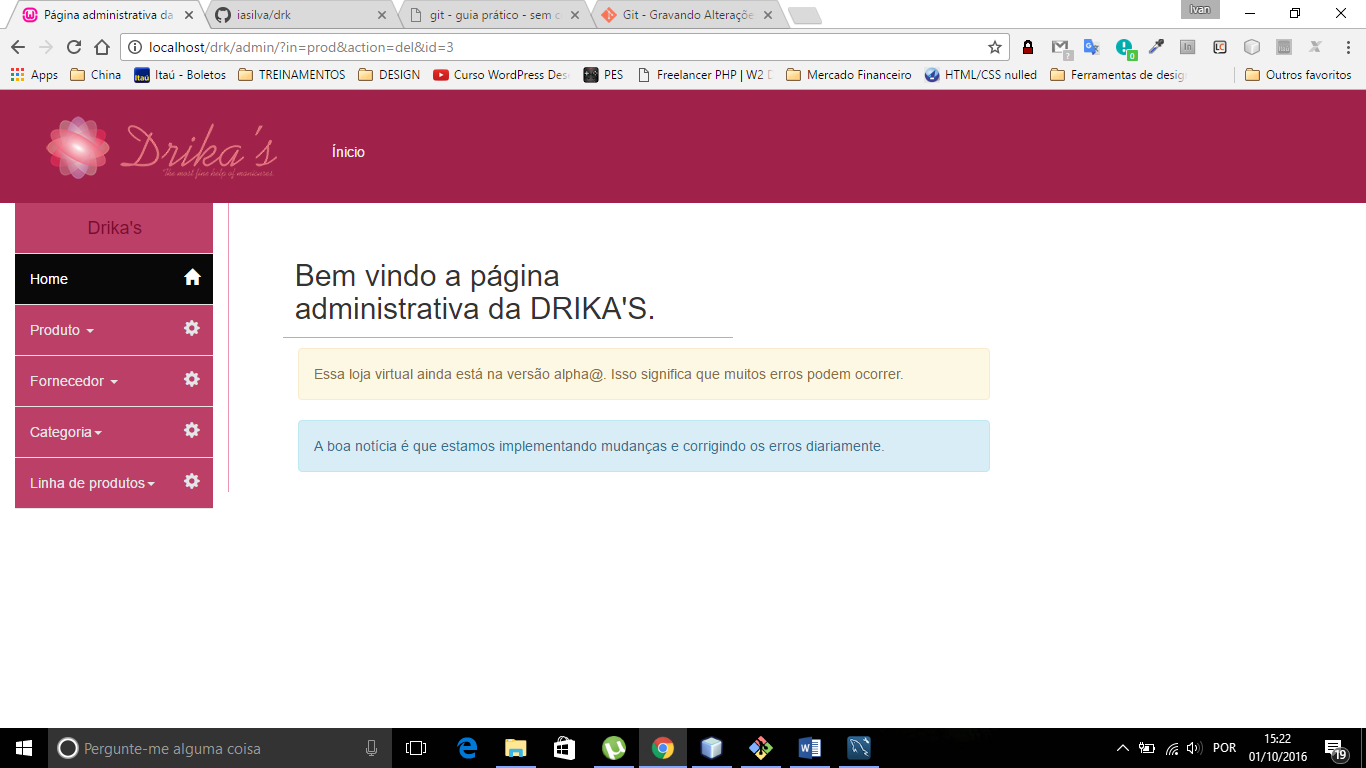
Controller



IN -> indica a tabela em que estamos trabalhando

Action-> indica a ação que desejamos fazer

Id-> indica o código do produto

**Verificações**

1. Identificar qual controller chamar
2. O controller precisa verificar a action que será utilizada
3. A action do coontroller deve utilizar o id e processar a exclusão através do Model

Estrutura dos controllers

Todo controle deve possuir um método verificador <verify()> que deve ser chamado em cada interação na página administrativa.

**verify() –** Primeiramente verifica se a intenção do sistema é falar com ele. Ele vai sempre verificar isso através da variável $\_GET **in**. Geralmente a variável in indicará o prefixo dos campos da tabela.

1. **isWithMe()**

**False->** Retorna FALSE e elimina esse objeto da memória

**True->** Chama o método **action()** do controller que deve dar prosseguimento

**isWithMe() –** Esse método deve ser chamado por verify() e retornar **true** caso exista uma chamada para esse controller ou **false** caso não exista.

**action(string $action**=’GET’**) –** Tem por objetivo chamar os métodos necessários à executar o que está sendo solicitado. Caso não seja passada uma action, por padrão esse método tentará encontrar essa informação na URL.

**Actions:**

**create**-> chamará os métodos responsáveis para salvar dados no Banco de dados.

**up**-> chamará os métodos responsáveis para atualizar dados de um registro no Banco de dados.

**del**-> chamará os métodos responsáveis para excluir um registro no Banco de dados.

**read**-> chamará os métodos responsáveis para buscar dados no Banco de dados.

**view**-> chamará os métodos responsáveis para exibir dados na tela.

Constantes obrigatórias das classes de Controller

**TABLE\_NICK\_NAME**

Refere-se ao apelido da tabela que será usado nas actions. O método **isWithMe()** fará a comparação da variável **in** com o valor dessa constante.

**Atributos obrigatórios e úteis**

**$validActions=array("GET","POST","create","up","del","read","view"[,add])**   
Array com todas as actions que serão consideradas válidas.

Atualização

**actionUp()** Método responsável por desencadear todas as ações necessárias à atualização do dados da tabela.

Ações necessárias:

* + 1. Verifica a passagem de um ‘**id**’ pelo método $\_GET e caso não exista tenta verificar também por POST;
    2. Busca os dados do formulário vindo por $\_POST e configura o objeto com os dados vindos do formulário;
    3. Faz verificações necessárias de integridade nos dados vindos do formulário;
       1. Dados obrigatórios
       2. Formato e tipos de dados
          1. Datas, números e etc...
    4. Atualiza os dados no banco de dados.