

Programação Orientada a Objetos - Segundo Exercício Programa

Jacqueline Rodrigues 5210320
Márcio Vinícius dos Santos 5146346
Renato Shirakashi de Sousa 5123027

Instalação

O arquivo preamble contém um unico método necessário para a instalação:
PaginaInicial registerAsApplication: 'paginainicial'.

Navegação

A nossa aplicação começa com uma página inicial "Bem vindo" com dois links: "[Acesso Garçom](#)" e "[Acesso Gerente](#)".

Garçom

O link "Acesso Garçom" direciona o usuário para a página "Bem vindo" do garçom. Essa página também contém dois links: Cardápio e Pedido.

Cardápio

O link cardápio da página "Bem vindo" do garçom direciona o usuário para uma listagem do cardápio do restaurante. Essa listagem contém o id de cada item do cardápio, o nome de cada item e o preço, sendo que cada item é um prato do restaurante.

Pedido

O link pedido da página "Bem vindo" do garçom direciona o usuário para uma listagem dos pratos do restaurante disponíveis para a escolha de um cliente. Essa página é para o garçom fazer/registrar um pedido de um cliente. Ao lado de cada prato há um input text para o garçom colocar a quantidade de cada prato que o cliente pediu. Além disso há um botão "fecha pedido". Quando o cliente termina de pedir tudo oque deseja esse botão deve ser pressionado.

Fecha Pedido

O botão "fecha pedido" direciona o usuário para uma listagem dos itens que foram pedidos pelo cliente. Essa lista apresenta o id de cada item, o nome e o preço. Há um input text para o garçom colocar a quantidade de dinheiro que o cliente deu ao garçom para pagar o seu pedido. Nessa tela também há um botão "Calcula Troco".

Calcula Troco

Direciona o garçom à uma página que indica o troco que deve ser retornado ao cliente.

Gerente

O link "Acesso Gerente" direciona o usuário à página "Bem vindo" do Gerente. Essa página contém um único link "Editar Cardápio".

Editar Cardápio

O link editar cardápio da página "Bem vindo" do gerente direciona o usuário para uma listagem do cardápio do restaurante. Porém essa listagem é diferente da listagem do garçom, além de conter o id de cada item do cardápio, o nome de cada item e o preço, ela contém mais dois links para cada item da lista: "editar" e "remover".

Editar

O link editar da página "Editar Cardápio" direciona o usuário para um formulário já preenchido com os dados do item que o usuário deseja editar. O usuário pode então editar o item usando esse formulário e depois clicar no botão atualizar.

Remover

O link remover da página "Editar Cardápio" remove o item que o usuário deseja remover e já faz uma relistagem nessa página.

Classes

Caixa

Essa classe foi implementada na fase 1.

Ela tem a função de *iniciar* um pedido que é feito pelo cliente e *fechar* esse pedido. *Calcular o troco* que deve ser fornecido ao cliente à partir do tanto de dinheiro que o cliente deu para o garçom e do total do seu pedido.

Cardápio

Essa classe foi implementada na fase 1.

Essa classe lida com o cardápio do restaurante. Cria um Cardápio (essa classe é singleton), adiciona e remove pratos a esse cardápio.

Cozinheiro

Essa classe foi implementada na fase 1.

A sua única função é criar um robô e pedir para que esse robô prepare um pedido.

Página Inicial

Essa classe foi implementada na fase 2.

A função dessa classe é montar a página principal da aplicação, a página "Bem vindo" que contém os dois links: "Acesso Garçom" e "Acesso Gerente".

PaginalInicialGarcom

Essa classe foi implementada na fase 2.

A sua função é criar a página inicial do usuário garçom, a página "Bem vindo" que contém os links: "Cardápio" e "Pedido".

PaginalInicialGerente

Essa classe foi implementada na fase 2.

A sua função é criar a página inicial do usuário gerente, a página "Bem vindo" que contém o link: "Editar Cardápio".

Pedido

Essa classe foi implementada na fase 1.

A função dessa classe é montar um pedido. Ela cria um pedido, adiciona itens a esse pedido e calcula o preço do pedido.

PedidoFazComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página onde o usuário garçom irá registrar os pedidos dos clientes.

PedidoListaComponent

Essa classe foi implementada na fase 2.

A função dessa classe é de criar a página que faz a listagem dos itens que o cliente pediu com o input text para o garçom digitar a quantidade de dinheiro que o usuário forneceu para o garçom e com o botão calcula troco.

PedidoQuantidadeDeComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página onde o usuário garçom irá registrar os pedidos dos clientes, escolhendo a quantidade de cada item q o cliente deseja.

Prato

Essa classe foi implementada na fase 1.

Essa classe tem a função de lidar com um prato do restaurante, ela atribui o nome do prato, os ingredientes desse prato e o preço.

PratoAdicionaComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página onde o usuário gerente adiciona novos pratos ao cardápio do restaurante.

PratoEditaComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página onde o usuário gerente vai editar os pratos que pertencem ao cardápio do restaurante.

PratoListaComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página que contém a listagem dos pratos do cardápio.

PratoRemoveComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página de edição e remoção dos pratos que pertencem ao cardápio do restaurante.

Robo

Essa classe foi implementada na fase 1.

Essa classe tem a função de juntar e preparar um prato que foi passado para ele.

TrocoMostraComponent

Essa classe foi implementada na fase 2.

Essa classe tem a função de criar a página onde será exibido o troco que deve ser retornado ao cliente pelo usuário garçom.

Exemplo de código

Para fazer as páginas html usamos sempre um código parecido com esse. Se há uma tabela na página fazemos um loop para fazer a tabela

```
renderContentOn: html
  | prato |
  prato _ Prato new.
  html form: [
    html table:[
      html tableRow: [
        html tableData: [html bold: 'Nome: '].
        html tableData: [html textInputWithCallback: [:v | prato nome: v]; id: 'nome'].
      ].
      html tableRow: [
        html tableData: [html bold: 'Preço: '].
        html tableData: [html textInputWithCallback: [:v | prato preco: v]; id: 'preco'].
      ].
      html tableRow: [
        html tableData: [html bold: 'Ingredientes: '].
        html tableData: [html textInputWithCallback: [:v | prato ingredientes: (v subStrings: ';')]; id:
'ingredientes'].
      ].
    ].
  ].
```

html break.

html submitButtonWithAction: [self edita: prato] text: 'Adicionar'; id: 'prato'.]

Testes

Os testes foram mudados para podermos incluir algumas páginas a mais na aplicação e assim a navegação ficar mais clara.

Importante: muitos testes alteram o cardápio, podendo assim, falhar os outros testes. Para evitar esse problema, use o método de instancia da classe Cardapio **inicializarCardapio**.

Exemplos

Teste Original:

teste-adicao-prato		
open	SUA_PAGINA_DO_GERENTE	
clickAndWait	id=editar	
assertTitle	Edição do Cardápio	
clickAndWait	id=adicionar	
assertTitle	Novo Prato	
type	id=nome	Suco de Uva
type	id=preco	4.00
type	id=ingredientes	uva;água;gelo
submit	id=prato	
assertTitle	Edição do Cardápio	
assertTextPresent	Suco de Uva	

Nosso Teste:

teste-adicao-prato		
open	http://localhost:9090/seaside/paginainicial	
clickAndWait	link=Acesso Gerente	
clickAndWait	link=Editar cardapio	
assertTitle	Editar Cardapio	
clickAndWait	link=Novo Prato	
assertTitle	Novo Prato	
type	id=nome	Suco de Uva
type	id=preco	4
type	id=ingredientes	uva;água;gelo
clickAndWait	id=adicionar	
assertTitle	Editar Cardapio	
assertTextPresent	Suco de Uva	

Teste Original:

teste-cardapio		
open	SUA_PAGINA_INICIAL	
assertTitle	Bem vindo	
clickAndWait	id=cardapio	
assertTitle	Cardápio	
verifyTextPresent	Suco de Laranja	
verifyText	id=sucodelaranja	3.00

Nosso Teste:

teste-cardapio		
open	http://localhost:9090/seaside/paginainicial	
assertTitle	Bem vindo	
clickAndWait	link=Acesso Garcom	
clickAndWait	link=Cardapio	
assertTitle	Cardapio	
verifyTextPresent	Suco de Laranja	
verifyTextPresent	3	

Novas funcionalidades Escolhidas

- Cupons de Desconto.

As classes PedidoListaComponent, TrocoMostraComponent e Caixa implementam juntas essa funcionalidade.

- Montagem de Prato.

A PedidoMontaComponent implementa essa funcionalidade.

- Filtro por preço e ingrediente.