O que é JPA?

JAVA PERSISTENCE API.
UMA API DE PERSISTENCIA QUE POSSUI AMPLO SUPORTE PELA MAIORIA DOS GRANDES PLAYERS DO MERCADO. (APACHE, ORACLE, JBOSS, ETC).

ORM = MAPEAMENTO OBJETO RELACIONAL.

E quais as principais funcionalidades do JPA?

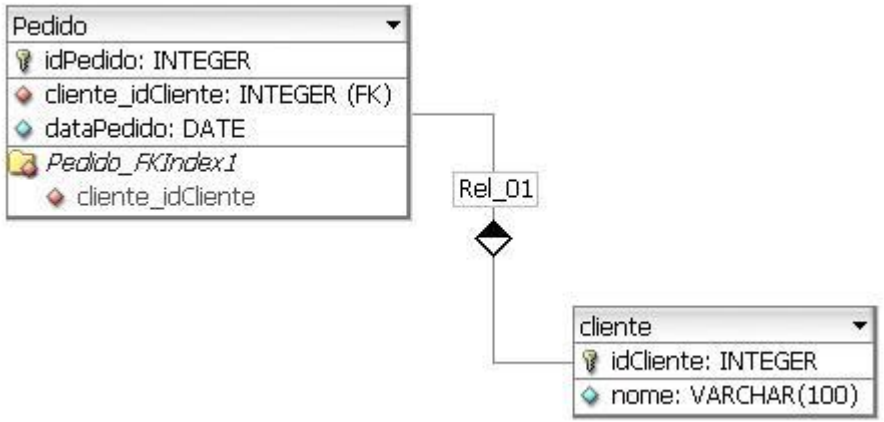
- * Padronizar o Mapeamento Objeto-Relacional
  * Utilizar POJO's ao invés de Entity Beans.
  * Pode ser utilizado com JSE e JEE.
  * Suporta a utilização de diferentes Providers.
  * Suporta herança, polimorfismo.

Pequena receita de bolo que poderá ser seguida.

- 1) Download do JPA Provider (http://jpa.hibernate.org)
  2) Preparar banco de dados e driver JDBC (http://www.mysql.com/downloads/mysql/)
  3) Mapeamento Objeto-Relacional (ORM)
  4) Configurar arquivo persistence.xml
  5) Implementar aceso a dados via EntityManager

Modelo Relacional

- **Modelo Relacional**



Vamos dar “nomes aos bois”, o que usamos para criar um mapeamento.

- **@Entity** => com esta anotação (annotations) dizemos que esta classe Pedido, é uma entidade.
  obs.: Entidade, é um objeto que pode ser persistido.
- **@Table** => Especifica o nome da tabela no banco de dados. Caso fosse necessário, ainda poderíamos ter passado em qual *schema* esta tabela estava localizada (name = “pedido”, schema = “venda”)
- **@Column** => Mapeia um atributo ou uma propriedade (getter) a um campo do banco de dados. Aceita diversas opções de validações.
- **@Id** => Mapeia uma chave primária simples. Podemos gerar a chave automaticamente caso necessite.
- **@Embeddable** => Define que uma classe pode fazer parte de uma entidade Podemos imaginar a tabela, itemPedido, recebendo as chaves da tabela produto e da tabela pedido.

Relacionamentos:

- **@ManyToOne** , **@OneToMany**, @**ManyToMany**
- Agora alguns recursos, que poderão nos ajudar bastante se usados corretamente.

  @**Version =>** Define uma coluna para armazenar a informação de versão para o controle de lock otimista. Este recurso, é interessantíssimo, pois podemos imaginar um cenário que duas pessoas estão trabalhando em um mesmo 'formulário', o de cadastro de pessoas, e coincidentemente, estão atualizando dados da mesma pessoa, o Fulano de Tal. A João terminou a sua atualização primeiro que o Pedro, como nosso mapeamento está corretinho, o Pedro irá receber uma Exception, dizendo algo do tipo: "outra transação".
- Operações em Cascata:
  Persist: Quando uma entidade é persistida, todas as outras nas coleções tbm são.
  Merge: Quando uma entidade desconectada é atualizada, todas as outras entidades na coleção são atualizadas.
  Remove: Quando uma entidade existente é removida, todas as outras na coleção tbm são.
  ALL: Todas a regras anteriores.

Continuação...

- Uma das funcionalidades que acho mais interessantes, são os métodos de Callback, quando usados corretamente, podem facilitar bastante a vida .
- > PostLoad, PostPersist, PostRemove, PostUpDate
  > PrePersist, PreRemove, PreUpdate
- Podem ser utilizados para adicionar funcionalidades extras, como validações, atualizações de tabelas auditadas, etc.
  Em um próximo post, falaremos com mais detalhes dos tão Famosos Métodos de Callback.
- Cenas para os próximos capítulos: Consultas Estáticas, Consultas Dinâmicas, Criação de Objetos, Configuração do Persistence.xml,
- E para concluirmos, fica claro que o JPA nos provê uma API simples e padronizada de persistência, e o uso de annotations simplifica e muito a configuração das entidades.

O que é Spring boot ?

Spring boot:

O Spring Boot é um projeto da Spring que veio para facilitar o processo de configuração e publicação de nossas aplicações. A intenção é ter o seu projeto rodando o mais rápido possível e sem complicação.
Ele consegue isso favorecendo a **convenção sobre a configuração**. Basta que você diga pra ele quais módulos deseja utilizar (WEB, Template, Persistência, Segurança, etc.) que ele vai reconhecer e configurar.
Você escolhe os módulos que deseja através dos *starters* que inclui no pom.xml do seu projeto. Eles, basicamente, são dependências que agrupam outras dependências. Inclusive, como temos esse grupo de dependências representadas pelo *starter*, nosso pom.xml acaba por ficar mais organizado.
Apesar do Spring Boot, através da convenção, já deixar tudo configurado, nada impede que você crie as suas customizações caso sejam necessárias.
O maior benefício do Spring Boot é que ele nos deixa mais livres para pensarmos nas regras de negócio da nossa aplicação.

O que é Hibernate?

Hibernate:

- O Hibernate é um framework para o **mapeamento objeto-relacional** escrito na linguagem Java, mas também é disponível em .Net com o nome NHibernate. Este framework facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) ou anotações Java (veja Annotation (java)).