



Universidade Federal de Uberlândia
Graduação em Gestão da Informação



Análise e Projeto de Sistemas

Prof. Luiz Cláudio Theodoro

Object-Oriented Diagrams:
Diagrama de Sequência

Mariana Belloube

DIAGRAMA DE SEQUÊNCIA

Antes de começar...

UML: o que é?



Unified Modeling Language

Uma **linguagem de notação** para uso em projetos de sistemas.

É expressa através de **diagramas** que se dividem em dois grandes grupos: **estruturais** e **comportamentais**.



parte estática



parte dinâmica

Princípio de Pareto no uso da UML:

**20% dos
recursos
da UML**

**atendem 80% dos cenários
a serem especificados**

6
diagramas

20% dos
recursos
de cada
diagrama

*Dos **21 diagramas** disponíveis na
versão atual da UML (2.5.1):*

atendem 80% do que precisamos

e para cada um dos seis:

resolvem 80% do que precisamos fazer

Finalmente,

Diagrama de Sequência

(Também conhecido como *diagrama de eventos* ou *cenário de eventos*)

OBJETIVO

Representar graficamente o comportamento de uma funcionalidade, considerando a interação entre todos os componentes de software relacionados ao seu uso.

TROCANDO EM MIÚDOS...

Tem o objetivo de mostrar como as **mensagens** entre os **objetos** são trocadas no decorrer do tempo para a realização de uma operação.

Ele mostra a **sequência de mensagens transmitidas entre objetos**.

QUANDO USÁ-LO?

O diagrama pode ser usado para diferentes finalidades no contexto da produção de software, destacando:

1. Design (projeto de software);
2. Engenharia Reversa e arqueologia de código;
3. Esboço de ideias.



Símbolos e componentes básicos



Símbolos de
objetos

Representa uma classe ou objeto. O símbolo do objeto demonstra como ele se comportará no contexto do sistema. Atributos de classe não devem ser listados dessa forma.



Símbolo de ator

Mostra as entidades que interagem com, ou são externas ao sistema.



Símbolos e componentes básicos



Caixa de
ativação

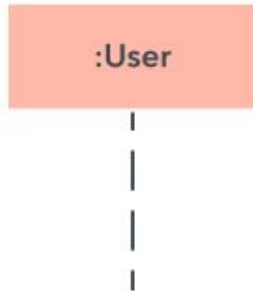
Representa o tempo necessário para que um objeto conclua uma tarefa. Quanto mais tempo a tarefa levar, mais alongada a caixa de ativação se torna.

Símbolos e componentes básicos



Símbolo de
pacote

Usada para conter elementos interativos do diagrama. Também conhecida como quadro, essa forma retangular tem um pequeno retângulo interno para rotular o diagrama.



Símbolo de
linha da vida

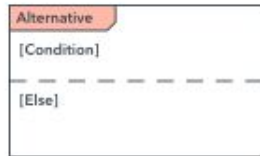
Representa a passagem do tempo, conforme se estende para baixo. Essa linha vertical tracejada mostra os eventos sequenciais que ocorrem a um objeto durante o processo no gráfico. Linhas da vida podem começar com uma forma de retângulo rotulado ou um símbolo de ator.

Símbolos e componentes básicos



**Símbolo de
opção em loop**

Usado para modelar cenários "if/else", ou seja, uma circunstância que ocorrerá apenas sob determinadas condições..



**Símbolo
alternativo**

Simboliza uma escolha (geralmente mutuamente exclusiva) entre duas ou mais sequências de mensagens. Para representar alternativas, usa-se o retângulo rotulado com uma linha tracejada em seu interior.

Símbolos comuns de mensagem

Mensagens síncronas

O remetente deve esperar pela resposta antes de continuar.



Mensagens assíncronas

Não necessitam de uma resposta para que o remetente continue.



Mensagens de retorno assíncronas



assíncrono de criar mensagem

Essa mensagem cria um novo objeto.



Responder a mensagem

Essa mensagem é resposta às chamadas.



Excluir a mensagem

Essa mensagem destrói um objeto



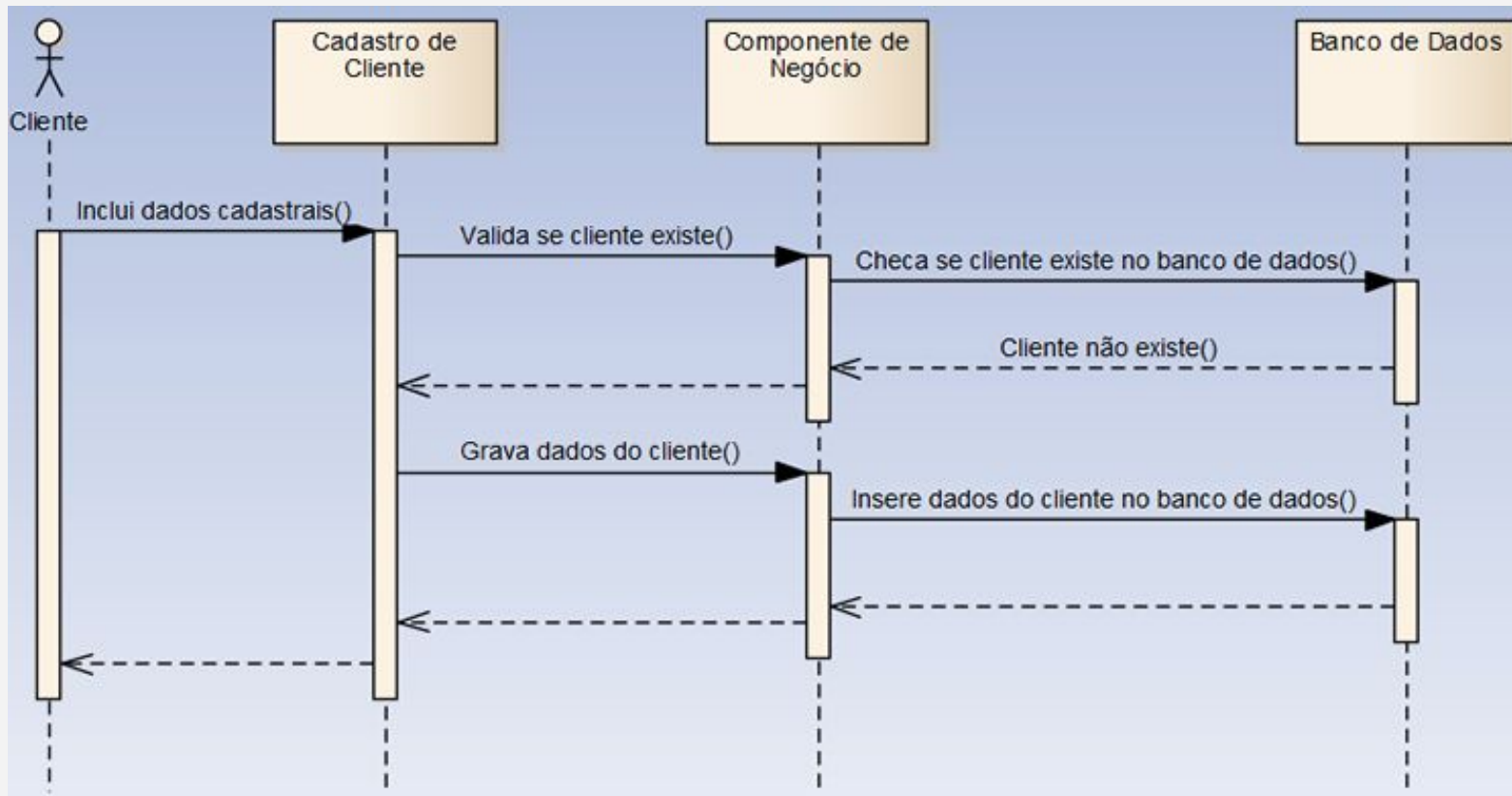
função PESQUISAR CLIENTE

da funcionalidade CADASTRO DE CLIENTE

SEQUÊNCIA DE INSTRUÇÕES

1. O usuário (ator) "Atendente" abre a página "Cadastro de Cliente".
2. O usuário (ator) "Atendente" digita o nome do cliente e clica no comando "Pesquisar".
3. A página "Cadastro de Cliente" aciona o método "CadastroClienteUI.PesquisarClientePorNome(...)" passando a expressão informada como termo de pesquisa.
4. A página "Cadastro de Cliente", através do método "CadastroClienteUI.PesquisarClientePorNome(...)", instancia a classe de negócio e chama o método "ClienteNegocio.PesquisarClientePorNome(...)" passando a string com a expressão informada para pesquisa.
5. O método "ClienteNegocio.PesquisarClientePorNome(...)" instancia a classe de dados e chama o método "ClienteDados.PesquisarClientePorNome(...)".
6. Método "ClienteDados.PesquisarClientePorNome(...)" retorna uma lista com os registros encontrados na busca.
7. Método "ClienteNegocio.PesquisarClientePorNome(...)" recebe uma lista com os registros informados pelo método "ClienteDados.PesquisarClientePorNome(...)".
8. A página "Cadastro de Cliente", através do método "CadastroClienteUI.PesquisarClientePorNome(...)" recebe uma lista com os registros informados pelo método "ClienteNegocio.PesquisarClientePorNome(...)".
9. A página "Cadastro de Cliente" exibe a lista de registros recebidos para o usuário chamador.

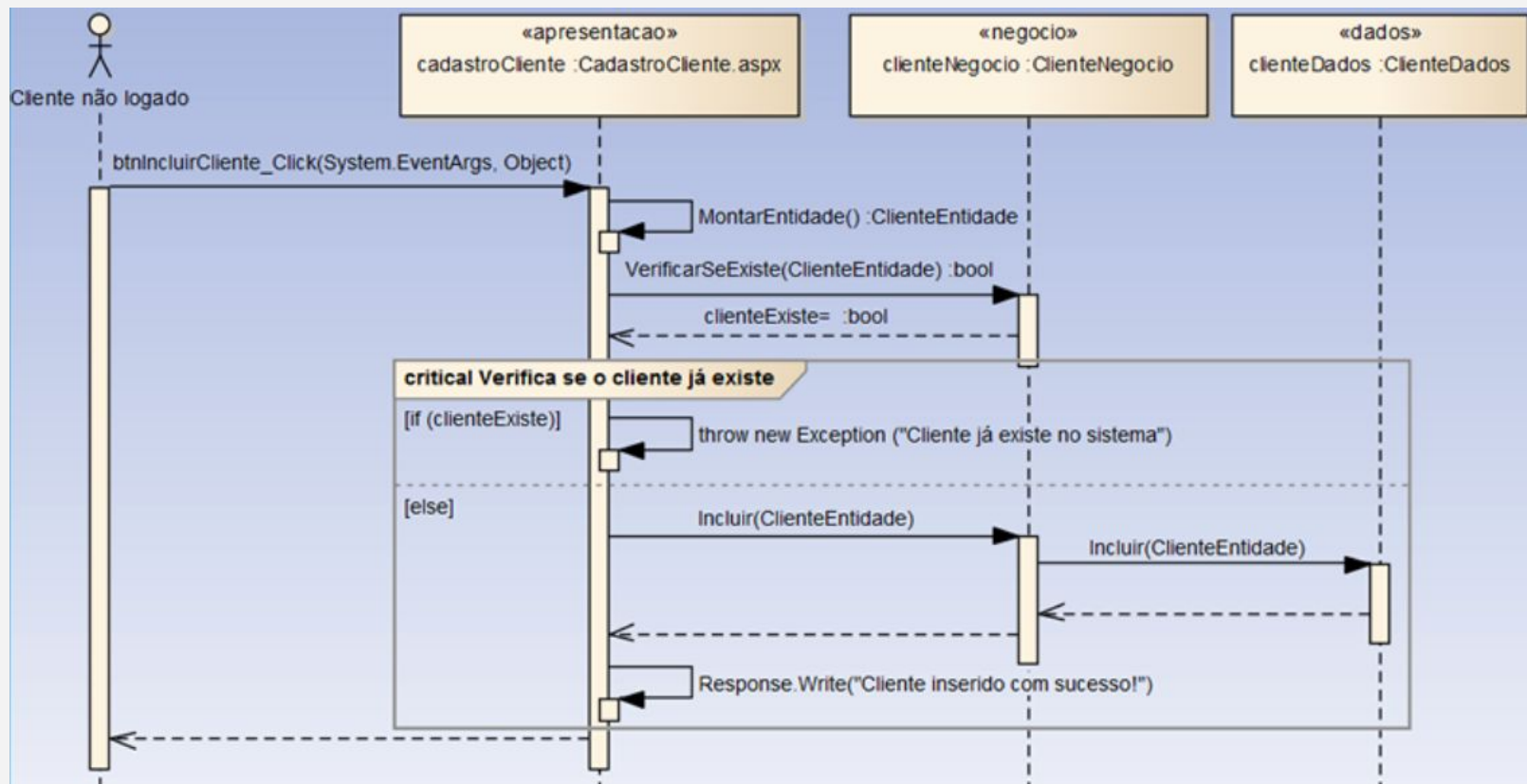
função **PESQUISAR CLIENTE** da funcionalidade **CADASTRO DE CLIENTE**



ESBOÇO DE IDEIAS

função PESQUISAR CLIENTE

da funcionalidade CADASTRO DE CLIENTE



Concluindo...

O diagrama de sequência da UML é uma excelente ferramenta para ilustrar **como as funções de uma funcionalidade interagem entre si** (cliques de botão, validações internas, acessos entre camadas de aplicação etc.)

Fazendo uso *na medida certa* (seja como um artefato de design rigoroso ou como rascunho num quadro de desenho), ele pode clarear as ideias de analistas e desenvolvedores e proporcionar **excelente entendimento do escopo**.

Referências

Diagramas UML:

<https://www.ateomomento.com.br/diagramas-uml/>

Diagrama de Sequência:

<https://www.ateomomento.com.br/diagrama-de-sequencia-uml/>

<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml>

Muito obrigada!
