

## Descripción general del proyecto

Este proyecto es una 'Java Application' que utiliza hibernate sobre base de datos MySQL para la persistencia de los datos. La configuración de hibernate se ha realizado en base a ficheros xml.

***Nota importante:*** *se ha intentado realizar la práctica bajo los principios del 'agilismo':*

- *producto incremental.*
- *añadiendo valor / funcionalidad en cada 'sprint'.*
- *el objetivo era poner en practica los conocimientos de persistencia, no realizar el mejor diseño o abarcar el alcance total del ejercicio propuesto.*

## Descripción de los archivos principales del del proyecto

- `src/hibernate.cfg.xml`: fichero de configuración:
  - `Session-factory`.
  - `Mappings`.
- `src/hibernate.properties`: fichero de configuración hibernate con la base de datos.
- `src/log.txt`: log de progreso del proyecto.
- `src/bbdd/Aena.sql`: fichero 'sql' que contine las secuencias para crear y configurar la base de datos MySQL utilizada en el proyecto. Contiene la inserción manual de algunos datos a fin de comprobar la aplicación.
- `src/resources/com/innova4b/aena/persistent`: mappings utilizados en la aplicación:
  - `Airplane.hbm.xml`
  - `Airport.hbm.xml`
  - `Boardingpass.hbm.xml`
  - `Gate.hbm.xml`
- `src/diagramas/er`: diagrama Entidad-Relación.
- `src/diagramas/uml`: diagrama UML.
- `src/com/innova4b/aena/dao`: interfaces de acceso a los datos.
  - `AirplaneDAO.java`
  - `AirportDAO.java`
- `src/com/innova4b/aena/daoImpl`: implementación de los interfaces de acceso a datos.
  - `HibernateUtil.java`: singleton de la session factory de acceso a la base de datos via hibernate.
  - `AirplaneDAOImpl.java`
  - `AirportDAOImpl.java`
- `src/com/innova4b/aena/persistent`: POJO's persistentes.
  - `Airplane.java`
  - `Airport.java`
  - `Boardingpass.java`
  - `Gate.java`
- `src/com/innova4b/aena/trans`: main de la aplicación java.

## Opciones de ejecución del proyecto

El proyecto responde a los siguiente comando:

- `getAllAirplanes`: muestra la información referente a todos los aviones de la base de datos.
- `newAirplane plateNumber`: crea un nuevo avión en la base de datos. No permite repetir matrícula (`plateNumber`).
- `deleteAirplaneById plateNumber`: elimina de la base de datos el avión con la matrícula indicada. Eliminará todos los billetes (`boardingpasses`) relacionados con el avión.
- `getAirplaneById plateNumber`: muestra la información referente al avión con la matrícula especificada.
- `configureAirplaneById plateNumber totalNumSeats totalNumSeatsToBook`: configura el avión con la matrícula especificada: número total de asientos y número total de billetes que se pueden vender / reservar (`book`).
- `bookSeatAtAirplaneById plateNumber`: crea una reserva (`boardingpass`) para el avión con la matrícula indicada si no ha sobrepasado el número máximo configurado para el mismo.
- `confirmSeatAtAirplaneById plateNumber`: confirma que una reserva (`boardingpass`) se ha utilizado (el pasajero ha subido al avión) si el avión con la matrícula indicada tiene alguna reserva disponible.
- `getAllAirports`: muestra la información referente a todos los aeropuertos de la base de datos: nombre, número de puertas (`gate`), número de puertas disponibles y cuáles son y número de puertas no disponibles y cuáles son.
- `GetAirportById name`: muestra la información del aeropuerto especificado: nombre, número de puertas (`gate`), número de puertas disponibles y cuáles son y número de puertas no disponibles y cuáles son.
- `newAirport name`: crea un nuevo aeropuerto con el nombre indicado. No permite repetir nombre.
- `deleteAirport name`: elimina el aeropuerto con el nombre indicado. Eliminará también todas las puertas (`gates`) asociadas al mismo.
- `addGateToAirport name`: añadir una puerta (`gate`) al aeropuerto indicado. La nueva puerta tendrá como número uno más que la anterior (se debería de realizar un control más completo: si se añaden y borran puertas, es fácil llegar a tener varias puertas con el mismo número (que no id) ). Las nuevas puertas siempre están disponibles.
- `deleteGateFromAirport name gateNumber`: borra la puerta (`gate`) indicada del aeropuerto indicado.

- `changeGateStatusFromAirport` `name` `gateNumber`: cambia el estado de la puerta indicada del aeropuerto indicado. Si está disponible a no disponible, y viceversa.
- `Dfgdfg`
- `gatesAvailableAtAirport` `airportName` `metodo`: dado un aeropuerto, determina el número de puertas de embarque (gates) disponibles. Si `metodo` es 'HQL' realiza la consulta vía HQL, si `metodo` es 'Criteria' realiza la consulta vía Criteria.
- `numBoardingpassesBookedFromAirplane` `plateNumber` `metodo`: dado un avión, determina el número de asientos reservados (booked). Este número se corresponde con el número de billetes comprados para un avión determinado. Si `metodo` es 'HQL' realiza la consulta vía HQL, si `metodo` es 'Criteria' realiza la consulta vía Criteria.
- `NumBoardingpassesConfirmedFromAirplane` `plateNumber` `metodo`: dado un avión, determina el número de asientos ocupados. Este número se corresponde con el número de personas que están dentro de un avión determinado. Si `metodo` es 'HQL' realiza la consulta vía HQL, si `metodo` es 'Criteria' realiza la consulta vía Criteria.