

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA



PROYECTO DE GRADO

**“SISTEMA DE SEGUIMIENTO Y CONTROL PARA
COMPONENTES DE AERONAVES.**

CASO: FUERZA AÉREA BOLIVIANA (FAB)”

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

POR: LIMBER VLADIMIR ZENTENO HUARACHI
TUTOR: M. SC. ALDO RAMIRO VALDEZ ALVARADO
ASESOR: LIC. MARCELO GERMÁN ARQUIPA CHAMBI

LA PAZ – BOLIVIA
2014



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



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

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA

“A mis padres, Zenón Zenteno y Rufina Huarachi, que me dieron la fuerza para salir adelante, dándome su confianza incondicional, a mis queridos hermanos Steyner, Cristhiam, Marcia que siempre han estado apoyándome”

AGRADECIMIENTOS

A Dios, por haberme dado la fuerza y salud para culminar este paso importante en mi vida, e impulsarme para seguir adelante.

A mi tutor M. Sc. Aldo Ramiro Valdez Alvarado por haberme guiado y dado la enseñanza concluir el presente proyecto de grado.

A mi asesor Li. Marcelo Aruquipa Chambi, por haberme guiado en la realización de este proyecto de grado, dándome las respectivas correcciones para que el mismo sea correcto.

A la Fuerza Aérea Boliviana, por permitir aplicar mis conocimientos obtenidos en la universidad, y haberme brindado su apoyo en el acceso a la información.

A todos mis amigos y amigas de la carrera por haber compartido buenos y malos momentos en el transcurso de mi formación académica.

RESUMEN

El proyecto “SISTEMA DE SEGUIMIENTO Y CONTROL DE COMPONENTES PARA AERONAVES” fue desarrollado para la Fuerza Aérea Boliviana, la institución tiene como objetivo, en particular, realizar el control del ciclo de vida de sus aeronaves.

El presente proyecto de grado tiene como objetivo automatizar el proceso de registro y control de las horas y/o ciclos de vuelo efectuados por las aeronaves y sus componentes, que forman parte de la Fuerza Aérea Boliviana, dicha información procesada sirven a la institución para ayudar a la toma de decisiones.

Cabe mencionar que dicho proyecto fue desarrollado bajo la metodología OpenUp, una metodología de desarrollo ágil orientado a la WEB que permite el desarrollo de software de una forma iterativa proporcionando cuatro fases: inicio, elaboración, construcción y transición. El modelado utilizado para el análisis y diseño de software es el UWE, herramienta que proporciona una serie de diagramas que facilitan la comprensión del desarrollo del sistema.

Las herramientas a emplearse fueron: PHP como lenguaje de programación, Postgres como base de datos, CakePHP como framework para el desarrollo, dichas herramientas tienen la característica de ser software libre, requisito especificado por parte de la institución.

Se empleó el ISO 9126 como forma para medir el software, tomando en cuenta los siguientes factores: funcionalidad, confiabilidad, mantenibilidad y portabilidad, mismos que según formulas, el proyecto tuvo resultados satisfactorios.

Palabras claves: aeronaves, horas, ciclos, control, componentes.

ABSTRACT

The "MONITORING AND CONTROL SYSTEM FOR AIRCRAFT COMPONENTS" project was developed for the Bolivian Air Force, the institution aims in particular perform control lifecycle of its aircraft.

This graduation project aims to automate the process of registration and control of hours and / or flight cycles by the aircraft and its components, which are part of the Bolivian Air Force, the information processed serve the institution to help to decision-making.

It is worth mentioning that the project was developed under the OpenUP methodology, agile development methodology oriented WEB that allows software development in an iterative manner providing four phases: inception, elaboration, construction and transition.

The modeling used for the analysis and design of software is the UWE, a tool that provides a series of diagrams to facilitate understanding of system development. The tools used were: PHP programming language, Postgres as a database, CakePHP as a framework for development, these tools have the characteristic of being free software requirement specified by the institution.

ISO 9126 was used as a method to measure software, taking into account the following factors: functionality, reliability, maintainability and portability, same as formulas, the project had satisfactory results.

Keywords: aircraft, hours, cycles, control components.

INDICE

CAPITULO I	1
MARCO INTRODUCTORIO	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	2
1.2.1. INSTITUCIONALES.....	2
1.3. TRABAJOS SIMILARES EN LA CARRERA DE INFORMÁTICA	5
1.4. PROBLEMA	6
1.4.1. IDENTIFICACION DEL PROBLEMA	6
1.4.2. PROBLEMA CENTRAL	8
1.4.3. PROBLEMAS SECUNDARIOS	8
1.5. OBJETIVO GENERAL	9
1.6. OBJETIVOS ESPECIFICOS	9
1.7. LIMITES Y ALCANCES	9
1.7.1. LIMITES	9
1.7.2. ALCANCES	9
1.8. APOORTE	10
1.9. JUSTIFICACIÓN.....	10
1.9.1. JUSTIFICACIÓN TÉCNICA.....	10
1.9.2. JUSTIFICACIÓN OPERATIVA	10
1.9.3. JUSTIFICACIÓN ECONÓMICA	10
1.9.4. JUSTIFICACIÓN SOCIAL.....	11
1.10. MÉTODOS DE INVESTIGACIÓN Y METODOLOGÍA	11
CAPITULO II	13
MARCO TEÓRICO.....	13
2.1. MARCO INSTITUCIONAL.....	13
2.1.1. ANTECEDENTES DEL OBJETO DE ESTUDIO.....	13
2.1.2. MARCO LEGAL	14
2.1.2.1. BASES LEGALES	14
2.2. INGENIERÍA DEL SOFTWARE	16
2.3. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	17
2.3.1. MODELOS ITERATIVOS E INCREMENTALES	17
2.3.2. METODOLOGÍA EN CASCADA.....	18
2.3.3. METODOLOGIA INCREMENTAL	20
2.3.4. PROTOTIPADO EVOLUTIVO.....	20
2.4. METODOLOGÍA OPENUP	21
2.4.1. OPENUP/BASIC.....	28
2.4.2. FASES DE OPEN/UP	31
2.4.2.1. FASE DE INICIO O INCEPCIÓN.....	31
2.4.2.2. FASE DE ELABORACIÓN.....	31
2.4.2.3. FASE DE CONSTRUCCIÓN.....	31
2.4.2.4. FASE DE TRANSICIÓN	31

2.4.3.	DISCIPLINAS DE OPEN/UP	32
2.4.3.1.	ANÁLISIS Y DISEÑO.....	32
2.4.3.2.	GESTIÓN DE CONFIGURACIÓN Y CAMBIOS	33
2.4.3.3.	IMPLEMENTACIÓN	33
2.4.3.4.	GESTIÓN DE PROYECTO.....	33
2.4.3.5.	REQUERIMIENTOS	34
2.4.3.6.	PRUEBA	34
2.5.	INGENIERÍA WEB	34
2.6.	METODOLOGÍAS WEB.....	35
2.6.1.	UML-BASED WEB ENGINEERING (UWE).....	35
2.6.1.1.	PROCESO	36
2.6.1.2.	ANÁLISIS DE REQUISITOS (CASOS DE USO)	38
2.6.1.3.	MODELO DE CONTENIDO.....	40
2.6.1.4.	MODELO DE USUARIO	41
2.6.1.5.	MODELO DE NAVEGACIÓN	43
2.6.1.6.	MODELO DE PRESENTACIÓN	44
2.6.1.7.	MODELO DE PROCESO	45
2.7.	HERRAMIENTAS	47
2.8.	PATRÓN MODELO VISTA CONTROLADOR (MVC)	47
2.9.	HYPERTEXT PRE-PROCESSOR (PHP).....	48
2.10.	POSTGRES.....	49
2.11.	FRAMEWORK CAKEPHP	50
2.12.	CALIDAD DE SOFTWARE	53
2.12.1.	FACTORES DE CALIDAD ISO 9126	53
2.13.	SEGURIDAD.....	55
2.13.1.	RESGUARDO DE LA INFORMACIÓN	55
2.13.2.	CONTROL DE CAMBIOS	56
CAPITULO III.....		57
MARCO APLICATIVO		57
3.1.	INTRODUCCIÓN.....	57
3.2.	DESARROLLO DEL SISTEMA	58
3.2.1.	CUADRO DE DOCUMENTOS ENTREGABLES.....	59
3.2.2.	FASE DE INICIO.....	60
3.2.2.1.	IDENTIFICACIÓN DE STAKEHOLDERS	60
3.2.2.2.	DEFINICIÓN DE LA POSIBLE SOLUCIÓN CON FUNCIONALIDAD CLAVE	61
3.2.2.3.	VISIÓN GENERAL DEL SISTEMA.....	64
3.2.3.	FASE DE ELABORACIÓN.....	65
3.2.3.1.	REQUERIMIENTOS TECNOLÓGICOS	65
3.2.3.2.	ANÁLISIS.....	66
3.2.3.3.	DISEÑO	90
3.2.3.4.	ARQUITECTURA	100
3.2.4.	FASE DE CONSTRUCCIÓN	102
3.2.5.	JUSTIFICACIÓN DE LAS HERRAMIENTAS ELEGIDAS	102
3.2.6.	PANTALLAS DEL SISTEMA	103

3.2.7.	FASE DE TRANSICIÓN	111
3.2.7.1.	PRUEBAS DEL SISTEMA.....	111
CAPITULO IV		116
CALIDAD DE SOFTWARE		116
4.1.	DEFINICIÓN	116
4.2.	FUNCIONALIDAD	116
4.3.	CONFIABILIDAD.....	120
4.4.	PORTABILIDAD	121
4.5.	FACILIDAD DE MANTENIMIENTO.....	122
4.5.1.	MANTENIMINETO ADAPTATIVO	122
4.5.2.	MANTENIMINETO PERFECTIVO	123
4.6.	USABILIDAD	123
4.7.	SEGURIDAD DEL SISTEMA	124
4.7.1.	SEGURIDAD DE BASE DE DATOS	124
4.7.2.	SEGURIDAD DE AUTENTICACIÓN.....	124
4.7.3.	SEGURIDAD DE LA APLICACIÓN.....	124
CAPITULO V		125
EVALUACIÓN COSTO BENEFICIO		125
5.1.	ANALISIS DE COSTOS	125
5.2.	CALCULO DE BENEFICIOS VAN Y TIR	127
5.2.1.	VALOR ACTUAL NETO (VAN).....	127
5.2.2.	TASA DE RENTABILIDAD INTERNA (TIR).....	129
5.2.3.	COSTO BENEFICIO	130
CAPITULO VI.....		131
CONCLUSIONES Y RECOMENDACIONES		131
6.1.	CONCLUSIONES.....	131
6.2.	RECOMENDACIONES	132
BIBLIOGRAFÍA		134

INDICE DE FIGURAS

Figura 2.1 Estratos de la ingeniería del software	16
Figura 2.2 Desarrollo iterativo e incremental	18
Figura 2.3 OpenUP capas: micro incrementos, ciclo de vida e iteración del ciclo de vida del proyecto	21
Figura 2.4 OpenUp un proceso ágil.....	25
Figura 2.5 Flujo de trabajo	30
Figura 2.6 Fases de OpenUp	32
Figura 2.7 Notación UWE.....	38
Figura 2.8 Caso de uso	40
Figura 2.9 Modelo de contenido.....	41
Figura 2.10 Modelo de usuario.....	42
Figura 2.11 Estereotipos de estructura de navegación	43
Figura 2.12 Modelo de navegación	44
Figura 2.13 Modelo de presentación	45
Figura 2.14 Modelo de procesos	46
Figura 2.15 Diagrama de actividad Login	47
Figura 2.16 Esquema de cakePHP.....	51
Figura 3.1 Relación fase de inicio UWE	57
Figura 3.2 Relación fase de elaboración UWE.....	58
Figura 3.3 Relación fase de construcción UWE	58
Figura 3.4 Relación fase de transición UWE.....	58
Figura 3.5 Caso de uso principal	70
Figura 3.6 Caso de uso registrar y visualizar base aérea.....	71
Figura 3.7 Caso de uso registrar y visualizar aeronave.....	72
Figura 3.8 Caso de uso registrar y visualizar horas o ciclos de vuelo	73
Figura 3.9 Diagrama de secuencia: Registrar y visualizar base aérea.....	85
Figura 3.10 Diagrama de secuencia: Registrar y visualizar aeronave.....	86
Figura 3.11 Diagrama de secuencia: Registrar y visualizar control aeronave	87
Figura 3.12 Modelo de contenido.....	88
Figura 3.13 Modelo del dominio	89
Figura 3.14 . Modelo navegacional Registrar y visualizar base aérea	90
Figura 3.15 Modelo navegacional Registrar y visualizar aeronave	91
Figura 3.16 Modelo navegacional Registrar y visualizar horas o ciclos de vuelo.....	91
Figura 3.17 Modelo de presentación Inicio de sesión.....	92
Figura 3.18 Modelo de presentación Panel principal.....	92

Figura 3.19 Modelo de presentación Panel principal.....	93
Figura 3.20 Modelo de presentación Registro y visualización de aeronaves	93
Figura 3.21 Modelo de presentación Panel principal.....	94
Figura 3.22 Modelo de presentación Registro de programación de manteniendo.....	94
Figura 3.23 Diagrama Entidad-Relación	95
Figura 3.24 Login.....	96
Figura 3.25 Menú principal del sistema	97
Figura 3.26 Interfaces Registrar y visualizar base aérea.....	98
Figura 3.27 Interfaces Registrar y visualizar aeronaves	99
Figura 3.28 Interfaces Registrar y visualizar control aeronaves	100
Figura 3.29 Arquitectura del sistema.....	101
Figura 3.30 Interfaz Login.....	104
Figura 3.31 Interfaz Panel principal	105
Figura 3.32 Interfaz Listado de Brigadas	105
Figura 3.33 Interfaz Listado de Brigadas	106
Figura 3.34 Interfaz Listado de unidades	106
Figura 3.35 Interfaz Listado de aeronaves	107
Figura 3.36 Interfaz Registrar aeronave	108
Figura 3.37 Interfaz Listar componentes	109
Figura 3.38 Interfaz Listar horas y/o ciclos de vuelo.....	110
Figura 3.39 Interfaz registrar horas y/o ciclos de vuelo.....	110

INDICE DE TABLAS

Tabla 2.1 Características de OpenUp	22
Tabla 3.1 Cuadro de documentos entregables	60
Tabla 3.2 Identificación de stakeholders	61
Tabla 3.3 Problemas relacionados al personal del departamento IV	62
Tabla 3.4 Problemas relacionados al personal de las unidades	63
Tabla 3.5 . Problemas relacionado al personal gerencia	63
Tabla 3.6 Control de asignación de aeronaves a las unidades aéreas.	64
Tabla 3.7 Registro automatizado de las horas y/o ciclos de vuelo	64
Tabla 3.8 Conocer el estado de las aeronaves de la FAB	65
Tabla 3.9 Controlar la programación de mantenimiento de las aeronaves.	65
Tabla 3.10 Requerimientos funcionales-Registro de grandes, pequeñas unidades	66
Tabla 3.11 Requerimientos funcionales-Componentes.....	67
Tabla 3.12 Requerimientos funcionales-Seguimiento y control de componentes	67
Tabla 3.13 Requerimientos funcionales-Programación de mantenimineto	68
Tabla 3.14 Descripción de actores.....	69
Tabla 3.15 Descripción de actores identificados	74
Tabla 3.16 Especificación Registrar horas o ciclos de vuelo	75
Tabla 3.17 Especificación Actualizar horas o ciclos de vuelo	76
Tabla 3.18 Especificación Actualizar horas o ciclos de vuelo	78
Tabla 3.19 Especificación Actualizar horas o ciclos de vuelo	79
Tabla 3.20 Especificación Actualizar horas o ciclos de vuelo	80
Tabla 3.21 Especificación Registrar programación de inspección.....	81
Tabla 3.22 Especificación Modificar programación de inspección	82
Tabla 3.23 Especificación Registrar horas o ciclos de vuelo	83
Tabla 3.24 Herramientas de construcción	102
Tabla 3.25 Caso de prueba Ingreso al sistema	111
Tabla 3.26 Caso de prueba Registrar material aéreo	112
Tabla 3.27 Caso de prueba Registrar componente.....	113
Tabla 3.28 Caso de prueba Registrar horas o ciclos de vuelo	114
Tabla 3.29 Caso de prueba Asignar componente.....	114
Tabla 3.30 Caso de prueba Registrar programación de inspección.....	115
Tabla 4.1 Resultados del sistema.....	117
Tabla 4.2 Puntos obtenidos	118
Tabla 4.3 Ponderación de las respuestas.....	119
Tabla 4.4 Ponderación en relación al sistema.....	120

Tabla 4.5 Facilidad de uso.....	123
Tabla 5.1 Relación de valores en el modelo COCOMO.....	126

CAPITULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

Los avances tecnológicos actuales nos proporcionan herramientas informáticas que permiten a todas las instituciones implementar sistemas de información para una administración eficiente de los medios, en la FAB (Fuerza Aérea Boliviana) no se ha incentivado esta práctica, tomando en cuenta que cuenta con los recursos materiales y humanos necesarios para realizarlo. Debido a esto, la FAB. se encuentra sumamente rezagada en cuanto a una rápida, eficaz administración y evaluación de los componentes de las aeronaves.

En la actualidad la Fuerza Aérea Boliviana, a pesar del desarrollo tecnológico, no cuenta con un instrumento adecuado que le permita orientar sus actividades hacia el cumplimiento diario de objetivos claramente establecidos, es decir, contar con un sistema de información y almacenamiento de datos, el cual contenga la descripción de aeronaves, seguimiento de repuestos y control de mantenimiento de los componentes de las aeronaves de la Fuerza Aérea Boliviana.

En este contexto, dada la rapidez con que se presentan los acontecimientos en la actualidad, y considerando que una de las características más importantes de la FAB. es la Acción Inmediata, el presente proyecto tiene como propósito aportar al desarrollo de sistemas a través del análisis, diseño e implementación de un sistema para la administración de componentes mayores y rotables de las aeronaves de la FAB., coadyuvando de esta manera al cumplimiento de su misión.

En el presente proyecto se desea realizar un seguimiento y control de los repuestos que se instalan como también los que se desmontan de las aeronaves de la FAB. e identificando el

tiempo de vuelo, en horas y/o ciclos, de los componentes mayores y rotables de las aeronaves, lo que mejorará la eficiencia en el mantenimiento, control y seguimiento de las aeronaves y sus respectivos componentes, así también como un apoyo fundamentalmente al Departamento IV-Logística de la FAB. que es el que carece de contar de manera precisa y oportuna con la información relacionada al tema.

1.2. ANTECEDENTES

1.2.1. INSTITUCIONALES

El 26 de septiembre de 1957, mediante Decreto Supremo N° 04743 se crea el Comando de la Fuerza Aérea, lo que significó el nacimiento de una nueva institución militar, la Fuerza Aérea Boliviana (FAB) , al mismo nivel del Ejército Nacional.

La Fuerza Aérea es una institución del Estado, miembro de las Fuerzas Armadas de la Nación, que tiene la misión:

“Defender la soberanía del espacio Aéreo Nacional, la seguridad de la infraestructura aeronáutica estratégica del País y utilizar el poder Aeroespacial Militar en el Desarrollo del Estado Plurinacional, A FIN DE contribuir al logro de la misión constitucional de las Fuerzas Armadas”.

La actual estructura orgánica de la FAB. De manera general presenta la siguiente composición:

- Comando General
- Estado Mayor General
- Grandes Unidades
- Pequeñas Unidades
- Institutos Militares
- Direcciones Militares
- Reparticiones Militares

Una de las áreas que corresponde a la FAB es el Departamento IV-Logística EMGFAB, de acuerdo a su manual de organización y funciones vigentes, asume la responsabilidad de cumplir con la siguiente tarea¹:

“Prever y proveer medios y materiales comunes, en apoyo a las unidades o institutos de la fuerza aérea y satisfacer sus necesidades para el cumplimiento de la misión asignada, tanto en tiempo de paz como en tiempo de guerra”.

Para el cumplimiento de este cometido desempeña las funciones de fiscalización, evaluación supervisión y control, en las cuatro funciones que desempeña la logística:

- Abastecimiento
- Mantenimiento
- Adquisición
- Transporte Terrestre

Donde la función de mantenimiento es el área de interés para la elaboración del presente trabajo, el cual está a cargo de la Sección Material Aéreo y sus sub secciones correspondientes, la misma que coadyuva a la tarea asignada al departamento IV-Logística asumiendo, de la misma manera, la responsabilidad de cumplir con las siguientes tareas:

- De la Sección Material Aéreo, “Llevar el control de la documentación y estado de las aeronaves que pertenecen a la Fuerza Aérea, así como aquellas incautadas y confiscadas que se encuentran en poder de la Institución, en cumplimiento a la Tarea del Departamento IV Log. EMGFAB”.
- De la Sub Sección Control de Mantenimiento, “Asesorar al Jefe de Sección en relación a trabajos de mantenimiento, que deberán realizar en las diferentes aeronaves de la Fuerza Aérea”.
- De la Sub Sección Material Aéreo, “Llevar al día el registro de aviones y sus matrículas del material de vuelo de las aeronaves de la Fuerza Aérea Boliviana”.

¹ Manual de Organización y Funciones del Departamento IV Logística EMGFAB, Gestión 2013

Para cumplir con este propósito, en esta sección se mantienen cuadros operativos donde se registra datos de aeronaves por unidades y su estado actual, además de actualizar archivos de kardex (comunes) y una hoja electrónica, donde se almacena el resumen de la información procesada.

En cuanto a aeronaves en la actualidad la Fuerza aérea. cuenta con aproximadamente de 300 entre operables y no operables, las mismas se clasifican en²:

- Empleos tácticos (K-8, T-33, pilatus PC-7 y otros)
- Ejecutivas (Falcon 900EX, Beechcraft, Sabreliner, Super King)
- Transportes (BAe 146-200, MA-60, Fokker F-27, casa C-212, Boing 727-200, Boing 737-200, Douglas DC-10-10 y C-130 Hercules)
- Helicópteros (Eurocopter, Super puma, Aérospatiale SE 319B ALOUETTE III – Astazau, UH1-H y Helicóptero EC-145)
- Aerofotogrametría (Lear Jet 25B y 25D y Cessna 402 C)
- Entrenamiento (Diamond” D-40, UIRAPURU T- 23 y Beechcraft T-34 “Mentor)

Asimismo realizada la revisión de la Sección Planeamiento de la Fuerza Aérea, no se ha encontrado ningún documento relacionado con este tema, asimismo se ha podido evidenciar que no existen trabajos similares relacionados con la administración de componentes para aeronaves o el control del mantenimiento de aeronaves, pero si existe una base de datos para la sección abastecimiento la cual no cuenta con la documentación respectiva.

De la misma manera, según información obtenida, en la Dirección de Informática de la Fuerza Aérea, en el cual se realiza un sistema integrado para la Fuerza Aérea, donde actualmente existen los siguientes módulos o sistemas:

- Horas de Vuelo de los tripulantes de la Fuerza Aérea.
- Sistema de Personal.
- Cargos de cuenta
- Correspondencia

² Página web www.fab.bo

- Sistema de Seguimiento de Procesos Judiciales

En lo que se refiere a los componentes de las aeronaves, se proyectaba la implementación de este sistema, pero quedó sin efecto, funcionando solamente una tabla en Microsoft Excel donde se detalla algunas características sobre la administración de los componentes de las aeronaves de alto costo.

1.3. TRABAJOS SIMILARES EN LA CARRERA DE INFORMÁTICA

Sistema de gestión y seguimiento para almacén de material de vidrio. [Manueco-2009]

Se desarrolló e implementó un sistema de gestión y seguimiento de información para la unidad de almacén de materiales de vidrio, que facilite el manejo y seguimiento de la información al encargado de almacén como a los usuarios, facilitando el registro, actualización, control del movimiento de materiales como ser préstamos y devoluciones, además de tener un seguimiento de ellos y la obtención de un pronóstico de la calidad óptima de adquisición de los materiales más necesitados y se solventa los materiales existentes en cada gestión, además de controlar el ingreso y salida del material, con el objetivo de que no se realicen compras insulsas.

Sistema de administración de inventarios de materiales y equipos. [Morante-2010]

Este proyecto fue realizado en la facultad de agronomía, se desarrolla un sistema de administración e inventario para materias y equipos que tiene por objeto la administración y control, para dar solución a varios problemas particularmente en la existencia, pedidos, ingresos y salidas de materiales no contando con control y salida óptima.

Se desarrolló un sistema de administración para mejorar el manejo y control de los materiales, optimizando los procesos y el tiempo de requerimiento de la institución.

Sistema de control y seguimiento de inventario de fármacos. Caso clínica San Damian. [Valle-2009]

El siguiente proyecto de grado, tiene como objetivo el desarrollo de un sistema de control y seguimiento de inventarios de fármacos, la misma herramienta permite un adecuado control

de inventario que cumpla con los requerimientos cambiantes de la clínica y de los usuarios a través del registro de transacciones, informes que apoyen a los inventarios físicos, elaboración de reportes para los clientes, proveedores, ítems y otros que van desacuerdo a las exigencias de las áreas involucradas.

Sistema de administración y control de almacén e inventario GMLP, Caso Mallasa. [Matta-2011]

El siguiente proyecto de grado tiene como objeto implementar un sistema de administración y control de almacén en la unidad de almacenes de la Sub alcaldía de Mallasa que permita mejorar el registro de ingreso, salida y existencias de los materiales en almacén de manera que permita brindar información confiable y oportuna para una toma de decisiones.

1.4. PROBLEMA

1.4.1. IDENTIFICACION DEL PROBLEMA

Dentro de la industria aeronáutica, la vida de un avión se mide en horas de vuelo y/o ciclos (despegue-aterizaje). El término Overhaul se refiere a una inspección en la que se realizan pruebas de fatiga de materiales y se determina la aeronavegabilidad de una aeronave, para realizar el mantenimiento de la estructura de la aeronave se debe tener un registro adecuado, en la Fuerza Aérea existen aeronaves que no tienen un registro de horas de vuelo y/o ciclos de la estructura o partes que componen la aeronave, porque son aviones que fueron decomisadas por el estado y transferidos a la FAB.

La FAB. maneja una considerable cantidad de información relacionada a los componentes de las aeronaves, los mismos que se dividen en cuatro grupos:

- **Mayores**, son aquellos componentes que tienen un elevado costo monetario y pueden ser reparados como también sustituidos, como ser: motor, hélices, tren de aterrizaje y otros.
- **Rotables**, son aquellos componentes que se pueden reparar y puede ser sustituido por la misma clase, como ser: equipos de comunicación, generadores AC/DC, Unidades de control, FCU/PCU y otros.

- **Fungibles**, es el que tiene tales características físicas que se consume completamente durante el uso y se vuelve inservible por el uso y puede ser sustituido por la misma clase, como ser: Llantas, tornillos, arandelas y otros.
- **Misceláneos**, son los artículos que están compuestos por varias cosas distintas o de géneros diferentes, como ser: asientos, cinturones de seguridad, arneses de hombro y otros.

Los componentes mayores y rotables deben responder a:

- Un seguimiento, es decir, contar con la información general de un determinado componente como su descripción, estado, ubicación y otros.
- Un control, el mismo que se realiza a través de dos indicadores: horas de vuelo, y ciclos (despegue + aterrizaje= 1 ciclo).
- Una programación, el mismo que se establece según tiempos definidos.

Al no contar de manera oportuna y óptima con esta información, la FAB. atraviesa varios problemas desde los más básicos como la pérdida de tiempo, hasta los más críticos como la pérdida económica, en fin, al no contar con una herramienta que provea de manera oportuna, precisa e instantánea de esta información, se deriva a que se desconozca el estado de las aeronaves en general, se corra el riesgo de que una aeronave esté en funcionamiento con sus componentes en un estado nada óptimo, que se pierda eficiencia operativa al no contar con una programación adecuada de mantenimiento de los componentes de las aeronaves.

De acuerdo a la recolección de datos, se ha podido evidenciar que la sección material aéreo de la FAB., Carece de un instrumento adecuado que le permita efectuar un control óptimo de los componentes de las aeronaves y tener a disponibilidad información oportuna y actualizada, es decir, contar con un sistema de almacenamiento de datos, el cual contenga la descripción de las diferentes aeronaves, su estado actual en cuanto a horas de vuelo y/o ciclos (despegue y aterrizaje) de sus componentes, procedencia de las aeronaves y las necesidades para una correcta operabilidad (referente a estructura y componentes mayores), que le permita elaborar un plan de seguimiento, mantenimiento y proyectar a la institución hacia el futuro en condiciones mucho más favorables .

No se cuenta con información suficiente sobre los repuestos de aeronaves de alto costo, sabiendo que estas partes de aeronaves demandan una inversión considerable para la FAB. Asimismo estos componentes debería manejarse con un control recomendable y de esta manera tener una información adecuada.

En la actualidad los documentos correspondientes al seguimiento y control de los componentes de las aeronaves se encuentran en archivos físicos, entonces no existe un control exacto de los componentes que se instalan o se desmontan de las aeronaves de la FAB.

1.4.2. PROBLEMA CENTRAL

¿Cómo lograr un mantenimiento eficiente de los componentes, mayores y rotables, de las aeronaves de la Fuerza Aérea Boliviana, a partir de la información sobre los repuestos y el control de las horas y/o ciclos de vuelo?

1.4.3. PROBLEMAS SECUNDARIOS

- Se carece de un seguimiento a los repuestos de las aeronaves para realizar el mantenimiento de los componentes, lo que ocasiona pérdida de tiempo al momento de establecer una planificación del mantenimiento.
- No se cuenta con un control adecuado de repuestos de alto valor instalados en las aeronaves, lo que ocasiona pérdida de eficiencia en las operaciones realizadas por las aeronaves.
- No se cuenta con información inmediata y precisa sobre el control de las horas de vuelo y/o ciclos de las aeronaves y sus componentes, por lo cual no se puede prevenir la adquisición o acelerar el mantenimiento de los componentes.
- Se carece de información inmediata y precisa sobre la asignación y el estado de las aeronaves que se encuentra en las diferentes unidades o grupos de la FAB, por lo cual se maneja registros parcialmente completos o en el peor de los casos no se cuentan con registros de los componentes de las aeronaves.
- No se cuenta con un historial (kardex de registro) de los componentes que tienen las aeronaves, lo que ocasiona pérdida de tiempo y recursos.

1.5. OBJETIVO GENERAL

Desarrollar un sistema Web de seguimiento y control de mantenimiento de los componentes, mayores y rotables, de las aeronaves de la Fuerza Aérea Boliviana, para mejorar la eficiencia de operatividad de las aeronaves.

1.6. OBJETIVOS ESPECIFICOS

- Realizar el seguimiento de los componentes de las aeronaves
- Efectuar el control de horas de vuelo y/o ciclos para el mantenimiento de los componentes de las aeronaves.
- Realizar el control de las horas y/o ciclos de la estructura de la aeronave
- Emitir reportes acerca de: aeronaves por brigadas y unidades, estado de componentes de aeronaves específicas, detalle de componentes de aeronaves, detalle completo de aeronaves.
- Diseñar e implementar una base de datos, que permita responder a las necesidades que tenga la institución.

1.7. LIMITES Y ALCANCES

1.7.1. LIMITES

El presente trabajo abarcará la administración (seguimiento y control) de los componentes MAYORES y ROTABLES que pertenecen a las aeronaves, dejando de lado a los componentes fungibles y misceláneos, los mismos que por el análisis efectuado son de menor importancia que los dos componentes mencionados anteriormente, así como también el proyecto de grado a implementar no tomará en cuenta el área contable, es decir manejar el precio y costo de las aeronaves o sus componentes.

1.7.2. ALCANCES

El alcance del presente trabajo es la implementación de un sistema que permita: el registro de la composición de la Fuerza Aérea (Brigadas y unidades), registro de aeronaves, registro de componentes (mayores y rotables) de las aeronaves, registro de horas y/o ciclos y

procesos con las horas y/o ciclos de las aeronaves, así como la emisión de reportes según requerimiento de la institución.

1.8. APORTE

El aporte que se desea brindar con el presente proyecto es el de proponer la aplicación de la metodología OpenUp, al desarrollo de un sistema, una metodología ágil que permite el buen desarrollo de software evitando la extensa documentación, el mismo que: es extensible para cualquier módulo, ajustable para el entorno necesario, ligero y ágil en su implementación, cuenta con un proceso iterativo y no requiere un tecnología específica.

1.9. JUSTIFICACIÓN

1.9.1. JUSTIFICACIÓN TÉCNICA

La FAB., a través de la Dirección de Informática, ha iniciado un proceso de actualización tecnológica mediante la compra de nuevos equipos los cuales permiten almacenar y manipular grandes cantidades de información acordes con las exigencias de la institución, de la misma manera ha incorporado a la mencionada dirección a personal especialista para su manejo y operación.

1.9.2. JUSTIFICACIÓN OPERATIVA

A través de este trabajo se proporcionará a la Sección Material Aéreo del Departamento IV de la FAB., un sistema de información que le permita almacenar y procesar información referente al estado de las aeronaves y sus componentes mayores y rotables, facilitando de esta manera la planificación de mantenimiento y el control de su ejecución.

1.9.3. JUSTIFICACIÓN ECONÓMICA

La FAB., como producto del apoyo a la integración nacional a través del servicio de transporte aéreo, genera recursos propios los cuales son destinados al mejoramiento del material aéreo, en este sentido una adecuada administración de los componentes de las

aeronaves, permitirá a la Sección Material Aéreo asegurar la operabilidad de las aeronaves garantizando el servicio de transporte y por ende los ingresos necesarios para la subsistencia de la institución aérea.

1.9.4. JUSTIFICACIÓN SOCIAL

La implementación del presente proyecto, permitirá mejorar el control de las aeronaves, en términos de mantenimiento de sus componentes, lo cual garantizará la integridad de los vuelos efectuados por las aeronaves de la Fuerza Aérea Boliviana, mostrando seguridad en vuelo, así como también una mejora operativa presentado eficiencia en operaciones de vuelo a la sociedad.

1.10. MÉTODOS DE INVESTIGACIÓN Y METODOLOGÍA

- Para la realización del presente proyecto, se emplean la investigación descriptiva, que se detalla a continuación:
- Investigación descriptiva, los estudios descriptivos buscan especificar las propiedades importantes de personas, grupos, comunidades o cualquier otro fenómeno que sea sometido a análisis [Hernández, Fernández, Baptista, 2006]. Miden o evalúan diversos aspectos, dimensiones o componentes del fenómeno o fenómenos a investigar.

La Metodología a usarse en el desarrollo del presente proyecto es la siguiente:

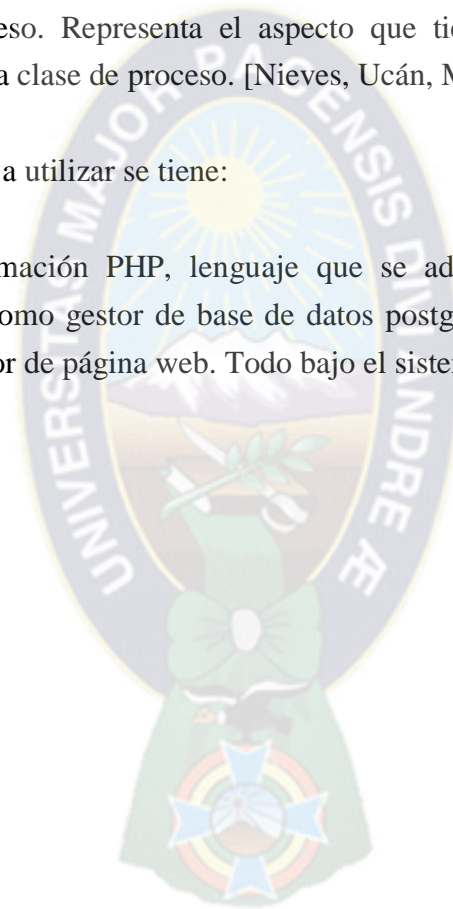
- OpenUp, una metodología de desarrollo ágil, es un proceso de desarrollo de software mínimamente suficiente, esto quiere decir que contiene sólo el contenido fundamental, esto es que no provee orientación sobre temas en los que el proyecto tiene que lidiar.
- UWE. Es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo.

En su implementación se deben contemplar las siguientes etapas y modelos.

- Análisis de requisitos. Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.
- Modelo de contenido. Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- Modelo de navegación. Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- Modelo de proceso. Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso. [Nieves, Ucán, Menendez, 2014]

Dentro de las herramientas a utilizar se tiene:

- Lenguaje de programación PHP, lenguaje que se adecua a la programación de aplicaciones Web. Como gestor de base de datos postgresql, de distribución libre y Apache como servidor de página web. Todo bajo el sistema operativo Windows.



CAPITULO II

MARCO TEÓRICO

2.1. MARCO INSTITUCIONAL

2.1.1. ANTECEDENTES DEL OBJETO DE ESTUDIO.

El Departamento IV, de acuerdo a su manual de organización y funciones vigente, asumido la responsabilidad de cumplir con la siguiente tarea³:

“PREVER Y PROVEER MEDIOS Y MATERIALES COMUNES, EN APOYO A LAS UNIDADES O INSTITUTOS DE LA FUERZA AÉREA Y SATISFACER SUS NECESIDADES PARA EL CUMPLIMIENTO DE LA MISIÓN ASIGNADA, TANTO EN TIEMPO DE PAZ COMO EN TIEMPO DE GUERRA”.

Para el cumplimiento de este cometido desempeña las funciones de fiscalización, evaluación supervisión y control, en las cuatro funciones que desempeña la logística:

- Abastecimiento
- Mantenimiento
- Adquisición
- Transporte Terrestre

Donde la función de mantenimiento es el área de interés para la elaboración del presente trabajo, el cual está a cargo de la Sección Material Aéreo y sus sub secciones correspondientes, la misma que coadyuva a la tarea asignada al departamento IV Logística asumiendo, de la misma manera, la responsabilidad de cumplir con las siguientes tareas:

- **De la Sección Material Aéreo**, “Llevar el control de la documentación y estado de las aeronaves que pertenecen a la Fuerza Aérea, así como aquellas incautadas

³ Manual de Organización y Funciones del Departamento IV Logística EMGFAB, Gestión 2005

y confiscadas que se encuentran en poder de la Institución, en cumplimiento a la Tarea del Departamento IV Log. EMGFAB”.

- **De la Sub Sección Control de Mantenimiento**, “Asesorar al Jefe de Sección en relación a trabajos de mantenimiento, que deberán realizar en las diferentes aeronaves de la Fuerza Aérea”.
- **De la Sub Sección Material Aéreo**, “Llevar al día el registro de aviones y sus matrículas del material de vuelo de las aeronaves de la Fuerza Aérea Boliviana”.

Para cumplir con este propósito, en esta sección se mantienen cuadros operativos donde se registra datos de aeronaves por unidades y su estado actual, además de actualizar archivos de kardex (comunes) y una hoja electrónica, donde se almacena el resumen de la información procesada.

2.1.2. MARCO LEGAL

2.1.2.1. BASES LEGALES

- **Constitución Política del Estado.**

Título VII, Régimen de las fuerzas Armadas, el artículo 208, que a la letra dice: “La misión fundamental de las Fuerzas Armadas de la Nación, es defender y conservar la Independencia Nacional, la Seguridad y Estabilidad de la Republica, el Honor y Soberanía Nacional; asegurar el Imperio de la Constitución Política, garantizar la estabilidad del Gobierno legalmente constituido y cooperar con el desarrollo del País”.

- **La Ley Orgánica de las Fuerzas Armadas.**

Capitulo III, en su artículo 6to., inciso b) sostiene que las Fuerzas Armadas deben: “Precautelar la seguridad, soberanía y honor nacional” y en el inciso e) indica: ”Defender, controlar y conservar la integridad territorial, las aguas territoriales y el

espacio aéreo, así como contribuir a la protección del medio ambiente, los recursos naturales y de todo el patrimonio nacional”

En el artículo 22, inciso m), dentro sus atribuciones y responsabilidades del Ministerio de Defensa es de: “Planificar y desarrollar la investigación científica-tecnológica a los fines de la seguridad y defensa nacional en coordinación con el Comando en Jefe de las Fuerzas Armadas” y en el inciso q) indica que debe: “Disponer en forma exclusiva de un segmento electromagnético a los fines de las comunicaciones para la Seguridad y Defensa Nacional”.

En el artículo 62, inciso i), dentro de las atribuciones de los Comandos de Fuerza indica que deben de: “Realizar la investigación científica, tecnológica y la elaboración de estudios de interés para su Fuerza”.

- **La Directiva Presidencial 07/91.**

Establece: “Que las Fuerzas Armadas deben adecuar su estructura y sus medios al avance de la situación general del país en los aspectos económicos, social, tecnológico, etc.”, recomendando que las diferentes Fuerzas generen procesos de modernización de sus sistemas, armamentos, equipos y comunicaciones.

- **Orden del Día de la FAB No. 122/02.**

Mediante esta, el 9 de Octubre de 2002, durante la gestión de Comando de Gral. Fza. Ae. Erland Camacho Mancilla, con el propósito de mejorar los recursos en el campo de la Informática y estar acorde al avance tecnológico, se crea la “DIRECCION GENERAL DE INFORMATICA”.

- **Directiva de la FAB No.21/2004**

Donde se establece parámetros para la asignación de matrículas para aeronaves de la FAB, además de mencionar el procedimiento para la asignación de las mismas.

2.2. INGENIERÍA DEL SOFTWARE

A pesar de que cientos de autores han definido en forma individual la ingeniería del software, la definición que propuso Fritz Bauer en una conferencia fundamental sobre la materia aún se puede utilizar como base para el debate:

[La ingeniería del software es] el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. [Pressman, 2006]

La IEEE presenta una definición más comprensible:

Ingeniería del software: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería del software. 2) El estudio de enfoques como en 1). [Pressman, 2006]

La ingeniería del software es una tecnología estratificada. Como se muestra en la figura, cualquier enfoque de la ingeniería (incluido el de la ingeniería del software) debe estar sustentado en un compromiso con la calidad. La Gestión de la Calidad Total, Sigma Seis y enfoques similares fomentan una cultura de mejora continua del proceso, y es esta cultura que al final conduce al desarrollo de enfoques muy efectivos para la ingeniería del software. [Pressman, 2006].

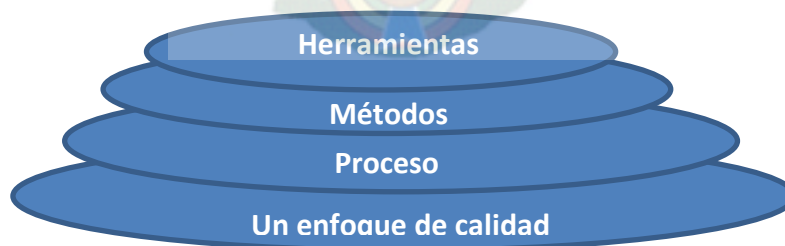


Figura 2.1 Estratos de la ingeniería del software
Fuente: Pressman, 2006

La base que soporta la ingeniería del software en un *enfoque de calidad*, como se muestra en la figura 2.1.

La base de la ingeniería del software es el estrato del proceso. El proceso de la ingeniería del software es el elemento que mantiene juntos los estratos de la tecnología y que permite el desarrollo racional y a tiempo del software de computadora.

Los métodos de la ingeniería del software proporcionan los “como” técnicos para construir software. Los métodos abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte.

Las herramientas de la ingeniería del software proporcionan el soporte automatizado o semiautomatizado para el proceso y los métodos. Cuando las herramientas se integran de forma que la información que cree una de ellas pueda usarla otra, se dice que se ha establecido un sistema para el soporte de desarrollo del software, que con frecuencia se denomina *ingeniería asistida por computadora*. [Pressman, 2006]

2.3. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

2.3.1. MODELOS ITERATIVOS E INCREMENTALES

El desarrollo iterativo e incremental es un proceso de desarrollo de software cíclico en respuesta a la debilidad del modelo en cascada. Empieza con una planificación inicial y termina con el despliegue con la iteración cíclica en el medio.

El desarrollo incremental e iterativo es una parte esencial de SCRUM, RUP, DSDM, XP y generalmente de los marcos de desarrollo de software ágil.



Figura 2.2 Desarrollo iterativo e incremental
 Fuente: Ingeniería del software: Metodologías y ciclos de vida, 2009

El desarrollo incremental es una estrategia programada y en etapas, en la que las diferentes partes del sistema se desarrollan en diferentes momentos o a diferentes velocidades, y se integran a medida que se completan.

El desarrollo iterativo es una estrategia de programación de reproceso en la que el tiempo se separa para revisar y mejorar partes del sistema. Esto no presupone desarrollo incremental, pero trabaja muy bien con él. Una diferencia típica es que la salida de un incremento no está necesariamente sujeta a más refinamiento, y sus pruebas o la realimentación del usuario no se usa como entrada para revisar los planes o especificaciones de los incrementos sucesivos. Por el contrario, la salida de una iteración se examina para modificación y especialmente para revisar los objetivos de las sucesivas iteraciones.

2.3.2. METODOLOGÍA EN CASCADA.

La versión original del modelo en cascada, fue presentada por Royce en 1970, aunque son más conocidos los refinamientos realizados por Boehm (1981), Sommerville (1985) y

Sigwart (1990). En este modelo, el producto evoluciona a través de una secuencia de fases ordenadas en forma lineal, permitiendo iteraciones al estado anterior. El número de etapas suele variar, pero en general suele ser:

- Análisis de requisitos del sistema.
- Análisis de requisitos del software.
- Diseño preliminar.
- Diseño detallado.
- Codificación y pruebas.
- Explotación (u operación) y mantenimiento.

Las características de este modelo son:

- Cada fase empieza cuando se ha terminado la anterior.
- Para pasar a la fase posterior es necesario haber logrado los objetivos de la previa.
- Es útil como control de fechas de entregas.
- Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto.

Mc Cracken y Jackson (1982) han realizado algunas críticas al modelo:

- Sostienen que los proyectos reales rara vez siguen una linealidad tal, y que siempre hay iteraciones que van más allá de la etapa anterior.
- Además, como el sistema no estará en funcionamiento hasta finalizar el proyecto, el usuario, recibe el primer producto al haber consumido casi la totalidad de los recursos.

Otra limitación que se argumenta es que el modelo supone que los requisitos pueden ser “congelados” antes de comenzar el diseño y esto significa un hardware asociado durante el tiempo que dure el proyecto.

2.3.3. METODOLOGIA INCREMENTAL

Las etapas son las mismas que en el ciclo de vida en cascada y su realización sigue el mismo orden, pero corrige la problemática de la linealidad del modelo en cascada. Este modelo incremental fue desarrollado por Lehman (1984) y en cada paso sucesivo agrega al sistema nuevas funcionalidades o requisitos que permiten el refinamiento a partir de una versión previa. El modelo es útil cuando la definición de los requisitos es ambigua y poco precisa, porque permite el refinamiento, o sea se puede ampliar los requisitos y las especificaciones derivadas de la etapa anterior.

Uno de los problemas que puede presentar es la detección de requisitos tardíamente, siendo su corrección tan costosa como en el caso de la metodología en cascada.

2.3.4. PROTOTIPADO EVOLUTIVO

El uso de prototipos se centra en la idea de ayudar a comprender los requisitos que plantea el usuario, sobre todo si este no tiene una idea muy acabada de lo que desea. También pueden utilizarse cuando el ingeniero de software tiene dudas acerca de la viabilidad de la solución pensada. Esta versión temprana de lo que será el producto, con una funcionalidad reducida, en principio, podrá incrementarse paulatinamente a través de refinamientos sucesivos de las especificaciones del sistema, evolucionando hasta llegar al sistema final.

Al usar prototipos, las etapas del ciclo de vida clásico quedan modificadas de la siguiente manera.

- Análisis de requisitos del sistema.
- Análisis de requisitos del software.
- Diseño, desarrollo e implementación del prototipo.
- Prueba del prototipo.
- Refinamiento iterativo del prototipo.
- Refinamiento de las especificaciones del prototipo.
- Diseño e implementación del sistema final.

- Explotación (u operación) y mantenimiento.

Si bien el modelo de prototipos evolutivos, fácilmente modificables y ampliables es muy usado, en muchos casos pueden usarse prototipos descartables para esclarecer aquellos aspectos del sistema que no se comprenden bien [Juzgado, 1996].

2.4. METODOLOGÍA OPENUP

OpenUP pertenece al grupo de metodologías ágiles de desarrollo, es un proceso de desarrollo iterativo del software que es mínimo, completo, y extensible.

El proceso es mínimo en que solamente el contenido fundamental es incluido; es completo en que puede ser manifestado como todo el proceso para construir un sistema; extensible en que puede ser utilizado como fundamento sobre el cual el contenido de proceso se pueda agregar o adaptar según lo necesitado.

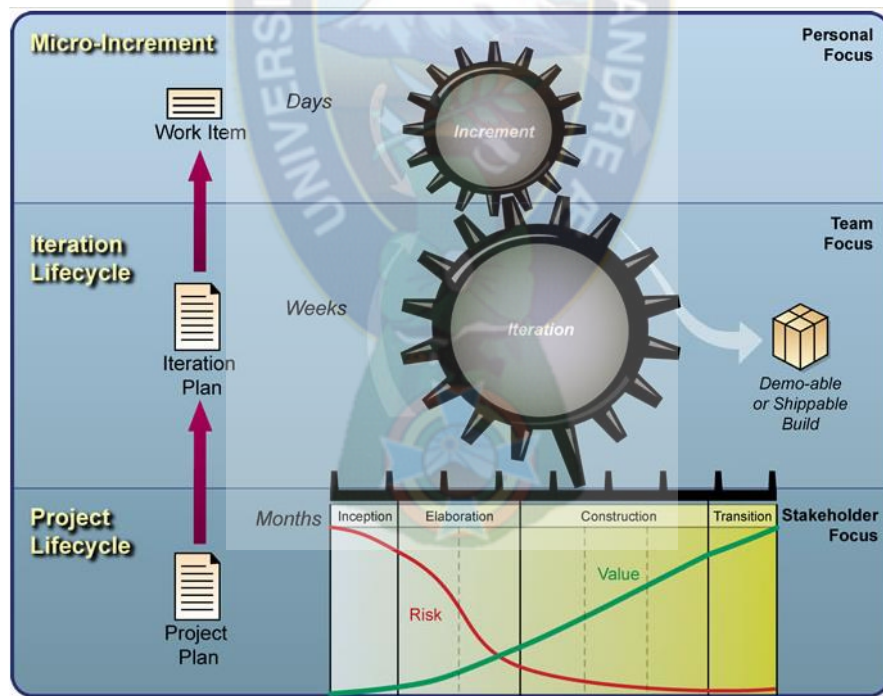


Figura 2.3 OpenUP capas: micro incrementos, ciclo de vida e iteración del ciclo de vida del proyecto

Fuente: The Eclipse Foundation, 2006

OpenUP se organiza en micro incrementos. Estos representan unidades cortas de trabajo que producen un ritmo constante, medibles del avance del proyecto (normalmente medido en horas o días). El proceso se aplica una intensa colaboración donde el sistema está desarrollado de forma incremental por un equipo comprometido, auto-organizado. Estos micro incrementos proporcionan un bucle de retroalimentación extremadamente corto que impulsa las decisiones de adaptación dentro de cada iteración.

OpenUP divide el proyecto en iteraciones: intervalos planificados, encajados en tiempos que normalmente se miden en semanas. El plan de iteración define lo que debe ser entregado dentro de una iteración, y el resultado es un entregable.

CARACTERÍSTICA	OPENUP
Equipo muy grande de desarrollo	No
Seguimiento de cumplimiento	No
Situaciones contractuales	No
Aplicaciones de misión crítica	No
Tecnología específica	No
Extensible para cualquier modulo	Si
Ajustable para el entorno necesario	Si
Ligero y ágil en su implementación	Si
Proceso iterativo	Si
Contiene características esenciales	Si

Tabla 2.1 Características de OpenUp
Fuente: The Eclipse Foundation, 2012

OpenUp estructura el ciclo de vida en cuatro fases: inepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados y miembros del equipo visibilidad y puntos de decisión a través del proyecto. Esto permite vigilancia efectiva y permite decisiones de “Continuar y no continuar” en momentos apropiados. Un plan de proyecto define el ciclo de vida, y el resultado final es una aplicación lanzada.

Principios Básicos de OpenUP

OpenUp se rige por cuatro principios fundamentales, ellos crean las bases para la interpretación de los roles y productos de trabajo así como desarrollar las tareas [The Eclipse Foundation, 2006].

Colaborar para alinear los intereses y un entendimiento compartido

El software es creado por personas con diferentes intereses y habilidades quienes trabajan juntos para crear software eficientemente.

Por consiguiente, desarrolle prácticas que fomenten un ambiente de equipo saludable. Un ambiente de equipo saludable permite la colaboración efectiva, lo cual encuadra los intereses de los participantes del proyecto (equipo de desarrollo, aseguramiento de la calidad, stakeholders del producto, clientes) y ayuda a los participantes del proyecto a desarrollar un entendimiento compartido del proyecto.

Balancear las prioridades en contradicción para maximizar el valor de los stakeholder

Los sistemas necesitan ser desarrollado por equilibrar las diferencias y algunas veces perspectivas contradictorias. Por ejemplo, ¿usted prefiere minimizar el costo reutilizando una capacidad existente o hacer un desarrollo personalizado de la capacidad para que haga exactamente lo que usted quiere?

Por consiguiente, los participantes del proyecto y los stakeholders deben colaborar para desarrollar una solución que maximice el beneficio de los stakeholders y cumpla con las restricciones planteadas en el proyecto. Lograr un balance es un proceso dinámico porque si

los stakeholders y los participantes del proyecto aprenden más acerca del sistema, entonces sus prioridades y restricciones cambian.

Enfocarse en articular la arquitectura

Sin un fundamento arquitectónico, un sistema evolucionará en una forma ineficiente y casual. Tal sistema frecuentemente presenta dificultades para evolucionar, reutilizarse o integrarse sin una reconstrucción sustancial. Esto también dificulta organizar el equipo o comunicar las ideas sin el enfoque técnico común que la arquitectura proporciona.

Por consiguiente, use la arquitectura como un punto focal para que los desarrolladores alineen sus intereses e ideas, articulando las decisiones técnicas esenciales a través de una arquitectura creciente.

Evolucionar para obtener continuamente retroalimentación y progreso

Usualmente no es posible conocer todas las necesidades de los stakeholders, ser consciente de todos los riesgos, comprender todas las tecnologías del proyecto, o saber como trabajar con sus colegas. Aún si fuese posible conocer todas estas cosas, es probable que cambien durante la vida del proyecto.

Por consiguiente, divida el proyecto en mini proyectos, iteraciones encajadas en tiempo para demostrar valor incremental y obtener retroalimentación temprana y continua.

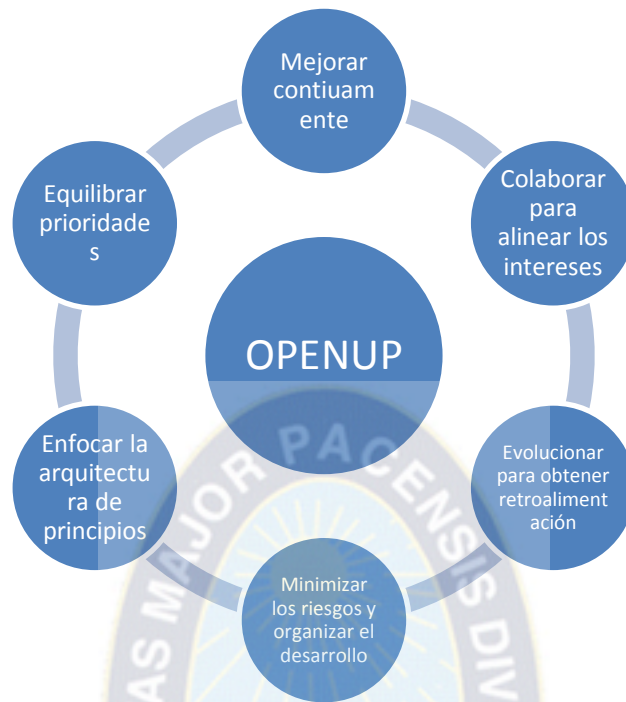


Figura 2.4 OpenUp un proceso ágil
 Fuente: The Eclipse Foundation, 2012

OpenUp es un proceso ágil, y cabe aclarar que hay mucho más a la agilidad que simplemente ser ligero. La mayoría de prácticas ágiles reconocidas están destinadas a conseguir que el equipo se comunique entre sí proporcionando una comprensión compartida del proyecto, suprimiendo de esta manera la entrega de resultados improductivos y la pérdida de tiempo. [Balduino, 2007 en Novoa, Ramos, 2013].

Cada principio de OpenUp soporta una declaración del manifiesto ágil [The Eclipse Foundation, 2006], como se ve en la siguiente tabla:

Principio de OpenUP	Declaración del manifiesto ágil
Colaborar para alinear los intereses y compartir comprensión	Individuos e interacciones sobre procesos y herramientas
Balancear prioridades que compiten para maximizar el valor de los clientes	Colaborar con el cliente sobre negociación de contratos

Enfocar rápidamente la arquitectura para minimizar riesgos y organizar el desarrollo	Software funcionando sobre documentación
Evolucionar para obtener retroalimentación continuamente y mejorar	Responder al cambio sobre seguir un plan.

Tabla 2.2. Mapeo entre los principios de OpenUp y el manifiesto ágil

Fuente: The Eclipse Foundation, 2006

También, teniendo en cuenta lo descrito anteriormente, es necesario mostrar que el contenido de OpenUp se centra en las siguientes disciplinas [The Eclipse Foundation, 2006]

ENTORNO

Esta disciplina explica como personalizar el proceso y las herramientas para un proyecto u organización

GERENCIA DE PROYECTO

Esta disciplina explica como entrenar, facilitar y apoyar al equipo, ayudando a hacer frente a los riesgos y los obstáculos encontrados en la construcción de software. Los objetivos de esta disciplina son:

Fomentar consenso entre los interesados para priorizar la secuencia de trabajo.

Estimular la colaboración en equipo en la creación de planes a largo plazo y corto plazo del proyecto.

Centrarse en que el equipo entregue continuamente software probado para la evaluación de los Stakeholders.

Ayudar a crear un entorno de trabajo eficaz para maximizar la productividad de los equipos.

Mantener informadas a las partes interesadas y al equipo del avance del proyecto.

Proporcionar un marco para gestionar los riesgos del proyecto y continuamente adaptarse a los cambios.

REQUERIMIENTOS

Esta disciplina explica como obtener, analizar, especificar, validar y gestionar los requisitos para el sistema a desarrollar. El propósito de esta disciplina es:

- Entender el problema a ser resuelto
- Entender las necesidades de los interesados directos (lo que los usuarios quieren).
- Definir los requisitos para la solución (lo que el sistema debe hacer).
- Definir los límites (alcance) del sistema.
- Identificar interfaces externas para el sistema.
- Identificar las limitaciones técnicas de la solución.
- Proporcionar las bases para la planificación de iteraciones.
- Proporcionar las bases iniciales para la estimación de costos y cronograma.

ARQUITECTURA.

Esta disciplina explica como crear una arquitectura de software a partir de los requisitos de gran importancia arquitectónicos. El propósito de esta disciplina es desarrollar una arquitectura robusta para el sistema. La arquitectura se construye en la disciplina de desarrollo.

DESARROLLO

Esta disciplina explica como diseñar e implementar una solución técnica que ajuste a la arquitectura y que sea compatible con los requerimientos. Tiene los siguientes objetivos:

- Transformar los requisitos en un diseño del futuro sistema.
- Adaptar el diseño para que coincida con el entorno de ejecución.

- Construir el sistema de forma incremental.
- Verificar que las unidades técnicas utilizadas para construir el sistema funcionen como se especifica.
- Con cada iteración, las tareas de esta disciplina se desarrollan para generar una versión de la construcción cada vez con más capacidad y más estable.

Al trabajar en el sistema, los desarrolladores usarán y estarán limitados por la arquitectura.

PRUEBAS

Esta disciplina define el conjunto mínimo de tareas necesarias para planificar, implementar, ejecutar y evaluar las pruebas de un sistema. El propósito de esta disciplina es:

- Buscar defectos en los documentos.
- Validar las suposiciones formuladas en las especificaciones de diseño y los requisitos a través de la demostración concreta.
- Validar que los requisitos se aplican adecuadamente.
- Validar que el producto de software funciona como fue diseñado.

DESPLIEGUE

Esta disciplina explica como planificar y desplegar una solución.

2.4.1. OPENUP/BASIC

OpenUP/Basic es un subconjunto de OpenUP que da un acercamiento ágil para el desarrollo del software, con solo un contenido fundamental provee un conjunto simplificado de artefactos, roles, tareas y guías de trabajo [The Eclipse Foundation, 2006].

Con un grupo de desarrollo, la clave del desafío de administración es siempre mantener una unidad e integridad de diseño. OpenUP/Basic es un proceso iterativo del desarrollo del

software para equipos de desarrollo pequeños y que le dan valor a la colaboración y a las necesidades de los Stakeholder, es un proceso mínimo, completo, y extensible:

Mínimo: Incluyendo únicamente el contenido del proceso fundamental

Completo: Tiene la posibilidad de ser declarado como todo un proceso para construir un sistema.

Extensible: Tiene la posibilidad de ser utilizado como base para agregar o para adaptar más procesos

El equipo tiene que participar a plenitud de la interacción diaria cara a cara. Los miembros del equipo incluidos los Stakeholders, desarrolladores, arquitectos, gestores de proyecto y testers.

Los miembros del equipo participan en una colaboración significativa, tomando sus propias decisiones en cuanto a lo que se necesita trabajar, cuales son las prioridades, y la mejor manera de abordar las necesidades de los Stakeholders. La presencia de los Stakeholders como miembros del equipo es crítica para realizar exitosamente OpenUP/Basic. Los miembros del equipo participan a diario en las reuniones stand-up para comunicar el estado y sus asuntos. Los problemas se abordan fuera de las reuniones diarias.

El ciclo de vida de desarrollo de software de OpenUP/Basic

OpenUp es un proceso iterativo cuyas iteraciones se distribuyen a través de cuatro fases: Concepción, elaboración, construcción y transición. (Ver Figura 2.5)

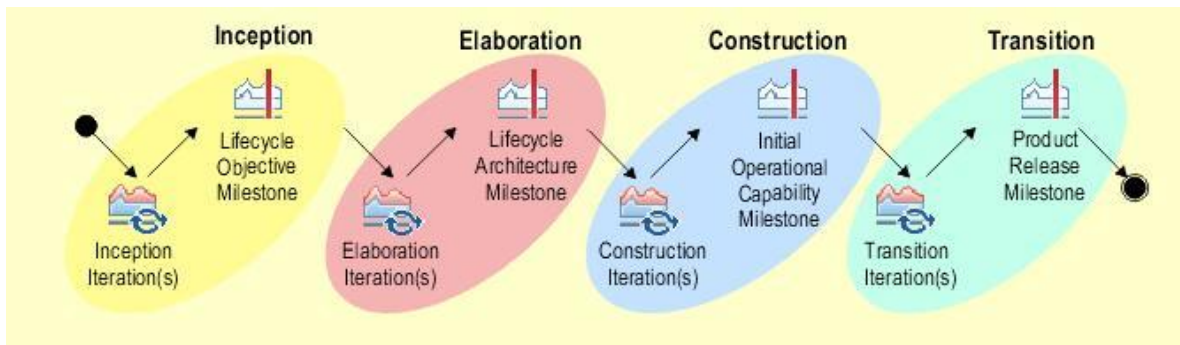


Figura 2.5 Flujo de trabajo
 Fuente: Colaboradores de Eclipse y otros, 2004, 2006

Cada fase podrá tener tantas iteraciones como se requiera dependiendo del grado de novedad del dominio de negocio, de la tecnología a ser utilizada, de la complejidad, de la arquitectura de la solución y del tamaño del proyecto, entre otros factores.

Para que los equipos de trabajo planeen sus iteraciones de forma rápida, el OpenUp/Basic provee plantillas con estructuras para la división del trabajo (EDT) en cada iteración y una plantilla de EDT para todo el proceso considerado de principio a fin.

Las iteraciones pueden tener duraciones variables dependiendo de las características del proyecto. Iteraciones de un mes son recomendadas típicamente ya que este periodo de tiempo provee:

- Una cantidad de tiempo razonable para que los proyectos entreguen incrementos considerables en funcionalidad.
- Retro alimentación temprana y frecuente por parte de los usuarios.
- Administración a tiempo de los riesgos y problemas encontrados durante el curso del proyecto.
- El OpenUp/Basic esta diseñada para ofrecer una guía en el proceso de desarrollo en proyectos pequeños:
- Equipos de 3 a 6 personas.
- 3 a 6 meses de trabajo. [The Eclipse Foundation, 2006].

2.4.2. FASES DE OPEN/UP

2.4.2.1.FASE DE INICIO O INCEPCIÓN

En esta fase se define el negocio y la factibilidad del proyecto que se va a realizar. Se describe el modelo de negocio, la visión, las metas del proyecto, se identifican actores, conceptos de dominio y deseos de usuario. Adicionalmente se define la viabilidad del proyecto.

2.4.2.2.FASE DE ELABORACIÓN

En la fase de elaboración se obtiene la visión refinada del proyecto identificando nuevos alcances, requisitos y estimaciones. Se tiene en cuenta la resolución de los riesgos más altos y de acuerdo a todos los factores podría llegarse a detener el proyecto por complejidad técnica.

2.4.2.3.FASE DE CONSTRUCCIÓN.

En la fase de construcción se realiza la implementación iterativa de elementos más sencillos y requisitos de menor riesgo. Representa la evolución hasta llegar a un producto terminado, incluyendo el 100% de los requisitos. Al final de esta fase el sistema contiene todos los casos de uso pactados entre cliente y la dirección. En caso de existir casos de uso no desarrollados, estos se convertirán en nuevas iteraciones.

2.4.2.4.FASE DE TRANSICIÓN

En esta fase se entrega el producto terminado al cliente, se instala y comienza el ciclo de pruebas.

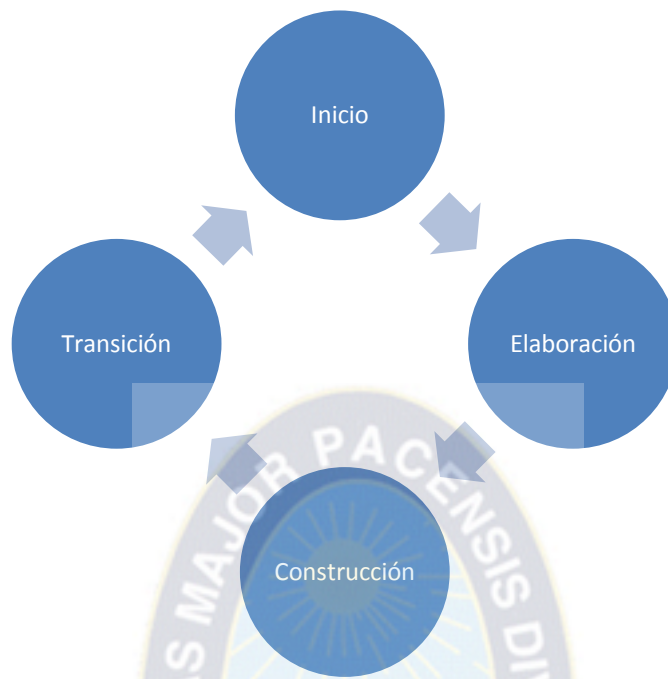


Figura 2.6 Fases de OpenUp
 Fuente: The Eclipse Foundation, 2012

2.4.3. DISCIPLINAS DE OPEN/UP

A continuación se describen las disciplinas que contempla Open/Up:

2.4.3.1. ANÁLISIS Y DISEÑO

Esta disciplina explica cómo crear un diseño desde los requerimientos que pueden ser implementados por los desarrolladores.

Los propósitos del Análisis y Diseño son:

Transformar los requerimientos en un diseño en que se construirá el sistema.

- Desarrollar una arquitectura sólida para el sistema.
- Adaptar el diseño para que coincida con el entorno de implementación.

2.4.3.2.GESTIÓN DE CONFIGURACIÓN Y CAMBIOS

Esta disciplina explica cómo controlar los cambios en artefactos, asegurando una evolución sincronizada del conjunto de Productos de trabajo que componen un sistema de software.

El objetivo de esta disciplina es:

- Mantener un conjunto consistente de productos de trabajo a medida que evolucionan.
- Proporcionar un medio eficiente para adaptarse a los cambios y los problemas y el trabajo de re-planificar en consecuencia.
- Proporcionar datos para medir el progreso.

2.4.3.3.IMPLEMENTACIÓN

Esta disciplina se explica cómo implementar una solución técnica que cumpla con el diseño, trabaja dentro de la arquitectura y es compatible con los requisitos.

El objetivo de esta disciplina es:

- Incrementos para construir el sistema.
- Verifique que las unidades técnicas utilizadas para construir el sistema funcionen como se especifica.

2.4.3.4.GESTIÓN DE PROYECTO

El objetivo de esta disciplina es:

- Ayudar a priorizar la secuencia de trabajo.
- Ayudar a crear un entorno de trabajo eficaz para maximizar la productividad del proyecto.
- Mantener las partes interesadas y que el equipo informe sobre los avances del proyecto.

- Proporcionar un marco para gestionar los riesgos del proyecto y continuamente adaptarse al cambio.

2.4.3.5.REQUERIMIENTOS

Esta tarea define las tareas mínimas necesarias para obtener, analizar, especificar, validar y gestionar los requisitos para el sistema a desarrollar.

El objetivo de esta disciplina es:

- Entender el problema a resolver.
- Entender las necesidades de las partes interesadas
- Definir los requisitos de la solución.
- Definir los límites del sistema.
- Identificar las limitaciones técnicas de la solución.

2.4.3.6.PRUEBA

Esta tarea define las tareas mínimas necesarias para obtener, analizar, especificar, validar y gestionar los requisitos para el sistema a desarrollar.

2.5. INGENIERÍA WEB

Según Pressman, la ingeniería Web (IWeb) aplica “sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitosos de sistemas y aplicaciones basados en Web de alta calidad” [Pressman, 2006].

Actualmente existen millones de sitios Web que cubren diferentes necesidades de la vida, seguramente la comunicación es la más importante teniendo en cuenta que las distancias pasaron a segundo plano.

El avance de internet y las nuevas tecnologías de comunicación, marcan una nueva tendencia en el mercado del software. La exigencia de los usuarios, cada vez más

numerosos, hizo que Internet creciera en forma acelerada y desprolija, produciendo un impacto social, económico y político impensado.

2.6. METODOLOGÍAS WEB

Algunas metodologías de desarrollo de aplicaciones Web como OOHDM (Schwabe & Rossi, 2000), OO-H (Cachero & Gómez, 2003) y OOWS (Fons, Pastor, Valderas & Ruiz, 2002) proponen todas las fases necesarias para el desarrollo de una aplicación Web, desde el levantamiento de los requerimientos, hasta el desarrollo de la aplicación, no toman en cuenta la fase de prueba e implantación del aplicativo. UWE (Koch & Kraus, 2002), WebML (Ceri, Fraternali, & Bongio, 2000) y NDT (Escalona, Torres & Reina, 2002), sólo contemplan las fases de levantamiento de requerimiento, análisis y diseño. Algunas de estas metodologías utilizan esquemas de modelado haciendo uso de notaciones estándares tales como UML y E-R. Otras proponen la combinación de notaciones propias estándares.

2.6.1. UML-BASED WEB ENGINEERING (UWE)

UWE (Ingeniería Web basada en UML), desarrollado por Nora Koch, del Instituto de Informática de la Universitat Manchen de Alemania, es un método de desarrollo de aplicaciones Web basado en UML estándar.

Además de la notación UML, utilizan el proceso unificado de desarrollo de software (Rational Unified Process o RUP) como metodología para realizar aplicaciones hipermedia, por lo cual el proceso es iterativo e incremental. El método es muy similar a OOHDM, y la principal diferencia radica en la notación.

La estrategia de diseño UWE se basa en modelos que se construyen durante la fase de análisis, principalmente el modelo conceptual y el modelo de procesos.

UWE introduce clases específicas de procesos como parte de un modelo separado, que ofrece una interfaz al modelo de navegación. [Rodríguez, 2009]

UWE es un enfoque de ingeniería de software para el dominio Web con el objetivo de cubrir todo el ciclo de vida de desarrollo de aplicaciones Web. El aspecto clave que distinguen a UWE es la confianza en las normas. UWE (UML) es una propuesta basada en UML y en el proceso unificado para modelar aplicaciones Web. Esta propuesta está formada por una notación para especificar el dominio (basada en UML) y en un modelo para llevar a cabo el desarrollo del proceso de modelado. Los sistemas adaptativos y la sistematización son dos aspectos sobre los que se enfoca UWE. Además de estar considerado como una extensión del estándar UML, también se basa en otros estándares como por ejemplo: XMI como modelo de intercambio de formato, MOF para el meta-modelado, los principios de modelado de MDA, el modelado de transformación del lenguaje QVT y XML. [Quispe, 2011].

2.6.1.1.PROCESO

1. Análisis de requisitos: El análisis de requisitos se expresa a través de la especificación de los casos de uso del sistema.
2. Diseño conceptual: En esta etapa se representa el dominio del problema con un diagrama de clases de UML. Los casos de uso sirven como entrada para elaborar tarjetas Clase-Responsabilidad-Colaborador (CRC), o para la identificación de verbos y sustantivos, entre otras técnicas, que permiten determinar las clases, métodos y atributos.
3. Diseño navegacional: tiene dos etapas (1) la definición del espacio de navegación y (2) el diseño de las estructuras de navegación.

(1) La definición del espacio de navegación se trata de una vista del diagrama conceptual, se define mediante el diagrama de clases UML y se distinguen con el estereotipo NC.

Los Espacios navegacionales tienen las siguientes características:

Las clases conceptuales que son importantes para el usuario, permanecen en el modelo navegacional.

Las clases que no se visitan, pero que contienen atributos importantes, no aparecen en el modelo navegacional, y sus atributos se muestran como parte de otras clases.

En el caso de vistas complejas se emplea Object Query Language para construirlas.

Para evitar caminos navegacionales profundos, se incorporan al modelo de navegación, asociaciones adicionales que están etiquetadas con un estereotipo y representan la navegación directa entre clases.

Las composiciones en el diagrama de clases navegacionales son interpretadas como la creación de un nodo de hipermedia compuesto, en la que varios nodos se muestran juntos.

(2) El diseño de las estructuras de navegación establece las estructuras de acceso que permiten visitar los objetos del espacio navegacional. Están constituidas por menús, índices, visitas guiadas, y formularios. Todos ellos son clases con estereotipos. La notación de las estructuras de acceso es similar a la de RMM y se muestra en la figura 2.7.

Los índices tienen referencias a una colección de objetos, y permiten la navegación directa a ellos.

Las visitas guiadas contienen una colección de referencias, y permiten la navegación secuencial a través de la misma. Los índices y visitas guiadas pueden definir la colección de objetos a la que están asociados de forma dinámica mediante el uso de formularios de entrada y condiciones de selección. Por supuesto, los índices y visitas guiadas pueden referirse a colecciones fijas de objetos.

Un menú es un objeto navegacional que tiene un número fijo de asociaciones a estructuras de acceso u objetos.

Un formulario permite al usuario ingresar información para completar las condiciones de selección de objetos pertenecientes a las colecciones de índices y visitas guiadas.

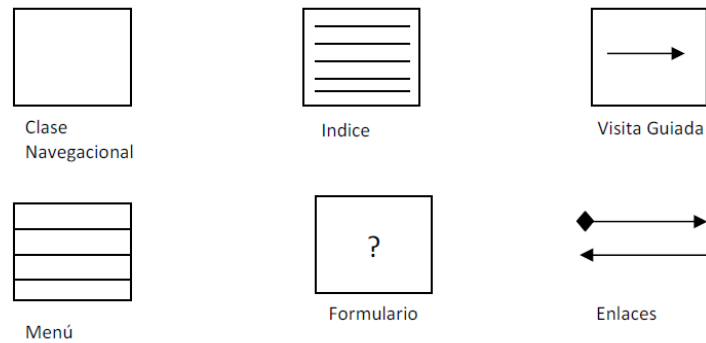


Figura 2.7 Notación UWE
Fuente: Rodríguez, 2009

(3) Diseño de la presentación: El modelo de presentación UWE está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos UML. Los elementos del modelo de presentación son: anclas, entradas de texto, imágenes, audio y botones. Cada clase del modelo navegacional tiene asignada una clase del modelo de presentación; las clases del modelo de presentación son equivalentes a las ADV de OOHDM.

A continuación se detallan cada uno de los modelos correspondientes a UWE, los mismos serán aplicados un determinado ejemplo, para fines prácticos.

2.6.1.2. ANÁLISIS DE REQUISITOS (CASOS DE USO)

Una de las primeras actividades en la construcción de aplicaciones Web es la identificación de los requisitos, y en UEW se especifican mediante el modelo de requerimientos, que involucra el modelado de casos de uso con UML.

En UWE el modelado de requisitos consiste en dos partes:

- Casos de uso
- Actividades describiendo los casos de uso a detalle.

Casos de uso

Los casos de uso se utilizan para visualizar las diferentes funcionalidades que la aplicación tiene que proporcionar, como por ejemplo: crear a un nuevo usuario, identificar al usuario, realizar una búsqueda, realizar la composición de un nuevo objeto y guardar el objeto compuesto. [Nieves, Ucan, Menendez, 2014]

El diagrama de casos de uso está conformado por los elementos actor y caso de uso.

- Actores. Los actores representan un tipo de usuario de sistema. En los diagramas de casos de uso, los actores se dibujan como una silueta humana.
- Casos de uso. Un caso de uso es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando.

Se representa mediante un ovalo.

Cada uno de los casos de uso debe detallarse, habitualmente con una descripción textual.

- Asociaciones. Hay una asociación mediante un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso.
- Actividades. Como con casos de uso solamente es posible capturar poca información, cada caso de uso puede ser descrito más detalladamente mediante un proceso. Es decir, las acciones que son parte de un caso de uso así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades.

A continuación se muestra un ejemplo de casos de uso, aplicado a un portal de música en la Web.

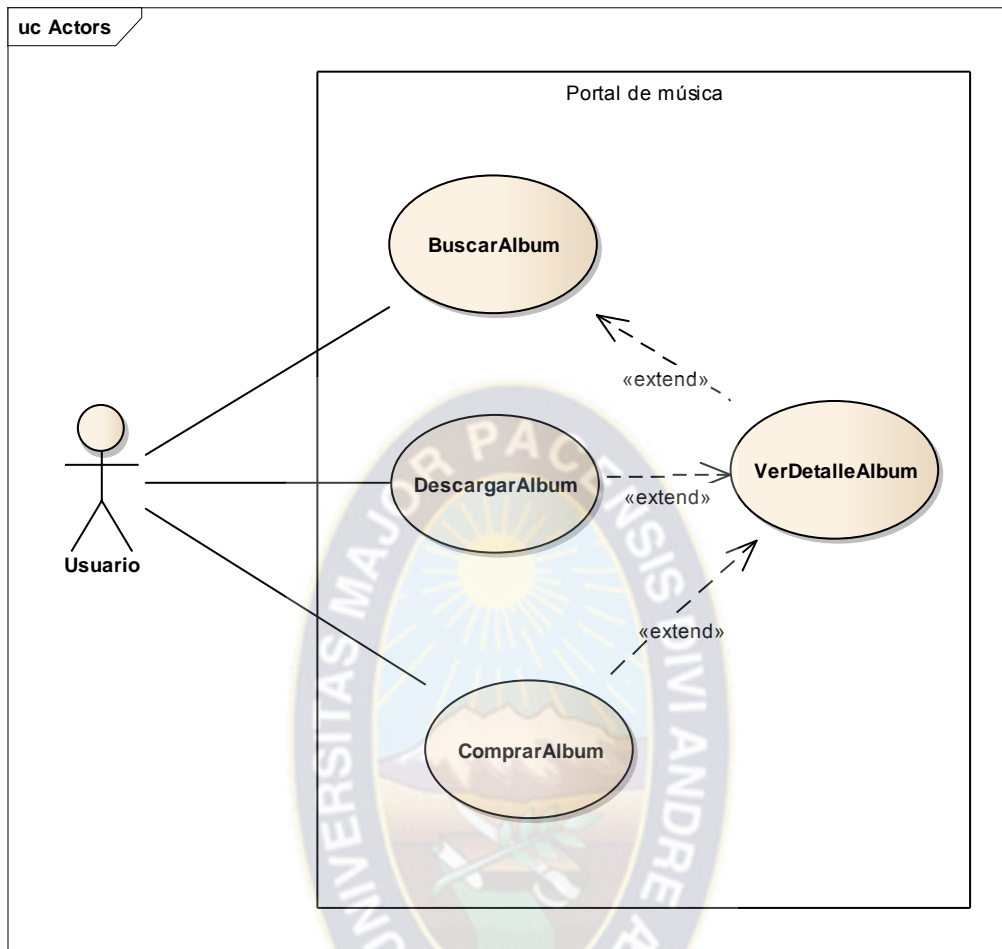


Figura 2.8 Caso de uso
Fuente: Nieves, Ucán, Menéndez, 2014

2.6.1.3. MODELO DE CONTENIDO

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio Web. [Quispe, 2011]

Este es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias para el caso de estudio presentado.

Siguiendo el ejemplo presentado en el punto 2.6.1.2. Casos de uso, se muestra el modelo de contenido para el portal de música.

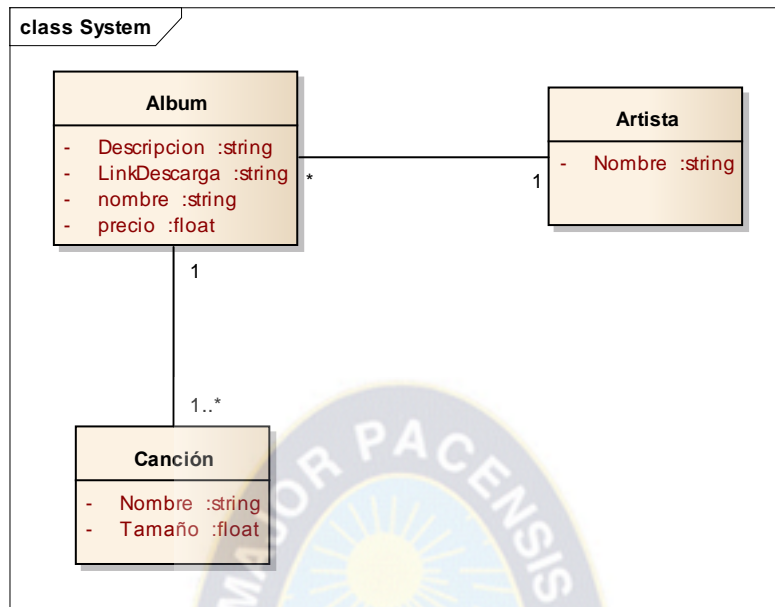


Figura 2.9 Modelo de contenido
Fuente Nieves, Ucán, Menéndez, 2014

2.6.1.4. MODELO DE USUARIO

La diferencia entre el modelo de contenido y este modelo es que el modelo de contenido está relacionado a los datos almacenados por la aplicación, mientras el modelo de usuario tiene dos objetivos diferenciados:

Contiene las clases que definen que información es almacenada en el contexto de una sesión. En este ejemplo, vemos que en una sesión puede haber un usuario actual, el mismo que puede tener una colección de álbumes de su propiedad. Por otra parte, las clases en el modelo de usuario proporcionan operaciones que se pueden utilizar en los procesos de negocio.

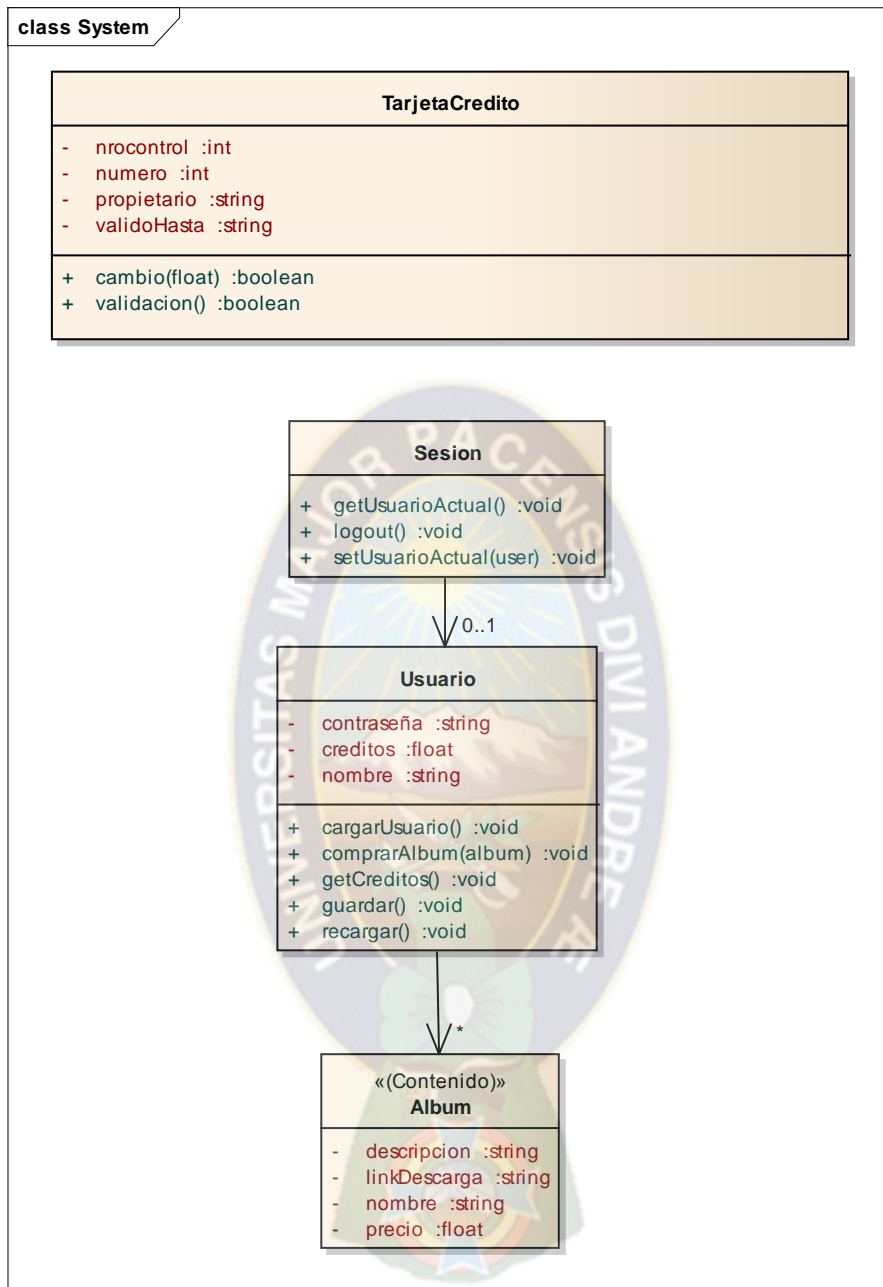


Figura 2.10 Modelo de usuario
Fuente Nieves, Ucán, Menéndez, 2014

2.6.1.5.MODELO DE NAVEGACIÓN

Este modelo indica como el sistema de páginas Web del sitio está relacionado internamente.

Es decir cómo se enlazan los elementos de navegación. [Quispe, 2011]

En una aplicación para la Web es útil saber cómo están enlazadas las páginas. Ello significa que se requiere un diagrama de navegación con nodos y enlaces. Este diagrama se modela con base en el análisis de los requisitos y el modelo de contenido. [Nieve, Ucan, Menéndez, 2014]

UWE provee diferentes estereotipos para el modelado de navegación:

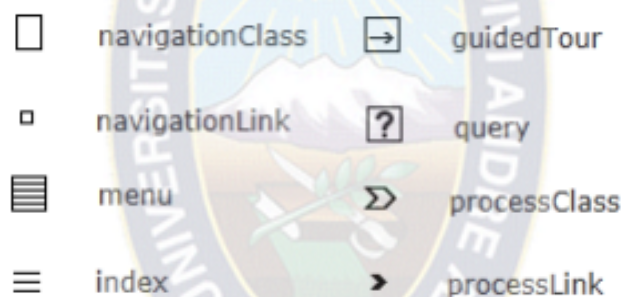


Figura 2.11 Estereotipos de estructura de navegación
Fuente: Nieve, Ucan, Menéndez, 2014

Las clases de navegación («navigationClass») representan nodos navegables de la estructura de hipertexto; los enlaces de navegación («navigationLink») muestran vínculos directos entre las clases de navegación; las rutas alternativas de navegación son manejadas por menú («menu»). Los accesos se utilizan para llegar a múltiples instancias de una clase de navegación («index» o « guidedTour») o para seleccionar los elementos («query»). Las clases de procesos («processClass») forman los puntos de entrada y salida de los procesos de negocio en este modelado y la inculación entre sí y a las clases de navegación se modela por enlaces de procesos («processLink»). [Nieves, Ucan, Menéndez, 2014].

Para el ejemplo portal de música, el modelo de navegación es el siguiente:

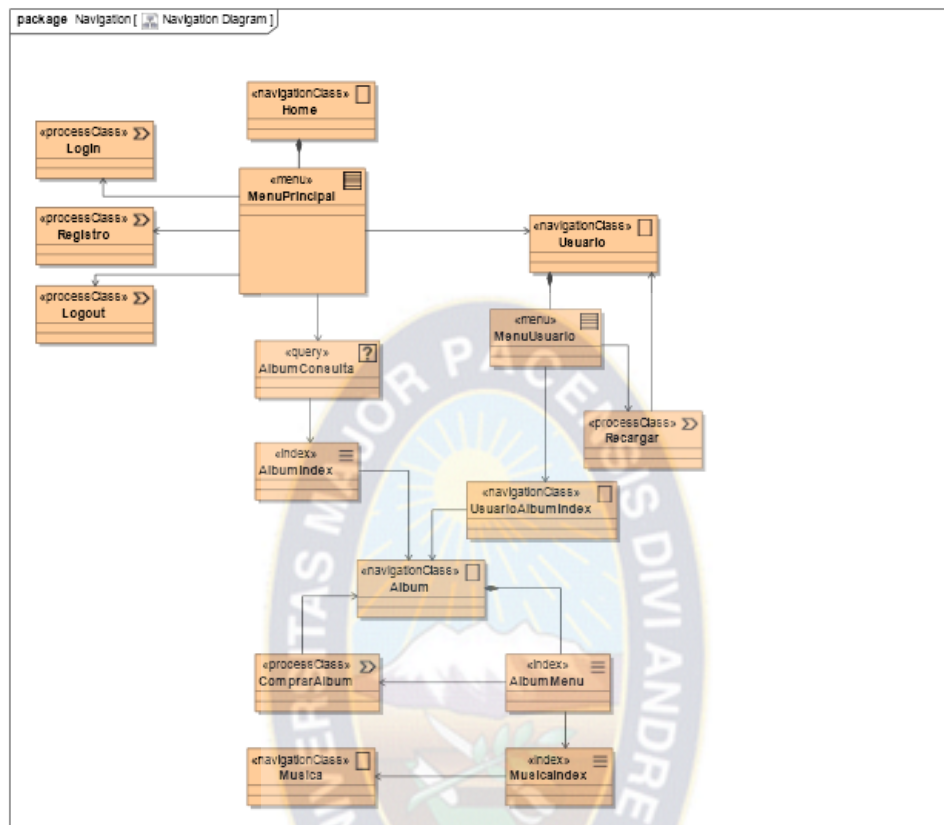


Figura 2.12 Modelo de navegación
Fuente: Nieves, Ucán, Menéndez, 2014

2.6.1.6. MODELO DE PRESENTACIÓN

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario (IU). Describe la estructura básica de la IU, es decir, ¿qué elementos de interfaz de usuario (por ejemplo texto, imágenes, enlaces, formularios) se utilizan para presentar los nodos de navegación? Su ventaja es que es independiente de las técnicas actuales que se utilizan para implementar un sitio Web, lo que permite a las partes interesadas discutir la conveniencia de la presentación antes de que realmente se aplique.

Una clase de presentación está compuesta de elementos de IU como texto («text»), enlaces (anchor), botones («button»), imágenes («image»), formularios («form») y colecciones de enlaces («anchored collection»).[Nieves, Ucan, Menéndez, 2014]

A continuación se presenta parte del modelo de presentación, siguiendo el ejemplo del portal de música.

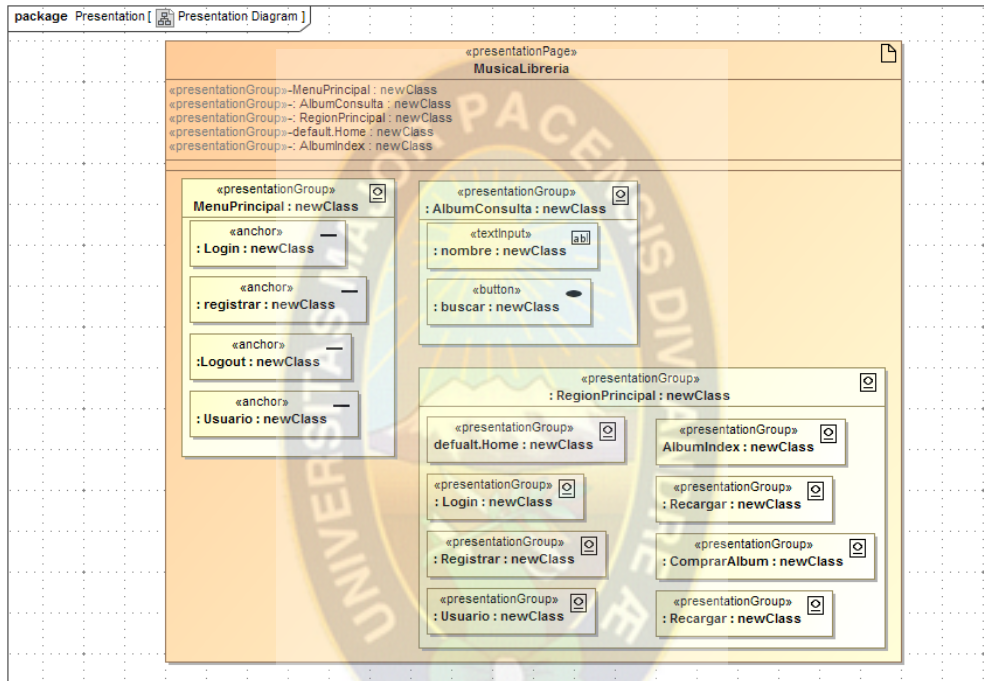


Figura 2.13 Modelo de presentación
Fuente: Nieves, Ucan, Menéndez, 2014

2.6.1.7. MODELO DE PROCESO

La estructura de navegación puede ser extendida mediante clases de procesos que representan la entrada y salida de procesos de negocio. El modelo de proceso representa el aspecto que tienen las acciones de las clases de proceso. En este modelo se tienen dos tipos de modelos:

Modelo de estructura del proceso, que describe las relaciones entre las diferentes clases de proceso, y

Modelo de flujo del proceso, que especifica las actividades conectadas con cada «processClass».

A continuación se describen cada uno de ellos:

Modelo de estructura del proceso. Es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso.

Modelo del flujo del proceso. Siguiendo el principio de la utilización de UML se han refinado los requisitos con los diagramas de actividad UML. Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades.

A continuación se presenta el modelo de procesos, siguiendo el ejemplo portal de música, tomado como ejemplo en la generación de todos los diagramas correspondientes a UWE.

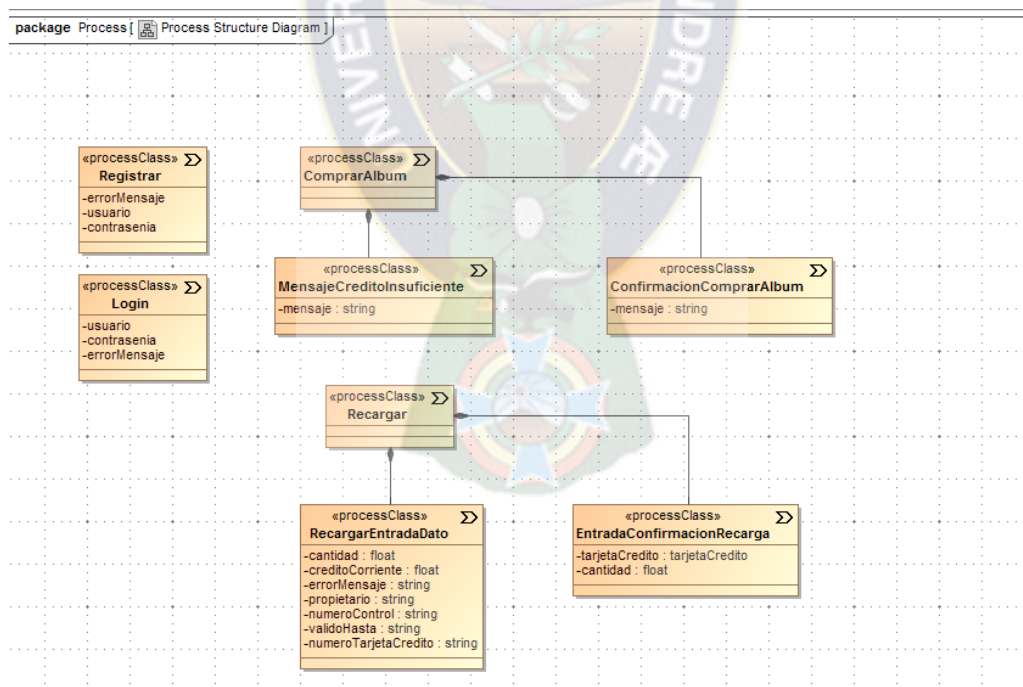


Figura 2.14 Modelo de procesos
Fuente: Nieves, Ucán, Menéndez, 2014

El siguiente ejemplo muestra el diagrama de actividad, en función al “Login” presentado en el diagrama de procesos.

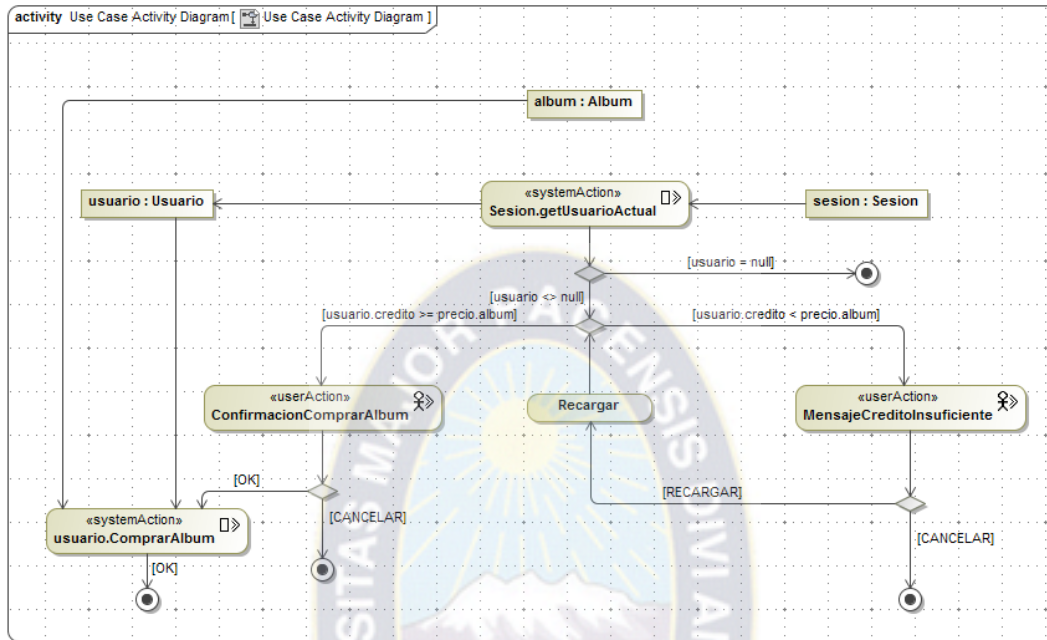


Figura 2.15 Diagrama de actividad Login
Fuente: Nieves, Ucán, Menéndez, 2014

2.7. HERRAMIENTAS

2.8. PATRÓN MODELO VISTA CONTROLADOR (MVC)

Modelo Vista Controlador (MVC), es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones WEB, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción del patrón

Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facultando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos; en líneas generales del MVC corresponde al modelo.

La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y accesos a datos, el MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la vista y el controlador en paralelo, ya que ambos cumplen ciclos de vida distintos.

2.9. HYPERTEXT PRE-PROCESSOR (PHP)

PHP (Hypertext Pre-processor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Existen principalmente tres campos principales donde se usan scripts de PHP.

Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Se necesitan tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor, con una instalación de PHP conectada. Se puede acceder al resultado del programa PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.

Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente usando cron (en Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.

Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

2.10. POSTGRES

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS,s) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesado de datos. El modelo relacional sustituyó modelos previos en parte por su "simplicidad espartana". Sin embargo, como se ha mencionado, esta

simplicidad también hace muy dócil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema

- Clases
- Herencia
- Tipos
- Funciones
- Sistema Gestor de Base de Datos PostgreSQL

PostgreSQL está basado en una arquitectura cliente servidor. El programa servidor se llama Postgres y entre los muchos programas cliente tenemos, por ejemplo, pgbrowser (un cliente gráfico) y psql (un cliente en modo texto). Un proceso servidor Postgres puede atender exclusivamente a un solo cliente; es decir, hacen falta tantos procesos servidor Postgres como clientes haya. El proceso Postmaster es el encargado de ejecutar un nuevo servidor para cada cliente que solicite una conexión es así que siempre se ejecuta solamente un proceso postmaster y múltiples procesos Postgres. Los clientes pueden ejecutarse en el mismo sitio o en equipos remotos conectados por TCP/IP.

2.11. FRAMEWORK CAKEPHP

CakePHP es un framework de desarrollo rápido que usa los más conocidos diseños de patrones y provee un framework estructurado que habilita a los usuarios de php a usarlo en todos los niveles para desarrollar aplicaciones web robustas rápidamente sin ninguna pérdida de flexibilidad. Esto significa que se puede desarrollar una aplicación mucho más rápido, el código es mejor y permite realizar aplicaciones tipo web 2.0 sorprendentemente rápido.

Sigue fuertemente en su estructura el patrón de Modelo -Vista- Controlador (MVC).

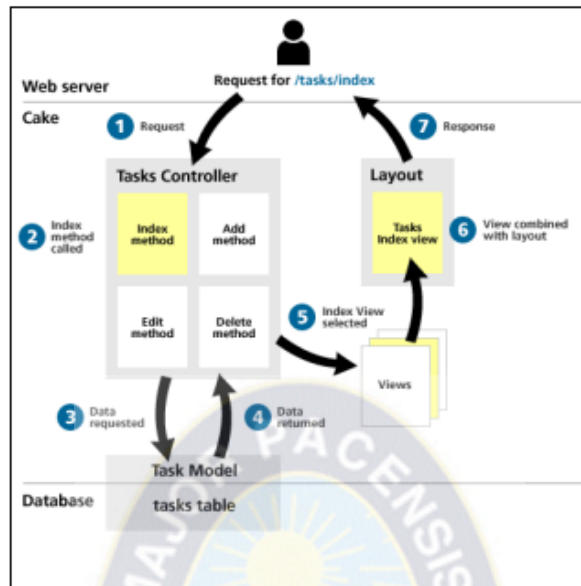


Figura 2.16 Esquema de cakePHP
Fuente: Bolívar, Hernández, Parra, Silva, 2011

Las ventajas que nos ofrece CakePHP son muchas ya que posee unas estructuras que permite a los desarrolladores crear aplicaciones robustas, de gran desempeño y muy fáciles de administrar.

Entre las muchas ventajas que ofrece CakePHP se pueden mencionar las siguientes:

Operaciones CRUD incorporadas: El desarrollo de aplicaciones Web esta intrínsecamente relacionado con el uso de base de datos y por supuesto operaciones sobre las mismas. CRUD por sus siglas en ingles significan create, read, update y delete, que son todas las operaciones que podemos realizar en una base de datos. Cake hace que la implementación del CRUD sea algo sencillo, ya que el framework incorpora estas funciones de forma automática, en lugar de codificar cada una de las operaciones CRUD a mano, cake proporciona a los desarrolladores un conjunto de clases que realizan esta tarea, generando el código para cada operación basado en la estructura de tablas de la base de datos que el desarrollador haya especificado.

Convenciones y organización: CakePHP posee una organización que permite a los programadores el desarrollo de aplicaciones de manera ordenada, ya que todos los desarrolladores que hagan parte de un mismo proyecto, deben seguir las mismas convenciones y reglas para el desarrollo de la aplicación, algo que en la programación en equipo suele ser una gran problema sobre todo cuando se trata de un lenguaje de programación tan flexible como lo es php. Cake contribuye a esta organización proporcionando un conjunto de convenciones y reglas muy fáciles de aprender e implementar.

Soporte de scaffolding: El scaffolding es un método que se utiliza para crear aplicaciones basadas en bases de datos. crear aplicaciones web será mucho más rápido y fácil usando las habilidades que tiene CakePHP para utilizar este método, ya que con solo escribir una línea de código se puede invocar el scaffolding, el cual hace un mapeo de todos los elementos de nuestra base de datos en elementos visuales, es decir, convierte las columnas de las tablas de nuestra base de datos en elementos de un formulario HTML , sin escribir ni una sola línea de HTML y obviamente integrando todas las operaciones CRUD de la base de datos a nuestra aplicación.

Estructuración de Código: CakePHP nos permite organizar el código en segmentos modulares, lo cual les permite a los desarrolladores poder llevar a cabo rápidamente cualquier cambio que sea requerido, en consecuencia la modificación y la depuración se convierte en una labor mucho más sencilla.

Menos Configuración: Uno de los principios fundamentales dentro de la filosofía de diseño de CakePHP es tener un número mínimo de configuraciones, las configuraciones son normalmente requeridas para que el sistema funcione de un modo adecuado, pero con cake estas configuraciones son mantenidas al mínimo, únicamente lo que se requiere para que CakePHP funcione es indicarle el nombre de la base de datos con la que se va a trabajar y no hay que especificar nada más, ya que CakePHP es una framework basado en convenciones y no en configuraciones.

Autogeneración de código: Otra de las grandes ventajas y que acelera el desarrollo de aplicaciones es la autogeneración de código, ya que al crear una aplicación en CakePHP ya se cuenta con una gran cantidad de código escrito sin que el programador tenga que modificar una sola línea de código. El generador de código incorporado en CakePHP se conoce con el nombre de baking script, lo único que se debe hacer es especificar las tablas de la base de datos para el proyecto, baking automáticamente crea los modelos, los controladores y las vistas que se necesitan para nuestra aplicación. [Bari, 2008 en Bolívar, Hernández, Parra, Silva, 2011].

2.12. CALIDAD DE SOFTWARE

La medición de la calidad de software se realiza a través de métricas de control de calidad, para medir aspectos del software como ser: funcionalidad, instalación/mantenibilidad y su portabilidad, los cuales se detallan a continuación:

2.12.1. FACTORES DE CALIDAD ISO 9126

Normas de gestión ISO 17799.- La norma NB-ISO/IEC 17799 define la seguridad de la información como la preservación de:

- Confidencialidad.- Solo quienes estén autorizados pueden acceder a la información.
- Integridad.- La integración y sus métodos de procesos son exactos y complejos.
- Disponibilidad.- Los usuarios autorizados tienen acceso a la información y a sus activos asociados cuando lo requieran.

Entre los aspectos fundamentales para la calidad de las aplicaciones Web destacamos:

- La medición y evaluación del producto software
- El control del proceso de desarrollo

Entre los estándares y modelos de evaluación y mejora de los procesos software que están relacionados con la calidad, en cualquiera de los términos analizados. Tenemos a ISO 9001

ISO 9001 – CALIDAD.- Es un modelo para conseguir la calidad total en el diseño, desarrollo, producción, instalación y servicio post-venta.

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica seis atributos clave de calidad:

- **Funcionalidad.** El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperabilidad, conformidad y seguridad.
- **Confiabilidad.** Cantidad de tiempo que el software está disponible para su uso. Esta referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- **Usabilidad.** Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- **Eficiencia.** Grado en que el software hace óptimo el uso de los recursos del sistema. esta indicado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- **Facilidad de mantenimiento.** La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- **Portabilidad.** La facilidad con que el software puede ser llevado de un entorno a otro. Esta referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Los factores ISO 9126 no necesariamente son utilizados para medidas directas. En cualquier caso, facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad del sistema.

2.13. SEGURIDAD

Cuando se habla de la función informática generalmente se tiende a hablar de tecnología nueva, de nuevas aplicaciones, nuevos dispositivos, nuevas formas de elaborar información más consistente, etc.

Sin embargo se suele pasar por alto o se tiene muy implícita la base que hace posible la existencia de los anteriores elementos. Esta base es la información.

Es muy importante conocer su significado dentro de la función informática, de forma esencial cuando su manejo está basado en tecnología moderna, para esto se debe ser confidencial para algunas personas o a escala institucional, puede ser mal utilizada o divulgada, puede estar sujeta a robos, sabotaje o fraudes.

2.13.1. RESGUARDO DE LA INFORMACIÓN

Efectuar copias de seguridad periódicas del sitio, con un esquema adecuado de frecuencia, tipo de resguardo, rotación y reutilización de medios de almacenamiento. Las copias deben ser almacenadas en un lugar físicamente seguro, para prevenir el posible daño, robo o pérdida de las mismas.

Probar periódicamente la correcta restauración de las copias de seguridad.

Efectuar copias de seguridad de los logs del sistema y almacenarlas según esquema de rotación lo suficientemente largo.

En los casos en los cuales se disponga la eliminación de información o bien la reutilización de medios de almacenamiento, efectuar la eliminación de la información de forma segura, en función a la criticidad de la misma.

2.13.2. CONTROL DE CAMBIOS

Gestionar los cambios de forma de garantizar de manera de garantizar que los mismos se efectúan de manera autorizada segura. Documentar e implementar procedimientos para la solicitud, evaluación, diseño, prueba e implementación de los cambios.

Implementar un mecanismo de solicitud de cambios formalizado.

Documentar e implementar un procedimiento de análisis de los cambios solicitados, incluyendo la evaluación del impacto del cambio sobre la seguridad de la información.

Probar los cambios junto al usuario para garantizar que cumplan con los requerimientos.

La implementación de cambios en el sitio (ya sea de diseño o contenido) debe encontrarse formalmente documentada y autorizada.

Mantener una bitácora de los cambios realizados en la configuración del sistema. Esta bitácora no debe ser almacenada en el mismo servidor y debe estar debidamente resguardada.

Documentar los procedimientos para solucionar problemas comunes y lo referido a mantenimiento periódico.

Mantener un entorno de desarrollo/prueba separado físicamente y lógicamente del sitio de producción.

CAPITULO III

MARCO APLICATIVO

3.1. INTRODUCCIÓN

En el presente capítulo se presenta el desarrollo del sistema, en el cual se especifican los pasos a seguir según la metodología OpenUp, definida en el marco teórico, así como también el uso de la UWE aplicado a la metodología de desarrollo, detallando los diagramas intervinientes.

A continuación se describe la interrelación de la metodología de desarrollo de software OpenUp con la metodología UWE (UML Basado en Ingeniería Web) y la arquitectura MVC (Modelo Vista Controlador):

Fase de inicio

DISCIPLINA	UWE
Requerimientos	- Modelo de requerimientos
Gestión de configuración y cambios	
Gestión de proyectos	No aplica

Figura 3.1 Relación fase de inicio UWE
Fuente: Elaboración propia

Fase de elaboración

DISCIPLINA	UWE
Análisis y diseño	<ul style="list-style-type: none"> - Modelo de contenido - Modelo de navegación - Modelo de presentación
Arquitectura	- Patrón MVC (Modelo Vista Controlador)

Gestión de configuración y cambios	No aplica
Gestión de proyectos	No aplica

Figura 3.2 Relación fase de elaboración UWE
Fuente: Elaboración propia

Fase de construcción

DISCIPLINA	UWE
Implementación	No aplica
Gestión de configuración y cambios	No aplica
Gestión de proyectos	No aplica

Figura 3.3 Relación fase de construcción UWE
Fuente: Elaboración propia

Fase de transición

DISCIPLINA	UWE
Prueba	No aplica
Gestión de configuración y cambios	No aplica
Gestión de proyectos	No aplica

Figura 3.4 Relación fase de transición UWE
Fuente: Elaboración propia

3.2. DESARROLLO DEL SISTEMA

En esta sección se describe el desarrollo del sistema basado en la metodología OpenUp desde el levantamiento de información con la institución y los interesados, pasando por el análisis de la información, la definición de la arquitectura, el diseño del prototipo hasta el desarrollo final del sistema.

Para el siguiente sistema se realizaron un total de 5 iteraciones, en las cuales se fue afinando el análisis y diseño del sistema, específicamente trabajando en las fases de inicio y

elaboración de la metodología de desarrollo OpenUp, de las 5 iteraciones realizadas se obtuvo como producto los siguientes entregables que se detallan a continuación.

3.2.1. CUADRO DE DOCUMENTOS ENTREGABLES

A continuación se presenta el cuadro de los documentos entregables, agrupados por las fases que presenta la metodología de desarrollo.

FASE	ACTIVIDAD	ENTREGABLE
Fase de inicio	Definición del tamaño de la solución	Descripción del problema
	Definición de la visión del proyecto	<ul style="list-style-type: none"> - Identificación de stakeholders - Definición de la posible solución con funcionalidad clave. - Visión general del sistema
Fase de elaboración	Análisis	<ul style="list-style-type: none"> - Cuadro de recursos tecnológicos - Modelo de requerimientos (UWE) - Diagramas de secuencia. - Modelo de contenido (UWE). - Modelo del dominio.
	Diseño	<ul style="list-style-type: none"> - Modelo Navegacional (UWE). - Modelo de presentación (UWE). - Diseño de interfaces. - Modelo Entidad Relación.
	Arquitectura	<ul style="list-style-type: none"> - Arquitectura MVC (Modelo Vista Controlador)

Fase de construcción	Selección de herramientas	- Justificación de las herramientas elegidas
	Implementación	- Implementación de la solución
Fase de transición	Prueba	- Prueba de la solución

Tabla 3.1 Cuadro de documentos entregables

Fuente: Elaboración propia

3.2.2. FASE DE INICIO

3.2.2.1. IDENTIFICACIÓN DE STAKEHOLDERS

Resulta importante la descripción de los usuarios en la fase de inicio de la metodología, ya que los mismos se beneficiaran de forma directa con el sistema que automatizara una serie de procesos y funciones de las cuales están encargados dentro de la institución.

La siguiente tabla muestra en grandes rasgos los principales interesados del sistema (Stakeholders):

NOMBRE	DESCRIPCIÓN	RESPONSABILIDADES
Personal del departamento IV	Persona que se encarga de administrar el registro de aviones en general, así como del manejo de la información total.	Realizar el registro de aeronaves pertenecientes a la FAB. Efectuar el control de los componentes de las aeronaves pertenecientes a la FAB. Realizar correcciones en registros emitidos por las unidades aéreas dependientes de la FAB.
Personal de las unidades	Persona que se encarga de la	Realizar el registro de las

aéreas	administración del control de los componentes de las aeronaves	horas y/o ciclos de vuelo de las aeronaves. Realizar el registro de la programación de mantenimiento de las aeronaves. Realizar la administración de los componentes de las aeronaves.
Personal Gerencia	Persona que visualiza la información del estado de las aeronaves.	Visualiza el estado de las aeronaves con referencia a las horas y/o ciclos de vuelo de las mismas, y la condición de las aeronaves en su totalidad.

Tabla 3.2 Identificación de stakeholders

Fuente: Elaboración propia

3.2.2.2.DEFINICIÓN DE LA POSIBLE SOLUCIÓN CON FUNCIONALIDAD

CLAVE

A continuación se brindara un resumen amplio de los problemas que serán resueltos, en beneficio de los interesados, en el desarrollo del sistema Web propuesto, tomando en cuenta a los involucrados del sistema.

Se presenta la tabla con los problemas a ser resueltos relacionados al involucrado Personal del departamento IV

Para	Personal del departamento IV
Quiénes	Se encargan de la administración de la información que

	se genera con respecto al control e información general de las aeronaves.
El	Sistema de seguimiento y control para componentes de aeronaves
Que	Proporciona el manejo de información de forma automatizada.
Nuestro producto	Pretende mejorar la eficiencia en la presentación de la información, haciendo que la misma se efectúe más rápidamente, de la misma manera ayudar a mejorar el control que se tiene con respecto a las aeronaves.

Tabla 3.3 Problemas relacionados al personal del departamento IV

Fuente: Elaboración Propia

A continuación se presenta la tabla con los problemas a ser resueltos relacionados al involucrado Personal de las unidades aéreas:

Para	Personal de las unidades aéreas
Quiénes	Se encargan del control de las aeronaves y de las horas y/o ciclos de vuelo de las mismas, así como también de la programación de mantenimiento de las aeronaves y sus respectivos componentes.
El	Sistema de seguimiento y control para componentes de aeronaves
Que	Pretende mejorar el proceso de control de horas y/o ciclos de vuelo de las aeronaves.

Nuestro producto	Brinda mecanismos para el cálculo automático de control con referencia a las horas y/o ciclos de vuelo de las aeronaves, así como también un control de mantenimiento de las mismas.
------------------	--

Tabla 3.4 Problemas relacionados al personal de las unidades
Fuente: Elaboración propia

Se presenta la tabla con los problemas a ser resueltos relacionados al involucrado Personal gerencia:

Para	Personal gerencia
Quiénes	Representa a la alta gerencia, que se encarga de la toma de decisiones correspondientes al mantenimiento de las aeronaves.
El	Sistema de seguimiento y control para componentes de aeronaves
Que	Tiene como fin, revisar el estado de las aeronaves en general, así mismo revisar los estados de los componentes de las aeronaves.
Nuestro producto	Brinda información actualizada sobre el estado de las aeronaves y sus componentes, así como también dar un aviso previo para mejorar la eficiencia del servicio que prestan las aeronaves.

Tabla 3.5 . Problemas relacionado al personal gerencia
Fuente: Elaboración propia

3.2.2.3.VISIÓN GENERAL DEL SISTEMA

A continuación se brindara una descripción de las características del sistema a nivel general, señalando las necesidades de la institución y la solución propuesta con la implementación del sistema web a desarrollar.

Necesidad	Control de asignación de aeronaves a las unidades aéreas.
Prioridad	Alta
Características	Importancia sobre la información del número de aeronaves y la descripción de las mismas asignada a las unidades aéreas dependientes de las brigadas aéreas.
Solución Sugerida	Implementación de un módulo que permita almacenar la información de las bases aéreas (Brigadas y unidades) así como también las aeronaves y la respectiva asignación de las mismas.

Tabla 3.6 Control de asignación de aeronaves a las unidades aéreas.

Fuente: Elaboración propia

Necesidad	Registro automatizado de las horas y/o ciclos de vuelo
Prioridad	Alta
Características	El tiempo de vida de una aeronave y sus respectivos componentes se miden en horas y/o ciclos de vuelo, las mismas vienen con unas horas y ciclos límite, para tal efecto debe realizarse el control mediante estos dos parámetros.
Solución Sugerida	Por medio del registro de horas y/o ciclos de vuelo a aeronaves, el procedimiento para componentes y subcomponentes, se realizaran de manera interna, haciendo del registro un proceso fácil.

Tabla 3.7 Registro automatizado de las horas y/o ciclos de vuelo

Fuente: Elaboración propia

Necesidad	Conocer el estado de las aeronaves de la FAB
Prioridad	Alta

Características	Es necesario conocer el estado de las aeronaves, para prever la compra o mantenimiento de los componentes afectados, así como también extraer información para medir la operatividad de la flota con la que cuentan.
Solución Sugerida	Control y alertas para las aeronaves y componentes, que avisen si necesitan un mantenimiento, o si está por llegar a superar las horas y/o ciclos límite de vuelo.

Tabla 3.8 Conocer el estado de las aeronaves de la FAB

Fuente: Elaboración propia

Necesidad	Controlar la programación de mantenimiento de las aeronaves.
Prioridad	Alta
Características	Las aeronaves tienen una programación de mantenimiento, ya sea por horas, ciclos o fechas en las cuales se debe efectuar un chequeo general , y si es necesario un mantenimiento a la aeronave como tal.
Solución Sugerida	Permitir el registro de una programación de mantenimiento, el mismo que controle y avise al usuario el tiempo exacto que falta para que una aeronave entre en inspección.

Tabla 3.9 Controlar la programación de mantenimiento de las aeronaves.

Fuente: Elaboración propia

3.2.3. FASE DE ELABORACIÓN

3.2.3.1. REQUERIMIENTOS TECNOLÓGICOS

A continuación, se describen los requerimientos tecnológicos necesarios para el desarrollo del sistema Web propuesto:

- RAM de 6GB
- Computador i3 de primera generación o superior
- Servidor web compatible con la plataforma del sistema
- Motor de base de datos Postgresql

- Lenguaje de programación PHP5, HTML5.
- Conexión a internet mínimo 512 Kb

3.2.3.2.ANÁLISIS

i. MODELO DE REQUERIMIENTOS

Los requerimientos funcionales fueron elaborados a partir de reuniones sostenidas con los usuarios involucrados de la institución, como resultado de las mismas se presentan los siguientes requerimientos que debe cumplir el sistema, organizados por módulo.

Tabla 3.11: Requerimientos funcionales modulo – Registro de Grandes , pequeñas unidades y aeronaves

Código del Requerimiento	Prioridad	Descripción del Requerimiento
RF-RT-01	Alta	El sistema debe permitir el registro de las grandes y pequeñas unidades dependientes de la FAB.
RF-RT-02	Alta	El sistema debe permitir el registro de aeronaves pertenecientes a las unidades aéreas
RF-RT-03	Media	El sistema debe permitir la emisión de reportes especificando las grandes y pequeñas unidades, así como las aeronaves asignadas a estas.

Tabla 3.10 Requerimientos funcionales-Registro de grandes, pequeñas unidades
Fuente: Elaboración Propia

Requerimientos funcionales – Componentes

Código del Requerimiento	Prioridad	Descripción del Requerimiento
--------------------------	-----------	-------------------------------

RF-CP-01	Alta	El sistema debe permitir realizar el registro de componentes hasta un 2do nivel, pertenecientes a las aeronaves.
RF-CP-02	Alta	El sistema debe permitir monitorear los pagos realizados por el paciente, a través de alarmas que indiquen el estado de cancelación de pagos de un tratamiento.
RF-CP-03	Media	El sistema debe permitir al usuario administrador, en caso de tratamientos especiales, seleccionar el tipo de caja a la entrara el pago cancelado por el paciente.

Tabla 3.11 Requerimientos funcionales-Componentes
Fuente: Elaboración Propia

Requerimientos funcionales – Seguimiento y control de componentes

Código del Requerimiento	Prioridad	Descripción del Requerimiento
RF-HC-01	Alta	El sistema debe permitir el registro de seguimiento a los componentes que forman parte de las aeronaves
RF-HC-02	Media	El sistema debe permitir realizar el seguimiento a los componentes que sirven de repuesto, de las aeronaves
RF-HC-03	Media	El sistema debe permitir el registro de las horas y/o ciclos de vuelo de las aeronaves.
RF-HC-04	Baja	El sistema debe permitir la visualización del estado de las aeronaves, en cuanto a horas y/o ciclos.

Tabla 3.12 Requerimientos funcionales-Seguimiento y control de componentes
Fuente: Elaboración Propia

Requerimientos funcionales – Programación de mantenimiento

Código del Requerimiento	Prioridad	Descripción del Requerimiento
RF-CM-01	Alta	El sistema debe permitir el registro de la programación de mantenimiento de las aeronaves.
RF-CM-02	Media	Se debe tener un control de mantenimiento, de acuerdo a la programación efectuada.

Tabla 3.13 Requerimientos funcionales-Programación de mantenimiento
Fuente: Elaboración Propia

• DESCRIPCIÓN DE ACTORES

Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier persona externa que interactúa con el sistema. El actor es un usuario que juega un rol con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Descripción de actores.

Actores	Definición
Personal de las unidades aéreas	El usuario Personal de las unidades aéreas, tiene como tarea el registro del control de las aeronaves, específicamente el registro de las horas y/o ciclos de vuelo de las aeronaves, así como de sus componentes, tanto mayores como rotables. También se encarga del registro de la programación de mantenimiento,

	y del control del mismo.
Personal del departamento IV	El usuario Personal del departamento IV, se encarga del registro de las grandes y pequeñas unidades aéreas, así como también del registro de las aeronaves que forman parte de cada unidad. Al mismo tiempo goza de los privilegios del usuario Personal de las unidades aéreas.
Personal gerencia	El usuario Personal gerencia, tiene acceso a la visualización de toda la información generada, así como también de los reportes facilitados por el sistema, dicho usuario no puede realizar cambios de los datos generados por el usuario Personal del departamento IV ni por el usuario Personal de las unidades aéreas.

Tabla 3.14 Descripción de actores

Fuente: Elaboración Propia

- **ESPECIFICACIÓN DE CASOS DE USO**

A continuación se detallan las especificaciones de los casos de uso.

CASO DE USO PRINCIPAL

El caso de uso principal, que corresponde al sistema es el siguiente:

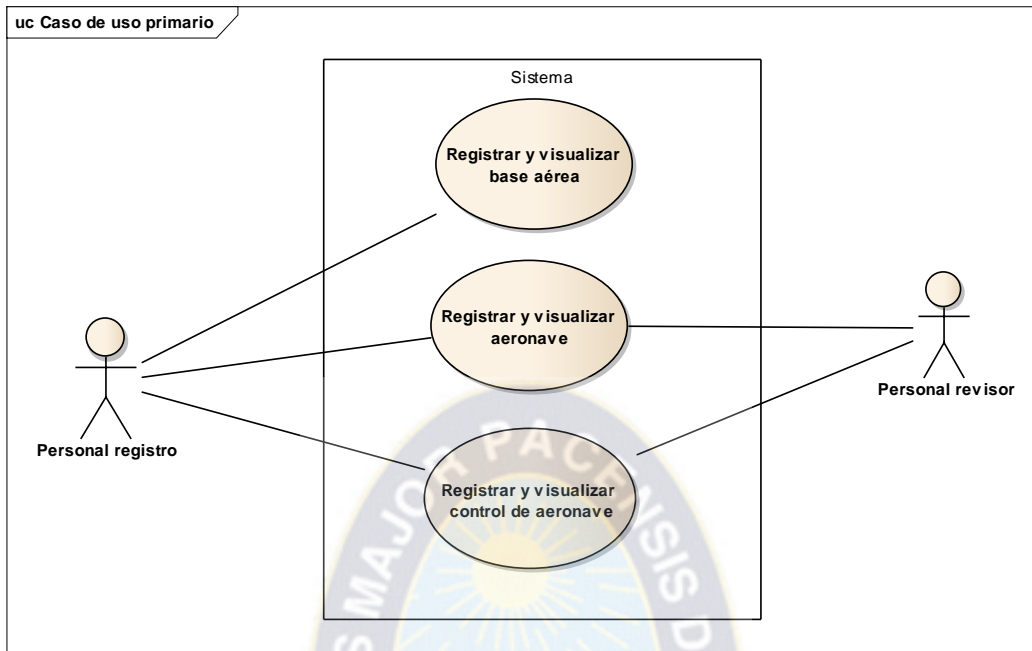


Figura 3.5 Caso de uso principal
Fuente. Elaboración propia

CASOS DE USO SECUNDARIOS

A continuación se muestran los casos de usos secundarios, correspondientes al caso de uso principal:

Registrar y visualizar base aérea

A continuación se presenta el caso de uso correspondiente al registro y visualización de base aérea.

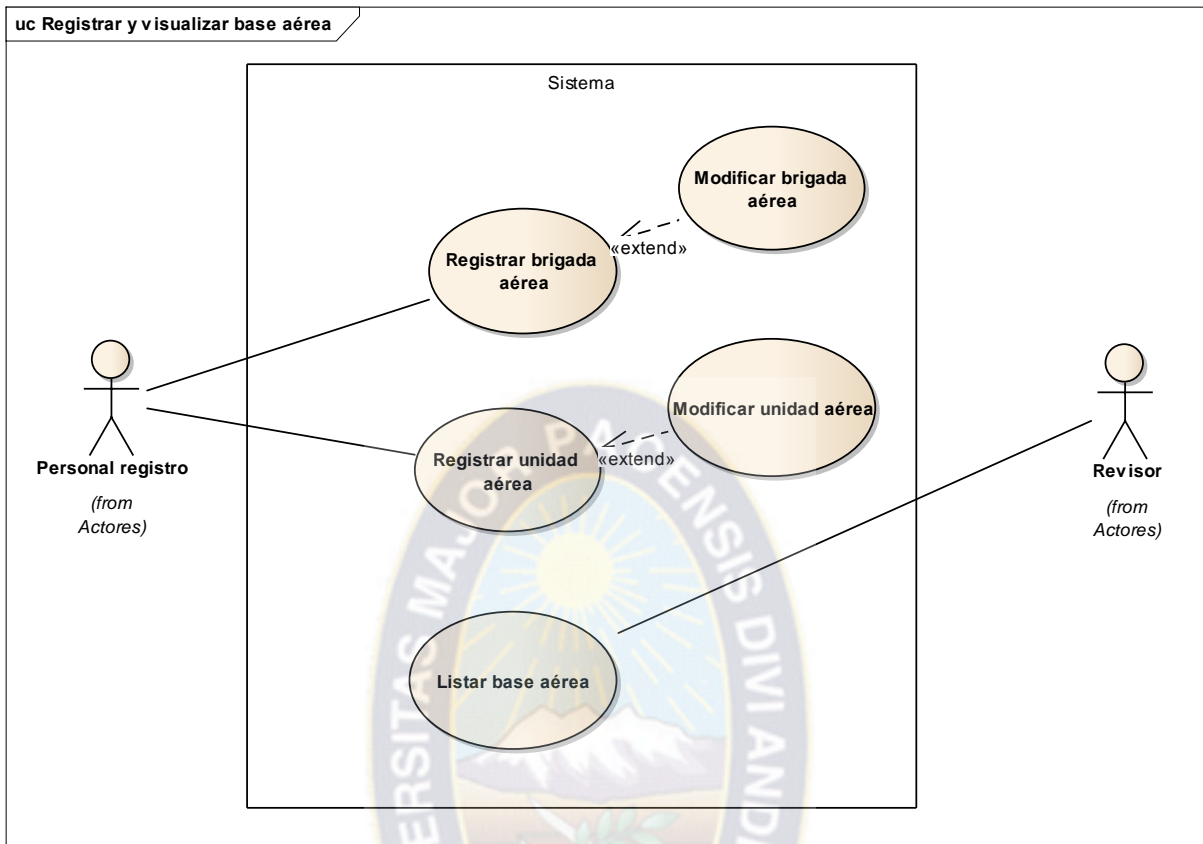


Figura 3.6 Caso de uso registrar y visualizar base aérea
Fuente. Elaboración propia

Registra y visualizar aeronave

A continuación se presenta el caso de uso correspondiente al registro y visualización de aeronave.

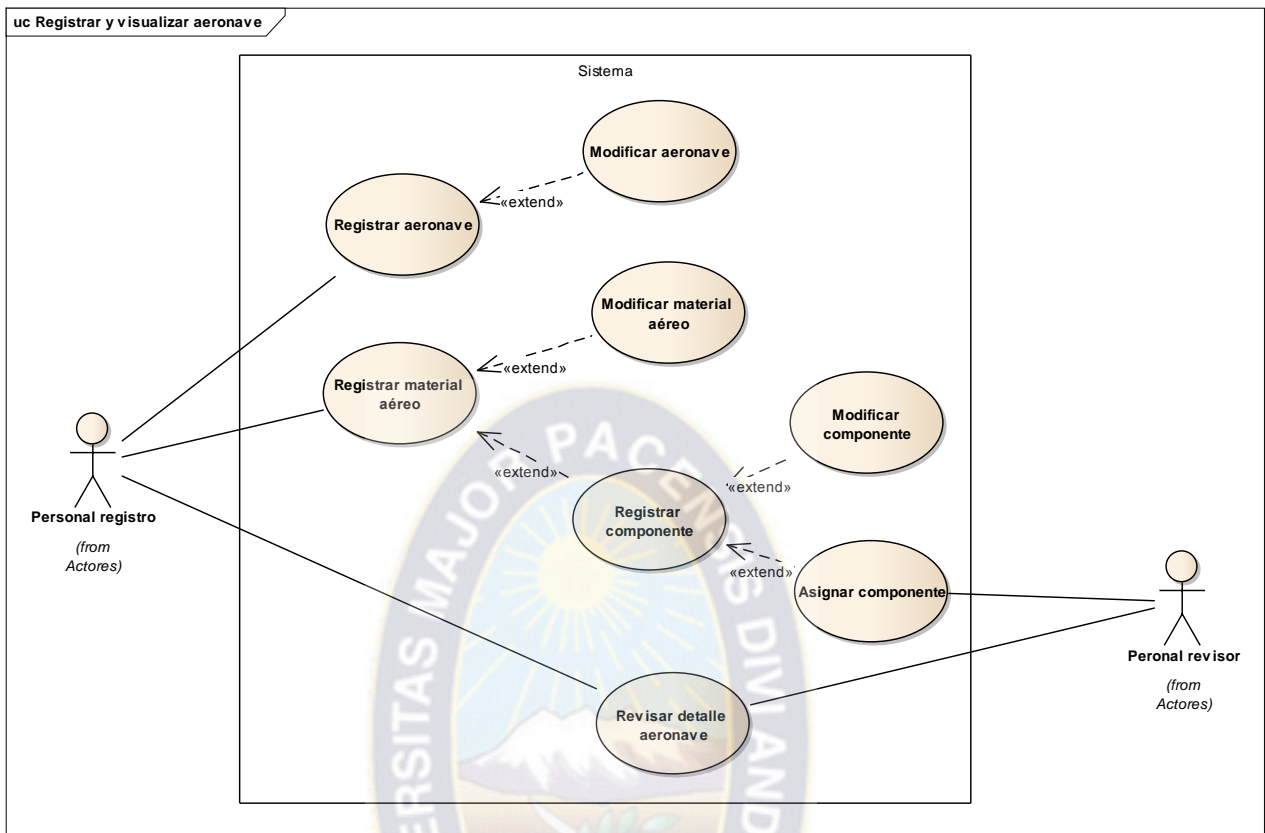


Figura 3.7 Caso de uso registrar y visualizar aeronave
Fuente. Elaboración propia

Registrar y visualizar horas o ciclos de vuelo

A continuación se presenta el caso de uso correspondiente al registro y visualización de horas o ciclos de vuelo.

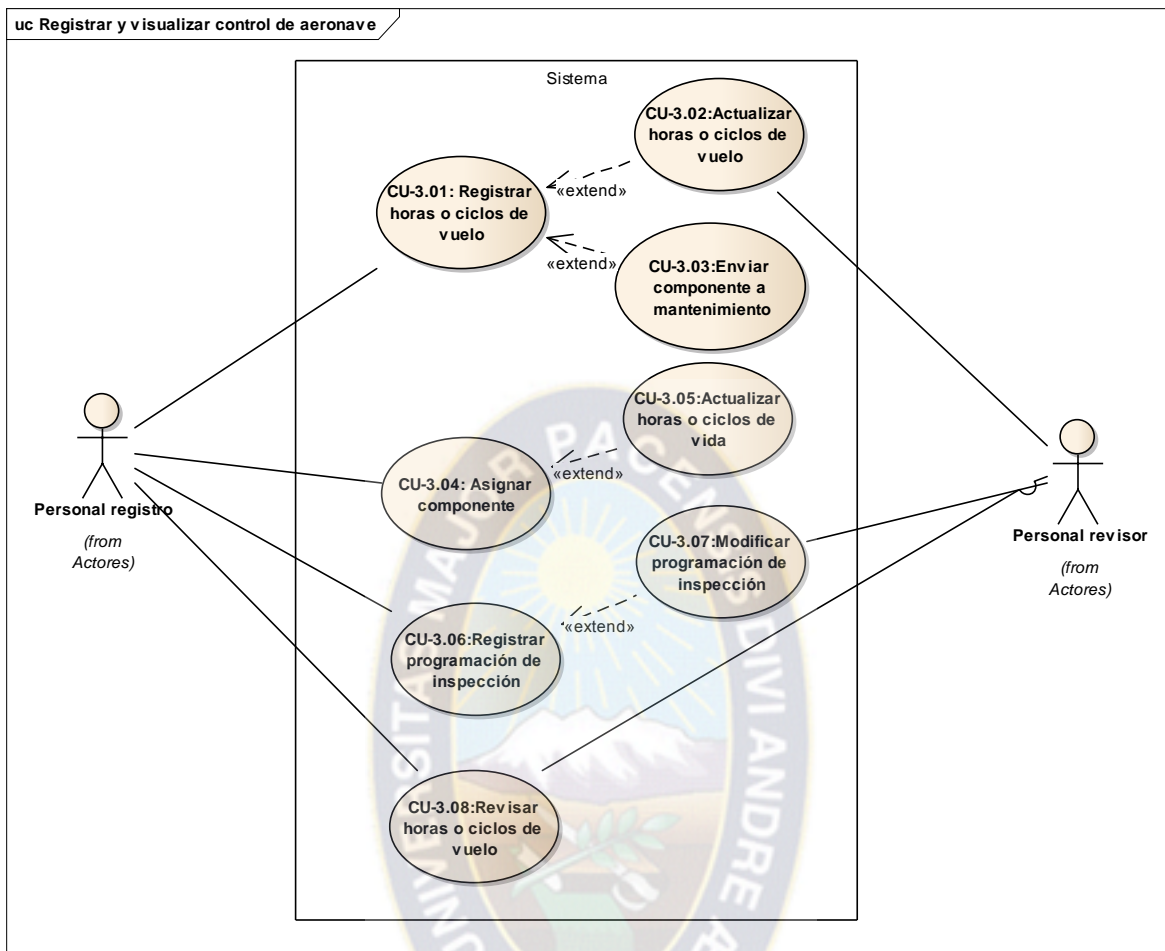


Figura 3.8 Caso de uso registrar y visualizar horas o ciclos de vuelo
Fuente. Elaboración propia

- **ESPECIFICACIÓN DE ACTORES**

La identificación de actores en términos generales son los usuarios los cuáles interactúan con el sistema, aportan y reciben información del sistema para coadyuvar a sus tareas cotidianas o necesidades demandadas.

A continuación se detallan la descripción de los actores identificados:

Actores	Definición
Personal de	Es el usuario responsable de registrar y visualizar la información

registro de material aéreo	general correspondiente a las aeronaves, y sus componentes; así como administrar el registro de las grandes y pequeñas unidades aéreas que son parte de la FAB.
Personal de mantenimiento	Es el usuario responsable del registro de las horas o ciclos de vuelo de las aeronaves y sus componentes, así como también de la programación y seguimiento de los mismos. Así mismo de visualizar los datos generados a partir de los registros.
Personal revisor	Es el usuario responsable del control y evaluación de los datos registrados en el sistema.

Tabla 3.15 Descripción de actores identificados
Fuente: Elaboración Propia

Especificación de casos de uso

A continuación se presentan las especificaciones del caso de uso “Registrar y visualizar control de la aeronave”, cuya prioridad es mayor a los otros, las especificaciones del resto de los casos de uso puede vérselas en anexos.

Especificación de casos de uso para el caso de uso Registrar y visualizar control de aeronave.

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-03.01

Nombre	Registrar horas o ciclos de vuelo		
Código	CU-03.01	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		

Precondición	El material aéreo debe estar previamente registrado
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un material aéreo y selecciona la opción “Ver horas/ciclos”. 2. El sistema despliega una interfaz que contiene el listado de los registros de horas y ciclos de vuelo registrados. 3. El usuario Personal de registro selecciona la opción Registrar horas/ciclos. 4. El sistema despliega el formulario de registro de horas/ciclos de vuelo. 5. El usuario Personal de registro introduce los datos requeridos, y selecciona la opción “Guardar”. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de horas y ciclos registrados.
Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	Los datos de horas y ciclos de vuelo se almacenan en la base de datos

Tabla 3.16 Especificación Registrar horas o ciclos de vuelo
Fuente: Elaboración Propia

A continuación se muestra la especificación correspondiente al caso de uso CU-03.02:

Nombre	Actualizar horas o ciclos de vuelo		
Código	CU-03.02	Estado (Fase)	Elaboración
Actor (es)	Personal revisor		
Precondición	El material aéreo debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un material aéreo y selecciona la opción “Ver horas/ciclos”. 2. El sistema despliega una interfaz que contiene el listado de los registros de horas y ciclos de vuelo registrados. 3. El usuario Personal revisor elige un registro de hora y/o ciclo y selecciona la opción modificar. 4. El sistema despliega el formulario de modificación de horas/ciclos de vuelo. 5. El usuario Personal revisor introduce los datos actualizados, y selecciona la opción “Guardar”. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de horas y ciclos registrados. 		
Escenarios alternativos			
Alternativa 1			
PostCondición	Los datos de horas y ciclos de vuelo se almacenan en la base de datos		

Tabla 3.17 Especificación Actualizar horas o ciclos de vuelo
Fuente: Elaboración Propia

A continuación se muestra la especificación correspondiente al caso de uso CU-03.03

Nombre	Asignar componente		
Código	CU-03.03	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El material aéreo debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro selecciona del menú principal “Control” el submenú “Almacén” 2. El Personal de registro elige un material aéreo y selecciona la opción “Ver horas/ciclos”. 3. El sistema despliega una interfaz que contiene el listado de los registros de horas y ciclos de vuelo registrados. 4. El usuario Personal revisor elige un registro de hora y/o ciclo y selecciona la opción modificar. 5. El sistema despliega el formulario de modificación de horas/ciclos de vuelo. 6. El usuario Personal revisor introduce los datos actualizados, y selecciona la opción “Guardar”. 7. El sistema valida los datos ingresados y los almacena en la base de datos. 8. El sistema despliega el listado de horas y ciclos registrados. 		
Escenarios alternativos			
Puntos de extensión			

PostCondición	Los datos de horas y ciclos de vuelo se almacenan en la base de datos
---------------	---

Tabla 3.18 Especificación Actualizar horas o ciclos de vuelo
Fuente: Elaboración Propia

A continuación se muestra la especificación correspondiente al caso de uso CU-03.04

Nombre	Asignar componente		
Código	CU-03.04	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El material aéreo debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro selecciona del menú principal “Control” el submenú “Almacén” 2. El Personal de registro elige un componente y selecciona la opción “Asignar a aeronave”. 3. El sistema despliega una interfaz que contiene el listado de aeronaves a asignar. 4. El usuario Personal de registro selecciona una aeronave. 5. El usuario Personal de registro selecciona la opción “Guardar”. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de horas y ciclos registrados. 		
Escenarios alternativos			
Alternativa 1			

Puntos de extensión	
PostCondición	El componente se asigna a una determinada aeronave.

Tabla 3.19 Especificación Actualizar horas o ciclos de vuelo
Fuente: Elaboración Propia

A continuación se muestra la especificación correspondiente al caso de uso CU-03.0

Nombre	Actualizar horas o ciclos de vida		
Código	CU-03.05	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El componente debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un componente y selecciona la opción “Actualizar horas y/o ciclos de vida”. 2. El sistema despliega un formulario de actualización de horas y/o ciclos de vida. 3. El usuario Personal de registro selecciona una aeronave, y selecciona la opción Actualizar horas y/o ciclos de vida. 4. El usuario Personal de registro ingresa las nuevas horas y/o ciclos de vida. 5. El usuario Personal de registro selecciona la opción Guardar. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de horas y ciclos registrados. 		

Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	El componente se asigna a una determinada aeronave.

Tabla 3.20 Especificación Actualizar horas o ciclos de vuelo
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-03.06

Nombre	Registrar programación de inspección		
Código	CU-03.06	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El material aéreo debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un material aéreo y selecciona la opción “Ver programación”. 2. El sistema despliega una interfaz que contiene el listado de la programación de inspección del material aéreo seleccionado. 3. El usuario Personal de registro selecciona la opción Registrar programación. 4. El sistema despliega el formulario de registro de programación de inspección. 5. El usuario Personal de registro introduce los datos requeridos, y 		

	<p>selecciona la opción “Guardar”.</p> <p>6. El sistema valida los datos ingresados y los almacena en la base de datos.</p> <p>7. El sistema despliega el listado de programación de inspección.</p>
Escenarios alternativos	
Alternativa 1	<p>5A. Cancelar</p> <p>1. El Usuario Personal de registro selecciona la opción Cancelar</p> <p>2. El sistema sale de la interfaz sin almacenar los datos</p>
Puntos de extensión	
PostCondición	Los datos de la programación se almacenan en la base de datos

Tabla 3.21 Especificación Registrar programación de inspección
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-03.07

Nombre	Modificar programación de inspección		
Código	CU-03.07	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El Personal de registro debe estar autenticado en el sistema.		
Escenario básico	1. El Personal de registro elige un material aéreo y selecciona la opción “Ver programación”.		

	<ol style="list-style-type: none"> 2. El sistema despliega una interfaz que contiene el listado de la programación de inspección del material aéreo seleccionado. 3. El usuario Personal de registro selecciona elije un registro y selecciona la opción Modificar. 4. El sistema despliega el formulario de registro de programación de inspección. 5. El usuario Personal de registro introduce los datos actualizados, y selecciona la opción “Guardar”. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de programación de inspección.
Escenarios alternativos	
Alternativa 1	<ol style="list-style-type: none"> 3A. Cancelar 3. El Usuario Personal de registro selecciona la opción Cancelar 4. El sistema sale de la interfaz sin almacenar los datos
Puntos de extensión	
PostCondición	Los datos actualizados se almacenan en la base de datos

Tabla 3.22 Especificación Modificar programación de inspección

Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-03.08

Nombre	Revisar horas o ciclos de vuelo
--------	---------------------------------

Código	CU-03.08	Estado (Fase)	Elaboración
Actor (es)	Personal revisor		
Precondición	El Personal de registro debe estar autenticado en el sistema. Debe existir al menos un material aéreo registrado.		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal revisor elige un material aéreo y selecciona la opción “Ver horas o ciclos de vuelo”. 2. El sistema despliega una interfaz que contiene la información general del material aéreo, los componentes registrados correspondientes al material aéreo y el registro de horas o ciclos de vuelo del mismo. 3. El usuario Personal revisor selecciona la opción Descargar Excel. 4. El sistema genera un documento Excel con la información presentada en el punto 2. 		
Escenarios alternativos			
Alternativa 1	3A. Descargar PDF <ol style="list-style-type: none"> 1. El Usuario Personal revisor selecciona la opción Descargar PDF 2. El sistema genera un documento PDF con la información presentada en el punto 2 		
Puntos de extensión			
PostCondición	Se genera un reporte con la información del material aéreo.		

Tabla 3.23 Especificación Registrar horas o ciclos de vuelo
Fuente: Elaboración Propia

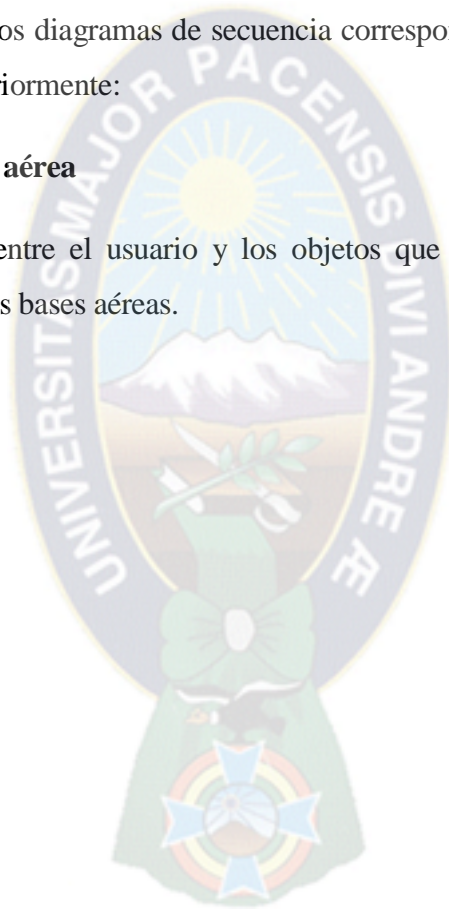
ii. DIAGRAMAS DE SECUENCIA

El diagrama de secuencia muestra la forma en la que se comunican los objetos al transcurrir el tiempo en el orden de las llamadas/eventos del sistema. El evento del sistema es una entrada externa que origina una operación del sistema como respuesta al evento, representados en secuencias, el detalle del diagrama de secuencia depende de la fase en la que se encuentre, lo que se pretenda contar con el diagrama y a quién esté dirigido.

A continuación se detallan los diagramas de secuencia correspondientes al sistema, según los casos de uso detallados anteriormente:

Registrar y visualizar base aérea

Describe la comunicación entre el usuario y los objetos que participan en el proceso del registro y visualización de las bases aéreas.



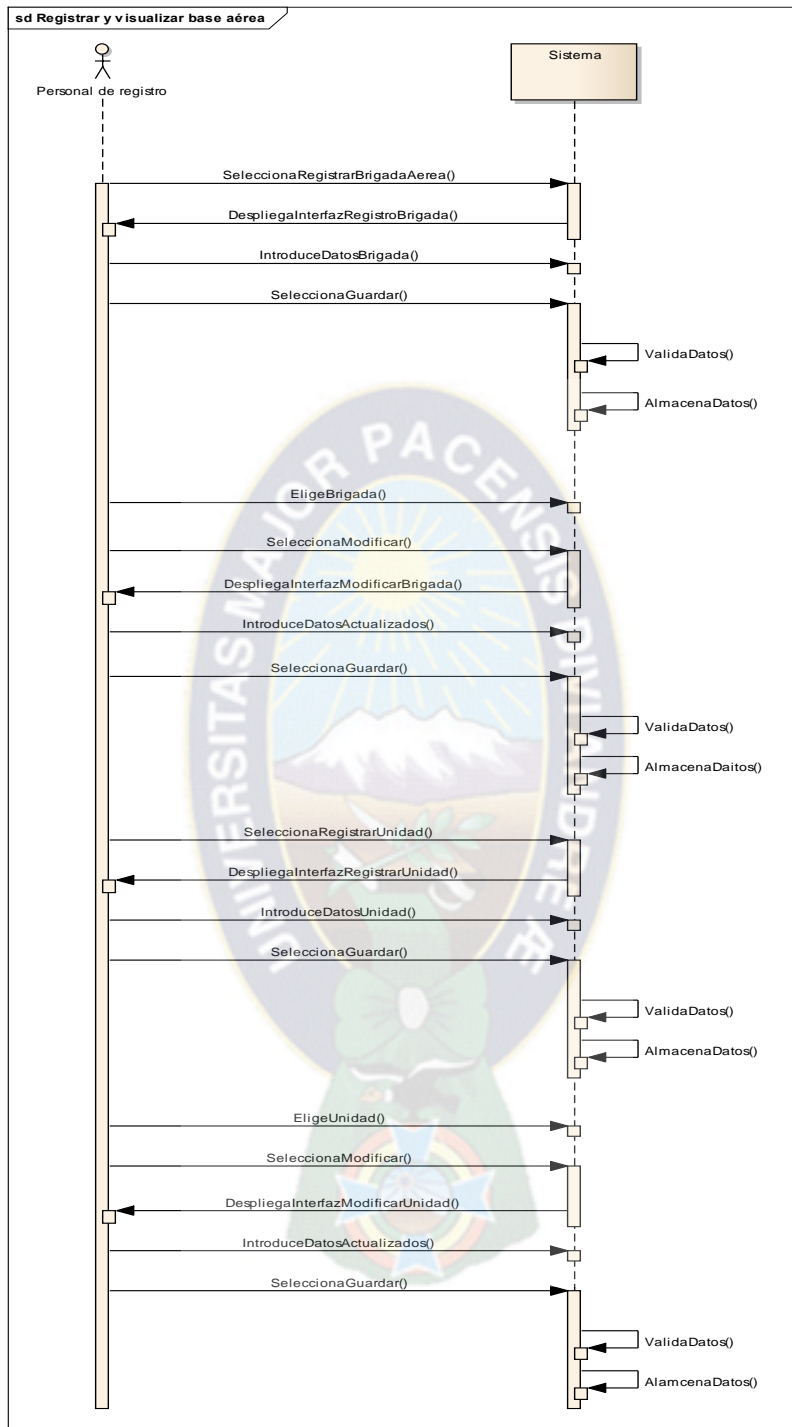


Figura 3.9 Diagrama de secuencia: Registrar y visualizar base aérea
Fuente. Elaboración propia

Registrar y visualizar aeronave

Describe la comunicación entre el usuario y los objetos que participan en el proceso del registro y visualización de las bases aéreas.

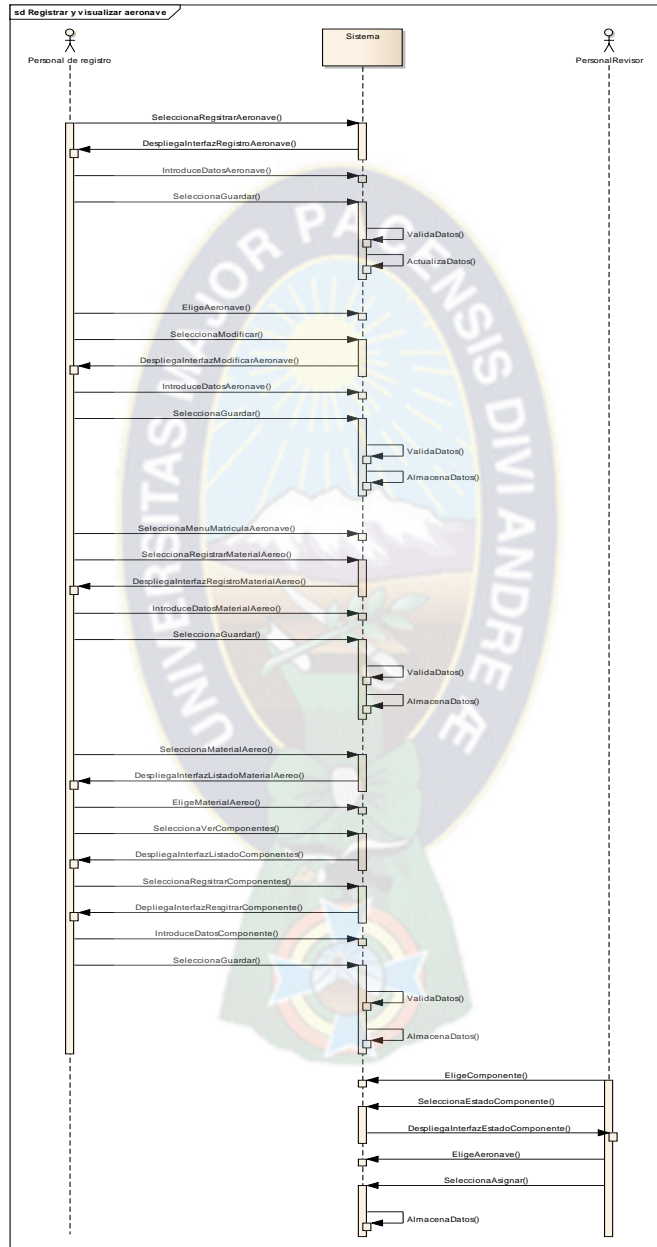


Figura 3.10 Diagrama de secuencia: Registrar y visualizar aeronave
Fuente. Elaboración propia

Registrar y visualizar control de aeronave

Describe la comunicación entre el usuario y los objetos que participan en el proceso del registro y visualización del control de las aeronaves.

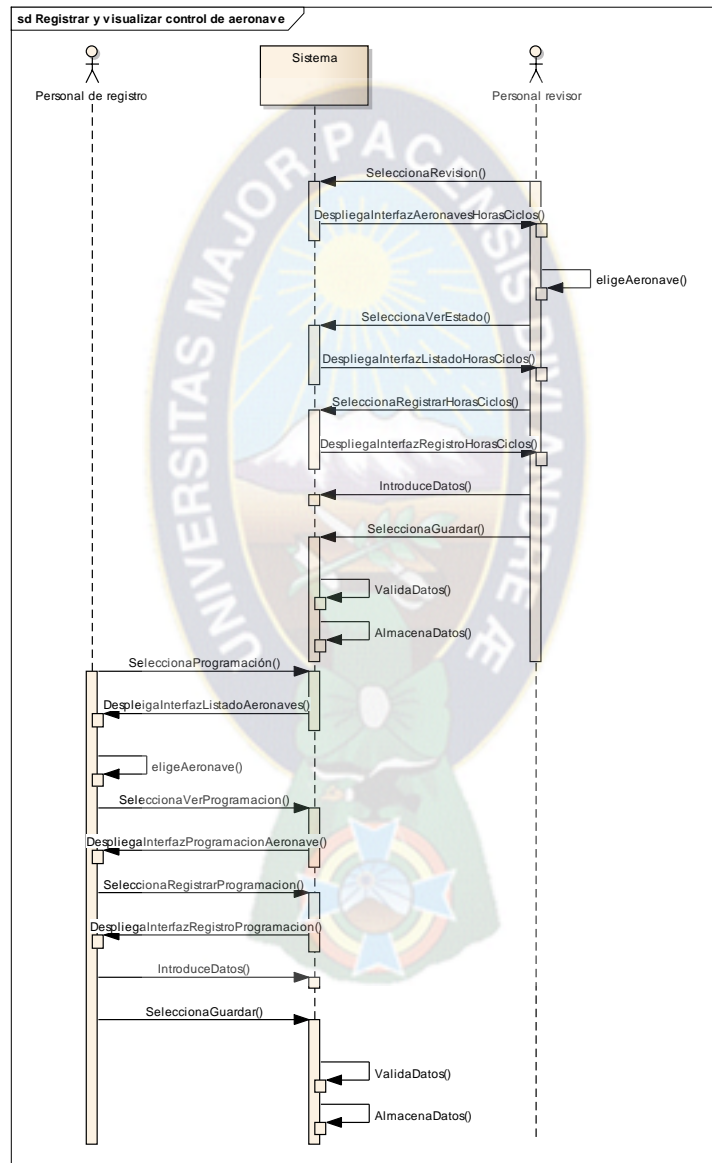


Figura 3.11 Diagrama de secuencia: Registrar y visualizar control aeronave
Fuente. Elaboración propia

iii. MODELO DE CONTENIDO

A continuación se presenta el diagrama de clases para el modelo de contenido del sistema.

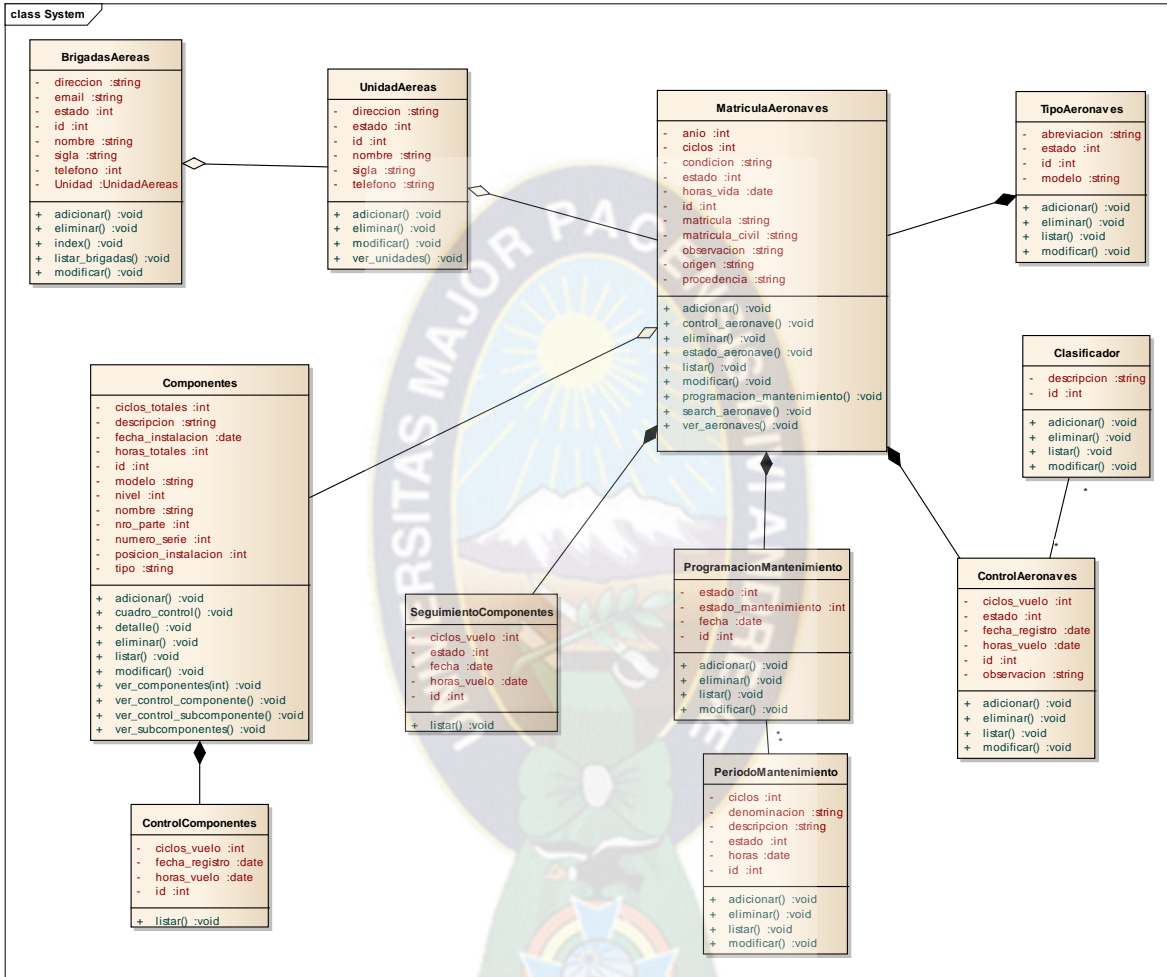


Figura 3.12 Modelo de contenido
Fuente. Elaboración propia

iv. MODELO DEL DOMINIO

El modelo de dominio muestra los conceptos básicos del dominio del problema, sus propiedades importantes y las relaciones de dichos conceptos.

A continuación la figura muestra el modelo del dominio correspondiente al sistema.

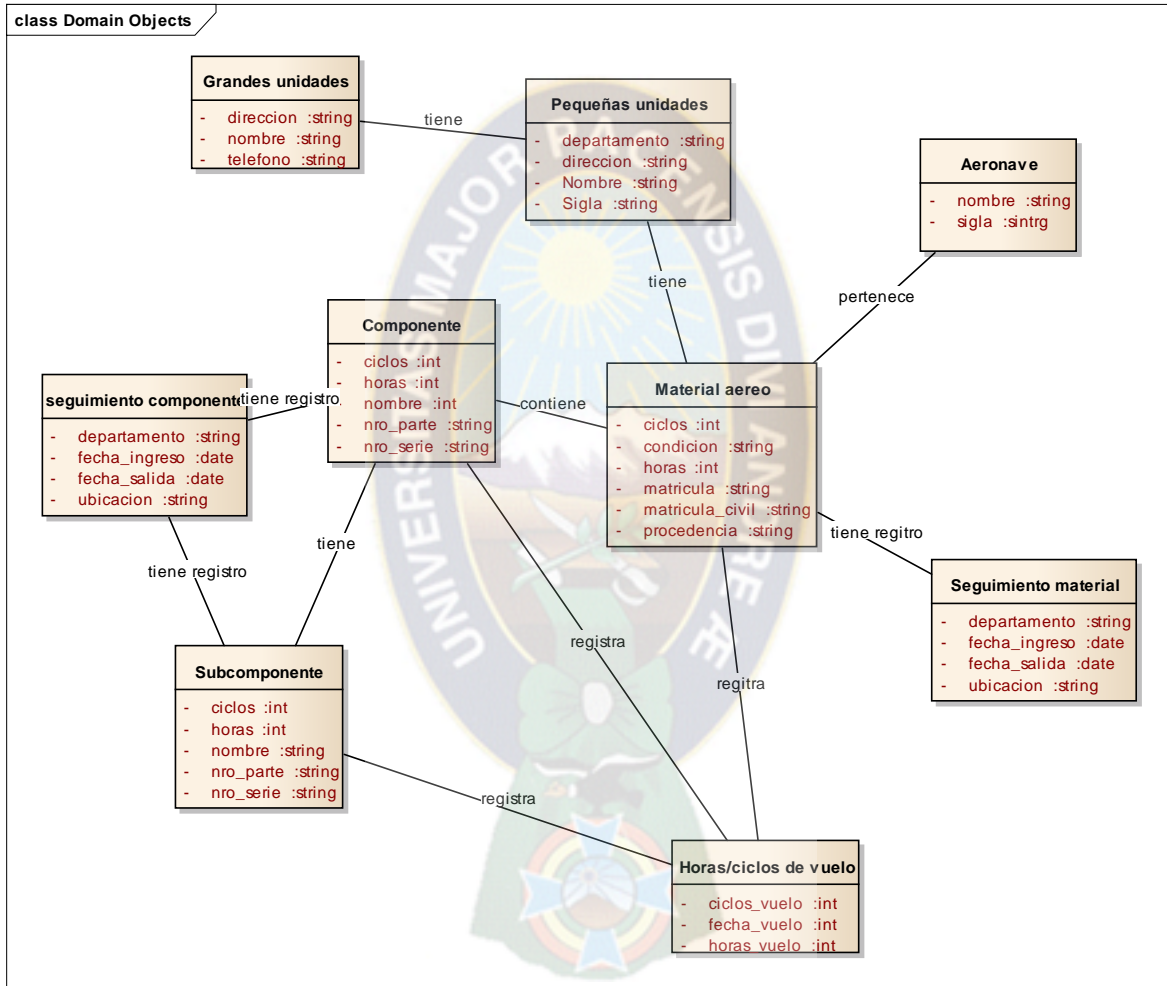


Figura 3.13 Modelo del dominio
Fuente. Elaboración propia

3.2.3.3.DISEÑO

i. MODELO NAVEGACIONAL

A continuación se presentan los modelos navegacionales correspondientes a los distintos casos de uso, citados anteriormente:

Modelo Navegacional para el caso de uso Registrar y visualizar base aérea:

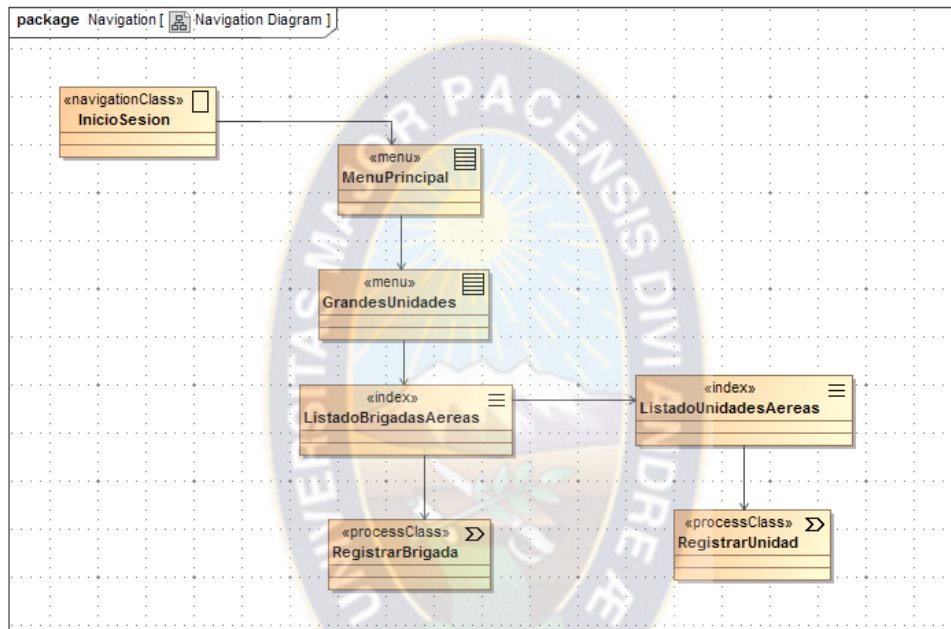


Figura 3.14 . Modelo navegacional Registrar y visualizar base aérea
Fuente. Elaboración propia

Modelo Navegacional correspondiente al caso de uso Registrar y visualizar aeronave:

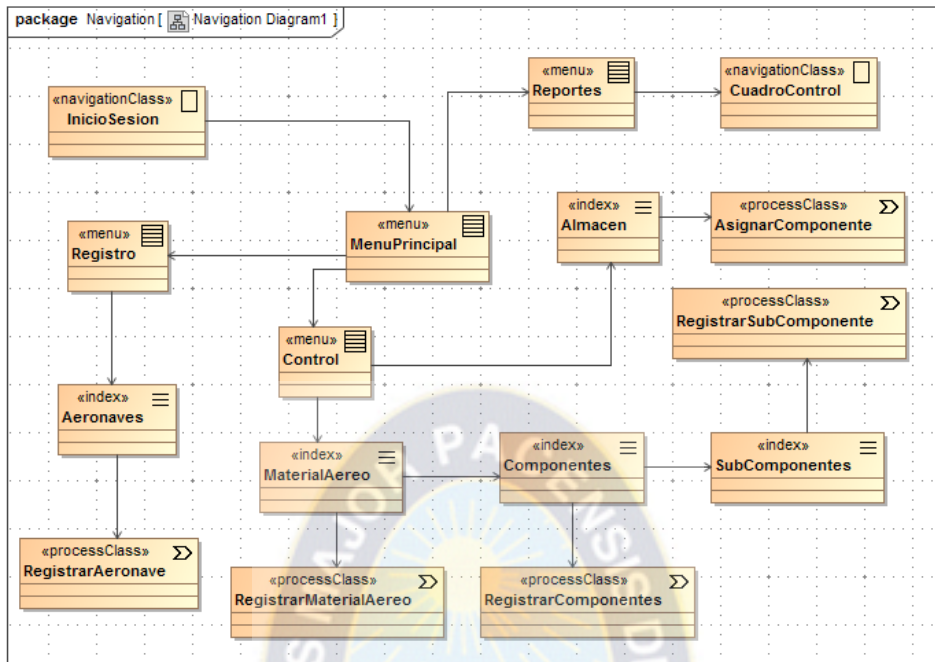


Figura 3.15 Modelo navegacional Registrar y visualizar aeronave
Fuente. Elaboración propia

Modelo Navegacional correspondiente al caso de uso Registrar y visualizar horas o ciclos de vuelo

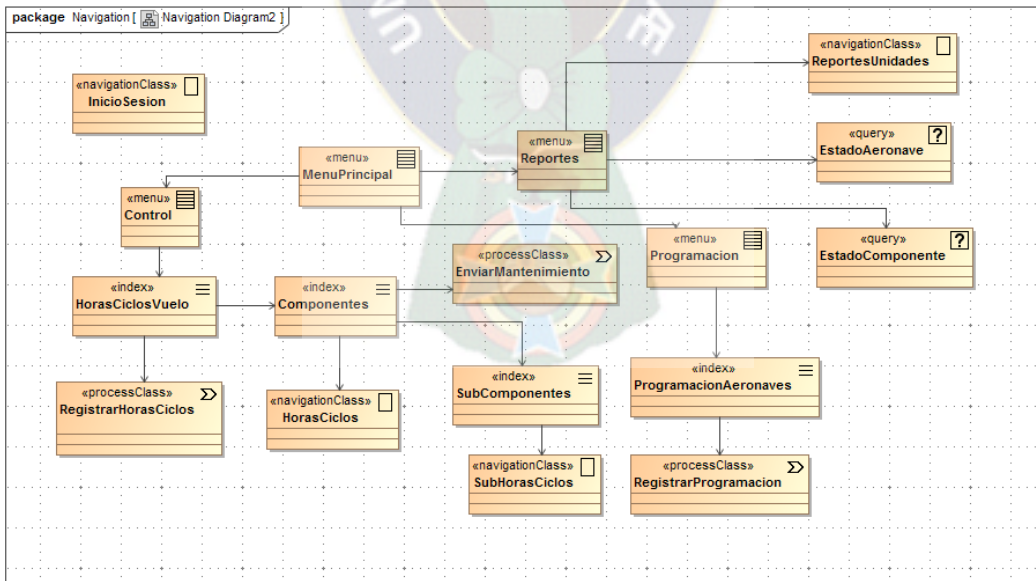


Figura 3.16 Modelo navegacional Registrar y visualizar horas o ciclos de vuelo
Fuente. Elaboración propia

ii. MODELO DE PRESENTACIÓN

A continuación se detallan los modelos de presentación correspondientes al sistema.

Se presenta el modelo de presentación relacionado al inicio de sesión:

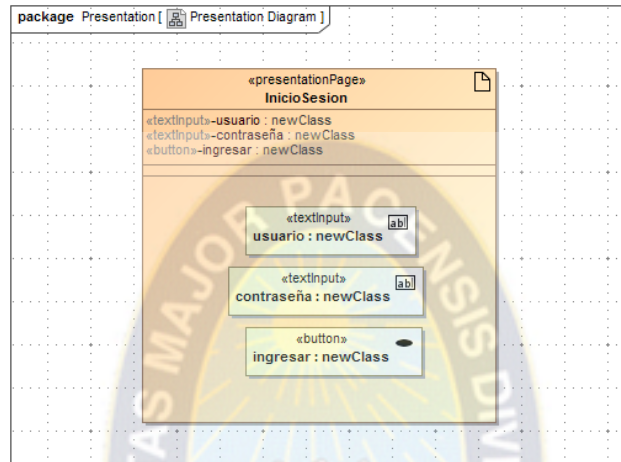


Figura 3.17 Modelo de presentación Inicio de sesión
Fuente. Elaboración propia

A continuación se presenta el modelo de presentación para el panel principal

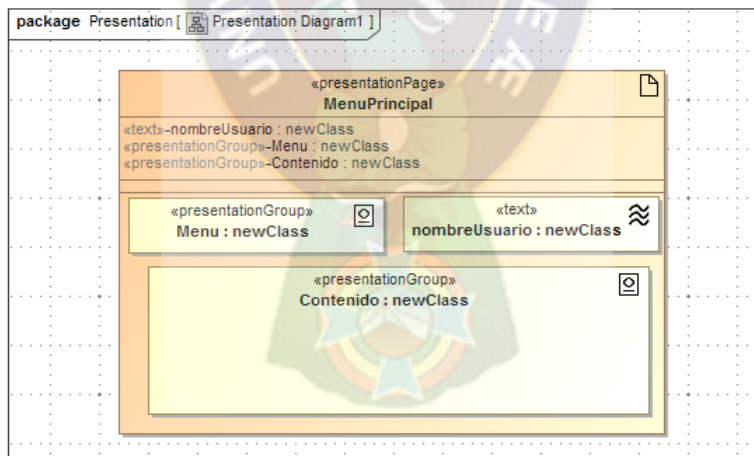


Figura 3.18 Modelo de presentación Panel principal
Fuente. Elaboración propia

La siguiente figura muestra el modelo presentación para el registro y control de unidades aéreas

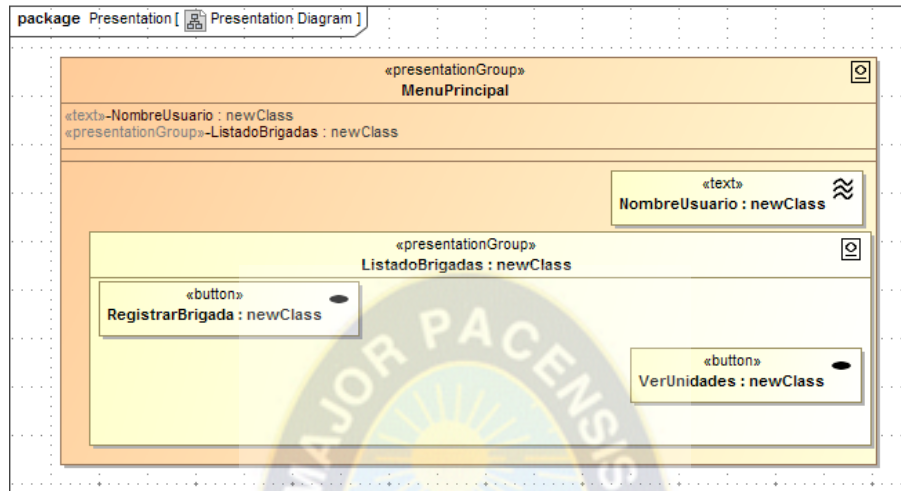


Figura 3.19 Modelo de presentación Panel principal
Fuente. Elaboración propia

La siguiente figura muestra el modelo presentación para el registro y visualización de aeronaves.

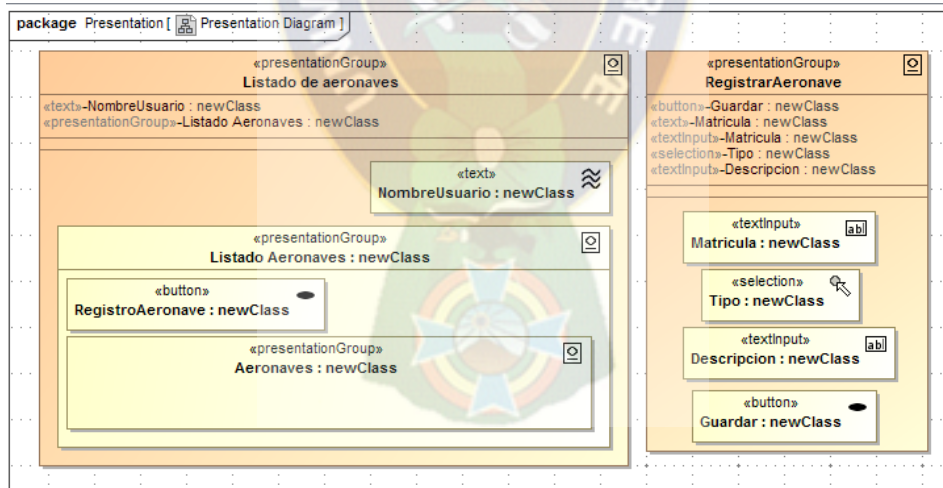


Figura 3.20 Modelo de presentación Registro y visualización de aeronaves
Fuente. Elaboración propia

La siguiente figura muestra el modelo presentación para el registro y control de horas o ciclos de vuelo.

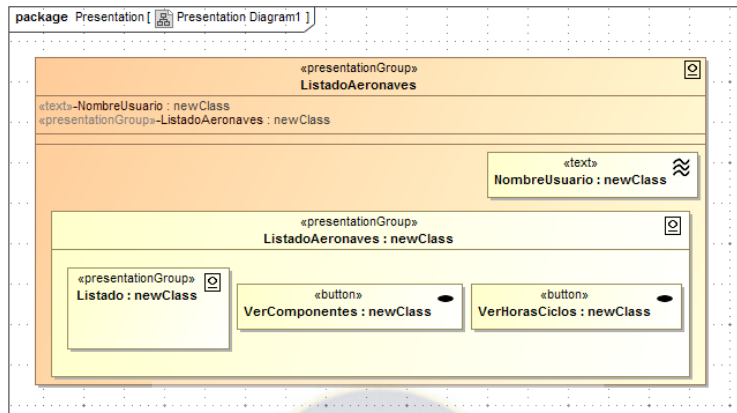


Figura 3.21 Modelo de presentación Panel principal
Fuente. Elaboración propia

La siguiente figura muestra el modelo presentación para el registro de programación de mantenimiento.

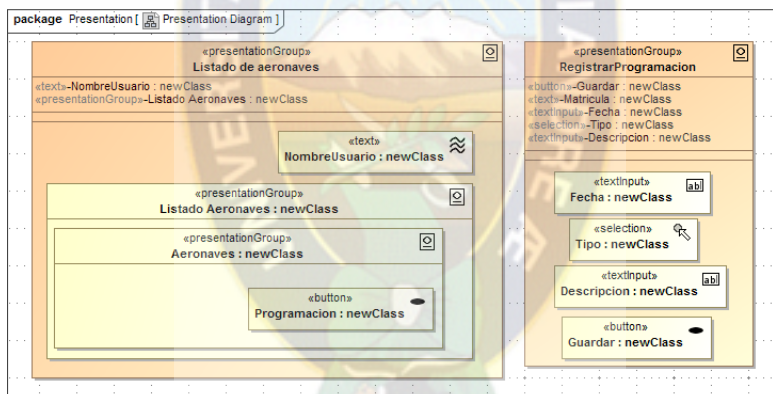


Figura 3.22 Modelo de presentación Registro de programación de manteniendo
Fuente. Elaboración propia

iii. MODELO ENTIDAD RELACIÓN

A partir de las especificaciones que se tienen para transformar un diagrama de clases a un modelo entidad relación, a continuación se presenta de manera gráfica en la figura el modelo entidad relación correspondiente al sistema.

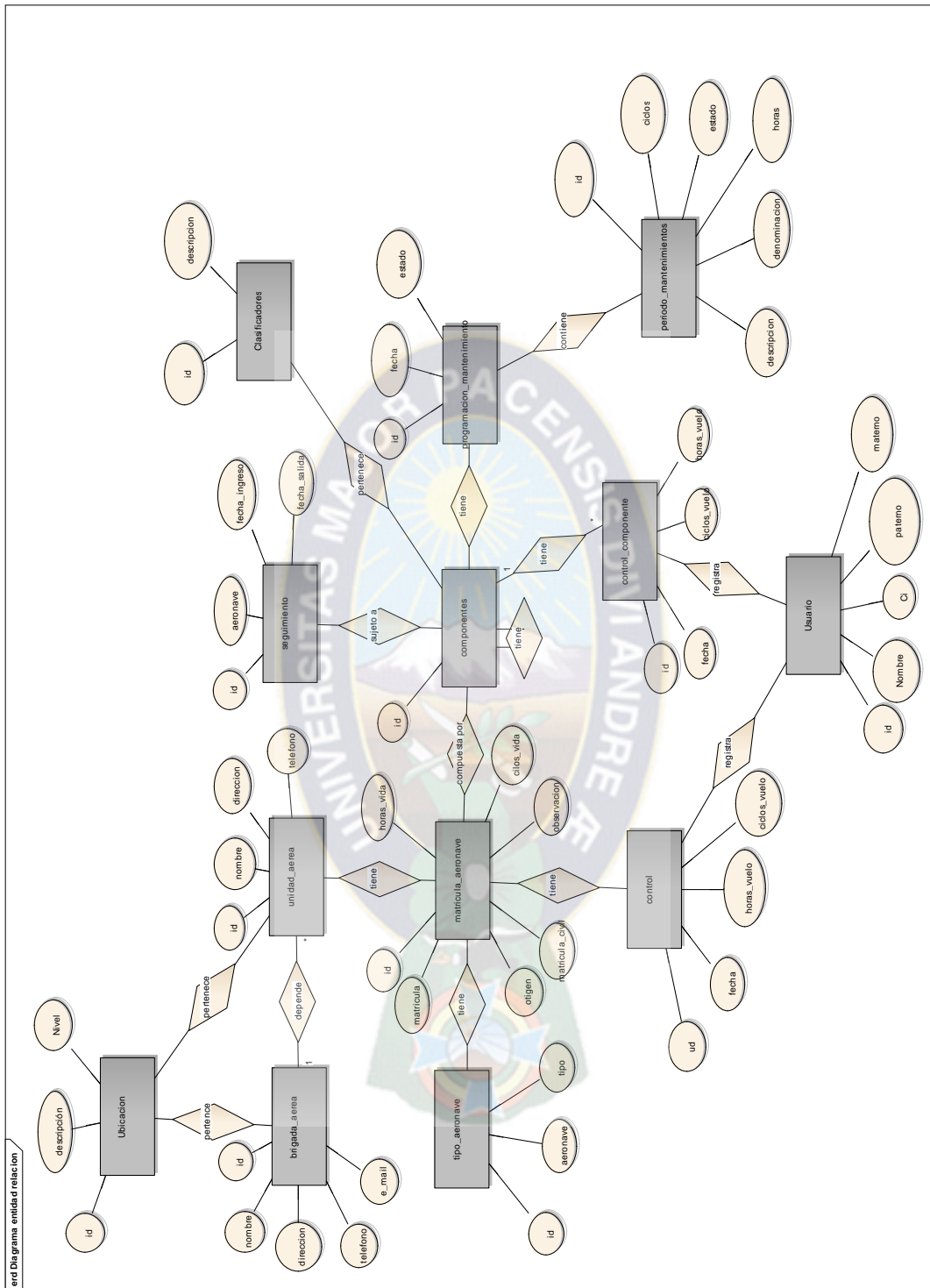


Figura 3.23 Diagrama Entidad-Relación
Fuente. Elaboración propia

iv. DISEÑO DE INTERFACES

A continuación se describen las interfaces de interacción del sistema con el usuario. Mediante las interfaces el sistema capturará datos y brindará información a los usuarios del sistema.

También se especifica la información de entrada y salida del sistema que se constituyen en los insumos del procesamiento del sistema.

Login

La figura describe la interfaz de acceso al sistema.

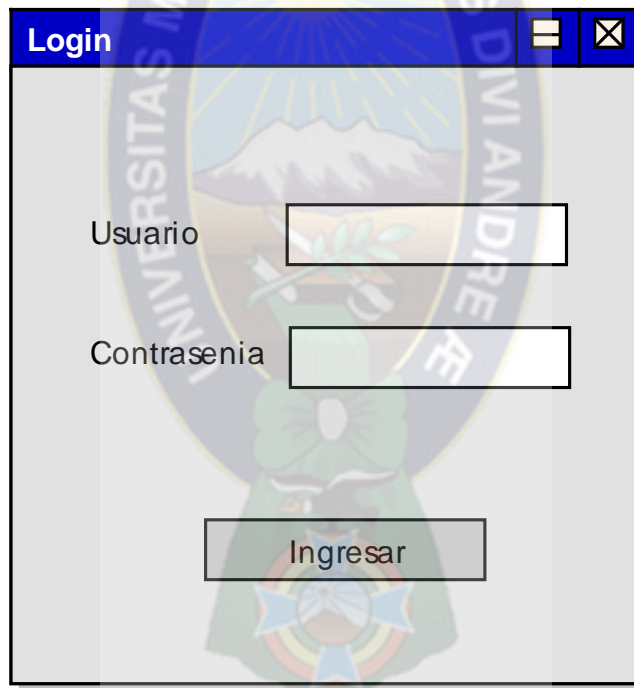
The image shows a screenshot of a web-based login interface. At the top, there is a blue header bar with the word "Login" in white text on the left and standard window control icons (minimize, maximize, close) on the right. Below the header, the main content area has a light gray background. It features two input fields: the first is labeled "Usuario" and the second is labeled "Contraseña". Both fields are empty and have a white background with a thin black border. Below these fields is a rectangular button with a gray background and the text "Ingresar" in black. In the background, there is a large, semi-transparent watermark of the logo of Universidad Mayor Pacensis Divi Andrés, which includes a sun, mountains, and a figure.

Figura 3.24 Login
Fuente: Elaboración Propia

Interfaz de inicio del sistema

La figura describe el panel de presentación de datos, que es la pantalla inicial de acceso al sistema.

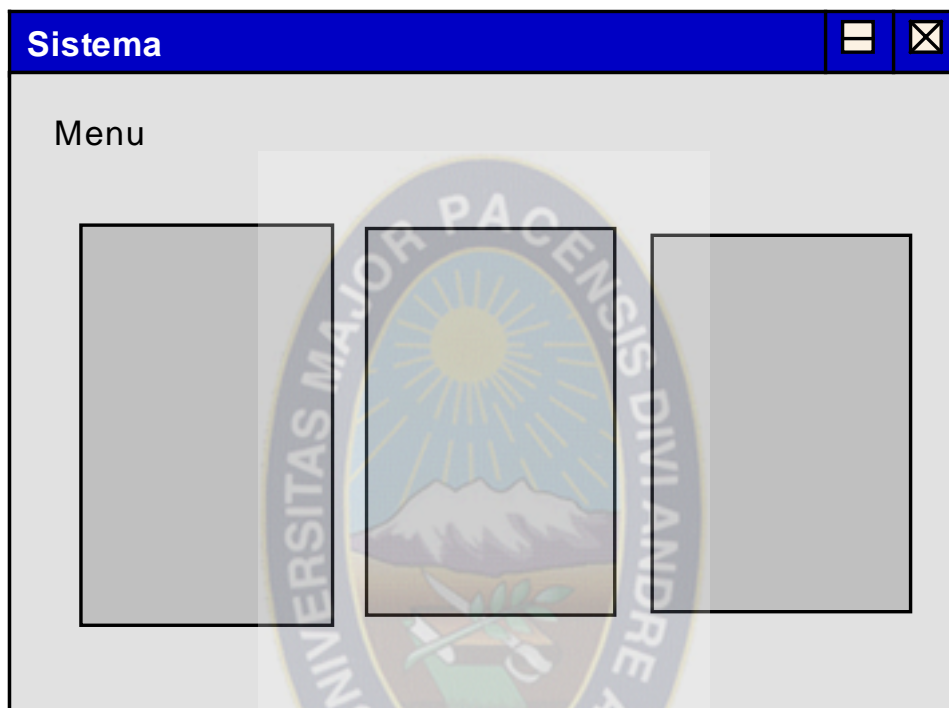


Figura 3.25 Menú principal del sistema
Fuente: Elaboración Propia

A continuación se detallan el prototipo de las interfaces correspondientes a cada tarea a realizarse por parte de los usuarios del sistema.

Registrar y visualizar base aérea

La figura muestra las interfaces relacionadas al registro de brigadas y unidades aéreas (Bases aéreas), y la interfaz de salida de las mismas.

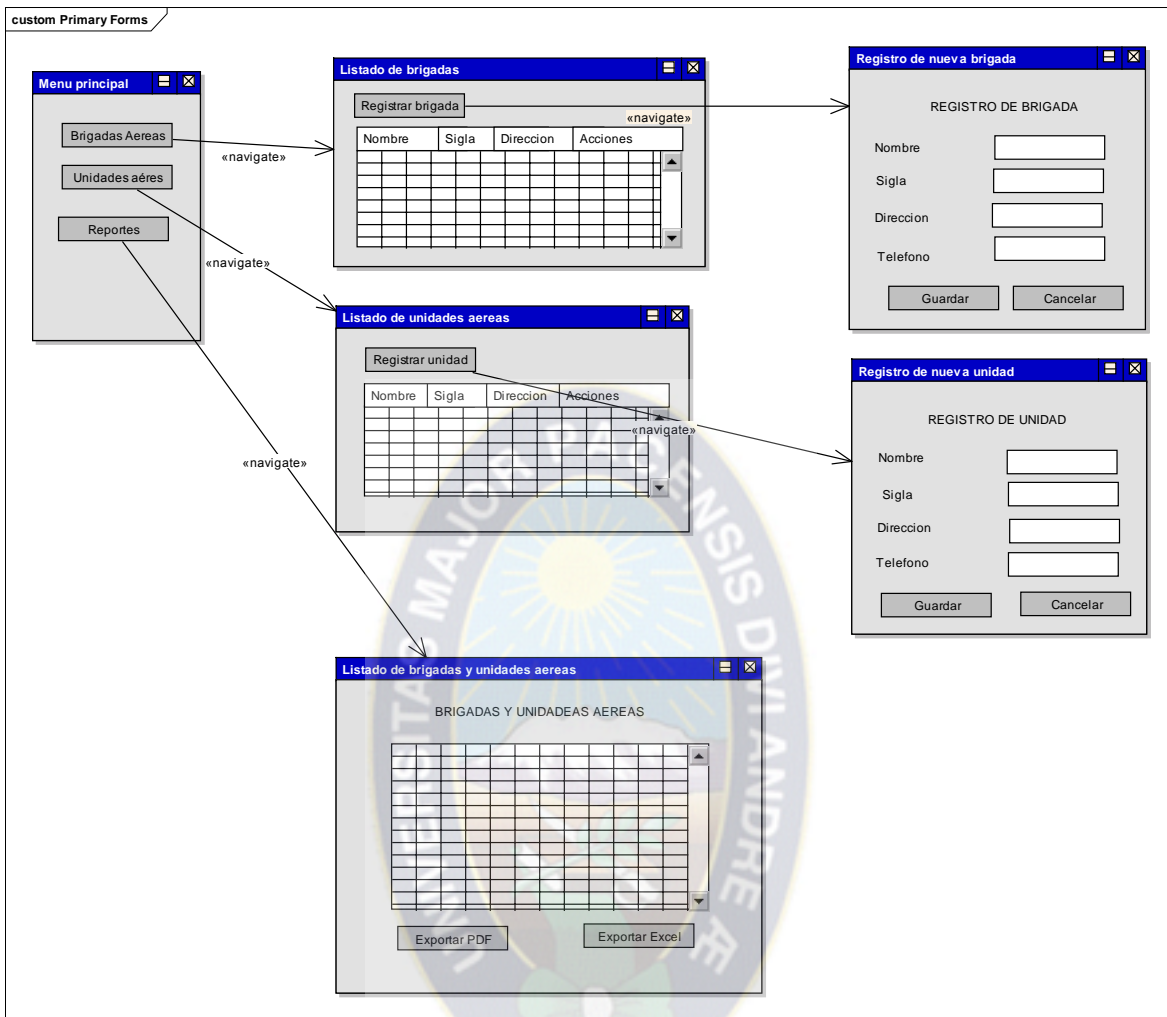


Figura 3.26 Interfaces Registrar y visualizar base aérea
Fuente: Elaboración Propia

Registrar y visualizar aeronaves

La figura muestra las interfaces relacionadas con el registro de aeronaves, material aéreo así como el listado de los mismos y el registro de sus componentes.

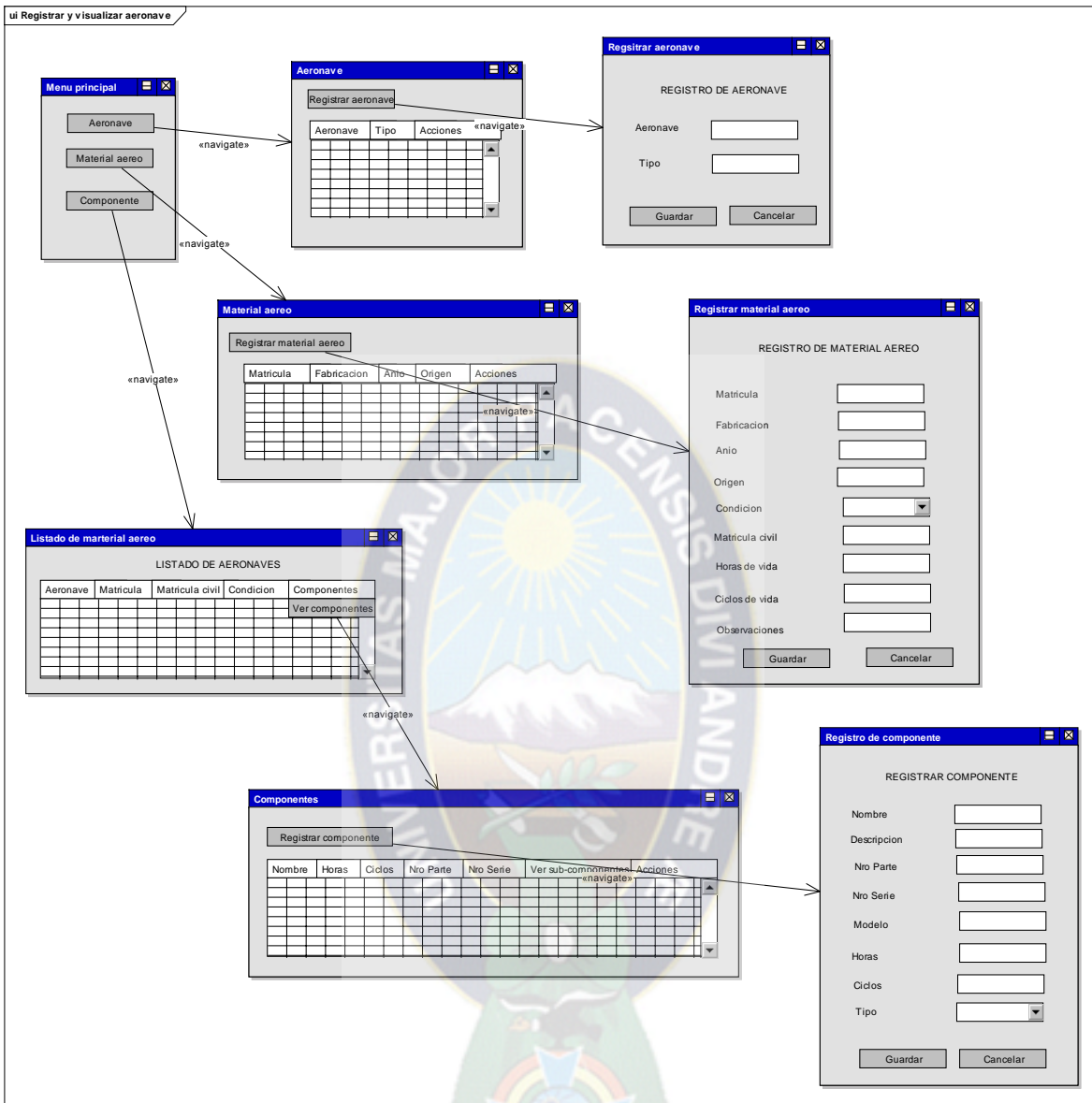


Figura 3.27 Interfaces Registrar y visualizar aeronaves
Fuente: Elaboración Propia

Registrar y visualizar control de aeronaves

La siguiente figura muestra las interfaces correspondientes al registro de las horas o ciclos de vuelo por aeronave, así como también el registro de la programación de mantenimiento y el listado de aeronaves según programación y horas y/o ciclos de vuelo.

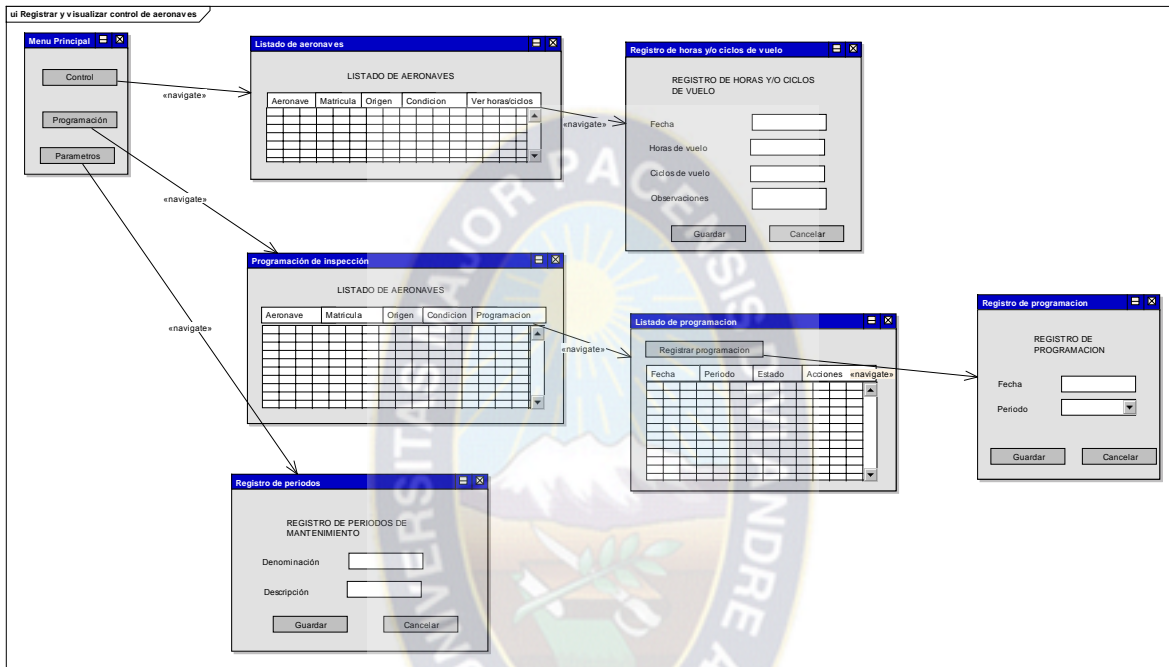


Figura 3.28 Interfaces Registrar y visualizar control aeronaves

Fuente: Elaboración Propia

3.2.3.4.ARQUITECTURA

A continuación se detalla la arquitectura del sistema, en la cual se puede evidenciar tres niveles de usuarios, los cuales están asociados a diferentes privilegios de acceso al sistema.

Al mismo tiempo, y apoyando los requerimientos de los usuarios, se puede evidenciar la facilidad de acceso del usuario al sistema, por medio de diferentes dispositivos (móviles y de escritorio), con el objetivo de que el entorno de trabajo de los usuarios sea lo más accesible.

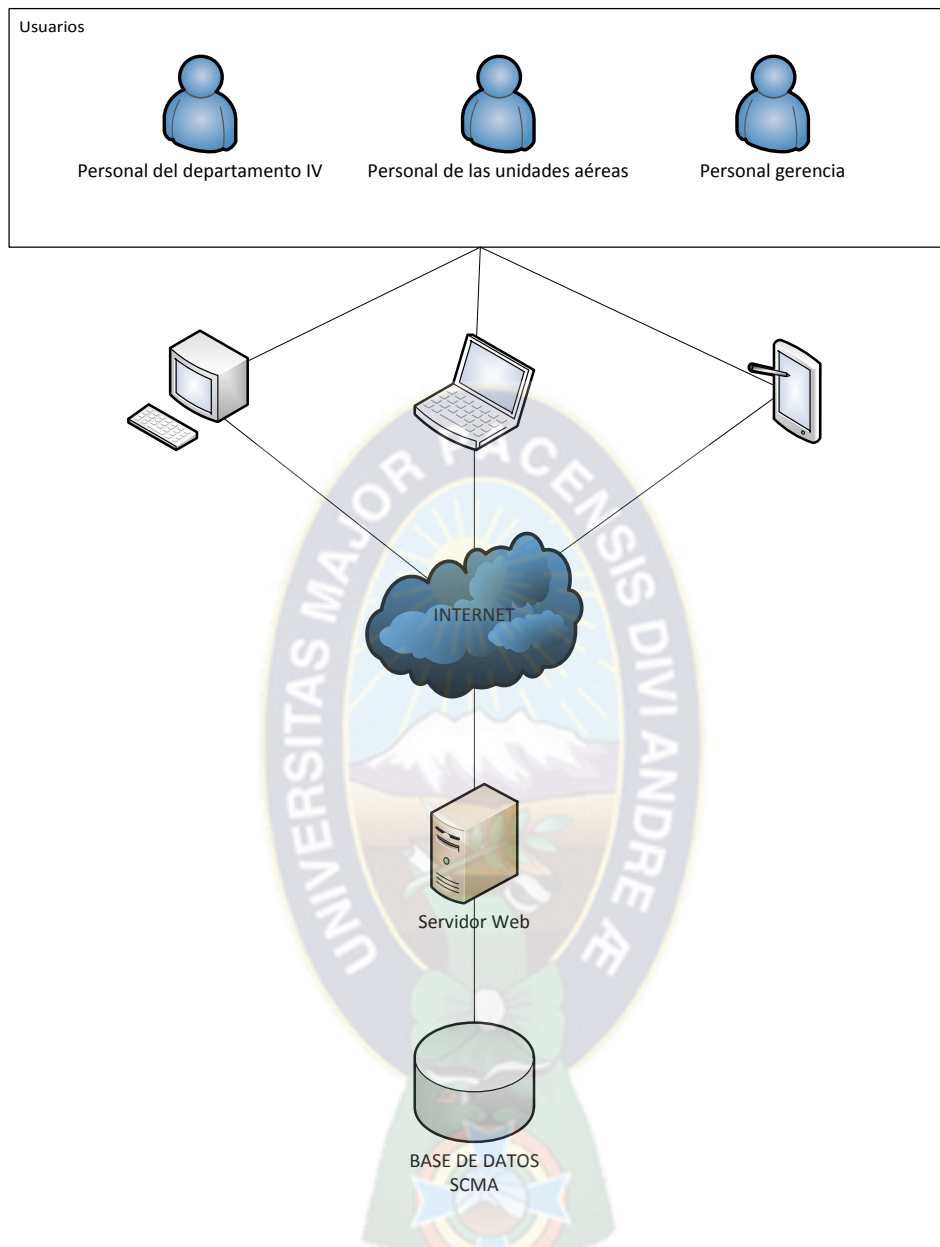


Figura 3.29 Arquitectura del sistema
Fuente: Elaboración Propia

3.2.4. FASE DE CONSTRUCCIÓN

3.2.5. JUSTIFICACIÓN DE LAS HERRAMIENTAS ELEGIDAS

- **Selección de herramientas de construcción**

A continuación se detallan las herramientas de construcción del software.

HERRAMIENTA	UTILIDAD
PHP	Lenguaje de programación
Apache	Servidor Web
Postgresql	Base de datos
PgAdmin III	Modelador de base de datos
MagicDraw (Plugin MagicUWE)	Diseñador UWE
CakePHP	Framework
Metronic	Plantilla de estilos

Tabla 3.24 Herramientas de construcción

Fuente: Elaboración propia

- **Justificación de la selección de herramientas**

PHP

En el sistema se utiliza software libre, por ser un sistema destinado a una entidad pública, tomando en cuenta la dirección que toma el estado en la actualidad, es uno de los lenguajes más utilizados en ambientes Web.

Específicamente para este proyecto cumple el objetivo de ser software libre, posee suficiente documentación, lo cual hace que el aprendizaje sea mucho más rápido, se adapta plenamente al entorno de desarrollo, además sigue la línea de sistemas que se están implementando y se implementarán en la FAB.

APACHE

Para este proyecto en específico, cumple el objetivo de ser software libre, posee amplia compatibilidad con navegadores actuales, esta aplicado en ambientes de la FAB, lo que facilita la puesta en producción del sistema.

POSTGRESQL

Para este proyecto en específico, cumple el objetivo de ser software libre, es uno de los motores de base de datos más completos en su línea, posee herramientas de seguridad y respaldos incorporadas, muy buena compatibilidad para proyectos en entornos Web.

PGADMIN III

Es una aplicación gráfica para gestionar el gestor de base de datos PostgreSQL, siendo la más completa y popular en la actualidad.

Para este proyecto en particular, cumple el objetivo de ser software libre, cuenta con una agradable y eficiente interfaz gráfica.

MAGICUWE

Es una herramienta CASE, desarrollada para el diseño asistido por ordenador de las aplicaciones Web, utilizando la metodología de ingeniería Web basado en UML (UWE), MagicUWE es un plugin de MagicDraw.

Facilita el realizado de los diagramas correspondientes UWE, ya que es una herramientas completa y de fácil uso, provee de todos los diagramas especificados por UWE.

3.2.6. PANTALLAS DEL SISTEMA

A continuación se muestran las principales interfaces del sistema.

El sistema de control y seguimiento de material aéreo presenta la siguiente interfaz de inicio de sesión, como interfaz inicial.

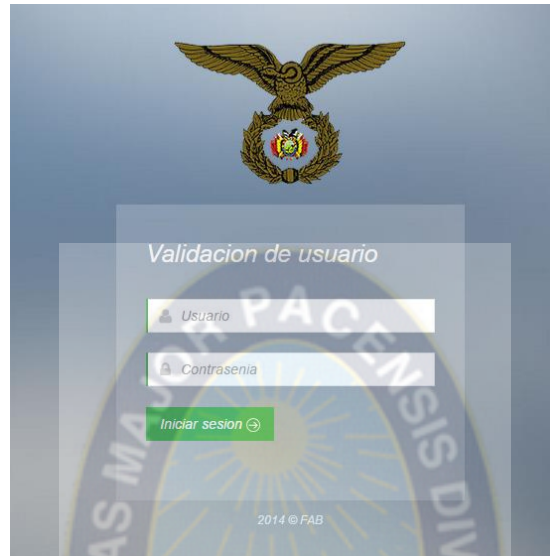


Figura 3.30 Interfaz Login

Fuente: Elaboración propia

En la siguiente figura se muestra el panel de presentación información, panel inicial. en esta pantalla se puede identificar el menú principal asignado al usuario y el nombre de usuario en la parte superior derecha.



Indicadores Configuración admin

Control y seguimiento de material aéreo Pagina Principal

Fuerza Aérea Boliviana Brigadas Aereas

Fuerza Aérea Boliviana Unidades Aereas

Estado actual

Nro	Aeronave	Tipo	Matricula civil	Matricula	Condición
1	Foocker	F-27		FAB-93	W
2	Foocker	F-27		FAB-92	W
3	Foocker	F-27		FAB-90	W
4	Foocker	F-27	ACD-665	FAB-101	P
5	BAE	B-146-200		FAB-103	W
6	MA-60	B-858L		FAB-96	W
7	SUPER KING	B-200C		FAB-018	W
8	AEROCOM	690-B		FAB-028	W
9	BEECHCRAFT	B-58-P	CP-1618	FAB-031	P
10	LEAR JET	25R-192		FAB-008	W
11	LEAR JET	25D-211		FAB-010	P
12	CESSNA	C-402		FAB-021	P
13	CESSNA	U-206G	CP-1796	FAB-277	W

Figura 3.31 Interfaz Panel principal
Fuente: Elaboración propia

En la siguiente figura se muestra la pantalla de listado y registro de brigadas aéreas, se puede identificar el formulario de registro en la parte izquierda y el listado de registros en la parte derecha.

Brigadas aéreas Listado de brigadas

Brigadas

Registrar brigada +

5 Registros por página

Buscar:

Nombre	Sigla	Dirección	Teléfono	Email	Unidades	Acciones
I Brigada Aérea	I BRIG. AE	Plaza España S/N	72548135	primerabrig@fab.bo	Ver unidades	Opciones
II BRIGADA	II BRIG. AE	COCHABAMBA	3343333		Ver unidades	Opciones
III Brigada aérea	III BRIG. AE	Santos du mont, 3er anillo	45551661		Ver unidades	Opciones
Nueva	NN	Av America	21578412	brig@yahoo.es	Ver unidades	Opciones
Nueva Brigada	NNBB	Av Puente	2285015	nn@orig.com	Ver unidades	Opciones

Mostrando 1 al 5 de 5 registros

← Anterior 1 Siguiente →

Figura 3.32 Interfaz Listado de Brigadas
Fuente: Elaboración propia

Brigadas aéreas Registro de brigadas

Registrar brigada

⚠ Los campos marcados con * son requeridos

Nombre *

Sigla *

Dirección *

Teléfono

Correo electrónico

Figura 3.33 Interfaz Listado de Brigadas

Fuente: Elaboración propia

En la siguiente figura se muestra la pantalla de listado de unidades aéreas, se puede identificar el formulario de registro de unidades en la parte izquierda y el listado de los registros en la parte derecha.

Unidades aéreas Listado de unidades

Unidades 🔍 ↻ 🗖

5

Nombre	Sigla	Dirección	Teléfono	Acciones
Grupo Aereo "31"	G.A. "31"	El alto, zona 16 de julio	54564556	<input type="button" value="Opciones"/>
Grupo aereo presidencial	GAP 1	El Alto		<input type="button" value="Opciones"/>
Grupo aéreo 71	G.A. "71"	El alto, zona 16 de julio	75963212	<input type="button" value="Opciones"/>
Servicio nacional de aerofotogrametría	S.N.A.	Av fuerza aerea	28578541	<input type="button" value="Opciones"/>

Mostrando 1 al 4 de 4 registros ← Anterior 1 Siguiente →

Figura 3.34 Interfaz Listado de unidades

Fuente: Elaboración propia

A continuación, en la siguiente figura se muestra el listado de material aéreo registrado, se puede identificar las opciones: Registrar aeronaves, ver componentes y ver horas/ciclos, esta última opción son los registros transaccionales que realiza el usuario.

Registro de aeronaves Listado de aeronaves

Aeronave	Matricula	Matricula civil	Origen	Condición	Componentes	Horas/ciclos	Opciones
AEROCOM	FAB-028		INCAUTADO	W	Ver componentes	Ver horas/ciclos	Opciones
BAE	FAB-103		FAB		Ver componentes	Ver horas/ciclos	Opciones
BEECHCRAFT	FAB-031	CP-1618	FAB	P	Ver componentes	Ver horas/ciclos	Opciones
CESSNA	FAB-021		FAB		Ver componentes	Ver horas/ciclos	Opciones
CESSNA	FAB-277	CP-1796	Incautado		Ver componentes	Ver horas/ciclos	Opciones
Focker	FAB-101	ACD-665	3	P	Ver componentes	Ver horas/ciclos	Opciones

Figura 3.35 Interfaz Listado de aeronaves
Fuente: Elaboración propia

La siguiente figura muestra el formulario de registro de material aéreo, en la cual se detallan los campos a ser llenados por parte del usuario.

Matricula - Aeronaves Registro de matriculas

Registrar matricula

⚠ Los campos marcados con * son requeridos

Unidad Aerea	-- Seleccione una unidad --
Tipo aeronave	-- Seleccione una tipo de aeronave --
Matricula*	Matricula asignada
Fabricación*	Pais de fabricacion
Año*	Anio de fabricacion
Origen	-- Seleccione origen --
Condición	Condicion de la aeronave
Matricula civil	Matricula civil
Horas de vida	Horas de vida de la nave
Ciclos de vida	Ciclos de la aeronave
Observación	Observaciones

Guardar Cancelar

Figura 3.36 Interfaz Registrar aeronave
Fuente: Elaboración propia

En la siguiente figura se muestra el listado de componentes pertenecientes a una determinada aeronave, se puede identificar las opciones: Registrar componente y ver sub-componentes, la primera despliega el formulario de registro de un nuevo componente, y la segunda opción despliega el listado de los sub-componentes pertenecientes al componente registrado.

Registro de componentes Listado de componentes

Listado de componentes

Registrar componente +

5 ▼ Registros por página Buscar:

Nombre	Horas limite	Ciclos limite	Nro parte	Nro serie	Ver sub-componentes	Ver horas/ciclos	Acciones
motor	180.00:00	300	2-043-002-11	1720001	Ver sub-componentes ↪	Ver horas/ciclos ↪	Opciones ▼

Mostrando 1 al 1 de 1 registros
← Anterior | 1 | Siguiente →

Figura 3.37 Interfaz Listar componentes
Fuente: Elaboración propia

La siguiente figura muestra el listado de registro de horas y/o ciclos de vuelo registrados, pertenecientes a una determinada aeronave. Se puede identificar la opción registrar horas/ciclos, mediante la misma se accede al formulario de registro de horas o ciclos de vuelo.

Registro de horas y/o ciclos de vuelo

Fecha	Horas	Ciclos	Opciones
2014-05-21 02:46:09	08:00:00	2	Opciones ▾
2014-05-21 02:46:38	04:00:00	4	Opciones ▾
2014-05-21 02:47:56	08:00:00	4	Opciones ▾
2014-05-27 17:56:40	08:00:00	2	Opciones ▾

Figura 3.38 Interfaz Listar horas y/o ciclos de vuelo
Fuente: Elaboración propia

La siguiente figura muestra el formulario de registro de horas y/o ciclos de vuelo pertenecientes a una determinada aeronave.

Registro Horas y/o ciclos de vuelo

≡ Nuevo registro

⚠ Los campos marcados con * son requeridos

Fecha* 18/06/2014 19:01:58

Horas de vuelo* 00:00

Ciclos de vuelo* 00

Observaciones*

Guardar Cancelar

Figura 3.39 Interfaz registrar horas y/o ciclos de vuelo
Fuente: Elaboración propia

3.2.7. FASE DE TRANSICIÓN

3.2.7.1. PRUEBAS DEL SISTEMA

Una vez desarrollado el sistema, se procede a efectuar las pruebas correspondientes, con el objetivo de verificar la funcionalidad de los procedimientos que se efectúan.

Las pruebas se efectuaron según los procedimientos más importantes, los que se consideran críticos en el sistema.

El siguiente caso de prueba corresponde al ingreso al sistema, el cual tiene como objeto verificar la funcionalidad de la verificación de usuario y contraseña.

Caso de prueba: Ingreso al sistema	
Descripción	El sistema debe realizar el control de ingreso al sistema
Tipo	Funcional
Precondiciones	El usuario no debe estar autenticado en el sistema
Postcondiciones	El sistema ingresa a la página principal del sistema.
PROCEDIMIENTO DE PRUEBA	
Actor	Respuesta Sistema
1. El actor introduce el nombre de usuario y contraseña. 2. El actor selecciona la opción Ingresar	3. El sistema valida el nombre de usuario y contraseña. 4. El sistema ingresa a la interfaz principal.
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
Cumple	Comentario
<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas

Tabla 3.25 Caso de prueba Ingreso al sistema

Fuente: Elaboración propia

El siguiente caso de prueba corresponde al registro de aeronaves.

Caso de prueba: Registrar material aéreo	
Descripción	El sistema despliega la interfaz de registro para que el actor ingrese los datos requeridos y sean almacenados
Tipo	Funcional
Precondiciones	El usuario debe estar autenticado en el sistema

Postcondiciones	El sistema almacena la información en la base de datos
PROCEDIMIENTO DE PRUEBA	
<u>Actor</u>	<u>Respuesta Sistema</u>
<ol style="list-style-type: none"> 1. El actor selecciona del menú principal “Control” el submenú “Material aéreo”. 2. El actor selecciona la opción Registrar material aéreo 3. El actor selecciona la opción Registrar material aéreo 4. El actor introduce los datos requeridos, y selecciona la opción “Guardar”. 	<ol style="list-style-type: none"> 5. El sistema despliega una interfaz que contiene el listado de todo el material aéreo registrado por unidades. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de materiales aéreos..
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
<u>Cumple</u>	<u>Comentario</u>
<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas

Tabla 3.26 Caso de prueba Registrar material aéreo
Fuente: Elaboración propia

El siguiente caso de prueba corresponde al registro de componentes.

Caso de prueba: Registrar componente	
Descripción	El sistema despliega la interfaz de registro de componentes para que el actor ingrese los datos requeridos y sean almacenados
Tipo	Funcional
Precondiciones	El usuario debe estar autenticado en el sistema
Postcondiciones	El sistema almacena la información en la base de datos
PROCEDIMIENTO DE PRUEBA	
<u>Actor</u>	<u>Respuesta Sistema</u>
<ol style="list-style-type: none"> 1. El actor elige un material aéreo y selecciona la opción “Ver componentes”. 2. El actor selecciona la opción Registrar componente. 3. El actor introduce los datos requeridos, y selecciona la opción 	<ol style="list-style-type: none"> 5. El sistema despliega una interfaz que contiene el listado de todos los componentes registrados del material aéreo seleccionado 6. El sistema despliega el formulario de registro de nuevo componente. 7. El sistema valida los datos

“Guardar”.	ingresados y los almacena en la base de datos.
4. El actor introduce los datos requeridos, y selecciona la opción “Guardar”.	8. El sistema despliega el listado de componentes registrados.
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
Cumple	Comentario
<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas

Tabla 3.27 Caso de prueba Registrar componente
Fuente: Elaboración propia

El siguiente caso de prueba corresponde al registro de horas y/o ciclos de vuelo de las aeronaves y sus componentes.

Caso de prueba: Registrar horas o ciclos de vuelo	
Descripción	El sistema despliega una interfaz en la cual se listan las aeronaves junto a la opción registrar horas y/o ciclos, botón mediante el cual se realiza dicho registro.
Tipo	Funcional
Precondiciones	El usuario debe estar autenticado en el sistema
Postcondiciones	El sistema lista las aeronaves y las horas y/o ciclos de vuelo
PROCEDIMIENTO DE PRUEBA	
Actor	Respuesta Sistema
9. El actor elige un material aéreo y selecciona la opción “Ver horas/ciclos”.	12. El sistema despliega una interfaz que contiene el listado de las horas y/o ciclos de vuelo
10. El actor de registro selecciona la opción Registrar horas/ciclos.	13. El sistema despliega el formulario de registro de horas/ciclos de vuelo.
11. El actor de registro introduce los datos requeridos, y selecciona la opción “Guardar”.	14. El sistema valida los datos ingresados y los almacena en la base de datos.
	15. El sistema despliega el listado de horas y ciclos registrados.
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
Cumple	Comentario

<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas
--	--------------------------

Tabla 3.28 Caso de prueba Registrar horas o ciclos de vuelo

Fuente: Elaboración propia

El siguiente caso de prueba corresponde a la asignación de componentes a aeronaves a partir del ingreso a almacén.

Caso de prueba: Asignar componente	
Descripción	Se realiza la asignación de un determinado componente, a una aeronave específica a partir de almacén.
Tipo	Funcional
Precondiciones	El usuario debe estar autenticado en el sistema
Postcondiciones	El sistema asigna el componente a la aeronave especificada
PROCEDIMIENTO DE PRUEBA	
Actor	Respuesta Sistema
1. El actor selecciona del menú principal “Control” el submenú “Almacén” 2. El actor selecciona una aeronave. 3. El actor selecciona la opción “Guardar”.	4. El Personal de registro elige un componente y selecciona la opción “Asignar a aeronave”. 5. El sistema despliega una interfaz que contiene el listado de aeronaves a asignar. 6. El sistema valida los datos ingresados y los almacena en la base de datos.
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
Cumple	Comentario
<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas

Tabla 3.29 Caso de prueba Asignar componente

Fuente: Elaboración propia

El siguiente caso de prueba corresponde al registro de la programación de 114mantenimiento.

Caso de prueba: Registrar programación de inspección	
Descripción	Se realiza la programación de mantenimiento, el cual esta orientado a una aeronave en específica, por ende a los componentes que pertenecen a dicha aeronave

Tipo	Funcional
Precondiciones	El usuario debe estar autenticado en el sistema El material aéreo debe estar previamente registrado
Postcondiciones	El sistema asigna el componente a la aeronave especificada
PROCEDIMIENTO DE PRUEBA	
<u>Actor</u>	<u>Respuesta Sistema</u>
<ol style="list-style-type: none"> 1. El actor elige un material aéreo y selecciona la opción “Ver programación”. 2. El actor selecciona la opción Registrar programación. 3. El actor introduce los datos requeridos, y selecciona la opción “Guardar”. 	<ol style="list-style-type: none"> 4. El sistema despliega una interfaz que contiene el listado de la programación de inspección del material aéreo seleccionado. 5. El sistema despliega el formulario de registro de programación de inspección. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de programación de inspección.
RESULTADO OBTENIDO	
N° Intento	Observaciones
1	Sin Observaciones
<u>Cumple</u>	<u>Comentario</u>
<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	No se encontraron fallas

Tabla 3.30 Caso de prueba Registrar programación de inspección
Fuente: Elaboración propia

CAPITULO IV

CALIDAD DE SOFTWARE

4.1. DEFINICIÓN

La calidad de software es el conjunto de características de una entidad que le confiere su aptitud para satisfacer las necesidades expresadas y las implícitas, además indicando la dirección hacia donde debemos buscar las soluciones.

Para valorar la calidad de los productos software o sistemas, se proporciona información adecuada sobre los datos necesarios referentes a la calidad del producto, permitiendo una visión profunda sobre el cumplimiento de los objetivos del proyecto.

Entre las varias técnicas para medir la calidad de software se encuentra el estándar ISO 9126, que identifica los siguientes atributo clave de calidad para el software:

- Funcionalidad
- Confiabilidad
- Facilidad de mantenimiento
- Portabilidad
- Usabilidad

4.2. 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:

- Números de entradas del usuario. Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.

- Número de salidas de usuarios. Estas se refieren a informes, mensajes de error, es decir salidas que proporcionen al usuario información orientada a la aplicación.
- Número 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.
- Número de archivos. Se cuenta cada archivo maestro lógico.
- Número de interfaces externas. Se cuenta todas las interfaces legibles por el ordenador que son solicitadas para transmitir información a otro sistema.

De acuerdo a lo mencionado, la tabla muestra los resultados obtenidos en relación al sistema.

Parámetro	Resultado
Entradas de usuario	15
Salidas de usuario	15
Consultas de usuario	4
Número de archivos	22
Interfaces externas	0

Tabla 4.1 Resultados del sistema
Fuente: Elaboración Propia

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

Parámetros de medición	Cuenta		Factor de ponderación MEDIO		Totales
Entradas de usuario	15	X	4	=	60
Salidas de usuario	15	X	5	=	75

Consultas de usuario	4	X	4	=	16
Número de archivos	22	X	10	=	220
Interfaces externas	0	X	7	=	0
CUENTA TOTAL					371

Tabla 4.2 Puntos obtenidos
Fuente: Elaboración Propia

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

$$PF = CUENTA_TOTAL * (Grado_de_Confiabilidad + Tasa_de_error * \Sigma fi)$$

Dónde:

PF = Medida de funcionabilidad

CUENTA_TOTAL = Suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos.

Grado_de_Confiabilidad = Confiabilidad estimada del sistema

Tasa_de_error = Probabilidad subjetiva estimada del dominio de la información, este error estimado es del 1%.

Fi = Valores de ajuste de complejidad que toman los valores de la tabla y que dan respuesta a las preguntas de la tabla

En la siguiente tabla se detalla la ponderación de las respuestas a las preguntas efectuadas.

Sin importancia	0
-----------------	---

Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 4.3 Ponderación de las respuestas
Fuente: Elaboración Propia

A continuación se detalla la ponderación que se asigna a cada factor, en relación al sistema:

ESCALA	Sin	Incidental	Moderado	Medio	Significant	Esencial
Factor	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de recuperación fiables?						X
¿Se requiere comunicación de datos?						X
¿Existen funciones de procesos distribuidos?	X					
¿Es crítico el rendimiento?			X			
¿Será ejecutado el sistema en SO existente?					X	
¿Requiere el sistema de entrada interactiva?						X
¿Requiere el sistema de entrada de datos interactiva sobre múltiples ventanas?					X	
¿Se actualizan los archivos maestros de manera interactiva?						X
¿Son complejos las entradas, salidas, archivos o peticiones?					X	
¿Es complejo el procesamiento interno?				X		
¿Se ha diseñado el código para ser reutilizable?						X
¿Están incluidas en el diseño la conversión e instalación?					X	
¿Se ha diseñado el sistema para soportar múltiples instalaciones?					X	

TOTAL	50
-------	----

Tabla 4.4 Ponderación en relación al sistema
Fuente: Elaboración Propia

Con la obtención de los anteriores datos y considerando un grado de confiabilidad del 65% es que a continuación calculamos el valor del PF.

$$PF = CUENTA_TOTAL * (Grado_de_Confiabilidad + Tasa_de_error * \Sigma fi)$$

$$PF = 371 * (0.65 + 0.01 * 50)$$

$$PF = 426.65$$

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

$$PF = 371 * (0.65 + 0.01 * 70)$$

$$PF_{m\acute{a}ximo} = 500.85$$

Entonces si Σfi es considerada como el 100%, la relación obtenida entre los puntos será.

$$\frac{PF}{PF_{m\acute{a}ximo}} = \frac{426.65}{500.85} = 0.85$$

Por lo tanto la funcionabilidad del sistema es del 85% tomando en cuenta el punto de función máximo

4.3. CONFIABILIDAD

Es la probabilidad de operación libre de fallos de un programa en un entorno determinado y durante un tiempo específico.

Se observa el trabajo hasta que se produzca un fallo en un instante t, se halla la probabilidad de falla con una variable aleatoria continua T, en una función exponencial. La relación es la siguiente:

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$

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

Dónde:

$Fc = 0.85$: Funcionalidad del sistema

$\lambda = 1$: Tasa de fallos en 7 ejecuciones, dentro de un mes

Se realiza el cálculo para la confiabilidad durante el próximo año:

$$F(t) = 0.85 * (e^{(-\frac{\lambda}{7} * 12)})$$

$$F(t) = 0.15$$

La probabilidad de hallar una falla dentro del próximo año es de 15%.

$$\text{CONFIABILIDAD} = 1 - F(t)$$

$$\text{CONFIABILIDAD} = 1 - 0.15$$

$$\text{CONFIABILIDAD} = 85\%$$

Entonces se concluye que, la probabilidad de no hallar una falla dentro del próximo año es de un 85%.

4.4. PORTABILIDAD

El software es apto para funcionar bajo distintas plataformas, tales como Windows o Linux debido a que es una aplicación desarrollada bajo el lenguaje de programación PHP con una base de datos Postgresql y servidor Apache los cuales tienen la característica de ser multiplataforma.

Con respecto al tamaño físico de la aplicación y la base de datos, estos ocupan un espacio aproximado de 100 MB debido a la utilización del framework CakePHP, por lo tanto se

concluye que el sistema no requiere de un gran esfuerzo para su traslado de un entorno de Hardware y Software a otro.

4.5. FACILIDAD DE MANTENIMIENTO

El índice de madurez de software proporciona una indicación de la estabilidad de un producto de software y sirve de métrica de la calidad del mantenimiento del sistema.

Se calcula el índice de madurez del software con la siguiente relación:

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

Donde:

M_t : 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 anterior que se han eliminado en la versión actual.

Reemplazando valores, según la información relacionada al proyecto, se tiene:

$$IMS = [5 - (0 + 1 + 0)]/5$$

$$IMS = 0.8$$

Lo cual indica que la facilidad de mantenimiento, con la que se puede corregir el software en caso de encontrar un error, o se puede adaptar si su entorno cambia o mejora de acuerdo a los requerimientos de los usuarios, es de un 80%

4.5.1. MANTENIMIENTO ADAPTATIVO

El mantenimiento adaptativo ocurrirá cuando se cambien las políticas o estructura organizacional de la institución, cambios para los cuales el sistema podrá adaptarse hasta cierta medida, pero para cambios más profundos será necesario realizar ajustes en los procesos y su adaptación con los nuevos cambios que se vayan a generar.

4.5.2. MANTENIMIENTO PERFECTIVO

El sistema está abierto a añadir o adicionar nuevas funcionalidades de acuerdo a los nuevos requerimientos del cliente, siempre y cuando tengan relación con la lógica del negocio.

4.6. USABILIDAD

La medición de la facilidad de uso se puede entender como la facilidad que el usuario tiene para interactuar con el sistema, tanto como para comprenderlo, aprenderlo y operarlo. A continuación se presenta en la tabla los resultados obtenidos en la capacitación del software a los usuarios del sistema

USUARIOS	FACILIDAD DE COMPRENSIÓN (%)	FACILIDAD DE APRENDIZAJE (%)	FACILIDAD DE OPERACIÓN (%)
Usuario 1	90	90	90
Usuario 2	92	91	90
Usuario 3	95	93	93
Usuario 4	91	90	91
Promedio	92	91	91

Tabla 4.5 Facilidad de uso
Fuente: Elaboración Propia

Por lo tanto, de acuerdo a los resultados de la tabla se obtuvo que la facilidad de uso del sistema es de un 91%, tomando el promedio más bajo obtenido en las consultas realizadas a los usuarios del sistema.

4.7. SEGURIDAD DEL SISTEMA

4.7.1. SEGURIDAD DE BASE DE DATOS

La información en la institución es muy valiosa, es por ello que a nivel de base de datos se tomó en cuenta la validación de los campos a ingresar, con el propósito de evitar cualquier ataque, como por ejemplo el SQL Injection.

Al mismo tiempo, el framework aplicado, establece una conexión segura con la base de datos utilizada, que es Postgresql.

4.7.2. SEGURIDAD DE AUTENTICACIÓN

El sistema es capaz de controlar el acceso a partir de la evaluación de un nombre de usuario y contraseña.

Así mismo el sistema cuenta con un módulo, para la creación de roles y usuarios, es por ello que se contrala también un nivel de acceso a la información mediante estos dos factores.

4.7.3. SEGURIDAD DE LA APLICACIÓN

Mediante el uso de logs de auditoria. El sistema maneja un módulo que se encarga del control de las actividades que realizan los usuarios, ya sea ingreso al sistema o cualquier operación que el mismo vaya a realizar (altas, bajas y modificaciones entre otros).

CAPITULO V

EVALUACIÓN COSTO BENEFICIO

Como su nombre lo sugiere, el método de análisis costo / beneficio se basa en la razón de los beneficios a los costos asociado con un proyecto en particular. Se considera que un proyecto es atractivo, cuando los beneficios derivados de su implementación exceden a sus costos asociados.

5.1. ANALISIS DE COSTOS

Para la determinación del costo del software desarrollado, se usa el modelo constructivo de costo COCOMO.

Para realizar el cálculo de costos relacionados al sistema se toma en consideración los siguientes aspectos:

E: Esfuerzo en hombres /mes

KLDC: Miles de líneas de código

Obtenido en la anterior sección que $PF = 426.65$

Conversión a los puntos de función a KLDC

Tomando en cuenta el número de líneas de código del lenguaje de programación con el que se desarrolla el sistema en base al punto de función obtenido:

La equivalencia dice: 29 líneas de código PHP = 1 Punto de función.

Entonces:

$$LDC = PF * Factor LDC$$

$$LDC = 426.65 * 29$$

$$LDC = 12372.85$$

$$KLDC = (LDC) / 1000$$

$$KLDC = 12.37$$

Por tanto existen 12.37 líneas de código distribuidas para el proyecto.

Ahora se aplica las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones básicas del COCOMO básico tienen la siguiente forma:

$$E = a_b(KLDC)^{b_b}$$

$$D = C_b(E)^{d_b}$$

Dónde:

E: Esfuerzo aplicado en personas

D: Tiempo de desarrollo en meses cronológicos

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

Proyecto de software	a_b	b_b	C_b	d_b
Orgánico	2.1	1.01	2.2	0.34
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 5.1 Relación de valores en el modelo COCOMO

Fuente: Soto, 2011

Reemplazando los datos en las ecuaciones se tiene:

$$E = 2.1(12.37)^{1.01}$$

$$D = 2.2(19.68)^{0.34}$$

$$E = 26.64 \text{ programadores/mes} \quad D = 6.72 \text{ meses} \approx 7 \text{ meses}$$

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

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

$$\text{Número de programadores} = 26.64/6.72$$

$$\text{Número de programadores} = 3.96 \approx 4 \text{ programadores}$$

El salario de un programador puede oscilar entre los 350 USD, cifra que es tomada en cuenta para la estimación siguiente.

$$\text{Costo del software} = \text{Número de programadores} * \text{salario de un programador}$$

$$\text{Costo del software por persona} = 4 * 300 \text{ USD} = 1200 \text{ USD}$$

$$\text{Costo del software} = 1200 \text{ USD} * 7 = 8400 \text{ USD}$$

Por lo tanto, se concluye que para el desarrollo del software se requieren de 4 desarrolladores y un tiempo aproximado de 7 meses, dando como costo total del desarrollo del sistema 9800 USD.

5.2. CALCULO DE BENEFICIOS VAN Y TIR

Para evaluar los beneficios que se obtendrán al implementar el proyecto se calcula con el método VAN y TIR.

5.2.1. VALOR ACTUAL NETO (VAN)

El VAN (Valor Actual Neto) es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá el proyecto, para determinar si luego de descontar la inversión inicial quedará alguna ganancia. Si el resultado es positivo, el proyecto es viable.

Para hallar el VAN del proyecto de inversión se requiere tres valores de acuerdo a la siguiente fórmula:

$$VAN = BNA - INVERSIÓN$$

En donde BNA es el beneficio neto, actualizado, el cual debe ser actualizado de acuerdo a la tasa de descuento TD que es la tasa de oportunidad, rendimiento o rentabilidad mínima, que se espera ganar.

La regla del VAN, que indica que decisión tomar es:

- Si el VAN es mayor a cero, se debe aceptar.
- Si el VAN es igual a cero, se debe ser indiferente.
- Si el VAN es menor que cero, se debe rechazar.

La fórmula que permite el cálculo del VAN es la siguiente:

$$VAN = \sum \frac{Rt}{(1 - i)^t} - I_0$$

Donde:

Rt: Flujo de efectivo (beneficio-costo del periodo t)

t: Periodo de tiempo que oscila de 1 a n.

i: Tasa de rendimiento esperada (10%)

I_0 : Inversión inicial (8400 USD)

Los valores esperados para el presente proyecto se calculan para 5 año esperando ganar 6000 USD por cada año.

Entonces aplicando la fórmula del VAN, se tiene:

$$VAN = BAN - INVERSION$$

$$VAN = \sum \frac{ganancias}{(1 + 0.1)^6} - INVERSION$$

$$VAN = \sum \frac{6000 USD}{(1 + 0.1)^5} - 9800 USD$$

$$VAN = 14344.72 USD$$

Por lo tanto, el VAN es mayor a la inversión producirá ganancias por encima de la rentabilidad, además según el $VAN > 0$ el proyecto es factible.

5.2.2. TASA DE RENTABILIDAD INTERNA (TIR)

Es la tasa de descuento TD de un proyecto de inversión, que permite que el BNA sea igual a la inversión (VAN igual a cero). La TIR es la máxima TD que puede tener un proyecto para que sea rentable.

La regla de la TIR, es la siguiente:

- Si la TIR es mayor que la tasa de descuento, se debe aceptar.
- Si la TIR es igual a la tasa de descuento, se debe ser indiferente.
- Si la TIR es menor a la tasa de descuento, se debe rechazar.

Teniendo en cuenta la siguiente ecuación, se calcula el TIR asociado al presente proyecto.

$$0 = \sum \frac{ganancias}{(1 + i)^5} - INVERSION$$

$$0 = \sum \frac{2000 USD}{(1 + 0.1)^5} - 8400 USD$$

$$i = 66\%$$

Por lo tanto, el proyecto está en las condiciones para retornar el 66% de la inversión, además al ser la TIR superior a la TD, se concluye que el proyecto es viable.

5.2.3. COSTO BENEFICIO

Extrayendo, los costos y beneficios de los anteriores puntos, se tiene:

$$C/B = \frac{\sum \text{ganancias}}{\sum \text{costos}}$$

$$C/B = \frac{14344.72}{8400}$$

$$C/B = 1.70\text{USD}$$

De este resultado se concluye que por 1 USD invertido se tiene una ganancia de 0.70 USD, entonces se concluye que el proyecto es rentable.



CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

La aplicación de tecnologías orientadas a la WEB, permite el acceso a la información desde diferentes lugares, sin la necesidad de instalar un software adicional para que el sistema entre en funcionamiento. Por otro lado esta misma tecnología al permitir la integración por medio de una interfaz única, el explorador, se adapta perfectamente para hacer posible la implementación del sistema de una forma ágil y adecuada.

Todas estas características son fundamentales y adecuadas para el control de componentes de las aeronaves, ya que el control de las horas y/o ciclos de vuelo se realizan en diferentes lugares físicos, lo que hace que los usuarios cuenten con información al instante y puedan al mismo tiempo actualizar dicha información según sus necesidades.

6.1. CONCLUSIONES

Los siguientes objetivos específicos, ayudaron a cumplir el objetivo principal:

- “Realizar el seguimiento de los componentes de las aeronaves”, este objetivo se cumplió mediante el desarrollo de un módulo que permite el registro de componentes y la asignación del mismo desde almacén a una aeronave específica.
- “Efectuar el control de horas de vuelo y/o ciclos para el mantenimiento de los componentes de las aeronaves”, objetivo cumplido mediante el desarrollo de un módulo que permite el registro de horas y/o ciclos de vuelo efectuados por las aeronaves.
- “Realizar el control de las horas y/o ciclos de la estructura de la aeronave”, el mismo se cumplió mediante la implementación de un módulo que permite el registro de los componentes de las aeronaves, así como también el registro de horas y/o ciclos de vuelo por componente.
- “Emitir reportes acerca de: aeronaves por brigadas y unidades, estado de componentes de aeronaves específicas, detalle de componentes de aeronaves,

detalle completo de aeronaves”, este objetivo se cumplió mediante la construcción de diferentes módulos, los cuales permiten el registro de brigadas aéreas, registro de unidades aéreas, registro de aeronaves (incluidos el detalle de las mismas), registro de componentes de aeronaves (mayores y rotables), registro de horas y/o ciclos de vuelo, registro de programación de mantenimiento, mismos que alimentan a la base de datos de información necesaria, la cual es procesada por el sistema para la emisión de reportes en formato PDF y Web.

- “Diseñar e implementar una base de datos, que permita responder a las necesidades que tenga la institución”, el objetivo fue alcanzado mediante el diseño y la implementación de una base de datos, efectuada según la necesidad de información de los usuarios del sistema.

Evaluando los distintos objetivos específicos citados anteriormente y las tareas desarrolladas para cumplirlos, se puede concluir que el objetivo general “Desarrollar un sistema Web de seguimiento y control de mantenimiento de los componentes, mayores y rotables, de las aeronaves de la Fuerza Aérea Boliviana, para mejorar la eficiencia de operatividad de las aeronaves” se cumplió.

Así como también las métricas de calidad, evaluadas en el Capítulo IV, muestran la factibilidad del proyecto realizado.

6.2. RECOMENDACIONES

Teniendo en cuenta la institución tiene como planes migrar los sistemas que tiene a software libre, entonces se recomienda:

- Realizar evaluaciones periódicas del sistema en cuanto a mantenimiento y actualización, con el objeto de contar con un sistema vigente y sea útil para el manejo por parte de los usuarios.
- Realizar copias de seguridad, periódicas con el propósito de garantizar la integridad de los datos almacenados.

- Implementar un módulo que permita el seguimiento financiero de las aeronaves y sus componentes.
- Implementar un módulo que permita la administración del inventario en almacenes de los componentes correspondientes a las aeronaves.
- Implementar un módulo de seguimiento, desde el punto de vista de la tripulación de las aeronaves, con el objeto de tener un control específico en vuelo de las mismas.



BIBLIOGRAFÍA

Referencias Bibliográficas

1. [Hernández, Fernández, Baptista, 2006] Hernández Sampieri Roberto, Fernández Collado Carlos, Baptista Lucio Pilar, 2006, METODOLOGÍA DE LA INVESTIGACIÓN CUARTA EDICIÓN, Iztapalapa México D.F.
2. [Nieves, Ucán, Menendez, 2014] Citlali G. Nieves-Guerrero, Juan P. Ucán-Pech, Víctor H. Menéndez-Domínguez, 2014, UWE EN SISTEMAS DE RECOMENDACIÓN DE OBJETOS DE APRENDIZAJE. APLICADO INGENIERÍA WEB: UN MÉTODO EN CASO DE ESTUDIO, Mérida, Yucatán, México.
3. [Pressman, 2006] Pressman Roger S., 2006, INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO SEXTA EDICIÓN.
4. INTECO, 2009, INGENIERÍA DEL SOFTWARE: METODOLOGÍA Y CICLOS DE VIDA, España.
5. [Juzgado, 1996] J. Juzgado, N., 1996, PROCESOS DE CONSTRUCCIÓN DEL SOFTWARE Y CICLOS DE VIDA, Universidad Politécnica de Madrid.
6. [Balduino, 2007 en Novoa, Ramos, 2013] Novoa Infante Andrés Felipe, Ramos Garzón Cristian Duván, 2013, DESARROLLO DE UN PROTOTIPO DE SOFTWARE PARA EL SEGUIMIENTO DE SOLICITUDES DE CRÉDITO DIRIGIDO A LA COMPAÑÍA SOFINANCOL S.A.S., Bogotá, Colombia.
7. [Rodríguez, 2009] Rodriguez Ana Nieves del Valle, 2009, METODOLOGÍAS DE DISEÑO USADAS EN INGENIERÍA WEB, SU VINCULACIÓN CON LAS NTICS.
8. [Quispe, 2001] Quispe Apaza Santos, 2011, SISTEMA WEB DE REGISTRO, CONTROL Y SEGUIMIENTO DE LOS PROCESOS DE AUDITORÍA INTERNA CASO: SENAPE (SERVICIO NACIONAL DE PATRIMONIO DEL ESTADO), La Paz, Bolivia.

9. [Bolívar, Hernández, Parra, Silva, 2011] Bolívar Bryan, Hernández Jorge, Parra Aquileo, Silva Ángel, 2011, CAKEPHP VS ZEND FRAMEWORK.

Referencias de Internet

1. [The Eclipse Foundation, 2006] The Eclipse Foundation, 2006, OPENUP/BASIC
<http://epf.eclipse.org/wikis/openupsp/index.htm>
2. [The Eclipse Foundation, 2012] The Eclipse Foundation, 2012, OPENUP.
<http://epf.eclipse.org/wikis/openup/>



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

ANEXOS

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-01.01

Nombre	Registrar brigada aérea		
Código	CU-01.01	Estado (Fase)	Elaboración
Actor (es)	Personal registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<p>El Personal registro selecciona del menú principal “Registrar” el submenú “Grandes unidades”.</p> <p>El sistema despliega una interfaz que contiene el formulario para el registro de nuevas brigadas, y el listado de brigadas aéreas registradas.</p> <p>El usuario Personal registro introduce los datos requeridos, y selecciona la opción “Guardar”.</p> <p>El sistema valida los datos ingresados y los almacena en la base de datos.</p> <p>El sistema agrega en la lista la nueva brigada registrada.</p>		
Escenarios alternativos			
Alternativa 1			
Puntos de extensión			
PostCondición	Los datos de la nueva brigada se almacenan en la base de datos		

Tabla A.1. Especificación Registrar base aérea
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-01.02

Nombre	Modificar brigada aérea		
Código	CU-01.02	Estado (Fase)	Elaboración
Actor (es)	Personal registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<p>El usuario Personal registro elige una brigada aérea y selecciona de opciones la opción “Modificar”</p> <p>El sistema despliega los datos de la brigada aérea en la parte izquierda de la interfaz.</p> <p>El usuario Personal registro introduce los datos actualizados, y selecciona la opción “Guardar”.</p> <p>El sistema valida los datos ingresados y los almacena en la base de datos.</p> <p>El sistema actualiza en la lista la brigada aérea modificada.</p>		
Escenarios alternativos			
Alternativa 1	3A. El usuario Personal registro puede seleccionar la opción “Cancelar”		
Puntos de extensión			
PostCondición	Los datos actualizados de la brigada aérea se almacenan en la base de datos		

Tabla A.2. Especificación Modificar base aérea

Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-01.03

Nombre	Registrar unidad aérea		
Código	CU-01.03	Estado (Fase)	Elaboración
Actor (es)	Personal registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal registro selecciona del menú principal “Registrar” el submenú “Pequeñas unidades”. 2. El sistema despliega una interfaz que contiene el formulario para el registro de nuevas unidades, y el listado de unidades aéreas registradas. 3. El usuario Personal registro introduce los datos requeridos, y selecciona la opción “Guardar”. 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema agrega en la lista la nueva brigada registrada. 		
Escenarios alternativos			
Alternativa 1			
Puntos de extensión			
PostCondición	Los datos de la nueva brigada se almacenan en la base de datos		

Tabla A.3. Especificación Registrar unidad aérea
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-01.04

Nombre	Modificar unidad aérea		
Código	CU-01.04	Estado (Fase)	Elaboración
Actor (es)	Personal registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<ol style="list-style-type: none"> 1. El usuario Personal registro elige una unidad aérea y selecciona de opciones la opción “Modificar” 2. El sistema despliega los datos de la unidad aérea en la parte izquierda de la interfaz. 3. El usuario Personal registro introduce los datos actualizados, y selecciona la opción “Guardar”. 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema actualiza en la lista la brigada aérea modificada. 		
Escenarios alternativos			
Alternativa 1	3A. El usuario Personal registro puede seleccionar la opción “Cancelar”		
Puntos de extensión			
PostCondición	Los datos actualizados de la unidad aérea se almacenan en la base de datos		

Tabla A.4. Especificación Modificar unidad aérea
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-01.05

Nombre	Listar base aérea		
Código	CU-01.05	Estado (Fase)	Elaboración
Actor (es)	Personal revisor		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal revisor selecciona del menú principal “Bases aéreas” el submenú “Listar brigadas aéreas”. 2. El sistema despliega una interfaz que contiene el listado de las brigadas aéreas registradas. 3. El usuario Personal revisor elige una brigada aérea y selecciona “Ver unidades”. 4. El sistema despliega una interfaz que contiene el listado de todas las unidades registradas pertenecientes a la brigada aérea seleccionada. 		
Escenarios alternativos			
Alternativa 1	<p>2A. Ver detalle de la brigada</p> <ol style="list-style-type: none"> 1. El usuario elige una brigada aérea y selecciona ver detalle. 2. El sistema despliega la interfaz que contiene toda la información de la brigada aérea seleccionada. 		
Alternativa 2	<p>4A. Ver detalle de la brigada</p> <ol style="list-style-type: none"> 1. El usuario elige una brigada aérea y selecciona ver detalle. 		

	2. El sistema despliega la interfaz que contiene toda la información de la brigada aérea seleccionada
Puntos de extensión	
PostCondición	Se listan los registros de las brigadas y unidades aéreas.

Tabla A.5.. Especificación Listar base aérea
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.01

Nombre	Registrar aeronave		
Código	CU-02.01	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro selecciona del menú principal “Registrar” el submenú “Aeronaves”. 2. El sistema despliega una interfaz que contiene el formulario para el registro de nuevas aeronaves, y el listado de aeronaves registradas 3. El usuario Personal de registro introduce los datos requeridos, y selecciona la opción “Guardar”. 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema agrega en la lista la nueva aeronave registrada. 		

Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	Los datos de la nueva aeronave se almacenan en la base de datos

Tabla A.6.. Especificación Registrar aeronave
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.02

Nombre	Modificar aeronave		
Código	CU-02.02	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	La aeronave debe estar previamente registrada		
Escenario básico	<ol style="list-style-type: none"> 1. El usuario Personal de registro elige una unidad aeronave y selecciona de opciones la opción “Modificar” 2. El sistema despliega los datos de la aeronave en la parte izquierda de la interfaz. 3. El usuario Personal de registro introduce los datos actualizados, y selecciona la opción “Guardar”. 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema actualiza en la lista la aeronave modificada. 		

Escenarios alternativos	
Alternativa 1	3A. El usuario Personal registro puede seleccionar la opción “Cancelar”
Puntos de extensión	
PostCondición	Los datos actualizados de la aeronave se almacenan en la base de datos

Tabla A.7. Especificación Modificar aeronave
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.03

Nombre	Registrar material aéreo		
Código	CU-02.03	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El usuario debe estar autenticado en el sistema		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro selecciona del menú principal “Control” el submenú “Material aéreo”. 2. El sistema despliega una interfaz que contiene el listado de todo el material aéreo registrado por unidades. 3. El usuario Personal de registro selecciona la opción Registrar material aéreo. 4. El sistema despliega el formulario de registro de nuevo material aéreo. 5. El usuario Personal de registro introduce los datos requeridos, y 		

	<p>selecciona la opción “Guardar”.</p> <p>6. El sistema valida los datos ingresados y los almacena en la base de datos.</p> <p>7. El sistema despliega el listado de materiales aéreos.</p>
Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	Los datos del nuevo material aéreo se almacenan en la base de datos

Tabla A.8.. Especificación Registrar matricula
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.04

Nombre	Modificar material aéreo		
Código	CU-02.04	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	La aeronave debe estar previamente registrada		
Escenario básico	<ol style="list-style-type: none"> 1. El usuario Personal de registro elige registro de material aéreo y selecciona de opciones la opción “Modificar” 2. El sistema despliega los datos del material aéreo seleccionado. 3. El usuario Personal de registro introduce los datos actualizados, y selecciona la opción “Guardar”. 		

	<ol style="list-style-type: none"> 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema despliega el listado de materiales aéreos registrados.
Escenarios alternativos	
Alternativa 1	3A. El usuario Personal registro puede seleccionar la opción “Cancelar”
Puntos de extensión	
PostCondición	Los datos actualizados del material aéreo se almacenan en la base de datos

Tabla A.9. Especificación Modificar matricula
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.05

Nombre	Registrar componente		
Código	CU-02.05	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El material aéreo debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un material aéreo y selecciona la opción “Ver componentes”. 2. El sistema despliega una interfaz que contiene el listado de todos los componentes registrados del material aéreo seleccionado. 3. El usuario Personal de registro selecciona la opción Registrar componente. 		

	<ol style="list-style-type: none"> 4. El sistema despliega el formulario de registro de nuevo componente. 5. El usuario Personal de registro introduce los datos requeridos, y selecciona la opción “Guardar”. 6. El sistema valida los datos ingresados y los almacena en la base de datos. 7. El sistema despliega el listado de componentes registrados.
Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	Los datos del nuevo componente se almacenan en la base de datos

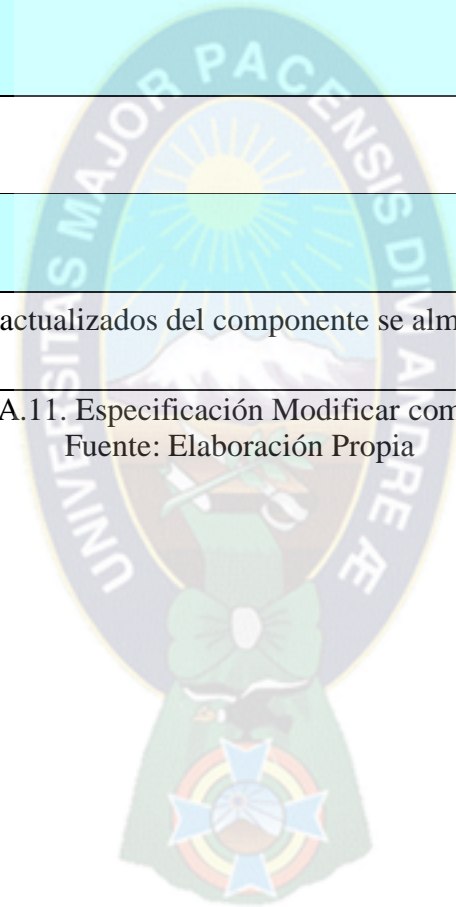
Tabla A.10. Especificación Registrar componente
Fuente: Elaboración Propia

La siguiente tabla muestra la especificación del caso de uso correspondiente al CU-02.06

Nombre	Modificar componente		
Código	CU-02.06	Estado (Fase)	Elaboración
Actor (es)	Personal de registro		
Precondición	El componente debe estar previamente registrado		
Escenario básico	<ol style="list-style-type: none"> 1. El Personal de registro elige un componente en específico y selecciona de opciones “Modificar”. 2. El sistema despliega una interfaz que contiene los datos del 		

	<p>componente a modificar.</p> <ol style="list-style-type: none"> 3. El usuario Personal de registro introduce los datos actualizados y selecciona la opción “Guardar”. 4. El sistema valida los datos ingresados y los almacena en la base de datos. 5. El sistema despliega el listado de componentes registrados.
Escenarios alternativos	
Alternativa 1	
Puntos de extensión	
PostCondición	Los datos actualizados del componente se almacenan en la base de datos

Tabla A.11. Especificación Modificar componente
Fuente: Elaboración Propia



DOCUMENTACIÓN