

[Uniovi Virtual](#) / [Mis cursos](#) / [Repositorios de Información \(Grado en Ingeniería Informática del Software\)](#) / [Tema 3](#)

/ [Cuestionario. Desadaptación de impedancias y más.](#)

Pregunta **7**

Respuesta guardada

Puntúa como 1,00

Una vez cargado en memoria el primer objeto del grafo, desde él se puede acceder a los demás objetos asociados. Es como si virtualmente el grafo estuviese cargado por completo en memoria. ¿De qué formas consigue el mapeador este efecto?

Seleccione una:

- ☐ a. Basándose en el principio de localidad espacio temporal (acuérdate de sistemas operativos), el mapeador predice qué zona vas a visitar a continuación y la carga junto con el primer objeto que le pides.
- ☐ b. Carga el objeto que le pides y todos los que están asociados con él.
- ☐ c. Carga el objeto que le pides y todos los que están asociados con él y esto recursivamente.
- ☒ d. Le puedo especificar que los objetos asociados a uno ya cargado los cargue bajo demanda o de forma agresiva.

[Quitar mi elección](#)

Pregunta **8**

Respuesta guardada

Puntúa como 1,00

Dos usuarios en concurrencia cargan copias de la misma entidad. Si cada uno modifica su versión ¿cómo relaciona el mapeador las dos copias con la fila de la tabla de la que provienen?

Seleccione una:

- ☐ a. Es cosa del mapeador, no me afecta para nada como programador.
- ☒ b. Debo definir hashCode() y equals() sobre los atributos mutables de la entidad.
- ☐ c. Porque le habré dicho al mapeador qué atributos del objeto forman la clave en la base de datos.
- ☐ d. Es cosa de la base de datos y no le afecta al mapeador (ni a mí).

[Quitar mi elección](#)

Pregunta **9**

Respuesta guardada

Puntúa como 1,00

Echa un vistazo al Javadoc de las interfaces `java.util.Set` y `java.util.Map`. ¿Qué condición deben cumplir los objetos que se almacenan en `Set` y los que hacen de clave en `Map`?

Seleccione una:

- ☒ a. No pueden cambiar el valor de ningún atributo que afecte al `hashCode()` (y `equals()`) mientras están contenidos en esas colecciones.
- ☐ b. No se especifica ninguna restricción.
- ☐ c. Pueden cambiar el valor del `hashCode()` pero `equals()` debe seguir devolviendo el mismo resultado.
- ☐ d. Pueden cambiar el resultado de `equals()` pero no el de `hashCode()`.

[Quitar mi elección](#)Pregunta **10**

Respuesta guardada

Puntúa como 1,00

¿Sobre qué atributos se deben definir los métodos `hashCode()` y `equals()` para un objeto que representa una entidad?

Seleccione una:

- ☐ a. No hace falta, la implementación por defecto es suficiente.
- ☐ b. Sobre todos.
- ☒ c. Sobre los que definen su identidad.
- ☐ d. Es irrelevante.

[Quitar mi elección](#)Pregunta **11**

Respuesta guardada

Puntúa como 1,00

¿Cuáles de las siguientes son diferencias estructurales?

Seleccione una:

- ☐ a. Granularidad, control de transacciones, subtipado e identidad.
- ☐ b. Granularidad, subtipado, identidad y navegación.
- ☒ c. Subtipado, asociaciones, identidad y granularidad.
- ☐ d. Concurrencia, navegación y cacheado.

[Quitar mi elección](#)

Pregunta **12**

Respuesta guardada

Puntúa como 1,00

La diferencia por navegación se refiere a que:

Seleccione una:

- ☐ a. Partiendo de una entidad se pueda acceder a las que están relacionadas con ella. Eso es posible en un grafo de objetos por las referencias entre ellos y en el modelo relacional haciendo consultas con join.
- ☐ b. En las bases de datos se puede navegar libremente porque para eso son las claves ajenas mientras que en un grafo de objetos esto no es posible ya que éstas no existen.
- ☒ c. En el grafo cargado en memoria, por cada asociación que se navega (a través de getters por ejemplo) hay que indicárselo expresamente al mapeador en el momento de hacerlo y eso no es necesario en las bases de datos.
- ☐ d. Esa diferencia se resuelve con poner las referencias cruzadas entre los objetos.

[Quitar mi elección](#)

* Las respuestas de esta página serán guardadas automáticamente al pulsar el botón 'Siguiente'

[◀ Algunas preguntas interesantes](#)[Cuestionario: Gestión de objetos persistentes ►](#)