

## ARTIGO: 11398

### Dica em Grails - Utilizando projections

Projections são objetos do tipo Closure inseridos dentro de outra Closure, essa última passado como parâmetro ao criarmos um objeto do tipo Criteria utilizando o método createCriteria do grails ([clique aqui para saber mais sobre createCriteria e projections](#)).

Vamos dar continuidade ao exemplo publicado [nesse artigo](#), e criar um caso de uso prático para saber quando podemos tirar proveito das projections.

Imagine que agora precisamos gerar um relatório para saber quais clientes estão vinculados a quais planos. Poderíamos fazer simplesmente:

Java Code

```
def results = ClientePlano.list()
```

Caso resolvido. Quase.. O problema é que para cada registro o Grails irá trazer o Cliente e Plano com todos os seus atributos. Por razões óbvias não queremos que isso aconteça quando tivermos muitos clientes vinculados.

Vamos utilizar uma projection para obtermos somente o nome do cliente e a descrição do plano, e nada mais.

Java Code

```
def results = ClientePlano.createCriteria().list {
  cliente {
    projections {
      property "nome"
    }
  }

  plano {
    projections {
      property "descricao"
    }
  }
}
```

Note que podemos alterar os resultados operando na relação através de outra Closure, como acima fizemos com cliente e plano. Dessa forma iremos obter uma collection com registros nessa estrutura:

Java Code

```
[ ["Cliente 1", "Plano 1"] , ["Cliente 2", "Plano 2"] ]
```

...com objetos do tipo String e não do tipo Cliente e Plano com todos os atributos, o suficiente para o nosso simples relatório.

Se precisamos ter o telefone e e-mail do cliente, podemos facilmente adicionar essas propriedades:

Java Code

```
cliente {
  projections {
    property "telefone"
    property "email"
    property "nome"
  }
}
```

Projections são úteis para várias coisas como funções de agregação, count, distinct entre outros.

Espero que a dica tenha sido útil.

Até a próxima.