

Engenharia de Requisitos

Aula TP6: Diagrama de Sequência

Mary Barreto
Faculdade de Ciências Exatas e Engenharias

baseado nos slides do Prof. Filipe Quintal

2023/2024




1

DIAGRAMA DE SEQUÊNCIA

- Enquanto que o diagramas de transição de estados modela os estados de um objeto
- Os **diagramas de sequência** (ou diagrama de sequência de mensagens) **modelam a comunicação entre objetos**
- Como os **dados são transferidos**
- Podem ser usados para modelar
 - **Um conversa entre 2 pessoas**
 - **Protocolos de comunicação**
 - **Série de chamadas a métodos de diferentes objetos**
- Em UML usamos os diagramas de sequência para modelar cenários específicos de interação

2023/2024




2

DIAGRAMA DE SEQUÊNCIA

- Nos diagramas de sequência **não modelamos como os dados são tratados em cada** objetos
- Mas **sim** como a **sequência de comunicação** acontece com diferentes níveis de detalhe
- É provavelmente o diagrama mais fácil de compreender
 - Desde experts no âmbito até utilizadores finais

2023/2024



3

DIAGRAMA DE SEQUÊNCIA: dicas para realizar

- Defina o **contexto para a interação**
 - Por exemplo, um caso de uso de utilização/ use case.
- Exprese o fluxo da **esquerda para a direita** e de **cima para baixo**.
- Coloque as instâncias **ativas à esquerda/superior** e as **passivas à direita/inferior**.
- Desenhe diagramas de sequência **para cada caso de utilização** se quiser observar o comportamento de vários objetos

2023/2024

4

DIAGRAMA DE SEQUÊNCIA - *LIFELINE*



2023/2024

5

DIAGRAMA DE SEQUÊNCIA - *LIFELINE*

- Num diagrama de sequência os objetos envolvidos na interação são representados nas *lifelines* ou *linhas de vida*
- Representada como uma linha vertical (normalmente tracejada)
- Esta *direção vertical representa a ordem cronológica das mensagens*
- No topo da *lifeline* a classe ou papel são identificados num retângulo separados por :
- Quem podem não estar presentes
 - Se a classe não for especificada então os : podem ser removidos
 - Terão que estar presentes se a classe for especificada

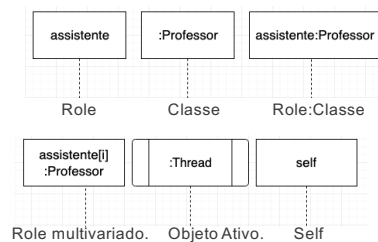


2023/2024

6

DIAGRAMA DE SEQUÊNCIA - *LIFELINE*

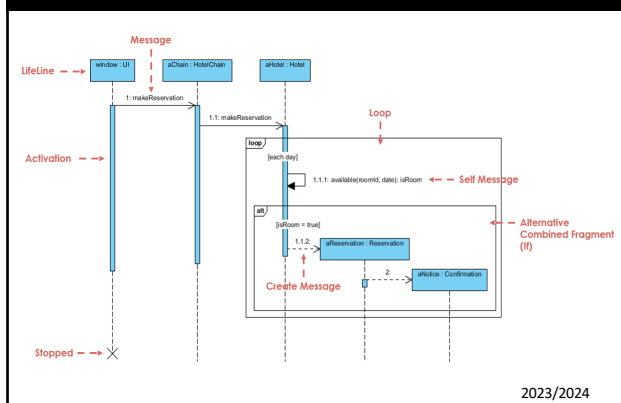
- O diagrama de sequência poderá ser **modelado do ponto de vista da instância ou da classe**
- Maior flexibilidade utilizando instâncias e roles



2023/2024

7

Hotel System: Reservation window (message sequence)



2023/2024

8

DIAGRAMA DE SEQUÊNCIA - TROCA DE MENSAGENS



2023/2024

9

DIAGRAMA DE SEQUÊNCIA - TROCA DE MENSAGENS

- O diagrama de sequência é **bi-dimensional**
- Os envolvidos são **representados no eixo horizontal, enquanto que o eixo vertical modela uma ordem cronológica**
- As interações são consideradas uma sequência de especificações de **eventos**
- Estes **eventos** englobam o **envio** e **recepção** de **mensagens** ou a ocorrência de eventos baseados no tempo.
 - O eixo vertical no diagrama determina a sequência de eventos numa *lifeline*
- É assumido que a troca de mensagens não tem duração

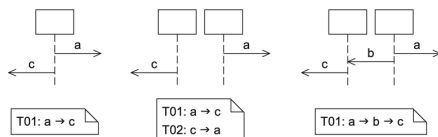


2023/2024

10

DIAGRAMA DE SEQUÊNCIA - TROCA DE MENSAGENS

- A ordem cronológica entre um evento a e b é representada pelo símbolo \rightarrow .
- $a \rightarrow b$ significa que a mensagem a precede a mensagem b



- Caso esta ordem não seja especificada então a ordem de apresentação é respeitada

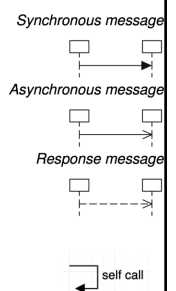


2023/2024

11

DIAGRAMA DE SEQUÊNCIA - MENSAGENS

- Num diagrama de sequência uma mensagem é representada por uma seta entre o emissor e recetor
- Uma **mensagem síncrona** é representada com uma seta com uma linha contínua e com a extremidade sólida
 - Neste caso o **emissor espera pela resposta**
- Uma mensagem **assíncrona** é representada da mesma forma mas com a extremidade vazia
- Uma **mensagem de resposta** é representada por uma seta aberta e tracejada
 - Caso a resposta e ocasião da resposta seja óbvia então a **mensagem de resposta pode ser omitida**
- Podem também existir mensagens de um objeto a si próprio (*self-messages*)
 - Síncronas dentro da *lifeline*
 - Análogo à chamada de um método dentro da classe



2023/2024

12

DIAGRAMA DE SEQUÊNCIA - MENSAGENS



- **Considerando o exemplo acima**
 - Ambos os modelos representam a comunicação com o professor para o registo numa disciplina
 - À esquerda o registo é feito por email, o **aluno não espera por confirmação**
 - À direita o registo é feito pessoalmente e o **aluno espera pela resposta**

DIAGRAMA DE SEQUÊNCIA - MENSAGENS

- As mensagens são **identificadas pelo nome**, com parâmetros e tipos de retorno opcional
 - O tipo de retorno pode ser associado, também a uma variável.

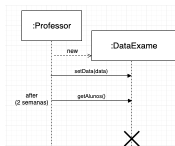
- Existe um tipo de mensagem particulares para criar objetos

- Identificada com uma seta aberta tracejada com a keyword new

- Pode também ser útil **remover objetos através de um *destruction event***



DIAGRAMA DE SEQUÊNCIA - MENSAGENS



- No diagrama acima a classe Professor cria uma DataExame mensagem new
- Depois de duas semanas quando o objeto DataExame é removido
- Se o emissor de uma mensagem **é desconhecido ou irrelevante** pode ser modelado como uma *found message*
- Especificado com um círculo preto

- Por outro lado uma *lost message* é uma mensagem em que o recetor é irrelevante



DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS

- **Fragmentos** ou **operadores** são usados para modelar **estruturas de controle**
- Representados por um **retângulo** com o **tipo de operador num pentágono**
- UML modela 12 tipos de operadores/fragmentos organizados em **3 tipos**
 - Ciclos e ramos
 - Concorrência ou ordem
 - Filtragem e afirmação
- Os fragmentos podem ser encapsulados



2023/2024

17

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS

	Operator	Purpose
Branches and loops	alt	Alternative interaction
	opt	Optional interaction
	loop	Iterative interaction
	break	Exception interaction
Concurrency and order	seq	Weak order
	strict	Strict order
	par	Concurrent interaction
	critical	Atomic interaction
Filters and assertions	ignore	Irrelevant interaction parts
	consider	Relevant interaction parts
	assert	Asserted interaction
	neg	Invalid interaction



2023/2024

18

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- Podemos usar o **fragmento alt** para representar **sequências alternativas**
- Possuem **2 operandos**
- Cada operando representa **uma forma alternativa de modelar o comportamento**
 - **Análogo ao switch** numa linguagem de programação
 - **Guardas** são utilizados **para escolher** o fragmento a executar
- **Pode ser definida uma guarda else**, avaliada como **true** caso nenhuma das outras seja verdadeira



2023/2024

19

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- Podemos usar o **fragmento opt** para representar **uma sequência alternativa**
- Após a avaliação de uma **guarda** o conteúdo **as mensagens dentro do fragmento são processadas**
- Se a guarda é positiva então as mensagens são ignoradas
 - O processamento continua "fora" do opt
- **Análogo a um if sem else** numa linguagem de programação



2023/2024

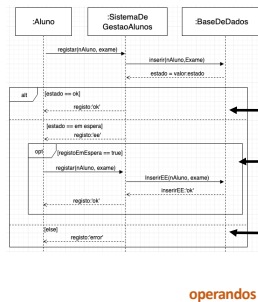
20

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- Quando um aluno regista-se num exame envia uma mensagem com o nº e qual o exame

- Mensagem síncrona** (irá esperar por resposta)

- O Sistema de gestão direciona o processamento para a base de dados
- A Base de dados retorna o estado da operação na variável estado
- Depois o processamento entra no primeiro fragmento



2023/2024

21

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- O fragmento alt depende do resultado recebido da base de dados

- Se existirem lugares disponíveis o diagrama chega ao seu final
- Se só existirem lugares em fila de espera.

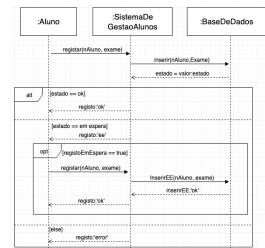
- O aluno terá que decidir se fica em fila de espera

- Modelado no opt**

- Se não existirem lugares o aluno recebe um erro

- Mensagem de resposta

- Guarda else**



2023/2024

22

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- O **fragmento loop** representa uma sequencia de mensagens/eventos que deverá ser repetida

- A keyword loop poderá ser **seguida por um número de iterações no formato (min,max)**.

- Se min e max forem idênticos poderão ser omitidos, se não existir valor superior podemos usar um *

- Pode ser uma **guarda** especificada para quebrar o loop

- O **fragmento break** é parecido a um fragmento opt,

- Possui uma guarda

- Quando verdadeira** as as interações dentro do break são executadas

- As mensagens seguintes ao break são ignoradas**

2023/2024

23

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONTROLE

- Neste exemplo é modelado **um sistema que necessita de login**

- O utilizador tem especificar password e nome de utilizador

- Tem 3 tentativas

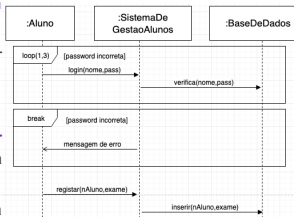
- Mas o loop **poderá ser finalizado antes** se a password inserida for correta

- Após o fragmento loop, se a guarda [password incorreta] continuar verdadeira

- O **fragmento break é executado**

- Retorna a mensagem de erro

- E termina a execução



2023/2024

24

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- Como já foi discutido o **eixo vertical representa uma ordem cronológica**
- Existem **outros métodos** para **explicitamente controlar ordem** das mensagens num diagrama de sequência.
- O **fragmento seq** representa a ordem por defeito para a troca de mensagens
- Tem **pelo menos um operando**, **representa uma sequência fraca especificada pelo padrão UML como:**
 - A ordem de eventos dentro de cada operando é mantida no resultado
 - Os eventos em diferentes *lifelines* de diferentes operandos podem ocorrer em qualquer ordem
 - Eventos na mesma *lifeline* de diferentes operandos são ordenados de forma a que o primeiro operando aconteça antes do segundo



2023/2024

25

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- O fragmento seq poder ser **usado para agrupar mensagens num fragmento break**
- Se a condição do break é verdadeira as **mensagens no seq que ainda não foram executadas são ignoradas**
- A execução continua fora do seq
- Este fragmento é útil nestas situações

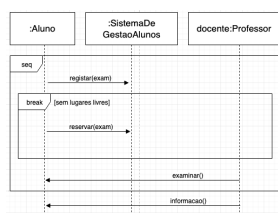


2023/2024

26

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- Neste exemplo um aluno comunica com o sistema de gestão para o registo num exame
- De seguida o aluno reserva o exame
- Durante este processo **se a guarda [sem lugares livres] é verdadeira**
 - O fragmento break é executado
- **A execução continua fora do seq** com a mensagem informação



Esta mensagem
seria ignorada

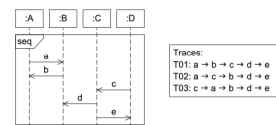
- **Este modelo implementado sem o seq resultaria na finalização da comunicação entre os objetos especificados**



2023/2024

27

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA



- Neste diagrama a ordem de c não está ligada a 'a' e 'b'
- E desta forma pode ser independente de ambas
- 'b' e 'd' são enviada/recebidas por B, logo **aqui existe uma ordem cronológica entre estas mensagens**
- **'e' é sempre a ultima mensagem**

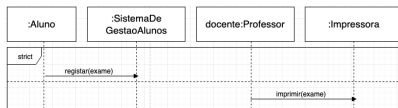


2023/2024

28

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- O **fragmento strict** representa uma ordem de interação rígida
- As mensagens num operando acima na representação visual serão sempre trocadas antes das inferiores



- Neste exemplo a mensagem **registrar** entre a classe Aluno e SistemaDeGestaoAluno **será sempre trocada antes da mensagem imprimir** entre o docente e a classe Impressora



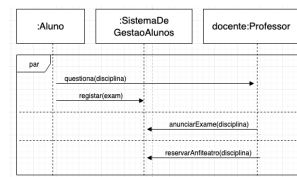
2023/2024

29

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- O **fragmento par** permite ignorar qualquer ordem cronológica, entre diferentes operandos
- Estes podem ser intercalados
 - No entanto as restrições dentro da cada operando deverão ser respeitadas
- Logo, a ordem dos operandos é irrelevante

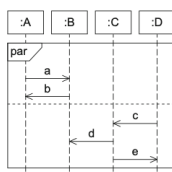
- Neste exemplo **um aluno nunca registra para um exame antes de questionar a matéria**
- Mas estas mensagens **podem acontecer antes ou depois** do docente anunciar exame ou reservar sala



2023/2024

30

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA



Traces:

T01: a → b → c → d → e
 T02: a → c → b → d → e
 T03: a → c → d → b → e
 T04: a → c → d → e → b
 T05: c → a → b → d → e
 T06: c → a → d → b → e
 T07: c → a → d → e → b
 T08: c → d → a → b → e
 T09: c → d → a → e → b
 T10: c → d → e → a → b

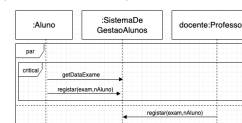


2023/2024

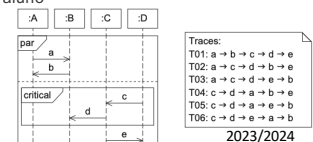
31

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - CONCORRÊNCIA

- O **fragmento critical** garante que certas partes da interação **não são interrompidas por eventos imprevistos**.



- A modelação acima **garante o tempo entre a mensagem getExamData e registrar não podem acontecer outra mensagem** que por exemplo registre outro aluno



Traces:

T01: a → b → c → d → e
 T02: a → c → b → d → e
 T03: a → c → d → b → e
 T04: c → a → b → d → e
 T05: c → d → a → b → e
 T06: c → d → e → a → b



2023/2024

32

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - FILTRAGEM

- Geralmente o diagrama de sequência não representa todos os aspetos de interação.
- Vamos agora compreender como apresentar :
 1. Mensagens que podem acontecer mas não são relevantes
 2. Que mensagens podem ocorrer
 3. Que mensagem não podem ocorrer



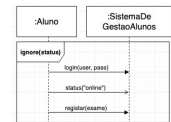
2023/2024

33

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - FILTRAGEM

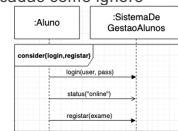
- Mensagens irrelevantes são indicadas com o fragmento {ignore}

- Neste exemplo as mensagens status
- Ocorrem mas não são importantes
 - Para o cenário modelado



- Por outro lado o fragmento {consider} realça mensagem de particular importância para o cenário modelado

- As mensagens não definidas são classificadas como ignore
- O diagrama à direita é equivalente ao apresentado acima

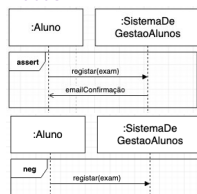


2023/2024

34

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - FILTRAGEM

- O fragmento **assert** identifica certas partes do modelo como obrigatórias
- O modelo representa o que terá que ser implementado sem desvios
- Por exemplo quando um aluno inscreve-se no exame **terá obrigatoriamente** que receber um email
- O fragmento **neg** modela interações não permitidas
- Situações que não podem ocorrer
- Úteis para modelar erros comuns
 - Evitar que sejam implementados
 - Encontrar soluções!



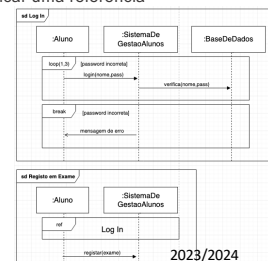
2023/2024

35

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - REFERÊNCIAS

- Pode ser útil integrar um diagrama num outro
 - Evitar repetir modelação
 - Simplificar modelações mais complexas
- É usado o fragmento **ref** para especificar uma referência
- Poderão ser definidos parâmetros
 - Entre parêntesis
 - E tipo de retorno

- No exemplo à direita, a sequência de mensagens de login são incluídas no diagrama do registo de exame

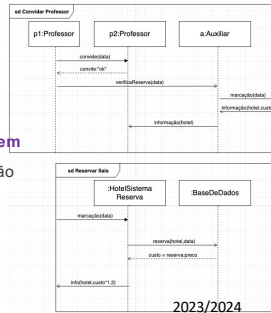


2023/2024

36

DIAGRAMA DE SEQUÊNCIA - FRAGMENTOS - GATES

- Por definição **as mensagens não devem extrapolar a fronteira do modelo**
- Para modelar comunicações **para fora e de fora** do modelo usamos os gates
- Permitem **receber/enviar mensagens "para fora" da interação** modelada
- Os gates são representados como setas de e para fora do modelo
- Podem ter um nome na mensagem
- Nos exemplo à direita existe comunicação entre os 2 modelos através de gates



37

Realização Ficha 6

38