

## ARTIGO: 11397

### Dica em Grails - Modelando Many-to-Many

Quando precisamos modelar uma relação Many-To-Many com grails, podemos fazer de algumas formas diferentes. A maneira que eu prefiro fazer é criar uma classe responsável por unir as duas tabelas. Imagine essa relação:

- Um cliente pode ter muitos planos
- Um plano pode ser de muitos clientes

De forma resumida, podemos ter as classes cliente e plano assim:

Java Code

```
class Cliente {
    String nome

    static hasMany = [clientesPlano: ClientePlano]
}

class Plano {
    String descricao
    Double valor

    static hasMany = [clientesPlano: ClientePlano]
}
```

Observe o atributo hasMany de ambas as classes. A classe ClientePlano irá relacionar, dessa forma:

Java Code

```
class ClientePlano {
    Cliente cliente
    Plano plano
}
```

O benefício de modelarmos dessa forma é que podemos encapsular responsabilidades na classe ClientePlano, possibilitando um melhor controle.

Por exemplo, podemos criar métodos estáticos responsáveis por salvar e remover, a classe ClientePlano ficaria assim:

Java Code

```
class ClientePlano {
    Cliente cliente
    Plano plano

    static ClientePlano link(cliente, plano) {
        def cp = new ClientePlano()
        cliente.addToClientesPlano(cp)
        plano.addToClientesPlano(cp)
        cp.save()
    }

    static ClientePlano unlink(cliente, plano) {
        def cp = ClientePlano.findByClienteAndPlano(cliente, plano)
        if(cp) {
            cliente.removeFromClientesPlano(cp)
            plano.removeFromClientesPlano(cp)
            cp.delete()
        }
    }
}
```

Quem precisar salvar e remover planos, peça para a class ClientePlano, dessa forma:

Java Code

```
// ids 1 de exemplo
def plano = Plano.read(1)
def cliente = Cliente.read(1)
ClientePlano.link(cliente, plano)
```

Modelado dessa forma, seria fácil implementar e encapsular novas funcionalidades, como limite de clientes por plano.

Até a próxima!