**Testes de Integração**

***Obs:*** *Alguns testes de integração não foram implementados.*

**Classes PEDIDO e ITEM:**

1. Através do teste “testNaoAdicionarItemNoPedidoEstoqueInsuficiente()” está sendo validado que quando um item não tem ingredientes suficientes/não está no estoque, ele não poderá ser adicionado em um pedido.

**pedido.addItem(item)**

1. Através do teste “testAdicionarItemNoPedido()” está sendo validado a adição de um novo item a um pedido quando todos os ingredientes existem no estoque em quantidade suficiente.

**item.getReceita().devolveItensReceitaParaEstoque()  
pedido.addItem(item)**

1. Através do teste “testSettersAndsGetters()” está sendo validado que os dados de um item inserido em um pedido estão retornando com os mesmos valores que foram adicionados.  
   **pedido.addItem(item)  
   item.getCodigo()**

**item.getNome()**

**item.getCusto()**

**item.getValorDeVenda()**

**item.getTempoDePreparo()**

**item.getReceita()**

**item.getEstadoItem()**

**item.getCategoriaItem()**

**Classes PEDIDO e GARCOM:**

1. Através do teste “testGarcomDoPedido()” está sendo validado a inclusão de um novo garçom à um pedido e sendo verificado se os dados do garçom estão retornando com os mesmos valores que foram adicionados.

**pedido.setGarcom((Garcom) garcom)**

**pedido.getGarcom().getPorcentagemComissao()  
pedido.getGarcom().getSalarioFixo()  
pedido.getGarcom().getSalarioVariavel()  
pedido.getGarcom().getSalarioTotalTurno()**

**Classes ITEM e RECEITA:**

1. Através do teste “testInstrucaoDoItem()” está sendo validado que as instruções de receita adicionadas à um item estão retornando conforme foram adicionadas.  
   **item.getReceita().setInstrucoes("Nova Instrução")**

**item.getReceita().getInstrucoes()**

1. Através do teste “testAdicionarNovoIngredienteReceitaDoItemCardapio()” está sendo validado se as instruções de receita de um item do cardápio estão retornando conforme configuradas inicialmente e também se estão retornando corretamente após adicionar um novo ingrediente à receita do item.  
   **item.getReceita().addIngrediente("alho", 0.5)**

**item.getReceita().getIngredientes()**

1. Através do teste ”testReceitaDoItemNovo()” está sendo validado que os ingredientes de um item pertencente à uma receita estão retornando conforme foram adicionados.  
   **item.setReceita(receita)  
   item.getReceita().addIngrediente("alho", 0.5)**

**item.getReceita().getIngredientes()**

1. Através do teste “testItemEstoque()” está sendo validado se os ingredientes de que compõem a receita de um item são suficientes no estoque.  
   **Dados.getInstance().addNoEstoque("linguiças", 2.0)**

**item.getReceita().isEstoqueSuficiente()**

1. Através do teste “testItemRemoverDoEstoque()” está sendo validado se os ingredientes de que compõem a receita de um item podem ser removidos do estoque.  
   **Dados.getInstance().addNoEstoque("linguiças", 2.0)**

**item.getReceita().removeItensReceitaDoEstoque()**

1. Através do teste “testItemDevolverAoEstoque()” está sendo validado que é possível devolver ao estoque todos os ingredientes que compõem a receita de um item de uma única vez.

**item.getReceita().devolveItensReceitaParaEstoque()  
Dados.getInstance().consultaQuantidadeNoEstoque("linguiças")==2.0)**

**Classes CONTA e PEDIDO:**

1. Através do teste “testValidandoPedidos()” está sendo validado a adição de pedidos à uma conta bem como está sendo validado que os dados dos pedidos adicionados estão retornando conforme o esperado.

**pedido.addItem(item)**  
**conta.addPedido(pedido)  
conta.getPedido(pedido.getCodigo()).getValorPedido()  
conta.getPedido(pedido.getCodigo()).getCustoPedido()  
conta.getPedido(pedido.getCodigo()).getEstadoPedido()  
conta.getPedido(pedido.getCodigo()).getGarcom().getID()**

1. Através do teste “testPedidoNaoAdicionadoAConta()” está sendo validado que um pedido vazio não poderá ser adicionado à uma conta.

**conta.addPedido(pedido)  
conta.getPedidos().isEmpty()**

1. Através do teste “testContaSemComissao()” está sendo validado se o custo total, o valor total e a comissão (neste método deve ser igual à zero) de uma conta está com os valores correspondentes aos itens inclusos em cada um dos pedidos adicionados à conta.  
   **pedido.addItem(item)**  
   **conta.setComissaoInclusa(false)  
   conta.addPedido(pedido)  
   conta.fechaConta()  
   conta.getValorTotal()  
   conta.getCustoTotal()  
   conta.getComissao()**
2. Através do teste “testContaComComissao()” está sendo validado se o custo total, o valor total e a comissão (neste método deve ser corresponder à 10% do valor total) de uma conta está com os valores correspondentes aos itens inclusos em cada um dos pedidos adicionados à conta.

**pedido.addItem(item)**  
**conta.setComissaoInclusa(true)  
conta.addPedido(pedido)  
conta.fechaConta()  
conta.getValorTotal()  
conta.getCustoTotal()  
conta.getComissao()**

**Classes CONTA e AUXILIAR DE COZINHA:**

1. Através do teste “testAdicionarAuxiliar()” está sendo validado que após adicionar um auxiliar de cozinha para uma conta, os valores de “salário fixo” e “porcentagem de comissão” estão retornando de acordo com os valores configurados.

**conta.setAuxiliarDeCozinha((AuxiliarDeCozinha) auxiliar)**

**conta.getAuxiliarDeCozinha().getSalarioFixo()==100.0)  
conta.getAuxiliarDeCozinha().getPorcentagemComissao()==0.03)**

**Classes MESA e CONTA:**

1. Através do teste “testVincularMesaComUmaConta()” está sendo validado que após vincular uma conta à uma mesa, os dados da conta estão retornando conforme o esperado, por exemplo o auxiliar de cozinha, o estado da conta, o custo total, o valor total e o código.  
   **mesa.setConta(conta)  
   mesa.getConta().isComissaoInclusa()  
   mesa.getConta().getAuxiliarDeCozinha().getPorcentagemComissao()  
   mesa.getConta().getAuxiliarDeCozinha().getSalarioVariavel()  
   mesa.getConta().getAuxiliarDeCozinha().getSalarioTotalTurno()  
   mesa.getConta().getEstadoConta()**  
   **mesa.getConta().getCustoTotal()  
   mesa.getConta().getValorTotal()  
   mesa.getUltimaConta()**
2. Através do teste “testValidarIDDaUltimaContaDaMesa()” está sendo validado que independente de quantas contas diferentes são vinculadas à uma mesa, o sistema está retornando o código da última conta vinculada corretamente.

**mesa.getUltimaConta()  
mesa.setConta(conta1)  
mesa.getUltimaConta()  
mesa.setConta(conta2)  
mesa.getUltimaConta()  
mesa.setConta(conta3)  
mesa.getUltimaConta()  
mesa.setConta(conta1)  
mesa.getUltimaConta()**