

Relatório P3 - Lucas de Paiva Carianha

1 - Padrão de projeto

O padrão de projeto escolhido foi o **Facade**. No projeto ele é o responsável por instanciar todas as classes de gerenciamento do programa, além de ser o responsável por chamar todos os métodos de CRUD de cada classe de gerenciamento, além de chamar os métodos que geram os relatórios. Como pode ser visto na imagem abaixo, todos os métodos de CRUD da classe GerClientes são chamados pela classe MenuFacade, o que acontece com os outros gerenciamentos instanciados.

```
26 public class MenuFacade {
27
28     private GerClientes clientes;
29
30     private Cardapio card;
31     private GerGerentes ger;
32     private GerFuncionarios func;
33     private GerFornecedores forn;
34     private GerProdutos prod;
35     private GerVendas vendas;
36     private GeraRelatorio relatorio;
37     private GerEstoque estoque;
38
126 public boolean addCliente(String nome, String cpf, String email, String telefone) {
127     if (clientes.verificaAtributosCliente(nome, cpf, email, telefone)) {
128         return clientes.add(nome, cpf, email, telefone);
129     } else {
130         Alerts.showAlertComum("Erro", "Um ou mais campos vazios", AlertType.WARNING);
131         return false;
132     }
133
134 }
135
136 public Clientes remCliente(String cod) {
137     if (cod.isBlank()) {
138         Alerts.showAlertComum("Erro", "Codigo vazio", AlertType.WARNING);
139         return null;
140     }
141     return clientes.remove(cod);
142 }
143
144 public Clientes editCliente(Clientes c, String nome, String cpf, String email, String telefone) {
145     if (c != null) {
146         return clientes.edit(c, nome, cpf, email, telefone);
147     }
148     return null;
149 }
```

Dessa forma, o padrão de projeto fica como um agente intermediário do Controller e da View, que acessam os dados da Model pelo MenuFacade de uma forma mais simplificada

2 - MVC

Como mostra a imagem abaixo, cada View de uma função do CRUD possui um pacote próprio, assim como cada **Controller** correspondente possui seu próprio pacote de CRUD, já que para cada função desta é aberta uma janela pop-up diferente no programa. O pacote de **Controller** e **View** do menu do gerente corresponde a respectivamente: controller.gerente e view.gerente; Já o de funcionário são: controller.funcionario e view.funcionario. Outros pacotes de **Views** e **Controllers** como os de relatório e tables, foram feitas pois algumas entidades/métodos precisam de uma janela adicional, ou para ver dados (como os produtos de um fornecedor), ou para pegar algum atributo necessário para a chamada de um método (como pegar as duas datas de um período de venda). E por fim, Os **Controllers** e **Views** responsáveis pelo funcionamento da tela de login.

O Model do projeto possui 3 pacotes: o pacote de entidades, correspondente às entidades que possuem o código único do sistema; o pacote de gerenciamento, correspondente a todas as classes de gerenciamento das entidades; e um model de terminal, na qual fica o MenuFacade citado anteriormente.

