

# Exercício Programa 2

## Programação Orientada a Objetos

Guilherme Otranto  
João Enomoto  
Gustavo Braga

### 1 Introdução

Nesse ep foi necessário desenvolver duas interfaces web para o programa feito na primeira fase: uma para o garçom, quem faz efetivamente os pedidos e serve como interface para o cozinheiro e seu robo no preparo dos pratos, e outra para o gerente, análogo a um administrador do sistema que pode fazer alterações no cardápio, entre outras funcionalidades a serem descritas posteriormente.

A implementação foi feita em uma imagem do Squeak 3.9 com Seaside 2.7.

### 2 Interfaces

Optamos por fazer interfaces em HTML bastantes simples, tentando dar um maior enfoque na funcionalidade do que propriamente em recursos estilísticos (folhas de estilo, figuras, etc). A interface do garçom está no pacote "RestauranteFastFood-IfaceGarcom". A interface do gerente é uma classe dentro de "RestauranteFastFood" (não houve tempo suficiente para terminar de fazer o refatoramento da interface).

#### 2.1 Interface do Garçom

Possui as funcionalidades de exibir o menu e realizar pedidos. Para tal, o pedido deve ser inicializado pelo garçom e itens devem ser adicionados através de um menu com botões, cada qual representando um prato do cardápio. É possível para o garçom verificar o que foi pedido bem como o total do pedido até o momento. Assim que todos os itens pedidos forem cadastrados, o garçom deve fechar o pedido recolher o dinheiro. Caso o dinheiro seja suficiente, uma mensagem é transmitida ao cozinheiro para que prepare os pratos e o garçom retorna um troco de acordo com a quantia recebida.

## 2.2 Interface do Gerente

Além das funcionalidades de garçom (pressupomos que o gerente também pode fazer pedidos), duas funcionalidades são garantidas ao gerente: edição do cardápio, permitindo que novos pratos sejam adicionados ou pratos existentes sejam removidos, e uma listagem de mensagens do robô, uma saída de dados das mensagens do robô, informando sobre a preparação dos ingredientes e a montagem dos pratos.

## 3 Funcionalidades Extras

Além do básico desenvolvido para este EP, novas funcionalidades foram adicionadas de modo a incrementar o sistema existente.

### 3.1 Filtro do Cardápio (3 pontos)

Uma funcionalidade extra que decidimos implementar é o filtro de cardápio, onde o usuário de qualquer uma das interfaces pode pedir uma listagem dos pratos no cardápio baseado no preço de cada prato.

Na interface de exibição do cardápio, há opções que permitem definir se o usuário quer ou não limitar seus gastos (mínimo e/ou máximo). De acordo com as opções de filtro, a lista de itens do cardápio é redefinida.

### 3.2 Montagem de Prato (3 pontos)

Criamos um recurso em que o cliente pode especificar os ingredientes com os quais deseja criar um prato totalmente novo. Para tanto, foi criada uma nova classe na camada lógica que representa um ingrediente, com nome e preço. No cardápio foi criada uma lista de ingredientes contendo ingredientes disponíveis para o cliente.

Foi criada uma interface na parte do garçom que lista os vários ingredientes bem como o seu custo e permite ao usuário adicionar cada um deles quantas vezes achar necessário para que o prato fique do seu gosto. O preço calculado é conforme o estipulado, ou seja, o preço de cada ingrediente (por unidade adicionada) mais dois reais. Quando solicitada, a interface fecha o novo prato e o adiciona ao pedido com o nome de pedido personalizado, informando ao usuário qual o prato que foi criado.

Na interface do gerente foi criada uma tela de edição de ingredientes, onde o gerente pode adicionar, editar e remover ingredientes. Essas funções não afetam os pratos já existentes no cardápio que possuem os ingredientes modificados.

### **3.3 Limite nas Calorias (1 ponto)**

Foi adicionada na classe Ingrediente um atributo calorias. Esse atributo também é adicionado nas interfaces, tanto para exibição quanto para edição. A modificação fundamental foi feita na interface de montagem do prato: foram adicionadas restrições ao total de calorias do prato. O limite pode ser superior ou inferior (ou seja, o usuário pode querer ingerir no mínimo ou no máximo um número determinado de calorias). Enquanto o prato não possuir o número de calorias suficientes, o prato não pode ser adicionado ao pedido.

## **4 Testes de aceitação**

Pegamos os testes de aceitação e tentamos ao máximo adaptá-los. Infelizmente não conseguimos fazer com que todos os testes fossem funcionais pelo Selenium. Resolvemos aplicar os testes manualmente para verificar eventuais erros com sucesso.

## **5 Links externos**

Para efeitos de deixar o trabalho no ar, conseguimos reservar um endereço na web para colocar o trabalho: <http://fastfood.seasidehosting.st>. Como não houve tempo suficiente de mexer com elementos estilísticos das interfaces, planejamos (talvez ao final da terceira fase) deixar nesse site ambas interfaces como exemplos de aplicação do framework Seaside.