

USE CASE: DESCRÍÇÃO CENÁRIOS:	Fazer Pedido Cliente faz pedido no restaurante 1 e 2	1. Dividir os fluxos 2. Identificar responsabilidades da UI 3. Descrever API (Identificar parâmetros)	4. Identificar sub-sistemas (especificar operações)
PÓS-CONDIÇÃO:	Sistema registra pedido com sucesso e cliente recebe fatura		
FLUXO NORMAL:	<p>1. Sistema mostra no ecrã taléus as propostas e menus disponíveis</p> <p>2. Cliente adiciona propostas e menus ao pedido</p> <p>3. Sistema informa cliente de alergénios</p> <p>4. Cliente escolhe que vai comer no restaurante</p> <p>5. Cliente finaliza o pedido</p> <p>6. Sistema registra a execução do pagamento</p> <p>7. Sistema emite fatura e talão com o número do pedido</p> <p>8. Sistema registra o pedido</p>	<p>Determinar propostas e menus disponíveis obter_emental() : Ementa</p> <p>Determinar alergénios contidos no pedido obter_alergenicos_vPendingente(codPedido : String, idRestaurante : String) : Collection<Pair<String, Ingrediente>></p> <p>Registrar pagamento do pedido regista_pagamento(valor : double, codPedido : String, metodoPagamento : String) : String</p> <p>Cesar fatura e talão referente ao pedido gera_fatura(idPagamento : String, pedido : Pedido) : Fatura & gera_talao(codPedido : String) : Talao</p> <p>Registrar pedido regista_pedido(codPedido : String, idRestaurante : String) : void</p>	Subsistema do Pedido Subsistema do Pedido Subsistema Commercial Subsistema Commercial Subsistema do Pedido
FLUXO ALTERNATIVO	<p>[1] [Cliente quer fazer alterações ao pedido] (passo 5)</p> <p>5.a.1 Cliente seleciona propostas ou menus que deseja alterar</p> <p>5.a.2 Sistema mostra ingredientes que pode alterar</p> <p>5.a.3 Cliente seleciona o que pretende alterar</p> <p>5.a.4 Sistema regista alterações</p> <p>5.a.5 Regressa a 5.</p>	<p>Determinar ingredientes que se pode alterar alterarIngredientesProposta(codMenu : String) : String & lisPropostasIngredientes(codProposta : String) : String</p> <p>Registrar alterações alterarIngredientesProposta(pedido : Pedido, codProposta : String, lista : List<String>, adicionar : boolean) : void & alterarIngredientesPropostaMenu(pedido : Pedido, codMenu : String, codProposta : String, lista : List<String>, adicionar : boolean) : void</p>	Subsistema do Pedido
FLUXO ALTERNATIVO	<p>[2] [Cliente quer adicionar nota ao pedido] (passo 5)</p> <p>5.b.1 Cliente escreve nota</p> <p>5.b.2 Cliente submete a nota</p> <p>5.b.3 Sistema adiciona nota ao pedido</p> <p>5.b.4 Regressa a 5.</p>	<p>Registrar nota do pedido regista_nota(not : String, codPedido : String, idRestaurante : String) : void</p>	Subsistema do Pedido
FLUXO ALTERNATIVO	<p>[3] [Cliente escolhe que o pedido é de take away] (passo 4)</p> <p>4.1 Sistema regista que o pedido é do tipo take away</p> <p>4.2 Regressa a 5.</p>	<p>Registrar tipo de pedido register_tipo_pedido(tipo : String, codPedido : String, idRestaurante : String) : void</p>	Subsistema do Pedido
FLUXO ALTERNATIVO	<p>[4] [Cliente quer fatura com contribuinte] (passo 5)</p> <p>5.c.1 Cliente indica o contribuinte</p> <p>5.c.2 Sistema regista contribuinte</p> <p>5.c.3 Regressa a 5.</p>	<p>Registrar contribuinte do cliente register_contribuinte_Pedido(nc : int, codPedido : String, idRestaurante : String) : void</p>	Subsistema do Pedido
FLUXO DE EXCEÇÃO	<p>[5] [Sistema não recebe pagamento] (passo 6)</p> <p>6.1 Sistema informa cliente que o pagamento não foi recebido</p> <p>6.2 Sistema cancela o pedido</p>	<p>Cancelar pedido cancelar_pedido(codPedido : String, idRestaurante : String) : void</p>	Subsistema do Pedido