

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL SUBSISTEMA DE INGRESO Y
SELECCIÓN DE CANDIDATOS A LAS ESCUELAS DE FORMACIÓN DEL
EJÉRCITO ECUATORIANO UTILIZANDO HERRAMIENTAS OPEN SOURCE**



Previo a la obtención del título de:

INGENIERÍA EN SISTEMAS E INFORMÁTICA

POR: EDWIN RICARDO MENESES BECERRA

SANGOLQUÍ, DICIEMBRE 2011

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. EDWIN RICARDO MENESES BECERRA, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, Diciembre de 2011.

ING. RAMIRO DELGADO
Director

DEDICATORIA

Dedico el presente trabajo a mi esposa Anita Torres, porque ha sido el pilar fundamental para obtener mi título profesional y es quien me da la fuerza necesaria para seguir adelante todos los días, a mi hija Dayanna y a mi hijo que esta por nacer porque por ellos trato de superarme cada día más y darles un mejor futuro.

Edwin Meneses

AGRADECIMIENTOS

Un especial agradecimiento a mi madre Cenia Becerra, por la gran dedicación y apoyo a todos sus hijos durante todo el transcurso de nuestra vida y habernos formado como personas de bien.

A los ingenieros Ramiro Delgado, Santiago Salvador por el gran apoyo durante la elaboración del presente proyecto, dando las pautas y consejos para poder culminarlo con éxito.

Al Ingeniero Mauricio Campana por ser una de las personas que me acompañó desde el principio de mi carrera y me ha apoyado en todo lo que he necesitado en materia académica.

Edwin Meneses

CARRERA DE INGENIERIA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Edwin Ricardo Meneses Becerra

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la institución del proyecto de grado titulado “ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL SUBSISTEMA DE INGRESO Y SELECCIÓN DE CANDIDATOS A LAS ESCUELAS DE FORMACIÓN DEL EJÉRCITO ECUATORIANO UTILIZANDO HERRAMIENTAS OPEN SOURCE” cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, Diciembre de 2011.

Edwin Ricardo Meneses Becerra

ÍNDICE DE CONTENIDOS

RESUMEN	1
CAPÍTULO I	3
1.1. INTRODUCCIÓN.....	3
1.2. JUSTIFICACIÓN	4
1.3. OBJETIVOS.....	5
1.3.1. OBJETIVO GENERAL.....	5
1.3.2. OBJETIVOS ESPECÍFICOS.....	5
1.4. ALCANCE	5
1.5. METODOLOGÍA.....	5
1.6. HERRAMIENTAS.....	6
CAPÍTULO II	8
2.1. MARCO TEÓRICO	8
2.1.1. DESCRIPCIÓN Y CONCEPTOS DE AUTOMATIZACIÓN.....	8
2.1.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE RUP	9
2.1.3. HERRAMIENTAS DE DESARROLLO	17
2.1.4. TÉCNICAS Y HERRAMIENTAS PARA ESPECIFICAR REQUERIMIENTOS.....	20
CAPÍTULO III	62
3.1. ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA	62
3.1.1. ESTRUCTURA ORGANIZACIONAL	62
3.1.2. DESCRIPCIÓN DE LOS PROCESOS	62
3.1.2.5. ACTIVIDADES DE CALIDAD:	63
3.1.2.5.3. PUBLICAR LAS LISTAS DEL PERSONAL PRESELECCIONADO	64
3.1.2.5.4. TOMAR CONOCIMIENTO DE LOS PLANES REMITIDOS POR LAS DIRECCIONES.....	64
3.1.3. ESPECIFICACIÓN DE REQUERIMIENTOS.....	66
CAPÍTULO IV.....	77
4.1. DESARROLLO DEL SUBSISTEMA DE INGRESO Y SELECCIÓN.....	77
4.1.1. ANÁLISIS Y DISEÑO DEL SISTEMA.....	77

4.1.1.1.	ARQUITECTURA DE SOFTWARE	77
4.1.1.2.	ESPECIFICACIÓN DE CASO DE USO	89
4.1.1.3.	JERARQUÍA MODULAR.....	109
4.1.1.4.	MODELO LÓGICO	111
4.1.2.	CONSTRUCCIÓN.....	121
4.1.2.1.	MANUAL USUARIO.....	121
CAPÍTULO V.....		131
CONCLUSIONES Y RECOMENDACIONES.....		131
5.1.	CONCLUSIONES.....	131
5.2.	RECOMENDACIONES	132
BIBLIOGRAFÍA		133

INDICE DE FIGURAS

Figura 2. 1 Iteraciones de las etapas de RUP	11
Figura 2. 2 Visión ampliada de las herramientas como una capa de la IS.	18
Figura 2. 3 Mejoras que provee el uso de HDS.....	19
Figura 2. 4 Clasificación de HDS desde diferentes perspectivas.	19
Figura 2. 5 Tópicos de las HDS, como sub área de conocimiento de la IS.....	20
Figura 2. 6 El empleado de ventas es un actor del sistema de ventas.....	30
Figura 2. 7 Cuando el actor es otro sistema se puede cambiar la notación	30
Figura 2. 8 Los casos de uso se representan con óvalos.....	31
Figura 2. 9 Una relación de extensión entre dos casos de uso	32
Figura 2. 10 Relaciones de Uso entre Casos de Uso.....	33
Figura 2. 11 El nombre de los actores abstractos suele no ser feliz.....	34
Figura 2. 12 Herencia de actores.....	34
Figura 3. 1 Estructura Orgánica de la Dirección de Recursos Humanos	62
Figura 4. 1 Vista de la Arquitectura desde el Punto de Vista del Usuario.....	81
Figura 4. 2 Vista de la Arquitectura desde el Punto de Lenguaje de Modelamiento Unificado	82
Figura 4. 3 Descripción de Paquetes Arquitectónicos.....	85
Figura 4. 4 Caso Uso Administración Documentos Requeridos.....	90
Figura 4. 5 Caso Uso Administración Deportes y Habilidades.....	94
Figura 4. 6 Caso Uso Administración Tipo Mención Honorífica.....	97
Figura 4. 7 Caso Uso Registro Documento.....	101
Figura 4. 8 Caso de Uso Reportes	107
Figura 4. 9 Modelo Lógico Inscripción de Candidatos.....	111
Figura 4. 10 Familiares de Candidatos.	112
Figura 4. 11 Parámetros de Selección.....	113
Figura 4. 12 Pruebas Psicológicas.	114
Figura 4. 13 Pruebas Académicas.....	115

Figura 4. 14 Pruebas Físicas.....	116
Figura 4. 15 Pruebas Médicas.....	117
Figura 4. 16 Ingreso al Portal Web	123
Figura 4. 17 Ingreso de Contraseña	124
Figura 4. 18 Selección del Sistema Ingreso y Selección	124
Figura 4. 19 Ingreso al Unidades Activas	125
Figura 4. 20 Ingreso al Menú de Sise.....	125
Figura 4. 21 Administración de Actividades	126
Figura 4. 22 Modificar Actividad	127
Figura 4. 23 Eliminar Actividad.....	127
Figura 4. 24 Limpiar Actividades	128
Figura 4. 25 Administración de Documentos Requeridos.....	128
Figura 4. 26 Parámetros de Selección	129
Figura 4. 27 Calendario Selección Fechas.....	130

RESUMEN

En la actualidad el Departamento de Obtención de la Dirección de Recursos Humanos no consta con un sistema informático que le permita realizar el proceso de Selección y Reclutamiento de una manera optima, en esta tesis se realizo el Análisis, Diseño y Construcción de un sistema que llene las expectativas del Departamento de Obtención y les ayude con los procesos de selección y reclutamiento. El desarrollo del sistema se realizó con herramientas open source con excepción de la base de datos que es Oracle, debido a que el Ejército Ecuatoriano tiene desarrollados todos sus sistemas con esta Base de Datos, Para elaborar este sistema se optó por una aplicación web desarrollada con Java, debido a que es necesario trabajar en diferentes repartos militares y la manera de interactuar más segura y eficiente es por medio del internet puesto que estos repartos militares no tiene conexión directa con la base de datos. Como servidor de aplicaciones Glassfish V2.0 debido a la integración con Netbeans y Java, este servidor de aplicaciones se encontrará administrado por la Dirección de Comunicaciones y Sistemas del Ejército Ecuatoriano y se utiliza instalado conjuntamente con la Base de datos Oracle en el cuarto de servidores de la Comandancia del Ejército.

ABSTRACT

Currently the department to obtain the direction of Human Resources has not a computer system that allows you to make the selection and recruitment process in an optimum way, this thesis is the analysis, design and construction of a complete system department's expectations and help them obtain the recruitment and selection processes. The system development was done with open source tools with the exception of the database is Oracle, because the Ecuadorian Army has developed all systems on this tool, to develop this system we opted for a web application developed with Java, because it is necessary to work in various divisions military and how to interact more safely and efficiently is through the internet as these military deals has no direct connection to the database. As Glassfish Application Server V2.0 use due to integration with Netbeans and Java, this application server is managed by the find of Communications and Systems Ecuadorian Army and was found installed in conjunction with the Oracle database in the room servants of the Army Command.

CAPÍTULO I

1.1. INTRODUCCIÓN

El Ejército Ecuatoriano cada año recluta cerca de 5000 candidatos para las Escuelas de Formación, de estos candidatos selecciona a los aspirantes que cumplan con los requisitos y aprueben las diferentes pruebas, por lo tanto; es indispensable desarrollar un sistema de información que permita administrar esta selección de una manera técnica y de forma eficiente que esté de acuerdo a las necesidades y requerimientos de la Institución.

El proceso de selección requiere de un sistema de información, que permita administrar la información relevante y llevar un control real en cada una de las etapas de selección que se realiza en un periodo determinado.

En el mercado nacional e internacional, se han comercializado sistemas de software, que pretenden satisfacer las necesidades del Ejército Ecuatoriano, pero estos no cubren todas las expectativas y requerimientos particulares que el proceso de selección tiene.

Se necesita elaborar un sistema que tome en cuenta otros factores personalizados que satisfagan las perspectivas del nivel gerencial y apoyen al nivel operativo, por lo que se ha visto indispensable la construcción de una herramienta informática, que permita estructurar un Sistema de Ingreso y Selección eficaz y eficiente.

Esta solución está orientada a proporcionar esa herramienta, utilizando métodos y técnicas de Ingeniería de Software que logren un producto de calidad, en funcionalidad, oportunidad y costo.

1.2. JUSTIFICACIÓN

En la actualidad el proceso de Ingreso y Selección del Ejército Ecuatoriano se lleva de una manera semiautomatizada, y en cada etapa del proceso se realiza de diferente forma así:

- El registro de la información de los candidatos se lo hace mediante un aplicativo web básico que guarda la información en la base de datos propia de cada Escuela de Formación.
- La verificación de la información física o carpetas con los documentos que deben presentar los candidatos se la realiza en una hoja de Excel.
- Las pruebas psicológicas tomadas a los candidatos son introducidas en un lector óptico y este a su vez genera una hoja Excel con la información generada para que esta pueda ser manipulada por los psicólogos.
- Las pruebas académicas se encuentran automatizadas pero sus registros son guardados en bases de datos individuales.
- Las pruebas médicas son registradas en hojas y luego ingresadas en el lector óptico y este posteriormente genera una hoja Excel para su manipulación por parte de los instructores, las pruebas médicas son manejadas en hojas Excel.
- Una vez recuperada esta información se consolida en una hoja electrónica parametrizando manualmente cada una de las pruebas, para que la junta de selección tome las decisiones correspondientes a los candidatos que posteriormente llegarán a ser aspirantes a oficiales o voluntarios del Ejército Ecuatoriano.

El proceso de Ingreso y Selección necesita de un sistema ágil y veraz, que permita a los miembros de la junta como a cada una de las personas involucradas en este proceso tomar decisiones de una manera correcta en el menor tiempo posible.

Para tener una seguridad optima y evitar contratiempos con la información se contará con el sistema de seguridades que está desarrollado para todos los sistemas que maneja el Ejército Ecuatoriano.

El manejo de la información debe ser centralizada en una sola base de datos que se encuentra en el edificio de la comandancia del Ejercito con sus debidas seguridades y planes de contingencia.

Por todo lo anteriormente citado, se hace indispensable realizar el Análisis, Diseño y Construcción de un Sistema de Información, que permita manejar el proceso de Ingreso y Selección a las escuelas de formación del Ejército, el mismo que deberá considerar metodologías y herramientas de ingeniería de software de calidad, a fin de conseguir un producto que satisfaga las expectativas y requerimientos del Ejército Ecuatoriano.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Analizar, Diseñar y Construir un Sistema de Información para el control del Proceso de Ingreso y Selección de los candidatos a las Escuelas de Formación del Ejército Ecuatoriano.

1.3.2. OBJETIVOS ESPECÍFICOS

1.3.2.1. Describir y conceptualizar los procesos automatizables de ingreso y selección.

1.3.2.2. Analizar las metodologías de desarrollo de software RUP.

- Rational Unified Process, debido a que el Ejército Ecuatoriano tiene como estándar la utilización de la metodología RUP.

1.4. ALCANCE

Desarrollar un Sistema para Control y Selección de candidatos a las Escuelas de Formación del Ejército Ecuatoriano, en ambiente Web que ofrezca a los usuarios soluciones para optimizar el proceso y permita a los mandos del ejército tomar decisiones acertadas para el manejo del personal entrante, este sistema va a satisfacer las necesidades del usuario final en cuanto a contenido, reportes, funcionalidad, tiempo y seguridad, a partir de un nuevo de Sistema Informático, utilizando una metodología estandarizada y herramientas de carácter open source que proporcionen estabilidad, escalabilidad y seguridad al Sistema.

La aplicación creada será instalada para pruebas, en la institución auspiciante.

1.5. METODOLOGÍA

El trabajo planteado exige inicialmente un estudio detallado de cada uno de los procesos de Ingreso y Selección, así también una Recopilación de información en cada uno de las tres Escuelas de Formación con las que cuenta el Ejército Ecuatoriano en este momento, para desarrollar reglas de negocios certeras y con validades.

Posteriormente se utilizará para el desarrollo del Sistema, las metodologías combinadas RUP.

1.5.1. PROCESO UNIFICADO RACIONAL (*Rational Unified Process*, RUP)

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, **y que además es el estándar que utiliza el Ejército Ecuatoriano.**

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

1.6. HERRAMIENTAS

Las Herramientas que se utilizan para el desarrollo del Sistema se detallan según las funciones para las que se utilizará cada una:

1.6.1. NETBEANS

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

1.6.2. JAVA

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Fue desarrollado por la compañía Sun Microsystems.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, entre otros. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema

que hace de puente entre el sistema operativo y el programa y posibilita que este último se entienda perfectamente.

1.6.3. GLASSFISH.

Es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia *TopLink*. GlassFish tiene como base al servidor *Sun Java System Application Server* de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

1.6.4. ORACLE

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

1.6.5. POWERDESIGNER 12.

Herramienta case intuitiva y estructurada, que permite el modelamiento de datos con UML, arquitectura, metadatos, etc., para el diseño y análisis del software de manera fácil y ágil, para que así las diferentes tecnologías de desarrollo estructuren soluciones para la arquitectura empresarial y de información.

CAPÍTULO II

2.1. MARCO TEÓRICO

2.1.1. DESCRIPCIÓN Y CONCEPTOS DE AUTOMATIZACIÓN DEL SUBSISTEMA DE INGRESO Y SELECCIÓN.

2.1.1.1. Receptar y verificar la documentación de los candidatos

Los Institutos de Formación, con sus delegados designados en las Divisiones, Brigadas y demás organismos militares, receptan y verifican que la documentación exigida a los ciudadanos este completa; luego se realiza la entrevista en dónde se les da indicaciones generales de la Institución, y se conoce las expectativas del ciudadano y su representante con respecto a la Institución.

2.1.1.2. ANALIZAR LA DOCUMENTACIÓN DE LAS CARPETAS

La Junta de Reclutamiento y Selección, previo conocimiento de las actividades constantes en el cronograma, se reúne para analizar la documentación presentada por los ciudadanos y proceder a la selección y aprobación de las mismas, luego remitir los listados del personal preseleccionado al Dpto. de Obtención para conocimiento.

2.1.1.3. PUBLICAR LAS LISTAS DEL PERSONAL PRESELECCIONADO

El Dpto. de Obtención en coordinación con los Institutos de Formación, publican en la página Web la lista del personal que ha sido preseleccionado para continuar en el proceso de selección, y con esta actividad finaliza el proceso de Reclutamiento.

2.1.1.3.1. APLICAR LOS EXÁMENES MÉDICOS

La Dirección de Sanidad, por intermedio de las Unidades de Salud Militar, aplicará los exámenes médicos a los ciudadanos candidatos a ingresar a los Institutos de Formación de acuerdo con lo planificado. Luego determina los resultados de la evaluación médica y una vez realizado el análisis por la Junta de Reclutamiento y Selección, remite al Dpto. de Obtención los listados de los ciudadanos idóneos y no idóneos, y en coordinación con los Institutos de Formación publicarán los resultados en la página Web.

2.1.1.3.2. APLICAR LOS EXÁMENES PSICOLÓGICOS

El Departamento de Salud Integral por intermedio de la Sección Psicosocial de la Dirección de Sanidad de la F.T., aplica los exámenes Psicológicos a los ciudadanos candidatos a ingresar a los Institutos de Formación, posteriormente remitirá al Dpto. de

Obtención los listados de los ciudadanos idóneos y no idóneos para continuar en el proceso.

2.1.1.3.3. APLICAR LOS EXÁMENES ACADÉMICOS

El Departamento de Planificación Educativa del Comando de Educación y Doctrina de la F.T. en coordinación con los Departamentos de Enseñanza de los Institutos de Formación (ESMIL, ESFORST y EIWIAS) y/o Unidades designadas, aplican los exámenes y remiten los resultados obtenidos al Dpto. de Obtención de la Dirección General de Recursos Humanos de la F.T., para la consolidación respectiva.

2.1.1.3.4. APLICAR LOS EXÁMENES FÍSICOS

La Sección de Cultura Física del Comando de Operaciones Terrestre en coordinación con las Secciones de Educación Física de los Institutos de Formación y/o Unidades designadas, aplican los exámenes físicos y remiten los resultados al Dpto. de Obtención, para la consolidación respectiva.

2.1.1.4. CONSOLIDAR LOS RESULTADOS DE EXÁMENES Y ENVIAR LOS REPORTES A LA JUNTA DE RECLUTAMIENTO Y SELECCIÓN

El Dpto. de Obtención en coordinación con la Junta de Reclutamiento y Selección, consolidan los resultados obtenidos en los diferentes exámenes aplicados (Médicos, Psicológicos, Académicos y Físicos).

2.1.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE RUP, PROCESO UNIFICADO RACIONAL, SUS FASES Y CARACTERÍSTICAS.

El Proceso Unificado Racional, Rational Unified Process en inglés, y sus siglas RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí.

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

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

RUP se divide en 4 fases, dentro de las cuales se realizan varias iteraciones según el proyecto y en las que se hace mayor o menos esfuerzo en las distintas actividades.

En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades:

2.1.2.1. FASE DE INICIO (Inspección y Concepción).

Se hace un plan de fases, donde se identifican los principales casos de uso y se identifican los riesgos. Se concreta la idea, la visión del producto y como se enmarca en el negocio así como se determina el alcance del proyecto.

2.1.2.2. FASE DE ELABORACIÓN.

Se realiza el plan de proyecto, donde se completan los casos de uso y se mitigan los riesgos. Se planifica las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura.

2.1.2.3. FASE DE CONSTRUCCIÓN.

Se basa en la elaboración de un producto totalmente operativo y en la elaboración del manual de usuario. Construir el producto, la arquitectura y los planes, hasta que esté listo para ser enviado a la comunidad de usuarios.

2.1.2.4. FASE DE TRANSICIÓN.

Se realiza la instalación del producto en el cliente y se procede al entrenamiento de los usuarios. Se realiza el entrenamiento en el uso del producto a los usuarios, lo cual incluye: construcción, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente quede satisfecho, por tanto en esta fase suelen ocurrir cambios.

Con estas fases se logra ejecutar el conjunto de mejores prácticas que se listan a continuación:

- Desarrollar Software Iterativamente
- Modelar el software visualmente
- Gerenciar los Requerimientos
- Usar arquitecturas basadas en componentes
- Verificación continua de la calidad

- Gerenciar los cambios

La **Figura 2.1** muestra las interacciones en las etapas de RUP.

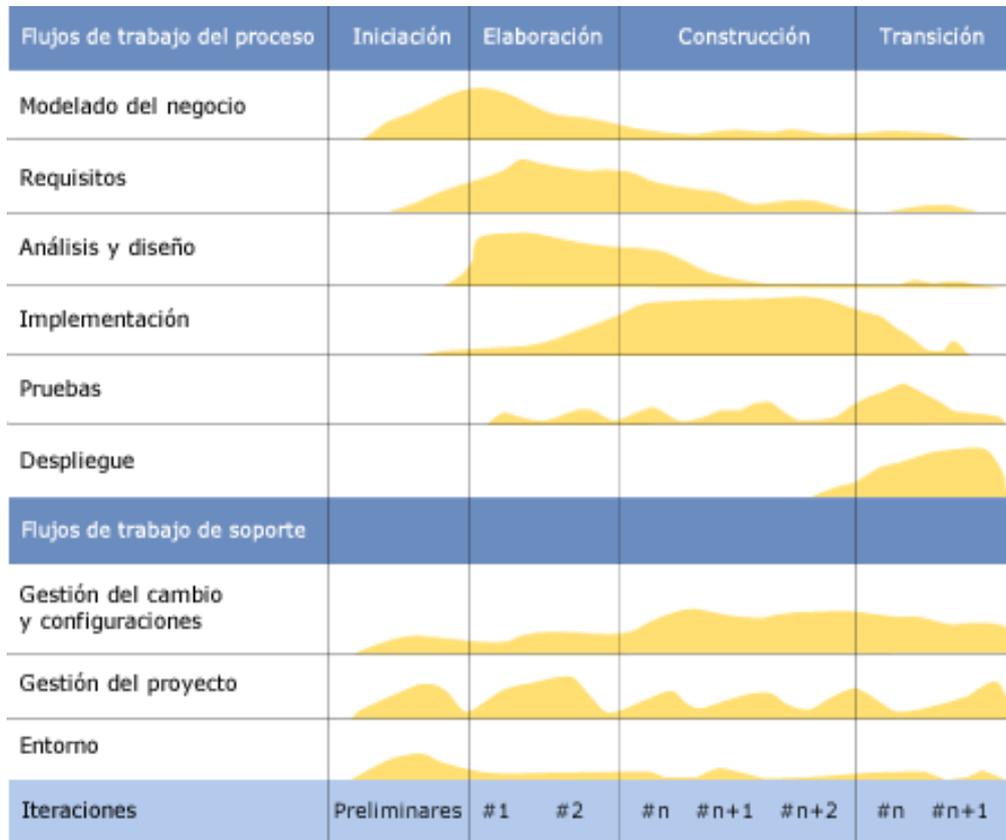


Figura 2. 1 Iteraciones de las etapas de RUP

2.1.2.5. DESCRIPCIÓN DE LAS FASES

Dependiendo de la iteración el equipo de desarrollo puede realizar diferentes tipos de actividades. Veamos de qué trata cada fase.

2.1.2.5.1. FASE DE INICIO

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

En esta fase se elaboran los siguientes entregables.

- Un documento con la visión del proyecto.
- El modelo de Casos de Uso con una lista de todos los Casos de Uso y los actores que puedan ser identificados.
- Un glosario inicial del proyecto.
- Un Caso de Uso inicial de Negocio el cual incluye: contexto del negocio, criterios de éxito y planificación financiera.

- Un estudio inicial de riesgos.
- Un plan del proyecto que muestre las fases y las iteraciones.

El objetivo de esta fase, es definir el modelo de negocio que permita entender las funciones de la organización y del cliente, tanto en estructura como en sus procesos. En esta fase se modela las funciones y roles que realiza la organización para optimizarla mediante una reingeniería de procesos y la implantación del nuevo sistema. También se describe lo que el sistema tendrá que realizar y permitir que los desarrolladores y el cliente estén de acuerdo con esta descripción.

Para ello se realizarán las siguientes subfases:

- Describir los requerimientos funcionales y no funcionales (rendimiento esperado, plataformas soportadas, integración con sistemas externos, etc.).
- Capturar un glosario o vocabulario del sistema o proyecto (mediante documento y clases conceptuales).
- Encontrar actores y casos de uso.
- Describir los casos de uso mediante su flujo principal, variaciones y excepciones.
- Asignar prioridades a los casos de uso encontrados para poder planificar la iteración en forma de análisis, diseño e implementación.
- Modelar la interfaz de usuario (diseño lógico).
- Desarrollar un prototipo de la interfaz de usuario (diseño físico).

2.1.2.5.2. FASE DE ELABORACIÓN

En esta fase las iteraciones se orientan al desarrollo de la arquitectura, que incluye los flujos de trabajo de requerimientos, modelo de negocio (refinamiento), análisis, diseño y una parte de implementación orientado a la arquitectura.

En esta fase se elabora:

- Un modelo de Casos de Uso con todos los actores identificados y la mayor parte de las descripciones de estos.
- Requerimientos adicionales: no funcionales o pseudo requerimientos.
- Descripción de la arquitectura del software.
- Prototipo ejecutable de arquitectura.
- Una lista revisada de riesgos.
- Plan del proyecto, incluyendo iteraciones y criterios de evaluación para cada iteración.
- Manual preliminar de usuario.

En esta fase se especifican los requerimientos y se describe cómo se van a implementar en el sistema, esto es, transformar los requisitos al diseño del sistema, se desarrolla la arquitectura para el sistema, y se adapta el diseño para que sea consistente con el entorno de implementación.

2.1.2.5.3. FASE DE CONSTRUCCIÓN

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

Para ello se realiza lo siguiente:

- Se integra el producto de software sobre la plataforma adecuada.
- Se elaboran los manuales de usuario.
- Se dé una descripción de la versión actual.
- Se planifica 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 se encuentra errores de diseño, se notifica.
- Se integra el sistema siguiendo el plan.

En la fase de Pruebas se evalúa la calidad del producto, pero no para aceptar o rechazar el mismo al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Se debe encontrar y documentar defectos en la calidad del software. Generalmente proporciona datos sobre la calidad del software, 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 y que los requisitos tengan su apropiada implementación.

En la parte de despliegue se produce distribuciones del producto y se entrega 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.

Durante todo el proyecto se ejecutan las fases de gestión del proyecto, donde se vigila el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

En esta etapa se realizan las siguientes tareas:

- 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 fase de configuración y control de cambios, permite mantener la integridad de todas las actividades que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

En la fase de Entorno, la finalidad 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.

2.1.2.6. ROLES QUE SE CUMPLEN EN EL RUP.

2.1.2.6.1. ANALISTAS

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

2.1.2.6.2. DESARROLLADORES

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

2.1.2.6.3. 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.

2.1.2.6.4. APOYO

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

2.1.2.6.5. ESPECIALISTA EN PRUEBAS

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

2.1.2.6.6. OTROS ROLES

- Stakeholders.
- Revisor
- Coordinación de revisiones

- Revisor técnico
- Cualquier rol
- Para grandes organizaciones con un número de equipos de ingenieros y la comunicación entre cada equipo es crítica por lo tanto es necesario que los artefactos sean completos y bastante comprensivos
- En tanto que para pequeños proyectos no es recomendable presentarse tanto rigor en las preparaciones de los artefactos, la eficiencia del proceso depende más de las habilidades de cada trabajador.

2.1.2.6.7. BENEFICIOS DE LA METODOLOGÍA ORIENTADA A OBJETOS.

- Promueve la reusabilidad.
- Reduce la complejidad del mantenimiento (extensibilidad y facilidad de cambios).
- Riqueza semántica.
- Disminuye la brecha semántica entre la visión interna y la visión externa del sistema.
- Facilita la construcción de prototipos.

2.1.2.6.8. VENTAJAS DE LA METODOLOGÍA ORIENTADA A OBJETOS.

- Reutilización
- El diseñador piensa en términos del comportamiento de objetos y no en detalles de bajo nivel
- Confiabilidad, Integridad y Estabilidad.
- Mantenimiento más sencillo. Modificaciones locales.
- Modelado más realista.
- Modelos empresariales inteligentes.
- Independencia del diseño.
- Mejores herramientas CASE.
- Bibliotecas de clases para las empresas.
- Se construyen clases cada vez más complejas.
- Nuevos mercados para el software.
- Diseño de mayor calidad.
- Programación más sencilla.
- Mejor comunicación entre los profesionales de los Sistemas de Información y los empresarios.
- Mayor nivel de automatización de las bases de datos.

- La comprensión del sistema es más fácil porque la semántica entre el sistema y la realidad son similares.

2.1.2.7. JUSTIFICACIÓN DE LA METODOLOGÍA RUP.

En el Desarrollo de un Sistema de Información para llevar el Registro y Control de las Víctimas y Afectados por Accidentes de Tránsito y las Denuncias sobre el estado de la Infraestructura Vial para la Asociación Civil ASOTRANSITO, se utilizara la metodología RUP. Esta metodología se encuentra dividida en 4 fases: fase de inicio, fase de elaboración, fase de construcción y fase de transiciones. En la fase de inicio identificaremos los principales caso de uso, los riesgos y el alcance del proyecto. En la fase de elaboración se realizara el plan de proyecto, competición de los casos de uso y se mitigan riesgos. En la fase de construcción se concreta la elaboración del sistema totalmente operativo y eficiente, y la realización del manual de usuarios, en la fase de transición se entregará el sistema al usuario haciendo la respectiva instalación del mismo con el fin que el usuario pueda realizar las pruebas correspondientes.

2.1.3. HERRAMIENTAS DE DESARROLLO.

2.1.3.1. RESUMEN

Las herramientas para el desarrollo de software (HDS) desempeñan un importante papel en el desarrollo de aplicaciones. Como parte de la ingeniería de software (IS), las HDS han experimentado cambios en los últimos años. Aún ante la existencia de numerosas herramientas, es necesario abordar temas genéricos sobre las HDS, identificar tópicos relevantes y sus relaciones para facilitar su comprensión y análisis. Esta investigación en progreso, establece la construcción de una ontología que describe conceptos y relaciones para la investigación en HDS. Parte de las HDS en el contexto de la IS: las disciplinas, las áreas de conocimiento y las herramientas como una capa de la IS, para dar lugar a una exploración mayor sobre las definiciones de HDS, las características que definen sus beneficios y las taxonomías disponibles. Ello procura establecer conexiones y motivar una visión de conjunto para su análisis. Palabras clave: herramientas, ingeniería de software, ontología, desarrollo de software.

2.1.3.2. HERRAMIENTAS DE DESARROLLO DE SOFTWARE

Actualmente se considera a las HDS como herramientas basadas en computadoras que asisten el proceso de ciclo de vida de software, consolidadas en la literatura en la forma de Ingeniería de software asistida por computadora (CASE, por sus siglas en

inglés). Esto es, software que se utiliza para ayudar a las actividades del proceso de software o software que es utilizado para diseñar y para implementar otro software. Permiten automatizar acciones bien definidas, reduciendo también la carga cognitiva del ingeniero de software, quien requiere libertad para concentrarse en los aspectos creativos del proceso. Este soporte se traduce en mejoras a la calidad y la productividad en el diseño y desarrollo. Las HDS automatizan metodologías de software y desarrollo de sistemas y se vinculan con los diferentes conceptos involucrados en el desarrollo (Figura 2.2).

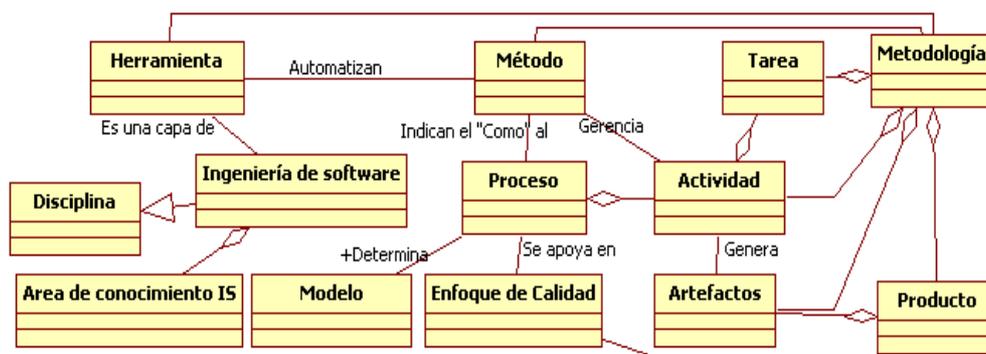


Figura 2. 2 Visión ampliada de las herramientas como una capa de la IS.

2.1.3.3. MEJORAS QUE PROVEE EL USO DE HDS

El soporte que brindan las HDS al proceso de desarrollo proporciona importantes ventajas para el equipo de trabajo de IS (Figura 2.3). Estas mejoras se sintetizan en:

- a. Apoyar a las metodologías y métodos, integrando actividades y propiciando visión de continuidad entre fases metodológicas.
- b. Mejorar la comunicación entre los actores involucrados, facilitándoles compartir su trabajo y desempeñarlo de forma dinámica e iterativa.
- c. Establecer métodos efectivos para almacenar y utilizar los datos, lo que permite organizar y correlacionar componentes, para accederlos a través de un repositorio.
- d. Agregar eficiencia al mantenimiento, ya que los programas son construidos sobre las mismas estructuras y estándares, facilitando la adherencia a la disciplina de diseño y facilitan también la conversión automática de programas a versiones más recientes de lenguajes de programación.
- e. Automatizar porciones del análisis y diseño tediosos y propensos a error, con influencia sobre la generación de código, las pruebas y el

control. Resalta la consideración de que los beneficios potenciales sólo pueden ser alcanzados si las HDS son utilizadas de forma correcta.

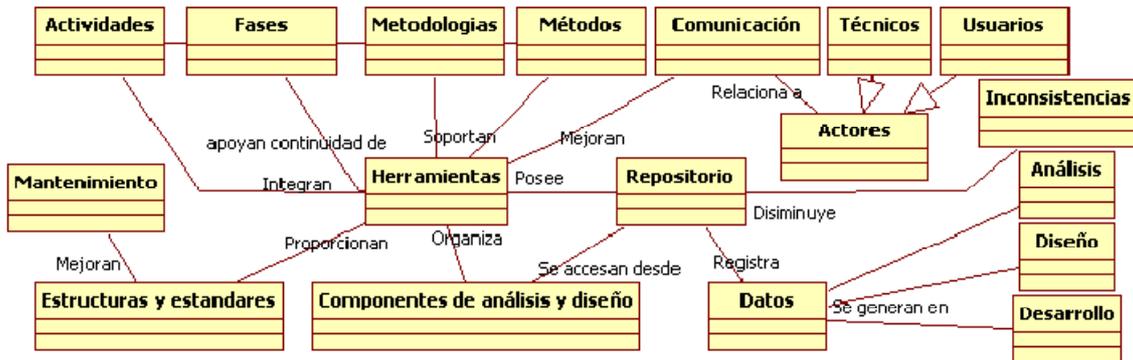


Figura 2. 3 Mejoras que provee el uso de HDS.

2.1.3.4. CLASIFICACIONES DE LAS HERRAMIENTAS

Establecer una clasificación resulta importante para comprender los tipos de herramientas disponibles y su soporte al proceso del software. En adelante se exponen las principales perspectivas, o formas de clasificación de las HDS, disponibles en la literatura.

En la práctica, los límites entre estas clasificaciones son difusos, lo cual contribuye al argumento de que no es fácil ubicar una herramienta utilizando una clasificación particular. En la Figura 2.4 se representan estas tres perspectivas, de una manera bastante general (razón por la cual las HDS funcionales no se describen en detalle).

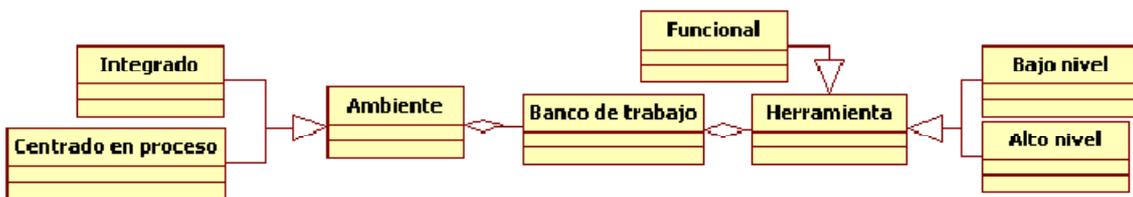


Figura 2. 4 Clasificación de HDS desde diferentes perspectivas.

2.1.3.4.1. DE ACUERDO CON LAS FUNCIONES QUE LA HERRAMIENTA SOPORTA.

En esta perspectiva, las HDS se clasifican según su función específica dentro del proceso de desarrollo.

En la literatura se identifican hasta 22 tipos de herramientas clasificadas funcionalmente. Estas clasificaciones tienen una intención orientadora, apuntan a mostrar la amplitud de las herramientas y sus aportes a distintas etapas del desarrollo, más que a establecer taxonomías rigurosas.

2.1.3.4.2. DE ACUERDO CON EL ACTOR QUE ES SOPORTADO POR LA HERRAMIENTA.

Esta perspectiva se basa en el sujeto a quien soporta la herramienta, y las define como:

- a. De Alto nivel, que ayudan principalmente a los analistas y diseñadores, permitiéndoles crear y modificar el sistema.
- b. De Bajo nivel, utilizadas por programadores y trabajadores quienes deben implementar los sistemas diseñados.

2.1.3.4.3. DE ACUERDO CON EL PROCESO.

En esta clasificación se definen tres categorías:

- a. Las herramientas, que ayudan a tareas puntuales y se utilizan a discreción del ingeniero de software.
- b. Los bancos de trabajo, que agrupan herramientas que mantienen algún grado de integración y soportan a un método que incluye un modelo del proceso.
- c. Los ambientes, que apoyan a los procesos de software, incluyen bancos de trabajo integrados y soportan a los datos, al control y a la integración.

2.1.3.4.4. TÓPICOS DE LAS HERRAMIENTAS, COMO SUB ÁREA DE CONOCIMIENTO DE LA IS.

En esta clasificación existe un conjunto de herramientas dentro de cada tópico, cuyo campo de acción se asocia no sólo con las actividades propias de cada uno de estos, sino también se considera el caso de herramientas que prestan soporte con un carácter transversal, como el caso de las herramientas misceláneas (Figura 2.5).

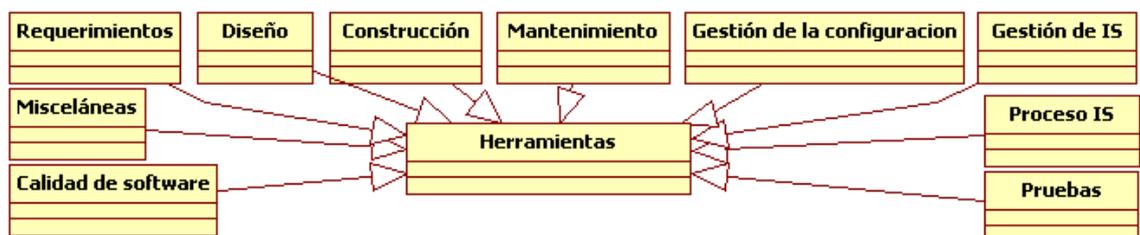


Figura 2. 5 Tópicos de las HDS, como sub área de conocimiento de la IS.

2.1.4. TÉCNICAS Y HERRAMIENTAS PARA ESPECIFICAR REQUERIMIENTOS.

2.1.4.1. INTRODUCCIÓN

En la actualidad, son muchos los procesos de desarrollo de software que existen. Con el pasar de los años, la Ingeniería de Software ha introducido y

popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso de desarrollo en sí. Se han publicado muchos libros y artículos relacionados con este tema, con el modelado de procesos del negocio y la reingeniería. Un número creciente de herramientas automatizadas han surgido para ayudar a definir y aplicar un proceso de desarrollo de software efectivo. Hoy en día la economía global depende más de sistemas automatizados que en épocas pasadas; esto ha llevado a los equipos de desarrollo a enfrentarse con una nueva década de procesos y estándares de calidad.

Sin embargo, ¿cómo explicamos la alta incidencia de fallos en los proyectos de software? ¿Por qué existen tantos proyectos de software víctimas de retrasos, presupuestos sobregirados y con problemas de calidad? ¿Cómo podemos tener una producción o una economía de calidad, cuando nuestras actividades diarias dependen de la calidad del sistema?

Tal vez suene ilógico pero, a pesar de los avances que ha dado la tecnología, aún existen procesos de producción informales, parciales y en algunos casos no confiables.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

La razón principal para escoger este tema se fundamentó en la gran cantidad de proyectos de software que no llegan a cumplir sus objetivos. En nuestro país somos partícipes de este problema a diario, en donde se ha vuelto común la compra de sistemas extranjeros, para luego "personalizarlos" supuestamente a la medida de las empresas.

Tal "personalización", la mayoría de las veces, termina retrasando el proyecto en meses, o incluso en años. La problemática del año 2000 trajo como consecuencia una serie de cambios apresurados en los sistemas existentes; cambios que, desde mi punto de vista, no fueron bien planificados.

El reemplazo de plataformas y tecnologías obsoletas, la compra de sistemas completamente nuevos, las modificaciones de todos o de casi todos los programas que forman un sistema, entre otras razones, llevan a desarrollar proyectos en calendarios sumamente ajustados y en algunos casos irreales; esto ocasiona que se omitan muchos pasos importantes en el ciclo de vida de desarrollo, entre estos, la definición de los requerimientos.

Estudios realizados muestran que más del 53% de los proyectos de software fracasan por no realizar un estudio previo de requisitos. Otros factores como falta de participación del usuario, requerimientos incompletos y el cambio a los requerimientos, también ocupan sitios altos en los motivos de fracasos.

2.1.4.2. TÉCNICAS UTILIZADAS EN LA ACTIVIDADES DE INGENIERIA DE REQUERIMIENTOS

Existen varias técnicas para la Ingeniería de Requerimientos (IR), sin embargo, en este documento se van a estudiar sólo algunas de ellas. Cada técnica puede aplicarse en una o más actividades de la IR; en la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose.

2.1.4.2.1. ENTREVISTAS Y CUESTIONARIOS

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema.

Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él.

Las preguntas que deben realizarse en esta técnica, deben ser preguntas de alto nivel y abstractas que pueden realizarse al inicio del proyecto para obtener información sobre aspectos globales del problema del usuario y soluciones potenciales.

Con frecuencia, se utilizan preguntas abiertas para descubrir sentimientos, opiniones y experiencias generales, o para explorar un proceso o problema. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado y no están influenciadas por el conocimiento de la solución.

Las preguntas pueden ser enfocadas a un elemento del sistema, tales como usuarios, procesos, etc. El siguiente ejemplo muestra algunos tipos de preguntas abiertas.

Del Usuario

- ¿Quién es el cliente?
- ¿Quién es el usuario?
- ¿Son sus necesidades diferentes?
- ¿Cuáles son sus habilidades, capacidades, ambiente?

Del Proceso

- ¿Cuál es la razón por la que se quiere resolver este problema?
- ¿Cuál es el valor de una solución exitosa?
- ¿Cómo usted resuelve el problema actualmente?
- ¿Qué retrasos ocurren o pueden ocurrir?

Del Producto

- ¿Qué problemas podría causar este producto en el negocio?
- ¿En qué ambiente se usará el producto?
- ¿Cuáles son sus expectativas para los conceptos fácil de usar, confiable, rendimiento?
- ¿Qué obstáculos afectan la eficiencia del sistema?

El éxito de esta técnica combinada, depende de la habilidad del entrevistador y de su preparación para la misma. Los analistas necesitan ser sensibles las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales. Asimismo, necesitan considerar no sólo la información que adquieren a través del cuestionario y la entrevista, sino también, su significancia.

2.1.4.3. LLUVIA DE IDEAS (BRAINSTORM)

Este método comenzó en el ámbito de las empresas, aplicándose a temas tan variados como la productividad, la necesidad de encontrar nuevas ideas y soluciones para los productos del mercado, encontrar nuevos métodos que desarrollen el pensamiento creativo a todos los niveles, etc. Pero pronto se extendió a otros ámbitos, incluyendo el mundo de desarrollo de sistemas; básicamente se busca que los involucrados en un proyecto desarrollen su creatividad, promoviendo la introducción de los principios creativos.

A esta técnica se le conoce también como torbellino de ideas, tormenta de ideas, desencadenamiento de ideas, movilización verbal, bombardeo de ideas, sacudidas de cerebros, promoción de ideas, tormenta cerebral, avalancha de ideas, tempestad en el cerebro y tempestad de ideas, entre otras.

2.1.4.3.1. PRINCIPIOS DE LA LLUVIA DE IDEAS

- Aplazar el juicio y no realizar críticas, hasta que no agoten las ideas, ya que actuaría como un inhibidor. Se ha de crear una atmósfera de trabajo en la que nadie se sienta amenazado.
- Cuantas más ideas se sugieren, mejores resultados se conseguirán: "la cantidad produce la calidad". Las mejores ideas aparecen tarde en el periodo de producción de ideas, será más fácil que encontremos las soluciones y tendremos más variedad sobre la que elegir.
- La producción de ideas en grupos puede ser más efectiva que la individual.
- Tampoco debemos olvidar que durante las sesiones, las ideas de una persona, serán asociadas de manera distinta por cada miembro, y hará que aparezcan otras por contacto.

El equipo en una lluvia de ideas debe estar formado por:

- **EL DIRECTOR**

Es la figura principal y el encargado de dirigir la sesión. Debe ser un experto en pensamiento creador. Su función es formular claramente el problema y que todos se familiaricen con él. Cuando lo haga, debe estimular ideas y hacer que se rompa el hielo en el grupo. Es el encargado de que se cumplan las normas, no permitiendo las críticas. Debe permanecer callado e intervenir cuando se corte la afluencia de ideas, por lo que le será útil llevar ya un listado de ideas. Debe hacer que todos participen y den ideas. Además, es la persona que da concede la palabra y da por finalizada la sesión. Posteriormente, clasificará las ideas de la lista que le proporciona el secretario.

- **EL SECRETARIO**

Registra por escrito las ideas según van surgiendo. Las enumera, las reproduce fielmente, las redacta y se asegura que todos están de acuerdo con lo escrito. Por último realizará una lista de ideas.

- **LOS PARTICIPANTES**

Pueden ser habituales o invitados; cualquier involucrado en el proyecto entra en esta categoría. Su función es producir ideas. Conviene que entre ellos no haya diferencias jerárquicas.

Las personas que componen el grupo deben estar motivadas para solucionar el problema, y con un ambiente que propicie la participación de todos. Todos pueden sentirse confiados y con la sensación de que pueden hablar sin que se produzcan críticas. Todas las ideas en principio deben tener el mismo valor, pues cualquiera de ellas puede ser la clave para la solución. Es necesario prestar mucha atención a las frases que pueden coartar la producción de ideas. Además durante la celebración no deben asistir espectadores.

Se debe evitar todos los bloqueos que paralizan la ideación: como son hábitos o ideas preconcebidas, el desánimo o falta de confianza en sí mismo, el temor y la timidez.

2.1.4.3.2. FASES DE APLICACIÓN EN EL BRAINSTORM

a) DESCUBRIR HECHOS

Al menos con un día de antelación, el director comunica por escrito a los miembros del grupo sobre los temas a tratar. El director explica los principios de la Tormenta de ideas e insiste en la importancia de tenerlos en cuenta. La sesión comienza con una ambientación de unos 10 minutos, tratando un tema sencillo y no comprometido. Es una fase especialmente importante para los miembros sin experiencia. Se determina el problema, delimitándolo, precisándolo y clarificándolo. A continuación se plantea el problema, recogiendo las experiencias que se poseen o consultando documentación. Cuando es complejo, conviene dividirlo en partes. Aquí es importante la utilización del análisis, desmenuzando el problema en pequeñas partes para conectar lo nuevo y lo desconocido.

b) PRODUCIR IDEAS (es la fase de tormenta de ideas propiamente dicha)

Se van aplicando alternativas. Se busca producir una gran cantidad de ideas, aplicando los principios que hemos visto. Además, es útil cuando se ha trabajado mucho, alejarse del problema, pues es un buen momento para que se produzcan asociaciones. Muchas de las nuevas ideas serán ideas antiguas, mejoradas o combinadas con varias ya conocidas.

Al final de la reunión, el director da las gracias a los asistentes y les ruega que no abandonen el problema, ya que al día siguiente se le pedirá una lista de ideas que les puedan haber surgido. Se incorporan las ideas surgidas después de la reunión.

c) DESCUBRIR SOLUCIONES

Se elabora una lista definitiva de ideas, para seleccionar las más interesantes. La selección se realiza desechando las ideas que no tienen valor y se estudia si son válidas las que se consideran interesantes. Lo mejor es establecer una lista de criterios de conveniencia para cada idea. Se seleccionan las ideas más útiles y si es necesario se ponderarán. Pueden realizarlo los mismos miembros del grupo o crear otros para esta tarea; la clasificación debe hacerse por categorías (tarea que corresponde al director). Se presentan las ideas de forma atractiva, haciendo uso de soportes visuales.

d) PROTOTIPOS

Los prototipos permiten al desarrollador crear un modelo del software que debe ser construido.

Al igual que todos los enfoques al proceso de desarrollo del software, el prototipado comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un "diseño rápido". El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo. El prototipo es evaluado por el cliente y el usuario y utilizado para refinar los requerimientos del software a ser desarrollado. Un proceso de iteración tiene lugar a medida que el prototipo es "puesto a punto" para satisfacer las necesidades del cliente y permitiendo al mismo tiempo una mejor comprensión del problema por parte del desarrollador.

Existen principalmente dos tipos de prototipos:

i. PROTOTIPO RÁPIDO (concept prototipe)

El prototipado rápido es un mecanismo para lograr la validación pre-compromiso. Se utiliza para validar requerimientos en una etapa previa

al diseño específico. En este sentido, el prototipo puede ser visto como una aceptación tácita de que los requerimientos no son totalmente conocidos o entendidos antes del diseño y la implementación. El prototipo rápido puede ser usado como un medio para explorar nuevos requerimientos y así ayudar a "controlar" su constante evolución.

ii. **PROTOTIPO EVOLUTIVO**

Desde una perspectiva diferente, todo el ciclo de vida de un producto puede ser visto como una serie incremental de detallados prototipos acumulativos. Tradicionalmente, el ciclo de vida está dividido en dos fases distintas: desarrollo y mantenimiento. La experiencia ha demostrado que esta distinción es arbitraria y va en contra de la realidad ya que la mayor parte del costo del software ocurre después de que el producto se ha entregado. El punto de vista evolutivo del ciclo de vida del software considera a la primera entrega como un prototipo inicial en el campo. Modificaciones y mejoras subsecuentes resultan en nuevas entregas de prototipos más maduros. Este proceso continúa hasta que se haya desarrollado el producto final. La adopción de esta óptica elimina la distinción arbitraria entre desarrollo y mantenimiento, resultando en un importante cambio de mentalidad que afecta las estrategias para la estimación de costos, enfoques de desarrollo y adquisición de productos.

2.1.4.4. ADMINISTRACIÓN DE REQUERIMIENTOS CON CASOS DE USO

2.1.4.4.1. ¿Qué son los Casos de Uso?

Los casos de uso son una técnica para especificar el comportamiento de un sistema: "Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios."

Todo sistema de software ofrece a su entorno una serie de servicios. Un caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa. Cuando decimos "alguien o algo" hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software.

Por ejemplo, un sistema de ventas, si pretende tener éxito, debe ofrecer un servicio para ingresar un nuevo pedido de un cliente. Cuando un usuario accede a este servicio, podemos decir que está "ejecutando" el caso de uso ingresando pedido.

Los Casos de Uso fueron introducidos por Jacobson en 1992 [Jacobson92]. Sin embargo, la idea de especificar un sistema a partir de su interacción con el entorno es original de McMenamin y Palmer, dos precursores del análisis estructurado, que en 1984 definieron un concepto muy parecido al del caso de uso: el evento. Para McMenamin y Palmer, un evento es algo que ocurre fuera de los límites del sistema, ante lo cual el sistema debe responder. Siguiendo con nuestro ejemplo anterior, nuestro sistema de Ingreso y Selección tendrá un evento "Aspirante". En este caso el sistema deberá responder al estímulo que recibe.

Sin embargo, existen algunas diferencias entre los casos de uso y los eventos. Las principales son:

- Los eventos se centran en describir qué hace el sistema cuando el evento ocurre, mientras que los casos de uso se centran en describir cómo es el diálogo entre el usuario y el sistema.
- Los eventos son "atómicos": se recibe una entrada, se la procesa, y se genera una salida, mientras que los casos de uso se prolongan a lo largo del tiempo mientras dure la interacción del usuario con el sistema. De esta forma, un caso de uso puede agrupar a varios eventos.
- Para los eventos, lo importante es qué datos ingresan al sistema o salen de él cuando ocurre el evento (estos datos se llaman datos esenciales), mientras que para los casos de uso la importancia del detalle sobre la información que se intercambia es secundaria. Según esta técnica, ya habrá tiempo más adelante en el desarrollo del sistema para ocuparse de este tema.

Los casos de uso combinan el concepto de evento del análisis estructurado con otra técnica de especificación de requerimientos bastante poco difundida, aquella que dice que una buena forma de expresar los requerimientos de un sistema es escribir su manual de usuario antes de construirlo. Esta técnica, si bien ganó pocos adeptos, se basa en un concepto muy interesante, al definir requerimientos, es importante describir al sistema desde el punto de vista de aquél que lo va a usar, y no desde el punto de vista del que lo va a construir. De esta forma, es más fácil validar que los requerimientos documentados son los verdaderos requerimientos de los usuarios, ya que éstos comprenderán fácilmente la forma en la que están expresados.

2.1.4.4.2. LOS CASOS DE USO Y UML

A partir de la publicación del libro de Jacobson, gran parte de los más reconocidos especialistas en métodos Orientados a Objetos coincidieron en considerar

a los casos de uso como una excelente forma de especificar el comportamiento externo de un sistema. De esta forma, la notación de los casos de uso fue incorporada al lenguaje estándar de modelado UML (Unified Modeling Language) propuesto por Ivar Jacobson, James Rumbaugh y Grady Booch, tres de los precursores de las metodologías de Análisis y Diseño Orientado a Objetos, y avalado por las principales empresas que desarrollan software en el mundo. UML va en camino de convertirse en un estándar para modelado de sistemas de software de amplia difusión.

A pesar de ser considerada una técnica de Análisis Orientado a Objetos, es importante destacar que los casos de uso poco tienen que ver con entender a un sistema como un conjunto de objetos que interactúan, que es la premisa básica del análisis orientado a objetos "clásico". En este sentido, el éxito de los casos de uso no hace más que dar la razón al análisis estructurado, que propone que la mejor forma de empezar a entender un sistema es a partir de los servicios o funciones que ofrece a su entorno, independientemente de los objetos que interactúan dentro del sistema para proveerlos. Como era de esperar, es probable que en el futuro los métodos de análisis y diseño que prevalezcan hayan adoptado las principales ventajas de todos los métodos disponibles en la actualidad (estructurados, métodos formales, métodos orientados a objetos, etc.).

2.1.4.4.3. DEFINICIONES Y NOTACIONES

a) ACTORES

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que estamos construyendo de la misma forma. Por ejemplo, para nuestro sistema de ingreso y selección las personas que quieren registrarse para ingresar al curso de aspirantes a oficiales o voluntarios, si pueden hacer las mismas cosas con el sistema, son considerados un único actor: "Usuarios externos".

Los actores son externos al sistema que vamos a desarrollar. Por lo tanto, al identificar actores estamos empezando a delimitar el sistema, y a definir su alcance. Definir el alcance del sistema debe ser el primer objetivo de todo analista, ya que un proyecto sin alcance definido nunca podrá alcanzar sus objetivos.

Es importante tener clara la diferencia entre usuario y actor. Un actor es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol. De esta forma, un usuario puede acceder al sistema

como distintos actores. La forma más simple de entender esto es pensar en perfiles de usuario de un sistema operativo. Una misma persona puede acceder al sistema con distintos perfiles, que le permiten hacer cosas distintas. Los perfiles son en este caso equivalentes a los actores.

Los actores se representan con dibujos simplificados de personas, llamados en inglés "stick man" (hombres de palo). Como muestra la Figura 2.6.

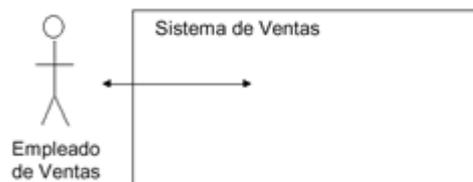


Figura 2. 6 El empleado de ventas es un actor del sistema de ventas

Si bien en UML los actores siempre se representan con "hombres de palo", a veces resulta útil representar a otros sistemas con alguna representación más clara. Como muestra la Figura 2.7.

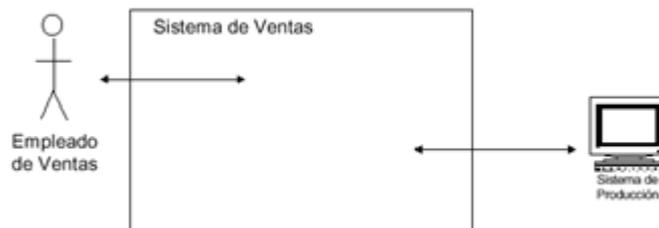


Figura 2. 7 Cuando el actor es otro sistema se puede cambiar la notación

Las flechas, que existían en la propuesta original de Jacobson, pero que desaparecieron del modelo semántico de UML, puede usarse para indicar el flujo de información entre el sistema y el actor. Si la flecha apunta desde el actor hacia el sistema, esto indica que el actor está ingresando información en el sistema. Si la flecha apunta desde el sistema hacia el actor, el sistema está generando información para el actor.

Identificar a los actores es el primer paso para usar la técnica de casos de uso. Por ejemplo, en el sistema de ingreso y selección, sin conocer prácticamente ningún detalle sobre cómo funcionará, se puede decir que:

- El grupo de usuarios que ingrese sus datos al sistema será un actor.

- El grupo de usuarios que haga otras operaciones con los datos registrados, como por ejemplo verificar carpetas, cancelarlos y modificar sus plazos de entrega, será un actor.
- Todo grupo de usuarios que reciba ciertos informes del sistema, como por ejemplo estadísticas de registros, será un actor.

b) CASOS DE USO

Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Un caso de uso es iniciado por un actor. A partir de ese momento, ese actor, junto con otros actores, intercambian datos o control con el sistema, participando de ese caso de uso.

El nombre de un caso de uso se expresa con un verbo en gerundio, seguido generalmente por el principal objeto o entidad del sistema que es afectado por el caso. Gráficamente, los casos de uso se representan con un óvalo, con el nombre del caso en su interior. Como muestra la Figura 2.8.

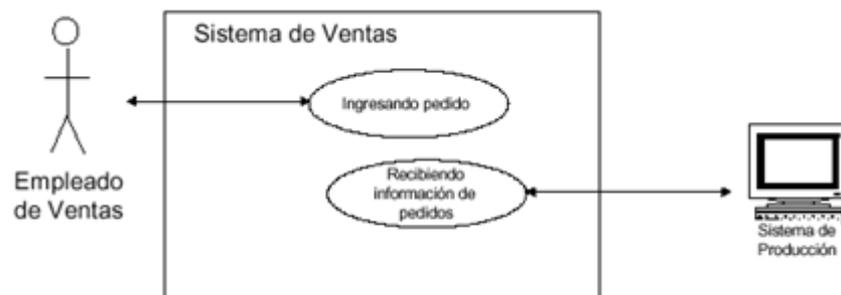


Figura 2. 8 Los casos de uso se representan con óvalos

Los casos de uso tienen las siguientes características:

- Están expresados desde el punto de vista del actor.
- Se documentan con texto informal.
- Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él, aunque el énfasis está puesto en la interacción.
- Son iniciados por un único actor.
- Están acotados al uso de una determinada funcionalidad del sistema, claramente diferenciada.

c) EXTENSIONES (EXTENDS)

Muchas veces, la funcionalidad de un caso de uso incluye un conjunto de pasos que ocurren sólo en algunas oportunidades. Supongamos que estamos especificando un sistema en el cual los candidatos pueden registrarse interactivamente, y que dentro de la funcionalidad del registro el usuario puede solicitar al sistema que le haga una presentación sobre los pasos o fases para el ingreso al ejército. En este caso, tengo una excepción dentro del caso de uso "Registro de Datos". La excepción consiste en interrumpir el caso de uso y pasar a ejecutar el caso de uso "Revisando Presentación de Fases". En este caso decimos que el caso de uso "Revisando Presentación de Fases", extiende el caso de uso "Registro de Datos" y se representa por una línea de trazos desde el caso que 'extiende a' al caso que es extendido. Como muestra la Figura 2.9.

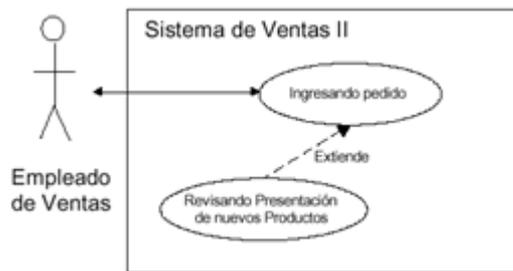


Figura 2. 9 Una relación de extensión entre dos casos de uso

Las extensiones tienen las siguientes características:

- Representan una parte de la funcionalidad del caso que no siempre ocurre.
- Son un caso de uso en sí mismas.
- No necesariamente provienen de un error o excepción.

d) USOS (USES)

Es común que la misma funcionalidad del sistema sea accedida a partir de varios casos de uso. Por ejemplo, la funcionalidad de buscar un candidato puede ser accedida desde la verificación de carpetas, desde las consultas de candidatos, o desde los reportes de candidatos. ¿Cómo hago para no repetir el texto de esta funcionalidad en todos los casos de uso que la acceden? La respuesta es simple: sacando esta funcionalidad a un nuevo caso de uso, que es usado por los casos de los cuales fue sacada. Este tipo de relaciones se llama relaciones de uso y se representa por una línea punteada desde el caso que 'usa a' al caso que es 'usado'.

Decimos, por ejemplo, que el caso de uso "Obteniendo reporte de candidatos por provincia", usa al caso de uso Buscando candidato. Como muestra la Figura 2.10.



Figura 2. 10 Relaciones de Uso entre Casos de Uso

Este concepto no es novedoso, es simplemente el concepto de la subrutina o subprograma usado en un nivel más alto de abstracción.

Las características de las relaciones de uso son:

- Aparecen como funcionalidad común, luego de haber especificado varios casos de uso.
- Los casos usados son casos de uso en sí mismos.
- El caso es usado siempre que el caso que lo usa es ejecutado. Esto marca la diferencia con las extensiones, que son opcionales.

e) ABSTRACCIÓN

Al identificar relaciones de uso y extensión, puede ser que extraigamos casos de uso que son accedidos por varios actores. Por ejemplo, el caso de uso "buscando datos de candidato" es accedido por muchos actores (el oficial o voluntario que verifica los datos en las carpetas, el oficial que quiere obtener estadísticas del proceso, el supervisor que quiere consultar la información de algún candidato, etc.). Ahora bien, como el caso de uso nunca se ejecuta fuera del contexto de otro caso de uso, decimos que es un caso de uso abstracto. Lo llamamos abstracto porque no es implementable por sí mismo: sólo tiene sentido como parte de otros casos.

De la misma forma, el actor que participa de este caso de uso, que reúne características comunes a todos los actores de los casos de uso que lo usan, es un actor abstracto. En nuestro ejemplo, podemos decir que tenemos un actor abstracto "Buscador de Datos de Candidato". Los actores abstractos,

entonces, son necesarios para no dejar sin actores a los casos de uso abstractos. Como muestra la Figura 2.11.

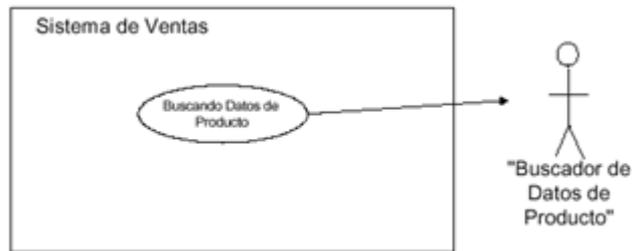


Figura 2. 11 El nombre de los actores abstractos suele no ser feliz

f) HERENCIA

La duda ahora es cómo relacionar este actor abstracto con los actores concretos: los que sí existen en la realidad y ejecutan casos de uso concretos, como "registro" y "obteniendo estadísticas de proceso".

Para esto se puede usar el concepto de herencia, uno de los conceptos básicos de la orientación a objetos. Como todos los actores concretos también ejecutan el caso "buscando datos de candidato", a través de la relación de uso, podemos decir que los actores concretos heredan al actor abstracto. Como muestra la Figura 2.12.



Figura 2. 12 Herencia de actores

La relación de herencia no necesariamente implica la existencia de un caso abstracto. Puede ocurrir que un actor ejecute todos los casos que ejecuta otro actor, y algunos más. En nuestro sistema, el oficial a cargo puede hacer todo lo que hace el usuario normal, pero además puede autorizar la verificación de carpetas. En este caso, podemos decir que el oficial a cargo hereda al usuario normal, aunque el usuario normal no sea un actor abstracto. De esta forma, toda la funcionalidad que está habilitada para el usuario normal también lo está para el oficial a cargo.

2.1.4.4.4. EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS CON CASOS DE USO

A continuación, se describen los pasos a seguir para aplicar la técnica con casos de uso a la IR.

a) IDENTIFICAR LOS ACTORES

Si la primera pregunta que un analista debe hacer a sus usuarios es ¿Para qué es este sistema?, la segunda es claramente ¿Para quiénes es este sistema? Como mencionamos al hablar sobre los actores, identificar a todos ellos es crítico para un buen análisis de requerimientos. Por lo tanto, antes de avanzar con los casos de uso, debo tratar de identificar todos los tipos de usuario diferentes que tiene el sistema. Si el sistema será implementado en una empresa, debo preguntar cuáles de las áreas afectadas usarán o actualizarán su información.

A pesar de hacer una identificación inicial de los actores, también debo repetirla a medida que empiezo a describir los casos de uso, ya que al conocer más detalles del sistema, pueden aparecer nuevos tipos de usuarios.

b) IDENTIFICAR LOS PRINCIPALES CASOS DE USO DE CADA ACTOR

El siguiente paso es enunciar los nombres de los principales casos de uso de cada uno de los actores que se identificaron en el paso anterior. No es necesario especificar cuáles son las acciones dentro del caso de uso. Tampoco debo preocuparme si no aparecen muchos casos, ya que existen técnicas para encontrar nuevos casos de uso a partir de los existentes.

c) IDENTIFICAR NUEVOS CASOS A PARTIR DE LOS EXISTENTES

Uno de los principales errores que se pueden cometer al identificar requerimientos es algo que parece obvio, pero que muchas veces ocurre: ¡olvidarse de algún requerimiento! Como los requerimientos están en la cabeza de los usuarios, el éxito de esta tarea depende de la habilidad del analista.

Para ayudarnos a identificar nuevos casos de uso a partir de los casos existentes, podemos aplicar las mismas técnicas utilizadas para identificar eventos según el análisis estructurado. Algunas de las preguntas que debemos hacernos son:

- ¿Cuáles son las tareas de un actor?
- ¿Necesita el actor estar informado de ciertas ocurrencias del sistema?
- ¿Necesita el actor informar al sistema de cambios externos súbitos?
- ¿Proporciona el sistema el comportamiento correcto al negocio?
- ¿Pueden ser todos los requerimientos funcionales, desarrollados por los casos de uso?
- ¿Qué casos de uso soportarán y mantendrán al sistema?
- ¿Qué información debe ser modificada o creada?

d) DOCUMENTACIÓN DE CASOS DE USO

Una vez que identificamos todos los casos de uso, se empieza a documentar sus pasos; este documento se crea para cada caso de uso, detallando lo que el sistema debe proporcionar al actor cuando el caso de uso es ejecutado. Esta tarea no es estrictamente secuencial de la anterior: es posible que, mientras empezamos a documentar los casos, se sigue buscando otros nuevos.

Un contenido típico de un documento de caso de uso sería:

- Describir cómo comienza el caso de uso y cómo termina.
- Realizar un flujo normal de eventos.
- Realizar un flujo alterno de eventos.
- Detallar las excepciones al flujo de eventos.

e) DEFINIR PRIORIDADES

Una vez documentados los casos de uso, es conveniente definir las prioridades de los distintos requerimientos, expresados como casos de uso. Para los escenarios claves y los casos de uso que serán analizados en su iteración, se debe:

- Representar la funcionalidad central de cada caso de uso.
- Cubrir varios aspectos de arquitectura.
- Poner bajo estrés un punto delicado de la arquitectura.

f) GRÁFICOS A UTILIZAR

Dependiendo del tamaño del sistema, es probable que un único gráfico con todos los casos de uso nos quede chico. No olvidemos que los modelos gráficos son para aclarar el texto, y no para confundir. Si el gráfico de casos de uso es una maraña indescifrable, no está cumpliendo su objetivo. Por lo tanto,

podemos usar las siguientes reglas para incluir gráficos de casos de uso dentro de la SRS.

- Un gráfico de casos de uso no debe mostrar más de 15 casos
- Si es necesario particionar el gráfico, debe hacerse por actor. La primera partición debe separar los casos centrales de los casos auxiliares, ya que probablemente les interesen a personas distintas.
- Si las relaciones de uso y las extensiones entran en el diagrama principal, sin dejar de cumplir con la regla 1, debo dejarlas ahí. Lo mismo se aplica a los actores abstractos.
- Si las relaciones de uso no entran en el diagrama principal, debo mostrarlas en gráficos teniendo en cuenta que siempre debo mostrar todos los casos de uso que usan a otro en un mismo diagrama.
- Si tengo un caso de uso que es usado por gran parte de los otros casos, como por ejemplo el caso de uso "Identificándose ante el sistema", debo evitar mostrarlo en el gráfico principal, ya que las flechas serán imposibles de organizar. Es probable que no haga falta mostrar esta relación de uso en un gráfico.

2.1.4.5. HERRAMIENTAS AUTOMATIZADAS PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS

La Ingeniería de Software cuenta con una serie de herramientas automatizadas destinadas a diferentes propósitos. Dentro de las herramientas CASE que sirven de apoyo a los procesos de Ingeniería de Software, están las especializadas en la administración de requisitos. Estas herramientas se concentran en capturar requerimientos, administrarlos y producir una especificación de requisitos.

Las ventajas que nos proporcionan las herramientas automatizadas para IR son:

- Permiten un mayor control de proyectos complejos.
- Permiten reducir costos y retrasos en la liberación de un proyecto.
- Permiten una mayor comunicación en equipos de trabajo.
- Ayudan a determinar la complejidad del proyecto y esfuerzos necesarios.

2.1.4.6. IEEE-STD-830-1998 : ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE.

2.1.4.6.1. DEFINICIONES

En general las definiciones de los términos usados en estas especificaciones están conforme a las definiciones proporcionadas en IEEE Std 610.12-1990.

a) CONTRATO:

Un documento es legalmente obligatorio y en el estarán de acuerdo las partes del cliente y proveedor. Esto incluye los requisitos técnicos y requerimientos de la organización, costo y tiempo para un producto. Un contrato también puede contener la información informal pero útil como los compromisos o expectativas de las partes involucradas.

b) CLIENTE:

La persona que paga por el producto y normalmente (pero no necesariamente) define los requisitos. En la práctica el cliente y el proveedor pueden ser miembros de la misma organización.

c) PROVEEDOR:

La persona que produce un producto para un cliente.

d) USUARIO:

La persona (s) que operan o actúan recíprocamente directamente con el producto. El usuario y el cliente no es a menudo la misma persona.

2.1.4.6.2. LAS CONSIDERACIONES PARA PRODUCIR UN BUEN SRS.

Estas cláusulas proporcionan información a fondo que deben ser consideradas al momento de producir un SRS. Esto incluye lo siguiente:

- a)** La Naturaleza del SRS;
- b)** El Ambiente del SRS;
- c)** Las Características de un buen SRS ;
- d)** La preparación de los Joins del SRS;
- e)** La evolución de SRS;
- f)** Prototipos;
- g)** Generando el diseño en el SRS;
- h)** Generando los requisitos del proyecto en el SRS.

a) NATURALEZA DEL SRS

El SRS son especificaciones para un producto del software en particular, programa, o juego de programas que realizan ciertas funciones en un ambiente específico. El SRS puede escribirse por uno o más representantes del proveedor, uno o más representantes del cliente, o por ambos. Se recomienda ambos.

Los problemas básicos que se presentan al escribir un SRS van dirigidos a lo siguiente:

- La Funcionalidad.
¿Qué se supone va hacer el software ?
- Las interfaces Externas.
¿Cómo el software actúa recíprocamente con las personas, el hardware de los sistemas, otro hardware, y otro software?
- La Actuación.
¿Cuál es la velocidad, la disponibilidad, tiempo de la contestación, tiempo de la recuperación de varias funciones del software, etc.?
- Los Atributos.
¿Qué portabilidad tiene, exactitud, el mantenimiento, la seguridad, las consideraciones etc.?
- Las restricciones del diseño que impusieron en una aplicación.
¿Hay algún requerimiento Standard, idioma de aplicación, las políticas para la integridad del banco de datos, los límites de los recursos, operando en qué ambiente etc.?

b) AMBIENTE DEL SRS

Es importante considerar la parte que el SRS representa en el diseño del proyecto total que se define en IEEE Std 610.12-1990. El software puede contener toda la funcionalidad del proyecto esencialmente o puede ser parte de un sistema más grande.

En el último caso habrá un SRS que declarará las interfaces entre el sistema y su software modular, y pondrá qué función externa y requisitos de funcionalidad tiene con el software modular.

Desde que el SRS tiene un papel específico en el proceso de desarrollo de software, el que define el SRS debe tener el cuidado para no ir más allá de los límites de ese papel.

Esto significa que:

- Debe definir todos los requisitos del software correctamente. Un requisito del software puede existir debido a la naturaleza de la tarea a ser resuelta o debido a una característica especial del proyecto.
- No debe describir cualquier plan o detalles de aplicación. Éstos deben describirse en la fase del diseño del proyecto.
- No debe imponer las restricciones adicionales en el software. Éstos se especifican propiamente en otros documentos.

c) CARACTERÍSTICAS DE UN BUEN SRS.

Un SRS debe ser:

- Correcto
- Inequívoco
- Completo
- Consistente
- Delinear que tiene importancia y/o estabilidad
- Comprobable
- Modificable
- Identificable.

d) CORRECTO

Un SRS es correcto si, y sólo si, cada requisito declarado se encuentra en el software.

No hay ninguna herramienta o procedimiento que aseguran la exactitud.

Alternativamente el cliente o el usuario pueden determinar si el SRS refleja las necesidades reales correctamente. Identificando los requerimientos hace este procedimiento más fácil y hay menos probabilidad al error.

e) INEQUÍVOCO

Un SRS es inequívoco si, y sólo si, cada requisito declarado tiene sólo una interpretación. Como un mínimo, se requiere que cada característica de la última versión del producto se describa usando un único término.

En casos dónde un término en un contexto particular tenga significados múltiples, el término debe ser incluido en un glosario dónde su significado es hecho más específico.

Un SRS es una parte importante del proceso de requisitos del ciclo de vida de software y se usa en el diseño, aplicación, supervisión, comprobación, aprobación y pruebas como está descrito en IEEE Std 1074-1997.

El SRS debe ser inequívoco para aquéllos que lo crean y para aquéllos que lo usan. Sin embargo, estos grupos no tienen a menudo el mismo fondo y por consiguiente no tienden a describir los requisitos del software de la misma manera.

- **TRAMPAS DEL IDIOMA NATURAL.**

Los requisitos son a menudo escritos en el idioma natural (por ejemplo, inglés) el idioma natural es inherentemente ambiguo. Un idioma natural que SRS podría ser revisado por una parte independiente para identificar el uso ambiguo del idioma para que pueda corregirse.

- **IDIOMAS DE ESPECIFICACIÓN DE REQUISITOS**

Una manera de evitar la ambigüedad inherente en el idioma natural es escribir el SRS en un idioma de especificación de requisitos particular. Sus procesadores del idioma descubren muchos errores léxicos, sintácticos, y semánticos automáticamente.

Una desventaja en el uso de tales idiomas es que la premura de tiempo exigió aprenderlos. También, muchos usuarios no-técnicos los encuentran ininteligible. Es más, estos idiomas tienden a ser buenos a expresar ciertos tipos de requisitos y dirigirse a ciertos tipos de sistemas. Así, ellos pueden influir en los requisitos de las maneras sutiles.

- **REPRESENTACIÓN HECHA CON HERRAMIENTAS**

En general, los métodos de requisitos e idiomas y las herramientas que los apoyan entran en tres categorías generales - el objeto, procesos y conductual.

El término objetos-orientados organizan los requisitos en lo que se refiere a los objetos en el mundo real , sus atributos, y los servicios realizados por esos objetos.

El término procesos organizan los requisitos en las jerarquías de funciones que comunican vía el flujo de datos.

El términos conductuales describen conducta externa del sistema por lo que se refiere a alguna noción de lo abstracto, las funciones matemáticas o el estado de las máquinas.

El grado en que se usan estas herramientas y los métodos pueden ser útiles preparando un SRS pero depende del tamaño y complejidad del programa. Aún usando cualquiera de estos términos es mejor retener las descripciones del idioma natural. Así, clientes poco familiar con las anotaciones el SRS puede entender todavía.

f) COMPLETO

Un SRS está completo si, y sólo si, incluye los elementos siguientes:

- a) Los requisitos están relacionados a la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas. En particular debe reconocerse cualquier requisito externo impuesto por una especificación del sistema y debe tratarse.
- b) La definición de las respuestas del software a todos los posibles datos de la entrada del sistema y a toda clase de situaciones. Una nota que es importante especificar son las contestaciones a las entradas válidas e inválidas a ciertos valores.
- c) Tener todas las etiquetas llenas y referencias a todas las figuras, tablas, diagramas en el SRS y definición de todas las condiciones y unidades de medida.

g) CONSISTENTE

La consistencia se refiere a la consistencia interior. Si un SRS no está de acuerdo con algún documento del superior-nivel, como una especificación de requisitos de sistema, entonces no es correcto.

- **CONSISTENCIA INTERIOR**

Un SRS es internamente consistente si, y sólo si, ningún subconjunto de requisitos individuales genero conflicto en él.

Los tres tipos de conflictos probables en un SRS son:

- a) Las características especificadas en el mundo real de los objetos pueden chocar. Por ejemplo:
 - 1. El formato de un informe del rendimiento puede describirse en un requisito como tabular pero en otro como textual.
 - 2. Un requisito puede declarar que todo las luces serán verdes mientras otro puede declarar que todo las luces sean azules.

- b) Puede haber conflicto lógico o temporal entre dos acciones especificadas. Por ejemplo:
 - 1. Un requisito puede especificar que el programa sumará dos entradas y otro puede especificar que el programa los multiplicará.
 - 2. Un requisito puede declarar que "A" siempre debe seguir "B", mientras otro puede requerir que "A" "y B" ocurran simultáneamente.

Dos o más requisitos pueden describir el mismo mundo real del objeto pero uso las condiciones diferentes para ese objeto. Por ejemplo, una demanda del programa para una entrada del usuario puede llamarse una "sugerencia" en un requisito y una "señal" en otro. El uso de terminología normal y definiciones promueve la consistencia

h) DELINEAR QUE TIENE IMPORTANCIA Y/O ESTABILIDAD

Un SRS debe delinear la importancia y/o estabilidad si cada requisito en él tiene un identificador para indicar la importancia o estabilidad de ese requisito en particular.

Típicamente, todos los requisitos que relacionan a un producto del software no son igualmente importantes. Algunos requisitos pueden ser esenciales, sobre todo para las aplicaciones de vida crítica, mientras otros pueden ser deseables.

Cada requisito en el SRS debe identificarse para representar estas diferencias, aclarar y ser explícito. Identificando los requisitos de la manera siguientes:

- a) Tienen los clientes que dar las consideraciones muy cuidadosamente a cada requisito para que se clarifique cualquier omisión que ellos pueden tener.

b) Tener diseñadores que hagan diseños correctos y pongan el mismo esfuerzo en todos los niveles del producto del software.

- **GRADO DE ESTABILIDAD**

Puede expresarse la estabilidad por lo que se refiere al número de cambios esperados a cualquier requisito basado en experiencia o conocimiento de eventos venideros que afectan la organización, funciones y a las personas que apoyan el sistema del software.

- **GRADO DE NECESIDAD**

Otra manera de alinear los requisitos es distinguir las clases de requisitos que hay: el esencial, el condicional y optativo.

- a) **Esencial.**

Implica que el software no será aceptable a menos que estos requisitos se proporcionan de una manera convenida.

- b) **El Condicional.**

Implica que éstos son requisitos que reforzarían el producto del software, pero no lo haría inaceptable si ellos están ausentes.

- c) **Optativo.**

Implica una clase de funciones que pueden o no pueden valer la pena. Esto le da la oportunidad de proponer algo que excede el SRS al proveedor.

i) COMPROBABLE

Un SRS es comprobable si, y sólo si, cada requisito declarado es comprobable. Un requisito es comprobable si, y sólo si, allí existe algún proceso rentable finito con que una persona o la máquina puede verificar que el producto del software reúne el requisito. En general cualquier requisito ambiguo no es comprobable.

Los requisitos de No-verificable incluyen las declaraciones como "trabaja bien", "interface humana buena" y "normalmente pasará" no pueden verificarse los requisitos de esos porque es imposible de definir las condiciones "bueno," "bien" o "normalmente". La declaración que "el programa nunca entrará en una vuelta infinita" es el no-verificable porque la comprobación de esta calidad es teóricamente imposible.

Un ejemplo de una declaración comprobable es:

- El rendimiento del programa se producirá dentro de 20 seg de evento 60% del tiempo; y se producirá dentro de 30 seg de evento 100% del tiempo.

Esta declaración puede verificarse porque usa condiciones concretas y las cantidades mensurables.

Si un método no puede inventarse para determinar si el software reúne un requisito particular, entonces ese requisito debe quitarse o debe revisarse.

j) MODIFICABLE

Un SRS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requisitos fácilmente, completamente y de forma consistente mientras conserva la estructura y estilo. Para que sea modificable se requiere un SRS que contenga:

- a) Tiene un coherente y fácil de usar en la organización de volúmenes de información, un índice y las referencias cruzadas explícitas;
- b) No sea redundante (es decir, el mismo requisito no debe aparecer en más de un lugar en el SRS);
- c) Exprese cada requisito separadamente, en lugar de intercalarlas con otros requisitos.

La redundancia no es un error, pero puede llevar fácilmente a los errores. La redundancia puede ayudar hacer un SRS más leíble de vez en cuando, pero un problema puede generarse cuando el documento redundante se actualiza. Por ejemplo, un requisito puede alterarse en un solo lugar dónde aparece. El SRS se pone incoherente entonces. Siempre que la redundancia sea necesaria.

k) IDENTIFICABLE

Un SRS es identificable si el origen de cada uno de sus requisitos está claro y si facilita las referencias de cada requisito en el desarrollo futuro o documentación del mismo. Lo siguiente que se recomiendan dos tipos de identificabilidad:

- a) El identificable dirigido hacia atrás (es decir, a las fases anteriores de desarrollo).

Esto depende explícitamente en cada requisito la referencias de su fuente en los documentos más antiguos.

- b) El identificable delantero (es decir, a todos los documentos desovados por el SRS).

Esto depende en cada requisito en el SRS que tiene un único nombre o número de la referencia. El identificable delantero del SRS es especialmente importante cuando el producto del software entra en el funcionamiento y fase de mantenimiento. Como el código y documentos del plan se modifican, es esencial poder determinar el juego completo de requisitos que pueden afectarse por esas modificaciones.

l) PREPARACIÓN DE LOS JOIN DEL SRS

El proceso de desarrollo de software debe empezar con el proveedor y con el acuerdo del cliente en lo que el software completado debe hacer. Este acuerdo, en la forma de un SRS, debe prepararse juntamente. Esto es importante porque ni el cliente ni el proveedor son calificables para escribir exclusivamente un buen SRS.

- a) Clientes normalmente no entienden bien el diseño del software y proceso de desarrollo bastante bien como para escribir un SRS utilizable.
- b) Los Proveedores normalmente no entienden bien el problema de los clientes y campo de acción bastante bien como para que especifique los requisitos para un sistema satisfactorio.

Por consiguiente, el cliente y el proveedor deben trabajar para producir juntos un buen escrito y completamente entendible SRS.

Una situación especial existe cuando el sistema y su software los dos se están definiéndose concurrentemente. Entonces la funcionalidad, interfaces, desarrollo y otros atributos, restricciones del software no son los predefinidos, sino se definen juntamente y están sujetos a la negociación y al cambio.

m) EVOLUCIÓN DE SRS

El SRS puede necesitar evolucionar así como el desarrollo de las actualizaciones del producto de software. Puede ser imposible de especificar un poco a detalle en el momento que el proyecto se inicia (por ejemplo, puede ser imposible de definir toda la estructura de la pantalla para un programa interactivo durante la fase de requisitos). Los cambios adicionales pueden suceder según como las deficiencias se vayan descubriendo, las limitaciones e inexactitudes en el SRS.

Dos consideraciones en este proceso son las siguientes:

- a) Deben especificarse los requisitos completamente como se es conocido en el momento, aun cuando las revisiones evolutivas pueden preverse como inevitable. El hecho que ellos están incompletos debe ser anotado.
- b) Un proceso de cambio formal debe comenzarse para identificarse, el control, dejar huella e informe de lo que proyectaron los cambios.

Los cambios aprobados en los requisitos deben incorporarse en el SRS de semejante manera acerca de que:

1. Proporcione un lineamiento de la auditoria exacta y completa de cambios;
2. el permiso de la revisión actual y reemplazó de los cambios en SRS.

n) PROTOTIPOS.

Los prototipos frecuentemente se usan durante una fase de los requisitos de un proyecto.

Muchas herramientas existen para generar un prototipo para exhibir algunas características de un sistema, ser creado muy rápidamente y fácilmente.

Los prototipos son útiles por las razones siguientes:

- a) El cliente puede ver el prototipo y reaccionar a él que leer el SRS y reaccionar a él.
Así, el prototipo proporciona la regeneración rápida.
- b) El prototipo despliega aspectos de anticiparse a la conducta de los sistemas. Así, no sólo produce las respuestas sino también las nuevas preguntas. Esto ayuda a ver el alcance en el SRS.

- c) Un SRS basado en un prototipo tiende a sufrir menos cambios durante el desarrollo, así se acorta el tiempo de desarrollo.

Un prototipo debe usarse como una manera de sacar los requisitos del software. Pueden extraerse algunas características como pantalla o formatos del reporte directamente del prototipo. Otros requisitos pueden ser inferidos ejecutando los experimentos con el prototipo.

o) GENERANDO EL DISEÑO EN EL SRS

Un requisito especifica una función externa visible o atributo de un sistema. Un diseño describe un subcomponente particular de un sistema y/o sus interfaces con otros subcomponentes. El diseñador del SRS debe distinguir claramente entre identificar las restricciones del diseño requeridos y proyectar un plan específico. La nota es que cada requisito en el SRS limita las alternativas del plan. Esto no significa, sin embargo, que cada requisito es el plan.

El SRS debe especificar qué funciones serán realizadas, con qué datos, para producir qué resultados, en qué situación y para quien. El SRS se debe enfocar en los servicios a ser realizados. El SRS normalmente no debe especificar los puntos del plan como lo siguiente:

- Partir el software en módulos;
- Asignando las funciones a los módulos;
- Describiendo el flujo de información o controles entre los módulos;
- Escogiendo las estructuras de los datos.

2.1.4.6.3. LAS PARTES DE UN SRS

Estas partes se colocan en un contorno que puede servir como un ejemplo por escribir un SRS.

Un SRS no tiene que seguir este contorno o usar los nombres dado aquí para sus partes, un buen SRS debe incluir toda la información que se mencionó aquí.

a) INTRODUCCIÓN

La introducción del SRS debe proporcionar una apreciación global del SRS completo.

Debe contener las subdivisiones siguientes:

➤ **PROPÓSITO**

Esta subdivisión debe:

- Delinear el propósito del SRS;
- Especifique a que público intencional va dirigido el SRS.

➤ **ALCANCE**

Esta subdivisión debe:

- Identifique el producto (s) del software para ser diseñado por el nombre (por ejemplo, Anfitrión DBMS, el Generador del Reporte, etc.);
- Explique eso que el producto (s) del software que hará y que no hará.
- Describe la aplicación del software especificándose los beneficios pertinentes, objetivos, y metas;
- Sea consistente con las declaraciones similares en las especificaciones de niveles superiores (por ejemplo, las especificaciones de los requisitos del sistema), si ellos existen.

➤ **DEFINICIONES, SIGLAS, Y ABREVIACIONES**

Esta subdivisión debe proporcionar las definiciones de todas las condiciones, las siglas, y abreviaciones que exigen interpretar el SRS propiamente. Esta información puede proporcionarse por la referencia a uno o más apéndices en el SRS o por la referencia a otros documentos.

➤ **REFERENCIAS**

Esta subdivisión debe:

- Proporcione una lista completa de todas las referencias de los documentos en otra parte en el SRS;
- Identifique cada documento por el título, número del reporte (si es aplicable), fecha, y publicación de la organización;
- Especifique las fuentes de las referencias de donde se obtuvieron.

Esta información puede proporcionarse por la referencia a un apéndice o a otro documento.

➤ **APRECIACIÓN GLOBAL**

Esta subdivisión debe:

- Describa lo que el resto del SRS contiene;
- Explica cómo el SRS es organizado.

b) DESCRIPCIÓN GLOBAL

Esta sección del SRS debe describir los factores generales que afectan el producto y sus requisitos. Esta sección no declara los requisitos específicos. En cambio, mantiene un fondo de esos requisitos que se definen en detalle en Sección 3 del SRS y les hacen más fácil entender.

Esta sección normalmente consiste en seis subdivisiones, como sigue:

a. PERSPECTIVA DEL PRODUCTO

Esta subdivisión del SRS debe poner el producto en la perspectiva con otros productos relacionados. Si el producto es independiente y totalmente autónomo, debe declararse que así es. Si el SRS define un producto que es un componente de un sistema más grande, como frecuentemente ocurre, entonces esta subdivisión debe relacionar los requisitos de ese sistema más grande a la funcionalidad del software y debe identificar las interfaces entre ese sistema y el software.

Un diagrama del bloque que muestra los componentes mayores del sistema más grande, las interconexiones, y las interfaces externas pueden ser útiles.

Esta subdivisión también debe describir cómo el software opera dentro de las varias restricciones. Por ejemplo, estas restricciones podrían incluir:

➤ INTERFACES DEL SISTEMA.

Esto debe listar cada interfaz del sistema y debe identificar la funcionalidad del software para lograr el requisito del sistema y la descripción de la interfaz para empatar el sistema.

➤ INTERFACES CON EL USUARIO.

Esto debe especificar a lo siguiente:

- Las características lógicas de cada interfaz entre el producto del software y sus usuarios.

Esto incluye las características de la configuración (por ejemplo, formatos de la pantalla requeridos, página o esquemas de la

ventana, los reportes o menús o disponibilidad de llaves de la función programables) necesario para lograr los requisitos del software.

- Todos los aspectos para perfeccionar la interfaz con la persona que debe usar el sistema.

Esto puede comprender una lista de lo que hace y no hace simplemente delante de cómo el sistema aparecerá al usuario. Un ejemplo puede ser un requisito para la opción de mensajes de error largos o cortos. Como todos, estos requisitos deben ser comprobables, debe especificarse en los Atributos de Sistema de Software bajo una sección titulada Facilidad de Uso.

➤ **INTERFACES CON EL HARDWARE.**

Esto debe especificar las características lógicas de cada interfaz entre el producto del software y los componentes del hardware del sistema. Esto incluye las características de la configuración (el número de puertos, la instrucción set, etc.), también cubre como qué dispositivos serán apoyados, cómo ellos serán apoyados y protocolos. Por ejemplo, el apoyo de las terminales puede especificarse cuando tienen full-screen.

➤ **INTERFACES CON EL SOFTWARE.**

Esto debe especificar el uso de otros productos del software requeridos (por ejemplo, un sistema de dirección de datos, un sistema operativo o un paquete matemático) e interfaces con otros sistemas de la aplicación (por ejemplo, la unión entre el Sistema de Cuentas, el Sistema por Cobrar y un Sistema del Mayor General). Para cada uno el producto del software requirió proporcionarse:

- El nombre;
- El código mnemotécnico;
- El número de la especificación;
- El número de la versión;
- La fuente.

Para cada interfaz, lo siguiente debe proporcionarse:

- La discusión del propósito de la interfaz del software en relación con el producto del software.

- La definición de la interfaz por lo que se refiere a los mensajes contenidos y formatos.

No es necesario detallar cualquiera bien la documentación de la interfaz, pero una referencia al documento que define la interfaz se requiere.

➤ **INTERFACES DE COMUNICACIONES**

Esto debe especificar las varias interfaces a las comunicaciones como los protocolos de las redes locales, etc.

➤ **RESTRICCIONES DE MEMORIA**

Esto debe especificar cualquier característica aplicable y límites en la memoria primaria y la memoria secundaria.

➤ **FUNCIONAMIENTOS**

Esto debe especificar los funcionamientos normales y especiales requeridos por el usuario como:

- Los varios modos de funcionamientos en la organización del usuario (por ejemplo, los funcionamientos de iniciar el usuario);
- Los periodos de funcionamientos interactivos y periodo de funcionamientos desatendido;
- Datos que procesan las funciones de apoyo;
- El apoyo y funcionamientos de la recuperación.

NOTA Esto a veces se especifica como la parte del User Interfaces Sección.

➤ **REQUISITOS DE ADAPTACIÓN DEL SITE.**

Esto debe:

- Defina los requisitos para cualquier dato o la secuencia de inicialización que son específico a un sitio dado, la misión o el modo operacional (por ejemplo, los límites de seguridad, etc.);
- Especifique el sitio o los rasgos que se deben relacionar que deben modificarse para adaptar el software a una instalación particular.

b. FUNCIONES DEL PRODUCTO

Esta subdivisión del SRS debe proporcionar un resumen de las funciones mayores que el software realizará.

Por ejemplo, un SRS para un programa de contabilidad puede acostumbrar esta parte a dirigirse al mantenimiento de Cuenta de Cliente, declaración del cliente y preparación de la factura sin mencionar la inmensa cantidad de detalle que cada uno de esas funciones requiere.

A veces el resumen de la función que es necesario para esta parte puede tomarse directamente de la sección de la especificación en el nivel superior (si uno existe) eso asigna las funciones particulares al producto del software. Note que eso es por causa de la claridad.

- Las funciones deben organizarse en cierto modo eso hace la lista de funciones entendible al cliente o a cualquiera nada más leyendo el documento la primera vez.
- Pueden usarse los métodos Textuales o gráficos para mostrar las funciones diferentes y sus relaciones.

No se piensa que el diagrama muestra un diseño de un producto, sino simplemente muestra la relación lógica entre las variables.

c. CARACTERÍSTICAS DEL USUARIO

Esta subdivisión del SRS debe describir esas características generales de los usuarios intencionales del producto que incluye nivel educativo, experiencia, y la especialización técnica.

d. RESTRICCIONES

Esta subdivisión del SRS debe proporcionar una descripción general de cualquier otro punto que limitará las opciones de los diseñadores. Éstos incluyen:

- a) las políticas reguladoras;
- b) las limitaciones del Hardware;
- c) las Interfaces a otras aplicaciones;
- d) el funcionamiento Paralelo;
- e) las funciones de la Auditoría;
- f) las funciones de Control;

- g) los requisitos de lenguaje;
- h) los requisitos de Fiabilidad;
- i) Credibilidad de la aplicación;
- j) la Seguridad y consideraciones de seguridad.

e. ATENCIONES Y DEPENDENCIAS

Esta subdivisión del SRS debe listar cada uno de los factores que afectan los requisitos declarados en el SRS.

Estos factores no son las restricciones del diseño en el software pero son, más bien, cualquier cambio a ellos eso puede afectar los requisitos en el SRS. Por ejemplo, una suposición puede ser que un sistema operativo específico estará disponible en el hardware designado para el producto del software. Si, de hecho, el sistema operativo no está disponible, los SRS tendrían que cambiar de acuerdo con entonces.

f. PRORRATEAR LOS REQUISITOS

Esta subdivisión del SRS debe identificar requisitos que pueden tardarse hasta las versiones futuras del sistema.

c) REQUISITOS ESPECÍFICOS

Esta sección del SRS debe contener todos los requisitos del software a un nivel de detalle suficiente para permitirles a los diseñadores diseñar un sistema para satisfacer esos requisitos, y a los auditores a probar que el sistema satisface esos requisitos. A lo largo de esta sección, cada requisito declarado debe ser externamente perceptible por los usuarios, operadores u otros sistemas externos. Estos requisitos deben incluir por lo menos una descripción de cada entrada (el estímulo) en el sistema, cada salida (la contestación) del sistema, y todas las funciones realizadas por el sistema en la salida a una entrada o en el apoyo de la salida. Esta es la parte más grande y más importante del SRS, los principios siguientes aplican:

- Deben declararse los requisitos específicos en la conformidad con todas las características descritas.
- Los requisitos específicos deben tener referencias cruzadas a documentos más actuales que los relacionan.
- Todos los requisitos deben ser singularmente identificables.

- Debe prestarse la atención debida a organizar los requisitos para aumentar al máximo la legibilidad.

➤ **INTERFACES EXTERNAS**

Esta debe ser una descripción detallada de todas las entradas y salidas del sistema del software. Debe complementar las descripciones de la interfaz y no debe repetirse la información.

Debe incluir ambas entradas/salidas y debe estructurarse como sigue:

- El nombre de la entrada;
- La descripción de propósito;
- La fuente de entrada o destino de salida;
- El rango válido, exactitud, y/o tolerancia;
- Las unidades de medida;
- Tiempos;
- Las relaciones a otras entradas/salidas;
- El formato de pantalla /organización;
- El formato de ventanas/organización;
- Los formatos de los datos;
- Los formatos de los comandos;
- Fin de mensajes.

➤ **FUNCIONES**

Los requisitos funcionales deben definir las acciones fundamentales que deben tener lugar en el software, aceptando y procesando las entradas, procesando y generando las salidas. Éstos generalmente se listan como "debe" declaraciones que empiezan con "El sistema debe"

Éstos incluyen:

- Verificar la validez sobre las entradas
- La secuencia exacta de las operaciones
- Las contestaciones a las situaciones anormales, incluyendo
 - Overflow
 - Facilidades de comunicación
 - Manejo de errores y recuperación
- El efecto de parámetros
- La relación de salidas a las entradas, incluyendo

- Las secuencias de entrada/salidas
- Las fórmulas de entrada y su conversión a la salida

Puede ser apropiado dividir los requisitos funcionales en subfunciones o subprocesos.

Esto no implica que el plan del software también se dividirá así.

➤ **REQUISITOS DEL DESARROLLO.**

Esta subdivisión debe especificar los requerimientos estáticos y dinámicos que se pusieron en el software o en la interacción humana con el software en conjunto. Los requisitos estáticos pueden incluir a lo siguiente:

- El número de terminales a ser apoyadas;
- El número de usuarios simultáneos ser apoyados;
- La cantidad y tipo de información que se maneja.

A veces se identifican los requisitos estáticos bajo una sección separada titulada la Capacidad. Por ejemplo, los requisitos dinámicos pueden incluir los números de transacciones, tareas y la cantidad de datos a ser procesado dentro de ciertos periodos de tiempo para las condiciones del trabajo normales y máximas.

Todos que estos requisitos deben declararse en las condiciones mensurables. Por ejemplo, 95% de las transacciones se procesarán en menos de 1 seg.

NOTA: Normalmente se especifican límites numéricos aplicados a una función específica como la parte de la descripción de subinciso de proceso de esa función.

➤ **REQUISITOS DEL BANCO DE DATOS LÓGICOS**

Esto debe especificar los requisitos lógicos para cualquier información que será puesta en un banco de datos. Esto puede incluir a lo siguiente:

- Los tipos de información usadas por varias funciones;
- La frecuencia de uso;
- Accediendo las capacidades;
- Las entidades de los datos y sus relaciones;

- Las restricciones de integridad;
- Requerimientos en la retención de datos.

➤ **RESTRICCIONES DEL DISEÑO.**

Esto debe especificar las restricciones del diseño que pueden imponerse por otros estándares, las limitaciones del hardware, etc.

i. ACEPTACIÓN DE LAS NORMAS

Esta subdivisión debe especificar los requisitos derivados de estándares existentes o regulaciones. Ellos pueden incluir a lo siguiente:

- El formato del reporte;
- Los nombres de los datos;
- Los procedimientos de contabilidad;
- Los lineamientos de la Auditoría.

Por ejemplo, esto podría especificar los requisito para el software y rastrear la actividad del proceso. Se necesita rastrear algunas aplicaciones para encontrarse al menos las normas reguladoras o financieras. Por ejemplo, un requisito de rastro de auditoría puede declarar que deben grabarse todos los cambios a un banco de datos de la nómina en un archivo del rastro con los valores antes del proceso y después del proceso.

ii. ATRIBUTOS DEL SOFTWARE DEL SISTEMA.

Hay varios atributos del software que puede servir como los requisitos. Es importante que los atributos se especifiquen para que su logro pueda verificarse objetivamente.

- **FIABILIDAD**

Esto debe especificar que los factores exigieron establecer la fiabilidad requerida del sistema del software al momento de la entrega.

- **DISPONIBILIDAD**

Esto debe especificar que los factores exigieron garantizar un nivel de disponibilidad definido para el sistema como un punto de control, la recuperación y al iniciar.

- **SEGURIDAD**

Esto debe especificar los factores que protegen el software del acceso accidental o malévolo, uso, modificación, destrucción o descubrimiento. Los requisitos específicos en esta área podrían incluir la necesidad a:

- Utilice ciertas técnicas de encriptación;
- Tenga Log de entrada o históricos de datos;
- Asigne ciertas funciones a módulos diferentes;
- Restrinja las comunicaciones entre algunas áreas del programa;
- La integridad de datos se verifique para variables críticas.

- **MANTENIMIENTO**

Esto debe especificar atributos de software que relaciona a la facilidad de mantenimiento del propio software. Puede haber algún requisito con toda seguridad de modularidad, interfaces, la complejidad, etc. no deben ponerse los requisitos aquí.

- **PORTABILIDAD**

Esto debe especificar atributos de software que relaciona a la facilidad de poner el software a otro servidor y/o sistemas operativos. Esto puede incluir a lo siguiente:

- El porcentaje de componentes con código cliente-servidor;
- El porcentaje de código del cliente-servidor;
- El uso de un idioma portátil probado;
- El uso de un compilador particular o subconjunto de lenguajes;
- El uso de un sistema operativo particular.

iii. **ORGANIZAR LOS REQUISITOS ESPECÍFICOS.**

Por algo los requisitos detallados de los sistemas triviales tienden a ser extenso. Por esta razón, se recomienda que sean cuidadosos de organizar éstos de una manera óptima para que sean entendibles.

iv. **COMENTARIOS ADICIONALES**

Siempre que un nuevo SRS se contemple, organice los requisitos específicos para jerarquías múltiples detalladas a las necesidades específicas del sistema bajo la especificación.

Hay muchas anotaciones, métodos y herramientas de apoyo automatizadas disponibles para ayudar en la documentación de requisitos. La mayor parte, su utilidad es una función de organización. Por ejemplo, al organizar por el modo, máquinas de estado finitas o los mapas estatales pueden demostrar utilidad; al organizar por el objeto, el análisis objeto-orientado puede demostrar utilidad; al organizar por el rasgo, las secuencias de estímulo-contestación pueden demostrar utilidad y al organizar por la jerarquía funcional, los datos fluyen según los diagramas y los diccionarios de datos pueden demostrar también utilidad.

En cualquiera de los contornos dados, esas secciones llamadas "Requisito Funcional" puede describirse en el idioma nativo (por ejemplo, inglés), en el pseudo código, en un idioma de definición de sistema, o en cuatro subdivisiones tituladas: La introducción, Entradas, Proceso, y Rendimientos.

➤ **INFORMACIÓN DE APOYO**

La información de apoyo hace más fácil al SRS para usarse. Incluye a lo siguiente:

i. **TABLA DE CONTENIDOS E ÍNDICE**

La tabla de contenidos e índice es bastante importante y debe seguir las prácticas de las composiciones generales.

ii. APÉNDICES

Los apéndices no siempre son considerados parte del SRS real y no siempre son necesarios. Ellos pueden incluir:

- Ejemplos de formatos de las entradas/salidas, las descripciones del análisis del costo que se estudiaron o resultados de estudios del usuario;
- Apoyando o dando información a fondo que puede ayudar a los lectores del SRS;
- Una descripción de los problemas a ser resuelto por el software;
- las instrucciones del empaquetamiento especiales para el código y los medios de comunicación para reunir la seguridad, exportar la carga inicial u otros requisitos.

Cuando los apéndices son incluido, el SRS debe declarar explícitamente sí o no los apéndices serán considerados parte de los requisitos.

2.1.4.6.4. LAS PLANTILLAS DE SRS

d) La Plantilla de A.1 de SRS Sección 3 organizada por el modo: Versión 1

- Los requisitos específicos
 - requisitos de las interfaces externas
 - interfaz con el usuario
 - interfaz con el hardware
 - interfaz con el software
 - interfaces de comunicaciones
 - requisitos funcionales
 - modo 1
 - requisito 1.1 funcional
 -
 -
 -
 - .n requisito 1.n Funcional
 - modo 2
 -
 -

- .
 - Modo m
 - requisito Funcional m.1
 - .
 - .
 - .
 - requisito Funcional m.n
- Requisitos del desarrollo
- Restricciones del diseño
- Atributos de sistema de software
- Otros requisitos

CAPÍTULO III

3.1. ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA

3.1.1. ESTRUCTURA ORGANIZACIONAL.

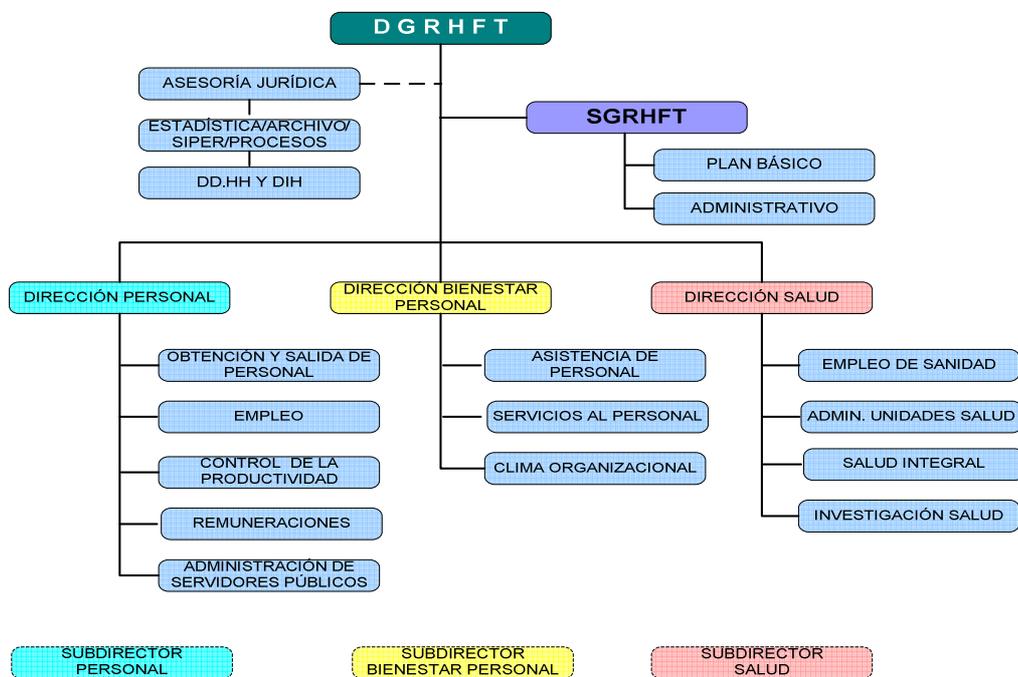


Figura 3. 1 Estructura Orgánica de la Dirección de Recursos Humanos

3.1.2. DESCRIPCIÓN DE LOS PROCESOS DEL SUBSISTEMA DE INGRESO Y SELECCIÓN

3.1.2.1. OBJETIVO:

Seleccionar mediante un proceso técnico a los ciudadanos idóneos para que ingresen en calidad de Aspirantes a Oficiales y Voluntarios a la Fuerza Terrestre.

3.1.2.2. ALCANCE:

Este procedimiento es aplicable al personal de Aspirantes a Oficiales y Voluntarios que han sido reclutados, se inicia con los listados del personal idóneo, como resultado del proceso del reclutamiento y finaliza con la publicación del Alta en la Orden General y su respectivo registro en el sistema.

3.1.2.3. RESPONSABLES:

RESPONSABLE	ACTIVIDAD
El Comandante General de la F.T.	➤ Aprobar y Legalizar

La Junta de Reclutamiento y Selección	➤ Seleccionará al personal de Aspirantes a Oficiales y Tropa.
El Jefe del Dpto. de Obtención	➤ Controlará y supervisará que el proceso de Selección se cumpla conforme a lo planificado.
Los Directores y/o Comandantes de las Direcciones, Divisiones, Brigadas y Escuelas de Formación	• Darán a conocer los resultados del personal idóneo, mediante lo publicado en la página Web e Institutos respectivos.

3.1.2.4. DEFINICIONES:

COT Comando de Operaciones Terrestres

CEDFT Comando de Educación y Doctrina de la Fuerza Terrestre

DIFT Dirección de Inteligencia de la Fuerza Terrestre

DGRHFT Dirección General de RR. HH. De la Fuerza Terrestre

DSNDFT Dirección de Sanidad de la Fuerza Terrestre

DCSFT Departamento de Comunicación Social de la Fuerza Terrestre

ESMIL Escuela Superior Militar "Eloy Alfaro"

ESFORSFT Escuela de Formación de Soldados de la Fuerza Terrestre

EIWIAS Escuela de Iwias.

3.1.2.5. ACTIVIDADES DE CALIDAD:

3.1.2.5.1. Receptar y verificar la documentación de los candidatos

Los Institutos de Formación, con sus delegados designados en las Divisiones, Brigadas y demás organismos militares, receptan y verifican que la documentación exigida a los ciudadanos este completa; luego se realiza la entrevista en dónde se les da indicaciones generales de la Institución, y se conoce las expectativas del ciudadano y su representante con respecto a la Institución.

3.1.2.5.2. Analizar la documentación de las carpetas

La Junta de Reclutamiento y Selección, previo conocimiento de las actividades constantes en el cronograma, se reúne para analizar la documentación presentada por los ciudadanos y proceder a la selección y aprobación de las mismas, luego remitir los listados del personal preseleccionado al Dpto. de Obtención para conocimiento.

3.1.2.5.3. Publicar las listas del personal preseleccionado

El Dpto. de Obtención en coordinación con los Institutos de Formación, publican en la página Web la lista del personal que han sido preseleccionados para continuar en el proceso de selección, y con esta actividad finaliza el proceso de Reclutamiento.

3.1.2.5.4. Tomar conocimiento de los Planes remitidos por las Direcciones.

El Dpto. de Obtención, toma conocimiento de los Planes remitidos por las Direcciones involucradas en el proceso, para realizar la aplicación de las pruebas de selección y análisis en los diferentes Institutos de Formación.

3.1.2.5.5. Aplicar los exámenes médicos

La Dirección de Sanidad, por intermedio de las Unidades de Salud Militar, aplicará los exámenes médicos a los ciudadanos candidatos a ingresar a los Institutos de Formación de acuerdo con lo planificado, luego determina los resultados de la evaluación médica y una vez realizado el análisis por la Junta de Reclutamiento y Selección, remite al Dpto. de Obtención los listados de los ciudadanos idóneos y no idóneos, y en coordinación con los Institutos de Formación publicarán los resultados en la página Web.

3.1.2.5.6. Aplicar los exámenes psicológicos

El Departamento de Salud Integral por intermedio de la Sección Psicosocial de la Dirección de Sanidad de la F.T., aplica los exámenes Psicológicos a los ciudadanos candidatos a ingresar a los Institutos de Formación, posteriormente remitirá al Dpto. de Obtención los listados de los ciudadanos idóneos y no idóneos para continuar en el proceso.

3.1.2.5.7. Aplicar los exámenes académicos

El Departamento de Planificación Educativa del Comando de Educación y Doctrina de la F.T. en coordinación con los Departamentos de Enseñanza de los Institutos de Formación (ESMIL, ESFORSFT y EIWIAS) y/o Unidades designadas, aplican los exámenes y remiten los resultados obtenidos al Dpto. de Obtención de la Dirección General de Recursos Humanos de la F.T., para la consolidación respectiva.

3.1.2.5.8. Aplicar los exámenes físicos

La Sección de Cultura Física del Comando de Operaciones Terrestre en coordinación con las Secciones de Educación Física de los Institutos de Formación y/o

Unidades designadas, aplican los exámenes físicos y remiten los resultados al Dpto. de Obtención, para la consolidación respectiva.

3.1.2.5.9. Consolidar los resultados de exámenes y enviar los reportes a la Junta de Reclutamiento y Selección

El Dpto. de Obtención en coordinación con la Junta de Reclutamiento y Selección, consolidan los resultados obtenidos en los diferentes exámenes aplicados (Médicos, Psicológicos, Académicos y Físicos).

3.1.2.5.10. Analizar resultados de los exámenes aplicados para determinar los cupos establecidos

La Junta de Reclutamiento y Selección, una vez culminado la aplicación de cada uno de los exámenes (Médicos, Psicológicos, Académicos y Físicos), analiza los resultados de las pruebas para determinar los cupos establecidos en el orgánico y reportar los listados del personal de ciudadanos aprobados, luego remite el informe final al Dpto. de Obtención para conocimiento y trámite correspondiente.

3.1.2.5.11. Remitir reporte de aprobados

El Departamento de Obtención, remite los reportes del personal aprobado en las pruebas de selección a la (ESMIL, ESFORST y EIWIAS) debidamente legalizados por el Director General de Recursos Humanos de la Fuerza Terrestre, a los Institutos de Formación para conocimiento.

3.1.2.5.12. Realizar las entrevistas personales

Los Institutos de Formación en coordinación con la Sección de Psicología Educativa, realizan la entrevista personal a los Aspirantes a Oficiales y Voluntarios, utilizando los siguientes formatos “Entrevista aplicada al Aspirante” de código TH1.2.2.F1, “Entrevista aplicada al padre y/o apoderado” de código TH1.2.2.F2, y “Entrevista aplicada al personal de Soldados de la F.T” de código TH1.2.2.F3, luego elaboran el reporte de aprobados y remite al Dpto. de Obtención, con esta actividad los Institutos de Formación finalizan el proceso de selección.

3.1.2.5.13. Receptar reporte de aprobados

El Dpto. de Obtención, recepta los reportes del personal de Aspirantes a Oficiales y Voluntarios aprobados en las entrevistas personales-apoderados, y toma conocimiento, para luego disponer a los Institutos de Formación, que publiquen en la página Web, para información de los interesados.

3.1.2.5.14. Publicar resultados de aprobados

Los Institutos de Formación, hacen conocer al interesado los resultados mediante la pagina Web, y con un reporte del personal aprobado que son colocados en los informativos de los Institutos de Formación respectivos.

3.1.2.5.15. Conocer resultados finales

El ciudadano una vez que conoce que ha sido aprobado como alumno para el curso de Aspirante a Oficial o Voluntario, procede a cancelar los aranceles tanto de matrícula, pensión y fondo de compensación en el caso de Aspirantes a Oficiales.

3.1.2.5.16. Receptar aranceles y remitir reporte

Las Unidades Financieras de los Institutos de Formación, reciben la copia del pago de aranceles realizados por los Aspirantes a Oficiales o Voluntarios en las diferentes entidades bancarias y remiten los reportes al Dpto. de Obtención para conocimiento.

3.1.2.5.17. Remitir listado para publicar el alta en la Orden General

El Dpto. de Obtención, remite mediante oficio y en magnético el listado del personal de Aspirantes a Oficiales y Voluntarios aprobados, para la publicación del alta en la Orden General respectiva.

3.1.2.5.18. Registrar en el sistema

El Dpto. de Obtención, verificará en la Orden General de la Fuerza Terrestre, la publicación del alta del personal de Aspirantes a Oficiales y Voluntarios, luego se registra en el sistema.

3.1.3. ESPECIFICACIÓN DE REQUERIMIENTOS.

3.1.3.1. INTRODUCCIÓN

3.1.3.1.1. PROPÓSITO

El propósito de este documento es recopilar los requisitos funcionales del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército.

3.1.3.1.2. ALCANCE

Este documento se centra en mostrar la información a ser manejada, de la misma manera se mostrarán los procesos a ser automatizados por el Sistema.

3.1.3.2. DESCRIPCIÓN

3.1.3.2.1. PRODUCTOS A REALIZAR

Se desarrollará el Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército el cual contendrá los siguientes módulos:

- Administración
- Procesos
- Consultas y Reportes

3.1.3.2.2. LIMITACIÓN DE RESPONSABILIDADES

El desarrollo del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército no contemplará los siguientes ítems:

- Un módulo personalizado de seguridades. Para manejar las seguridades se utilizará el módulo de seguridades que se encuentra desarrollado para la Dirección de Comunicaciones del Ejército.
- No se realizará el diseño de la imagen institucional¹ de la Dirección de Recursos Humanos o de las unidades donde se implantará el sistema
- No se realizará un portal Web personalizado para la Dirección de Recursos Humanos del Ejército. Lo que se entregará es una página de acceso al sistema, a la cual se puede acceder vía Intranet o vía Internet a través de un link designado por la DISICOME con los permisos y perfiles entregados por la Dirección de Recursos Humanos del Ejército.
- La carga inicial y depuración de datos estará a cargo del personal técnico de la Dirección de Recursos Humanos.
- La información estará centralizada de acuerdo a la infraestructura proporcionada por la Dirección de Recursos Humanos del Ejército para el sistema, en ninguna de las unidades se tendrá un servidor local con la información referente a la misma.
- No se realizará la creación, modificación, actualización u observación en los procesos normados por Dirección de Recursos Humanos del Ejército. El desarrollo del software se basa en los procesos definidos en la Norma vigente difundida por la Dirección de Recursos Humanos del Ejército.
- No es responsabilidad del proyecto, el proporcionar el canal de comunicaciones, equipos ni licenciamiento de software (Base de Datos, Servidor de Aplicaciones, Sistema Operativo) en las Direcciones y/o Unidades en las que se implantará el Sistema, esta responsabilidad recae

¹ Logos, escudos, slogan

sobre la DISICOME y la Dirección de Recursos Humanos del Ejército, la cual se encargará de este requisito.

- El proyecto se encargará de capacitar al personal designado por la DISICOME y la Dirección de Recursos Humanos para que éste efectúe la instalación, capacitación e implantación del sistema desarrollado en las unidades a ser consideradas dentro del plan piloto².
- Para la puesta en producción del Sistema, se usará el servidor de Aplicaciones proporcionado por la DISICOME.
- No se realizará el desarrollo de módulos que no se encuentren contemplados dentro del presente documento de requerimientos.
- La fase de migración de datos estará a cargo de los técnicos del SIPER, con el soporte de los técnicos de la ESPE, enmarcados dentro de un cronograma de actividades

3.1.3.3. DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS

3.1.3.3.1. ADMINISTRACIÓN

- **OBJETIVOS**

En este módulo, el Administrador podrá realizar la configuración de los parámetros generales del Sistema de Administración de Archivos para su debido uso y funcionamiento.

- **DETALLES DE FUNCIONALIDAD**

La información que se registrará en este módulo será la siguiente:

- Administración Actividades

- Guarda los datos de los respectivas Actividades.
- Todos los datos detallados son obligatorios.
- No se permitirán dos Departamentos con el mismo Código de Identificación o Descripción.

- Administración Cronograma

- Guarda los datos del Cronograma que se seguirá durante el periodo de Ingreso y Selección.
- Código, Descripción.
- Todos los datos detallados son obligatorios.
- No se permitirán dos Actividades en el Cronograma con el mismo

² Dirección de Recursos Humanos

Código de Identificación.

- Administración Documentos Requeridos
 - Guarda los datos de los Documentos Requeridos a cada uno de los Candidatos.
 - Código, Descripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Documentos Requeridos con el mismo Código de Identificación o Descripción.

- Administración Deportes y Habilidades
 - Guarda los datos de los Deportes y Habilidades que posee un Candidato.
 - Código, Descripción
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Deportes y Habilidades con el mismo Código de Identificación o Descripción.

- Administración Tipo Mención Honorífica
 - Guarda los datos del Tipo Mención Honorífica.
 - Código, Descripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Tipos Mención Honorífica con el mismo Código de Identificación o Descripción.

- Administración Partido Político
 - Guarda los datos del Partido Político.
 - Código, Descripción, Siglas.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Partidos Políticos con el mismo Código de Identificación, Descripción o Siglas.

- Administración Niveles de Calificación
 - Guarda los datos de los Niveles de Calificación.
 - Código, Descripción, Nivel.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Niveles de Calificación con el mismo Código

de Identificación o Descripción.

- Administración Tipo Referencia
 - Guarda los datos de Tipo Referencia.
 - Código, Descripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Tipos Referencia con el mismo Código de Identificación o Descripción.

- Administración Tipo Conocimiento
 - Guarda los datos de Tipo Conocimiento.
 - Código, Descripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Tipo Conocimiento con el mismo Código de Identificación o Descripción.

- Administración Factores Evaluar
 - Guarda los datos de Factores Evaluar.
 - Código, Descripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Factores Evaluar con el mismo Código de Identificación o Descripción.

- **DETALLES DE ACCESO Y SEGURIDAD**

A ésta área, tendrán acceso los usuarios y el personal que tengan o hayan sido definidos con el perfil de administradores. El acceso se lo realizará especificando - y previa validación - del nombre de usuario y la respectiva contraseña.

Todos los usuarios autorizados, tendrán acceso a las opciones definidas en el perfil, pudiendo acceder a las opciones de mantenimiento del sistema.

3.1.3.3.2. PROCESOS

- **OBJETIVOS**

En este módulo, el personal de Obtención, podrá hacer uso de las siguientes opciones:

- Registro de los candidatos.
- Recepción y Verificación de la documentación.

- Registro de las Pruebas Psicológicas.
- Registro de Pruebas Académicas.
- Registro de Pruebas Médicas.
- Registro de Pruebas Físicas.

- **DETALLES DE FUNCIONALIDAD**

Dentro de éste módulo, se registrará la siguiente información por cada proceso:

- Registro de los candidatos.
 - Guarda los datos principales del Candidato.
 - Código, Fotografía, Cedula, Apellidos, Nombres, Dirección, Teléfono, Fecha de Nacimiento, Libreta Militar, Estatura, Código Interno, Fecha de Admisión, Fecha de Presentación, Password, Cabeza , Cuello, Calzado, Peso, email, Tipo de Sangre, Unidad, Provincia, Cantón, Parroquia, Periodo, Estado Civil, Sexo, Color de Ojos, Tipo Licencia de Conducir.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos Candidatos con el mismo Código de Identificación o Nombre.

- Validación Documentos Candidato.
 - Guarda la información de verificación de cada uno de los candidatos en lo referente a verificación de cada uno de los documentos e información suministrada por ellos a cada uno de las personas encargadas de la verificación..
 - Código, Código del Documento, Fecha que presenta el documento, Número copias del documento, Cumple el requisito, Novedades encontradas en el documento, Observaciones que se agreguen al documento entregado, Código del candidato al que pertenece el documento, Periodo en el que se presenta ese documento, el candidato debe cumplir con la entrega de todos los documentos requeridos en la inscripción.
 - Todos los datos detallados son obligatorios.
 - No se permitirán dos procesos de recepción de documentos con el mismo código de Identificación.

- Proceso de Pruebas Psicológicas.
 - Guarda los datos registrados por el lector óptico de cada una de las pruebas psicológicas que rindieron los candidatos en un periodo de tiempo determinado.
 - Califica cada una de las pruebas psicológicas registradas en el sistema por medio de claves especiales ingresadas en el sistema.
 - Código en letras del test, Código numérico del test, Resultado del test, grupo al que pertenece el candidato, sub grupo al que pertenece el candidato, Fecha en la que registra el resultado.
 - Todos los datos detallados son obligatorios.
 - El registro de las respuestas del test por candidato debe estar relacionado con los test psicológicos y sus respectivas claves.
 - No se permitirán dos Procesos de registro de test por candidato con el mismo código de Identificación.

- Proceso de Pruebas Académicas.
 - Guarda los datos de las pruebas académicas que cada uno de los candidatos debe rendir para poder avanzar en el proceso de selección.
 - Código, Fecha de evaluación, Estado de la evaluación, Tiempo de la evaluación, Calificación que obtuvo en la evaluación.
 - Código, código de la pregunta, Respuesta.
 - Todos los datos detallados son obligatorios.
 - El registro de las pruebas académicas por candidato debe estar relacionado con los pruebas académicas y sus respuestas.
 - No se permitirán dos Procesos de registro de pruebas académicas por candidato con el mismo código de Identificación.

- Proceso de Pruebas Físicas.
 - Guarda los datos de las pruebas físicas que se procedió a realizar a cada uno de los candidatos y que son suministradas por un lector óptico para el ingreso al sistema por medio de archivos planos.
 - Código secuencial, Fecha de prueba física, si es idóneo, salto decisión, Promedio, Marca, Nota, Observaciones, Talento.
 - Fecha, Marca, Minutos, segundos, milisegundos, Baremo, Calificación, Idóneo detalle prueba física.
 - Todos los datos detallados son obligatorios.

- El registro de las pruebas físicas por candidato debe estar relacionado con los tipos de pruebas físicas y sus respuestas.
- No se permitirán dos Procesos de registro de pruebas físicas por candidato con el mismo código de Identificación.

➤ Proceso de Pruebas Médicas.

- Guarda los datos de las pruebas medicas que se les practica a cada uno de los candidatos a las diferentes escuelas de formación del ejercito, estos valores serán ingresados por medio de un aplicativo en el sistema.
- Código, Fecha de prueba médica, Aprueba, Tipo Enfermedad.
- En el Detalle tenemos Código, Fecha, Observación.
- Todos los datos detallados son obligatorios.
- El registro de las pruebas médicas por candidato debe estar relacionado con los tipos de pruebas médicas.
- No se permitirán dos Procesos de registro de pruebas médicas por candidato con el mismo código de Identificación.

• **DESCRIPCIÓN DE PROCESOS:**

➤ Registro de los candidatos.

Se permitirá el registro de los candidatos por medio del portal web durante un periodo determinado, los candidatos deberán registrar toda la información requerida e imprimir la hoja de registro que permitirá posteriormente verificar su registro como también la verificación de los documentos requeridos para su ingreso a las diferentes escuelas de formación del Ejército Ecuatoriano.

➤ Recepción y Verificación de la documentación

- Una vez culminado el periodo de inscripciones los candidatos deben acercarse a los diferentes puntos de verificación de documentos que fueron requeridos en el momento de la inscripción.
- Las personas encargadas de la verificación de los documentos a cada uno de los candidatos registrara en el sistema si cumplen o no con cada uno de los documentos requeridos.
- Una junta especial de selección especificara los parámetros de estatura y edad para verificar que candidatos continúan en el proceso de selección.

- Registro de las Pruebas Psicológicas.
 - Se importa los datos emitidos por un lector óptico de cada una de las pruebas psicológicas dadas por los candidatos a las escuelas de formación.
 - Se califica las pruebas psicológicas de cada uno de los candidatos a las escuelas de formación por medio de las claves correspondientes a los test psicológicos que son utilizados en ese periodo de selección.
 - Se emiten los reportes necesarios con las calificaciones obtenidas por cada uno de los candidatos.
 - Una junta especial de selección analizara los datos mediante parámetros para seleccionar los candidatos más idóneos que continuaran en el proceso.

- Registro de pruebas Académicas.
 - Se debe registrar los bancos de preguntas por cada una de las materias que se tomaran las pruebas académicas a los candidatos a las escuelas de formación.
 - Se debe generar el examen en base al banco de preguntas de manera aleatoria se generara el examen de cada una de las materias.
 - Se procede a tomar el examen a cada uno de los candidatos a las escuelas de formación y al culminar el tiempo el sistema calificara automáticamente indicando el puntaje e imprimiendo un reporte con la nota obtenida por el candidato.
 - Una junta especial analizara los resultados mediante parámetros para seleccionar a los candidatos más idóneos para continuar en el proceso.

- Registro de Pruebas Médicas.
 - Se registra cada una de las pruebas medicas tomada a cada uno de los candidatos a las escuelas de formación por parte de los médicos designados para ese evento.
 - Una junta especial analizara las pruebas médicas para seleccionar a los candidatos más idóneos para seguir en el proceso.

- Registro de las Pruebas Físicas.
 - Se genera los grupos de los candidatos que están aptos para rendir las pruebas físicas.
 - Se generan los subgrupos de los candidatos que fueron seleccionados en un grupo para rendir las diferentes pruebas físicas.
 - Se procede a tomar las pruebas físicas a cada uno de los candidatos según el grupo y subgrupo al que pertenezcan.
 - Se emiten los reportes con los resultados de las pruebas físicas que fueron tomadas a cada uno de los candidatos seleccionados.

3.1.3.3.3. DETALLES DE ACCESO Y SEGURIDAD

A ésta área, tendrán acceso los usuarios y el personal que tengan o hayan sido definidos con el perfil de administradores, analistas o jefe de departamento. El acceso se lo realizará especificando - y previa validación - del nombre de usuario y la respectiva contraseña.

Todos los usuarios autorizados, tendrán acceso a las opciones definidas en el perfil, pudiendo acceder a las opciones de registro de información del sistema.

El acceso por parte del personal a las opciones de almacenamiento de información, se lo realizará a través de un proceso de autenticación con contraseña asignada por el administrador.

Los niveles superiores tendrán permisos de visualización e impresión.

3.1.3.3.4. CONSULTAS Y REPORTES

- **OBJETIVOS**

Es el módulo encargado de mostrar los reportes del sistema que darán la información necesaria a los usuarios.

- **DETALLES DE FUNCIONALIDAD**

Candidatos Inscritos.

Listado de Documentación Aprobada.

Estadístico por departamento.

- Pruebas Psicológicas.
 - Reporte de Pruebas Psicológicas.
 - Reporte Numérico por rango.
 - Detalle Psicológicas Resultados.
 - Reporte final Psicológicas.

- Pruebas Académicas.
 - Reporte Pruebas Académicas.
 - Detalle Académicas por Materias y Promedio.
- Pruebas Médicas.
 - Listado Pruebas Medicas Fases.
- Pruebas Físicas.
 - Grupos de Evaluación.
 - Subgrupos de Evaluación.
 - Lista de Evaluación.
 - Candidatos Atletas.
 - Candidatos Presentes y Ausentes Pruebas Físicas.
 - Candidatos No Saltaron, No Nadaron.
 - Resultados Pruebas Físicas.
- Reportes Junta Final.
 - Detalle Junta Final.
 - Listado Junta Final.

- **DETALLES DE ACCESO Y SEGURIDAD:**

A ésta área, tendrán acceso los usuarios y el personal que tengan o hayan sido definidos con el perfil de administrador, analistas o jefe de departamento. El acceso se lo realizará especificando y previa validación del nombre de usuario y la respectiva contraseña.

Todos los usuarios autorizados por el administrador del sistema, tendrán acceso a las opciones definidas en el perfil.

CAPÍTULO IV

4.1. DESARROLLO DEL SUBSISTEMA DE INGRESO Y SELECCIÓN.

4.1.1. ANÁLISIS Y DISEÑO DEL SISTEMA

4.1.1.1. ARQUITECTURA DE SOFTWARE

4.1.1.1.1. INTRODUCCIÓN

- **PROPÓSITO**

La Arquitectura de Software (también denominada Arquitectura Lógica) es el arte de proyectar y construir sistemas informáticos para pequeñas, medianas y grandes empresas; tanto del sector público como del sector privado.

La Arquitectura de Software, permite crear un plan unificado y coherente; basado en patrones y abstracciones que proporcionan el marco de referencia para la construcción de cualquier sistema de información.

Es así, que el presente documento tiene como finalidad proporcionar una descripción comprensible de la arquitectura de los sistemas que serán desarrolladas dentro del Proyecto de Desarrollo de Software ESPE 2010, para posteriormente ser implementados en la Dirección de Recursos Humanos del Ejército Ecuatoriano. Aquí se recogerán, analizarán y definirán los requerimientos, detalles y características de la arquitectura que se manejará centrándose en las necesidades (hardware y software) de la arquitectura que se implementara.

- **ALCANCE**

En este documento se establecerán las reglas sobre las que se va a diseñar la arquitectura del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, así como de los requerimientos básicos de la misma; tanto de software como de hardware.

- **DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES**

- **Arquitectura**

Estructura fundamental de los componentes de un sistema, las relaciones entre ellos, su entorno y los principios y guías que gobiernan su diseño y evolución.

- **CORBA**(Common Object Request Broker Architecture)

Es una arquitectura estándar de objetos distribuidos desarrollado por el Object Management Group (OMG), establecida en 1989, incluye a más de 800 compañías como IBM, SUN, Oracle, Svbase, etc...., pero no incluye a Microsoft, DECOM compete con CORBA, es de vendedores independientes, separa la interfaz de la implementación.

- **JSP:** (Java Server Pages)

Es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo.

Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

- **Nivel**

Es la forma en que las capas lógicas se encuentran distribuidas de forma física.

- **ORB** (Object Request Broker)

Maneja todas las comunicaciones entre los objetos con los métodos situados en máquinas remotas, a través de una red:

 1. Acepta los requerimientos desde los clientes
 2. Localiza y activa los objetos destinos
 3. Identifica la máquina que ejecuta el objeto servidor
 4. Pide una conexión al ORB de la maquina del servidor
 5. Direcciona los requerimientos y recibe las respuestas

- **n-capas**

Capas especializadas absolutamente en la funcionalidad que deben brindar (procesamiento en las reglas de negocios o presentación de datos en la capa cliente) sin importar cuál es el origen de los datos procesados. Pueden encontrarse físicamente en lugares diferentes, están conformadas por librerías con código especializado que son invocadas remotamente.

➤ **Servlets**

Pequeños programas en Java que se ejecutan de forma persistente en el servidor, y que por lo tanto, tienen una activación muy rápida y una forma más simple de hacerlo.

➤ **Servidor de Aplicaciones**

Programa que contiene la implementación de uno o más tipos de objetos. Provee un ambiente para almacenar los objetos, instancia los objetos CORBA, aplica la seguridad, asimismo maneja las Transacciones, el Failover y el balanceo de carga.

➤ **Servidor Remoto**

Un servidor remoto agrupa servicios Web que están instalados en el mismo equipo y requieren el mismo tipo de autenticación.

➤ **Skeleton**

Implementa los mecanismos para la invocación de los requerimientos hacia el servidor y que puedan ser descriptados y direccionados a los métodos correctos. Pone en espera a la implementación del objeto en tiempo de ejecución para el ORB, desempaqueta los argumentos de los métodos, dispone los métodos para la instancia de la implementación del objeto, también es conocido como la clase base de la implementación.

➤ **WEB**

La World Wide Web consiste en ofrecer una interface simple y consistente para acceder a la inmensidad de los recursos de Internet. Es la forma más moderna de ofrecer información. La información se ofrece en forma de páginas electrónicas.

• **REFERENCIAS**

- Documento de los modelos de casos de uso del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército.
- Documento de los modelos en términos de clases del análisis realizado.

• **DESCRIPCIÓN**

Este documento establecerá las normas básicas y definiciones que han de ser utilizadas en la Arquitectura del Sistema de Ingreso y Selección de la

Dirección de Recursos Humanos del Ejército, así como también la plataforma que deberá ser montada y desde la cual se correrán todas las aplicaciones.

4.1.1.1.2. REPRESENTACIÓN DE LA ARQUITECTURA

En la actualidad, la Dirección de Recursos Humanos no cuenta con un sistema de Ingreso y Selección de Candidatos a Aspirantes a las escuelas de formación del Ejército³ Ecuatoriano, sino que cada una de las escuelas cuenta con subsistemas aislados que realizan parte del proceso de Ingreso y Selección.

Por tal motivo y ante la necesidad de integrar, actualizar la tecnología y globalizar la información, la Dirección de Sistemas y Comunicación del Ejército⁴ con apoyo de la Escuela Politécnica del Ejército⁵; pondrán en marcha la planeación, construcción e implantación del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, como parte del Proyecto de Desarrollo de Software ESPE 2010.

El mencionado sistema, estará conformado de los siguientes módulos:

- Módulo de Mantenimiento
- Módulo de Procesos
- Módulo de Reportes

Las actividades que se implementaran dentro de cada uno de estos módulos serán las siguientes:

- Módulo de Mantenimiento
 - Mantenimiento general del Sistema
- Modulo de Procesos
 - Registro de los candidatos.
 - Recepción y Verificación de la documentación.
 - Registro de las Pruebas Psicológicas.
 - Registro de Pruebas Académicas.
 - Registro de Pruebas Médicas.
 - Registro de Pruebas Físicas.
 - Parametrización para el ingreso y selección de las Juntas de Calificación.
- Modulo de Reportes
 - Reportes solicitados por los usuarios

³ ESFOSE: Escuela de Formación de Soldados.

⁴ DISICOME: Dirección de Sistemas y Comunicación del Ejército

⁵ ESPE

La construcción y posterior implantación del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, será diseñado en el modelo de n-capas. Las herramientas de desarrollo, servidor de aplicaciones y Base de Datos a utilizarse serán:

- **Herramientas de desarrollo**
 - NetBeans como IDE de desarrollo de aplicativos WEB
- **Servidor de Aplicaciones WEB**
 - GlassFish
- **Base de Datos**
 - Oracle

Las herramientas tecnológicas seleccionadas, han sido elegidas debido a las ventajas que los mismos ofrecen para el desarrollo, seguridad y portabilidad de los aplicativos.

Tanto los servidores de base de datos como los servidores de aplicaciones del Sistema de Ingreso y Selección, estarán ubicados físicamente en el Edificio de la Comandancia de la Fuerza Terrestre y la administración de los mismos es de exclusiva responsabilidad de la Dirección de Sistemas y Comunicación del Ejército⁶. A continuación se referencia en las figuras 4.1 y 4.2 la vista de Arquitectura desde dos puntos de vista.

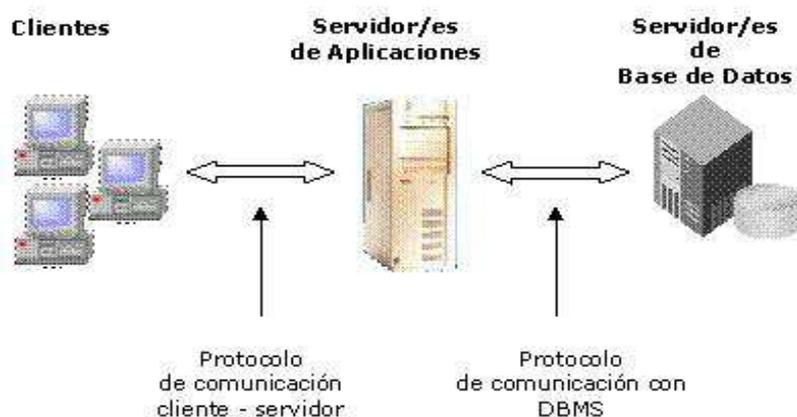


Figura 4. 1 Vista de la Arquitectura desde el Punto de Vista del Usuario

⁶ DISICOME: Dirección de Sistemas y Comunicación del Ejército

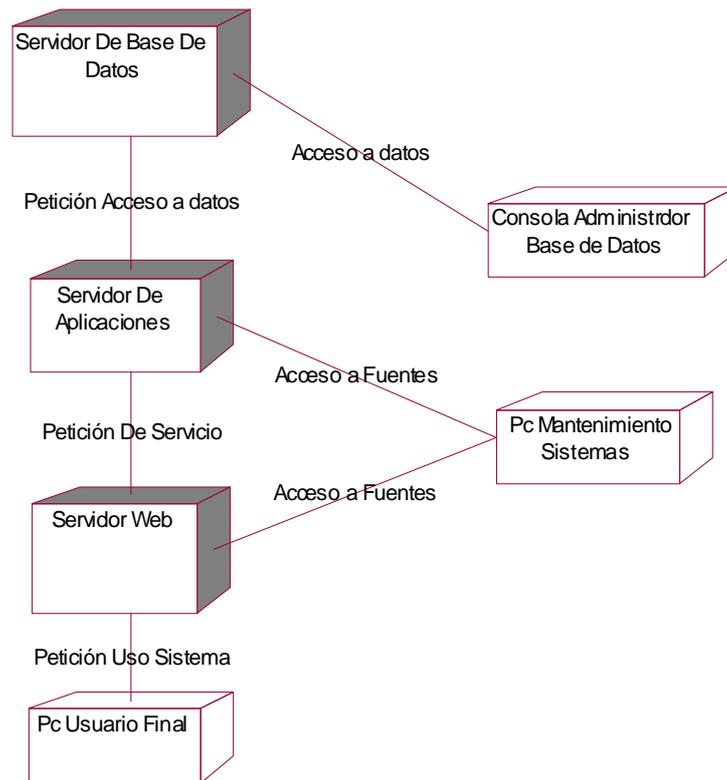


Figura 4. 2 Vista de la Arquitectura desde el Punto de Lenguaje de Modelamiento Unificado

4.1.1.1.3. RESTRICCIONES Y METAS DE LA ARQUITECTURA

a) RESTRICCIONES

La implantación de los sistemas tanto distribuidos como transaccionales no podrá ser realizada si se dieran las siguientes situaciones:

- 1) Fallos en el sistema de comunicaciones: El Canal de Comunicación debe contemplar la integración de todas las unidades ubicadas geográficamente en diferentes lugares a través de todo el territorio nacional. El no tener el sistema de comunicación cien por ciento operable, dificultaría enormemente la implantación de los sistemas a ser desarrollados.
- 2) Falta de apoyo de las autoridades: Uno de los puntos más importantes que facilitara la implantación del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, es el apoyo que las autoridades brinden al mismo. La falta de apoyo de las autoridades al personal técnico delegado por la ESPE en tareas tales como: Entrega de requerimientos, no asistencia del personal

civil y militar a los talleres de capacitación y difusión de los sistemas o cualquier otra acción que imposibilite el normal desarrollo del proyecto imposibilitara el éxito en el cumplimiento de sus metas.

- 3) Continuidad de las personas involucradas en el desarrollo e implantación de los sistemas. La falta de continuidad del personal militar⁷ o del personal civil imposibilitaría enormemente la feliz culminación de la implementación de los sistemas.

b) METAS

El desarrollo de aplicaciones Web, tiene la finalidad de:

- 1) Construir software que aporte a las autoridades de la Dirección de Recursos Humanos del Ejército y del departamento de Ingreso y Selección en la toma de decisiones.
- 2) Propiciar un desarrollo orientado a demostraciones periódicas de la funcionalidad de los productos desarrollados.
- 3) Crear interfaces que permiten una cooperación eficiente entre los encargados del diseño de los sistemas y los encargados de realizar la implementación de los procesos.
- 4) Tener clara la etapa de diseño e implementación de las estructuras de software de alto nivel. Con la arquitectura de software implementada, se pretende realizar el ensamblaje de un cierto número de elementos arquitectónicos de forma adecuada para satisfacer de mejor manera la funcionalidad y requerimientos de desempeño de un sistema. De igual manera se pretende implementar los requerimientos no funcionales tales como: la confiabilidad, escalabilidad, portabilidad y disponibilidad.

4.1.1.1.4. VISTA DE LOS CASOS DE USO⁸

Esta sección lista los casos de uso o escenarios desde el modelo de caso de uso si estos representan algún significado, la funcionalidad central del sistema final, o si ellos tienen una cobertura arquitectónica grande, o constituyen un punto delicado de la arquitectura.

⁷ Los pases a los que son sujetos los militares en el transcurso del año

⁸ Para mayor detalle se debe revisar el documento de casos de uso

- **MÓDULO DE MANTENIMIENTO**
 - **MANTENIMIENTO DE LAS TABLAS PRINCIPALES**

- **MÓDULO DE PROCESOS**
 - Registro de los candidatos.
 - Recepción y Verificación de la documentación.
 - Registro de las Pruebas Psicológicas.
 - Registro de Pruebas Académicas.
 - Registro de Pruebas Médicas.
 - Registro de Pruebas Físicas.
 - Parametrización para el ingreso y selección de las Juntas de Calificación.

- **MÓDULO DE REPORTES**
 - Generación de los Reportes Solicitados por el Usuario

4.1.1.1.5. VISTA LÓGICA

- **DESCRIPCIÓN**

Las fases de diseño, desarrollo y posterior implantación de los módulos que conformaran el Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, será realizado mediante la distribución de sus funcionalidades con la finalidad de:

- Mejorar el rendimiento
- Escalabilidad de los aplicativos
- Optimización de los recursos
- Mejorar el mantenimiento del sistema, entre otros

- **PAQUETES ARQUITECTÓNICOS**

A continuación se referencia en las figuras 4.3 los Paquetes Arquitectónicos.

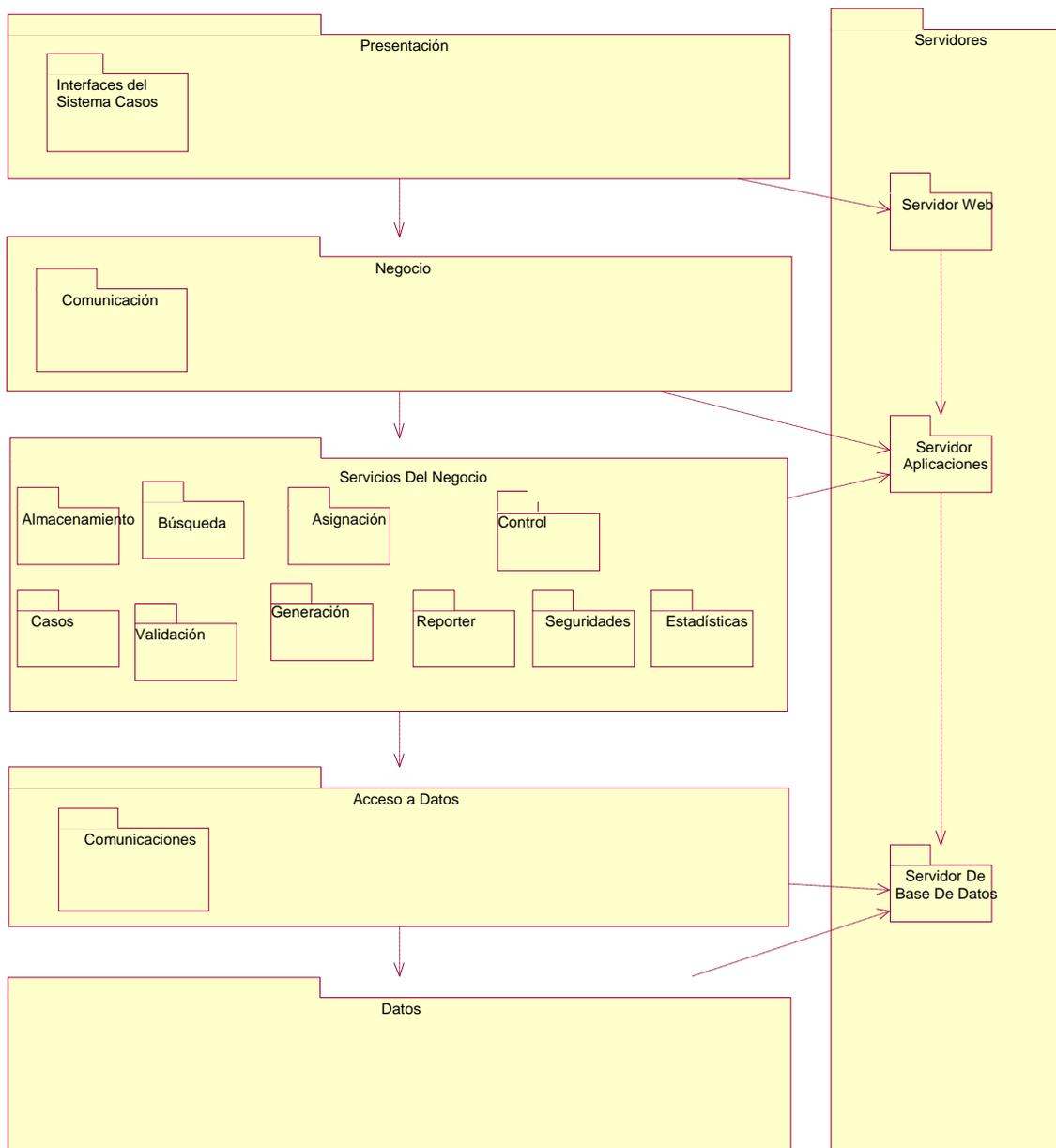


Figura 4. 3 Descripción de Paquetes Arquitectónicos

- **DESCRIPCIÓN DE CAPAS**

- **INTERFAZ**

La interfaz de usuario (también conocida como Presentación) es donde el usuario responsable de proceso interactuara con los aplicativos desarrollados. Entre algunas de las tareas a desarrollarse en la interface, se incluyen tareas tales como ingreso de datos, presentación

de los datos, mantenimiento de los datos existentes, comienzo de los procesos de búsqueda, entre otros. La capa de la presentación no contendrá la lógica de negocio o la lógica de la transacción. El contenido de ésta capa debe ser mínima, con la finalidad de prevenir que las modificaciones o cambios sean realizadas en la capa que contenga la lógica del negocio. A continuación se detallaran algunas características que se incluirán en la interfaz:

- Presionar el botón de búsqueda
- Ingreso de datos de un caso
- Buscar datos en una lista de selección

- **REGLA DEL NEGOCIO**

En esta capa, es donde las decisiones del negocio deben ser resueltas. Por definición, “su estructura se encuentra basada en un juego de funcionamiento diseñado para satisfacer las necesidades de negocio”. Esta capa se encargara de validar la información mientras se la va procesando, así como ejecutar acciones embebidas sobre la base de datos dependiendo de las solicitudes que los usuarios la vayan realizando. Algunos ejemplos de la lógica del negocio incluyen:

- Conexión a la Base de Datos
- Autenticación de un usuario autorizado
- Ejecutar el proceso de búsqueda.
- Generación de código clave

- **DATOS**

Es la capa encargada del manejo de los datos. Se incluyen tareas como la definición, manipulación y almacenamiento de los datos. En esta capa se albergara principalmente los servicios de la base de datos destinados a aumentar al máximo su rendimiento. En esta capa se incluirán tareas como:

- Generación de códigos clave
- Ejecución de procedimientos almacenados
- Ejecución de funciones de base de datos

- **NIVELES**

Los niveles de una aplicación dependen del tipo de arquitectura implementada.

- **Arquitectura Distribuida**

- **Nivel Web**

Es el nivel y ambiente definido para el usuario final, todos los subsistemas desarrollados e implementados en la WEB para los usuarios del Sistema de Administración de Archivos de la Dirección de Inteligencia del Ejército, podrán ser accedidos desde la Intranet.

El acceso a los sistemas WEB será válido; siempre y cuando la autenticación sea correcta y la configuración de la red lo permita.

El objetivo de las páginas WEB que se desarrollaran en este nivel, es crear una presencia sólida y eficaz en la Intranet, de tal manera que el sitio se convierta en una herramienta para los usuarios del Sistema de Administración de Archivos de la Dirección de Inteligencia del Ejército. Para que un sitio cumpla con este objetivo, debe de tener las siguientes características:

- Contar con información actualizada
- Tener información completa
- Presentación adecuada de la información

- **Nivel Aplicación/Interface**

El nivel aplicación y/o interface se constituye como un repositorio de librerías, archivos, recursos u otros. Los objetos que forman la interface, no son visibles para el usuario final. Cualquier cambio realizado en éste nivel afectara directamente a lo que el usuario final vera. El acceso a este nivel solo lo podrá realizar el desarrollador o programador, así como el administrador de los sistemas delegado por las autoridades de respectivas.

- **Nivel de la Capa del Negocio**

En éste nivel se realizará la implementación y parametrización de las reglas del negocio. El contenido de la Capa de Negocio es visible únicamente para el Administrador del sistema.

- **Nivel de Base de Datos**

En este nivel se encuentran todos los datos físicos que se

encuentran almacenados en el servidor de base de datos. Estos datos así como las estructuras físicas de la Base de Datos no son visibles para los usuarios, siendo visibles únicamente para el administrador de la base de datos.

4.1.1.1.6. TAMAÑO Y RENDIMIENTO

Los beneficios que se pretende alcanzar al desarrollar sistemas WEB son los siguientes:

- Proveer a los usuarios de la Unidad Rectora⁹, y unidades ejecutoras; la información generada como resultado de sus actividades así como llegar a realizar – solo si esto fuese necesario – la actualización a medida que se van desarrollando o ejecutando nuevas tareas. Esto es mucho más sencillo y económico que imprimir nuevos catálogos o hacer publicaciones cada vez que se lo requiera.
- Otorgar información detallada y específica acerca de las actividades realizadas.
- Un sistema Web estará disponible fácilmente para cualquier persona autorizada y los datos que ellos desean almacenar, se guardarán de manera centralizada; de esta manera se evitará que los datos se dupliquen o se corrompan.
- El desarrollo de un sistema Web está orientado a realizar un balance de la necesidad de una solución tecnológica con la realidad de las personas que lo usarán y la facilidad administrativa de utilizarlo. Si esto no se hace así, la solución desarrollada podría ser la mejor desde el punto de vista tecnológico, pero un fracaso total a la hora de convencer a los usuarios a tomar ventajas de los sistemas o podría causar tal demanda de soporte por parte de los administradores del sistema que se vuelva impráctico desde el punto de vista económico.

4.1.1.1.7. CALIDAD

Los sistemas desarrollados como parte del Proyecto Desarrollo de Software ESPE 2011, basado en la implementación de la arquitectura Web deberán ser software de alta calidad y que deberán presentar las siguientes características

⁹ Dirección de Inteligencia del Ejército

1. **Confiabilidad:** Los sistemas desarrollados y que necesiten acceder a los datos almacenados en el servidor de Base de Datos, solo serán validos si estos son autenticados a través del usuario y contraseña.
2. **Seguridad:** Para el mejoramiento de la seguridad se deberá crear procesos que permita el manejo de perfiles y niveles de acceso. Esto permitirá que el software sea seguro y difícil de violar.
3. **Portabilidad:** Los sistemas Web permitirán sistemas portables, que permitan una funcionalidad independientemente del sistema operativo que posea al equipo del usuario.
4. **Privacidad:** La información que maneje el Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército, no saldrá fuera de la red interna de la misma con excepción del modulo de Inscripción de los candidatos a las Escuelas de Formación y Reportes de Aceptados y Rechazados.

4.1.1.2. ESPECIFICACIÓN DE CASO DE USO

4.1.1.2.1. INTRODUCCIÓN

a) PROPÓSITO

El propósito de este documento es dar un bosquejo general del Sistema de Ingreso y Selección "SISE" en forma de Casos de Uso y su Especificación.

De la misma manera específica como el Sistema interactúa con los usuarios y con otros Sistemas de la Fuerza.

b) ALCANCE

Este documento muestra los Casos de Uso que serán automatizados con el desarrollo del Sistema de Ingreso y Selección "SISE" de la Dirección de Recursos Humanos del Ejército, de la misma manera muestra una especificación más detallada de cada uno de ellos.

c) DESCRIPCIÓN

A continuación se especifican los Casos de Uso que dan una visión general del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército.

- **MÓDULO DE ADMINISTRACIÓN**
 - **ADMINISTRACIÓN DOCUMENTOS REQUERIDOS**

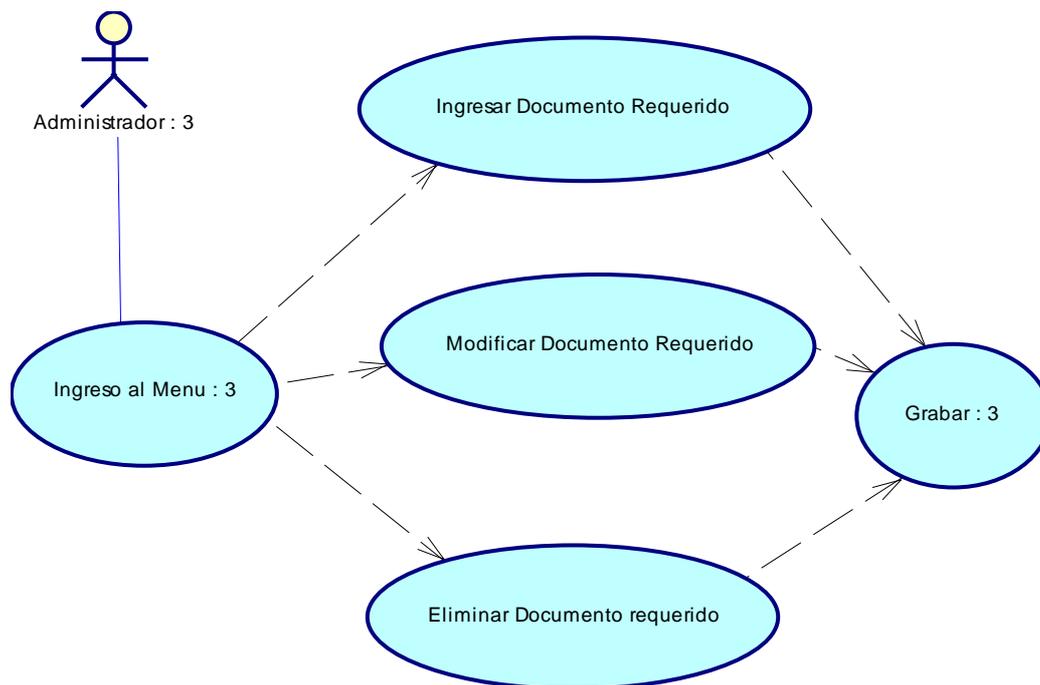


Figura 4. 4 Caso Uso Administración Documentos Requeridos

Caso Uso Documentos Requeridos	
Nombre:	Administración Documentos Requeridos
Descripción:	Usuario Administrador del Sistema, permite almacenar los datos de los Documentos Requeridos en el Periodo de Ingreso y Selección. Es una actividad no frecuente que será ingresada en un inicio por el Administrador del Sistema.
Requerimiento:	Guardar los Datos del Documento Requerido .
Precondición:	El Administrador este autenticado en el sistema.
Flujo Normal:	
El proceso de Administración Documentos Requeridos comprende las siguientes tareas: ingresar, buscar, modificar y eliminar. A continuación se describe cada uno de los procesos:	

Caso Uso Documentos Requeridos	
Actor	Sistema
<p>1. Ingresar Documento Requerido:</p> <p>1.1. Usuario escoge ingresar Documento Requerido.</p> <p>1.2. Usuario ingresa la información en los campos correspondientes.</p> <p>1.3. Usuario da clic en la opción Grabar.</p> <p>2. Buscar Documento Requerido:</p> <p>2.1. Usuario escoge buscar Documento Requerido.</p> <p>2.2. Usuario ingresa la información del Documento Requerido.</p> <p>2.3. Usuario da clic en el botón Buscar.</p> <p>3. Modificar Documento Requerido</p> <p>3.1. Usuario escoge Modificar Documento requerido.</p> <p>3.2. Usuario selecciona el Documento Requerido a modificar.</p> <p>3.3. Usuario modifica la información en los campos correspondientes.</p> <p>3.4. Usuario da clic en la opción Actualizar.</p> <p>4. Eliminar Documento Requerido</p>	<p>1. Ingresar Documento Requerido:</p> <p>1.1. El sistema despliega la pantalla en la que se debe registrar la siguiente información:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. Asignado automáticamente por el sistema. • Descripción: Campo Alfanumérico de hasta 50 caracteres. <p>1.2. El sistema valida que la información ingresada por el usuario sea correcta.</p> <p>1.3. Los datos son guardados en la base y se presenta al usuario una pantalla de confirmación de que han sido guardados correctamente.</p> <p>2. Buscar Documento Requerido:</p> <p>2.1. El sistema desplegará la pantalla en la que se puede realizar la búsqueda de acuerdo a los siguientes criterios:</p> <ul style="list-style-type: none"> • Descripción del Documento Requerido. <p>2.2. A continuación se visualizará la información respectiva</p> <p>3. Modificar Documento Requerido:</p> <p>3.1. El sistema despliega la pantalla en la que el usuario selecciona el Documento requerido que desea modificar.</p> <p>3.2. A continuación el administrador</p>

Caso Uso Documentos Requeridos	
<p>4.1. Usuario escoge Eliminar Documento Requerido.</p> <p>4.2. Usuario selecciona el Documento Requerido a eliminar.</p> <p>4.3. Usuario da clic en la opción Actualizar.</p>	<p>podrá modificar la siguiente información :</p> <ul style="list-style-type: none"> • Descripción <p>3.3. Los datos son actualizados en la base. Se emite un mensaje de actualización exitosa.</p> <p>3.4. A continuación se visualizará la información respectiva.</p> <p>4. Eliminar Documento Requerido:</p> <p>4.1. El sistema despliega la pantalla en la que el usuario selecciona el Documento requerido que desea eliminar.</p> <p>4.2. El sistema despliega un mensaje de confirmación de la acción.</p> <p>4.3. La información es borrada de la Base de Datos.</p>
Flujo Alternativo:	
A continuación se muestran las excepciones que se pueden tener en el sistema.	
Actor	Sistema
<p>1. Ingresar Documento Requerido:</p> <p>1.1. Ingresar un Documento requerido con el mismo nombre.</p> <p>2. Buscar Documento Requerido:</p> <p>2.1. Usuario ingresa como parámetro de búsqueda un Documento Requerido inexistente.</p> <p>3. Modificar Documento Requerido</p> <p>3.1. Usuario modifica un Documento Requerido</p>	<p>1. Guardar Documento Requerido:</p> <p>1.1. El sistema despliega un mensaje indicando que el Documento requerido ya se encuentra registrado.</p> <p>2. Buscar Documento Requerido:</p> <p>2.2. El sistema despliega un mensaje indicando que el Documento requerido no existe.</p> <p>3. Modificar Documento Requerido</p>

Caso Uso Documentos Requeridos	
<p>inexistente.</p> <p>4. Eliminar Documento Requerido</p> <p>4.1. Usuario elimina un Documento requerido inexistente.</p>	<p>4.4. El sistema despliega un mensaje indicando que el Documento requerido a modificar no existe.</p> <p>4. Eliminar Documento Requerido</p> <p>1.1. El sistema despliega un mensaje indicando que el Documento requerido a eliminar no existe.</p>
Pos condición:	Cada Documento requerido se registrará una sola vez.
Requerimientos Especiales:	<p>Validación de Campos:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. (Obligatorio) • Descripción: Campo Alfanumérico de hasta 50 caracteres.(Obligatorio)
Puntos de Extensión:	

○ **ADMINISTRACIÓN DEPORTES Y HABILIDADES**

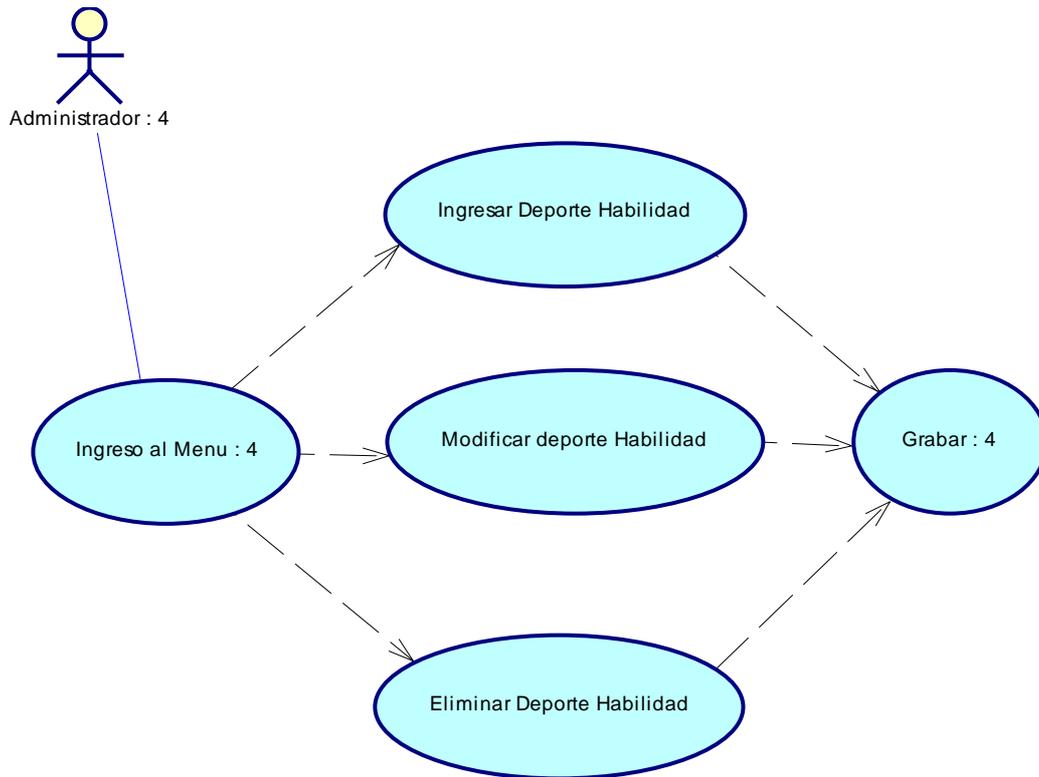


Figura 4. 5 Caso Uso Administración Deportes y Habilidades

Caso Uso Deportes y Habilidades	
Nombre:	Administración Deportes y Habilidades
Descripción:	Usuarios Administrador del Sistema. Es una actividad no frecuente que será ingresada en un inicio por el Administrador del Sistema.
Requerimiento:	Guardar los datos de los Deportes y Habilidades. (Ej. Futbol, Atletismo, etc.).
Precondición:	El Administrador, este autenticado en el sistema.
Flujo Normal:	
El proceso de Administración Tipo Calificación comprende las siguientes tareas: ingresar, buscar, modificar y eliminar. A continuación se describe cada uno de los procesos:	

Caso Uso Deportes y Habilidades	
Actor	Sistema
<p>1. Ingresar Deportes y Habilidades:</p> <p>1.1. Usuario escoge ingresar Deportes y Habilidades.</p> <p>1.2. Usuario ingresa la información en los campos correspondientes.</p> <p>1.3. Usuario da clic en la opción guardar.</p> <p>2. Buscar Deportes y Habilidades:</p> <p>2.1. Usuario escoge buscar Deportes y Habilidades.</p> <p>2.2. Usuario ingresa la información del Deporte o Habilidad.</p> <p>2.3. Usuario da clic en el botón Buscar.</p> <p>3. Modificar Deportes y Habilidades</p> <p>3.1. Usuario escoge Modificar Deportes y Habilidades.</p> <p>3.2. Usuario selecciona el Deportes y Habilidades a modificar.</p> <p>3.3. Usuario modifica la información en los campos correspondientes.</p> <p>3.4. Usuario da clic en la opción Actualizar.</p> <p>4. Eliminar Deportes y Habilidades</p>	<p>1. Ingresar Deportes y Habilidades:</p> <p>1.1. El sistema despliega la pantalla en la que se debe registrar la siguiente información:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. Asignado automáticamente por el sistema. • Descripción: Campo Alfanumérico de hasta 50 caracteres. <p>1.2. El sistema valida que la información ingresada por el usuario sea correcta.</p> <p>1.3. Los datos son guardados en la base y se presenta al usuario una pantalla de confirmación de que han sido guardados correctamente.</p> <p>2. Buscar Deportes y Habilidades:</p> <p>2.1. El sistema desplegará la pantalla en la que se puede realizar la búsqueda de acuerdo a los siguientes criterios:</p> <ul style="list-style-type: none"> • Descripción del Deportes y Habilidades. <p>2.2. A continuación se visualizará la información respectiva</p> <p>3. Modificar Deportes y Habilidades:</p> <p>3.1. El sistema despliega la pantalla con un listado de los Deportes y Habilidades.</p> <p>3.2. A continuación el administrador podrá modificar la siguiente información :</p>

Caso Uso Deportes y Habilidades	
<p>4.1. Usuario escoge Eliminar Deportes y Habilidades.</p> <p>4.2. Usuario selecciona el Deportes y Habilidades a eliminar.</p> <p>4.3. Usuario da clic en la opción Actualizar.</p>	<ul style="list-style-type: none"> • Descripción <p>3.3. Los datos son actualizados en la base. Se emite un mensaje de actualización exitosa.</p> <p>3.4. A continuación se visualizará la información respectiva.</p> <p>4. Eliminar Deportes y Habilidades:</p> <p>4.1. El sistema despliega la pantalla en la que el usuario selecciona el Deportes y Habilidades que desea eliminar.</p> <p>4.2. El sistema despliega un mensaje de confirmación de la acción.</p> <p>4.3. La información es borrada de la Base de Datos.</p>
Flujo Alternativo:	
A continuación se muestran las excepciones que se pueden tener en el sistema.	
Actor	Sistema
<p>1. Ingresar Deportes y Habilidades:</p> <p>1.1. Ingresar un Deporte y Habilidad con el mismo nombre.</p> <p>2. Buscar Deportes y Habilidades:</p> <p>2.1. Usuario ingresa como parámetro de búsqueda un Deporte y Habilidad inexistente.</p> <p>3. Modificar Deportes y Habilidades</p> <p>3.1. Usuario modifica un Deporte y Habilidad inexistente.</p> <p>4. Eliminar Deportes y</p>	<p>1. Guardar Deportes y Habilidades:</p> <p>1.1. El sistema despliega un mensaje indicando que el Deporte y Habilidad ya se encuentra registrado.</p> <p>2. Buscar Deportes y Habilidades:</p> <p>2.1. El sistema despliega un mensaje indicando que el Deporte y Habilidad no existe.</p> <p>3. Modificar Deportes y Habilidades</p> <p>3.1. El sistema despliega un mensaje indicando que el Deporte y Habilidad a modificar no existe.</p> <p>4. Eliminar Deportes y Habilidades</p>

Caso Uso Deportes y Habilidades	
Habilidades 4.1. Usuario elimina un Deporte y Habilidad inexistente.	4.1. El sistema despliega un mensaje indicando que el Deporte y Habilidad a eliminar no existe.
Pos condición:	Cada Deporte y Habilidad se registrará una sola vez.
Requerimientos Especiales:	Validación de Campos: <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. (Obligatorio) • Descripción: Campo Alfanumérico de hasta 50 caracteres. (Obligatorio)
Puntos de Extensión:	

○ **ADMINISTRACIÓN TIPO MENCIÓN HONORIFICA**

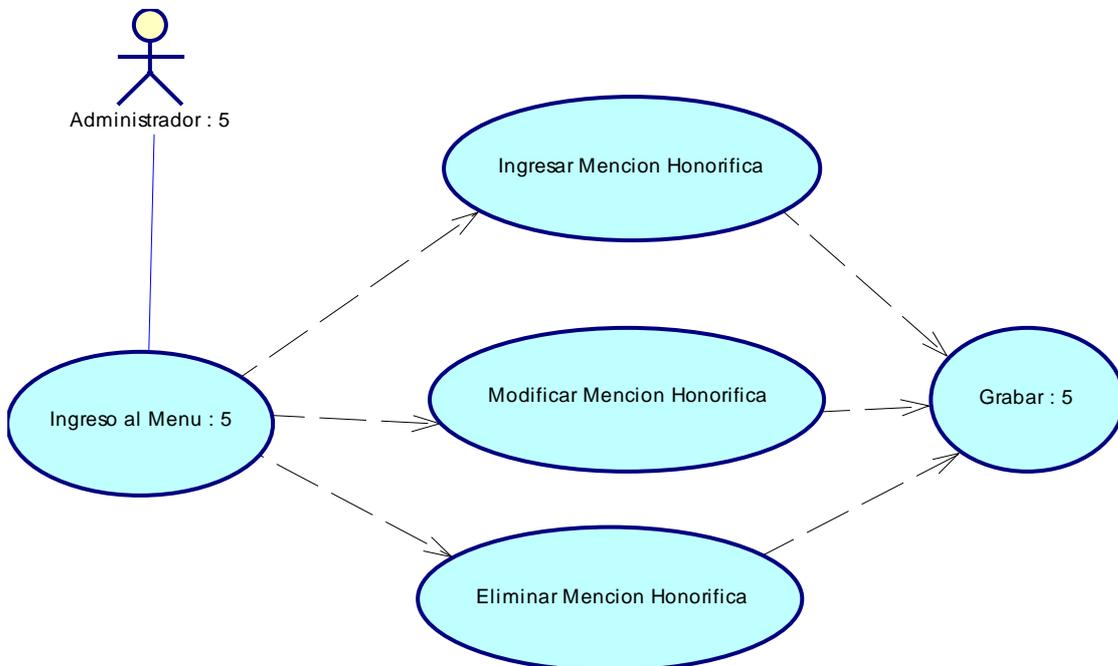


Figura 4. 6 Caso Uso Administración Tipo Mención Honorifica

Caso Uso Tipo Mención Honorifica	
Nombre:	Administración Tipo Mención Honorifica
Descripción:	Usuarios Administrador del Sistema. Es una actividad no frecuente que será ingresada en un inicio por el Administrador del Sistema.
Requerimiento:	Guardar los datos del Tipo de Mención Honorifica. (Ej. Mejor Egresado, Poeta)
Precondición:	El Administrador, este autenticado en el sistema.
Flujo Normal:	
El proceso de Administración Tipo Mención Honorifica comprende las siguientes tareas: ingresar, buscar, modificar y eliminar. A continuación se describe cada uno de los procesos:	
Actor	Sistema
<p>1. Ingresar Tipo Mención Honorifica:</p> <p>1.1. Usuario escoge ingresar Tipo Mención Honorifica.</p> <p>1.2. Usuario ingresa la información en los campos correspondientes.</p> <p>1.3. Usuario da clic en la opción guardar.</p> <p>2. Buscar Tipo Mención Honorifica:</p> <p>2.1. Usuario escoge buscar Tipo Mención Honorifica.</p> <p>2.2. Usuario ingresa la información del Tipo Mención Honorifica.</p> <p>2.3. Usuario da clic en el botón Buscar.</p>	<p>1. Ingresar Tipo Mención Honorifica:</p> <p>1.1. El sistema despliega la pantalla en la que se debe registrar la siguiente información:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. Asignado automáticamente por el sistema. • Descripción: Campo Alfanumérico de hasta 50 caracteres. <p>1.2. El sistema valida que la información ingresada por el usuario sea correcta.</p> <p>1.3. Los datos son guardados en la base y se presenta al usuario una pantalla de confirmación de que han sido guardados correctamente.</p> <p>2. Buscar Tipo Mención Honorifica:</p>

Caso Uso Tipo Mención Honorífica	
<p>3. Modificar Tipo Mención Honorífica</p> <p>3.1. Usuario escoge Modificar Tipo Mención Honorífica.</p> <p>3.2. Usuario selecciona el Tipo Mención Honorífica a modificar.</p> <p>3.3. Usuario modifica la información en los campos correspondientes.</p> <p>3.4. Usuario da clic en la opción Modificar.</p> <p>4. Eliminar Tipo Mención Honorífica</p> <p>4.1. Usuario escoge Eliminar Tipo Mención Honorífica.</p> <p>4.2. Usuario selecciona el Tipo Entrega a eliminar.</p> <p>4.3. Usuario da clic en la opción Eliminar.</p>	<p>2.1. El sistema desplegará la pantalla en la que se puede realizar la búsqueda de acuerdo a los siguientes criterios:</p> <ul style="list-style-type: none"> • Descripción del Tipo Mención Honorífica. <p>2.2. A continuación se visualizará la información respectiva</p> <p>3. Modificar Tipo Mención Honorífica:</p> <p>3.1. El sistema despliega la pantalla con un listado de Tipos de Mención Honorífica.</p> <p>3.2. A continuación el administrador podrá modificar la siguiente información :</p> <ul style="list-style-type: none"> • Descripción <p>3.3. Los datos son actualizados en la base. Se emite un mensaje de actualización exitosa.</p> <p>3.4. A continuación se visualizará la información respectiva.</p> <p>4. Eliminar Tipo Mención Honorífica:</p> <p>4.1. El sistema despliega la pantalla en la que el usuario selecciona el Tipo Mención Honorífica que desea eliminar.</p> <p>4.2. El sistema despliega un mensaje de confirmación de la acción.</p> <p>4.3. La información es borrada de la Base de Datos.</p>
Flujo Alternativo:	
A continuación se muestran las excepciones que se pueden tener en el sistema.	
Actor	Sistema

Caso Uso Tipo Mención Honorífica	
<p>1. Ingresar Tipo Mención Honorífica:</p> <p>1.2. Ingresar un Tipo Mención Honorífica con el mismo nombre.</p> <p>2. Buscar Tipo Mención Honorífica:</p> <p>2.2. Usuario ingresa como parámetro de búsqueda un Tipo Mención Honorífica inexistente.</p> <p>3. Modificar Tipo Mención Honorífica</p> <p>3.1. Usuario modifica un Tipo Mención Honorífica inexistente.</p> <p>4. Eliminar Tipo Mención Honorífica</p> <p>4.1. Usuario elimina un Tipo Mención Honorífica inexistente.</p>	<p>1. Guardar Tipo Mención Honorífica:</p> <p>1.2. El sistema despliega un mensaje indicando que el Tipo Mención Honorífica ya se encuentra registrado.</p> <p>2. Buscar Tipo Mención Honorífica:</p> <p>2.2. El sistema despliega un mensaje indicando que el Tipo Mención Honorífica no existe.</p> <p>3. Modificar Tipo Mención Honorífica</p> <p>3.1. El sistema despliega un mensaje indicando que el Tipo Mención Honorífica a modificar no existe.</p> <p>4. Eliminar Tipo Mención Honorífica</p> <p>4.1. El sistema despliega un mensaje indicando que el Tipo Mención Honorífica a eliminar no existe.</p>
Pos condición:	Cada Tipo Mención Honorífica se registrará una sola vez.
Requerimientos Especiales:	Validación de Campos: <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. (Obligatorio) • Descripción: Campo Alfanumérico de hasta 50 caracteres. (Obligatorio)
Puntos de Extensión:	

- **MÓDULO DE PROCESOS**
 - **REGISTRO DOCUMENTO**

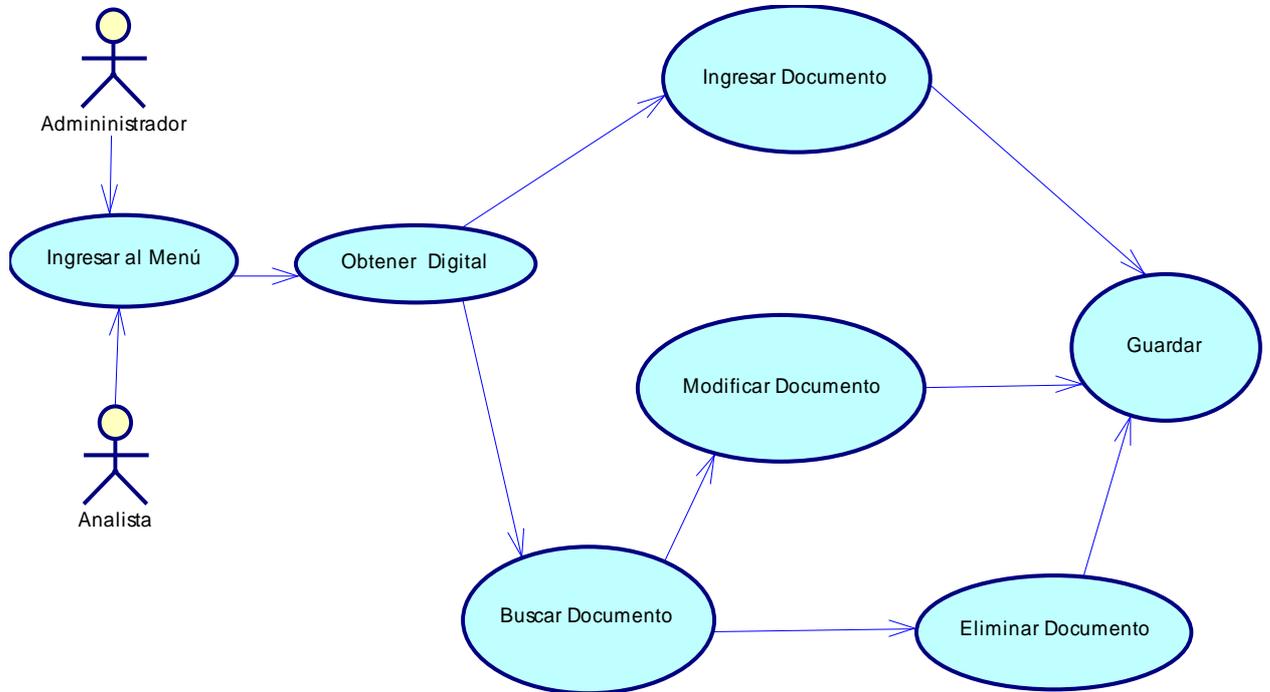


Figura 4. 7 Caso Uso Registro Documento

Caso Uso Registro Documento	
Nombre:	Registro Documento
Descripción:	Usuario Administrador del Sistema, Analista. Es una actividad frecuente que será utilizada por el Administrador del Sistema y el Analista.
Requerimiento:	Guardar los datos principales del Documento.
Precondición:	El Administrador/Analista este autenticado en el sistema. Se encuentre Documento Digital.
Flujo Normal:	
El proceso de Registro Documento comprende las siguientes tareas: ingresar, buscar y modificar. A continuación se describe cada uno de los procesos:	

Caso Uso Registro Documento	
Actor	Sistema
<p>1. Ingresar Documento :</p> <p>1.1. Usuario escoge Registro Documento.</p> <p>1.2. Usuario selecciona el documento digital.</p> <p>1.3. Usuario ingresa la información en los campos correspondientes.</p> <p>1.4. Usuario da clic en la opción guardar.</p> <p>2. Buscar Documento :</p> <p>2.1. Usuario escoge buscar Documento.</p> <p>2.2. Usuario ingresa la información del Documento.</p> <p>2.3. Usuario da clic en la opción Buscar.</p> <p>3. Modificar Documento</p> <p>3.1. Usuario escoge modificar Documento.</p> <p>3.2. Usuario selecciona el Documento a modificar.</p> <p>3.3. El usuario modifica la información en los campos correspondientes.</p> <p>3.4. Usuario da clic en la opción Modificar.</p> <p>4. Eliminar Documento</p> <p>4.1. El usuario escoge eliminar Documento.</p>	<p>1. Ingresar Documento</p> <p>1.1. El sistema despliega una pantalla que permite buscar/examinar el archivo.</p> <p>1.2. El sistema despliega la pantalla en la que se debe registrar la siguiente información:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. Asignado automáticamente por el sistema. • Código el Doc: Campo Alfanumérico de hasta 100 caracteres. • Fecha del Doc: Campo tipo Date • Fecha de Ingreso: Campo tipo Date • Cédula quien subió el Doc: Campo Alfanumérico de hasta 10 caracteres. • Nombre del Doc: Campo Alfanumérico de hasta 250 caracteres. • Sección: Campo numérico de hasta 6 dígitos. • Departamento: Campo numérico de hasta 6 dígitos. • Tipo Documento: Campo numérico de hasta 6 dígitos. • Código Calificación Documento: Campo numérico de hasta 6 dígitos. • Ubicación física: Campo Alfanumérico de hasta 250 caracteres. • Resumen Documento: Campo

Caso Uso Registro Documento	
<p>4.2. Usuario selecciona el Documento a eliminar.</p> <p>4.3. El usuario da clic en la opción Eliminar.</p>	<p>Alfanumérico de hasta 1024 caracteres.</p> <ul style="list-style-type: none"> • Cédula persona implicada: Campo Alfanumérico de hasta 10 caracteres. • Nombre Organización implicada: Campo Alfanumérico de hasta 250 caracteres. <p>1.3. El sistema valida que la información ingresada por el usuario sea correcta.</p> <p>1.4. Los datos son guardados en la base y se presenta al usuario una pantalla de confirmación de que han sido guardados correctamente.</p> <p>2. Buscar Documento :</p> <p>2.1. El sistema desplegará la pantalla en la que se puede realizar la búsqueda de acuerdo a los siguientes criterios:</p> <ul style="list-style-type: none"> • Código el Doc • Fecha del Doc • Nombre del Doc • Tipo Documento • Departamento • Cedula/Organización implicada • Criterios en blanco o Todos (Con lo que se mostrará toda la información). <p>2.2. A continuación se visualizará la información respectiva</p> <p>3. Modificar Documento :</p> <p>3.1. El sistema despliega la pantalla</p>

Caso Uso Registro Documento	
	<p>en la que selecciona el Documento que desea modificar.</p> <p>3.2. Sistema verifica el permiso para modificar el Documento.</p> <p>3.3. A continuación el administrador podrá modificar la siguiente información :</p> <ul style="list-style-type: none"> • Código el Doc • Fecha del Doc • Nombre del Doc • Tipo Documento • Departamento • Cedula/Organización implicada • Resumen Documento • Ubicación física <p>3.4. Los datos son actualizados en la base. Se emite un mensaje de actualización exitosa.</p> <p>3.5. A continuación se visualizará la información respectiva.</p> <p>4. Eliminar Documento :</p> <p>4.1. El sistema despliega la pantalla en la que el usuario selecciona el Documento que desea eliminar.</p> <p>4.2. Sistema verifica el permiso para modificar el Documento.</p> <p>4.3. El sistema despliega un mensaje de confirmación de la acción.</p> <p>4.4. La información es borrada de la Base de Datos.</p>
Flujo Alternativo:	
A continuación se muestran las excepciones que se pueden tener en el sistema.	
Actor	Sistema

Caso Uso Registro Documento	
<p>1. Ingresar Documento : 1.1. Ingresa un Documento con el mismo nombre.</p> <p>2. Buscar Documento : 2.1. Usuario ingresa como parámetro de búsqueda el nombre de un Documento inexistente.</p> <p>3. Modificar Documento 3.1. Usuario modifica un Documento inexistente.</p> <p>4. Eliminar Documento 4.1. Usuario elimina un Documento inexistente</p>	<p>1. Guardar Documento : 1.1. El sistema despliega un mensaje indicando que el Documento ya se encuentra registrado.</p> <p>2. Buscar Documento : 2.1. El sistema despliega un mensaje indicando que el Documento no existe.</p> <p>3. Modificar Documento 3.1. El sistema despliega un mensaje indicando que el Documento a modificar no existe.</p> <p>4. Eliminar Documento 4.1. El sistema despliega un mensaje indicando que el Documento a eliminar no existe</p>
<p>Pos condición:</p>	<p>Cada Documento se registrará una sola vez.</p>
<p>Requerimientos Especiales:</p>	<p>Validación de Campos:</p> <ul style="list-style-type: none"> • Código: Campo Numérico de hasta 6 dígitos. (Obligatorio). • Código el Doc: Campo Alfanumérico de hasta 100 caracteres. (No Obligatorio) • Fecha del Doc: Campo tipo Date. (Obligatorio) • Fecha de Ingreso: Campo tipo Date. (Obligatorio) • Cédula quien subió el Doc: Campo Alfanumérico de hasta 10

Caso Uso Registro Documento	
	<p>caracteres. (Obligatorio)</p> <ul style="list-style-type: none"> • Nombre del Doc: Campo Alfanumérico de hasta 250 caracteres. (Obligatorio) • Sección: Campo numérico de hasta 6 dígitos. (Obligatorio) • Departamento: Campo numérico de hasta 6 dígitos. (Obligatorio) • Tipo Documento: Campo numérico de hasta 6 dígitos. (Obligatorio) • Código Calificación Documento: Campo numérico de hasta 6 dígitos. (No Obligatorio) • Ubicación física: Campo Alfanumérico de hasta 250 caracteres. (No Obligatorio) • Resumen Documento: Campo Alfanumérico de hasta 1024 caracteres. (No Obligatorio) • Cédula persona implicada: Campo Alfanumérico de hasta 10 caracteres. (Obligatorio) • Nombre Organización implicada: Campo Alfanumérico de hasta 250 caracteres. (Obligatorio)
Puntos de Extensión:	

- **MÓDULO DE REPORTEES**
 - **REPORTES**

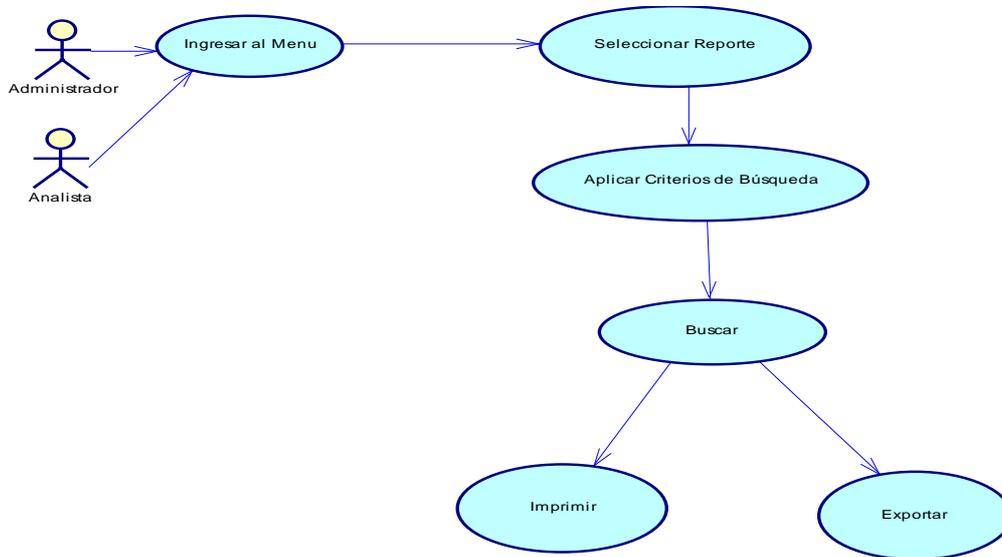


Figura 4. 8 Caso de Uso Reportes

Caso Uso Reportes	
Nombre:	Renovar Reportes
Descripción:	Permite desplegar reportes en base a datos ingresados. Es una actividad altamente frecuente que se realizará en base a los procesos realizados.
Requerimiento:	Un usuario desea visualizar reportes de los procesos realizados.
Precondición:	El Administrador/Analista este autenticado en el sistema.
Flujo Normal:	
El proceso de Reportes comprende las siguientes tareas:	
Actor	Sistema
1. Usuario escoge la opción del menú correspondiente al reporte que	1. El sistema desplegará la pantalla en la que se puede realizar la búsqueda, del

Caso Uso Reportes	
<p>desea generar.</p> <p>2. Usuario ingresa la información de los criterios de búsqueda y da clic en el botón buscar.</p> <p>3. A continuación el usuario podrá visualizar, imprimir o exportar el reporte.</p>	<p>reporte respectivo:</p> <ul style="list-style-type: none"> • Criterios en blanco o Todos (Con lo que se mostrará toda la información) (Por Defecto) <p>2. El sistema genera la información correspondiente al reporte y a los criterios ingresados para la búsqueda.</p> <p>3. El sistema valida el permiso para la impresión o la exportación del archivo.</p> <p>4. Si tiene permiso, el sistema graba la información con la fecha de impresión o exportación del archivo.</p>
Flujo Alternativo:	
A continuación se muestran las excepciones que se pueden tener en el sistema	
Actor	Sistema
<p>1. Usuario ingresa criterios de búsqueda que no generan datos en el reporte.</p> <p>2. Usuario no tiene el permiso de impresión, o exportación.</p>	<p>1. Sistema despliega un mensaje indicando que no existen datos para generar el reporte.</p> <p>2. Sistema despliega un mensaje indicando que el usuario no tiene el permiso respectivo.</p>
Pos condición:	
Requerimientos Especiales:	<ul style="list-style-type: none"> • Los parámetros de búsqueda que ingresará el usuario deben estar correctamente validados según el tipo de dato que corresponde.
Puntos de Extensión:	

4.1.1.3. JERARQUÍA MODULAR

4.1.1.3.1. INTRODUCCIÓN

- **PROPÓSITO**

El propósito de este documento es mostrar la estructura modular del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército y la distribución del menú dentro del mismo.

- **ALCANCE**

Este documento se centra en mostrar la distribución del menú del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército.

4.1.1.3.2. DESCRIPCIÓN

Jerarquía Modular del Sistema de Ingreso y Selección de la Dirección de Recursos Humanos del Ejército.

SISTEMA INGRESO Y SELECCIÓN DE LA DIRECCIÓN DE RECURSOS HUMANOS DEL EJÉRCITO		
Módulo	Submódulo	Opciones
ADMINISTRACIÓN		Actividades
		Cronograma
		Documentos Requeridos
		Deportes y Habilidades
		Tipo Mención Honorífica
		Partido Político
		Niveles de Calificación
		Tipo Referencia
		Profesión Candidato
		Tipo Conocimiento.
PROCESOS		Factores Evaluar
		Registro de los candidatos
		Recepción y Verificación de la documentación.
		Registro de las Pruebas Psicológicas.
		Registro de Pruebas Académicas
		Registro de Pruebas Médicas
	Registro de Pruebas Físicas	

		Parametrización Juntas de Calificación
REPORTES		Estadísticas de Personal Registrado.
		Listado de Personal que debe Rendir Pruebas Psicológicas.
		Estadísticas de Personal que Rindió Pruebas Psicológicas.
		Listado de Personal que debe rendir Pruebas Académicas.
		Estadísticas de Personal que Rindió Pruebas Académicas.
		Listado de Personal que debe rendir Pruebas Físicas.
		Estadísticas de Personal que Rindió Pruebas Medicas
		Listado de Personal que es Aceptado como Estudiante

4.1.1.4. MODELO LÓGICO

4.1.1.4.1. INSCRIPCIÓN DE CANDIDATOS

Figura 4. 9 Modelo Lógico Inscripción de Candidatos

4.1.1.4.2. FAMILIARES CANDIDATOS

Figura 4. 10 Familiares de Candidatos.

4.1.1.4.3. PARÁMETROS DE SELECCIÓN

Figura 4. 11 Parámetros de Selección.

4.1.1.4.4. PRUEBAS PSICOLÓGICAS

Figura 4. 12 Pruebas Psicológicas.

4.1.1.4.5. PRUEBAS ACADÉMICAS

Figura 4. 13 Pruebas Académicas.

4.1.1.4.6. PRUEBAS FÍSICAS

Figura 4. 14 Pruebas Físicas.

4.1.1.4.7. PRUEBAS MÉDICAS

Figura 4. 15 Pruebas Médicas.

4.1.1.5. SCRIPT GENERACION BASE DE DATOS

4.1.1.5.1. GENERACIÓN DE LA TABLA DE CANDIDATOS.

```
CREATE TABLE CASP_ASPMI (  
    ASP_FOTO          BLOB,  
    CAN_CODIGO1      NUMBER(10)          NOT NULL,  
    TSA_CODIGO       NUMBER(5),  
    CAB_CODIGO       NUMBER(5),  
    UNI_CODIGO2      VARCHAR2(25),  
    PRO_CODIG        NUMBER(8),  
    PPL_PPOLIT       SMALLINT,  
    CSE_CLACOD       VARCHAR2(5)        DEFAULT 'M'  
    CONSTRAINT CKC_CSE_CLACOD_CASP_ASP CHECK (CSE_CLACOD IS  
NULL OR (CSE_CLACOD IN ('C','M'))),  
    TEF_CODIGO       NUMBER(5),  
    SEX_CODIGO       CHAR(1)            DEFAULT 'M' NOT NULL  
    CONSTRAINT CKC_SEX_CODIGO_CASP_ASP CHECK (SEX_CODIGO IN  
( 'M','F' )),  
    CPI_CODIGO       NUMBER(5),  
    MPR_PRV_CODIGO   VARCHAR2(2),  
    MPR_CAN_CODIGO   VARCHAR2(3),  
    PRQ_CODIG        VARCHAR2(4),  
    OJO_CODIGO       NUMBER(5),  
    PRV_CODIGO       VARCHAR2(2),  
    CAN_CODIGO       VARCHAR2(3),  
    MPR_PRQ_CODIG    VARCHAR2(4),  
    TLI_CODIGO       NUMBER(5),  
    CAN_CEDULA       VARCHAR2(10),  
    CAN_APELLI       VARCHAR2(50),  
    CAN_NOMBRE       VARCHAR2(50),  
    CAN_DIRECC       VARCHAR2(50),  
    CAN_TELEFO       VARCHAR2(12),  
    CAN_FECNAC       DATE,  
    CAN_LIBMIL       VARCHAR2(12),  
    CAN_ESTATU       NUMBER(3),  
    CAN_NUMINT       VARCHAR2(10)       NOT NULL,  
    CAN_FECADM       DATE,  
    CAN_NUMCAR       INTEGER,  
    CAN_FECPRE       DATE,  
    CAN_PASSWORD     VARCHAR2(10)       NOT NULL,  
    CONSTRAINT PK_CASP_ASPMI PRIMARY KEY (CAN_CODIGO1)
```

```

);
COMMENT ON TABLE CASP_ASPMI IS
'CANDIDATO A INGRESAR A LA FUERZA TERRESTRE';
COMMENT ON COLUMN CASP_ASPMI.TSA_CODIGO IS
'Código del tipo de sangre';
COMMENT ON COLUMN CASP_ASPMI.UNI_CODIGO2 IS
'Código de la unidad de la FT';

COMMENT ON COLUMN CASP_ASPMI.PRO_CODIG IS
'Código de la profesión';
COMMENT ON COLUMN CASP_ASPMI.PPL_PPOLIT IS
'Código partidos políticos';
COMMENT ON COLUMN CASP_ASPMI.CSE_CLACOD IS
'Código de la clase de efectivo';
COMMENT ON COLUMN CASP_ASPMI.TEF_CODIGO IS
'Código del tipo de efectivo';
COMMENT ON COLUMN CASP_ASPMI.CPI_CODIGO IS
'Coding piel';
COMMENT ON COLUMN CASP_ASPMI.MPR_PRV_CODIGO IS
'Código de la Provincia';
COMMENT ON COLUMN CASP_ASPMI.MPR_CAN_CODIGO IS
'Código del Cantón';
COMMENT ON COLUMN CASP_ASPMI.PRQ_CODIG IS
'Código Parroquia';
COMMENT ON COLUMN CASP_ASPMI.OJO_CODIGO IS
'Código ojos';
COMMENT ON COLUMN CASP_ASPMI.PRIV_CODIGO IS
'Código de la Provincia';
COMMENT ON COLUMN CASP_ASPMI.CAN_CODIGO IS
'Código del Cantón';
COMMENT ON COLUMN CASP_ASPMI.MPR_PRQ_CODIG IS
'Código Parroquia';
COMMENT ON COLUMN CASP_ASPMI.TLI_CODIGO IS
'Código del tipo de licencia';
COMMENT ON COLUMN CASP_ASPMI.CAN_NUMINT IS
'Número de intento';
COMMENT ON COLUMN CASP_ASPMI.CAN_FECADM IS
'Fecha de admisión';
COMMENT ON COLUMN CASP_ASPMI.CAN_NUMCAR IS
'Número de carpeta';

```

COMMENT ON COLUMN CASP_ASPMI.CAN_FECPRE IS
'Fecha presentación';

```
CREATE INDEX CR_PPL_ASP_FK ON CASP_ASPMI (  
    PPL_PPOLIT ASC  
);
```

```
CREATE INDEX PR_SEX_MEM_FK ON CASP_ASPMI (  
    SEX_CODIGO ASC  
);
```

```
CREATE INDEX PRPRQ_ME__FK ON CASP_ASPMI (  
    PRV_CODIGO ASC,  
    CAN_CODIGO ASC,  
    MPR_PRQ_CODIG ASC  
);
```

```
CREATE INDEX PR_PRQ_MEM_FK ON CASP_ASPMI (  
    MPR_PRV_CODIGO ASC,  
    MPR_CAN_CODIGO ASC,  
    PRQ_CODIG ASC  
);
```

```
CREATE INDEX PR_UNI_MEM_FK ON CASP_ASPMI (  
    UNI_CODIGO2 ASC  
);
```

```
CREATE INDEX PR_TEF_MEM_FK ON CASP_ASPMI (  
    CSE_CLACOD ASC,  
    TEF_CODIGO ASC  
);
```

```
CREATE INDEX BBBCCCDD_FK ON CASP_ASPMI (  
    PRO_CODIG ASC  
);
```

```
CREATE INDEX COLCABCANDIDATO_FK ON CASP_ASPMI (  
    CAB_CODIGO ASC  
);
```

```
CREATE INDEX COLOR_DE_PIEL_DEL_CANDIDATO_FK ON CASP_ASPMI (  
    CPI_CODIGO ASC  
);
```

```
CREATE INDEX COLOR_DE_OJOS_CANDIDATO_FK ON CASP_ASPMI (  
    OJO_CODIGO ASC  
);
```

```
CREATE INDEX TIPO_SANGRE_DEL_CANDIDATO_FK ON CASP_ASPMI (  
);
```

```
TSA_CODIGO ASC
);
CREATE INDEX LICENCIA_CONDUCCION_CANDIDATO_FK ON CASP_ASPMI (
    TLI_CODIGO ASC
);
```

El contenido completo del script de la base de datos se encuentra anexado en el cd que se adjunta a este Proyecto

4.1.2. CONSTRUCCIÓN

4.1.2.1. MANUAL USUARIO

4.1.2.1.1. INTRODUCCIÓN

- **ALCANCE**

El presente documento está estructurado para que el usuario pueda primeramente conocer el sistema según su entorno visual y posteriormente le sea sencillo y rápido utilizarlo de manera funcional.

Como ya se indicó, el sistema se encuentra diseñado de manera tal, en la que su principal objetivo es su comodidad durante su uso. Así que en una breve sinopsis enumeraremos los pasos que el proceso seguirá, desde su inicio al hacer su ingreso por vez primera, hasta el registro de las diferentes calificaciones en cada uno de sus respectivos subprocesos y el despliegue de reportes.

Así, presentado y distribuido, el proceso se encuentra integrado por las siguientes etapas:

- Registro de los Candidatos
- Autenticación y Autorización
- Módulo de Mantenimiento.
- Módulo de Procesos y Sub-Procesos de Selección
- Módulo de Reportes.

A continuación daremos una ligera visión de lo que tratan todos los ítems enumerados

- **AUTENTICACIÓN Y AUTORIZACIÓN**

Este proceso lo realiza el Sistema de Ingreso al Portal Web (Menú) el cual se encargara de autenticar y autorizar a los usuarios en el sistema, para esto el sistema pedirá el ingreso del código de usuario y la contraseña. El código de usuario es la cédula de ciudadanía del usuario.

En caso de ser la primera vez en la que un usuario ingresa al sistema, la contraseña de ingreso será su número de cédula de ciudadanía, y automáticamente el sistema pedirá el cambio de contraseña.

- **MÓDULO DE MANTENIMIENTO DE TABLAS**

En este modulo, el Administrador podrá realizar la configuración de los parámetros generales del Sistema de Ingreso y Selección para su debido uso y funcionamiento.

- **MÓDULO DE PROCESOS Y SUB-PROCESOS DEL CASO MILITAR**

En este modulo el usuario, ingresará la información correspondiente a:

- Candidatos
- Validación Documentos Candidatos
- Pruebas Psicológicas
- Pruebas Académicas
- Pruebas Físicas
- Pruebas Medicas

- **MÓDULO DE REPORTE**

Es el modulo encargado de mostrar los reportes del sistema que darán la información necesaria a los usuarios de los procesos y subprocesos anteriormente mencionados.

El sistema de Ingreso y Selección se encuentra relacionado con el sistema de Ingreso al Portal Web del Ejército (Menú). A continuación se va a realizar una breve descripción de la relación existente.

- Sistema de Ingreso al Portal Web del Ejército (Menú).- La Dirección de Comunicaciones y Sistemas DISICOME cuenta con el sistema de seguridades desarrollado por la ESPE para controlar el acceso a cada uno de los sistemas utilizados en el Ejército Ecuatoriano.

4.1.2.1.2. DESCRIPCIÓN

- **AUTENTICACIÓN Y AUTORIZACIÓN**
 - Para ingresar al sistema es necesario digitar en el Internet Explorer la dirección <http://dominio/Menu>. Dar clic en el hipervínculo “clic aquí”.

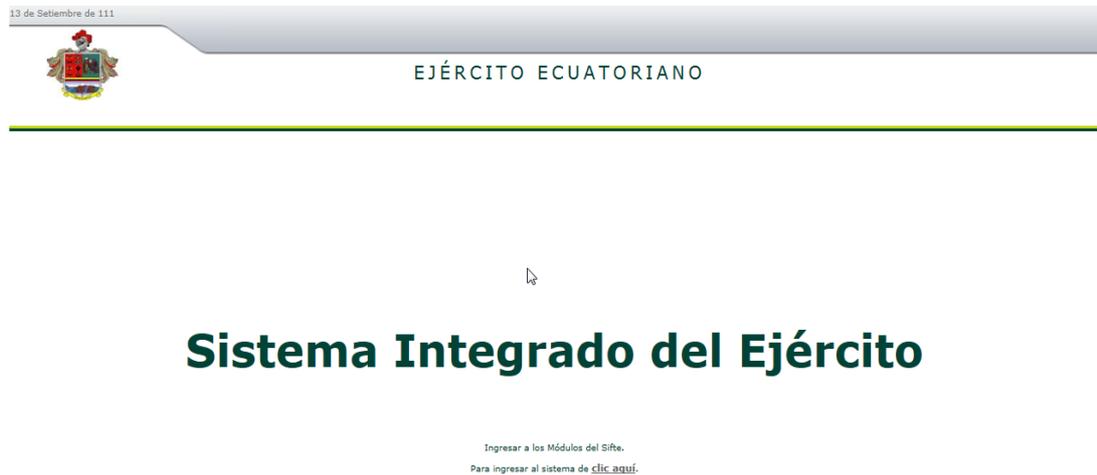


Figura 4. 16 Ingreso al Portal Web

- A continuación tendremos la siguiente pantalla, en donde se procede a ingresar el Usuario, Contraseña.
- Ingresar su Cédula en el recuadro de texto denominado “Usuario”, número que no variará, ya que esta es su cuenta de usuario, válida para su acceso.
- Nuevamente ingrese su Cédula en el Recuadro Inferior, denominado “Contraseña”, (tan solo si ingresa por primera ocasión, o si no ha modificado su contraseña todavía), para luego oprimir “Aceptar”.



INGRESO DE USUARIOS

Usuario:

Contraseña:

Si ingresa por primera vez, la Contraseña es su número de cédula.

Figura 4. 17 Ingreso de Contraseña

- A continuación tendremos la siguiente pantalla, con un listado de Sistemas a los que podemos ingresar, si tenemos el perfil asignado, escoger el Sistema de Ingreso y Selección “Sise”.

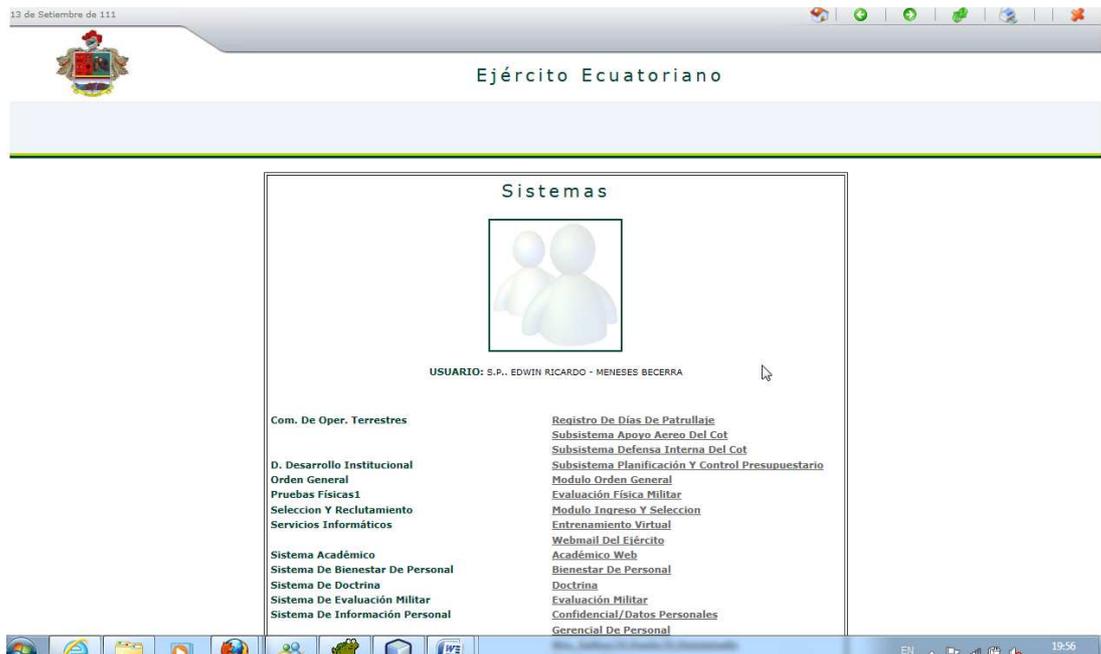


Figura 4. 18 Selección del Sistema Ingreso y Selección

- A continuación se despliega la siguiente pantalla, en la que se puede elegir una de las tres escuelas de formación.

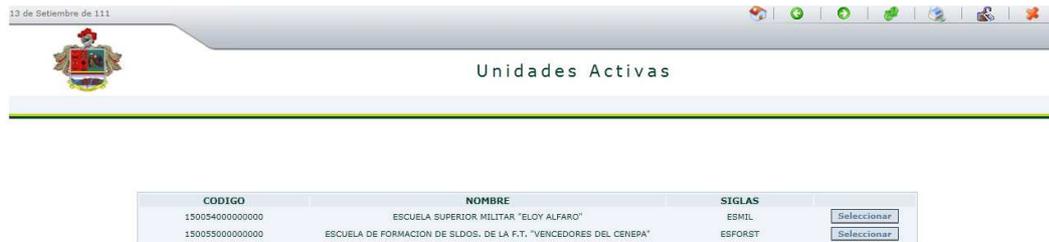


Figura 4. 19 Ingreso al Unidades Activas

- A continuación se muestra la siguiente pantalla, en la cual se puede visualizar en la parte izquierda de la ventana el menú con las diferentes opciones asignadas según el perfil que se le asigne a cada usuario del sistema.

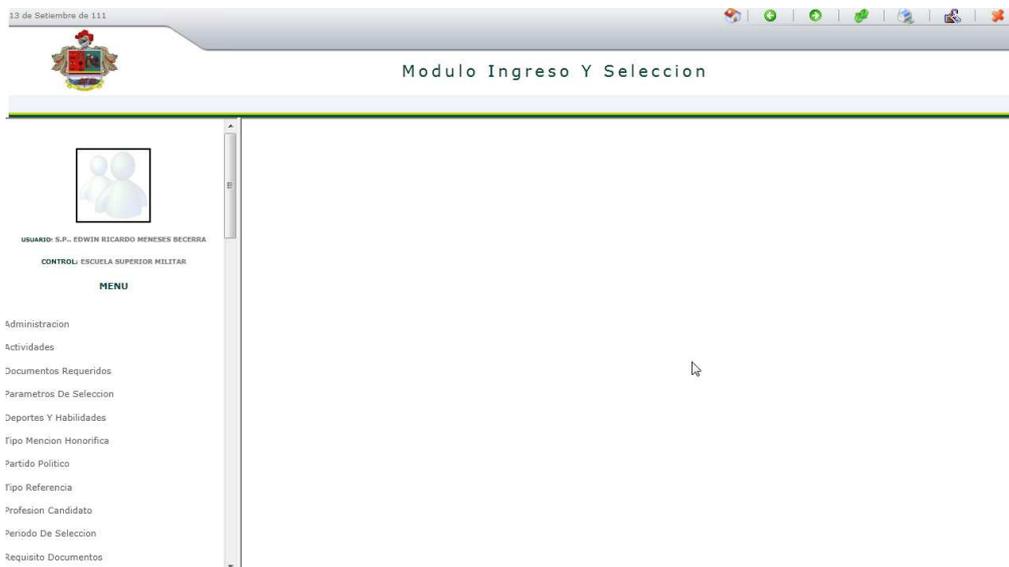


Figura 4. 20 Ingreso al Menú de Sise

➤ MÓDULO DE ADMINISTRACIÓN

• ADMINISTRACIÓN DE ACTIVIDADES

- Ingresar la descripción de las actividades que se van a cumplir durante el periodo de selección, si el cuadro tiene un color azul claro, indica que el ingreso es obligatorio. Dar clic en el botón Insertar.

13 de Setiembre de 111

Modulo Ingreso Y Seleccion

ACTIVIDADES DURANTE EL PERIODO DE SELECCION

Código:
Descripción:

INSERTAR ACTUALIZAR ELIMINAR LIMPIAR

Código	Descripción	Eliminar
1	PROCESO DE INSCRIPCIÓN EN LA PÁGINA WEB	<input type="checkbox"/>
5	RESULTADOS PRUEBAS PSICOLÓGICAS	<input type="checkbox"/>
6	RECEPCIÓN DE PRUEBAS ACADÉMICAS	<input type="checkbox"/>
7	RESULTADOS PRUEBAS ACADÉMICAS	<input type="checkbox"/>
8	RECEPCIÓN DE EXÁMENES MÉDICOS	<input type="checkbox"/>
9	RESULTADOS EXÁMENES MÉDICOS	<input type="checkbox"/>
10	RECEPCIÓN DE PRUEBAS FÍSICAS	<input type="checkbox"/>
12	ENTREVISTA	<input type="checkbox"/>
13	PERSONL IDONEO PROCESO SELECCIÓN	<input type="checkbox"/>
18	JUNTA DE SELECCION PARA RESULTADOS DE PRUEBAS FISICAS	<input type="checkbox"/>

Page: 1 of 2 GO

Figura 4. 21 Administración de Actividades

- Para Modificar un registro se da clic en el hipervínculo de la descripción y subimos los datos a la parte superior y modificamos los datos. Al dar clic en el hipervínculo se nos activa el botón de actualizar. Luego de realizar los cambios dar clic en el botón actualizar.

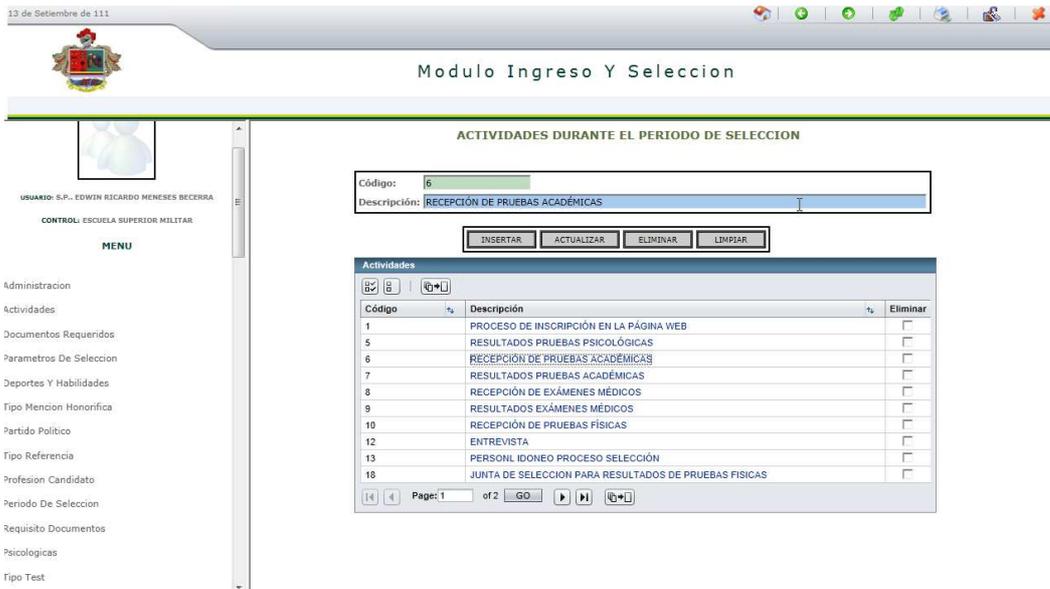


Figura 4. 22 Modificar Actividad

- Para Eliminar un registro se selecciona el registro o registros mediante el check box. Al realizar esta acción se nos activa el botón Eliminar. Dar clic en el botón eliminar.

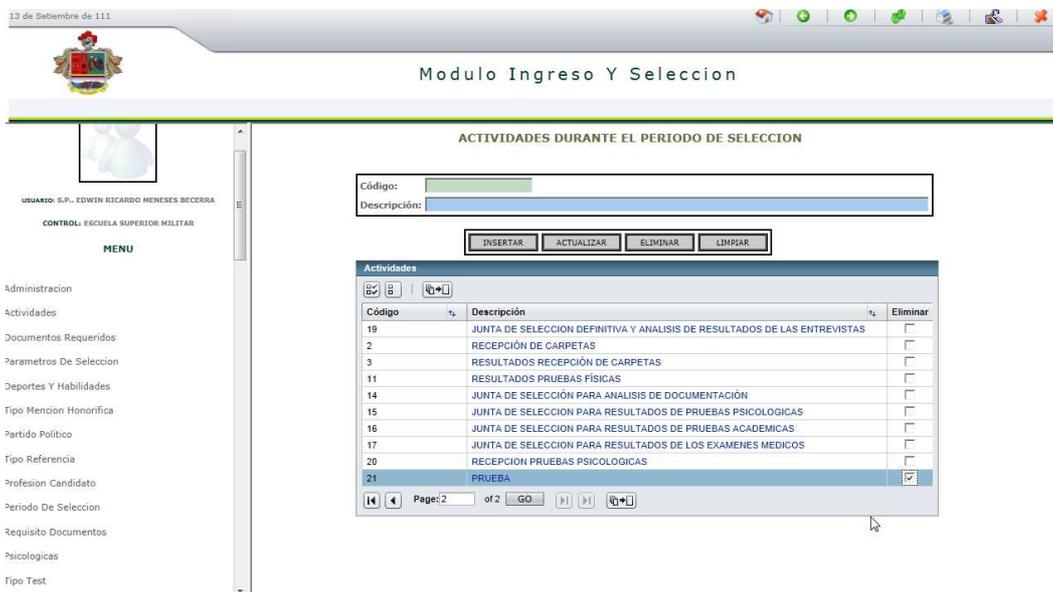


Figura 4. 23 Eliminar Actividad

- El Botón limpiar permite dejar en blanco los campos que están de color azul claro para volver a realizar otro tipo de consulta.

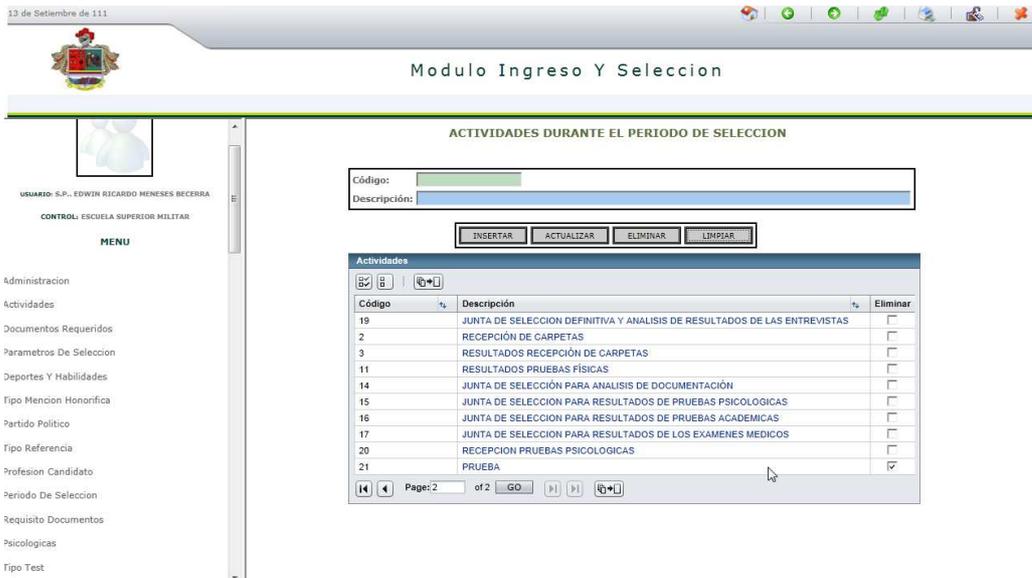


Figura 4. 24 Limpiar Actividades

- ADMINISTRACIÓN DE DOCUMENTOS REQUERIDOS

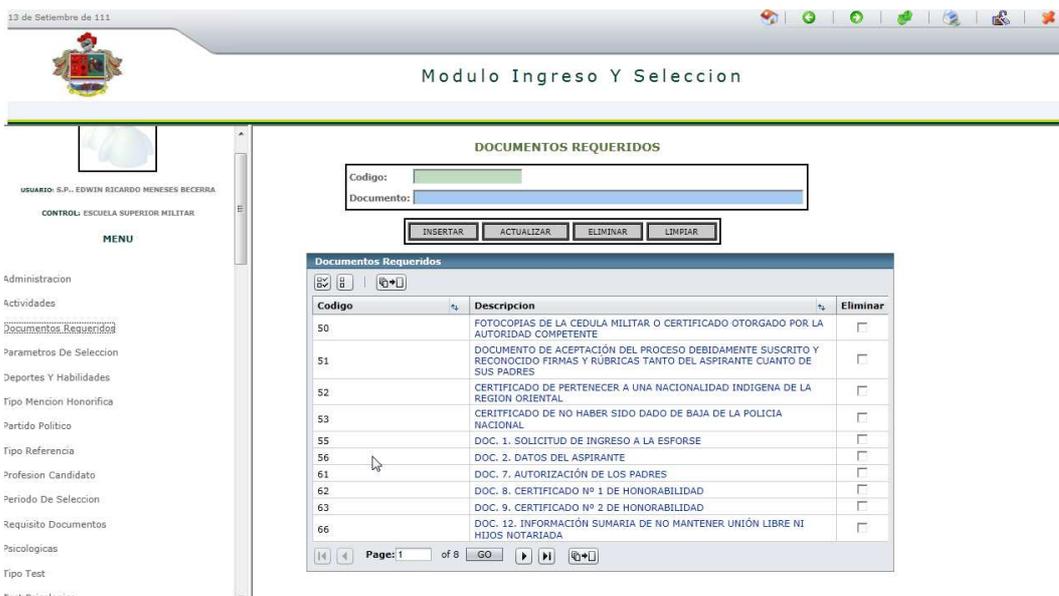


Figura 4. 25 Administración de Documentos Requeridos

- Ingresar la descripción del documento requerido a los candidatos para el periodo de selección, si el cuadro tiene un color azul claro, indica que el ingreso es obligatorio. Dar clic en el botón Insertar.
- Para Modificar un registro se da clic en el hipervínculo de la descripción y se suben los datos a la parte superior y modificamos los datos. Al dar clic en el hipervínculo se nos activa el botón de actualizar. Luego de realizar los cambios dar clic en el botón

actualizar.

- Para Eliminar un registro se selecciona el registro o registros mediante el check box. Al realizar esta acción se nos activa el botón Eliminar. Dar clic en el botón eliminar.
- El Botón limpiar nos permite dejar en blanco los campos que están de color azul claro para volver a realizar otro tipo de consulta.

- **ADMINISTRACIÓN DE LOS PARÁMETROS DE SELECCIÓN**

- Esta ventana permite registrar los parámetros que se utilizaran en el periodo de selección a continuación se describe cada uno de los campo necesarios así como los botones utilizados:
- En el campo Escuela de Formación no se puede ingresar ni modificar porque es un campo automático es decir al momento que usted selecciono la escuela de formación esa aparecerá en ese campo.

>

Figura 4. 26 Parámetros de Selección

- En puesto usted debe seleccionar una de las opciones posibles.
- En Fecha de Inicio Selección debe especificar la fecha en la que inicia el periodo de selección por medio del calendario que se muestra a continuación:

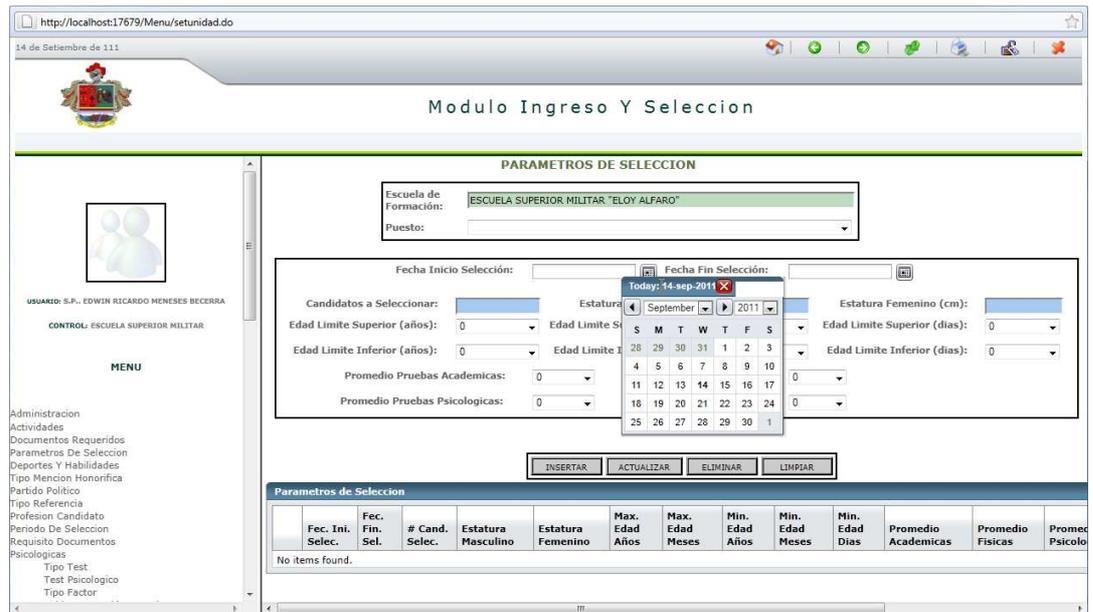


Figura 4. 27 Calendario Selección Fechas

- En Fecha Fin Selección debe seleccionar la fecha de finalización del periodo de selección por medio del calendario que se muestra en la imagen anterior.
- En Candidatos a Seleccionar debe registrar el número de candidatos que serán seleccionados después de terminar todo el proceso.
- En Estatura Masculino debe registrar la estatura mínima que debe cumplir un candidato Masculino para continuar en el proceso.
- En Estatura Femenino debe registrar la estatura mínima que debe cumplir una candidata de sexo Femenino para continuar en el proceso.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.

- La metodología RUP se acopla a las aplicaciones desarrolladas hasta el momento por el Ejército Ecuatoriano de una manera aceptable y en especial con este proyecto porque con la ayuda del lenguaje unificado de modelamiento UML se pudo tener una visión clara de cuáles son las necesidades de la sección de Selección y Reclutamiento.
- El levantamiento de requerimientos se llevo a cabo utilizando la norma IEEE-STD-830 con cada una de las recomendaciones y plantillas de la norma es para un sistema con una calidad alta y es por esta razón que el sistema de Ingreso y Selección se encuentra desarrollado de acuerdo a las necesidades verdaderas de cada uno de los usuarios.
- Las herramientas utilizadas en el desarrollo del sistema de Ingreso y Selección se complementan de una manera excelente entre ellas dando lugar a una aplicación robusta y confiable.
- El Diseño del sistema de Selección y Reclutamiento se realizó de manera eficiente ya que se uso Powerdesigner 12.5 puesto que es una herramienta muy completa y de fácil uso, se pudieron desarrollar todos los diagramas necesarios como son casos de uso, modelo entidad relación y script de base de datos.
- La integración con la base de datos SIFTE utilizada por el Ejército Ecuatoriano con motor de base de datos Oracle 11g se llevo a cabo sin ninguna novedad y se tuvo como resultado un sistema de Ingreso y Selección relacionado e integrado con todos los sistemas del Ejército Ecuatoriano.
- El sistema de Ingreso y Selección se implantó por medio de la página web del Ejército Ecuatoriano, y se procedió a realizar el primer proceso de Ingreso y Selección de candidatos a las diferentes Escuelas de formación utilizando el sistema desarrollado sin existir ninguna novedad de consideración.

5.2. RECOMENDACIONES.

- El Ejército Ecuatoriano por medio de la Dirección de Comunicaciones y sistemas debe realizar un análisis pormenorizado de la metodología RUP que se utiliza en la actualidad para el desarrollo de sus sistemas y optar por tecnologías modernas que le permitirán obtener mejores sistemas y de muy buena calidad.
- Realizar un estudio más detallado de lo que son las diferentes pruebas que se toman a los candidatos a las Escuelas de Formación para poder unificarlas en próximas pruebas y estandarizarlas en todo el Ejército Ecuatoriano.
- En el futuro se debería implementar un sistema de pistas de auditoría directamente en la base de datos debido a que existe el peligro de manipulación de la información por personas ajenas al sistema y sería una manera más eficiente de protegerla de cambios no deseados.
- Se debe solicitar a la Dirección de Comunicaciones del Ejército Ecuatoriano una conexión más fluida con las escuelas de formación porque las actuales comunicaciones tienen defectos y los enlaces se caen cada determinado tiempo.

BIBLIOGRAFÍA

- Brink, T.; Gergle D.; Wood, S.D.: Usability for the Web: M- Kauf. 2002
- Bruegge B., Dutoit A.H. Ingeniería de Software Orientado a Objetos, Prentice Hall – Pearson educación, México, 2002.
- Pressman R.S. Ingeniería del Software. Un enfoque práctico (5ª ed.) Mc Graw- Hill; New York , 2001.
- Rumbaugh J., Jacobson I., Booch G. El Lenguaje Unificado de Modelado. Manual de Referencia, Addison-Wesley, Madrid, 2000.
- Sommerville I. Ingeniería de software, 6ª edición, Prentice Hall – Pearson educación, México, 2002.
- Stevens P., Pooley R. Utilización de UML en Ingeniería del Software con Objetos y Componentes, Addison Wesley, Madrid, 2002.
- Silva, Andrés; Mercerat, Bárbara; Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Internet.

HOJA DE LEGALIZACION DE FIRMAS

ELABORADO POR

Edwin Ricardo Meneses Becerra

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña

Lugar y Fecha: Sangolquí, Diciembre 2011