

Prueba Practica M06

Índice de contenido

1Descripción del ejercicio.....	3
1.1Base de datos.....	3
1.1.1Enunciado.....	3
1.1.2Esquema inicial.....	5
1.2Enunciado practica.....	5
1.3BBDD.....	7
1.3.1Base de datos modificada.....	7
1.3.2Diagrama UML.....	9

1 Descripción del ejercicio

1.1 Base de datos

1.1.1 Enunciado

AENA (Aeropuertos Españoles y Navegación Aérea) quiere crear una aplicación web para gestionar los aeropuertos españoles. Los aeropuertos tienen un nombre único que los identifica y un número de puertas de embarque, que dependen del tamaño de cada aeropuerto. Las puertas de embarque limitan el número de aviones que un aeropuerto puede mantener en tierra. Los aviones son de un modelo determinado y se identifican mediante un código alfanumérico. En cada avión puede ir hasta un número máximo de pasajeros, incluyendo al personal de abordó. Aunque un avión pertenece a una compañía determinada, puede realizar rutas para otras compañías. Los aviones realizan rutas entre dos aeropuertos: el aeropuerto origen y el destino. De estos aeropuertos, al menos uno de ellos está en territorio español. Este dato es importante, ya que la aplicación solo debe gestionar aquellos aviones que utilizan o utilizarán los aeropuertos españoles. Los aviones tienen un horario de salida y de llegada previsto que podrá ser utilizado en los paneles de cada aeropuerto para avisar a los pasajeros. A AENA le gustaría conocer en cada momento el estado de un avión: averiado, retrasado, saliendo, llegando o correcto. Cuando un avión está averiado, ocupa una puerta de embarque que probablemente tarde en liberarse. Un avión retrasado indica que la puerta de embarque debería estar ocupada pero que actualmente no lo está. Cuando se activan los estados saliendo y llegando, se conoce la puerta de embarque desde donde sale un avión o a la que llega. Además, AENA quiere poder añadir más estados desde la aplicación web. Estos estados vienen definidos, también, por un código numérico, que el usuario de la aplicación puede utilizar para buscar un estado determinado. Por otra parte, un avión siempre está en un aeropuerto o realizando alguna ruta entre dos aeropuertos.

Una vez han desembarcado los pasajeros, el aeropuerto destino pasa a ser el aeropuerto origen. No se asigna un nuevo aeropuerto destino hasta que despegue el avión. Para embarcar en un avión los pasajeros, excepto el personal de abordaje, han comprado un billete de avión en el que se reflejan su datos personales (nombre, apellidos, fecha de nacimiento y DNI, siendo este último único por pasajero), el asiento, fecha del vuelo y el código de billete. De esta manera, se puede conocer quién viaja en el avión y cuantos asientos están siendo ocupados.

Hay que considerar que algunos de los posibles pasajeros pueden perder el avión, por lo que no viajarán en él a pesar de haber comprado el billete.

Finalmente, las compañías tienen un nombre y un código de licencia que le permiten operar. AENA esta interesada en almacenar las fechas de concesión de las licencias y su fecha de caducidad con el objetivo de poder renovarlas.

1.1.2 Esquema inicial

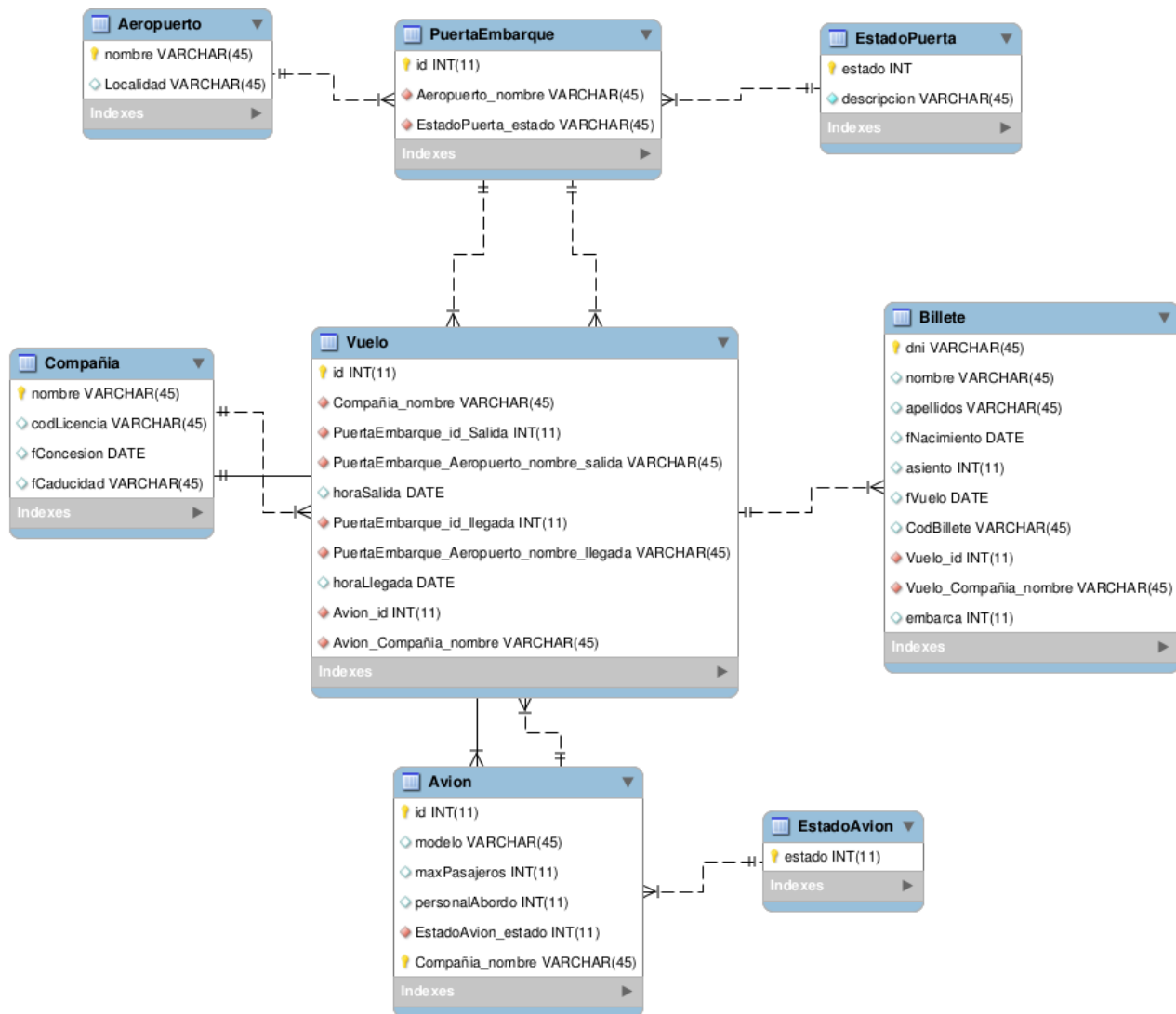


Imagen 1. Esquema Relacional Inicial

1.2 Enunciado practica

En el examen del módulo 3, se realizó el modelo E-R de una aplicación web para que AENA (Aeropuertos Españoles y Navegación Aérea) pudiese gestionar los aeropuertos españoles. Ahora, tú modelo está ubicado en tú carpeta personal dentro del site del curso. A partir de este modelo, se desarrollará individualmente una aplicación con Hibernate que utilice objetos persistentes

considerando los siguientes puntos:

a. El modelo E-R no es estático, de tal forma que los problemas/errores y simplificaciones que se encuentren harán que éste evolucione. La granularidad podría ser un ejemplo de simplificación. Cualquier modificación y decisión de diseño que se tome deberá documentarse.

b. A partir de la versión final del modelo E-R que definas, genera la aplicación, trabajando exclusivamente desde Java utilizando métodos del tipo insertEmpleado(), tal y cómo se ha visto en el módulo. Sin embargo, será más enriquecedor si se conecta Hibernate con otros frameworks.

Esto último permitirá introducir datos de una forma más amigable ahorrando tiempo a la hora de probar y depurar la aplicación.

c. Desarrolla el modelo UML correspondiente a la aplicación, de tal forma que se documente la dirección de las asociaciones implementadas.

d. A nivel de configuración, se deberá establecer con Hibernate las correspondencias de entidades, columnas y relaciones, e implementar la persistencia de objetos. Por lo tanto, dentro de la aplicación se permitirá introducir datos individuales dentro de cada una de las tablas implementadas, y relacionar elementos.

2. AENA ha especificado en el enunciado del módulo 3 funcionalidad extra a implementar. Entre esta funcionalidad, es importante para AENA la implementación y visualización de los resultados de las siguientes consultas, utilizando Criteria o HQL:

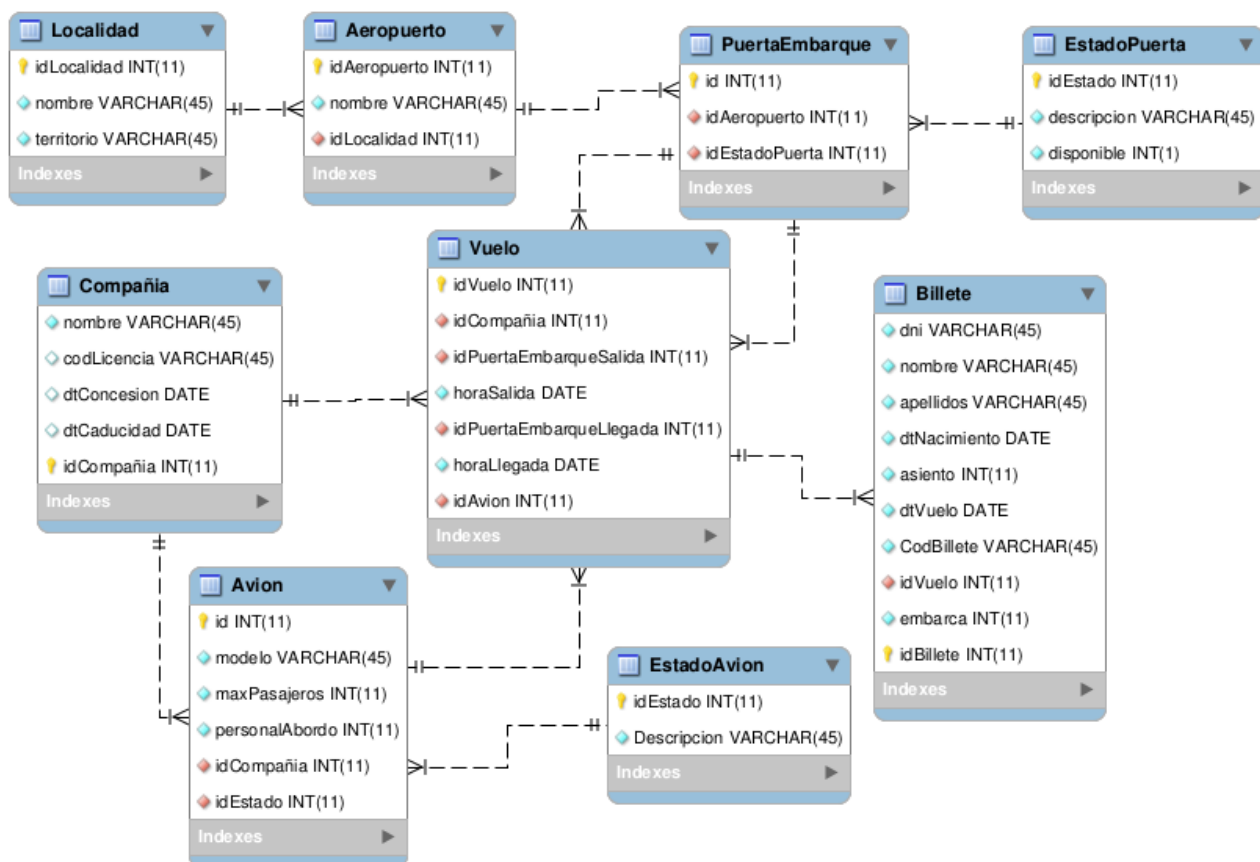
- a. Dado un aeropuerto, determinar el número de puertas de embarque disponibles.
- b. Dado un avión, determinar el número de asientos reservados. Este número se corresponde con el número de billetes comprados para un avión determinado.
- c. Dado un avión, determinar el número de asientos ocupados. Este número se corresponde con el número de personas que están dentro de un avión determinado.
- d. Listar los aviones con licencias de vuelo caducadas.

e. Listar los aviones que utilicen aeropuertos españoles, ya sea porque están en ellos o porque realizan una ruta donde el origen o el destino es español.

Cómo mínimo, hay que realizar las tareas considerando el material visto durante el curso. Se valorarán aquellos conceptos que no se hayan practicado en clase. Por ejemplo, en caso de la herencia, ésta se ha implementado utilizando una tabla por jerarquía. Por lo tanto, se valorarán más otros métodos diferentes. También se valorarán modificaciones del enunciado para realizar mejoras en el sistema de AENA, siempre que estén implementadas. Finalmente, tendrá una valoración especial, la utilización de elementos no vistos durante el módulo.

1.3 BBDD

1.3.1 Base de datos modificada



Se han realizado distintas modificaciones sobre la base de datos para poder seguir el modelo.

Las modificaciones son las siguientes:

- Se ha añadido un id identificativo a cada tabla respetando la estructura existente.
- Se ha creado una nueva tabla localidad para el control del territorio donde esta situado el aeropuerto.
- Para saber si una puerta esta o no disponible para el uso se ha añadido un nuevo campo al estado de la puerta.
- Se ha añadido un campo descripción para que aporte mas información sobre el significado del id.

1.3.2 Diagrama UML

