

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



PROYECTO DE GRADO

**“SISTEMA INFORMATICO PARA LA ADMINISTRACION DE
PLANILLAS DE HABERES ADMINISTRATIVAS DE LA
UNIVERSIDAD MAYOR DE SAN ANDRES”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POSTULANTE: IVAN SERGIO ESPINAL ALVAREZ
TUTOR METODOLOGICO: LIC. GROVER ALEX RODRÍGUEZ RAMIREZ
ASESOR: LIC. MOISES MARTIN SILVA CHOQUE.**

LA PAZ – BOLIVIA

2015



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



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA.

Dedicado con mucho cariño a mis queridos padres

Por todo el apoyo y sacrificio que me dieron

En los malos y mejores momentos

AGRADECIMIENTOS.

A la Universidad Mayor de san Andrés por haberme acogido en sus aulas de estudio y brindado una educación de excelencia que impacto en mi vida en el ámbito profesional. Agradecer a la carrera de informática de la facultad de ciencias puras y naturales por permitirme integrar en sus filas de profesionales y ayudarme a consolidar como una mejor persona a la sociedad a través de mi profesión.

Al Lic. Grover Alex Rodríguez Ramírez, que muy profesionalmente me guio en las revisiones del proyecto absorbiendo mis dudas a todo momento.

Al Lic. Moises Martin Silva Choque por su asesoramiento, paciencia y apoyo para lograr la culminación de este proyecto.

A los funcionarios del Departamento de Recursos Humanos de la Universidad Mayor De San Andrés, en especial a la unidad de Escalafón.

Gracias.

RESUMEN

Existe una necesidad tecnológica de comunicación y de contar con toda información de manera ágil, oportuna y eficaz, esta necesidad también la presentan todas las instituciones y organizaciones, optando así por tener una importante presencia en la web.

El presente proyecto de grado denominado “**SISTEMA DE ADMINISTRACION DE PLANILLAS DE HABERES ADMINISTRATIVAS DE LA UNIVERSIDAD MAYOR DE SAN ANDRES**”, que permite almacenar información relevante de las planillas de haberes , ha sido desarrollado para la unidad de escalafón dependiente del dependiente de Recursos Humanos Administrativos de la Universidad Mayor de San Andrés. Es así que el objetivo es implementar el sistema informático para facilitar información detallada de la información almacenada en las planillas de haberes físicas de manera oportuna, eficaz además de la emisión de certificados de aportes.

En el proyecto se realizó un análisis previo y con el cual se efectuó el diseño de cada uno de los procesos, utilizando como metodología de desarrollo de software el Proceso Unificado de Desarrollo de software (RUP), que permite modelar comportamiento que va a tener el sistema utilizando el modelo Lenguaje de Modelo Unificado (UML), el cual modelara cada uno de los procesos, también se utilizó para el desarrollo de software Modelo Vista Controlador (MVC), que ayudo mantener estructurado el sistema para de esta manera poder ordenar nuestro código de una manera estándar.

Las herramientas utilizadas para su implementación son Lenguaje de Programación Java, PostgreSQL como gestor de Base de Datos, además del uso de JSF y PrimeFaces para el Frontend.

Finalmente mediante el análisis de resultados se determina que el sistema desarrollado cumple los requisitos planteados.

INDICE GENERAL

CAPITULO 1. MARCO REFERENCIAL

1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	3
1.2.1 INSTITUCIONAL	3
1.2.2 SOFTWARE.....	3
1.3 PLANTEAMIENTO DEL PROBLEMA	3
1.3.1 ANÁLISIS DEL PROBLEMA	3
1.4 OBJETIVOS	4
1.4.1 OBJETIVO GENERAL.....	4
1.4.2 OBJETIVOS ESPECÍFICOS	5
1.5 JUSTIFICACIÓN	5
1.5.1 TÉCNICA	5
1.5.2 ECONÓMICA	5
1.5.3 SOCIAL	5

CAPÍTULO 2. MARCO TEÓRICO

1.6 LIMITES Y ALCANCES.....	6
2.1 MARCO INSTITUCIONAL.....	7
2.1.1 DIVISIÓN DE RECURSOS HUMANOS ADMINISTRATIVOS.....	7
2.2 INGENIERÍA DE REQUERIMIENTOS	9
2.2.1 REQUERIMIENTOS	9
2.2.2 ACTIVIDADES DE LA INGENIERÍA DE REQUERIMIENTOS	11
2.3 METODOLOGÍAS E INGENIERÍA DE SOFTWARE	13
2.3.1 DESARROLLO ÁGIL DE SOFTWARE	13
2.3.2 METODOLOGÍA RUP	13
2.3.3 PROCESO DE VIDA DE RUP	14
2.3.4 INCEPCIÓN O CONCEPCIÓN	14
2.4 UML.....	16
2.4.2 UML Y FUNCIONALIDADES	16
2.4.3 TIPOS DE DIAGRAMAS UML	17
2.4.3.5 DIAGRAMA DE SECUENCIA.....	18
2.4.3.6 DESCRIPCIÓN DE CASOS DE USO REAL	19
2.4.3.7 DIAGRAMA DE CLASE.....	20
2.4.4 DEFINICIÓN DE ESQUEMA DE BASE DE DATOS	21
A. MODELO ENTIDAD RELACIÓN.....	22
B) DIAGRAMA ENTIDAD RELACIÓN	22
2.5 TECNOLOGÍAS DE SOFTWARE	24
A) NETBEANS.....	24
B. JAVA.....	25
C. JAVA SERVER FACES (JSF)	25
D. MVC (MODELO VISTA CONTROLADOR)	26

E. FRAMEWORK PRIMEFACES	27
F) GESTOR DE BASE DE DATOS POSTGRESQL	28
G) GLASSFISH	28
2.6 CALIDAD DE SOFTWARE ISO 9126.....	29
2.6.1 ISO 9126	29
A) FUNCIONALIDAD.....	29
B) FIABILIDAD.....	29
C) USABILIDAD	30
D) EFICIENCIA.....	30
E) MANTENIBILIDAD	30
F) PORTABILIDAD	31
2.7 ESTUDIO DE COSTOS Y BENEFICIO.....	31
2.7.1 ESTIMACIÓN DE COSTOS DE DESARROLLO DE SOFTWARE	31
2.7.2 COCOMO.....	32
2.7.3 LÍNEAS DE CÓDIGO Y PUNTOS DE FUNCIÓN.	32
2.8 SEGURIDAD.....	33
2.8.1 BENEFICIOS DE PRUEBAS DE SEGURIDAD	34
2.9 PRUEBAS	35
2.9.1 FUNDAMENTOS.....	35
2.9.3 PRUEBA DE CAJA BLANCA.....	35
2.9.4 PRUEBA DE CAJA NEGRA	36
CAPÍTULO 3. MARCO APLICATIVO	
3.1 FASE DE INICIO.....	37
3.1.1 MODELADO DEL NEGOCIO.....	37
3.1.1.1 MODELADO DE CASOS DE USO DEL NEGOCIO.....	37
3.1.1.2 DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO	38
3.1.1.3 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO	38
3.1.2 REQUERIMIENTOS.....	40
3.1.2.1 REQUERIMIENTOS FUNCIONALES.....	40
3.1.2.2 REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	41
3.1.2.3 REQUERIMIENTOS NO FUNCIONALES	42
3.1.3 DESCRIPCIÓN DE LOS ACTORES	42
3.1.3.1 DIAGRAMA DE CASOS DE USO	43
3.2 FASE DE ELABORACIÓN	43
3.2.1 ANÁLISIS Y DISEÑO.....	44
3.2.1.2 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA	44
3.2.1.3 DIAGRAMA DE CLASES.....	45
3.2.1.4 DIAGRAMA DE DESPLIEGUE	47
3.2.1.4.1 DESCRIPCIÓN DE CASOS DE USO EXPANDIDO	47
3.2.1.5 DIAGRAMA DE SECUENCIA DE SISTEMA.....	50
3.2.1.6 DIAGRAMA DE ACTIVIDADES	55
3.3 FASE DE CONSTRUCCIÓN.....	59
3.3.1 DIAGRAMA DE PAQUETES.....	59

c) DISEÑO DE INTERFAZ	61
3.4 FASE DE TRANSICIÓN	62
3.4.1 MANTENIMIENTO DE SOFTWARE	62
3.4.1.1 CONTROL DE ACCESO FÍSICO.....	63
3.5 SEGURIDAD	63
3.5.1 Control De Acceso.....	63
3.5.2 Limitación Del Tiempo De Conexión (Inactividad).....	64
3.5.3 Protección Y Respaldo De La Información.....	64
3.5.4 NIVELES DE ACCESO	65
3.5.5 COPIAS DE SEGURIDAD DE LA BASE DE DATOS.....	65
CAPÍTULO 4. MÉTRICAS DE CALIDAD	
4.1 FUNCIONALIDAD	66
4.2 CONFIABILIDAD.....	67
4.3 PORTABILIDAD	68
4.4 MANTENIBILIDAD	68
4.4.1 MANTENIMIENTO ADAPTIVO	68
4.4.2 MANTENIMIENTO PERFECTIVO	68
4.5 USABILIDAD	68
CAPITULO 5. EVALUACIÓN DE COSTOS Y BENEFICIOS	
5.1 MÉTODO DE COCOMO	70
CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.	
6.1 CONCLUSIONES.....	74
6.2 RECOMENDACIONES	75
BIBLIOGRAFÍA	76

ANEXOS

ANEXO A	INTERFAZ GRÁFICA.
ANEXO B	ÁRBOL DE PROBLEMAS.
ANEXO C	MARCO LÓGICO.
ANEXO D	DICCIONARIO DE DATOS.

ÍNDICE DE FIGURAS

Figura 2.1 (Página Oficial de la UMSA)	9
Figura 2.2 Ejemplo de Diagrama de Clases.....	18
Figura 3.1 Modelo de casos de Uso	37
Figura 3.2 Diagrama Casos de Uso	43
Figura 3.3 Diagrama Casos de uso del sistema.....	44
Figura 3.4 Diagrama de Clases.....	45
Figura 3.5 Diagrama Relacional.....	46
Figura 3.6 (Caso de Uso expandido, administra Gestión de Funcionarios).....	47
Figura 3.7 (Diagrama de Secuencia: Gestión de Funcionarios)	51
Figura 3.8 (Diagrama de Actividades).....	52
Figura 3.9 (Diagrama de Secuencia: Gestión de Planillas)	53
Figura 3.10 (Diagrama de Secuencia: Gestión de Usuarios)	54
Figura 3.11 (Diagrama de Secuencia: Gestión de Reportes).....	55
Figura 3.12 (Diagrama de Actividades Gestión de Usuarios)	56
Figura 3.13 (Diagrama de Actividades Gestión de Planillas)	57
Figura 3.14 (Diagrama de Actividades Gestión de Funcionarios)	58
Figura 3.15 (Diagrama de Actividades Gestión de Reportes).....	59
Figura 3.16 (Diagrama de paquetes: registro de Datos)	60
Figura 3.17 (Diagrama de paquetes: Elaboración de Reportes)	61

ÍNDICE DE TABLAS

Tabla 3.1 Descripción de caso de uso: Envío y recepción de solicitud.	38
Tabla 3.2 Descripción de caso de uso: Recepción de Solicitud y Realización de certificado de aportes.	39
Tabla 3.3 Descripción de caso de uso: Entrega y recepción de Planillas de haberes.	40
Tabla 3.4 Requerimientos del Sistema.....	41
Tabla 3.5 Registro de datos Funcionarios, Planillas, usuarios	41
Tabla 3.6 Muestra registros, de búsqueda de funcionarios.....	41
Tabla 3.7 Emisión de Certificado de aportes.....	42
Tabla 3.8 Emisión de Certificado de aportes.....	42
Tabla 3.9 Descripción de caso de uso: Administrar Funcionarios (CRUD) Funcionarios.....	48
Tabla 3.10 Descripción de caso de uso: Administrar Planillas (CRUD) Planillas. .	48
Tabla 3.11 Descripción de caso de uso: Administrar Usuarios (CRUD).....	49
Tabla 3.12 Descripción de caso de uso: Generar listados, reportes.	50
Tabla 4.1 Rangos de Nivel de Calidad para la Fiabilidad	66
Tabla 4.2 Calculo de la Confiabilidad. (Elaboración Propia)	67
Tabla 4.3 Resultados para el cálculo de facilidad de uso.....	69
Tabla 5.1 Ponderación del esfuerzo	71
Tabla 5.2 Factores y Valores.....	71

MARCO REFERENCIAL

1.1 INTRODUCCIÓN

La informática se ha incorporado a todas las actividades del quehacer del hombre y conforme se difunde con gran rapidez el uso de las computadoras dentro de las organizaciones, surgen muchas inquietudes acerca de la forma de usarlas para mejorar la productividad y lograr mejorar los objetivos de la organización. (Kendal & Kendal, 2002)

Con la evolución de las tecnologías informáticas en nuestra sociedad, tiende cada día a automatizar labores cotidianas que antes se realizaban de forma manual. Esto ha llevado a varias instituciones u organizaciones a automatizar y aprovechar de este modo los recursos tecnológicos para un mejor desempeño de sus actividades.

En la actualidad, para muchas organizaciones los sistemas de información basados en computadoras son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones. (Senn, 1992). La información juega un papel muy importante en las instituciones, es base para la toma de decisiones, debe estar disponible en cualquier momento de tal forma que sea la más oportuna, confiable y efectiva.

Si bien hoy en día no todas las instituciones cuentan con un sistema informático que les ayude a mejorar sus servicios en forma óptima, eficiente y eficaz. Pero muchas de estas, van por ese camino de automatizar las tareas que se realizan así de esta en forma proporcionar rápidos y buenos servicios.

La administración de documentos tiene una importancia para el funcionamiento adecuado de organismos de administración pública como privadas. Entonces, se puede decir, que la gestión de dicha información almacenada en documentos tiene gran importancia, por lo cual, se debe dar una atención inmediata al tratamiento de información que ingresa y sale de las instituciones.

El avance de las tecnologías informáticas, objetivamente los sistemas de información, han adquirido gran importancia en el ámbito social y sobre todo organizacional por las utilidades que este presta; siendo su mayor aporte la eficaz, segura y rápida comunicación, además que logra concentrar grandes cantidades de información en lapsos de tiempo, de esta forma los procesos que presentaban el uso de una amplia cantidad de recursos, tiempo y espacio

son menos desventajosos al implementar un sistema de software informático. Como en todo estudio también existe la posibilidad de que los sistemas de información sean mal empleados en las organizaciones; ya que en estas no es necesario su uso o simplemente requieren de un solo proceso, cuando en realidad un sistema representa varios de estos. De implementar un sistema informático innecesario a procesos que cumplen eficientemente su trabajo, esto significa gastos insulsos en su desarrollo, para evitar estos inconvenientes es que se debe realizar un sistema bajo una estructura acorde a las metodologías más apropiadas que adquiere una gran importancia al momento de implementar un sistema de información.

El desarrollo e implementación de un sistema de información no es la única solución que un informático o ingeniero de sistemas puede dar a un determinado problema institucional, las soluciones van desde la reingeniería hasta el rediseño de procesos, pero es la solución con mayor requerimiento al interior de las organizaciones; que gracias al avance tecnológico producido por la computadora y sus aplicaciones incrementan sus beneficios económicos al automatizar procesos, integrarlos e incluso trabajar con software inteligente a nivel de decisiones de gerencia; en este sentido, el presente proyecto de grado implementara la metodología necesaria para proporcionar una solución óptima y de esta manera dar solución a los inconvenientes encontrados.

En el Departamento de Recursos Humanos de la Universidad Mayor de San Andrés se encuentra la unidad de Escalafón Administrativo, dicha unidad se encarga de almacenar la información del personal administrativo, También se almacena información histórica de las planillas de haberes, se genera un libro de datos por mes y es esta unidad la que se encarga de almacenar los libros, y también de emitir reportes solicitados por autoridades para la toma de decisiones.

El presente proyecto de grado pretende desarrollar un sistema de información para proporcionar soluciones a las necesidades actuales que tienen relación con procesos de acceso de información que ayude a resolver problemas de optimización del tiempo en la elaboración de reportes además de brindar más información a través otros tipos de reportes, listas y resúmenes que ayudaran a la unidad escalafón, departamento de recursos humanos y demás, a la adecuada toma de decisiones.

El proyecto está elaborado con el objetivo de organizar, fortalecer, automatizar y sistematizar la información contenida en los fondos documentales que existe en el Archivo de Escalafón Administrativo.

1.2 ANTECEDENTES

1.2.1 Institucional

Universidad Mayor de San Andrés, Departamento de Recursos Humanos Administrativos, Unidad Escalafón Administrativo con Una población promedio 1500 funcionarios activos registrados en las planillas de 2014, 180 personas a contrato eventual y 133 Beca – Trabajo, la Unidad de Escalafón administrativo, que es la encargada de administrar la información, la información esta resguardada en Files Personales, Planillas Alfabéticas y Planillas de Haberes. Las Planillas de haberes son las que almacenan la información:

- Ítem.
- Renta.
- Descuentos (Aportes, etc.).

La unidad de recursos humanos no cuenta con la tecnología apropiada que coadyuve a administrar la información recepcionada en las planillas de haberes. Por esta razón al momento de recibir requerimientos de autoridades superiores, se procede a revisar planilla por planilla de cada registro almacenado para poder construir el reporte solicitado, en promedio la tardanza es de dos días para la emisión de informes, dependiendo del requerimiento (Cantidad de Años Solicitados).

1.2.2 Software

No existes Ningún tipo de Tecnología de información (Software) que ayude a realizar el proceso de selección de información, sin embargo se cuenta con un banco de datos que almacena información que no pertenece a esta unidad, es tal el caso que la existencia de estos no influye en el desarrollo en el trabajo

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Análisis Del Problema

Por los grandes beneficios que hoy en día nos brindan las herramientas informáticas, El Departamento de Recursos Humanos en conjunto con la Unidad de Escalafón no se puede

dar el lujo de prescindir de ellas, es por esta razón que se pudo detectar las siguientes problemáticas.

- La búsqueda de información almacenada en las planillas de haberes es manual, y se emplea un tiempo considerable en su procesamiento, dado que recurren a la búsqueda y clasificación de los registros de manera individual.
- El tiempo para generar un certificado de aportes es largo y tedioso, y no se genera de manera oportuna.
- La generación de reportes no permite una toma de decisiones adecuada por el tiempo que se emplea para su desarrollo, dado que el tiempo de procesamiento es largo por la cantidad de información existente.
- En los procesos de la elaboración de planillas existen fallas humanas (Visión, Manipulación, etc.) al momento de elaborar informes.
- No existen procesos que controlen la seguridad de la información y el procesamiento de datos.

En el transcurso del análisis, se pudo divisar la necesidad de salvaguardar la información contenida en cada planilla, para que no exista manipulación en el manejo de datos y la no Transparencia. Por ello se plantea la siguiente pregunta.

¿Con un sistema informático de administración de datos registrados en la planilla de haberes se agilizará los procesos de elaboración de reportes, en el departamento de recursos humanos administrativos?

1.4 OBJETIVOS

1.4.1 Objetivo General

Diseñar, desarrollar e implementar un sistema informático que permitirá sistematizar, clasificar, organizar y actualizar el proceso de emisión de certificados de aportes para de esta manera optimizar recursos humanos y económicos de tal manera poder realizar un trabajo más eficiente al momento de recibir requerimientos, garantizando un proceso seguro, rápido y transparente con el resguardo electrónico de los registros.

1.4.2 Objetivos Específicos

- Centralizar la información en una Base de Datos con estándares de seguridad, y con el debido control de usuarios.
- Mejorar el proceso de búsqueda y la generación de reportes mediante procesos de consulta a la base de datos centralizada que permita optimizar el tiempo.
- Automatizar los procesos de registros, modificaciones y dar seguimientos a la información de los funcionarios con la información de las planillas de haberes.
- Elaborar interfaz de usuario que ayude a que se tenga fácil acceso a la información.
- Generar de manera confiable y oportuna Certificados de Aportes con un módulo que permita su generación para los funcionarios del área administrativa.

1.5 JUSTIFICACIÓN

Las justificaciones del proyecto “Sistema Informático de Planillas Haberes” (S.I.P.HA.), son desarrollados de acuerdo a tres aspectos técnica, económica y social.

1.5.1 Técnica

La Unidad de Escalafón administrativos dependiente del Departamento de Recursos Humanos Administrativos, cuenta con el soporte técnico necesario para llevar a cabo la implementación del sistema web, cabe mencionar que actualmente no se cuenta con ningún tipo de sistema relacionado la administración de información de las planillas de Haberes. El usuario final, cuenta con los medios tecnológicos para hacer uso del producto final.

1.5.2 Económica

Un sistema de administración de información de planillas, permitirá que los funcionarios administrativos puedan optimizar sus principales tareas, mejorando el tiempo de servicio y la emisión de información en un tiempo oportuno, el sistema que será implementado para ser utilizado mediante la web permitirá que el usuario lo utilice desde su oficina.

1.5.3 Social

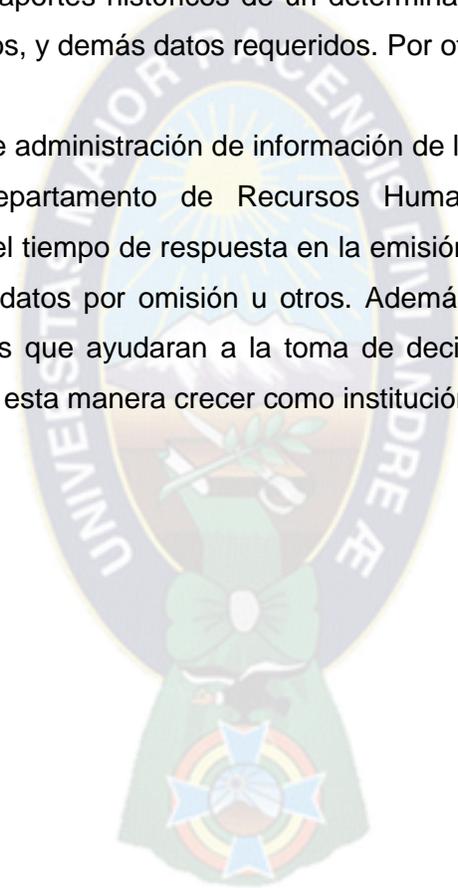
El recurso humano administrativo, es la población beneficiada del proyecto, ya que reducirá el tiempo en la consulta de datos que se requiera. Los funcionarios de esta unidad, están

capacitados en el uso de plataformas web, por lo que el sistema puede desarrollarse y ejecutarse en la institución. Así también otras unidades serán beneficiadas indirectamente; mejorando la imagen de servicio oportuno, confiable y seguro a la comunidad administrativa de la Universidad Mayor de San Andrés.

1.6 LIMITES Y ALCANCES

LÍMITES. El sistema informático se realizara la ejecución de dos tareas específicas. Emisión de certificado de aportes históricos de un determinado funcionario detallando el total de aportes, descuentos, y demás datos requeridos. Por otro lado,

ALCANCES. El sistema de administración de información de las planillas de haberes será de beneficio para el departamento de Recursos Humanos unidad de escalafón administrativo, disminuirá el tiempo de respuesta en la emisión de certificado de aportes y evitara posibles fugas de datos por omisión u otros. Además el sistema será capaz de generar reportes generales que ayudaran a la toma de decisiones para tener un mejor funcionamiento y poder de esta manera crecer como institución.



MÁRCO TEÓRICO

2.1 MARCO INSTITUCIONAL

2.1.1 División De Recursos Humanos Administrativos

MISIÓN

Optimizar la aplicación de los recursos humanos administrativos de la Universidad Mayor de San Andrés, orientando su accionar a una constante mejora en la calidad de los servicios que presta la unidad.

OBJETIVOS

Administrar los recursos humanos administrativos de la Universidad Mayor de San Andrés con la mayor eficacia, eficiencia y competitividad en la prestación de los servicios en cumplimiento a normas y políticas emanadas por los órganos superiores del Gobierno Universitario.

Optimizar la gestión de los recursos humanos administrativos mediante los procesos técnicos, eficientes, eficaces, oportunos y transparentes que permitan el aporte productivo y el compromiso institucional del funcionario administrativo a la administración. (UMSA, 2015)

2.1.1.1 Funciones De La Unidad

- Planificar, organizar, ejecutar y controlar el sistema de gestión de recursos humanos, de la Universidad.
- Elaborar, revisar, consensuar y ejecutar la planilla Presupuestaria del personal administrativo.
- Coordinar y ejecutar los procesos de reclutamientos, selección, transparencia, promoción y cesantía del personal.
- Proponer a las autoridades universitarias las modificaciones al Manual de Clasificación, Descripción y Valoración de Cargos, Escalafón Administrativo, Escala Salarial y otras disposiciones en materia de recursos humanos, sobre la base de cambios que se produzcan en la institución.
- Elaborar la programación de puestos.

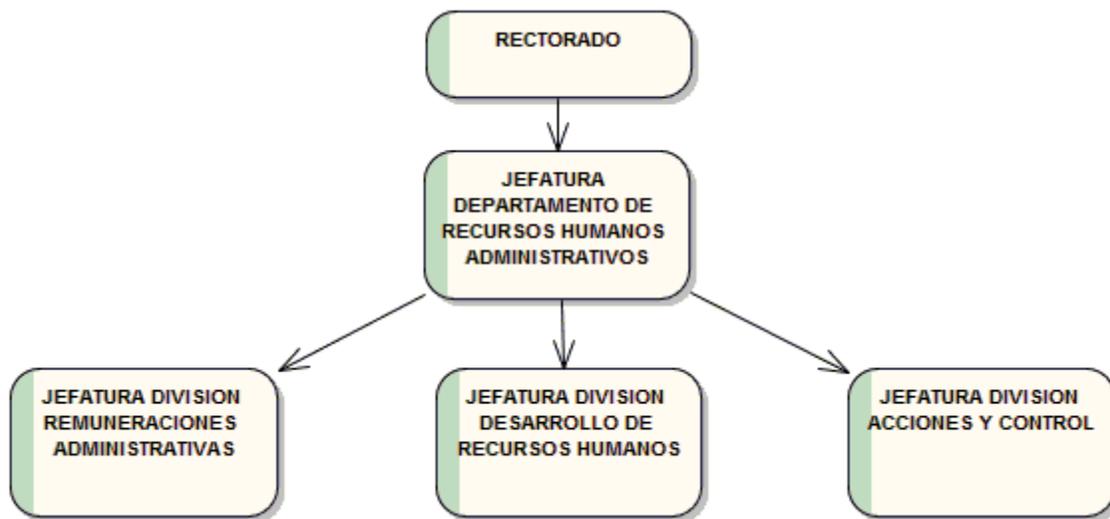
- Dirigir y coordinar las acciones destinadas a mantener un eficiente archivo y registro de antecedentes de los funcionarios, mediante control sistemático y actualizado de documentos contenida en los files personales.
- Expedir certificaciones de años de servicio, aportes, de categorías de personal administrativo.
- Elaborar informes sobre las actividades desarrolladas en el departamento.
- Coordinar y dirigir las comisiones de Procesos Administrativos.
- Aplicar el Régimen Disciplinario del Reglamento Internos de Personal en coordinación con las unidades académicas y administrativas.

2.1.1.2 Disposiciones Legales

- Constitución Política del Estado.
- Estatuto Orgánico de la U.M .S.A., aprobado por el 1er. Congreso Interno de la Universidad Mayor de San Andrés el 31 de octubre de 1988.
- Ley N° 1178 “De Administración y Control Gubernamentales” (SAFCO) de 20 de julio de 1990.
- Reglamento Específico del Sistema de Organización Administrativa de la Universidad Mayor de San Andrés de las Normas Básicas del Sistema de Organización Administrativa. Resolución del Honorable Consejo Universitario No. 065/2002 de fecha 3 de abril de 2002.
- Reglamento Interno del Personal Administrativo; C.E.H.C.U. 344/91 (25 Octubre de 1991).
- Manual de Clasificación, Descripción y Valoración de Cargos; C.E.H.C.U. 73/91 (12 Marzo 1991).
- Reglamento Escalafón Administrativo, U.M.S.A. 1994. (Umsa.Bo, 2014)

Estructura Orgánica del Departamento.

Figura 2.1 (Página Oficial de la UMSA)



Fuente: (UMSA, 2015)

2.2 INGENIERÍA DE REQUERIMIENTOS

2.2.1 Requerimientos

Normalmente, un tema de la Ingeniería de Software tiene diferentes significados. De las muchas definiciones que existen para requerimiento, a continuación se presenta la definición que aparece en el glosario de la IEEE. (Pressman., 1993)

- i. Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- ii. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- iii. Una representación documentada de una condición o capacidad como en (i) o (ii).

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad.

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes.

Necesario: Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.

Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento. (Senn, 1992)

No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación.

Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Dificultades para definir los Requerimientos.

- i. Los requerimientos no son obvios y vienen de muchas fuentes.
- ii. Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- iii. Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- iv. La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- v. Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.

- vi. Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- vii. Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- viii. Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- ix. Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

2.2.2 Actividades De La Ingeniería De Requerimientos

A. Extracción De Datos Y Estudios De Factibilidad

En esta etapa los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar, etc. Es importante, que la extracción sea efectiva, ya que la aceptación del sistema dependerá de cuan bien éste satisfaga las necesidades del cliente. Ejemplo, Observar las necesidades del nuevo sistema o mejora de uno existente. Para extraer las funciones que no han de cambiar y las nuevas que surgirán en el nuevo sistema, identificar las necesidades del Sistema para el control y seguimiento de asistencia del personal administrativo con la colaboración del cliente para entender de forma clara y precisa lo que se quiere del sistema, que hay que resolver, restricciones del sistema y servicio que prestará el sistema.

El resultado de esta etapa es producir un informe de factibilidad como se ilustra en la tanto en realizar una recolección y evaluación de la información, como redactar el informe del estudio de la factibilidad. (Somerville, 1996).

B. Análisis De Requerimientos

Sobre la base de la extracción realizada previamente, comienza esta fase en la cual se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento. Estudiar sobre la base de extracción los requerimientos del cliente los problemas existentes, como solucionarlos, entre otros puntos de interés. Usualmente se hace un análisis luego de haber producido un BOSQUEJO inicial del documento de requerimientos; en esta etapa se leen los requerimientos, se conceptúan, se investigan,

se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requerimientos. (Somerville, 1996)

C. Especificaciones

En esta etapa se establece la especificación de los requerimientos, es decir lo que el sistema debe realizar. Esta etapa es muy complicada debido a que la naturaleza de los problemas es muy compleja. Cada especificación dependiendo de la complejidad.

Es menester destacar que la especificación puede verse como un proceso independiente del modo en que se realice, todo esto con el objetivo de lograr una adecuada implementación de software. Además se han determinado los siguientes principios para representar los requisitos de software (Somerville, 1996)

- Separar la funcionalidad de la implementación
- Desarrollar un modelo de comportamiento de un sistema que comprenda los datos y las respuestas funcionales de un sistema a varios estímulos del entorno.
- Establecer los componentes del sistema que interactúan con él.
- Definir el entorno en que operara el sistema
- Crear un modelo intuitivo
- Considerar que una especificación es una abstracción de una situación real por lo cual será incompleta y existirá a muchos niveles de detalle.
- Definir un contenido y estructura que sea susceptible a cambios

D. Validación De Requerimientos

En esta etapa se establecen los requerimientos finales ó completos que definirán el sistema que el cliente desea. Se puede apreciar que el proceso de ingeniería de requerimientos es un conjunto estructurado de actividades, mediante las cuales se obtiene, se valida y se logra dar un mantenimiento adecuado al documento de especificación de requerimientos, que es el documento final, de carácter formal, que se obtiene de este proceso. Es necesario recalcar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura

organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. (Somerville, 1996).

2.3 METODOLOGÍAS E INGENIERÍA DE SOFTWARE

2.3.1 Desarrollo Ágil De Software

Se entiende como Desarrollo Ágil de Software a un paradigma de Desarrollo de Software basado en procesos ágiles. Los procesos ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos Ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos Ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, diseñadores de iteración, escritores de documentación y ayuda y directores de proyecto.

Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica. (López y Mtra)

2.3.2 Metodología RUP

Siempre que empezamos discutiendo métodos en la arena OO, inevitablemente salimos con el papel del Rational Unified Process. El Proceso Unificado fue desarrollado por Philippe Kruchten, Ivar Jacobson y otros de la Rational como el proceso complementario al UML. El RUP es un armazón de proceso y como tal puede acomodar una gran variedad de

procesos. De hecho ésta es mi crítica principal al RUP - como puede ser cualquier cosa acaba siendo nada.

Como resultado de esta mentalidad de armazón de procesos, el RUP puede usarse en un estilo muy tradicional de cascada o de una manera ágil. Como resultado usted puede usar el RUP como un proceso ágil, o como un proceso pesado - todo depende de cómo lo adapte a su ambiente. (López y Mtra)

2.3.3 Proceso De Vida De RUP

El proceso de ciclo de vida de RUP se divide en cuatro fases bien conocidas llamadas Incepción, Elaboración, Construcción y Transición. Esas fases se dividen en iteraciones, cada una de las cuales produce una pieza de software demostrable. La duración de cada iteración puede extenderse desde dos semanas hasta seis meses. Las fases son:

2.3.4 Incepción O Concepción

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Significa “comienzo”, pero la palabra original (de origen latino y casi en desuso como sustantivo) es sugestiva y por ello la traducimos así. Se especifican los objetivos del ciclo de vida del proyecto y las necesidades de cada participante. Esto entraña establecer el alcance y las condiciones de límite y los criterios de aceptabilidad. Se identifican los casos de uso que orientarán la funcionalidad.

Se diseñan las arquitecturas candidatas y se estima la agenda y el presupuesto de todo el proyecto, en particular para la siguiente fase de elaboración. Típicamente es una fase breve que puede durar unos pocos días o unas pocas semanas.

2.3.4.1 Elaboración

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Se analiza el dominio del problema y se define el plan del proyecto. RUP presupone que la fase de elaboración brinda una arquitectura suficientemente sólida junto con requerimientos y planes bastante estables. Se describen en detalle la infraestructura y el ambiente de desarrollo, así como el soporte de herramientas de automatización. Al cabo de esta fase, debe estar identificada la mayoría de los casos de uso y los actores, debe quedar descripta la arquitectura de software y se debe crear un prototipo de ella. Al final de la fase se realiza un análisis para determinar los riesgos y se evalúan los gastos hechos contra los originalmente planeados.

2.3.4.2 Construcción

Se desarrollan, integran y verifican todos los componentes y rasgos de la aplicación. RUP considera que esta fase es un proceso de manufactura, en el que se debe poner énfasis en la administración de los recursos y el control de costos, agenda y calidad. Los resultados de esta fase (las versiones alfa, beta y otras versiones de prueba) se crean tan rápido como sea posible. Se debe compilar también una versión de entrega. Es la fase más prolongada de todas.

2.3.4.3 Transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. Este tipo de metodología no ha sido aplicada probablemente por su complejidad de administración o desconocimiento de la misma, desaprovechando sus considerables ventajas respecto a los métodos tradicionales. Por esto, es necesario entonces desarrollar mecanismos de apropiación tecnológica más eficaces, que permitan mantener actualizadas las prácticas organizacionales y los marcos de referencia aquí mencionados. Es aquí, donde es necesario considerar que el conocimiento de la metodología y desarrollo de habilidades de los analistas, programadores, administradores de bases de bases de datos y demás miembros del equipo de desarrollo, comienzan desde su preparación universitaria donde es necesario conocer este enfoque y aplicarlo en proyectos en donde utilicen las guías de trabajo definidas en el RUP y desarrollen los artefactos asociados;

Esperando que con la práctica alcancen un nivel de madurez en la asimilación del proceso unificado (RUP).

De esta manera en la asignatura de análisis y diseño de sistemas de información II, que se imparte a los estudiantes del programa educativo de Tecnologías de la información y comunicación, se desarrolla un proyecto de simulación tipo basado en la metodología de trabajo del Proceso Unificado Rational, utilizando la herramienta CASE “Rational Unified Process” que es un sitio WEB en línea que los alumnos consultan para entender los términos en que debe ser realizada la documentación y diseño de los programas informáticos que construyen.

Con el objetivo de promover el conocimiento de la estructura metodología del RUP y en complemento al uso de herramienta CASE descrita anteriormente, se diseñó una Tecnología Educativa de Alto Impacto (TEAI) en su modalidad de material didáctico que permite que los alumnos conozcan, comprendan y analicen la naturaleza iterativa de desarrollo de proyectos con el proceso unificado en relación al avance ó estatus del proyecto y la evolución de los artefactos generados.

Con esta (TEAI) se busca que los estudiantes manipulen directamente la estructura del ciclo de vida del RUP, mediante el manejo de piezas adheribles al pizarrón le permiten identificar roles o artefactos por fases o por disciplinas disminuyendo la complejidad que resulta el trabajar por primera vez con procesos evolutivos promoviendo de la misma manera la participación activa del alumno en la asimilación de su propio conocimiento. (López y Mtra)

2.4 UML

2.4.1 DEFINICIÓN

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

2.4.2 UML Y Funcionalidades

El término “lenguaje” ha generado bastante confusión respecto a lo que es UML. En realidad el término lenguaje quizás no es el más apropiado, ya que no es un lenguaje propiamente dicho, sino una serie de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software. Mucha gente piensa por confusión que UML es un lenguaje de programación y esta idea es errónea: UML no es un lenguaje de

programación. Como decimos, UML son una serie de normas y estándares que dicen cómo se debe representar algo.

UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales (aquellos que definen qué debe hacer un programa sin entrar a escribir el código) y analistas-programadores (aquellos que dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python o cualquier otro). Por tanto si estás dando tus primeros pasos en programación, te recomendaríamos que te olvides de UML hasta que tengas unos conocimientos mínimos como uso de condicionales, bucles, y conocimiento de la programación orientada a objetos. Esto es solo una recomendación, en realidad prácticamente cualquier persona puede usar UML, incluso podría usarse para realizar esquemas o documentación de procesos que no tengan que ver con la informática.

Hemos dicho que UML es un estándar. Vamos a aclarar primero qué es un estándar. Supongamos que vamos a definir un estándar llamado “LMAPR” o lenguaje de modelado. (J. Pedro Caraca, 1997.)

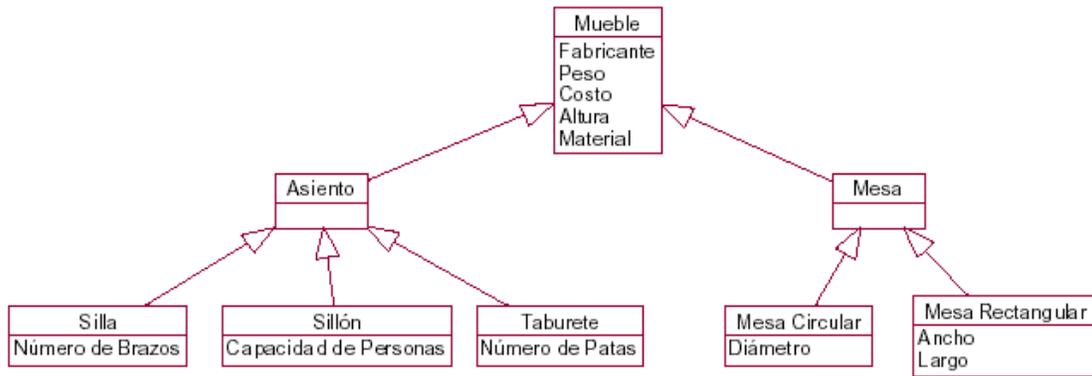
2.4.3 Tipos De Diagramas UML

Usando UML se pueden construir numerosos tipos de diagramas. Vamos a citar algunos: diagramas de casos de uso: representan a los actores y casos de uso (procesos principales) que intervienen en un desarrollo de software. (James Rumbaugh, 1996)

2.4.3.1 Diagramas De Clases

Para UML una clase es una entidad, no una clase software. Un diagrama de clases UML puede ser un diagrama del dominio o representación de conceptos que intervienen en un problema, o también un diagrama de clases software. El sentido de un diagrama UML se lo da la persona que lo construye. (Grady Booch, 1999)

Figura 2.2 Ejemplo de Diagrama de Clases



Fuente (Grady Booch, 1999)

2.4.3.2 Diagramas De Secuencia

Suelen usarse para representar objetos software y el intercambio de mensajes entre ellos, representando la aparición de nuevos objetos de izquierda a derecha.

2.4.3.3 Diagramas De Colaboración

Suelen usarse para representar objetos o clases y la forma en que se transmiten mensajes y colaboran entre ellos para cumplir un objetivo.

2.4.3.4 Diagramas De Estados

Suelen usarse para representar cómo evoluciona un sistema (cómo va cambiando de estado) a medida que se producen determinados eventos.

E) Otros Diagramas

Diagramas de actividad, diagramas de paquetes, diagramas de arquitectura software, etc.

2.4.3.5 Diagrama De Secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. En inglés se pueden encontrar como "sequencediagram", "event-trace diagrams", "eventscenarios" o "timingdiagrams"

A) Utilidad

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales. (Fowler , 2003)

b) Tipos De Mensajes

Existen dos tipos de mensajes: sincrónicos y asincrónicos. Los mensajes sincrónicos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada. Este tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asincrónicos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta.

También se representa la respuesta a un mensaje con una flecha discontinua.

2.4.3.6 Descripción De Casos De Uso Real

Un caso real de uso describe el diseño concreto del caso de uso a partir de una tecnología particular de entrada y salida, así como de su implementación global. Por ejemplo, si interviene una interfaz gráfica para el usuario, el caso de uso real incluirá diagramas de las ventanas en cuestión y una explicación de la interacción de bajo nivel con los artefactos de la interfaz.

Los casos de uso tienen las siguientes características:

- a) Están expresados desde el punto de vista del actor.
- b) Se documentan con texto informal.

c) 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 –claramente diferenciada– del sistema

2.4.3.7 Diagrama De Clase

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos.

- El diagrama de clases puede tener como ejemplo: una clase que sería un objeto o persona misma en la cual se especifica cada acción y especificación.
- Propiedades de objetos que tienen propiedades y/u operaciones que contienen un contexto y un dominio, los primeros dos ejemplos son clases de datos y el tercero clase de lógica de negocio, dependiendo de quién diseñe el sistema se pueden unir los datos con las operaciones.
- El diagrama de clases incluye mucha más información como la relación entre un objeto y otro, la herencia de propiedades de otro objeto, conjuntos de operaciones/propiedades que son implementadas para una interfaz gráfica.
- Presenta las clases del sistema con sus relaciones estructurales y de herencia.
- El diagrama de clases es la base para elaborar una arquitectura MVC o MVP.

2.4.3.7.1 Contenido

Un diagrama de clases comúnmente con tiene lo siguiente:

- Clases
- Interfaces
- Colaboraciones
- Dependencia
- Generalización

Relaciones de asociación

Los otros diagramas de clase pueden contener notas y restricciones. Los diagramas de clase pueden también contener paquetes o subsistemas ambos de los cuales son usados para agrupar elementos de su modelo. Algunas veces se quieren instancias de lugar en el

diagrama de clases, como también especialmente cuando se quiere visualizar el tipo de una instancia (posibilidad dinámica).

2.4.3.7.2 Usos Comunes

Modelado del diseño estático de un sistema. Esta vista en primer lugar soporta los requerimientos funcionales de un sistema - el servicio del sistema debería de proveer este a los usuarios finales.

Para el modelo de diseño estático de la vista de un sistema, típicamente se usan diagramas de clases en alguna de estas tres alternativas:

- Modelo del vocabulario de un sistema. El modelo del vocabulario de un sistema involucra tomar decisiones acerca de las cuales son parte del sistema y cuales quedan fuera del ambiente. Los diagramas de clase especifican estas abstracciones y sus responsabilidades.
- Modelado simple de colaboraciones. Una colaboración es una sociedad de clases, interfaces, y otros elementos, estos trabajan juntos para proveer igual comportamiento de colaboración, esto es más grande que la suma de todos los elementos. Por ejemplo, cuando se está modelando la semántica de una transacción en un sistema distribuido, no se puede fijar la vista en una simple clase, para entender cuál irá. Esta semántica es llevada fuera por un conjunto de clases que trabajan juntas. Los diagramas de clases se usan para visualizar y especificar este conjunto de clases y sus relaciones.
- Modelo lógico del esquema de la base de datos. Pensar en un esquema como la heliografía (dibujo) para el diseño conceptual de una base de datos. En muchos dominios se quiere almacenar mucha información persistente en una base de datos relacional o en base de datos orientada a objetos. Se pueden modelar esquemas para estas bases de datos usando diagramas de clases.

2.4.4 Definición De Esquema De Base De Datos

El esquema de una base de datos (en inglés, database schema) describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS). En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un diccionario de datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.

Generalmente en la práctica el término esquema de la base de datos se refiere al diseño físico de la base de datos.

A. Modelo Entidad Relación

Un diagrama o modelo entidad-relación (a veces denominado por sus siglas DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

- Se elabora el diagrama (o diagramas) entidad-relación.
- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

El modelado de datos no acaba con el uso de esta técnica. Son necesarias otras técnicas para lograr un modelo directamente implementable en una base de datos. Brevemente:

Permite mostrar resultados entre otras entidades pertenecientes a las existentes de manera que se encuentre la normatividad de archivos que se almacenaran

- Transformación de relaciones múltiples en binarias.
- Normalización de una base de datos de relaciones (algunas relaciones pueden transformarse en atributos y viceversa).
- Conversión en tablas (en caso de utilizar una base de datos relacional).
(Monografias.com, 2015)

B) Diagrama Entidad Relación

Diagrama Entidad Relación: Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de un Esquema gráfico empleando los terminología de Entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas Atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo.

En un DER, cada entidad se representa mediante un rectángulo, cada relación mediante un rombo y cada dominio (conjunto donde toma valores el atributo) mediante un círculo. Mediante líneas se conectan las entidades con las relaciones, igual que las entidades con los dominios, representando a los atributos. Los Atributos Llaves se representan subrayando el correspondiente conjunto de valores.

En ocasiones, una entidad no puede ser identificada únicamente por el valor de sus propios atributos. En estos casos, se utilizan conjuntamente las relaciones con los atributos para lograr la requerida identificación unívoca. Estas entidades reciben el nombre de entidades débiles y se representan en el DER con un doble rectángulo. El MER restringe las relaciones a usar para identificar las entidades débiles a relaciones binarias del tipo 1: N. Así, por ejemplo, una ocurrencia de "trabajador" puede tener N ocurrencias "persona-dependiente" asociadas, donde además, la existencia de las ocurrencias en la segunda entidad depende de la existencia de una ocurrencia que le corresponda en la primera entidad. Por ejemplo, en el modelo habrá personas dependientes de un trabajador sólo si ese trabajador existe. Para indicar esa dependencia en la existencia se usa una saeta en el DER. La llave de una entidad débil se forma combinando la llave de la entidad regular que la determina con algún otro atributo que defina unívocamente cada entidad débil asociada a una entidad regular dada. (Una entidad se denomina regular si no es débil).

En una relación, la llave es la combinación de las llaves de todas las entidades asociadas. Para cada relación se determina su tipo (simple o complejo) y en el DER se escribe el tipo de correspondencia. Por ejemplo, una empresa puede tener varios (n) trabajadores asociados y un trabajador pertenece a una sola empresa (1). En la relación Trabajador-Máquina-Pieza, un trabajador puede trabajar en n máquinas, produciendo p piezas, o una pieza puede ser producida por m trabajadores en n máquinas. Aquí, m, n y p no identifican un número específico, sino solamente el tipo de correspondencia que se establece en la relación.

I) Entidad

Las entidades son el fundamento del modelo entidad relación. Podemos adoptar como definición de entidad cualquier cosa o parte del mundo que es distinguible del resto. Por ejemplo, en un sistema bancario, las personas y las cuentas bancarias se podrían interpretar como entidades. Las entidades pueden representar entes concretos, como una persona o un avión, o abstractas, como por ejemplo un préstamo o una reserva.

ii) Atributo

Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

iii) Relaciones

Se representa mediante un rombo etiquetado en su interior con un verbo. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.

Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos.

2.5 TECNOLOGÍAS DE SOFTWARE

a) NETBEANS

Es un entorno de desarrollo integrado libre, que utiliza el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

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.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación

Java (J2SE, web, EJB¹ y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

B. JAVA

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, runanywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

C. JAVA SERVER FACES (JSF)

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServerPages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo

¹ Enterprise Java Beans

de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

La especificación de JSF fue desarrollada por la Java Community Process, que definía JSF 1.0 y 1.1, JSR 252 que define JSF 1.2 y JSR 314 para JSF 2.0

D. MVC (MODELO VISTA CONTROLADOR)

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre la Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a

datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

E. FRAMEWORK PRIMEFACES

PrimeFaces es una librería de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces.

Propiedades.

- Conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte de ajax con despliegue parcial, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.
- 25 temas prediseñados
- Componente para desarrollar aplicaciones web para móviles-celulares, especiales para Iphones, Palm, Android y teléfonos móviles Nokia.

Existen las siguientes versiones.

- Primefaces 1: Trabaja con JSF 1.2
- Primefaces 2: Trabaja con JSF 2
- Primefaces 3: Trabaja con JSF 2
- Primefaces 4: Trabaja con JSF 2
- Primefaces 5: Trabaja con JSF 2.2

Ventajas Y Desventajas De Primefaces

En cuanto a la experiencia de los usuarios finales los componentes de Primefaces son amigables al usuario además que cuentan con un diseño innovador. Pero en lo que respecta al programador, es que sus desarrolladores no respetan un principio básico del desarrollo de componentes: la compatibilidad hacia atrás, es decir, un componente de una nueva versión de Primefaces por lo general no es compatible al 100% con una aplicación desarrollada con la versión previa a la misma.

F) GESTOR DE BASE DE DATOS POSTGRESQL

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Características De Postgresql.

La última serie de producción es la 9.3. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

G) GLASSFISH

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 Enterprise Edition (EE) y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código libre y se distribuye bajo un licenciamiento. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation; este 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.

2.6 CALIDAD DE SOFTWARE ISO 9126

2.6.1 ISO 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos

El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características. (SQuare, 2005)

Subcaracterísticas.

a) Funcionalidad

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

- Adecuación - Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.
- Exactitud - Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- Interoperabilidad - Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados.
- Seguridad - Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos.
- Cumplimiento funcional.

b) Fiabilidad

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

- Madurez - Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.

- Recuperabilidad - Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- Tolerancia a fallos - Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.
- Cumplimiento de Fiabilidad - La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.

c) Usabilidad

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- Aprendizaje- Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Comprensión - Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Operatividad - Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.
- Atractividad.

d) Eficiencia

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

- Comportamiento en el tiempo - Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.
- Comportamiento de recursos - Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

e) Mantenibilidad

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

- Estabilidad - Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.
- Facilidad de análisis - Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
- Facilidad de cambio - Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
- Facilidad de pruebas - Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.

f) Portabilidad

Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.

- Capacidad de instalación - Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
- Capacidad de reemplazamiento - Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado.
- Adaptabilidad - Atributos del software relacionados con la oportunidad para su adaptación a diferentes ambientes especificados sin aplicar otras acciones o medios que los proporcionados para este propósito por el software considerado.
- Co-Existencia - Coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes. (DBpedia , 2005)

2.7 ESTUDIO DE COSTOS Y BENEFICIO

2.7.1 Estimación De Costos De Desarrollo De Software

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismos, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente.

La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo incidiendo en la gestión

empresarial. En la actualidad existen un conjunto de métricas que no se utilizan, y que pueden ser aplicables a cualquier tipo de proyecto de software para calcular el costo de los mismos.

2.7.2 Cocomo

El Modelo COCOMO Creado por Barry Boehm en 1981. Su nombre significa COConstructiveCOst MOdel (Modelo constructivo de costo) constituye una jerarquía de modelos de estimación para el software. La jerarquía está constituida por los siguientes modelos (COCOMO Model definition manual., 1990).

- COCOMO básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LOC.
- COCOMO intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- COCOMO detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

COCOMO es una herramienta basada en las líneas de código la cual le hace muy poderoso para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño. Es necesario para un administrador de proyectos una herramienta de estimación de costos; y esta herramienta puede estar relacionada con COCOMO ya que esta técnica representa uno de los más completos modelos empíricos para la estimación de software (Bumett, 1998).

Una de las deficiencias detectadas en el modelo COCOMO es que para el análisis del costo del proyecto solo analizan el salario del desarrollador sin tener en cuenta otros elementos de gastos que inciden en los costos del proyecto de software. (Prentice, 1981)

2.7.3 Líneas De Código Y Puntos De Función.

Las métricas para puntos de función están basadas en las guías proporcionadas por el International Function Point User Group (IEPUG, 1994). Los puntos de función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en

información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

Los datos de líneas de código (LDC) y los puntos de función (PF) se emplean de dos formas durante la estimación del proyecto de software:

- Variables de estimación, utilizadas para calibrar cada elemento del software.
- Métricas de base, recogidas de anteriores proyectos utilizadas junto con las variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Estas técnicas son diferentes pero tienen características comunes. El planificador del proyecto comienza con una declaración restringida del ámbito del software y, a partir de esa declaración, intenta descomponer el software en pequeñas subfunciones que pueden ser estimadas individualmente. Entonces, estima las LDC o PF (la variable de estimación) para cada subfunción. Luego, aplica las métricas básicas de productividad a la variable de estimación apropiada y deriva el costo y el esfuerzo para la subfunción. Combinando las estimaciones de las subfunciones se produce la estimación total para el proyecto entero.

Difieren en el nivel de detalle que requiere la descomposición, cuando se utiliza LDC como variable de estimación, la descomposición funcional es absolutamente esencial y, a menudo, se lleva hasta considerables niveles de detalle. Actualmente esta técnica no refleja el costo real de un proyecto por el análisis de sus líneas de código, puede existir un proyecto con pocas líneas de código y que el esfuerzo, tiempo, recursos humanos y materiales dedicados al mismo sea superior a otro proyecto que fuese desarrollado con anterioridad y cuente con más líneas de código. (Pressman, 1998)

2.8 SEGURIDAD

Las Pruebas de Seguridad son un proceso que permite verificar que un sistema de información protege sus datos y funciona de acuerdo a los propósitos para los que fue diseñado.

Por tanto, las Pruebas de Seguridad verifican que la aplicación y la infraestructura que la soporta no evidencian vulnerabilidades que puedan ser aprovechadas por terceros para uso no deseado.

Las Pruebas de Seguridad se enfocan, estratégicamente, en uno o varios de los siguientes aspectos de la información:

- **Confidencialidad de la información:** evitar que usuarios o sistemas no autorizados accedan a la información.
- **Integridad de la información:** asegurar la exactitud y la completitud de la información, así como los métodos que se utilizan para su procesamiento.
- **Disponibilidad de la información:** asegurar que la información esté disponible para usuarios y sistemas autorizados en el momento que lo requieran.

2.8.1 Beneficios De Pruebas De Seguridad

Al realizar procesos formales de Pruebas de Seguridad Usted percibe los siguientes beneficios:

- Aprovechamiento de la base de conocimiento desarrollada por Avantica a lo largo de años de experiencia.
- Medir la exposición real a posibles ataques de hackers.
- Controlar accesos no autorizados a información sensible.
- Contar con estrategias para la mitigación de riesgo de explotación de vulnerabilidad.
- Asegurar que los riesgos de seguridad identificados se mantienen en niveles aceptables y medibles y por tanto lograr protección real de los datos.

2.8.2 COMPROMETERSE A HACER DE LA SEGURIDAD UNA PARTE DE LA ADQUISICIÓN

Al igual que con el desarrollo interno, los usuarios de negocios primero tienen que querer la seguridad como un aspecto necesario de la calidad general. Una parte crítica de la creación de este compromiso es comunicarse con ellos acerca de los riesgos y beneficios, y establecer de objetivos mutuamente acordados.

2.8.3 ENCONTRAR LOS PUNTOS CORRECTOS DE CONTACTO DEL PROCESO

Los controles de seguridad obligatorios en la gestión de proyectos, revisión de contratos y contratación son una necesidad. Pero ya que la disponibilidad de software gratuito y de pago a medida hace que sea fácil para los usuarios individuales de negocios adquirir el software por sí mismos, probablemente se necesitarán herramientas adicionales para descubrir el software instalado y el uso de aplicaciones de nube.

2.8.4 ACTIVIDADES DE PRUEBA A MEDIDA PARA EL SOFTWARE Y EL TIPO ABASTECIMIENTO

Todo el software debe ser probado respecto a su seguridad al menos una vez antes de entrar en producción, pero no todas las necesidades del software necesitan ser probadas con la misma frecuencia o en la misma profundidad. Busque alguna combinación de los siguientes: evidencia del proceso y prueba del proveedor; evidencia de pruebas de terceros; validaciones y certificaciones de terceros (por lo que valen); pruebas internas durante la codificación y las pruebas (por algún tercero); pruebas internas antes del lanzamiento; y pruebas de software una vez desplegado en producción. En todos los casos, haga las pruebas frente a requisitos documentados.

2.9 PRUEBAS

2.9.1 Fundamentos

En la etapa de prueba del software se crean una serie de casos de prueba que intentan "destruir" el software desarrollado.

La prueba requiere que se descarten ideas preconcebidas sobre la "calidad o corrección" del software desarrollado.

2.9.2 Objetivos De La Prueba

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces
- Una prueba tiene éxito si descubre un error no detectado hasta entonces

El objetivo es diseñar casos de prueba que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo.

La prueba no puede asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software.

2.9.3 Prueba De Caja Blanca

En programación, se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra

ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles (se verifican los bucles para 0,1 e interacciones, y luego para las interacciones máximas, máximas menos uno y más uno).

Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración).

En los sistemas orientados a objetos, las pruebas de caja blanca pueden aplicarse a los métodos de la clase, pero según varias opiniones, ese esfuerzo debería dedicarse a otro tipo de pruebas más especializadas (un argumento podría ser que los métodos de una clase suelen ser menos complejos que los de una función de programación estructurada). Dentro de las Pruebas de Caja Blanca encontramos las llamadas coberturas (sentencia, decisión, condición y múltiple además de los mencionados caminos ciclomáticos).

2.9.4 Prueba De Caja Negra

En teoría de sistemas y física, se denomina Caja Negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea (en ocasiones, otros elementos que también podrían ser cajas negras) entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento. (Perez Darling, 2008)

MARCO APLICATIVO

En el presente capítulo se mostrara el análisis, diseño e implementación, con el objetivo de identificar y resolver los problemas en la administración digital de la información almacenada en las planillas de haberes.

3.1 FASE DE INICIO

Se realizara un estudio del modelado de negocio y los requerimientos funcionales y no funciones, identificando los casos de uso.

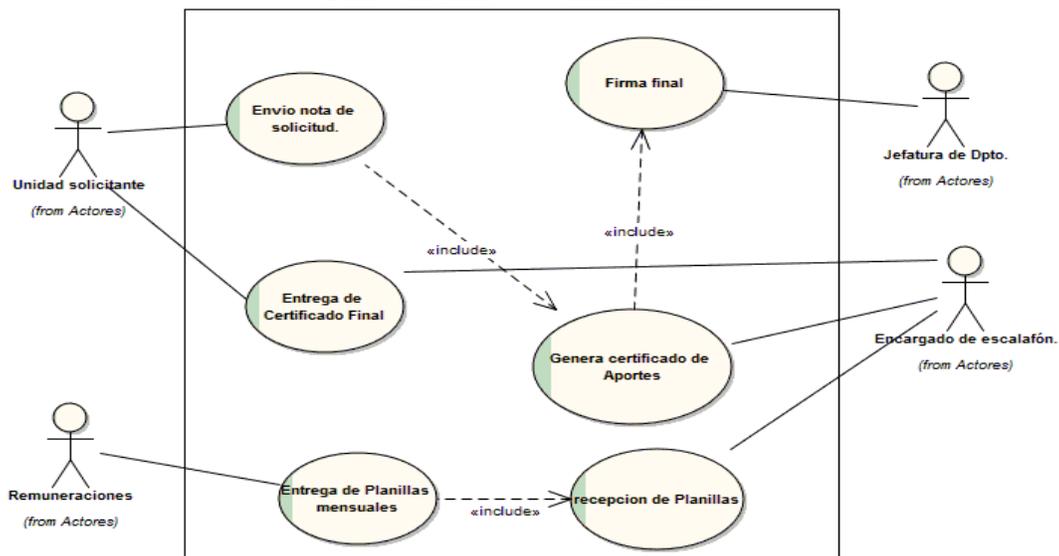
3.1.1 Modelado Del Negocio

Nos permite capturar los requisitos requeridos para desarrollar el sistema, de acuerdo a las necesidades del usuario:

- Se identifican los casos de uso de los actores
- Se tiene el modelo de actividades del negocio.
- Se identifican las acciones de cada caso de uso.

3.1.1.1 Modelado De Casos De Uso Del Negocio

Figura 3.1 Modelo de casos de Uso



Fuente (Elaboración propia)

3.1.1.2 Descripción De Los Actores Del Negocio

A) UNIDAD SOLICITANTE (FUNCIONARIO). Es la persona que solicita información, almacenada en las planillas de haberes, en un resumen (certificado de Aportes).

B) REMUNERACIONES. Es la unidad que se encarga de proveer de las planillas de haberes mensualmente a las distintas unidades, divisiones en este caso a escalafón.

C) JEFATURA DE DEPARTAMENTO. Es la que decepciona las solicitudes, y el q realiza la previa firma del Jefe de RRHH previa entrega de los certificado de aportes.

D) ENCARGADO DE ESCALAFÓN. En esta unidad se genera los reportes (Certificado de Aportes), esta acción se realiza revisando planilla por planilla los datos requeridos, de esta manera se completa la solicitud, una vez concluido el trabajo, deriva el certificado a la Jefatura de Departamento para la firma final.

3.1.1.3 Descripción De Los Casos De Uso Del Negocio

Tabla 3.1 Descripción de caso de uso: Envío y recepción de solicitud.

Casos de Uso	Envío y recepción de solicitud.
Actores.	Unidad solicitante, Jefatura de departamento.
Propósito.	El propósito de este caso es iniciar con el proceso de la solicitud.
Resumen.	La unidad solicitante realiza el pago de valores por el respectivo certificado de aportes y realiza una nota (Carta) dirigida al jefe en turno de Recursos Humanos, adjuntando la boleta de pago, procede a la entrega en recepción de la jefatura.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor (Funcionario)	Respuesta del Sistema (Jefatura)
1. Realiza el pago de valores por un certificado de Aportes.	4. Recepciona la solicitud. 5. Deriva el tramite (Solicitud), a la unidad de escalafón administrativo.

<p>2. Realiza una nota de solicitud de un certificado de aportes.</p> <p>3. Realiza la entrega de la nota escrita</p>	
---	--

Fuente: [Elaboración Propia]

Tabla 3.2 Descripción de caso de uso: Recepción de Solicitud y Realización de certificado de aportes.

Casos de Uso	Recepción de Solicitud y Realización de certificado de aportes.
Actores.	Jefatura de departamento, Encargado de escalafón.
Propósito.	Continuar el proceso e iniciar el llenado de reporte solicitado.
Resumen.	Se procede al llenado de datos en el reporte de la información almacenada en las planillas de haberes.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. El Mensajero perteneciente a la jefatura de Recursos Humanos deriva dicha solicitud.	<p>2. el encargado de escalafón recepciona la solicitud.</p> <p>3. El Encargado de escalafón procede a revisar las planillas de haberes mensuales y llenar el certificado de aportes.</p> <p>4. Una vez concluido el certificado se envía a jefatura para posterior Firma final y entrega del certificado al interesado.</p>

Fuente: [Elaboración Propia]

Tabla 3.3 Descripción de caso de uso: Entrega y recepción de Planillas de haberes.

Casos de Uso	Entrega y recepción de Planillas de haberes.
Actores.	Remuneraciones, Encargado de escalafón.
Propósito.	Alimentar de Información a la unidad de escalafón.
Resumen.	Remuneraciones realiza la entrega de las planillas de haberes, y planillas alfabéticas al encargado de escalafón.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor(Remuneraciones)	Respuesta del Sistema (Encargado de Escalafón)
1. El mensajero de remuneraciones realiza la entrega de Planillas de haberes, alfabéticas al encargado de escalafón	2. El encargado de escalafón recepciona las planillas y procede a su almacenamiento en la unidad.

Fuente: [Elaboración Propia]

3.1.2 REQUERIMIENTOS

Se hace una aplicación más extensa de los casos de uso profundizando el uso de esta herramienta para detallar la arquitectura del sistema.

3.1.2.1 Requerimientos Funcionales

En las siguientes tablas se reflejan las funciones del sistema, donde la primera columna hace referencia a la cantidad de funciones en si que engloba un módulo, la tercera columna muestra las clasificaciones que pueden tener cada función, y entre ellas:

- **Evidente.** Función que debe realizarse, y el usuario debe saber que se ha realizado
- **Oculto.** Debe Realizarse, aunque no es visible para los usuarios.
- **Superflua.** Opcionales, su inclusión no repercute significativamente en el costo ni en otras funciones.

Tabla 3.4 Requerimientos del Sistema

REF. #	FUNCIÓN	CATEGORÍA
R1.1	El sistema debe ser fácil, el uso por parte del usuario.	Evidente
R1.2	El sistema debe dar reportes actualizados.	Evidente
R1.4	El sistema debe tener la base de datos actualizados para dar respuesta a las consultas ingresadas por el usuario.	Evidente

Fuente (Elaboración propia)

3.1.2.2 Requerimientos Funcionales Del Sistema

Se identifican los requerimientos funcionales que hará como proceso, mostrando los eventos que se desarrolla dentro del sistema.

Tabla 3.5 Registro de datos Funcionarios, Planillas, usuarios

REF. #	FUNCIÓN	CATEGORÍA
R2.1	El sistema proporciona formularios de llenado de datos	Evidente
R2.2	El sistema verifica el llenado del formulario	Evidente
R2.3	El sistema registra y almacena los datos insertados	Ocultas
R2.4	Se registra usuarios, Funcionarios, Planillas	Evidente

Fuente (Elaboración propia)

Tabla 3.6 Muestra registros, de búsqueda de funcionarios.

REF. #	FUNCIÓN	CATEGORÍA
R3.1	Mostrar información sobre seguimiento administrativo	Evidente
R3.2	Mostrar Planillas en las cuales puede Adicionarse al funcionario.	Evidente
R3.3	Emitir listas.	Evidente
R3.4	Cumplir con las búsquedas necesarias.	Evidente

Fuente (Elaboración propia)

Tabla 3.7 Emisión de Certificado de aportes

REF. #	FUNCIÓN	CATEGORÍA
R3.1	El sistema Muestra Listado de funcionarios	Evidente
R3.2	El sistema muestra datos almacenados en las planillas del funcionarios seleccionado	Evidente
R3.3	El sistema retorna reporte aportes	Evidente
R3.4	El sistema exporta el reporte	Evidente

Fuente (Elaboración propia)

3.1.2.3 Requerimientos No Funcionales

Tabla 3.8 Emisión de Certificado de aportes

REF. #	FUNCIÓN	CATEGORÍA
R3.1	El sistema Muestra Listado de funcionarios	Evidente
R3.2	El sistema muestra datos almacenados en las planillas del funcionarios seleccionado	Evidente
R3.3	El sistema retorna reporte aportes	Evidente
R3.4	El sistema exporta el reporte	Evidente

Fuente (Elaboración propia)

3.1.3 Descripción De Los Actores

a) ADMINISTRADOR. Es la persona q se hace responsable del Sistema, en modo administrador. Este tipo de usuario:

- Accede al sistema para obtener reportes Actualizados de los Funcionarios, Planillas, Aportes, descuentos, etc.
- Adicionara / Modificar / eliminar Usuarios.
- Adicionar / eliminar Permisos a los tipos de Usuarios.
- Adicionara / Modificar / Eliminar Funcionarios.
- Adicionar / Modificar / eliminar Planillas.

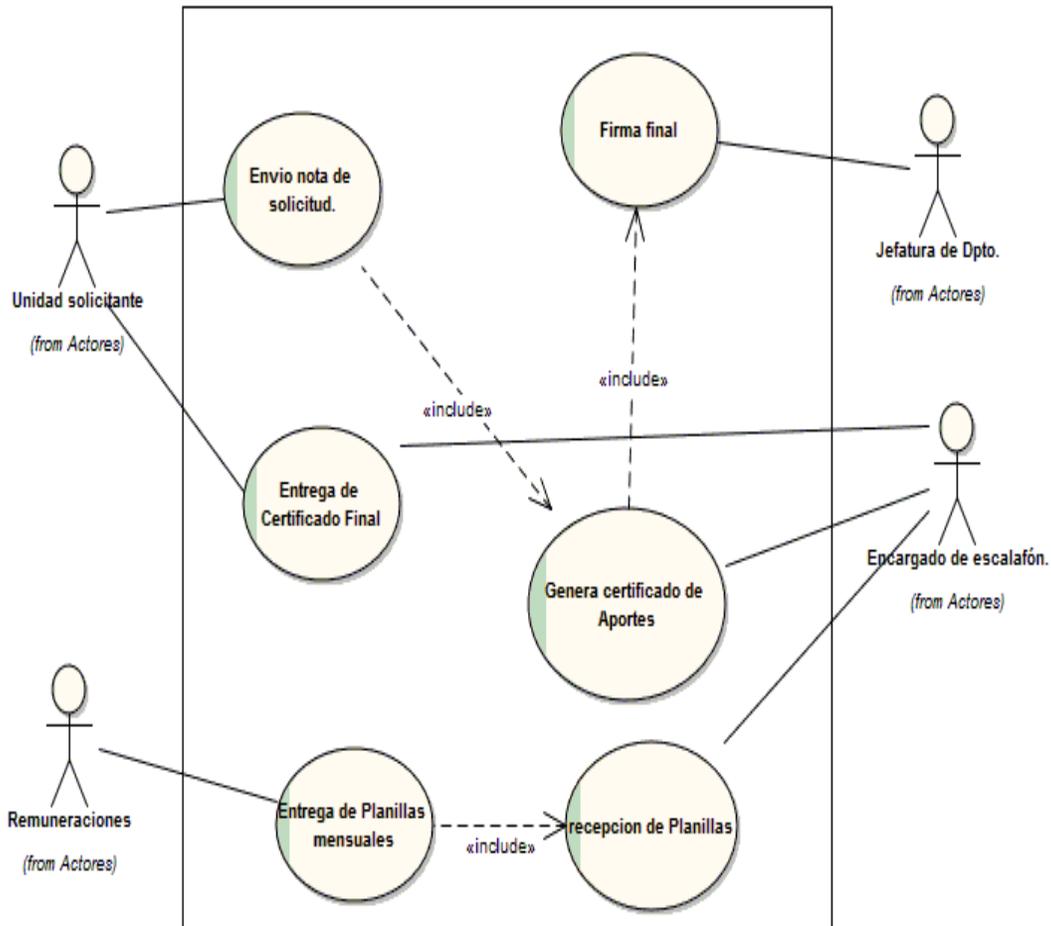
b) TRANSCRIPTOR (USUARIO COMÚN). Es la persona que accede al sistema para realizar la inserción de datos, encargado de Alimentar la Base de Datos, además de:

- Adicionar / eliminar Permisos a los tipos de Usuarios.
- Adicionara / Modificar / Eliminar Funcionarios.

- Adicionar / Modificar / eliminar Planillas.

3.1.3.1 Diagrama De Casos De Uso

Figura 3.2 Diagrama Casos de Uso



Fuente (Elaboración Propia)

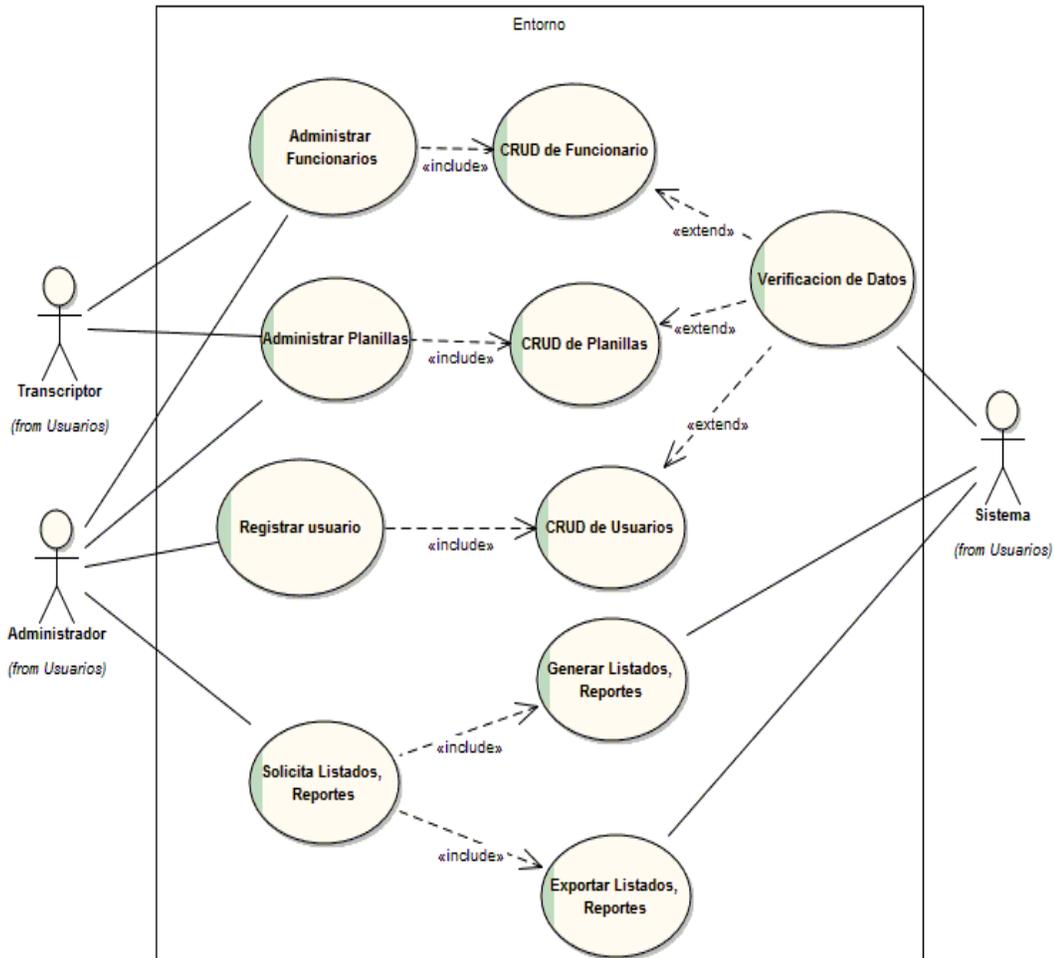
3.2 FASE DE ELABORACIÓN

En esta Fase todos los componentes se desarrollan e incorporan al sistema donde los requerimientos son estables y detallados.

3.2.1 Análisis Y Diseño

3.2.1.1 Diagrama De Casos Del Sistema

Figura 3.3 Diagrama Casos de uso del sistema



Fuente (Elaboración Propia)

3.2.1.2 Descripción De Los Actores Del Sistema

a) **ADMINISTRADOR.** Es la persona q se hace responsable del Sistema, en modo administrador. Este tipo de usuario:

- Accede al sistema para obtener reportes Actualizados de los Funcionarios, Planillas, Aportes, descuentos, etc.
- Adicionara / Modificar / eliminar Usuarios.
- Adicionar / eliminar Permisos a los tipos de Usuarios.

- Adicionara / Modificar / Eliminar Funcionarios.
- Adicionar / Modificar / eliminar Planillas.

b) TRANSCRIPTOR (USUARIO COMÚN). Es la persona q accede al sistema para realizar la inserción de datos, encargado de Alimentar la Base de Datos, además de:

- Adicionar / eliminar Permisos a los tipos de Usuarios.
- Adicionara / Modificar / Eliminar Funcionarios.
- Adicionar / Modificar / eliminar Planillas.

3.2.1.3 Diagrama De Clases

Figura 3.4 Diagrama de Clases



Fuete (Elaboración Propia)

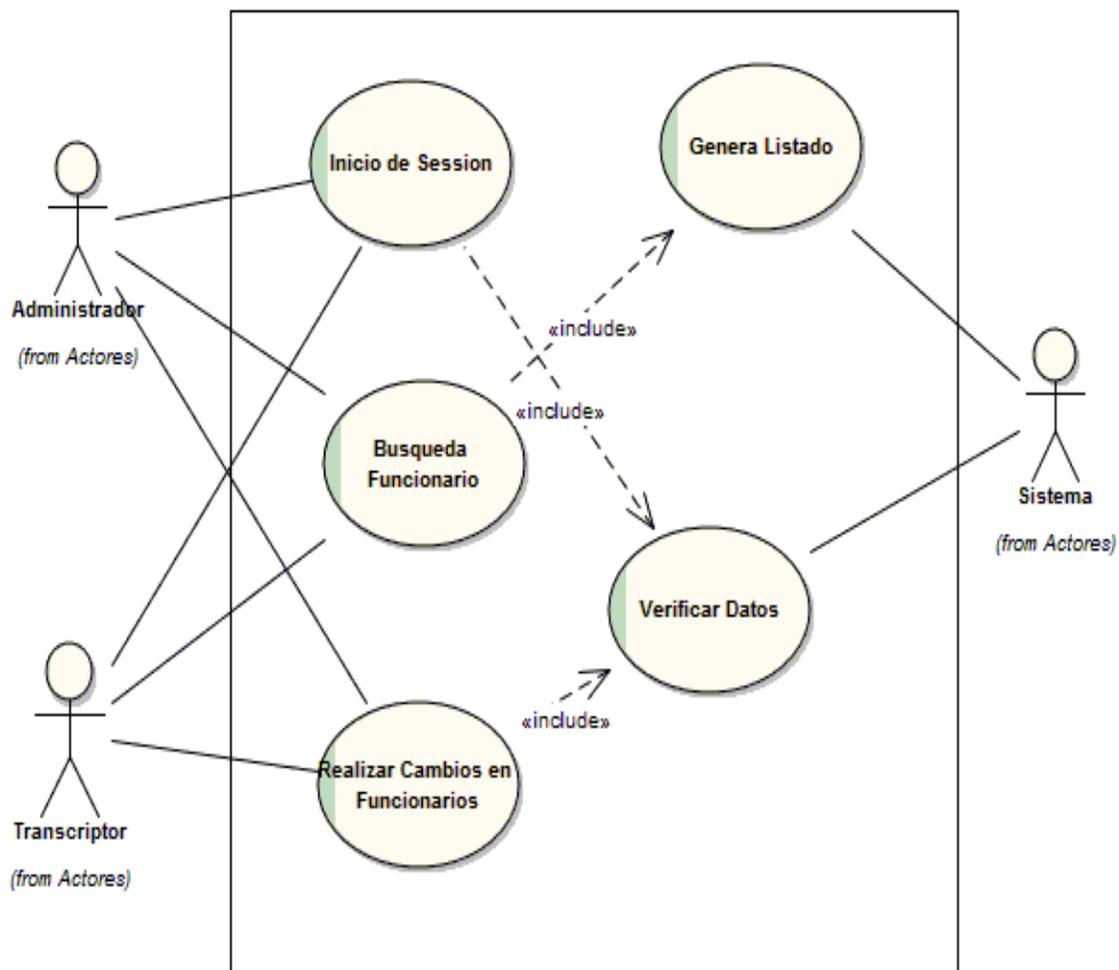
3.2.1.4 Diagrama De Despliegue

Es un tipo de diagrama del Lenguaje Unificado de Modelado (UML), que se utiliza para modelar la disposición física de los artefactos software en nodos (Usualmente plataforma de hardware).

3.2.1.4.1 Descripción De Casos De Uso Expandido

a) Administración De Funcionarios

Figura 3.6 (Caso de Uso expandido, administra Gestión de Funcionarios)



FUENTE (Elaboración Propia)

Tabla 3.9 Descripción de caso de uso: Administrar Funcionarios (CRUD) Funcionarios.

Casos de Uso	Administrar Funcionarios (CRUD) Funcionarios
Actores.	Administrador. Transcriptor.
Propósito.	Realizar Modificaciones En La Base De Datos, En La Gestión Funcionario
Resumen.	Los usuarios, realizada Altas, Bajas y Cambios de la información de los funcionarios.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita iniciar sesión 2. Solicita listado de Funcionarios 3. Solicita formulario de acción (adición, modificación). 5. Llena debidamente los datos requeridos por el formulario.	2. Verifica los permisos de usuario (login). 4. Entrega formulario solicitado (adición, modificación). 6. Se registran los datos recepcionados.

b) Administración De Planillas

Tabla 3.10 Descripción de caso de uso: Administrar Planillas (CRUD) Planillas.

Casos de Uso	Administrar Planillas (CRUD) Planillas
Actores.	Administrador. Transcriptor.
Propósito.	Realizar modificaciones en la base de datos, en la gestión planillas
Resumen.	Realizar Inserciones de nuevas planillas.
Tipo	Primario
Curso Normal de eventos	

Acción del Actor	Respuesta del Sistema
1. Solicita iniciar sesión como administrador. 2. Solicita listado de Entorno planilla. 3. Solicita formulario de acción (Creación, modificación). 5. Llena debidamente los datos requeridos por el formulario.	2. Verifica los permisos de usuario (login). 4. Entrega formulario solicitado (adición, modificación). 6. Se registran los datos decepcionados.

c) Administración De Usuarios

Tabla 3.11 Descripción de caso de uso: Administrar Usuarios (CRUD)

Casos de Uso	Administrar Usuarios (CRUD)
Actores.	Administrador.
Propósito.	Realizar Modificaciones En La Base De Datos, En La Gestión Usuarios.
Resumen.	Los usuarios, realizada Altas, Bajas y Cambios de la información de los usuarios.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita iniciar sesión 2. Solicita listado de Usuarios. 4. Solicita formulario de acción (adición, modificación). 6. Llena debidamente los datos requeridos por el formulario.	2. Verifica los permisos de usuario (login). 3. Retorna listado. 5. Entrega formulario solicitado (adición, modificación). 7. Se registran los datos recepcionados.

d) Administración De Reportes

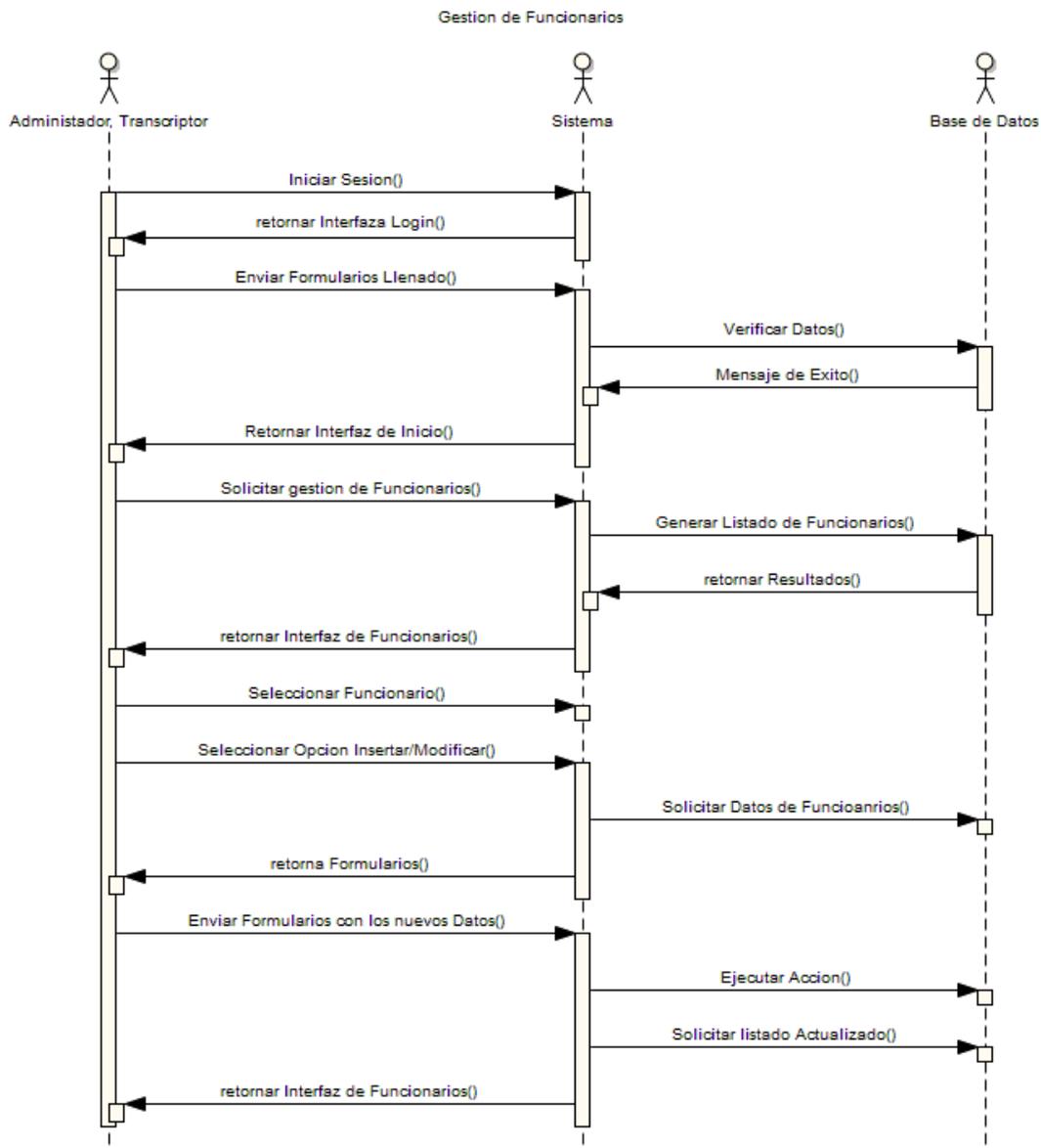
Tabla 3.12 Descripción de caso de uso: Generar listados, reportes.

Casos de Uso	Generar listados, reportes.
Actores.	Administrador.
Propósito.	Conocer, visualmente y mediante un reporte los aportes de los funcionarios realizados durante su carrera administrativa.
Resumen.	Los funcionarios activos e inactivos Solicitan un Certificado de Aportes, a las instancias correspondientes.
Tipo	Primario
Curso Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador realiza la recepción de la solicitud escrita. 2. Solicita iniciar sesión como administrador. 4. El administrador busca el funcionario, y solicita el reporte de aportes.	3. Verifica los permisos de Usuario (Login). 5. En el certificado de Aportes es procesado y posteriormente presentado en forma de reporte.

3.2.1.5 Diagrama De Secuencia De Sistema

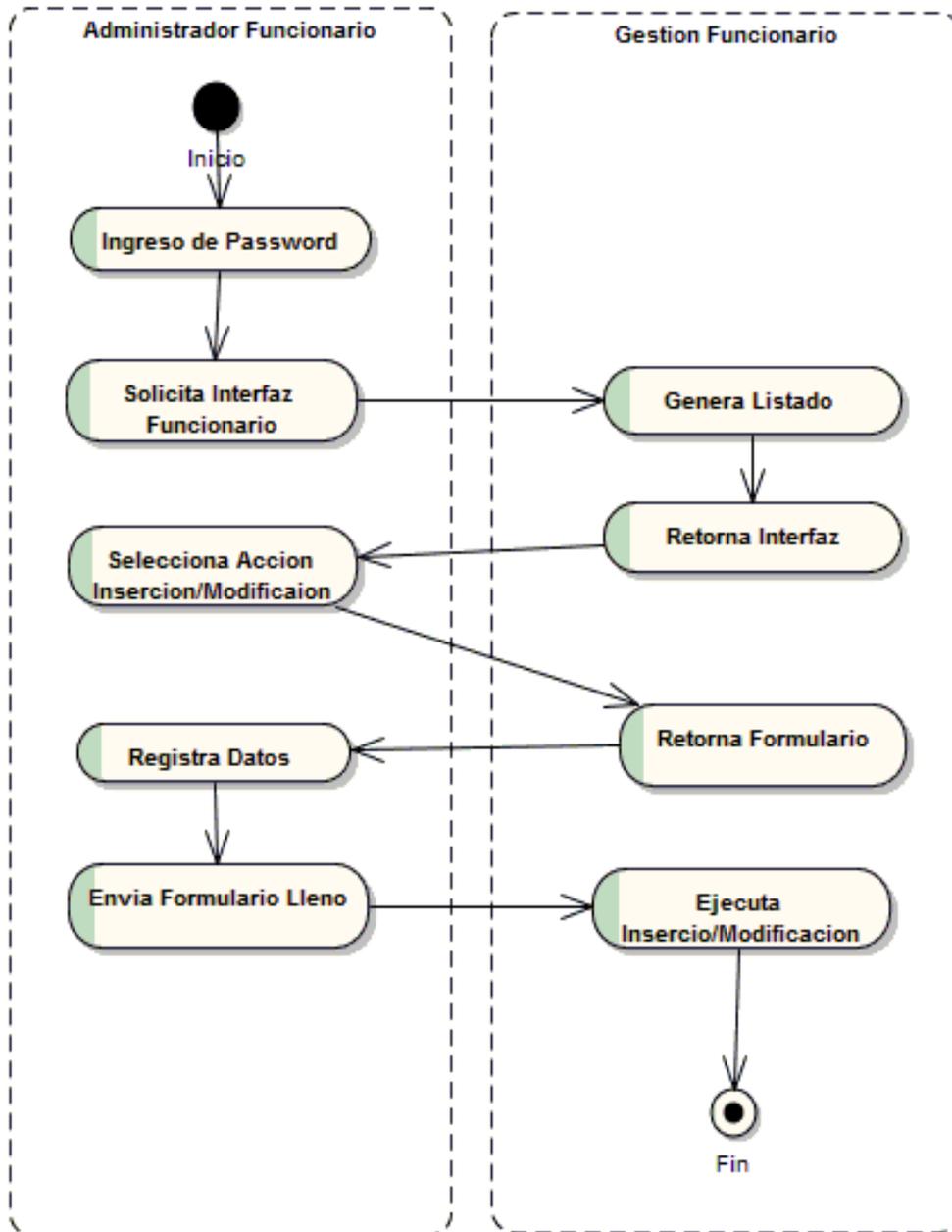
a) Gestión De Funcionarios

Figura 3.7 (Diagrama de Secuencia: Gestión de Funcionarios)



Fuente (Elaboración Propia)

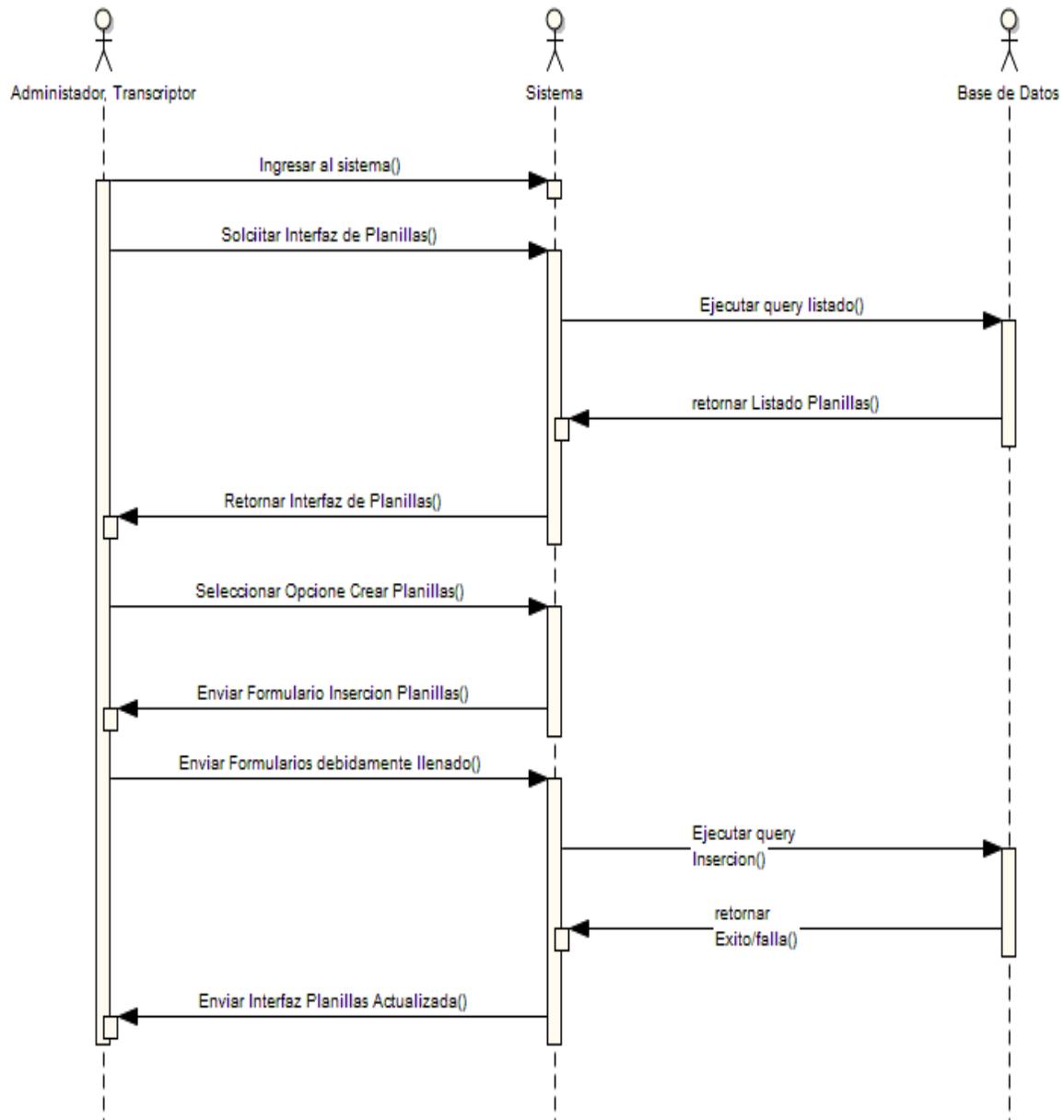
Figura 3.8 (Diagrama de Actividades)



Fuente (Elaboración Propia)

b) Gestión De Planillas

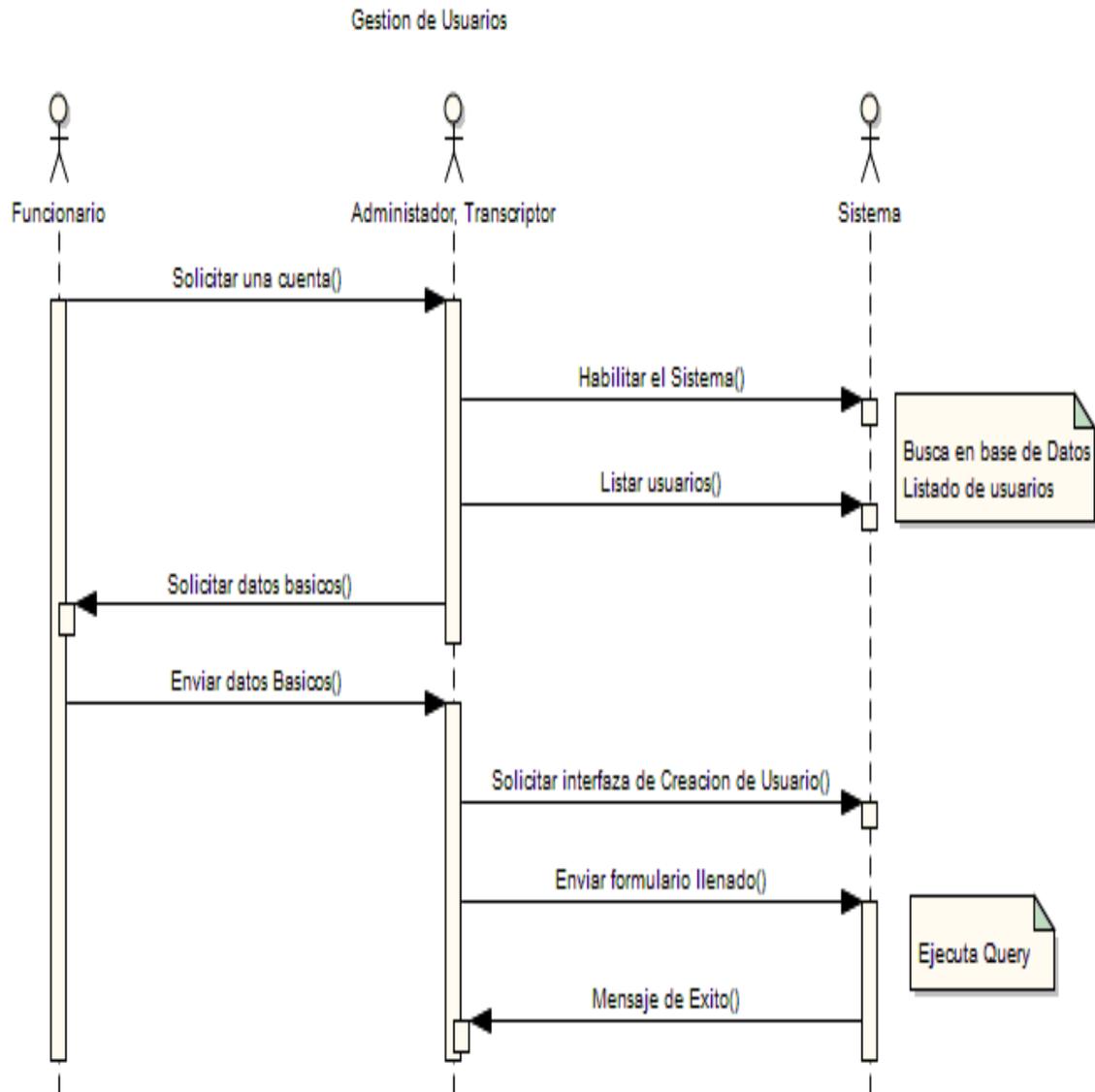
Figura 3.9 (Diagrama de Secuencia: Gestión de Planillas)



Fuente (Elaboración Propia)

c) Gestión De Usuarios

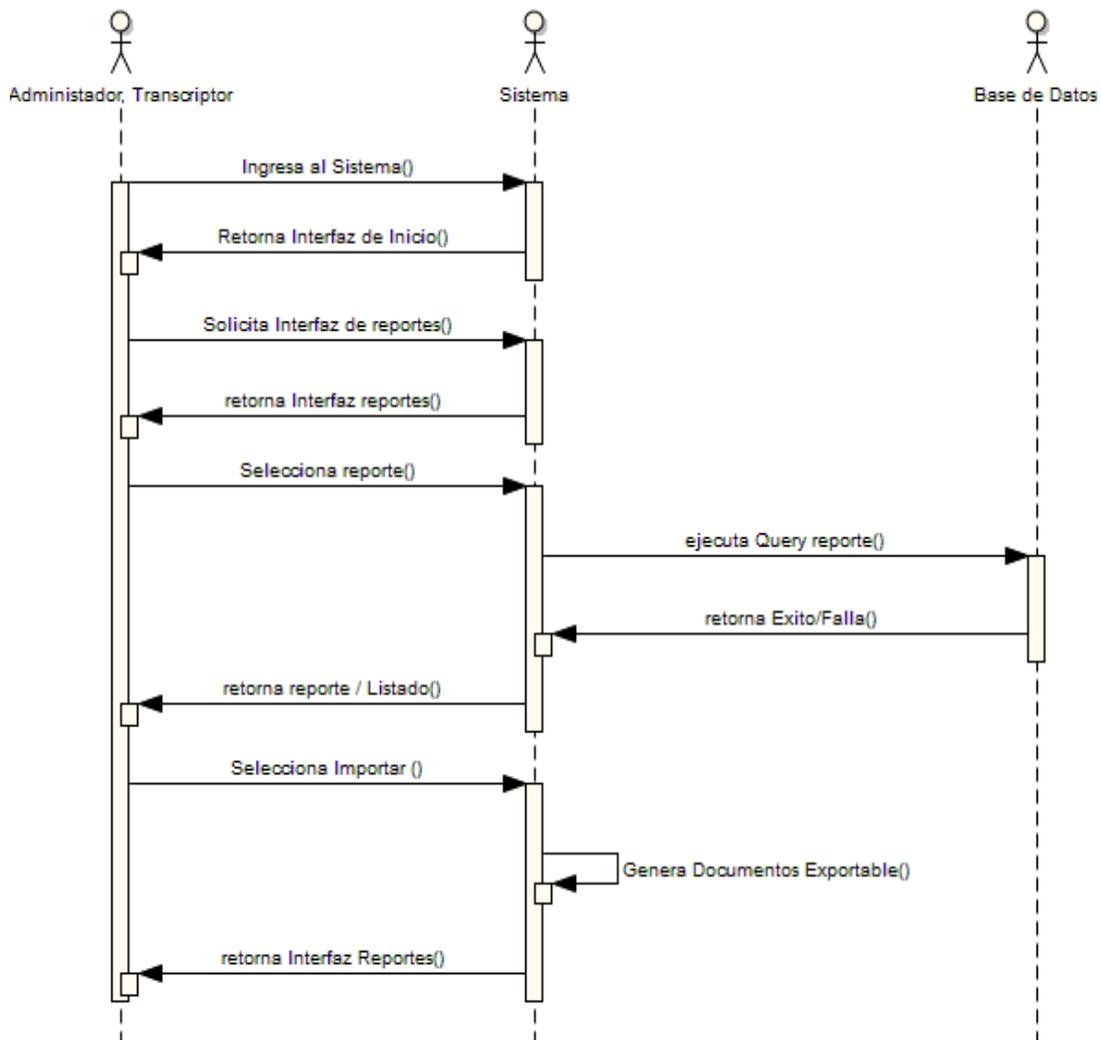
Figura 3.10 (Diagrama de Secuencia: Gestión de Usuarios)



Fuente (Elaboración Propia)

d) Gestión De Reportes

Figura 3.11 (Diagrama de Secuencia: Gestión de Reportes)

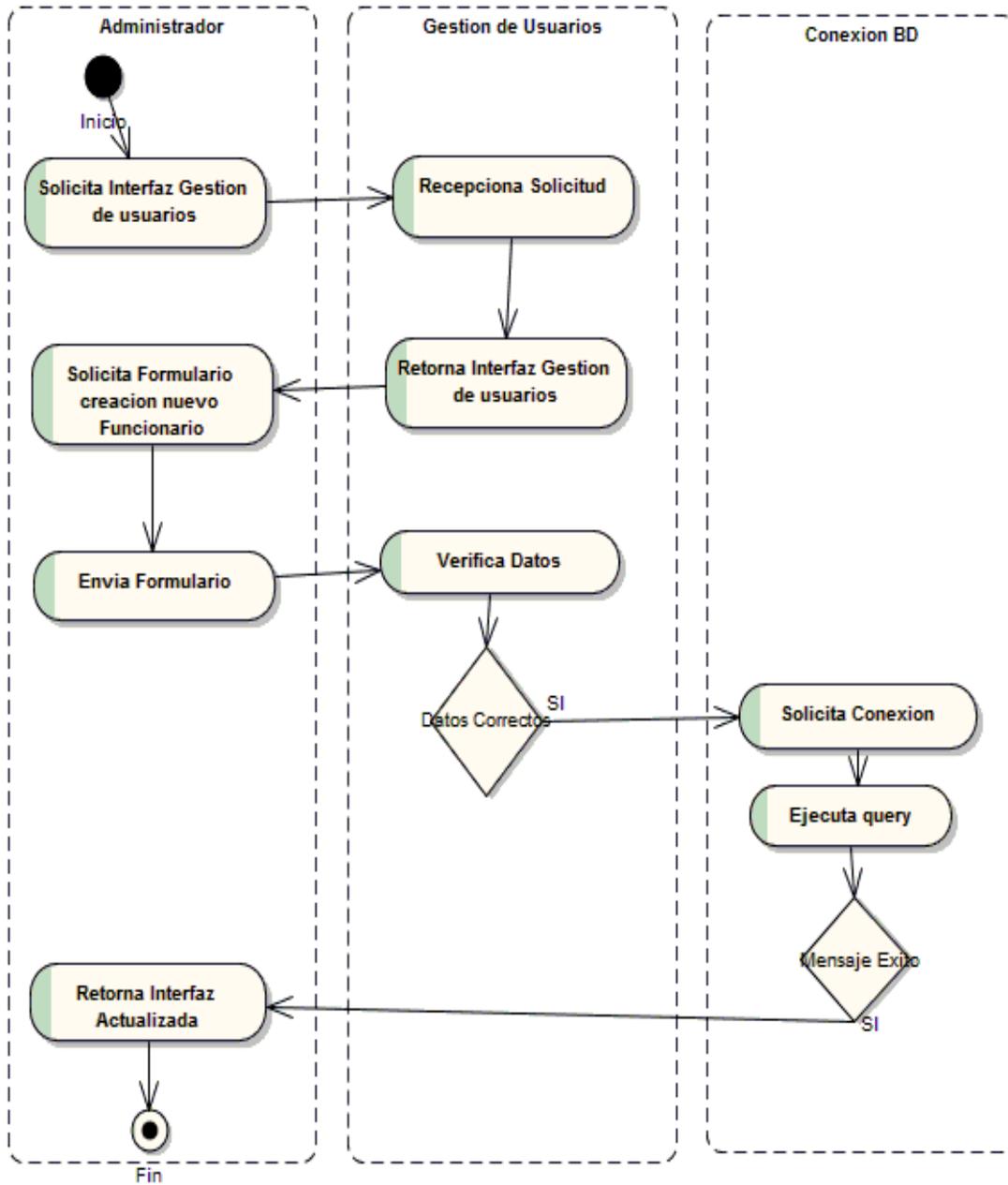


Fuente (Elaboración Propia)

3.2.1.6 Diagrama De Actividades

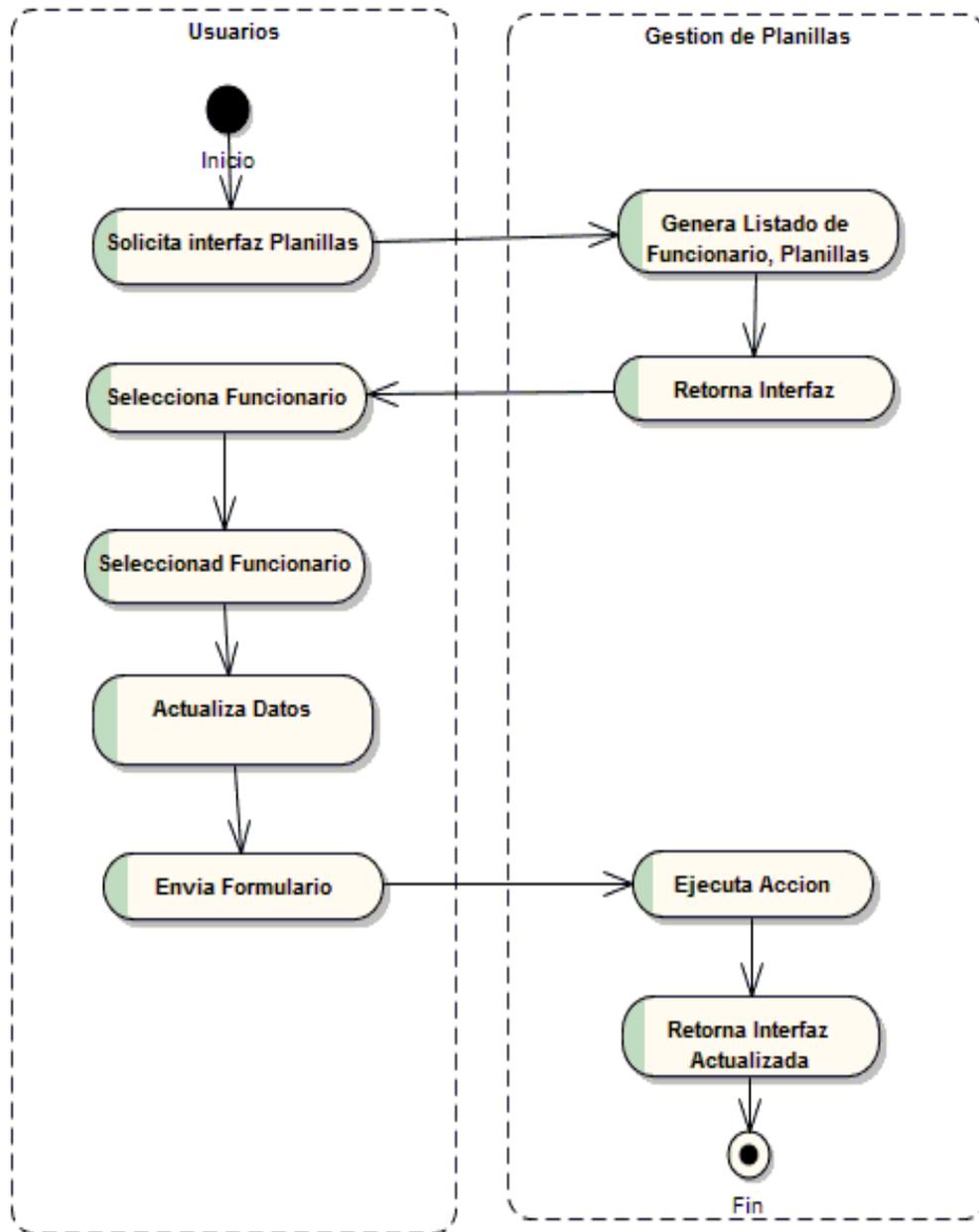
En las siguientes figuras se representa los diagramas de actividades en formato carril:

Figura 3.12 (Diagrama de Actividades Gestión de Usuarios)



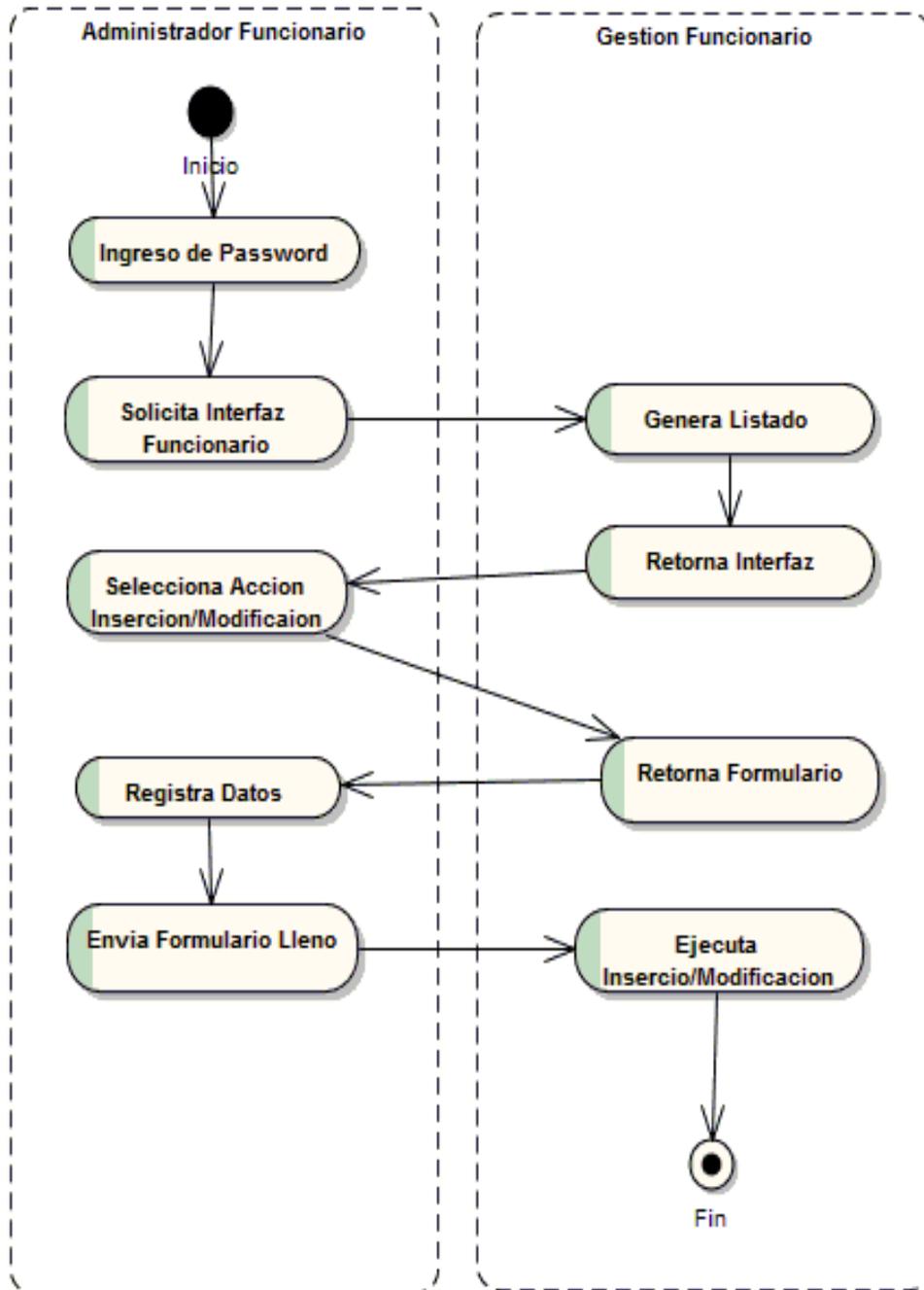
FUENTE (Elaboración Propia)

Figura 3.13 (Diagrama de Actividades Gestión de Planillas)



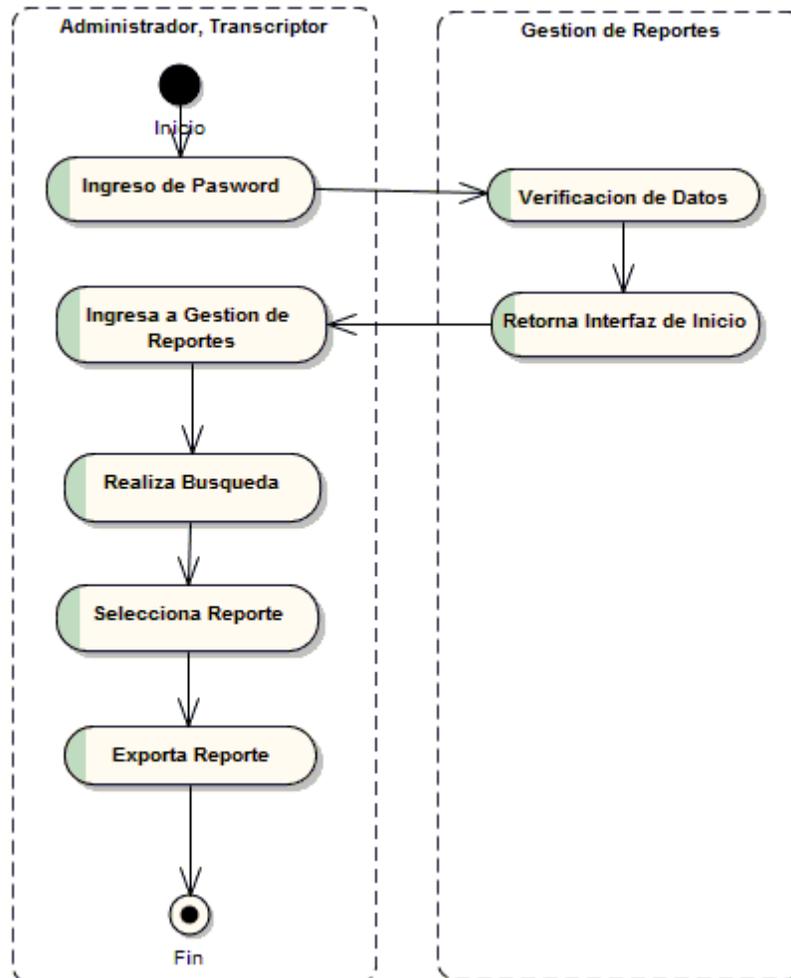
FUENTE (Elaboración Propia)

Figura 3.14 (Diagrama de Actividades Gestión de Funcionarios)



FUENTE (Elaboración Propia)

Figura 3.15 (Diagrama de Actividades Gestión de Reportes)



FUENTE (Elaboración Propia)

3.3 FASE DE CONSTRUCCIÓN

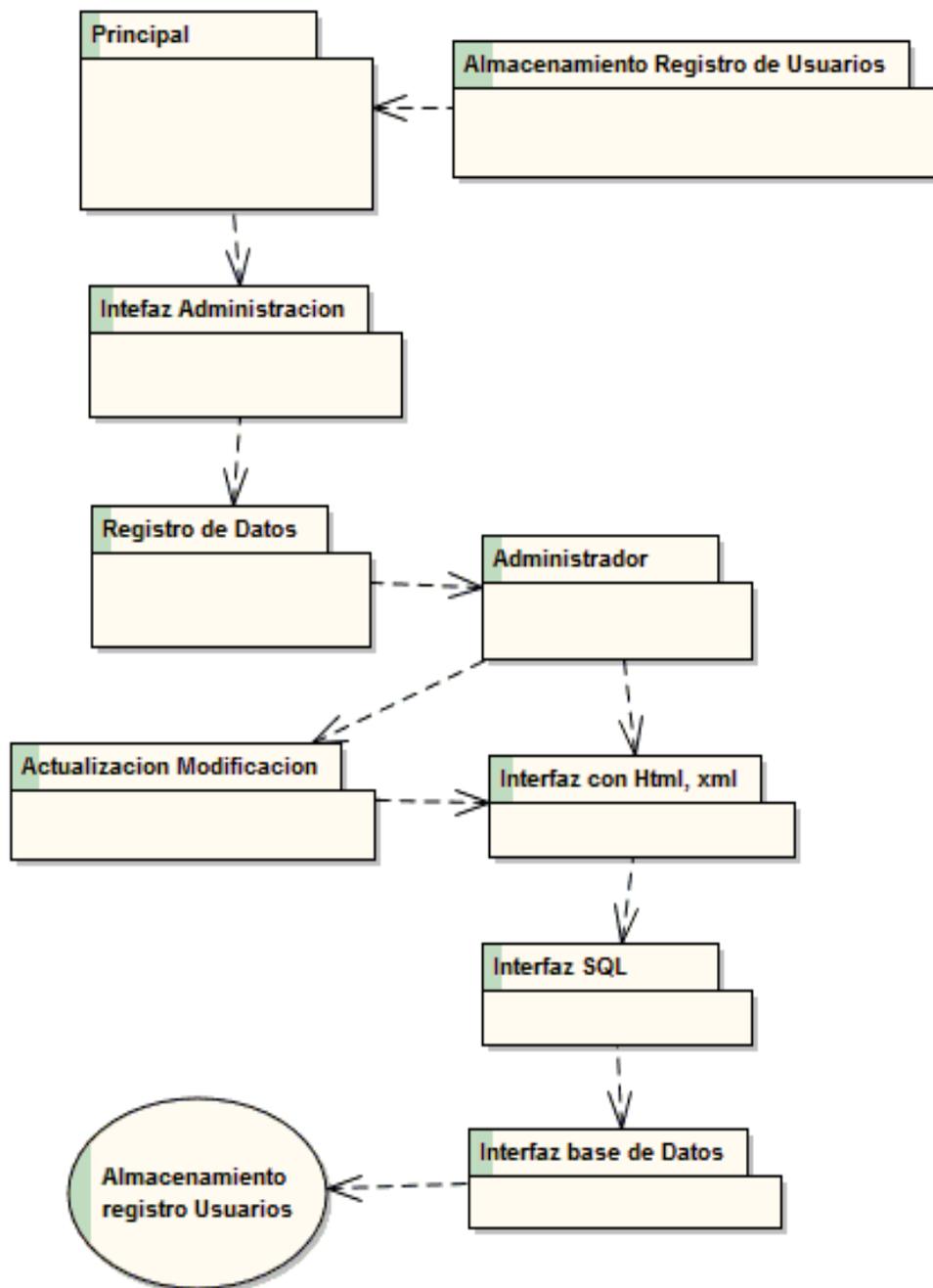
En esta fase el foco principal es el desarrollo de componentes y de otras características del sistema que está siendo diseñado. En proyectos que son bastante complejos, varias iteraciones de la construcción se pueden desarrollar en un esfuerzo de dividir dos casos en segmentos manejables que producen prototipos demostrables.

3.3.1 Diagrama De Paquetes

Permite Identificar el proceso de desarrollo del sistema anidado.

a) Registro De Datos

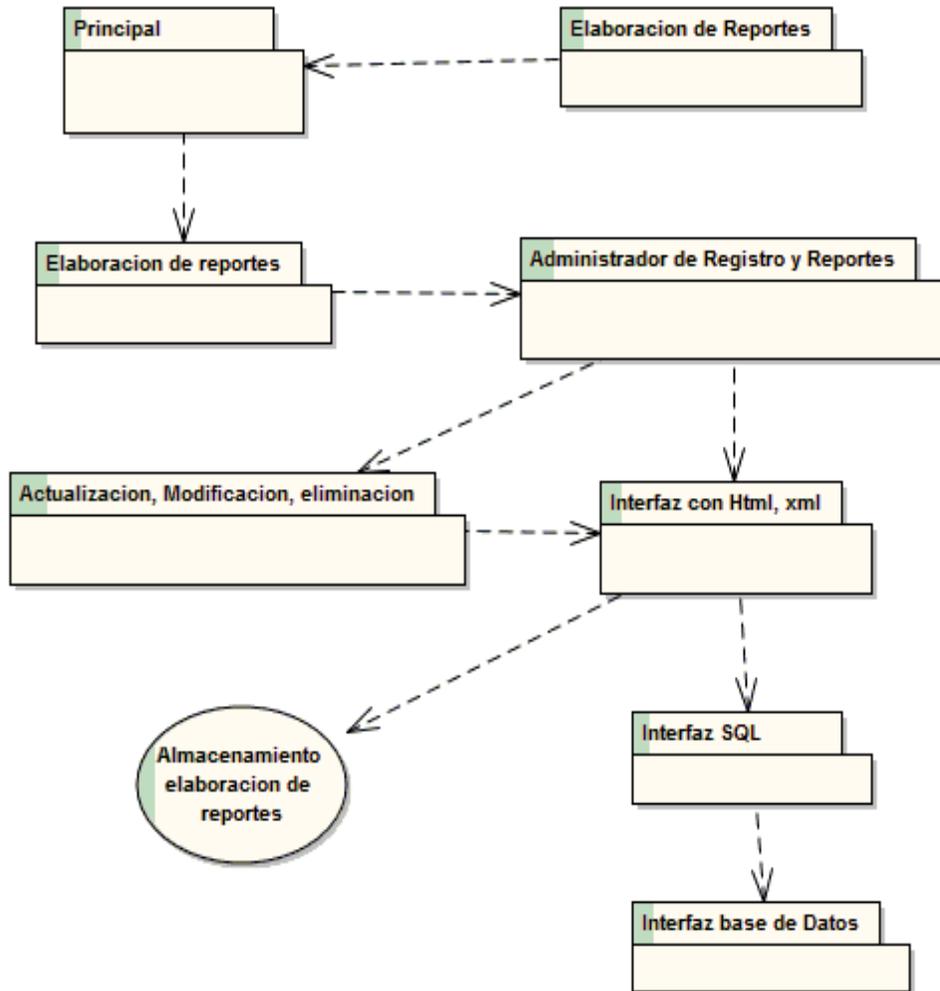
Figura 3.16 (Diagrama de paquetes: registro de Datos)



FUENTE (Elaboración Propia)

b) Elaboración De Reportes

Figura 3.17 (Diagrama de paquetes: Elaboración de Reportes)



FUENTE (Elaboración Propia)

c) Diseño De Interfaz

Formularios

En estas pantallas se observan los formularios para la inserción de datos, ya sea para insertar actualizar e incluso seleccionar datos en el caso de los reportes, ver anexo:

- Creación de Usuario Anexo Fig. **A.3**.
- Creación de Funcionario Anexo Fig. **A.5**.
- Creación de Planilla Fig. **A.7**.

- Modificación de Usuario Fig. **A.4.**
- Modificación de Funcionario **A.6.**
- Creación de haberes y Descuentos Fig. **A.10**
- Modificación de Haberes y Descuentos Fig. **A.13**

LISTADOS.

En estas vistas se observan los resultados de búsquedas y reportes.

- Listado de usuarios Fig. **A.3.**
- Listado de Funcionarios (Búsqueda) Fig. **A.5.**
- Listado de planillas Fig. **A.7.**
- Listado de Funcionarios en Planillas Seleccionada Fig. **A.9.**
- Listado de aportes de Funcionario Fig. **A.9.**

REPORTES.

- Certificado de Aportes Fig. **A.16.**
- Aportes Históricos de Funcionario Fig. **A.23.**

El usuario Ingresa su identificador y su contraseña. Estos datos serán proporcionados por el administrador del sistema.

3.4 FASE DE TRANSICIÓN

Esta es la última fase, el objetivo de esta fase es transportar el software desarrollando a los usuarios.

3.4.1 Mantenimiento De Software

En este caso el sistema informático desarrollado para plataforma web, se le aplicara el mantenimiento correctivo.

Correctivo

Durante la implementación del sistema pueden surgir problemas detectados por los usuarios finales en este caso el Transcriptor y Administrador, algunos de este supuesto podrían ser:

- El funcionamiento defectuoso de algunas herramientas del sistema, la causa más probable seria la no compatibilidad de los recursos usados para el desarrollo del

sistema con la actualización de los navegadores por ejemplo Firefox, Google Chrom, etc.

3.4.1.1 Control De Acceso Físico

Es uno de los controles principales para restringir el acceso físico a los dispositivos (Servidores y estaciones de trabajo) lo componentes a menudo encontrados son:

- Candados en los cuartos donde se Encuentra el Servidor.
- Controlar quien tiene permiso de uso directo con la base de datos.

3.5 SEGURIDAD

3.5.1 Control De Acceso

a) Requerimientos Para El Control De Acceso

Los controles de acceso deberán contemplar:

- Requerimientos de seguridad en el sistema web.
- Definir los perfiles o privilegios de acceso de los usuarios a las aplicaciones de acuerdo a su perfil de cargo en la entidad.

b) Administración De Accesos De Usuarios

La unidad de escalafón establece procedimientos para controlar la asignación de derechos de acceso a los sistemas, datos y servicios de información.

c) Creación De Usuarios

Los datos de acceso a los sistemas de información están compuestos por un ID, nombre de usuario y contraseña que es único por cada usuario.

Cuando se retire o cambie de contrato cualquier funcionario, se aplica la eliminación o cambios de privilegios en el sistema de información a los que el usuario estaba autorizado.

d) Administración De Contraseñas De Usuario

Todos los usuarios deberán cambiar su contraseña de acceso a los diferentes sistemas de información con una frecuencia mínima de 3 meses, a excepción de aquellos que contengan información confidencial o secreta en cuyo caso el cambio se debe realizar cada mes. (Tecnologías, 2005).

e) Uso De Contraseñas

Los usuarios deben cumplir las siguientes normas:

- Mantener los datos de acceso en secreto.
- Contraseñas fáciles de recordar y difíciles de adivinar.
- Que las contraseñas no estén basadas en algún dato que otra persona pueda adivinar u obtener fácilmente mediante información relacionada con la persona, por ejemplo nombres, números de teléfono, fecha de nacimiento, etc. (Políticas de Seguridad de Activos de Información)

3.5.2 Limitación Del Tiempo De Conexión (Inactividad)

Las restricciones al horario de conexión deben suministrar seguridad adicional a las aplicaciones de alto riesgo:

- a) Limitar los tiempos de conexión al horario normal de oficina, de no existir un requerimiento operativo de horas extras o extensión horaria.
- b) Llevar cuenta del tiempo de inactividad que un usuario lleva, para cerrar la sesión.

3.5.3 Protección Y Respaldo De La Información

Se deberá preservar la seguridad de la información dando cumplimiento a los principios de Confidencialidad, Integridad y Disponibilidad de la información de la institución.

La información de la institución deberá mantenerse disponible a las personas autorizadas para ello en el momento en que se necesite.

La institución deberá identificar mecanismos que permitan que las actividades de respaldo y recuperación de la información sean adecuadas costo / beneficio.

Los niveles de protección y clasificación establecidos para la información de la institución deberán ser mantenidos en todo momento. (Acceso, toma de respaldo, backup, transporte, recuperación, otros). Por lo tanto se deben mantener los controles y medidas establecidas para esto.

Los usuarios de la institución son responsables de alojar la información que necesita ser respaldada en los lugares establecidos para ello.

Los usuarios respaldarán y protegerán, con medidas que eviten accesos de personas no autorizadas, aquellos activos digitales de información que estén almacenados en elementos de TI de uso personal, que les hayan sido asignados. Se deberán preservar los lineamientos de acuerdo a la sensibilidad y nivel de clasificación de seguridad.

Los usuarios son responsables de aplicar los controles para la protección de la información según su nivel de clasificación. Así mismo deberán alertar al área del CRIE y la División de Sistemas cuando un activo digital de información requiera medidas especiales de protección.

Los funcionarios de la institución deberán seguir los procedimientos de respaldo de la información y realizar su seguimiento a partir de una bitácora de respaldos a la información personal. (Ecuador, 2011)

3.5.4 Niveles De Acceso

El sistema puede ser utilizado por el personal administrativo que se encuentra asignado a la división de Remuneraciones administrativas, de la misma manera al personal asignado a la división de escalafón, por niveles, el sistema ofrecerá facilidades de proceso de datos de acuerdo a las necesidades del personal, es flexible para habilitar a cualquier usuario, como también para modificar los niveles de acceso.

3.5.5 Copias De Seguridad De La Base De Datos

Una de las prioridades es contar con copias de seguridad, que consiste en guardar en un medio extraíble la información del sistema y de la base de datos, como resguardo de cualquier percance no previsto. Por lo que el administrador encargado del sistema deberá guardar los datos con fecha y hora copiada, la frecuencia con la que se crearan las copias de seguridad se determinara con un análisis previo considerando el volumen de datos que se introducen en la base de datos en un tiempo determinado.

MÉTRICAS DE CALIDAD

A la conclusión del desarrollo del sistema es menester conocer la calidad obtenida del sistema siendo este un factor importante por tanto esta calidad se basara en la descripción de las dimensiones de calidad.

Si bien no se llega a obtener calidad perfecta se desea lograr una calidad necesaria o suficiente para el momento cuando el usuario así lo requiera.

4.1 FUNCIONALIDAD

La funcionalidad mide el grado en el que el software satisface las necesidades indicadas por la adecuación, exactitud, interoperabilidad y seguridad de acceso.

Es por tanto que mediremos la eficacia en la Eliminación De Defectos (EED). Dentro del ámbito de desarrollo del sistema virtual, la EDD se define de la siguiente manera:

$$EED = E / (E - D)$$

Dónde:

E = Numero de errores antes de la entrega del software al usuario final.

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

Si el número de defectos presentados después de la entrega al usuario final es 0 entonces el valor de EDD es 1.

La correspondiente ponderación se representa en la siguiente tabla:

Tabla 4.1 Rangos de Nivel de Calidad para la Fiabilidad

Nivel de Calidad	Valor de EDD
Primer Nivel	Entre 1.15 y 1.10
Segundo Nivel	Entre 1.10 y 1.05
Tercer Nivel	Entre 1.05 y 1.00

Fuente (Elaboración Propia)

$$\text{Remplazando tenemos } EDD = \frac{E}{E-D} = \frac{21}{21-2} = \frac{21}{19} = 1.11$$

Si observamos los resultados de la EDD, tenemos un 1.11, comparando con la ponderación de la tabla # tenemos que en el rango del segundo nivel de funcionalidad.

4.2 CONFIABILIDAD

En la cantidad de tiempo que el software está disponible para su uso, es posible medir la confiabilidad tomando en cuenta la probabilidad del sistema que esté libre de fallos en un contexto determinado y durante un periodo de tiempo.

Probabilidad de hallar una falla:

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

Probabilidad de no hallar una falla:

$$P(T > t) = 1 - F(t)$$

Con

$$F(t) = Fc * \left(e^{-\frac{\lambda}{7} * 12} \right)$$

Dónde:

Fc: Funcionalidad del sistema.

λ : Taza de fallos en 7 ejecuciones dentro de un mes

Hallando la confiabilidad:

Tabla 4.2 Calculo de la Confiabilidad. (Elaboración Propia)

Hallando Confiabilidad	Prob. Hallar una falla (F(t))	Prob. De no hallar una falla
- $F(t) = Fc * (e^{-\lambda / 7 * 12})$	$P(T \leq t) = F(t)$	$1 - F(t)$
- $F(t) = 1.11 * (e^{-1 / 7 * 12})$	$F(t) = 0.08$	$1 - 0.08 = 0.92$
- $F(t) = 0.08$		

4.3 PORTABILIDAD

Para El presente Sistema se tiene Que el Hardware en el que funciona de manera estable del lado del servidor, está dado por un equipo Core i7 de 4Gb de RAM, el acceso a este servidor es a través de un intranet donde solo pueden acceder los usuarios autorizados del sistema. Las terminales de donde se accede al servidor tiene características de todo tipo desde Pentium IV hasta Core I7, Por otro lado el Sistema operativo del lado del servidor es Windows 7.

Al tratarse de un sistema informático desarrollado para una plataforma Web, puede ser utilizado desde cualquier Sistema Operativo (Windows, Linux, Mac Os), explorador (Firefox, Opera, Explorer, Chrom, etc.)

4.4 MANTENIBILIDAD

4.4.1 Mantenimiento Adaptivo

El mantenimiento adaptivo ocurrirá cuando se cambien las políticas o cuando se cambie la estructura de las planillas de haberes, estos factores harán que el sistema cambien en poca o gran medida, cambios para los cuales el sistema está preparado en adaptarse a algunos de estos casos, pero para otros complejos se deberá hacer revisión de los procesos y su adaptación con los nuevos cambios que se generen. Una de las principales razones por las que se tenga que revisar código es cuando los cambios en los exploradores de internet (Firefox, Opera, Explorer, Chrom, etc.) sufran cambios trascendentales en funcionamiento originando q algunos de los componentes del sistema funcionen defectuosamente.

4.4.2 Mantenimiento Perfectivo

El sistema está completamente abierto a añadir o adicionar nuevas funcionalidades de acuerdo a los nuevos requerimientos del cliente, siempre y cuando sean relacionados con el servicio e información que brinda el sistema.

4.5 USABILIDAD

La medición de la facilidad de uso se puede entender como la facilidad que el usuario tiene para acceder al sistema, tanto como para comprenderlo, aprenderlo y operarlo. A continuación presentamos en la tabla # los resultados obtenidos en la enseñanza de la manipulación del software a los 2 usuarios.

Tabla 4.3 Resultados para el cálculo de facilidad de uso.

Usuarios	Porcentaje Facilidad de Comprensión	Porcentaje Facilidad de aprendizaje	Porcentaje Facilidad de Operación
Usuario 1	91	90	91
Usuario 2	94	93	95
Promedio	92.5	91.5	93

Fuente (Elaboración Propia)

Por lo tanto de acuerdo a los resultados de la tabla # se obtuvo que la usabilidad del sistema es de un 92%.



EVALUACIÓN DE COSTOS Y BENEFICIOS

En este capítulo se evaluará los costos y beneficios para el sistema.

5.1 MÉTODO DE COCOMO

El modelo constructivo de costos es un modelo matemático de bases empíricas utilizado para estimación de costos de software.

Las ecuaciones que se utilizan en los tres modelos son:

- $E = a(KI)^b * m(X)$, en persona – mes
- $T_{dev} = c(E)^d * m(X)$, en meses
- $P = \frac{E}{T_{dev}}$, en personas

E = Esfuerzo requerido por el proyecto, en persona-mes.

T_{dev} = Tiempo requerido por el proyecto, en meses.

P = Número de personas requerido por el proyecto.

a, b, c y d = Constantes con valores definidos en una tabla, según cada submodelo.

KI = Cantidad de líneas de código, en miles.

m(X) = Es un multiplicador que depende de 15 atributos.

Se tiene un total de líneas de código Klcd = 8363

Así

$$KI = \frac{Klcd}{1000}$$

$$KI = \frac{8363}{1000} = 8.363 \text{ KI}$$

En este caso es el tipo orgánico será más apropiado ya que el número de líneas de código no supera los 50KI

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

Tabla 5.1 Ponderación del esfuerzo

MODO	a	b	c	d
Orgánico	3.2	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Fuente (Elaboración propia)

Y también debemos de hallar la variable FAE es el factor de ajustes de esfuerzo la cual se obtiene mediante la multiplicación de los valores evaluados.

Tabla 5.2 Factores y Valores

		Valor					
		Muy Bajo	Bajo	Nominal	alto	Muy Alto	Extra Alto
a)	Fiabilidad	0.75	0.88	1.00	1.15	1.40	-
b)	Tamaño de base de datos	-	0.94	1.00	1.08	1.16	-
c)	Complejidad	0.70	0.85	1.00	1.15	1.30	1.65
d)	Restricciones de tiempo de ejecución	-	-	1.00	1.11	1.30	1.66
e)	Tiempo de respuesta	-	0.87	1.00	1.07	1.30	-
f)	Capacidad de análisis.	1.46	1.19	1.00	0.86	0.71	-
g)	Experiencia en la aplicación	1.29	1.29	1.00	0.91	0.82	-
h)	Calidad de los programadores	1.42	1.17	1.00	0.86	0.70	-
i)	Experiencia en el lenguaje	1.14	1.07	1.00	0.95	-	-
j)	Técnicas actualizadas de programación	1.24	1.10	1.00	0.91	0.82	-
k)	Utilización de herramientas software	1.24	1.10	1.00	0.91	0.83	-
l)	Restricciones de tiempo de desarrollo	1.23	1.23	1.00	1.04	1.10	-

Fuente (Elaboración Propia)

$$m(X) = 1.15 * 1.00 * 1.00 * 1.00 * 1.00 * 0.86 * 0.91 * 0.86 * 0.95 * 1.00 * 0.91 * 1.00$$

El Resultado a ser procesado es:

$$m(X) = 0.67$$

La justificación de valores es:

- a) **Fiabilidad.** Si se produce un fallo por alguna mensualidad o registro, puede ocasionar pérdida de información (valoración alta).
- b) **Tamaño de base de datos.** La base de datos es de tipo estándar.
- c) **Complejidad.** La aplicación no realiza cálculos complejos.
- d) **Restricciones de tiempo de ejecución.** En los requerimientos se exige un alto rendimiento
- e) **Tiempo de respuesta.** Deberá ser interactivo con el usuario.
- f) **Capacidad de análisis.** Capacidad media, debido a la experiencia en análisis en proyecto similar
- g) **Experiencia en la aplicación.** Se tiene cierta experiencia en aplicaciones de esta envergadura.
- h) **Calidad de los programadores.** Teóricamente deberá tenerse una capacidad muy alta por la capacidad de un programador
- i) **Experiencia en el lenguaje.** Es alta, dado que se controlan las nociones básicas y propias del proyecto.
- j) **Técnicas actualizadas de programación.** Se usaran prácticas de programación mayormente convencional.
- k) **Utilización de herramientas software.** Se usaran herramientas estándar que no exigirán formación.
- l) **Restricciones de tiempo de desarrollo.** Existen pocos límites de planificación.

Calculo del esfuerzo del desarrollo.

$$E = a(KI)^b * m(X), \text{ en persona mes.}$$

$$E = 3.2 * (8.363)^{1.05} * 0.549$$

$$E = 16.33$$

Calculo tiempo de desarrollo.

$$T_{dev} = c(E)^d, \text{ en meses.}$$

$$T = 2.5 \cdot 12.25 \cdot 0.38$$

$$T = 7.22 \text{ meses}$$

Productividad

$$P = \frac{16.33}{7.22} = 2.261$$

Estimación de costos.

Al tener los datos definidos, se realiza el supuesto estimado del costo.

La hora por trabajar = 7.8Bs.

Día = 7.8 Bs * 8 Horas = 62.4Bs/Día.

Semana = 62.4 Bs * 5 días = 312 bs / Semana

Mes = 312bs/Semana * 4 Semanas = 1248.

Total 1248Bs / Mes * 4 Meses = 6420Bs

CONCLUSIONES Y RECOMENDACIONES.

6.1 CONCLUSIONES

Luego de haber realizado el análisis y diseño del sistema informático de administración de planillas de haberes caso Universidad Mayor de San Andrés. Se logró obtener las siguientes Conclusiones:

- Se ha elaborado interfaces de usuario para un fácil acceso a la información, utilizando las herramientas de JSF, PrimeFaces, JSF, y Java.
- La información ahora se maneja de manera automatizada lo que reduce el manejo de información de forma manual, así mismo la implementación de una base de datos para el almacenamiento produce y genera que la información sea rápida y oportuna evitando así problemas y errores en el manejo mismo.
- Acortar el tiempo de búsqueda y acceso a la información almacenada en las planillas de haberes
- Permite realizar una mejor atención a la Unidades Solicitantes, con un tiempo mucho menor al tiempo empleado con el proceso anterior con información precisa y confiable.
- Interfaz entendible para el usuario con pantallas comprensibles y de fácil manejo
- Se logró automatizar la emisión de certificados de aportes de manera oportuna, precisa y confiable.

Se dio cumplimiento a los requerimientos planteados, a criterio de la institución el sistema cubre con los requerimientos establecidos hacia los usuarios. Por ende el sistema ya se implementó y está en funcionamiento en el servidor perteneciente al departamento de recursos humanos de la Universidad Mayor De San Andrés.

6.2 RECOMENDACIONES

Con la finalización del presente proyecto se pueden efectuar las siguientes recomendaciones:

- Realizar el mantenimiento al Sistema Web, para el correcto desempeño del sistema y prevenir posibles fallas.
- Realizar módulos que permitan el compartir la información de la organización con otros sistemas.
- Implementar un módulo de lectura de archivos para de esta manera lograr reducir el tiempo de inserción de datos en la base de datos, ya que de las planillas más recientes se cuenta almacenada en forma digital.
- Desarrollar procesos que permitan obtener backup de forma automática.



BIBLIOGRAFÍA

- Accenture, ©. 2. (2014). <http://www.accenture.com/>. Obtenido de <http://www.accenture.com/>.
- DBpedia . (2005). ISO/IEC 9126 ISO/IEC 9126.
- Dni, R. c. (2014). www.adictosaltrabajo.com. Obtenido de www.adictosaltrabajo.com.
- Ecuador, U. C. (2011). Seguridad de la Información. Quito - Ecuador.
- Fowler , M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (Object Technology Series).
- Grady Booch, J. R. (1999). El lenguaje de unificado de modelado. Edit:Addison Wesley.
- J. Pedro Caraca, V. y.-U. (1997.). Curso de Ingeniería del Software - Unidad .
- James Rumbaugh, M. B. (1996). Modelos y diseños orientados a objetos. Prentice Hall.
- Kendal & Kendal. (2002). Analisis y Diseño de sistemas 913 Tercera Edición Editorial. Amy Kohen, Mexico.
- López y Mtra, M. (s.f.). Awdtools. Obtenido de Awdtools: <http://www-306.ibm.com/software/awdtools/rup/>
- Pereira, U. T. (s.f.). Políticas de Seguridad de Activos de Informacion.
- Perez Darling, D. (2008). Análisis y Diseño de Componentes para Pruebas caja blanca y negra. de La Habana.
- Prentice, H. (1981). Software Engineering Economics.
- Pressman, R. (1998). Ingeniería del Software. Un enfoque práctico. McGraw-Hill.
- Pressman., R. S. (1993). IEEE Task Force on Requirements Engineering. Software Engineering.
- S.A., ©. M. (15 de Abril de 2015). Monografias.com. Obtenido de www.monografias.com
- S.L.:, M. (2014). Guiarte Multimedia S.L. . Obtenido de Guiarte Multimedia S.L. .
- Senn, J. (1992). Análisis y Diseño de Sistemas de Información. McGraw Hill.
- Somerville, I. (1996). Ingeniería del Software. Mexico: Addison Wesley.

Somerville, I. (1996). Ingeniería del Software. Mexico: Addison Wesley.

SQuare. (2005). Software Engineering Iso/lec.

Tecnologias, O. N. (2005). Modelo de Política de Seguridad de la Información para Organismos del Sector. Documento Publico.

UMSA. (14 de abril de 2015). Departamento de Recursos Humanos. Obtenido de Departamento de Recursos Humanos: <http://rrhh.umsa.bo/>

Umsa.Bo. (14 de 04 de 2014). Universidad Mayor de San Andres. Obtenido de <http://www.umsa.bo/>: <http://www.umsa.bo/>

Wikipedia. (2014). Metodología de Desarrollo. En Wikipedia, Metodologías de desarrollo de Software.



The logo of Universidad Mayor Pacensis Divi Andreae is an oval emblem. It features a sun with rays at the top, a mountain range in the middle, and a green banner at the bottom with a blue cross and a central figure. The text "UNIVERSITAS MAJOR PACENSIS DIVI ANDRAE" is written around the perimeter of the oval.

ANEXOS

ANEXOS A. INTERFAZ GRÁFICA

FIGURA A.1 (El usuario ingresa su identificador y contraseña)

The screenshot shows a web interface for 'Recursos Humanos Administrativos - S. I. P. L. H. A.'. At the top, there is a dark header with the text 'Recursos Humanos Administrativos - S. I. P. L. H. A.' and 'Bienvenido' below it. The main content area is light gray and contains a central login form. The form has a title 'RR, HH. - S. I. P. L. H. A.' and a subtitle 'Complete el formulario para ingresar al sistema'. It includes two input fields: 'Usuario: *' and 'Contraseña: *', both with asterisks indicating they are required. Below the fields is a blue button labeled 'Ingresar'. At the bottom of the page, there is a dark footer with the text '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

FUENTE (Elaboración Propia)

FIGURA A.2 (Pantalla de Bienvenida)

The screenshot displays the main dashboard of the 'Sistema Informático Administración de Planillas'. The top header is dark and contains the system name 'Sistema Informático Administración de Planillas' and 'Recursos Humanos Administrativos' on the left. On the right, it shows 'Session Iniciada Por: admin ADMINISTRADOR' and a 'Salir' button. A left sidebar menu is visible with categories: 'GESTIÓN USUARIOS' (with sub-item '+Usuarios'), 'GESTIÓN FUNCIONARIOS' (with sub-item '+Funcionarios'), 'GESTIÓN PLANILLAS' (with sub-items '+Edicion de Planillas' and '+Detalles Planillas'), and 'GESTIÓN REPORTES' (with sub-items '+Certificado de Aportes' and '+Reportes Hist. Funcionario'). The main content area has a dark header with 'Welcome to Siplha' and a subtitle 'Sistema Informatico Administrador de Planillas de Haberes'. The footer at the bottom contains the text '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

FUENTE (Elaboración Propia)

FIGURA A.3 (Gestión de usuarios)

The screenshot shows the 'Gestionar Usuarios' page. At the top, there is a header with the logo 'HR' and the text 'Sistema Informático Administración de Planillas' and 'Recursos Humanos Administrativos'. The session information indicates 'Session Iniciada Por: admin ADMINISTRADOR' with a 'Salir' button. A left sidebar contains a 'MENÚ' with options like 'GESTION USUARIOS', 'GESTIÓN FUNCIONARIOS', 'GESTIÓN PLANILLAS', and 'GESTIÓN REPORTES'. The main area features a search form with fields for 'Usuario' and 'C.I.', and buttons for 'Buscar #', 'Crear Usuario +', and 'Cancelar'. Below the form is a table with 11 columns: Nro, CI, Nombre Completo, Usuario, Password, Tipo, Estado, Ver, Edit, and Elim. The table contains two rows of user data.

Nro	CI	Nombre Completo	Usuario	Password	Tipo	Estado	Ver	Edit	Elim
1	123456	admin	admin	admin	ADMINISTRADOR	ACTIVO			Elimin *
2	ivan	ivan	ivan	ivan	ADMINISTRADOR	ACTIVO			Elimin *

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.4 (Creación de Usuario, Se ingresa los datos Correspondientes)

This screenshot shows the same 'Gestionar Usuarios' page as Figure A.3, but with a modal window titled 'Creando Usuario' open in the center. The modal contains several input fields: 'Cedula de Identidad' (with value 'NuevoUsuario'), 'Nombre Completo' (with value 'NuevoUsuario'), 'Usuario' (with value 'NuevoUsuario'), 'Contraseña' (with value 'NuevoUsuario'), 'Tipo de Usuario' (with a dropdown menu showing 'Administrador'), and 'Estado' (with a dropdown menu showing 'Activo'). At the bottom of the modal are buttons for 'Guardar Usuario' and 'Cerrar'. The background table and search form are partially visible behind the modal.

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.5 (Gestión de Funcionarios)

The screenshot shows the 'Gestión de Funcionarios' interface. At the top, there is a header with the logo 'hr', the title 'Sistema Informático Administración de Planillas', and the subtitle 'Recursos Humanos Administrativos'. The session information indicates 'Session Iniciada Por: admin ADMINISTRADOR' with a 'Salir' button. A left sidebar menu contains options like 'GESTION USUARIOS', 'GESTIÓN FUNCIONARIOS', 'GESTIÓN PLANILLAS', and 'GESTIÓN REPORTES'. The main area features a search form with fields for 'Nombres', 'Ap. Paterno', 'Ap. Materno', and 'Carnet Identidad', along with 'Buscar' and 'Crear Funcionario' buttons. Below the form is a table with 5 rows of employee data. The table has columns for 'Nro', 'Nombres', 'Paterno', 'Materno', 'Unidad', 'Cargo', 'Nivel', 'Item', 'Ver', 'Edit', and 'Elim'. The footer shows '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

Nro	Nombres	Paterno	Materno	Unidad	Cargo	Nivel	Item	Ver	Edit	Elim
1	MARIA ISABEL	ABAN	BOTELLO	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	SECRETARIA	5	1			Elim
2	MARTIN	ABELO	MAMANI	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	SECRETARIA	5	2			Elim
3	JOSE LUIS	ABREGO	SALAZAR	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	SECRETARIA	1	3			Elim
4	JUAN JOSE	ACARAPI	GUZMAN	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	4			Elim
5	JORGE	ACARAPI	LECONA	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	5			Elim

FUENTE (Elaboración Propia)

FIGURA A.6 (Creación de Nuevo Funcionario, llenado de datos)

The screenshot shows the 'Gestión de Funcionarios' interface with a 'Creando Funcionario' modal form open. The modal is divided into 'Datos Personales' and 'Datos Funcionario'. The 'Datos Personales' section includes fields for 'Primer Nombre', 'Segundo Nombre', 'Tercer Nombre', 'Ap. Paterno', 'Ap. Materno', 'Ap. Casada', 'Fec. Nac.', 'C.I.', 'Expedido', and 'Sexo'. The 'Datos Funcionario' section includes 'Estado' (radio buttons for 'ACTIVO' and 'INACTIVO'), 'Unidad', 'Cargo', 'Item', and 'Nivel'. There are 'Guardar' and 'Cerrar' buttons at the bottom of the modal. The background shows the same table as in Figure A.5. The footer shows '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

FUENTE (Elaboración Propia)

FIGURA A.7 (Gestión de Planillas, El usuario Selecciona una planilla)

Sistema Informático Administración de Planillas
Recursos Humanos Administrativos

Session Iniciada Por: admin
ADMINISTRADOR [Salir](#)

MENÚ

- GESTION USUARIOS
 - Usuarios
- GESTIÓN FUNCIONARIOS
 - Funcionarios
- GESTIÓN PLANILLAS**
 - Edición de Planillas
 - Detalles Planillas
- GESTIÓN REPORTES
 - Certificado de Aportes
 - Reportes Hist. Funcionario

GESTIÓN DE PLANILLAS [Crear Planilla](#)

(1 of 1) Total: 2 Planillas

Nro	MES	AÑO	Opciones	Ver Totales
1	Diciembre	1995	Seleccionar Planilla Eliminar	Totales
2	Febrero	2015	Seleccionar Planilla Eliminar	Totales

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.8 (Visualización de Sumas Totales de las Planillas).

Sistema Informático Administración de Planillas
Recursos Humanos Administrativos

Session Iniciada Por: admin
ADMINISTRADOR [Salir](#)

MENÚ

- GESTION USUARIOS
 - Usuarios
- GESTIÓN FUNCIONARIOS
 - Funcionarios
- GESTIÓN PLANILLAS**
 - Edición de Planillas
 - Detalles Planillas
- GESTIÓN REPORTES
 - Certificado de Aportes
 - Reportes Hist. Funcionario

Calculo de Totales

Diciembre 1995

HABERES

Total Basico : 1500.0
Total Categoria: 5517.7
Total Ganado 2099.1

DESCUENTOS

Total SSU 1148.34

[Atras](#)

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.9 (Administración de planilla seleccionada, adiciona funcionarios a la planilla, insertando datos de la planilla de haberes)

Sistema Informático Administración de Planillas
Recursos Humanos Administrativos

Session Iniciada Por: admin
ADMINISTRADOR [Salir x](#)

MENÚ

- GESTIÓN USUARIOS
 - Usuarios
- GESTIÓN FUNCIONARIOS
 - Funcionarios
- GESTIÓN PLANILLAS
 - Edición de Planillas
 - Detalles Planillas
- GESTIÓN REPORTES
 - Certificado de Aportes
 - Reportes Hist. Funcionario

FUNCIONARIOS Diciembre 1995 Listado de Funcionarios

FUNCIONARIOS Diciembre 1995

Seleccionar Otra Planilla Ver Totales

(1 of 1) 1 10 Total: 8 funcionarios en la Planilla

Nro	Nombres	Paterno	Materno	Unidad	Cargo	Nivel	Item	Opciones
1	MARIA ISABEL	ABAN	BOTELLO	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	1	Ver Edit Eliminar
2	SIMON	AMARU	HUAYGUA	RECTORADO	OFICINISTA	8	5	Ver Edit Eliminar
3	ZAIDA BEATRIZ	MARTINEZ	PLAZA	RECTORADO	SECRETARIA EJECUTIVA	12	3	Ver Edit Eliminar
4	SONIA LUCRECIA	CENTELLAS	REGUERIN	RECTORADO	SECRETARIA ADMINISTRATIVA	15	2	Ver Edit Eliminar
5	ORLANDO NATALIO	ALBARRACIN	RODRIGUEZ	DEPARTAMENTO PLANIFICACIÓN Y COORDINACION	ANALISTA	20	10	Ver Edit Eliminar

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.10 (Adición de un Nuevo Registro en la planilla Seleccionada)

Sistema Informático Administración de Planillas
Recursos Humanos Administrativos

Session Iniciada Por: admin
ADMINISTRADOR [Salir x](#)

MENÚ

- GESTIÓN USUARIOS
 - Usuarios
- GESTIÓN FUNCIONARIOS
 - Funcionarios
- GESTIÓN PLANILLAS
 - Edición de Planillas
 - Detalles Planillas
- GESTIÓN REPORTES
 - Certificado de Aportes
 - Reportes Hist. Funcionario

Información Planilla Diciembre 1995

Datos Personales

Primer Nombre * WALDO Segundo Nombre DT 0.0 Básico 0.0
 Tercer Nombre Ap. Paterno * ALBARRACIN Categ 0.0 Tot.Mes 0.0
 Ap. Materno SANCHEZ Ap. Casada Tot.Des 0.0 Liquido 0.0
 C.I. * 2312846 46

Datos Funcionario

Unidad * DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS
 Cargo ABOGADO
 Item * 46
 Nivel * 1

Haberes

DT 0.0 Básico 0.0
 Categ 0.0 Tot.Mes 0.0
 Tot.Des 0.0 Liquido 0.0

Descuentos

Tdes 0.0 Renta 0.0
 Verios 0.0 Fonvi 0.0
 Cbiofer 0.0 Seu 0.0
 Stumse 0.0 Casid 0.0
 Codavi 0.0 Sbas 0.0
 Smsu 0.0 Clulum 0.0
 Suspen 0.0 Agrom 0.0
 Multa 0.0 Rjud 0.0
 Cinfra 0.0 Cob 0.0
 Comoru 0.0 Anticp 0.0
 Ueduc 0.0 Convum 0.0
 Consumo 0.0 Acred 0.0
 Pulpen 0.0

Insertar a Planilla Cerrar

Nro	Nombres	Paterno	Materno	Cargo	Item	Unidad	Nivel	Opciones
1	ELIZABETH LOURDES							Ver Func
2	MARIA ELENA							Ver Func
3	ORLANDO NATALIO							Ver Func
4	WALDO							Ver Func

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.11 (Visualización de Totales Actualizados)

The screenshot shows the 'Sistema Informático Administración de Planillas' interface. A modal window titled 'Información Planilla Diciembre 1995' is open, displaying financial data for a specific planilla. The data is organized into sections: 'HABERES' (Total Básico: 1500.0 Bs., Total Categoría: 5517.7 Bs., Total Ganado: 2099.1 Bs.) and 'DESCUENTOS' (Total SSU: 1148.34 Bs.). The modal also includes a 'Cerrar' button. In the background, a table lists employees with columns for 'Nro', 'Nombres', 'Paterno', 'Materno', 'Cargo', 'Nivel', 'Item', and 'Opciones'. The footer of the application reads '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

FUENTE (Elaboración Propia)

FIGURA A.12 (Selecciona la Opción Detalles de Planillas y selecciona una planilla).

The screenshot shows the 'GESTION DE PLANILLAS' screen in the 'Sistema Informático Administración de Planillas'. A table displays two planillas for selection, with columns for 'Nro', 'MES', and 'AÑO'. The first row is for 'Diciembre 1995' and the second for 'Febrero 2015'. Each row has a 'Seleccionar Planilla' button. The interface also includes a 'MENÚ' on the left and a footer with '© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0'.

FUENTE (Elaboración Propia)

FIGURA A.13 (Una vez seleccionado el tema, se observa un apartado de registros almacenados en la planilla seleccionada con la opción de ver detalles)

The screenshot shows the 'Sistema Informático Administración de Planillas' interface. The main window displays a table of 'FUNCIONARIOS' with columns for Nro, Nombres, Ap. Paterno, Ap. Materno, C.I., Unidad, Cargo, Nivel, and Item. A modal window titled 'Información Planilla Diciembre 1995' is open, showing detailed information for a selected employee, including 'Datos Personales', 'Datos Funcionario', 'Haberes', and 'Descuentos'.

Nro	Nombres	Ap. Paterno	Ap. Materno	C.I.	Unidad	Cargo	Nivel	Item
1	MARIA ISABEL	ABAN	BOTELLO	4772940	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	44
2	SIMON	AM					8	
3	ZAIDA BEATRIZ	MA					12	
4	SONIA LUCRECIA	CE					15	
5	ORLANDO NATALIO	ALE					20	
6	MARUJA	SERRUDO	ORMACHEA		DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	JEFE DIVISION	8	1500.0
7	RUTH CATALINA	REYES	OCAMPO		DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	ANALISTA	13	10

FUENTE (Elaboración Propia)

FIGURA A.14 (Nuevamente se observa la opción de Búsqueda, en este vista se selecciona un funcionario).

The screenshot shows the 'Sistema Informático Administración de Planillas' interface. The main window displays a 'CERTIFICADO DE APORTES' form with search fields for 'Nombres', 'Ap. Paterno', 'Ap. Materno', and 'Carnet Identidad'. Below the search fields is a 'Buscar' button. The results are displayed in a table with columns for Nro, Nombres, Paterno, Materno, Unidad, Cargo, Nivel, Item, and Opciones.

Nro	Nombres	Paterno	Materno	Unidad	Cargo	Nivel	Item	Opciones
1	MARIA ELENA	ALBARRACIN	IRUSTA	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	44	Seleccionar
2	ORLANDO NATALIO	ALBARRACIN	RODRIGUEZ	DEPARTAMENTO PLANIFICACIÓN Y COORDINACIÓN	ANALISTA	20	10	Seleccionar
3	WALDO	ALBARRACIN	SANCHEZ	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	46	Seleccionar

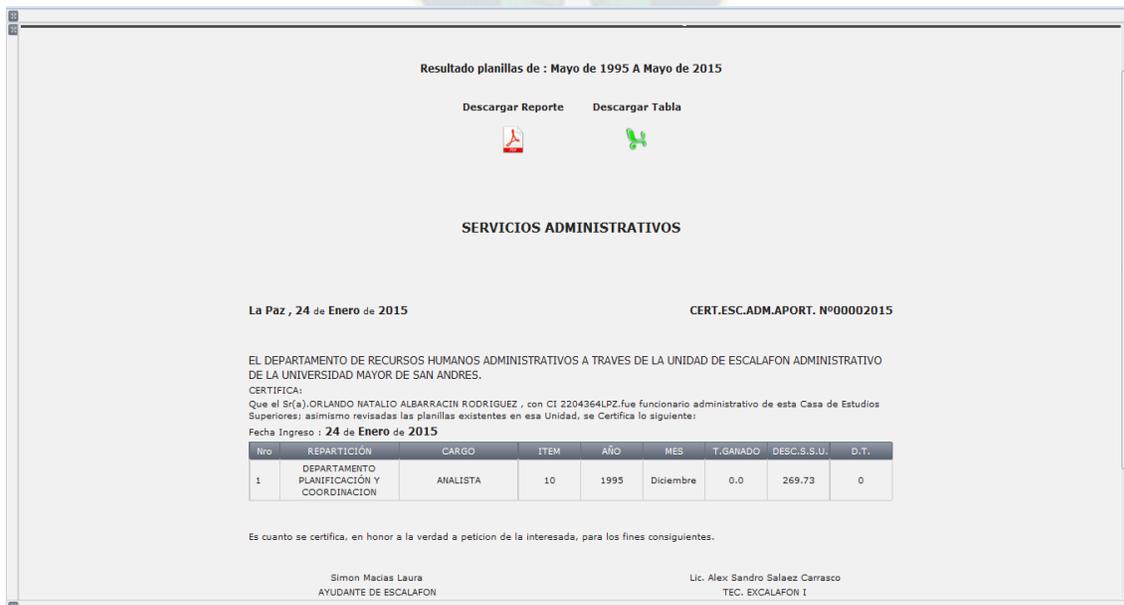
FUENTE (Elaboración Propia)

FIGURA A.15 (Aquí se selecciona el rango de fechas en los que se desea se genere el reporte de aportes)



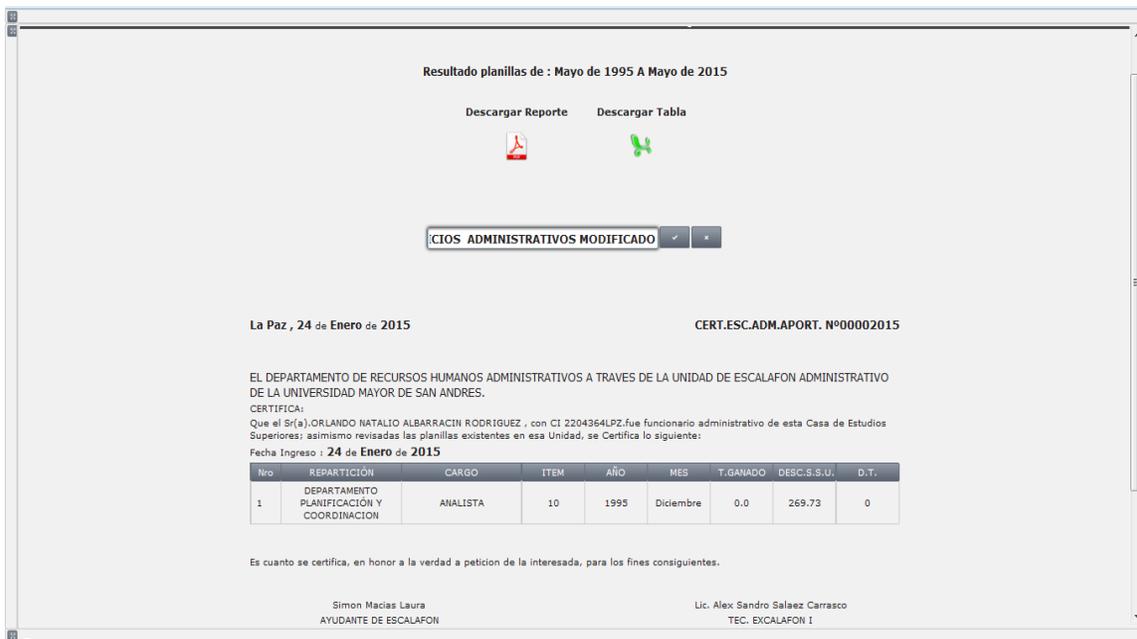
FUENTE (Elaboración Propia)

FIGURA A.16 (Aquí se observa una vista previa del certificado de aportes)



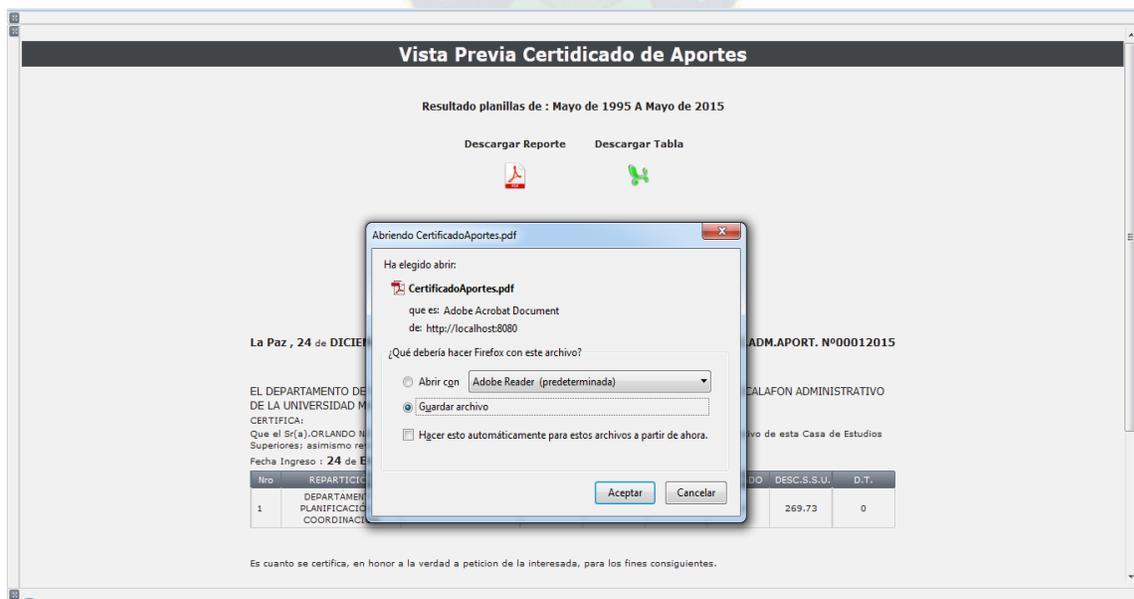
FUENTE (Elaboración Propia)

FIGURA A.17 (Todo los datos No calculados son totalmente editables, en este caso se muestra cómo se puede modificar el título del certificado de aportes)



FUENTE (Elaboración Propia)

FIGURA A.18 (Una vez modificado el certificado se elige la opción de exportar PDF, click en el icono PDF)



FUENTE (Elaboración Propia)

FIGURA A.19 (Aquí el detalle del documento exportado PDF).

SERVICIOS ADMINISTRATIVOS

La Paz, 19 de Enero de 2015 CERT.ESC.ADM.APORT. N° 0000/2015

EL DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS A TRAVES DE LA UNIDAD DE ESCALAFON ADMINISTRATIVO DE LA UNIVERSIDAD MAYOR DE SAN ANDRES.

CERTIFICA:

Que el Sr

SONIA LUCRECIA CENTELLAS REGUERIN , con CI.23109 LPZ

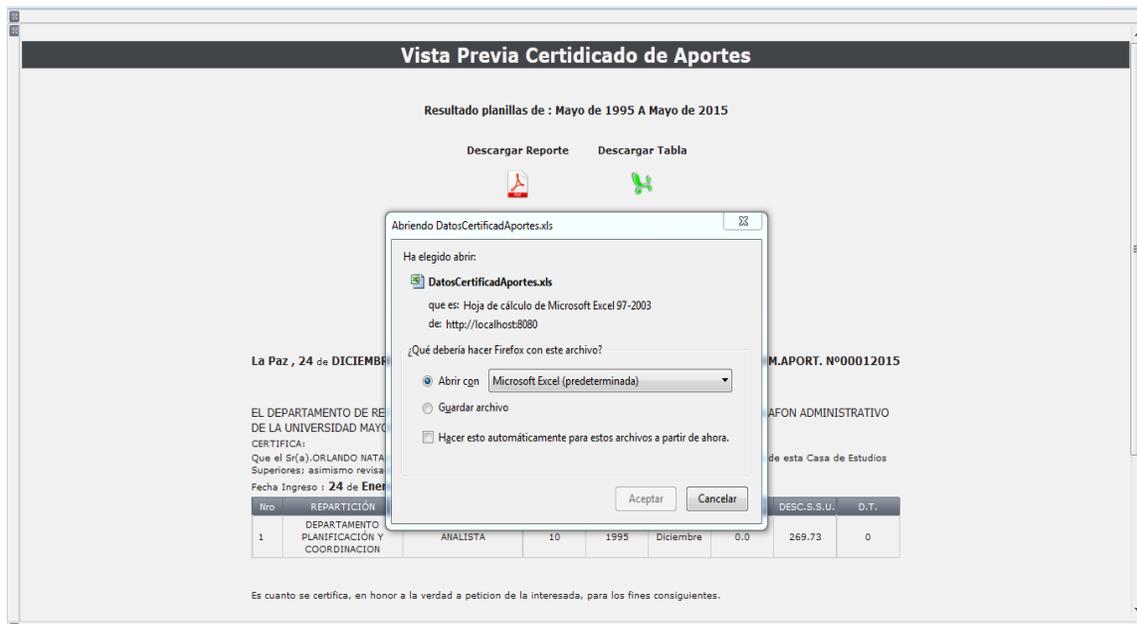
fue funcionario administrativo de esta Casa de Estudios Superiores; asimismo revisadas las planillas existentes en esa Unidad, se Certifica lo siguiente:

Fecha Ingreso: 19 de Enero de 2015

Repartición	Cargo	Item	Año	Mes	T:Ganado	Desc.S.S.U	D.T.
RECTORADO	SECRETARI ADMINISTRATIVA	2	1995	10	2221.32	177.71	30
RECTORADO	SECRETARI ADMINISTRATIVA	2	1995	11	2221.32	177.71	30
RECTORADO	SECRETARI ADMINISTRATIVA	2	1995	12	2221.32	177.71	30

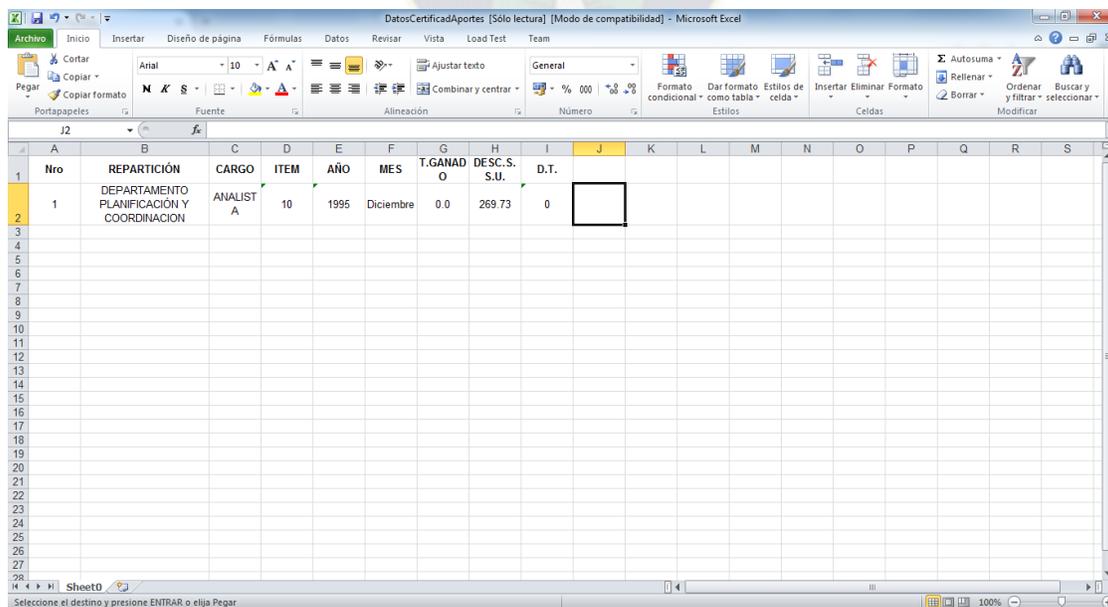
FUENTE (Elaboración Propia)

FIGURA A.20 (Dando Click en el icono Excel, accedemos a la opción de descargar la información mostrada en la tabla de contenidos para ser utilizado).



FUENTE (Elaboración Propia)

FIGURA A.21 (Vista del documento Excel Exportado, con la información de la tabla de contenidos mostrada en la vista previa al reporte)



FUENTE (Elaboración Propia)

FIGURA A.22 (Opción Reportes Histórico de Funcionarios, y búsqueda de funcionario)

The screenshot shows the 'Sistema Informático Administración de Planillas' interface. The header includes the logo, title, and user session information: 'Session Iniciada Por: admin ADMINISTRADOR' with a 'Salir' button. A left sidebar contains a 'MENÚ' with options like 'GESTION USUARIOS', 'GESTIÓN FUNCIONARIOS', 'GESTIÓN PLANILLAS', and 'GESTIÓN REPORTES'. The main content area is titled 'CERTIFICADO DE APORTES' and features a search form with fields for 'Nombres', 'Ap. Paterno' (filled with 'albarraci'), 'Ap. Materno', and 'Carnet Identidad', along with a 'Buscar' button. Below the search form is a table with 3 rows of employee data and a 'Total: 3 funcionarios' indicator.

Nro	Nombres	Paterno	Materno	Unidad	Cargo	Nivel	Item	Opciones
1	MARIA ELENA	ALBARRACIN	IRUSTA	DEPARTAMENTO DE RECURSOS HUMANOS ADMINISTRATIVOS	PORTERO	1	44	Seleccionar
2	ORLANDO NATALIO	ALBARRACIN	RODRIGUEZ	DEPARTAMENTO PLANIFICACIÓN Y COORDINACION	ANALISTA	20	10	Seleccionar
3	WALDO	ALBARRACIN	SANCHEZ	ASESORIA JURIDICA	ABOGADO	20	20	Seleccionar

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA A.23 (Una vez seleccionado el funcionario, se observa todos los detalles almacenadas en todas las planillas del funcionario seleccionado)

The screenshot shows the 'REPORTE HISTORICO DE FUNCIONARIOS VISTA' screen. It displays the details for the selected employee: 'Funcionario Seleccionado ORLANDO NATALIO ALBARRACIN RODRIGUEZ'. Below this, it lists the employee's details: 'Unidad : DEPARTAMENTO PLANIFICACIÓN Y COORDINACION', 'Cargo : ANALISTA', 'Item : 10', and 'Nivel : 20'. There is a 'Cambiar de Funcionario' button. At the bottom, there is a table showing the employee's record history across different months and years.

Nro	Unidad	Cargo	Nivel	Item	Mes	Año	DíasT
1	DEPARTAMENTO PLANIFICACIÓN Y COORDINACION	CONTADOR I	20	10	Octubre	1995	30
2	DEPARTAMENTO PLANIFICACIÓN Y COORDINACION	CONTADOR I	20	10	Noviembre	1995	30
3	DEPARTAMENTO PLANIFICACIÓN Y COORDINACION	CONTADOR I	20	10	Diciembre	1995	0

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)

FIGURA # (También se puede exportar la tabla de contenidos en formato Excel)

Session Iniciada Por: admin ADMINISTRADOR Salir

DESCARGAR REPORTE EN FORMATO EXCEL

Funcionario Seleccionado ORLANDO NATALIO ALBARRACIN RODRIGUEZ
Unidad : DEPARTAMENTO PLANIFICACIÓN Y COORDINACIÓN
Cargo : ANALISTA
Item : 10
Nivel : 20

Cambiar de Funcionario

Nro	Unidad	Cargo	Nivel	Item	Mes	Año	DíasT
1	DEPARTAMENTO PLANIFICACIÓN Y COORDINACIÓN	CONTADOR I	20	10	Octubre	1995	30
2	DEPARTAMENTO PLANIFICACIÓN Y COORDINACIÓN	CONTADOR I	20	10	Noviembre	1995	30
3	DEPARTAMENTO PLANIFICACIÓN Y COORDINACIÓN	CONTADOR I	20	10	Diciembre	1995	0

© Sergio-Systems | 2015 Recursos Humanos - SIPLA - Ver. 1.0

FUENTE (Elaboración Propia)



ANEXO B. ÁRBOL DE PROBLEMAS.

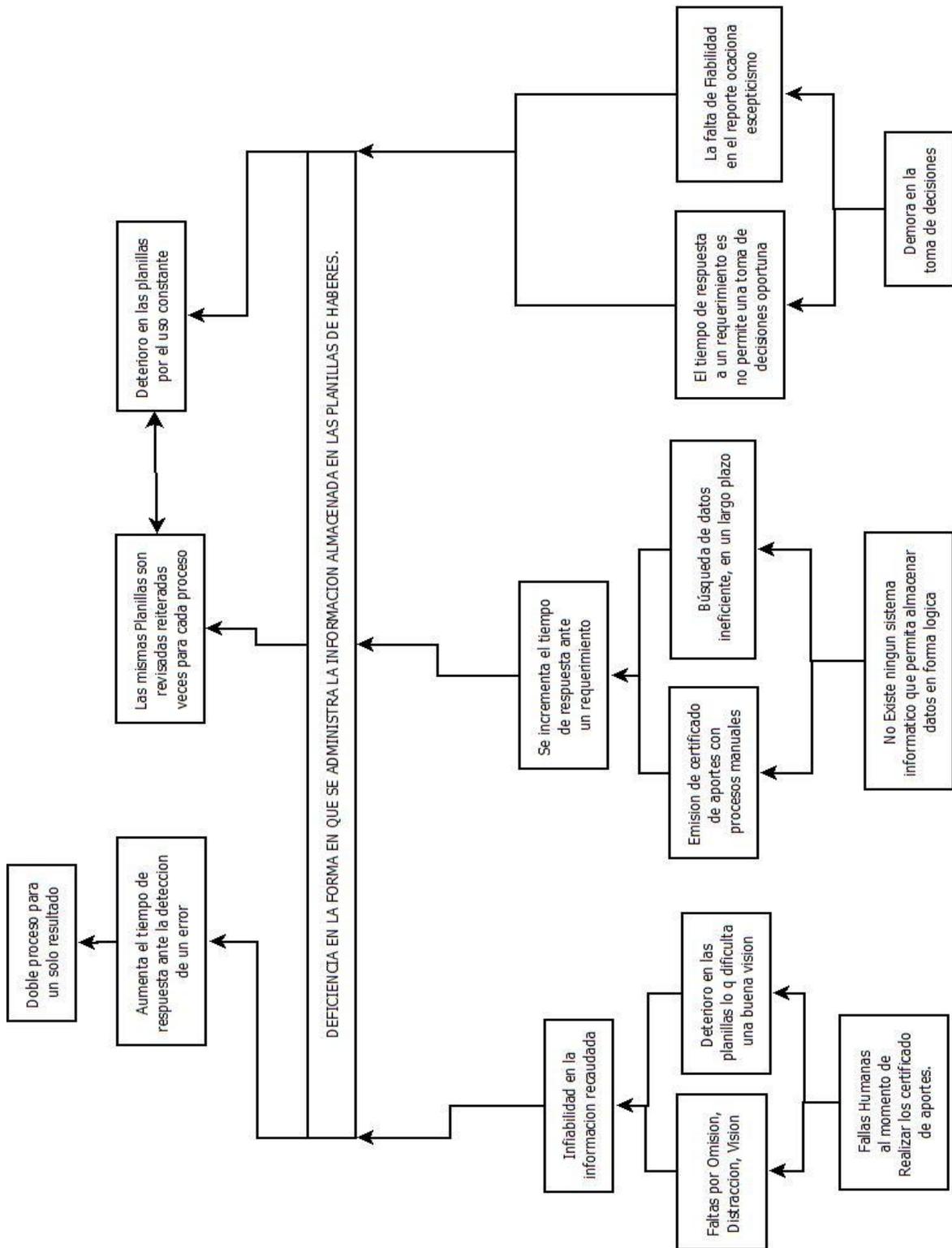


FIGURA B.1 Fuente (Elaboración Propia)

ANEXO C. MARCO LÓGICO

Tabla C.1 (Marco Logico)

RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN.</p> <p>Mejorar el tratamiento de la información de planillas de haberes, que implica la toma de dediciones</p>	<p>Incrementar la velocidad de tiempo de respuesta ante una búsqueda de datos y elaboración de reportes</p>	<p>Verificar información física con los datos lógicos almacenados en la base de datos. Comparar Reportes realizados en procesos pasados y verificar la veracidad con los reportes del sistema</p>	<p>Recibir el apoyo y aprobación de las máximas autoridades de la institución para la elaboración del proyecto</p>
<p>PROPÓSITO.</p> <p>Centralizar la información de las planillas en una base de datos para de esta manera mejorar el tiempo de respuesta en los requerimientos dados. Automatizar el proceso de generación de certificado de aportes,</p>	<p>Brindar seguridad y evitar errores de emisión de información, incrementando la eficiencia en los accesos a los datos.</p>	<p>Generar reportes solicitados por los usuarios de la información.</p>	<p>Contar con la aprobación colaboración del personal de la entidad, para que este proceso tenga utilidad.</p>
<p>PRODUCTO.</p> <p>Sistema Informático para la administración de la información</p>	<p>I/2015 se contara con un sistema informático para la unidad de escalafón dependiente del departamento de</p>	<p>Presentación de Avances según cronograma.</p>	<p>Existencia la colaboración en buenos términos del departamento UMSATIC encargado de</p>

<p>almacenada en las planillas de haberes, manual del usuario.</p>	<p>RRHH administrativos de la Universidad Mayor de San Andrés</p>	<p>Entrega de Código fuente del sistema y manual de usuario.</p>	<p>gestionar todas las tecnologías de la institución. Aprobación de la unidad interesada.</p>
<p>ACTIVIDADES.</p> <p>Entrevistas a los funcionarios que realizan las tareas mecánicas y conoedores en el tema de planilla de haberes.</p> <p>Planificación general del proceso.</p> <p>Desarrollo de la base de datos..</p> <p>Desarrollo de Software.</p> <p>Pruebas y el respectivo mantenimiento.</p> <p>Entrega final y capacitación del personal designado para la manipulación del sistema.</p>	<p>Conclusión con la planificación general de acuerdo a los tiempos establecidos en cronograma hasta el primer periodo de la gestión 2015.</p>	<p>Informe de aprobación del Jefe del departamento y el visto bueno de UMSATIC</p>	<p>Contar con equipos computacionales conectados a la red local para el uso del sistema.</p> <p>Contar con un personal capacitado para el uso adecuado del sistema.</p>

ANEXO D. DICCIONARIO DE DATOS

TABLA - PERSONA			
COLUMNA	TIPO DATO	LONGITUD	DESCRIPCIÓN
idPersona	Serial	-	Es la Clave Primaria de la tabla identificador
nombre1	Varchar	35	Es el Primer nombre de la Persona
Nombre2	Varchar	35	Es el Segundo nombre de la Persona
Nombre3	Varchar	35	Es el Tercer nombre de la Persona
Materno	Varchar	35	Apellido Materno de la Persona
Paterno	Varchar	35	Apellido Paterno de la Persona
apCasada	Varchar	35	El apellido de casada de la persona
Completo	Varchar	100	Nombre Completo de la Persona
Ci	Varchar	15	Numero de Cedula de Identidad
Fechanac	Date	-	Fecha de Nacimiento de la Persona
Expedido	Varchar	10	Descripción de la ciudad expedida de la cedula de identidad
Genero	Varchar	10	El género de la persona (Masculino/Femenino)
TABLA - FUNCIONARIO			
COLUMNA	TIPO DATO	LONGITUD	DESCRIPCIÓN
idFuncionario	Serial	-	Identificador de la Tabla
idPersona	Serial	-	Es la Clave Primaria de la tabla identificador
Estado	Varchar	10	Estado del registro(Activo/Inactivo)
Ítem	Integer	-	Ítem designado a la fecha, este dato varía de acuerdo a la movilidad que se le da a cada funcionario.
idUnidad	Serial	-	Identificador de la tabla de Unidad.
idCargo	Serial	-	Identificador de la tabla de Cargo.
idNivel	Serial	-	Identificador de la tabla de Nivel.

TABLA – PLANILLA			
COLUMNA	TIPO DATO	LONGITUD	DESCRIPCIÓN
idPlanilla	Serial	-	Identificador de la Tabla Planilla
Mes	Integer	-	Este valor representa un mes ejemplo 1 = Enero
Año	Integer	-	Este valor representa un año.
TABLA – HABERES			
COLUMNA	TIPO DATO	LONGITUD	DESCRIPCIÓN
idHaberres	Serial	-	Identificador de la Tabla Haberres
Diast	Double	-	Días Trabajados
Básico	Double	-	Este valor representa un año.
Categoría	Double	-	Representa el monto de Categoría
Totalmes	Double	-	Representa el monto de del Total del Mes
Totaldesc	Double	-	Representa el monto de del Total de Descuentos
Liquidop	Double	-	Representa el monto de del Líquido Pagable

DOCUMENTACIÓN

