

02

## Marcando os produtos como vendidos: condicionais e mais migrations

Já conseguimos efetuar o pedido de uma compra, gravando uma venda no nosso banco de dados, mas o produto continua aparecendo na primeira tela. Produtos comprados não devem aparecer na listagem, como fazer isso?

Uma solução é marcar na nossa tabela quais produtos já foram vendidos, um campo boolean (true ou false). Se vou alterar nosso banco, crio uma migration, agora uma `002_adiciona_vendido_ao_produto.php`.

```
<?php
class Migration_Adiciona_vendido_ao_produto extends CI_Migration {
    public function up() {
    }
    public function down() {
    }
}
```

Para avançar desejamos adicionar uma nova coluna:

```
<?php
class Migration_Adiciona_vendido_ao_produto extends CI_Migration {
    public function up() {
        $this->dbforge->add_column();
    }
    public function down() {
    }
}
```

Quero adicionar na tabela `produtos` um único campo, `vendido`, que é boolean com valor padrão `0` (false):

```
<?php
class Migration_Adiciona_vendido_ao_produto extends CI_Migration {
    public function up() {
        $this->dbforge->add_column('produtos', array(
            'vendido' => array(
                'type' => 'boolean',
                'default' => '0'
            )
        )));
    }
    public function down() {
    }
}
```

E quando desejar voltar atrás em nossa migração, em nossa evolução do banco, desejamos remover a coluna `vendido`:

```
public function down() {
    $this->dbforge->drop_column('produtos', 'vendido');
}
```

Alteramos no `migration.php`:

```
$config['migration_version'] = 2;
```

E agora acessamos `/mercado/utils/migrate`. Ele indica sucesso e conferimos que o campo foi criado em nossa tabela através do phpmyadmin. Mas como o Code Igniter sabe que estou na versão um ou dois do banco? Repara no phpmyadmin que existem as tabelas que criamos e também uma chamada `migrations`, com um único campo chamado `version`, e valor `2`. De acordo com esse campo ele sabe para onde deve migrar. O que acontece se alterarmos nosso `migration.php` para 1 novamente?

```
$config['migration_version'] = 1;
```

Acessamos `/mercado/utils/migrate`, pedi para ele sair da 2 para 1. Abrimos a tabela `produtos` e o campo foi removido, foi `dropado`. Isso pois chamamos o método `down` de nossa migration. Através do arquivo de migration podemos ir ou voltar nas estruturas do nosso banco - tome cuidado pois um `drop remove` também remove os dados, que você não tem como recuperar.

Atualizamos novamente para 2:

```
$config['migration_version'] = 1;
```

Acessamos `/mercado/utils/migrate`. Mas preciso atualizar agora o produto como vendido toda vez que isso acontecer, isto é, sempre que for chamado o método de salvar uma venda em nosso modelo:

```
<?php
class Vendas_model extends CI_Model {
    public function salva($venda) {
        $this->db->insert("vendas", $venda);
    }
}
```

Devemos atualizar também a tabela `produtos` para que o campo `vendido` tenha o valor de `1` para o id do produto vendido:

```
public function salva($venda) {
    $this->db->insert("vendas", $venda);
    $this->db->update("produtos",
        array("vendido" => 1),
        array("id" => $venda["produto_id"]))
};
```

Testamos agora nossa compra e o produto some da lista inicial, se olharmos no banco de dados verificamos que ele tem vendido setado para 1.

