

CAPITULO III

PROPUESTA

3.1. TEMA

Implementación de un sistema para el control del recurso humano en la Base Aérea Cotopaxi de la Fuerza Aérea Ecuatoriana

3.2. Presentación

De acuerdo al avance tecnológico que soporta el país, la Base Aérea Cotopaxi se ha visto en la necesidad de implementar una herramienta que le brinde mayor agilidad y control en la información que se genera ya que los procesos que se llevan a cabo son completamente automatizados generando así el ahorro de los recursos que posee el reparto.

Uno de los beneficios que ofrece este sistema a la Base Aérea Cotopaxi es que se conseguirá acceder a la información más relevante del personal tanto Oficiales como Aerotécnicos que prestan sus servicios en esta importante institución militar, todo el proceso que se realiza en la asignación de guardias, guardias especiales, semanas, ubicación en cada uno de los pelotones, control de la información personal y sobre todo el seguimiento que van a poder realizar el personal a sus respectivos roles de guardia.

3.3. Justificación

Los procesos automatizados, son factores de vital importancia en la institución, es necesario que la información sea procesada y almacenada de una forma más efectiva para agilizar los procesos de control y así lograr seguridad y efectividad en las actividades que se realizan dentro de la institución. Con el desarrollo de un sistema automatizado para el control del Recurso Humano que abarque las necesidades y requerimientos del personal y la institución, esto proporcionará un mejor flujo de la información así como efectividad y seguridad al momento de tomar decisiones por parte del mando.

Al implementar este sistema habrá un impacto psicosocial positivo en el personal que labora en el Departamento de Personal de la institución, el mismo que proporcionará, una información confiable y veraz, agilizando y facilitando el trabajo en el proceso de control.

La automatización del proceso será un aporte técnico muy importante para la institución y su personal, ya que esto permitirá mantenerse actualizada y competitiva. Además servirá como aporte científico para futuras implementaciones.

Se ha conseguido un total apoyo a la idea de implementar un sistema automatizado para el control del Recurso Humano.

Todo el personal se siente partidario por hacer realidad un proyecto acorde a los avances de la era tecnológica de la información.

Las herramientas que permitieron el desarrollo del sistema como ya se mencionó anteriormente son Open Source y Firewall porque son gratuitas y de alto nivel, tales como AJAX, XML, PHP, APACHE, LA PLATAFORMA LINUX, MySql, mismas que nos apoyaron en el proceso de construcción de las diferentes características que debe tener la aplicación, tales como las entradas, salidas procedimientos y especificaciones de control.

3.4. Objetivos

3.4.1. Objetivo General

Implementar un sistema automatizado para el control del Recurso Humano, que controle y registre de manera segura, precisa y eficiente la información aprovechando los recursos económicos, humanos y computacionales que posee la institución.

3.4.2. Objetivos Específicos

- Analizar la forma en que se realiza actualmente el proceso de control del Recurso Humano (concesión de permisos, asignación de pelotones de guardia, prestación de servicios, comisiones, cursos, vacaciones, situación médica, etc.) en la Base Aérea Cotopaxi de la Fuerza Aérea Ecuatoriana.
- Diseñar un sistema para controlar de una manera eficiente y segura la disponibilidad del recurso humano de la Base Aérea Cotopaxi.

- Implementar un sistema que permita automatizar el control de personal militar y civil para concesión de permisos, asignación de pelotones de guardia, prestación de servicios, comisiones, cursos, vacaciones, situación médica, etc. de la Base Aérea Cotopaxi de la Fuerza Aérea Ecuatoriana.

3.5. Desarrollo del Sistema

3.5.1. Lenguaje de Modelado Unificado (UML)

Diagramas de Casos de Uso

Para el diseño de las aplicaciones o software informáticos se está usando el lenguaje UML (Lenguaje de Modelado Unificado), ya que este lenguaje presenta una visión general y clara de las definiciones, objetivos, alcances y requisitos que se debe tener en cuenta para el desarrollo de una aplicación o proyecto informático, además se basa en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado.

UML consta de varios diagramas que conforman el conjunto de esquemas propuesto por la notación, estos diagramas son utilizados de acuerdo con el proyecto informático a desarrollarse. Provee de beneficios significativos para los ingenieros de software y las organizaciones ayudando a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo. A continuación se mencionan los diagramas que se tomarán en cuenta para el diseño del sistema de personal de la Base Aérea Cotopaxi.

Son los principales medios para capturar la funcionalidad del software desde la perspectiva del usuario y muchas veces puede remplazar al documento "requisitos funcionales".

En los casos de uso se utiliza una metodología de tal forma que permite enfocar una primera aproximación del funcionamiento del sistema a desarrollarse.

Para los casos de uso UML presenta una metodología de fácil comprensión tanto para el diseñador del software como para los usuarios, a continuación presentamos la simbología para realizar los casos de uso.

Actor: Un Actor es un rol que un usuario desempeña con respecto al software. 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 software.

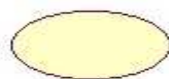
GRÁFICO N° 3.1: ACTOR



FUENTE: http://es.wikipedia.org/wiki/Imagen:Notacion_Caso_de_Uso_actor.png

Caso de Uso: Un Caso de Uso es una representación de una unidad discreta de trabajo realizada por un usuario (u otro software) usando el software en operación. Se ejecuta en su totalidad o no se ejecuta nada, devolviendo algo de valor al usuario.

GRÁFICO N° 3.2: CASO DE USO



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>

Relación de asociación: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

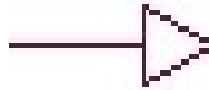
GRÁFICO N° 3.3: ASOCIACIONES ENTRE OBJETOS



FUENTE: http://www.dcc.uchile.cl/~psalinas/uml/aso_obj.html

Relación de Generalización: Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>). Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

GRÁFICO N° 3.4: GENERALIZACIÓN



FUENTE: <http://es.wikipedia.org/wiki/Diagramadecasosdeuso#Generalizacion>

Extends: Es recomendable utilizar cuando un caso de uso es similar a otro (características).

Includes: Es recomendable utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

3.5.2. Diagrama de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el software, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

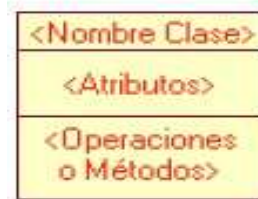
Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Clase: Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones:

GRÁFICO N° 3.5: CLASE



FUENTE: [http://en.wikipedia.org/wiki/Clase_\(UML\)](http://en.wikipedia.org/wiki/Clase_(UML))

En la figura se destacan las siguientes partes:

- **Superior:** Contiene el nombre de la Clase.
- **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase.
- **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno.

Para identificar las clases, se usan los diagramas de casos de uso y de la documentación de los mismos, además la búsqueda de las clases se debe realizar desde 3 perspectivas diferentes:

- El límite entre el software y los actores, dichas clases se denominan clases de interfaz.
- Toda la información que usa y/o genera el sistema, dichas clases se denominan clases de entidad.
- Lógica del control del sistema, dichas clases se denominan clases de control.

Atributos y métodos: Los atributos o características y los métodos de una clase pueden ser de tres tipos, estos son:

- **Public:** Indica que será visible tanto dentro como fuera de la clase, es decir es accesible desde todos lados.
- **Private:** Indica que será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).
- **Protected:** Indica que será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven.

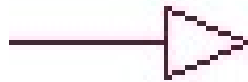
Relaciones entre Clases: Una vez definido el concepto de Clase, es necesario explicar cómo se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- **uno o muchos:** 1..* (1..n)
- **0 o muchos:** 0..* (0..n)
- **número fijo:** m (m denota el número).

Relación de Herencia (Especialización/Generalización): Indica que una subclase hereda los métodos y atributos especificados por una Súper Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Súper Clase (public y protected):

GRÁFICO N° 3.6: HERENCIA



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/herencia.html>

Relación de agregación: Para modelar objetos complejos, bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

- **Por Valor:** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada **Composición** (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo"). La composición (por Valor) se destaca por un rombo relleno.
- **Por Referencia:** Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada.

Agregación (el objeto base utiliza al incluido para su funcionamiento). La agregación (por Referencia) se destaca por un rombo transparente.

La flecha en este tipo de relación indica la navegabilidad del objeto referenciado. Cuando no existe este tipo de particularidad la flecha se elimina.

GRÁFICO N° 3.7: RELACIÓN DE AGREGACIÓN



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/agregacion.html>

Relación de asociación: La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

GRÁFICO N° 3.8: RELACIONES DE ASOCIACIÓN



FUENTE: http://usuarios.lycos.es/oopere/uml_relaciones_de_asociacion.htm

Relación de dependencia o Instanciación (uso): Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra.

GRÁFICO N° 3.9: RELACIÓN DE DEPENDENCIA



FUENTE: <http://www-gris.det.uvigo.es/~avilas/UML/node18.html>

Es necesario indicar que este diagrama puede contener clases sencillas, es decir sin atributos ni operaciones, ya que es una primera aproximación para realizar el diagrama de clases de diseño, en el cual se utiliza la misma simbología y conceptos expuestos en este tema. En estos diagramas de Diseño se añaden los detalles referentes al lenguaje de programación que se vaya a usar. Por ejemplo, los tipos de los atributos y parámetros se expresarán según la sintaxis del lenguaje de implementación escogido.

3.5.3. Diagramas de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes como Cliente (Actor) u Objetos (Clases) en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases de análisis o el de Casos de Uso. Así los componentes de un diagrama de interacción son: un Objeto o Actor, mensaje de un objeto a otro objeto, mensaje de un objeto a si mismo.

Objeto/Actor: En el eje horizontal se colocan los rectángulos que representa una instancia de un Objeto en particular o actores participantes en la interacción, mientras que el eje vertical (línea punteada) representa el tiempo en la que se colocan las llamadas a métodos del objeto sin un orden prefijado. Cada objeto o actor tiene una línea vertical. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o junto a las transiciones o activaciones a las que se refieren.

GRÁFICO N° 3.10: OBJETO/ACTOR



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/objeto/actor.html>

Mensaje de un objeto a otro objeto: Se representa por una flecha entre un objeto y otro, representa la llamada a un método (operación) de un objeto en particular.

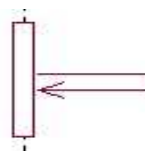
GRÁFICO N° 3.11: PASO DE MENSAJES DE UN OBJETO A OTRO



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html#objeto>

Mensaje al Mismo Objeto: No solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.

GRÁFICO N° 3.12: PASO DE MENSAJES AL MISMO OBJETO



FUENTE: <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html#objeto>

3.6. Análisis del Sistema.

Para el diseño del sistema se utilizó la recopilación de la información así como también se lo represento gráficamente en la herramienta de Rational Rose. Para lo que es el grafico de Caos de Uso general se obtuvo la siguiente representación la misma que está dada por:

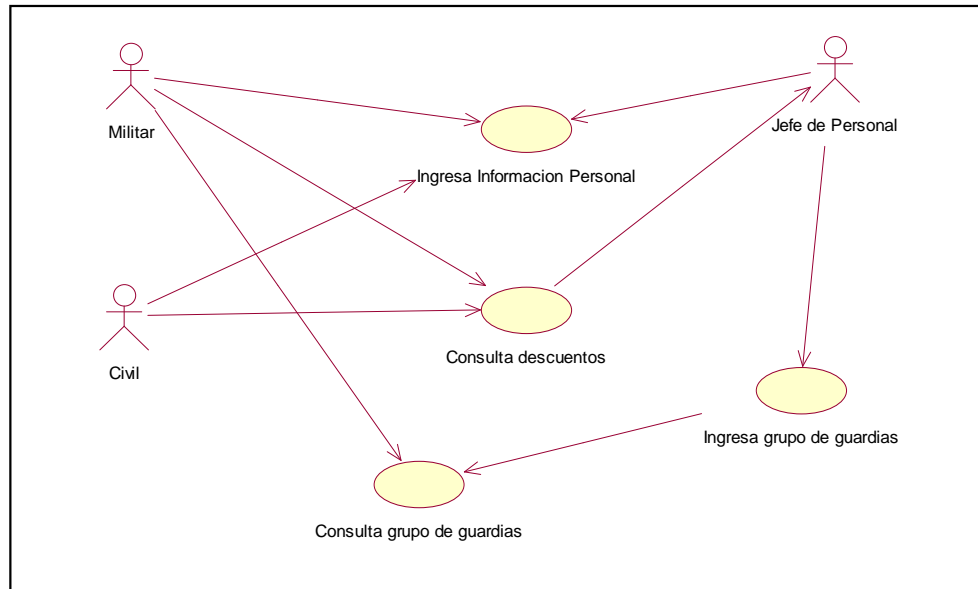


GRÁFICO N° 3.13: Diagrama de Casos de Uso general del sistema de personal

FUENTE: Los Investigadores

Como se puede observar en el sistema de administración de la Base Aérea Cotopaxi se puede observar claramente que los actores que van a interactuar en la aplicación son el administrador en calidad de jefe de personal, y los otros dos actores son los que conforman al personal militar y al personal civil.

De igual manera para un mejor entendimiento de que es lo que se va a lograr con esta aplicación se realizó el análisis del sistema para la administración de personal mediante un Diagrama de Secuencias.

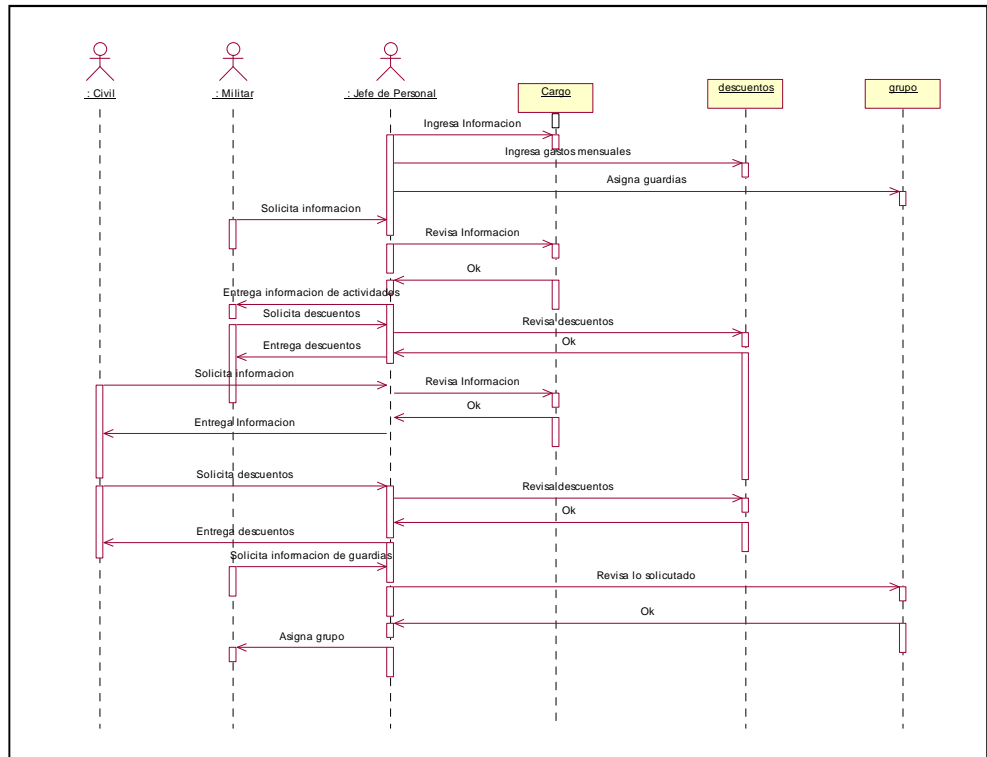


GRÁFICO N° 3.14: Diagrama de Secuencia del sistema de personal

FUENTE: Los Investigadores

3.7. Diseño del Sistema.

Para el diseño del sistema se optó por seguir con la herramienta de Rational Rose para el diseño mediante el diagrama de clases en lo cual se puede escoger la multiplicidad de acuerdo a las necesidades de los sistemas y particularmente de este sistema de administración de personal la misma que garantiza el flujo de información de acuerdo a las necesidades institucionales.

3.8. Desarrollo del Sistema.

El Desarrollo del presente sistema se lo ha realizado en varios lenguajes de programación como son: PHP, JavaScript, Ajax; el lenguaje de programación PHP lo hemos utilizado ya que tiene comandos y funciones que trabajan en conjunto con MySQL, el lenguaje de programación Ajax nos ha permitido realizar la conexión que

utiliza el sistema ya que en este se especifica el nombre y la base de datos que se usa, hemos utilizado esta técnica porque es aceptada en cualquier plataforma y navegador, no es compleja su utilización, y beneficia las aplicaciones Web, teniendo una mayor interactividad, velocidad y usabilidad de los procesos.

Para obtener una mejor visualización de los cada uno de los elementos que posee el Sistema se ha utilizado el lenguaje JavaScript.

Para iniciar la aplicación de Administración de personal de la Base Aérea Cotopaxi se cuenta con una pantalla de ingreso de contraseñas la misma que se encuentra desarrollada mediante el algoritmo de encriptación de MD5 el mismo que controla a través de la base de datos de usuarios del sistema el mismo que cuenta con los privilegios de acuerdo al perfil del usuario y con el estado es decir si se encuentra activo o en pasivo dentro del sistema.



The image shows a web-based login interface for the 'Control de Personal' system. At the top left is the logo of the Fuerza Aérea Ecuatoriana (FAE). A banner across the top displays the text 'Fuerza Aérea Ecuatoriana' over a background of clouds and aircraft. Below the banner, the title 'Control de Personal' is centered, followed by the word 'Acceso'. To the left of the login fields is a graphic of a padlock and a large red '@' symbol. The login form consists of two input fields: 'Username: MARIO' and 'Password: *****'. Below these fields is a 'Login In' button.

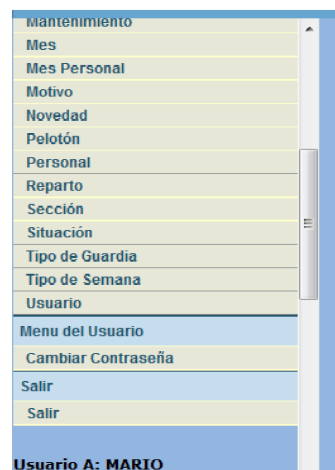
El ingreso esta dado como se puede ver solamente el nombre de usuario y la contraseña la misma que está representada por puntos.



La pantalla de ingreso de usuario consta con la presentación con un diseño clásico de la Fuerza Aérea Ecuatoriana al lado izquierdo tenemos el menú, el mismo que tiene el menú de consultas de usuario, el de administración y por ultimo tenemos el menú de ingreso de usuario.



La pantalla de ingreso de usuarios, esta es la que controla la administración de todos y los privilegios de acuerdo a los perfiles de los usuarios la pantalla está dada por el código que se auto genera, se debe ingresar nombre se recomienda que solo sea el nombre o el apellido o buscar un nombre de usuario que identifique la contraseña se puede poner una combinación de números y letras, el rol del usuario que puede ser administrador, usuario avanzado, o usuario normal que tiene privilegios solamente para realizar las consultas que se requiere como se lo va a ver más adelante .



La información en el menú de consulta de los usuarios está dado de acuerdo a lo que se necesita es decir reportes de forma general de los usuarios como informes de guardias tanto especiales como las que regularmente se realiza a lo largo del mes así como también cuando se tiene comisión de servicios sea en otro reparto o ciudad.

Como se puede observar un usuario llamado Juan solo tiene privilegios para poder desplegar este tipo de consulta, no así que el usuario Mario tiene los privilegios completos ya que este usuario al igual que Wladimir hasta el momento cuenta como usuario con privilegios de Administrador es decir tiene los privilegios completos para poder realizar cualquier cambio o agregar nuevos usuarios o perfiles de acuerdo a las actividades que se generan.

Nombres	Situación	Pelotón	Grupo Tarea	Grado	Sección	Estado Civil
MARIO MONGE	ACCIDENTAL	ALFA	UNO	SARGENTO SEGUNDO	REPARTO	SOLTERO
WLADIMIR HERRERA	ACCIDENTAL	ALFA	UNO	SARGENTO SEGUNDO	REPARTO	SOLTERO

s del 1 al 2 del total de 2

Todas las pantallas tienen el mismo formato como por ejemplo la edición, guardar, eliminar y las barras de navegación se lo tiene que realizar de acuerdo a las necesidades que se tenga al momento del ingreso.



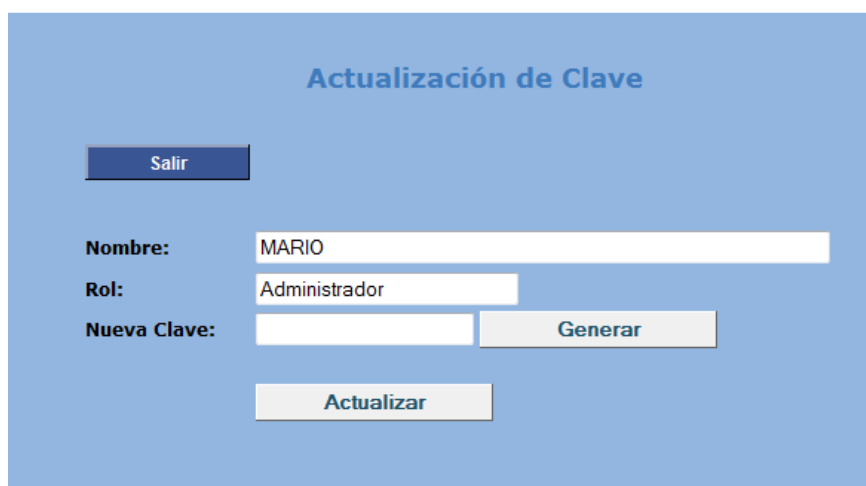
The image shows a web-based form titled "Ingreso de Personal" (Personnel Entry) for the "Fuerza Aérea Ecuatoriana" (Ecuadorian Air Force). The form is set against a blue background with a header image of clouds and the text "Fuerza Aérea Ecuatoriana". A "Salir" (Exit) button is located at the top left. The form fields are as follows:

- Cédula:** Text input field.
- Nombres:** Text input field.
- Situación:** Dropdown menu with "Selecione.." and a downward arrow.
- Pelotón:** Dropdown menu with "Selecione.." and a downward arrow.
- Grupo Tarea:** Dropdown menu with "Selecione.." and a downward arrow.
- Grado:** Dropdown menu with "Selecione.." and a downward arrow.
- Sección:** Dropdown menu with "Selecione.." and a downward arrow.
- Estado Civil:** Dropdown menu with "Selecione.." and a downward arrow.
- Reparto:** Dropdown menu with "Selecione.." and a downward arrow.
- Antigüedad:** Text input field.
- Cefae:** Text input field.
- Estado:** Dropdown menu with "Seleccionar Opción" and a downward arrow.
- Observación:** Text input field.

La pantalla de ingreso de toda la aplicación tiene el mismo formato para todos solo que varía de acuerdo al número de campos que se tenga es decir entre mayor número de campos más largo será el ingreso de la información ya que solo eso varía entre formularios de ingreso y consultas.



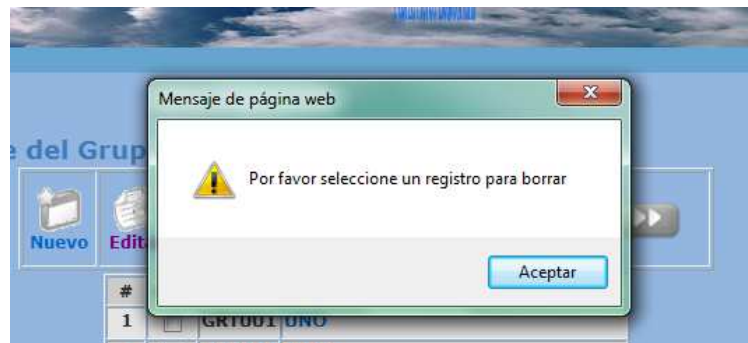
Para los reportes se los tiene que realizar de acuerdo a los ingresos es así que el combo box en la barra de navegación se lo puede realizar manualmente pero si se ingresa la primera letra del nombre la barra baja hasta donde se encuentra el usuario que se necesita consultar.



Para la actualización de claves de seguridad se tiene una pantalla para migración de la contraseña nueva y vieja ya que es un requisito fundamental en todo sistema ya que por seguridad informática se tiene que dar mantenimiento a las contraseñas cada cierto tiempo para prevenir posibles ataques.



El sistema consta de todo tipo de seguridades particularmente para actividades que no tienen definición alguna como por ejemplo al momento de la edición se tiene que tener marcado el registro para poder proceder a editar y luego grabar los cambios realizados.



Para poder proceder con la eliminación de un registro hay que tomar en cuenta que no sean registros que tenga relación con otras tablas ya que esto causaría un mensaje de error ya que no se puede eliminar en cascada.

3.9. Implementación

En esta parte del desarrollo del Sistema el grupo investigador ha realizado la instalación de cada uno de los elementos que se necesitan así como: el servidor http Apache, el motor de base de datos MySQL, los programas que facilitaran el buen funcionamiento del sistema desarrollado.

Cabe indicar que la instalación de los efectos se debe realizar de una forma minuciosa ya que se necesita una configuración especial.

Para completar la parte de la implementación se adjunta a esta tesis en un CD en la que consta toda la información tanto teórica como el sistema terminado.

3.10. Pruebas

En esta parte, nosotros hemos realizado cada una de las pruebas pudiendo así comprobar que cada uno de los procesos que posee el sistema funciona correctamente, las pruebas que se realizaron fueron de tres tipos como son: primero se realizó una prueba unitaria la cual es la que verifica cada uno de las aplicaciones de forma individual comprobando que funcione lo más mínimo, la segunda parte fue que las aplicaciones trabajen en conjunto correctamente y verificamos que los resultados sean los esperados, y por último se procedió a realizar la prueba en el Internet y procedimos a comprobar los procesos; ya que las dos anteriores pruebas se realizaron en la maquina en la que trabajamos.