

TRABAJO PRÁCTICO

PERSISTIDOR DE OBJETOS

Desarrollar un *persistidor de objetos* que permita almacenar en una base de datos instancias únicas de objetos de diferentes clases.

API

```
public class PersistentObject
{
    // Almacena la instancia del objeto o asociada a la clave sId,
    // o actualiza la instancia existente retornando true o false
    // segun actualiza o almacena.
    // El objeto o puede ser null, en tal caso el valor que se
    // almacenara sera null.
    public boolean store(long sId, Object o){ ... };

    // Devuelve la instancia del objeto o asociada a la clave sId.
    public <T> T load(long sId, Class<T> clazz){ ... };

    // Retorna true o false según exista o no una instancia
    // de clazz (aunque sea null) asociada a la clave sId.
    public boolean exists(long sId, Class<T> clazz){ ... };

    // Retorna (en milisegundos) el tiempo transcurrido
    // desde el ultimo acceso registrado para la clave sId,
    // sin considerar las llamadas a este metodo ni a exists.
    public long elapsedTime(long sId){ ... };

    // retorna y elimina la instancia de clazz vinculada a la
    // clave sId, o retorna null si no existe dicha instancia
    public <T> T delete(long sId, Class<T> clazz){ ... };
}
```

ANNOTATIONS

@Persistable

Permite indicar qué campos de una clase (o sus subclases) deben ser persistidos. Si esta *annotation* se aplica a una clase, entonces todos los campos de la misma serán persistidos.

@NotPersistable

En caso de que la clase esté anotada con @Persistable, esta *annotation* permite indicar qué campos deben ser omitidos.

CONSIDERACIONES

- En caso de que la estructura de un objeto persistido haya cambiado (porque se agregaron, quitaron o modificaron campos de la clase), se deberá actualizar el objeto invocando al método `store`. Si esto no ocurre y se invoca al método `load` sin la correspondiente previa actualización del objeto, el método arrojará la excepción `StructureChangedException`.
- Las clases cuyos objetos podrán ser persistidos pueden tener campos de cualquier tipo, excepto *arrays*. Pero sí colecciones de tipos primitivos o de otros objetos.

SPRING BOOT

El *framework* debe desarrollarse para ser utilizado dentro del contexto de Spring Boot, con JPA.