

Tarea No. 10. ORM

Barrera Pérez Carlos Tonatihu
Profesor: José Asunción Enríquez Zárate
Web Application Development
Grupo: 3CM9

5 de marzo de 2019

Índice

1. Introducción	3
2. Desarrollo	3
2.1. Ventajas y desventajas de los ORM	3
3. Conclusiones	4

1. Introducción

La presente tarea es una investigación sobre el Mapeo Objeto-Relacional (ORM por sus siglas en inglés), sus ventajas y desventajas y con información general sobre este tema.

2. Desarrollo

ORM es un mecanismo que hace posible acceder y manipular objetos sin tener que considerar como esos objetos están relacionados a la fuente de datos. Los ORM permiten a los programadores mantener la consistencia en los objetos a través del tiempo incluso cuando las fuentes de las que provienen, que las consumen o que las aplicaciones que acceden a ellos cambia. [1]

2.1. Ventajas y desventajas de los ORM

Ventajas

- No tienes que escribir código SQL en la mayoría de los casos.
- Te dejan sacar partido a las bondades de la programación orientada a objetos.
- Nos permiten aumentar la reutilización del código y mejorar el mantenimiento del mismo, ya que tenemos un único modelo en un único lugar, que podemos reutilizar en toda la aplicación, y sin mezclar consultas con código o mantener sincronizados los cambios de la base de datos con el modelo y viceversa.
- Hacen muchas cosas por nosotros: desde el acceso a los datos, hasta la conversión de tipos o la internacionalización del modelo de datos. [2]

Desventajas

- Pueden llegar a ser muy complejos.
- No son ligeros por regla general.
- Si no sabes bien lo que estás haciendo y las implicaciones que tiene en el modelo relacional puedes construir modelos que generen consultas monstruosas y muy poco óptimas contra la base de datos, agravando el problema del rendimiento y la eficiencia.

- La configuración inicial que requieren se puede complicar dependiendo de la cantidad de entidades que se manejen y su complejidad, del gestor de datos subyacente, etc. [2]

Al trabajar con Java existen varias opciones a considerar, el ORM más popular es Hibernate de Red Hat. Hay muchos otros como Jooq, ActiveJDBC que trata de emular los Active Records de Ruby On Rails, o QueryDSL, pero en realidad ninguno llega ni por asomo al nivel de uso de Hibernate.

3. Conclusiones

El utilizar ORM puede llegar a ser un arma de doble filo, debido a que en un inicio parece facilitar el desarrollo de aplicación al hacer el acceso de datos una parte más homogénea respecto al resto de desarrollo que se tiene; pero si no se tiene la suficiente experiencia con estas herramientas puede resultar en grandes problemas en la aplicación.

Es por esto que la elección de usar o no debe de estar bien fundamentada y no simplemente usar un ORM porque se ve bonito.

Referencias

- [1] M. Rouse, “Object-relational Mapping (ORM).” <https://searchwindevelopment.techtarget.com/definition/object-relational-mapping>, 2008. [Online; consultado: 5-marzo-2019].
- [2] J. M. Alarcón, “¿Qué es un ORM?.” <https://www.campusmvp.es/recursos/post/que-es-un-orm.aspx>, 2018. [Online; consultado: 5-marzo-2019].