

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



PROYECTO DE GRADO

“SISTEMA INTEGRADO DE CONTROL Y ADMINISTRACION DE PERSONAL “

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

AUTOR: Claudia Gimena Apaza Poma

TUTOR: Lic. Nancy Orihuela Sequeiros

REVISOR: Lic. Brígida Carvajal Blanco

PAZ - BOLIVIA

2007

Dedicatoria

A Dios por iluminarme y
acompañarme en todo
momento de mi vida.

A mi querida familia por su
por su apoyo incondicional para el
logro de mis metas

AGRADECIMIENTOS

Deseo expresar mi agradecimiento en primer lugar a Dios padre que esta en el cielo, porque sin el nada de esto seria realidad.

A mi Tutor Lic. Nancy Orihuela, por el entusiasmo mostrado y aliento infundido durante la elaboración de este proyecto y confiar en mi persona, dando sugerencias y enseñanzas enriquecedoras, guía y apoyo constante a mi tarea.

A mi Revisor Lic. Brígida Carvajal Blanco, por su disponibilidad en el proceso de revisión del proyecto, por las correcciones, observaciones y sugerencias durante el transcurso del desarrollo del presente proyecto.

Al Dr. Sabino Carme Zarate Director Nacional de RRHH por darme la oportunidad de aprender y trabajar en el desarrollo del presente proyecto, y colaboración para seguir adelante en mi proyecto.

Al Lic. Héctor Wilfredo Pacheco administrador de RRHH quien me brindo su tiempo desinteresadamente apoyándome en el transcurso del desarrollo del sistema.

A mi familia quienes sin tomar en cuenta mis fracasos ni mis fallas han estados siempre dispuestos animándome y ofreciendo lo necesario para que se lleve adelante este proyecto de grado.

A mis compañeros de la carrera en especial a mi tan apreciado Juan Carlos , por acompañarme en esta etapa tan hermosa, que es la vida universitaria.

RESUMEN

Yacimientos Petrolíferos Fiscales Bolivianos (Y.P.F.B.) es una empresa estatal petrolera encargada de la administración y fiscalización de los recursos petrolíferos existentes en Bolivia, la empresa cuenta con un departamento de Recursos Humanos, el cual se encarga de la administración y control del personal. Debido al crecimiento de la empresa, y el incremento en el personal, se producen retrasos en el procesamiento de la información, a causa de que algunos procesos se siguen realizando en forma manual.

En este sentido se ha desarrollado el Sistema Integrado de Control y Administración de Personal, realizando el mismo a través de la metodología ágil "ASD" (Desarrollo de Software Adaptable), el cual permite realizar iteraciones de pruebas veloces en el transcurso del desarrollo del sistema, con las que se ha logrado agilizar el trabajo y conseguir la adaptabilidad del sistema en la empresa. Como herramienta de apoyo se utilizó el Lenguaje de Modelado Unificado "UML".

El Sistema Integrado de Control y Administración de Personal proporciona una adecuada organización de la información manejada en el departamento de R.R.H.H., además cuenta con una integración centralizada de la información en una sola base de datos uniendo todos los sistemas de control de la división.

En la implementación del sistema, en la autenticación de usuarios se utiliza el algoritmo de encriptación SHA-1, finalmente cuenta con una generación de reportes gerenciales, en el que se evidencia toda la información operada o manejada con el sistema.

ÍNDICE

CONTENIDO	Pág.
CAPITULO 1 INTRODUCCIÓN	
1.1 Introducción.....	1
1.2 Antecedentes.....	2
1.3 Planteamiento del Problema.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo General.....	5
1.4.2 Objetivos Específicos.....	5
1.5 Justificación.....	5
1.5.1 Justificación Económica.....	5
1.5.2 Justificación Social.....	6
1.5.3 Justificación Técnica.....	6
1.6 Aporte Practico.....	6
1.7 Alcances.....	7
CAPITULO 2 MARCO TEÓRICO	
2.1 Introducción.....	9
2.2 Ingeniería de software.....	10
2.2.1 El aporte de una metodología de desarrollo de software....	11
2.2.2 Metodologías Agiles.....	12
2.3 Metodologías Agiles vs metodologías Tradicionales.....	15
2.4 ASD – Adaptive Software Development.....	16
2.4.1 Fase de Especulación.....	16
2.4.2 Fase de Colaboración.....	17
2.4.3 Fase de Aprendizaje.....	18
2.5 Lenguaje de modelado unificado.....	21
2.5.1 Diagrama de Casos de Uso.....	23
2.5.2 Diagrama de Clases.....	26

2.5.3	Diagrama de secuencia.....	26
2.5.4	Diagrama de Colaboración.....	27
2.5.5	Diagrama de Paquetes.....	27
2.5.6	Diagrama de Componentes.....	27
2.6	Herramientas de Construcción de Software.....	28
2.6.1	Sql Server.....	28
2.6.2	Visual Basic.....	28
2.7	Seguridad.....	29
2.7.1	Criptografía.....	29
2.8	Evaluación/ Pruebas del Software.....	30
2.8.1	Caso de Prueba.....	31
2.8.2	Procedimiento de Prueba	31
2.8.3	Componente de Prueba	31
2.8.4	Diseño de Casos de Prueba	31
2.8.5	Pruebas de Caja Blanca.....	31
2.9	Evaluación de la Calidad del Software.....	32
2.10	Estimación del Costo del Proyecto.....	36
2.10.1	Las Métricas Orientadas al Tamaño.....	36
2.10.2	El Método COCOMO II	37
CAPITULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA		
3.1	Fase de Especificación.....	40
3.1.1	Análisis y Determinación de Requerimientos.....	40
3.1.2	Captura de Requisitos	45
3.1.3	Modelo de Casos de Uso	46
3.1.4	Diagrama de Casos de Uso	47
3.1.5	Diagramas de Secuencia y Colaboración.....	66
3.2	Fase de Colaboración.....	69
3.2.1	Diagrama de Paquetes	69
3.2.2	Diagrama de Componentes	70
3.2.3	Diseño del Diagrama de clases	72
3.2.4	Diseño del Diagrama de Entidad-Relación.....	74

3.2.5	Diseño de la Base de Datos	74
3.3	Fase de Aprendizaje.....	76
3.3.1	Esquema de Funcionamiento de la Empresa	93

CAPITULO 4 PRUEBAS Y CALIDAD DEL SOFTWARE

4.1	Introducción	94
4.2	Pruebas de Caja Blanca	94
4.3	Funcionalidad	97
4.4	Confiabilidad	101
4.5	Mantenibilidad del Software	102
4.6	Portabilidad del Software	103
4.7	Facilidad de Instalación	103
4.8	Costo de Elaboración del Proyecto	105
4.8.1	Costo Total	106

CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusiones	107
5.2	Recomendaciones.....	108

BIBLIOGRAFIA	109
ANEXO A	110
ANEXO B	118
ANEXO C	125
ANEXO D.....	131
ANEXO E.....	

1.1 INTRODUCCIÓN

Yacimientos Petrolíferos Fiscales Bolivianos (Y.P.F.B.), nació de las cenizas de la guerra del chaco. La inmolación de 50.000 hombres defendiendo la riqueza petrolera, sirvió de marco al inicio de la entidad estatal petrolera Boliviana.

Yacimientos Petrolíferos Fiscales Bolivianos (Y.P.F.B.) empresa estatal petrolera de Bolivia tiene como función y responsabilidad principal administrar y fiscalizar los recursos petroleros existentes en Bolivia, realizar contratos con las empresas extranjeras que vienen a invertir al país, todo este trabajo se realiza en un marco legal de acuerdo a las leyes y reglamentos existentes en Bolivia.

Y.P.F.B. con casi 60 años de vida constituye una de las empresas más importantes en el ámbito nacional e internacional cuando menos en el ámbito de empresas estatales de Latinoamérica.

Los sistemas de información de control de personal son parte integral para desarrollar el trabajo de todo ejecutivo que supervisa, dirige o administra las actividades de subordinados.

El presente trabajo surge de la necesidad de automatizar las tareas del departamento de personal en Y.P.F.B., siendo el objetivo principal relacionar y mejorar el tiempo de procesamiento de las diferentes tareas contar con información rápida y precisa que requieran en su momento.

Se plantea el Sistema Integrado de Control y Administración de Personal (SICAP) orientado a cliente servidor donde su labor principal será relacionar y concentrar las diferentes labores que desempeña cada responsable del departamento: Registro de Personal, Elaboración de Planillas de Viáticos, Recepción de formulario IVA, planilla de lactancia, control de vacaciones, control de asistencia,

elaboración de planillas de fin de mes, elaboración de finiquitos, planilla del servicio de te, planilla de prenatal, natalidad y lactancia, todo este trabajo se automatizará haciendo uso de los recursos disponibles que la empresa tiene y sobre todo utilizando tecnología actual que permita contar con un sistema de información para la toma de decisiones.

El Proyecto comprende 5 capítulos los cuales son:

Capitulo 1 Introducción, se describe brevemente los antecedentes, la situación de la problemática que tropiezan la empresa, el cual constituyó la base para formular el problema principal, además se representan los objetivos específicos y el general.

Capitulo 2 Marco Teórico, se presentan y describen tanto las definiciones teóricas de la metodología, así como conceptos que se utilizaron en el presente proyecto.

Capitulo 3 Análisis y diseño, donde se realiza un análisis de la situación actual de los procesos de administración y control del personal para luego aplicar la metodología mencionada y realizar el diseño propuesto para llegar al objetivo trazado.

Capitulo 4 Calidad del Software, donde se realiza una serie de medidas de calidad del sistema.

Finalmente se presenta el Capitulo 5 Conclusiones y Recomendaciones, de todo el trabajo realizado en el presente proyecto.

1.2 ANTECEDENTES

Y.P.F.B. cuenta con diferentes Gerencias, Direcciones y Departamentos y Oficinas, toda la labor de control de personal se concentra en la gerencia de Administración y Finanzas específicamente en la oficina del Personal. En la investigación preliminar se evidenció que se lleva a cabo procesos semiautomáticos, los mismos que hacen que el trabajo avance con algunas falencias que se requiere mejorar.

Actualmente no existe un sistema que realice todas las tareas de la Oficina de Personal de Y.P.F.B.

Se tiene dos sistemas que ayudan a realizar el control de personal los mismos que tienen las siguientes características:

- Sistema Win Pack

Este sistema se desarrollo en el extranjero y fue comercializado en Bolivia por la Empresa Global Tech, utiliza una base de datos Access que tiene un entorno Visual y trabaja en plataforma WIN/98, existe tres paneles de control y cada empleado cuenta con una tarjeta magnética que permite realizar el mercado respectivo, como principales tareas tiene:

- Control de entrada y salida del personal.
- Reporte de horas trabajadas.
- Almacena datos del personal por Gerencia, Departamento y Oficina.
- Control de Atrasos.

Este sistema solo se encarga de la asistencia del personal, entradas y salidas.

- Sistema de Planillas de Y.P.F.B.

Fue desarrollada en la ciudad de Santa Cruz por el departamento de sistemas y tiene las siguientes funciones:

- Planilla de haberes
- Planilla de aportes
- Planilla de descuentos
- Planilla del IVA
- Planilla de personal temporal
- Planilla de horas extras
- Planilla de líquido pagable.
- Planilla del servicio de Té.

El sistema mencionado anteriormente (sistema de planillas de Y.P.F.B.), tiene un entorno en modo texto y necesariamente para la manipulación se debe contar con un usuario con mucho conocimiento del mismo sistema, además que el tiempo de procesamiento es muy lento.

Como se describió antes ambos sistemas son solo para un tipo de trabajo, el primero para el control asistencia y el segundo para la elaboración de planilla de fin de mes.

1.3 PLANTEAMIENTO DEL PROBLEMA

Dado el actual crecimiento de la información manejada en el departamento de Recursos Humanos surge la necesidad de que dicha división tenga un sistema de información, que facilite la automatización de la misma para lograr un manejo eficiente, el problema que se pudo observar es:

El problema El departamento de Recursos Humanos de Y.P.F.B., tiene demoras en el procesamiento de la información y pérdidas de datos.

Las actividades de control de asistencia, horas extras y faltas son aún realizadas manualmente, esto provoca datos erróneos y cálculos inexactos.

No se tiene un resumen del file por trabajador.

El control de vacaciones tiene falencias en el cálculo de días de vacación.

La elaboración de planillas de fin de mes se realiza en la ciudad de Santa Cruz y demora mucho en procesar.

Afecta a

- RRHH (estrictamente al departamento de recursos humanos).
- Funcionario (al mismo funcionario a quien se le realizan los pagos y beneficios).
- Y.P.F.B. (ala empresa por que puede percibir pérdidas económicas).

El impacto asociado es

- Proceso lento y costoso de las diferentes actividades.

Una solución adecuada seria Implementar el sistema informático que automatice el procesamiento de la información y el control de asistencia, usando una red local o intranet con un entorno cliente servidor y generar interfaces amigables y sencillas.

1.4 OBJETIVOS

1.4.1 Objetivo general

Diseñar e implementar un sistema informático de control de Personal de Y.P.F.B., Integrando los procesos de control y administración de personal en el sistema de información automatizado, considerando medidas de seguridad para la implementación del mismo.

1.4.2 Objetivos Específicos del proyecto son:

- Automatizar e implementar los procesos de control de asistencia, elaboración de planillas de sueldos y finiquitos.
- Administrar la configuración y los cambios, considerando la facilidad de actualización y posibilidad de reutilización de código.
- Establecer esquemas de seguridad para asegurar la integridad, consistencia y fiabilidad de la información.
- Optimizar el tiempo durante la elaboración de reportes y informes referentes a registros y afiliación del personal.
- Generar distintos reportes, para su publicación interna y externa.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Económica

Sistematizar las tareas que en la organización se realizan manualmente o semiautomáticamente, es de gran utilidad puesto que con la implementación del sistema el procesamiento de finiquitos es más rápido y preciso, esto evitará pérdidas económicas a la empresa por retraso en el pago de beneficios sociales, lo dicho anteriormente se podrá comprobar en los reportes de fin de año de finiquitos (que se podrá observar en el anexo C) ya que anteriormente por errores de cálculo cancelaron por de más las reliquidaciones (beneficios social que se da al trabajador por años de servicio en el momento de retiro de la empresa u finalización del contrato) al personal de la empresa eso provocando pérdidas

económicas a la empresa, el SICAP tendrá un mejor control de personal y es creado para mejorar beneficios económicos y laborales para la institución.

1.5.2 Justificación Social

En la actualidad los sistemas informáticos se han convertido en una necesidad muy importante para la sociedad, con el SICAP se realizara el trabajo mas eficiente, dinámico y preciso. El sistema beneficia a la empresa y a los empleados, tendrá gran aceptación por parte del usuario por que es de fácil manipulación y cumplirá las necesidades que se requieren respecto a cada trabajador de la empresa.

Este sistema ahorra tiempo en el proceso de elaboración de reportes de todos los empleados o de manera personal de cada empleado de la empresa. Con el nuevo sistema los beneficios para los empleados crecerán, ya que con ellos se tendrá una mejor atención.

1.5.3 Justificación Técnica

El desarrollo e implementación del sistema se realiza con los recursos existentes en la empresa, ya que se cuenta con software actual con sus respectivas licencias, también se cuenta con computadoras de ultima generación, lo cual representan costos bajos y de fácil aplicación, así mismo sistematizar ciertas actividades que en la organización la realizaran manualmente o semiautomático, es de gran utilidad por que este sistema esta creado para generar beneficios económicos.

Es posible plantear un modelo de solución a través de un sistema de información para el departamento de RRHH.

Eliminación de procesos manuales que originan demoras e inconsistencias de la información generada de cada proyecto.

1.6 APORTE PRÁCTICO

El sistema SICAP esta orientado a realizar un manejo más integrado al control de personal usando una estructura cliente servidor a diferencia de otros sistemas se

utiliza la aplicación de la metodología ágil ASD (Adaptive Software Development), brindando un mejor enfoque sistémico, también este sistema está orientado a la transparencia de la información para todo el personal de Y.P.F.B. en general utilizando procesamiento en tiempo real al realizar las actualizaciones correspondientes.

El aporte principal del presente proyecto es generar el informe gerencial de gastos y movimiento de personal de manera automática (ver anexo C), obteniendo así información sobre el presupuesto del mes, de los siguientes informes: importe de haberes, importes de horas extras, gastos por comisión de viaje, pagos adicionales, movimiento de personal en altas y bajas, número de personal profesionales y no profesionales, este informe se genera integrando y relacionando las diferentes labores de la oficina de personal, además nuevos sistemas contribuirán a un ambiente de trabajo más fácil, dinámico y seguro.

1.7 ALCANCES

El presente proyecto de grado se centra en el área de control de personal de una empresa, en donde se identificaron los siguientes módulos.

El sistema SICAP está orientado a:

- Implementar el módulo de control de asistencia del personal.
 - Control del personal, atrasos y horas extras.
- Implementar el módulo de planillas de fin de mes.
 - Emitir reportes de fin de mes personales y reportes generales.
 - Emitir reportes gerenciales.
 - Emitir reportes de consulta de viáticos y descuentos laborales.
- Implementar el módulo de viáticos.
 - Emitir reportes generales de fin de mes de viáticos realizados en la fecha.
 - Emitir reportes de egresos de viáticos realizados en el año.
- Implementar el módulo de vacaciones.
 - Realizar el cálculo exacto de las vacaciones de cada personal.
 - Realizar el cálculo exacto de reintegro a la empresa.

Referencias de hardware y software

- Pruebas de iteraciones con las cuales nos permitirán encontrar las fallas.
- Se realizara un entorno vía intranet y no vía Web
 - Realizar una red local dentro la empresa en un entorno cliente servidor.
- Se utiliza una plataforma Windows Xp, SQL server2000, y visual Basic 6.0.



MARCO TEÓRICO

2.1 INTRODUCCIÓN

El marco teórico presentado a continuación es el producto de la exposición de los principios de la ciencia de la computación relacionadas a los sistemas, métodos y enfoques metodológicos que se consideran validos para el desarrollo de proyectos de software en el medio. Como ser la utilización del lenguaje UML para especificación del problema y objetivos del sistema.

Hoy en día el software ha adquirido mayor importancia en el convivir cotidiano, pues se encuentra presente en la mayoría de las actividades de nuestra sociedad, ha pasado a ser una de las industrias más importantes del mundo. En este escenario es imperativo para los ingenieros de software encontrar formas de desarrollo más eficaces y eficientes que permitan enfrentar esta compleja tarea.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodologías que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usar. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo mas actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo mas complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al

individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software [COT1994].

2.2 INGENIERÍA DE SOFTWARE

La Ingeniería del Software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la Ingeniería del Software como una nueva área de la ingeniería, y el ingeniero de software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidades que cumplir, junto a una, ya, reconocida consideración social en el mundo empresarial y, por suerte, para esas personas con brillante futuro. La Ingeniería del Software trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc [press,2002].

Podemos rescatar las siguientes características:

- La confiabilidad se puede definir también como la probabilidad en que un producto software realizará su función prevista sin incidentes por un período de tiempo especificado y bajo condiciones indicadas.
- La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso
- La adaptabilidad es una de las características que más se busca de un sistema software, es la capacidad de lograr adaptarse a las exigencias del usuario, por ello se debe tener claro este concepto el cual puede

definirse de la siguiente manera: Acomodar, ajustar una cosa a otra. Hacer que un objeto o mecanismo desempeñe funciones distintas de aquellas para las que fue construido. Acomodarse a diversas circunstancias.

2.2.1 El aporte de una metodología de desarrollo de software

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto [COT1994].

La aplicación de metodologías en el desarrollo de un software nace por las fallas que cometemos al desarrollar nuestro software las más comunes fallas son:

- 75% de las empresas de software han sido calificadas como caóticas.
- Los desarrolladores son muy entusiastas y optimistas.
- Se requiere más gente.
- Subestimar el esfuerzo que se requiere.
- Presión externa de Mercadotecnia, clientes y la alta gerencia.
- El estrés causa el 40% de los errores en la creación o integración de aplicaciones.
- 84% de los proyectos de software fallan.
- 30% son cancelados.
- El resto en sobrepasan el presupuesto.

2.2.2 Metodologías Ágiles

Varios grupos de desarrolladores preocupados ante los problemas surgidos de la aplicación de las metodologías tradicionales proponen otras metodologías que tratan de adaptarse de mejor manera la realidad del desarrollo de software. Y es así como en el año 2001 los principales promotores de estas teorías se reunieron para debatir los puntos que tenían en común y el resultado fue la creación “The Agile Alliance” quienes dan a conocer el Manifiesto Ágil. Este manifiesto establece las ideas principales de las metodologías ágiles:

Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados

Es más importante crear un producto software que funcione, antes que escribir documentación exhaustiva

- La colaboración con el cliente debe prevalecer sobre la negociación de contratos
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Hay diversas metodologías ágiles, de las cuales podemos citar las siguientes:

Extreme Programming (XP)

Este es el método que más popularidad ha alcanzado entre las metodologías ágiles, y posiblemente sea también el más trasgresor de la ortodoxia basada en procesos.

Su creador, Kent Beck fue el alma matar del Manifiesto Ágil.

Extreme Programming (XP) se rige sobre la suposición de que es posible desarrollar software de gran calidad a pesar, o incluso como consecuencia del cambio continuo. Su principal asunción es que con un poco de planificación, un poco de codificación y unas pocas pruebas se puede decidir si se está siguiendo

un camino acertado o equivocado, evitando así tener que echar marcha atrás demasiado tarde [CIS II 01]..

Scrum

No es propiamente un método o metodología de desarrollo, e implantarlo como tal resulta insuficiente. Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas [CIS, II 01].

Los principios de Scrum son:

- Equipos autogestionados.
- Una vez dimensionadas las tareas no es posible agregarles trabajo extra.
- Reuniones diarias en las que los miembros del equipo se plantean 3 cuestiones:
 - ¿Qué has hecho desde la última revisión?
 - ¿Qué obstáculos te impiden cumplir la meta?
 - ¿Qué vas a hacer antes de la próxima reunión?
- Iteraciones de desarrollo de frecuencia inferior a un mes, al final de las cuales se presenta el resultado a los externos del equipo de desarrollo, y se realiza una planificación de la siguiente iteración, guiada por cliente.

Crystal Methodologies

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del

tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros) [CIS, II 01].

Dynamic Systems Development Method (DSDM)

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases [CIS, II 01].

Adaptive Software Development (ASD)

Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes de software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo [CIS, II 01].

Feature -Driven Development (FDD)

Prescribe un proceso iterativo de 5 pasos, con iteraciones de dos semanas. El punto de referencia son las características que debe reunir el software, y se centra en las fases de diseño e implementación del sistema [CIS II 01].

Después de haber estudiado estas metodologías todas buscan el manifiesto ágil, pero se puede observar que el ciclo de vida de XP es mayor a ASD por ello una iteración es más demorativa, Cabe observar que XP es óptimo al contar con un

equipo de trabajo, pero al trabajar individualmente XP es mas lento que ASD. Por otra parte scrum presenta iteraciones, denominadas sprints, con una duración de 30 días lo cual alarga el proyecto aun más. Como en el presente proyecto se desea satisfacer las necesidades de una empresa seria, el sistema debe lograr adaptarse a estas y se debe agilizar el desarrollo del sistema en tanto el producto sea lo suficientemente bueno, es por esta razón que en la línea de las metodologías ágiles se ha optado por ASD.

2.3 METODOLOGÍAS ÁGILES VS. METODOLOGÍAS TRADICIONALES

Vamos a enumerar las principales diferencias de una Metodología Ágil respecto de las Metodologías Tradicionales (llamadas peyorativamente “no ágiles” o “pesadas”). La Tabla 2.1 recoge estas diferencias que no se refieren sólo al proceso en sí, sino también al contexto de equipo y organización que es más favorable a cada uno de estas filosofías de procesos de desarrollo de software [JHC,2000].

Tabla 2.1 Características de las Metodologías

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Ciente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños.	Aplicables a proyectos de cualquier

Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Fuente: [JHC,2000].

2.4 ASD – ADAPTIVE SOFTWARE DEVELOPMENT

Jim Highsmith en su libro [Highsmith, 1999] es la mente detrás de este proceso ágil. ASD consiste en un cambio de filosofía en las organizaciones pasando de la transición del modelo Comando-Control al modelo Liderazgo-Colaboración.

ASD propone utilizar en cambio el ciclo de vida de la Figura 2.1, Especular Colaborar-Aprender.

2.4.1 Fase de Especulación

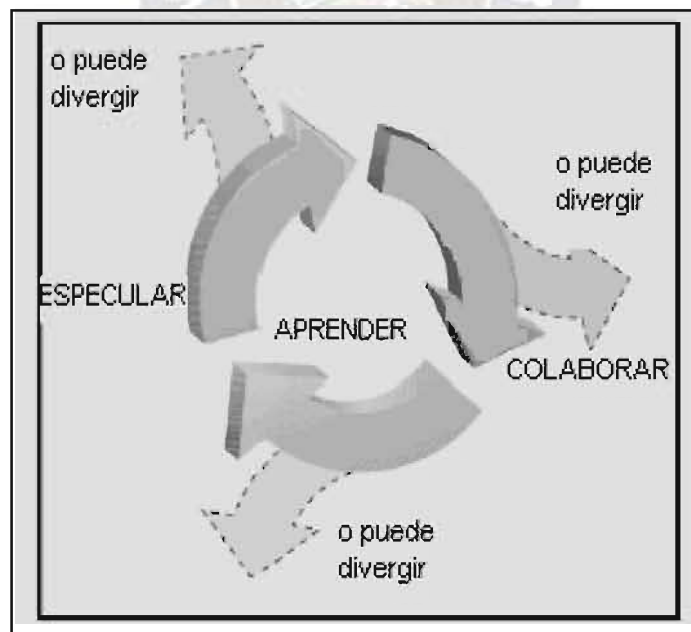
El proyecto comienza con una fase de especulación en que se lleva a cabo la planificación tentativa del proyecto en función de las entregas que se irán realizando.

La utilización del verbo Especular demuestra el interés de Highsmith en demostrar la naturaleza impredecible de los sistemas complejos. En esta etapa se fija un

rumbo determinado a ser seguido en el desarrollo, sabiendo a partir de ese momento que no será el lugar en que finalizará el proyecto.

En cada iteración, se aprenderán nuevas funcionalidades, se entenderán viejas cuestiones, y cambiarán los requerimientos. Gracias a centrarse en la especulación, ASD permite administrar estos proyectos de alto cambio y rápido desarrollo que se encuentran en el *borde del caos*. Sin embargo, no es más que una especulación ya que el carácter adaptativo del proceso permite pequeñas desviaciones en un sentido por lo que Highsmith sugiere que cada ciclo se componga de un mix entre funcionalidades críticas, útiles, y opcionales, previendo los posibles retrasos que puedan existir mediante el movimiento de las funcionalidades de menor prioridad a futuros ciclos y grandes desviaciones en otro, las cuales son utilizadas para la exploración del dominio y de la aplicación, que puede llevar a cambiar el rumbo del proyecto estos desvíos está representado por las flechas de divergencia [HIJ, 1999] Highsmith.

Figura 2.1 Ciclo de Vida Adaptativo



Fuente: [High, 2000]

2.4.2 Fase de Colaboración

La siguiente fase del ciclo de vida, Colaborar, es aquella en la que se construye la funcionalidad definida durante la especulación. ASD define un Componente como un grupo de funcionalidades o entregables a ser desarrollados durante un ciclo iterativo. Durante cada iteración el equipo colabora intensamente para liberar la funcionalidad planificada. También, existe la posibilidad de explorar nuevas alternativas, realizar pruebas de concepto, pudiendo eventualmente alterar el rumbo del proyecto profundamente. ASD no propone técnicas ni prescribe tareas al momento de llevar a cabo la construcción simplemente mencionando que todas las prácticas que sirvan para reforzar la colaboración serán preferidas, siguiendo de esta forma la línea de las metodologías ágiles respecto a la orientación a componentes. El énfasis se ubica en la relaciones entre las personas que deben estar lo suficientemente lubricadas para generar una propiedad imprescindible de los organismos complejos: emergencia. La emergencia es una propiedad de los sistemas adaptativos complejos que crea alguna propiedad más grande del todo (comportamiento del sistema) a partir de la interacción entre las partes (comportamiento auto-organizativo de los agentes). Gracias a esta propiedad los grupos de desarrollo logran sacar lo mejor de si en la el borde del caos [HIJ,1999] Highsmith.

2.4.3 Fase de Aprendizaje

La fase final de ASD, Aprender, consiste en la revisión de calidad que se realiza al final de cada ciclo. En la misma se analizan cuatro categorías de cosas para aprender [HIJ,2000]:

- Calidad del resultado de la desde la perspectiva del cliente
- Calidad del resultado de la desde la perspectiva técnica

Para evaluar la calidad desde el punto de vista del cliente se sugieren utilizar grupos de enfoque en el cliente, mediante los cuales se explora un modelo de la aplicación y se anotan los requerimientos de cambio del cliente.

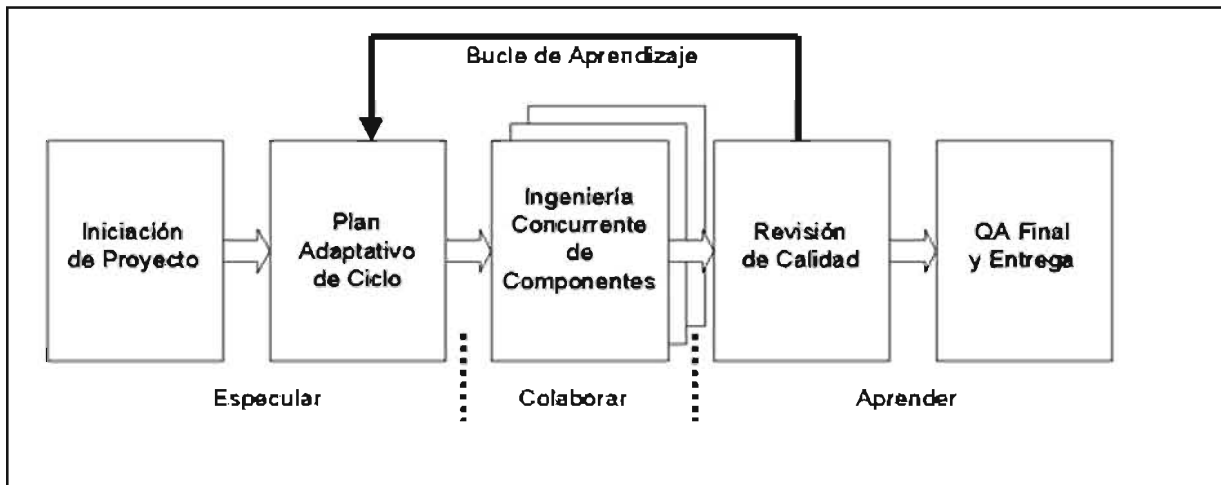
Las revisiones al diseño, al código o a las pruebas permitirán aprender sobre la calidad de los mismos. En este caso, el énfasis estará puesto en aprender cuales han sido los errores o desvíos y poder resolverlos, y no en encontrar culpables. Asimismo, está es la etapa en que se evaluarán las exploraciones que se hayan realizado dando la capacidad de poder modificar la arquitectura del sistema si se ha encontrado algún camino que se ajusta mejor a lo que necesita el usuario o si han cambiado los requerimientos. El tercer proceso de feedback está relacionado con la interacción entre las partes, la dinámica de grupo, y las técnicas empleadas. Para medir la performance y el grado de cohesión del mismo, se podrán realizar al final de cada ciclo pequeñas reuniones de postmortem. En las mismas se discuten los aspectos del proceso que contribuyen al desarrollo y aquellos que deben ser descartados por su influencia negativa.

En relación al status del proyecto, se realizarán revisiones para determinar el estado del mismo en relación a lo planificado. En este momento, se detectarán posibles diferencias que pueden surgir de la exploración y que cambiarán el rumbo a que apuntaba el proyecto.

En la Figura 2.2 se puede ver el detalle interno de cada fase como ya fue explicado, mostrándose con una flecha que trasciende las tres fases en sentido inverso, el bucle de aprendizaje. Este bucle es algo crítico para ASD ya que denota un cambio en el esquema tradicional de la vista de un sistema en que se tenía un bucle de control para detectar diferencias y corregirlas. Es decir, en las metodologías tradicionales las diferencias respecto a lo planificado eran vistas como errores que debían ser enmendados para que cumplieran lo pautado. ASD y las metodologías ágiles plantean la necesidad de que el feedback necesario sea para aprender, nos da la posibilidad de entender más respecto al dominio y construir la aplicación que mejor satisfaga las necesidades del cliente. Highsmith lo expone claramente en la siguiente frase:

En ambientes complejos, el seguir un plan al pie de la letra produce el producto que pretendíamos, pero no el producto que necesitamos.

Figura 2.2 Actividades del Ciclo Adaptativo



Fuente: [HIJ,2000]

Este método ágil pretende abrir una tercera vía entre el “desarrollo monumental de software” y el “desarrollo accidental”, o entre la burocracia y la adhocracia. En el cual se que indica que deberíamos buscar más bien, “el rigor estrictamente necesario”; para ello hay que situarse en coordenadas apenas un poco fuera del caos y ejercer menos control que el que se cree necesario [HIJ2000].

ASD presupone que las necesidades del cliente son siempre cambiantes. La iniciación de un proyecto involucra definir una misión para él, determinar las características y las fechas y descomponer el proyecto en una serie de pasos individuales, cada uno de los cuales puede abarcar entre cuatro y ocho semanas. Los pasos iniciales deben verificar el alcance del proyecto; los tardíos tienen que ver con el diseño de una arquitectura, la construcción del código, la ejecución de las pruebas finales y el despliegue.

Un ciclo de vida es una iteración; este ciclo se basa en componentes y no en tareas, es limitado en el tiempo, orientado por riesgos y tolerante al cambio. Que se base en componentes implica concentrarse en el desarrollo de software que trabaje, construyendo el sistema pieza por pieza. En este paradigma, el cambio es

bienvenido y necesario, pues se concibe como la oportunidad de aprender y ganar así una ventaja competitiva; de ningún modo es algo que pueda ir en detrimento del proceso y sus resultados.

Highsmith piensa que los procesos rigurosos (repetibles, visibles, medibles) son encomiables porque proporcionan estabilidad en un entorno complejo, pero muchos procesos en el desarrollo (por ejemplo, el diseño del proyecto) deberían ser flexibles. La clave para mantener el control radica en los “estados de trabajo” (la colección de los productos de trabajo) y no en el flujo de trabajo (workflow). Demasiado rigor, por otra parte, acarrea rigor mortis, el cual impide cambiar el producto cuando se introducen las inevitables modificaciones. En la moderna teoría económica del retorno creciente, ser capaz de adaptarse es significativamente más importante que ser capaz de optimizar [HIJ,2000].

El modelo de Highsmith es, naturalmente, complementario a cualquier concepción dinámica del método; no podría ser otra cosa que adaptable, después de todo, y por ello admite y promueve integración con otros modelos y marcos como el lenguaje de modelado UML. Un estudio de Dirk Riehle compara ASD con XP, encontrando similitudes y diferencias de principio que pueden conciliarse con relativa facilidad, al lado de otras variables que son incompatibles. La actitud de ambos métodos frente a la redundancia de código, por ejemplo, es distinta; en XP se debe hacer todo “una vez y sólo una vez”, mientras que en ASD la redundancia puede ser un subproducto táctico inevitable en un ambiente competitivo y debe aceptarse en tanto el producto sea “suficientemente bueno”. En materia de técnicas, ASD las considera importantes pero no más que eso; para XP, en cambio, los patrones y la refactorización son indispensables.

2.5 LENGUAJE DE MODELADO UNIFICADO UML.

El lenguaje Unificado de modelado UML es una notación que es el resultado de la evolución de las notaciones previas en ingeniería de software, toma los aspectos fuertes de tres metodologías anteriores: OMT, Booch y OOSE. [BOOCH]. El

Lenguaje de Modelado Unificado UML es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software".

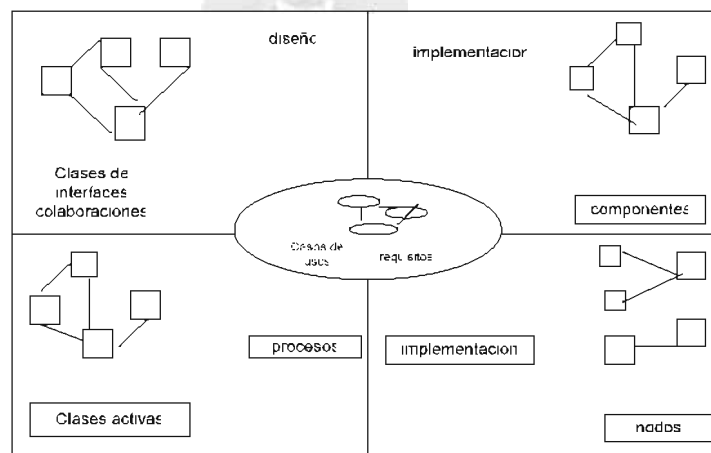
El UML es el Lenguaje de Modelado Unificado Orientado a Objetos, UML no es un método porque no tiene noción de proceso el cual es una parte importante de un método. Ahora bien si UML no es método; entonces, varios especialistas en desarrollo de sistemas de información arguyen de que existe la necesidad de adoptar un Proceso de Desarrollo de sistemas para enmarcar las fases importantes que sigue el UML, por ello los desarrolladores de proyectos de sistemas de información emplean el Procesos Unificado para dar soluciones adecuadas a las necesidades de los clientes [BOOCH].

La notación UML se fundamenta en principios de modelado, lo cual es importante para toda implementación de un sistema de información.

El UML debe adoptar el Proceso Unificado de Desarrollo para modelar las actividades de un proyecto. [BOOCH].

Los Diagramas a utilizar en las diferentes etapas del desarrollo de los sistemas de información, pueden variar dependiendo del tamaño y tipo de sistema, por lo que es necesario organizarlos según las fases del Proceso Unificado. [BOOCH].

Figura2.3 Arquitectura del Sistema con Uml.



Fuente: [BOOCH]

Diagramas de uml

Los elementos de UML se muestran mediante Diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como modelos. [UML,2002] UML presenta varios Diagramas donde cada uno representa un aspecto del sistema. Los Diagramas se emplean según el requerimiento de cada proyecto y según la complejidad del sistema de información a solucionar. [UML,2002].

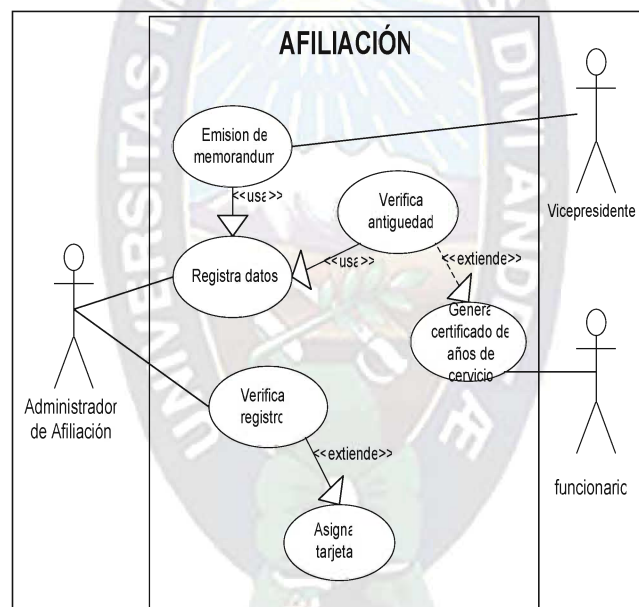
Es así que, dado un sistema a desarrollar no es necesario emplear todos los Diagramas; para sistemas sencillos un Diagrama de clases junto con un par de Diagramas de actividades e interacción sería suficiente, asimismo si los sistemas son complejos requieren de la utilización de más Diagramas, debido a que requieren de etapas incrementales e iterativas(ciclos de desarrollo) en el análisis, diseño e implementación, por ello es que el conjunto actividades deberá especificar la etapa de desarrollo y los Diagramas recomendados. [UML,2002].

- Un modelo UML está compuesto por tres clases de bloques de construcción.
 1. Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
 2. Relaciones: relacionan los elementos entre sí.
 3. Diagramas: Son colecciones de elementos con sus relaciones.
 - Diagramas de casos de uso
 - Diagramas de clases
 - Diagramas de interacción
 - Secuencia
 - Colaboración
 - Diagramas de paquetes
 - Diagramas de componentes

2.5.1 Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el Diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea [BOOCH].

Figura 2.4 Diagrama de casos de uso



Fuente: Elaboración Propia

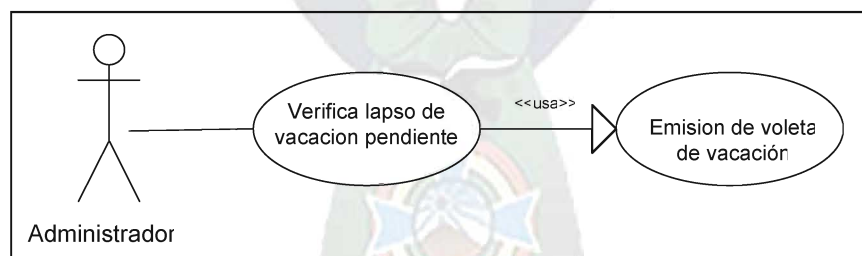
Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

- **Actores-** Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema. Se representa mediante

una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.

- **Casos de Uso.-** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.
- **Relaciones entre Casos de Uso.-** Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. Para el caso de que queramos utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos: • Incluye (<>): Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base.

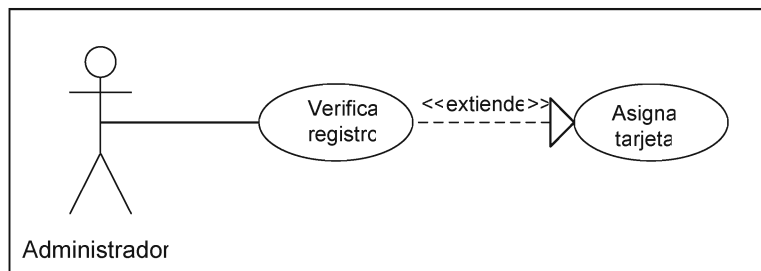
Figura 2.5 Ejemplo de Relación <<usa>>



Fuente: [BOOCH].

Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema (a diferencia de la relación incluye que se da siempre que se realiza la interacción descrita)

Figura 2.6 Ejemplo de Relación <<extend>>



Fuente: [BOOCH].

Ambos tipos de relación se representan como una dependencia etiquetada con el estereotipo correspondiente (<> o <<>>), de tal forma que la flecha indique el sentido en el que debe leerse la etiqueta. Junto a la etiqueta <> se puede detallar el/los puntos de extensión del caso de uso base en los que se aplica la extensión.

2.5.2 Diagrama de Clases

El Diagrama de Clase es el Diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el Diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo Diagrama, y se modifica para satisfacer los detalles de las implementaciones [BOOCH].

- **Clases.**- Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática, con los atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

2.5.3 Diagrama de secuencia

Un Diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas

en el contexto de una operación como se podrá ver en el capítulo siguiente [BOOCH].

2.5.4 Diagrama de colaboración

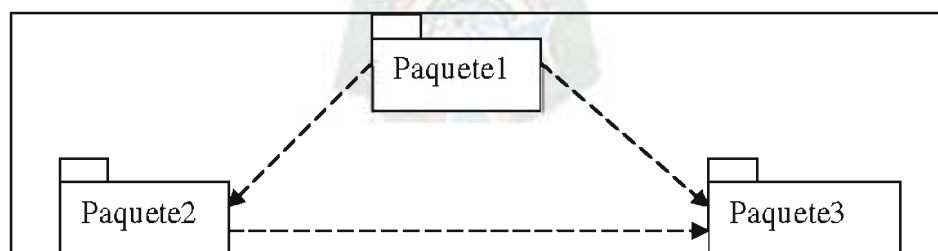
Los Diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los Diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los Diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los Diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los Diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de Diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa, como se podrá observar en el capítulo siguiente [BOOCH].

2.5.5 Diagrama de paquetes

Los paquetes son un mecanismo de estructuración, utilizados para agrupar cualquier tipo de elementos de los modelos que tienen alguna relación semántica. Cuyo propósito general es dividir el sistema en varios subsistemas para simplificar el modelado del mismo. [Booch 99].

Figura 2.7 Diagrama de paquetes

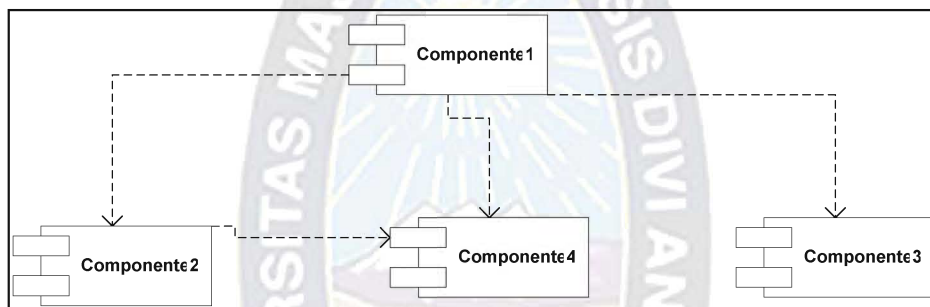


Fuente: Elaboración Propia

2.5.6 Diagrama de Componentes

Se utilizan para modelar la vista estática de un sistema. Muestra las organizaciones y dependencias entre un conjunto de componentes. No es necesario que un Diagrama incluya todos los componentes de un sistema, normalmente se realizan por partes. Cada Diagrama describe un apartado del sistema. En el se sitúan librerías, tablas, archivos ejecutables y documentos que formen parte del sistema. Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema [Booch, 99].

Figura 2.8 Diagrama de componentes



Fuente: Elaboración Propia

2.6 HERRAMIENTAS DE CONSTRUCCIÓN DE SOFTWARE

2.6.1 Sql Server

Microsoft SQL Server es un programa informático de gestión y administración de bases de datos relacionales basada en el lenguaje SQL, que incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Se eligió esta base de datos por que se adapta con el sistema a desarrollar por permitir el manejo óptimo de cliente servidor y una mejora considerable de todo el proceso de réplica.

También se adecuo a los requerimientos de la empresa por que ellos impusieron el software (SQL SERVER 2000) con la respectiva licencia de manejo.

2.6.2 Visual Basic

Visual Basic es un lenguaje de programación desarrollado por Microsoft. Visual Basic es un lenguaje visual que desciende del lenguaje de programación BASIC.

Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma.

Se eligió este tipo de interfaz por su buen entorno visual y amigable a los ojos del usuario. Una de las ventajas de Visual BASIC es que muchos de sus controles Standard son "data aware" o "data bound", es decir permiten conectarse o tomar sus datos directamente de una tabla de la base de datos.

También se adecuó a los requerimientos de la empresa por que ellos impusieron el software (VISUAL BASIC 6.0) con la respectiva licencia de manejo.

2.7 SEGURIDAD

La seguridad es uno de los aspectos más importantes y conflictivos en el uso de Intranet. La falta de una política de seguridad global esta frenando el desarrollo de Intranet en diferentes áreas, por lo que es necesario crear entornos seguros.

Para el desarrollo de sistema se desarrolla un entorno de seguridad para proteger la información enviada del servidor hacia el cliente y viceversa, de esta manera evitar el uso no autorizado de las funciones que ofrece el sistema, dotando al mismo de servicios de seguridad.

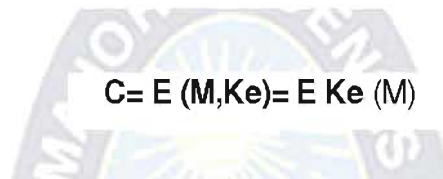
2.7.1 Criptografía

Las raíces etimológicamente de la palabra criptografía son Kriptos, que significa oculto y graphos, que significa como escribir, lo que da una clara idea de su definición clásica; arte de escribir mensajes en clave secreta o enigmáticamente.

Ahora bien la criptografía corresponde solo a una parte de la comunicación. Si se requiere secreto para la comunicación, es por que existe desconfianza o peligro de que el mensaje transmitido sea interpretado por un enemigo y ese tal vez quiera

alterar el mismo para su beneficio, para esto utilizará un conjunto de técnicas y métodos que constituyen una ciencia conocida como criptoanálisis [ASO05].

- **Modelo de Criptografía:** La encriptación esta asociada con la transformación de un mensaje inteligente (M) a una forma no inteligible (C) con la ayuda de una clave secreta (Ke) antes de que sea colocada en un medio inseguro. La encriptación se define como una función matemática (algoritmo) de la siguiente forma:


$$C = E(M, K_e) = E_{K_e}(M)$$

- Donde E puede ser una función computacional.
 - Un algoritmo de encriptación debe tener la propiedad de que el dato original pueda ser recuperado de su forma encriptada si se conoce en valor de su llave.
- **Criptografía de clave pública**

En los cifrados de la clave pública la característica fundamental es que la clave de descifrado no se puede calcular a partir del cifrado. Con un sistema asimétrico cualquier usuario puede enviar un mensaje cifrado usando la clave pública, de este último, pero solo aquellos que conozcan la clave secreta correspondiente pueden descifrar correctamente.

- Se podrá ver mas acerca de la criptografía en el ANEXO D.

2.8 EVALUACIO / PRUEBAS DEL SOFTWARE

La prueba es un proceso de ejecución con la intención de descubrir errores.

Un buen caso de prueba es aquel q tiene una probabilidad muy alta de descubrir un nuevo error.

Una prueba tiene éxito si descubre un error.

Se debe hacer un seguimiento hasta ver si se cumplen los requisitos del cliente.

El principio de Pareto es aplicable a la prueba del software.

- El 80% de los errores está en el 20% de los módulos.
- Hay que identificar esos módulos y probarlos muy bien.

2.8.1 Caso de prueba

Es un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados. Tiene un objetivo concreto (probar algo).

2.8.2 Procedimiento de prueba

Pasos que hay que llevar a cabo para probar uno o varios casos de prueba:
¿Cómo probar el caso de prueba y verificar si ha tenido éxito?

2.8.3 Componente de prueba

Programa que automatiza la ejecución de uno o varios casos de prueba.
Una vez escrito, se puede probar muchas veces (cada vez que hay un cambio en el código de una clase que pueda afectarle).

2.8.4 Diseño de casos de prueba

Definir los casos de prueba que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo.

- Prueba de caja blanca.
Encontrar casos de prueba "viendo" el código interno.

2.8.5 Pruebas de Caja Blanca

Aseguran que la operación interna del programa se ajusta a las especificaciones y que todos los componentes internos se han probado adecuadamente.

- Usa la estructura de control para obtener los casos de prueba.
- Intentan garantizar que todos los caminos de ejecución del programa queden probados.

2.9 EVALUACION DE LA CALIDAD DEL SOFTWARE

La calidad del software se define como: “La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [Pres, 1998].

Funcionalidad

La funcionalidad se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global. [Pres, 1998].

Los puntos de función se computan completando la tabla 2.1 se determinan cinco características de dominios de información. Los valores de dominio de información se definen de la siguiente forma:

Tabla 2.1 Cálculo de Métricas de Puntos Función

Parámetros de medida	de cuenta	Factor de peso			Total
		Simple	Medio	Complejo	
Numero de entradas del usuario	*	3	4	6	
Numero de salidas del usuario	*	4	5	7	
Numero de peticiones	*	3	4	6	

del usuario				
Numero de archivos	*	7	10	15
Numero de interfaces externas	*	5	7	10
Cuenta total				

Fuente: R. Presuman

- **Numero de entradas de usuario**

Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se encuentran de forma separada.

- **Numero de salidas del usuario**

Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. La salida se refiere a informes, pantallas, mensajes de error.

- **Numero de peticiones de usuario**

Una petición se define como una entrada interactiva que produce la generación de alguna respuesta inmediata en forma de salida interactiva.

- **Numero de archivos**

Se cuenta cada archivo maestro lógico.

- **Numero de interfaces externas**

Se cuenta todas las interfaces legibles por la maquina que se utilizan para transmitir a otro sistema.

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

$$PF = \text{cuenta total} \times [0,65 + 1,01 \times \sum(F_i)]$$

Donde cuenta total es la suma de todas las entradas obtenidas de la tabla 2.1, F_i ($i=1$ a 14) son valores de ajuste de la complejidad basadas en las respuestas de la tabla 2.3.

Tabla 2.2 Computación de Puntos de Función

evaluar cada factor en una escala de 0 a 5:					
0	1	2	3	4	5
sin influencia	incidental	Moderado	Medio	Significati	Esencial

Tabla 2.3 Preguntas de Puntos de Función

- 1.-¿Requiere el sistema copias de seguridad y de recuperación fiables?
- 2.-¿Se requiere comunicación de datos?
- 3.-¿Existen funciones de procesamiento distribuido?
- 4.-¿Es crítico el rendimiento?
- 5.-¿Se ejecutan el sistema en un entorno operativo existente y fuertemente utilizado?
- 6.-¿,Requiere el sistema entrada de datos interactiva?
- 7.-¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
- 8.-¿Se actualizan los archivos maestros de forma interactiva?
- 9.-¿Son complejas las entradas, las salidas, los archivos o las peticiones?
- 10.-¿Es complejo el procesamiento interno?
- 11.-¿Se ha diseñado el código para ser reutilizable?
- 12.-¿Están incluidas en el diseño la conversión y la instalación'?
- 13.-¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
- 14.-¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Fuente: R Presuman

Confiabilidad

Al efectuar un análisis de confiabilidad de un sistema, a menudo se congela el tiempo t , de t de tal forma que adecuado un análisis estático. La intención principal es obtener una medida de confiabilidad del sistema, conociendo la

confiabilidad de sus componentes. Para este propósito es adecuado distinguir dos situaciones: en un caso, se considera que el sistema entero falla si falla cualquiera de sus componentes, en otro caso, el sistema falla si y solo si fallan todos sus componentes, mientras que la segunda situación esta modelada por una combinación en paralelo de todos los componentes. El sistema en cada caso, se considera que ha fallado siempre que el flujo señal de entrada a la salida este bloqueado. [Maha,1987].

Para calcular la confiabilidad de un sistema se puede realizar mediante la siguiente función:

$$R(t)=e^{-\lambda t}$$

Donde la función $R(t)$ es la confiabilidad de un determinado modulo, r es el margen de error y t el tiempo que estima la confiabilidad.

Mantenibilidad

El estándar IEEE 982.1-1998 sugiere el índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software basada en los cambios que ocurre con cada versión del producto. Con el IMS se determina la siguiente información:

- MT =Número de módulos en la versión actual.
- F_c =Número de módulos en la versión actual que se han cambiado.
- F_a =Número de módulos en la versión actual que se han añadido.
- F_e =Número de módulos en la versión actual que se han eliminado.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS=[MT-(F_c + F_a + F_e)]/ MT$$

A medida que el IMS se aproxima a 1 el producto se empieza a estabilizar [Pres,2001].

Portabilidad

La portabilidad es la facilidad con la que el software puede ser llevado de un entorno a otro. Este criterio se subdivide: Facilidad de instalación, Facilidad de ajuste, Facilidad de adaptación al cambio [Nuñe,2005].

La portabilidad viene dado por la medida de la subcaracterística de facilidad de instalación, que puede ser medido respondiendo a la pregunta: ¿el usuario o quien mantiene el software puede fácilmente instalar el software en un ambiente operacional?, teniendo la siguiente relación:

$$X=A/B$$

Donde:

A: es el numero de instalaciones exitosas que el usuario realizo.

B: es el numero total de instalaciones que realizo el usuario.

2.10 ESTIMACIÓN DEL COSTO DEL PROYECTO

Existen una serie de métricas propuestas por la Ingeniería del Software para determinare! esfuerzo de un proyecto, el alcance del mismo y la productividad de sus programadores. Vamos a aplicar algunas de las mismas a este desarrollo, para calibrar su dificultad y rendimiento obtenido PRE2000.

2.10.1 Las Métricas Orientadas al Tamaño

La métrica del software es un factor realmente importante en el análisis de un proyecto. Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso por el cual se desarrolla. Se basan en la medición del número de Líneas De Código - LDC - que contiene el desarrollo, entendiendo por línea de código una sentencia del lenguaje de programación (se excluyen comentarios y líneas en blanco de los fuentes).

Tabla 2.4: clasificación de las líneas de código

Categoría	Programadores	Duración	Líneas de código	Ejemplo
Trivial	1	0 – 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 – 6 semanas	1k- 3k	Biblioteca de funciones

Media	2-5	0,5 – 2 semanas	3k - 50k	Compilador de C
Grande	5-20	2 – 3 años	50k- 100k	SO pequeño
Muy grande	100-1000	4 – 5 años	100k- 1M	Grandes SO

Fuente: R Presuman

Una forma de clasificarlos es atendiendo al número de líneas de código, como se muestra en la tabla.

Si en un determinado proyecto en este caso digamos el proyecto SpiderBot en el cual se han generado más de 8.200 líneas de código, esto quiere decir que nos enfrentamos a un proyecto software con una clasificación de complejidad media, para el cual se necesitarían de 2 a 5 programadores trabajando de medio año a 2 años.

2.10.2 El Método Cocomo II

Una metodología que se encarga de medir proyectos software es COCOMO. La metodología COCOMO (COConstructive COst MOdel) se debe a Barry Boehm, y está orientada a líneas de código.

Hay una jerarquía de modelos COCOMO: básico, intermedio y avanzado, la cual se aplica a tres tipos diferentes de software [BWB1990]:

1. **Orgánico;** proyectos relativamente sencillos, menores de 50.000 líneas de código. Se tiene experiencia en proyectos similares y se encuentra en un entorno estable.
2. **Semiacoplado:** proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

3. Empotrado: proyectos bastante complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Dado que sólo se va a emplear una variable para la estimación (la línea de código), se empleará CÓCOMO básico, ya que es un modelo uní variable estático, con lo que se obtiene una **valoración objetiva** del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que posee menos de 50.000 líneas de código.

La ecuación del esfuerzo de COCOMO básico tiene la siguiente forma:

$$E = \text{Esfuerzo} = a \text{ KLDC}^b \text{ (persona x mes)}$$

Donde KLDC es el número de líneas de código, distribuidas en millares, para el proyecto.

La ecuación del tiempo de desarrollo es:

$$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo}^d \text{ (meses)}$$

Por su parte los coeficientes a, b, c y d se obtienen empíricamente del estudio de una serie de proyectos, y sus valores son:

Tabla 2.5 Coeficientes COCOMO

Proyecto de software	A	B	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Fuente:[BWB1990]

En el desarrollo de SpiderBot se han codificado 8,2 miles de líneas de código.

Esfuerzo realizado = $2,4 * 8,2^{1,05} = 21,9$ personas-mes

$T = 2,5 * 21,9^{0,38} = 8,1$ mes [BWB1990]:

N° de personas para desarrollar el proyecto = $E/T = 21,9 / 8,1 = 3$ personas

- La controversia: Líneas de código frente a puntos de función

Existe en el mundo de la Ingeniería del Software una viva polémica sobre qué tipo de métricas son mejores para evaluar un proyecto: las orientadas a tamaño o las que utilizan puntos de función.

El centro de controversia está en considerar las líneas de código como medida clave, ya que los que se oponen a su uso, aducen que las medidas basadas en líneas de código son dependientes del lenguaje de programación.

En cualquier caso esta polémica queda apartada gracias a Casper Jones, quién creó la siguiente Tabla 2.4 de correspondencia entre algunos de los lenguajes de programación más conocidos con su número de equivalencia entre líneas de código por punto de función:

Tabla 2.6 Conversión Líneas de Código a Puntos de Función

Lenguaje	LDC / PF
Ensamblador	320
C	150
Cobol	106
Pascal	91
Basic	64
TCL	64
java	53
PHP	29
C++	29

Fuente: [COT1994]

ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se efectúa el análisis y diseño correspondiente al sistema, dando curso a la aplicación de ingeniería de requerimientos, también se aplican las fases correspondientes a la conceptualización, análisis, diseño del sistema y implementación de acuerdo al proceso de desarrollo, empleando la metodología ASD, utilizando los puntos mencionados en el marco teórico, siguiendo el respectivo plan de desarrollo de software.

3.1 FASE DE ESPECULACIÓN

Esta es la fase inicial del ciclo de vida ASD, que comienza con la Especulación esta consiste en el inicio del proyecto y la planificación de las características del mismo, para este fin sin duda el primer paso es la captura de requisitos, para posteriormente realizar una planificación adecuada del proyecto. Para lo cual se utilizaron herramientas del lenguaje de modelado unificado "UML", estas son los Diagramas de casos de uso, Diagramas de secuencia y los Diagramas de colaboración que se describieron en el capítulo anterior

3.1.1 Análisis y Determinación de Requerimientos

En la oficina del personal de Y.P.F.B. se realizan procesos los cuales se muestran a continuación.

Descripción de procesos

Los procesos básicos que realiza la oficina del personal de Y.P.F.B. son los siguientes:

- Afiliación del personal
- Control de asistencia
- Elaboración de finiquitos
- Control de Vacaciones
- Elaboración de planillas de viáticos
- Subsidios, prenatal, natalidad y lactancia

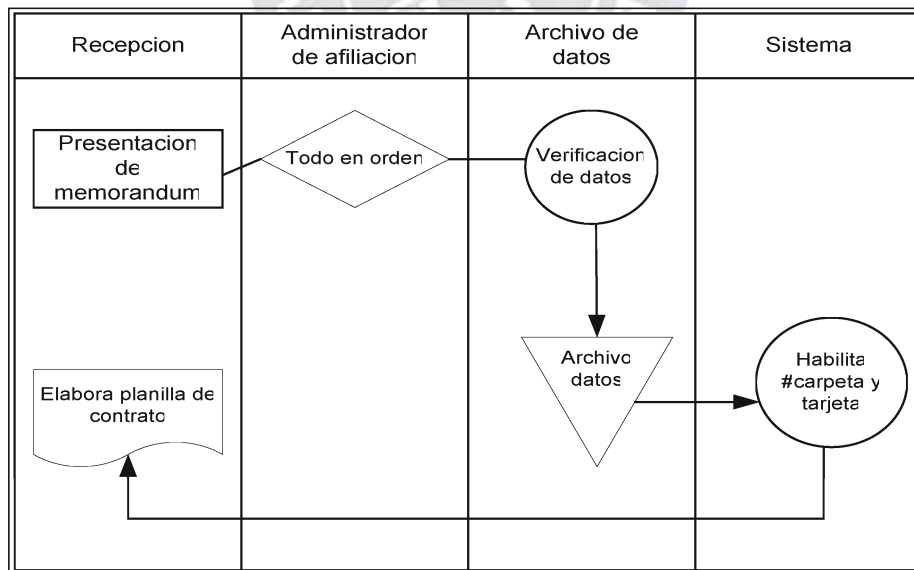
- Planilla de sueldos
- Control de aportes
- Con los datos del empleado se realiza el control de asistencia, atrasos, vacaciones, viáticos por comisión, planilla de subsidios, planilla de sueldos y elaboración de finiquitos en caso de retiro.

Se identifico las siguientes actividades que realizan cada una de los encargados con respecto al funcionamiento del control del personal.

- **AFILIACIÓN DE PERSONAL**

Este es el primer proceso que se realiza y se inicia con la instrucción de memorando emitido por el presidente de la Institución para una nueva contratación, se espera a que se presente el nuevo empleado, si se presenta se le exige la documentación respectiva y se le habilita una tarjeta de asistencia, inmediatamente se elabora un contrato de trabajo de acuerdo a las instrucciones realizadas y se abre un file de personal asignándole un número único de carpeta que le identificara dentro la empresa y que luego es archivada, el encargado de esta operación es el administrador de afiliación.

Figura 3.1 diagrama lógico de procesos de afiliación



Fuente: Elaboración Propia

- **CONTROL DE ASISTENCIA**

Luego de ser incorporado el nuevo empleado se le entrega una tarjeta magnética que sirve para el control de asistencia, esta tarjeta marca la entrada y salida en tres panales de entradas de datos, el empleado registra y marca su entrada o salida este sistema se llama Win Pack y realiza los listados de asistencia del personal, por razones de elaboración de planillas de sueldos, la asistencia es controlada de 21 a 20 de cada mes, el encargado de administrar o controlar la asistencia es el administrador de asistencia.

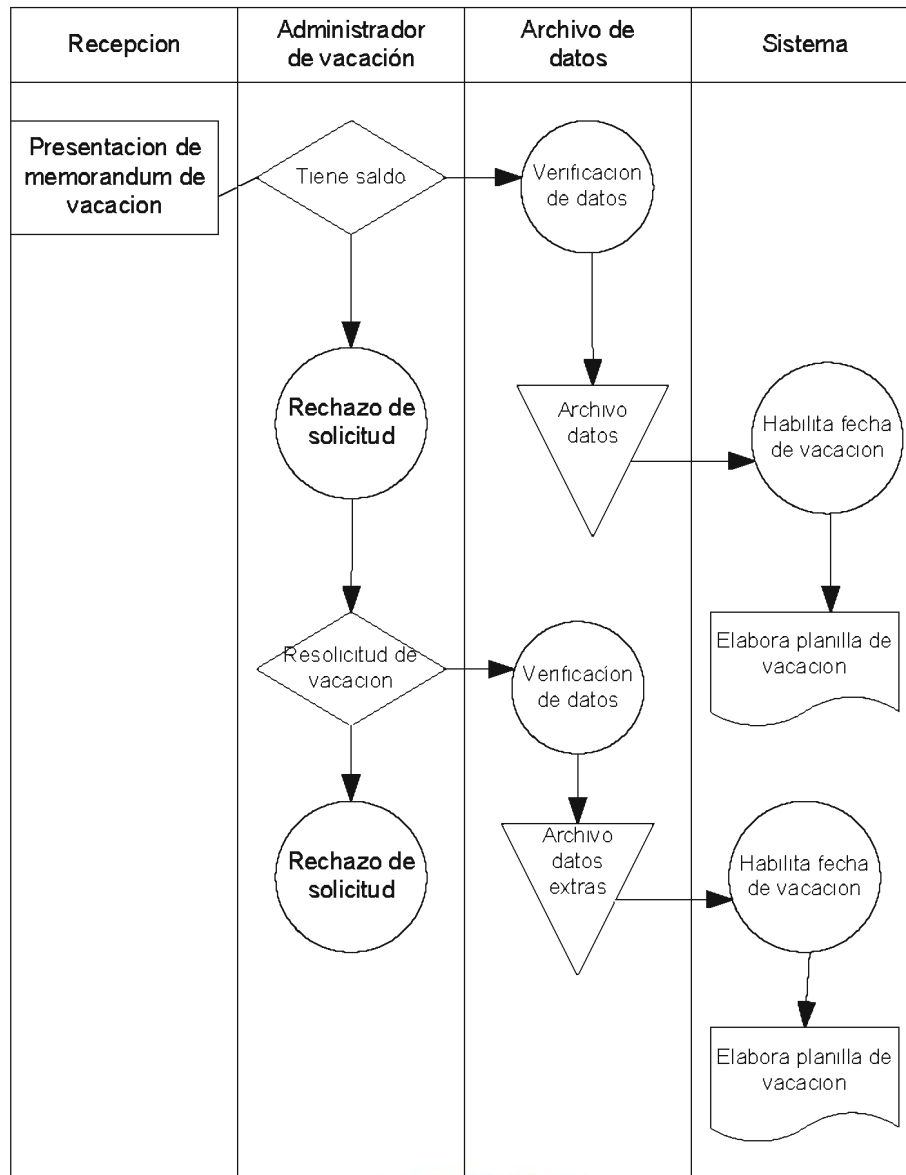
- **CONTROL DE FINIQUITO**

La Presidencia ejecutiva realiza el memorando dirigido al vicepresidente de negociaciones Internacionales y Contratos donde instruye la rescisión de contrato de un empleado, luego la vicepresidencia deriva la instrucción a la administración para que realice la carta de retiro, una vez que se realiza la carta se entrega a la Vicepresidencia para que la firmen y luego su oficina haga la entrega respectiva, el encargado de esta operación es el administrador de finiquitos.

- **CONTROL DE VACACIÓN**

En este proceso el empleado solicita vacaciones a la oficina del personal, y si tiene saldo del lapso anterior se llena la boleta de vacación con los datos necesarios, luego se le entrega al empleado para que pueda ser firmado y autorizado por su jefe de unidad, en caso de no tener saldo de días de vacación la oficina de personal realiza un informe a la administración para que en coordinación con el jefe de unidad del empleado solicitante determinen si corresponden otorgar la vacación adelantada, en caso de ser aprobado la oficina de personal llena la boleta, el encargado de esta operación es el administrador de vacación.

Figura 3.2 Diagrama lógico de procesos de Vacación



Fuente: Elaboración propia

- VIÁTICOS**

los viajes a realizar son instruidos por los ejecutivos de la empresa que en este caso se le denomina autoridad conferente puede ser el presidente, vicepresidente, gerentes, directores, esta autoridad designa a un empleado que viaje a realizar trabajos que se requieran para la empresa, primeramente hacen una comunicación interna donde se adjunta la hoja de comisión y en

esta indica la fecha de viaje , el tiempo y el objetivo del viaje, una vez que se registra esta instrucción se realiza la reserva de los pasajes y el calculo de viáticos que corresponde de acuerdo a la escala vigente en la empresa, una vez realizado el calculo de viáticos se deriva esta planilla a la coordinación financiera quienes hacen efectivo el pago al empleado, el encargado de este proceso es el administrador de viáticos.

- **PLANILLA DE SUELDOS**

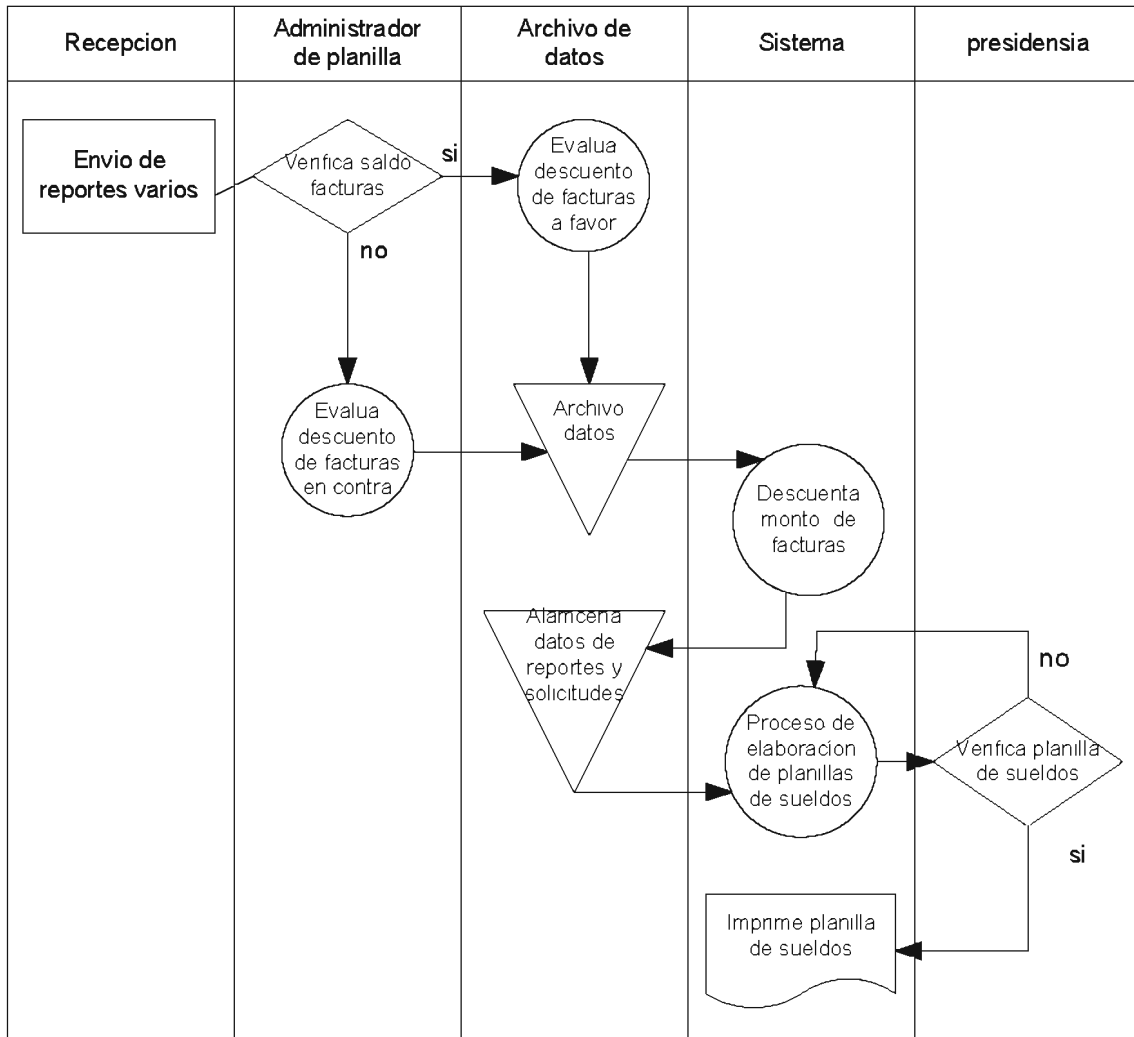
La planilla de sueldos se elabora con los reportes del control de asistencia, personal con descargo de facturas IVA, viáticos pagados y otros, esta información es enviada a la ciudad de Santa Cruz vía fax o correo electrónico quienes hacen el vaciado de los datos en el sistema de planillas.

Luego de hacer el proceso de elaboración de planillas envían la prueba para su respectiva revisión de los datos posteriormente se envía nuevamente con las correcciones o el visto bueno, el encargado de esta operación es el administrador de planillas.

Control de facturas IVA

Todo el personal que tiene facturas para el descargo de impuestos tiene hasta el 15 de cada mes para hacer llegar el formulario RC-87 llenado y con sus respectivas facturas para que se haga una revisión, una vez realizado esta tarea se hace un listado del personal que presento descargo y se informa al encargado de planilla de sueldos, el encargado de esta operación es el administrador de planillas.

Figura 3.3 Diagrama lógico de procesos de pago de Sueldos



Fuente: elaboración propia

3.1.2 CAPTURA DE REQUISITOS

La captura de requisitos es el proceso de averiguar normalmente en circunstancias difíciles lo que se debe construir. Los desarrolladores y clientes deben acordar qué es lo que el sistema debe hacer. Para identificar los requisitos funcionales del sistema utilizaremos los casos de uso, que se describen como conjuntos de secuencias.

Cada una de estas secuencias refleja la interacción entre los elementos externos al sistema y el propio sistema (se trata de la descripción de escenarios o situaciones posibles donde se pone de relieve el comportamiento del sistema ante su uso por parte del usuario). Para la óptima captura de requisitos, el proyecto se afianzará del modelado de casos de uso. Así pues, los objetivos principales de la realización de casos de uso son:

Definir el límite entre el sistema a desarrollar y los elementos externos a ese sistema (actores usuarios del sistema).

Capturar el conjunto de funcionalidades y comportamientos del sistema a desarrollar.

Cada caso de uso se documenta mediante una representación gráfica y un texto con la descripción de las situaciones o escenarios ante los que el usuario se pueda encontrar en su interacción con el sistema.

3.1.3 MODELO DE CASOS DE USO

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones.

A continuación se describen las características de los actores identificados en el manejo e implementación del sistema.

ACTORES



DESCRIPCIÓN

Administrador de afiliación. Este usuario será el encargado del manejo del modulo de afiliación tendrá acceso directo ala base de datos del servidor, realizara bajas altas y modificación del personal de la empresa estará a cargo de registrar todos los datos del personal y su historial de trabaja para su record de servicio, su antigüedad etc.



Administrador de Asistencia. Este usuario será el encargado del control de la asistencia del personal de la empresa, realizara reportes de asistencia, reportes de horas extras.



Administrador de Vacación. Este usuario será el encargado de emitir papeletas de solicitud de vacación, realizara el control de lapsos cumplidos y duodécimas, emisión de boletas de lapsos.



Administrador de Viáticos. Este usuario se encargara de realizar la emisión de la planilla de viáticos realizados de cada personal, realiza la verificación de las reliquidación ala conclusión del viático.



Administrador de Planillas. Este usuario es el encargado de realizar planillas de sueldos, realizar reportes de planillas del iva, planillas de AFP`s, recibe la información de las planillas de horas extras, asistencia, viáticos, para realización de los respectivos descuentos.



Funcionario. Es un usuario cualquiera de la empresa al cual se le realizan procesos de asignación de trabajos.

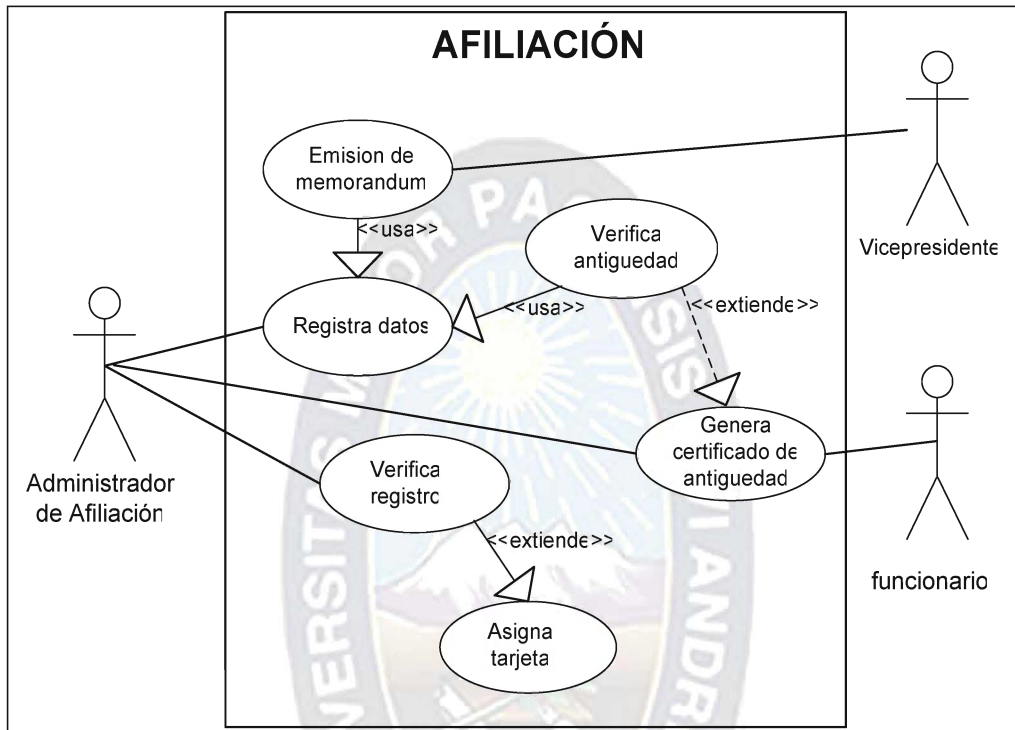


Vicepresidente. Este usuario es el encargado de aprobar los memorandums de afiliaciones, viáticos, vacaciones para el personal de la empresa.

3.1.4 DIAGRAMA DE CASOS DE USO

Un Diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema. A continuación se muestran los casos de uso principales agrupados por paquetes y la descripción narrativa de cada caso de uso.

Figura 3.4 Diagrama de Casos de Uso del modulo de afiliación



Fuente: Elaboración Propia

Nombre	REGISTRA DATOS	
Actores	Él actor que interviene en este caso es: administrador de afiliación.	
Propósito	Registra datos de afiliación del personal.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar datos deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de afiliación entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de	4 El Sistema autentifica código y clave y las casillas para

	afiliación ingresa código.	introducir información
	5. El encargado de afiliación ingresa los datos.	6. El sistema registra los datos de acuerdo al tipo de ingreso que se desee.
Precondición	El sistema tiene almacenada las nominas del personal de la empresa que hayan sido retirados o sigan trabajando.	
Poscondición	El sistema almacena las modificaciones o nueva información del personal.	
Nombre	VERIFICA REGISTRO	
Actores	Él actor que interviene en este caso es: administrador de afiliación.	
Propósito	Verifica si el nuevo personal que se esta dando de alta ya estuvo antes en la empresa para solo actualizar los datos.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de verificar el registro del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de afiliación entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de afiliación ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de afiliación realiza la verificación del personal.	6. El sistema verifica si el personal a afiliar es nuevo o ya trabajo anteriormente.
Precondición	El sistema tiene almacenada las nominas del personal de la empresa que hayan sido retirados o sigan trabajando.	
Poscondición	Ninguno	

Nombre	GENERA CERTIFICADO DE ANTIGUEDAD	
Actores	Él actor que interviene en este caso es: Funcionario, administrador de afiliación.	
Propósito	Emite o genera certificado de antigüedad del personal de acuerdo ala experiencia de trabajo u currículo.	
Resumen	El sistema deberá comportase como se describe en el presente caso de uso cuando reciba la solicitud de emitir certificado de antigüedad del personal deberá de realizarlo de acuerdo con la orden que se le de.	
Flujo principal	Evento Actor	Evento Sistema
	<ol style="list-style-type: none"> 1. El encargado de afiliación entra al sistema. 3. El encargado de afiliación ingresa código. 5. El encargado de afiliación realiza la emisión de certificado de antigüedad. 7. El encargado de afiliación entrega el respectivo certificado al funcionario 	<ol style="list-style-type: none"> 2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información 6. El sistema verifica la solicitud y emite el respectivo certificado de acuerdo a datos almacenados existentes.
Precondición	El sistema tiene almacenada las nominas del personal de la empresa.	
Poscondición	Ninguno	

Nombre	EMISION DE MEMORANDUM	
Actores	Él actor que interviene en este caso es: vicepresidente	
Propósito	Emite o genera certificado de memorando del personal de acuerdo la disponibilidad.	
Resumen	El sistema deberá comportase como se describe en el	

presente caso de uso cuando reciba la solicitud de emitir memorando del personal deberá de realizarlo de acuerdo con la orden que se le de.

	Evento Actor	Evento Sistema
Flujo principal	<ol style="list-style-type: none">1. El vicepresidente entra al sistema.3. El vicepresidente ingresa código.5. El vicepresidente realiza la emisión de memorando.	<ol style="list-style-type: none">2. El sistema le solicita código y clave.4 El Sistema autentifica código y clave y las casillas para introducir información6. El sistema verifica la solicitud y emite el respectivo memorando de acuerdo a datos almacenados existentes.
Precondición	El sistema tiene almacenada las nominas del personal de la empresa y modelos de memorandos.	
Poscondición	Ninguno	

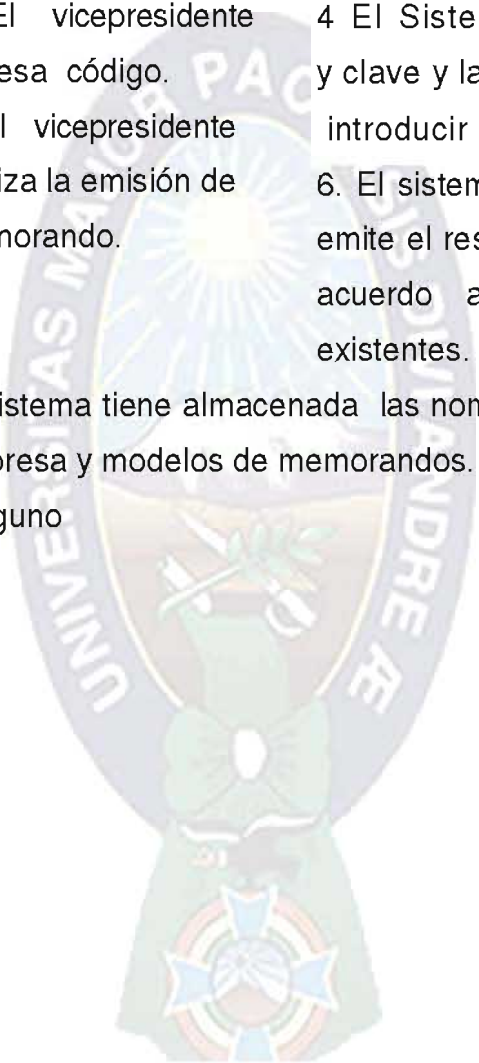
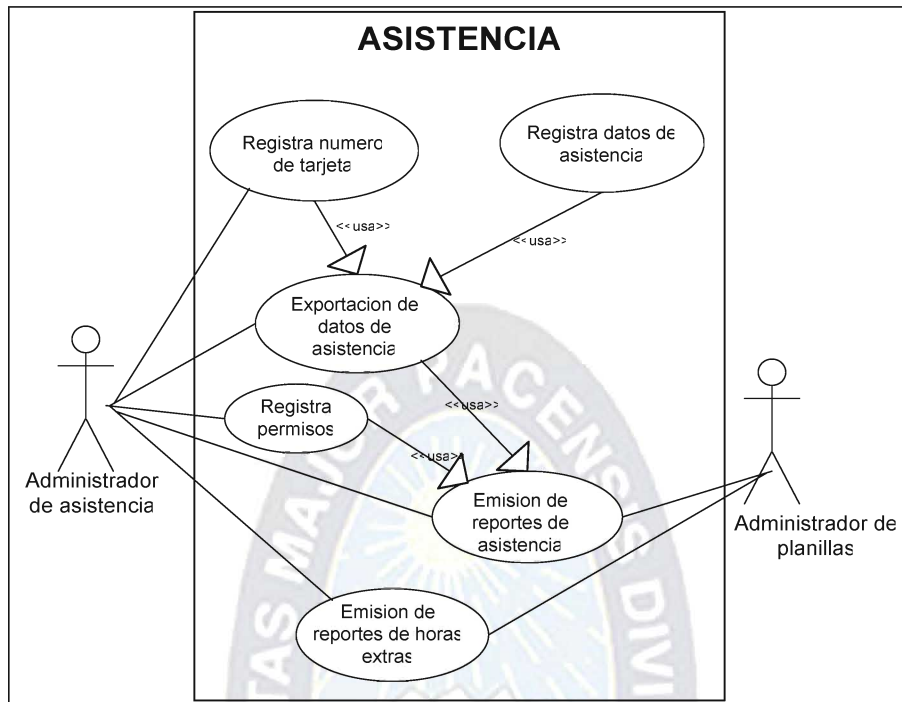


Figura 3.5 Diagrama de Casos de Uso del modulo de asistencia



Fuente: Elaboración Propia

Nombre	REGISTRA NUMERO DE TARJETA	
Actores	Él actor que interviene en este caso es: administrador de asistencia.	
Propósito	Registra el nuevo número de tarjeta del nuevo personal de la empresa.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar el numero de tarjeta del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de asistencia entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de afiliación ingresa	4 El Sistema autentifica código y clave y las casillas para introducir información.

	código.	6. El sistema verifica si el personal a afiliar es nuevo o ya trabajo anteriormente.
	5. El encargado de asistencia realiza el registro del número de tarjeta y datos del personal.	
Precondición	El sistema tiene almacenada las nominas del personal de la empresa.	
Poscondición	Ninguno	

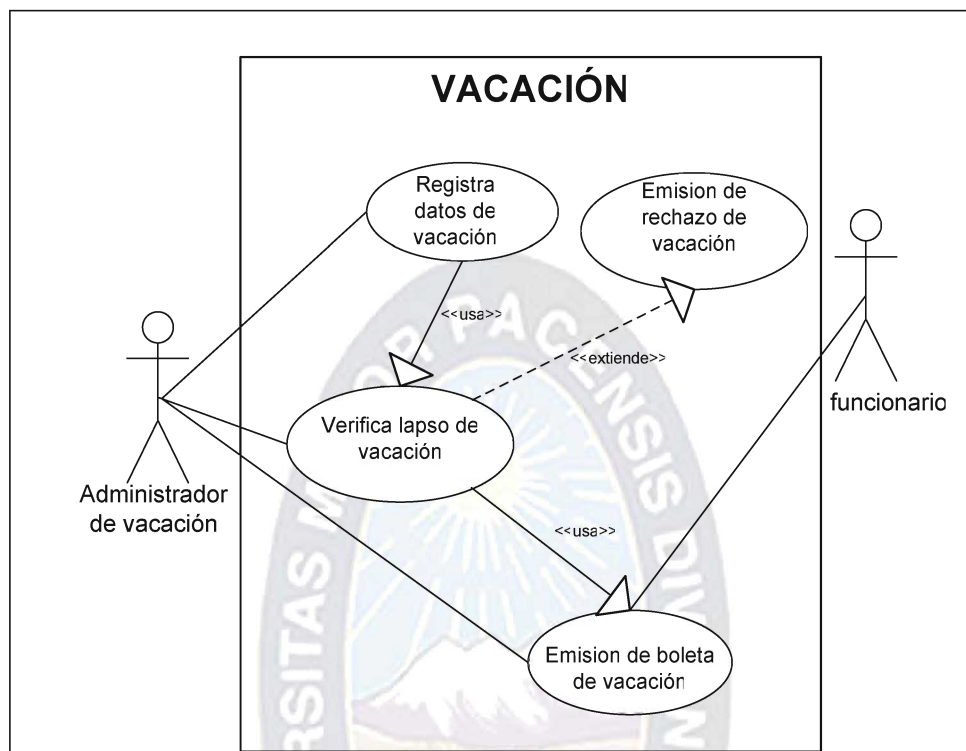
Nombre	REGISTRA PERMISOS	
Actores	Él actor que interviene en este caso es: administrador de asistencia.	
Propósito	Registra los permisos existentes en la empresa ya sea permisos o comisiones.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar permisos del personal deberá de realizarlo de acuerdo con la orden que se le de.	
Flujo principal	Evento Actor	Evento Sistema
	<p>1. El encargado de asistencia entra al sistema.</p> <p>3. El encargado de afiliación ingresa código.</p> <p>5. El encargado de asistencia ingresa los datos del permiso solicitado.</p>	<p>2. El sistema le solicita código y clave.</p> <p>4 El Sistema autentifica código y clave y las casillas para introducir información.</p> <p>6. El sistema registra y almacena los permisos solicitados</p>
Precondición	Ninguno.	
Poscondición	Ninguno	

Nombre	EMISION DE REPORTES DE ASISTENCIA	
Actores	Él actor que interviene en este caso es: administrador de asistencia, administrador de planilla.	
Propósito	Emite los reportes finales de asistencia.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de emitir reportes de asistencia deberá de realizarlo de acuerdo con la orden que se le de.	
Flujo principal	Evento Actor	Evento Sistema
	<ol style="list-style-type: none"> 1. El encargado de asistencia entra al sistema. 3. El encargado de afiliación ingresa código. 5. El encargado de asistencia elige opción de reporte deseado. 	<ol style="list-style-type: none"> 2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema realiza la opción deseada y emite el respectivo reporte.
Precondición	El sistema tiene almacenado un modelo de reporte final de asistencia y solo solicita datos a llenar.	
Poscondición	Ninguno	

Nombre	EMISION DE REPORTES DE HORAS EXTRAS	
Actores	Él actor que interviene en este caso es: administrador de asistencia, administrador de planilla.	
Propósito	Registra los permisos existentes en la empresa ya sea permisos o comisiones.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar permisos del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema

Flujo principal	<p>1. El encargado de asistencia entra al sistema.</p> <p>3. El encargado de afiliación ingresa código.</p> <p>5. El encargado de asistencia elige opción de reporte deseado.</p>	<p>2. El sistema le solicita código y clave.</p> <p>4 El Sistema autentifica código y clave y las casillas para introducir información.</p> <p>6. El sistema realiza la opción deseada y emite el respectivo reporte.</p>
Precondición	El sistema tiene almacenado las horas extras de cada personal de la empresa.	
Poscondición	Ninguno	
Nombre	EXPORTACION DE DATOS DE ASISTENCIA	
Actores	Él actor que interviene en este caso es: administrador de asistencia.	
Propósito	Exporta los datos reunidos de asistencia del personal.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de exportar datos deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	<p>1. El encargado de asistencia entra al sistema.</p> <p>3. El encargado de afiliación ingresa código.</p> <p>5. El encargado de asistencia elige la opción de exportación de datos</p>	<p>2. El sistema le solicita código y clave.</p> <p>4 El Sistema autentifica código y clave y las casillas para introducir información.</p> <p>6. El sistema realiza la opción pedida y exporta los datos solicitados</p>
Precondición	Ninguno.	
Poscondición	Ninguno	

Figura 3.6 Diagrama de Casos de Uso del modulo de vacaciones



Fuente: Elaboración Propia

Nombre	REGISTRA DATOS DE VACACIÓN	
Actores	Él actor que interviene en este caso es: Administrador de vacación.	
Propósito	Registra los datos de la vacación solicitada por el personal.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar permisos del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de vacación entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de vacación ingresa código.	4 El Sistema autentifica código y clave y las casillas para

	5. El encargado de introducir información.	
	vacación introduce los datos.	6. El sistema registra los datos de la vacación solicitada.
Precondición	El sistema tiene almacenado los datos de años de servicio de cada empleado.	
Poscondición	Ninguno	

Nombre	VERIFICA LAPSO DE VACACIÓN	
Actores	Él actor que interviene en este caso es: Administrador de vacación.	
Propósito	Verifica el lapso de vacaciones que tiene pendiente cada personal de la empresa.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de la verificación de lapsos pendiente de vacación del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de vacación entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de vacación ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de vacación solicita la verificación de lapso pendiente.	6. El sistema verifica el lapso pendiente de vacación solicitado.
Precondición	El sistema tiene almacenado los datos de años de servicio de cada empleado y lapsos de vacaciones pendientes.	
Poscondición	Ninguno	

Nombre	EMISION DE BOLETA DE VACACIÓN	
Actores	Él actor que interviene en este caso es: Administrador de vacación, funcionario.	
Propósito	Emite boleta de vacación solicitada	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de emitir boleta de vacación del personal deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de vacación entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de vacación ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de vacación solicita la emisión de la boleta de vacación.	6. El sistema emite la boleta de vacación solicitada.
Precondición	El sistema tiene registrado u almacenado los datos de vacación solicitada.	
Poscondición	Ninguno	

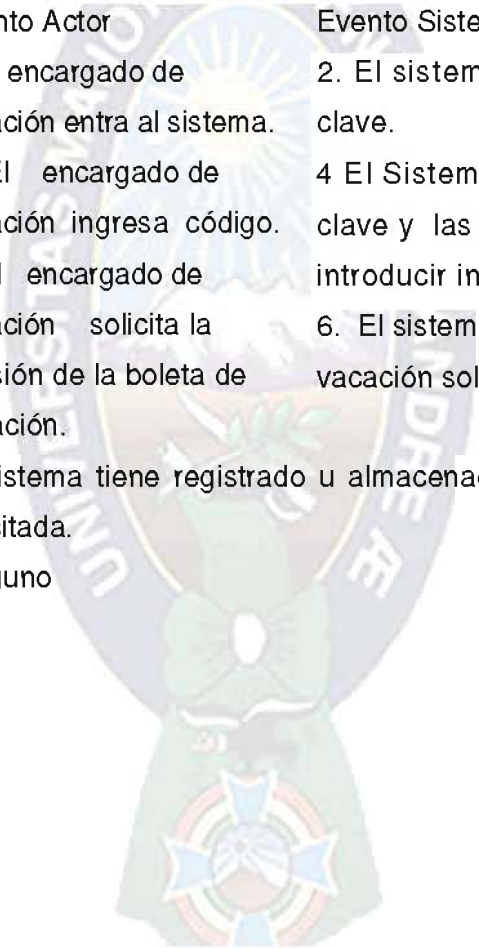
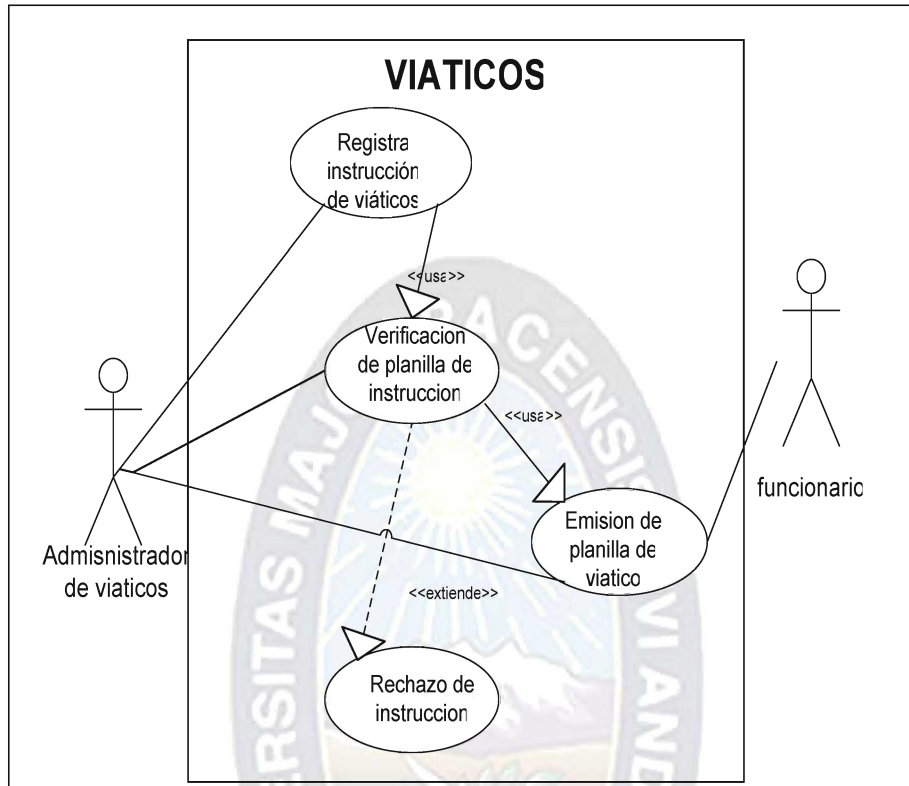


Figura 3.7 Diagrama de Casos de Uso del modulo de viáticos



Fuente: Elaboración Propia

Nombre	REGISTRA INSTRUCCIÓN DE VIÁTICO	
Actores	Él actor que interviene en este caso es: Administrador de viático.	
Propósito	Registra instrucción de viático solicitado.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registro de viático deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de viático entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de viático ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de viático introduce los	6. El sistema registra la instrucción

	datos.	del viático.
Precondición	Ninguno.	
Poscondición	Ninguno	

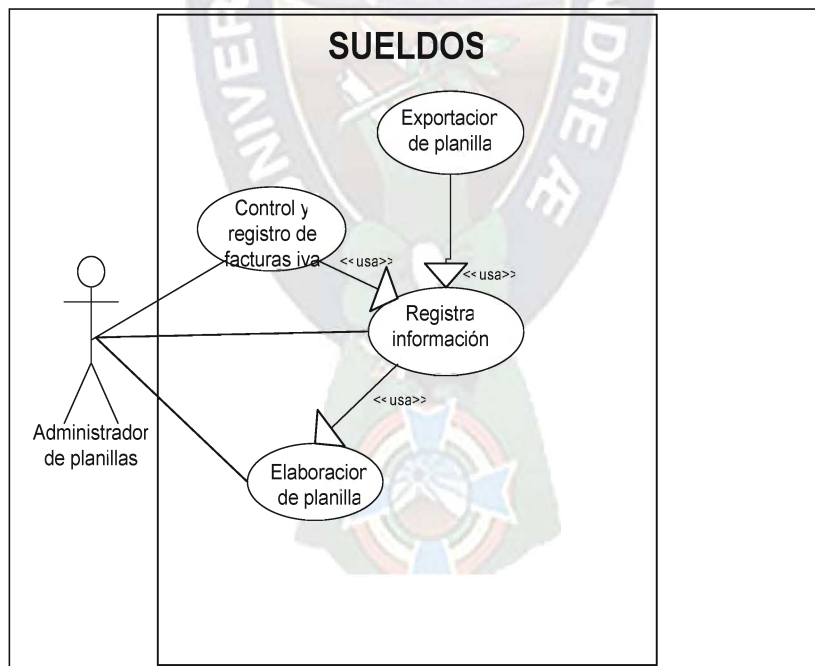
Nombre	VERIFICACION DE PLANILLA DE INSTRUCCIÓN	
Actores	Él actor que interviene en este caso es: Administrador de viático.	
Propósito	Verifica que este en orden la planilla de instrucción.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de verificar la planilla de instrucción deberá de realizarlo de acuerdo con la orden que se le de.	
Flujo principal	Evento Actor	Evento Sistema
	<ol style="list-style-type: none"> 1. El encargado de viático entra al sistema. 3. El encargado de viático ingresa código. 5. El encargado de viático elige la opción de verificación. 	<ol style="list-style-type: none"> 2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema verifica la planilla de instrucción.
Precondición	El sistema tiene almacenado todos los datos de la planilla de instrucción.	
Poscondición	Ninguno	

Nombre	EMISION DE PLANILLA DE VIÁTICO	
Actores	Él actor que interviene en este caso es: Administrador de viático, funcionario.	
Propósito	Emite la planilla de viático solicitado.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de emisión de planilla de viático deberá de realizarlo de	

acuerdo con la orden que se le de.

	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de viático entra al sistema. 3. El encargado de viático ingresa código. 5. El encargado de viático elige la opción emitir planilla.	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema realiza la opción y emite la respectiva planilla de viáticos.
Precondición	El sistema tiene almacenado los datos de la planilla de viático.	
Poscondición	Ninguno	

Figura 3.8 Diagrama de Casos de Uso del modulo de planillas de sueldos



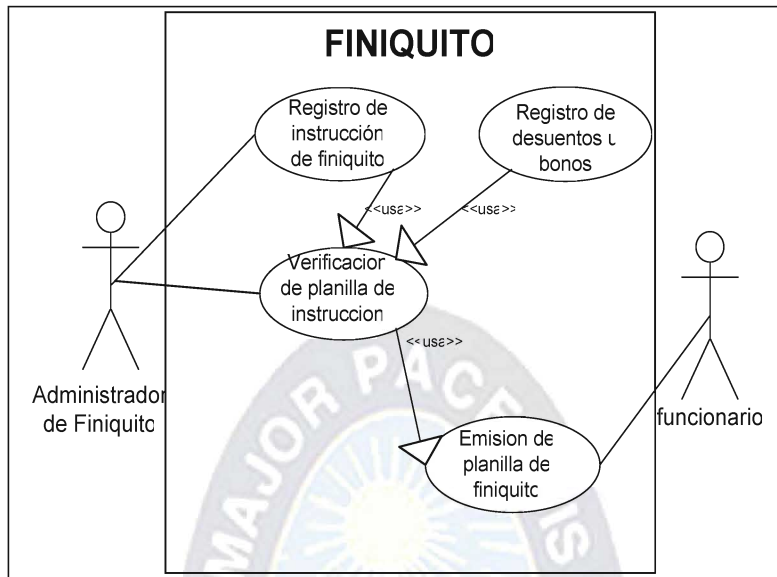
Fuente: Elaboración Propia

Nombre	CONTROL Y REGISTRO DE FACTURAS IVA	
Actores	Él actor que interviene en este caso es: Administrador de planillas.	
Propósito	Controla y registrar las facturas IVA.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registro de facturas deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de planilla entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de planilla ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de planilla introduce datos.	6. El sistema registra los datos introducidos.
Precondición	Ninguno.	
Poscondición	El sistema almacena la información registrada.	

Nombre	ELABORACIÓN DE PLANILLA	
Actores	Él actor que interviene en este caso es: Administrador de planillas.	
Propósito	Elabora la planilla de sueldos.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de elaboración de planilla deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de planilla entra al sistema.	2. El sistema le solicita código y clave.
	3. El encargado de planilla ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.
	5. El encargado de	

	planilla elige la opción 6. El sistema realiza la opción de elaboración de planilla. elegida.								
Precondición	El sistema tiene almacenado los datos para la emisión de planilla de sueldos.								
Poscondición	Ninguno.								
Nombre	REGISTRA INFORMACIÓN								
Actores	Él actor que interviene en este caso es: Administrador de planillas.								
Propósito	Registrar la información adquirida.								
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registrar deberá de realizarlo de acuerdo con la orden que se le de.								
Flujo principal	<table border="0"> <thead> <tr> <th>Evento Actor</th> <th>Evento Sistema</th> </tr> </thead> <tbody> <tr> <td>1. El encargado de planilla entra al sistema.</td> <td>2. El sistema le solicita código y clave.</td> </tr> <tr> <td>3. El encargado de planilla ingresa código.</td> <td>4 El Sistema autentifica código y clave y las casillas para introducir información.</td> </tr> <tr> <td>5. El encargado de planilla introduce datos.</td> <td>6. El sistema registra los datos introducidos.</td> </tr> </tbody> </table>	Evento Actor	Evento Sistema	1. El encargado de planilla entra al sistema.	2. El sistema le solicita código y clave.	3. El encargado de planilla ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.	5. El encargado de planilla introduce datos.	6. El sistema registra los datos introducidos.
Evento Actor	Evento Sistema								
1. El encargado de planilla entra al sistema.	2. El sistema le solicita código y clave.								
3. El encargado de planilla ingresa código.	4 El Sistema autentifica código y clave y las casillas para introducir información.								
5. El encargado de planilla introduce datos.	6. El sistema registra los datos introducidos.								
Precondición	Ninguno.								
Poscondición	El sistema almacena la información registrada.								

Figura 3.9 Diagrama de Casos de Uso del modulo de finiquitos



Fuente: Elaboración Propia

Nombre	REGISTRA INSTRUCCIÓN DE VIÁTICO	
Actores	Él actor que interviene en este caso es: Administrador de finiquito.	
Propósito	Registra instrucción de viático finiquito.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de registro de finiquito deberá de realizarlo de acuerdo con la orden que se le de.	
Flujo principal	<p>Evento Actor</p> <ol style="list-style-type: none"> 1. El encargado de finiquito entra al sistema. 3. El encargado de finiquito ingresa código. 5. El encargado de finiquito introduce los datos. 	<p>Evento Sistema</p> <ol style="list-style-type: none"> 2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema registra la instrucción del finiquito.
Precondición	Ninguno.	
Poscondición	Ninguno	

Nombre	VERIFICACION DE PLANILLA DE INSTRUCCIÓN	
Actores	Él actor que interviene en este caso es: Administrador de finiquito.	
Propósito	Verifica que este en orden la planilla de instrucción.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de verificar la planilla de instrucción deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de finiquito entra al sistema. 3. El encargado de finiquito ingresa código. 5. El encargado de finiquito elige la opción de verificación.	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema verifica la planilla de instrucción.
Precondición	El sistema tiene almacenado todos los datos de la planilla de instrucción.	
Poscondición	Ninguno	

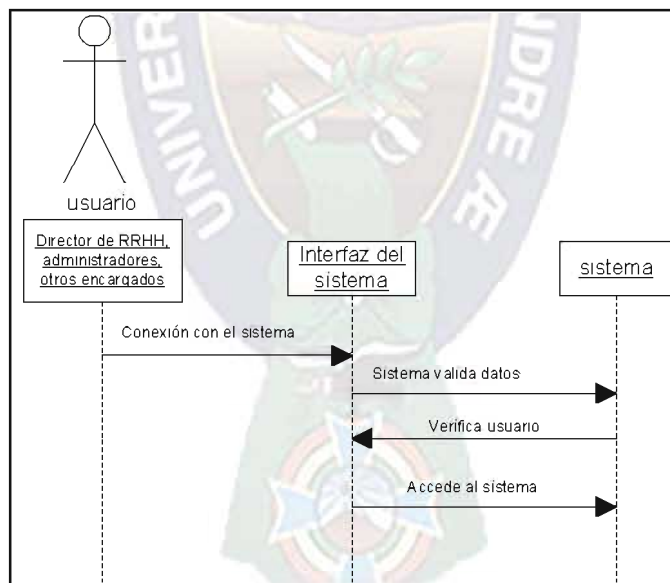
Nombre	EMISION DE PLANILLA DE FINIQUITO	
Actores	Él actor que interviene en este caso es: Administrador de finiquito, funcionario.	
Propósito	Emite la planilla de finiquito solicitado.	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba la solicitud de emisión de planilla de finiquito deberá de realizarlo de acuerdo con la orden que se le de.	
	Evento Actor	Evento Sistema
Flujo principal	1. El encargado de finiquito entra al sistema. 3. El encargado de finiquito ingresa código. 5. El encargado de finiquito elige la opción emitir planilla.	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información. 6. El sistema realiza la opción y emite la respectiva planilla de finiquito.
Precondición	El sistema tiene almacenado los datos de la planilla de finiquito.	

3.1.5 DIAGRAMAS DE SECUENCIA Y COLABORACIÓN

Los Diagramas de secuencia muestran de manera gráfica los eventos u operaciones del sistema, como es que este responde a alguna determinada operación y con la aplicación de los Diagramas de colaboración se tiene un enfoque más claro de la relación entre las operaciones, lo cual nos ayudara en la etapa de implementación.

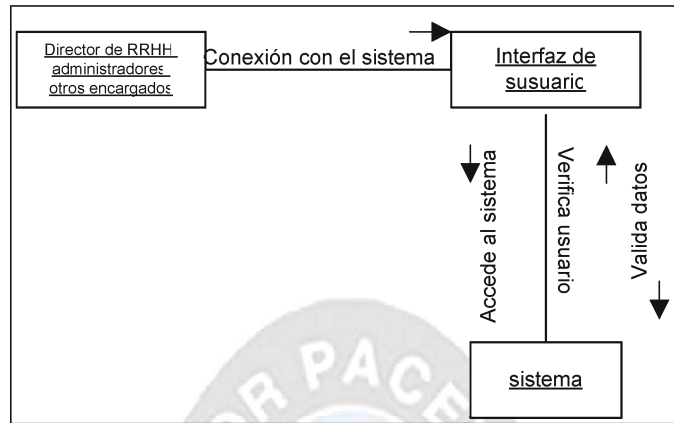
Con lo desarrollado hasta ahora se tiene una perspectiva como para poder elaborar los Diagramas de secuencia y sus respectivos Diagramas de colaboración, a continuación se detallarán las siguientes figuras más importantes: Es importante el control de acceso del usuario como muestra en la figura 3.10 y figura 3.11.

Figura 3.10 Secuencia Correspondiente al Control de Acceso del Usuario



Fuente: Elaboración Propia

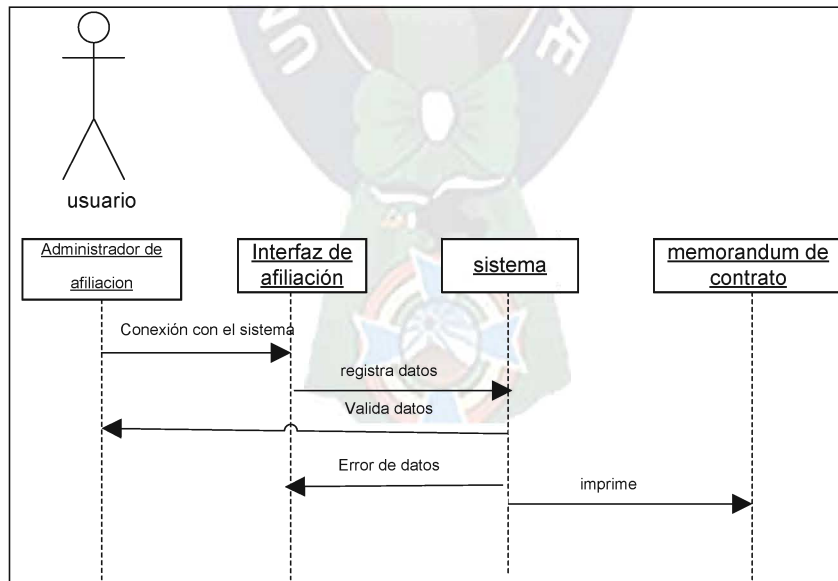
Figura 3.11 colaboración Correspondiente al Control de Acceso del Usuario



Fuente: Elaboración Propia

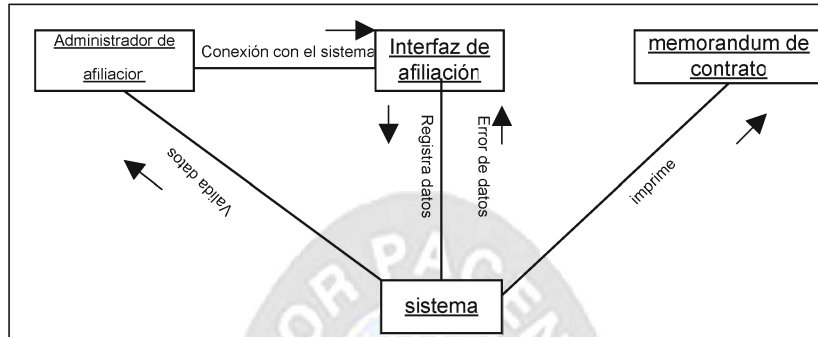
La secuencia correspondiente a la afiliación de nuevo personal se inicia cuando el vicepresidente ejecutivo de Y.P.F.B. manda un memorando de un nuevo funcionario para la empresa y el encargado de afiliación es el encargado de registrar los datos como se vera en la figura 3.12.

Figura 3.12 secuencia Correspondiente al Control de afiliación de nuevo funcionario



Fuente: Elaboración Propia

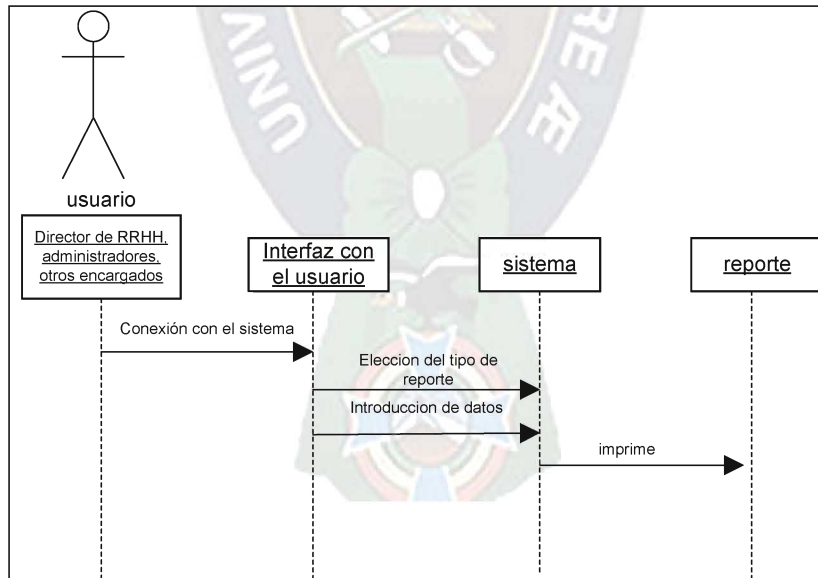
Figura 3.13 colaboración Correspondiente al Control de afiliación de nuevo funcionario



Fuente: Elaboración Propia

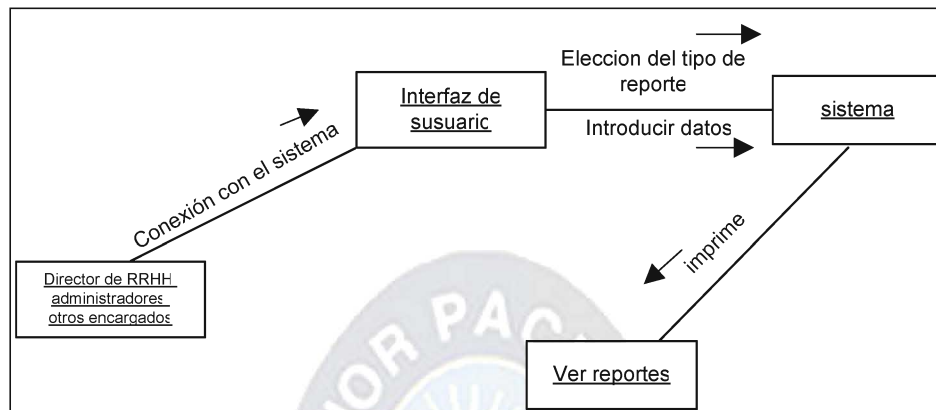
Luego pasado la etapa de afiliación el sistema podrá dar paso al control de asistencia del nuevo personal y la emisión de reporte de asistencia.

Figura 3.14 secuencia Correspondiente a la emisión de reportes



Fuente: Elaboración Propia

Figura 3.15 colaboración Correspondiente ala emisión de reportes



Fuente: Elaboración Propia

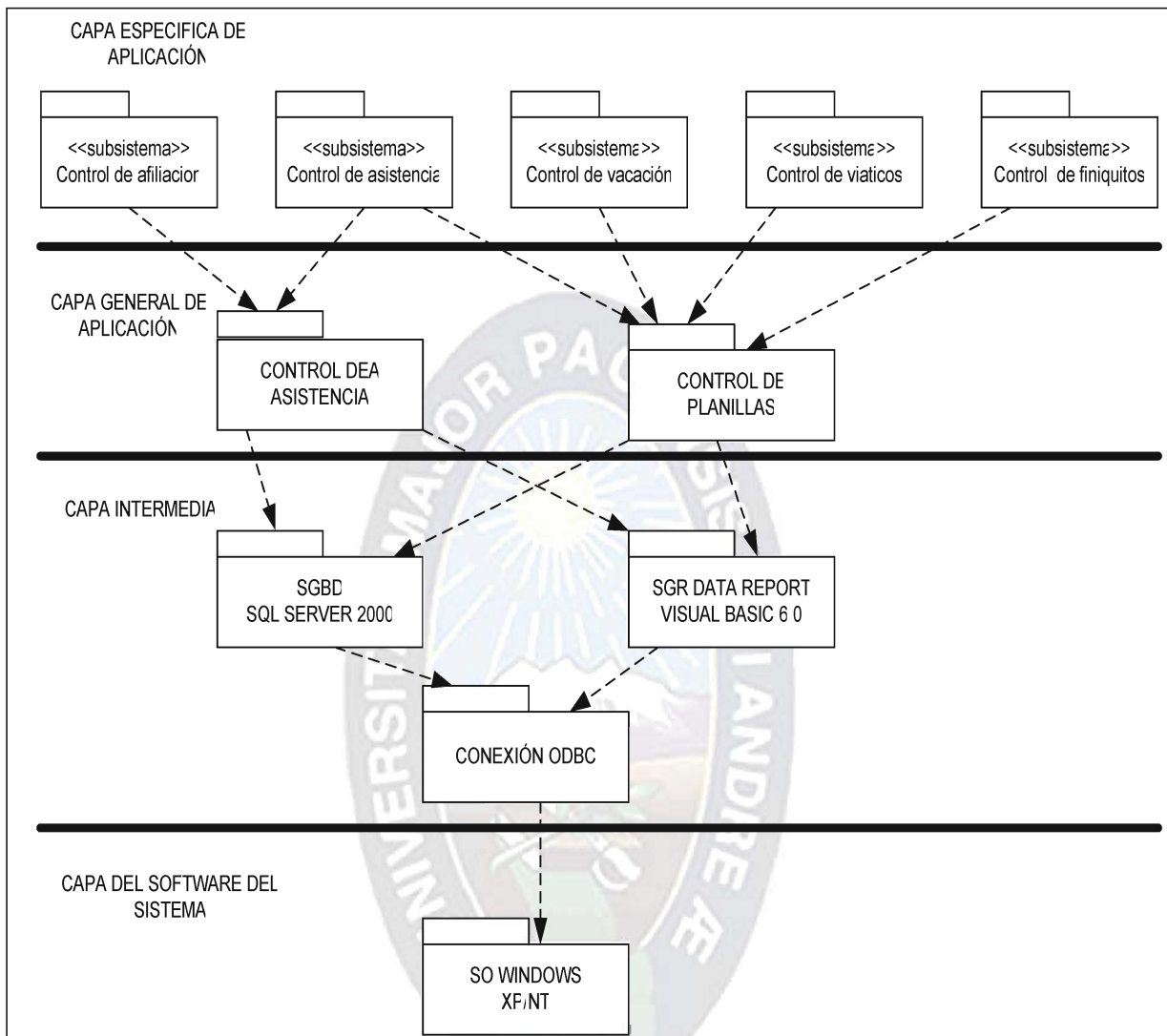
3.2 FASE DE COLABORACIÓN

Esta es la segunda fase del ciclo de vida ASD, se desarrollan las características del sistema, pero luego de haber realizado varias iteraciones que se vera mas adelante en el Anexo B, el sistema finalmente presenta el siguiente esquema en cuanto a su estructura de funcionamiento, el cual también se halla representado por herramientas del lenguaje de modelado unificado UML apoyándose en el Diagrama de clases y también se utilizo los Diagramas de E-R para modelar la estructura de la base de datos del sistema.

3.2.1 DIAGRAMA DE PAQUETES

Este diagrama nos permite ver el sistema en sus subsistemas.

Figura 3.16 diagrama de paquetes

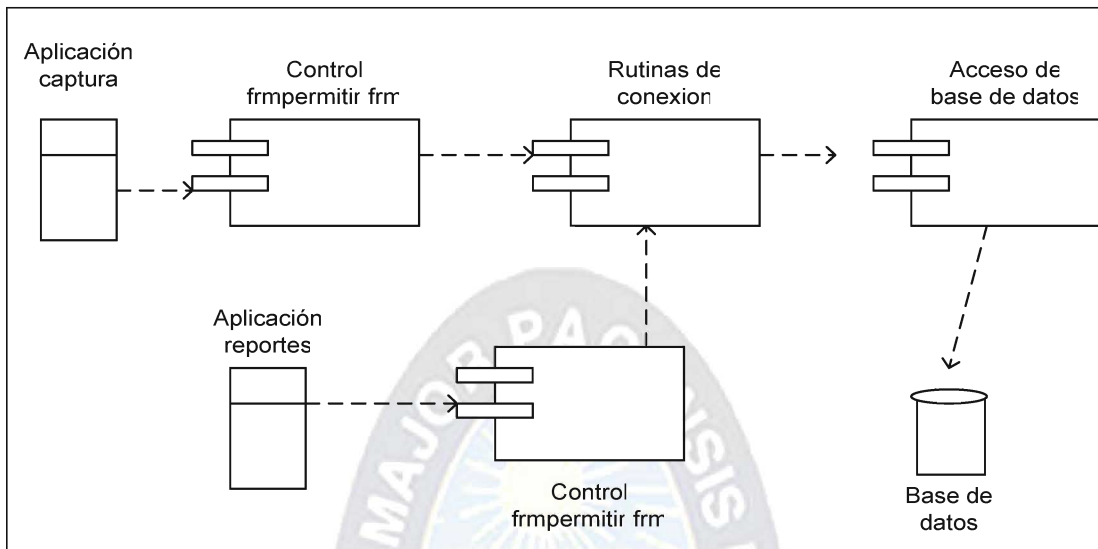


Fuente: Elaboración Propia

3.2.2 DIAGRAMA DE COMPONENTES

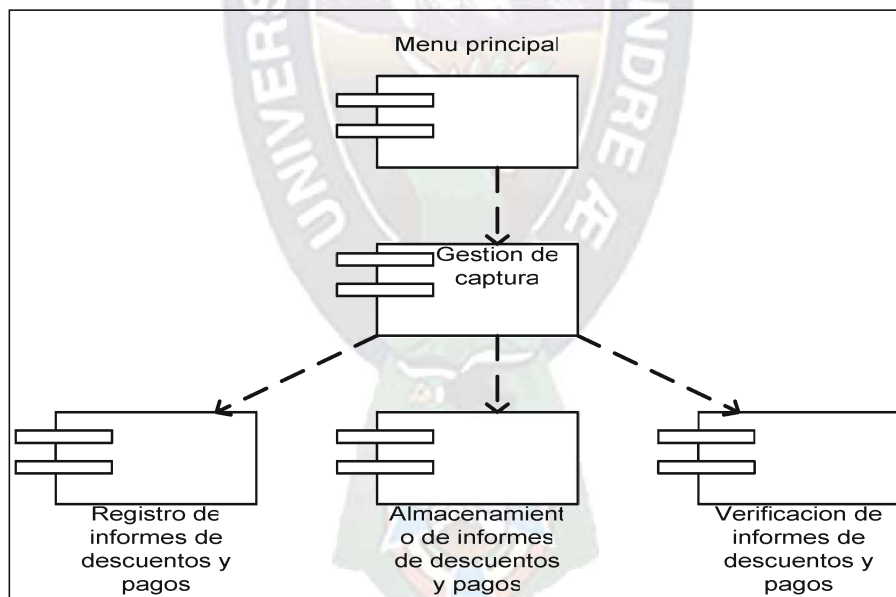
El diagrama de componentes nos permite ver la estructura de los módulos ejecutables del sistema.

Figura 3.17 diagrama de componentes comunes



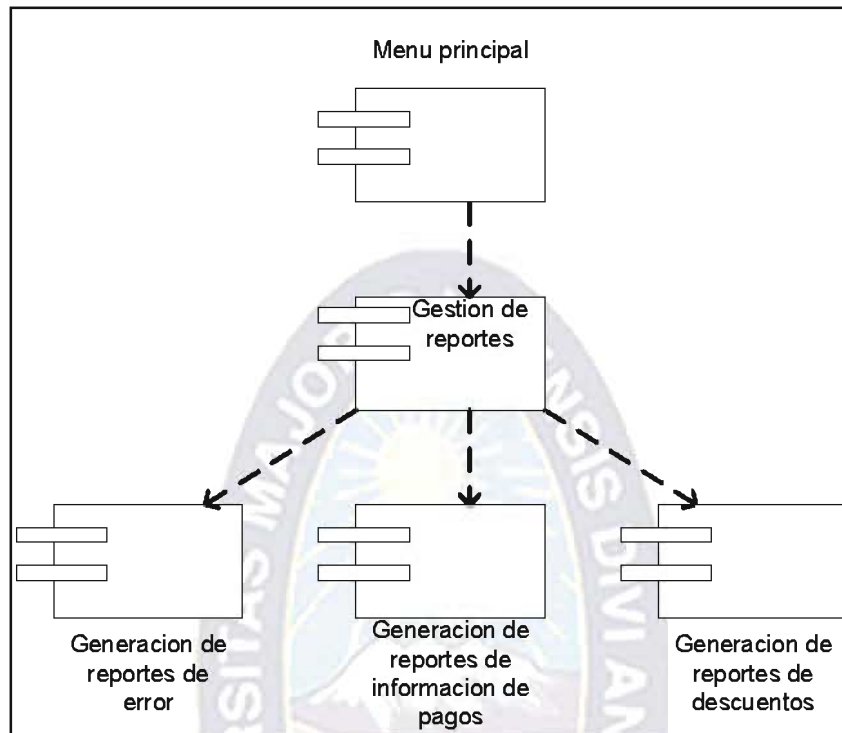
Fuente: Elaboración Propia

Figura 3.18 diagrama de componentes de gestión de captura



Fuente: Elaboración Propia

Figura 3.19 diagrama de componentes de gestión de reportes



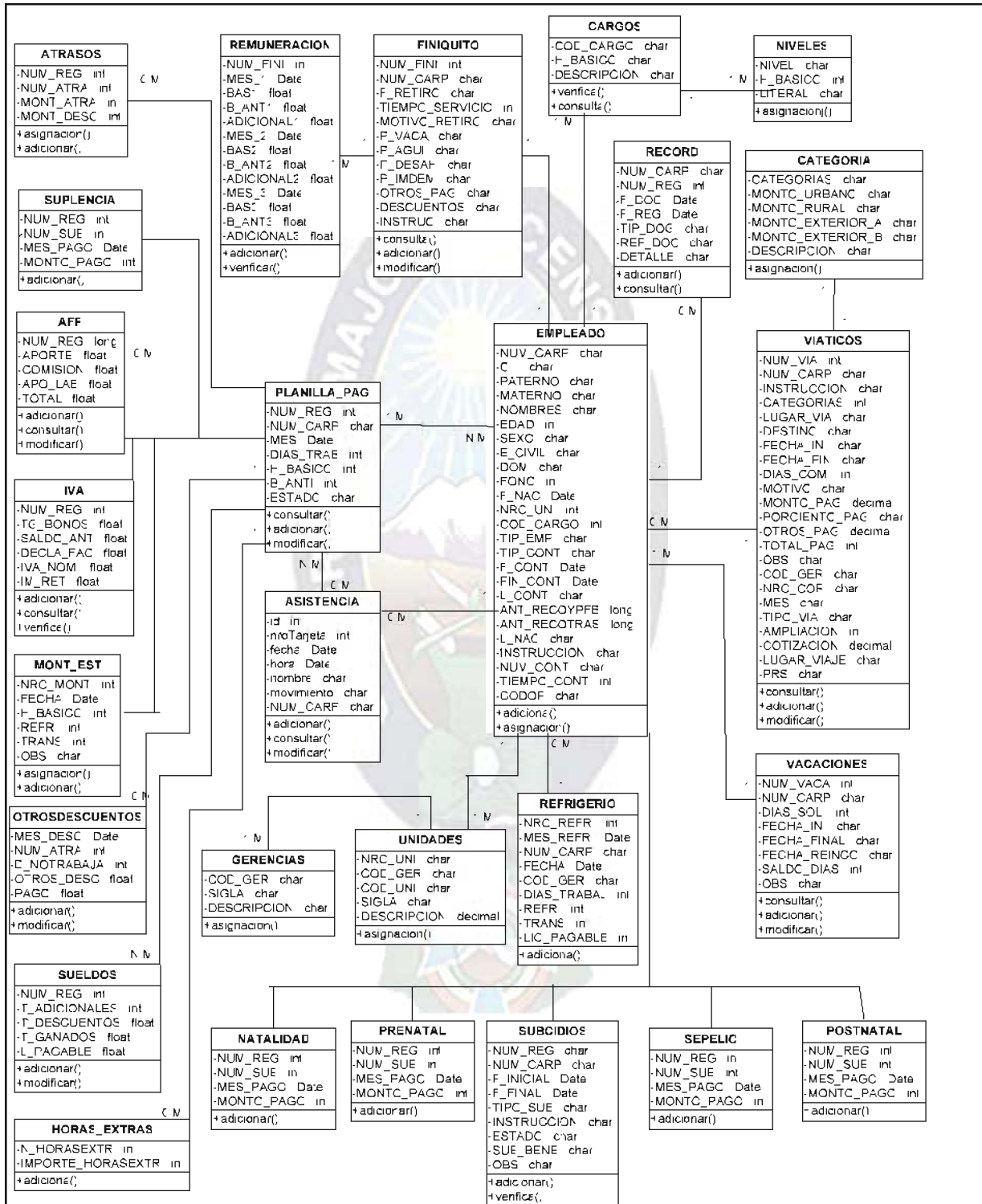
Fuente: Elaboración Propia

3.2.3 DISEÑO DEL DIAGRAMA DE CLASES

El sistema se ha logrado modelar de acuerdo al siguiente Diagrama de clases que se muestra.

Del Diagrama de clases de la Figura 3.20 se ha logrado identificar las entidades, atributos y relaciones, esto considerando sólo las clases más importantes.

Figura 3.20 Diagrama de Clases del Sistema



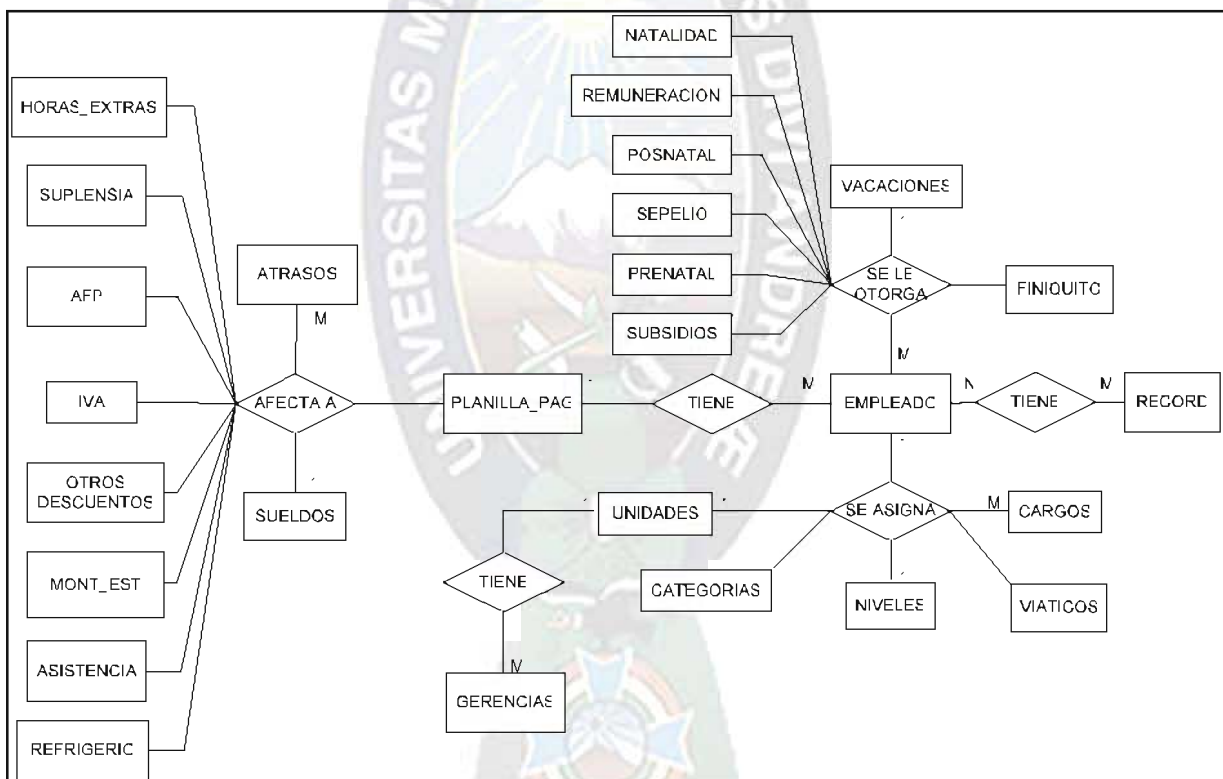
Fuente: Elaboración Propia

3.2.4 DISEÑO DEL DIAGRAMA DE ENTIDAD – RELACIÓN

Del Diagrama de clases de la Figura 3.20 se ha logrado identificar las entidades, atributos y relaciones, esto considerando sólo las clases más importantes.

De donde se ha logrado elaborar el Diagrama Entidad- Relación que nos proporciona una representación grafica mas clara de nuestra base de datos la cual se puede observar en la Figura 3.21

Figura 3.21 Diagrama E-R



Fuente: Elaboración Propia

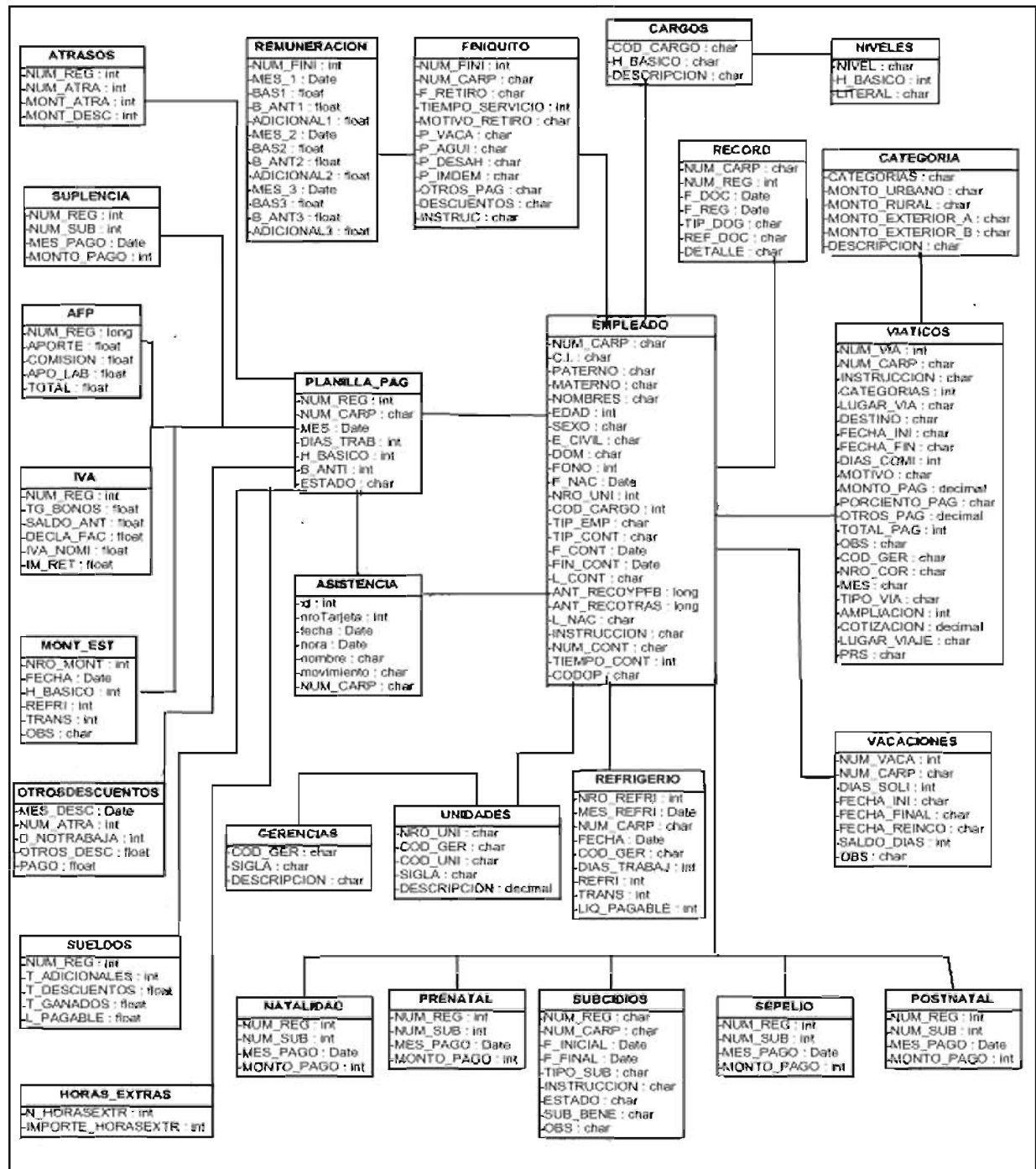
3.2.5 DISEÑO DE LA BASE DE DATOS

Se logro obtener una estructura de la base de datos, en la cual se busca que puedan efectuarse combinaciones de consultas dependiendo de la necesidad del usuario, además tomando en cuenta que los datos deben ser precisos y

consistentes, obteniendo un almacenamiento, una recuperación y actualización de datos eficiente

A continuación en la Figura 3.22 se ve el esquema de la base de datos con la que el sistema funciona de manera adecuada.

Figura 3.22 Esquema de la base de datos



Fuente: Elaboración Propia

3.3 FASE DE APRENDIZAJE

Esta es la tercera fase del modelo ASD esta consiste en la revisión de calidad que se realiza al final de cada ciclo. En la misma se analizan las siguientes categorías de cosas para aprender

- Calidad del resultado desde la perspectiva del cliente
- Calidad del resultado desde la perspectiva técnica

El calculo de calidad desde la perspectiva técnica se encuentra elaborado en el capitulo siguiente en el cual se ve que el proyecto se encuentra en el margen de calidad optimo. Una vez que estos puntos estén sin problemas puede realizarse la entrega al cliente. En el presente proyecto luego de optimizar los puntos mencionados, esto se logro después de varias iteraciones a través del ciclo de vida de ASD, se ha podido aprender de errores y finalmente se logro obtener el producto requerido. Los diseños finales con los que proyecto llevo a su conclusión los podemos observar a continuación.

PANTALLA INICIAL.- Esta es la encargada para recibir los datos de ingreso del usuario al sistema y si es correcta logra acceder al sistema SICAP, para entrar al sistema tendrá solo que ser personal administrativo de la empresa de R.R.H.H. u otros autorizados por la misma..

Figura 3.23 Interfaz se Usuario Pantalla Inicial

Fuente: Elaboración Propia

En esta primera pantalla que es la de inicio del sistema se identifica al usuario mas un código de contraseña o pas Word personal de cada operador del sistema en la primera celda se deberá introducir el apellido paterno del operador, en la segunda casilla se deberá introducir el pas Word del operador, posteriormente el operador deberá hacer un clic en el botón de ingresar para así poder ingresar al sistema.

- **PANEL DE CONTROL.-** El Panel de control para los diferentes usuarios es el siguiente:

- Panel de control de Afiliación

En acá los encargados de la administración de afiliación del sistema lograra realizar todas las funciones que aviamos descrito en nuestros casos de uso, siendo esta la pantalla para dar de alta a un nuevo personal.

Figura 3.24 Interfaz de Usuario Pantalla de afiliación

SISTEMA DE AFILIACION YPFB

Archivo Ayuda

Habilitar Registro

FORMULARIO DE AFILIACION

Datos Personales

Ap. Paterno : PACHECO Sexo : Masculino Femenino

Ap. Materno : FERNANDEZ Estado Civil : CASADO

Nombres : HECTOR Carnet Id. : 6101604 LP

Nacimiento : 01/03/1978 Telefono : 2420043

Lugar Nacimiento: LA PAZ Domicilio : ZONA ALTO SOPOCACHI

Datos de la Empresa

Lugar de Contrato : LA PAZ Fecha de Contrato : 01/06/2001

Unidad : PRESIDENCIA EJECUTIVA

Gerencia : PRESIDENCIA

Cargo : OFICIAL ADMINISTRATIVO III

Nivel Salarial : 06

Haber Básico : 3000

Antigüedad Reconocida	Años	Meses	Días
Y.P.F.B	0	0	0
Otras Entidades	1	2	0

Tipo de Empleado: Profesional No Profesional

Datos Adicionales

Tipo de contrato : Indefinido Temporal

Tiempo de Contrato : INDEFINIDO

Instruccion : PRS/234

No. Contrato : 1256

RESTAURAR

COPIA DE SEGURIDAD

NUEVO

GUARDAR

MODIFICAR

ELIMINAR

CONSULTAS

CANCELAR

SALIR

Fuente: Elaboración Propia

En esta pantalla se refleja todos los datos que el nuevo personal deberá de solicitar ala empresa para su respectivo registro y así poder admitirlo como un funcionario mas de la empresa.

Para el mejor funcionamiento del sistema se facilito una opción de ayuda para que el administrador de afiliación no tenga problemas al operar el sistema SISCAP.

-Panel de control del módulo de vacación

En acá los encargados de la administración de control de vacación del sistema lograra realizar todas las funciones que aviamos descrito en nuestros casos de uso, siendo esta la pantalla para dar uso de asignar vacación a un empleado de la empresa.

Figura 3.25 Interfaz de Usuario Pantalla de vacaciones

VACACION
— ○ ×

Ayuda
Ayuda del Sistema F6

SOLICITUD DE VACACIONES

Solicitud que Realiza

Número de carpeta : 161059 **Fecha de contrato** : 10/07/2005 12:58:23

Empleado : TORREZ LAURA CARLOS **Antigüedad Reconocida**

Domicilio : M **A M D**

Teléfono : 11 **YPFB** : 0 - 0 - 0

Fecha Actual : 18/07/2007 **Otros** : 1 - 6 - 0

SOLICITAR VACACION

GUARDAR SOLICITUD

MODIFICAR SOLICITUD

ELIMINAR SOLICITUD

CONSULTAS

CANCELAR

SALIR

Vacaciones Pendientes

Gestion	Años de Servicio	Días	Saldo
2006	2	15	15
2007	3	15	15
Total :		30	30

Solicitud que Realiza

Días Sol. : 15 Solicitar

Fecha Inicio : 18/07/2007

Motivo u Obs. : VIAJE

Calendario de Vacaciones

SABADO	DOMINGO	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
				18/07/2007	19/07/2007	20/07/2007
21/07/2007	22/07/2007	23/07/2007	24/07/2007	25/07/2007	26/07/2007	27/07/2007
28/07/2007	29/07/2007	30/07/2007	31/07/2007	01/08/2007	02/08/2007	03/08/2007
04/08/2007	05/08/2007	06/08/2007	07/08/2007			

Feriado 22/01/2004

Validar

Fecha Final 07/08/2007

Fecha Reincorp. 08/08/2007

Fuente: Elaboración Propia

Para la solicitud de vacación se deberá primero pulsar el botón de solicitud de vacación posteriormente se habilitara la celda para la búsqueda del personal al cual se le asignara vacación, luego de elegir al personal se mostrara en pantalla los datos personales y la antigüedad del personal en la empresa junto con los días de vacación que le corresponden hasta la fecha y con el saldo que le quedan de vacación, en el caso que no tenga saldo el solicitante se le rechaza la petición, una vez que se le aprueba la petición se registra los datos solicitados que se muestra en pantalla y se procede a guardarlo pulsando el botón guardar solicitud.

Para el mejor manejo de este modulo de vacaciones se le facilito al administrador de vacaciones una opción de ayuda en la cual explica todo acerca del manejo u manipulación del modulo de vacaciones.

-Panel de control del modulo de viático

En aquí el encargado de administrar el módulo de viático del sistema lograra realizar todas las funciones que aviamos descrito en nuestros casos de uso, siendo esta la pantalla la que asigna un viático a un empleado.

Figura 3.26 Interfaz de Usuario Pantalla de viáticos

VIATICOS
Ayuda

SOLICITUD DE VIATICOS

Datos del Empleado

Número de carpeta : PACHECO PACHECO HECTOR WILFREDO - 161034
Empleado : PACHECO PACHECO HECTOR WILFREDO
Cargo : OFICIAL ADMINISTRATIVO III
Gerencia : GERENCIA DE ADMINISTRACION Y FINA

Datos de la Comisión

Fecha de Elaboración : 18/07/2007
Lugar Elaboración : LA PAZ elija la categoria : 5
Instruido por : SAUL BARRIENTO Cotizacion del dolar : 8
Destino : SANTA CRUZ Viaje a: SANTA CRUZ
Motivo : REUNION ADINISTRATIVA
Porcentaje de Pago : 100 % 70 % 40 25
Tipo de Comisión : RURAL Otros Pagos : 0
Fecha Inicio : 18/07/2007 Fecha Final : 20/07/2007
Días de Comisión : 3
Observaciones : NINGUNA
DIGITE PRS : prs/546

SOLICITAR VIATICO
GUARDAR VIATICO
MODIFICAR VIATICO
AMPLIAR VIATICO
CONSULTAS
CANCELAR
SALIR

Fuente: Elaboración Propia

Para poder solicitar un viático se tiene que pulsar el botón solicitar viático posteriormente se procede a llenar todos los datos que se le solicitan en la

pantalla, una vez terminado de llenar los datos se procede a guardar el viático pulsando en el botón guardar viático.

Para un mejor implementación u manejo del SISCAP se facilitó al administrador de vacaciones una opción de preguntas comunes o ayuda del sistema.

-Panel de control del módulo de planillas

El encargado de administrar el módulo de planilla se encarga de elaborar las planillas de sueldos tomando en cuenta los datos reportados de los anteriores módulos como son: el módulo de vacaciones, de viático, asistencia, afiliación.

Figura 3.27 Interfaz de Usuario Pantalla de sueldos

Fuente: Elaboración Propia

En esta pantalla podemos ver tres módulos unidos que son el de control de pagos de planillas o sueldos, subsidios, servicio de te y transporte.

El primero que es el de sueldos nos sirve para realizar los pagos de los sueldos de los usuarios de la empresa asignándole el tipo de transacción que se le hará al usuario.

El segundo que es subsidios en este modulo se realiza los pagos de beneficios sociales que se le realiza al usuario como ser prenatal natalalidad etc.

En el tercer modulo que es el de servicio de te y transporte se realiza el calculo de te y transporte que hizo uso el usuario de la empresa a favor suyo.

Para el mejor manejo de este modulo de pagos de planilla se facilito al administrador de planillas una opción de ayuda como se puede observar en la figura 3.27

-Panel de control del modulo de finiquitos

El encargado de controlar el modulo de finiquitos que son los pagos de beneficios sociales de cada usuario de la empresa es el administrador de finiquitos.

Figura 3.28 Interfaz de Usuario Pantalla de finiquitos

The screenshot shows a web application window titled "FINIQUITO" with a standard Windows-style title bar. Inside the window, there is a header area with the word "FINIQUITO" in large blue letters and a "Fecha Actual" (Current Date) field displaying "18/07/2007". Below the header, there are two main sections: "Datos de Empleado" (Employee Data) and "Datos de Retiro" (Retirement Data). The "Datos de Empleado" section includes fields for "Número de Carpeta" (161034), "Nombre del Empleado" (PACHECO PACHECO HECTOR WILFREDO), and "Fecha de Ingreso" (05/06/2000 12:55:32). The "Datos de Retiro" section includes fields for "Fecha de Retiro" (18/07/2007), "Instruccion" (prs/546), "Otros Pagos" (5500), and "Otros Descuentos" (580). There are also radio buttons for "Motivo de Retiro" (Voluntario and Forzoso). At the bottom of the window, there is a row of buttons: "CREA FINIQUITO", "GUARDA FINIQUITO", "MODIFICAR", "ELIMINAR", "CONSULTAS", "CANCELAR", and "SALIR".

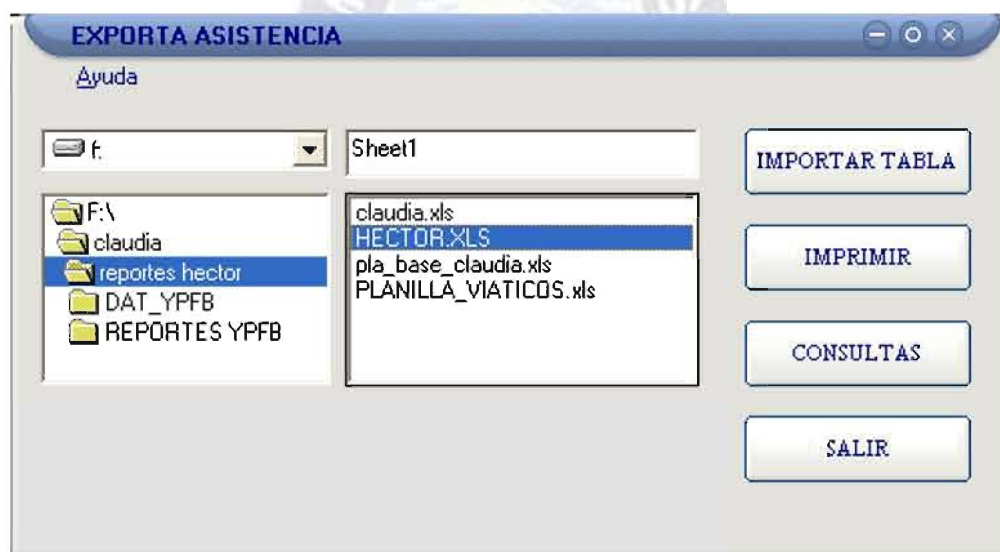
Fuente: Elaboración Propia

En este modulo se realiza la baja de un personal ya se a voluntario o forzoso este modulo se encarga de calcular el beneficio social que ele corresponde al empleado por años de servicio u otros, para el mejor manejo de este modulo se le facilito al administrador una opción de ayuda como se puede observar en la figura 3.28.

-Panel de control de exportación de datos para el modulo de asistencia

En este modulo se realiza la exportación de los datos de asistencia emitidos por el sistema winpack de la empresa.

Figura 3.29 Interfaz de Usuario Pantalla de exportación de asistencia



Fuente: Elaboración Propia

En este modulo se exporta los datos desde una planilla Excel ala base de datos del sistema, la planilla Excel a exportar es emitida por el sistema winpack con el que la empresa cuenta actualmente que consiste de un sistema que es Hardware y Software a la vez, el usuario tiene opción a una ayuda o manual de ayuda que le facilita el manejo del modulo de exportación.

-Panel de control del modulo de Asistencia

El encargado de administrar este módulo de asistencia es el administrador de asistencia, el cual se encarga de controlar la asistencia, faltas, horas extras, permisos, etc para la facilidad del manejo para el Administrador se le facilita una opción de ayuda como se puede observar en la figura 3.30 en la cual explica todo acerca del modulo de asistencia paso por paso para una mejor manipulación del mismo.

Figura 3.30 Interfaz de Usuario Pantalla de asistencia

CONSULTA ASISTENCIA

[Ayuda](#)

SABADO	DOMINGO	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
27/01/2007	28/01/2007	22/01/2007	23/01/2007	24/01/2007	25/01/2007	26/01/2007
03/02/2007	04/02/2007	29/01/2007	30/01/2007	31/01/2007	01/02/2007	02/02/2007
10/02/2007	11/02/2007	05/02/2007	06/02/2007	07/02/2007	08/02/2007	09/02/2007
17/02/2007	18/02/2007	12/02/2007	13/02/2007	14/02/2007	15/02/2007	16/02/2007

22/01/2007 20/02/2007 21/02/2007 17/07/2007-12

fecha inicial fecha final Elija el mes Buscar planilla por codigo

NUMCARP	AP	AM	NOM	22/01/2007	23/01/2007	24/01/2007	25/01/2007
161034	PACHECO	PACHECO	HECTOR W	A		ojo	
161042	APAZA	POMA	CLAUDIA G		A		P
161059	TORREZ	LAURA	CARLOS			BM	
161067	CAMARGO	BARRIENTE	JULIO CESA	V	V	V	

VERIFICAR TABLA IMPORTADA

ATRASOS

ERRORES DE ATRASOS

HORAS EXTRAS

ERRORES DE HORAS EXTRAS

CREA REPORTE_PLANILLA

LLENA REPORTE_PLANILLA

GUARDAR PLANILLA

SALIR

Fuente: Elaboración Propia

Políticas de seguridad

Los datos e información producto de la afiliación del personal, son uno de los recursos más valiosos de esta institución, por esta razón, deben ser controlados con los mismos cuidados que los demás módulos de control del personal.

Las políticas de control que considera el sistema de Administración de Personal (SICAP). Son:

- Control de acceso
- Control de identificación de usuarios
- Control de autorizaciones

El control de acceso, se refiere a determinar que usuarios están autorizados para acceder al equipo en el cual está instalado el SICAP, como también, a los usuarios que tendrán acceso al sistema por medio de un identificador. Para ello, el responsable encargado de administrar el sistema debe de identificar los requerimientos de cada uno de los usuarios para acceder a las instalaciones y a los módulos correspondientes. Es importante considerar que solo personas autorizadas pueden realizar modificaciones a los datos del personal de la empresa.

Por las características que tiene cada tarea, se presenta a continuación la tabla 3.1 De control de autorizaciones, que permitirá restringir el acceso a la base de datos para la operación de consultas, Transcripción, Verificación, Impresión de certificados u memorando y modificaciones.

Tabla 3.1 autorizaciones para usuarios del SICAP

USUARIO	CARGO	MÓDULOS
Administrador de afiliación	Usuario	-Afiliación de nuevo personal. -realiza altas, bajas y modificaciones del personal.
Administrador de asistencia	Usuario	-Control de asistencia del personal. -Emisión de reportes horas extras.
Administrador de planilla de sueldos	Usuario	-Control de pago de Sueldos. -Emisión de Planillas de pagos.
Administrador de vacaciones	Usuario	-Control de asignación de vacaciones.

Administrador de viáticos	Usuario	-Emisión de reporte de vacación. -Control de asignación de viático.
Administrador de finiquitos	Usuario	-Emisión de reporte de viático. -Control de Pago de finiquitos. -Emisión de planilla de finiquitos.
Jefe de personal/ administrador	Administrador	-Control del sistema SICAP -administra todos los módulos mencionados.

Fuente: elaboración propia

PLAN DE LA SEGURIDAD DE LA INFORMACIÓN

El plan o políticas de seguridad de la información, es un documento donde se establecen los principios organizacionales y funcionales de la seguridad informática en una organización.

Los objetivos que se persiguen en el plan de seguridad de la información del SICAP son los siguientes:

- Protección de las bases de datos del sistema de Control y Administración de Personal (SICAP).
- Obtención de copias de respaldo (Backup) en caso de contingencias.
- Controlar el acceso de usuarios al sistema que evite agresiones físicas o lógicas.

En ella se refleja una serie de normas, reglamentos a seguir, donde se definen las distintas medidas a tomar para proteger la seguridad del sistema, las funciones y responsabilidades de los distintos componentes de la organización y los mecanismos para controlar su correcto funcionamiento.

Vulnerabilidad, Amenazas y Contramedidas

Estos tres conceptos son considerados para mantener la seguridad del sistema informático SICAP.

La Vulnerabilidad, se refiere al punto o aspecto del sistema que es susceptible de ser atacado o de dañar la seguridad del mismo. Las Vulnerabilidades consideradas son:

- Vulnerabilidad Física, considera el ambiente donde se instalara el sistema. Para ello debe restringirse el acceso para evitar robos, modificaciones o destrucción del mismo. En el sistema SISCAP esto se toma en cuenta así que por medidas de seguridad física se previo que el sistema será instalado en el servidor de la empresa en el cual cuenta con un control del departamento de sistema de la empresa, al cual solo tiene acceso a personal autorizado.
- Vulnerabilidad Natural, considera fallos por desastre natural o ambiental que pueden dañar al sistema, como ser: fuego, inundaciones, fallos electrónicos o picos de potencia y temperaturas excesivas. Para la vulnerabilidad natural se tiene en cuenta que el sistema tendrá una copia de seguridad del sistema en un cd.
- Vulnerabilidad de las comunicaciones, considera los riesgos la interceptación de las comunicaciones a través de la red bueno el sistema SISCAP no es via Internet si no intranet y en el envío de datos este esta protegido mediante encriptación al enviar un dato este es automáticamente encriptado y al recibirlo se desencripta.

Las Amenazas principales consideradas son: la interceptación de usuarios no autorizados al sistema, la modificación de todo o en parte el contenido de la información de los registros de información, y la generación de información no autorizada en el sistema. Como se menciona anteriormente para prever estos

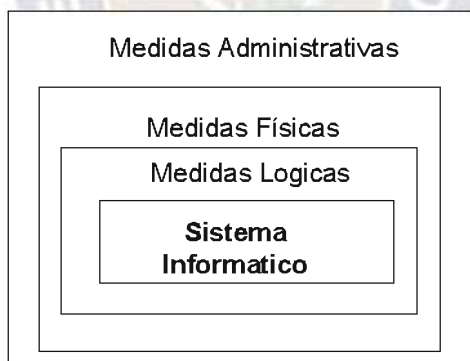
riesgos se toma en cuenta la restricción del personal mediante un password así restringiendo el acceso al sistema y evitando posibles interceptación de datos para ello se toma la encriptación de los datos para su protección.

Medidas de Seguridad

La seguridad de las instalaciones donde se instalara el sistema (se instalara el sistema SISCAP en el departamento de sistemas el cual cuenta con una seguridad desde ahí se distribuirá a la oficina del jefe de personal de RRHH y a los respectivos administradores modulo a modulo), de los datos y la información original y generada, es también parte de una conversión satisfactoria.

Básicamente se consideran cuatro aspectos interrelacionados: Seguridad Lógica, Seguridad Física y Seguridad de Comportamiento (figura 3.31), que permitirán disminuir la vulnerabilidad del sistema y mantener una calidad de seguridad alta.

Figura 3.31 Medidas de Seguridad



Fuente: Apuntes de Seguridad y Protección de la Información 1998

Medidas de Seguridad Lógica

La seguridad Lógica, básicamente se refiere a los controles de acceso y el uso correctos de los recursos e información del sistema SICAP. Los tipos de controles considerados son los siguientes:

- Se establecerán niveles de usuario, los cuales a partir de las contraseñas y códigos de autorización asignadas, tendrán acceso solamente a los módulos autorizados. Considera dos pasos:
Identificación, el usuario debe proporcionar su identificador de usuario, que puede estar compuesto por iniciales de su nombre o añadido un número, etc.
Autenticación, el usuario debe demostrar que es quien dice que es, para este propósito existen tres métodos:
 1. Algo que solo el usuario sabe.
 2. Algo que solo el usuario tiene.
 3. Algo que el usuario es.
- Debe mantenerse el resguardo una copia de respaldo del software de instalación eso se realizara en un cd extra ya que anteriormente se tiene una copia original.
- Debe obtenerse copia de seguridad una vez por semana(esto se realizara cada lunes de cada semana) de la base de datos almacenados en el sistema.
- Debe establecerse el grado de responsabilidad de cada tipo de usuarios considerando la siguiente clasificación:
 1. Administrador del sistema.
 2. Usuarios del sistema.
 3. Las personas relacionadas con el sistema pero sin necesidad de usarlo.
 4. Las personas ajenas al sistema.

Medidas de Seguridad Física

Se refiere a la aplicación de mecanismos para impedir el acceso directo o físico no autorizado al sistema. Para ello, se consideran los siguientes puntos.

- Los equipos en los cuales se instale el sistema SICAP, son ubicados de modo que un usuario típico no pueda moverlo.

- Debe establecerse una diferenciación jerárquica de las personas que tendrán acceso al servidor y los tiempos de permanencia de los mismos esto es controlado por el departamento de sistemas de la empresa.
- El ambiente debe garantizar el suministro de energía o UPS, actualmente la empresa tiene generadores de energía de respaldo por si un corte de luz llegara a ocurrir.
- Debe mantenerse lejos del alcance de agentes nocivos o contingencias.

Medidas Administrativas y Organizativas

Las medidas administrativas son aquellas que deben ser tomadas por las personas encargadas de definir la política de seguridad para ponerla en practica, hacerla viable y vigilar su correcto funcionamiento.

Los empleados y usuarios autorizados para acceder al sistema, deben ser totalmente concientes sobre las responsabilidades durante el manejo del mismo.

Para ello se sugiere la implantación de las siguientes medidas administrativas:

- Dar a conocer al personal a través de publicaciones y charlas, las políticas de seguridad y de las medidas tomadas para ponerlas en practica.
- Establecer las sanciones administrativas que puedan aplicarse al personal que infrinja la seguridad informática.
- Inventario periódico y frecuente de equipo, software y documentación importante.
- Clasificación y almacenamiento seguro de documento impresos y almacenados magnéticamente.

Las medidas administrativas referidas específicamente a los usuarios son las siguientes:

- El usuario no debe divulgar su nombre de usuario y contraseña.
- El usuario debe cerciorarse deque en el momento de digitar su nombre de usuario y contraseña no haya ninguna persona que lo observe.

- El usuario debe cambiar inmediatamente su contraseña en caso de sospecha el conocimiento del mismo por otras personas.
- En caso de olvido de contraseña, el usuario debe solicitar al administrador que nuevamente le facilite una nueva contraseña.
- El usuario es responsable de todas las operaciones que se efectúen con la contraseña asignada.

PLAN DE CONTINGENCIA DE LA SEGURIDAD DE LA INFORMACIÓN

El plan de contingencia, es un plan de recuperación en casos de desastre que prevee a la institución una guía rápida y organizada para posibilitar la capacidad de seguir trabajando en un plazo mínimo después de que se haya producido un problema.

Los objetivos principales del plan de contingencia son los siguientes:

- Minimizar el impacto del desastre en la empresa y los departamentos que la componen.
- Recuperación de la información a traves de copias de seguridad (Backup) diarias, para evitar posibles pérdidas de información ante posibles daños a la unidad de almacenamiento.
- Revisión periódica a los equipos y servidores y la recuperación del funcionamiento del sistema en base a la preparación de recursos alternativos.
- Garantizar las conexiones de energía y el suministro a través de un almacenador de energía (UPS).

Lo anterior mencionado se toma en cuenta ya que el sistema realiza copias de seguridad cada lunes de cada semana para la seguridad.

Factores posibles para la ejecución del Plan de Contingencia

A continuación se mencionaran los distintos factores posibles que provocarían el procesamiento de datos.

- Pérdida del personal clave u administrador.
- Interrupción por huelga.

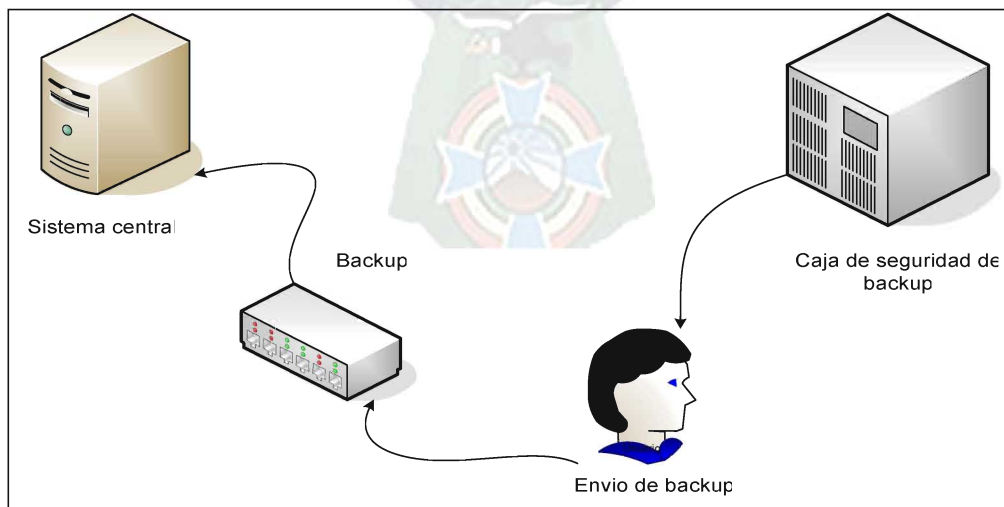
Si en el caso que se llegara a presentar la pérdida de algún administrador el jefe de personal tiene el poder para el asignar a algún otro personal la administración correspondiente tomando así una nueva asignación de operador dándole una nueva clave y contraseña para así poder ingresar al sistema que es de fácil manipulación.

Procedimiento de Emergencia

A continuación se detalla el plan de contingencia para el Sistema Integrado de Control y Administración de Personal SICAP.

- En caso de caída del sistema, se debe recurrir a la copia de seguridad últimos que se hicieron para su restablecimiento del sistema como se muestra en la figura 3.32

Figura 3.32 Recuperación con Backups



Fuente: elaboración propia

En caso de baja del servidor principal, el servidor de respaldo opera como principal hasta que el servidor principal este restaurado. Para ello, el servidor secundario debe poseer las mismas características de información que el principal figura 3.33

Figura 3.33 transferencia de información principal/secundario

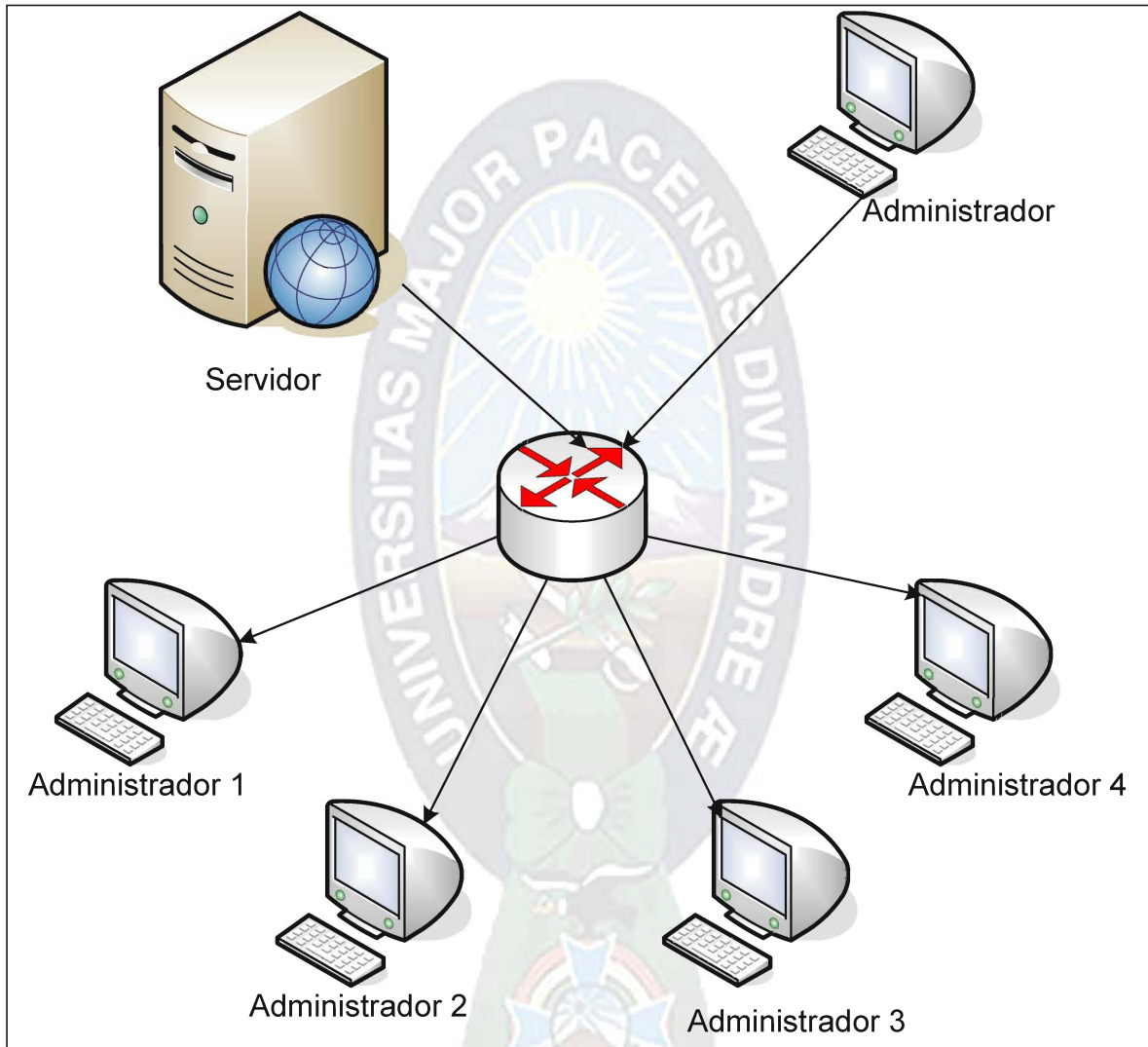


Fuente: elaboración propia

3.3.1 ESQUEMA DE FUNCIONAMIENTO DE LA EMPRESA

Cada administrador contara con el sistema estas trabajaran inicialmente de manera local, y posteriormente a través de la intranet de la empresa de tal modo que el servidor podrá relacionar y almacenar la información centralizándola.

Figura 3.34 Esquema de funcionamiento del sistema



Fuente: Elaboración Propia

PRUEBAS Y CALIDAD DEL SOFTWARE

4.1 INTRODUCCIÓN

Cuando se habla de calidad de software esto quiere decir que miramos el valor técnico del producto, medido normalmente con expresiones matemáticas que incluyen defectos, las fallas y el tiempo, los mismos que permiten evaluar la calidad del producto en este caso software. En este capítulo desarrollaremos algunas métricas para tener una visión de la calidad del sistema obtenido en el presente proyecto de grado.

4.2 PRUEBAS DE CAJA BLANCA

Las pruebas serán realizadas para la parte funcional del sistema tomando en cuenta un diagrama de flujo general utilizando una métrica de complejidad ciclomatica, la cual nos dará una cantidad aproximado de casos de prueba que serán aplicados al código fuente.

Tabla 4.1 Matriz de grafo de la métrica de complejidad ciclomatica

Conexión	A	B	C	D	E	F	G	H	I	SUMATORIA SUM
Nodos										
A		1								1-1=0
B			1						1	2-1=1
C				1						1-1=0
D					1				1	2-1=1
E						1		1		2-1=1
F							1			1-1=0
G										1-1=0
H									1	1-1=0
I										
										SUM= 3+1=4

Fuente: elaboración propia

Complejidad ciclomatica de acuerdo a:

$$V(G)=A-N+2$$

$$V(G)=P+1$$

Donde:

N=Numero de nodos A= Numero de Aristas P=Numero de nodos

Predicados

$$N=9 \quad A=11 \quad P=3$$

Remplazando:

$$V(G)=11- 9 +2$$

$$V(G)= 4$$

$$V(G)= 3+ 1$$

$$V(G)= 4$$

El valor $V(G)=4$ nos indica que son cuatro los casos de pruebas que deben de ejecutarse y diseñar para garantizar que se cubren todas las sentencias del programa.

Sacamos caminos independientes:

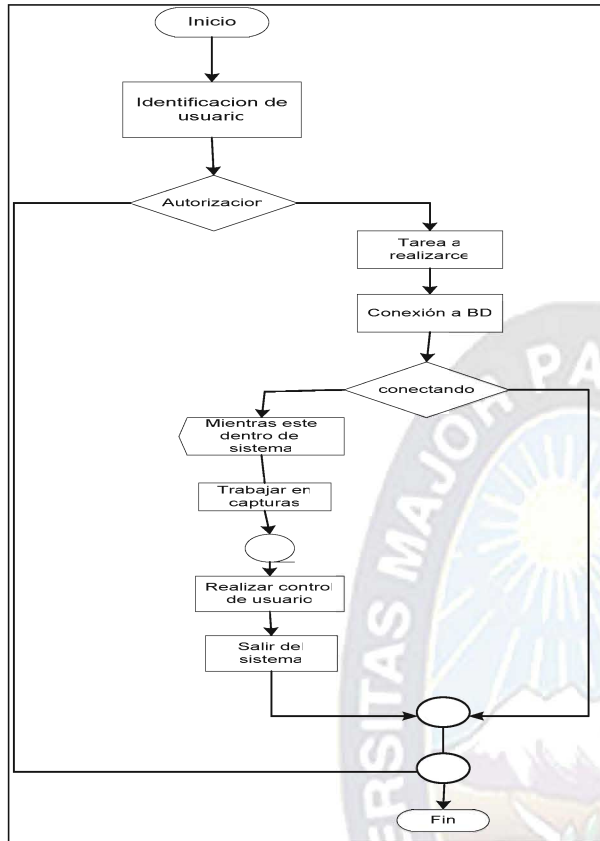
Camino 1: A – B - I

Camino 2: A – B – C – D – I

Camino 3: A – B – C – D – E – H – I

Camino 4: A – B – C – D – F – G – E – H – I

Figura 4.1 Diagrama de flujo general



Fuente: Elaboración Propia

Figura 4.2 Grafo de Métrica

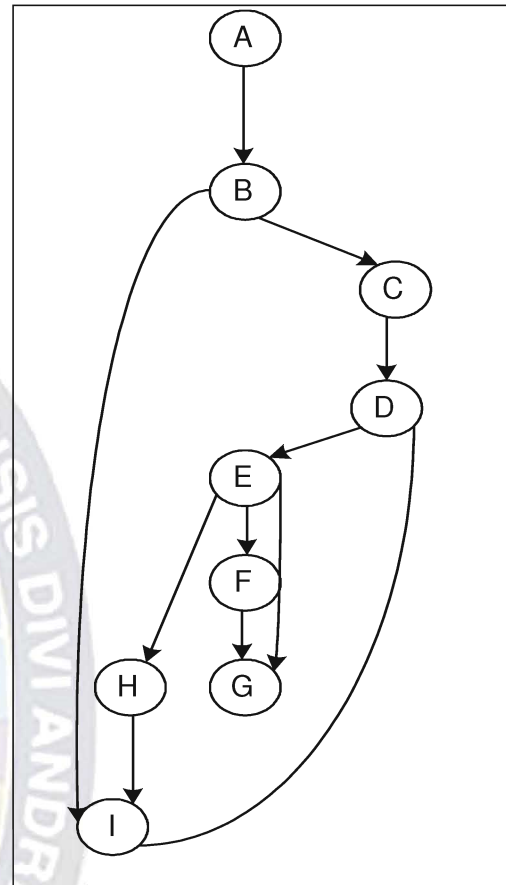
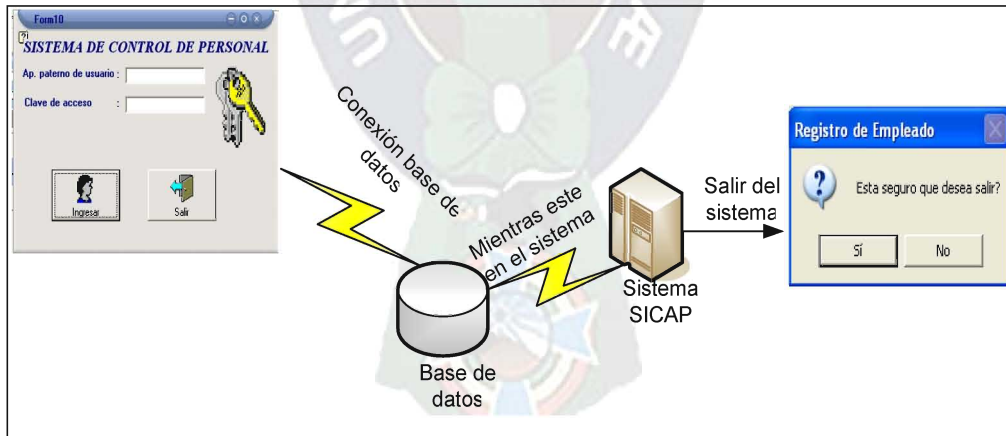


Figura 4.3 esquema funcional del flujo general



Fuente: Elaboración Propia

4.3 FUNCIONALIDAD

El punto función es una métrica orientada a la función del software y del proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa,

los puntos de función se calculan realizando una serie de actividades, comenzando por determinar los siguientes números:

- **Numero de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.
- **Numero de salidas de usuarios.** Estas se refieren a informes, mensaje de error, etc. Es decir, salidas que proporcionan al usuario información orientada a la aplicación.
- **Numero de peticiones de usuario.** Una petición esta definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida inactiva.
- **Numero de archivos.** Se cuenta cada archivo maestro lógico.
- **Numero de interfaces externas.** Se cuenta todas las interfaces legibles por el ordenador que son solicitados para transmitir información a otro sistema.

De acuerdo a lo mencionado es que se tiene los resultados en la siguiente tabla 4.2.

Tabla 4.2 entradas para el cálculo de funcionalidad

Entradas de usuario	28
Salidas de usuario	19
Consulta de usuario	16
Numero de archivos	23
Interfaces externas	2

Fuente: Elaboración Propia

Los puntos de función se calculan rellorando la tabla 4.2 con los datos obtenidos, considerando un factor de ponderación medio.

Tabla 4.3 calculo de puntos de función sin ajustar

Parámetros de medición	cuenta		Factor de ponderación medio		totales
Numero de entradas de usuario	28	X	4	=	112
Numero de salidas de usuario	19	X	5	=	95
Numero de consultas de usuario	16	X	4	=	64
Numero de archivos	23	X	10	=	230
Numero de interfaces externas	2	X	7	=	14
	Cuenta total				515

Fuente: Elaboración Propia

La relación que permite calcular los puntos de función es la siguiente:

$$PF = \text{Cuenta Total} * (\text{Grado de confiabilidad} + \text{tasa de error} * \sum Fi)$$

Donde:

- PF= Medida de Funcionalidad
- Cuenta Total = es la suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos.
- Grado de confiabilidad= Es la confiabilidad estimada del sistema.
- Tasa de error= probabilidad subjetiva estimada del dominio de la información, este error estimado es del 1 %.
- Fi = son valores de ajuste de complejidad que toman los valores de la tabla 4.4 y que dan respuesta a las preguntas de la tabla 4.5.

Tabla 4.4 valores de ajuste de complejidad

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

Fuente: Elaboración Propia

Tabla 4.5 ajuste de complejidad del punto función

ESCALA	Sin Importanci 0	Incidental 1	Moderado 2	Medio 3	Significati vo 4	Esencial 5
factor						
1.¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2.¿Se requiere comunicación de datos?				X		
3.¿Existen funciones de procesos distribuidos?				X		
4.¿Es crítico el rendimiento?		X				
5.¿Será ejecutado el sistema en S.O. existente?				X		
6.¿Requiere el sistema de entrada interactiva en linea ?					X	
7.¿Requiere el sistema entrada de datos interactiva sobre múltiples ventanas?					X	
8.¿Se actualizan los archivos maestros de manera interactiva?				X		
9.¿Son complejas las entradas, las Salidas, los archivos o las peticiones?			X			
10.¿Es complejo el procesamiento interno?				X		
11.¿Se ha diseñado el código para ser reutilizable?					X	

12. ¿Están incluidas en el diseño la conversión y la instalación?	X	
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones ?	X	
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?		X
Total ($\sum F_i$)		45

Fuente: Elaboración Propia

Con la obtención de los anteriores datos y considerando un grado de confiabilidad mínimo del % es que a continuación calculamos el valor de PF:

$$PF = \text{Cuenta Total} * (\text{Grado de Confiabilidad} + \text{Tasa de Error} * \sum F_i)$$

$$PF = 515 * (0.65 + 0,01 * 45)$$

$$PF = 566,5$$

Si consideramos el máximo valor de ajuste de complejidad como $\sum F_i = 70$, se tiene:

$$PF = 515 * (0,65 + 0,01 * 70)$$

$$PF = 695,25$$

Entonces si $\sum F_i$ es considerada como el 100%, la relación obtenida entre los puntos será:

$$\frac{PF}{PF \text{ maximo}} = \frac{566,5}{695,25} = 0,81$$

Por lo tanto la funcionalidad del sistema es del 81 % tomando en cuenta el punto de función máximo.

Conversión de los Puntos de Función a KLDC

Ahora se debe convertir los Puntos de Función a miles de líneas de código. Tomar en cuenta la siguiente tabla.

Tabla 4.6 Conversión de Puntos de Función a KLDC

LENGUAJE	NIVEL	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Ansi Cobol 74	3	107
Visual Basic	7.00	46
ASP	9.00	36
PHP	11.00	29
Visual C+ +	9.50	34

Fuente:[JONE,S96]

$$\text{LDC} = \text{PF} * \text{Factor LDC/PF}$$

$$\text{LDC} = 566,5 * 46$$

$$\text{LDC} = 26059$$

$$\text{KLDC} = (26059 / 1000) = 26,06$$

4.4 CONFIABILIDAD

Para determinar la confiabilidad del sistema especificamos el instante en el que este comienza a funcionar, determinado por $t_0 = 0$. A partir de este momento se observa el trabajo del sistema hasta que se produzca una falla en el instante T que se va aproximando a una variable aleatoria continua, que nos determina la confiabilidad en términos probabilísticos.

Entonces tenemos las siguientes probabilidades:

$$P(T \leq t) = F(t) \quad (1) \quad ; \text{Probabilidad de fallas}$$

$$P(T > t) = 1 - F(t) \quad (2) \quad ; \text{Probabilidad de trabajo sin fallas}$$

Debido a que se tiene tiempos de inicio como de fin, para el cálculo de estas probabilidades se utiliza la distribución exponencial. Entonces la función F(t) está dada por:

$$F(t) = 0,81 e^{-\lambda t} \quad ; \text{ecuación de confiabilidad}$$

Para el cálculo de error se consideran 10 ejecuciones en un mes en un periodo de 12 meses para obtener una probabilidad de fallas. En este caso, el margen de error λ será de 1/10. Además se utiliza el resultado obtenido del punto función igual a 0,81 para la distribución en un periodo $t=12$ meses.

Obteniendo la probabilidad de fallas se tiene:

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

$$\rightarrow F(t) = 0,81 * e^{(-1/10)*12}$$

$$\rightarrow F(t) = 0,24 \quad ; \text{Probabilidad de fallo}$$

Calculando el valor de la probabilidad de que el sistema trabaje sin fallos a partir del anterior resultado tenemos que: El software tendrá una confiabilidad de 81 % en una gestión, lo cual indica que se mantendrá estable y en funcionamiento.

4.5 MANTENIBILIDAD DEL SOFTWARE

Para calcular la estabilidad del sistema, es decir índice de madurez del software (IMS), se establecerá los cambios que ocurrieron con cada versión del producto.

Para esto el IMS se determina con la siguiente información:

- MT=Numero de módulos en la versión actual.
- Fc=Numero de módulos en la versión actual que se han cambiado
- Fa=Numero de módulos en la versión actual que se han añadido
- Fe=Numero de módulos en la versión actual que se han eliminado.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = [MT - (Fc + Fa + Fe)] / MT$$

En el sistema se obtuvieron los siguientes valores como muestra la tabla 4.6 para la información requerida para el IMS:

Tabla 4.7 información requerida por el IMS

Información	Valores obtenidos
MT	7
Fc	1
Fa	1
Fe	0

Fuente: Elaboración Propia

Ahora calculamos el índice de madurez del software sustituyendo los valores de la tabla 4.6 los cuales son resultados obtenidos del sistema.

$$IMS = [7 - (1+2+0)]/7$$

$$IMS = 0,71$$

Por lo tanto tomando en cuenta la escala siguiente se concluye que el IMS obtenido tiene una estabilidad alta al final de la evolución en las versiones logradas.

75% ≤ IMS ≤ 100% → Óptima

50% ≤ IMS < 75% → Buena

25% ≤ IMS < 50% → Suficiente

0% ≤ IMS < 25% → Deficiente

4.6 PORTABILIDAD DEL SOFTWARE

Mediante la métrica de facilidad de instalación se calcula el factor de portabilidad, mediante este se obtiene el porcentaje de éxitos de instalación de los usuarios responsables de realizar la instalación del sistema.

4.7 FACILIDAD DE INSTALACION

La facilidad de instalación viene dada de la siguiente relación:

$$X = A / B$$

A=Numero de casos de éxitos de la operación innatación por parte del usuario.

B= Numero total de operaciones de instalación que realizo el usuario.

Entonces se tiene que:

$$X = 8 / 10 = 0,8$$

Por tanto existe un 80% de probabilidad que el usuario instale exitosamente el sistema y de acuerdo a la escala que se muestra en la parte inferior se concluye que el sistema es portable.

75% ≤ X ≤ 100% → Óptima

50% ≤ X < 75% → Buena

25% ≤ X < 50% → Suficiente

0% ≤ X < 25% → Deficiente

Aplicación de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido

Las ecuaciones del CÓCOMO básico tienen la siguiente forma:

$$E = a_b (KLDC)^{b_b} \dots\dots\dots (Ecuación 2)$$

$$D = c_b (E)^{d_b} \dots\dots\dots (Ecuación 3)$$

Donde:

E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en meses cronológicos

KLDC: Número estimado de líneas de código distribuidas (en miles)

Tabla 4.8 Coeficientes a_b y c_b y los exponentes b_b , y d_b

Proyecto de Software	a_b	c_b	b_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi - acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: [PRE,2000]

Son proyectos de software semiacoplados, los proyectos intermedios (en tamaño y complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidos, tal es el caso del software desarrollado.

$$E = 3.0 * (26,06)^{1.12}$$

$$D = 2.5 * (115,6)^{0.35}$$

$$E = 115,6$$

$$D = 13.18$$

El personal requerido, en este caso el número de programadores se obtiene con la siguiente fórmula:

$$\text{Número de programadores} = E / D$$

$$\text{Número de programadores} = 115,6 / 13,14$$

$$\text{Número de programadores} = 8,77 = 9$$

El salario de un programador puede oscilar entre los \$us 200, cifra que será tomada en cuenta para la estimación siguiente:

Costo del software desarrollado = Número de programadores * salario de un programador

$$\text{Costo del software desarrollado} = 9 * 200$$

$$\text{Costo del Software desarrollado} = \$us 1800$$

4.8 Costo de Elaboración del Proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de análisis y recopilación principalmente, estos costos se representan en la siguiente tabla

Tabla 4.9 Costo de Elaboración del Proyecto

DETALLE	IMPORTE (\$us)
Análisis y Diseño del	300
Bibliografía	50
Material de Escritorio	30
Internet	40
Otros	10
TOTAL	430

Fuente: Elaboración Propia

4.8.1 Costo Total

El costo total es la sumatoria del costo del software desarrollado, costo de elaboración del proyecto, detallados en la tabla 4.9

Tabla 4.10 Costo Total del Proyecto

DETALLE	IMPORTE (\$us)
Costo del Software Desarrollado	1800
Costo de Elaboración del Proyecto	430
TOTAL	2230

Fuente: Elaboración Propia

De acuerdo al análisis de costo de COCOMO el presente proyecto de grado tiene un costo total de 2230 \$us



CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Terminando el proceso de construcción del sistema, se concluye que:

- A la culminación de la elaboración, análisis, diseño e implementación, se concluyó satisfactoriamente, llegando a los objetivos planteados.
- Se obtuvo un producto con las particularidades de un sistema de información, que contribuye como un aporte tecnológico. El sistema ha logrado adaptarse satisfactoriamente a los requerimientos de Empresa (Y.P.F.B.) con la que se trabajó, teniendo la capacidad de trabajar de manera local como a través de Intranet.
- La automatización de la información, facilita la organización de la misma, agilizando los procesos de control de personal de toda la empresa. En especial en el control de asistencia ya que anteriormente se realizaba las planillas de asistencia del personal mas o menos en un rango de 3 a 5 días perjudicando a la elaboración de planillas de sueldos, con el SICAP este proceso se realiza en un máximo de 2 días.
- Se desarrolló el sistema mediante la nueva corriente de metodologías que son las ágiles, en este caso usando ASD, gracias a su ciclo de vida permitió realizar iteraciones mas veloces, lo que favoreció que se pueda atender todos los requerimientos en menor tiempo, tomando en cuenta que el trabajo fue realizado de manera individual.
- El desarrollo del sistema logró apoyarse eficientemente en las herramientas de modelado que nos brinda el Lenguaje de Modelado Unificado UML con el cual se especifico el análisis y diseño del sistema.

- Debido al diseño, el sistema trabaja tranquilamente en Intranet dando la facilidad a la empresa en el manejo de la información, integrada y centralizada en un solo sistema.
- Debido a la seguridad en la autenticación del sistema, este esta libre de intercepciones de usuarios no admitidos, así logrando una optima seguridad en el ingreso y código del sistema esto se llevo acabo utilizando el algoritmo de encriptación SHA-1.
- El sistema SICAP se realizo en un entorno intranet y no via web debido a la seguridad de la información de la empresa.

5.2 RECOMENDACIONES

- El sistema ha tomado en cuenta el área de R.R.H.H. y el de administración de personal de la empresa, pero no el área contable ni el área de presidencia se recomienda diseñar estos módulos faltantes.
- Este sistema fue realizado en un entorno cliente servidor vía intranet y no Internet por la cual no tiene un acceso directo o en línea con el servidor central de YPFB el cual se encuentra en la ciudad de santa cruz, se recomienda realizar un sistema vía Internet que pueda unir al SISCAP con el servidor central para una mejor integridad y eficiencia.
- Para la consulta de datos personales, se recomienda realizar un sistema vía Web para que el personal de Y.P.F.B. pueda acceder a sus datos personales en la empresa desde cualquier lugar y a cualquier hora esto rigiéndose por políticas de la empresa.

BIBLIOGRAFÍA

- [JHC,2000] José H. Canós, Patricio Letelier y Carmen Penadés, "Metodologías Ágiles en el Desarrollo de Software" ,Universidad Politécnica de Valencia,2000
- [PRE,2000] Roger S. Presman, R. 2000: INGENIERIA DEL SOFTWARE UN ENFOQUE PRACTICO, Quinta Edicion, Editorial McGRAW-HILL, Mexico
- [JAC,1998] Ivar Jacobson, Grady booch, James Rumbaugh ,2000 "PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE"
- [COT,1994] Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13.
- [OMG,1999] Object Management Group. 1999. OMG Unified Modeling Language Specification (Draft). Versión 1.3. alfa R5, marzo de 1999. Disponible en: <http://www.rational.com/uml>
- [MAS,2001] *Manifesto for Agile Software Development*, Disponible en: <http://www.agilealliance.org/>

ANEXO A

Caso de Prueba

Numero de Caso de Prueba: 1

Numero Historia de Usuario: 1

Nombre de Caso de Prueba: Registra datos (caso de uso de afiliación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción para registrar datos, el sistema nos mostrara una pantalla para llenar con todos los datos para registrar de acuerdo a que operación se este realizando una vez realizado el registro de datos este se la almacenara en la base de datos.

Condiciones de Ejecución: las condiciones de ejecución son que el usuario u administrador

Este autorizado para realizar el registro de datos de dicha operación, una vez comprobado eso se puede pasar a digitar los datos solicitados para efectuar el registro de los mismos.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de afiliación.
- El encargado introduce en cada celda los datos correspondientes a cada una.
- El encargado pulsa el botón 'guardar'.

Resultado esperado: los datos introducidos por el encargado se almacena en la base de datos y son almacenados ahí para su posterior uso.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 2

Numero Historia de Usuario: 2

Nombre de Caso de Prueba: Verifica registro (caso de uso de afiliación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de verifica registro, el cual se almaceno anteriormente, el sistema nos mostrara una lista de los datos almacenados anteriormente para poder verificar que se halla almacenado correctamente todo los datos.

Condiciones de Ejecución: las condiciones de ejecución son que el usuario u administrador

Este autorizado para realizar la verificación de los registros de la información.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de consulta de afiliación para verificación de los registro.
- Pulsamos verifica registro.
- Nos aparecerá en la misma pantalla todos los datos registrados para su respectiva verificación.

Resultado esperado: los datos introducidos por el encargado se almacena en la base de datos y son almacenados ahí para su posterior uso..

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 3

Numero Historia de Usuario: 3

Nombre de Caso de Prueba: Genera Certificados de Antigüedad (caso de uso de afiliación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de generar certificado de antigüedad, el sistema nos mostrara una pantalla del modelo de impresión que se desea realizar.

Condiciones de Ejecución: las condiciones de ejecución son que el usuario u administrador

Este autorizado para realizar la verificación de los datos que se ban a imprimir .

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar “aceptar” de la aplicación
- Nos aparece la interfaz propia de afiliación.
- El encargado pulsa el botón imprimir certificado, nos aparecerá una vista de la información que se va a realizar

Resultado esperado: los datos introducidos por el encargado se almacena en la base de datos y son almacenados ahí para su posterior uso.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba:4

Numero Historia de Usuario: 4

Nombre de Caso de Prueba: Emisión de memorando (caso de uso de afiliación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción para realizar la emisión de memorando, el sistema nos mostrara el formato del memorando con los datos correctos pedidos, y eligiremos la opción de emisión de memorando.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de dicha operación, una ves comprobado eso se puede acceder a emitir el memorando correspondiente.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar “aceptar” de la aplicación.
- Nos aparece la interfaz propia de afiliación.

- El administrador selecciona la opción de emisión de reportes.

Resultado esperado: El formato de emisión de memorando este correcto y alineado con los

datos correctos y bien elaborados

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 5

Numero Historia de Usuario: 5

Nombre de Caso de Prueba: Registra permisos (caso de uso de asistencia)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción para realizar el registro de permisos, el sistema nos mostrara una pantalla para poder registrar los datos de permisos correspondientes de asistencia.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el

usuario administrativo este autorizado para realizar el registro de datos de dicha operación, una vez comprobado eso se puede acceder al registro de los datos.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de asistencia.
- Seleccionamos la opción de registrar permisos d asistencia.
- Nos aparece una interfaz para introducir los datos solicitados.
- El encargado pulsa el botón 'guardar'.

Resultado esperado: Los datos registrados se almacenan en la base de datos y se guardan ahí para su posterior uso.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 6

Numero Historia de Usuario: 6

Nombre de Caso de Prueba: Emisión de reportes de asistencia (casos de uso de asistencia).

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción para realizar la emisión de reportes de asistencia, el sistema nos mostrara el formato para la emisión de reporte con todos los datos adjuntos correspondientes.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el

usuario administrativo este autorizado para realizar la emisión de reporte, una vez comprobado eso se puede acceder ala emisión de los reportes de asistencia.

Entradas:

- El encargado u usuario entra al sistema.

- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar “aceptar” de la aplicación.
- Nos aparece la interfaz propia de asistencia.
- El administrador pulsa la opción de emisión de reportes, y emite el reporte correspondiente.

Resultado esperado: Emisión correcta del formato de asistencia con los datos solicitados.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 7

Numero Historia de Usuario:7

Nombre de Caso de Prueba: exportación de datos de asistencia (caso de uso de asistencia)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción para realizar la exportación de datos de asistencia, el sistema nos mostrara una interfaz para realizar la exportación de datos

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el

usuario administrativo este autorizado para realizar la exportación de los datos de asistencia, una vez comprobado eso se puede acceder a la emisión u exportación de datos.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar “aceptar” de la aplicación.
- Nos aparece la interfaz propia de asistencia.
- El administrador oprimirá la opción de exportación de datos de asistencia

Resultado esperado: exportar satisfactoriamente todos los datos solicitado

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 9

Numero Historia de Usuario: 9

Nombre de Caso de Prueba: registra datos de vacación (caso de uso de vacación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de registrar datos, el sistema nos mostrara una interfaz para realizar el registro de la informaron correspondiente.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el

usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder al registro de datos de vacación.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar “aceptar” de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de registrar datos

Resultado esperado: almacenamiento corrector de los datos registrados.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 10

Numero Historia de Usuario: 10

Nombre de Caso de Prueba: verifica lapso de vacación (caso de uso de vacación)

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de verificación de lapso de vacación, el sistema nos mostrara una interfaz para realizar la verificación requerida por el usuario.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una ves comprobado eso se puede acceder a la verificación correspondiente.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de verificación.
- El sistema nos muestra los datos a verificar o solicitados.

Resultado esperado: verificación correcta de los datos.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 11

Numero Historia de Usuario: 11

Nombre de Caso de Prueba: Emisión de boleta de vacación

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de emisión de boleta de vacación, el sistema nos mostrara una interfaz para realizar u observar el formato de la boleta a emitir.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una ves comprobado eso se puede acceder a la emisión correspondiente.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de emisión de boleta de vacación.
- El sistema nos mostrara una interfaz con el modelo de boleta a emitir.

Resultado esperado: emisión correcta de la boleta de vacación.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 12

Numero Historia de Usuario: 12

Nombre de Caso de Prueba: Registra instrucción de viático

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de registro de datos, el sistema nos mostrara una interfaz para realizar el registro correspondiente.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a realizar el registro correspondiente pedido por el usuario.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de registrar datos de viático.
- El sistema nos mostrara una interfaz para registrar los datos correspondientes.

Resultado esperado: almacenamiento correcto de la instrucción de viático solicitado.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 13

Numero Historia de Usuario: 13

Nombre de Caso de Prueba: Verificación de planilla de instrucción.

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de registro de datos, el sistema nos mostrara una interfaz para realizar la verificación de la instrucción solicitada.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a la optima verificación de la planilla de instrucción de viático.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de verificación de planilla.
- El sistema nos mostrara todos los datos a verificar que se solicitaron.

Resultado esperado: verificación de la planilla de viático en orden.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 14

Numero Historia de Usuario: 14

Nombre de Caso de Prueba: Emisión de planilla de viático.

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de emisión de planilla, el sistema nos mostrara una interfaz para realizar la emisión de planilla de viático antes de ser emitida u impresa.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a la emisión de la planilla.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de emisión de planilla de viático.
- El sistema nos mostrara una interfaz que nos permitirá ver un modelo de planilla antes de ser emitida o imprimida.

Resultado esperado: emisión correcta de planilla de viático.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 15

Numero Historia de Usuario: 15

Nombre de Caso de Prueba: Control y registro de facturas IVA

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de control y registro de facturas iva, el sistema nos mostrara una interfaz para realizar el registro de las facturas para su control esperado.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a registrar los datos de las facturas solicitados.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de registrar facturas iva.
- El sistema nos mostrara una interfaz para introducir los datos que se registraran.

Resultado esperado: almacenamiento correcto de los datos de las facturas iva y un control eficiente del mismo.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 16

Numero Historia de Usuario: 16

Nombre de Caso de Prueba: elaboración de planilla de sueldos

Descripción: : Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de elaboración de planilla de sueldos, el sistema nos mostrara una interfaz para realizar la elaboración de la planilla correspondiente.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a la opción de elaboración de planilla de sueldos.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de elaboración de planilla de sueldos.
- El sistema muestra una interfaz para la elaboración de la planilla solicitada.

Resultado esperado: una óptima elaboración de la planilla de sueldos.

Evaluación: Prueba superada con éxito

Caso de Prueba

Numero de Caso de Prueba: 17

Numero Historia de Usuario: 17

Nombre de Caso de Prueba: Registra información.

Descripción: Nos introducimos al sistema como administradores, accediendo a su

Funcionalidad y solicitamos la opción de registrar información, el sistema nos mostrara una interfaz para realizar el registro correspondiente.

Condiciones de Ejecución: las condiciones de ejecución del caso de prueba son que el usuario administrativo este autorizado para realizar el registro de datos de vacación, una vez comprobado eso se puede acceder a realizar el registro solicitado.

Entradas:

- El encargado u usuario entra al sistema.
- El encargado introduce nombre y contraseña personal.
- Pulsamos entrar "aceptar" de la aplicación.
- Nos aparece la interfaz propia de vacación
- El administrador oprimirá la opción de registro de datos.
- El sistema le mostrara una interfaz para que introduzca los datos que va a registrar.

Resultado esperado: almacenamiento correcto de los datos registrados.

Evaluación: Prueba superada con éxito

ANEXO B

CALENDARIO DEL PROYECTO

Disciplina/ artefactos generados o modificados durante la fase de inicio

Modelado del negocio

- modelo de casos de uso del Negocio

Comienzo

Semana 1
20/9/2006-28/9/2006

Aprobación

Semana 3
1/10/2006-6/11/2006

Requisitos

- Glosario

Semana 1
20/9/2006-28/9/2006

Semana 3
2/10/2006-6/11/2006

- Visión

Semana 2
25/9/2006-27/9/2006

Semana 3
1/10/2006-6/11/2006

- Modelo de casos de uso

Semana 3
1/10/2006-6/11/2006

Siguiente fase

- Especificación de casos de uso

Semana 3
1/10/2006-6/11/2006

Siguiente fase

- Especificación adicionales

Semana 3
1/10/2006-6/11/2006

Siguiente fase

Análisis y diseño

- Modelado de análisis y diseño

Semana 4
11/11/2006-20/11/2006

Siguiente fase

- Modelado de datos

Semana 4
11/11/2006-20/11/2006

Siguiente fase

Implementación

- Prototipos de interfaces de usuario

Semana 4
20/11/2006-30/11/2006

Siguiente fase

- Modelo de implementación

Semana 4
20/11/2006-30/11/2006

Siguiente fase

Pruebas

- Casos de pruebas funcionales

Semana 5
24/11/2006-10/12/2006

Siguiente fase

Despliegue

- Modelo de despliegue

Semana 5
24/11/2006-10/12/2006

Siguiente fase

Gestión de cambios y configuración

Gestión de proyectos

- Plan de desarrollo del software en su versión 1.0 y planes de las iteraciones

Durante todo el proyecto
Semana 5
24/11/2006-10/12/2006

Siguiente fase

Ambiente

Durante todo el proyecto

Disciplina/ artefactos generados o modificados durante la fase de elaboración

Modelado del negocio

- modelo de casos de uso del

Comienzo

Semana 1

Aprobación

Aprobado

Negocio	20/9/2006-28/9/2006	
Requisitos		
-Glosario	Semana 1 20/9/2006-28/9/2006	Aprobado
-Visión	Semana 2 24/9/2006-27/9/2006	Aprobado
-Modelo de casos de uso	Semana 10 16/01/2007-24/01/2007	Semana 12 1/02/2007-8/02/2007
-Especificación de casos de uso	Semana 11 20/01/2007-29/01/2007	Semana 13 8/02/2007-15/02/2007
-Especificación adicionales	Semana 11 19/01/2007-29/01/2007	Semana 13 8/02/2007-15/02/2007
Análisis y diseño		
-Modelado de análisis y diseño	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
-Modelado de datos	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
Implementación		
-Prototipos de interfaces de usuario	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
-Modelo de implementación	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
Pruebas		
-Casos de pruebas funcionales	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
Despliegue		
-Modelo de despliegue	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
Gestión de cambios y configuración	Durante todo el proyecto	
Gestión de proyectos		
-Plan de desarrollo del software en su versión 1.0 y planes de las iteraciones	Semana 13 8/02/2007-15/02/2007	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Disciplina /artefactos generados o

modificados durante la fase de construcción

Comienzo

Aprobación

(iteración 1)

Casos de uso negociados para el primer lanzamiento

Afiliación de nuevo personal(gestión de afiliación)	1/10/2006	30/11/2006
Consultas y reportes de afiliación	1/10/2006	30/11/2006

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 2)

Casos de uso negociados para el primer lanzamiento

Afiliación de nuevo personal(gestión de afiliación)	12/12/2006	Aprobado
Consultas y reportes de afiliación	12/12/2006	Aprobado

Casos de uso negociados para el próximo lanzamiento

Realizar solicitud de vacación (gestión de vacación)	26/01/2007	14/02/2007
Consulta y reportes de vacaciones (gestión de vacación)	26/01/2007	14/02/2007
Reformulación de boletas de vacación (gestión de vacación)	26/01/2007	14/02/2007

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 1)

Casos de uso negociados para el primer lanzamiento

Realizar solicitud de vacación (gestión de vacación)	26/01/2007	14/02/2007
Consulta y reportes de vacaciones (gestión de vacación)	26/01/2007	14/02/2007
Reformulación de boletas de vacación (gestión de vacación)	26/01/2007	14/02/2007

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 2)

Casos de uso negociados para el primer lanzamiento

Realizar solicitud de vacación (gestión de vacación)	16/02/2007	Aprobado
Consulta y reportes de vacaciones (gestión de vacación)	16/02/2007	Aprobado
Reformulación de boletas de vacación (gestión de vacación)	16/02/2007	Aprobado

de vacación)

Casos de uso negociados para el próximo lanzamiento

Asignar viático (gestión de viático)	23/02/2007	12/03/2007
Consultas y reportes de viáticos (gestión de viático)	23/02/2007	12/03/2007

Disciplina /artefectos generados o

modificados durante la fase de construcción (iteración 1)

Comienzo

Aprobación

Casos de uso negociados para el primer lanzamiento

Asignar viático (gestión de viático)	23/02/2007	12/03/2007
Consultas y reportes de viáticos (gestión de viático)	23/02/2007	12/03/2007

Disciplina /artefectos generados o

modificados durante la fase de construcción (iteración 2)

Comienzo

Aprobación

Casos de uso negociados para el primer lanzamiento

Asignar viático (gestión de viático)	14/03/2007	Aprobado
Consultas y reportes de viáticos (gestión de viático)	14/03/2007	Aprobado

Casos de uso negociados para el próximo lanzamiento

Control de asistencia (gestión de asistencia)	20/03/2007	04/04/2007
Consultas y reportes de asistencia (gestión de asistencia)	20/03/2007	04/04/2007

Disciplina /artefectos generados o

modificados durante la fase de construcción (iteración 1)

Comienzo

Aprobación

Casos de uso negociados para el primer lanzamiento

Control de asistencia (gestión de asistencia)	20/03/2007	04/04/2007
Consultas y reportes de asistencia (gestión de asistencia)	20/03/2007	04/04/2007

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 2)

Casos de uso negociados para el primer lanzamiento

Control de asistencia (gestión de asistencia)	06/04/2007	13/04/2007
Consultas y reportes de asistencia (gestión de asistencia)	06/04/2007	13/04/2007

Casos de uso negociados para el próximo lanzamiento

Control de planillas (gestión de planillas)	27/04/2007	11/05/2007
Consultas y reportes de planillas (gestión de planillas)	27/04/2007	11/05/2007

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 3)

Casos de uso negociados para el primer lanzamiento

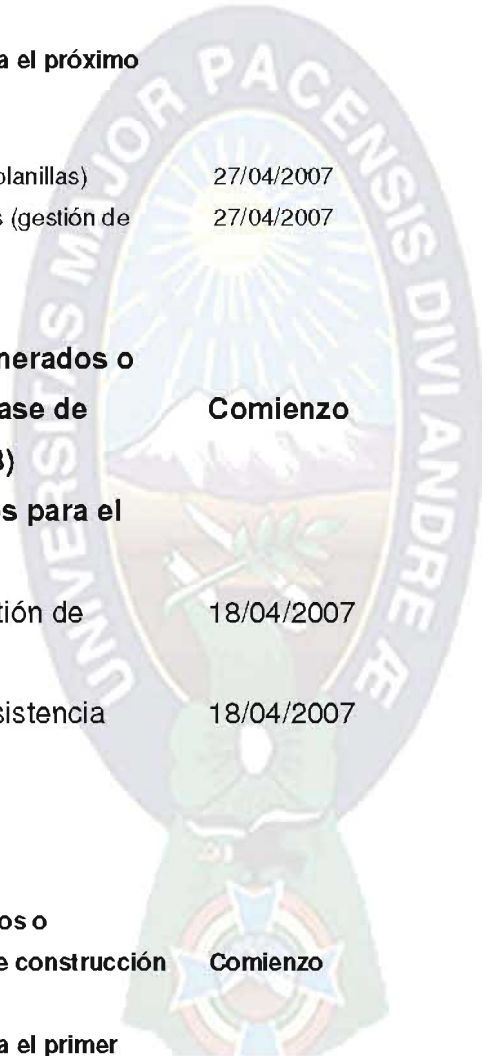
Control de asistencia (gestión de asistencia)	18/04/2007	Aprobado
Consultas y reportes de asistencia (gestión de asistencia)	18/04/2007	Aprobado

Disciplina /artefactos generados o modificados durante la fase de construcción (iteración 1)

Casos de uso negociados para el primer lanzamiento

Control de planillas (gestión de planillas)	27/04/2007	11/05/2007
Consultas y reportes de planillas (gestión de planillas)	27/04/2007	11/05/2007

Disciplina /artefactos generados o modificados durante la fase de construcción



(iteración 2)

Casos de uso negociados para el primer lanzamiento

Control de planillas (gestión de planilla)	14/05/2007	25/05/2007
Consultas y reportes de planillas (gestión de planilla)	14/05/2007	25/05/2007

Casos de uso negociados para el próximo lanzamiento

Control de finiquitos (gestión de finiquitos)	14/05/2007	31/05/2007
Consultas y reportes de finiquitos (gestión de finiquitos)	14/05/2007	31/05/2007

Disciplina /artefactos generados o modificados durante la fase de construcción **Comienzo** **Aprobación**
(iteración 3)

Casos de uso negociados para el primer lanzamiento

Control de planillas (gestión de planillas)	01/06/2007	Aprobado
Consultas y reportes de planillas (gestión de planillas)	01/06/2007	Aprobado

Disciplina /artefactos generados o modificados durante la fase de construcción **Comienzo** **Aprobación**
(iteración 2)

Casos de uso negociados para el primer lanzamiento

Control de finiquitos (gestión de finiquitos)	8/06/2007	Aprobado
Consultas y reportes de finiquitos (gestión de finiquitos)	8/06/2007	Aprobado

Casos de uso negociados para el próximo lanzamiento


Implementación y prestación final del sistema SICAP	15/06/2007	Aprobado
Pruebas del sistema SICAP	15/06/2007	Aprobado

HISTORIAL DE REVISIONES

FECHA	VERSION	DESCRIPCION	AUTOR
8/05/2007	0.1	Versión inicial de la fase de desarrollo del marco teórico	Claudia Apaza
21/05/2007	0.2	Versión segunda de la fase de marco teórico y inicio de análisis	Claudia Apaza
4/06/2007	0.1	Versión inicial de la fase de análisis y desarrollo del sistema	Claudia Apaza
8/06/2007	0.2	Versión final de la fase de análisis y desarrollo, inicio de pruebas de calidad	Claudia Apaza
11/06/2007	0.1	Versión inicial y casi concluida de la fase de pruebas	Claudia Apaza
20/06/2007	0,2	Versión final de la presentación del documento terminado	Claudia Apaza




REPORTE DE AFILIACION DE UN NUEVO PERSONAL

		PRS - VPNC		O R I G E N	SIGLA :		
		FILIACIÓN Y CONTRATO			NUMERO :		
					FECHA :		
PACHECO ----- APELLIDO PATERNO	PACHECO ----- APELLIDO MATERNO	HECTOR WILFREDO ----- NOMBRES		<div style="border: 1px solid black; width: 100%; height: 100%;"></div>			
LA PAZ ----- LUGAR DE NACIMIENTO	BOLIVIA ----- PAIS	BOLIVIA ----- PAIS CIUDAD	13/08/1975 ----- FECHA				
CASADO ----- ESTADO CIVIL	4849008-LP ----- CARNET	L. MILITAR ----- LIB. SERVIC. MILITAR					
LA PAZ ----- LUGAR DE CONTRATACIÓN	05/07/2007 16:01:49 ----- FECHA DE CONTRATACIÓN	05/07/2007 ----- FECHA DE INGRESO					
OFICIAL ----- CARGO INICIAL	COORDINACION ----- UNIDAD	3000 ----- HABER BASICO					
INDEFINIDO ----- CLASE CONTRATO	NN ----- DOMICILIO	0 ----- TELEFONO				CORREO	
FECHA ----- FECHA EXAMEN MEDICO	LUGAR DE ----- LUGAR	INFORME DEL MEDICO ----- SIN TESIS DEL INFORME				CASILLA DE CORREO ----- ESPECIALISTA ----- MEDICO CONSULTADO	

DEPENDIENTES DIRECTOS					
NOMBRE Y APELLIDO	FECHA			PARENTESC	DOCUMENTO PRESTADO
	DIA	MES	AÑO		
-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----

EDUCACION (Grado Alcanzado o Titulo)				
ELEMENTAL	SECUNDARIA	COMERCIAL O	UNIVERSITARIA	POST.
Etiqueta 1655	Etiqueta 1655	Etiqueta 1655	Etiqueta 1655	Etiqueta 1655
ANTECEDENTES DE TRABAJO				
EMPRESA	CARGO	DESDE	HASTA	MOTIVO DE
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----
IDIOMAS	CONVERSACION	ESCRITURA	COMPRENCION	OBSERVACIONES
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----
INDICAR : BUENA -				
MAQUINAS O EQUIPOS QUE MANEJA			REGISTRO	
-----			Etiqueta 1655	
-----			Etiqueta 1657	
INTERESADO			DIRECCION NAL.	
-----			-----	

REPORTE GERENCIAL DE GASTOS

		VICEPRESIDENCIA DE ADMINISTRACION Y FINANZAS					
GNAF - DNRH		INFORME GERENCIAL DE GASTOS (REPORTE MENSUAL)				CORRESPONDIENTE AL MES : MAY/06	
GERENCIA	VPAF						
DISTRITO	LA PAZ						
Cuenta Objetivo	Concepto	Frecuencia No Personal	Personal Profesional	Frecuencia No Personal	Personal No Profesional	Total Frecuencia	Total Importe
T T	TOTAL GENERAL :						
00	REMUNERACIONES :						
1.1	Haberes		0,00		0,00	0	0,00
1.2	Básico						0,00
1.3	Bonif. Antigüedad/Niv. Experiencia						0,00
1.4	Horas Extras						0,00
1.5	Horas Nocturnas No						
1.6	Pagos Adicionales						
1.7	Supletorias						
1.8	Retenciones						
1.9	Promedio Vocacional						
1.10	Subsidio de Frontera						
1.11	Convenio S. L. 26 (15%) CAMIRI						
02	CARGAS SOCIALES :						
2.1	Aguinaldo gestión 2007						
2.2	Puna unidades personal relacionado						
2.3	Puna sobre unidades Gestión 2001						
05	OTROS BENEFICIOS :		0,00		0,00		0,00
5.1	Servicio de Té					0	0,00
5.2	Fallas de Caja (Transporte)					0	0,00
06	GASTOS DE VIAJE :		0,00		0,00		0,00
6.1	Viático Único de Campo						
6.2	Viaticos por Comisión					0	0,00
6.3	Viaticos por Transferencia						
6.4	Pasajes por Comisión						0,00
6.5	Pasajes por Transferencia						
6.6	Transp. Ens P/Ret. y Transf.						0,00
NA	NUMERO DE PERSONAL EN EL MES:		0		0		0
	Varones						0
	Mujeres						0
NB	Numero de personal mes anterior		0		0		0
	Varones						0
	Mujeres						0
AA	Altas						0
AB	Contratos						0
AC	Transferencias						0
AD	Traspaso a Sector Profesional						0
BA	Bajas						0
BB	Retiros Forzosos						0
BC	Retiros Voluntarios						0
BD	Jubilaciones						0
BE	Traspaso a sector Profesional						0
BF	Transferencias						0
BG	TOTAL :						

REPORTE DE HOJA DE COMISION

		HOJA DE COMISIONES		D.V.P.R.
<i>Nombre y cargo del empleado:</i>				
<i>Comisión conferida por. (Nombre y Cargo)</i>		LIC. MARIELA SIMBRON PALACIOS - DNRH		
<i>Con Objeto de:</i>		COORDINAR EL TRABAJO DEL SISTEMA DE PLANILLAS		
<i>En el lugar o Lugares de:</i>				
<i>Tiempo que durará la comisión</i>		2 días, a partir del	7 de	JULIO
			de	2005
<i>Lugar y Fecha:</i> LA PAZ 1 DE JULIO DE 2005				
1	<i>Firma del Comisionado</i>	2	<i>Firma Autoridad Conferente</i>	
3	<i>Jefe de Personal</i>	4	<i>Vo Bo Gerente General, Gerente Jefe de División o Superintendente</i>	
<i>Fecha y hora retorno de la comisión</i>				
<i>Indicar si recibió alojamiento</i>				
<i>Informe Resumen labores cumplidas</i>				

REPORTE DE FINIQUITO Y TRANSFERENCIA

Y. P. F. B.	ESTADO ECONOMICO PARA	ORIGEN	Sigla :	DRH
Direcc. De Serv. Generales			Numero:	
Div. ADMIN - PERSONAL	FINIQUITO Y TRANSFERENCIA		Lugar :	LA PAZ
			Fecha :	12/dic/2005

Rogamos a los Señores Jefes de las DIVISIONES, REPARTICIONES Y ENTIDADES, mencionadas en el caminamiento, se sirvan informar a la DIRECCION DE SERVICIOS GENERALES, para los trámites consiguientes, las obligaciones pendientes del:

SEÑOR: DAGMAR MERINO RODO

que ha sido retirado de la Entidad

que ha sido transferido a la localidad de:


en fecha : 12 de diciembre de 2005

Informante	No Cuenta	Importe	Concepto	Fecha y Firma

1	División de Administración de Personal	2	Gerente de Personal
	Favor Informar		Confirme
HORA		HORA	
FECHA		FECHA	

Caminamiento:	DVCB	<input type="checkbox"/>	DVRM	<input type="checkbox"/>	DVTS	<input type="checkbox"/>	DVPP	<input type="checkbox"/>	DVSM	<input type="checkbox"/>
	DVAD	<input type="checkbox"/>	Entre Rios	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>
			Club Petrolero	<input type="checkbox"/>			Com. De Vivien. Pet. Est.	<input type="checkbox"/>		<input type="checkbox"/>

REPORTE DE PAPELETA DE PERMISO

 Form Y-90-430-22	PAPELETA DE PERMISO LICENCIA O COMISION		La Paz 24 de noviembre 2005		
	LOCAL		Lugar y Fecha		
HECTOR WILFREDO PACHECO Nombre Completo del Trabajador					
Cargo que Ocupa:	OFICIAL ADM. III	Repartición:	DRH		
Permiso	<input checked="" type="checkbox"/>	Lic. Sin Remuneración	<input type="checkbox"/>	Lic. Cargo Vacación	<input type="checkbox"/>
				Comisión Local	<input type="checkbox"/>
Solicita por:	1	días		horas	minutos
desde	25-nov-05		hasta	25-nov-05	
Motivo	FAMILIARES				
Salida a Hrs		del		de	de 200...
Retorno Hrs.		del		de	de 200...
Encargado Portería					
Observaciones:	COMPENSACION POR HORAS TRABAJADAS EN APLICACIÓN A LA DISPOSICION ADMINISTRATIVA No 418/2002 PUNTO 1,5				
1	Interesado	2	Jefe de Repartición		
Hora		Hora			
Fecha		Fecha			
3	Encarg. Control Asistencia	4	Vo Bo Jefe de Personal		
Hora		Hora			
Fecha		Fecha			

REPORTE DE PLANILLA DE VIATICO

Y.P.F.B Form. Y.158-430-37 **PLANILLA DE VIATICOS** No OFPR- 5 - 07/07/200

Nombre y apellido **HECTOR WILFREDO PACHECO PACHECO** SEGUN BB
 Cargo **OFICIAL ADMINISTRATIVO III**
 Gerencia **GERENCIA DE ADMINISTRACION Y FINA** Centro de costos **DOLAR** FECHA **8**
 Por su viaje a: **SANTA CRUZ** \$us 128 X 3 DIA(S)=\$us 384
 Motivo **REUNION**

Viático diario segun

128	%	100	%
	%		

Fecha	Recibido en ..	N° C.P.	Pagado del.....al.....	Importe	Apropiacion Contable	Firma Autorizada
10/07/2007	LA PAZ		11/07/2007 13/07/2007	564		
				Bs 384		
TOTAL				384		

LIQUIDACIÓN

Tiempo de Comisión

Recibido según planilla viáticos.

Derecho AASANA

Otros

SUMAS

Saldo a FAVOR / EN CONTRA del Trabajador

APROPIACION CONTABLE	DEBE	HABER
SUMAS		

Liquidador
 Firma del Trabajador
 Aprobado

NOTA- De acuerdo a disposiciones administrativas vigentes en la Empresa, el trabajador esta obligado a entregar la presente planilla, al Departamento de personal, en el término de (5) dias una vez concluida la comisión; caso contrario el Departamento de contabilidad procederá a girar la nota de débito por el total de la comisión (viáticos, hotel, pasajes, etc.)

Encriptación Algoritmo SHA-1

Primero un poco de historia:

La familia **SHA** (*Secure Hash Algorithm*, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la **Agencia de Seguridad Nacional de los Estados Unidos** y publicadas por el *National Institute of Standards and Technology* (NIST). El primer miembro de la familia fue publicado en **1993** es oficialmente llamado **SHA**. Sin embargo, hoy día, no oficialmente se le llama **SHA-0** para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de **SHA-1**. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: **SHA-224**, **SHA-256**, **SHA-384**, y **SHA-512** (todos ellos son referidos como **SHA-2**).

En **1998**, un ataque a SHA-0 fue encontrado pero no fue reconocido para SHA-1, se desconoce si fue la NSA quien lo descubrió pero aumentó la seguridad del SHA-1.

SHA-1 ha sido examinado muy de cerca por la comunidad criptográfica pública, y no se ha encontrado ningún ataque efectivo. No obstante, en el año **2004**, un número de ataques significativos fueron divulgados sobre funciones criptográficas de *hash* con una estructura similar a SHA-1; esto ha planteado dudas sobre la seguridad a largo plazo de SHA-1.

SHA-0 y SHA-1 producen una salida resumen de 160 bits de un mensaje que puede tener un tamaño máximo de 2^{64} bits, y se basa en principios similares a los usados por el profesor **Ronald L. Rivest** del MIT en el diseño de los algoritmos de resumen del mensaje **MD4** y **MD5**.

código para encriptar con este algoritmo:

Public Function SHA1(sMessage As String) As String

Dim HASH(4) As Long

Dim M() As Long

Dim W(79) As Long

Dim a, b, c, d, e As Long

Dim g, h, i, j As Long

Dim T1, T2 As Long

HASH(0) = &H67452301

HASH(1) = &HEFCDAB89

HASH(2) = &H98BADCFE

HASH(3) = &H10325476

HASH(4) = &HC3D2E1F0

M = ConvertToWordArray(sMessage)

For i = 0 To UBound(M) Step 16

 a = HASH(0)

 b = HASH(1)

 c = HASH(2)

 d = HASH(3)

 e = HASH(4)

 For g = 0 To 15

 W(g) = M(i + g)

 Next g

 For g = 16 To 79

 W(g) = LRot(W(g - 3) Xor W(g - 8) Xor W(g - 14) Xor W(g - 16), 1)

 Next g

 For j = 0 To 79

 If j <= 19 Then

 T1 = (b And c) Or ((Not b) And d)

 T2 = &H5A827999

 Elseif j <= 39 Then

 T1 = b Xor c Xor d

 T2 = &H6ED9EBA1

 Elseif j <= 59 Then



```

    T1 = (b And c) Or (b And d) Or (c And d)
    T2 = &H8F1BBCDC
Elseif j <= 79 Then
    T1 = b Xor c Xor d
    T2 = &HCA62C1D6
End If
h = AddUnsigned(AddUnsigned(AddUnsigned(AddUnsigned(LRot(a, 5), T1), e),
T2), W(j))
    e = d
    d = c
    c = LRot(b, 30)
    b = a
    a = h
Next j
HASH(0) = AddUnsigned(a, HASH(0))
HASH(1) = AddUnsigned(b, HASH(1))
HASH(2) = AddUnsigned(c, HASH(2))
HASH(3) = AddUnsigned(d, HASH(3))
HASH(4) = AddUnsigned(e, HASH(4))

Next i
SHA1 = LCase(Right("00000000" & Hex(HASH(0)), 8) & _
    Right("00000000" & Hex(HASH(1)), 8) & _
    Right("00000000" & Hex(HASH(2)), 8) & _
    Right("00000000" & Hex(HASH(3)), 8) & _
    Right("00000000" & Hex(HASH(4)), 8))
End Function

```

Se utilizo este algoritmo por su amplia seguridad aprobada ya en la actualidad y su amplia aceptación de tamaño de cifrado.