

Trabajo Práctico

2ºCuatrimestre

2015

Aerolínea - FRBA

Grupo Nro 46 - LAS_PELOTAS

Nombre y Apellido	Legajo
Adriano Alianiello	144.488-8
Juan Manuel Manrique	144.538-8
Guido Pablo Marsico	144.370-7

Índice

Estrategia General

Consideraciones iniciales

Funcionalidades:

1. ABM de Rol
2. Login y Seguridad
3. Registro de Usuario
4. ABM de Ciudades
5. ABM de Ruta Aérea
6. ABM de Aeronaves
7. Generar Viaje
8. Registro de Llegada a Destino
9. Compra de Pasaje / Encomienda
10. Cancelación / Devolución de Pasaje y/o Encomienda
11. Consulta de Millas de Pasajero Frecuente
12. Canje de Millas de Pasajero Frecuente
13. Listados Estadísticos

Estrategia General

Se comenzó el trabajo práctico analizando la tabla maestra otorgada por la cátedra con el fin de evaluar los datos existentes en la misma y normalizar estos mismos en el momento de la migración de datos de la tabla maestra que se encontraba desnormalizada.

De esta manera se empezó con el análisis de la tabla, identificando las distintas tablas a normalizar, nuevas tablas necesarias a crear y sus relaciones entre sí con el objetivo de poder cumplir con las funcionalidades pedidas. Luego de esto, creamos las constraint, primary key, foreign key, unique, check y not null que consideramos necesarias para cada tabla.

Más tarde por medio del enunciado del trabajo práctico, se empezó con el desarrollo de las ABM correspondientes y en paralelo el desarrollo de la migración. A medida que avanzamos con la aplicación y la migración, fuimos corrigiendo relaciones y creando columnas y tablas que consideramos necesarias.

Consideraciones iniciales

Registros inválidos: las tablas **vuelos**, **boletoDeCompra**, **Paquetes** y **Pasajes** tienen un campo inválido. En el caso de los vuelos, pueden darse la baja a una ciudad o una ruta desde la aplicación, por lo tanto, se invalidar los vuelos vinculados que aún no hayan salido, esto se hace seteando el campo inválido en 1 y no eliminando el registro del vuelo.

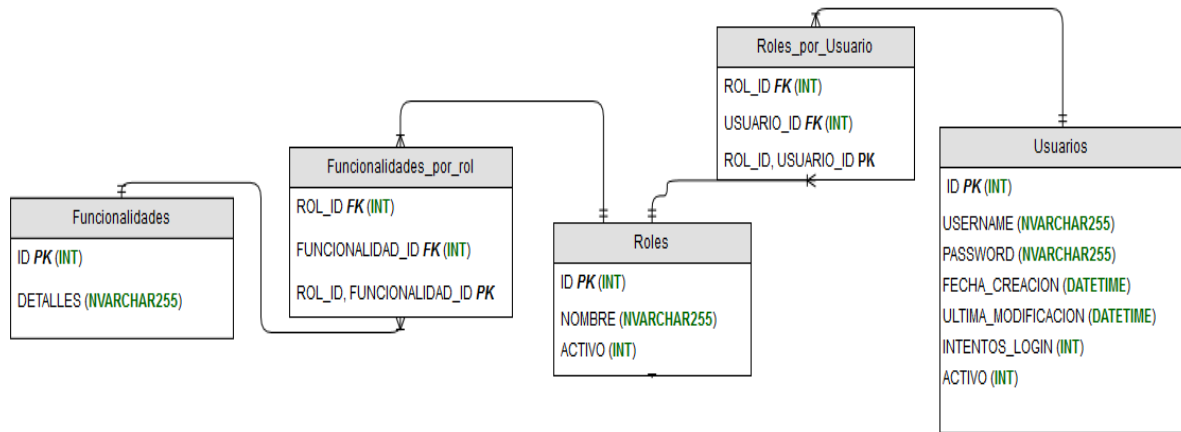
Por otro lado, en cuanto a la migración, se consideran inválidos a todos los vuelos que tengan asociada la misma aeronave y tienen intervalos de fecha superpuestas, con la información de la tabla maestra no se puede decidir cuál registro debe considerarse válido, por lo tanto se invalidan todos, esto se hace por medio del trigger after insert **insert_Vuelos**. De igual manera, los boletos de compra que esté asociados a un vuelo inválido, se dan de baja por medio del trigger **insertBoletosCompra**. Por último, los pasajes y paquetes insertados, que se correspondan a un boleto de compra invalido también serán seteados como inválidos.

Tomamos esta decisión con la intención de no perder registros de la tabla maestra luego de la migración, y además, dejar constancia de que hay un error lógico en los datos.

Cientes con igual DNI: En la tabla maestra encontramos una inconsistencia en los datos, ya que hay clientes con igual DNI y distinto nombre, decidimos mantener todos los clientes, entendiendo que distintos clientes pueden tener igual DNI, además no disponemos de información suficiente para conocer los clientes inválidos.

Funcionalidades

1. ABM de roles



Para esta funcionalidad creamos varias tablas, una de ellas la tabla **Roles**, con todos los roles existentes en la aplicación (como se indicaba en el enunciado actualmente solo existían el rol Administrativo y de cliente), compuesta por los campos de **ID** (PK), **NOMBRE** un nvarchar que especifica el nombre dado al rol y un campo numérico **ACTIVO** que permite saber si ese rol está activo o inactivo.

Otra tabla que creamos en este ABM es la tabla de **Funcionalidades**, que posee todas las funcionalidades disponibles del sistema, para poder relacionar los roles y las funcionalidades, la cardinalidad de relación entre ambas tablas es de muchos a muchos, por lo tanto creamos una tabla asociativa Funcionalidades_por_rol, en donde están todas las funcionalidades que puede ejecutar cada rol específico.

Como un rol puede estar asignado a muchos usuarios del sistema, pudiendo un usuario tener más de un rol se tuvo que crear también otra tabla asociativa entre la tabla **Roles** y **Usuarios** nombrada **Roles_por_usuario**.

En esta funcionalidad se tenía que implementar el alta, baja (lógica) y modificación de un rol. Al dar de alta un rol primero se debe escribir el nombre del rol a dar de alta, clicar el botón **Crear Rol**, si ya hubiera un rol con ese nombre en el sistema tira un Pop out, caso contrario se da de alta al nuevo rol y se nos habilita poder agregarle las funcionalidades del sistema como asignar ese rol a nuestros usuarios.

La baja del rol como lo especifica el enunciado es una baja lógica, por lo tanto al rol se lo deshabilita y no se elimina el registro, quitando el rol deshabilitado a todos los usuarios que lo poseen.

En cuanto a la modificación del rol, los campos permitidos a modificar son el nombre, sus funcionalidades, pudiendo agregar o quitar funcionalidades y también en la modificación de un rol es posible poder habilitar un rol que antes se encontraba deshabilitado.

2. Login y Seguridad

En esta funcionalidad, se permite dar el logueo al sistema, como se especificó en el enunciado de los 2 roles existentes sólo los usuarios con el rol de Administrador que tienen disponibles todas las funcionalidades del sistema se loguean en el sistema, ingresando su nombre de usuario y password.

En caso de que el usuario fallara en loguearse se pedía que se mostrará el Login nuevamente para que intente nuevamente y que se llevará un registros de cantidad de intentos fallidos que luego de tres intentos fallidos en cualquier momento al usuario se lo deshabilitará.

Para implementar esta funcionalidad lo que se hizo fue verificar que el usuario que intenta loguearse sea uno existente, sea de rol Administrador y que esté activo caso contrario tira un mensaje adecuado a la situación. A la hora de mantener un registro de los intentos fallidos lo que decidimos implementar fue agregarle un campo a la tabla **Usuarios** llamado **INTENTOS_LOGIN** de tipo INT NOT NULL DEFAULT 0, para que cada vez que un usuario existente, de rol Administrador y activo se equivocara en el login se iba incrementando ese campo con un stored procedure verificando que si llegara a los tres intentos se lo deshabilitará.



Aerolíneas FRBA

Usuario

Contraseña

3. Registro de usuarios

Usuarios
ID <i>PK</i> (INT)
USERNAME (NVARCHAR255)
PASSWORD (NVARCHAR255)
FECHA_CREACION (DATETIME)
ULTIMA_MODIFICACION (DATETIME)
INTENTOS_LOGIN (INT)
ACTIVO (INT)

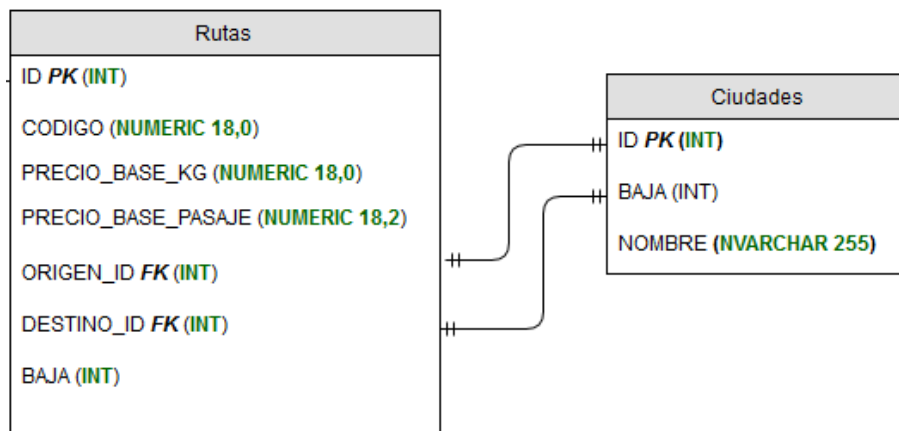
Como se indicaba en el enunciado no era necesario la implementación del ABM de usuarios, lo modelamos por medio de una tabla de **USUARIOS** que nombramos anteriormente, lo que sí se pedía, era un set de usuarios con rol Administrador para probar la funcionalidad del login, este set de usuarios debían tener asignados la password **w23e** y era necesario usar el algoritmo de encriptación SHA256.

La realización de la encriptación de la password por el algoritmo de SHA256 se la implementó a través de una biblioteca en C#.

Set de usuarios para el tp:

- admin
- guido
- juan
- adriano

4. ABM de ciudad



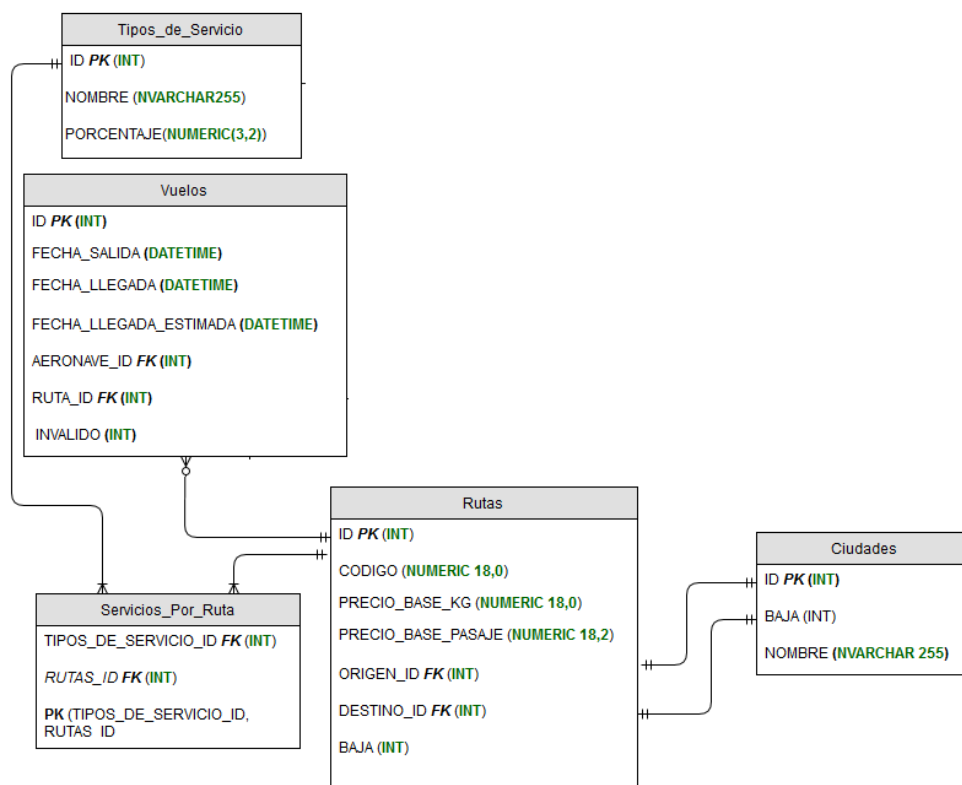
Para este ABM , no era necesario dar de alta o modificar ciudades, solo se debía poder dar de baja a las ciudades.

La manera en que realizamos la migración de la tabla maestra en relación a las ciudades, fue crear una tabla **Ciudades** que se relacionaba con la tabla **Rutas** por medio del origen y el destino de esta última tabla.

A partir de la migración cargamos la tabla **Ciudades**, a partir de los destinos u orígenes de los registros de la tabla maestra.

A la hora de dar de baja una ciudad en la aplicación se la busca y se la da de baja, lo que implicaría dar de baja en cascada a las rutas que tengan asociada la ciudad, siguiendo por la baja de los vuelos asociados a esas rutas, que no hayan salido todavía o no tengan registrada la llegada. La baja de vuelos implica la baja de los boletos de compra con sus pasajes y paquetes. Cabe aclarar que las bajas son lógicas, es decir que se setea el campo baja o inválido en 1 y no se elimina el registro.

5. ABM de ruta aérea



Lo modelamos con una doble relación hacia la tabla de ciudades, como se observa en el DER, una para el origen y otra para el destino. Teniendo en cuenta que una ruta puede tener asociada más de un tipo de servicio y además este último puede estar vinculado a más de una ruta, fue necesario crear la tabla intermedia **Servicios_Por_Ruta** para implementar la relación muchos a muchos.

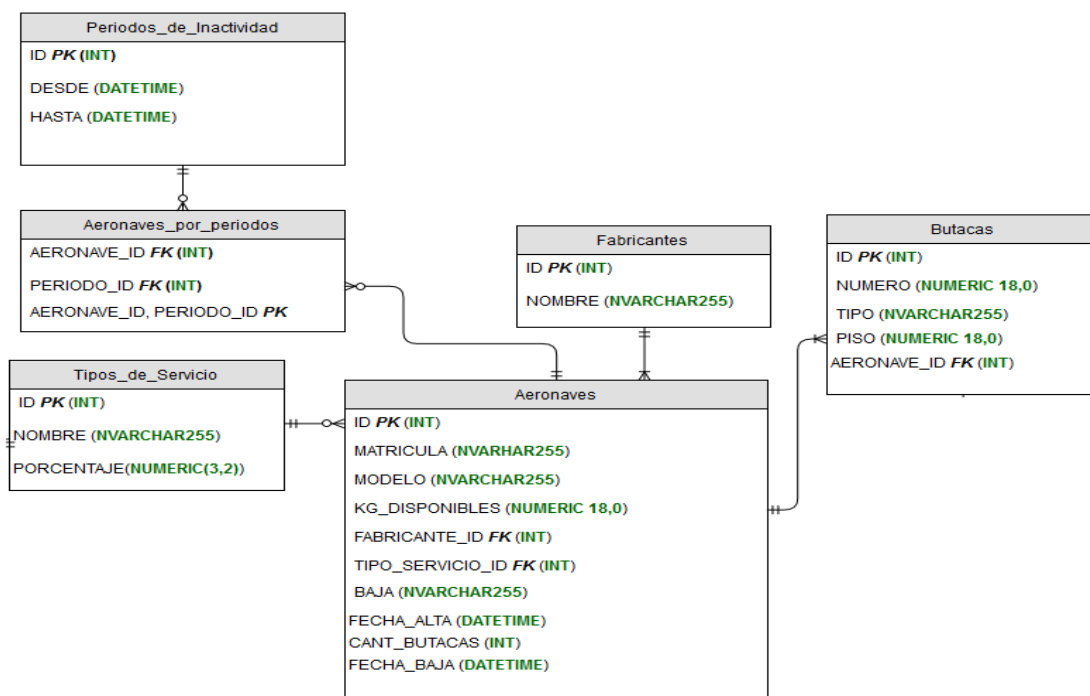
La columna porcentaje, de la tabla **Tipos_De_Servicio**, será utilizada para calcular el costo del pasaje junto con el **Precio_Base_Pasaje** de la ruta, el precio de las encomiendas se calcula a partir del **Precio_Base_KG** de la ruta. Al igual que con la ciudad la baja se da de forma lógico, por dicho motivo no se elimina el registro sino que el valor del campo Baja se pasa a 1.

Cuando damos de alta una ruta aérea es necesario ingresar todos sus campos, **ORIGEN**, **DESTINO**, siendo tomados estos por 2 comboBox con las ciudades de la base, **TIPO_SERVICIO** también es elegido por medio de un combo que tiene los servicios tomados de la base, **CÓDIGO_RUTA**, **Precio_Base_Pasaje**, **Precio_Base_KG** y con la validación correspondiente de todos los campos.

Para la modificación de una ruta tomamos la decisión de que los únicos campos lógicos que se debían modificar de una ruta eran el **Precio_Base_Pasaje** y **Precio_Base_KG**.

Al dar de baja a la ruta, como con la de ciudad, continúa dando de baja los vuelos que no hayan salido o no se hayan registrado una llegada que tengan esa ruta y a su vez los boletos de compra que tienen esos vuelos.

6. ABM de Aeronaves



Para esta funcionalidad se creó la tabla **Aeronaves**, en el alta de una aeronave se le debe indicar su matrícula , el modelo, los kg disponibles para enviar paquetes por encomiendas, la cantidad de butacas, la fecha en que fue dada de alta la aeronave, el fabricante (que se encuentra normalizado con la tabla **Fabricantes** ya que un fabricante puede fabricar muchas aeronaves pero una aeronave tiene un solo fabricante) y el tipo de servicio de la aeronave.

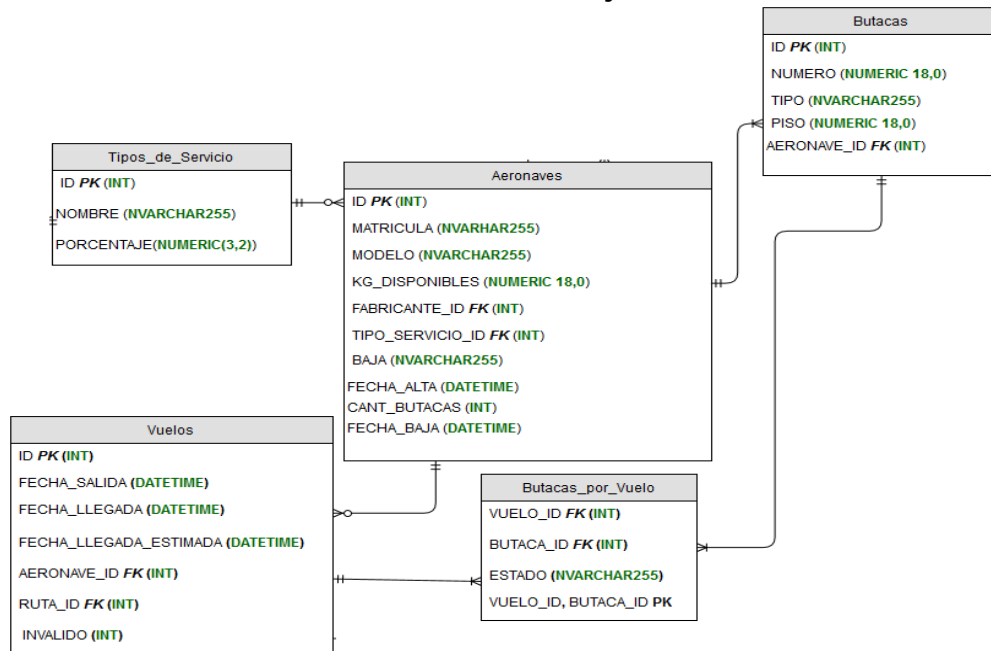
En la aplicación al validar todos los campos correctamente, al dar de alta la aeronave exitosamente se genera luego todas las butacas asociadas a esa aeronave en la tabla **Butacas** insertando la cantidad de butacas especificadas en la aeronave.

Para la modificación de una aeronave decidimos que lo unico logico que podía modificarse es el requisito que se pide de poder poner fuera de servicio momentánea una aeronave durante un intervalo dado, que se realizó creando la tabla **Periodos_de_inactividad** que tiene una relación muchos a muchos con la tabla **Aeronaves** que significó tener que crear una tabla asociativa **Aeronaves_por_periodos**.

A la hora en que un administrativo decida modificar una aeronave puede filtrar aeronaves por su matrícula,modelo,servicio y/o fabricante , seleccionando cuál modificar y indicando el intervalo en que se pondría en inactividad la aeronave. Si la aeronave a poner en inactividad tuviera vuelos durante ese periodo se le avisa al administrativo por medio de un Pop up indicando la situación y redirigiendo a una vista donde se le da la opción de cancelar todos los vuelos asociados a esa aeronave en ese periodo o de reemplazar por cada vuelo que la aeronave tenga en ese periodo por otra aeronave que pueda reemplazarla. Para que aeronave vuelva a estar disponible, cada vez que la aplicación se abre, se verifica por medio de la fecha de sistema si una aeronave que estuvo inactiva debería volver al estado disponible. Esto se implementó a través del procedure **ActivarAeronave**.

Para la baja de una aeronave, como indica en el enunciado, es una baja lógica que cuando se la ponía en periodo de inactividad y tenía vuelos pendientes en ese periodo, se lo indicaba al administrativo, en el caso de la baja se trata de la misma forma todos los vuelos pendientes de la aeronave dada de baja el administrativo puede optar por cancelar o reemplazar si es posible por otra aeronave. La diferencia está que cuando una aeronave es mandada a fuera de servicio momentáneamente, pasado ese periodo vuelve a estar disponible para volar y cuando se le da de baja de manera lógica, se entiende que se completo su vida útil y no volverá a volar.

7. Generar viaje



Se creó la tabla **Butacas_por_Vuelo**, intermedia entre **Butacas** y **Vuelos**, con la intención de identificar qué butaca está comprada de cada vuelo, ya que solo con la tabla de butacas de la aeronave no se puede identificar esto, por que setear una butaca como comprada para una aeronave no sirve, ya que la aeronave puede tener varios vuelos, con lo cual no se sabe en qué vuelo fue comprada el pasaje con dicha butaca. Se modela la capacidad máxima de las aeronaves por medio del campo **cant_butacas**, en el caso de las aeronaves que se insertan por la migración, se asume que la capacidad máxima es el máximo número de butaca para esa aeronave.

Para la implementación de esta funcionalidad en la aplicación tenemos una vista donde se tienen los campos de la hora de salida, de llegada estimada que son campos necesarios para poder generar un vuelo, también es necesario asignarle una aeronave y una ruta aérea para completar la creación del vuelo.

Para elegir una aeronave en la vista hay un listado de aeronaves disponibles que se las puede filtrar por su matrícula, como también un listado de rutas que se pueden filtrar por su código. A la hora de generar el vuelo luego de haber seleccionado todos los campos necesarios, se realizan todas las validaciones correspondientes:

- Que la fecha de salida sea menor a la de llegada estimada y no supere 24 hs de diferencia.
- Que la aeronave y la ruta aérea tenga el mismo tipo de servicio.
- Que la aeronave elegida esté disponible en ese horario de vuelo.

En caso de no cumplir alguna de las validaciones el sistema manda un mensaje de error indicando lo sucedido, caso contrario el vuelo es creado y se muestra un mensaje de éxito.

Generar Viajes

Horarios

Fecha y Hora Salida

21/11/2015 19:20:38

Fecha y Hora Llegada Estimada

21/11/2015 19:20:38

Aeronaves

Matrícula

Filtrar

Rutas

Código

Filtrar

Guardar

Limpiar

Volver

8. Registro de Llegada a destino

Para implementar esta funcionalidad hicimos uso de un listado de los vuelos que todavía no se les registró la llegada a su destino, eligiendo al que se quiera registrar la llegada introduciendo la fecha de llegada.

También es posible realizar un filtro de los vuelos que no tienen una llegada registrado, por los campos de la matrícula de la aeronave, por la ciudad de salida y la ciudad de llegada de los vuelos.

Al realizar el registro de la llegada del vuelo, que lo realizamos por medio de un stored procedure **registrarLlegada** que es llamado desde la aplicación, actualiza en la tabla de vuelos seteando la fecha de llegada al vuelo correspondiente y también se realiza un update a los boletos de compra de los pasajes y paquetes asociados al vuelo que se registró la llegada, seteando las millas calculadas como el **precio de total compra** dividido 10, dicho precio total de compra se calcula por medio de la función **precioTotal**, la cual recibe el id del vuelo y retorna la sumatoria de los precios de las pasajes y paquetes asociados.

Registrar Llegadas


Matrícula

Ciudad Origen

Ciudad Destino

Filtrar

--

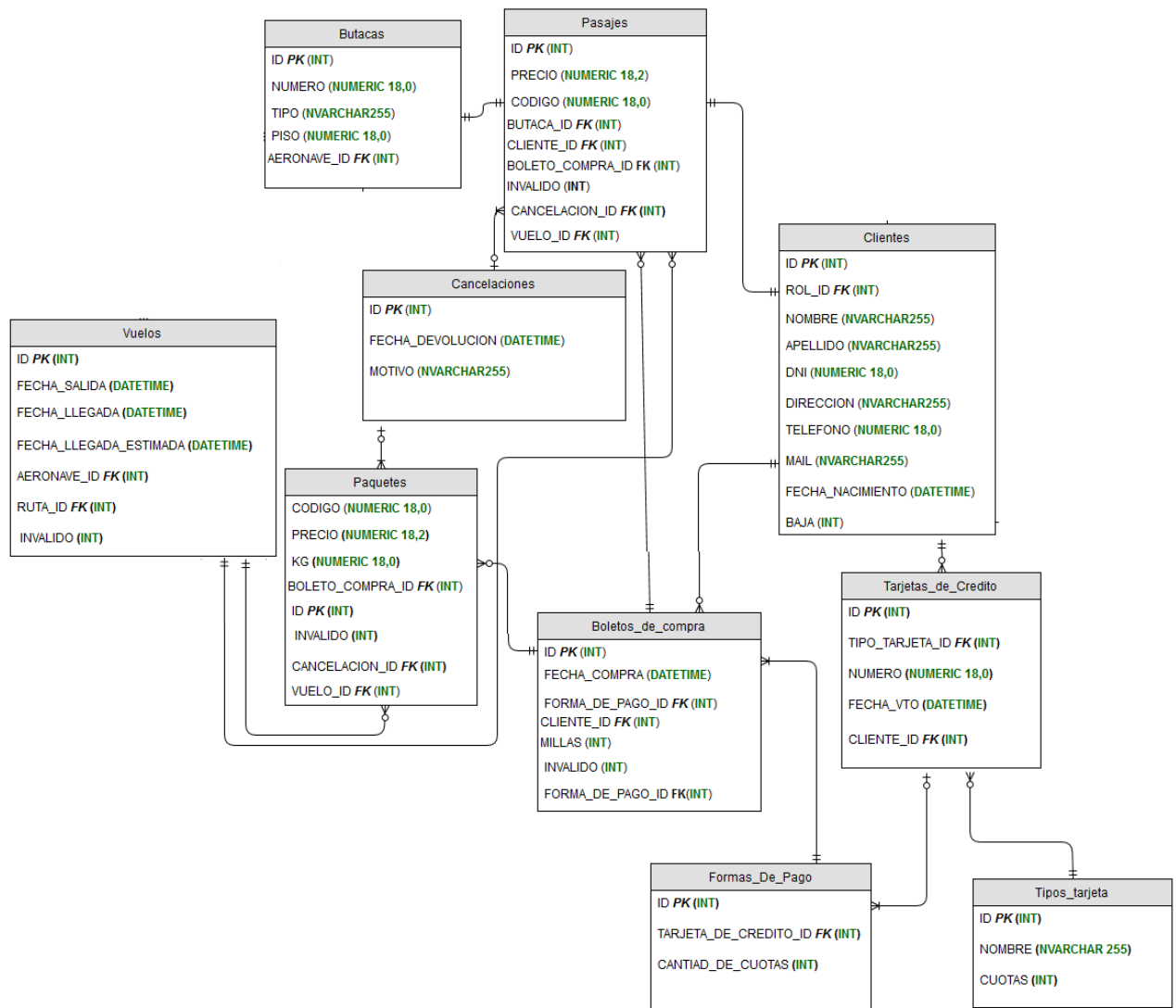
Fecha y Hora de Llegada 21/11/2015 20:10:22 

Registrar

Limpiar

Regresar

9. Compra pasajes/Encomiendas



En cuanto al modelo, se crearon las tablas pasajes y paquetes, las cuales referencian a un boleto de compra, cada boleto de compra tiene una única forma de pago, en principio no habíamos tenido en cuenta este concepto, aunque después nos dimos cuenta que era necesario para saber cómo realizó el pago de cierto boleto de compra un cliente en particular, conocer si lo hizo en efectivo, o con alguna de las tarjetas de crédito que posee y la cantidad de cuotas. De esta manera cada boleto de compra hace referencia a un registro de la tabla de formas de pago, puede ocurrir que varios boletos se paguen de igual manera, como por ejemplo en efectivo, por lo tanto decidimos que una forma de pago puede estar asociada a muchos boletos de compra.

Además, las tablas pasajes y paquetes, tienen una referencia al vuelo correspondiente, y el pasaje también tiene una referencia a cliente, ya que el cliente que compra, el que está asociado al boleto de compra, puede no ser el mismo que el cliente del pasaje comprado.

En cuanto a la aplicación, para la compra de un pasaje o envío de paquetes (Encomiendas), se tiene una primera pantalla donde al usuario le aparece un listado de los viajes disponibles para realizar una compra, pudiendo filtrar los viajes por la fecha de salida, el origen y el destino.

Compra

Fecha Origen Destino

Cantidad de Pasajes

Kg de Encomiendas

En esa primera vista se indica la cantidad de pasajes que se desea comprar y/o kg para enviar, al presionar comprar se valida que la cantidad de pasajes, kg sea posible de realizar esa compra, en caso de serlo, nos envía a la siguiente vista.

DNI

Buscar

Limpiar

Apellido
Nombre
Dirección

Teléfono
Mail
Fecha Nacimiento

Modificar Datos

Pasajes

Número
Ubicación

Cargar

Nombre	Apellido	DNI	Butaca	Ubicación	Precio

Eliminar

Limpiar

Encomiendas

Peso
Kg

Cargar

Kg	Precio

Eliminar

Limpiar

Confirmar Compra

Regresar

La segunda vista para la compra, es donde se cargan los datos necesarios para cargar pasajes y cargar paquetes para enviar. Para la carga de pasajes, si se eligió comprar pasajes se habilita poder cargar los datos del pasajero, primero cargando el dni del pasajero, si existe el cliente en la base los datos personales se cargan automáticamente, en caso de que el dni ingresado sea de un cliente inexistente se deja realizar un alta del pasajero y se cargan los datos necesarios. Luego se debe elegir la butaca para el pasajero que es un comboBox con las butacas libres del vuelo y al ir cargando los datos del pasajero se va llenando una grilla con el detalle del pasajero con sus datos y la butaca elegida.

Para la carga de paquetes, si se eligió enviar kgs se debe indicar la cantidad de kg que se quiere enviar y se va cargando en una grilla con los paquetes enviados con su detalle.

Al clickear comprar se pasa a la siguiente vista validando que todos los datos sean correctos, dependiendo el rol del usuario hay dos vista siguientes posibles:

Si el rol es de administrador le permite elegir el tipo de pago a usar si efectivo o con tarjeta.

Efectivo

Tarjeta

Volver

Forma pago en Efectivo, se debe indicar al cliente que realiza la compra, buscando al cliente por su número de DNI.

Pago en Efectivo

DNI

Buscar

Limpiar

Apellido
xxx

Nombre
xxx

Dirección
xxx

Mail
xxx

Teléfono
xxx

Fecha Nacimiento
xxx

Modificar Datos

Importe (\$) xx

Confirmar

Volver

Si el rol es de cliente lo redirige a la vista de forma de pago con tarjeta, donde debe ingresar los datos del comprador ingresando su dni y ingresando un número de tarjeta que el cliente tenga o si no tuviera ninguna tarjeta se puede dar de alta una tarjeta de crédito para el cliente en el momento.

Pagar con Tarjeta de Crédito

DNI

Buscar

Limpiar

Apellido

Mail

Nombre

Dirección

Fecha Nacimiento

Teléfono

Modificar Datos

Tarjeta

Número

Fecha Vencimiento

Cuotas

Tipo

Código Seguridad

(*) Ingreso Obligatorio
☐

Ingresar Tarjeta

Nueva Tarjeta

Limpiar

Confirmar

Regresar

Al confirmar la compra tanto por efectivo o por tarjeta, se realizan los insert por medio de llamados a stored procedures para el alta del boleto de compra, de los pasajes y de los paquetes.

La última vista de la compra, luego de confirmar la misma nos muestra el detalle de la operación.

Su código de compra es: XXXX

Fecha: XXX

Origen: XXX

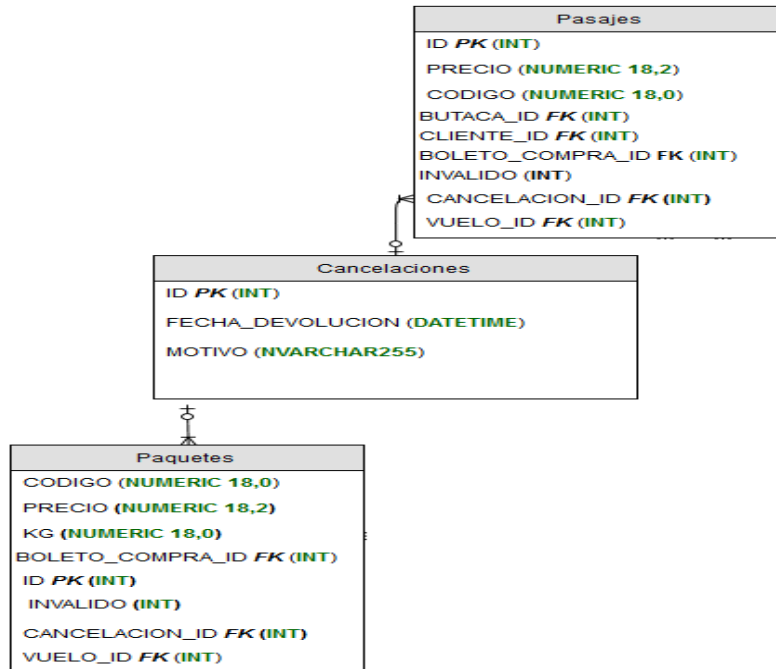
Destino: XXX

Pasajes Comprados: XXX

Paquetes Enviados: XXX

Aceptar

10. Cancelación de pasajes/Encomiendas



Para implementar la funcionalidad se realizó una vista donde se debe ingresar el dni del cliente para mostrar sus datos, luego debe ingresar el código del boleto de compra, que le muestra en dos grillas los pasajes y paquetes asociados a esa compra.

Para la cancelaciones de pasajes es posible elegir qué pasajes cancelar, seleccionandolos y clickeando el botón **Cancelar Pasaje**, que genera la ejecución de un stored procedure, éste inserta un registro en la tabla **Cancelaciones**, y actualiza en la tabla **Pasajes** el campo la referencia al registro insertado en cancelaciones, por último asigna estado de "LIBRE" a la butaca del vuelo asociada al pasaje.

Para la cancelación de paquetes, a diferencia de los pasajes, no se puede elegir qué paquete cancelar sino que al clickear el botón **Cancelar Paquetes** son cancelados todos los paquetes asociados a esa compra. Esto se implementó por medio de un stored procedure, que inserta un registro en la tabla de **Cancelaciones**, también realiza un update al registro del paquete cancelado, seteando la referencia al registro de cancelaciones insertado, por último realiza la actualización de kgs disponibles de la aeronave en ese vuelo, restando los kg ocupados por los paquetes cancelados.

No agregamos información acerca de la cancelación en la tabla boleto de compra, ya que no tiene mucho sentido cancelar un boleto de compra, no te permite identificar qué pasajes asociados a este se cancelaron o si se cancelaron los paquetes.

La vista correspondiente a la funcionalidad es la siguiente:

Cancelar Compra

DNI

Buscar

Limpiar

Apellido

Nombre

Dni

Dirección

Fecha Nacimiento

Teléfono

Mail

Codigo de Compra

Consultar

Cancelar Pasaje

Cancelar Paquetes

Regresar

11. Consulta de millas de pasajeros frecuentes

Para esta funcionalidad se tuvo en cuenta que tanto los paquetes como los pasajes suman a las millas de pasajeros frecuentes. La vista muestra un listado de de incrementos de millas y de decrementos ocasionados por los canjes, y además el total acumulado, teniendo en cuenta que las que tienen más de un año se vencen.

Consulta de Millas

DNI

1122696

Fecha	Motivo	Millas
-------	--------	--------

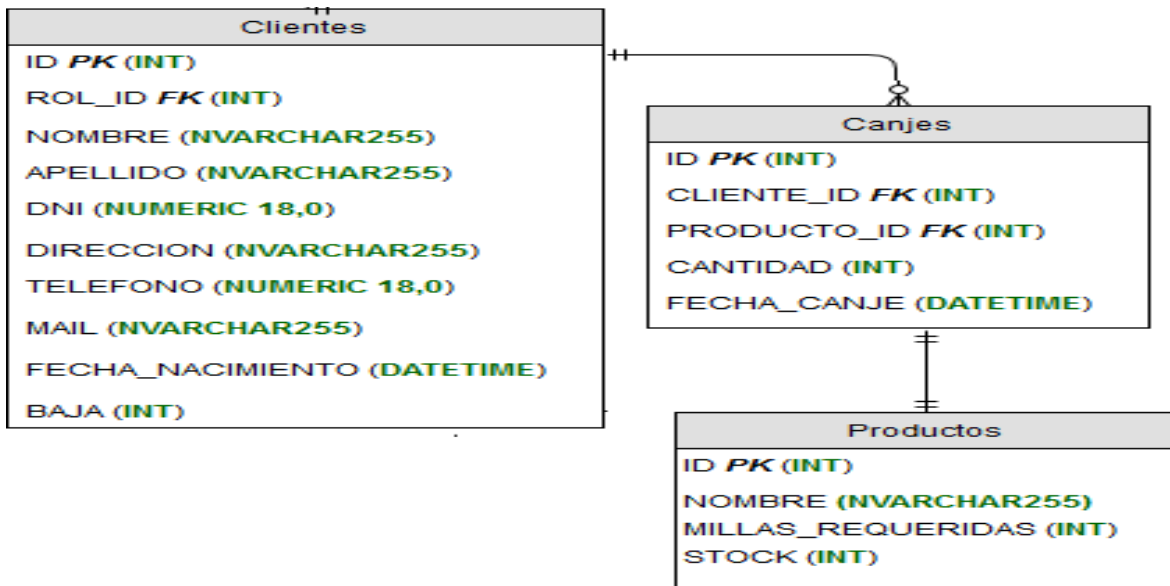
Mostrar

Limpiar

Acumulado
0

Regresar

12. Canje de millas de pasajeros frecuentes



Para modelar esta funcionalidad se utilizaron las tablas **Canjes** y **productos** como se observa en el DER. Por medio del procedure **altaCanje** se efectúan los canjes de los clientes con sus respectivos productos y se actualiza el stock de este último.

Se insertaron los siguientes productos para canjes:

	ID	NOMBRE	MILLAS_REQUERIDAS	STOCK
1	1	SABANAS	50	200
2	2	PS3	20000	5
3	3	TABLETA	1500	50
4	4	TV	10000	20
5	5	SILLA	60	100
6	6	MOTO	15000	10
7	7	HORNO	500	150

La siguiente vista de la aplicación, corresponde a la funcionalidad de caja:

Canje de Millas

DNI

Limpiar

Buscar

Nombre	Apellido	Dni	Fecha de Nacimiento	Millas

Ver Canjes

Cantidad

Canjear

Volver

13. Listado estadístico

Esta funcionalidad permite consultar un TOP 5 de listados estadísticos. Siendo necesario indicar el año y el semestre del que se quieren obtener los datos. Esta funcionalidad se implementa por medio de 5 stored procedure, que realizan la consulta correspondiente, generando los cruces de registros, agrupando y aplicando las funciones count o sum según corresponda. Los listados que el administrador podrá elegir para consultar son los siguientes

-Top 5 de los destinos con más pasajes comprados.

Estadísticas

Tipo de Listado

Top 5 Destinos con más pasajes comprados

Año

2015

Período

Segundo Semestre (Julio a Dic

Mostrar

Destino	Pasajes comprados

Regresar

-Top 5 de los destinos con aeronaves más vacías

Estadísticas

Tipo de Listado

Top 5 Destinos con aeronaves más vacías

Año

2015

Período

Segundo Semestre (Julio a Dic

Mostrar

Destino	Butacas Vacías
---------	----------------

Regresar

-Top 5 de los clientes con más puntos acumulados a la fecha.

Estadísticas

Tipo de Listado

Top 5 Clientes con más puntos acumulados a la fecha

Año

2015

Período

Primer Semestre (Enero a Juni

Mostrar

Cliente	Millas
---------	--------

Regresar

-Top 5 de los destinos con más pasajes cancelados.

Estadísticas

Tipo de Listado

Top 5 Destinos con más pasajes cancelados

Año

2015

Período

Primer Semestre (Enero a Juni

Mostrar

Destino	Cancelaciones
---------	---------------

Regresar

-Top 5 de las aeronaves con mayor cantidad de días fuera de servicio.

Estadísticas

Tipo de Listado

Top 5 Aeronaves con mayor cantidad de dias fuera de se

Año

2015

Período

Primer Semestre (Enero a Juni

Mostrar

Matrícula Aeronave	Días fuera de servicio
--------------------	------------------------

Regresar