



5408 Sistemas de informação – conceção

Sessão 6

26/03/2024

Margarida Fachada

Diagramas de Sequência

- Permitem representar o comportamento dinâmico do sistema, nomeadamente as interações entre os objetos.
- Consiste essencialmente na representação da sequência de chamada de métodos (troca de mensagens) entre os objetos que suportam a aplicação.
- Estes diagramas podem ser desenhados com vários níveis de detalhe e ao longo das diversas etapas do processo de desenvolvimento do sistema.

Diagramas de Sequência

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode ser usado para detalhar um Caso de Uso
- Identifica:
 - O evento gerador da funcionalidade modelada (ator responsável pelo evento)
 - Os objetos envolvidos na ação

Os benefícios de um diagrama de sequência

- Diagramas de sequência podem ser referências úteis para empresas e outras organizações.
- Pode-se desenhar um diagrama de sequência para:
 - Representar os detalhes de um caso de uso UML.
 - Modelar a lógica de um processo, função ou operação sofisticado.
 - Ver como objetos e componentes interagem uns com os outros para concluir um processo.
 - Planear e compreender a funcionalidade detalhada de um cenário existente ou futuro.

Diagramas de Sequência - O que representam?

- Mostram a sequência em que os eventos ocorrem num determinado processo
- Quais condições devem ser satisfeitas?
- Quais métodos devem ser disparados?
- Em que ordem os métodos são disparados ?
- Diagrama de Sequência não representa atributos

Diagramas de Sequência - Atores

- Exatamente os mesmos descritos no Diagrama de Casos de Uso
- Entidade externas que:
 - Interagem com o sistema
 - Solicitam serviços



Cliente

Diagramas de Sequência - Objetos

- Indicam instâncias de uma classe envolvidas no processo
 - As classes são mostradas em Diagramas de Classes
- Representados por retângulos
 - Nome do objeto (em minúsculo)
 - Nome da classe (inicial maiúscula)
 - Separação por dois pontos (:)

minha_conta: Conta



Diagramas de Sequência – Linha de Vida

- Linha vertical tracejada abaixo do objeto
 - Partem do retângulo que representa o objeto
- Representa o tempo em que um objeto existe durante o processo

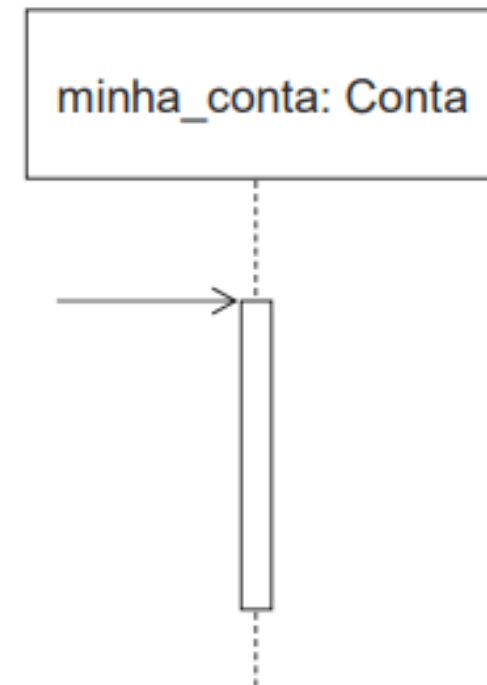


minha_conta: Conta



Diagramas de Sequência – Ativação do Objeto

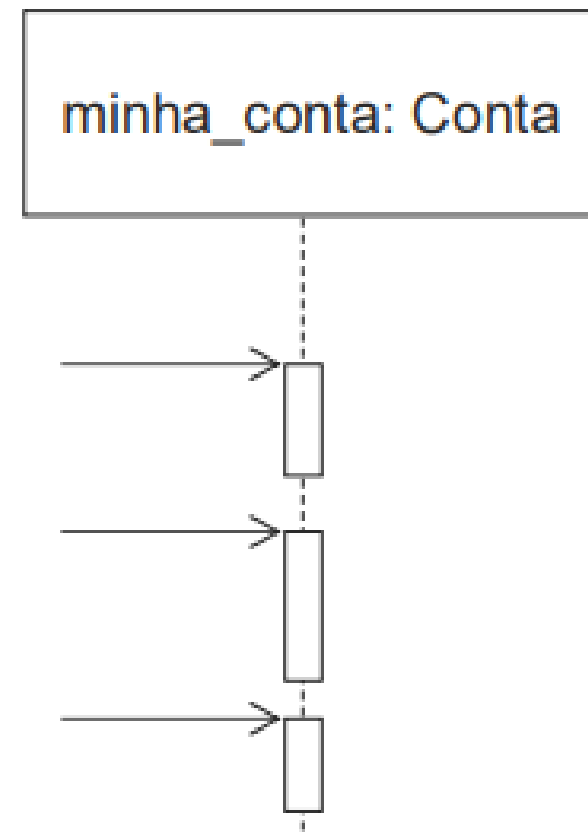
- O objeto é ativado quando este recebe um estímulo
 - Exemplo: recebimento de uma mensagem
- Um retângulo magro indica o período em que o objeto está a participar ativamente do processo





Diagramas de Sequência – Ativação do Objeto

- Podem haver vários períodos em que o objeto se encontra ativo.





Troca de Mensagens

- Representam a comunicação entre objetos e/ou atores
- Exemplos de mensagens
 - Chamadas de um método de um objeto por outro objeto
 - Comunicação entre dois atores

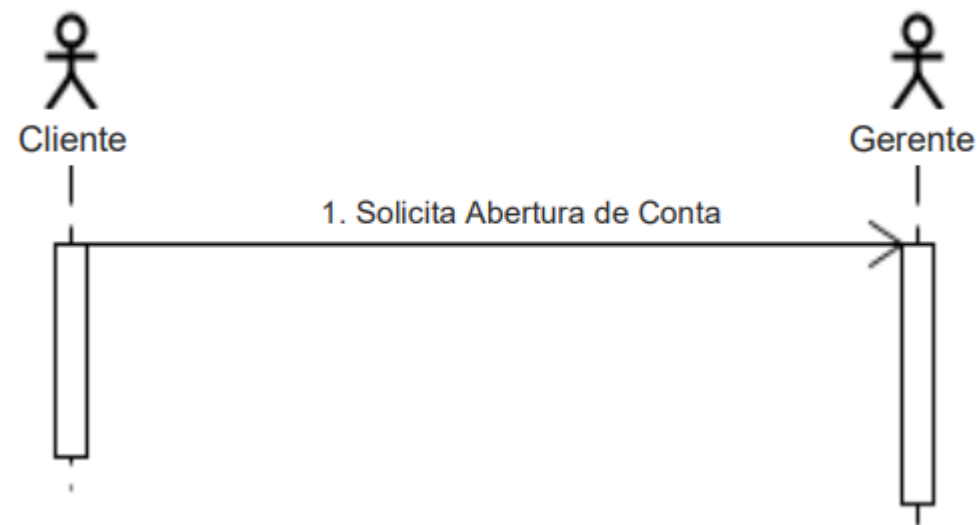


Tipos de Mensagens

- Ator para Ator
- Ator para Objeto
- Objeto para Objeto
- Objeto para Ator

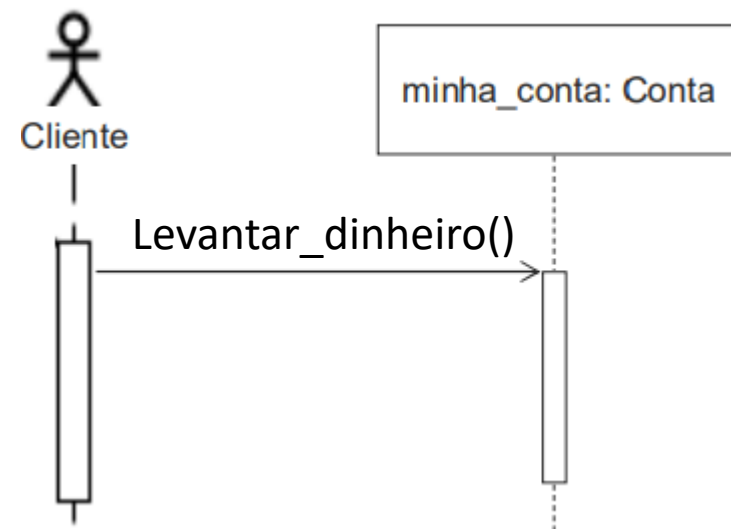
Mensagens entre atores

- Indica a conversa entre atores
- Os atores podem não fazer parte do sistema
 - Mas, facilita a compreensão do processo
- Não é muito comum de se modelar



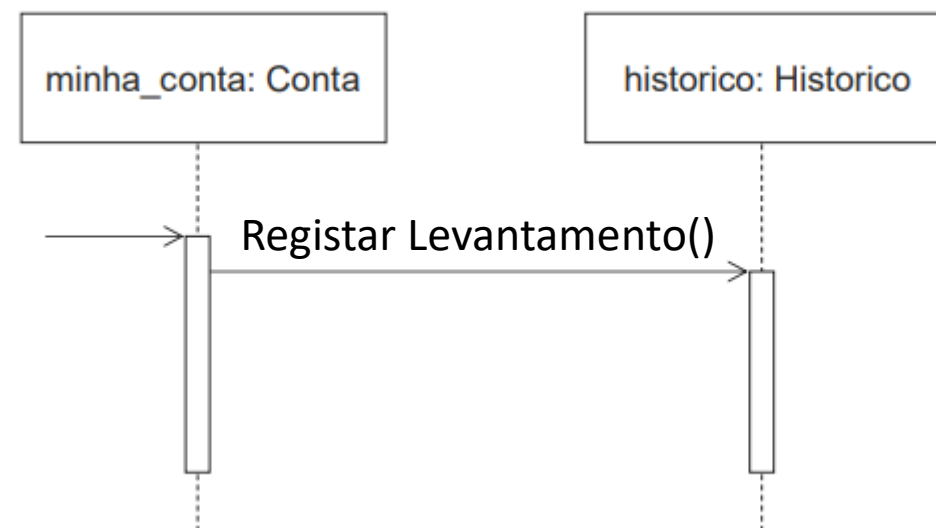
Mensagem Ator - Objeto

- Indica uma solicitação de serviço feita pelo ator ao sistema
- O ator produz um evento que força o disparo de um método
- Tipo comum quando se modela casos de uso



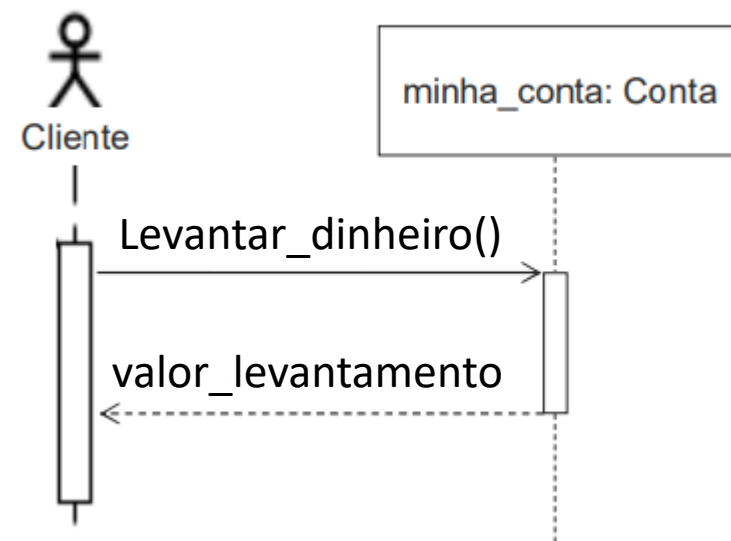
Mensagem entre objetos

- Indica que um objeto transmite uma mensagem para outro objeto
 - Exemplo: solicitando a execução de um método
- Tipo mais comum de troca de mensagens



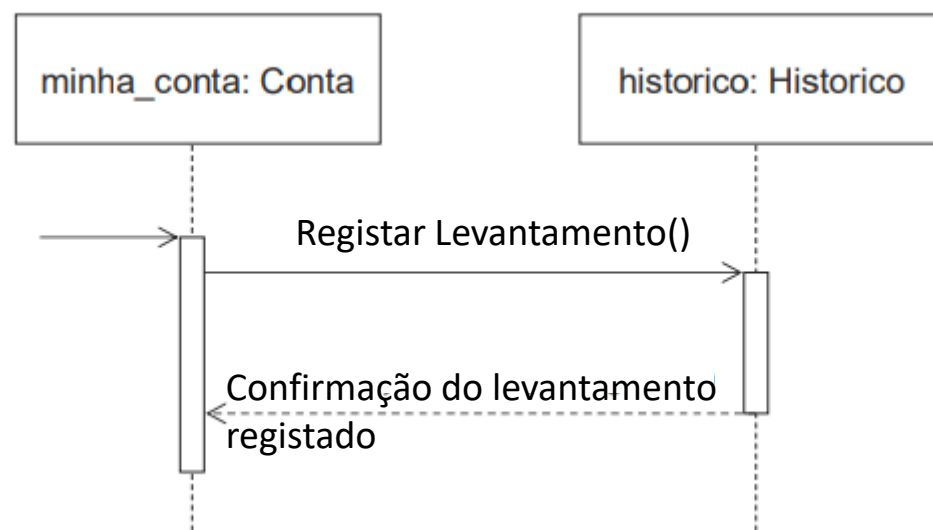
Mensagem Objeto-Ator

- Indica a resposta de uma solicitação de serviço feita pelo ator
 - O objeto envia uma mensagem de retorno
- Representadas por linhas tracejadas
 - Pode conter legenda indicando o retorno

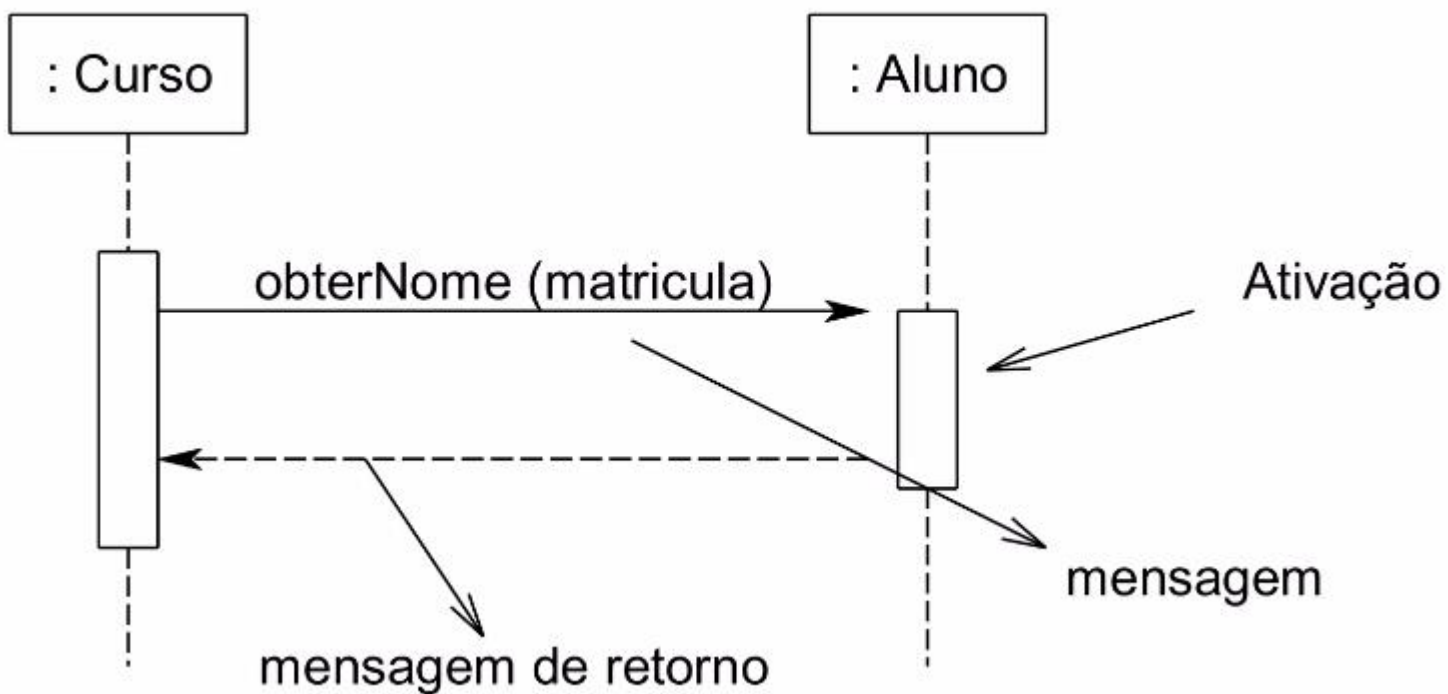


Mensagem de Retorno

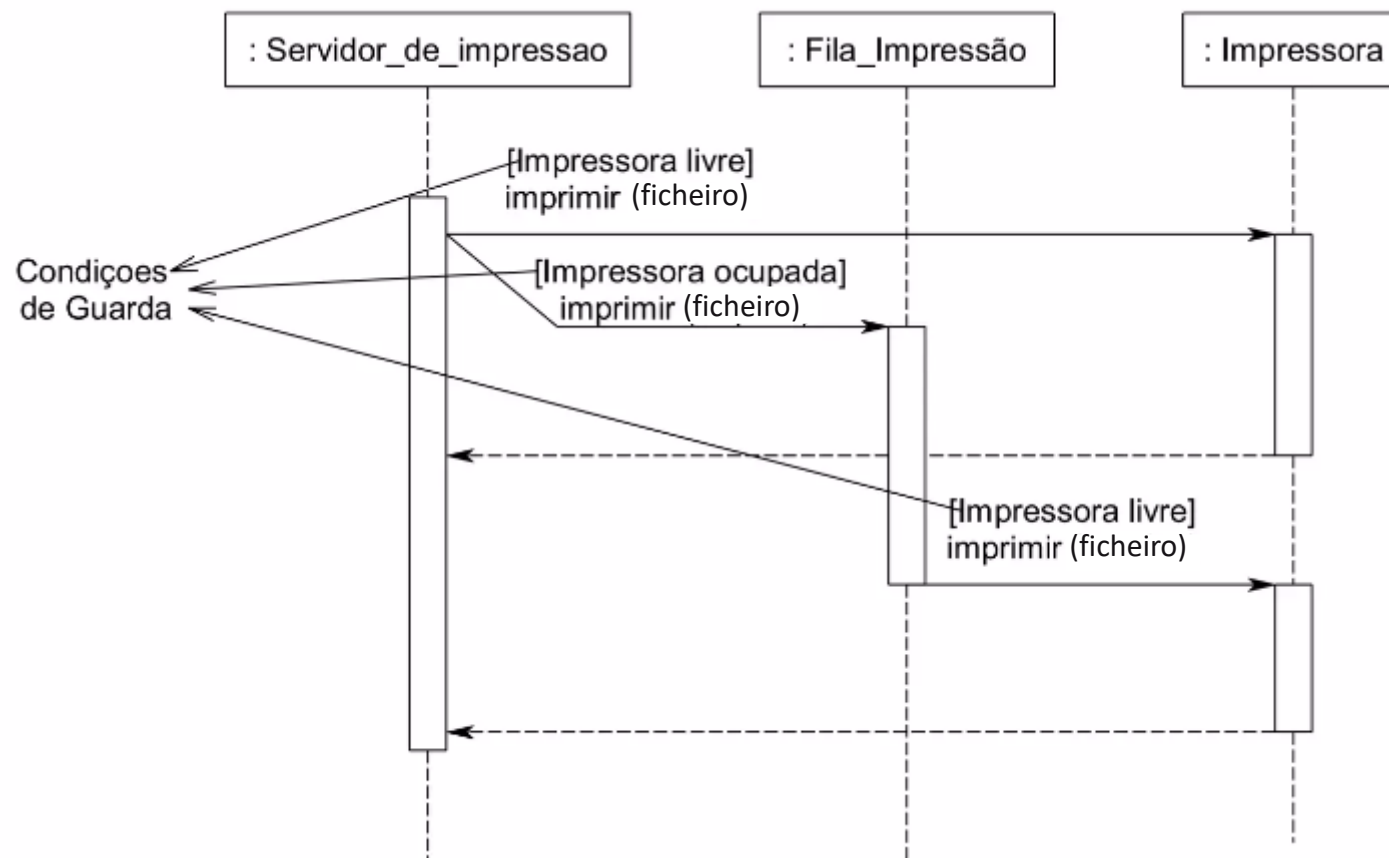
- Além de resposta ao ator, mensagens de retorno podem indicar respostas para objetos
 - Pode retornar informações específicas do método chamado



Representação - Ativação



Representação – Condições Guarda





Exercício 1 – Login no sistema

- O cliente solicita o login.
- O sistema recebe a solicitação e verifica as credenciais.
- O sistema retorna o resultado do login para o cliente.
- O cliente recebe o resultado e, se autenticado, prossegue com a interação no sistema.

Exercício 2 – Reserva de avião

- O cliente solicita uma reserva de bilhete ao sistema de reservas.
- O sistema de reservas confirma que recebeu a solicitação do cliente.
- O sistema de reservas envia uma solicitação de reserva para a companhia aérea.
- A companhia aérea confirma a reserva para o sistema de reservas.
- O sistema de reservas confirma a reserva para o cliente.

Exercício 3 – Pré-registo

- Descrição para o caso de uso PhonePizza:
 - Para que o cibernauta 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-requisito 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

Exercício 4 – Criar encomenda

- O diagrama que vamos ilustrar é um cenário em que um cliente cria uma encomenda adiciona itens e confirma encomenda.
- Neste caso um objeto cliente cria uma nova encomenda usando uma mensagem com o estereótipo <<create>>.
- O processo de adicionar produtos à encomenda é repetido por cada produto adicionar, por isso é utilizado um * como símbolo de **interação**.
- Este processo quer o envio de uma mensagem adicionaproduto() ao objeto Encomenda, que por sua vez irá adicionar o produto à encomenda após a consulta do respetivo preço através da mensagem devolvepreço().
- Por fim o cliente confirma a encomenda enviando uma mensagem confirmaencomenda() ao objeto Encomenda. Esta mensagem apenas será enviada se o número de itens adicionados a encomenda for superior zero