

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

P.PORTO

CTeSP DWDM

Análise e Arquitetura de Sistemas

UML: Diagramas de Interação - Sequência

Conceito

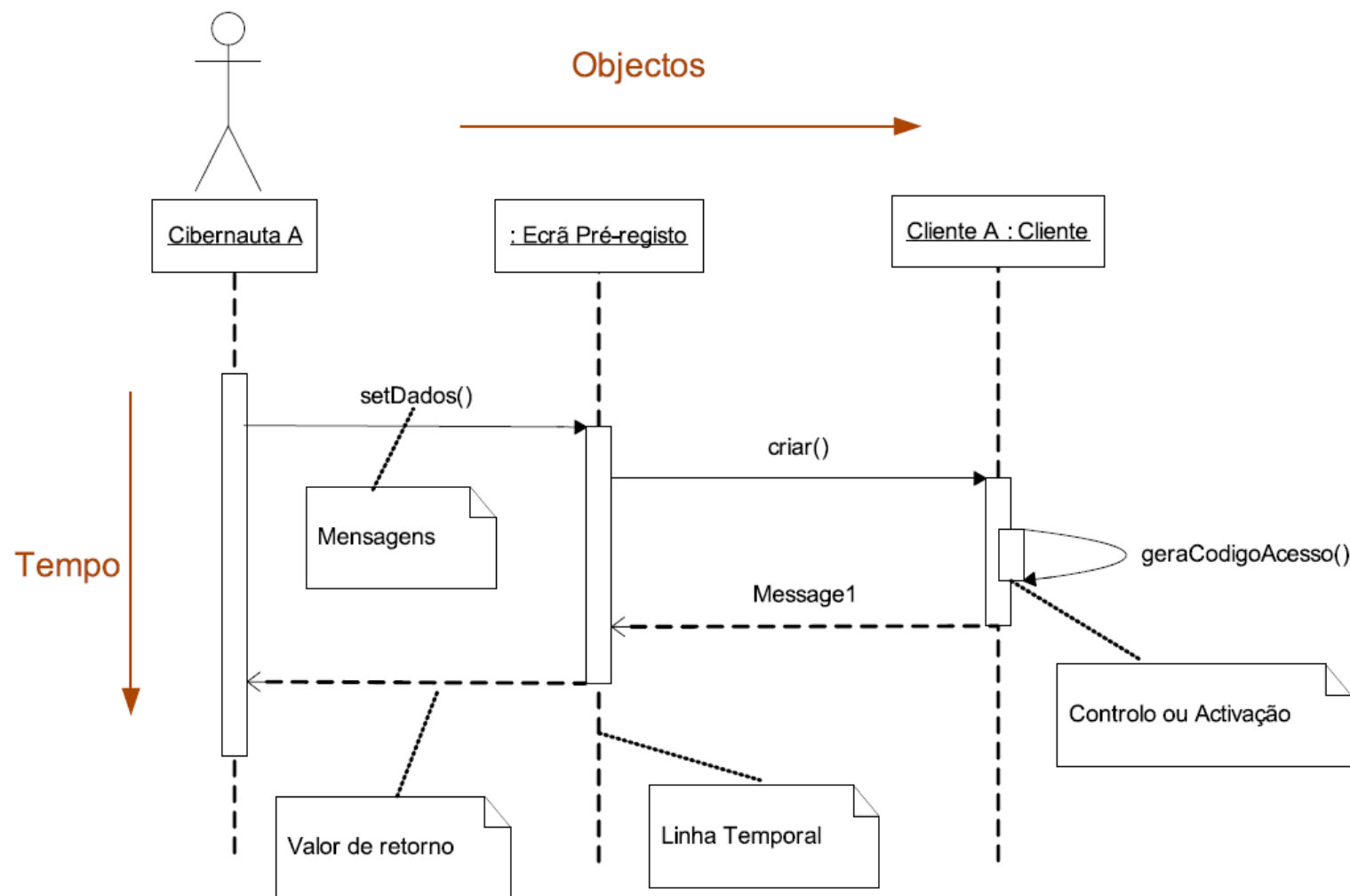
- São utilizados para modelar os aspetos dinâmicos do sistema em termos dos objetos e suas interações, tendo como base as mensagens trocadas entre objetos.
- Permitem definir e clarificar a colaboração entre as classes do sistema.
- Normalmente, são utilizados para ilustrar o comportamento do sistema num cenário de concretização de um use case.
- É frequente utilizar diagramas de interação em conjunto com a descrição textual dos use cases, pois facilitam a sua compreensão ao fornecer uma representação gráfica das interações entre os objetos.
- **Diagrama de interação** é uma designação genérica que na UML se aplica a:
 - **diagrama de sequência** – apresenta as interações entre objetos a partir do encadeamento temporal das mensagens.
 - **diagrama de colaboração** – descreve as mesmas interações mais centradas nos objetos intervenientes.
- Um diagrama de interação é composto pelos seguintes elementos:
 - Objetos
 - Ligações (links)
 - Mensagens

Diagramas de Sequência - Objetivo

- Um diagrama de sequência mostra uma interação, isto é, uma sequência de mensagens trocadas entre vários objetos num determinado contexto.
- Enfatiza a comunicação e passagem de controlo entre objetos ao longo do tempo.
- Útil para descrever uma sequência particular de funcionamento, mas não muitas sequências alternativas e ciclos nem ações realizadas por um objeto que não envolvem comunicação com outros objetos

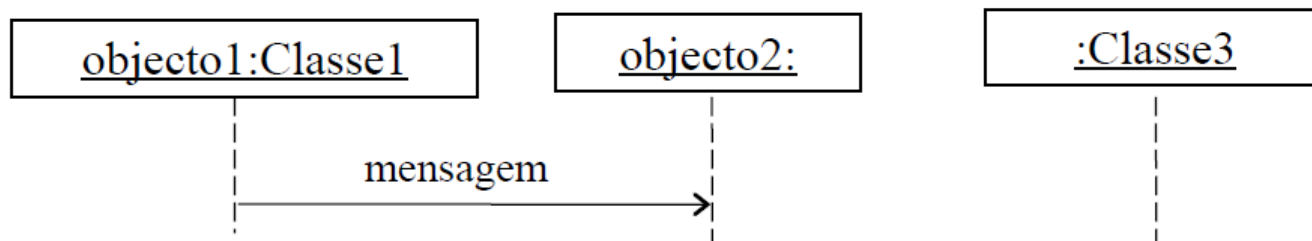
Exemplo

- Vejamos a seguinte descrição para o caso PhonePizza:
 - *“Para que o cliente possa efetuar encomendas através da Internet este terá que efetuar um pré-registo onde indicará o seu nome, morada, número de telefone, username e password. O pré-registo será confirmado através de um código de acesso que será enviado por correio eletrónico. O código será utilizado uma única vez pelo cliente para ativar os serviços de encomenda pela Internet”*
- Esta descrição caracteriza o use case “Pré-registo pela Internet”. Vamos usar um cenário de registo típico para apresentar os diagramas de interação.



Objetos e linha temporal

- Cada objeto participante é representado por uma caixa em cima duma linha vertical a traço interrompido (linha temporal ou linha de vida)
- Podem aparecer atores (objetos externos ao sistema), normalmente a iniciar interações
- O tempo cresce de cima para baixo



Mensagens

- Uma mensagem é uma comunicação entre objetos (emissor e recetor) que veicula informação na expectativa de provocar uma resposta (ação ou atividade)
 - Uma ação de um objeto capaz de provocar uma resposta noutro objeto pode ser modelada como uma mensagem do primeiro para o segundo objeto
- Uma mensagem é representada por uma seta horizontal, do emissor para o recetor, com o nome e possíveis argumentos

Tipos de mensagens:

- **síncrona** (\longrightarrow) – significa que o objeto emissor fica suspenso à espera de uma resposta, retomando posteriormente o controlo.
 - utiliza-se esta mensagem quando o objeto emissor necessita de dados provenientes do objeto recetor, para continuar o seu processamento.
- **assíncrona** (\longrightarrow) - o emissor não fica parado à espera de resposta.
 - particularmente útil para ilustrar sistemas com processos concorrentes.
- **retorno** (\longleftarrow) – utilizada para ilustrar o retorno da mensagem enviada.
 - para mensagens síncronas está implícita a existência de um retorno, sendo a sua representação opcional.
 - Para mensagens assíncronas deve-se representar.
- **simples** (\longrightarrow) - não se decide se é síncrona, de retorno ou assíncrona
 - usadas normalmente na modelação de interações na fronteira do sistema (entre atores e o sistema representado por um ou mais objetos).

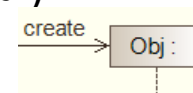
Mensagens

- As mensagens podem despoletar vários tipos de ação no objeto recetor.
- Tipos de ação para as mensagens:
 - **Call** – invoca uma operação de um objeto.
 - O mais utilizado
 - Mensagens síncronas
 - Este tipo de mensagem pode ser enviada ao próprio objeto.
 - **Return** – retorna um valor para o objeto emissor para mensagens síncronas ou um sinal para mensagens assíncronas.
 - **Send** – envia um sinal a um objeto.
 - Mensagens assíncronas
 - **Create** – cria um objeto.
 - **Destroy** – destrói um objeto.

Criação e destruição de objetos

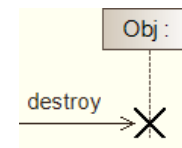
- Apenas os tipos Create e Destroy são explicitamente ilustrados nas mensagens, todas as outras estão implícitas ao tipo de mensagem.
- **Criação de objeto** é representada por mensagem dirigida à própria caixa que representa o objeto (em vez de ser dirigida à linha temporal)

- Mensagem de criação pode ter estereótipo «create»



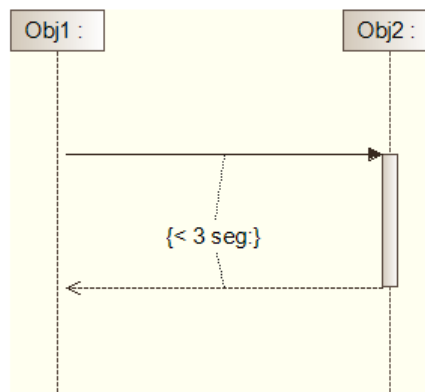
- **Destruição de objeto** é representada por um X no fim da linha temporal do objeto (a representação é opcional)

- Mensagem de destruição pode ter estereótipo «destroy»
- Pode ocorrer na receção de mensagem ou no retorno de chamada
- Objeto pode auto destruir-se



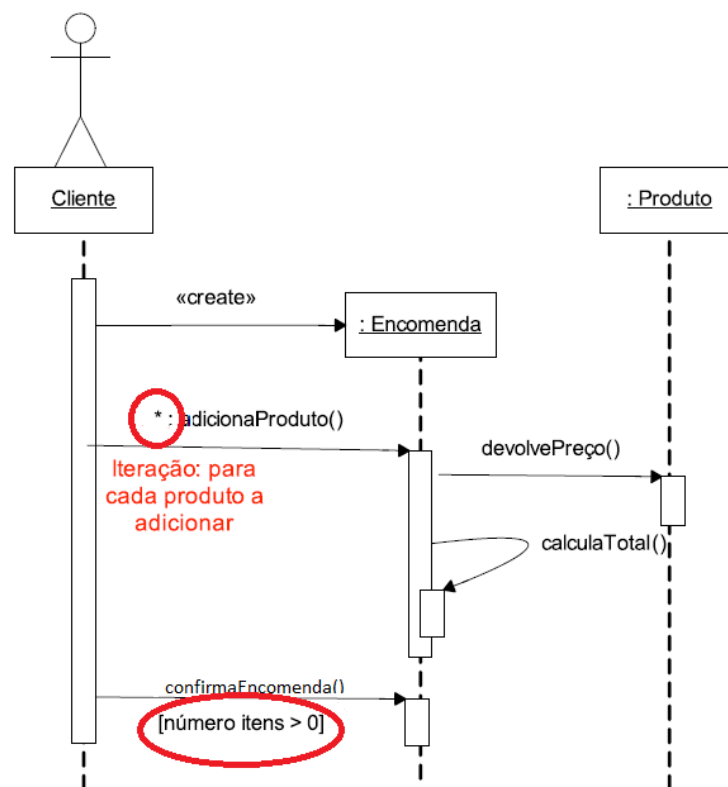
Linha Temporal e Controlo (Ativação)

- **Linha temporal** – acompanha o ciclo de vida dos objetos, onde também pode ser representado o intervalo de tempo.
- É possível impor uma **restrição temporal** entre o envio de uma mensagem e a respetiva resposta.



Linha dupla de controlo, símbolo de iteração e condição

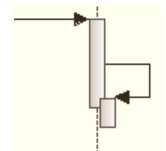
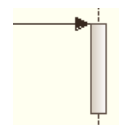
- **Linha dupla de controlo** – o objeto envia uma mensagem a si próprio.
- **Iteração** - é indicada com asterisco *, seguido ou não de uma fórmula de iteração.
- **Condição** – é indicada entre parêntesis retos perto da linha de controlo.
 - De forma a demonstrar que um conjunto de mensagens é enviado de acordo com uma condição, pode-se utilizar uma separação na linha de controlo.



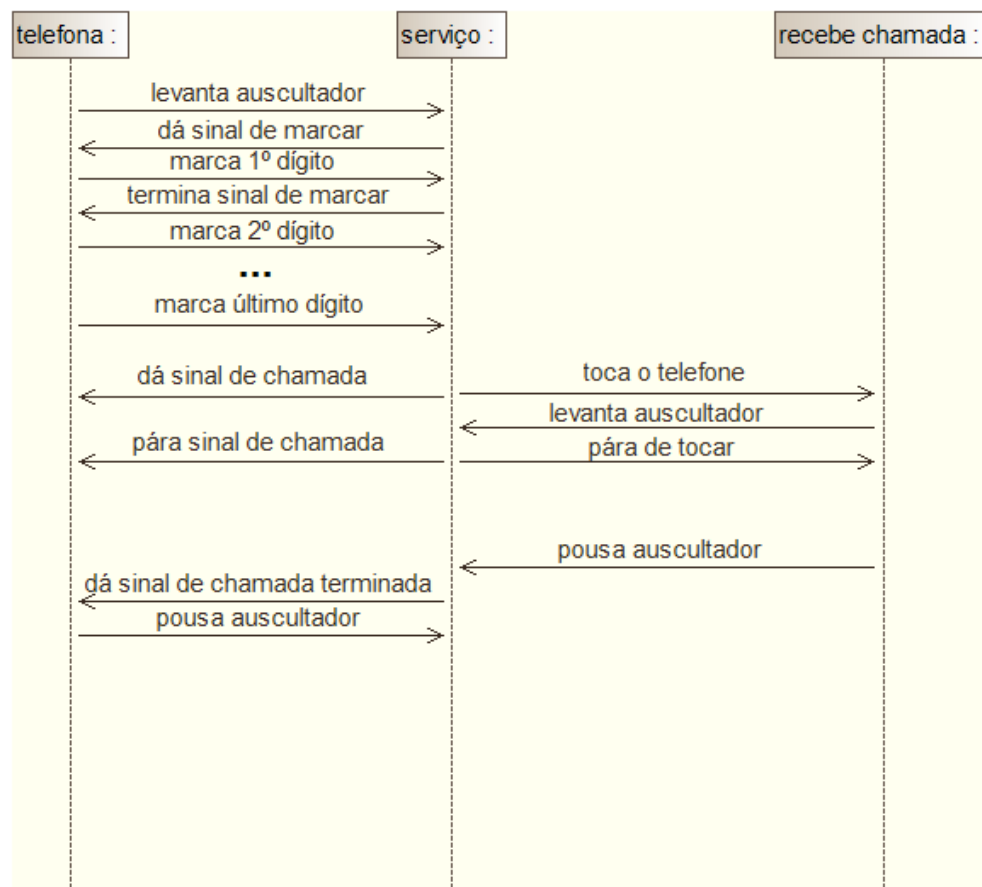
Não se devem representar muitas condições no diagrama de sequência pois corre-se o risco de o complicar.

Barra de ativação

- Uma barra de ativação mostra o período de tempo durante o qual um objeto está a executar uma ação, quer diretamente quer indiretamente através de um procedimento chamado
 - inclui situação em que está à espera de retorno de uma chamada síncrona
 - não inclui situação em que um processo está adormecido à espera de receber uma mensagem assíncrona que o acorde
- Em termos de processos, significa que o objeto tem um processo ou thread ativo associado
- A sua indicação é opcional
- Retorno de chamada é implícito no fim da barra de ativação
- Chamadas recursivas provocam barras empilhadas



Exemplo – chamada telefónica



Relação com outros diagramas

- **Colaboração**

- Diagramas de sequência e de colaboração coletivamente designados diagrama de interação
- Diagrama de colaboração = diagrama de objetos + diagrama de sequência

- **Casos de utilização**

- A um caso de utilização podem corresponder vários diagramas de sequência, para descrever sequências normais e sequências excecionais de funcionamento
- Inicialmente, o sistema pode ser representado por um único objeto; depois de conhecida a sua estrutura interna, podem-se representar objetos internos ao sistema

- **Estados**

- Enquanto que um diagrama de interação mostra um comportamento possível de um conjunto de objetos, com passagem de controlo entre objetos, um diagrama de estados mostra todos os comportamentos possíveis de um único objeto, com passagem de controlo entre estados
- Envio e receção de mensagens são ações e eventos nos diagramas de estados
- Intervalo de tempo entre duas mensagens é um estado no diagrama de estados

- **Atividades**

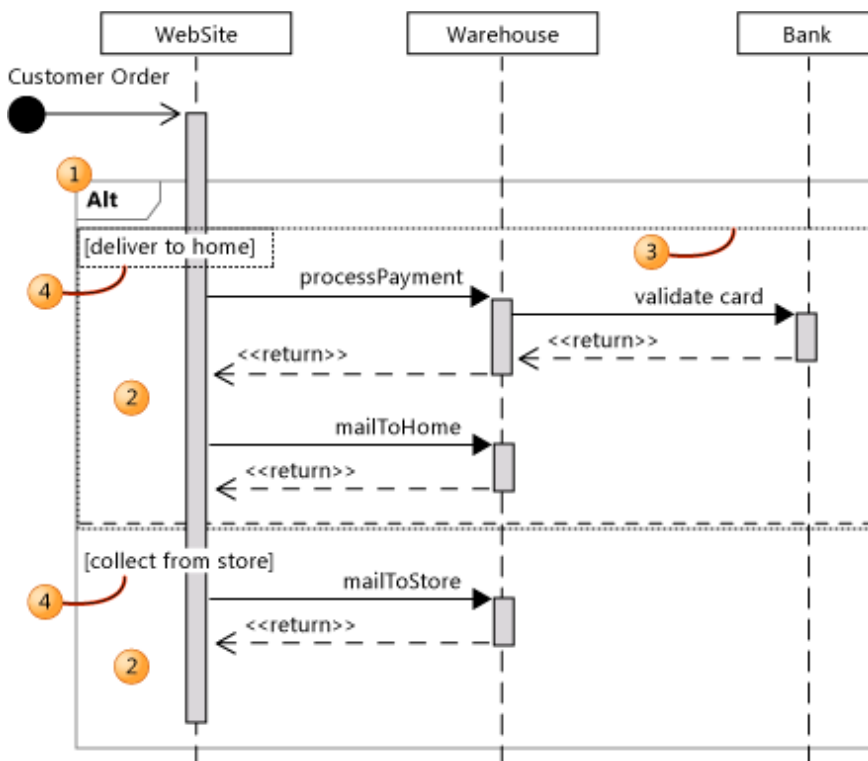
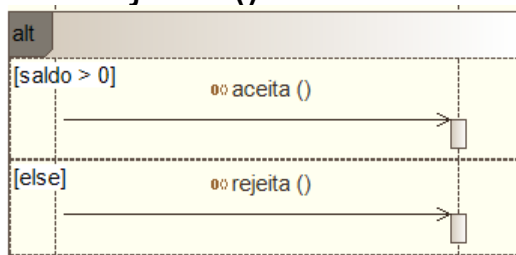
- Adequados para mostrar ações realizadas por um objeto que não envolvem comunicação com outros objetos
- Com "swimlanes", permitem mostrar sequências alternativas e ciclos envolvendo vários objetos

Fragmentos Combinados

- Consiste em um ou mais operadores de interação em que cada um deles contém uma ou mais mensagens, interações ou outros fragmentos combinados.
- Abordagem à reutilização e simplificação dos diagramas de sequência
 - junção de cenários alternativos semelhantes
 - Evitar repetir sequência de interação comuns

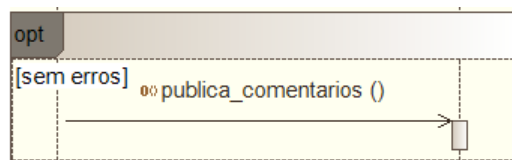
Fragmentos Combinados – Operador ALT

- “*alternatives*”
- Seleciona uma interação entre um conjunto de interações
- A interação selecionada estará associada à condição satisfeita por uma condição de guarda ou pelo else
 - opções: 1 ou mais condições de guarda e else
- **Exemplo:** Se o saldo for positivo chamar aceitar(). Caso contrário chamar rejeitar().



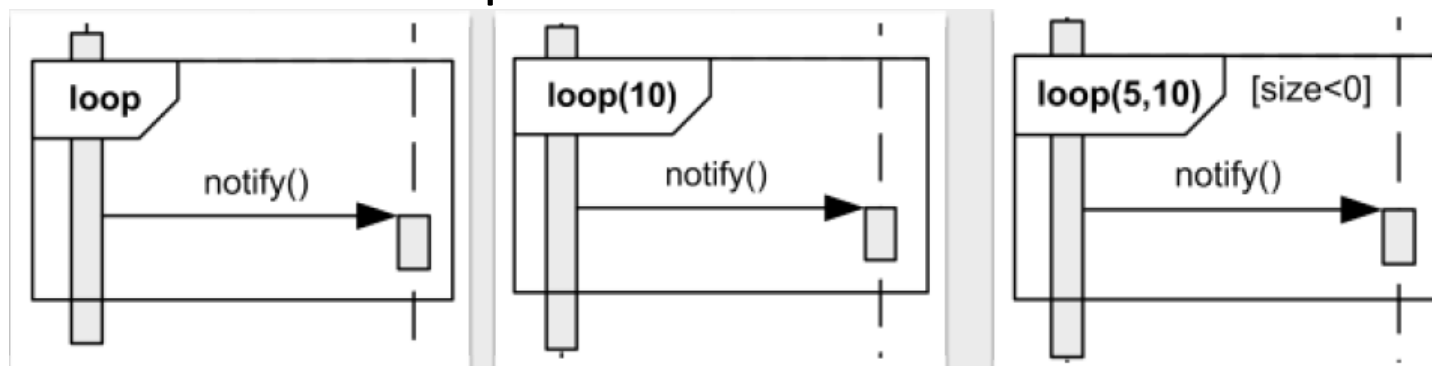
Fragmentos Combinados – Operador **OPT**

- “*option*”
- A interação neste fragmento ocorre se a guarda é satisfeita
 - opções: condição de guarda
- **Exemplo:** publica comentários se não existirem erros



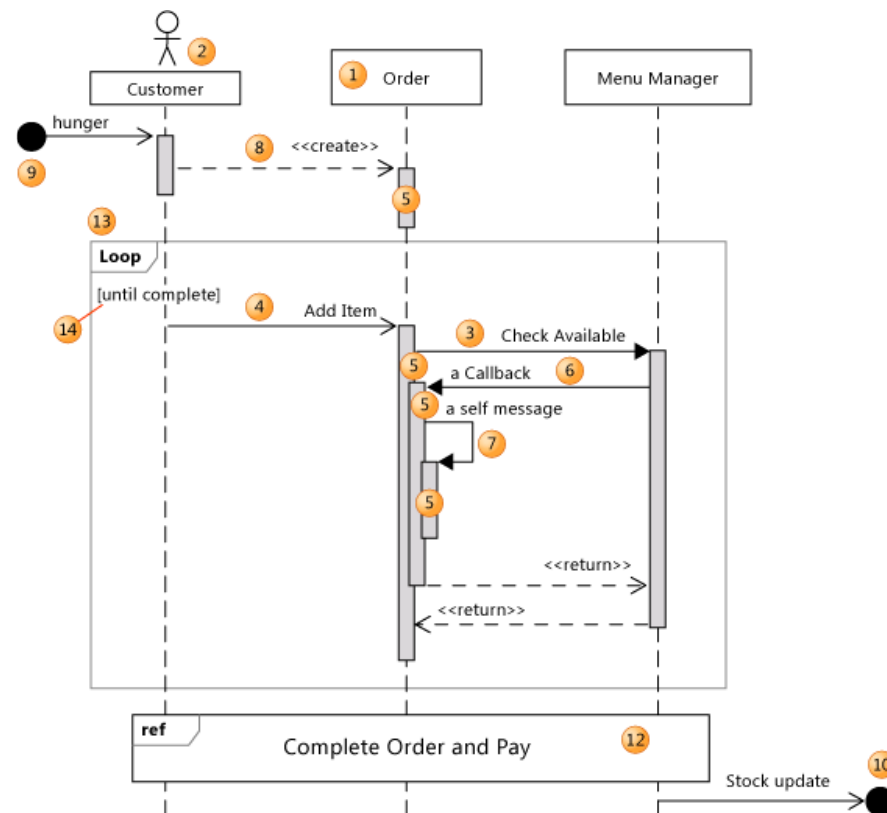
Fragmentos Combinados – Operador **LOOP**

- *Iteration*
- Executa a interação dentro do fragmento de minint até maxint vezes, enquanto a condição definida pela guarda é satisfeita
 - opções: minint, maxint, [guarda]
- **Exemplos:** notificar; notificar 10x; notificar entre 5x e 10x se o tamanho for maior que 0



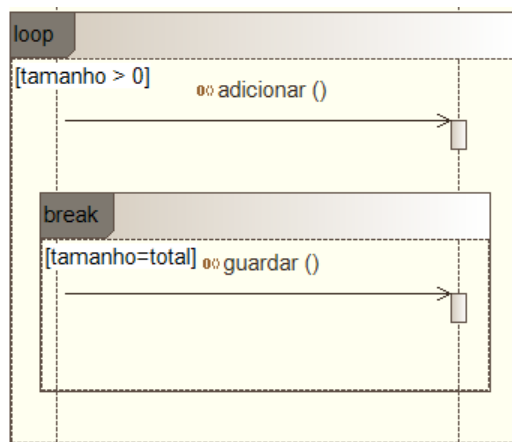
Fragmentos Combinados – Operador REF

- A interação neste fragmento foi definida noutra numa interação à parte (noutro diagrama de sequência), e é referenciada no ponto onde o fragmento é colocado.



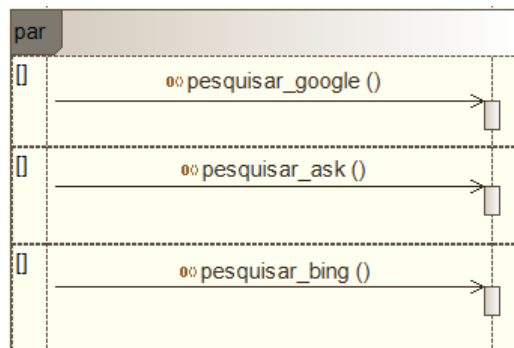
Fragmentos Combinados – Operador **BREAK**

- Se a interação no fragmento ocorrer, a interação exterior (tipicamente um ciclo) acaba
- **Exemplo:** Parar o ciclo e guardar a informação se o tamanho igualar o valor total



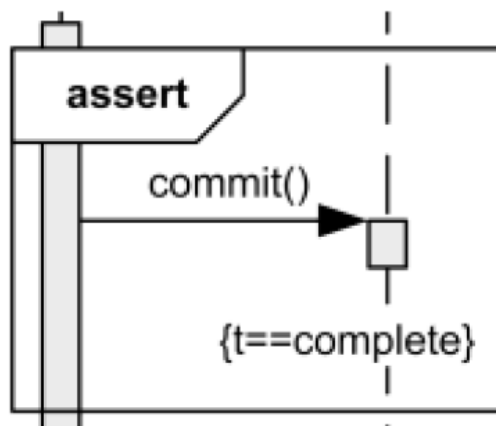
Fragmentos Combinados – Operador **PAR**

- “*Parallel*”
- A interação neste fragmento pode ter várias instâncias a ocorrer em paralelo
- Variantes de PAR para especificação da sequência das mensagens: **seq**; **strict**
- **Exemplo:** pesquisar no Google, ask e bing em paralelo



Fragmentos Combinados – Operador **ASSERT**

- A interação dentro do fragmento deve ocorrer exatamente da forma indicada ou será inválida
- **Exemplo:** o commit deve ocorrer. Após isso há uma avaliação do seu estado



Fragmentos Combinados – Operador **NEG**

- “*Negative*”
- Esta interação é inválida e não pode ocorrer
- **Exemplo:** A receção da mensagem de timeout indica que o login falhou

