05

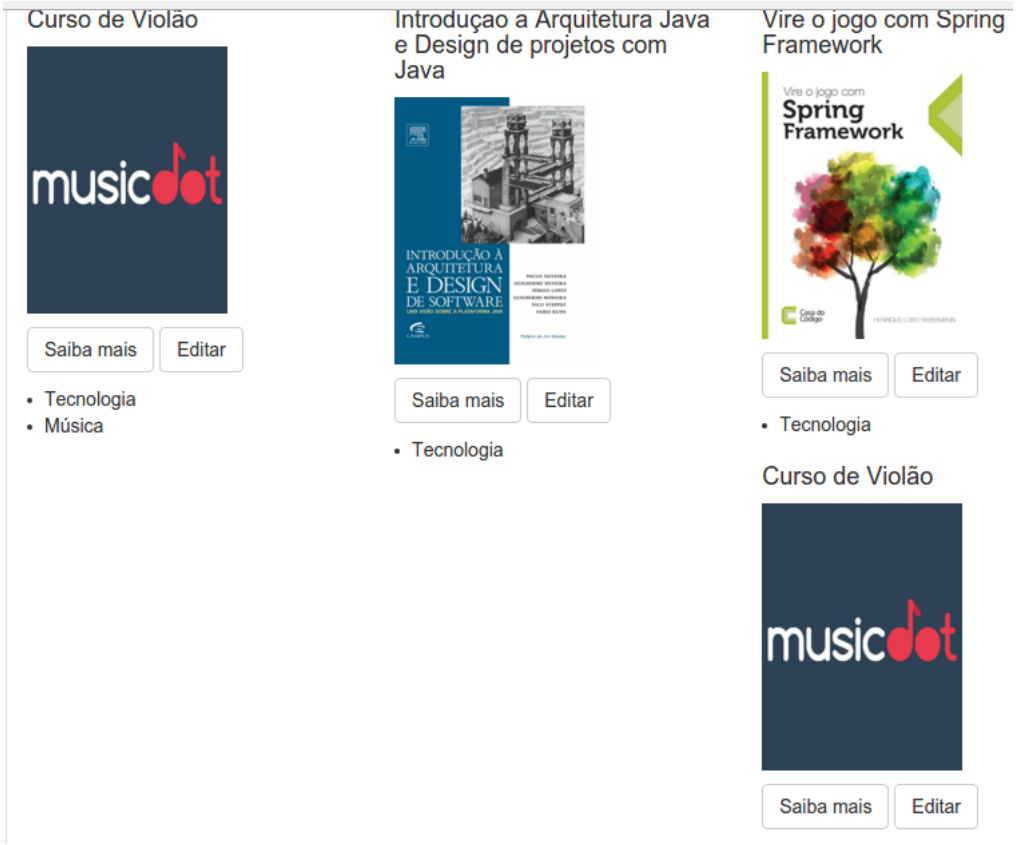
Planejando criteriosamente nossas queries

Para fugir do problema do *lazy load* abrimos o `EntityManager` e o deixamos aberto até o fim da requisição, ou, uma outra estratégica adotada é inicializar todos os dados antes de passar para a *view*. A ideia é planejar antecipadamente quais entidades e relacionamentos serão utilizados e, assim, só devolver dados já carregados sem causar o problema do *lazy load*. Para resolver o nosso problema o JPQL oferece uma forma de pedir explicitamente o carregamento dos relacionamentos. Isto é feito através de um **join** explícito junto à clausula `fetch` .

Por exemplo podemos carregar todas os produtos com as categorias:

```
public List<Produto> getProdutos() {
        return em.createQuery("from Produto p join fetch p.categorias", Produto.class)
                .getResultList();
}
```

Porém ao executarmos a `query` podemos ver que caso um produto possua mais de uma categoria, ele se repete. Como é caso da imagem abaixo:



Em um relacionamento `@ManyToMany` , como já foi visto, temos uma tabela de relacionamento . Essa tabela guarda a associação de cada produto com suas respectivas categorias. Quando buscamos algo usando um `join fetch` são trazidos todos os produtos relacionados aquela categoria. Caso um produto possua mais de uma categoria, ocorrerá repetição pois, haverá mais de uma associação para aquele produto na tabela de relacionamento:

```
mysql> select * from Produto_Categoria;
+------------+---------------+
| Produto_id | categorias_id |
+------------+---------------+
|          1 |             1 |
|          1 |             2 |
|          2 |             1 |
|          3 |             1 |
|          4 |             2 |
|          5 |             2 |
+------------+---------------+
6 rows in set (0.00 sec)
```

Para trazer apenas uma vez cada um dos produtos encontrados usamos a cláusula `distinct` :

```
public List<Produto> getProdutos() {
        return em.createQuery("select distinct p from Produto p join fetch p.categorias", Produt
                .getResultList();
}
```