

FMU – **F**aculdades **M**etropolitanas **U**nidas

Modelagem de Sistemas

Aula 1 - Introdução

Índice

- Apresentação do Professor
- Critérios de Avaliação
- Plano de Ensino
- Critérios de Avaliação
- Introdução
- Modelos orientados a objetos X modelos estruturados
- Orientação a Objetos
- UML
- Diagramas
- Ferramentas CASE
- Bibliografia

Apresentação do Professor

- **Nome:** Rafael de Alencar Segura
- **Titulação:**
 - **Graduação:** Ciências da Computação
 - **Pós Graduação** em Engenharia de Web Sites;
 - **Certificação:** OCA – Oracle Certified Associate – 10G.
 - **Mestre** – área: Ensino de Ciências
- **Profissão:** Professor Universitário e Administrador de Dados - Prodam [**Empresa de Tecnologia da Informação e Comunicação do Mun. SP**]
- **CV Lattes:** <http://lattes.cnpq.br/1472669515916215>
- **LinkedIn:** <https://br.linkedin.com/in/rafael-segura-26619629>
- **Interesses:** banco de dados , programação orientada a objetos, algoritmos, modelagem de sistemas, modelagem de dados, educação a distância, bateria, guitarra, aviação.

Dinâmica das Aulas

- Sala de Aula e Laboratório;
- Introdução teórica e aplicação prática da teoria.
- Exercícios diários para fixação da matéria.

Critérios de Avaliação

O processo avaliativo constará de uma Prova Regimental no valor de 7,0 (sete) pontos e de três instrumentos parciais de avaliação, que comporão juntos 3,0 (três) pontos.

7,0 – Regimental: avaliação marcada pela instituição

3,0 – Parcial = 2 provas (P1 e P2) + projeto (PRJ), cada um valendo até um ponto.

OBS: Não teremos prova substitutiva das provas P1 e P2.

Critérios de Avaliação

- Média = Regimental + Parcial

Média $\geq 7.0 \rightarrow$ Aprovado

Média < 7.0 e $\geq 4.0 \rightarrow$ Exame

Média $< 4.0 \rightarrow$ DP

Média Final = (Média + Prova) / 2

Média Final $\geq 5 \rightarrow$ Aprovado

Média Final $< 5 \rightarrow$ DP

Avaliações

- **Não teremos prova Substitutiva da P1, P2 e Projeto.**
- O aluno que perder a avaliação Regimental **poderá fazer a prova SUB (substitutiva) desde que solicite a prova diretamente na secretaria.**
- O conteúdo será sempre do semestre inteiro;

Exercícios propostos em sala

- Exercícios de Avaliação Continuada
 - Praticamente em todas as aulas
 - Exercícios sobre a aula do dia, entrega via Ambiente Virtual;

Ambiente Virtual para download das aulas

- Edmodo
- <https://www.edmodo.com/>

Introdução: Por que modelar software? [1/2]

- Modelagem > é a essência de uma determinada estrutura, suas relações internas, sua evolução e os fatores que determinam as possíveis mudanças.



- É importante, por mais simples que seja, que todo sistema possua um modelo antes de iniciar a implementação uma vez que os mesmos podem **aumentar o número de funcionalidades, complexidade e abrangência**.

Introdução: Por que modelar software? [2/2]

- Os sistemas normalmente são **dinâmicos** uma vez que estão em constante mudanças devido a alguns fatores conforme abaixo apresentados:
 - Solicitações de mudanças ou melhorias nos sistemas realizada pelos clientes;
 - Mudanças no mercado;
 - Governo através de novas leis ou alteração em leis já existentes.

Modelo de Software

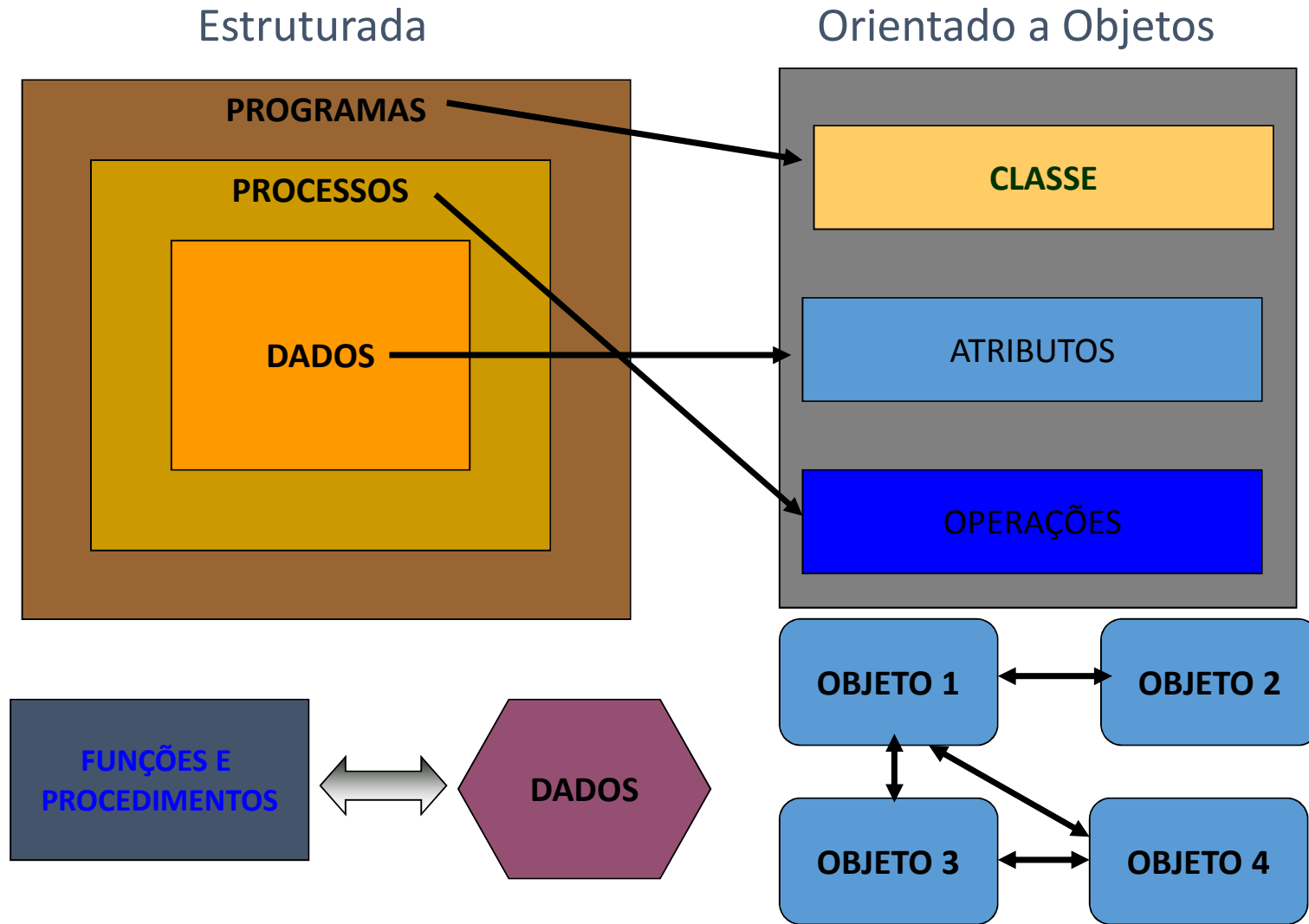
- Captura uma visão de um sistema físico abstraindo o mesmo em aspectos estruturais ou comportamentais do software;

Modelos orientados a objetos X modelos estruturados

- Orientação a Objetos e abordagem estruturada são paradigmas diferentes de desenvolvimento de sistemas.

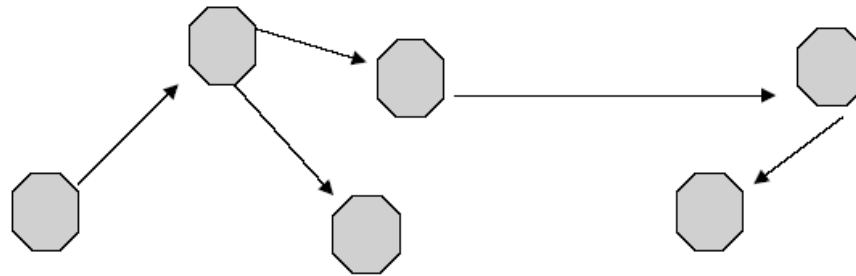
	Abordagem Estruturada	Abordagem Orientada a Objetos
Funcionalidades	são definidas por dfd	Conjunto de objetos interativos
Dados	separados das funcionalidades	Dados armazenados em cada objeto existente

Modelos orientados a objetos X Modelos estruturados



A Orientação a Objetos (OO)

- O Desenvolvimento Orientado a Objetos (OO) objetiva a construção de sistemas a partir do entendimento do mundo real como um conjunto de **objetos que comunicam-se entre si de forma coordenada.**



- Para trabalhar com orientação a objetos, é necessário:
 - Entender quais são os objetos envolvidos no domínio do problema
 - Entender como se comunicam no mundo real
 - Projetar a forma como devem ser implementados

UML – *Unified Modeling Language* ^[1/2]

- Anos 80 e 90, os desenvolvedores possuíam um conjunto de métodos e notações com iniciativas isoladas o que não contribuía para a melhoria no processo de desenvolvimento de software.
- Em 1996 a união entre Jacobson, Rumbaugh e Grady Booch teve início a proposta da **UML incorporando as melhores práticas de construção de sistemas orientados a objetos.**

UML É A UNIFICAÇÃO DOS MÉTODOS OO

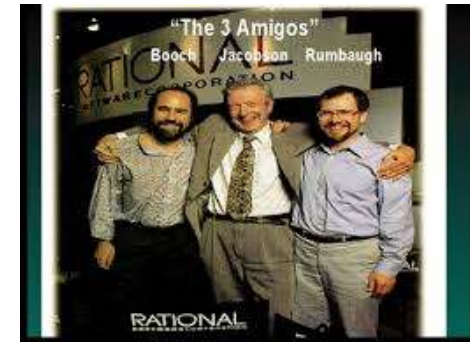
BOOCH



OMT



OOSE



UML – *Unified Modeling Language* [2/2]

- A UML é uma **linguagem de modelagem unificada** que é a especificação mais conhecida do OMG [**O**bject **M**anagement **G**roup] e é a norma da indústria da informática para descrever graficamente “software”. <http://www.omg.org/spec/UML/>
- O objetivo é propiciar aos arquitetos de sistemas e demais profissionais envolvidos **um conjunto de ferramentas para análise, projeto e implementação de sistemas**, bem como para modelagem de processos e similares.
- A UML **não é uma linguagem de programação** e sim uma linguagem de notação visual para especificação (modelagem) de sistemas de informação orientados a objeto (OO).

UML 2.0 : Diagramas

- Na versão 2.0 possuímos treze tipos de diagramas distribuídos em três categorias conforme abaixo:

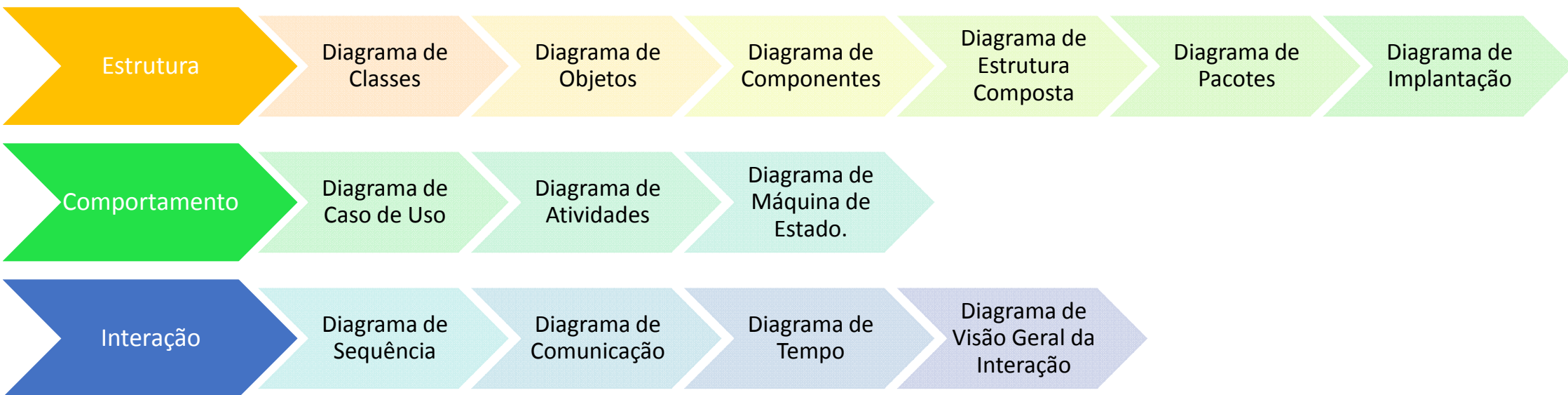
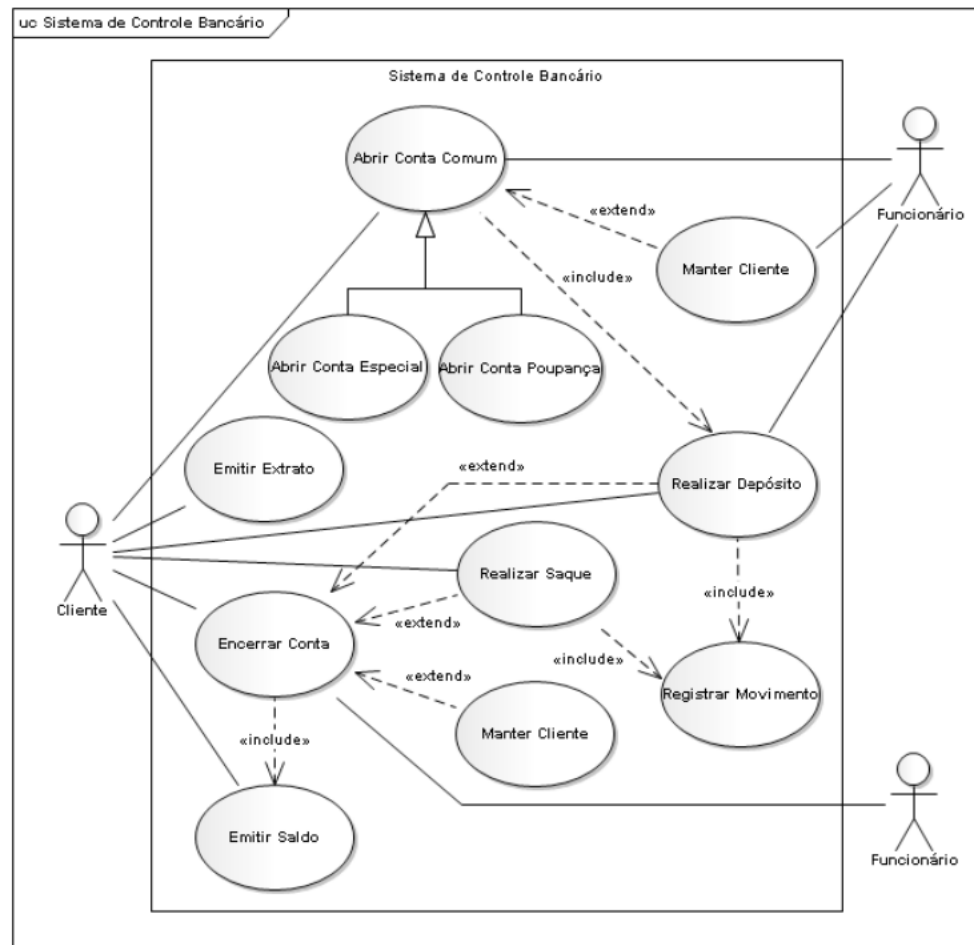


Diagrama de Caso de Uso

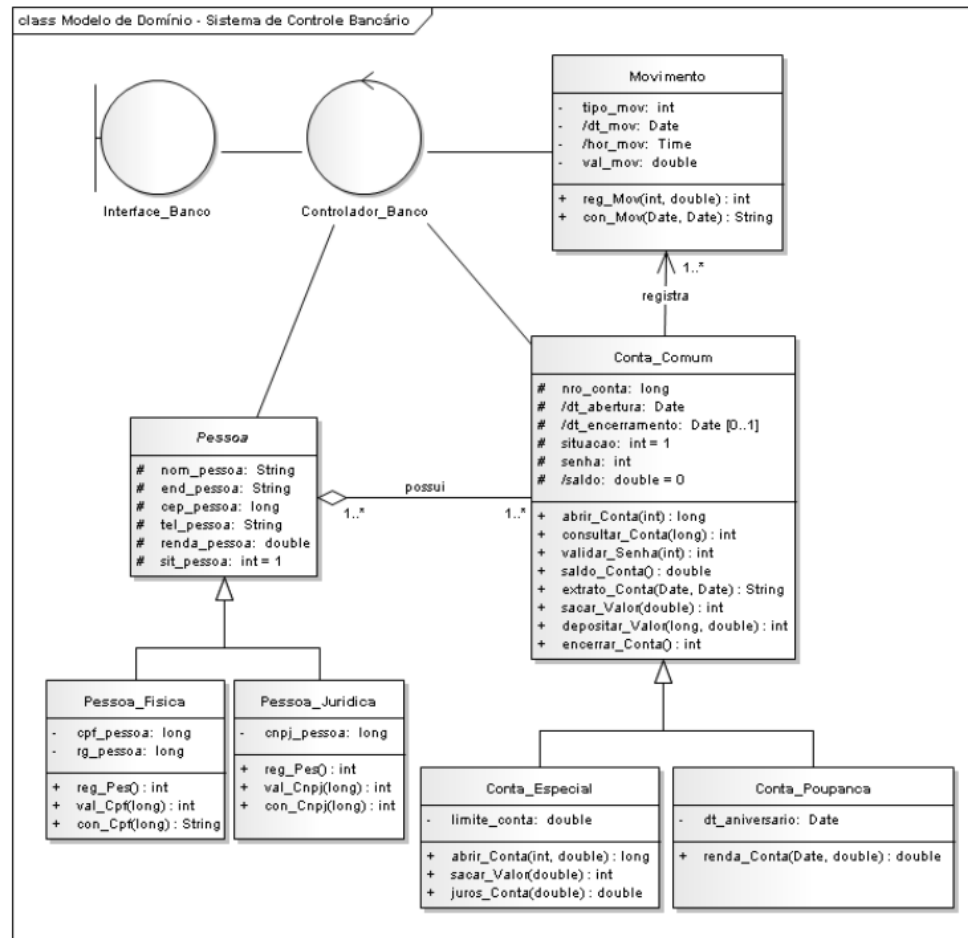
- Apresenta um sistema de informação pela visão do usuário, mostrando os módulos, seus usuários e papéis cada um vai desempenhar para seu funcionamento.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Classes

- Apresenta a estrutura das classes utilizadas pelo sistema, contendo os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Objetos

- Representa a instância de uma classe, ou seja mostra o estado do objeto (valores armazenados em um determinado momento do tempo).

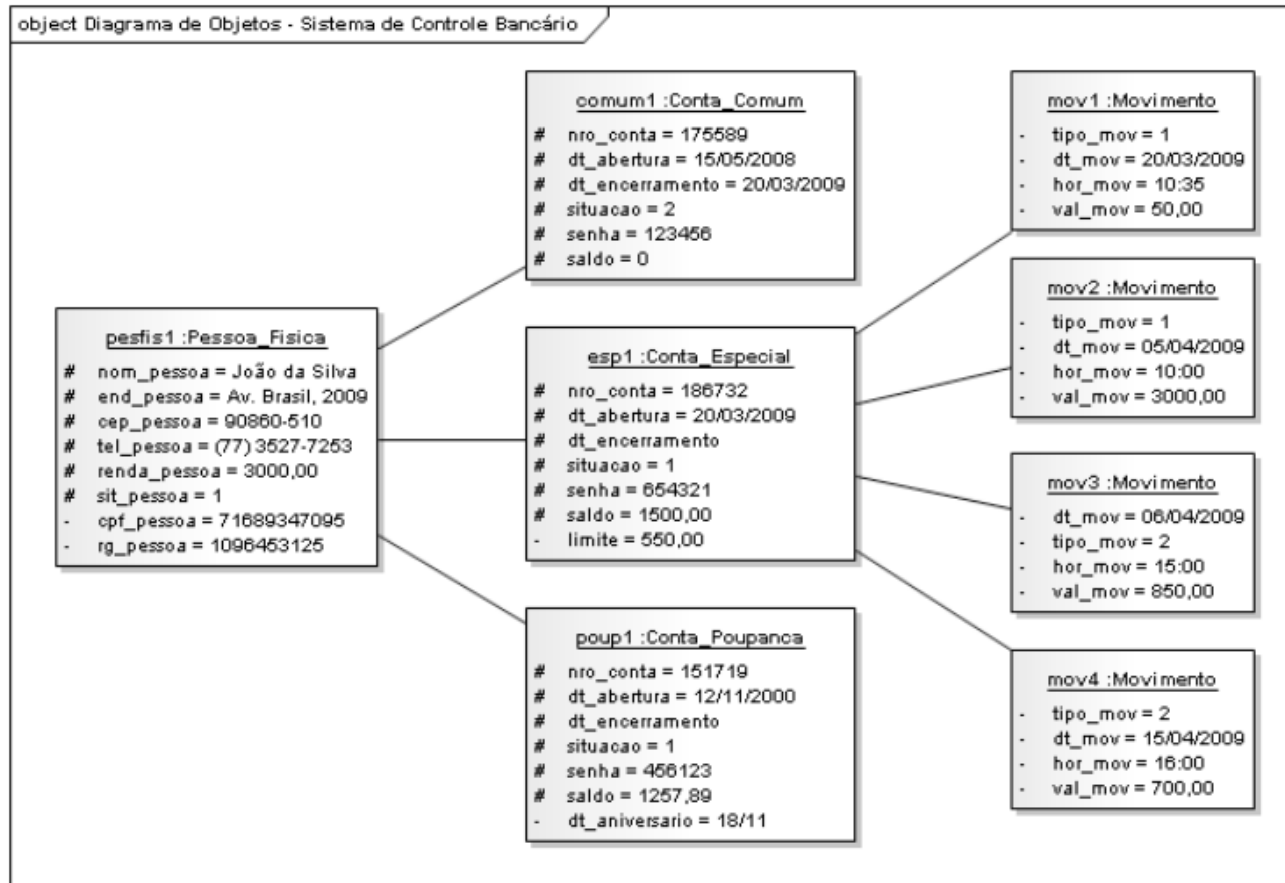
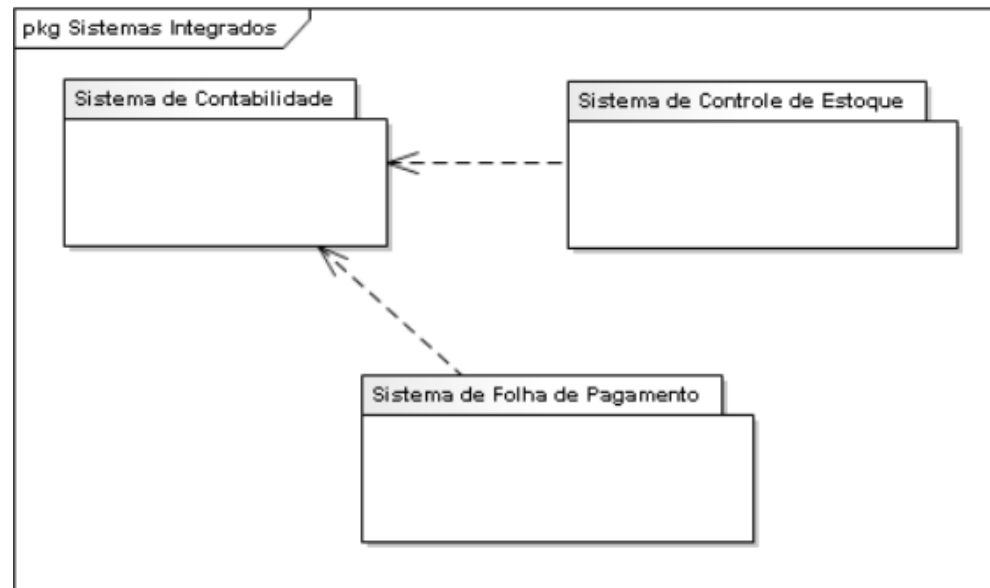


Diagrama de Pacotes

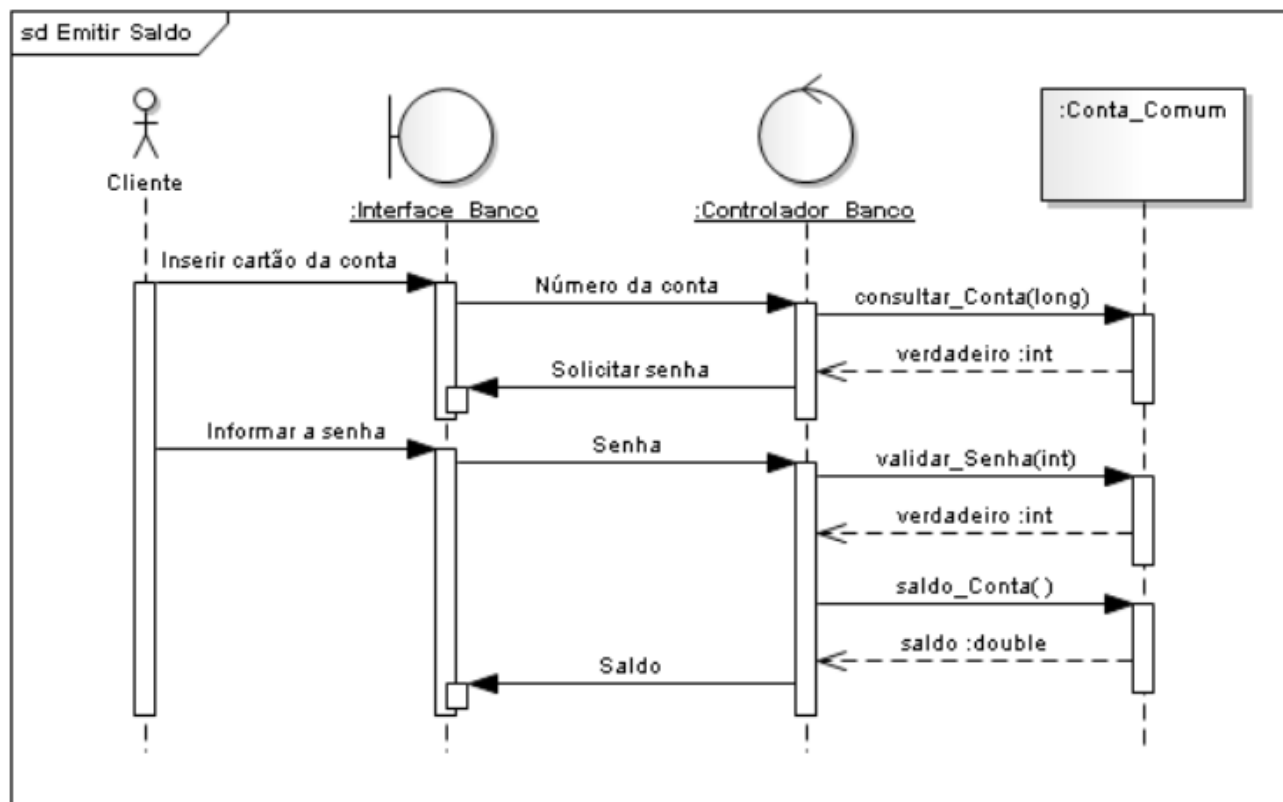
- Diagrama estrutural que tem por objetivo apresentar os sub módulos que compõe o sistema principal.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Sequência

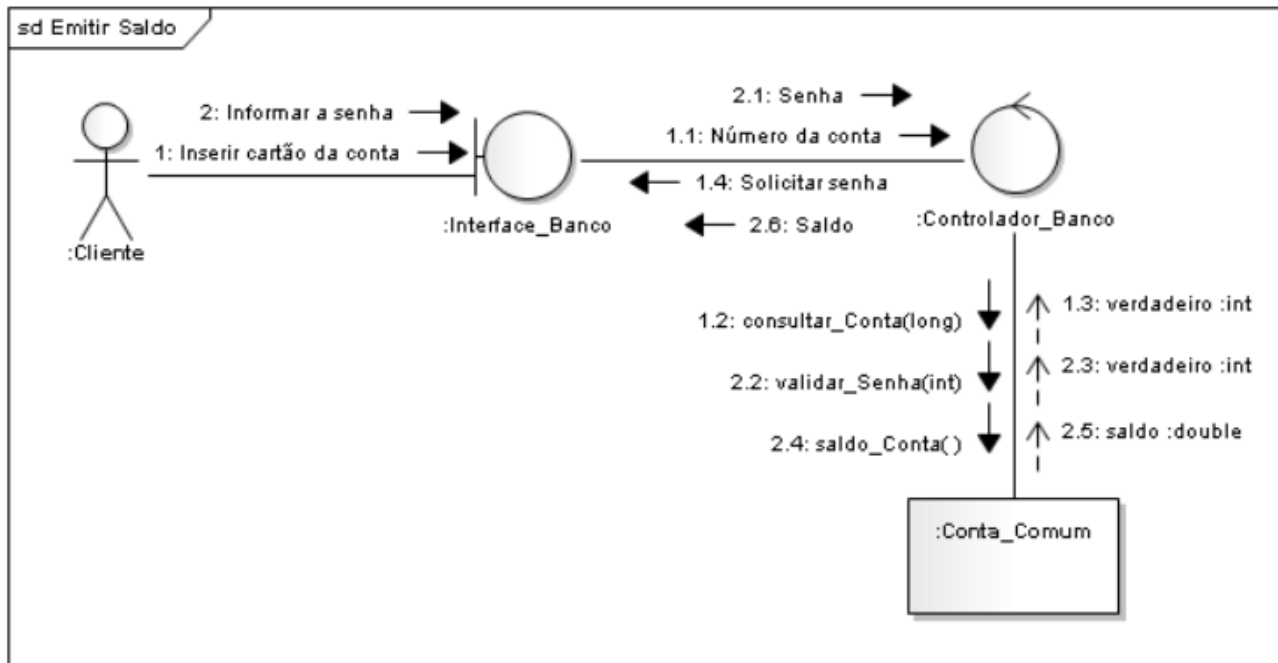
- Diagrama comportamental que define a ordem em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Comunicação

- Diagrama de colaboração até a versão 1.5 da UML;
- Associado ao diagrama de sequência porém com enfoque distinto: este diagrama se preocupa **não com a temporalidade** do processo e **sim como os elementos estão vinculados e como as mensagens são trocadas durante o processo**.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Máquina de Estados

- Apresenta os estados de uma instância de classe ou os estados de um processo representado por um caso de uso;
- Pode se basear em um caso de uso e se apoiar no Diagrama de Classes;

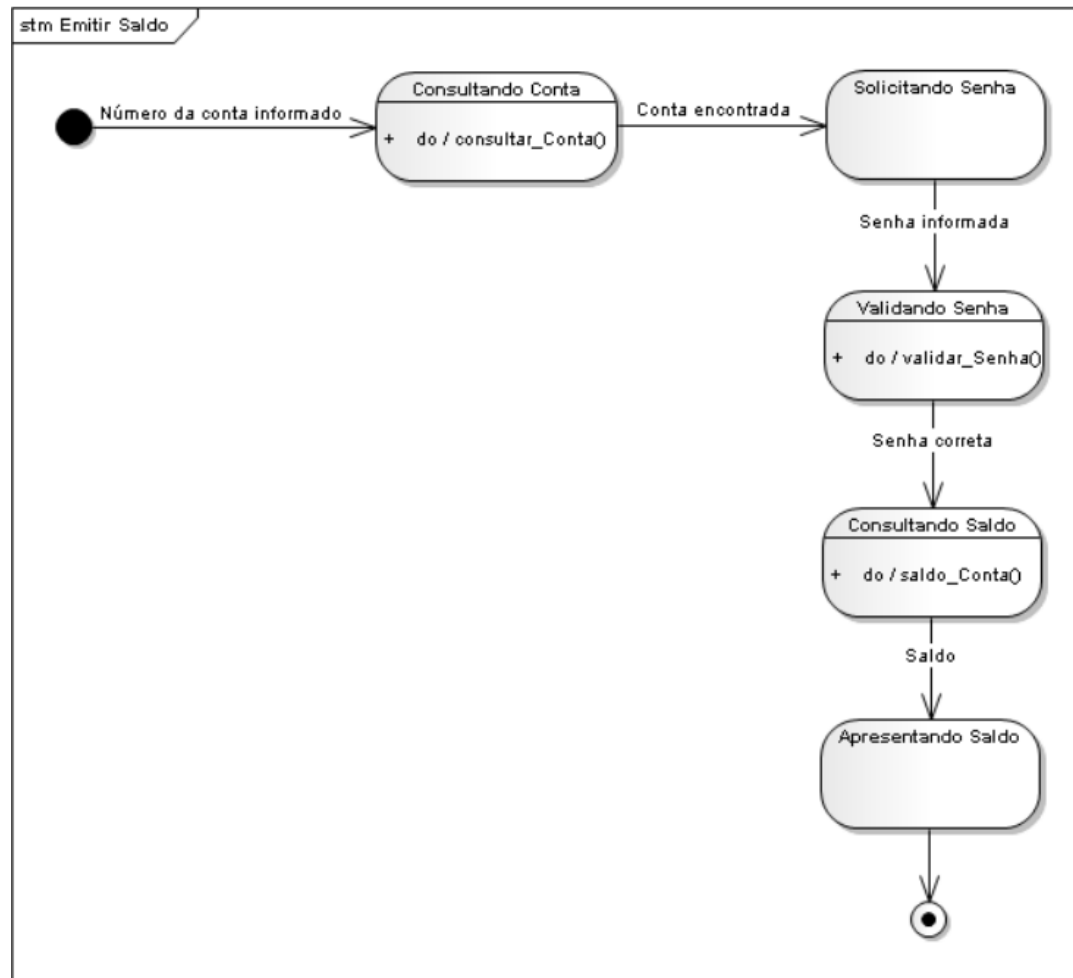
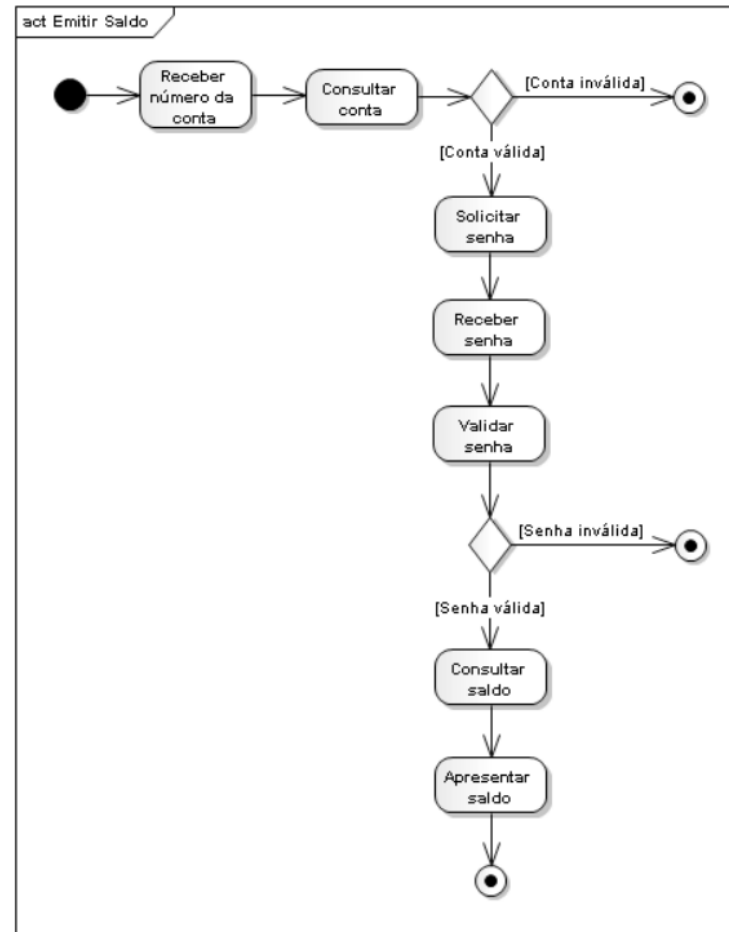


Diagrama de Atividade

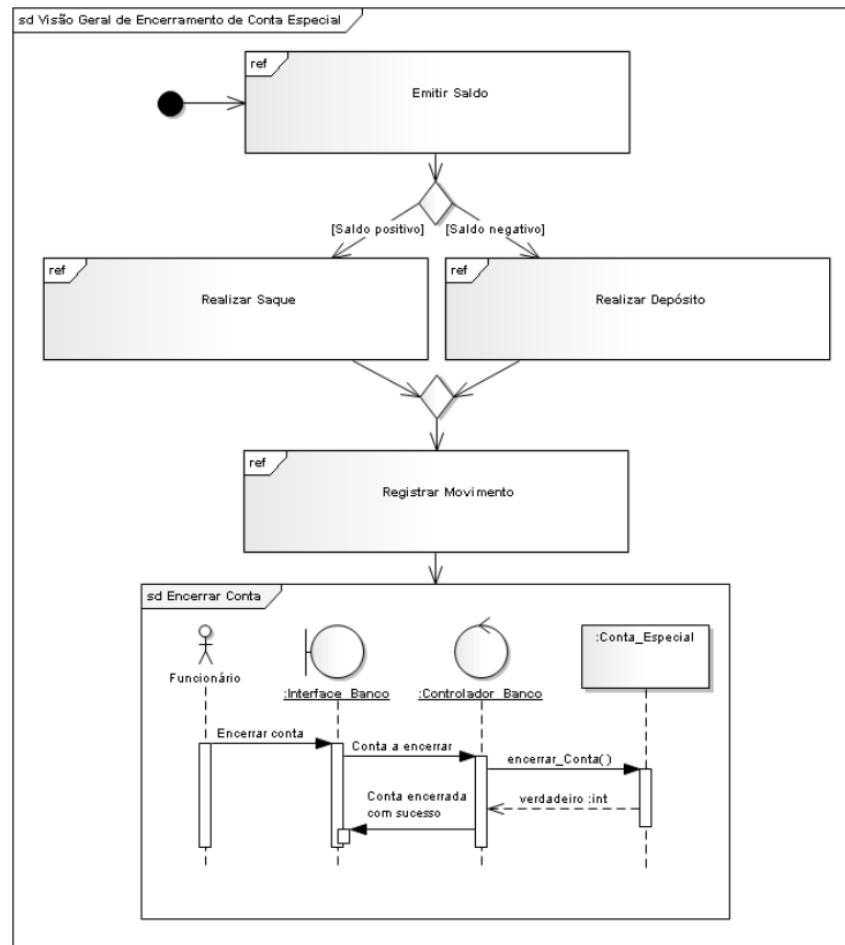
- Descrevem os passos a serem percorridos para a conclusão de uma atividade específica podendo modelar um processo completo. Concentra-se na representação do fluxo de controle.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Visão Geral de Interação

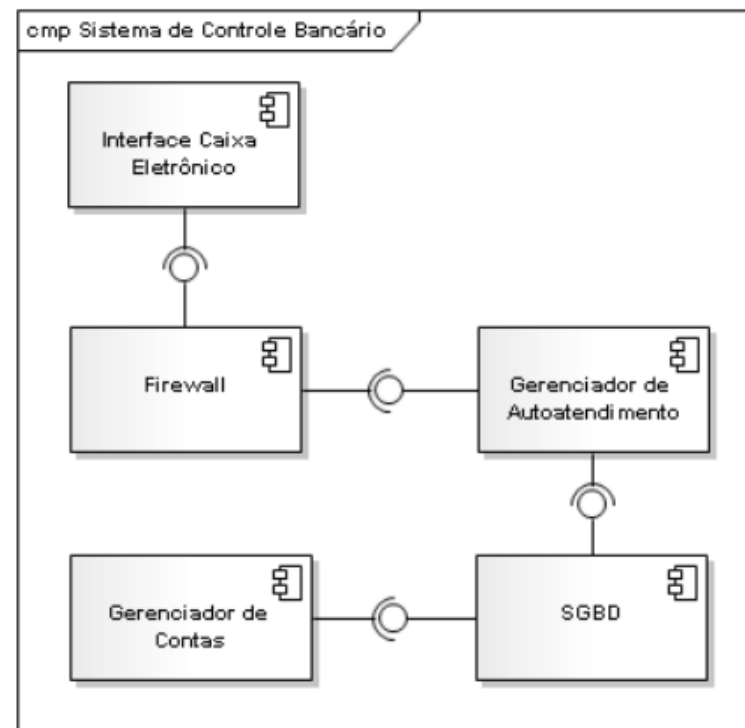
- É uma variação do diagrama de atividade que fornece uma visão geral dentro de um sistema ou processo de negócio. {existente apenas a partir da versão 2.0 da UML}



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Componentes

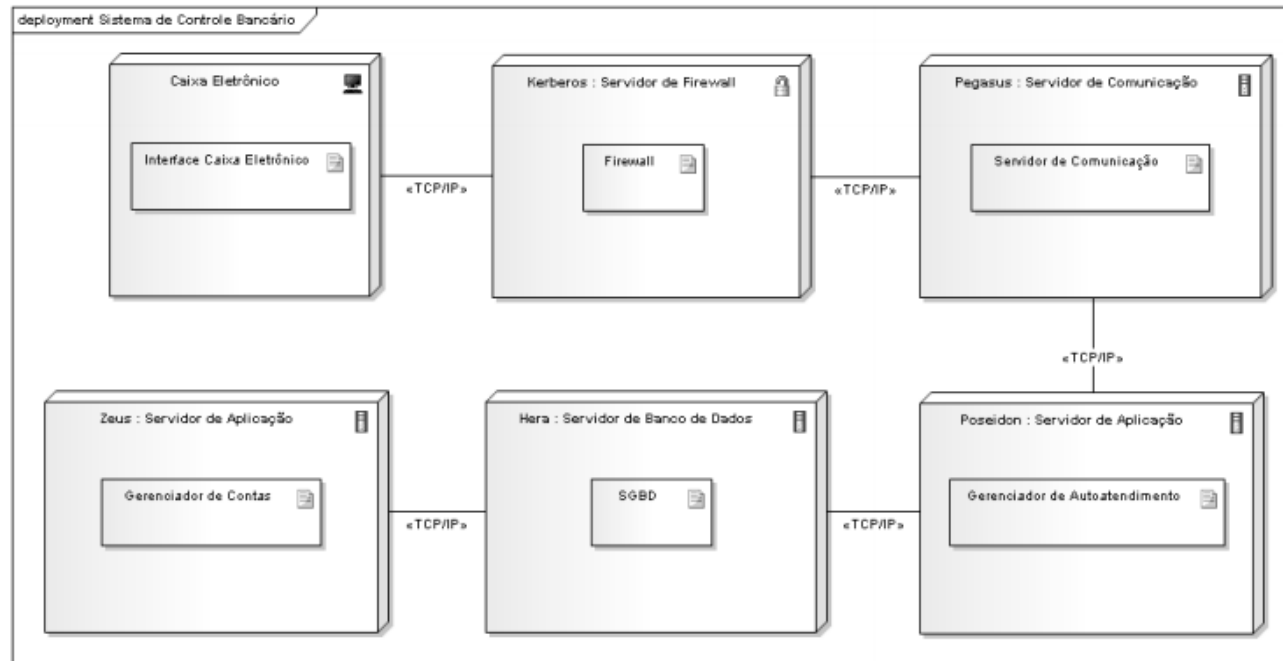
- Associado a linguagem de programação;
- Representa os componentes em termos de módulos de código fonte, bibliotecas, formulários, módulos executáveis.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Implantação

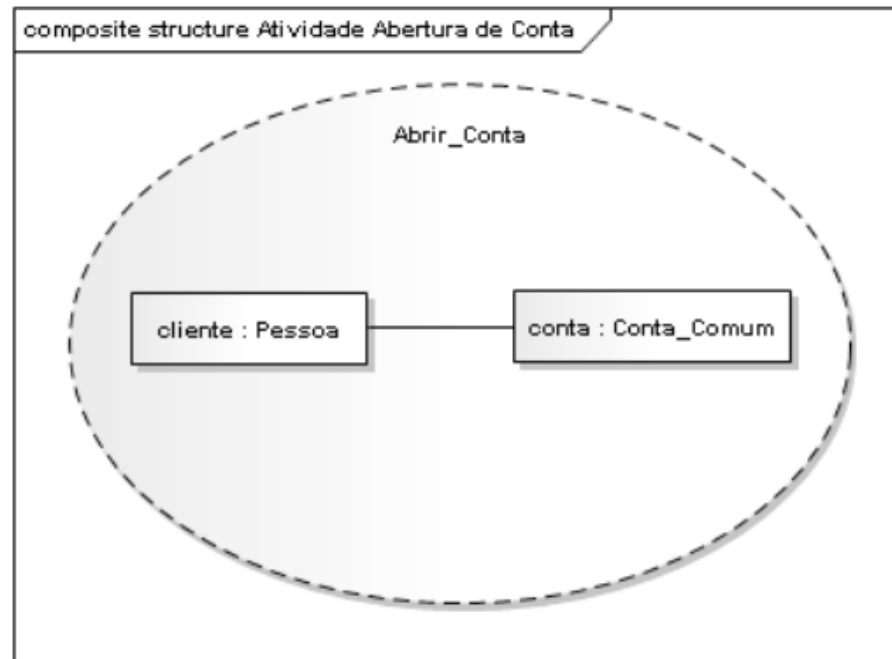
- Determinam as necessidades de hardware do sistema (servidores, protocolos de comunicação, ou seja, todo aparato físico sobre o qual o sistema deverá ser executado);



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Diagrama de Estrutura Composta

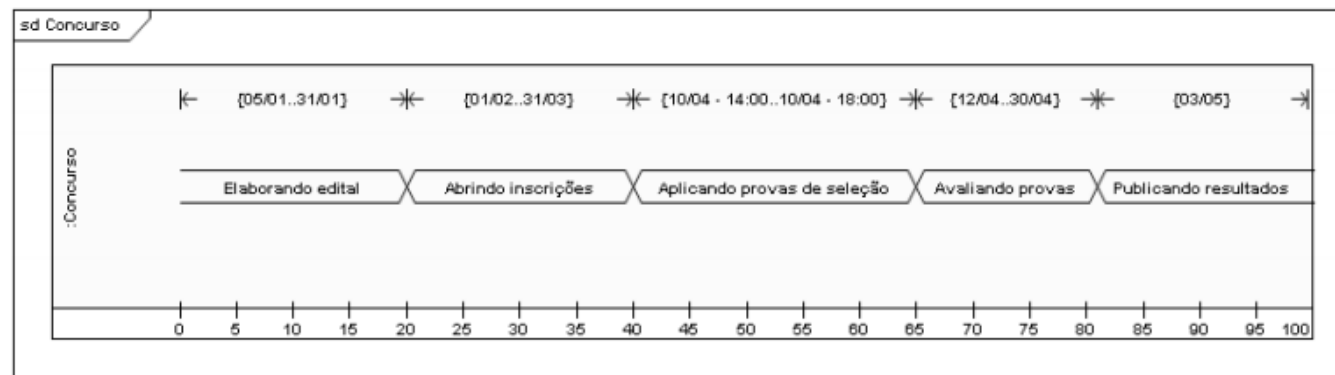
- Estrutura interna de um classificador [classe ou componente] detalhando as partes internas que o compõem.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

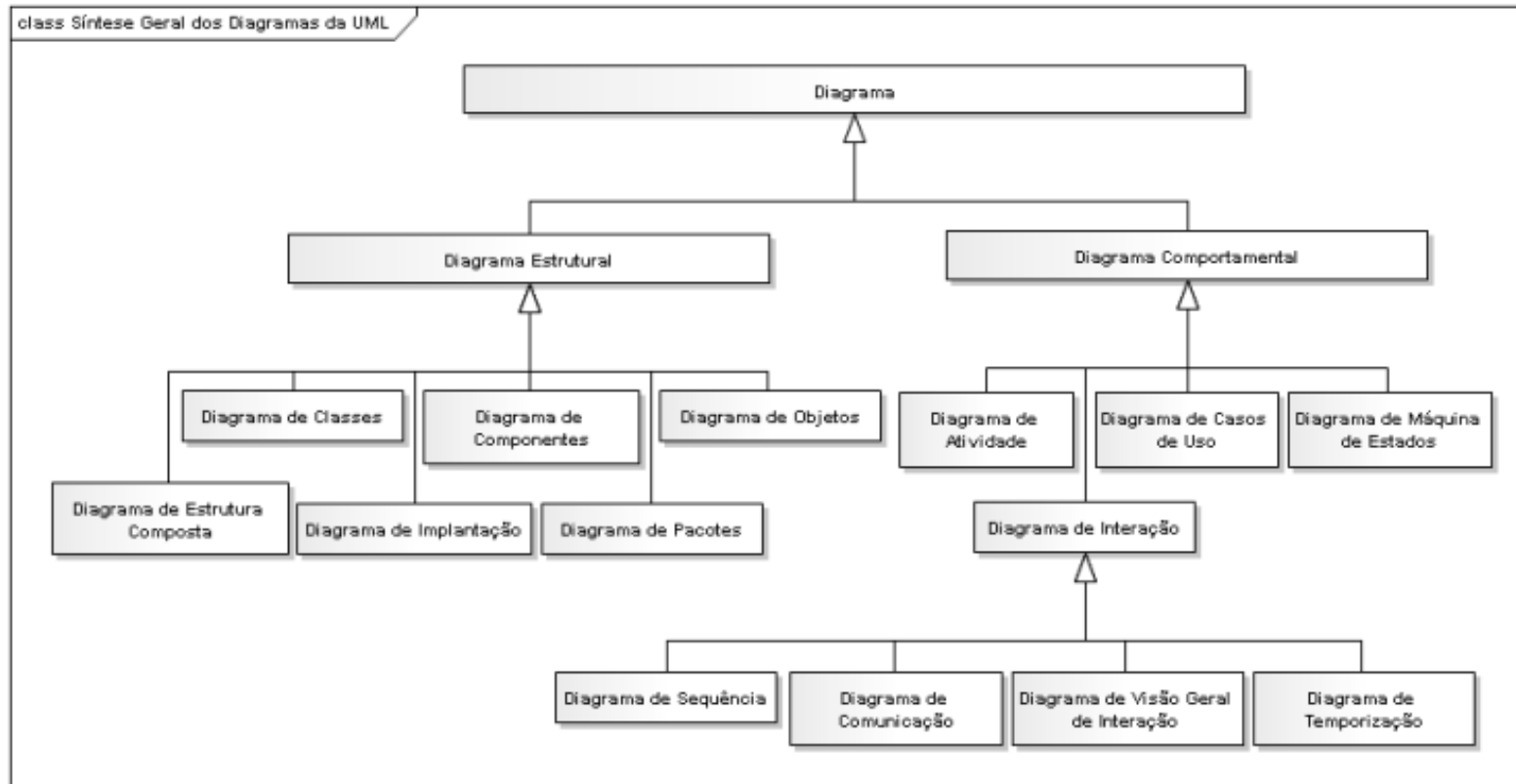
Diagrama de Tempo ou Temporização

- Mudança no estado ou condição de uma instância de uma classe ou seu papel durante um período.



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Síntese Geral dos Diagramas



Fonte: https://novatec.com.br/livros/abordagem_p/capitulo9788575222812.pdf

Ferramentas CASE baseadas em UML

- Ferramentas CASE (Computer-Aided Software Engineering ou Engenharia de Software Auxiliada por Computador) são softwares que colaboram para a execução de uma ou mais atividades realizadas durante o processo de engenharia de software. Algumas ferramentas CASE suportam a UML.

Ferramentas

- **Star Uml**- ferramenta gratuita – site: <http://staruml.io/>
- **Astah** – ferramenta gratuita – site: <http://astah.net/download>
- **Enterprise Architect** – é uma das ferramentas que mais oferecem recursos compatíveis com a UML em sua última versão. Apesar de não dispor de uma edição para a comunidade, a Sparx Systems, a empresa que produz a Enterprise Architect, disponibiliza uma versão trial, que pode ser utilizada por cerca de 60 dias, no site www.sparxsystems.com.au.
- **Visual Paradigm for UML** ou VP-UML – oferece uma edição para a comunidade e pode ser encontrada no site www.visual-paradigm.com
- **Poseidon for UML** – esta ferramenta também tem uma edição para a comunidade no site www.gentleware.com
- **ArgoUML** – é totalmente livre e pode ser baixada no site: www.argouml.tigris.org

Referências Bibliográficas

Livro: **Aprenda UML por meio de estudos de caso**

Autor: Wilson Moraes Góes

ISBN: 978-85-7522-346-8

Páginas: 288

Ano: 2014



Livro: **UML 2 - Guia Prático - 2ª Edição**

ISBN: 978-85-7522-385-7

Páginas: 192

Ano: 2014



Livro: **UML 2 - Uma Abordagem Prática - 2ª Edição**

Autor: Gilleanes T. A. Guedes

ISBN: 978-85-7522-193-8

Páginas: 488

Ano: 2011



OBJECT MANAGEMENT GROUP®

<http://www.omg.org/spec/UML/>