**Estrategia**

**Entidad Aeronave:** La entidad Aeronave aparte de contener los campos obligatorios que se especifican en el enunciado, se le agrego un campo estado. Este campo sirve para cuando la aeronave esta fuera de servicio temporalmente o cuando se da la baja definitiva, pues la baja de aeronave es una baja lógica por ende para mantener la integridad referencial, no se puede eliminar la fila que contiene a la misma. La aeronave en sí no posee una relación directa con las butacas (pero sí con su cantidad total, pues cada butaca tiene un estado, y el mismo varía a cada viaje). Por último la entidad tiene un servicio que es una clave foránea a la entidad servicio.

**Entidad Butaca:** La entidad Butaca es creada para poder tener registro de las todas las butacas vendidas en los pasajes de todos los viajes. La misma consta de datos propios como piso en el que se encuentra del avión; tipo, para saber si es ventanilla o pasillo y también de su estado para saber si está ocupada o no. Se relaciona con viaje porque el mismo puede variar (si estuviera relacionado con aeronave, no podría tener un estado, pues la aeronave tiene las cantidades, pero su estado difiere en cada viaje).

**Entidad Canje:** La entidad Canje es creada para tener registro de los canjes de millas por productos que pueden hacer los clientes. En ella se tienen los datos de fecha del canje, la cantidad del producto que se está canjeando, una clave foránea a la entidad Producto, y una clave foránea a la entidad Cliente, ambas referenciando el producto del canje y el cliente que canjea.

**Entidad Ciudad:** La entidad Ciudad es creada para tener registro de las ciudades, sean destino u origen de nuestras rutas aéreas o no. En esta se guardan el nombre de la ciudad.

**Entidad Cliente:** El cliente es toda aquella persona con interés en la empresa, pudiendo tener o no compras realizadas. Un usuario es quien ingresará las ventas realizadas por el cliente, asignando correctamente la relación (puede ser la misma persona física o un tercero). El cliente puede realizar canjes de productos por sus millas acumuladas, por lo que existen esas relaciones.

**Entidad Compra:** La compra es creada para asignar un pasaje o una encomienda, reservarlas y dejar registrada la terminal en la que se realizó, para un determinado viaje.

**Entidad Devolución**: La entidad Devolución es creada para tener registro de las cancelaciones de compras. En ella se encuentran datos como la fecha de devolución, el motivo, posee una clave foránea a la compra que se está cancelando y como clave principal tiene un código propio que es el PNR que se le entrega al cliente luego de realizar una compra. Además desde esta tabla se harán las consultas para el listado estadístico de ciudades con más pasajes cancelados.

**Entidad Encomienda:** La entidad Encomienda es creada para registrar encomiendas en una compra de la Aerolínea FRBA. La misma consta de la cantidad de kilogramos a transportar, la compra a la cual pertenece (la cual es clave foránea a la entidad Compra ya que debe referirse a una compra existente).

**Entidad Func\_X\_Rol:** A cada rol se le aplican múltiples funcionalidades y a cada funcionalidad puede corresponderle más de un rol. Para romper la relación muchos a muchos, y así cumplir con la forma normalizada, es que se crea la entidad funcionalidad por rol (Func\_X\_Rol), compuesta por las FK de Rol y Funcionalidad, que conforman su PK compuesta.

**Entidad Funcionalidad:** Representa a las diferentes responsabilidades que puede acceder y ejecutar un usuario, dado su rol.

**Entidad Millas\_Acumuladas:** Para cada viaje, a partir de ahora, se implementará un sistema de acumulación de millas, que serán de libre intercambio dentro de los 365 días del viaje por uno de los premios. La tabla comienza vacía, pues es una nueva funcionalidad del sistema.

**Entidad Pasaje:** Representa a una posición de la butaca ocupada por una compra de un viaje determinado. Sirve para saber que butacas todavía se encuentran libres de un cierto viaje al momento de elegir.

**Entidad Producto:** Representa a la serie de artículos intercambiables por las millas acumuladas para un usuario.

**Entidad Ruta\_Aerea:** Con referencias a las ciudades (aeropuertos) de origen y destino, la ruta aérea representa la trayectoria que los aviones pueden seguir en sus viajes. Es consistente el tipo de servicio de la ruta con el del avión,

**Entidad Servicio**: El servicio corresponde tanto al que proporciona una aeronave y como al que admite una ruta aérea determinada, que debe ser consistente en todos los viajes. Agrega un porcentaje de costo al vuelo. Para el caso de la aeronave solo se admite un solo tipo de servicio(dato del enunciado).

**Entidad Terminal:** Los tipos de terminal determinan que tipos de pago están habilitados para cada compra (no se puede comprar en efectivo en un autoservicio, por ejemplo).

**Entidad Tipo\_De\_Pago:** Une a las distintas posibilidades que tiene un cliente de formalizar su compra, según la terminal en la que esté haciendo la compra.

**Entidad Usuario:** Diferente de un cliente, un usuario es aquella entidad que tiene permiso para acceder al sistema (puede o no ser cliente, pero un cliente debe ser un usuario). El usuario tiene en su tabla si está habilitado para realizar operaciones, junto con su username de login y su contraseña (hasheada para ser indescifrable). Un usuario debe tener forzosamente asociado un rol, para así asignarle funcionalidades que pueda o no utilizar.

**Entidad Viaje:** El viaje representa a una ruta aérea en un momento determinado, con un avión asignado.

Respecto a los stored procedures que manejan filtros de distintas tablas, reciben una cierta cantidad fija de parámetros, que pueden ser nulos. En cuyo caso, tomará automáticamente ese valor como válido y no lo contemplará en el filtro. Esto permite que desde la aplicación se pueda filtrar por cualquier elemento o grupo de elementos sin mayor dificultad (get\_all\_aeronaves, por ejemplo, permite que se combinen entre 0 y 11 filtros distintos de cualquier manera, sin variar el procedure).

**Aplicación**

- Rol no tiene formulario de baja porque el enunciado dice que no se va a borrar un rol nunca, como mucho se van a modificar las funcionalidades que tiene. La baja sería deshabilitarlo pero para eso no necesitamos un form de baja.

- En la baja de las aeronaves sí necesitamos un formulario, donde tenemos que cargarle los motivos de baja, etc.

- Los controles que utilizamos en los forms fueron subclaseados para permitir que implementen una interfaz y poder hacer validaciones polimórficas entre otros métodos que entienden, así no se repite tanto código (implementan ISuperControls).

- Creamos métodos de extensiones en Extensions.cs y Validations.cs también (está comentado en las respectivas clases)

- En vez de validar que un usuario no ingrese en los textbox caracteres que no debe, ej.: en documento ingresar 10 caracteres y encima ingresó letras, directamente desde la aplicación se restringe la longitud de caracteres que puede ingresar y se lanza el evento onWrite que permite que si no es un dígito no lo escriba. Análogamente para otros textbox que requieren solamente caracteres alfanuméricos, o letras.

- En el form aeronave, en vez de permitir que el usuario checkee tipo de servicio encomienda o pasaje, tenemos dos numericUpDown que al incrementarse luego en la grid va a aparecer si eligió los dos servicios o uno.

- Creamos un objeto SqlConnector que hace polimórficas a las conexiones con la base de datos y una capa DAO que conecta la base de datos (usando el sqlConnector) con la aplicación.

- Consideramos que cada pasajero podía elegir una cantidad de kilos para mandar por encomienda. Entonces cuando se hace la compra de un pasaje se define la cantidad de kgs que se van a mandar, y luego en la pantalla siguiente, cada pasajero puede seleccionar lo que desee enviar de encomienda. Si se pasan de los kgs que pidió en el punto anterior, va a aparecer un mensaje que lo indique.

- En las aeronaves consideramos que los filtros por servicio son más útiles que por nombre de aeronave porque nos sirve más visualizar todas las que cumplen determinado servicio que saber cómo se llama.

- En cambio, en las rutas, es mucho mejor filtrar por ciudad origen o destino.

- En los roles no vimos útil el poner un filtro ya que contamos sólo con dos.

**MIGRACIÓN**

* Para la migración de los datos se tuvieron que tomar algunas decisiones para contemplar algunas inconsistencias con la que nos encontramos o simplemente para soportar el nuevo modelo.
* Ante clientes con el mismo DNI, lo cual es un dato inválido, se decidió dejar uno de los repetidos ya que si no se generarían inconsistencias al modelo o habría que cambiar el modelo de base de datos para la tabla clientes en relación a los campos tipo y numero de documento.
* Según se aclaró en el grupo de google de consulta de la página se modelaron las relaciones de la tabla compra respecto a pasaje o encomienda como una relación one-to-one opcional en ambas pero se agrega la validación de que no puede existir una compra que no tenga ninguna de las dos.
* El cálculo de los porcentajes adicionales para las rutas aéreas respecto al servicio que proveen se realizó en base al precio total y al precio de base de un pasaje que se obtuvo del set de datos provisto por la cátedra.
* Se incluyeron dos tipos de roles de usuarios válidos para utilizar el sistema, siendo los mismos “Administrativo” y “Cliente”.
* El usuario con rol Administrativo tiene los privilegios para ejecutar cualquier funcionalidad que el sistema provea. Por otro lado el cliente, solo realiza las funcionalidades de consulta y canje de millas y la devolución de pasaje o encomienda.
* Se insertaron una serie de productos a modo de ejemplo para poder utilizar la funcionalidad de canje de millas.
* Ante la falta de datos provistos se asumió como fecha de alta en el sistema para todas las aeronaves, a la fecha y hora del día en el que el sistema comience a funcionar.
* Con respecto a la funcionalidad de millas de pasajero frecuente, se decidió que entre en vigencia para los viajes que todavía no han sido realizados, es decir, que tengan fecha posterior a la fecha de implementación del sistema. Todo viaje realizado antes de dicha fecha no se tomara como válido para la acumulación correspondiente de millas.