

### Integrantes do Grupo:

Alexandre Oki Takinami - 5210532  
Carlos Eduardo Manssur - 5122815  
Diogo Vernier do Santos - 5123073

### Sobre a implementação

Para esse Ep2, foi mantido a arquitetura que usamos no Ep1 ( Modelo 3 do enunciado) e o Seaside 2.5 e Squeak 3.9. O sistema foi criado em abas, uma aba do garçom e a aba do gerente. Na interface básica do garçom é possível visualizar todos os pratos do cardápio, fazer o pedido e fechamento do pedido.

Foi criado o pacote *Seaside-RestauranteGrafica* para armazenar a parte gráfica do nosso Restaurante.

Os pratos do cardápio fica armazenado no método *pratosDefault* da classe *Cardapio* (implementado na fase1) , no qual o primeiro parâmetro é o nome do prato, o preço (multiplicado por 100 por causa do formato que é mostrado na página do garçom) e os ingredientes do prato. Esse é o cardápio completo do Restaurante.

```
pratosDefault
""
^ #(
    #('Xburger' 1000 'pao, carne, queijo')
    #('Suco de Laranja' 300 'laranja, gelo')
    #('Beirute' 2800 'pao, carne, queijo, presunto, ovo, tomate, alface, maiones, bacon')
    #('Salada' 900 'alface, tomate, azeite')
    #('Salada 2' 2000 'alface, palmito, azeite')
) asSortedCollection: [:a :b | a first <= b first]
```

Todos os pedidos feitos são adicionados a uma outra tela que mostra o nome do prato, o valor, a quantidade de cada prato e por fim a soma dos valores de todos os pedidos. Para fechamento do pedido é preciso que o garçom digite o valor que o cliente está pagando para o sistema retornar o troco, caso exista. E assim o pedido é enviado para o robo.

### Funcionalidades Novas

**- Filtro de cardápio (3 pontos):** Foi criado o filtro para busca de pratos por nome e por ingrediente. assim o cliente pode ver os pratos e adicionar ao pedido. Para executar a busca, basta digitar no campo o parametro de busca. Caso o prato seja único, já mostra os detalhes do prato.

Exemplos -> Busca: salada - é retornado os pratos salada e salada2 para a escolha do garçom.

-> Busca: queijo - é retornado todos os pratos que contém o ingrediente queijo em sua receita.

-> Busca: beirute - Como esse prato é unico no sistema, já é mostrado a descrição do prato.

Código da Procura: Foi criado o método *match* na classe *Prato* (criado no Ep1) que recebe algum parâmetro de busca (nome do prato ou ingrediente) e retorna a busca.

```
matches: umaChave
^ (nome includesSubstring: umaChave caseSensitive: false)
or: [ingredientes includesSubstring: umaChave caseSensitive: false]
```

### Testes de aceitação

Os testes de aceitação foram feitos usando o plugin do Firefox, o Selenium IDE. Os arquivos de

testes foram exportados em html e foi usado a classe runner do Selenium IDE. Os arquivos seguem anexado ao ep2.

Foram criados os seguintes teste, tela do garçom, fecha pedido certo, fecha pedido errado, cardápio e tela de gerente.