

pelo operador (*) e por condições de guarda, as quais são também usadas para estabelecer situações opcionais, representadas no diagrama de sequência por fragmentos combinados do tipo **opt**.

17.2.8 Diagramas de Máquinas de Estados da PizzaNet

Nesta seção, explicaremos os diagramas de máquina de estados referentes aos processos do sistema PizzaNet. Embora esse diagrama possa ser usado para modelar os estados de um único objeto, preferimos modelar os estados dos casos de uso do sistema, pois, uma vez que o sistema que estamos modelando é comercial, o número de estados assumidos por determinados objetos em certos processos é ínfimo, diversas vezes resumindo-se a um único estado. Assim, consideramos mais válido modelar os estados dos processos como um todo.

Diagrama de Máquina de Estados Escolher Pizza

Esse processo se inicia quando o cliente acessa a página da PizzaNet ou pressiona o botão **Pizzas**. Esse evento gera uma transição para o estado **ApresentandoTamanhos**, no qual são apresentados os tamanhos de pizzas oferecidos pela pizzeria. No momento em que todos os tamanhos forem apresentados, passa-se ao estado **ConsultandoSaboresDisponíveis**, em que é executado o método **consultarDescricaoSabor**, conforme demonstra a cláusula **Do**. Após a conclusão desse estado, passa-se ao estado **ApresentandoSabores**, no qual os sabores consultados são apresentados na interface. Em seguida, passa-se ao estado **RecebendoTamanho**, que é um estado estático na qual se espera que o cliente escolha o tamanho de pizza desejado. Quando o tamanho é escolhido, gera-se uma transição para o estado **ConsultandoQuantidadeSabores**, onde é executado o método **consultarQuantidadeSaboresTamanho** para verificar a quantidade máxima de sabores que a pizza pode conter (Figura 17.27).

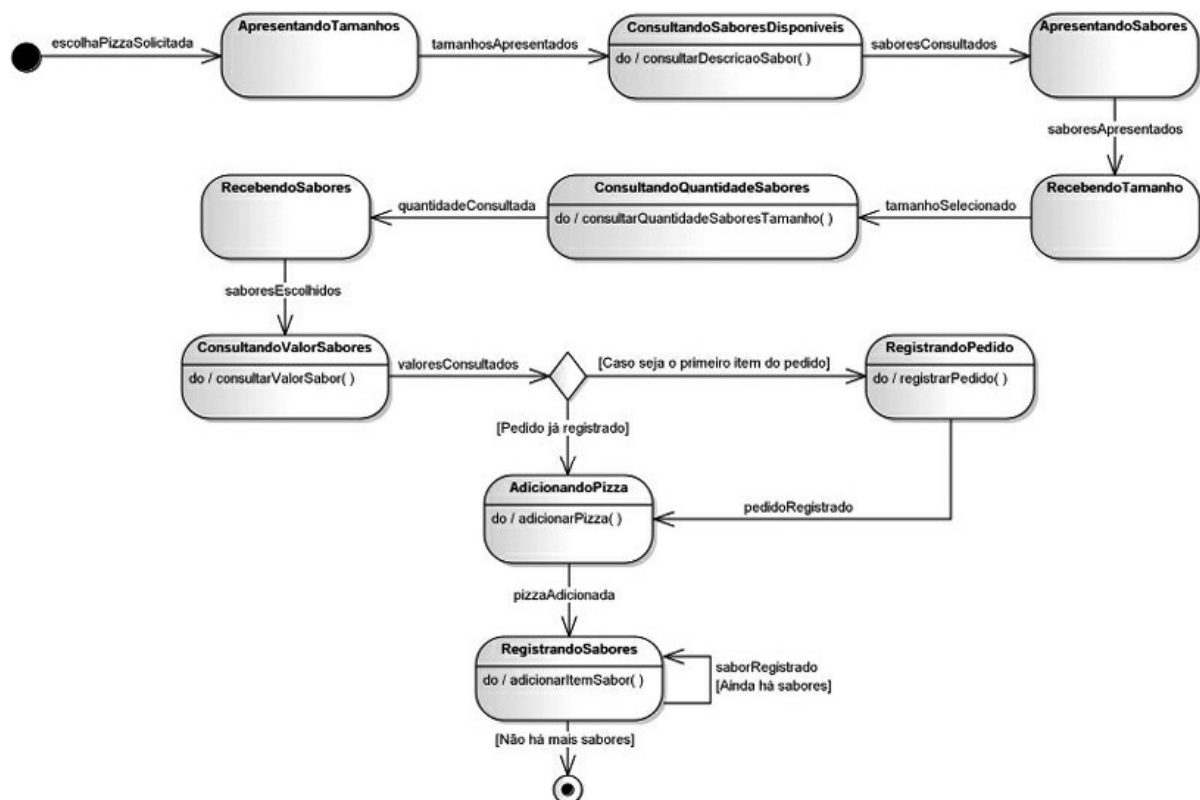


Figura 17.27 – Diagrama de Máquina de Estados Escolher Pizza.

Após a consulta, é gerada uma transição para um novo estado estático, **RecebendoSabores**, em que se espera que o cliente selecione os sabores desejados para a pizza. Em seguida, quando o cliente escolhe os sabores da pizza, é gerada uma transição para o estado **ConsultandoValoresSabores**, em que é executado o método **consultarValorSabor**, para recuperar o valor de cada sabor escolhido. Após a conclusão desse estado, o processo passa a um pseudoestado de escolha, no qual é preciso verificar se o item adicionado é o primeiro item do pedido. Se isso for verdade, gera-se uma transição para o estado **RegistrandoPedido**, no qual é disparado o método **registrarPedido**, para gerar um novo objeto da classe **Pedido**.

Ao concluir o registro do pedido ou se a pizza em questão não for o primeiro item do pedido, passa-se ao estado **AdicionandoPizza**, em que é chamado o método **adicionarPizza**. Quando a pizza for adicionada, gera-se o estado **RegistrandoSabores** e é executado o método **AdicionarItemSabor**, que gera um novo objeto da classe **ItemSabor** para cada sabor escolhido para a pizza. Observe que nesse estado existe uma

autotransição que retornará a esse último estado enquanto houver sabores para registrar.

Diagrama de Máquina de Estados Escolher Bebida

Quando o cliente escolhe essa opção, é gerada uma transição para o estado **ConsultandoTiposBebidas**, no qual são pesquisados todos os tipos de bebidas oferecidas na pizzeria. Nesse estado é executado o método **consultarTipoBebida**. Após os tipos de bebida terem sido consultados, passa-se ao estado **ApresentandoTiposBebidas**, em que são apresentados os tipos consultados. Depois disso, passa-se ao estado **RecebendoTipoBebida**, no qual se espera que o cliente selecione um tipo de bebida (Figura 17.28).

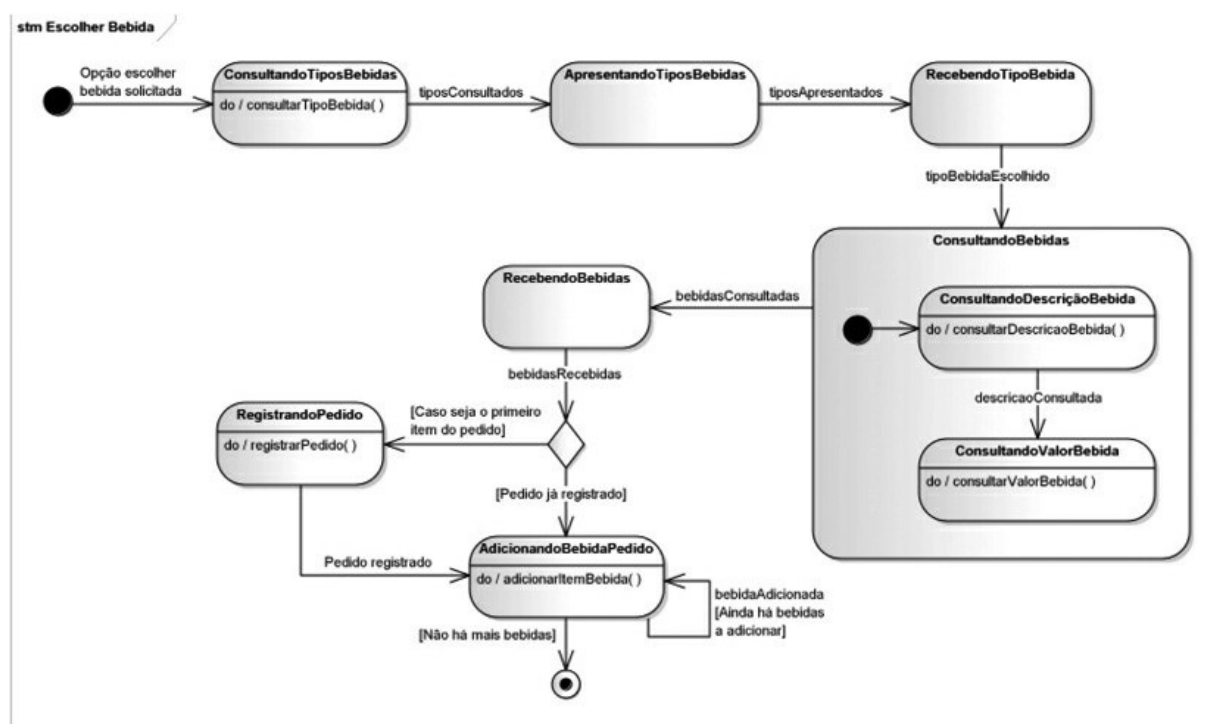


Figura 17.28 – Diagrama de Máquina de Estados Escolher Bebida.

No momento em que o cliente escolher um tipo de bebida, é gerada uma transição para o estado composto **ConsultandoBebidas**, no qual são consultadas as bebidas do tipo escolhido. Nesse estado composto são gerados os subestados **ConsultandoDescriçãoBebida** e **ConsultandoValorBebida** e executados os métodos **consultarDescricaoBebida** e **consultarValorBebida**.

Quando a consulta das bebidas for finalizada, será gerada uma transição

para um estado estático em que se aguarda até que o cliente informe as bebidas que deseja. Após o cliente escolher as bebidas e as quantidades desejadas, gera-se uma transição para um pseudoestado de escolha, em que é preciso verificar se a bebida escolhida refere-se ao primeiro item do pedido. Se esse for o caso, gera-se uma transição para o estado **RegistrandoPedido** e é executado o método **registrarPedido**. A seguir, caso a bebida escolhida não seja o primeiro item do pedido, será gerada uma transição para o estado **AdicionandoBebidaPedido**, em que é executado o método **adicionarItemBebida** para instanciar um novo objeto da classe **ItemBebida** para cada bebida escolhida pelo cliente. Observe que há uma autotransição para esse estado enquanto houver bebidas a adicionar.

Diagrama de Máquina de Estados Logar

No momento em que o cliente seleciona essa opção, uma transição para um pseudoestado de escolha é gerada. Esse pseudoestado representa a decisão do cliente em informar seu nome-login e senha, caso já possua cadastro, ou de se autorregistrar, caso contrário (Figura 17.29).

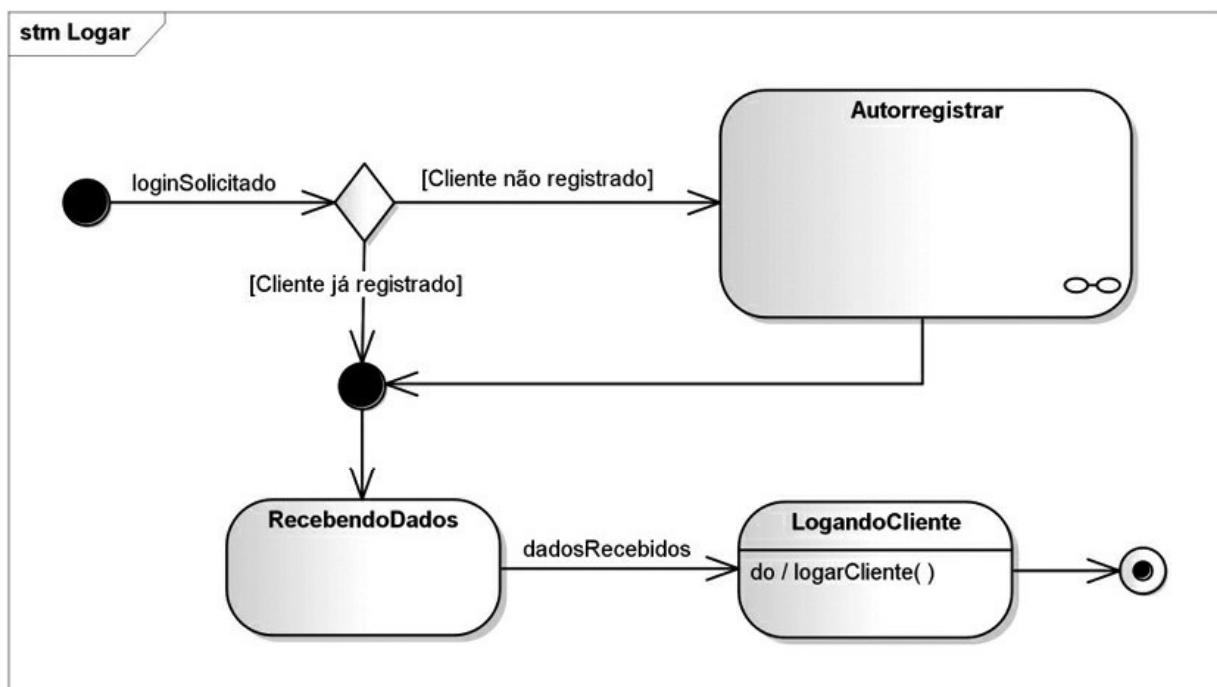


Figura 17.29 – Diagrama Máquina de Estados Logar.

Caso o cliente não esteja registrado, será gerada uma transição para um

estado de submáquina que se refere ao processo de **Autorregistrar**, descrito em outro diagrama. Após se autorregistrar, ou caso o cliente já possua cadastro, passa-se ao estado estático **RecebendoDados**, no qual o cliente informará seu nome-login e senha. Observe que se uniu o fluxo dividido pelo pseudoestado de escolha por meio de um pseudoestado de junção.

No momento em que os dados do cliente forem recebidos, será gerada uma transição para o estado **LogandoCliente**, onde será executado o método **logarCliente**, por meio do qual o cliente será logado no sistema, caso seu nome-login e senha estejam corretos.

Diagrama de Máquina de Estados Autorregistrar

Esse processo envolve apenas dois estados, sendo o primeiro estado **RecebendoDadosCliente**, gerado pela transição causada pela solicitação de registro pelo cliente. Nesse estado, o cliente deve informar seus dados e confirmar. Quando isso ocorre, é gerada uma transição para o estado **RegistrandoCliente**, em que é executado o método **registrarCliente**, para instanciar um novo objeto da classe **Cliente** (Figura 17.30).

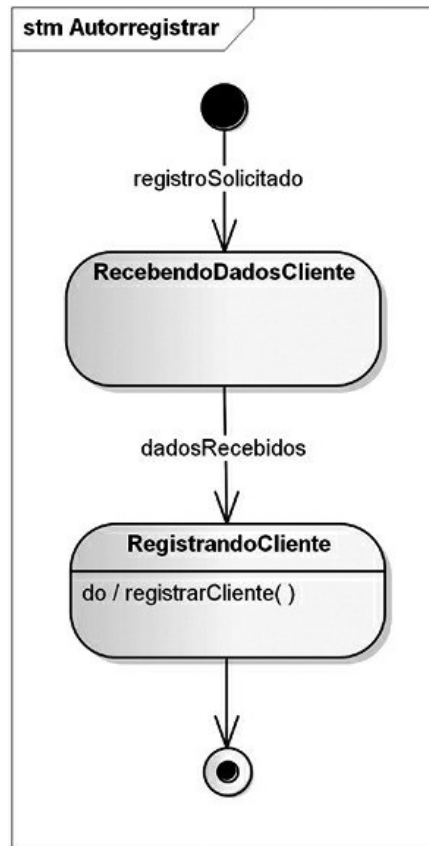


Figura 17.30 – Diagrama de Máquina de Estados Autorregistrar.

Diagrama de Máquina de Estados Opinar

Esse é um diagrama bastante simples composto de apenas dois estados. Inicia-se quando o cliente solicita a execução da funcionalidade de registrar opinião. Isso causa uma transição que gera o estado **RecebendoOpinião**, que espera que o cliente informe a sua opinião e confirme. Quando isso acontece, é gerado o estado **RegistrandoOpinião**, em que é executado o método `registrarOpiniao`, que instancia um novo objeto da classe `Opiniao` (Figura 17.31).