ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DEL SISTEMA WEB DE RESERVACIONES DE CUPOS DE AVION PARA PETROECUADOR CON LA HERRAMIENTA COLDFUSION MX 8

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

SUSANA MARGOTH CANENCIA GONZÁLEZ

maggy_tutty@hotmail.com

DIRECTOR: ING. LUZ MARINA VINTIMILLA
marina.vintimila@epn.edu.ec

Quito, Octubre 2013

DECLARACIÓN

Yo, Susana Margoth Canencia González, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y normatividad institucional vigente.

_____

Susana Margoth Canencia González

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Susana Margoth Canencia González, bajo mi supervisión

Ing. Luz Marina Vintimilla
DIRECTORA DE PROYECTO

AGRADECIMIENTOS

Agradezco de todo corazón a mi Dios por haberme brindado su bendición y sabiduría para alcanzar con paciencia mis objetivos planteados, a mi madre Susana González que ha sido el pilar de mi familia, a mi padre Alberto Canencia, a mi esposo Henry Zurita, a mi hijo Sebastián Zurita, a mi prima Lucila Canencia y a mis hermanos que perseverantemente me brindaron su apoyo, sabiduría e ímpetu para cumplir mis objetivos.

También agradezco a mi tutora de tesis la Ingeniera Luz Marina Vintimilla , por ser mi guía en la realización de este proyecto.

Susana Canencía

DEDICATORIA

Dedico este proyecto principalmente a San Sebastián, a mi abuelita Elena Escorza, a mis padres Susana Gonzáles, Alberto Canencia a mi hijo Sebastián Zurita y mi esposo Henry Zurita, por ser la inspiración, apoyo y fuerza para llegar a ser una profesional.

Susana

CONTENIDO

DECLARACIÓN…….. I
CERTIFICACIÓN……. II
AGRADECIMIENTOS…… III
DEDICATORIA…… IV
CONTENIDO…… V

CAPÍTULO I 1
1. ANTECEDENTES DEL PROYECTO 1
1.1 ÁMBITO 1
1.2 PLANTEAMIENTO DEL PROBLEMA 1
1.3 FORMULACIÓN DEL PROBLEMA 2
1.4 SISTEMATIZACIÓN DEL PROBLEMA 2
1.5 OBJETIVOS DE LA INVESTIGACIÓN 3
1.5.1 Objetivo General 3
1.5.2 Objetivos Específicos 3
1.6 JUSTIFICACIÓN DE LA PROPUESTA 4
1.7 ALCANCE 4
CAPITULO II 5
2. ASPECTOS TEÓRICOS 5
2.1 INGENIERÍA WEB 5
2.1.1 Proceso de Ingeniería Web 5
2.1.2 Control de Garantía de Calidad 7
2.1.3 Control de Configuración 7
2.1.4 Gestión de Proceso 7
2.1.5 Diferencia con la Ingeniería de Software 8
2.2 ARQUITECTURA DE DESARROLLO WEB 8
2.2.1 El Servidor Web 8
2.2.2 El Navegador Web, Browser 10

2.2.3 Aplicaciones Multinivel 11
2.3 HERRAMIENTAS DE DESARROLLO 13
2.3.1 ColdFusion 13
2.3.2 ActionScript 17
2.4 HERRAMIENTAS DE DISEÑO WEB 19
2.4.1 Macromedia Dreamweaver 19
2.4.2 Macromedia Flash 19
2.4.3 Gestor de base de datos Oracle 20
CAPÍTULO III 22
3. ASPECTOS METODOLÓGICOS 22
3.1 PARADIGMA ESPIRAL PARA LA WEB 22
3.2 METODOLOGIA 24
3.2.1 Clasificación de metodologías 24
3.2.2 Metodología OOHDM (Método de Diseño de Hipermedia Orientado a Objetos). 25
3.2.3 Ventajas y Desventajas de OOHDM(Método de Diseño de Hipermedia Orientado a Objetos). 42
3.2.4 UML 44
CONCLUSIONES Y RECOMENDACIONES 46
PRESUPUESTO Y FUENTES DE FINANCIAMIENTO 48
REFERENCIA BIBLIOGRAFICAS 48
LIBROS 48
INTERNET 48
MANUAL TÉCNICO 51
ANALISÍS 52
OBTENCIÓN DE REQUERIMIENTOS 53
DIAGRAMA DE CASOS DE USOS 54
MODELO CONCEPTUAL 59
DIAGRAMA DE CLASES 60
DIAGRAMA DE OBJETOS 70
DISEÑO 72
MODELO DINÁMICO 73
DIAGRAMAS DE SECUENCIA 74
MODELO FUNCIONAL 78
DIAGRAMA DE ACTIVIDADES 79
DISEÑO NAVEGACIONAL 93
MODELO NAVEGACIONAL 90
MANUAL DE INSTALACIÓN 91
INSTALACIÓN DEL SERVIDOR ORACLE ENTERPRISE 11g: 92
INSTALACIÓN DEL SERVIDOR COLDFUSION 100
INSTALACION TOAD 104

CONFIGURACIÓN DE TOAD 108
CONFIGURACION COLDFUSION 111
CONFIGURACIÓN SITIO EN DREAMWEAVER. 113
MANUAL DEL USUARIO 116
1.-Reservaciones (Cupos Pasajeros).- 117
2.-Reservaciones (Genera Cupos).- 122
3.-Despacho (Despacho Pasajeros).- 125
5- Pasajero (Pasajero Días Vuelo).- 134
6.-Itienerarios (Cupos por empresa) 136
7-Itienerarios (Creación y Modificación de Vuelos) 136

CAPÍTULO I

1. ANTECEDENTES DEL PROYECTO

En este capítulo se explica las principales razones por las que se lleva a cabo este trabajo, describiendo los problemas y sus soluciones, dando a conocer los objetivos planteados y el alcance que se proyecta llegar mediante el desarrollo del proyecto.

1.1 ÁMBITO

La unidad de Aviación de la Empresa de Hidrocarburos del Ecuador Petroecuador se encuentra ubicada en el aeropuerto de Quito y es la que se encarga del servicio de transporte aéreo para el personal de la Empresa hacia y desde la región del Oriente Ecuatoriano. Depende funcionalmente de la Gerencia Administrativa de la Empresa de Hidrocarburos del Ecuador Petroecuador y sus clientes internos son todos los empleados que necesitan viajar en el transporte aéreo.

1.2 PLANTEAMIENTO DEL PROBLEMA

En la actualidad Petroecuador y sus filiales utilizan el Sistema de Reservación de Cupos para avión, en una arquitectura cliente-servidor y ha sido desarrollado en Visual Fox Pro 6.0, cuenta con un sistema de administración propio de bases de datos ubicado en Lago Agrio, que controla la información relacionada con las reservas de cupos para vuelos en Petroecuador, el ejecutable de la aplicación se encuentra en el servidor intranet de Petroecuador.

Los encargados del sistema de reservas en Lago Agrio generan los cupos para pasajeros mediante el procesamiento por lotes batch, solamente de la Filial Petroproducción (encargada del sistema).

Al momento el Sistema actual genera los siguientes problemas:

- Pérdida de tiempo, tanto para el usuario como para la Empresa.
- La información almacenada no puede ser manipulada de acuerdo a las necesidades de los usuarios y autoridades de la Empresa.
- El sistema actual no permite generar reportes organizados ni confiables.
- El acceso y transferencia de datos no es confiable.
- Al no contar con una base de datos confiable, la información de los cupos de vuelos no se encuentra debidamente organizados.
- Demoras en el acceso y búsquedas de información

Petroecuador ha decidido desarrollar todas sus aplicaciones en plataforma Web, considerando el constante crecimiento de aplicaciones y sistemas en ambientes Web, esto ha proporcionado novedosas ventajas y facilidades en el manejo de información a través de una red o intranet sin importar la plataforma o sistema operativo en que esté la aplicación y ha permitido la comunicación entre filiales, reducir costos y otros beneficios inherentes.

Por lo expuesto, se ha decidido desarrollar el Sistema de Reservación de Cupos en plataforma Web.

1.3 FORMULACIÓN DEL PROBLEMA

¿Usando tecnología informática y de comunicación, cómo apoyar la optimización de la gestión de reservación de cupos realizados por PETROECUADOR de modo que su validez, confiabilidad y seguridad de datos puedan manipularse de acuerdo a los requerimientos de la unidad de Aviación?

1.4 SISTEMATIZACIÓN DEL PROBLEMA

- ¿Qué herramientas de desarrollo y que hardware son los apropiados?

- ¿Tengo los datos suficientes y qué relación, protección y control tendrán?
- ¿Cuáles son los procesos involucrados en la gestión de reservación de cupos en Petroecuador?
- ¿Cómo se validará la autenticidad de los usuarios?
- ¿Cómo administrar la información de cupos de reservación del avión?
- ¿Cómo administrar la información de consultas e informes de cupos de reservación del avión?

1.5 OBJETIVOS DE LA INVESTIGACIÓN

1.5.1 Objetivo General

DESARROLLAR EL SISTEMA WEB DE RESERVACIONES DE CUPOS DE AVION PARA PETROECUADOR CON LA HERRAMIENTA COLDFUSION MX 8

1.5.2 Objetivos Específicos

1) Determinar las políticas manejadas por Petroecuador.
2) Configurar el plan de arquitectura web.
3) Definir perfiles de usuarios.
4) Establecer políticas de seguridad y transparencia en el manejo de la información de acuerdo a la funcionalidad de los usuarios
5) Construir un buen administrador de datos mediante la herramienta Toad.
6) Generar las reservaciones de cupos de avión mediante el procesamiento por lotes batch para todas las diferentes filiales.
7) Generar reportes que ayuden al usuario a tener constancia del control de reservación de cupos para los pasajeros.
8) Probar la funcionalidad del sistema migrado.

1.6 JUSTIFICACIÓN DE LA PROPUESTA

Con la implementación de este sistema Petroecuador podrá controlar el manejo de la información de las reservaciones de cupos de pasajeros, usando un sistema informático de tal manera que sus datos puedan ser administrados de forma segura, para así certificar la claridad en el manejo de la información. Al mismo tiempo facilita el acceso al usuario mediante claves personales independiente del lugar donde se encuentre, garantizando la integridad de los datos, ya que el principal objetivo de una aplicación Web está en la accesibilidad universal, y que no requieren de alguna instalación previa, eso facilita las actualizaciones del mismo y se las realiza exclusivamente en el servidor.

1.7 ALCANCE

El sistema tiene como finalidad desarrollar una aplicación WEB, para mejorar de forma eficiente, segura y actualizada la información de reserva de cupos para el avión de Petroecuador.

El proceso objeto de estudio cubre las siguientes partes:

- Permitir el ingreso de usuarios con sus respectivas restricciones de acceso al sistema.
- Administrar la información actualizada de reservación de cupos para el avión.
- Administrar la información de consultas y reportes por usuario.
- Generar reportes y estadísticas de toda la información registrada.

CAPITULO II

2. ASPECTOS TEÓRICOS

En estecapítulo se redacta y describe los conceptos teóricos utilizados para el desarrollo de sistemas.

2.1 INGENIERÍA WEB

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.[1]

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

Desde que esto empezó a suceder el Internet se volvió más que una diversión y empezó a ser tomado más en serio, ya que el aumento de publicaciones y de informaciones hizo que la Web se volviera como un desafío para los ingenieros del software, a raíz de esto se crearon enfoques disciplinados, sistemáticos y metodologías donde tuvieron en cuenta aspectos específicos de este nuevo medio.

2.1.1 Proceso de Ingeniería Web

Existen algunas actividades importantes y que forman parte de este proceso, estas son: formulación, planificación análisis, modelización o ingeniería, generación de páginas, test y evaluación del cliente.

---

[1]Ingeniería Web referido a concepto de web (http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_la_Web#cite_note-autogenerated1-0)

Formulación: permite a los usuarios y equipo identifica objetivos y establece el alcance de la primera entrega.

Planificación: genera a estimación del costo, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.

Análisis: especifica los requerimientos e identifica el contenido.

Modelización o Ingeniería: consta de dos secuencias paralelas de tareas.

El diseño y producción del contenido que forma parte de la aplicación.

El diseño de la arquitectura, navegación e interfaz de usuario.

Generación de Páginas: se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de las aplicaciones, las páginas.

Test: el Test busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc.

Evaluación del Cliente: El resultado es sometido a la evaluación del cliente.[2]

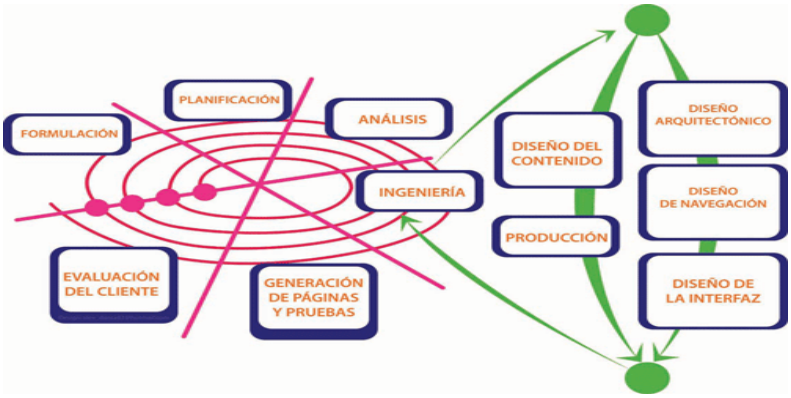


Fig. 2.1.1 Proceso de Ingeniería Web

[2]**Procesos de Ingeniera web referido de concepto:**
http://sg.com.mx/content/view/222

2.1.2 Control de Garantía de Calidad

Una de las tareas colaterales que forman parte del proceso es el Control y Garantía de la Calidad (CGC). Todas las actividades CGC de la ingeniería software tradicional como son: establecimiento y supervisión de estándares, revisiones técnicas formales, análisis, seguimiento y registro de informes, etc, son igualmente aplicables a la Ingeniería Web. Sin embargo, en la Web toman especial relevancia para valorar la calidad aspectos como: Usabilidad, Funcionabilidad, Fiabilidad, Seguridad, Eficiencia y Mantenibilidad [3]

2.1.3 Control de Configuración

Establecer mecanismos adecuados de control de la configuración para la Ingeniería Web es uno de los mayores desafíos a los que esta nueva disciplina se enfrenta. La Web tiene características únicas que demandan estrategias y herramientas nuevas..[4]

2.1.4 Gestión de Proceso

En un proceso tan rápido como es el proceso de Ingeniería Web, donde los tiempos de desarrollo y los ciclos de vida de los productos son tan cortos, ¿merece la pena el esfuerzo requerido por la gestión? La respuesta es que dada su complejidad es imprescindible.

Entre los aspectos que añaden dificultad a la gestión destacamos: - alto porcentaje de contratación a terceros, - el desarrollo incluye una gran variedad de personal técnico y no técnico trabajando en paralelo, - el equipo de desarrollo debe dominar aspectos tan variopintos como, software basado en componentes, redes, diseño de arquitectura y navegación, diseño gráfico y de interfaces, lenguajes y estándares en Internet, test de aplicaciones Web, etc., lo que hace que el proceso de búsqueda y contratación de personal sea arduo.

---

[3]**Control de calidad referido del libro**:
L. Olsina, G. Lafuente, G. Rossi. "Specifying Quality Characteristics and Attributes for Websites." Lecture Notes in Computer Science 2016 Springer 2001, pag. 266 – 278.
Control de configuración referido del libro:
[4]S. Dart, "Containing the Web Crisis Using Configuration Management," Proc. 1st ICSE Workshop on Web Engineering, ACM, Los Angeles, May 1999

2.1.5 Diferencia con la Ingeniería de Software

A modo de breve resumen enumeramos las siguientes diferencias:

- Confluencia de disciplinas: Sistemas de Información, Ingeniería Software y Diseño Gráfico que requiere equipos multidisciplinares y polivalentes.

- Ciclos de vida y tiempo de desarrollo muy cortos.

- Cambio continuo: Necesidad de soluciones que permitan flexibilidad y adaptación conforme el proyecto cambia. Requisitos fuertes de Seguridad, Rendimiento y Usabilidad.

2.2 ARQUITECTURA DE DESARROLLO WEB

La arquitectura Web es el conjunto de disciplinas que enseña el diseño además de coordinar el desarrollo de sitios web así también como la construcción de estructuras, organización de contenido, estableciendo contenido de sistemas de búsqueda, organización, y recuperación de cualquier información soportada en la web, con el fin de que el cliente directo cumpla sus objetivos, además de tener una experiencia optima al usar las aplicaciones web.[5]

2.2.1 El Servidor Web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta una página web o información de todo tipo de acuerdo a los comandos solicitados del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de

[5]**Arquitectura web concepto referido :**
http://www.slideshare.net/aimerodriguezrodriguez/arquitectura-de-paginas-web

aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor, *server side applications*, que vayamos a construir, ya que se ejecutarán en él.[6]

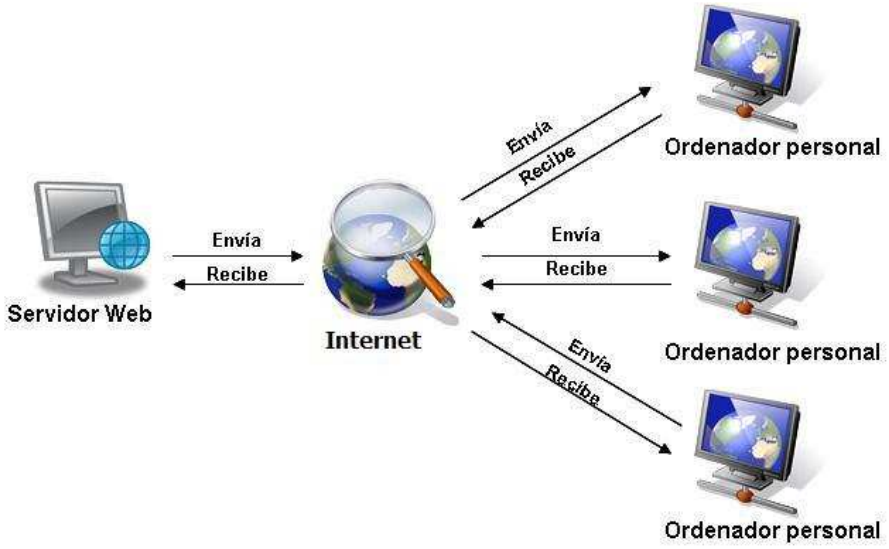


Fig. 2.2.1 Diagrama del servidor web

El 40% de los servidores en el mundo son Apache, después le sigue el servidor IIS de Microsoft con más del 20% y el resto se compone de otros servidores web como nginx, google, lighttpd entre otros.

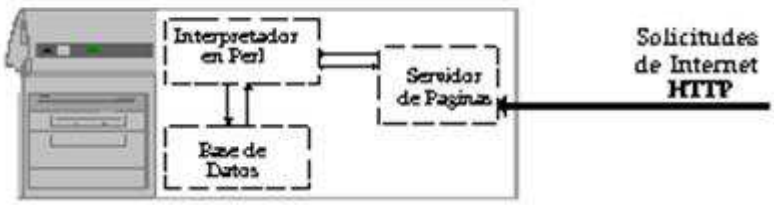


Fig. 2.2.1 Diagrama del servidor Apache

[6]**Servidor Web:**
http://www.monografias.com/trabajos75/servidores-web/servidores-web2.shtml

2.2.2 El Navegador Web, Browser

Es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de internet.

La comunicación entre el servidor web y el navegador se realiza mediante el protocoloHTTP, aunque la mayoría de los hojeadores soportan otros protocolos como FTP, Gopher, y HTTPS .

La función principal del navegador es descargar documentos HTML y mostrarlos en pantalla. En la actualidad, no solamente descargan este tipo de documentos sino que muestran con el documento sus imágenes, sonidos e incluso vídeos *streaming* en diferentes formatos y protocolos. Además, permiten almacenar la información en el disco o crear marcadores (*bookmarks*) de las páginas más visitadas.

Algunos de los navegadores web más populares se incluyen en lo que se denomina una Suite. Estas Suite disponen de varios programas integrados para leer noticias de Usenet y correo electrónico mediante los protocolos NNTP, IMAP y POP.

Los primeros navegadores web sólo soportaban una versión muy simple de HTML. El rápido desarrollo de los navegadores web propietarios condujo al desarrollo de dialectos no estándares de HTML y a problemas de interoperabilidad en la web. Los más modernos (como Google Chrome, Amaya, Mozilla, Netscape, Opera e Internet Explorer 9.0) soportan los estándares HTML y XHTML (comenzando con HTML 4.01, los cuales deberían visualizarse de la misma manera en todos ellos).

Los estándares web son un conjunto de recomendaciones dadas por el World Wide Web consortium W3C) y otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en la web. Su objetivo es crear

una web que trabaje mejor para todos, con sitios accesibles a más personas y que funcionen en cualquier dispositivo de acceso a Internet.[7]

2.2.3 Aplicaciones Multinivel

Es el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

La arquitectura tres-capas. Esta es diferente de la tradicional arquitectura cliente/servidor de manera que la aplicación en lugar de residir en el equipo del cliente, reside en la capa-media, es decir en el Servidor de Aplicación. Esto le permite a un usuario acceder a la aplicación desde el extremo del cliente, simplemente usando un navegador de red.

Las tres capas:

•**Capa de Presentación:** (o interfaz del usuario)Esta capa es la interfaz con el usuario y consiste de hardware como un PC o una estación de trabajo y un navegador de red. Esto puede constar de cualquier número de máquinas cliente.

•**Capa Lógica de Funcionalidad / Negocio:** Esta capa proporciona la funcionalidad al extremo de los usuarios y contienela lógica del negocio (aplicación). Proporciona el puente entre la primera y la tercera capa.

[7]**Navegador Web concepto referido**:
http://www.cavsi.com/preguntasrespuestas/que-es-un-navegador-web/

Dependiendo del número de puntos y del número de usuarios, cualquier número de servidores de aplicación puede colocarse en estacapa.

•**Capa de Datos**: Esta capa incluye la Base de Datos que contiene todos los datos de la organización y esta se encapsula desde los usuarios finales. Cualquier número de servidores de Base de Datos puede colocarse en esta capa, dependiendo del volumen de transacciones y dela cantidad de datos. [8]

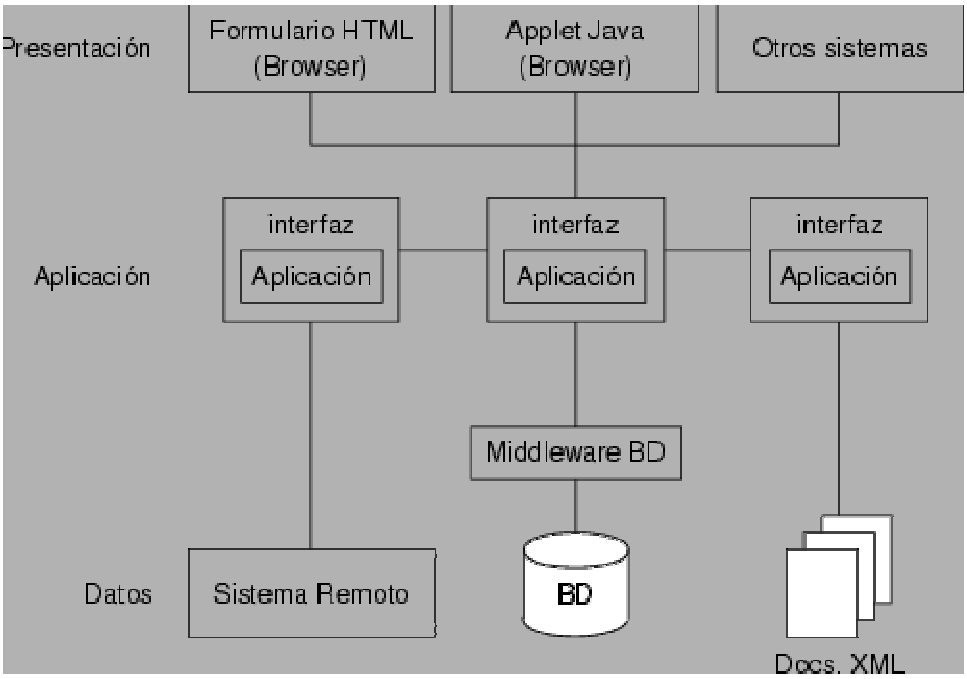


Fig 2.2.3 Arquitectura Multinivel.

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo

[8]**Aplicaciones Multinivel referido concepto**:
http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html

resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

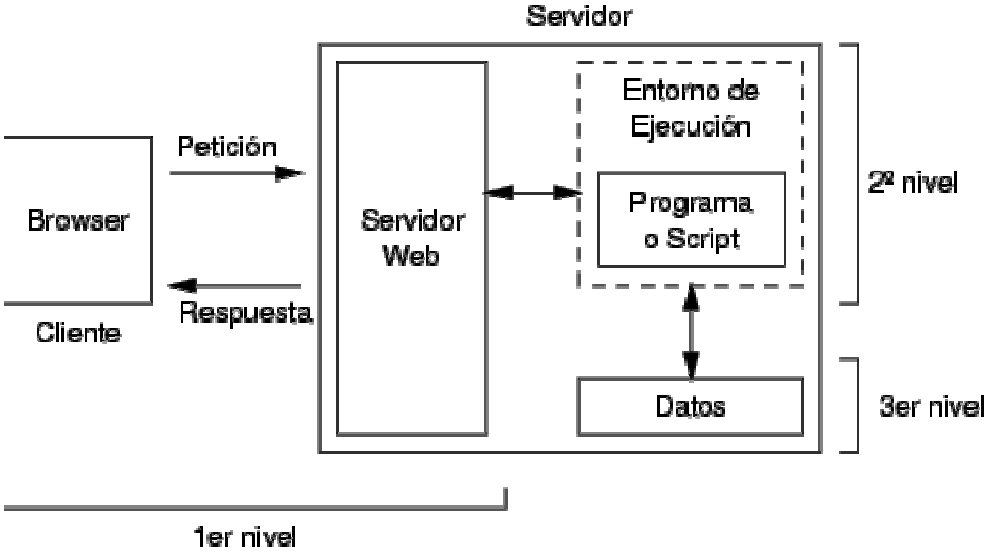


Fig. 2.2.3 Arquitectura Web de tres niveles.

2.3 HERRAMIENTAS DE DESARROLLO

La mayor parte de las páginas web que se encuentran en Internet se sustentan sobre bases de datos y sobre diversos lenguajes de desarrollo web que permiten realizar sitios web dinámicos y que generan las páginas según las peticiones de los usuarios, de forma que los creadores pueden actualizar con facilidad los contenidos del sitio y automatizar tareas.

Desde finales de los 90, diversos lenguajes de desarrollo web se prodigan en la mayoría de las páginas de Internet. El más utilizado, por su carácter gratuito y por su licencia libre, es PHP, aunque también presentan una buena cuota otros lenguajes como ASP.NET o Coldfusion.

2.3.1 ColdFusion

ColdFusiones uno de losentornos de programaciónmás fáciles de usaryle permite crearpotentesaplicaciones webdel lado del servidormuy rápido,conmuchomenos código queotras tecnologías comoASP, PHP, etc. ColdFusionintegramuchas de las tecnologíasde Internet, talescomo XML, Java,

servicios web, etc.

ColdFusionse compone de doselementos principales:

-Un servidorColdFusion(que se ejecutaen la parte superiorde su servidorweb),
-Plantillas deColdFusion(o archivos), que se escribenutilizandoColdFusionMarkup Language (CFML).

Eso esprobablemente una maneramuy simplistade decirlo, ya que hay muchoselementos que conformanun entorno deColdFusion,pero estosdos sonesenciales si sequiere construiruna aplicación deColdFusion/ sitio web.

2.3.1.1 Características y Funciones

ColdFusion centra su potencialidad en la confiabilidad y el control del manejo de datos. Reconoce la complejidad del manejo e interacción de escritos CGI, ofreciendo una potente seguridad, veloz carga de datos, procesamiento rápido de escritos CGI que posibilita el cumplimiento de tareas de entrada o devolución de datos.

Entre las funcionesde ColdFusion están:

Sirve a cualquier requisición de datos una vez cuente con la instalación y configuración de las fuentes de datos ODBC de 32-bits.

Detecta errores producidos por la mala configuración o por el registro completo de la bitácora del servidor SQL.

Funciona correctamente en una máquina remota. Se ejecuta sin problemas en el Microsoft Internet Information Server, aun teniendo gran cantidad de solicitudes. Gracias a ello brinda un correcto funcionamiento tanto en Internet como en Intranets.

Provee de ayuda para la configuración que permita generar páginas HTML en forma dinámica.

Crea estructuras condicionales dinámicamente para personalizar la solicitud de datos y el envío de los mismos hacia el cliente. Así mismo, diseña cadenas de datos para crear dinámicamente menús desplegables y para llenar listas de selección y listas de documentos.

2.3.1.2 Proceso de Coldfusion

Cuando unusuario navegaen un navegadora una páginaque termina conuna extensióncfm**.,** La solicitud se envíaa un servidorWeb**, que dirige** la solicitud alservidor de aplicacionesColdFusion(**CFA).**

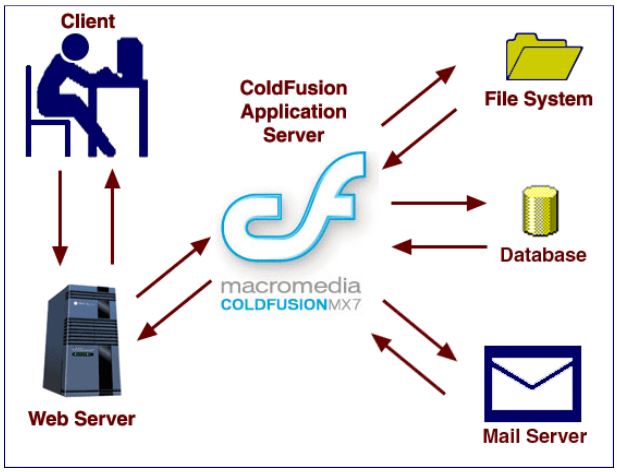


Fig 2.3.1.2. Arquitectura de ColdFusion para acceder bases de datos en el Web.

Como se muestraen eldiagrama anterior, elCFASprocesala página,la comunicación conlos sistemas de archivos, bases de datos y servidores decorreo electrónico, según sea necesario, y luegoentrega una páginawebal servidorwebpara regresaral navegador.

2.3.1.3 Componentes Coldfusion

ColdFusion posee los siguientes componentes los cuales lo hacen una herramienta muy eficaz para cualquier tipo de aplicación.

2.3.1.3.1 Tecnología del servidor

En el corazón de cada aplicación de ColdFusion hay un servidor ColdFusion, el cual, combina una arquitectura abierta y extensible que se integra fácilmente con sistemas existentes, así como también con aplicaciones built-in y servicios de infraestructura que ayudan a presentar la información de manera elegante y lograr un alto nivel de desempeño y confiabilidad.

2.3.1.3.2 Herramientas de desarrollo

Macromedia ofrece dos herramientas para el desarrollo. Para desarrolladores web, ColdFusion provee una tecnología poderosa de edición de código. Para diseñadores y desarrolladores web Dreamweaver UltraDev ofrece el mejor ambiente visual de desarrollo.

2.3.1.3.3 Ambiente de programación

ColdFusion soporta un poderoso lenguaje de scripting en el lado del servidor, ColdFusion Markup Language (CFML), que es extremadamente fácil de aprender y se integra limpiamente con todos los lenguajes y tecnologías web populares. ColdFusion trabaja con múltiples arquitecturas a través de la integración de COM, CORBA y EJB. También puede ser fácilmente extendido con nuevos componentes creados con Java Servlets, clases Java, o C/C++.

Permite construir aplicaciones web rápidamente.

Acelera el desarrollo con un conjunto poderoso de herramientas poderosas de diseño, programación, depuración e implantación.

Permite a los equipos de desarrollo colaborar de manera más efectiva compartiendo el mismo servidor y trabajando local o remotamente. Ensambla soluciones poderosas fácilmente.

Integración completa con la empresa, se conecta con todo el rango de sistemas backend, incluyendo bases de datos, servidores de mail, directorios, y aplicaciones empaquetadas. Se integra con tecnologías de empresa y de internet, incluyendo COM, CORBA, EJB, XML, C/C++ y Java.

Posee inteligencia de negocios. Permite crear planillas y reportes tabulares de calidad profesional.

Completa búsqueda de texto. Permite indexar fácilmente y buscar muchos tipos de contenido, incluyendo páginas web y documentos Microsoft Office 2000.

Arquitectura de alto desempeño. Asegura que las aplicaciones sean de implantación multiplataforma, entrega un avanzado thread pooling, caching de páginas built-in, consultas persistentes y pooling de conexiones a bases de datos.

Administración fácil. Simplifica la implantación y la administración del servidor a través de una poderosa consola de administración basada en web, reportes robustos de servidor y herramientas de análisis, además de integración con los sistemas de administración de la empresa.

Clustering de servidor. Provee balance de carga y recuperación automática para asegurar que las aplicaciones se mantengan consistentemente disponibles y se escala fácilmente para manejar tráfico creciente.[9]

2.3.2 ActionScript

[9] **Coldfusion definición referido concepto:**
http://es.wikipedia.org/wiki/Macromedia_ColdFusion

El ActionScript es el lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones Web animadas realizadas en el entorno Macromedia Flash, la tecnología de Adobe para añadir dinamismo al panorama Web. que ha utilizado Macromedia Flash desde sus comienzos, y que por supuesto, emplea Flash MX 2004. A grandes rasgos, podemos decir que el ActionScript nos permitirá realizar con Flash MX 2004 todo lo que nos propongamos, ya que nos da el control absoluto de todo lo que rodea a una película Flash.

2.3.2.1 Características de ActionScript

Es un lenguaje de script, esto quiere decir que no hará falta crear un programa completo para conseguir resultados, normalmente la aplicación de fragmentos de código ActionScript a los objetos existentes en nuestras películas nos permiten alcanzar nuestros objetivos.

En la mayor parte de las ocasiones, no será necesario "programar" realmente, Flash MX pone a nuestra disposición una impresionante colección de "funciones" ya implementadas que realizan lo que buscamos, bastará con colocarlas en el lugar adecuado.

Es el lenguaje de programación propio de Flash, tal y como el Lingo lo es de Macromedia Director, por ejemplo. El ActionScript está basado en la especificación ECMA-262, al igual que otros lenguajes como Javascript.[10]

---

[10] **AccionScript definición referido concepto:**
http://es.wikipedia.org/wiki/ActionScript

2.4 HERRAMIENTAS DE DISEÑO WEB

Son programas que nos ayuda al desarrollo,diseño, administración y programación del sistema web que vamos a realizar.

2.4.1 Macromedia Dreamweaver

Dreamweaver es una herramienta perfecta de desarrollo de páginas Web, es el número uno del mercado y permite a sus usuarios diseñar, crear y mantener de forma eficaz sitios y aplicaciones Web basados en normas.

Con Dreamweaver podemos usar un editor de diseño y código de máxima calidad en una sola herramienta y cuenta con un sistema de ayuda que hace más fácil su uso.
Podrás añadir rápidamente contenidos de vídeo en la Web. Arrastra y ubica vídeos de Flash en Dreamweaver para poner de forma rápida vídeos a sitios y aplicaciones Web.
Las versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos frameworks del lado servidor.[11]

2.4.2 Macromedia Flash

Es una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante un lenguaje de scripting llamado ActionScript. Flash es un estudio de animación que trabaja sobre "fotogramas" y está destinado a la producción y entrega de contenido interactivo para diferentes audiencias alrededor del mundo sin importar la plataforma.

Flash almacena sus archivos con varias extensiones. La extensión ".fla" contiene el programa fuente mientras que los archivos ".swf" contienen el gráfico que será mostrada en la web.

---

[11]**Dreamweaver referido :**
http://es.wikipedia.org/wiki/Adobe_Dreamweaver

Los archivos “.swf” son muy populares especialmente en la industria de la publicidad virtual (propaganda y avisos como banners, skyscrapers, etc.).

2.4.3 Gestor de base de datos Oracle

Es manejador de base de datos, es el conjunto de datos que proporciona la capacidad de almacenar y modificar Además es una suite de productos que ofrece una gran variedad de herramientas.

2.4.3.1 Características de Oracle

Desarrollado sobre Oracle Database, Oracle Content Database ha sido diseñada para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

2.4.3.2 Estructuras de Oracle

Una BD Oracle tiene una estructura física y una estructura lógica:

La estructura física se corresponde a los ficheros del sistema operativo.

La estructura lógica está formada por los tablespace y los objetos de un esquema de BD

2.4.3.2.1 Estructura lógica.

Se divide en unidades de almacenamiento lógicas: Tablespaces

Cada BD estará formada por uno o más tablespaces (al menos existe el tablespace SYSTEM catálogo del sistema), cada tablespace se corresponde con uno o más ficheros de datos.

Objetos: tablas, vistas, índices asociados a una tabla, clusters.

2.4.3.2.2 Estructura física

Una B.D. tiene uno o más ficheros de datos. Estos ficheros son de tamaño fijo y se establecen en el momento en que se crea la base de datos o en el momento en el que se crean tablespaces.

Los datos del fichero de datos son leídos cuando se necesitan y situados en una caché de memoria compartida para que el próximo acceso a los mismos sea más rápido. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

Soporte de transacciones.

- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

2.4.3.3 Arquitectura Oracle

La Arquitectura de Oracle tiene tres componentes básicos: las estructuras de memoria para almacenar los datos y el código ejecutable, los procesos que corren los sistemas de la base de datos y las tareas de cada usuario conectado a la base de datos y los archivos que sirven para el almacenamiento físico, en disco, de la información de la base de datos. [12]

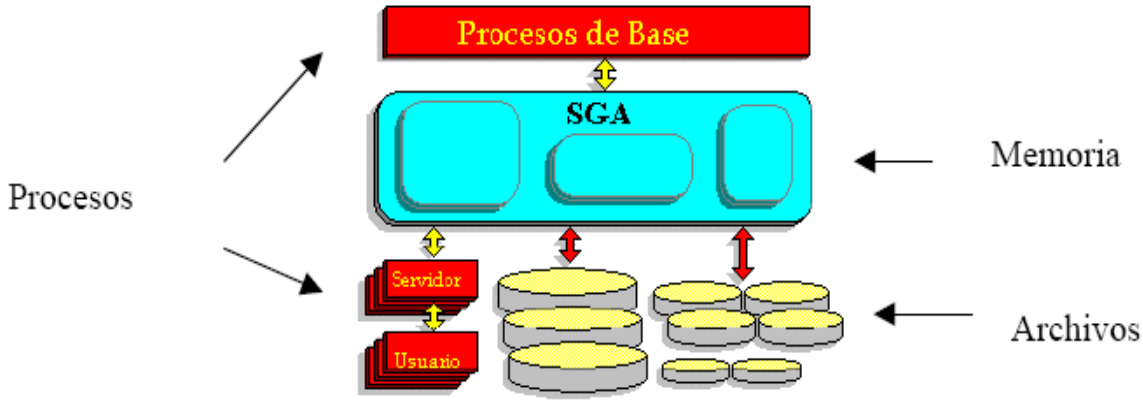


Fig.2 4.3.3 Arquitectura Oracle

[12]**Oracle referido:**
http://es.wikipedia.org/wiki/Oracle

CAPÍTULO III

3. ASPECTOS METODOLÓGICOS

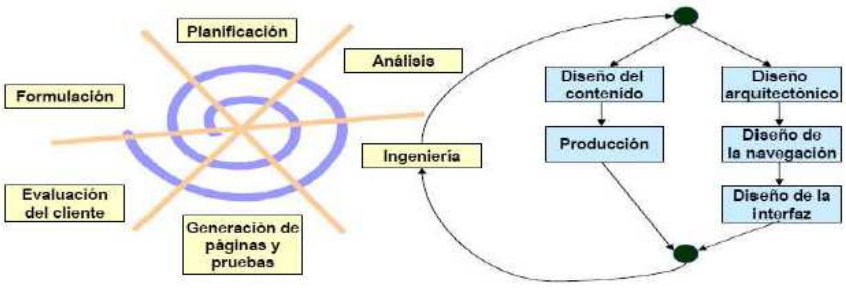
En el capítulo se describe el paradigma, la metodología y sus fases herramientas que fueron la guía de la ejecución de este proyecto.

3.1 PARADIGMA ESPIRAL PARA LA WEB

El modelo en espiral trata de desarrollar incrementalmente el proyecto, dividiéndolo en muchos subproyectos. Uno de los puntos más importantes del proceso es concentrarse primero en los aspectos más críticos del proyecto. La idea es definir e implementar las características más importantes primero, y con el conocimiento adquirido para hacerla, volver hacia atrás e implementar las características siguientes en pequeños sub proyectos.

El modelo en espiral orientado a la Web se divide en un número de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.

Modelo orientado al riesgo. Es el más versátil y flexible, pero también el más complejo. Cada vuelta de la espiral (ciclo) supone una refinación en el desarrollo.



3.1Paradigma espiral

A continuación se indican las etapas que presenta este Modelo:

Formulación. Se identifican las metas y objetivos.

Planificación. Estimación del coste global del proyecto, riesgos, etapas y sub-etapas.

Análisis. Establecimiento de los requisitos técnicos y de diseño (estéticos) e identificación de los elementos de contenido.

Ingeniería. Dos tareas paralelas.

Diseño del contenido y producción. Hechas por personal no técnico. Recopilación deinformación, medios audiovisuales, a integrar en la App. Diseño arquitectónico, de navegación y del interfaz: hecho por técnicos

Generación de páginas. Se adecua al diseño arquitectónico, de navegación y de interfaz, el contenido provisto para sacar las páginas HTML, XML, etc. Es en esta fase donde se integra la WebApp con el software intermedio (CORBA, DCOM, JavaBeans).

Pruebas. Se hace una navegación intensiva sobre la aplicación para descubrir errores, visualizarla en otros navegadores y ser consciente cuanto menos de las limitaciones y posibles “bugs”.

Evaluación del cliente. No es la última fase. Es una fase a ejecutar cada vez que se termina alguna de las anteriores. Los cambios se hacen efectivos por el flujo incremental del proceso.[13]

[13]Paradigma Espiral referido:
http://es.scribd.com/doc/47338452/PARADIGMA-DEL-MODELO-ESPIRAL

3.2 METODOLOGIA

Las Metodologías de Desarrollo de Software es una serie de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Una metodología está compuesta por:

- Cómo dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué restricciones deben aplicarse.
- Qué técnicas y herramientas se emplean.
- Cómo se controla y gestiona un proyecto.

3.2.1 Clasificación de metodologías

Las metodologías se clasifican de la siguiente forma:

- **Estructuradas**.
    - Orientadas a procesos
    - Orientadas a datos
    - Mixtas
- **No estructuradas**.
    - Orientadas a objetos
    - Sistemas de tiempo real

3.2.2Metodología OOHDM (Método de Diseño de Hipermedia Orientado a Objetos).

El modelo OOHDM u *Object Oriented Hypermedia Design Methodology,* para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa y es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web.

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cinco etapas:

3.2.2 Etapas de la metodología

3.2.2.1 Obtención de Requerimientos

La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema.

3.2.2.1.1 Diagramas de Casos de Uso

Estos permiten modelar el comportamiento de un sistema, el cual sirve para visualizar gráficamente cada uno de los componentes del sistema, la relación que existe entre ellos, quien hará uso de dicho comportamiento sin importar como estén implementados, esto nos ayuda a describir las funciones del sistema desde el punto de vista de los actores (usuarios, operadores, sistemas externos). En estos diagramas se especifica cada

función que realizará el sistema, obteniendo así el comportamiento total del sistema en desarrollo.

- **Actor**

Fig. *3.2.2.1.1* Actor

Un Actor es un rol que un usuario juega 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.

Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe de Local.

- **Casos de Uso**

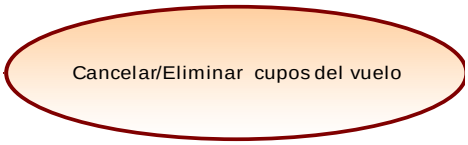


Fig. 3.2.2.1.1 Caso de Uso

Es una operación/tarea específica que se realiza tras una orden de algún

agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

- **Relaciones**
  - *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.

  - *Dependencia o Instanciación*

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

  - *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).

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda 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.

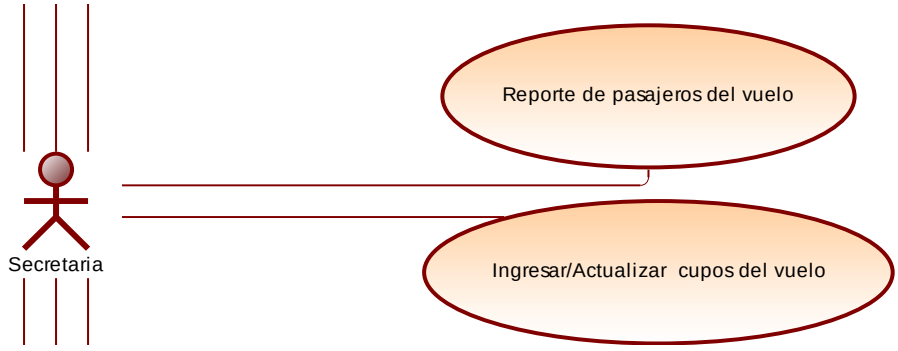


Fig. 3.2.2.1 Diagrama de Casos de Uso del Sistema Reservación de Cupos del Avión.

3.2.2.2 Modelo Conceptual

Durante esta actividad se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. En las aplicaciones hipermedia convencionales, cuyos componentes de hipermedia no son modificados durante la ejecución, se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intenten ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos.

En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real. Se usa notación similar a UML (Lenguaje de Modelado Unificado) y tarjetas de clases y relaciones similares a las tarjetas CRC (Clase Responsabilidad Colaboración). El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones que son usadas para construir enlaces.

Una clase describe un conjunto de objetos con características y comportamiento idéntico.

3.2.2.2.1 Diagrama de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos, son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro, se pueden distinguir principalmente dos elementos: clases y sus relaciones.

- **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.).

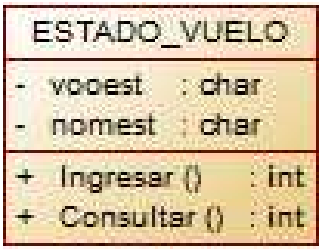


Fig. 3.2.2.2.1 Representación de una clase

Tanto los atributos como los métodos pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno que pueden ser:

- Public: es accesible desde todos lados, dentro y fuera de la clase.
- Private: solo será accesible dentro de la clase.
- Protected: no será accesible desde fuera de la clase, pero si de las que se deriven de esta.

- **RELACIONES**

Herencia/Generalización

Indica que una clase (clase derivada) hereda los métodos y atributos especificados por una clase (clase base), por lo cual una clase derivada además de tener sus propios métodos y atributos, podrá acceder a las características y atributos visibles de su clase base (public y protected)

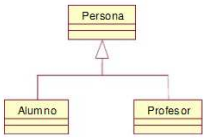


Fig. 3.2.2.2.1Relación Herencia

Agregación

La agregación es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye (el objeto base utiliza al incluido para su funcionamiento).

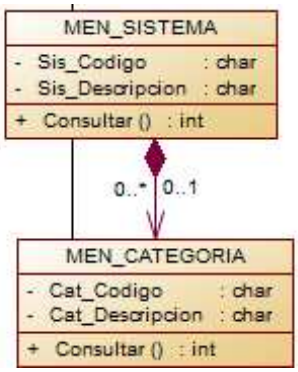


Fig. 3.2.2.2.1 Relación Agregación

Asociación ⟶

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

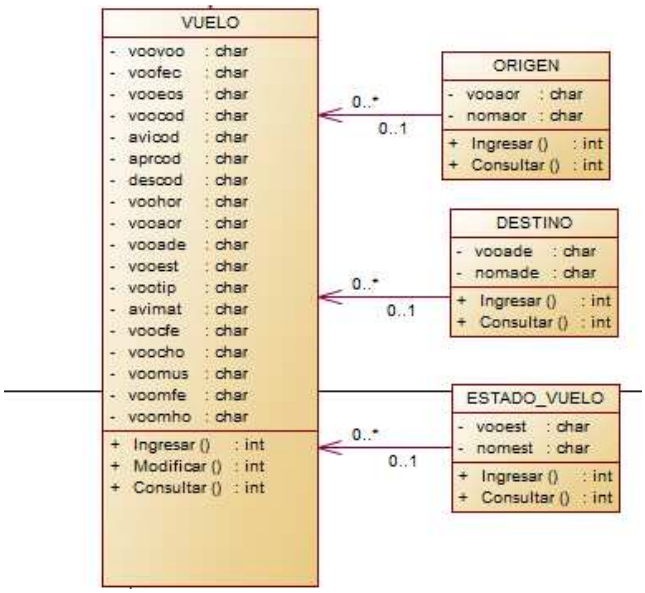


Fig. 3.2.2.2.1*Relación Asociación*

Dependencia o Instancia - - - - - - - - ⟶

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, como por ejemplo una aplicación grafica que instancia una ventana (la creación del Objeto Ventana está condicionado a la instanciación proveniente desde el objeto Aplicación)

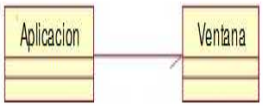


Fig. 3.2.2.2.1*Relación Dependencia*

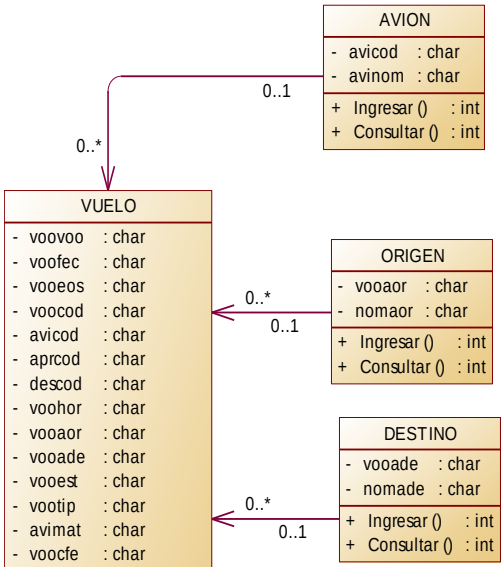


Fig. 3.2.2.2.2Diagrama de Clases de Sistema Reservación de Cupos del Avión.

3.2.2.2.2 Diagrama de Objetos

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre_del_objeto: nombre_de_la_clase. Puede representarse un objeto sin un nombre específico, entonces solo aparece el nombre de la clase.

Todas las relaciones son las mismas que en el diagrama de Clases por lo que no se las volverá a mencionar, la única diferencia que posee es que los atributos están instanciados como se muestra a continuación en la figura:

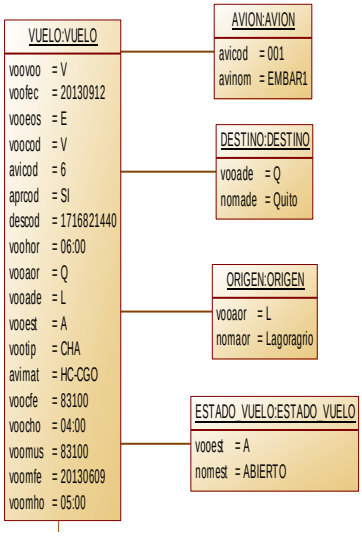


Fig. 3.2.2.2.2 Diagrama de objetosde Sistema Reservación de Cupos del Avión.

3.2.2.3 Diseño Navegacional

Para la Metodología OOHDM; la navegación es un paso clave en el diseño de aplicaciones web. Luego de obtener el esquema conceptual se establece la tercera fase que se denomina Diseño Navegacional**.** Un Diseño Navegacional es construido como una vista sobre un diseño conceptual, en la que se ha de definir la estructura de navegación a través del hiperdocumento mediante la construcción de modelos navegacionales de acuerdo con los diferentes perfiles de usuarios. En sí depende de las necesidades del usuario, donde lo que se realizará será organizar la información para cumplir con dichos requerimientos.

El Diseño de Navegación se expresa en dos esquemas o modelos:

- Esquema de Clases Navegacionales
- Esquema de Contextos Navegacionales

Clases Navegacionales están un conjunto de tipos predefinidos como: nodos, enlaces y las estructuras de acceso, tales como índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

3.2.2.3.1 Diagrama de Actividades

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos.

- Estados de Actividad y Estados de Acción

La representación de ambos es un rectángulo con las puntas redondeadas, en cuyo interior se representa bien una actividad o bien una acción. La forma de expresar tanto una actividad como una acción, no queda impuesta por UML, se podría utilizar lenguaje natural, una especificación formal de expresiones, un metalenguaje, etc. La idea central es la siguiente: “Un estado que represente una acción es atómico, lo que significa que su ejecución se puede considerar instantánea y no puede ser interrumpida” En la Figura, podemos ver ejemplos de estados de acción

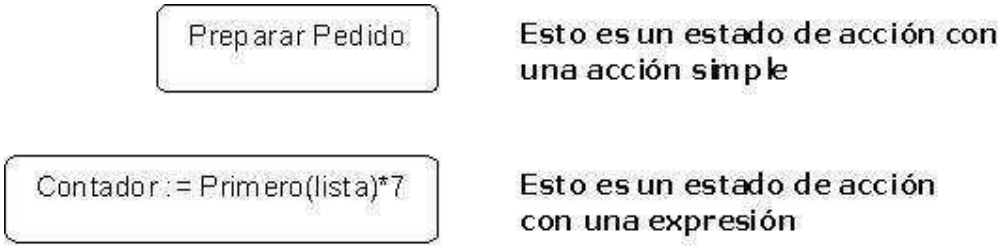


Fig. 3.2.2.3.1 Estado de Actividad y Acción

En cambio un estado de actividad, sí puede descomponerse en más sub-actividades representadas a través de otros diagramas de actividades. Además estos estados sí pueden ser interrumpidos y tardan un cierto tiempo en completarse. En los estados de actividad podemos encontrar otros elementos adicionales como son: acciones de entrada (entry) y de salida (exit) del estado en cuestión.

- **Transiciones**

Las transiciones reflejan el paso de un estado a otro, bien sea de actividad o de acción. Esta transición se produce como resultado de la finalización del estado del que parte el arco dirigido que marca la transición. Como todo flujo de control debe empezar y terminar en algún momento.

- **Bifurcaciones**

Un flujo de control no tiene porqué ser siempre secuencial, puede presentar caminos alternativos. Para poder representar dichos caminos alternativos o bifurcación se utilizará como símbolo el rombo. Dicha bifurcación tendrá una transición de entrada y dos o más de salida. En cada transición de salida se colocará una expresión booleana que será evaluada una vez al llegar a la bifurcación, las guardas de la bifurcación han de ser excluyentes y contemplar todos los casos ya que de otro modo la ejecución del flujo de control quedaría interrumpida. Para poder cubrir todas las posibilidades se puede utilizar la palabra ELSE, para indicar una transición obligada a un determinado estado cuando el resto de guardas han fallado.

- **División y Unión**

No sólo existe el flujo secuencial y la bifurcación, también hay algunos casos en los que se requieren tareas concurrentes. UML representa gráficamente el proceso de división, que representa la concurrencia, y el momento de la unión de nuevo al flujo de control secuencial, por una línea horizontal ancha. En la Figura podemos ver cómo se representa gráficamente.

- **Calles**

Cuando se modelan flujos de trabajo de organizaciones, es especialmente útil dividir los estados de actividades en grupos, cada grupo tiene un nombre concreto y se denominan calles. Cada calle representa a la parte de la organización responsable de las actividades que aparecen en esa calle.

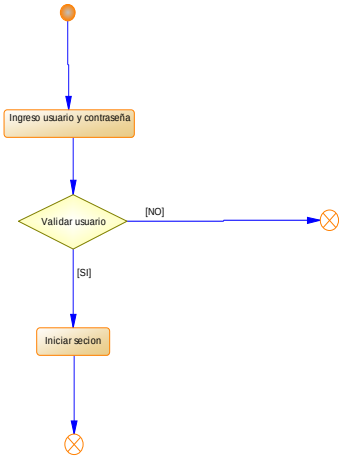


Fig. 3.2.2.3.1 Diagrama de Actividades de Sistema Reservación de Cupos del Avión.

3.2.2.3.2 Diagrama de Secuencia

Muestra la secuencia de mensajes entre objetos durante un escenario concreto. Cada objeto viene dado por una barra Vertical. El tiempo transcurre de arriba abajo. Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua.

- **Objeto/Actor**

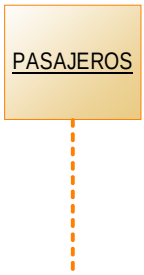


Fig. *3.2.2.3.2* Objeto

El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

- **Mensaje a otro Objeto**

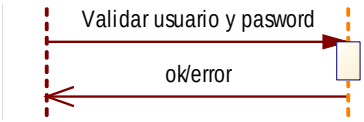


Fig. *3.2.2.3.2* Mensaje

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

- **Mensaje al mismo Objeto**

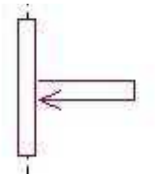


Fig. *3.2.2.3.2* Mensaje al mismo objeto

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

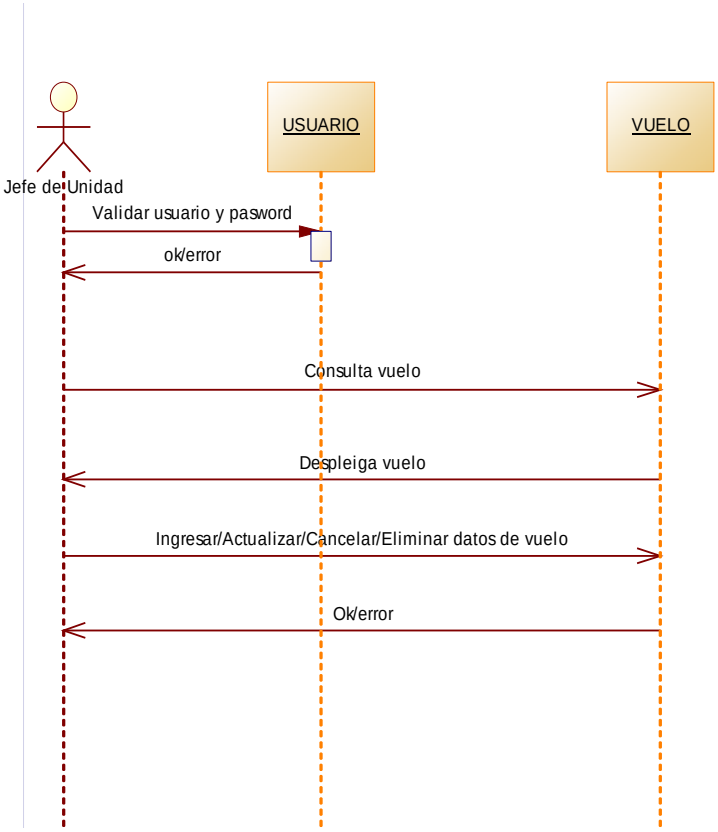


Fig. 3.2.2.3.2 Diagrama de Secuencia de Sistema de Reservas de Avión de Petroecuador

Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada. Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un contexto particular.

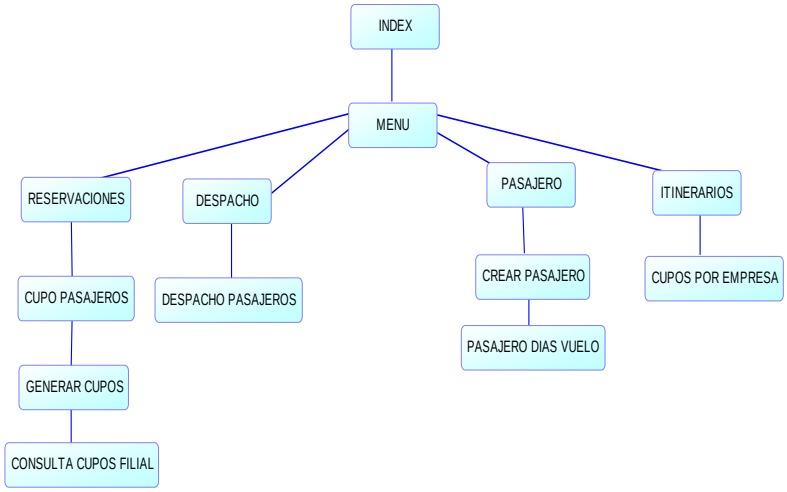


Fig. 3.2.2.3.2 .Diagrama Navegacional del Sistema de Reservas de cupos del avión

3.2.2.4 Diseño de Interfaz Abstracta

Esta es la cuarta fase de diseño de la metodología OOHDM, donde luego de definir las estructuras navegacionales, se deben especificar los aspectos de interfaz a través de un modelo, la estructura y el comportamiento de la interface del sistema hipermedia con el usuario. Este modelo es abstracto e independiente de la implementación. Y se basa en las Interfaces Gráficas de Usuario (IGUs).

En si se define la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, además qué transformaciones de la interfaz son pertinentes ycuándo es necesario realizarlas.

El modelo de la interface abstracta se expresa en tres tipos de diagramas los cuales se complementan entre sí.

- El primero es los diagramas de Vistas de Datos Abstractos (Abstract Data Views, ADVs), que OOHDM utiliza para modelar los aspectos estáticos de la interfaz de usuario, este incluyen una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue

establecida en el Diseño Navegacional. Este diagrama se compone de una serie de cajas (una caja es un ADV) que representan las diferentes clases de objetos que aparecerán ante el usuario.

- El segundo lugar se encuentra el Diagrama de Configuración, que representa los eventos externos (provocados por el usuario, como Clic del ratón o Doble clic) que maneja un ADV, los servicios que ofrece el ADV (como "visualización") y las relaciones estáticas entre las ADVs.
- Y el tercer diagrama de la metodología OODHM es el Diagrama de Estado que representa el comportamiento dinámico del sistema hipermedial mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (*oculto, desactivado, ampliado, reducido, normal, etc.)* y los eventos que originan los cambios de estado.

El Diseño de Interfaz Abstracta, simplifica las modificaciones de “percepción” y permite múltiples vistas de un mismo esquema.

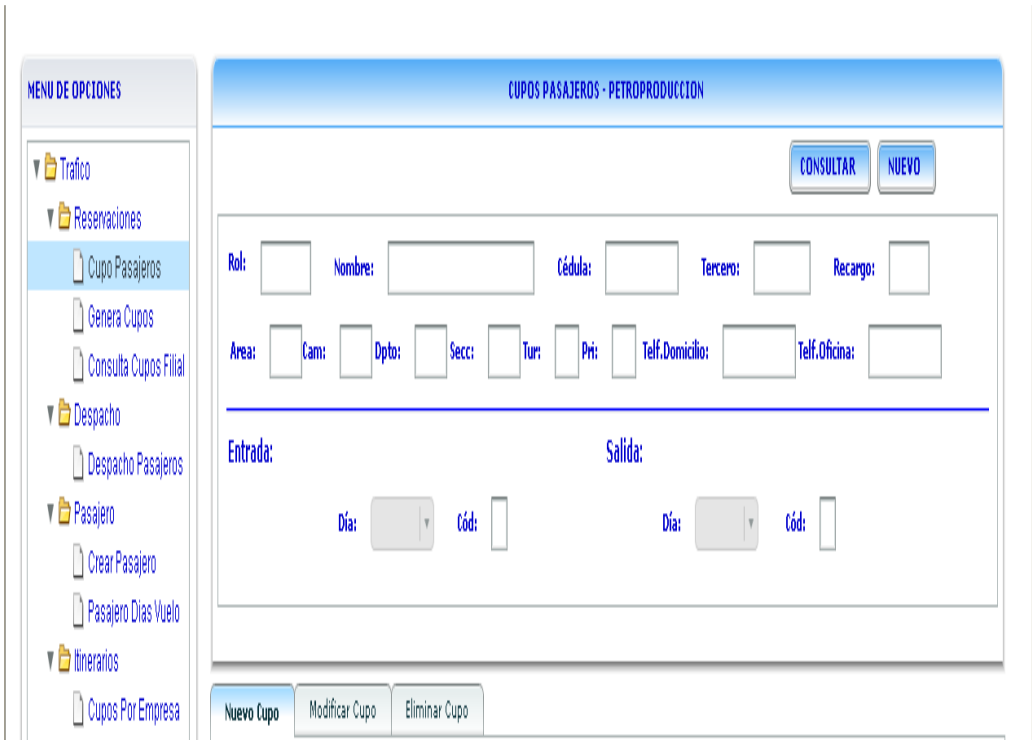


Fig.3.2.2.4 Diagrama de Interfaz de Sistema de Reservas de Cupos de Avión

3.2.2.5 Implementación

En esta fase, el diseñador debe implementar su diseño. Hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación.

Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada.[14]

3.2.2.6 Pruebas de sistema

La Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes´áreas en que se presenta, ha hecho que las personas tiendan a realizar todas sus actividades por esta vía. El incremento de la complejidad de este tipo de sistemas incrementa a su vez la necesidad de asegurar su calidad. La fase de pruebas del sistema ayuda a asegurar la calidad del software.

Las pruebas del sistema tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas, comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema.

---

[14]Modelo OOHDM referido:
http://pintojairo.wordpress.com/category/principal/metodologia-oohdm/
http://www2.rhernando.net/modules/tutorials/doc/ing/met_soft.html

Se prueba el sistema comprobando su funcionalidad y atributos de calidad. El sistema es probado en un ambiente lo más parecido posible al ambiente operacional.

3.2.3 Ventajas y Desventajas de OOHDM(Método de Diseño de Hipermedia Orientado a Objetos).

3.1.3.1 Ventajas

- OOHDM es una de las metodologías que más aceptación ha tenido, y sigue teniendo, en el desarrollo de aplicaciones multimedia. Actualmente está sirviendo como base para el desarrollo de nuevos sistemas de información web.
- OOHDM es una propuesta basada en el diseño, que ofrece una serie de ideas que han sido asumidas por bastantes propuestas y que han dado muy buenos resultados. La primera de ellas es que hace una separación clara entre lo conceptual, lo navegacional y lo visual.
- OOHDM hace uso también de la orientación a objetos y de un diagrama tan estandarizado como el de clases, para representar el aspecto de la navegación a través de las clases navegacionales: índices, enlaces y nodos.
- OOHDM nos ofrece una serie de ideas muy adecuadas a la hora de plantear una metodología de desarrollo que tenga en cuenta la navegación y la interfaz.
- OOHDM propone un modelo para representar a las aplicaciones multimedia, a su vez propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo.
- OOHDM toma como partida el modelo de clases que se obtiene en el análisis del Proceso Unificado de UML. A este modelo lo denomina modelo conceptual.
- OOHDM añade características que permitan incorporar a esta representación del sistema todos los aspectos propios de aplicaciones

multimedia.

- OOHDM no propone un modelo enriquecido para el dominio de la aplicación, por lo que deja libre al diseñador para elegir el modelo de especificación del dominio.
- OOHMD en las etapas de análisis y diseño, el usuario es considerado un integrante fundamental en la validación del producto obtenido. Esta interacción ayuda al desarrollador a entender y lograr en cada etapa lo que el usuario realmente necesita.
- OOHDM ofrece la posibilidad de crear estructuras de reuso, tales como los “esqueletos” o “frameworks”, cuyo principal objetivo es simplificar las tareas de diseño y disminuir su consumo de recursos.
- OOHDM utiliza una herramienta diagramática llamada UID, la cual es muy útil y sencilla de usar. Este instrumento es capaz de representar en forma precisa y con claridad los casos de uso obtenidos.

3.1.3.2 Desventajas

- Si bien es cierto los creadores de OOHDM señalan que la metodología fue creada principalmente para desarrollar aplicaciones hipermediales de gran extensión. Dicha orientación ha llevado a los creadores a desarrollar una serie de reglas y pasos (a veces bastante complicados de seguir) para realizar distintos mapeos entre un diagrama y otro, con el principal objetivo de simplificar y mecanizar las tareas de cada fase, este intento de mecanización puede traer como consecuencia el olvido de detalles fundamentales por parte del desarrollador.
- El diseño navegacional es un tanto tedioso, para resolverlo adecuadamente es necesario realizar una gran cantidad de diagramas que muchas veces entregan información similar a la entregada por los UIDs y las ADVs. Esta redundancia de información podría ser evitada graficando la información en un solo tipo de diagrama que sea capaz de reunir las capacidades de los UIDs, diagramas de contexto y ADVs.

- OOHDM presenta algunas deficiencias, deja fuera de su ámbito un aspecto esencial que es el tratamiento de la funcionalidad del sistema. El qué se puede hacer en el sistema y en qué momento de la navegación o de la interfaz se puede hacer, el cual lo deja como tarea de implementación.
- OOHDM no ofrece ningún mecanismo para trabajar con múltiples actores. Por ejemplo, imaginemos que la interfaz y la navegación de la aplicación varía sustancialmente dependiendo de quién se conecte a la aplicación. El diagrama navegacional, los contextos navegacionales y los ADVs resultarían muy complejos para representar esta variabilidad.
- OOHDM que no parece adecuada es la de los contextos navegacionales.

3.2.4 UML

Lenguaje Unificado de Modelado (LUM o **UML**, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema, para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.[15]

Los características de UML son las siguientes:

- UML no es una metodología.
- No cubre todas las necesidades de especificación de un proyecto software. Por ejemplo, No define los documentos textuales.
- UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

[15]UML referido :
https://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

La administración de perfiles de usuario para la seguridad y el acceso a la información, permitió dar un rastreo a la información modificada en la base de datos, dándonos una seguridad a la información.

La implementación del Sitio Web diseñado permitió a la Empresa tener mayor facilidad para el manejo de cupos del avión ya que facilito la actualización y manejo de la información en los tres aeropuertos (Quito, El Coca, Lago Agrio) de forma inmediata.

Oracle es uno de los sistemas de bases de datos más completos por su estabilidad, escalabilidad, soporte de transacciones, multiplataforma y sus mecanismos de seguridad para el manejo de la información.

La Metodología OOHDM permite una apropiada documentación del proceso, llevando un control en las pruebas de funcionamiento, lo cual incide en la calidad del producto de Software positivamente.

ColdFusion es un servidor de aplicaciones Web que permite desarrollar páginas en un tiempo muy corto, mediante un administrador de base de datos propio, de esta forma se las puede invocar desde cualquier página con extensión .cfm o .cfc y realizar consultas directamente a las bases de datos.

RECOMENDACIONES

Hay que tener cuidado en la Administración de la base de datos al tratar de eliminar alguna tabla y estar seguro de lo que se está haciendo en un área individual de la base de datos.

Los respaldos de datos se tendrán que realizar por lo menos una vez por semana o a su vez de acuerdo a las políticas de respaldo de datos que tenga compañía.

Antes de empezar a cargar la información al Sistema se recomienda disponer de un delegado por parte de la compañía para que haga una revisión de los datos a ingresar para no tener a futuro inconvenientes y estar haciendo muchas modificaciones, esto evitará pérdida de tiempo y recursos lo cual perjudica tanto la compañía como al desarrollador del proyecto.

El manejo de los tamaños de la multimedia utilizada en el proyecto es muy importante ya que al momento de estar vía internet el sistema puede tardar mucho tiempo, tiempo que los usuarios finales no disponen; se recomienda utilizar imágenes editadas que no ocupen mucho espacio en tamaño, pero tampoco tengan mala calidad.

Utilizar de forma continua el manual de usuarios; para un mejor uso y manejo de todas las herramientas de sistema.

Se recomienda un proceso de capacitación del usuario que va estar encargado de su funcionamiento.

.

PRESUPUESTO Y FUENTES DE FINANCIAMIENTO

	Costo por hora ($)	Horas	Total ($)
Recursos Humanos			
Diseñador	15	180	2700
Programador	15	360	5400
Software			
DreamWeaver	-	-	900
ColdFusion MX 8	-	-	6000
Oracle	-	-	70000
Hardware			
PC de Escritorio	-	-	900
Otros			100
Insumos	-	-	200
TOTAL			86200

REFERENCIA BIBLIOGRAFICAS

LIBROS

PRESSMAN, Roger S., *Iingeniería de Software Un Enfoque Práctico quinta edición.***(2002) págs. 344,345.**

GERMAN, D. *The Object Oriented Hypermedia Design Method.* **(2003).**

MARTIN, JAMES ODELL , *Análisis y diseño Orientado a objetos*

ROB BROOKS - BILSON, *Programming ColdFusion MX 2nd Edition***, august 2003.**

WILMINGTON, DELAWARE/ ADDISON-WESLEY, *Ingeniería de software: práctica de la programación***(1993).**

INTERNET

Cesar Ugaz Galarreta*IngenieríaWeb*

http://www.monografias.com/trabajos-ppt/ingenieria-web/ingenieria-web.shtml

Jet Aimee*Arquitectura web***on *Sep 04, 2012***
http://www.slideshare.net/aimerodriguezrodriguez/arquitectura-de-paginas-web

Perla Azucena Arredondo Morales*Servidores Web* **2013**

http://www.monografias.com/trabajos75/servidores-web/servidores-web2.shtml

CAVSI *Navegadores web*

http://www.cavsi.com/preguntasrespuestas/que-es-un-navegador-web/

JuanG83**Navegadores Web 2013**

http://es.wikipedia.org/wiki/Navegador_web

Juan Manuel Lemus*Aplicaciones Multinivel***2013**

http://www.maestrosdelweb.com/editorial/desarrollo-multinivel-para-aplicaciones-basadas-en-el-web/

María Jesús Lamarca Lapuente*Método OOHDM***2013**

http://www.hipertexto.info/documentos/oohdm.htm

Jose Luis Avalos*Diagrama de Clases***2013**

http://es.scribd.com/doc/31096724/Diagrama-de-Clases-en-UML

DaniSantia*Diagrama de objetos*on *Mar 13, 2013*

http://www.slideshare.net/DaniSantia/diagrama-de-objetos-y-de-componentes

Allforrous*Diagrama de Casos de Uso*

http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso

gmoreno*UML*

http://www.monografias.com/trabajos5/insof/insof.shtml

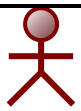
MANUAL TÉCNICO

ANALISÍS

OBTENCIÓN DE REQUERIMIENTOS

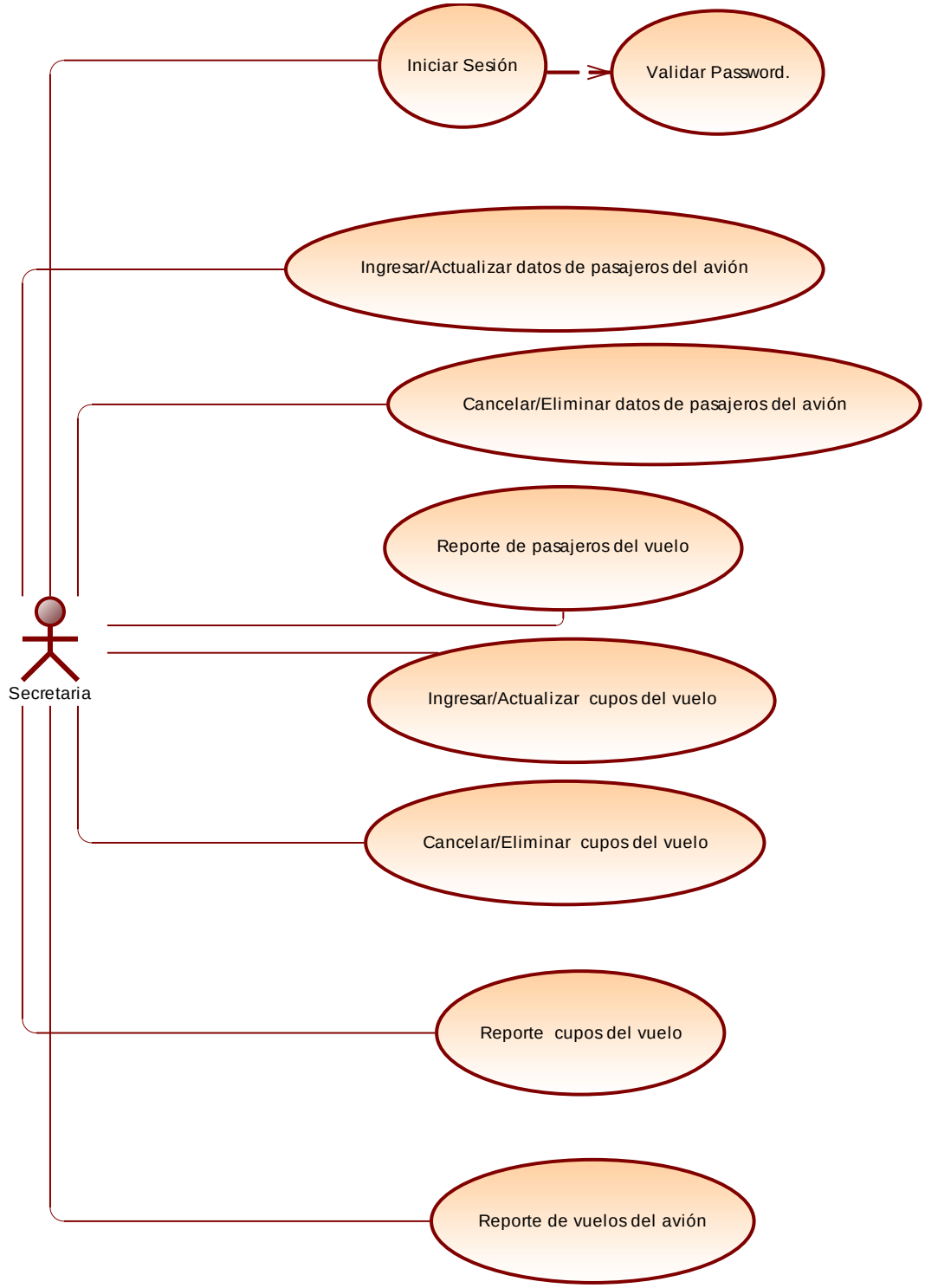
DIAGRAMA DE CASOS DE USOS

IDENTIFICACION DE ACTORES

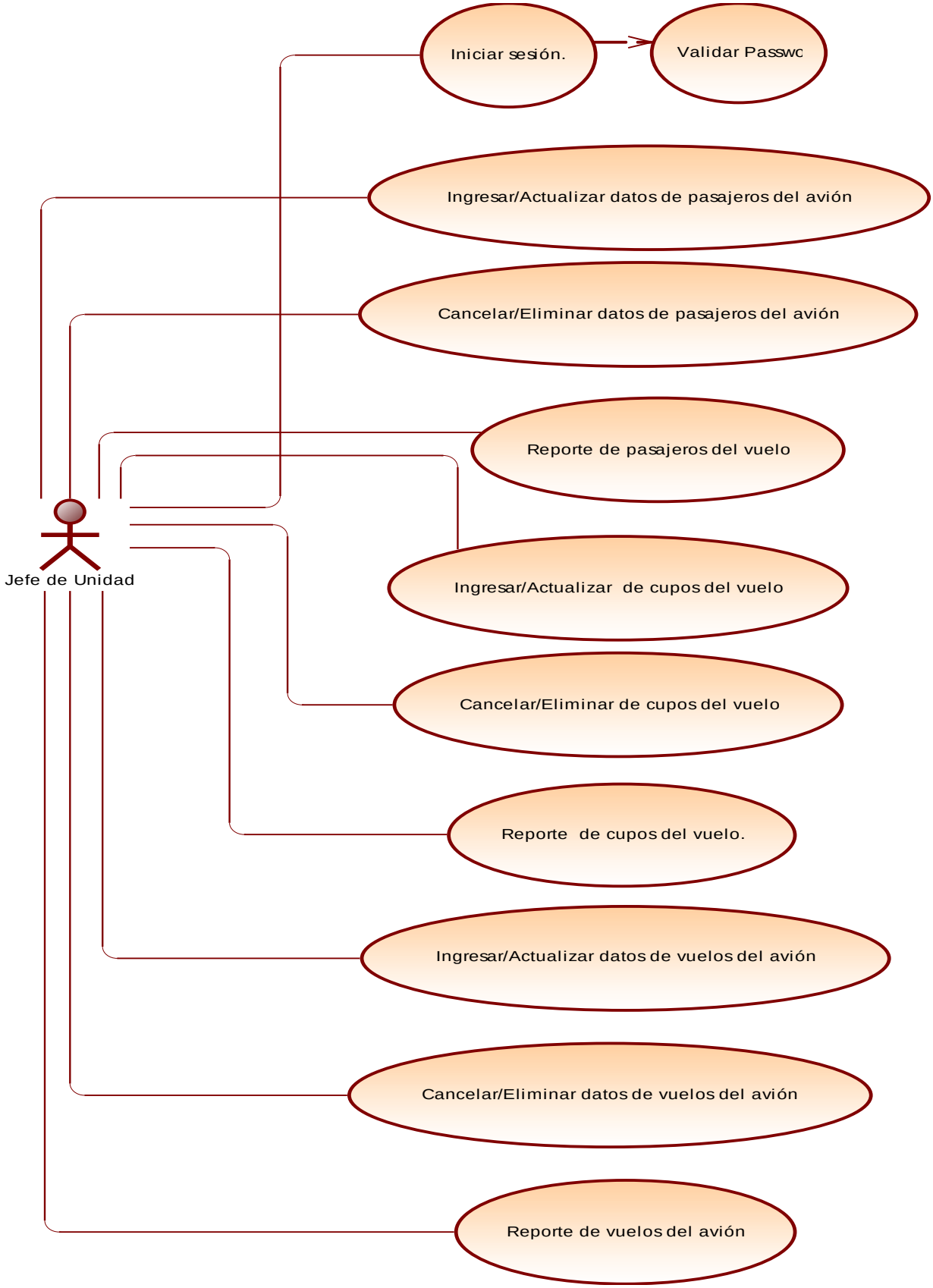
ACTORES	FUNCION
 Secretaria	Iniciar Sesión. Validar Password. Ingresar/Actualizar datos de pasajeros del avión. Cancelar/Eliminar datos de pasajeros del avión. Reporte de pasajeros del vuelo. Ingresar/Actualizar cupos del vuelo. Cancelar/Eliminar cupos del vuelo. Reporte cupos del vuelo. Reporte de vuelos del avión.
 Jefe de Unidad	Iniciar sesión. Validar Password Ingresar/Actualizar datos de pasajeros del avión. Cancelar/Eliminar datos de pasajeros del avión. Reporte de pasajeros del vuelo. Ingresar/Actualizar de cupos del vuelo. Cancelar/Eliminar de cupos del vuelo. Reporte de cupos del vuelo. Ingresar/Actualizar datos de vuelos del avión. Cancelar/Eliminar datos de vuelos del avión. Reporte de vuelos del avión.
 Despachador	Iniciar sesión. Validar Password. Cambio de cupos del vuelo. Consultar vuelos del avión. Reporte de vuelos del avión Consultar pasajeros del avión Reporte pasajeros del avión. Despacho de pasajeros. Consulta vuelos abiertos. Consulta vuelos cerrados. Reporte de pasajeros despachados.

DIAGRAMAS

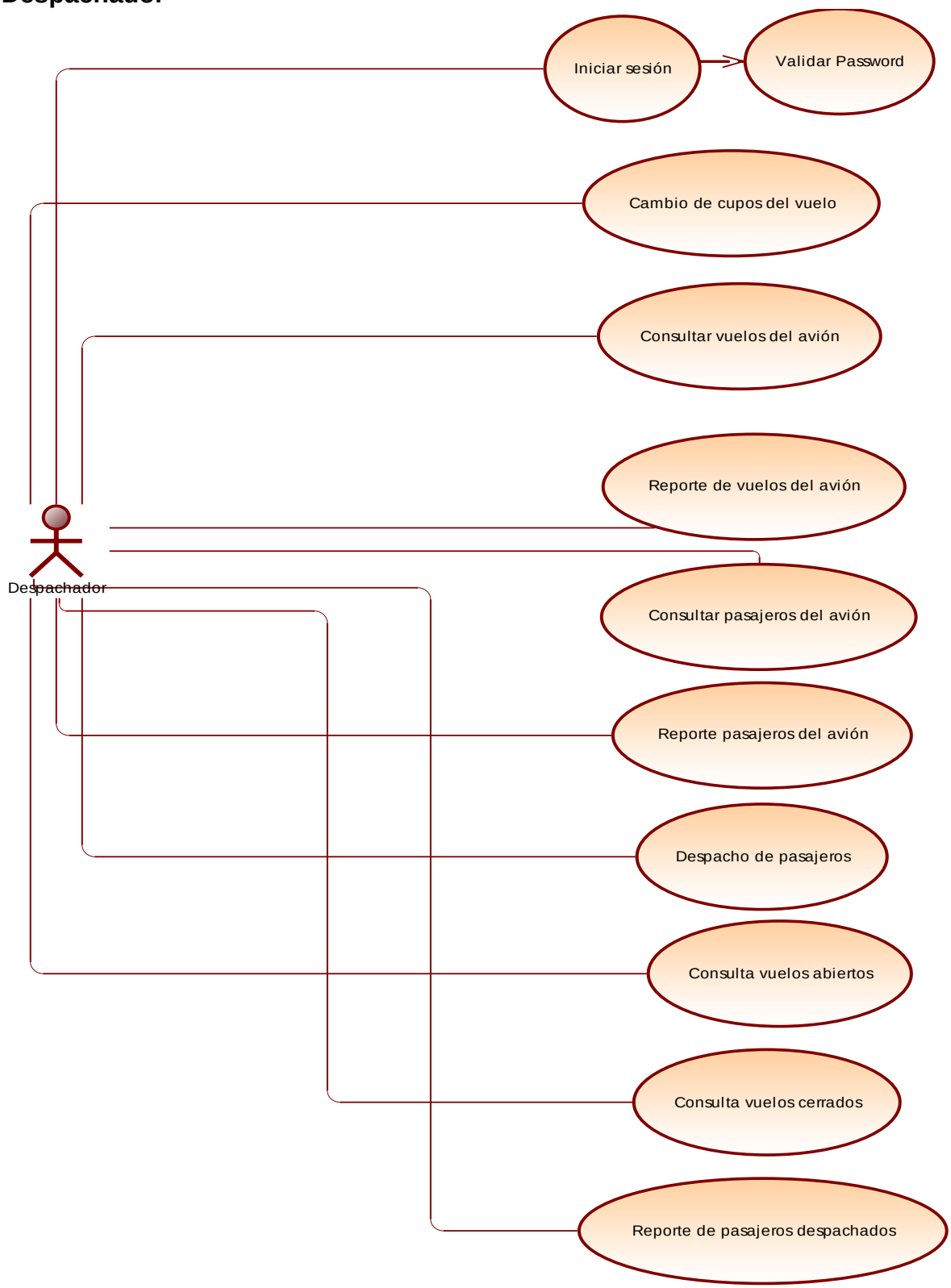
Secretaria



Jefe de Unidad

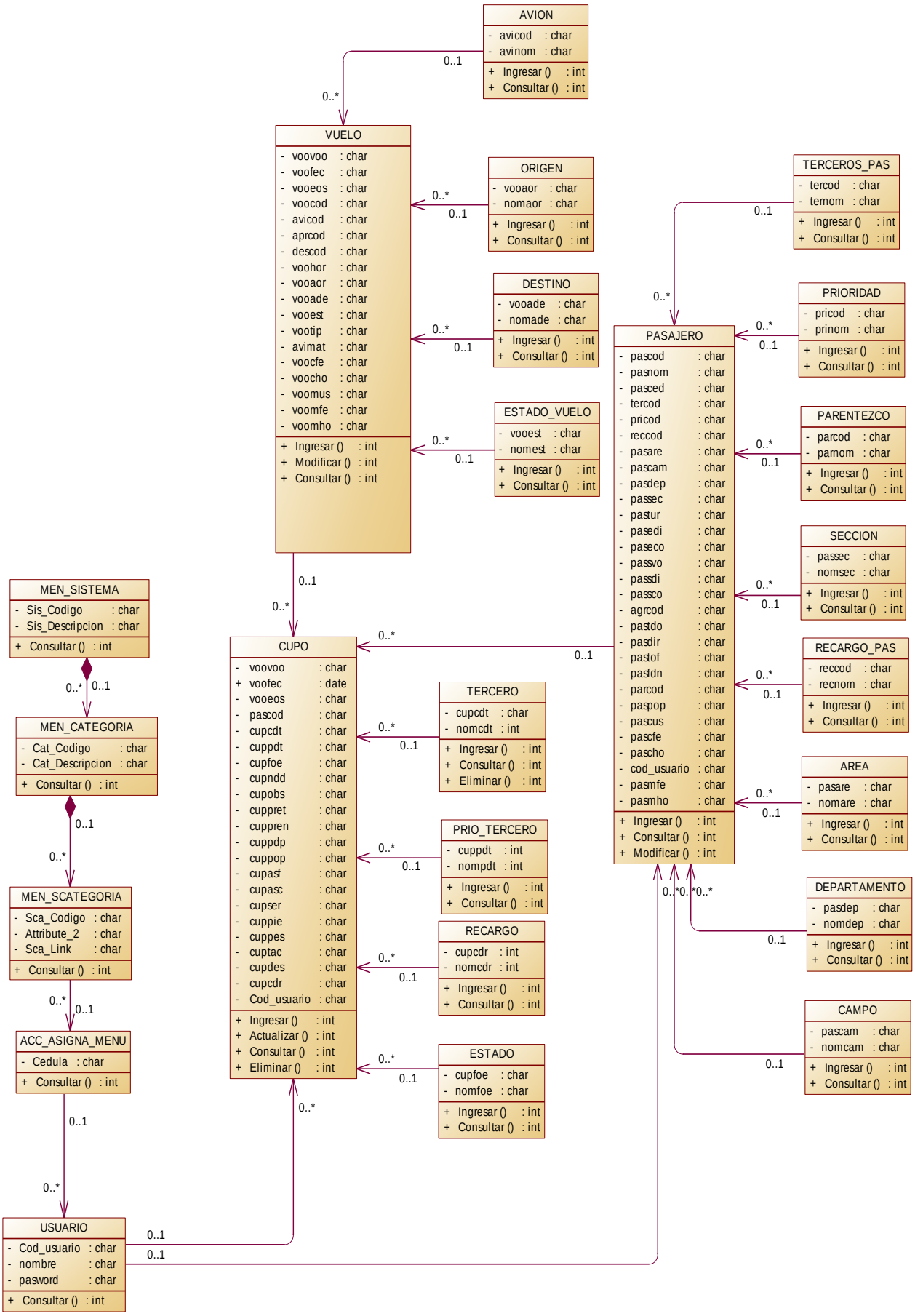


Despachador



MODELO CONCEPTUAL

DIAGRAMA DE CLASES


AVION
- avicod : char
- avinom : char
+ Ingresar () : int
+ Consultar () : int
VUELO
- voovoo : char
- voofec : char
- vooeos : char
- voocod : char
- avicod : char
- aprcod : char
- descod : char
- voohor : char
- vooaor : char
- vooade : char
- vooest : char
- vootip : char
- avimat : char
- voocfe : char
- voocho : char
- voomus : char
- voomfe : char
- voomho : char
+ Ingresar () : int
+ Modificar () : int
+ Consultar () : int
ORIGEN
- vooaor : char
- nomaor : char
+ Ingresar () : int
+ Consultar () : int
DESTINO
- vooade : char
- nomade : char
+ Ingresar () : int
+ Consultar () : int
ESTADO_VUELO
- vooest : char
- nomest : char
+ Ingresar () : int
+ Consultar () : int
TERCEROS_PAS
- tercod : char
- ternom : char
+ Ingresar () : int
+ Consultar () : int
PRIORIDAD
- pricod : char
- prinom : char
+ Ingresar () : int
+ Consultar () : int
PASAJERO
- pascod : char
- pasnom : char
- pasced : char
- tercod : char
- pricod : char
- reccod : char
- pasare : char
- pascam : char
- pasdep : char
- passec : char
- pastur : char
- pasedi : char
- paseco : char
- passvo : char
- passdi : char
- passco : char
- agrcod : char
- pastdo : char
- pasdir : char
- pastof : char
- pasfdn : char
- parcod : char
- paspop : char
- pascus : char
- pascfe : char
- pascho : char
- cod_usuario : char
- pasmfe : char
- pasmho : char
+ Ingresar () : int
+ Consultar () : int
+ Modificar () : int
PARENTEZCO
- parcod : char
- pamom : char
+ Ingresar () : int
+ Consultar () : int
SECCION
- passec : char
- nomsec : char
+ Ingresar () : int
+ Consultar () : int
RECARGO_PAS
- reccod : char
- recnom : char
+ Ingresar () : int
+ Consultar () : int
AREA
- pasare : char
- nomare : char
+ Ingresar () : int
+ Consultar () : int
DEPARTAMENTO
- pasdep : char
- nomdep : char
+ Ingresar () : int
+ Consultar () : int
CAMPO
- pascam : char
- nomcam : char
+ Ingresar () : int
+ Consultar () : int
MEN_SISTEMA
- Sis_Codigo : char
- Sis_Descripcion : char
+ Consultar () : int
MEN_CATEGORIA
- Cat_Codigo : char
- Cat_Descripcion : char
+ Consultar () : int
MEN_SCATEGORIA
- Sca_Codigo : char
- Attribute_2 : char
- Sca_Link : char
+ Consultar () : int
ACC_ASIGNA_MENU
- Cedula : char
+ Consultar () : int
USUARIO
- Cod_usuario : char
- nombre : char
- pasword : char
+ Consultar () : int
CUPO
- voovoo : char
+ voofec : date
- vooeos : char
- pascod : char
- cupcdt : char
- cuppdt : char
- cupfoe : char
- cupndd : char
- cupobs : char
- cuppret : char
- cuppren : char
- cuppdp : char
- cuppop : char
- cupasf : char
- cupasc : char
- cupser : char
- cuppie : char
- cuppes : char
- cuptac : char
- cupdes : char
- cupcdr : char
- Cod_usuario : char
+ Ingresar () : int
+ Actualizar () : int
+ Consultar () : int
+ Eliminar () : int
TERCERO
- cupcdt : char
- nomcdt : int
+ Ingresar () : int
+ Consultar () : int
+ Eliminar () : int
PRIO_TERCERO
- cuppdt : int
- nompdt : int
+ Ingresar () : int
+ Consultar () : int
RECARGO
- cupcdr : int
- nomcdr : int
+ Ingresar () : int
+ Consultar () : int
ESTADO
- cupfoe : char
- nomfoe : char
+ Ingresar () : int
+ Consultar () : int
0..1
0..*

DICCIONARIO DE CLASES

Nombre	Código	Descripción
CUPO	Cupo	Almacena información sobre cupos del avión
TERCERO	Tercero	Almacena información sobre nombre de filiales
PRIO_TERCERO	PrioTercero	Almacena toda la información sobre prioridad de vuelo.
RECARGO	Recargo	Almacena información a que filial se recarga del vuelo.
ESTADO	Estado	Almacena información sobre estado de cupo
PASAJERO	Pasajero	Almacena toda la información sobre el pasajero.
CAMPO	Campo	Almacena toda la información el campo de ubicación del pasajero
DEPARTAMENTO	Departamento	Almacena toda la información el departamento de ubicación del pasajero
AREA	Area	Almacena toda la información el área de ubicación del pasajero
RECARGO_PAS	RecargoPas	Almacena toda la información sobre a qué filial es el recargo.
SECCION	Seccion	Almacena información a que sección pertenece el pasajero.
PARENTEZCO	Parentezco	Almacena información sobre el parentesco con el empleado.
PRIORIDAD	Prioridad	Información sobre la prioridad de vuelo del pasajero.
TERCEROS_PAS	TercerosPas	Almacena toda la información sobre personas que no pertenecen a EP Petroecuador.
VUELO	Vuelo	Almacena toda la sobre los vuelos que realiza el avion
ESTADO_VUELO	EstadoVuelo	Almacena información sobre el estado del vuelo.
DESTINO	Destino	Almacena información sobre el destino del vuelo.
ORIGEN	Origen	Almacena información sobre el origen del vuelo.
AVION	Avion	Almacena información sobre el código del avion.
USUARIOS	USUARIOS	Almacena datos de los usuarios que administran el sistema.
MEN_SISTEMA	MEN_SISTEMA	Almacena información de código y nombre del sistema.
MEN_CATEGORIA	MEN_CATEGORIA	Almacena información referente

		a las categorías del menú de la aplicación (selecciones del menú).
MEN_SCATEGORIA	MEN_SCATEGORIA	Almacena información referente a las categorías del menú de la aplicación (sub-cesiones del menú del menú).

ATRIBUTOS DE CLASE

CUPO

Nombre	**Tipo de Dato**	**Descripción**
voovoo	char	Vuelo u otro
voofec	date	Fecha de vuelo
vooeos	char	Entrada o Salida de vuelo
pascod	char	Pasajero Código
cupcdt	char	Código de Terceros
cuppdt	char	Prioridad de Terceros
cupfoe	char	Cupo Fijo o En Espera
cupndd	char	Novedades de Despacho
cupobs	char	Observaciones de Despacho
cuppret	char	Referencia Tipo
cuppren	char	Número de Referencia
cuppdp	char	Prioridad de Pasajero
cuppop	char	Prioridad Operativa
cupasf	char	Asiento Fila
cupasc	char	Asiento Columna
cupser	char	Servicio Especial
cuppie	char	Piezas
cuppes	char	Peso
cuptac	char	Tac
cupdes	char	Novedades de Despacho
cupcdr	char	Código de Recargo

Cod_usuario	char	Código Usuario

PASAJERO

Nombre	**Tipo de Dato**	**Descripción**
pascod	char	Código pasajero
pasnom	char	Nombre Pasajero
pasced	char	Cédula pasajero
tercod	char	Tercero de pasajero
pricod	char	Prioridad Pasajero
reccod	char	Empresa recargo
pasare	char	Área del pasajero
pascam	char	Campo del pasajero
pasdep	char	Departamento pasajero
passec	char	Sección de pasajero
pastur	char	Turno de pasajero
pasedi	char	Día de entrada
paseco	char	Código de entrada
passvo	char	Salida vuelo
passdi	char	Día de salida
passco	char	Código de salida
agrcod	char	Código de agrupación
pastdo	char	Teléfono domicilio
pasdir	char	Dirección pasajero
pastof	char	Teléfono Oficina
pasfdn	char	Fecha de nacimiento pasajero
parcod	char	Código de parentesco
paspop	char	Prioridad operativa

pascus	char	Usuario
pascfe	Date	Fecha ingreso
pascho	Char	Hora ingreso
cod_usuario	Char	Código de usuario
pasmfe	Date	Fecha modificación
pasmho	Char	Hora modificación

VUELO

Nombre	**Tipo de Dato**	**Descripción**
voovoo	Char	Vuelo u otro
voofec	Date	Fecha de vuelo
vooeos	Char	Entrada o salida de vuelo
voocod	Char	Código de Vuelo
aprcod	Char	Aprobación de vuelo
descod	Char	Despacho del vuelo
voohor	Char	Hora del vuelo
vooaor	Char	Origen del vuelo
vooade	Char	Destino del vuelo
vooest	Char	Estado del vuelo
vootip	Char	Tipo de vuelo
avimat	Char	Matrícula de vuelo
voocfe	Date	Fecha de creación vuelo
voocho	Char	Hora de creación de vuelo
voomfe	Char	Fecha modificación vuelo
voomho	Char	Hora modificación vuelo

TERCEROS

Nombre	**Tipo de Dato**	**Descripción**
cupcdt	char	Código Terceros
nomcdt	char	Nombre terceros

RECARGO

Nombre	**Tipo de Dato**	**Descripción**
Cupcdr	char	Código recargo
nomcdr	char	Nombre recargo

PRIO_TERCEROS

Nombre	**Tipo de Dato**	**Descripción**
cuppdt	char	Código prioridad Terceros
nompdt	char	Nombre prioridad terceros

ESTADO

Nombre	**Tipo de Dato**	**Descripción**
cupfoe	char	Código Estado vuelo
nomfoe	char	Nombre Estado vuelo

AVION

Nombre	**Tipo de Dato**	**Descripción**
avicod	char	Código de avión
avimon	char	Nombre de avión

ORIGEN

Nombre	**Tipo de Dato**	**Descripción**
vooaor	char	Código de origen vuelo
nomaor	char	Nombre de origen vuelo

DESTINO

Nombre	**Tipo de Dato**	**Descripción**
vooade	char	Código de destino vuelo
nomade	char	Nombre de destino vuelo

ESTADO DE VUELO

Nombre	**Tipo de Dato**	**Descripción**
vooest	char	Código de estado vuelo
vomest	char	Nombre de estado vuelo

TERCEROS_PAS

Nombre	**Tipo de Dato**	**Descripción**
tercod	char	Código de terceros de pasajero
ternom	char	Nombre de terceros de pasajero

PRIORIADAD

Nombre	**Tipo de Dato**	**Descripción**
pricod	char	Código de prioridad de pasajero
prinom	char	Nombre de prioridad de pasajero

PARENTEZCO

Nombre	**Tipo de Dato**	**Descripción**
parcod	char	Código de parentesco pasajero
parnom	char	Nombre de parentesco pasajero

SECCION

Nombre	**Tipo de Dato**	**Descripción**
pascod	char	Código de sección pasajero
pasnom	char	Nombre de sección pasajero

RECARGO_PAS

Nombre	**Tipo de Dato**	**Descripción**
reccod	char	Código de recargo a pasajero

recnom	char	Nombre de recargo a pasajero

AREA

Nombre	**Tipo de Dato**	**Descripción**
pasare	char	Código de área del pasajero
nomare	char	Nombre de área pasajero

DEPARTAMENTO

Nombre	**Tipo de Dato**	**Descripción**
pasdep	char	Código de departamento pasajero
nomdep	char	Nombre de departamento pasajero

CAMPO

Nombre	**Tipo de Dato**	**Descripción**
pascam	char	Código de campo del pasajero
nomcam	char	Nombre de campo del pasajero

ACC_ASIGNA _MENU

Nombre	**Tipo dato**	**Descripción**
Cedula	Stirling	Cédula usuario.

MEN_CATEGORIA

Nombre	**Tipo dato**	**Descripción**
Cat_codigo	string	Código de categoría.
Cat-descripcion	string	Descripción de categoría.

MEN_SCATEGORIA

Nombre	**Tipo dato**	**Descripción**
sca_codigo	string	Código de subcategoria.
sca-descripcion	string	Descripción de subcategoria.
Sca_linksca	string	Relacion categoría con subcategoria.

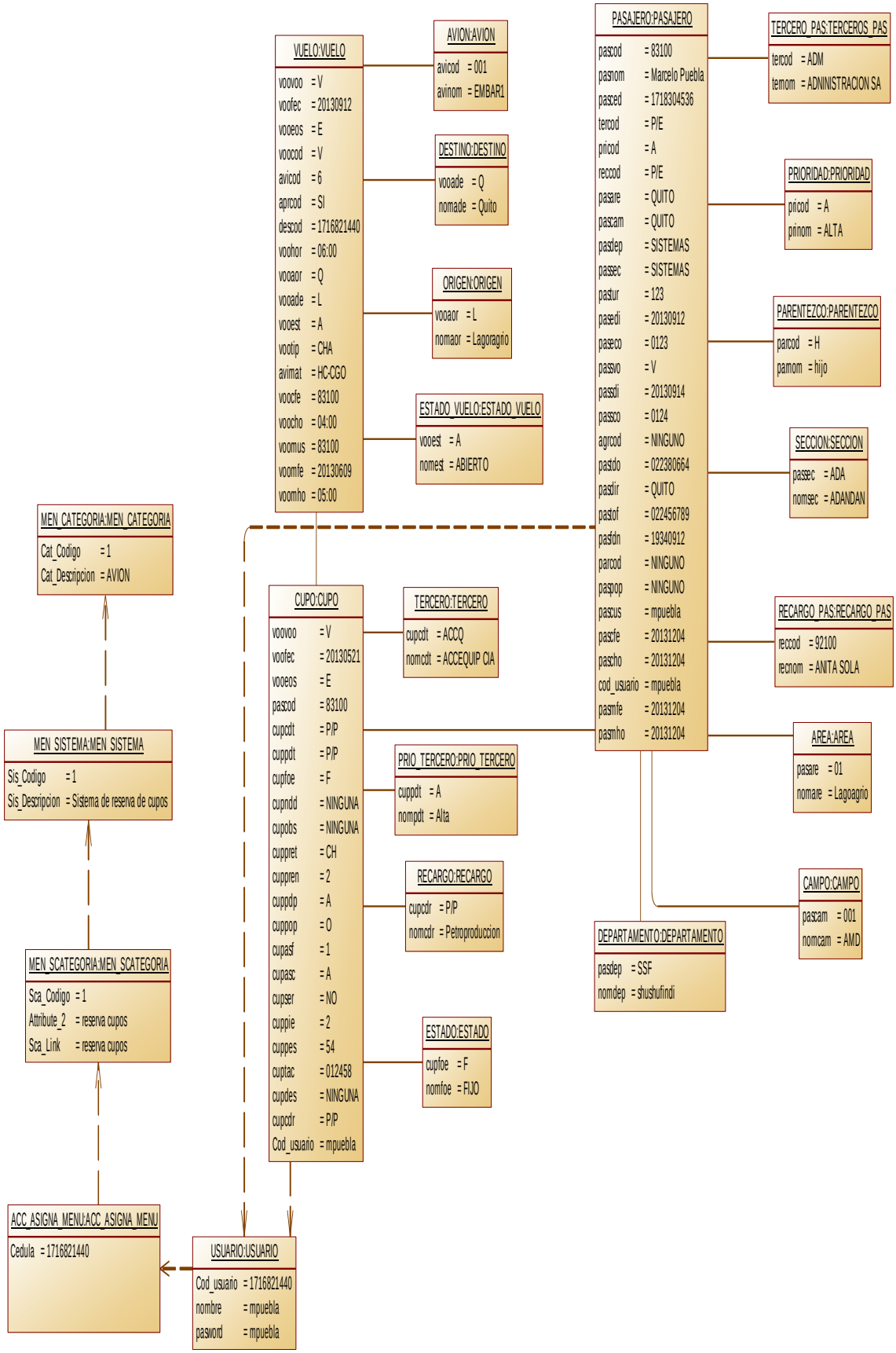
MEN_SISTEMA

Nombre	**Tipo dato**	**Descripción**
Sis_codigo	string	Código de sistema.
Sis_descripcion	string	Descripción de sistema.

USUARIOS

Nombre	**Tipo dato**	**Descripción**
Cod_usuario	string	Cédula.
Nombre	string	Nombre completo.
Pasword	string	Clave
usuario	string	Primera inicial del primer nombre + apellido

DIAGRAMA DE OBJETOS

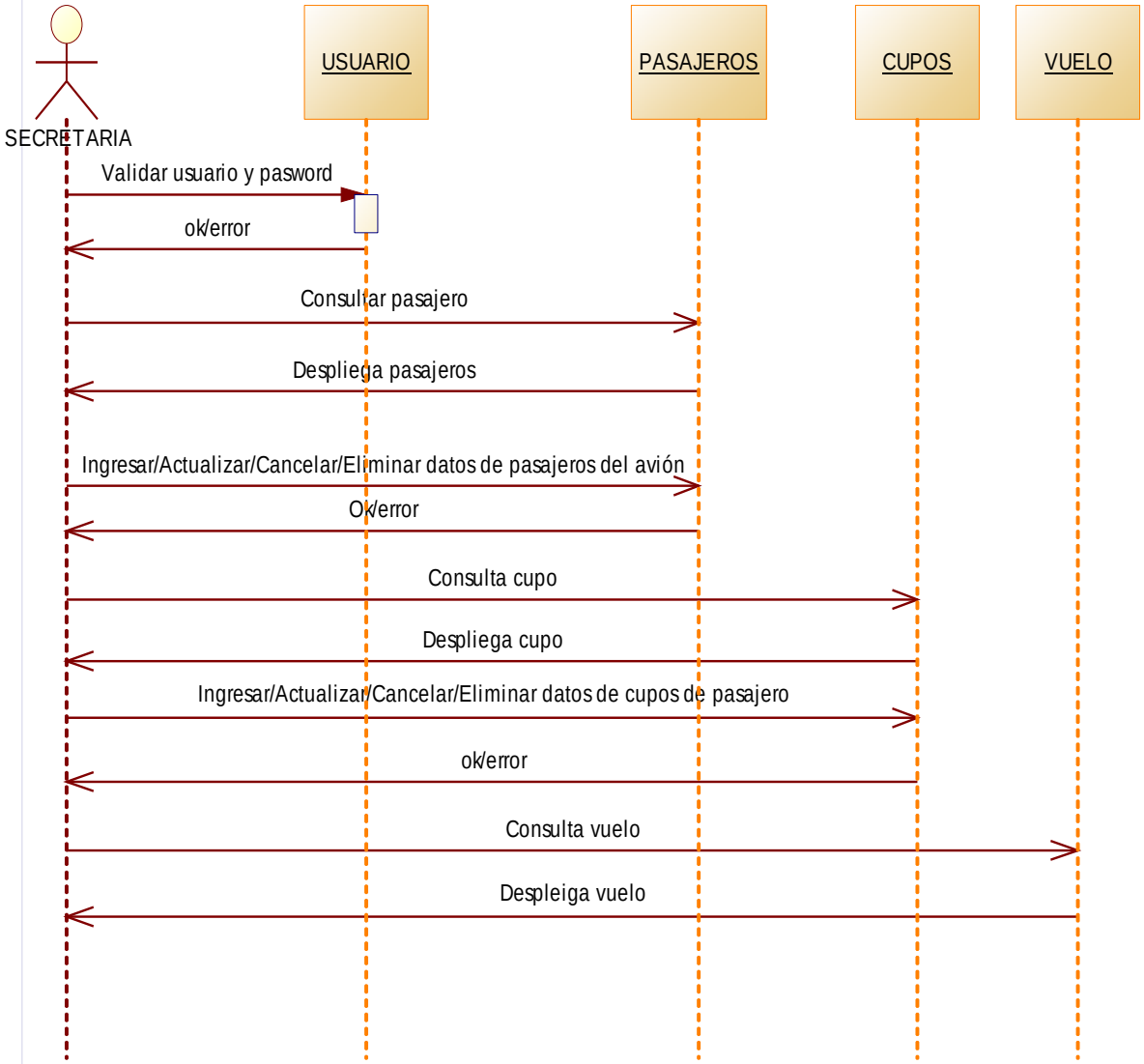

VUELO:VUELO
voovoo = V
voofec = 20130912
vooeos = E
voocod = V
avicod = 6
aprcod = SI
descod = 1716821440
voohor = 06:00
vooaor = Q
vooade = L
vooest = A
vootip = CHA
avimat = HC-CGO
voocfe = 83100
voocho = 04:00
voomus = 83100
voomfe = 20130609
voomho = 05:00
AVION:AVION
avicod = 001
avinom = EMBAR1
DESTINO:DESTINO
vooade = Q
nomade = Quito
ORIGEN:ORIGEN
vooaor = L
nomaor = Lagoragrio
ESTADO_VUELO:ESTADO_VUELO
vooest = A
nomest = ABIERTO
PASAJERO:PASAJERO
pascod = 83100
pasnom = Marcelo Puebla
pasced = 1718304536
tercod = P/E
pricod = A
reccod = P/E
pasare = QUITO
pascam = QUITO
pasdep = SISTEMAS
passec = SISTEMAS
pastur = 123
pasedi = 20130912
paseco = 0123
passvo = V
passdi = 20130914
passco = 0124
agrcod = NINGUNO
pasdo = 022380664
pasdir = QUITO
pastof = 022456789
pasfdn = 19340912
parcod = NINGUNO
paspop = NINGUNO
pascus = mpuebla
pascfe = 20131204
pascho = 20131204
cod_usuario = mpuebla
pasmfe = 20131204
pasmho = 20131204
TERCERO_PAS:TERCEROS_PAS
tercod = ADM
temom = ADNINISTRACION SA
PRIORIDAD:PRIORIDAD
pricod = A
prinom = ALTA
PARENTEZCO:PARENTEZCO
parcod = H
pamom = hijo
SECCION:SECCION
passec = ADA
nomsec = ADANDAN
RECARGO_PAS:RECARGO_PAS
reccod = 92100
recnom = ANITA SOLA
AREA:AREA
pasare = 01
nomare = Lagoagrio
CAMPO:CAMPO
pascam = 001
nomcam = AMD
DEPARTAMENTO:DEPARTAMENTO
pasdep = SSF
nomdep = shushufindi
MEN_CATEGORIA:MEN_CATEGORIA
Cat_Codigo = 1
Cat_Descripcion = AVION
MEN_SISTEMA:MEN_SISTEMA
Sis_Codigo = 1
Sis_Descripcion = Sistema de reserva de cupos
MEN_SCATEGORIA:MEN_SCATEGORIA
Sca_Codigo = 1
Attribute_2 = reserva cupos
Sca_Link = reserva cupos
ACC_ASIGNA_MENU:ACC_ASIGNA_MENU
Cedula = 1716821440
CUPO:CUPO
voovoo = V
voofec = 20130521
vooeos = E
pascod = 83100
cupcdt = P/P
cuppdt = P/P
cupfoe = F
cupndd = NINGUNA
cupobs = NINGUNA
cuppret = CH
cuppren = 2
cuppdp = A
cuppop = O
cupasf = 1
cupasc = A
cupser = NO
cuppie = 2
cuppes = 54
cuptac = 012458
cupdes = NINGUNA
cupcdr = P/P
Cod_usuario = mpuebla
TERCERO:TERCERO
cupcdt = ACCQ
nomcdt = ACCEQUIP CIA
PRIO_TERCERO:PRIO_TERCERO
cuppdt = A
nompdt = Alta
RECARGO:RECARGO
cupcdr = P/P
nomcdr = Petroproducción
ESTADO:ESTADO
cupfoe = F
nomfoe = FIJO
USUARIO:USUARIO
Cod_usuario = 1716821440
nombre = mpuebla
pasword = mpuebla

DISEÑO

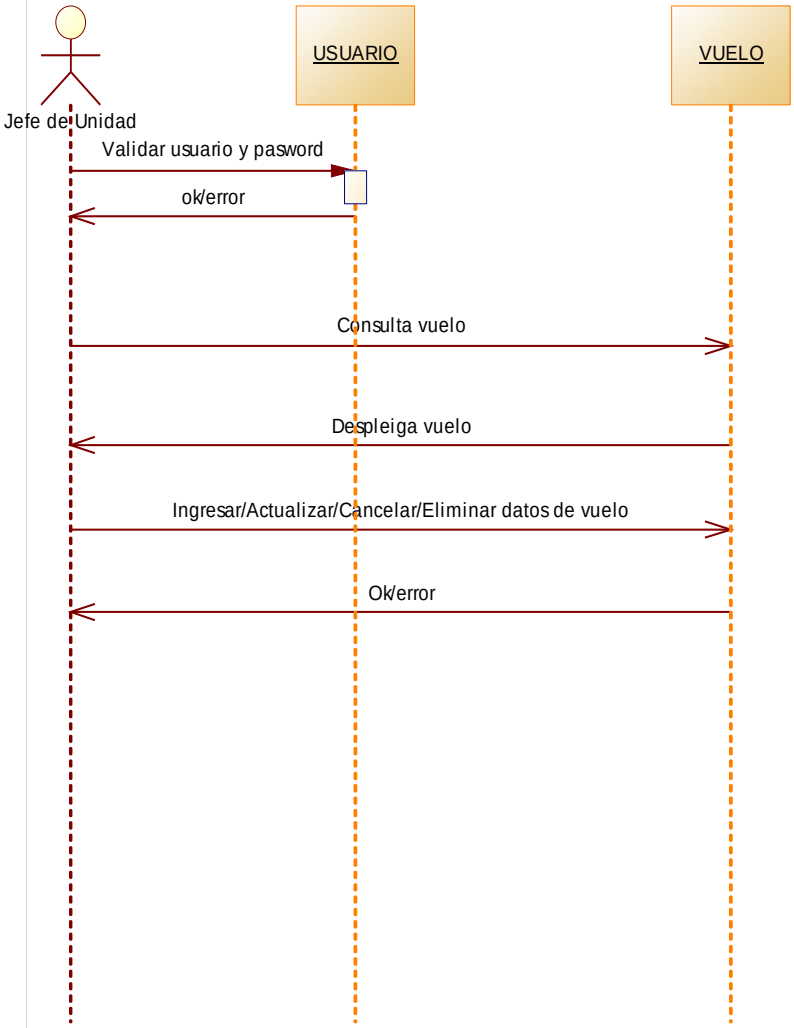
MODELO DINÁMICO

DIAGRAMAS DE SECUENCIA

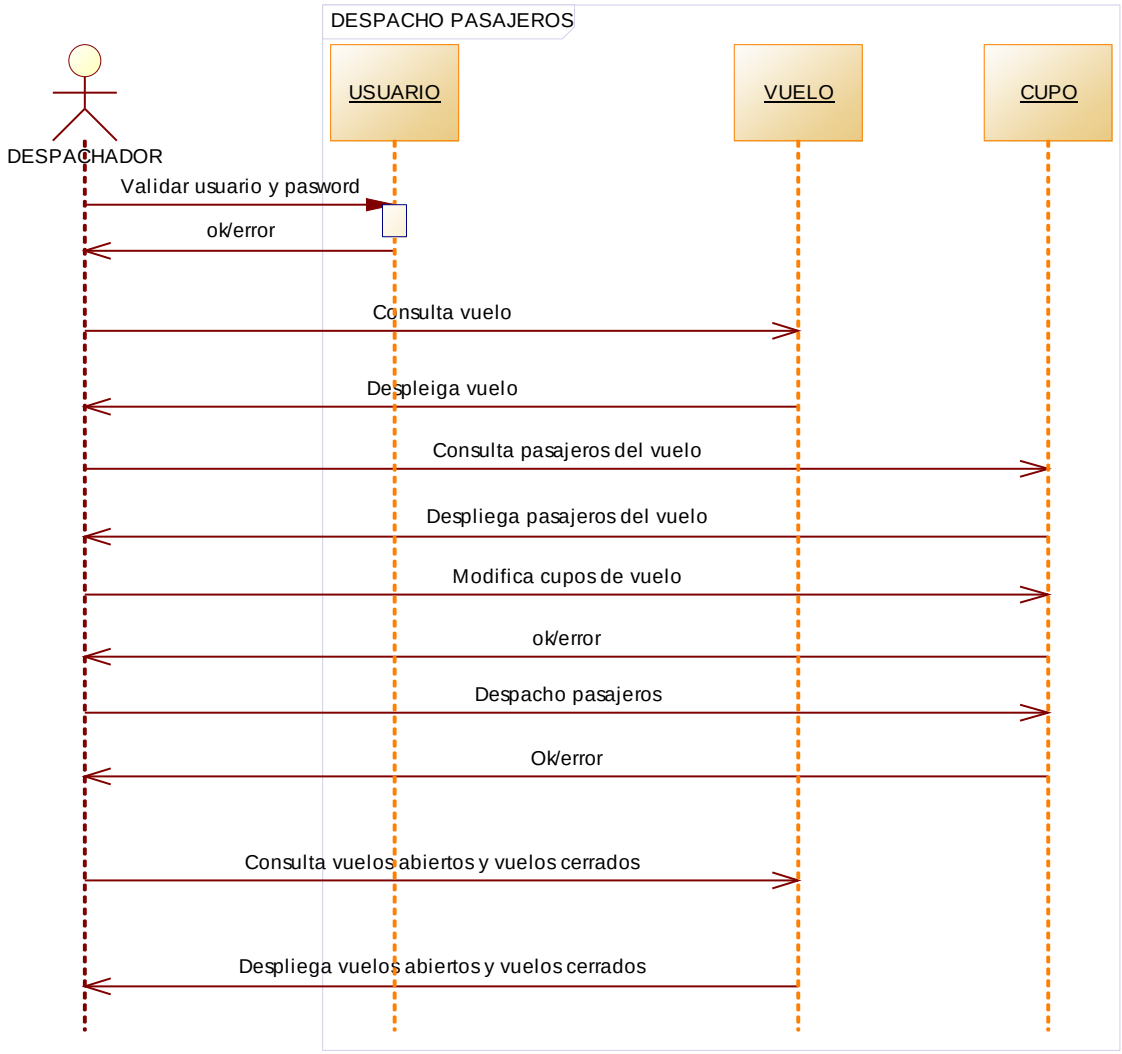
INGRESO PASAJEROS, CUPOS



INGRESO DE VUELO



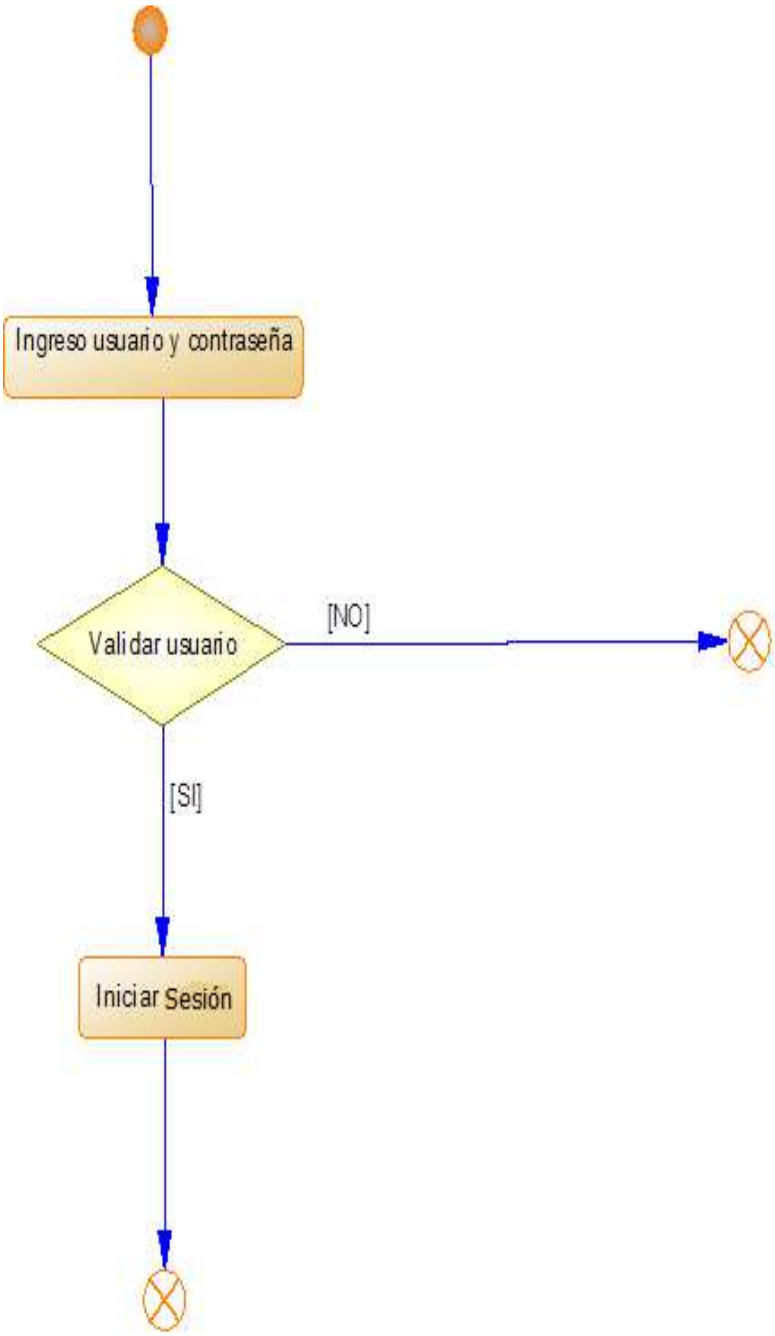
DESPACHO


DESPACHO PASAJEROS
DESPACHADOR
USUARIO
VUELO
CUPO
Validar usuario y pasword
ok/error
Consulta vuelo
Despleiga vuelo
Consulta pasajeros del vuelo
Despliega pasajeros del vuelo
Modifica cupos de vuelo
ok/error
Despacho pasajeros
Ok/error
Consulta vuelos abiertos y vuelos cerrados
Despliega vuelos abiertos y vuelos cerrados

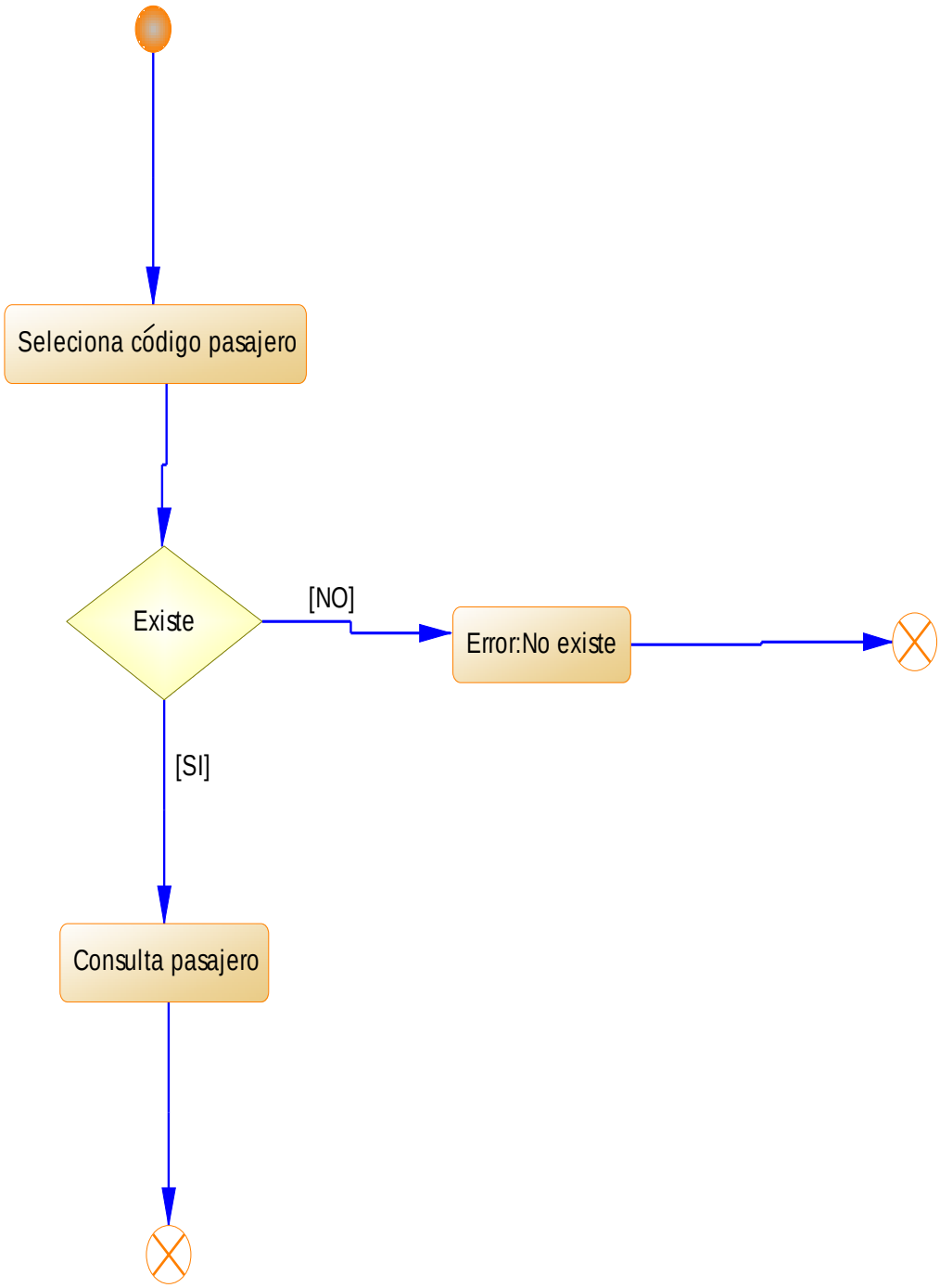
MODELO FUNCIONAL

DIAGRAMA DE ACTIVIDADES

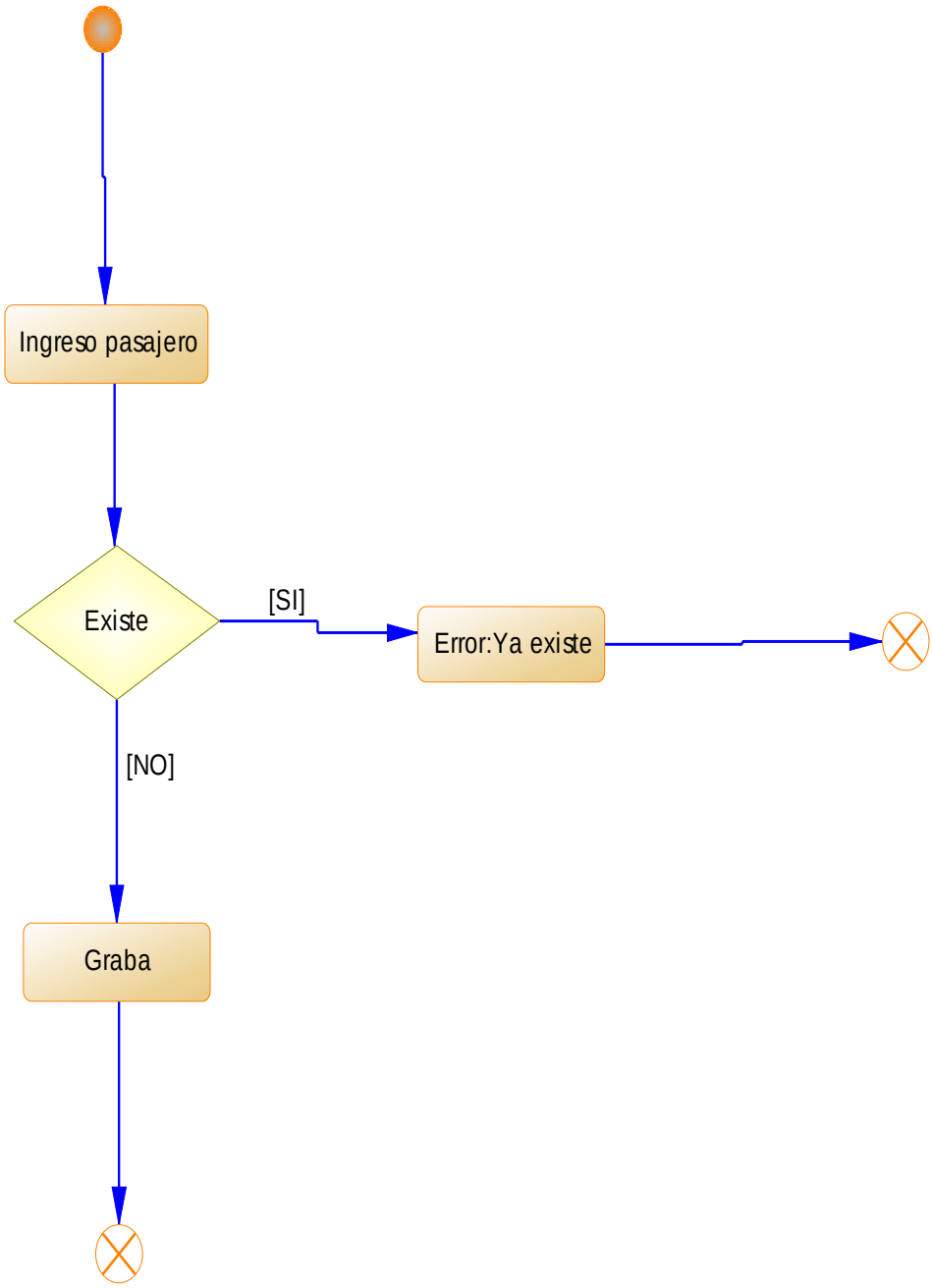
INICIO DE SESIÓN



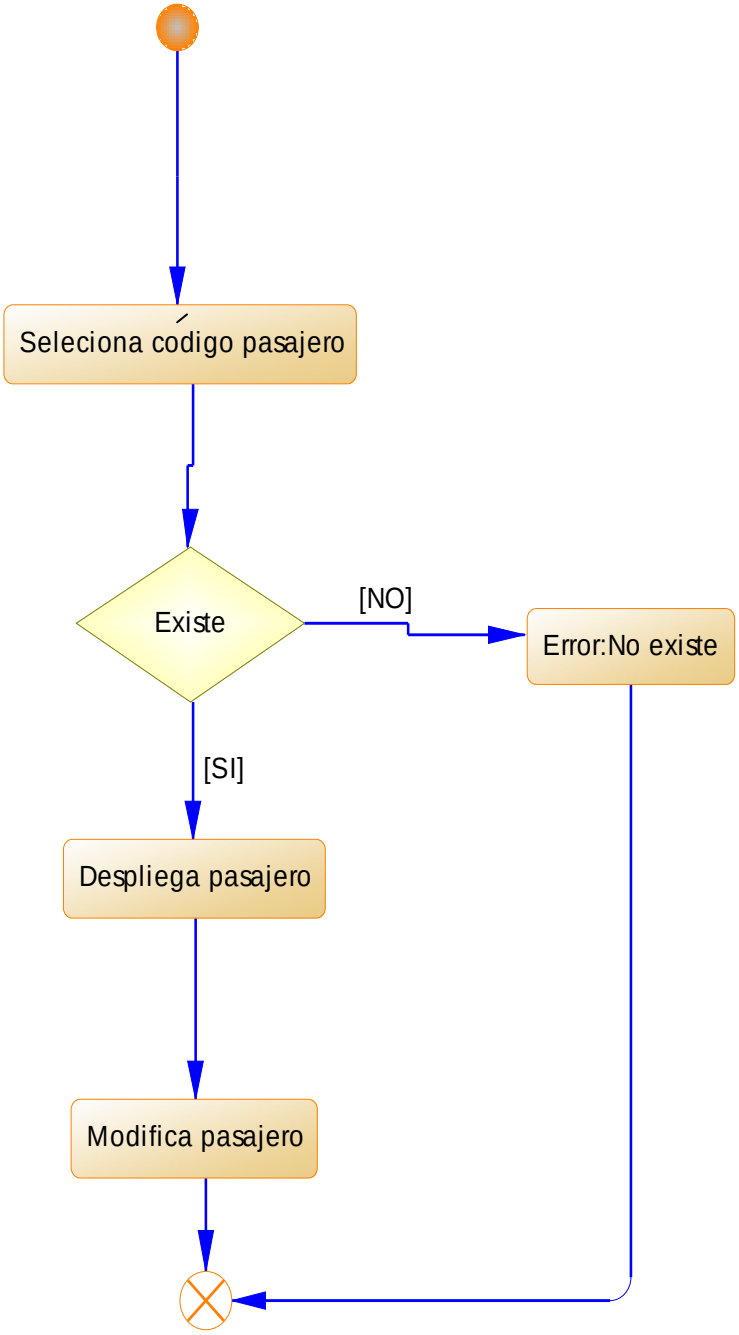
CONSULTA PASAJERO



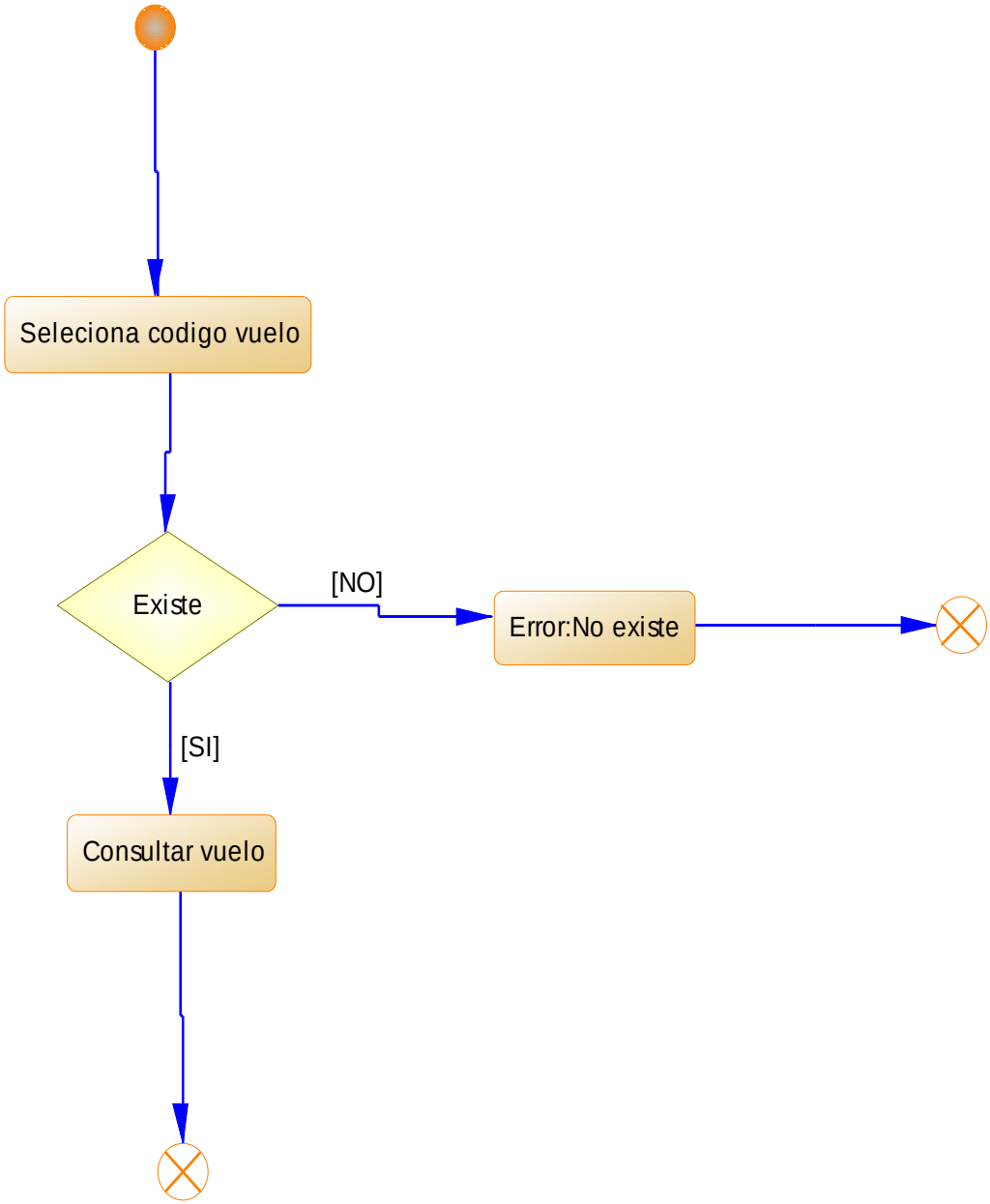
INGRESA DATOS PASAJERO



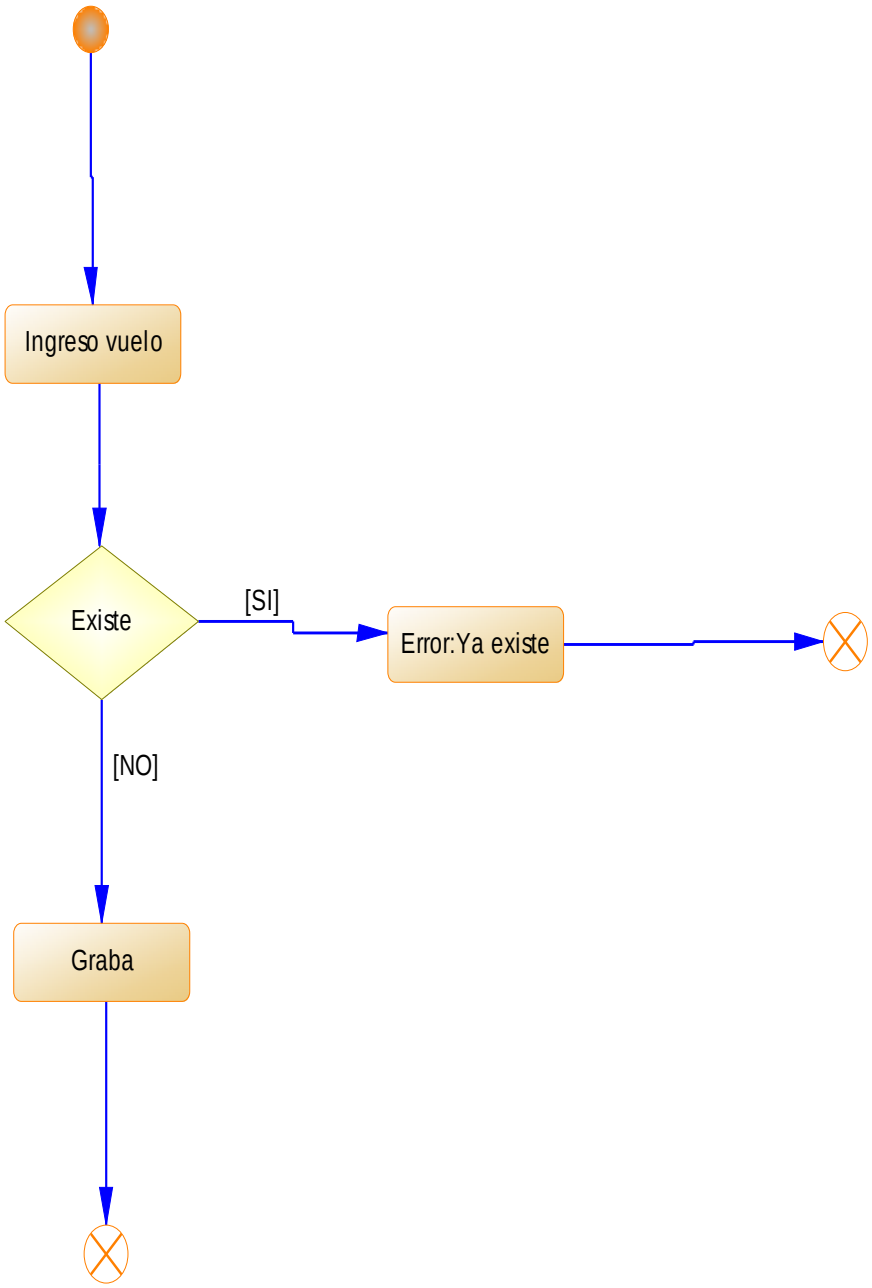
MODIFICA DATOS DE PASAJERO



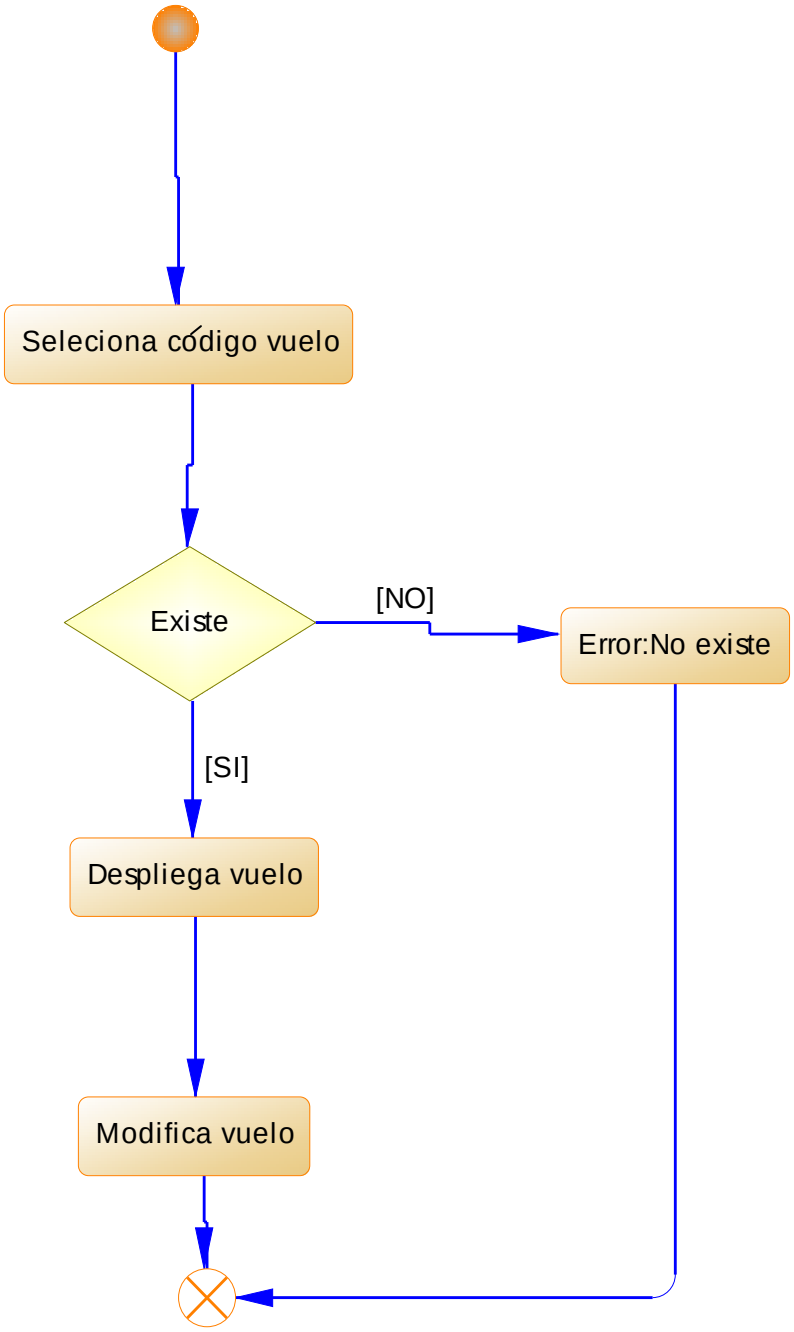
CONSULTA VUELOS



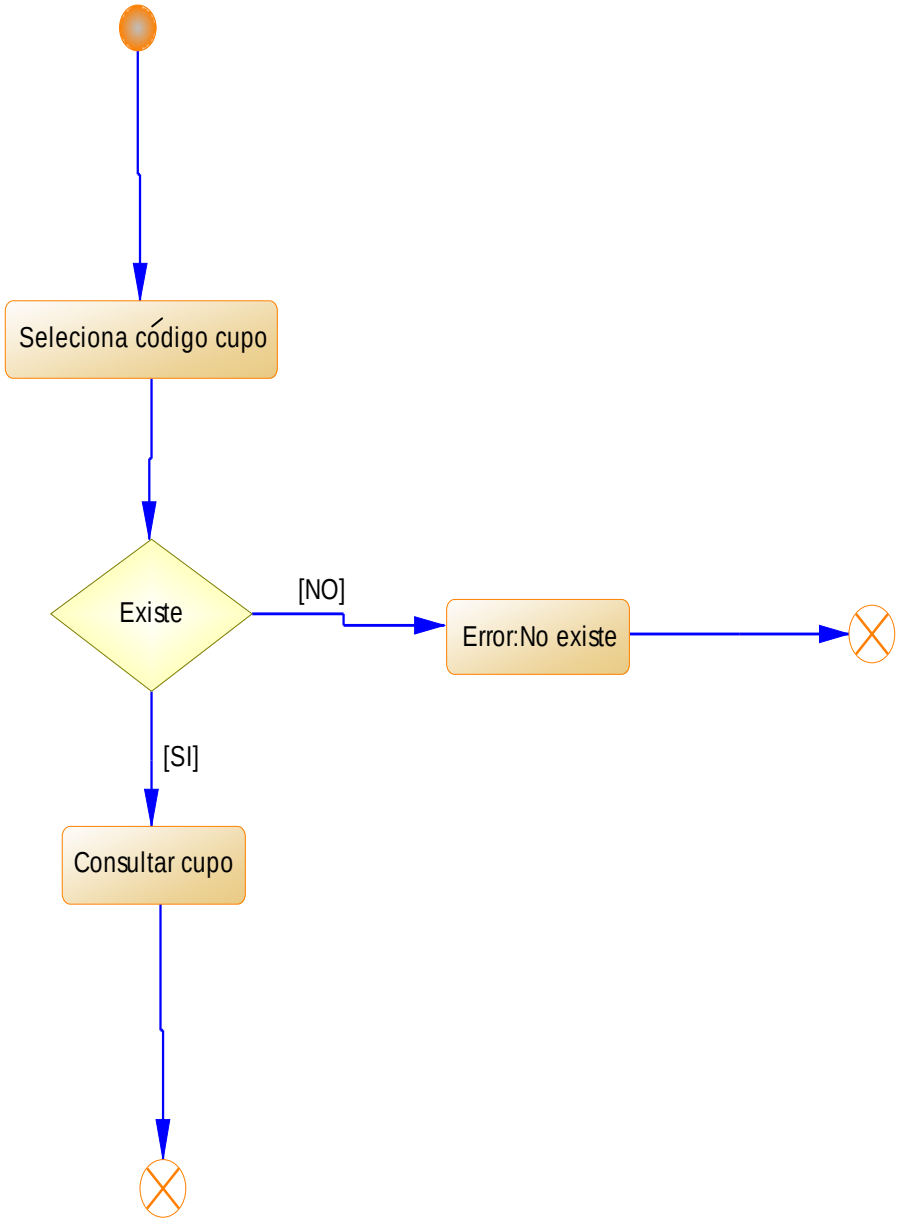
INGRESO VUELO



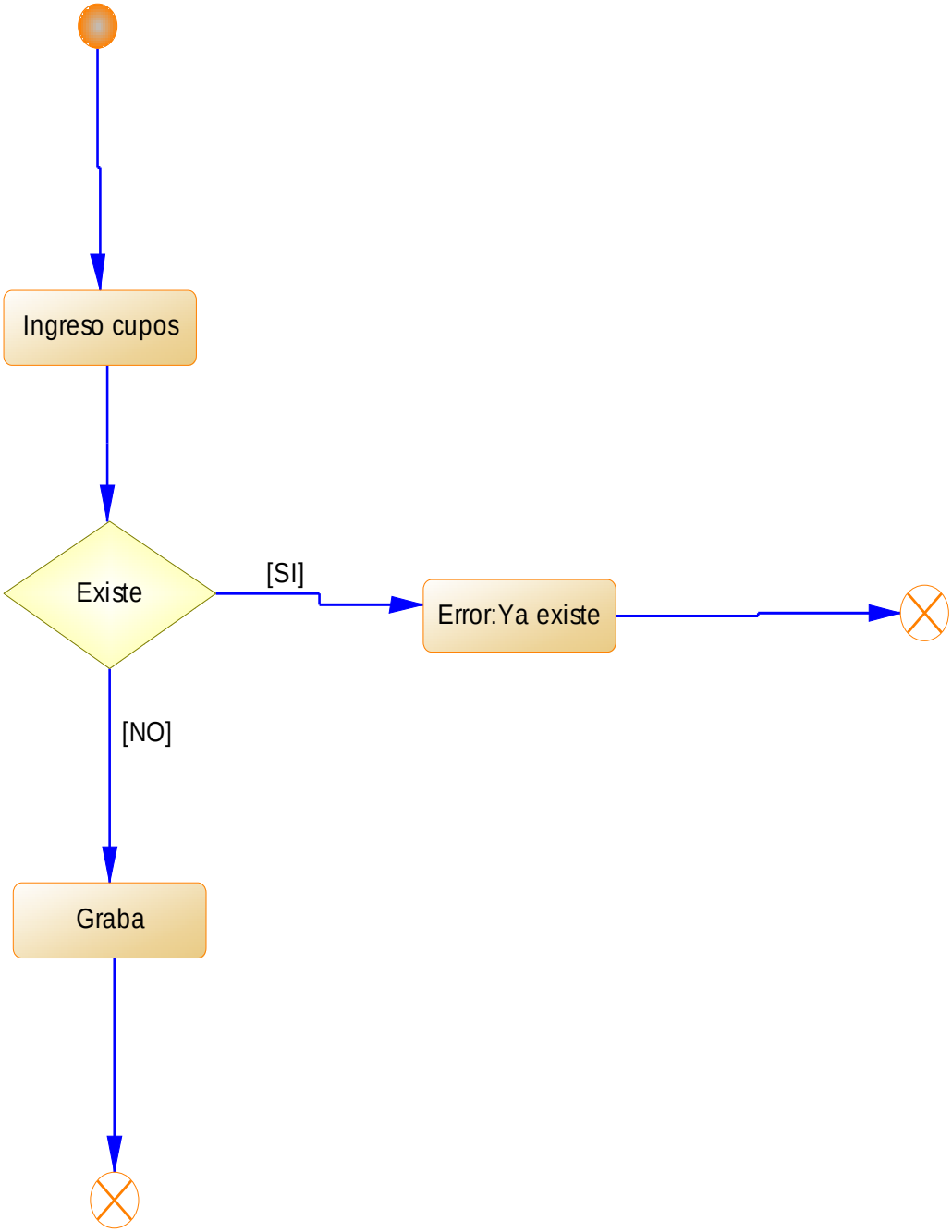
MODIFICA VUELO



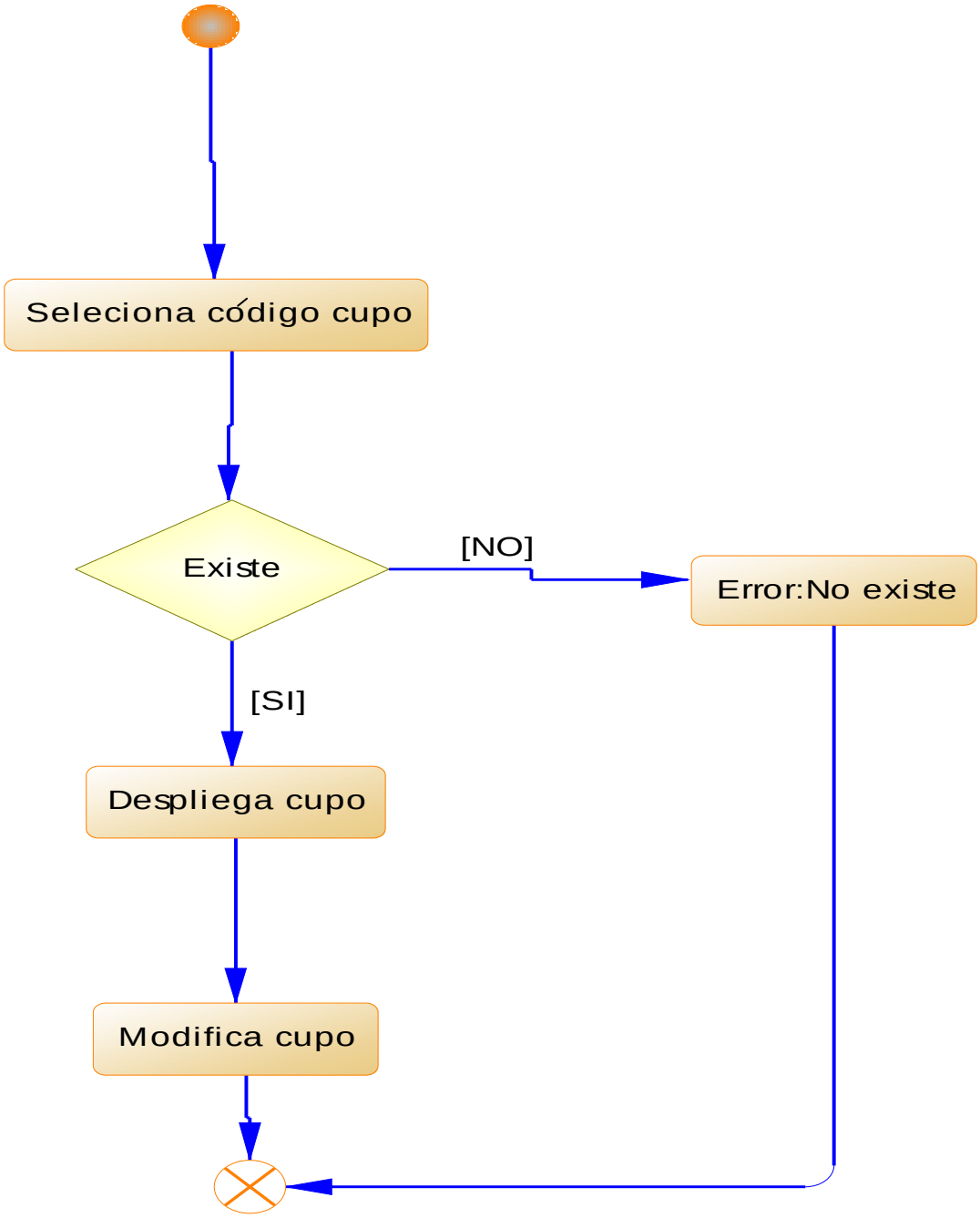
CONSULTA CUPOS DE VUELO



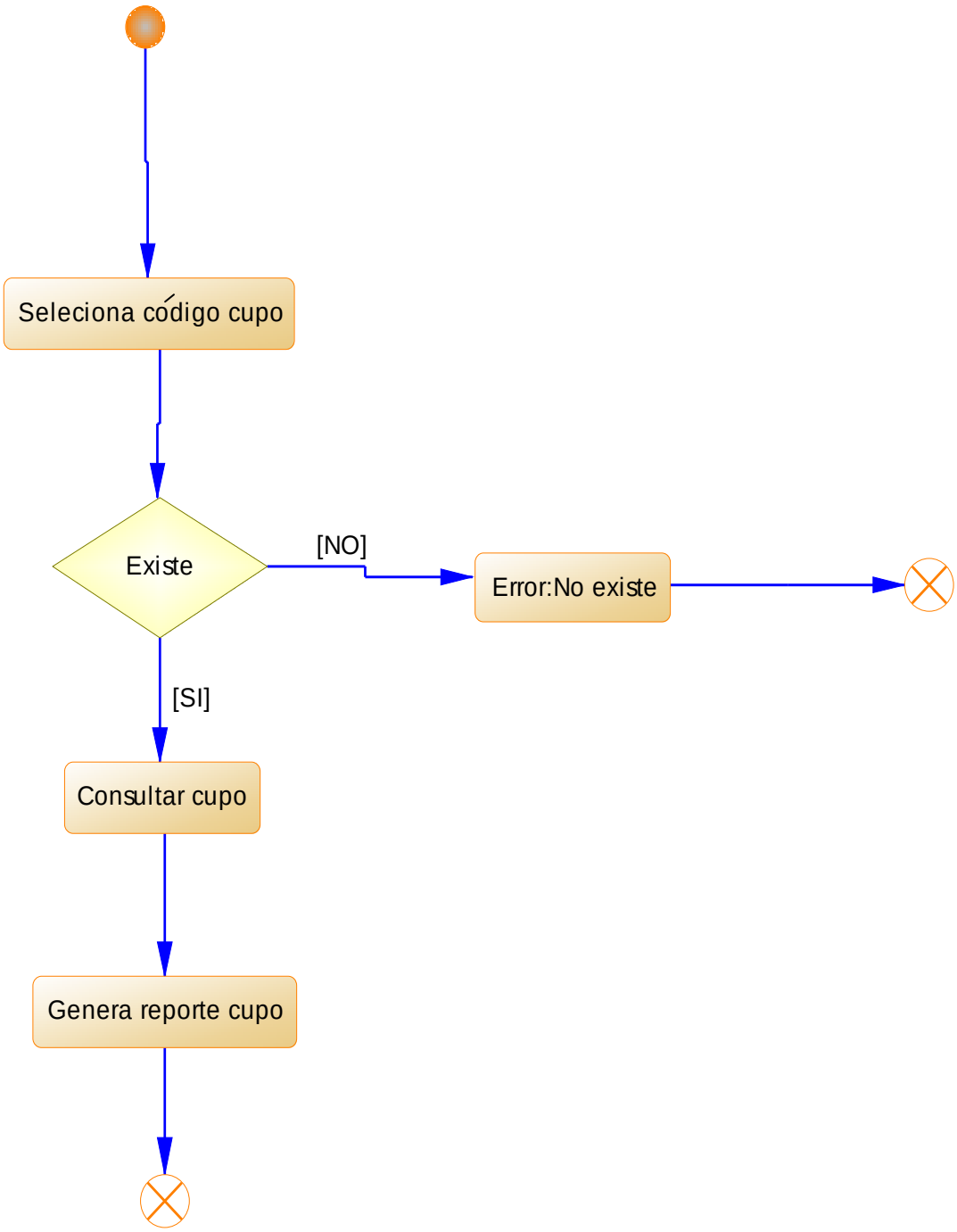
INGRESO CUPOS DE VUELO



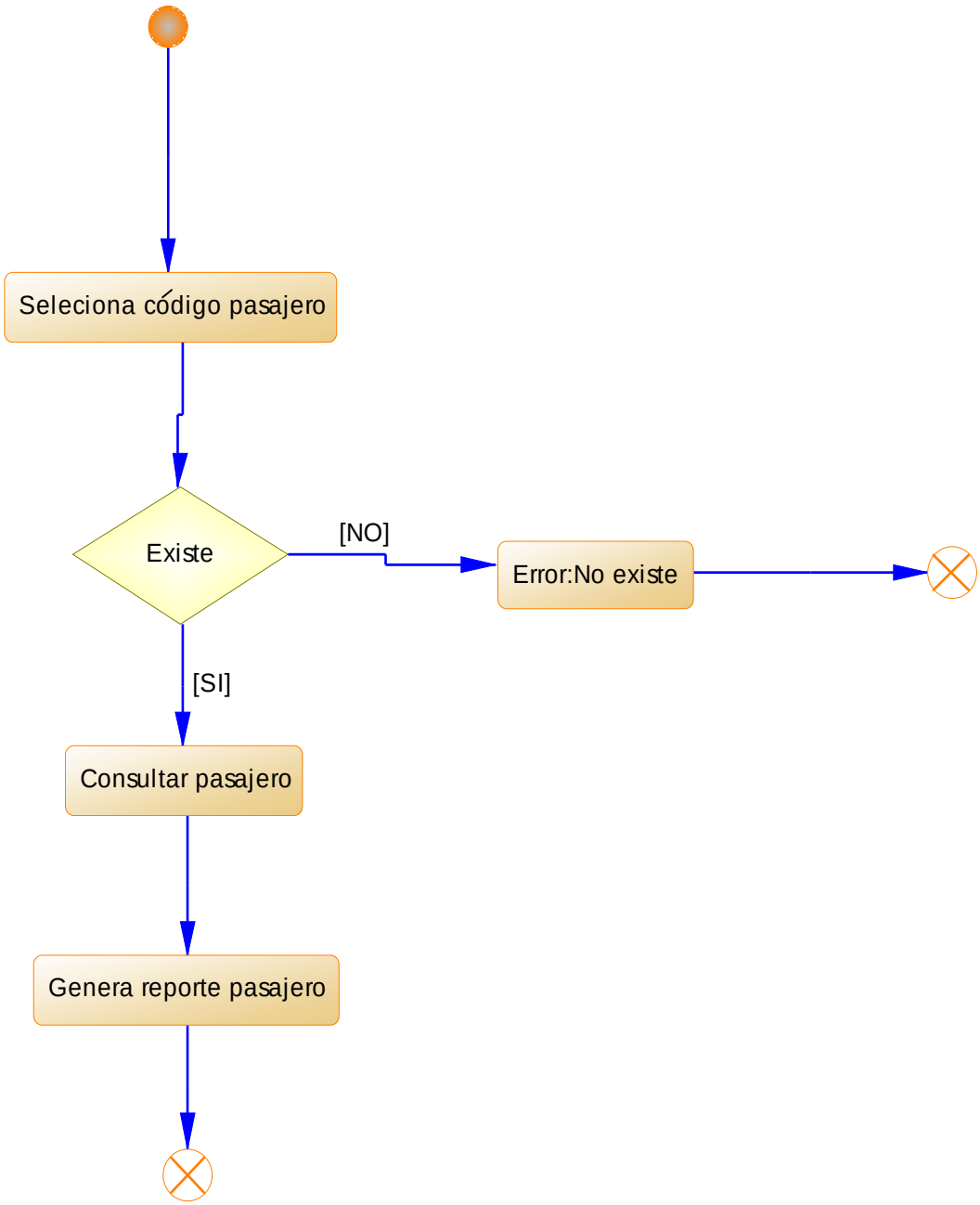
MODIFICA CUPOS DE AVIÓN



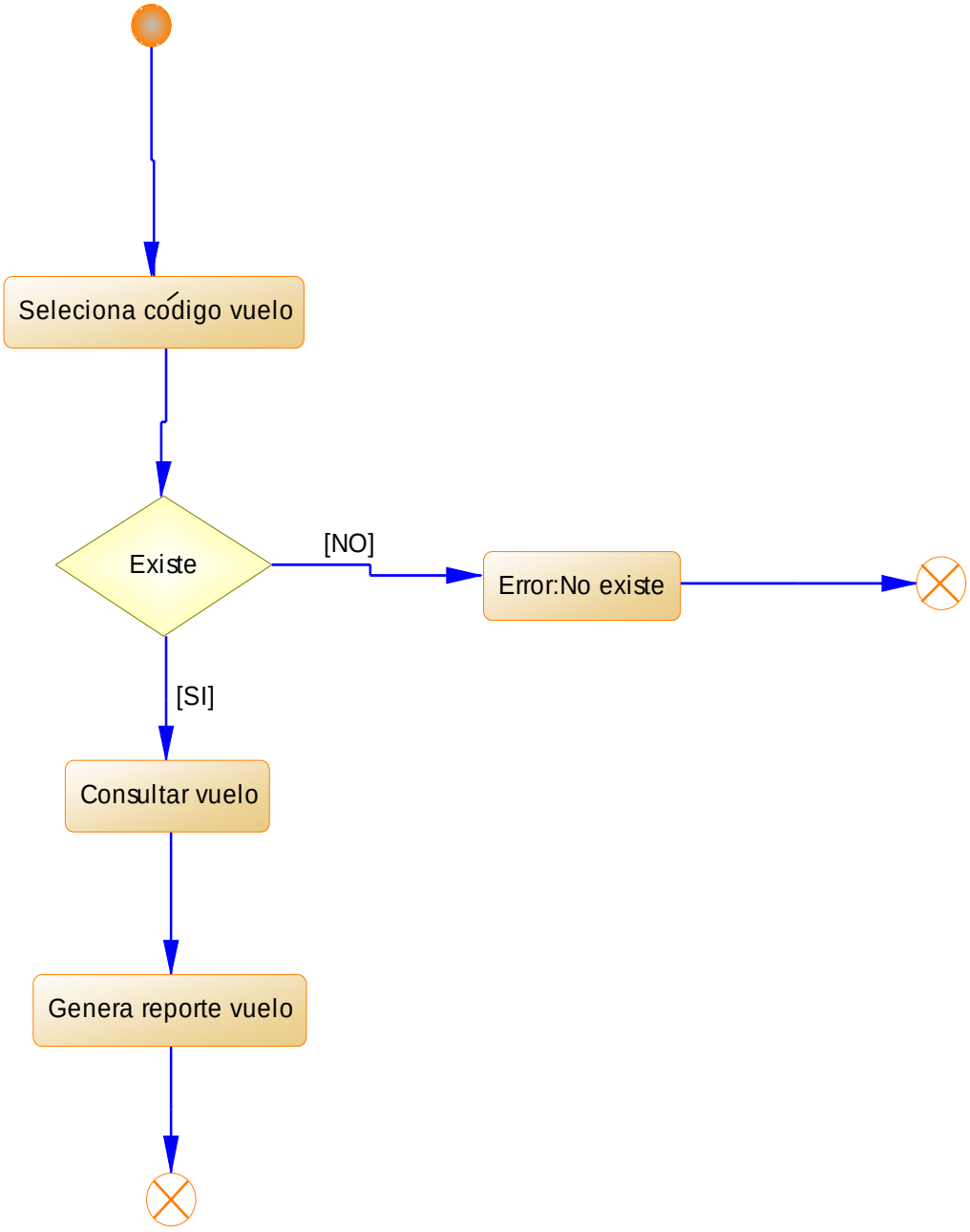
GENERAR REPORTE CUPO



GENERAR REPORTE PASAJERO

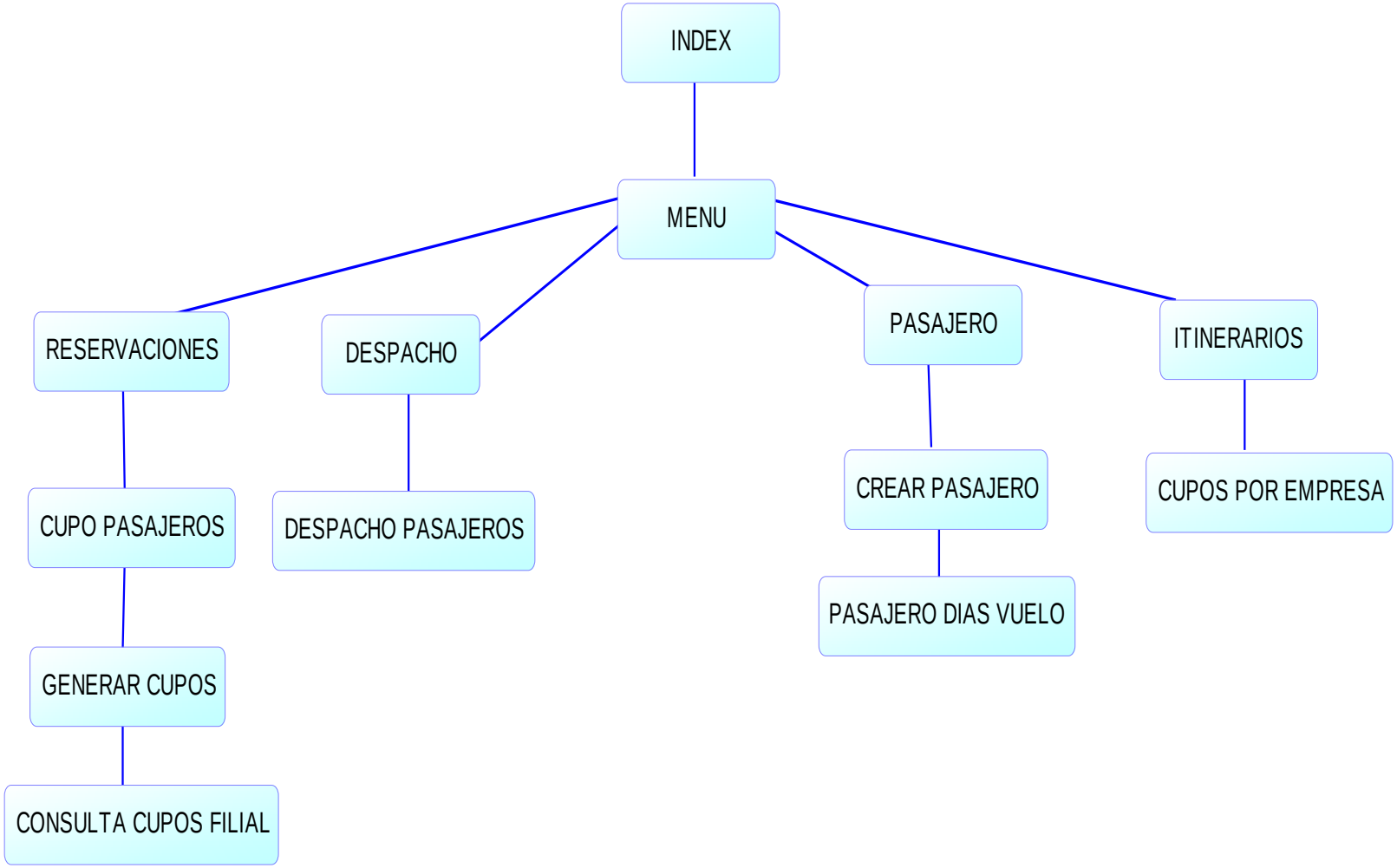


GENERAR REPORTE VUELO



DISEÑO NAVEGACIONAL

MODELO NAVEGACIONAL



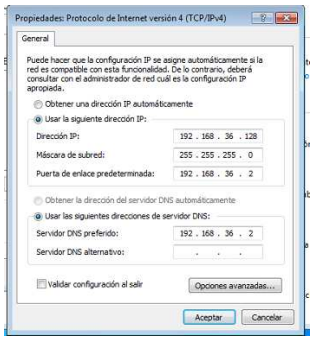
MANUAL DE INSTALACIÓN

SISTEMA **WEB DE RESERVACIONES DE CUPOS DE AVION PARA PETROECUADOR CON LA HERRAMIENTA COLDFUSION MX 8**

INSTALACIÓN DEL SERVIDOR ORACLE ENTERPRISE 11g:

Requisitos mínimos de Software:

Una dirección IP estática en el equipo en el que va a instalar Oracle 11g. Si intenta instalar Oracle 11g sin disponer de ella, se le indicará que instale un adaptador de bucle cerrado.

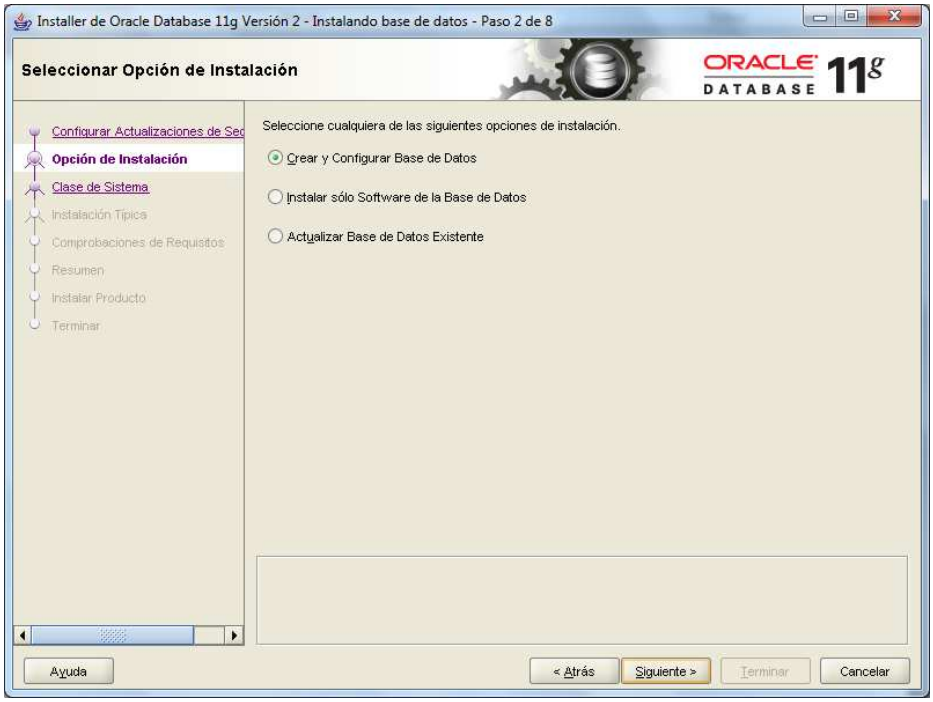


REQUISITOS	VALORES
Procesador:	Intel (x86), AMD64, e Intel EM64T Oracle proporciona versiones tanto para 32bits como para 64bits
RAM:	1 GB Ram (Recomendado 2GB en Windows 7 y Windows 2008 Server)
Disco duro:	5.39 GB para la instalación típica y 5.89GB para la instalación avanzada

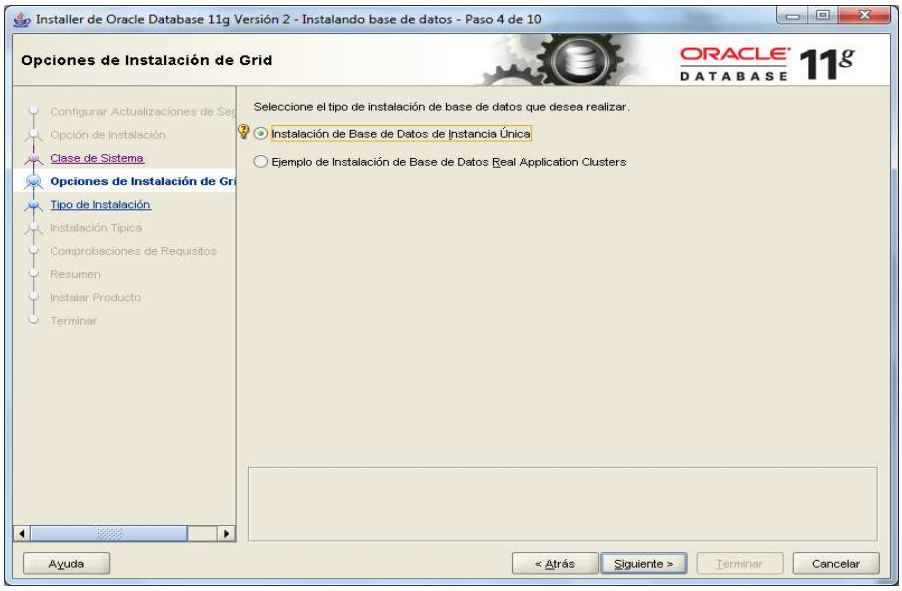
Pasos para la Instalación

1. Ingreso de una dirección de correo electrónico (opcional)

2. Seleccionar crear y configurar Base de Datos

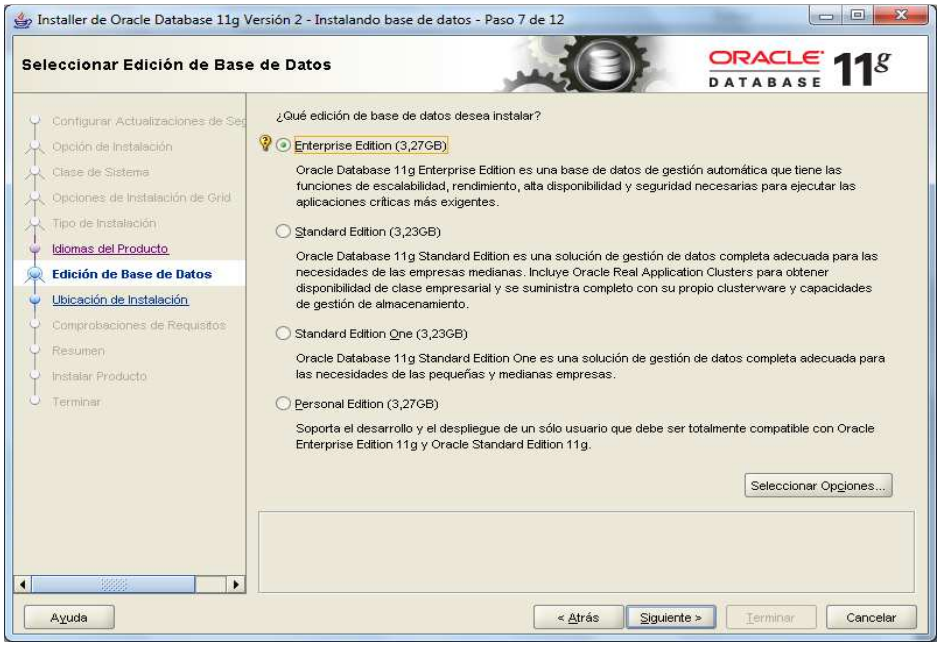


3. Seleccionar instalación de base de datos como una instancia única.

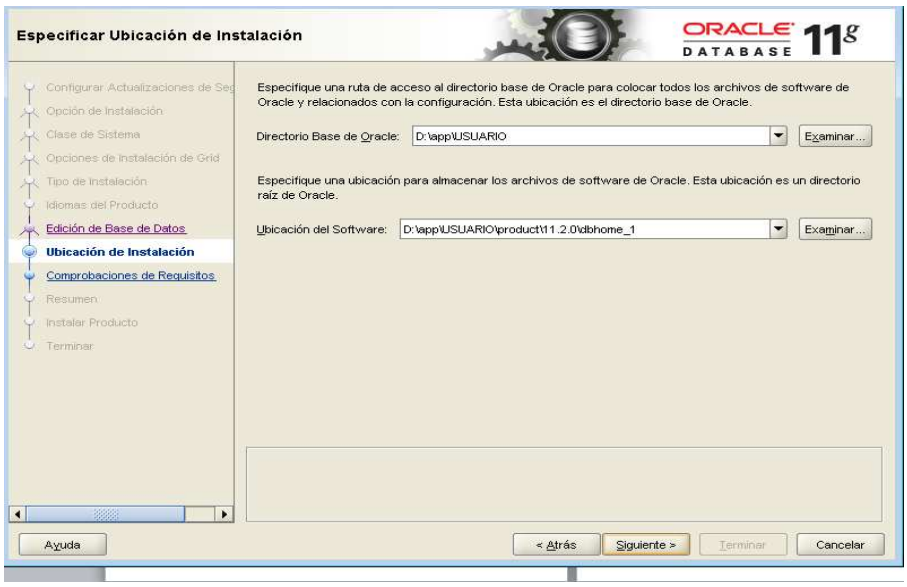


4. Seleccionar el idioma deacuerdo a su pais de recidencia.

5. Seleccionar la edición de base datos, Enterprise Edition, que gestiona automáticamente las funciones de rendimiento

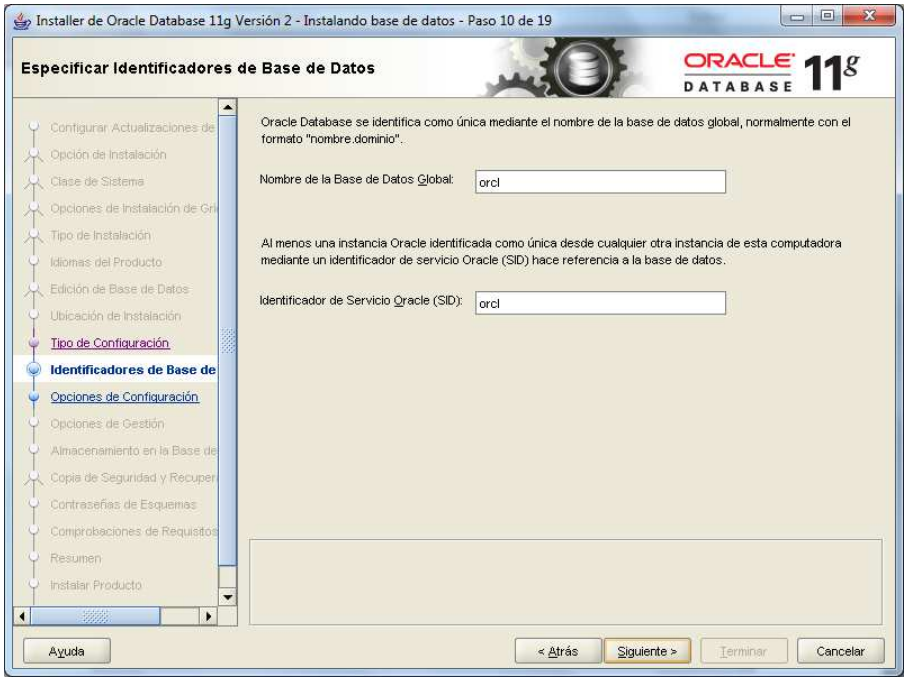


6. Seleccionar la ruta para la instación de los archivos relacionados con la configuración de Oracle 11g



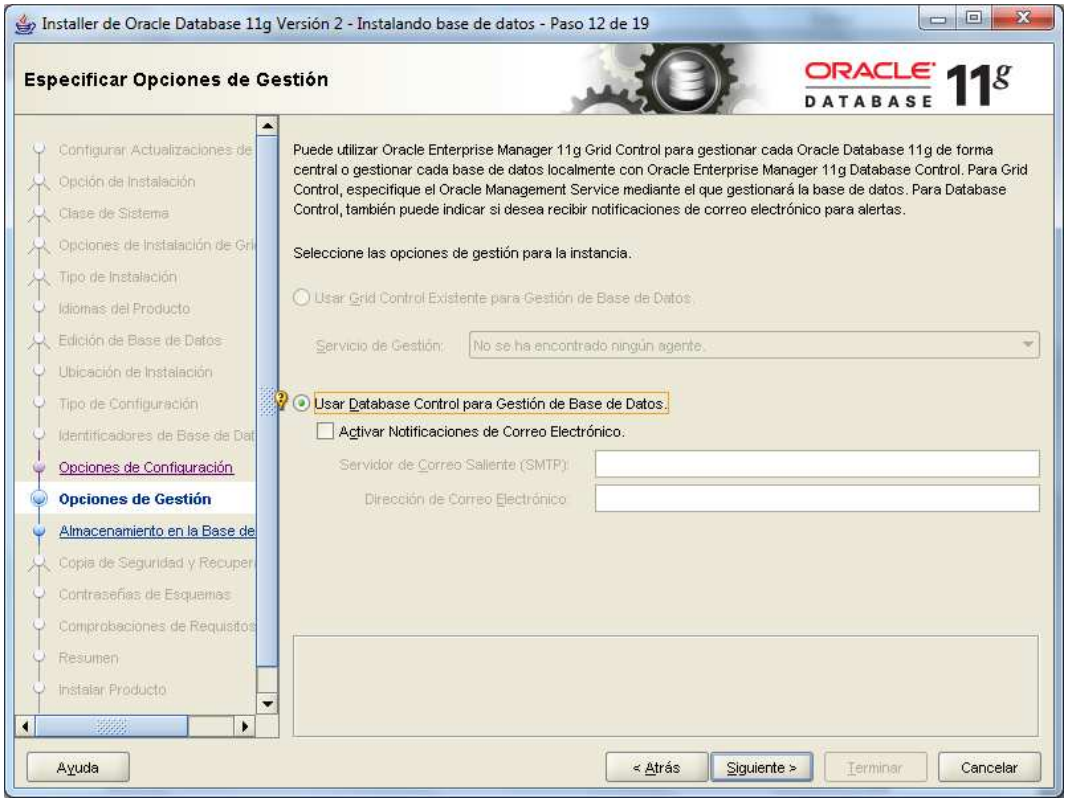
7. seleccionamos Uso General.

8. Ingresar el nombre de la base de datos Global

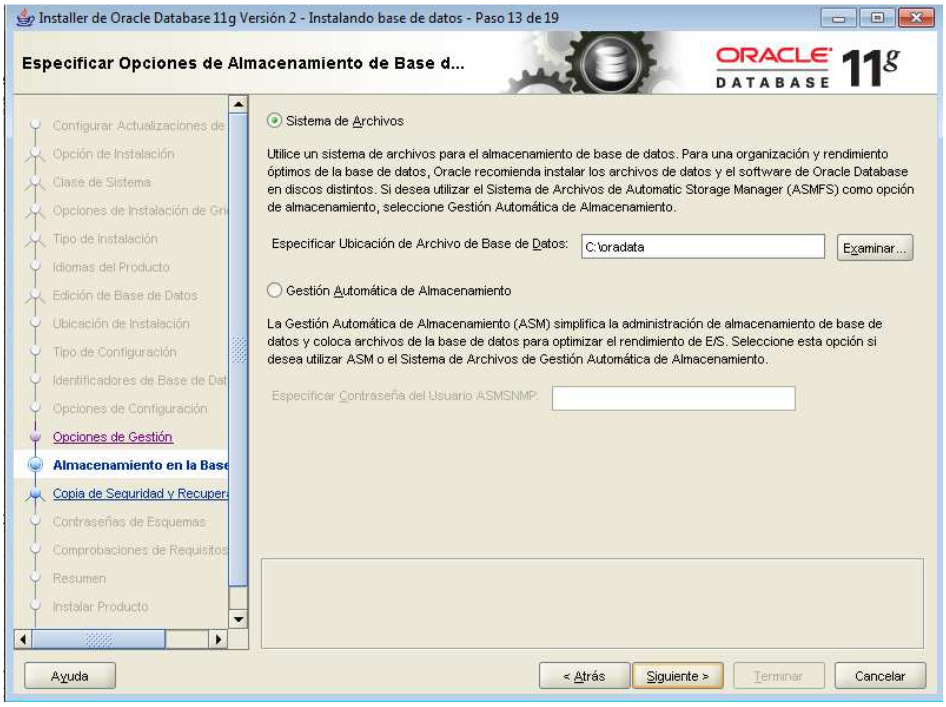


9. Seleccionamos la casilla Gestión automática de Memoria, para que el servidor gestione la memoria de una manera conveniente de acuerdo a la necesidad.

10. Activar User Database Control para gestional las bases por usuarios



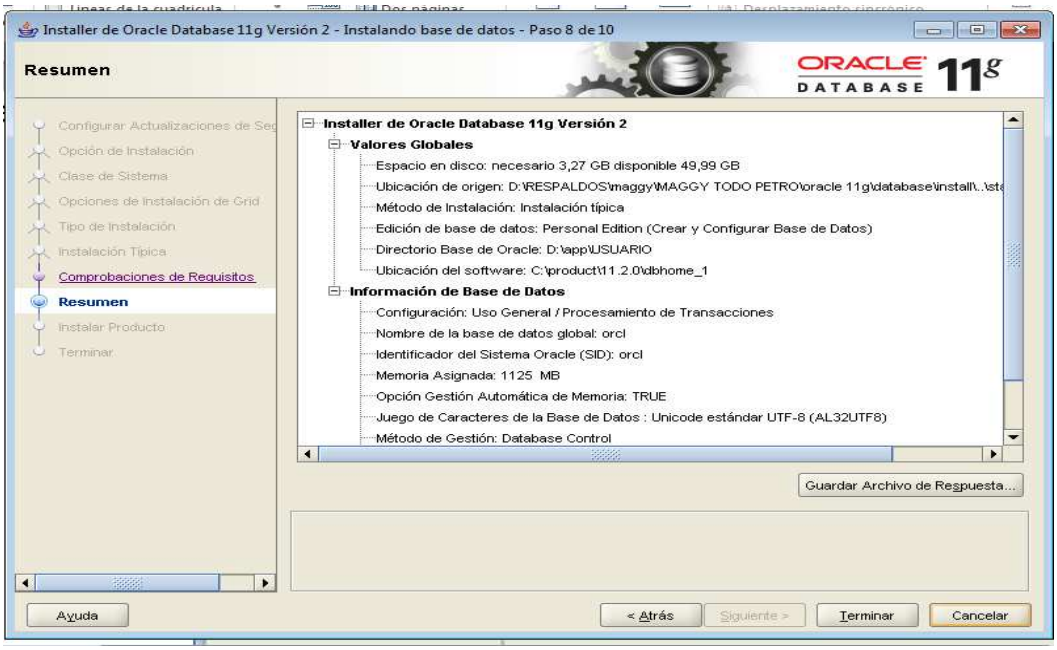
11. Seleccionar Sistema de Archivos para el almacenamiento de base de datos. Esto optimiza el rendimiento del servidor



12. Escoger No acivar Copias de seguridad Automáticas

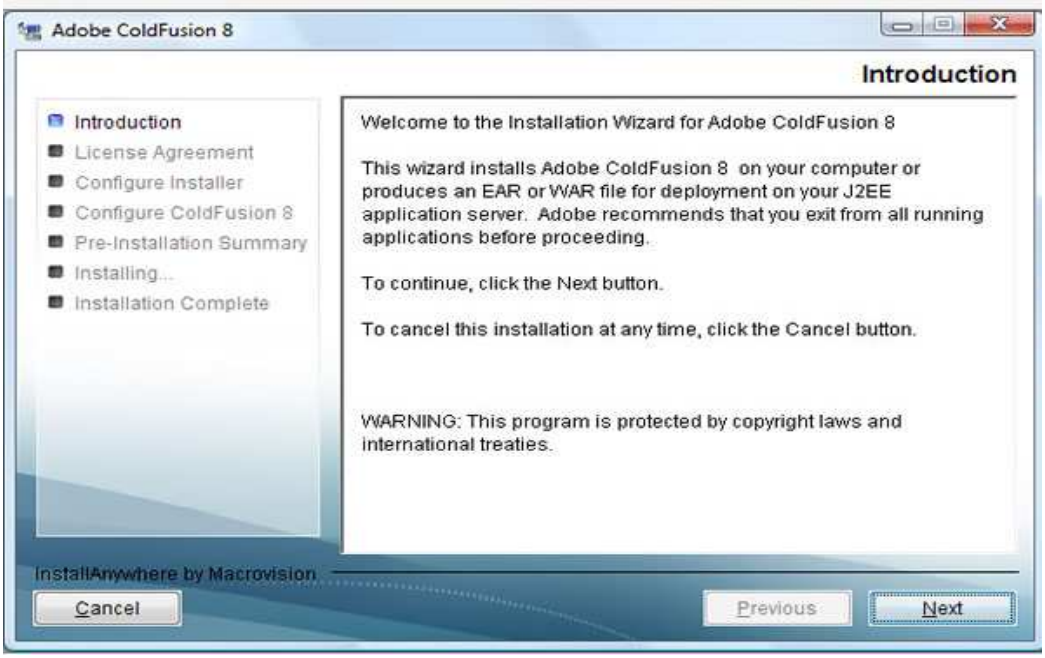
13. Ingresar una clave para los usuarios default de Oracle.

14. Reporte final de la configuración de nuestro servidor Oracle

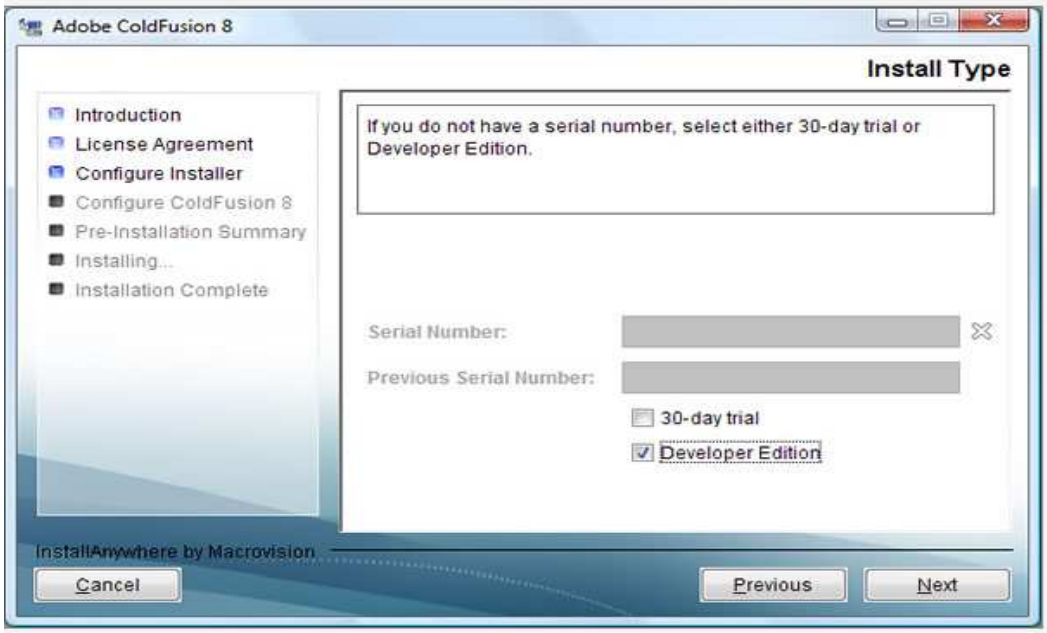


INSTALACIÓN DEL SERVIDOR COLDFUSION

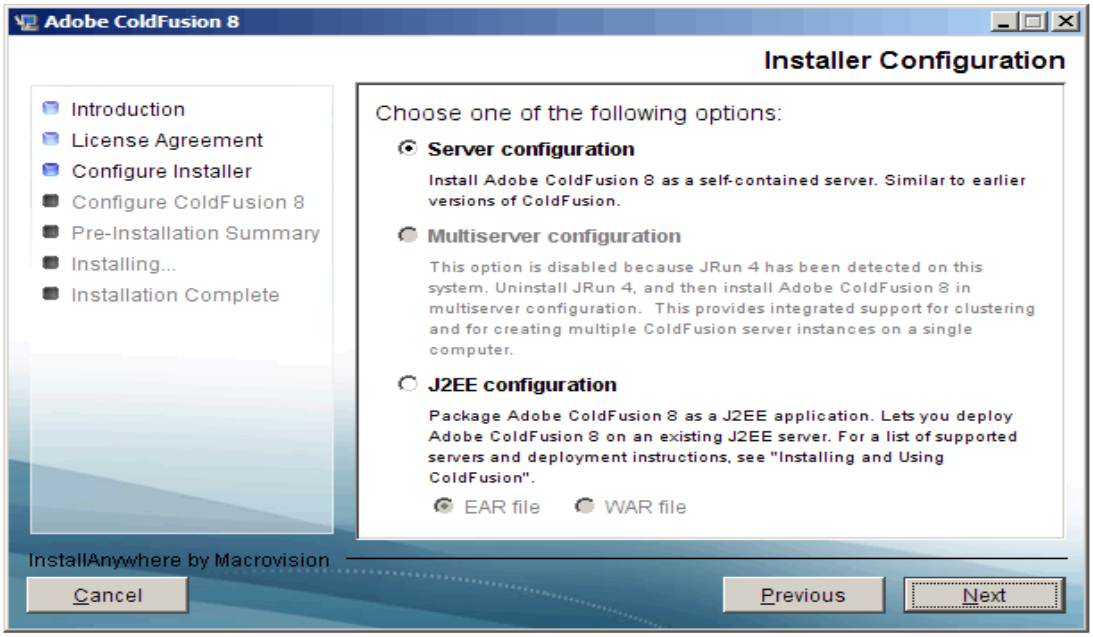
1.-Introducción sobre el software a instalar.



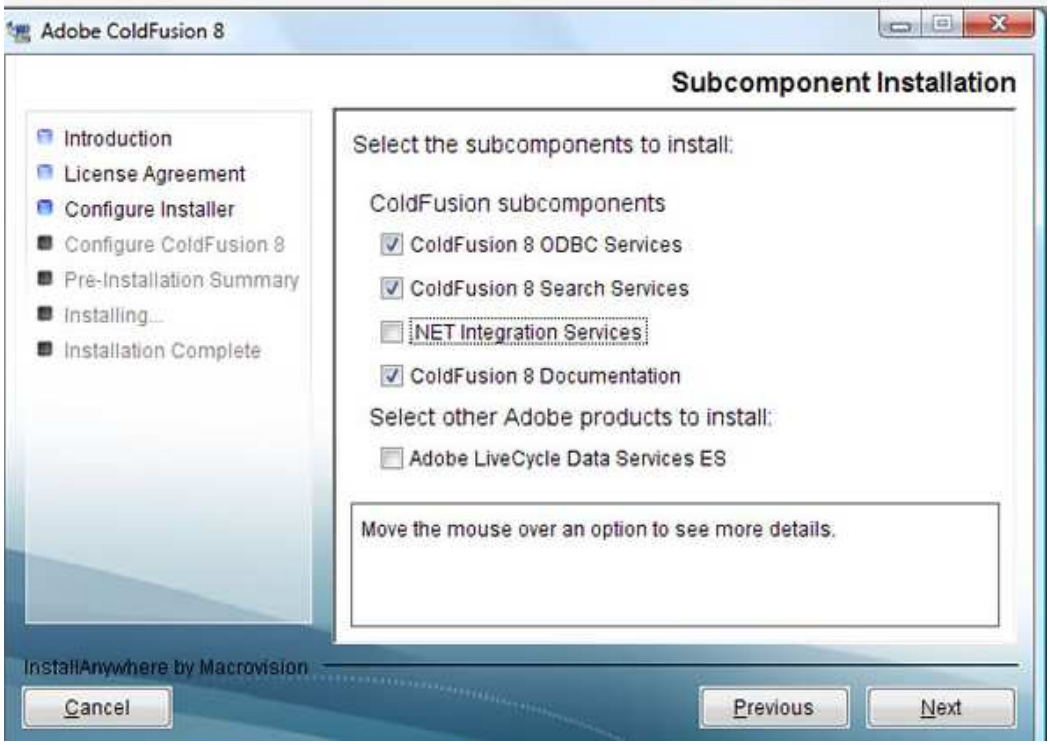
2.- Activar casilla ***Developer Edition***



3. Activar Server configuration



4.- Escoger todos los subcomponetes para configuración de base

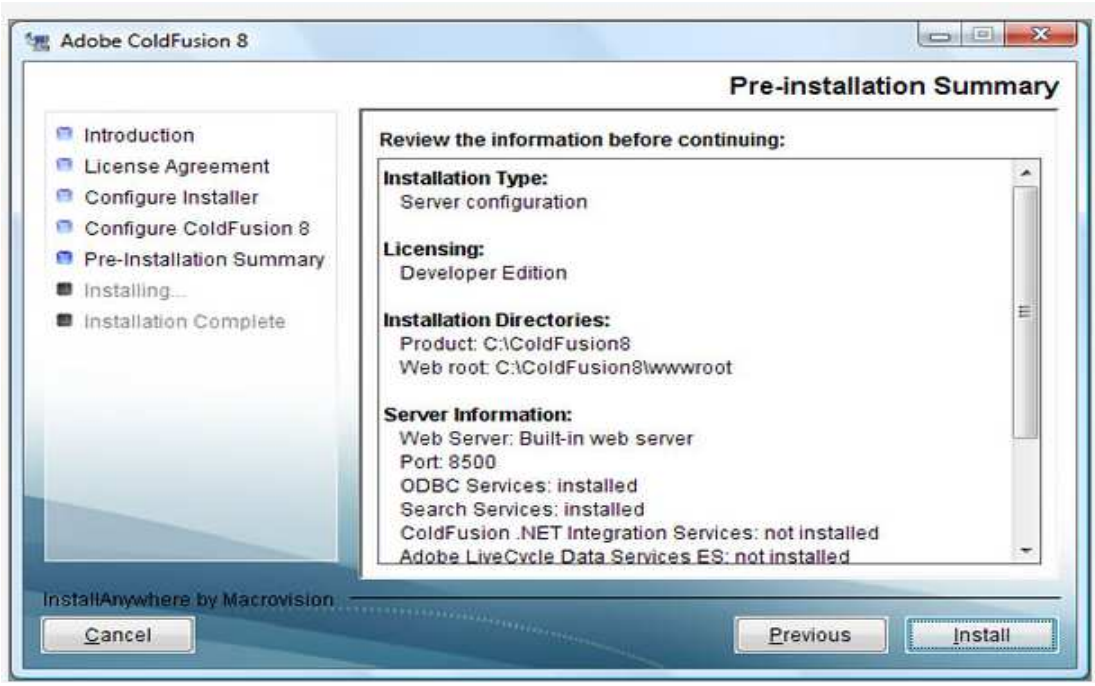


5.- Seleccionar directorio de Instalación

6.- Activar la siguiente selección.

7.- Ingresar password para el administrador de Coldfusion

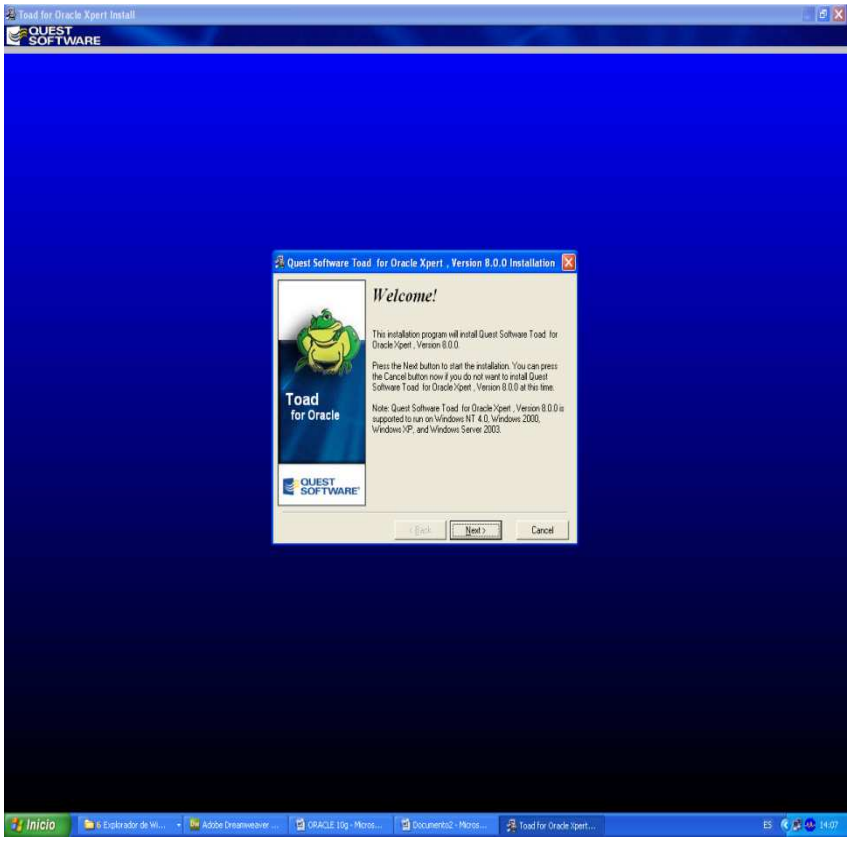
8. Confirmar datos de instalación y proceder a instalar



En este punto sale un mensaje indicando que instale la maquina virtual de java damos click en aceptar. Luego podemos instalarla

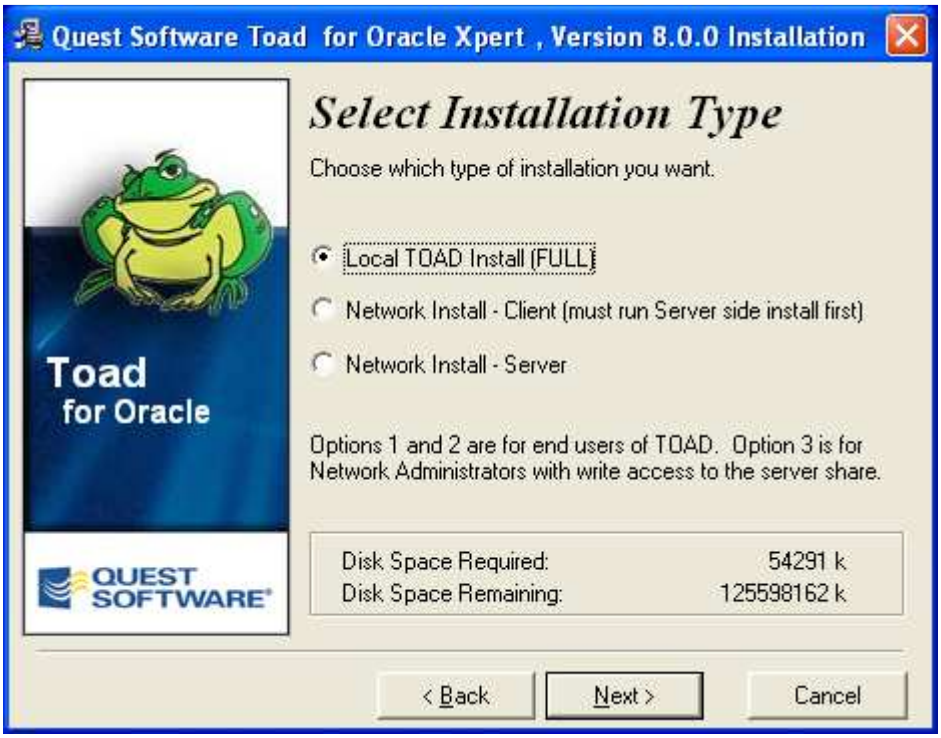
INSTALACION TOAD

1.-Presentación de interfaz



2.- Aceptar acuerdo de licencia software

3.-Elegir local toad para el tipo de instalación



4.- Seleccione directorio para la instalación

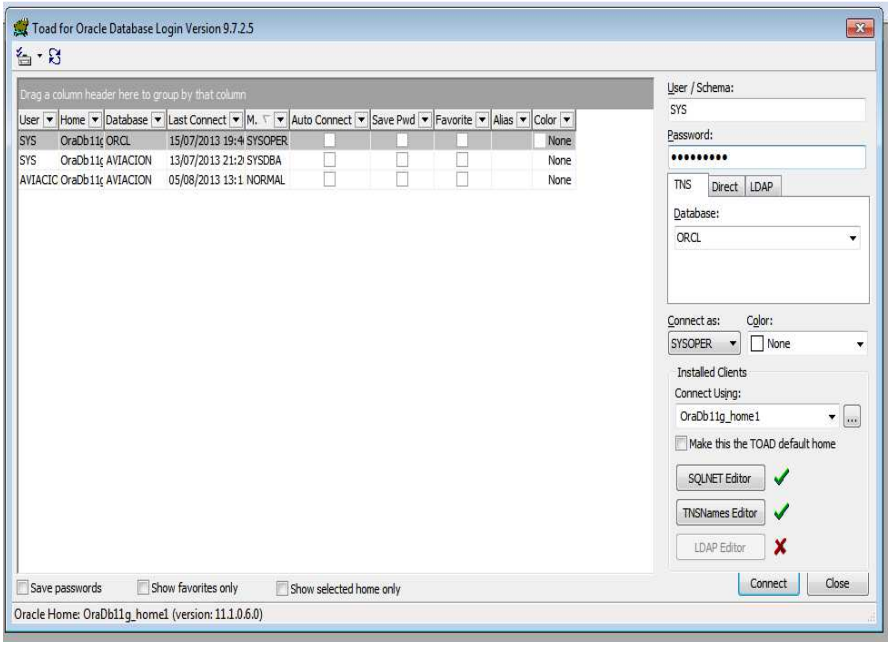
5.- Seleccione el directorio para backup

6.- Activar casilla Install unix job

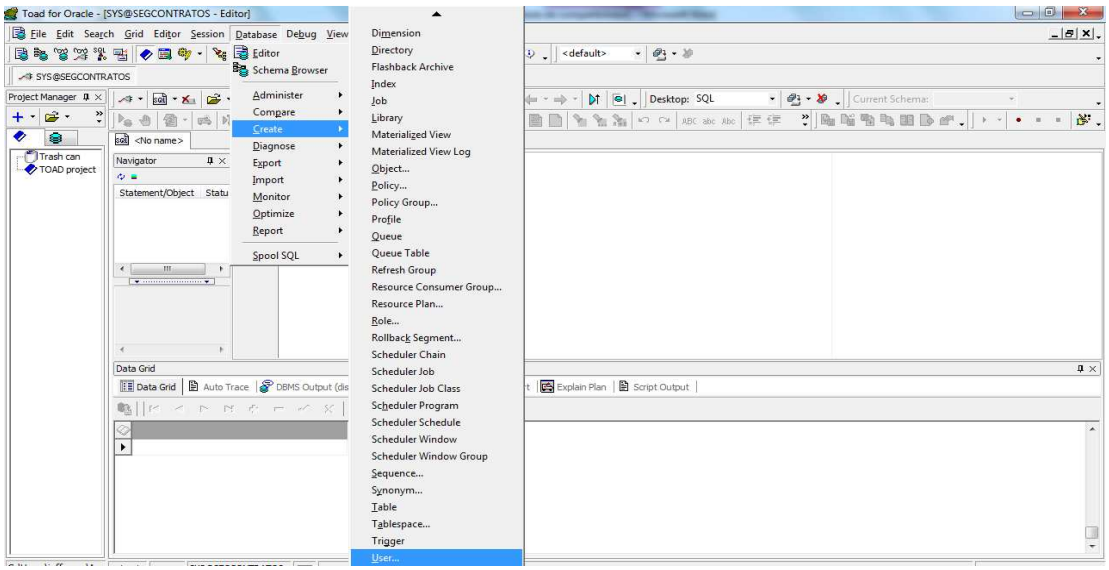
7.- Instalación completa

CONFIGURACIÓN DE TOAD

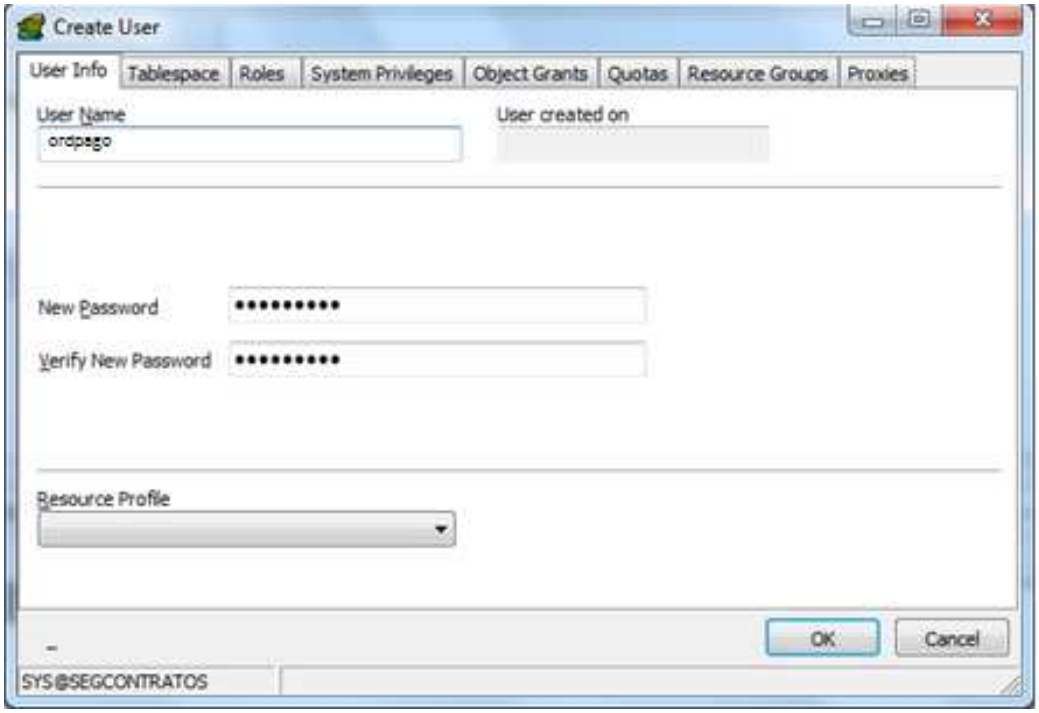
1.-Con la herramienta Toad for Oracle ingresamos a la base de datos orcl con el usuario SYS, y como Sysoper para crear un Schema y sobre ella crear nuestras tablas para el aplicativo aviacion.



2.-En la opción menú Database opción Create User, ingresar un nuevo usuario para crear las tablas del aplicativo.



3.-Ingresar el nombre del usuario o esquema y passsword



Asignamos al esquema a un*TABLESPACE.* Ejecutamos en el editor de SQL el siguiente script

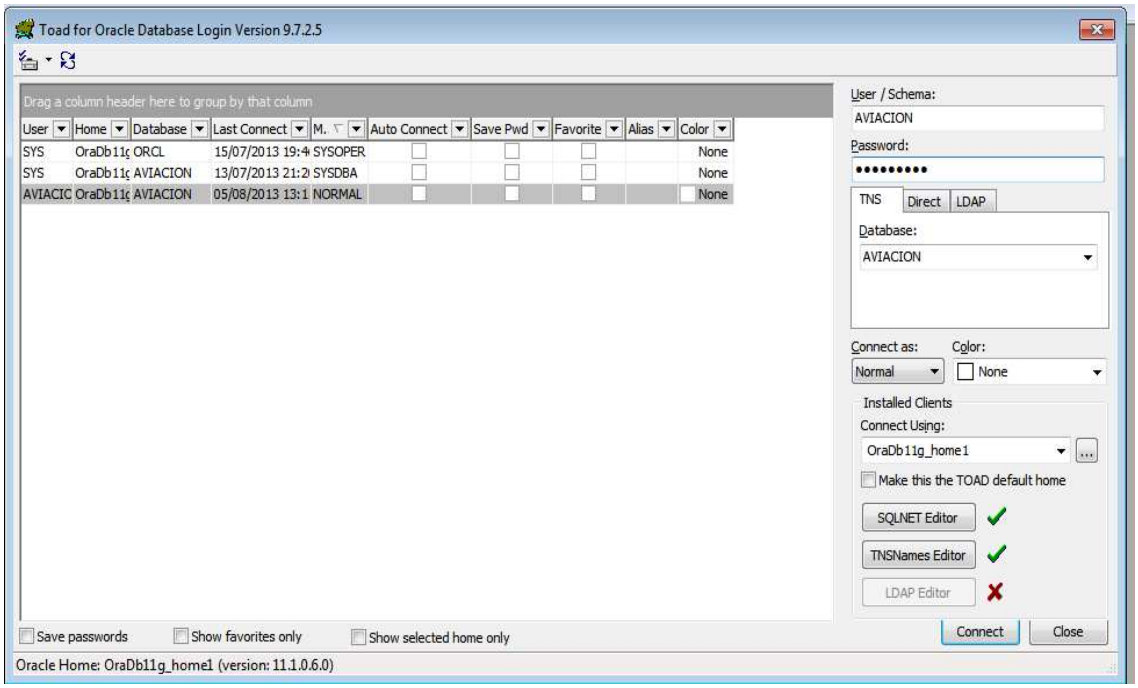
ALTER USER contratos DEFAULT TABLESPACE users.

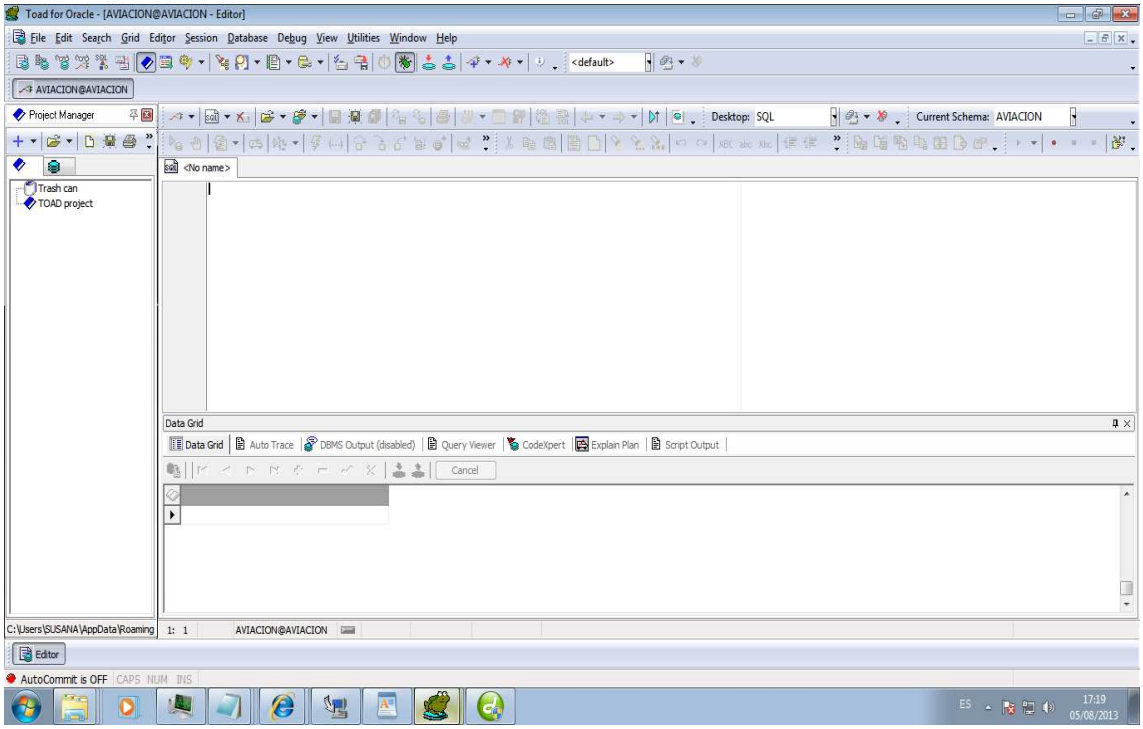
Asignar privilegios al esquema AVIACION En el sql editor ejecutamos el siguiente script.

GRANT ALL PRIVILEGES to AVIACION

Salir de la sesión SYS e ingresar con el usuario Contratos para crear las tablas del aplicativo.

4.- Ingresamos a nuestra base de datos



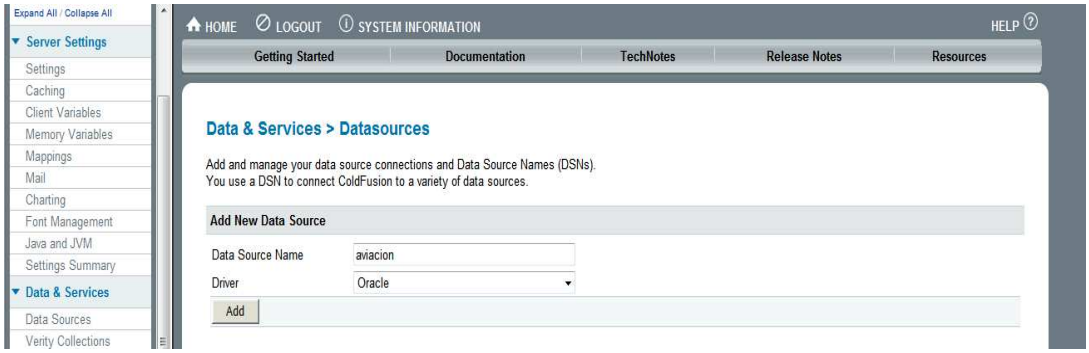


CONFIGURACION COLDFUSION

Ingresamos al administrador de coldfusion en la siguiente dirección:

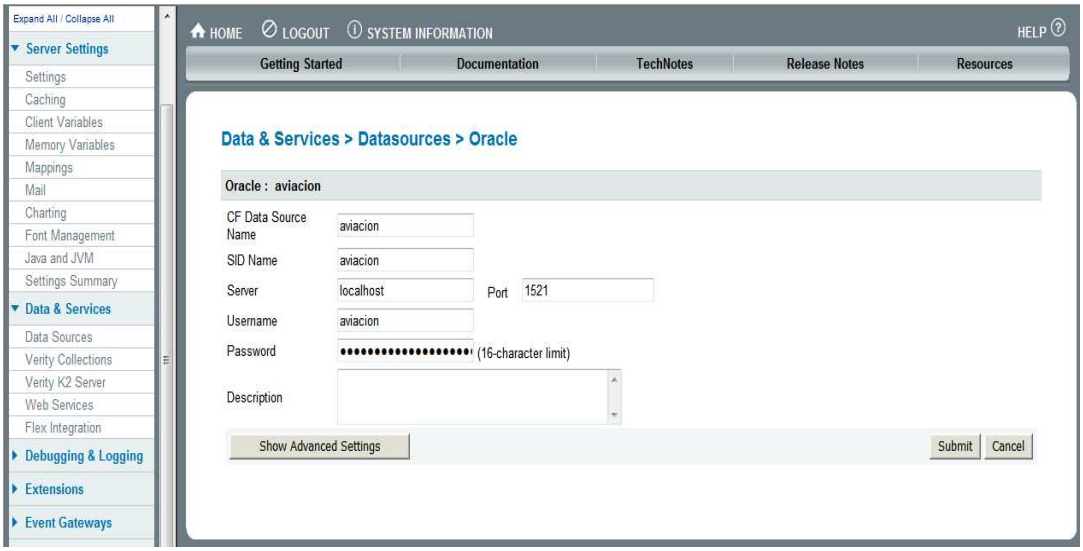
1.-http://127.0.0.1:8500/CFIDE/administrator/ , en la pantalla ingresamos el password que se asignó para el administrador al momento de la instalación.

2.-Seleccionar la opcion Data Source e ingresamos nombre de base .

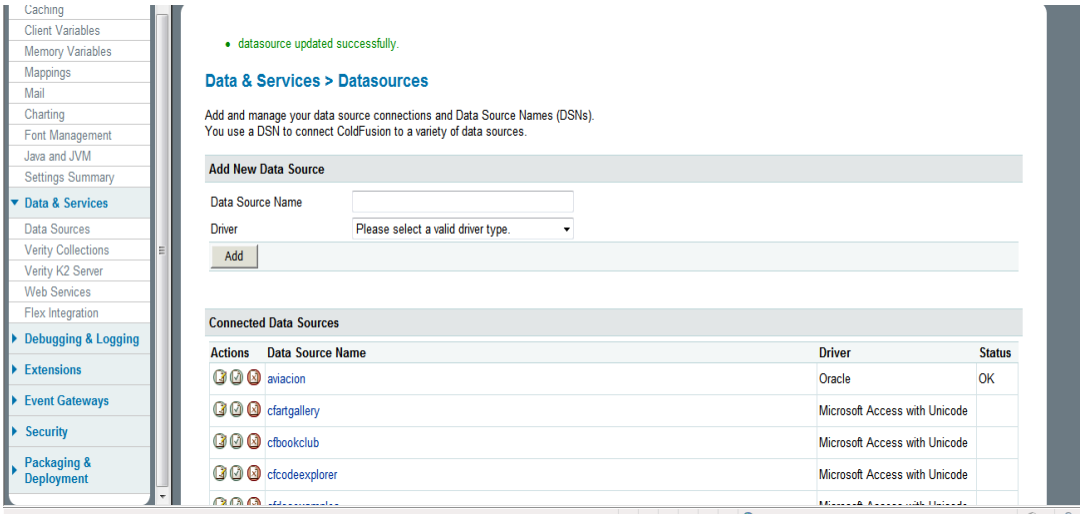


En esta pantalla ingresamos un nombre para la base de datos soporte, pulsamo añadir

3.- Ingresamos datos de la base.

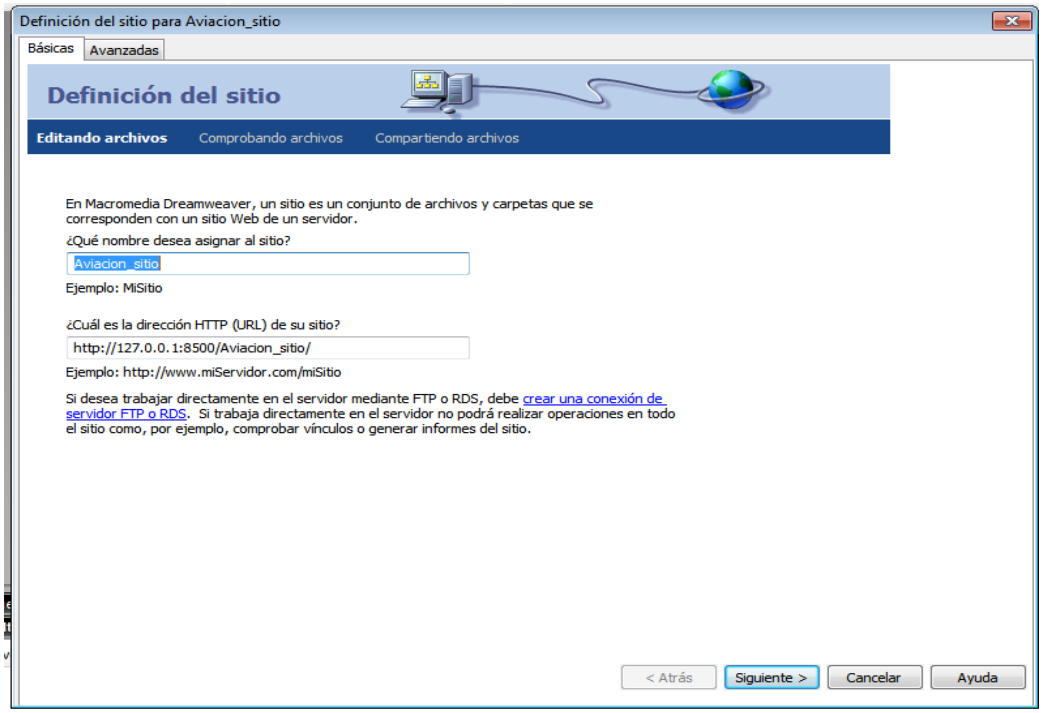


4.-Presionar *Submit.y* la conexión fue exitosa se mostrará un mensaje en color verde en la parte superior de página.

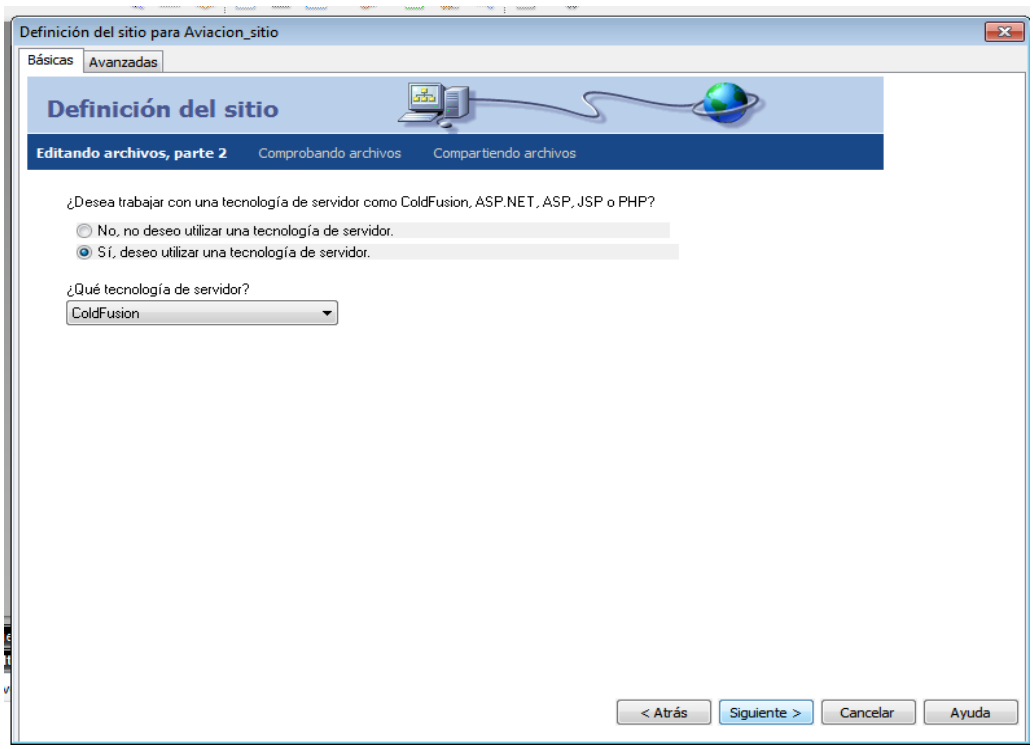


CONFIGURACIÓN SITIO EN DREAMWEAVER.

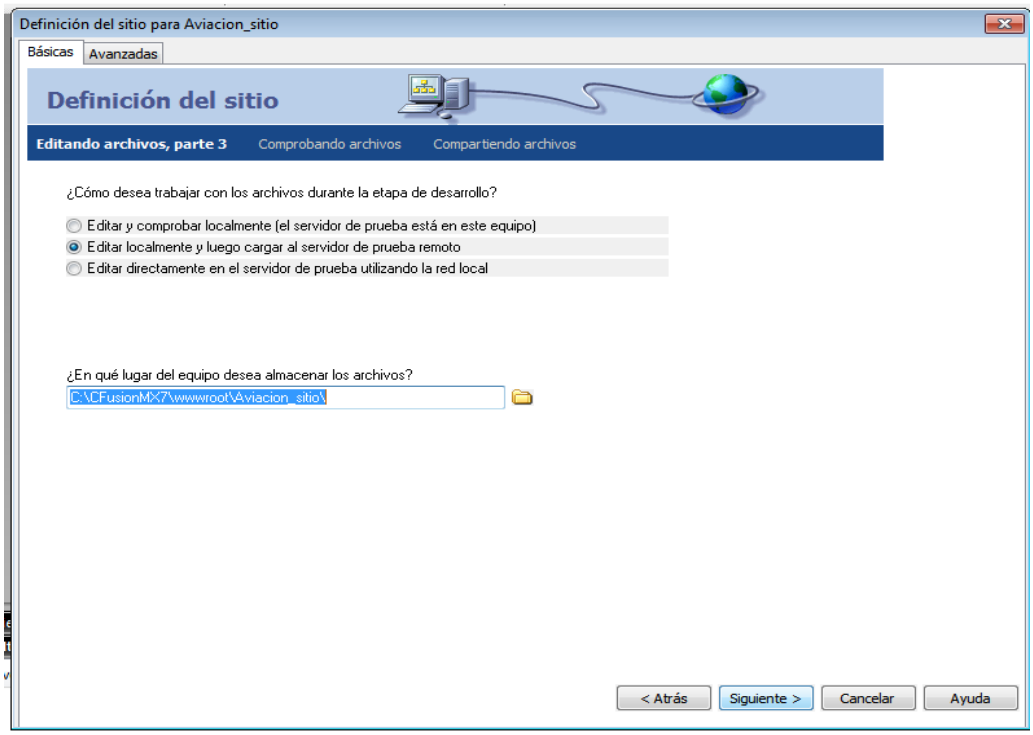
1.- Ingresamos nombre y dirección HTTP



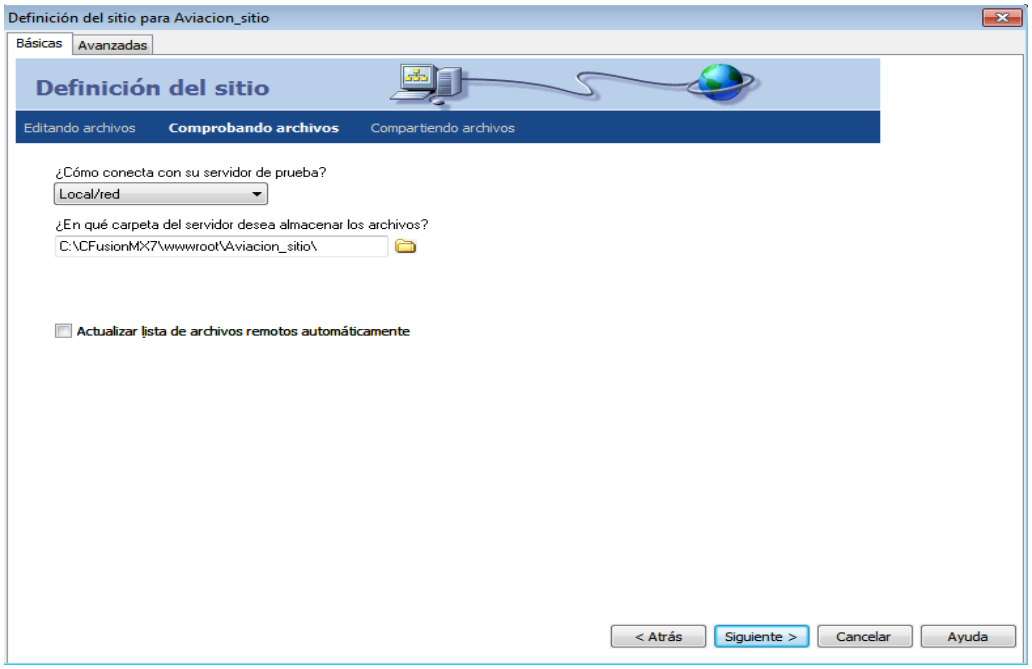
2.-Selecionamos utilizar tecnología de servidor



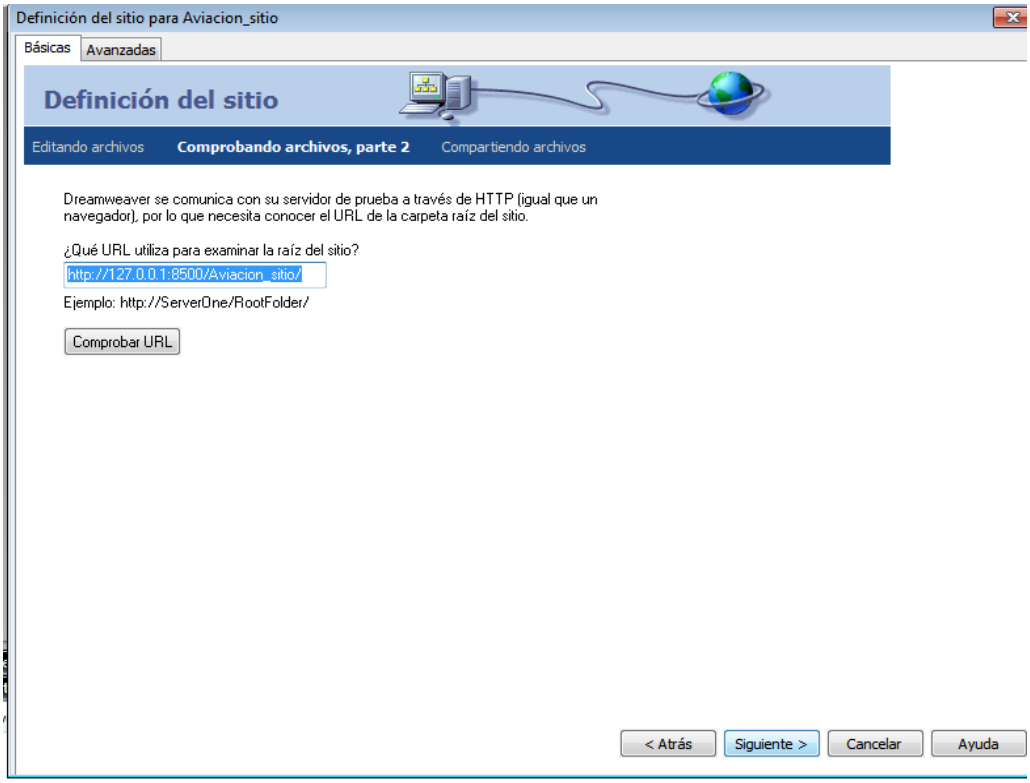
3.-Seleccionar como deseamos trabajar durante el desarrollo.



4.- Elegir el sitio de conexión y la dirección en donde archivaremos nuestro sitio



5.-Ingresar la dirección URL



Revisar configuración y finalizar.

Para ingresar al aplicativo aviacion, ingresar a la siguiente dirección: http://localhost:8500/aviacion_sitio

MANUAL DEL USUARIO

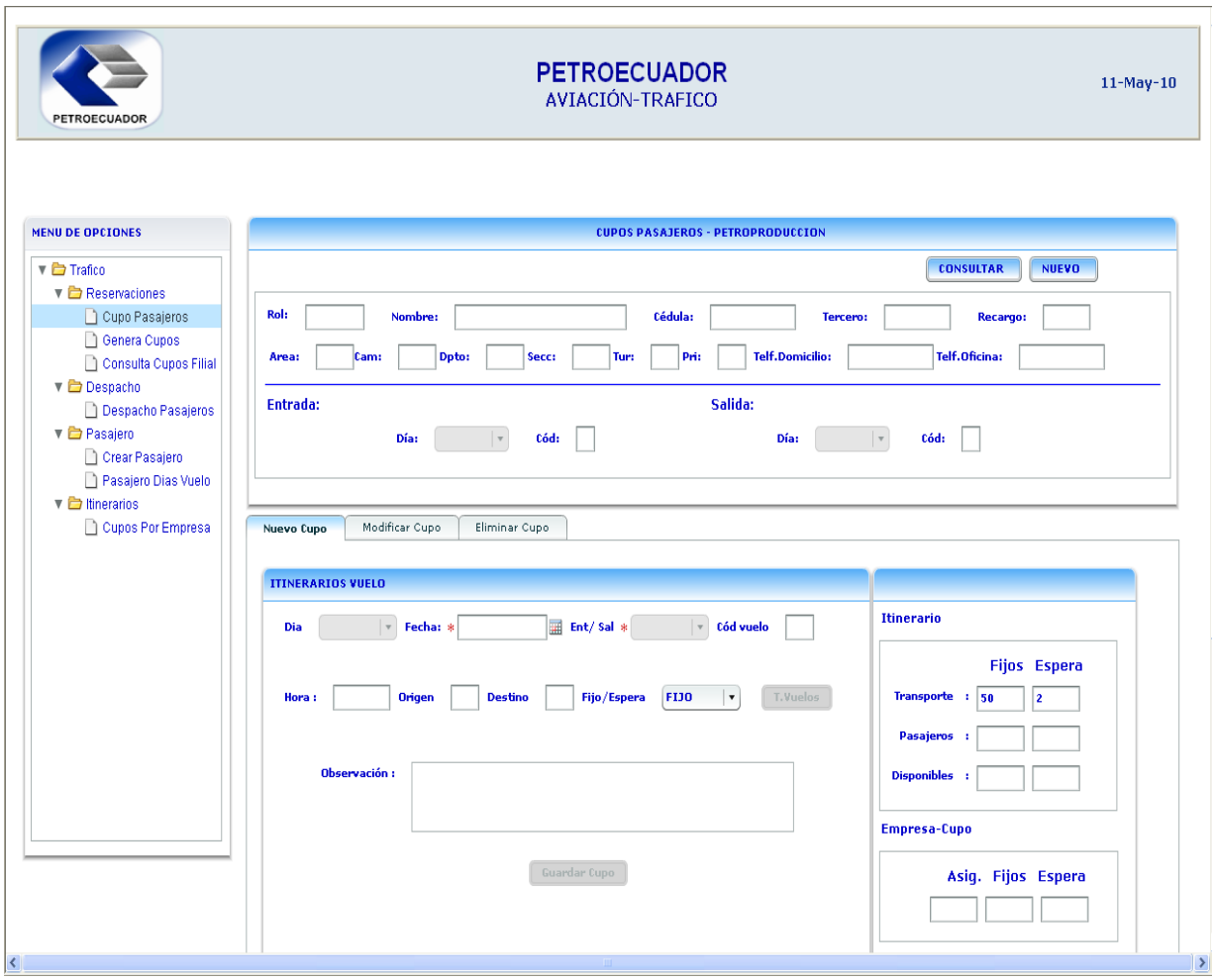
INGRESAR A LA SIGUIENTE DIRECCIÓN

http://localhost:8500/aviacion

En la pantalla se debe ingresar usuario y contraseña asignados para la intranet de EP Petroecuador; dependiendo de la clave que se ingrese se tendrá privilegios sobre los diferentes módulos.

1.-Reservaciones (Cupos Pasajeros).-

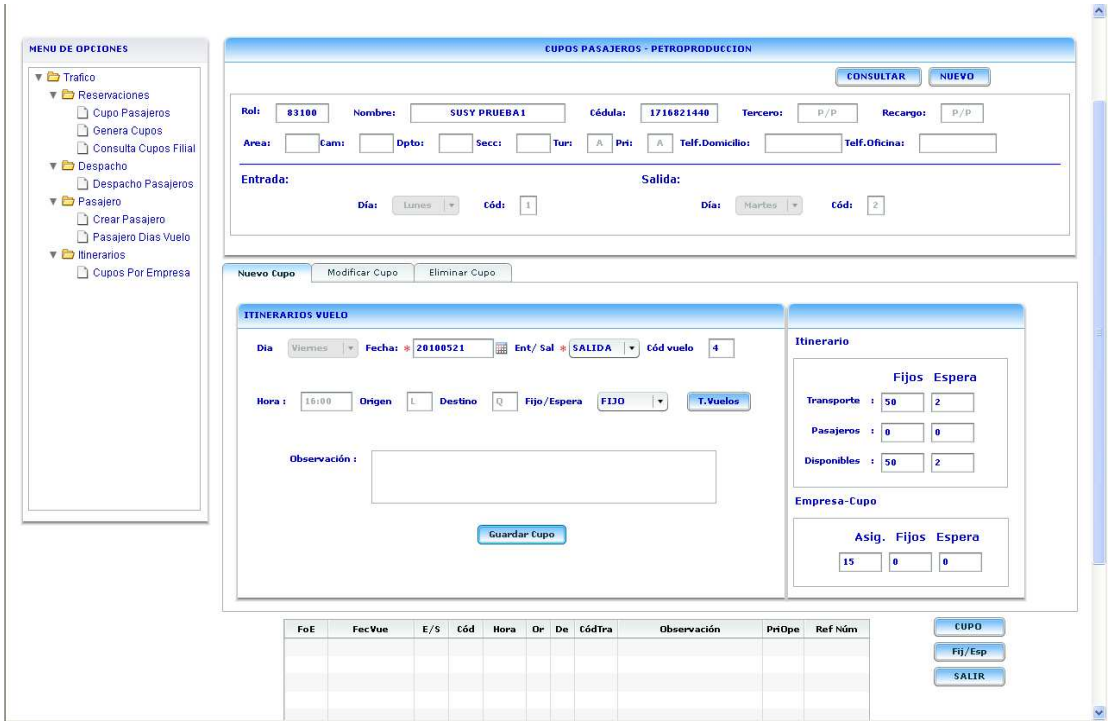
La interfaz nos ayuda a la creación , modificación, e eliminación de cupos por pasajero de acuerdo a cada subgerencia.

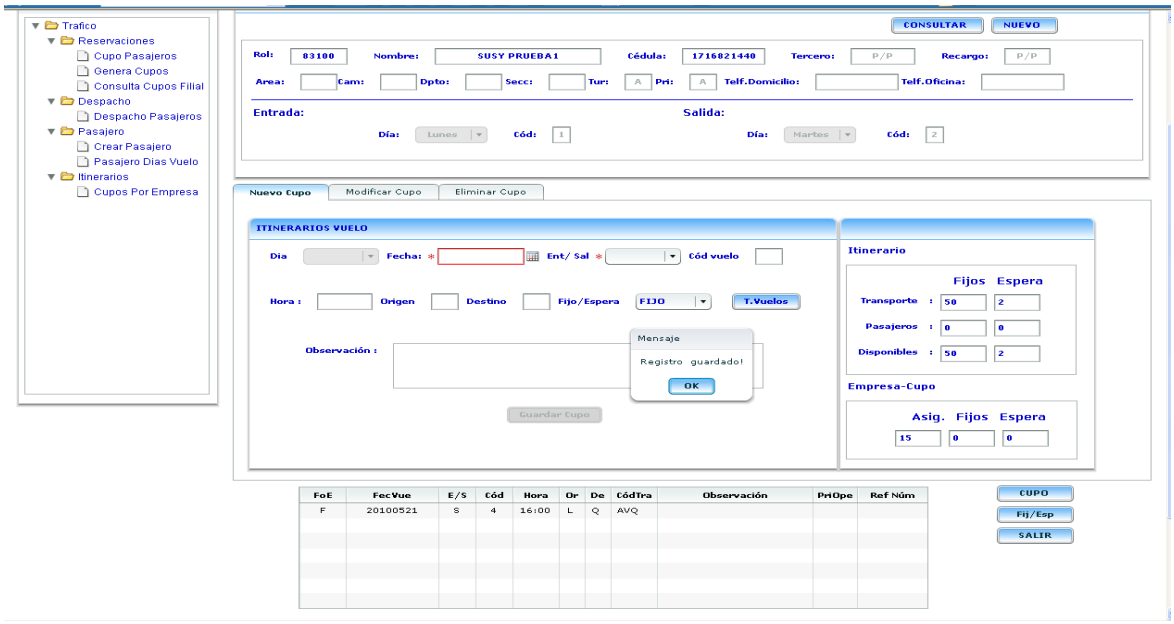


Para la consulta lo podemos realizar mediante consulta individual por rol, nombre o cedula.

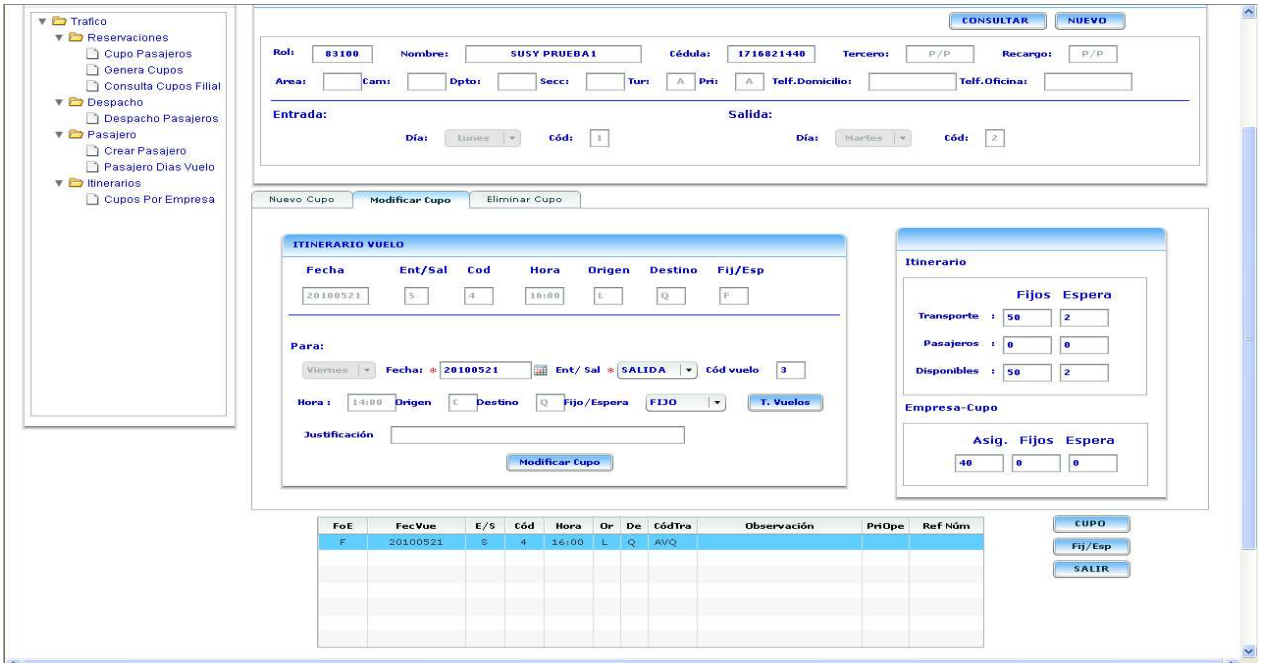
Para pasajeros que no existen despliega el mensaje

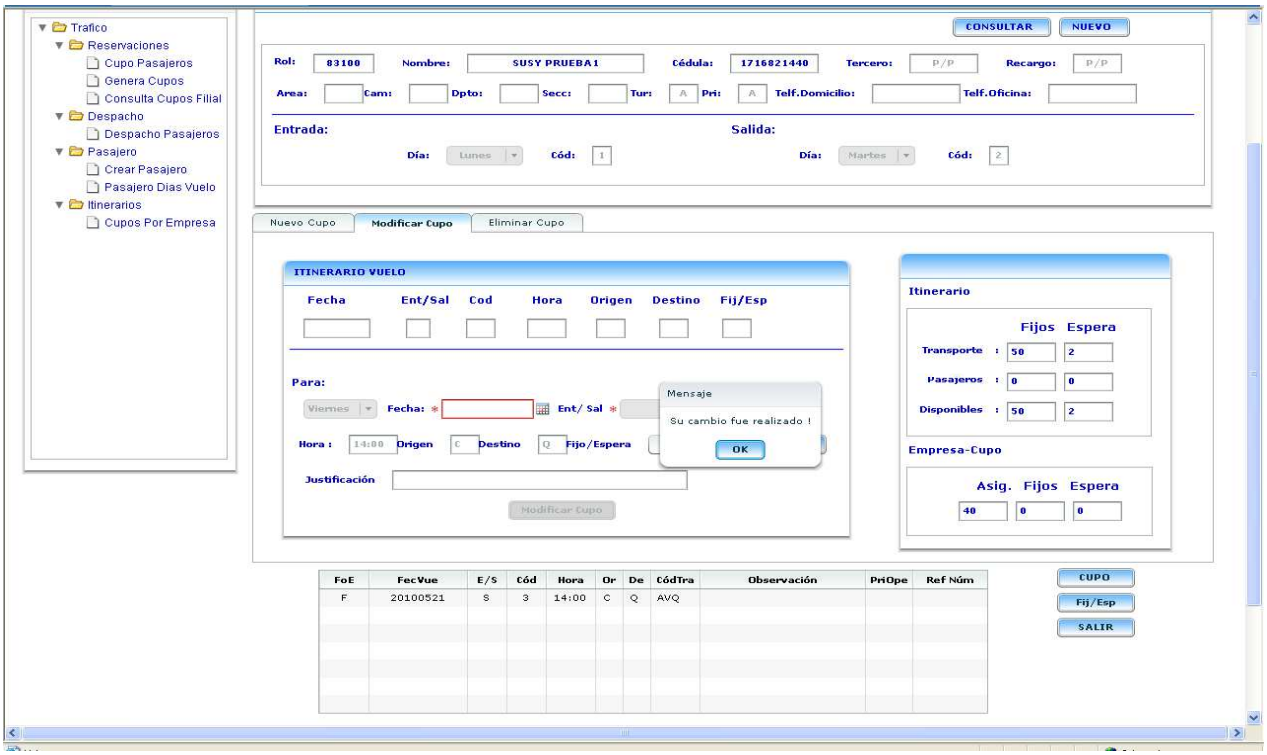
Guardar cupo



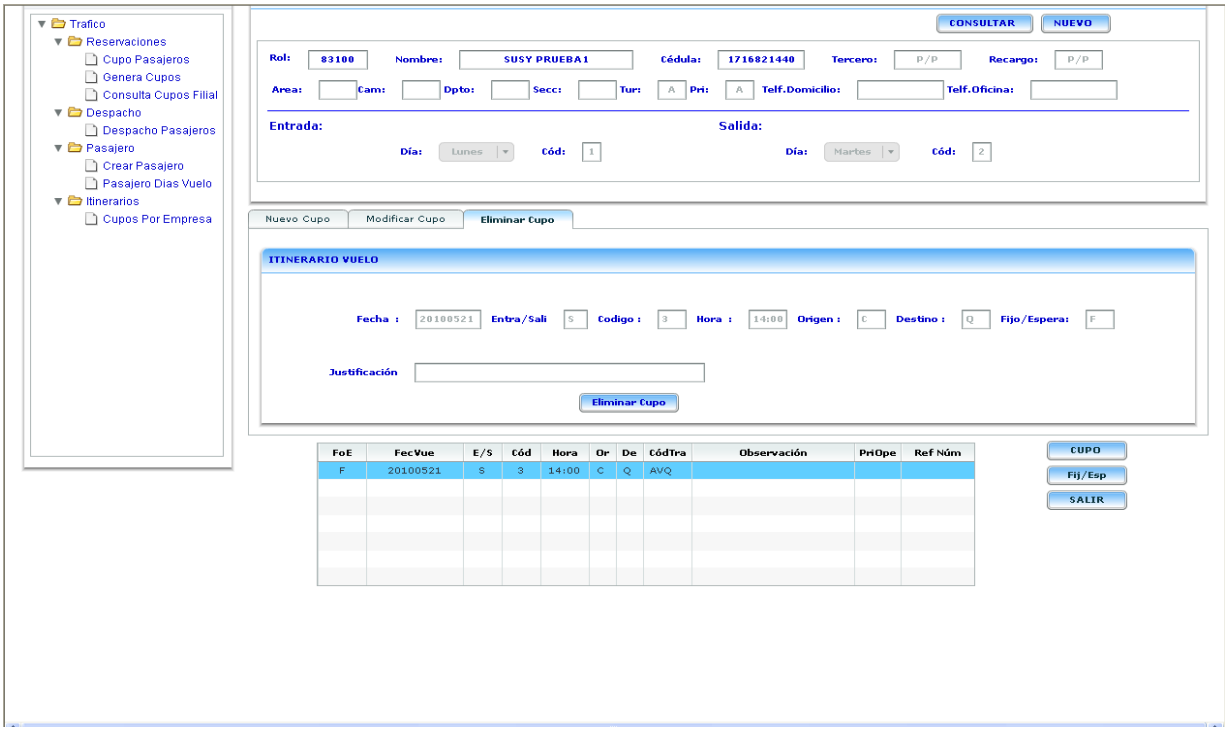


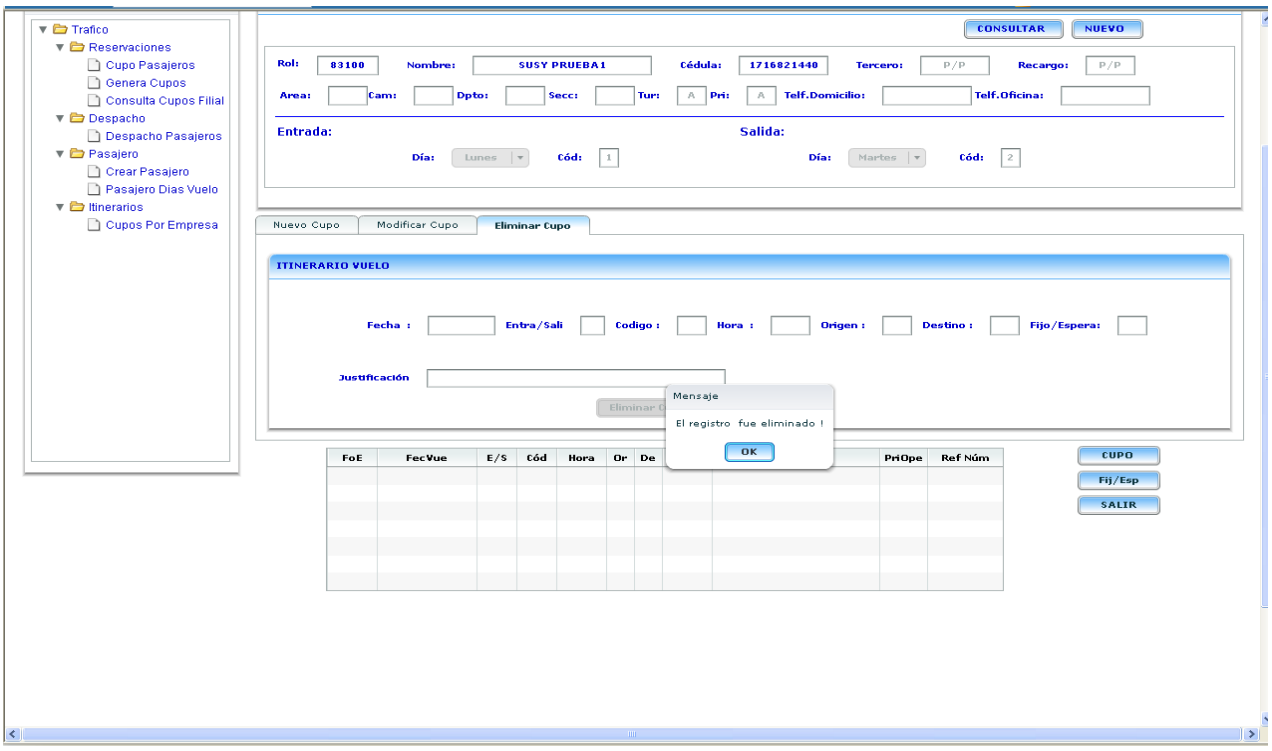
Modificar cupo





Eliminar cupo



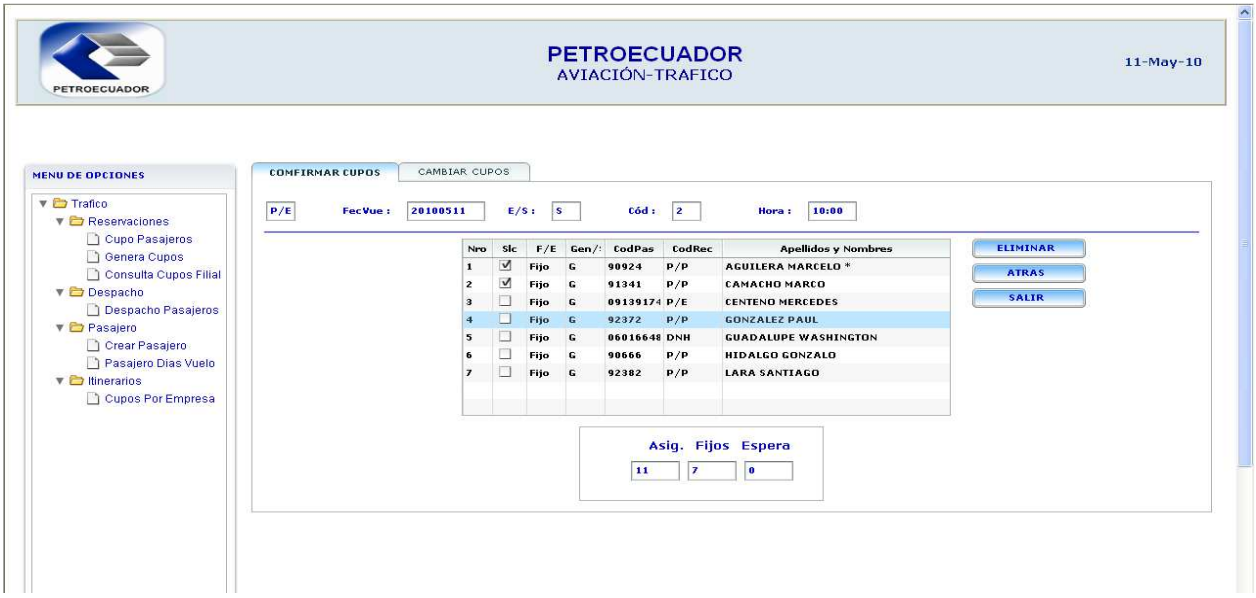


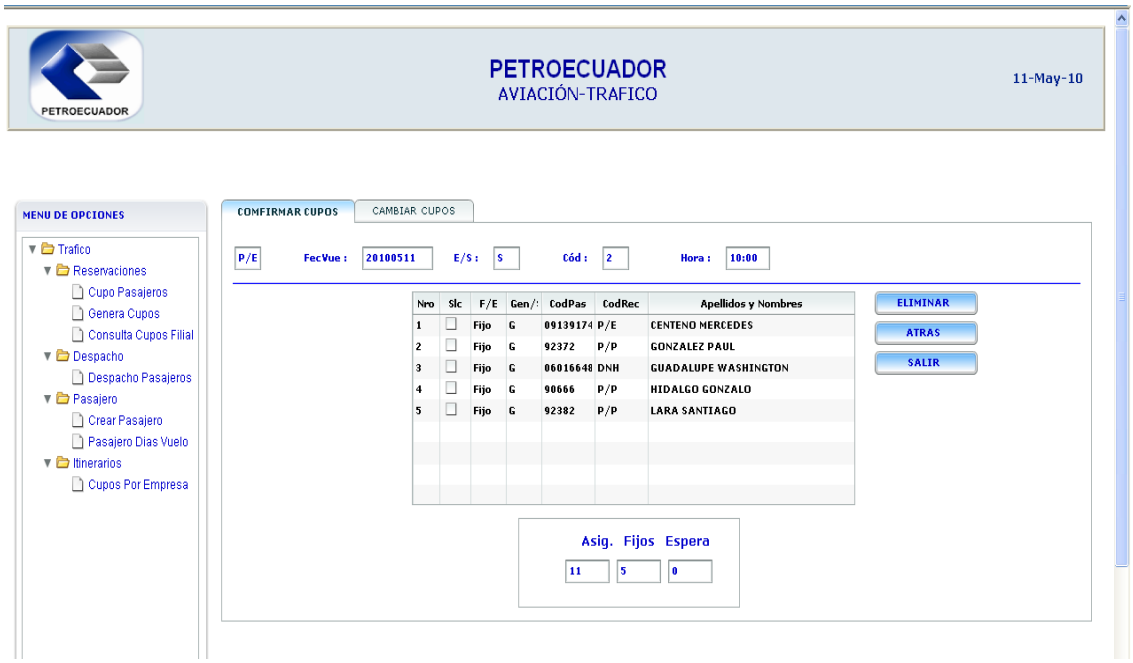
2.-Reservaciones (Genera Cupos).-

Interfaz encargada de eliminación y cambio de varios pasajeros.

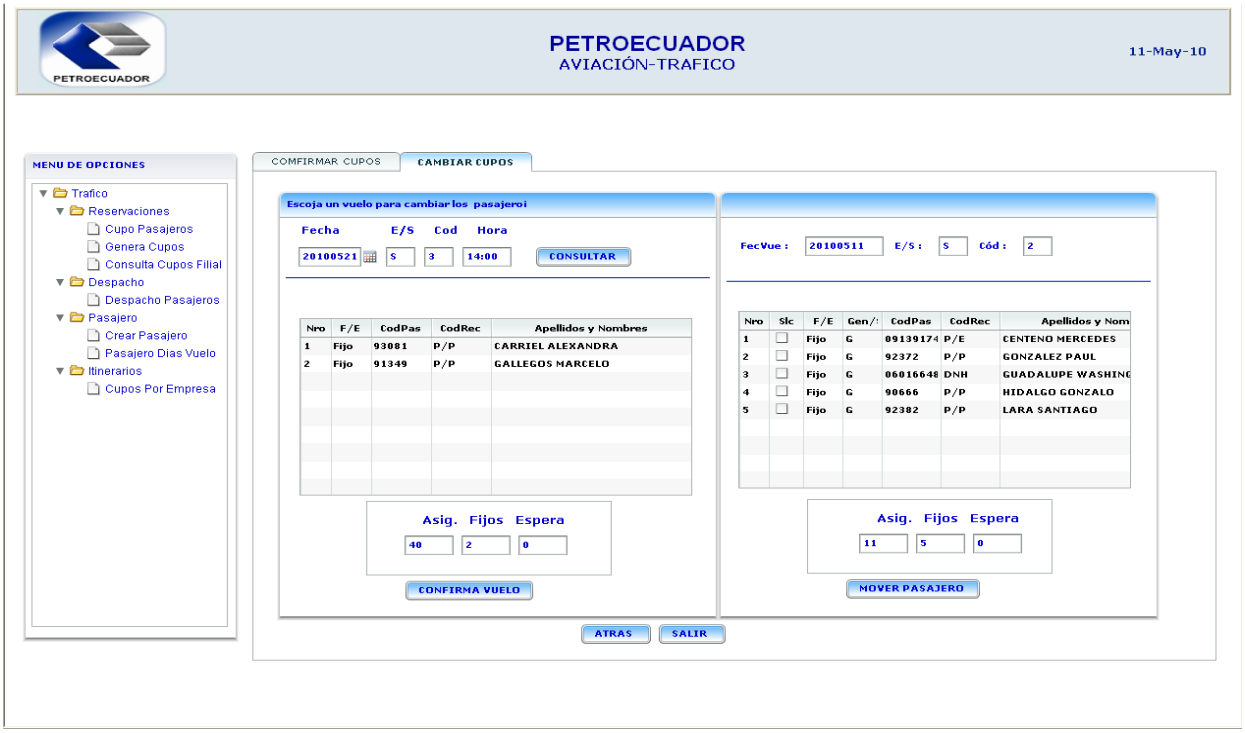
Escogemos un vuelo y pulsamos el botón (Confirmar)

Eliminar Pasajeros por vuelo

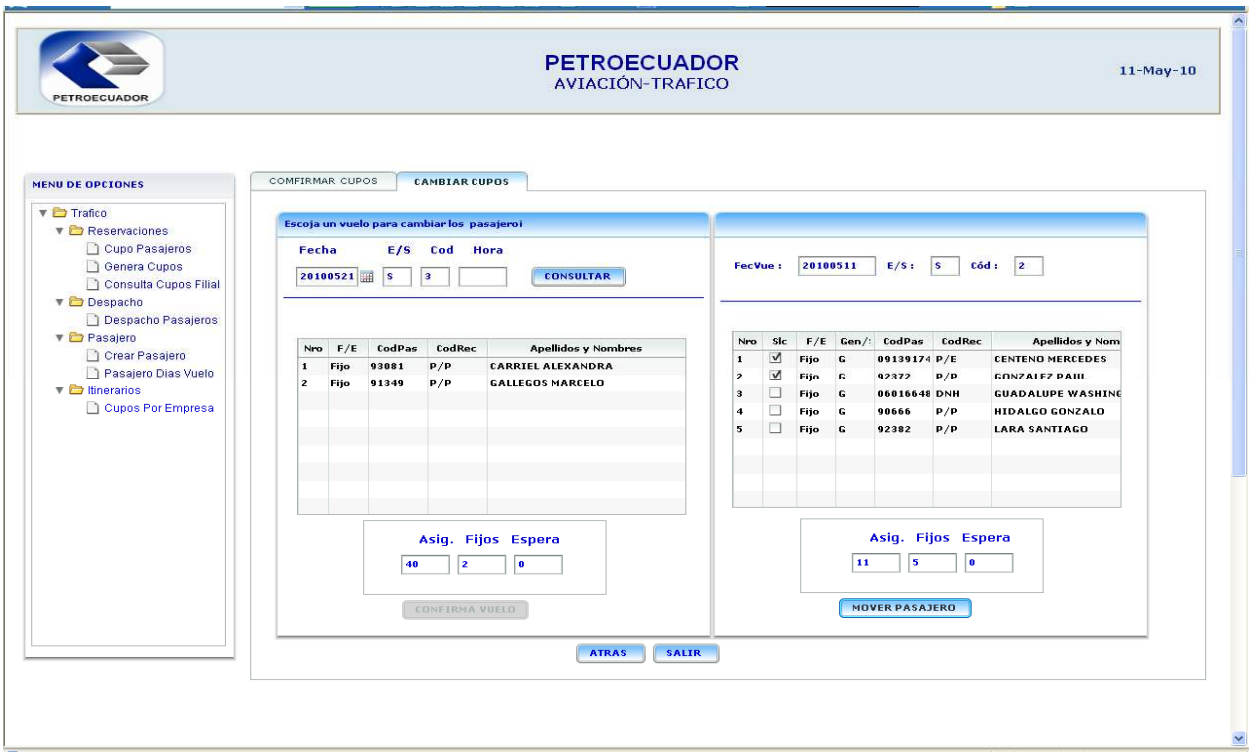




Mover pasajeros un vuelo a otro.(Consultamos el vuelo a donde queremos que los pasajeros cambien) y pulsamos el botón (confirmar)

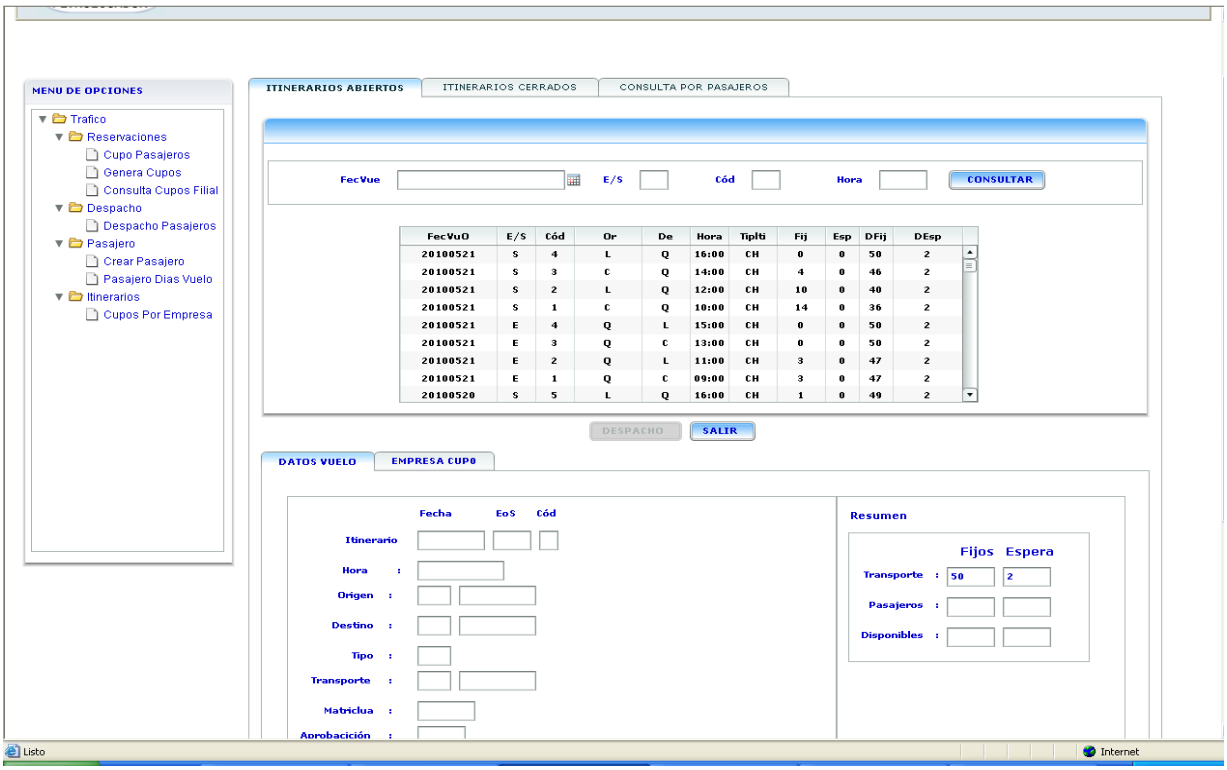


Una vez hecho esto escogemos los pasajeros que deseamos cambiar de vuelo y pulsamos el botón (Mover Pasajero)

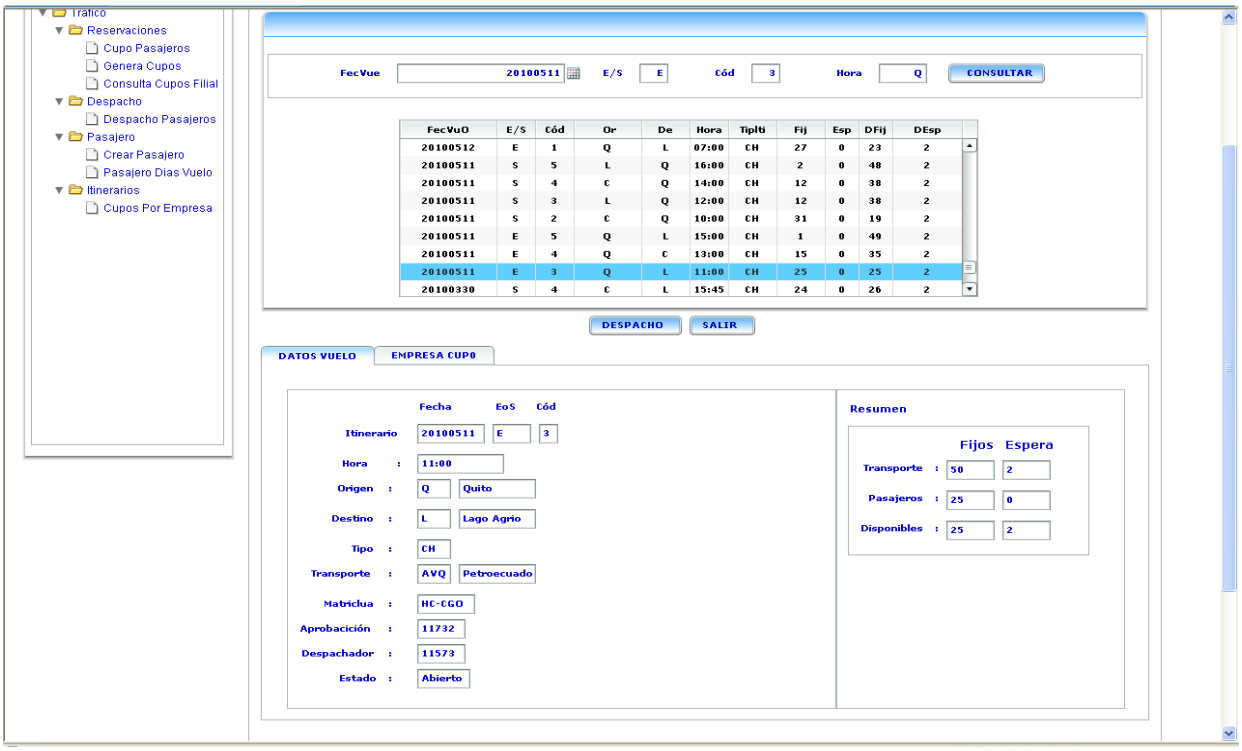


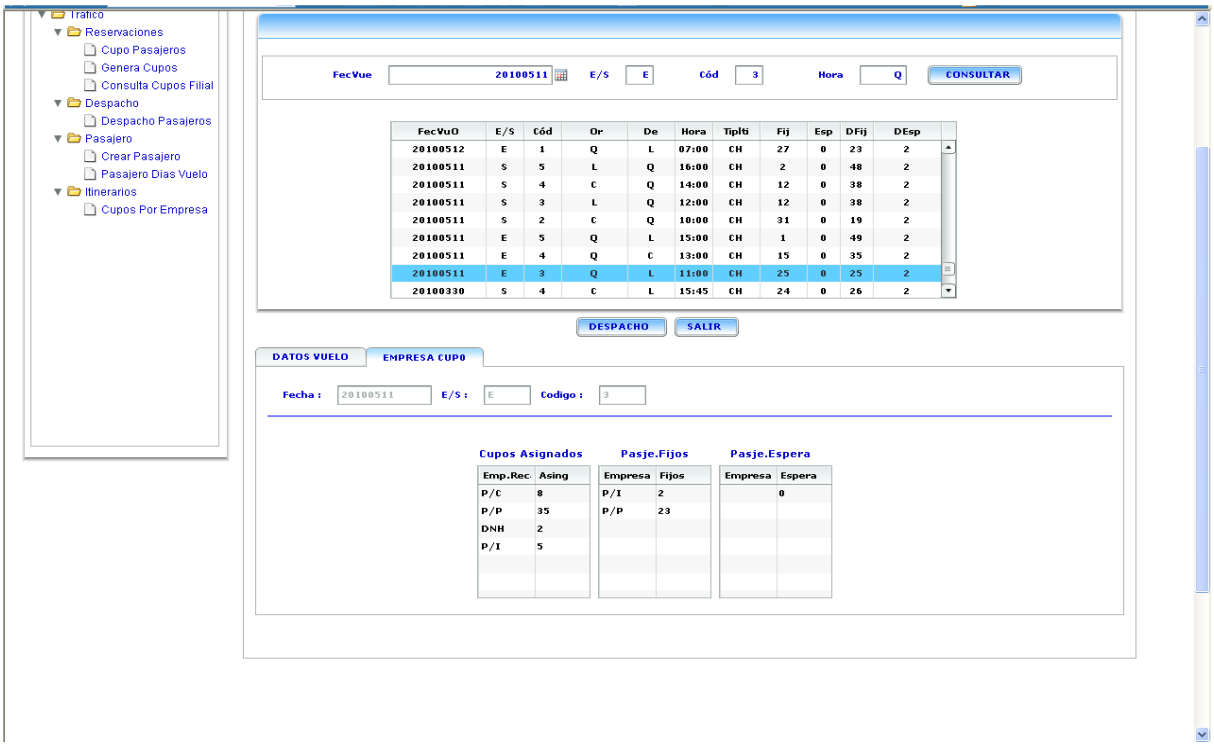
3.-Despacho (Despacho Pasajeros).-

Itinerarios pasajeros.- Vuelos abiertos para el despacho de pasajeros.



Encojemos un Itinerario, nos despliega los datos y cupos por empresa de cada Itinerario.

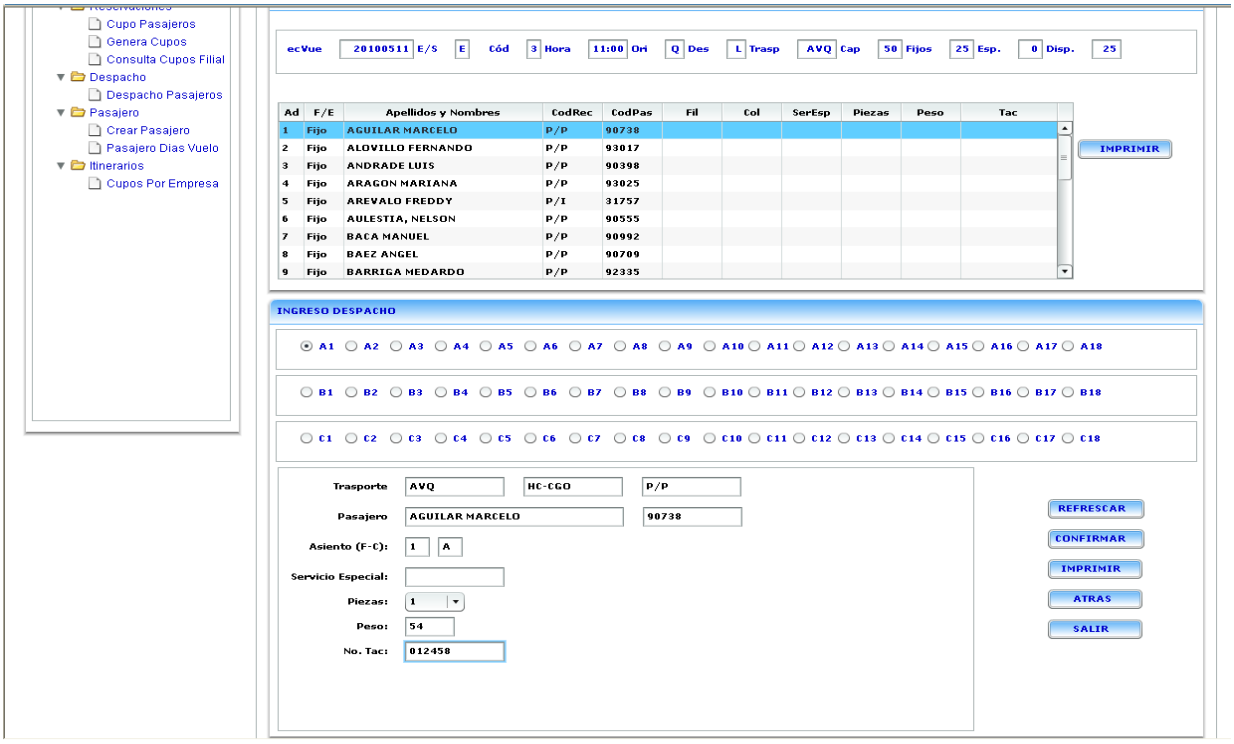




Para el ingreso del despacho, pulsamos el botón (despacho)

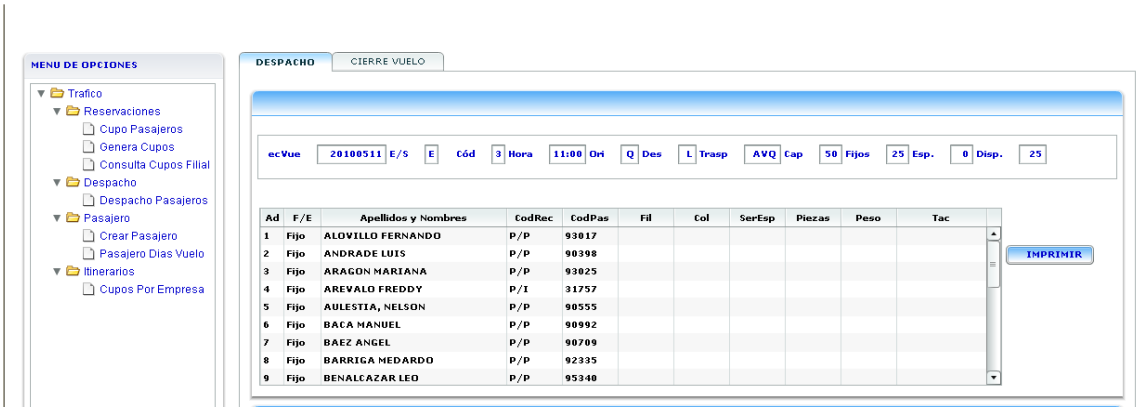
Pestaña Despacho.- Visualiza los pasajeros (no despachados) para su respectivo despacho.

Escogemos un pasajero y ingresamos todos sus datos de despacho.



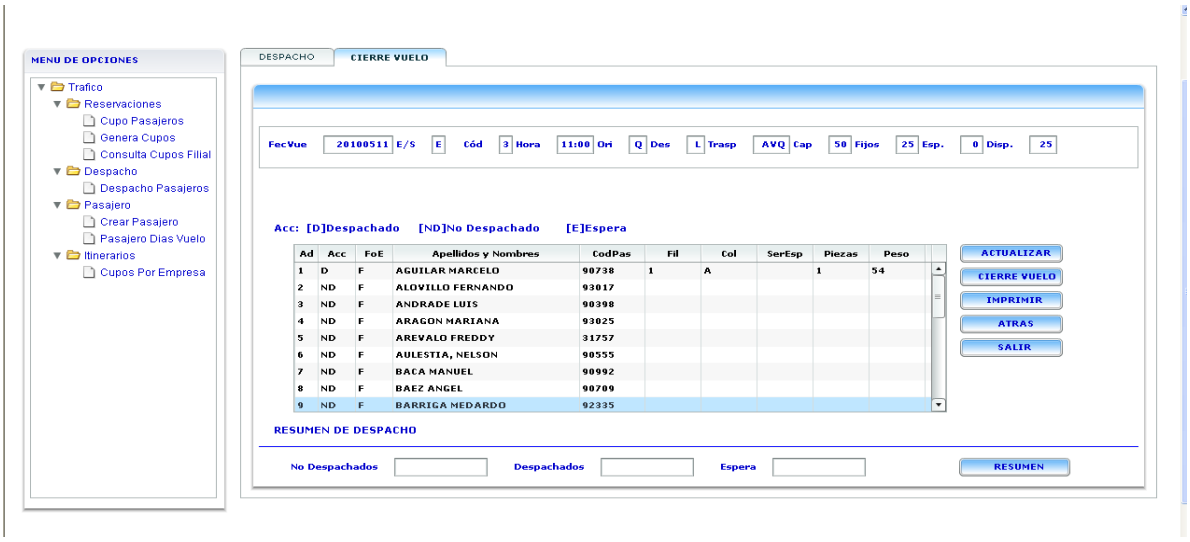
Pulsamos los botones (Confirmar y Refrescar)

Notaremos que nuestro pasajero despachado ya no se encuentra en lista.

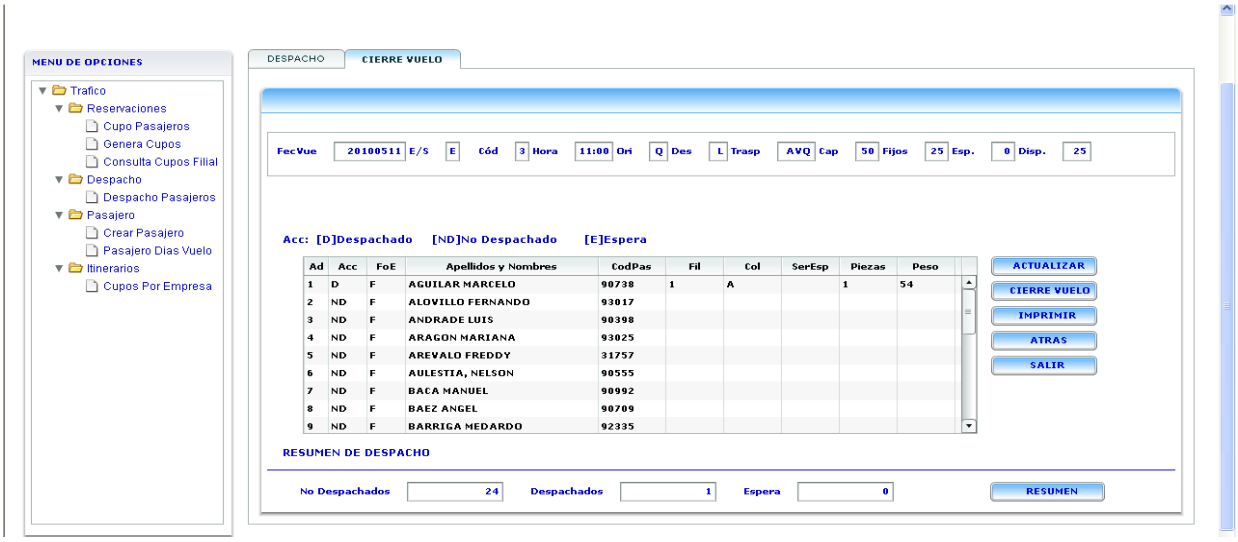


Luego de actualizar todos los datos de los pasajeros nos dirigimos a la pestaña de Cierre Vuelo.

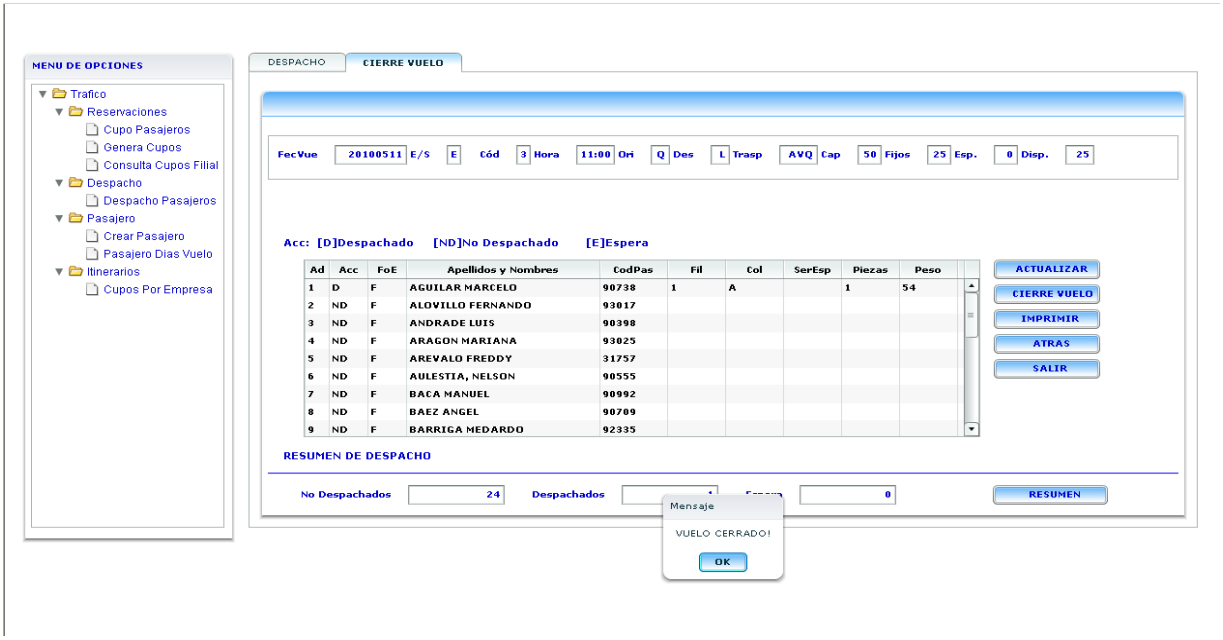
En donde podemos visualizar # pasajeros despachado y no despachados.



Pulsando el botón (Resumen)

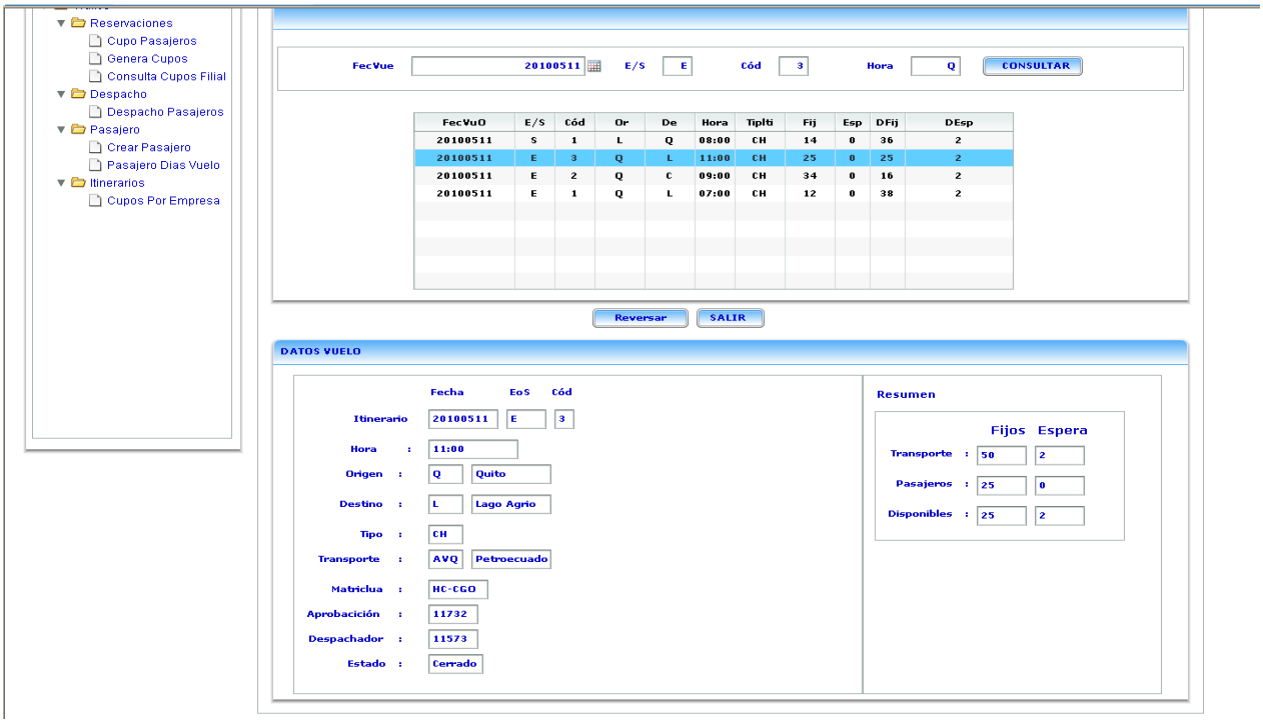


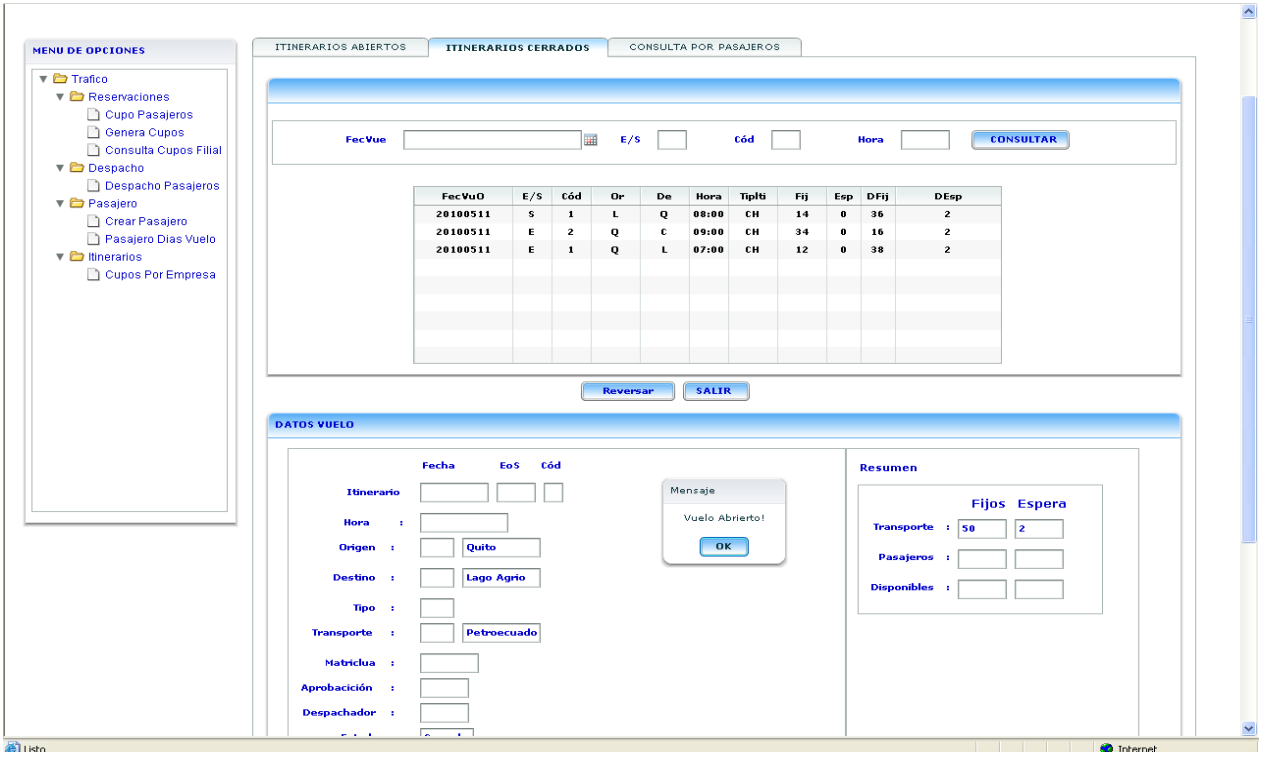
Botón (Cierre Vuelo) Cierra el vuelo una vez despachado los pasajeros.



Itinerarios Cerrados:_

Consultamos los vuelos cerrados que existen, podemos de igual manera visualizar datos y también reversar el vuelo.





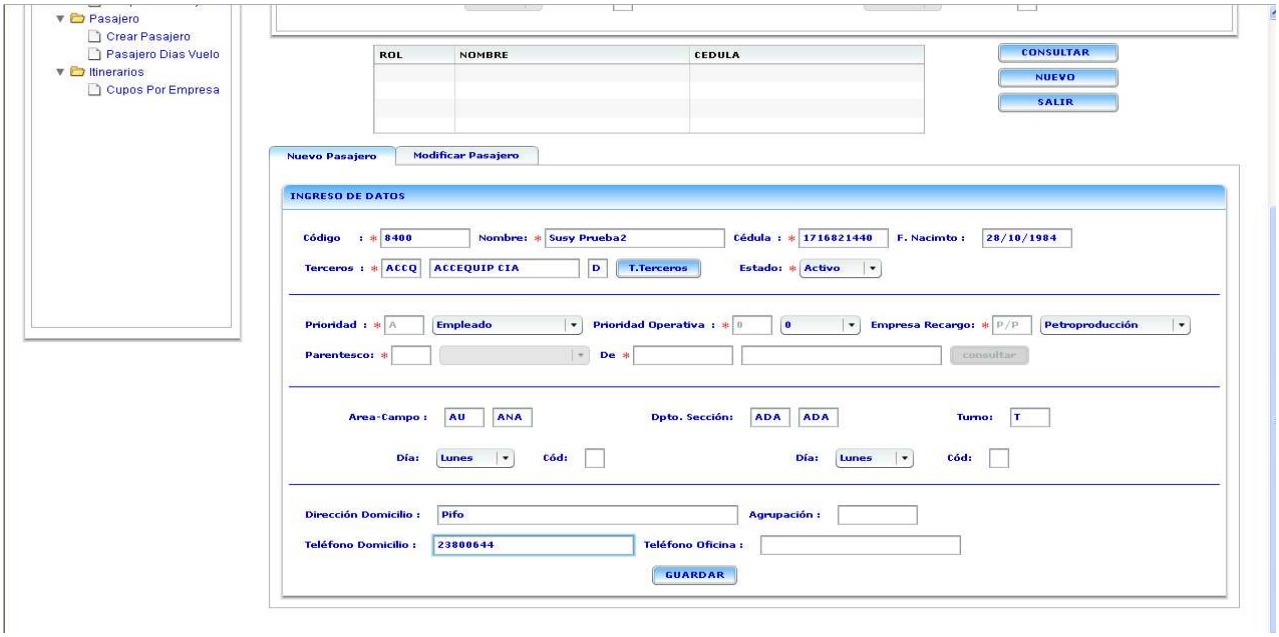
Consulta de cupos de pasajeros

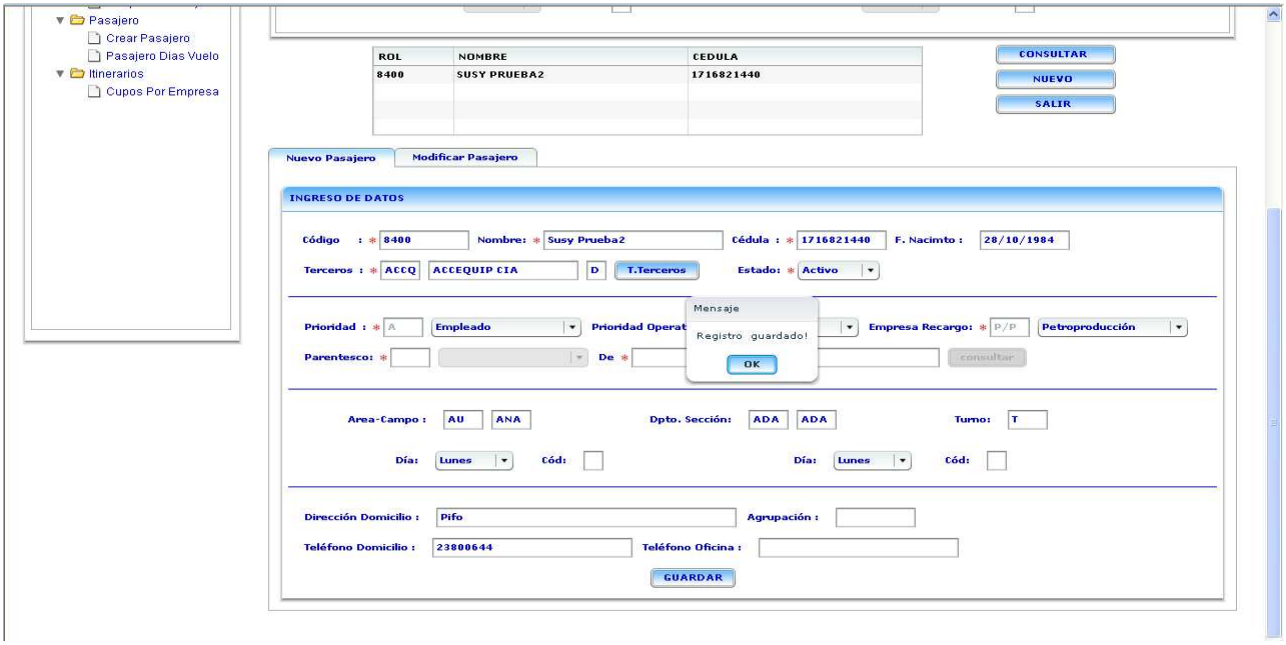
4.-Pasajero (Crear Pasajero).-

Interfaz para creación, modificación e eliminación de pasajero.

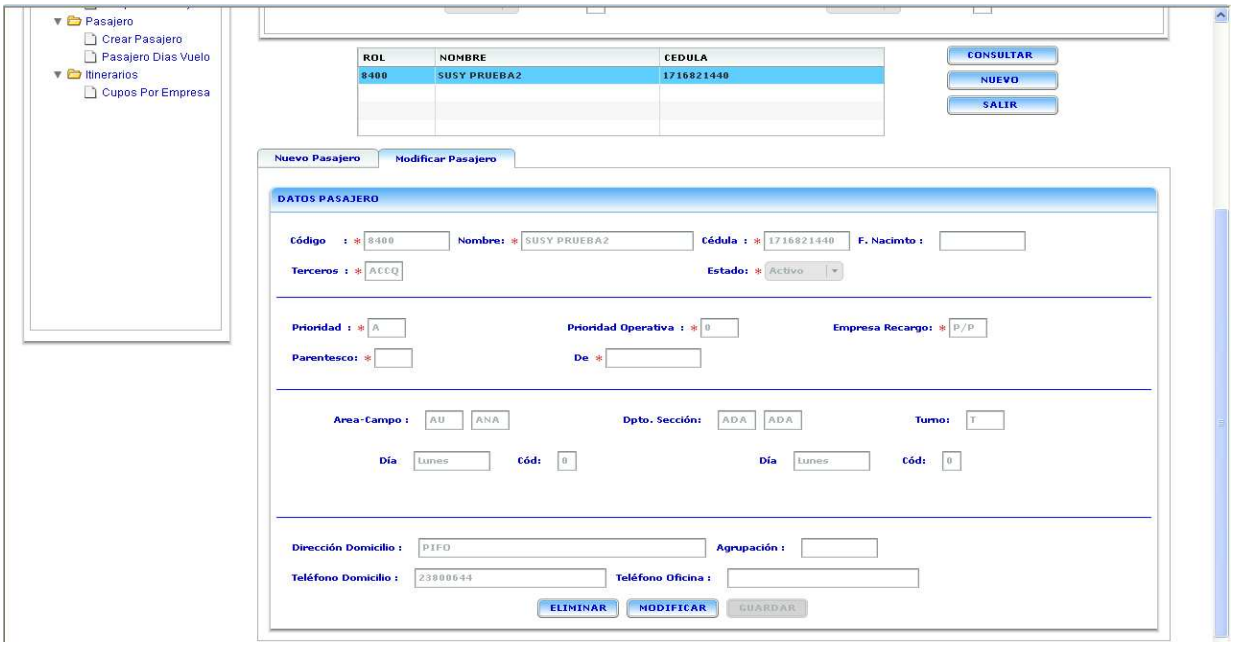
-(Podemos Consultar el pasajero para verificar si existe o no)

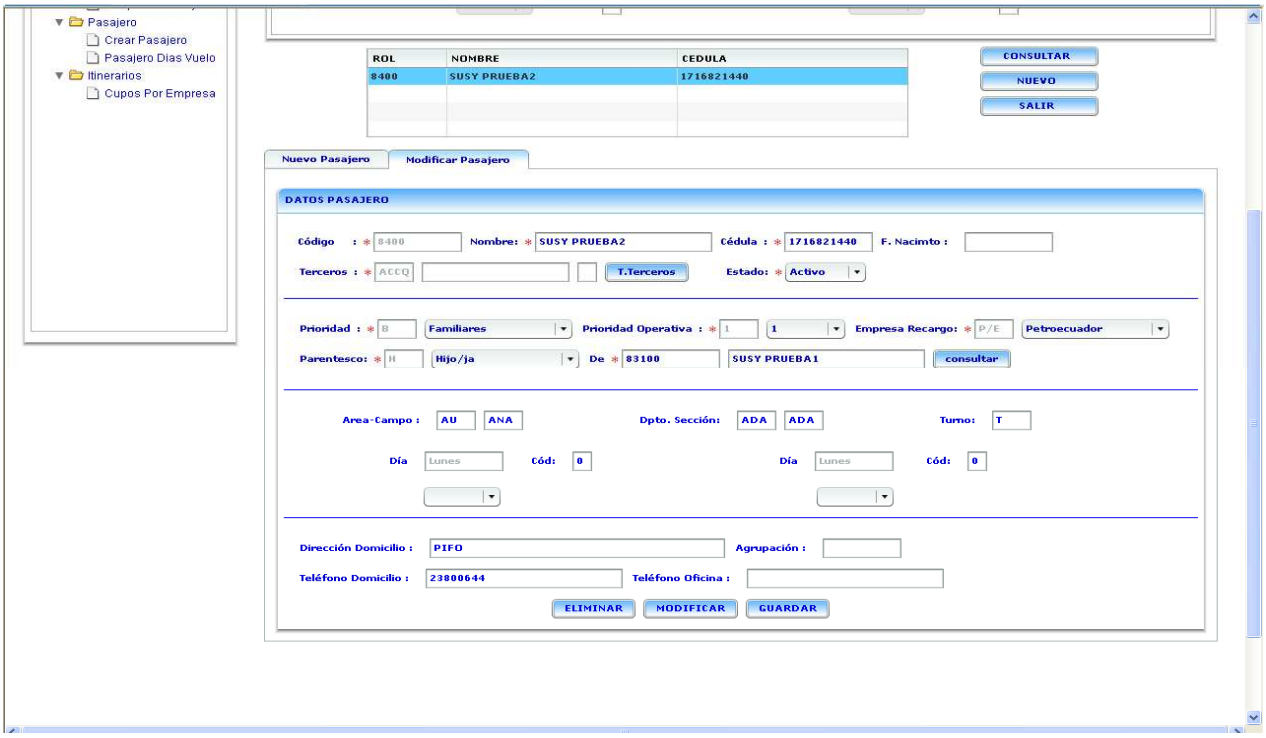
Nuevo Pasajero.

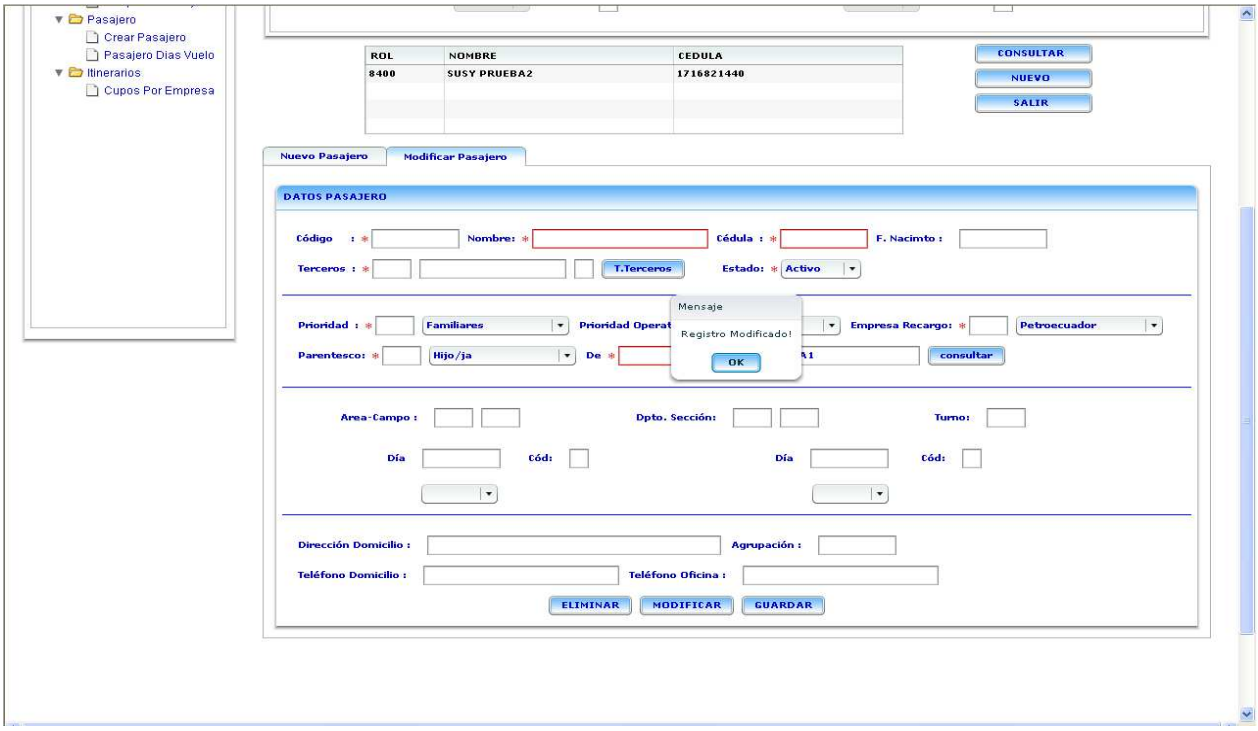




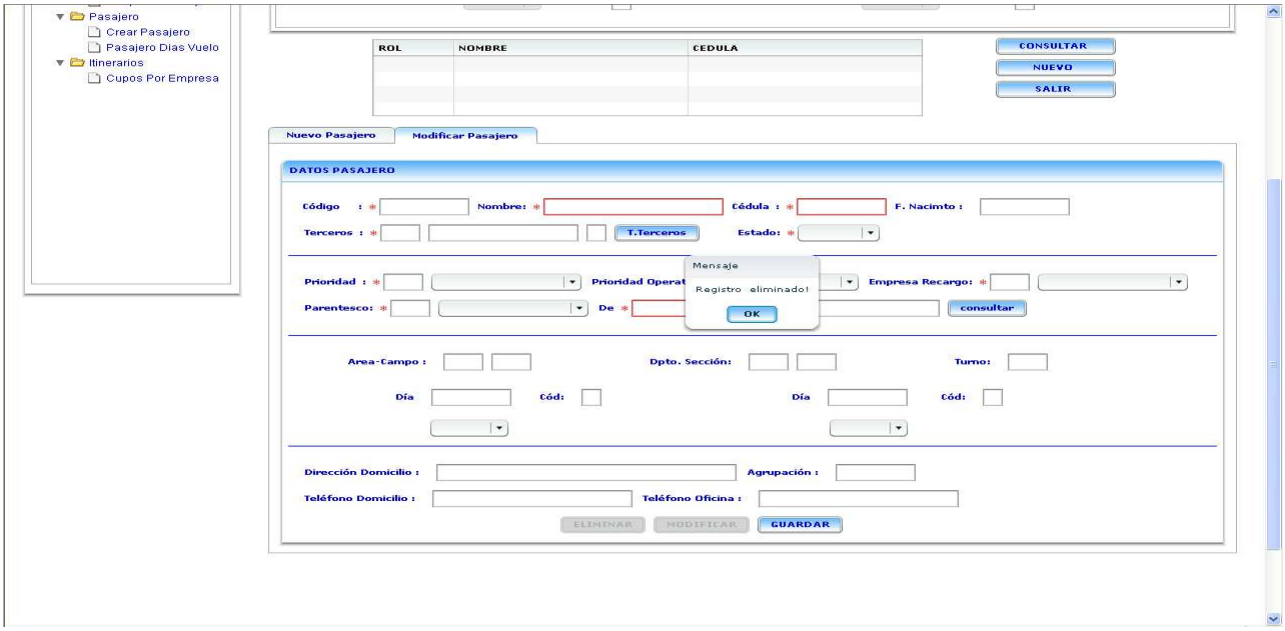
Modificar e Eliminar Pasajero




Pasajero
Crear Pasajero
Pasajero Dias Vuelo
Itinerarios
Cupos Por Empresa
ROL
NOMBRE
CEDULA
8400
SUSY PRUEBA2
1716821440
CONSULTAR
NUEVO
SALIR
Nuevo Pasajero
Modificar Pasajero
DATOS PASAJERO
Código
Nombre:
SUSY PRUEBA2
Cédula :
1716821440
F. Nacimto :
Terceros :
T.Terceros
Estado:
Activo
Prioridad :
Familiares
Prioridad Operativa :
Empresa Recargo:
Petroecuador
Parentesco:
Hijo/ja
De
83100
SUSY PRUEBA1
consultar
Area-Campo :
AU
ANA
Dpto. Sección:
ADA
ADA
Turno:
T
Día
Lunes
Cód:
Dirección Domicilio :
PIFO
Agrupación :
Teléfono Domicilio :
23800644
Teléfono Oficina :
ELIMINAR
MODIFICAR
GUARDAR


Pasajero
Crear Pasajero
Pasajero Dias Vuelo
Itinerarios
Cupos Por Empresa
ROL
NOMBRE
CEDULA
8400
SUSY PRUEBA2
1716821440
CONSULTAR
NUEVO
SALIR
Nuevo Pasajero
Modificar Pasajero
DATOS PASAJERO
Código
Nombre:
Cédula :
F. Nacimto :
Terceros :
T.Terceros
Estado:
Activo
Mensaje
Registro Modificado!
OK
Prioridad :
Familiares
Empresa Recargo:
Petroecuador
Parentesco:
Hijo/ja
De
consultar
Area-Campo :
Dpto. Sección:
Turno:
Día
Cód:
Dirección Domicilio :
Agrupación :
Teléfono Domicilio :
Teléfono Oficina :
ELIMINAR
MODIFICAR
GUARDAR

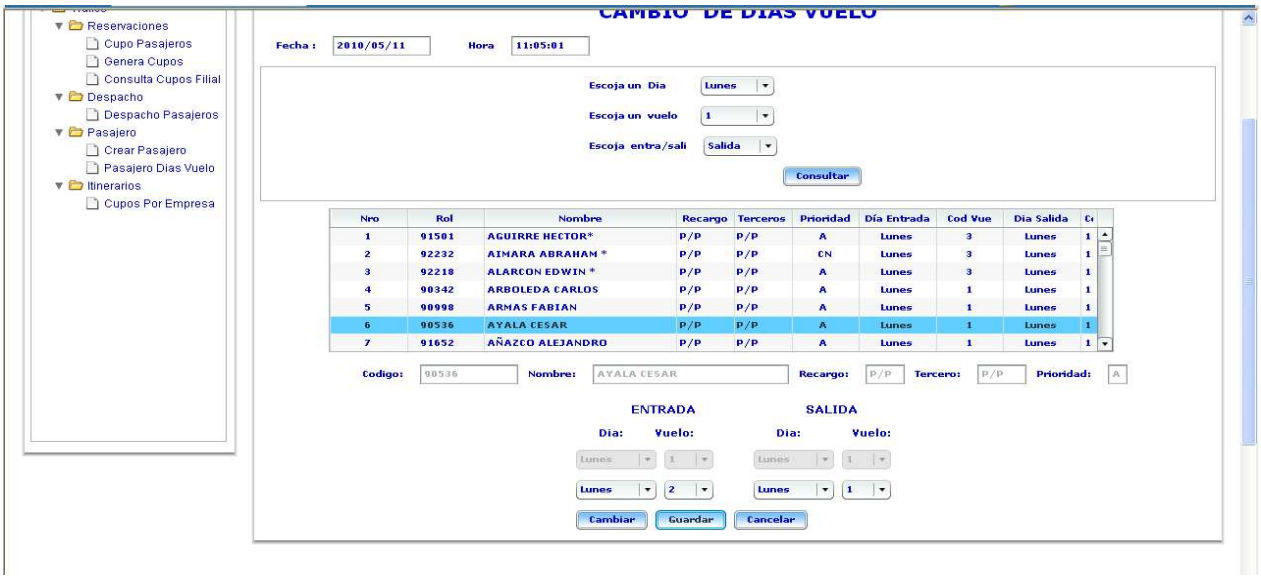
Eliminar

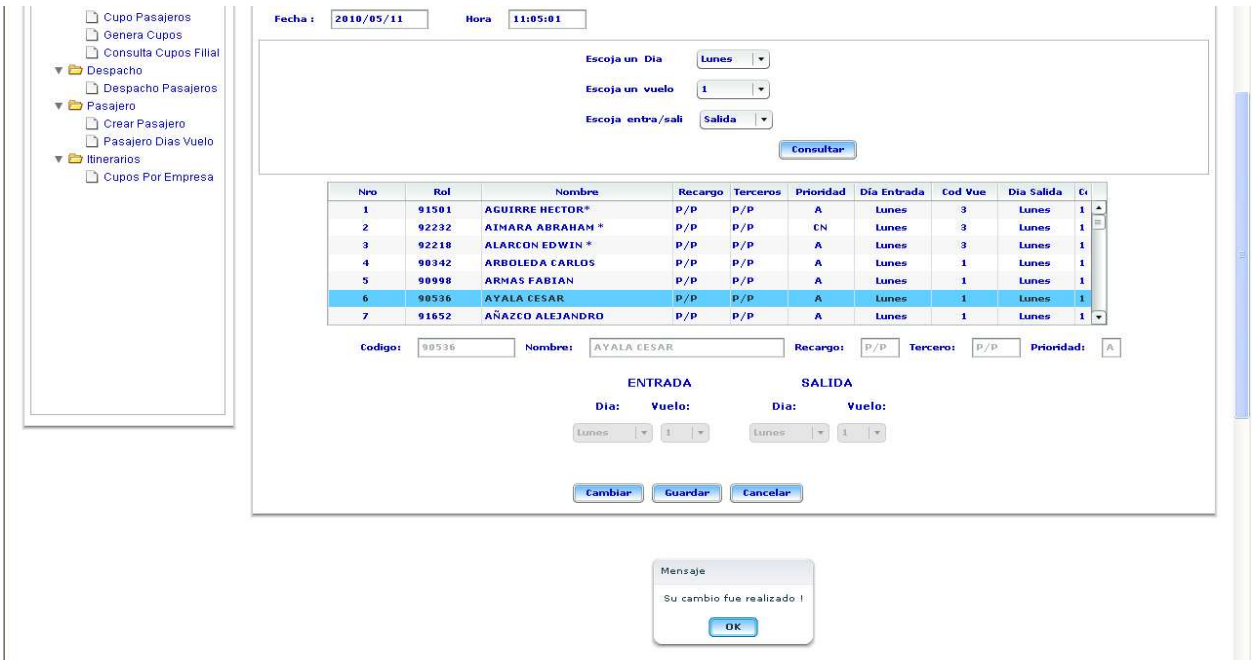


5-Pasajero (Pasajero Días Vuelo).-

La interfaz nos ayuda a consultar todos los pasajeros por día, código y entrada o salida del vuelo .

Y también poder modificar el dia de vuelo de cada pasajero.

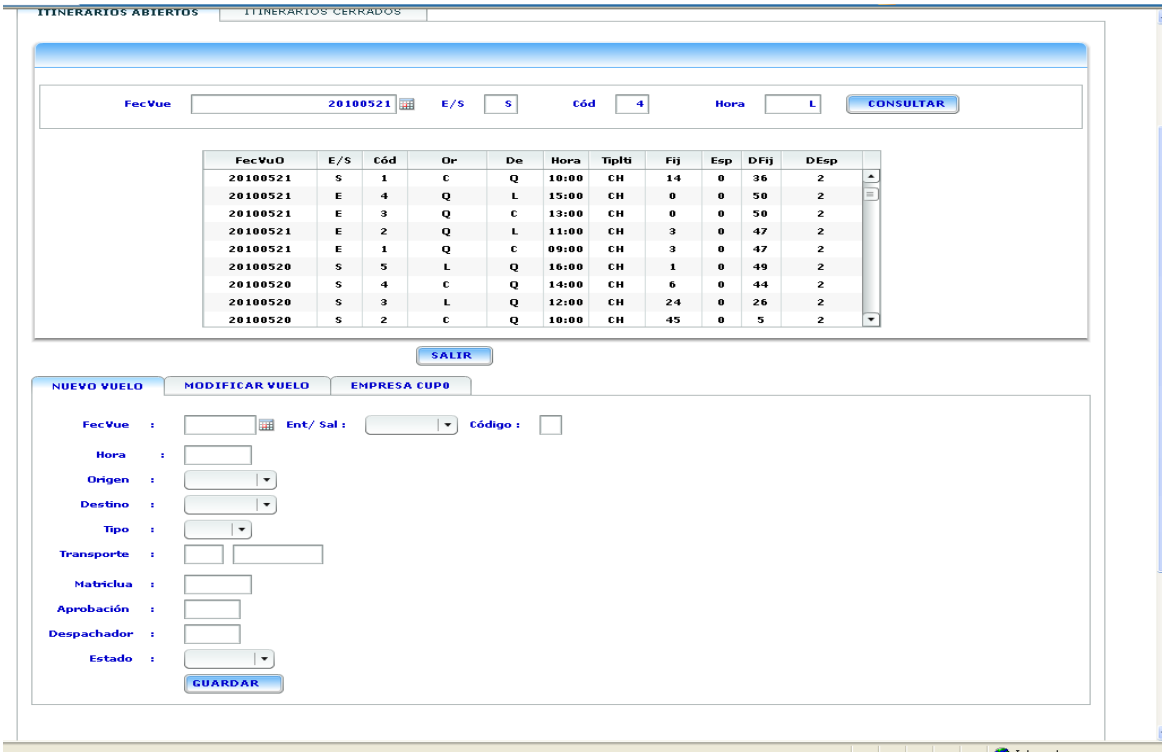




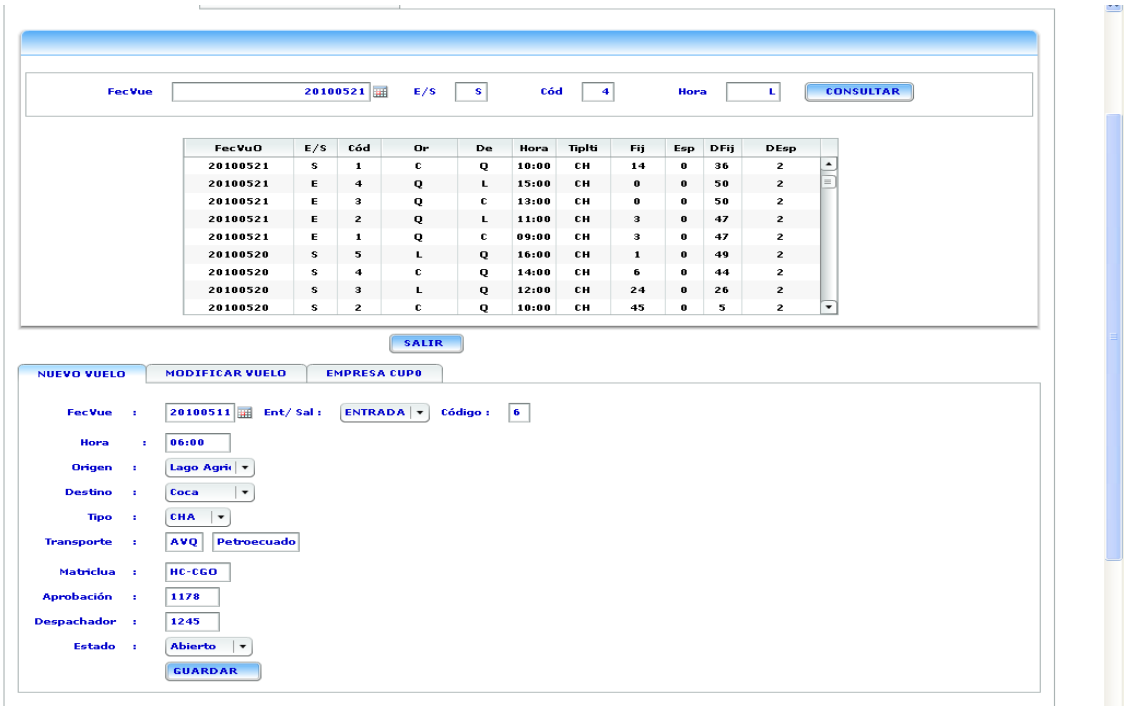
6.-Itienerarios (Cupos por empresa)

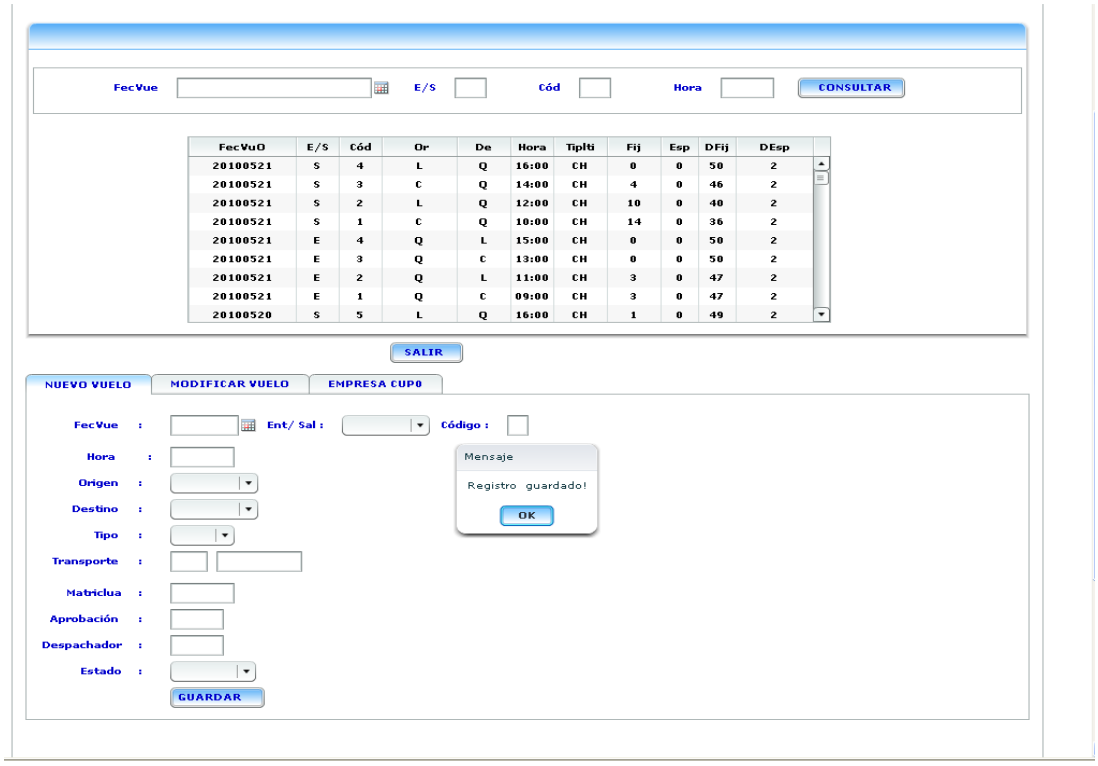
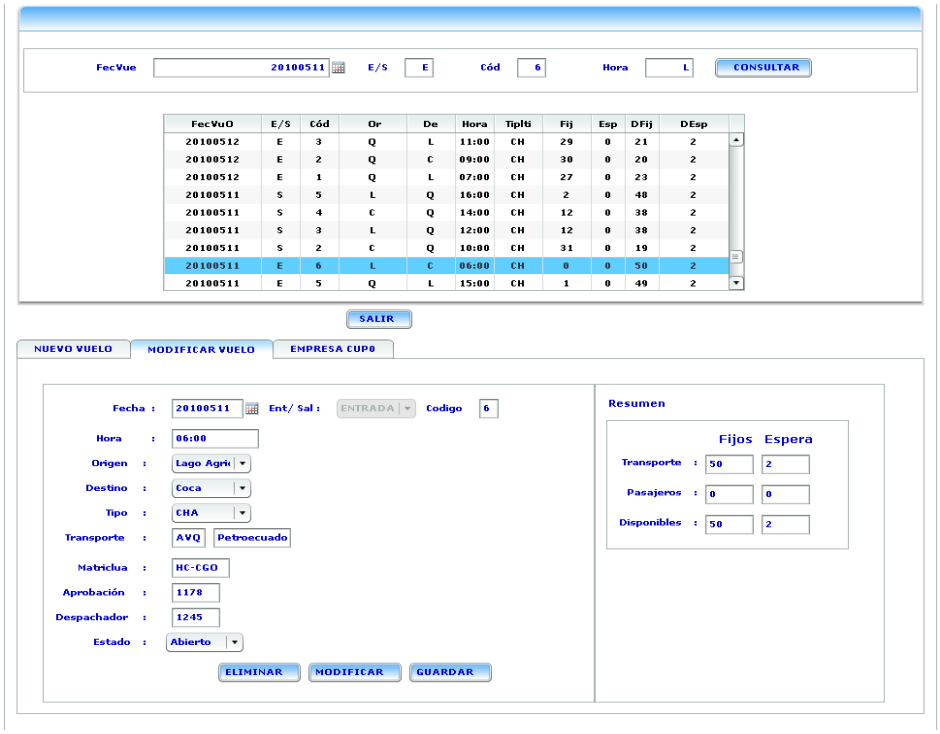
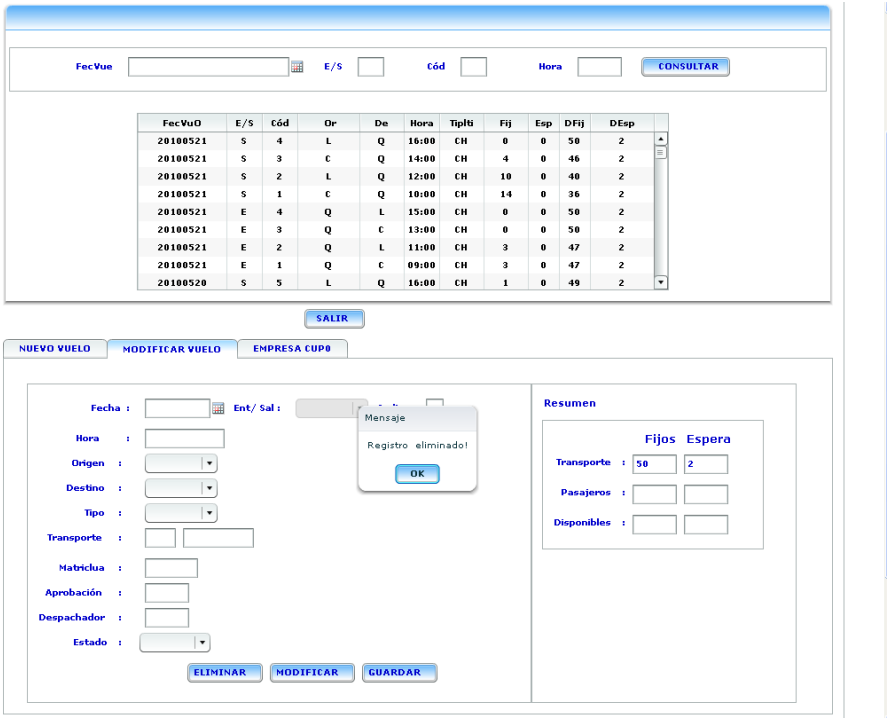
Ingreso de cupos por cada empresa.

7-Itienerarios (Creación y Modificación de Vuelos)



Nuevo Vuelo


FecVue 20100521 E/S S Cód 4 Hora L CONSULTAR
FecVuO E/S Cód Or De Hora Tiplti Fij Esp DFij DEsp
20100521 S 1 C Q 10:00 CH 14 0 36 2
20100521 E 4 Q L 15:00 CH 0 0 50 2
20100521 E 3 Q C 13:00 CH 0 0 50 2
20100521 E 2 Q L 11:00 CH 3 0 47 2
20100521 E 1 Q C 09:00 CH 3 0 47 2
20100520 S 5 L Q 16:00 CH 1 0 49 2
20100520 S 4 C Q 14:00 CH 6 0 44 2
20100520 S 3 L Q 12:00 CH 24 0 26 2
20100520 S 2 C Q 10:00 CH 45 0 5 2
SALIR
NUEVO VUELO
MODIFICAR VUELO
EMPRESA CUPO
FecVue : 20100511 Ent/ Sal : ENTRADA Código : 6
Hora : 06:00
Origen : Lago Agri
Destino : Coca
Tipo : CHA
Transporte : AVQ Petroecuado
Matriclua : HC-CGO
Aprobación : 1178
Despachador : 1245
Estado : Abierto
GUARDAR


FecVue E/S Cód Hora CONSULTAR
FecVuO E/S Cód Or De Hora Tiplti Fij Esp DFij DEsp
20100521 S 4 L Q 16:00 CH 0 0 50 2
20100521 S 3 C Q 14:00 CH 4 0 46 2
20100521 S 2 L Q 12:00 CH 10 0 40 2
20100521 S 1 C Q 10:00 CH 14 0 36 2
20100521 E 4 Q L 15:00 CH 0 0 50 2
20100521 E 3 Q C 13:00 CH 0 0 50 2
20100521 E 2 Q L 11:00 CH 3 0 47 2
20100521 E 1 Q C 09:00 CH 3 0 47 2
20100520 S 5 L Q 16:00 CH 1 0 49 2
SALIR
NUEVO VUELO
MODIFICAR VUELO
EMPRESA CUPO
FecVue : Ent/ Sal : Código :
Hora :
Origen :
Destino :
Tipo :
Transporte :
Matriclua :
Aprobación :
Despachador :
Estado :
GUARDAR
Mensaje
Registro guardado!
OK

Modificar y eliminar Vuelo