Criteria API - JPA

É uma API do JPA que ajuda a criar consultas de forma programática ***(através de código Java)***. Ela é bem **poderosa quando se trata de consultas complexas e dinâmicas que com JPQL não dá pra fazer**, porém ela é bem burocrática/verbosa.

Ela **permite que a gente monte uma consulta com código Java** e no final **monta uma consulta SQL pra ser usada no banco de dados.**

Estrutura de uma consulta com criteria API

Precisamos de um **CriteriaBuilder** que é uma interface, ou seja, **ele funciona como uma fábrica que pode ter diversas implementações diferentes de uma instância de CriteriaBuilder.** O EntityManager fornece uma implementação desse builder:

```
CriteriaBuilder builder = entityManager.getCriteriaBuilder();
```

Agora o builder vai oferecer uma instância de uma **CriteriaQuery<T>**, precisamos dessa **CriteriaQuery** pois é ela que **de fato possui as cláusulas da query**, lembra que o propósito da **Criteria API** é fornecer consultas programáticas, pois bem, as **cláusulas da consulta** são **métodos** da **CriteriaQuery:**

```
public List<Restaurante> achar(String nome, BigDecimal taxa){

    CriteriaBuilder builder = entityManager.getCriteriaBuilder();
    CriteriaQuery<Restaurante> criteriaQuery = builder.createQuery(Restaurante.class);

    criteriaQuery.from(Restaurante.class);

    return entityManager.createQuery(criteriaQuery).getResultList();
}
```

==========

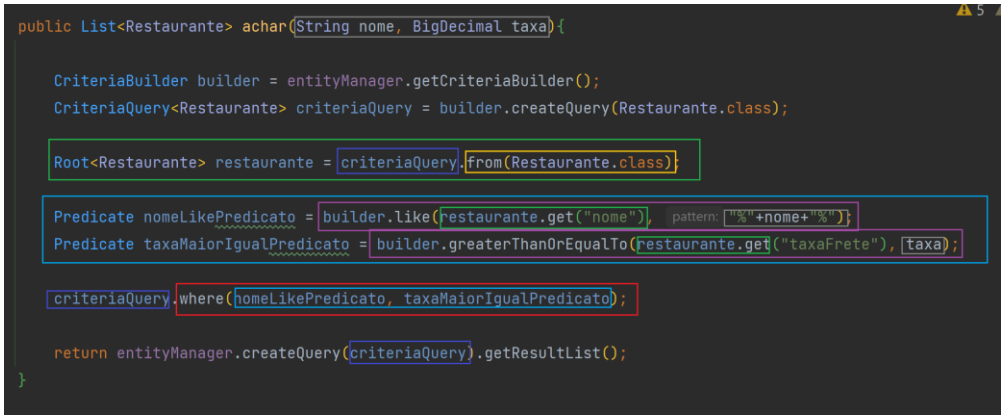
Criteria API com clausulas

Óbvio que o **criteria api** não é para ser usado em consultas simples, ela é usada por consultas que contém diversas clausulas.

O **Builder** funciona como **criador das condições da query**, então ele vai ter métodos como ***>=, distinct, like, <= etc....*** Para que essas condições que são métodos do builder funcionem, eles precisam de um Predicato.

O método from da **Criteriaquery** retorna um **Root** que pode ser usado como predicato, pois ele disponibiliza um **método get que aceita o nome da propriedade desse Root *(o root é a raiz da consulta, representa a entidade consultada).***

Esse predicato recebe a propriedade da ENTIDADE/ROOT da consulta e **o valor que será passado para usar na query.** Agora basta que a **CriteriaQuery** use o where com os predicatos como argumento:



==========

Criteria API filtros dinâmicos

Para tornar os **filtros dinâmicos e montar a query de acordo com os valores passados** é bem simples, basta **usar uma lista de predicatos** que vai ser preenchida se o valor do predicato passar na **verificação:**