

## ARTIGO: 11371

### Uso da API de Criteria

#### Introdução

Nesse artigo veremos como usar a API de Criteria do Veloster Framework. As Criterias são uma forma de consulta feita sobre objetos, que por baixo dos panos, geram as instruções SQL.

#### Objeto Criteria

Para obtermos um objeto Criteria, temos que solicitar a criação do mesmo pelo framework através da classe de serviço:

Java Code

```
Veloster<PersonGroup> veloster = VelosterRepository.getVeloster(PersonGroup.class);

Criteria<PersonGroup> criteria = manager.createCriteria(); // ou através de dynamic Finder RepositoryModel#createCriteria
criteria.add("name", new Like(group.getName(), Match.EXACT));
criteria.add("id", new Eq(group.getId()));

List<PersonGroup> items = criteria.list(); // select PersonGroup where name = ? and id = ?
```

#### Condições

A API de Criteria dispõe de uma serie de condições disponíveis:

Java Code

```
new Between(Object begin, Object end) //between
new Eq(Object value) // equal
new Ne(Object value) // not equal
new Ge(Object value) // greater or equal
new Gt(Object value) // greater than
new Le(Object value) // less or equal
new Lt(Object value) // less than
new Like(Object value, Match match) // like
new Like(Object value, Match match, boolean ignoreCase) // like upper case
new Like(Object value, Match match, boolean ignoreCase, boolean isNot) // like upper case not equal

new IsNull() // is null
new NotIsNull() // not is null

new OrderBy(Object orderBy) // order by
new OrderBy(Object orderBy, boolean desc) // order by desc
```

#### Classe Criteria

##### Condições e ordenação

Java Code

```
Criteria add(String field, Expression expression)
Criteria<T> orderBy(String... fields)
Criteria<T> orderByDesc(String... fields)
```

##### Paginação

Java Code

```
Criteria<T> setLimit(int limit)
Criteria<T> setOffset(int offset)
```

##### Execução

Java Code

```
T load()
List<T> list()
int count()
```