Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones web
Curso: 2022 -2023

1.8.- Ejercicio 8: Bases de datos OO

1.- Dado un modelo de datos relativo a coches, para convertir las clases en entidades que JPA pueda usar para almacenar la información, sabemos que se deben cumplir una serie de requisitos como que la clase debe anotarse con @Entity, etc., tal y como hemos visto en el tema. ¿Qué faltas/errores aprecias en el siguiente código?

```
@Entity
public final class Coche {

    private static final long serialVersionUID = 99488577171L;

    protected Coche (int codigo) {  }

}
```

a) Sobra el *final* en la declaración de la clase
b) Falta el *implements Serializable*
c) Sobra el parámetro del constructor o bien falta el constructor por defecto para evitar instanciaciones sin datos.
d) Todas son correctas

Elijo la opción D.

2.- Dando por hecho que hemos creado una instancia de la entidad `Coche` que podría ser por ejemplo el código siguiente:

```
Coche = new Coche("Seat Ibiza", 200, "AL1234F");
```

Empleando este método:

```
public static boolean persistirCoche(EntityManager em, Coche c) {
    boolean ok = false;

    /*1º Iniciamos la transacción.*/
    em.getTransaction().begin();
    try {
        /*2º Persistimos la entidad usando el método persist.*/

        /*3º Concluimos la transacción*/
        em.getTransaction().commit();

        /*4º (opcional, pero recomendable) Desligamos la entidad del
         EntityManager para que cambios en la misma no persistan en
        otro commit posterior */
        em.detach(p);

        ok = true;
```

```
        } catch (EntityExistsException ex) {

           /*5° Excepción que saltará si ya existe una entidad con el mismo ID,
         es conveniente capturarla.*/
            System.out.println("El coche ya existe.");

            /*6° Como la transacción ha ido mal, deshacemos la operación */
            em.getTransaction().rollback();
        }

        return ok;
    }
```

¿Qué código iría en el 2º para persistir la entidad?

a) coche.persist(c);
b) em.persist(coche);
c) em.persist(c);
d) car.persistence(c);

Elijo la opción C.

Nota: El paso 4, sería (em.detach(c);***) es lo malo del copiar/pegar.***

3.- Dado el siguiente código, qué instrucción serviría para eliminar una instancia de la entidad `Coche`

```
public static boolean borrarCoche (EntityManager em, String matricula)
    {
        boolean operacionRealizada = false;
        // Hacemos el find para buscarlo.
        Coche cocheAEliminar = em.find(Coche.class, matricula);
        //Si el existe podremos eliminarlo, en caso contrario no
        if (cocheAEliminar != null)
        {
            //Iniciamos transacción
            em.getTransaction().begin();

            //Borramos el coche


            //Confirmamos las operaciones.
            em.getTransaction().commit();
            operacionRealizada=true;
        }
        return operacionRealizada;
    }
```

a) em.remove(cocheAEliminar);
b) em.delete(cocheAEliminar);
c) em.AllDelete(cocheAEliminar);
d) em.lookDelete(cocheAEliminar);

Elijo la opción A.

4.- Dado el siguiente código para modificar el precio de un producto,

```
    public static boolean actualizarPrecioProducto (EntityManager em,
long idproducto, double precio)
    {
        boolean operacionRealizada=false;
        //Al hacer el find, el producto se convierte en gestionado.
        Producto productoAModificar=em.find(Producto.class,
idproducto);
        if (productoAModificar!=null) //Si el producto existe
        {
            //Establecemos el precio
            productoAModificar.setPrecio(precio);
            em.getTransaction().begin();

            //En este commit se almacenará la entidad

            //Desligamos la entidad, para evitar problemas
            em.detach(productoAModificar);
            operacionRealizada=true;
        }
        return operacionRealizada;
    }
```

Señala que sentencia se usaría para realizar el commit:

a) `get.getModify().commit();`
b) `commit.getTransaction().em();`
c) `producto.getTransaction().commit();`
d) `em.getTransaction().commit();`

Elijo la opción D.