

Nombre y Apellidos

EJERCICIO 1 CHANGE.ORG

El proyecto se enmarca dentro de una aplicación de reserva de coches, llamada Avis. Esta empresa se dedica al alquiler de vehículos a nivel nacional e internacional.



Tu reserva

Total

✓ **Dónde y cuándo**

18 feb 10:00 → 28 abr 10:00

Bilbao (centro de la ciudad)- Bilbao (centro de la ciudad)

EDITAR

2 Vehículo

Elige un vehículo

3 Cobertura







Elige un paquete de cobertura del vehículo

4 Extras

Elige los extras

12 COCHES EN LA LISTA SIGUIENTE DESDE 3 214,29 €

[Consulta nuestros Términos y Condiciones](#)

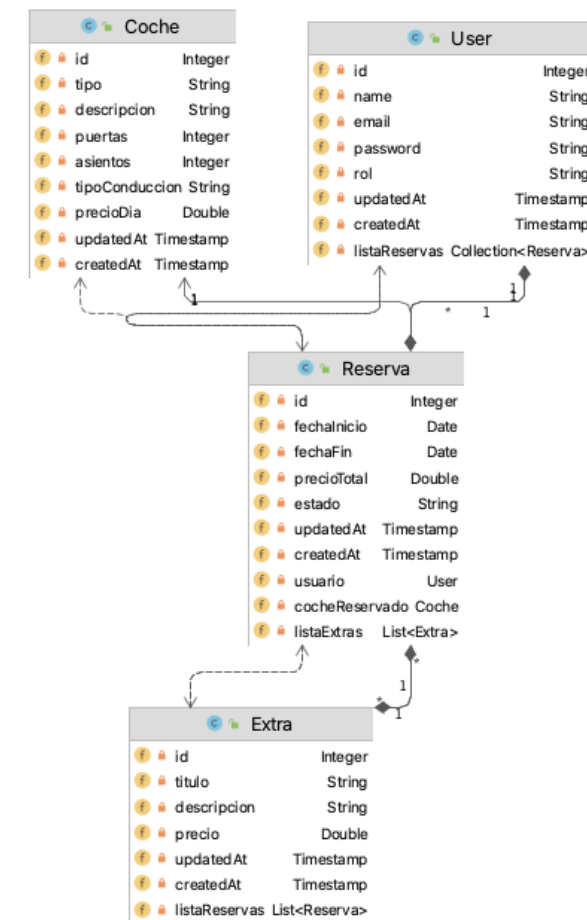
Pequeño/s <small>Por ejemplo Fiat Panda</small> 4 asientos 3 puertas Manual  PAGAR AL RECOGER 3 214,29 € SELECCIONAR <small>Cancelación sin coste*</small> <small>*Condiciones de cancelación: ?</small>	Pequeño/s <small>Por ejemplo Seat Ibiza</small> 5 asientos 5 puertas Manual  PAGAR AL RECOGER 3 535,73 € SELECCIONAR <small>Cancelación sin coste*</small> <small>*Condiciones de cancelación: ?</small>	Mediano/s <small>Por ejemplo Seat Leon</small> 5 asientos 5 puertas Manual  PAGAR AL RECOGER 4 062,95 € SELECCIONAR <small>Cancelación sin coste*</small> <small>*Condiciones de cancelación: ?</small>
Mediano/s <small>Por ejemplo Opel Crossland X</small> 5 asientos 5 puertas Manual  PAGAR AL RECOGER 5 284,38 € SELECCIONAR <small>Cancelación sin coste*</small>	Mediano/s <small>Por ejemplo Renault Captur</small> 5 asientos 5 puertas Automático  PAGAR AL RECOGER 6 338,56 € SELECCIONAR <small>Cancelación sin coste*</small>	Monovolúmen <small>Por ejemplo Renault Scenic (5+2 seats)</small> 5 asientos 5 puertas Manual  PAGAR AL RECOGER 6 338,56 € SELECCIONAR <small>Cancelación sin coste*</small>

En concreto, la funcionalidad que vamos a implementar está sobre todo centrada en el módulo de Reservas. Cuando una persona inicia una reserva para un rango de fechas concreto, puede elegir el tipo de coche que quiere alquilar y añadirle los extras que pueda necesitar.

Teniendo eso en cuenta os presento el diagrama de clases que tenéis que implementar así como el modelo de datos que lo soporta.



Powered by yFiles



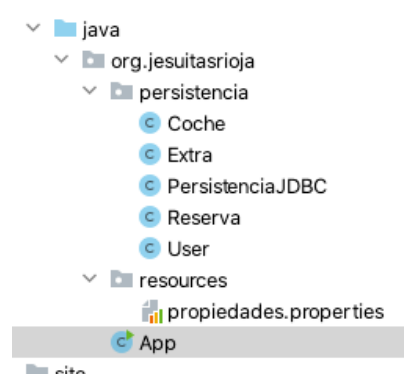
PASOS PREVIOS

NOTA: los siguientes pasos los realizarás junto con el profesor. No te adelantes a sus explicaciones. Una vez terminadas sus indicaciones, se dará paso al examen.

1. Lanza el script proporcionado que te creará la BBDD AD_avis.
2. Deberás importar los dos proyectos que se te proporcionan, uno para JDBC (Ejercicio 1.1→**P1R_Ejercicio1_JDBC**), y el otro para JPA-Hibernate (Ejercicio 1.2→**P1R_Ejercicio1_JPA**). Al final del examen tendrás que comprimir los proyectos a en un fichero llamado **P1R_NombreApellidos.zip**.
3. Configura las **dependencias** adecuadas al gestor de base de datos en cada uno de los proyectos.
4. Configura el fichero de configuración de persistencia de acuerdo con las características del gestor de base de datos en el proyecto JPA.

EJERCICIO 1.1 – JDBC (3,75 PUNTOS)

Dentro del proyecto JDBC tendrás la siguiente estructura de paquetes:



- a. org.jesuitasrioja.resources: paquete con el fichero properties donde configurarás la conexión a la BBDD.
 - b. org.jesuitasrioja.persistencia: paquete con las clases de la capa de persistencia.
 - c. App.java: fichero de prueba con el método main correspondiente.
1. **(0,5 pto.)** Modifica el fichero de propiedades que contenga los parámetros de conexión JDBC. Deberás utilizar dicho fichero para obtener los valores de la conexión.
 2. **(0.5 pto.)** Crea los POJO de forma conveniente.
 3. **(2.75 ptos.)** Implementa el siguiente método dentro de la clase PersistenciaJDBC atendiendo a la siguiente descripción:

(2.75 ptos.) aniadirExtras

```
public boolean aniadirExtras(Integer reservaId, List<Extra> extras) {
```

Añadirá los extras seleccionados por el usuario a la reserva indicada. El método debe comprobar que la reserva indicada existe, y que además cada extra también, antes de proceder a la inserción de cada uno de ellos. En caso de ocurrir algún error se volverá al estado original antes de la ejecución del método.

Parametros:

reservaId– identificador de la reserva a la cual se quieren añadir los extras.

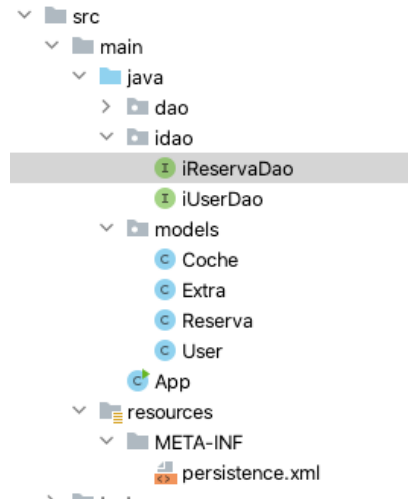
extras – Listado de extras a añadir.

Returns:

Devuelve true en caso de que se hayan podido añadir los extras a la reserva. False en caso contrario

EJERCICIO 1.2 – JPA-HIBERNATE (5.75 PUNTOS)

Dentro del proyecto JPA tendrás la siguiente estructura de paquetes:



- models: paquete con las clases correspondientes al modelo
 - iDao: paquete con las interfaces DAO
 - dao: paquete con las clases de la capa de persistencia que deberán usar una clase abstracta *DAO* que proporcionará el acceso a datos mediante el objeto *EntityManager* y además tendrá implementados los métodos básicos del CRUD.
 - App.java: donde deberás probar TODOS los métodos dao implementados.
- (0.25 ptos.) Crea una clase Dao abstracta dentro del paquete dao. Utilízala en el resto de clases de persistencia convenientemente.
 - (1 pto) Crea las anotaciones de persistencia necesarias para las entidades Coche, Extra, Reserva y User y usarlas así mediante Hibernate cumpliendo el estándar JPA. Debes representar tanto el mapeo de campos con las tablas de BBDD así como las relaciones entre las entidades.
 - (0.25 ptos.) Modifica el fichero de configuración persistence.xml para configurar la conexión convenientemente. También deberá mostrar por consola cada sentencia SQL ejecutada.
 - (4.25 puntos.) Implementa los siguientes métodos conforme a la especificación siguiente:

(0.5 ptos.) aniadirReserva

```
Reserva aniadirReserva(Reserva r)
```

Añadirá una nueva reserva a la BBDD. Dicha petición estará por defecto en estado 'pendiente' (de ser pagado) y el precio_total será 0 (de momento).

Parámetros:

r – Reserva que se desea añadir a la base de datos.

Returns:

Devuelve el objeto Reserva añadido a la BBDD.

(0.75 ptos.) `cambiarCocheReserva`

```
boolean cambiarCocheReserva(Integer reservaId, Integer cocheId)
```

Cambia el coche asociado a una reserva hecha anteriormente. Admite dos parámetros de entrada.

Parámetros:

reservaId – código de la reserva de la que se quiere cambiar el coche

cocheId – identificador del tipo de coche al que se quiere cambiar la reserva

Returns:

Devuelve true en caso de que se haya podido cambiar el coche. False en caso contrario.

(0.5 ptos.) `eliminarReserva`

```
boolean eliminarReserva(Reserva reserva)
```

Eliminará una reserva de la base de datos. Al eliminar dicha reserva, comprueba que se eliminan también los extras asociados a dicha reserva.

Parámetros:

reserva – objeto Reserva que se desea eliminar.

Returns:

Devuelve true en caso de que se haya podido eliminar la reserva. False en caso contrario.

(0.75 ptos.) `esReservaPagada`

```
boolean esReservaPagada(Integer reservaId)
```

Devolverá si una reserva está en estado “pagado” o no.

Parámetros:

reservaId – identificador de la reserva de la cual se quiere comprobar si está pagada.

Returns:

Devuelve true en caso de que la reserva esté en estado “pagado”. False en caso contrario

(1 pto.) `añadirExtras`

```
boolean añadirExtrasReserva(Integer reservaId, List<Extra> extras)
```

Añadirá a la base de datos la lista de extras que el usuario haya seleccionado en la aplicación. En caso de ocurrir algún error se volverá al estado original antes de la ejecución del método.

Parámetros

reservaId – identificador de la reserva a la cual se quieren añadir los extras.

extras – Listado de extras a añadir.

Returns:

Devuelve true en caso de que se hayan podido añadir los extras a la reserva. False en caso contrario

(0,75 pto.) listarReservasUsuario

```
List<Reserva> listarReservasUsuario(Integer usuarioId)
```

Este método listará las reservas que tiene asociados el usuario pasado por parámetro.

Parámetros

usuarioId– identificador de la reserva a la cual se quieren añadir los extras.

Returns:

Devuelve el listado de Reservas asociados al usuario.

EXTENSIÓN (0.5 PTOS)

Si te hago la siguiente afirmación, indica si es falsa o verdadera y justifica tu respuesta.
“Trabajar con Hibernate me permite ser más productiv@ una vez he pasado la curva de aprendizaje que es mayor”