**2. JPA**

A JPA, tratada como *Java Persistence API* é um framework baseado em linguagem Java para persistência de dados, ou seja, para minimizar a complexidade existente em interações relacionais com bases de dados. A especificação passou por uma história cheia de idas e vindas, e obteve algumas versões antes de se firmar na que usamos atualmente. O conjunto final saiu de uma junção de protótipos que diversos desenvolvedores experientes haviam formulado, sendo documentado e então fornecido ao mercado (JavaWorld, 2008).

**2.1. HIBERNATE**

O Hibernate é a mais comum ferramenta dentro da JPA , sendo um modelo para mapeamento de objetos a uma base de dados, abstraindo-os e economizando esforço (MAGALHÃES, 2013).

Podemos citar seus pontos principais como (Toolbox, 2005):

* **Múltiplos Banco de dados**: Totalmente orientado, o hibernate fornece uma configuração para diferentes plataformas de Banco de dados, sendo necessária apenas a troca da codificação a ser usada na aplicação;
* **Performance**: Como uma das principais características, o hibernate economiza performance, e deixa o acesso aos dados transparente e limpo, tratando tudo que é necessário, sem requerer código;
* **Produtividade e linha de aprendizado**: Fácil de ser ensinado e dominado, não existem muitas variâncias, ou seja, quando você o entende, será sempre da mesma maneira. Isso também ajuda em questões de produtividade em grandes corporações.

Não há muitas críticas sob a ferramenta, entretanto, podemos ressaltar duas, que podem influenciar direto em uma aplicação (JAVA4S, 2011):

* Boilerplate (Cópia de Código): O hibernate é fácil de ser utilizado, e segue sempre o mesmo design, porém, em casos onde são necessárias muitas interações com o banco, teremos algumas classes com o mesmo código padrão do Hibernate.
* Lentidão: Em algumas aplicações, ele pode se tornar um pouco mais lento do que o normal, porém, isso ainda não acata em um grande problema, visto todas as suas peculiaridades positivas.