Tarea AD03 – Breve Informe

Nombre de la BBDD: flyingdb

^^ IMPORTANTE MODIFICAR LOS DATOS DE ACCESO A LA BASE DE DATOS ^^
Los atributos a modificar se encuentran en la clase Conexión, dentro del paquete com.cypherstudios.ad03.dao

El proyecto consta de varios paquetes, siguiendo la norma Modelo, Vista, Controlador.

- **com.cypherestudios.ad03.app :** Contiene la clase main del proyecto.
- **com.cypherestudios.ad03.controller :** Aquí se almacena el controlador de la vista, en nuestro caso con una sola clase es suficiente.
- **com.cypherestudios.ad03.dao :** En este paquete se encuentran todas las clases relacionadas con el acceso y operaciones (CRUD) en la base de datos.
- **com.cypherestudios.ad03.exceptions :** Incluye la clase de excepción personalizada para manejar errores en la app.
- **com.cypherestudios.ad03.interfaces :** Guarda las interfaces que se implementan en las clases DAO, tanto para vuelos como para pasajeros
- **com.cypherestudios.ad03.model :** Contiene los modelos de vuelo y pasajero, asi como sendas clases ArrayList que se usan para insertar datos en la BBD, asi como el constructor de los datos de ejemplo.
- **com.cypherestudios.ad03.utils :** Incluye una clase con métodos de apoyo.
- **com.cypherestudios.ad03.view :** Se encuentra la vista formulario que interactura con el usuario.

Ya que el proyecto utiliza muchos archivos, paso a dar una breve descripción a los más importantes.

Para ampliar información, cada clase tiene sus comentarios y descripciones.

Clase Conexion

Se encarga de gestionar la conexión a la base de datos MySQL.

Esta clase define los parámetros de configuración, como la URL de conexión JDBC, el nombre de la base de datos, el usuario y la contraseña. El método getConexion se encarga de cargar el controlador JDBC y establecer la conexión utilizando los parámetros configurados. En caso de errores durante la conexión, muestra un mensaje de error y finaliza la aplicación. La conexión establecida se devuelve como un objeto Connection.

Métodos Principales:

- getConexion: Establece la conexión con la base de datos.

Clase CtrlOptionsPanel

Controlador que gestiona la lógica de la interfaz gráfica de usuario (GUI) representada por la clase OptionsPanel.

Métodos Principales:

- CtrlOptionsPanel: Constructor que recibe una instancia de OptionsPanel para gestionar sus eventos. (enviada desde la clase main principal de la app)
- launchApp: Inicia la aplicación y muestra la interfaz gráfica.
- evaluateOption: Método para evaluar el evento disparado por el usuario y realizar la operación correspondiente.

Clase FlightDAO

La clase FlightDAO es una implementación de la interfaz IFlightDAO que define operaciones relacionadas con la gestión de vuelos en la base de datos.

Esta clase implementa varias operaciones relacionadas con la gestión de vuelos, como la inserción, listado, creación, eliminación y verificación de existencia de vuelos en la base de datos MySQL. Además, proporciona métodos para listar códigos de vuelo en un ComboBox y verificar la existencia de la tabla de vuelos en la base de datos.

Métodos Principales:

- insertFlights: Inserta vuelos, de ejemplo, en la base de datos a partir de objetos recibidos en un ArrayList.
- createNewFlight: Crea un nuevo vuelo en la base de datos.
- deleteFlight: Elimina todos los vuelos, o el que reciba por parámetro.
- listCodVueloFlight: Escribe los vuelos en el combo Box, o lista desplegable.

Incluye otros métodos, como los propios para comprobar si un vuelo existe contando los vuelos que correspondan, comprobar que la tabla correspondiente existe en la base de datos y si no fuera así, crearla

Clase PassengerDAO

Clase que implementa operaciones relacionadas con la gestión de vuelos en la base de datos.

Esta clase es similar a la anterior, y aunque, comparten similitudes en términos de estructura y enfoque en el acceso a datos, las diferencias clave radican en las operaciones específicas que realizan y en las tablas de la base de datos con las que interactúan.

Tarea para AD03.

Enunciado.

Se trata de hacer una aplicación en Java que acceda a una base de datos Oracle de una aerolínea. Consiste en una conexión a las tablas VUELOS y PASAJEROS de una BD Oracle cuyas tablas y datos de ejemplo se implementarán con el fichero que se adjunta al final del enunciado.

La aplicación desarrollada deberá implementar las siguientes funcionalidades:

1. OK - Dar de alta los datos de ejemplo en la base de datos. (1.5 pto)
2. OK - Eliminar todos los datos de la base de datos. (1.0 pto)

```
DELETE FROM vuelos;

DELETE FROM pasajeros;
```

3. OK - Listar/Mostrar la información de todos los pasajeros. (1.5 pto)

```
SELECT num, cod_vuelo, tipo_plaza, fumador FROM pasajeros;
```

4. OK - Listar la información de los pasajeros de **un vuelo, que el usuario pueda seleccionar**. (1.5 pto) → Mirar de que el usuario pueda elegir el destino del vuelo

```
SELECT * FROM pasajeros WHERE cod_vuelo = ?;

SELECT p.NUM, p.COD_VUELO, p.TIPO_PLAZA, p.FUMADOR FROM pasajeros AS p
JOIN vuelos AS v ON v.COD_VUELO = p.COD_VUELO WHERE v.cod_vuelo = ?;
```

5. OK - Dar de alta un nuevo vuelo con todos sus valores. (1.5 pto)

```
INSERT INTO vuelos VALUES(?, ?, ?, ?, ?, ?, ?, ?);
```

6. OK - Eliminar un vuelo existente en la base de datos. (1.5 pto)

```
DELETE num, cod_vuelo, tipo_plaza, fumador FROM vuelos WHERE cod_vuelo =
?;
```

7. OK - Modificar los pasajeros de **un vuelo (seleccionado desde un desplegable)** permitiendo cambiar de fumadores a no fumadores. (1.5 pto)

```
SELECT * FROM pasajeros WHERE cod_vuelo = ?; -> Para mostrar los campos de
los pasajeros del vuelo elegido, en un JTable

UPDATE pasajeros SET fumador = ? WHERE num = ? AND cod_vuelo = ?;
```

Elabora un proyecto NetBeans el cual será enviado para evaluar, que resuelva los puntos anteriormente mencionados.