INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

ANÁLISE DE SISTEMAS
ASIS

Prof. Wallace Rodrigues

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Diagrama de Sequência

Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

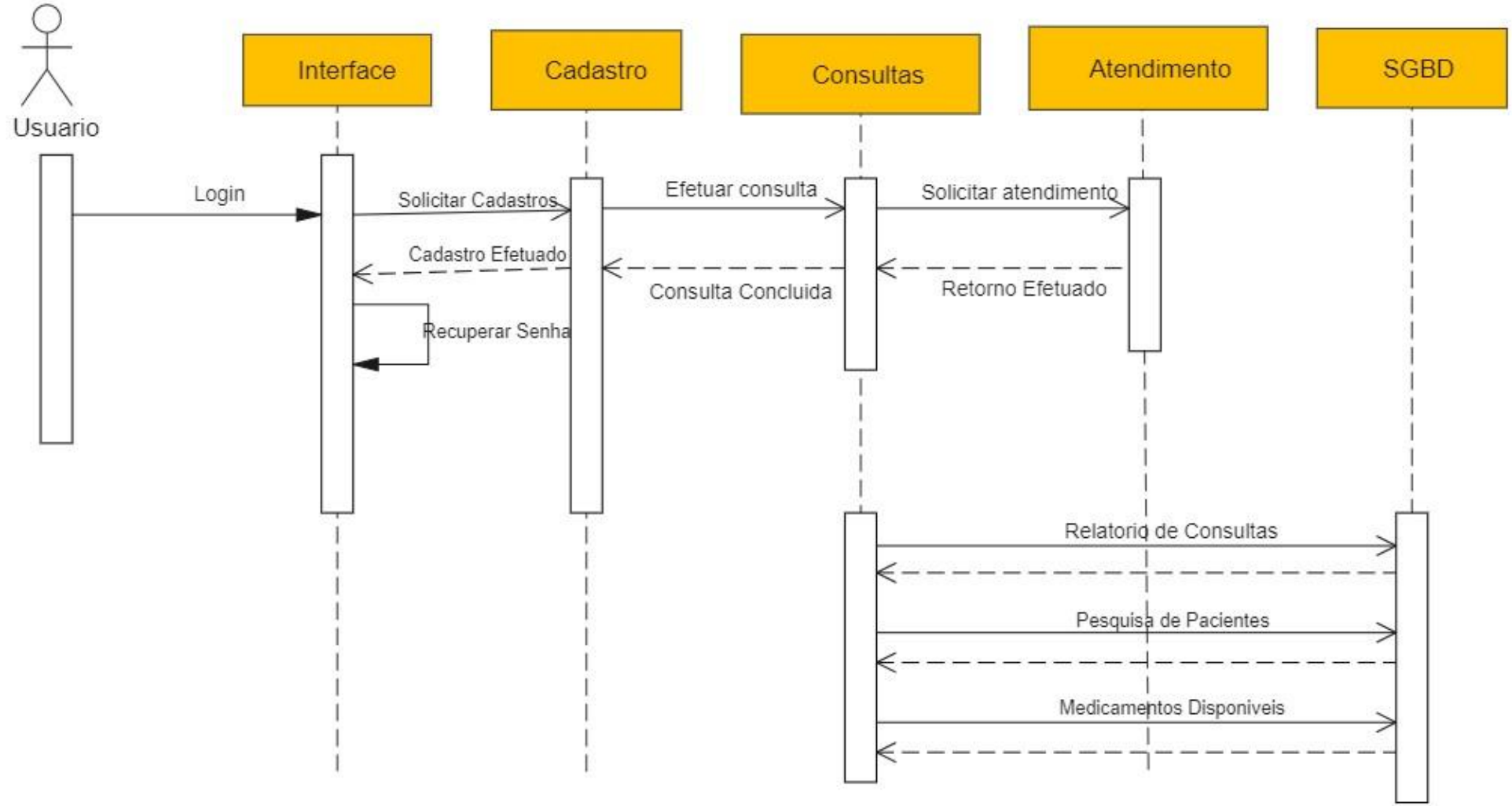


Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- **Definição**: O Diagrama de Sequência é um tipo de diagrama de interação da UML (Linguagem de Modelagem Unificada). Ele mostra a ordem cronológica na qual as mensagens (ou chamadas de método) são enviadas entre os objetos de um sistema para completar uma tarefa específica.

Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- **A "Fotografia" da Interação**: Enquanto o Diagrama de Classes mostra a estrutura estática do sistema (quem existe), o Diagrama de Sequência mostra a dinâmica e o fluxo de controle (como eles interagem no tempo). Ele responde à pergunta: "Como um grupo de objetos trabalha em conjunto para realizar uma função?".

Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- **O que é**: Um Diagrama de Sequência é uma ferramenta da UML (Linguagem de Modelagem Unificada) usada para mostrar como os objetos de um sistema interagem entre si ao longo do tempo.
- **O foco**: Ele se concentra na ordem cronológica das mensagens que são trocadas para que uma tarefa ou um processo seja concluído.
- **A diferença principal**: Enquanto um Diagrama de Classe mostra a estrutura estática do sistema (quais classes existem e como se relacionam), o Diagrama de Sequência mostra o comportamento dinâmico (como elas se comunicam para fazer algo acontecer).

Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Anatomia Básica

- Para visualizar melhor, imagine o diagrama de sequência como uma linha do tempo vertical, onde as interações acontecem de cima para baixo.
- O topo do diagrama lista os objetos e atores envolvidos. As linhas tracejadas que descem deles representam a sua "vida" durante a sequência.
- As setas horizontais entre as linhas de vida são as mensagens. Cada seta representa uma chamada de método ou uma comunicação entre os objetos.
- A ordem em que essas setas aparecem, de cima para baixo, indica a sequência temporal exata dos eventos.

Diagrama de Sequência

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Uma Metáfora Simples

- Pense em uma coreografia de dança. O Diagrama de Classe seria a lista de dançarinos e suas posições no palco (a estrutura).

- O Diagrama de Sequência seria a gravação completa da dança, mostrando quem interage com quem, em qual momento, e em que ordem. Ele detalha os passos, as trocas e o fluxo que formam a performance inteira.

Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

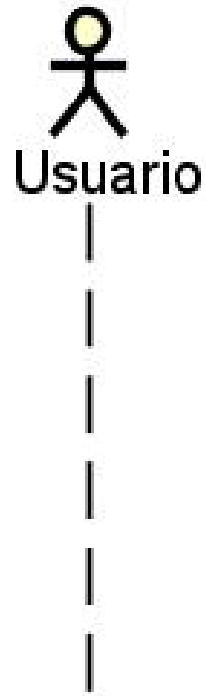
- Definimos o Diagrama de Sequência como a coreografia das interações do nosso sistema. Agora, vamos mergulhar nos elementos fundamentais que compõem essa coreografia. Pensem neles como os 'atores', os 'palcos' e as 'falas' que, juntos, narram a história de como os objetos de software trabalham em harmonia."

Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

O Ator (Actor)

- **O que é**: Representa uma entidade externa que inicia a sequência de eventos.
- **Exemplo**: Um usuário clicando em um botão, outro sistema enviando uma solicitação, ou um sensor acionando um evento.
- **Desenho**: É tradicionalmente representado por um boneco (stick figure).



Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

A Linha de Vida (Lifeline)

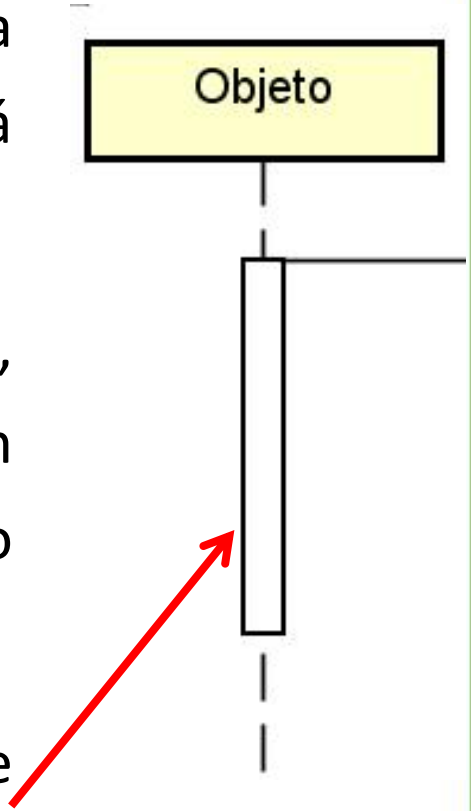
- **O que é**: Uma linha vertical tracejada que representa um objeto ou classe ao longo do tempo.
- **Exemplo**: Cada objeto do seu sistema, como TelaLogin, SistemaAutenticacao ou BancoDeDados, terá uma linha de vida.
- **Desenho**: Uma linha vertical tracejada com o nome do objeto (e, opcionalmente, sua classe) no topo. Ex: :LoginScreen.

Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

A Caixa de Ativação (Activation Bar)

- **O que é**: Uma barra retangular vertical que fica sobre a linha de vida. Ela indica o período de tempo em que um objeto está "ativo", ou seja, executando um método.
- **Exemplo**: Quando a TelaLogin recebe a mensagem de login, sua caixa de ativação começa. Quando ela envia a mensagem para o SistemaAutenticacao, a caixa de ativação deste último começa.
- **Desenho**: Uma barra retangular que "pula" de uma linha de vida para outra, conforme o controle do fluxo passa entre os objetos.



Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

As Mensagens (Messages)

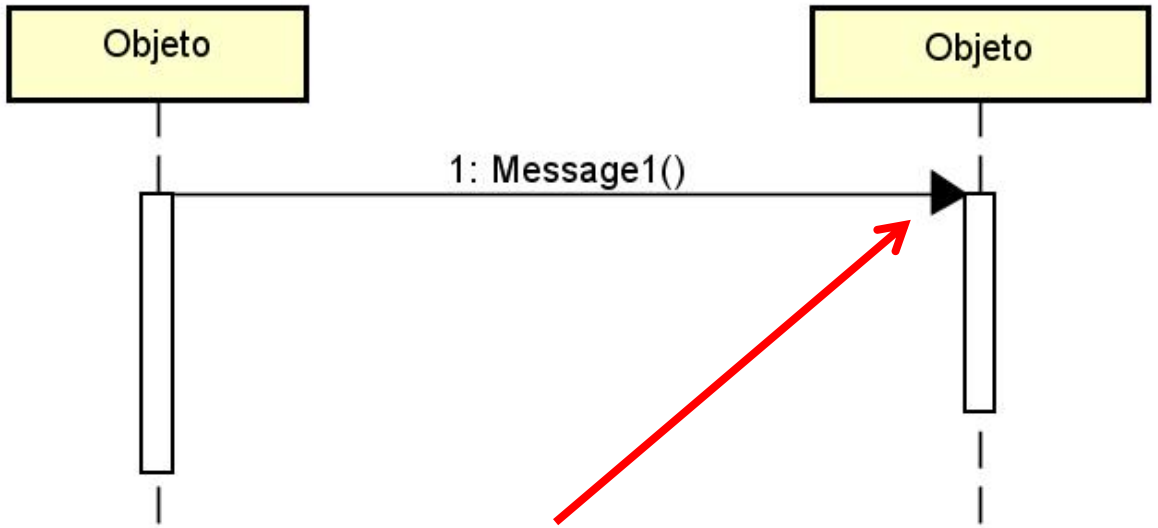
- **O que são**: As setas horizontais que conectam as linhas de vida. Elas representam a comunicação entre os objetos.

- **Tipos de Mensagens**:
  - Mensagem Síncrona (Synchronous)
  - Mensagem Assíncrona (Asynchronous)
  - Mensagem de Resposta ou Return (Reply Message)
  - Mensagem Self (Self Message)

Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- **Mensagem Síncrona (Synchronous)**: A mais comum. O remetente envia a mensagem e espera por uma resposta antes de continuar.
  - **Desenho**: Seta sólida com a ponta preenchida.
  - **Exemplo**: A TelaLogin envia autenticar() para o SistemaAutenticacao e fica "parada" esperando o resultado.

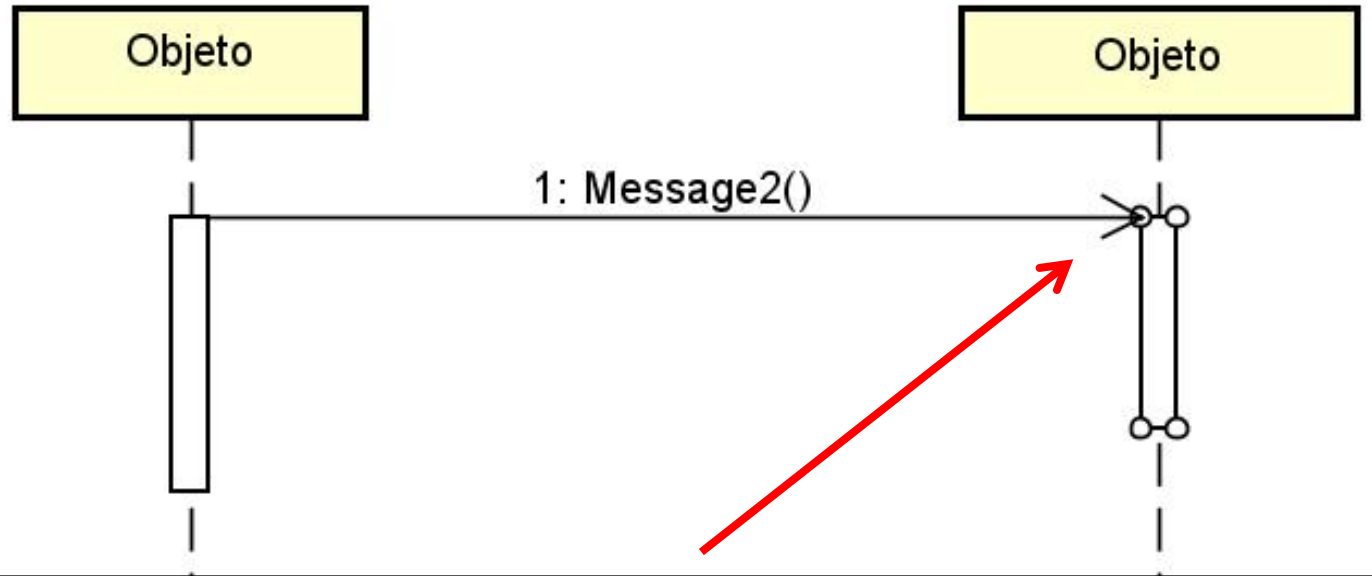


Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

➢ **Mensagem Assíncrona (Asynchronous)**: O remetente envia a mensagem e não espera pela resposta. Ele continua sua execução imediatamente.

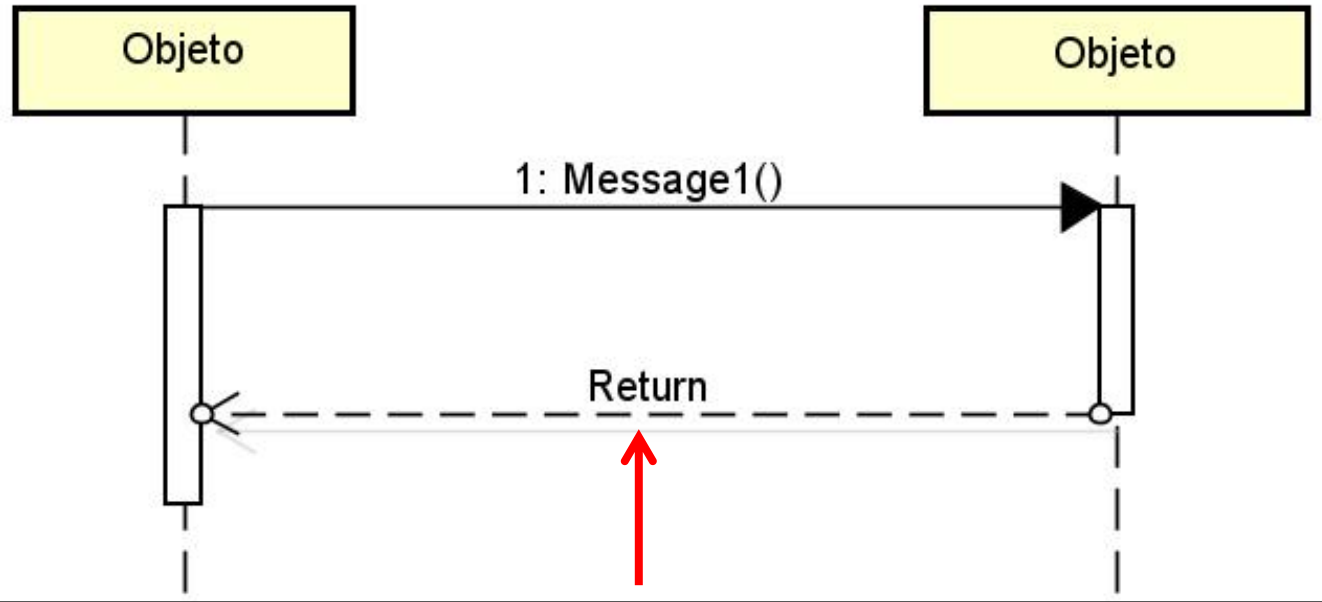
- **Desenho**: Seta sólida com a ponta aberta.



Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

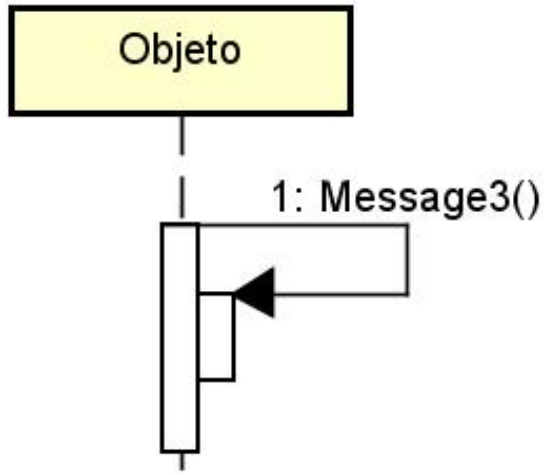
- **Mensagem de Resposta ou Return (Reply Message)**: Uma seta tracejada que retorna a um objeto, indicando o resultado de uma mensagem síncrona.
  - **Desenho**: Seta tracejada com a ponta aberta.



Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- **Mensagem Self (Self Message)**: É uma mensagem que um objeto envia para si mesmo. Ela representa uma chamada de método interno.
  - **Exemplo**: Imagine um objeto SistemaDeAutenticacao que, ao receber o método autenticar(username, password), chama um método interno validarCredenciais() para verificar a senha antes de interagir com o banco de dados.
  - **Desenho**: A seta de mensagem sai da caixa de ativação do SistemaDeAutenticacao e retorna para a mesma caixa de ativação, formando um "loop" ou "U" invertido.

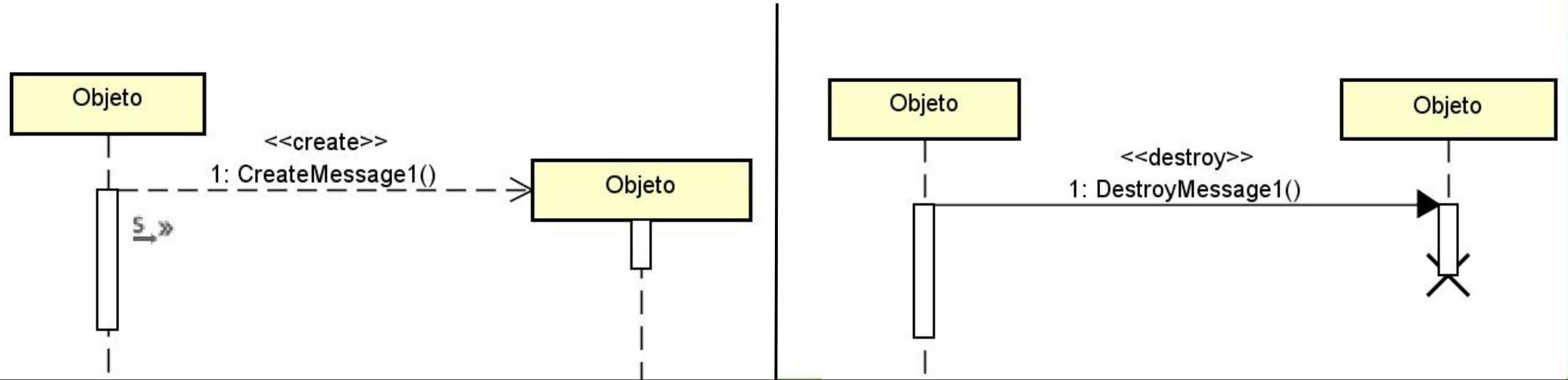


Componentes Principais

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Mensagens Especiais

- **Criação de Objeto (Create)**: Uma mensagem assíncrona que aponta para um novo objeto, indicando que ele foi criado durante a sequência.
- **Destruição de Objeto (Destroy)**: Uma mensagem que termina na linha de vida com um X, indicando que o objeto foi destruído da memória.



Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- Após aprender sobre os componentes e a sintaxe, é fundamental entender o valor real de um diagrama de sequência. Ele não é apenas um exercício de desenho, mas uma ferramenta estratégica para o desenvolvimento de software.

Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- Após aprender sobre os componentes e a sintaxe, é fundamental entender o valor real de um diagrama de sequência. Ele não é apenas um exercício de desenho, mas uma ferramenta estratégica para o desenvolvimento de software. O foco aqui é responder à pergunta: "Por que eu gastaria tempo criando um diagrama de sequência?".

1. Para Documentar o Comportamento do Sistema
2. Para Visualizar a Lógica de Funções Complexas
3. Para Identificar e Corrigir Problemas
4. Para Facilitar a Comunicação entre Equipes

Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

1. Para Documentar o Comportamento do Sistema

- O diagrama de sequência é excelente para criar documentação clara e visual. Ele atua como um "manual de instruções" de um processo.

- **Exemplo**: Em vez de descrever o fluxo de um processo complexo de reserva de passagens em um texto longo, um diagrama de sequência pode mostrar visualmente, e de forma inconfundível, quem envia a mensagem para quem e em que ordem.

Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

2. Para Visualizar a Lógica de Funções Complexas

- Use o diagrama para decompor e entender a lógica de um caso de uso que envolve muitos objetos.
- Ele ajuda a visualizar o fluxo de controle, as dependências e a ordem exata das chamadas de método, tornando mais fácil identificar gargalos, loops infinitos ou lógicas incoerentes.

Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

3. Para Identificar e Corrigir Problemas

- O diagrama de sequência é uma ferramenta poderosa para a análise de sistemas. Ao mapear o fluxo de comunicação, você pode facilmente encontrar:

  - **Gargalos de Desempenho**: Mensagens desnecessárias ou que causam latência.
  - **Dependências Circulares**: Objetos que dependem uns dos outros de forma ineficiente.
  - **Pontos de Falha**: Onde um objeto pode não responder ou falhar.

Quando Usar um Diagrama de Sequência?

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

4. Para Facilitar a Comunicação entre Equipes

- O diagrama de sequência é uma linguagem universal para desenvolvedores, arquitetos, analistas de negócios e gerentes de projeto.
- Ele elimina ambiguidades e garante que todos os membros da equipe tenham uma compreensão comum de como um recurso deve ser implementado. É uma forma visual e direta de validar o entendimento de um requisito.

Estruturas de Fluxo de Controle

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- Até agora, vimos sequências lineares de mensagens. Mas e se o fluxo do seu sistema tiver uma lógica mais complexa? A UML oferece fragmentos (ou operadores combinados) para representar esses cenários.

Estruturas de Fluxo de Controle

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Alternativa (alt)

- **O que é**: Usado para modelar um fluxo "se/senão" (if/else). O fragmento é dividido em seções, e cada seção contém um fluxo de mensagens que só é executado se a condição correspondente for verdadeira.
- **Desenho**: Uma caixa com um rótulo alt no canto superior esquerdo. Uma linha tracejada horizontal dentro da caixa separa as diferentes seções, e cada seção tem uma condição entre colchetes ([ ]).

Alternativa (alt)

- **Exemplo**: No nosso cenário de login, a validação de credenciais pode ter dois resultados:

  - [se sucesso]: Acesso concedido, e o sistema exibe a página principal.
  - [senão]: Acesso negado, e o sistema exibe uma mensagem de erro.

Estruturas de Fluxo de Controle

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Opção (opt)

- **O que é**: Usado para modelar um fluxo "se" simples. A sequência de mensagens dentro do fragmento só é executada se a condição for verdadeira.
- **Desenho**: Uma caixa com o rótulo opt. A condição é escrita entre colchetes na parte superior do fragmento.
  - **Exemplo**: "Se o usuário for um administrador, exiba o painel de controle." A ação de exibir o painel (exibirPainelAdmin()) só acontece sob a condição [se usuario.tipo == 'admin'].

Estruturas de Fluxo de Controle

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Loop (loop)

- **O que é**: Usado para modelar uma repetição. A sequência de mensagens dentro do fragmento se repete por um número definido de vezes ou até que uma condição de saída seja alcançada.
- **Desenho**: Uma caixa com o rótulo loop. A condição de repetição é especificada entre colchetes na parte superior.

Estruturas de Fluxo de Controle

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

Loop (loop)

- **O que é**: Usado para modelar uma repetição. A sequência de mensagens dentro do fragmento se repete por um número definido de vezes ou até que uma condição de saída seja alcançada.
- **Desenho**: Uma caixa com o rótulo loop. A condição de repetição é especificada entre colchetes na parte superior.
  - **Exemplo**: "Para cada item no carrinho de compras, adicione o valor ao total." A mensagem adicionarItemAoTotal() dentro do loop se repete para cada item. A condição pode ser [para cada item].

Conclusão

INSTITUTO FEDERAL
São Paulo
Câmpus Barretos

- Em resumo, o diagrama de sequência transforma a complexa dança de objetos em uma coreografia clara e fácil de entender, tornando o desenvolvimento de software mais eficiente e menos propenso a erros.
  - Se um diagrama de classes é a planta de uma casa, o diagrama de sequência é o guia de como as pessoas se movem por ela.
  - Ele mostra que o verdadeiro poder de um sistema não está apenas no que existe, mas em como tudo funciona junto.
  - Pense nisso: qual o próximo processo complexo que você precisa entender? Um diagrama de sequência pode ser a resposta.