

Programação Orientada a Objetos

Israel Danilo Lacerra #USP 5489057

Eduardo Santos Melo #USP 4897311

Ubiratan Augusto Lima #USP 4899268

Índice

A Arquitetura.....	3
Fase2.....	3
Cardapio.....	3
Ingrediente.....	3
CardapioIngredientes.....	3
GeradoraDeCupom e Registro.....	4
ItemDoCardapio.....	4
Combinado e Prato.....	4
Pedido.....	5
Robo.....	5
Fase2View.....	5
Intro.....	5
Principal.....	5
AdminView.....	5
RoboView.....	6
Adicionando itens ao cardápio: AdminAddIngrediente, AdminAddPrato e AdminAddCombinado	6
Editando itens do cardápio: AdminEditCombinado, AdminEditPrato e AdminEditIngrediente... ..	6
PedidoAddMontado.....	6
PedidoIntroView.....	6
PedidoView.....	6
FecharPedidoView.....	7
Fase2-Test.....	7

A Arquitetura

Procuramos dividir as classes em três pacotes bem definidos: Fase2, Fase2View e Fase2-Tests. A primeira possui as funções normais do sistema do restaurante, muitas delas já entregues na fase 1. A segunda possui as classes que formam a interface gráfica com o SeaSide. A terceira é apenas o pacote com as classes do Sunit. A seguir faremos uma breve descrição das classes e suas funcionalidades.

Fase2

Cardapio

Classe responsável pelo cardápio do restaurante. Possui, além de suas listas, uma variável estática “Ocardapio” que é usada na implementação do singleton. Para obter a instância do cardápio deve-se sempre chamar o método getInstance.

Ao adicionar um combinado ao cardápio, o mesmo é adicionado na lista de combinados desativados, para só depois poder ser ativado.

Ingrediente

Guarda informações sobre um ingrediente.

CardapioIngredientes

Assim como o “Cardapio” também é implementado com um singleton. Funciona de modo geral como o “Cardapio” mas serve apenas para guardar ingredientes disponíveis para se montar os pratos.

GeradoraDeCupom e Registro

A classe Registro apenas guarda informações sobre um cupom de desconto gerado para o cliente.

A classe GeradoraDeCupom é a responsável por instanciar os registros e guardá-los. A implementação também usa o padrão singleton.

Basicamente são guardados os códigos dos cupons na listaDeCodigos, que é uma lista de “Registros”.

ItemDoCardapio

É basicamente o que irá compor o cardápio.

Utilizamos o conceito de Interface/ClasseAbstrata, com esse design foi possível utilizar o padrão estratégia que inseriu flexibilidade. Exemplo disso é o método “prepareSe”, que possui diferente implementação para o Prato e para o Combinado(classes vistas a seguir). Além disso facilitou a criação de pedidos, onde adicionamos Combinados ou Pratos em uma mesma lista, mantendo coesão e fácil acesso a novas alterações.

Combinado e Prato

Subclasses de ItemDoCardapio. Ambos funcionam da mesma forma, com a diferença que o primeiro guarda uma lista de pratos e o segundo guarda uma lista de ingredientes.

O método calorias é apenas um get para a variável, diferente do método getCalorias que soma as calorias de todos os pratos do combinado. Na prática ambos devem retornar o mesmo valor. Coisa parecida acontece com preco e getPreco, onde o primeiro apenas retorna o valor da variável preco e o segundo soma os precos de todos os componentes do item. Aqui os métodos nem sempre retornarão o mesmo valor, pois nem sempre o valor do item será a soma dos valores de seus componentes.

Pedido

Classe que guarda as informações de um pedido.

O atributo “maxCalorias” foi adicionado à classe em função da implementação da funcionalidade sobre o limite de calorias de um pedido. Os atributos “desconto” e “cpf” foram adicionados para implementar a geração de cupons de descontos.

A “listaDePratos” é uma lista de ItemDoCardapio's.

Robo

Também possui um singleton. Em relação ao robo da primeira fase, tivemos que adicionar uma lista de mensagens, pois não fazia mais sentido usar transcript show e ficar imprimindo elas.

Fase2View

Intro

Apenas uma tela inicial do sistema, onde há um único link : “Iniciar”. Ao clicar nesse link, o cardapio e outros singletons são resetados. Para um sistema de verdade isso não deverá acontecer, mas para o nosso isso foi necessário para facilitar os testes.

Principal

Tela simples com duas opções: iniciar um pedido (chama-se a classe PedidoIntroView) ou entrar na área de administração (chama-se a classe AdminView).

AdminView

AdminView é a classe responsável pela principal tela de gerência. Nela são listados os pratos do cardápio, os combinados que estão ativados, os que estão desativados e os ingredientes. Para cada item temos a possibilidade de editar ou remover. É nessa tela também que podemos adicionar algum item, através das opções que são mostradas no topo da janela. Nessas opções também é ver as

mensagens do robo.

RoboView

Tela que apenas imprime as últimas mensagens contidas na lista de mensagens da classe Robo.

Adicionando itens ao cardápio: AdminAddIngrediente, AdminAddPrato e AdminAddCombinado

O cardápio aceita pratos ou combinados. Para se criar qualquer coisa, antes é necessário que se já tenha criado alguns ingredientes. Um prato é simplesmente um conjunto de ingredientes, e um combinado é um conjunto de pratos. As classes responsáveis pela criação de ingredientes, pratos e combinados são, respectivamente, AdminAddIngrediente, AdminAddPrato e AdminAddCombinado.

Editando itens do cardápio: AdminEditCombinado, AdminEditPrato e AdminEditIngrediente.

É importante salientar que ao editar um ingrediente na lista de ingredientes, a mudança não se propaga nos ingredientes que já estão adicionados em um prato. O mesmo acontece com os pratos em relação aos combinados. As telas de edição são muito semelhantes e sem maiores complicações.

PedidoAddMontado

Tela que da opção do cliente montar seu próprio prato. É muito semelhante à tela que da opção de criação de pratos para o gerente.

O preço é obtido somando os preços dos ingredientes.

PedidoIntroView

Tela de entrada para novos pedidos. O CPF do cliente já é obtido para verificar se há desconto cadastrado no sistema. O limite de calorias é também uma das funcionalidades criadas.

PedidoView

Tela onde o pedido é feito. A tela é dividida em dois, onde de um lado é mostrado o cardápio com

todas as suas opções, e do outro é mostrado o status atual do pedido, onde é possível remover itens já adicionados. Ao encerrar o pedido, a classe `FecharPedidoView` é chamada.

FecharPedidoView

Tela que fecha o pedido efetivamente. Ela indica se o cliente terá desconto em uma próxima oportunidade, além de verificar o pagamento e calcular o troco. Ao clicar “preparar”, o robo recebe os pedidos de preparo de todos os pratos e/ou combinados do pedido e o browser volta à tela inicial.

Fase2-Test

Apenas a implementação de alguns testes no SUnit.