INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Disciplina

Modelagem de Domínio e Conceitos de Orientação a Objetos

PROF. FERNANDO TAMBERLINI ALVES

Revisão

- **Engenharia de Software**: técnicas, métodos e ferramentas de apoio que nos auxiliam no desenvolvimento de software;
- **Arquitetura de Software:** características de como o software será desenvolvido;
- **Modelagem de Domínio:** identificação do contexto e levantamento dos requisitos do software que será desenvolvido;
- **Metodologia de Desenvolvimento:** etapas específicas a serem percorridas durante o desenvolvimento de um software;
- **Tipos de Dados em JavaScript:** linguagem dinamicamente e fracamente tipada.

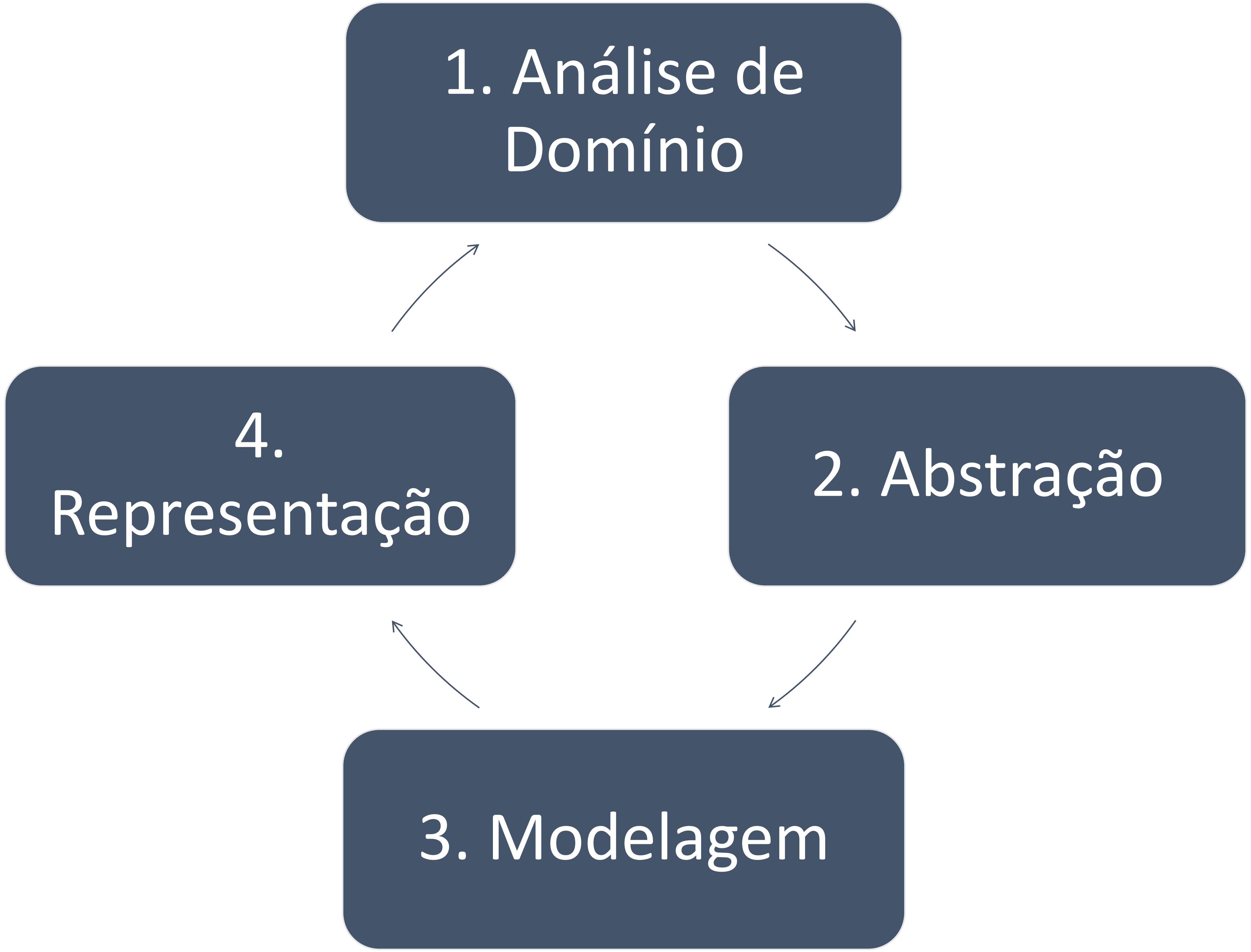
Modelagem de Domínio

1. Identificar o contexto do Sistema de Informação (Dados, Processo e Atores Envolvidos);

2. Idealizar como funcionará o Sistema de Informação pretendido;

3. Modelar a solução idealizada;

4. Representar o modelo idealizado numa linguagem clara e legível.



Modelo de Desenvolvimento Ágil

- O modelo ágil no desenvolvimento de software é uma abordagem moderna e flexível que se concentra na entrega contínua de software funcional, **de modo iterativo e incremental**, e na colaboração próxima com os clientes. Algumas das metodologias ágeis mais conhecidas incluem Scrum, Kanban e Extreme Programming (XP).

Visão Geral do Processo de Desenvolvimento Extreme Programming

Javascript - Tipo de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

JavaScript é uma linguagem de programação que é fracamente e dinamicamente tipada, o que significa que o tipo de dados de uma variável é determinado em tempo de execução e sem a necessidade de declarar o seu tipo.

Em JavaScript, os tipos de dados podem ser divididos em duas categorias principais: primitivos e não primitivos (também chamados de objetos).

Primitive Data Types

- Boolean
- Number
- BigInt
- String
- Undefined
- Null
- Symbol

Reference Data Types

- Array
- Object
- RegEx
- Date
- ....

Ambiente Javascript

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Ambiente de Desenvolvimento Integrado (IDE)
  - Editor (VSCode)
  - Navegador (Interpretador de Javascript)

- Forma de Execução
  - Tag <script> no HTML
  - Arquivo .js separado de HTML
  - Console do Navegador

- Forma de Saída
  - Alert()
  - Console.log()
  - Tag HTML

Atividade Prática

1. Abrir o editor de texto VSCODE;

2. Criar o arquivo aula03.html contendo os seguintes elementos:

   a) Título <title>
   b) Cabeçalho <h1>
   c) Formulário <form> cos os seguintes campos:
      - Nome <input type="text>
      - E-mail <input type="email">
      - Data Nascimento <input type="date">
      - Telefone <input type="tel">
      - Sexo: M/F <Input type="radio>
      - Peso: <input type="text>
      - Altura: <input type="text>
      - Botão Limpar <input type="reset">
      - Botão Calcular IMC<input type="submit">

Atividade Prática

3. Inserir a tag <script> com as instruções para calcular o IMC e imprimir o resultado no console e na caixa de diálogo do navegador.

4. Criar um arquivo imc.js no VSCODE com o conteúdo disponível na página do professor (Função calcularIMC – Java Script)

5. Comparar o seu código com o código similar disponível na página do professor (Exemplo HTML – Sem Erro e Exemplo HTML – Com CSS)

Utilize uma ferramenta de comparação !!!

Programação Orientada a Objetos

A Orientação a Objetos (OO) é um paradigma de programação que se baseia na ideia de organizar o código em torno de objetos, que são instâncias de classes. É uma abordagem poderosa e amplamente utilizada no desenvolvimento de software, pois ajuda a criar sistemas **mais organizados, modulares e reutilizáveis**.

A Orientação a Objetos promove uma abordagem mais natural e organizada para modelar o mundo real em código, tornando-o mais legível, manutenível e flexível. É amplamente utilizada em linguagens de programação como Java, C++, Python, C#, entre outras, e é aplicável a uma variedade de domínios de desenvolvimento de software, desde aplicações desktop até sistemas distribuídos e aplicativos web.

Programação Orientada a Objetos

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Aqui estão os principais conceitos do paradigma de Orientação a Objetos:

Objeto: Um objeto é uma instância concreta de uma classe. Ele representa uma entidade do mundo real e possui atributos (propriedades) que descrevem seu estado e métodos que definem seu comportamento. Por exemplo, um objeto "Carro" pode ter atributos como "cor" e "velocidade" e métodos como "ligar" e "desligar".

Classe: Uma classe é um modelo ou uma definição abstrata que descreve as características comuns a um conjunto de objetos. Ela serve como um plano para a criação de objetos. A classe define os atributos e métodos que seus objetos terão em comum. Por exemplo, uma classe "Carro" define os atributos e métodos que todos os carros terão

Criando o 1º Objeto

- Criar o objeto "pessoa" em Javascript utilizando a tag script, arquivo separado e no console. Imprimir no console, na caixa de diálogo – Alert e no HTML.

```javascript
let pessoa = {
	nome: 'Maria',
	peso: '58',
	altura: '1.58',

	calcularIMC(){
		return this.peso/(this.altura*this.altura);
	}
}

pessoa.calcularIMC()
```

Criando o 1º Objeto

- Página HTML contendo o objeto pessoa dentro a TAG <script> e imprimindo o valor de IMC em 3 formas.

aula03.html > html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MDCOO</title>
</head>
<body>
    <h1>Aula 03 - TAG SCRIPT</h1>
    <p id="saida"></p>
    <script>
        let pessoa = {
            nome: 'Maria',
            peso: '58',
            altura: '1.58',
        calcularIMC(){
            return this.peso/(this.altura*this.altura);
        }
       }
    console.log('IMC = '+pessoa.calcularIMC())
    alert('IMC = '+pessoa.calcularIMC())
    document.getElementById("saida").innerHTML = "IMC = " + pessoa.calcularIMC();
    </script>
</body>
</html>
```

Como representar um Software?

A Linguagem de Modelagem Unificada, ou **UML (Unified Modeling Language),** foi criada através da colaboração de um grupo de pessoas influentes na área de engenharia de software nos anos 90.

Nos anos 90, a engenharia de software estava passando por uma fase de transição. As metodologias tradicionais estavam sendo questionadas e uma necessidade crescente de uma linguagem padrão para modelagem de sistemas de software estava surgindo. Havia várias metodologias de modelagem em uso, como Booch, OMT (Object Modeling Technique) e OOSE (Object-Oriented Software Engineering), cada uma com suas próprias notações e abordagens.

Como representar um Software?

A UML foi inicialmente desenvolvida como uma fusão de três metodologias principais:

Booch Method:

- Criado por Grady Booch, era uma das metodologias mais antigas e amplamente adotadas para modelagem de software.
- Focava em diagramas de classes, diagramas de estado, diagramas de sequência, entre outros.

Object Modeling Technique (OMT):

- Desenvolvido por James Rumbaugh, era outra abordagem popular para modelagem de sistemas orientados a objetos.
- Tinha um foco especial em diagramas de objetos, diagramas de estados, entre outros.

Object-Oriented Software Engineering (OOSE):

- Desenvolvido por Ivar Jacobson, foi outra metodologia influente com forte ênfase na análise e design orientados a objetos.
- Introduziu conceitos como casos de uso.

Como representar um Software?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Unificação, Desenvolvimento e Evolução

Em 1995, Grady Booch, James Rumbaugh e Ivar Jacobson uniram forças para criar uma linguagem de modelagem padrão, unificando o melhor de suas respectivas metodologias. Eles foram apoiados por outras empresas e especialistas em engenharia de software, formando o "Three Amigos" (Três Amigos).

O objetivo era criar uma linguagem que pudesse abranger todos os aspectos do processo de desenvolvimento de software, desde a concepção até a implementação. A UML foi destinada a ser uma linguagem gráfica para visualizar, especificar, construir e documentar os artefatos de um sistema de software.

O trabalho inicial resultou na primeira versão da UML, conhecida como UML 0.8, em 1996. Ela foi seguida pela UML 0.9 e, finalmente, pela UML 1.0 em 1997, que foi amplamente adotada pela indústria de software. Desde então, a UML passou por várias revisões e atualizações. A mais recente versão 2.5 com uma ampla gama de diagramas e elementos para modelagem de sistemas complexos.

Como representar um Software?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Contribuições e Aceitação

A UML se tornou a linguagem padrão para modelagem de software. Sua aceitação foi impulsionada por uma série de fatores:

Padronização: Tornou-se um padrão da indústria, facilitando a comunicação entre equipes de desenvolvimento e organizações.

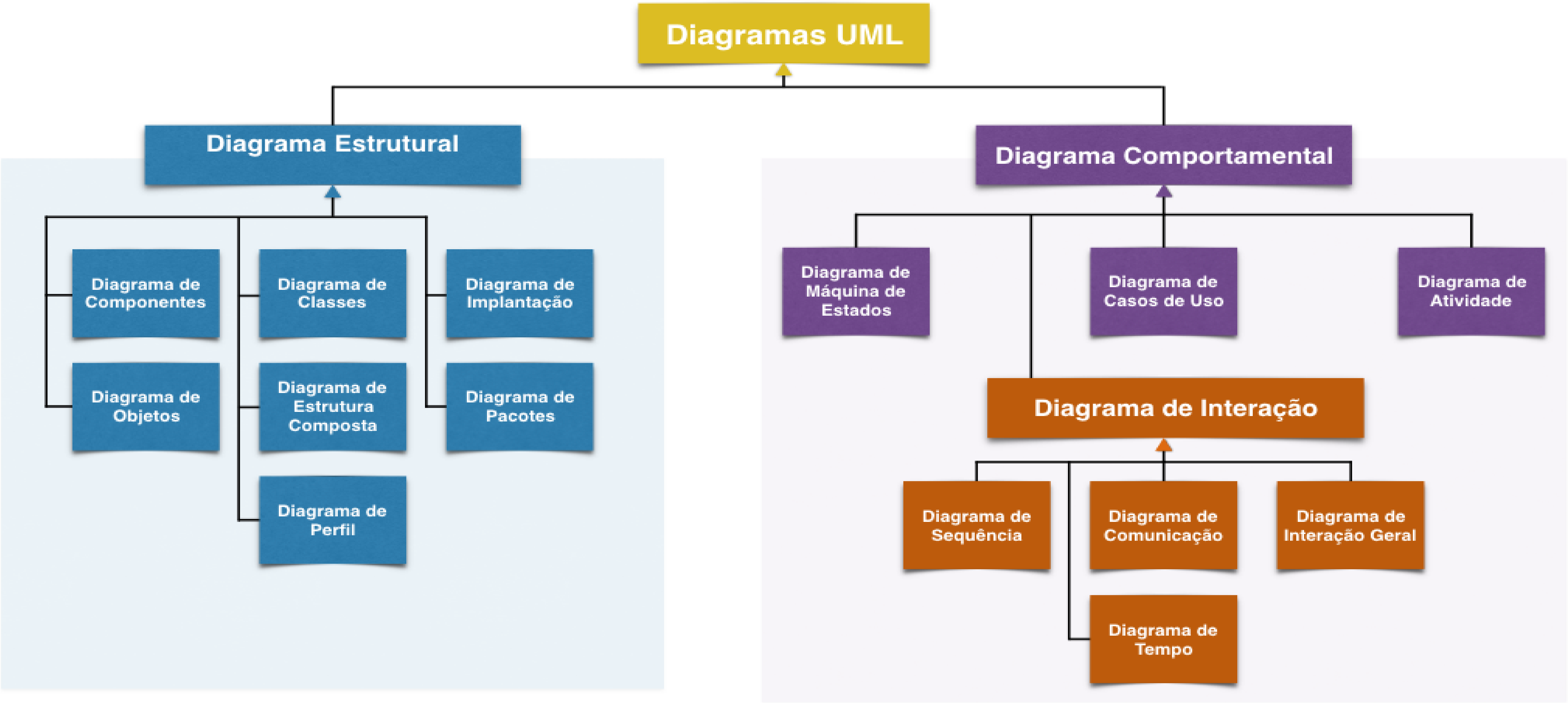
Flexibilidade: Oferece uma variedade de diagramas para atender às diferentes fases e aspectos do desenvolvimento de software.

Ampla Comunidade: Conta com uma comunidade ativa de desenvolvedores, profissionais de engenharia de software e pesquisadores, que contribuem para seu desenvolvimento contínuo.

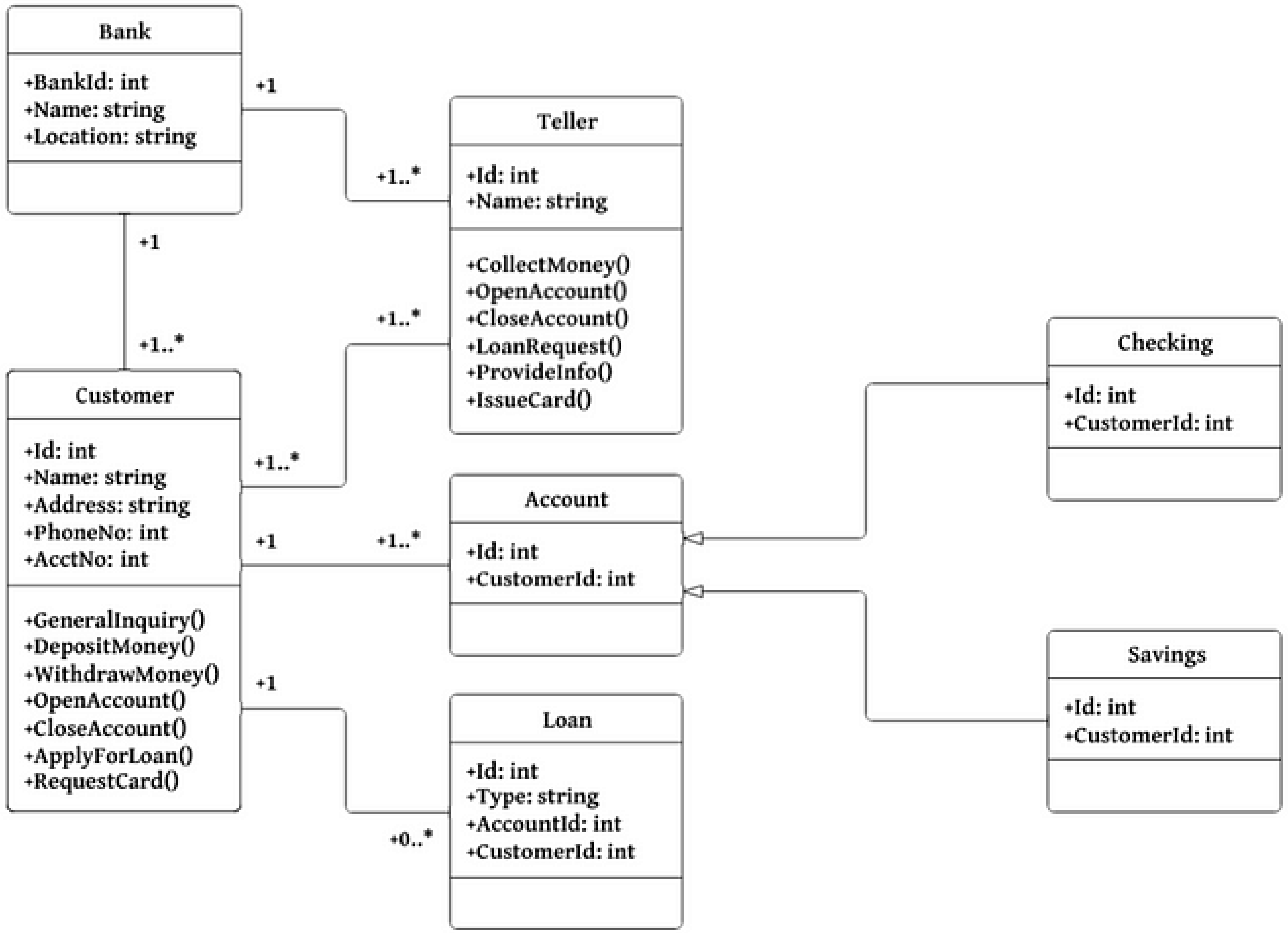
Ferramentas de Suporte: Há uma ampla variedade de ferramentas de modelagem UML disponíveis, tanto comerciais quanto de código aberto.

Diagramas UML

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro



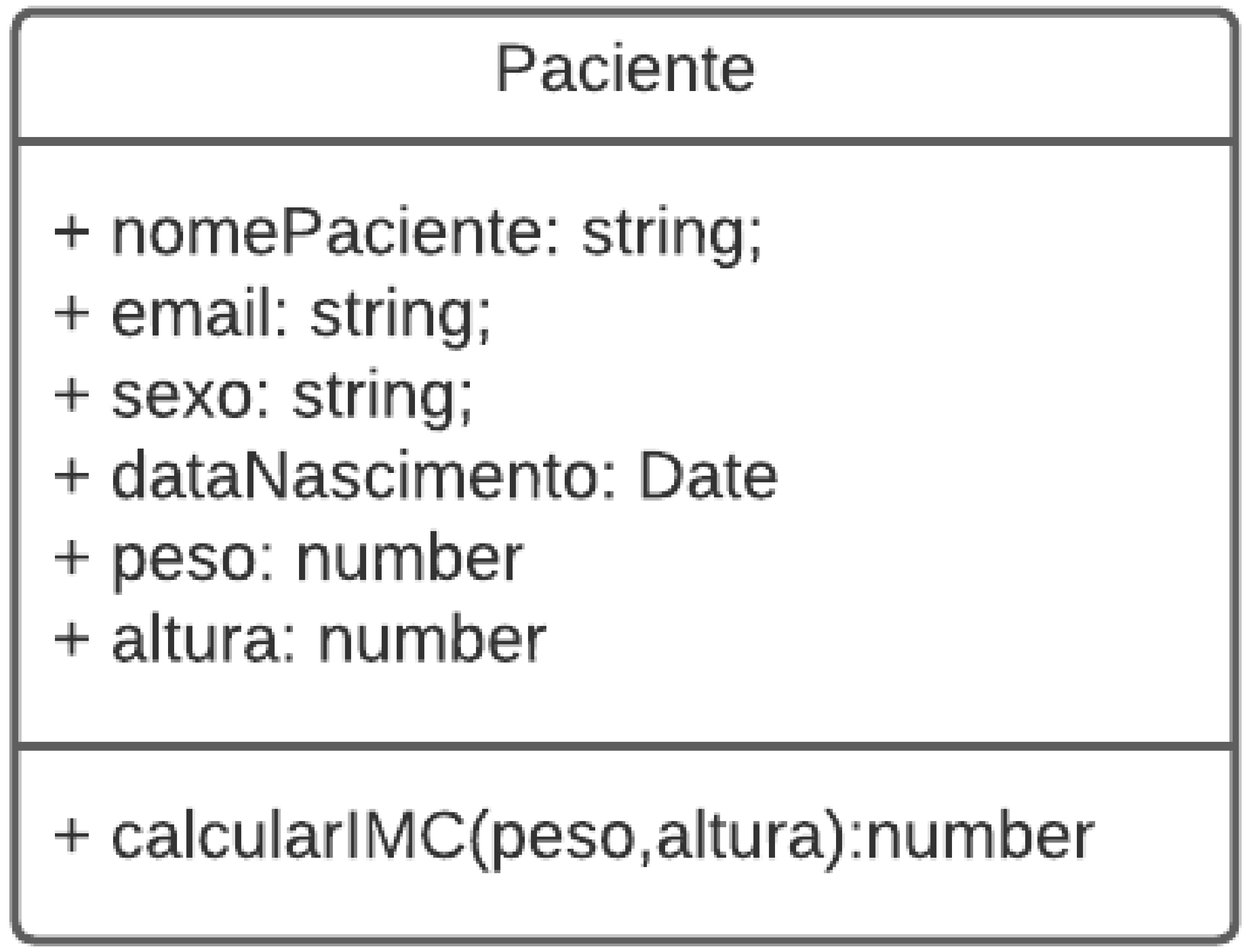
Diagramas de Classe



Representando o 1º Objeto

- Representando um objeto a partir de um Diagrama de Classe

- Vamos utilizar a ferramenta LucidChart ou Diagrams (Draw.io)

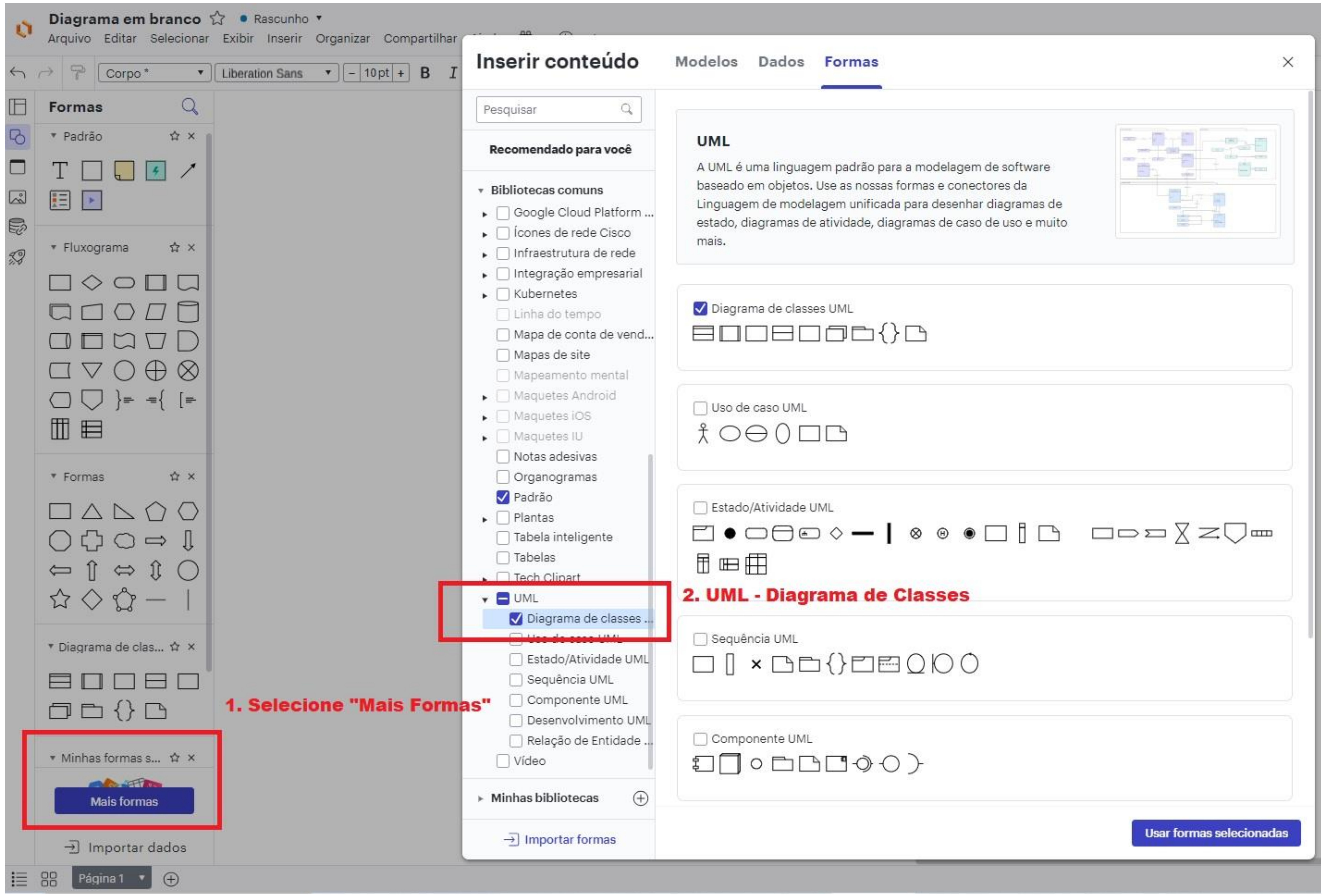


Representando o 1º Objeto

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Acesse o link do aplicativo LucidChart disponível no site do professor

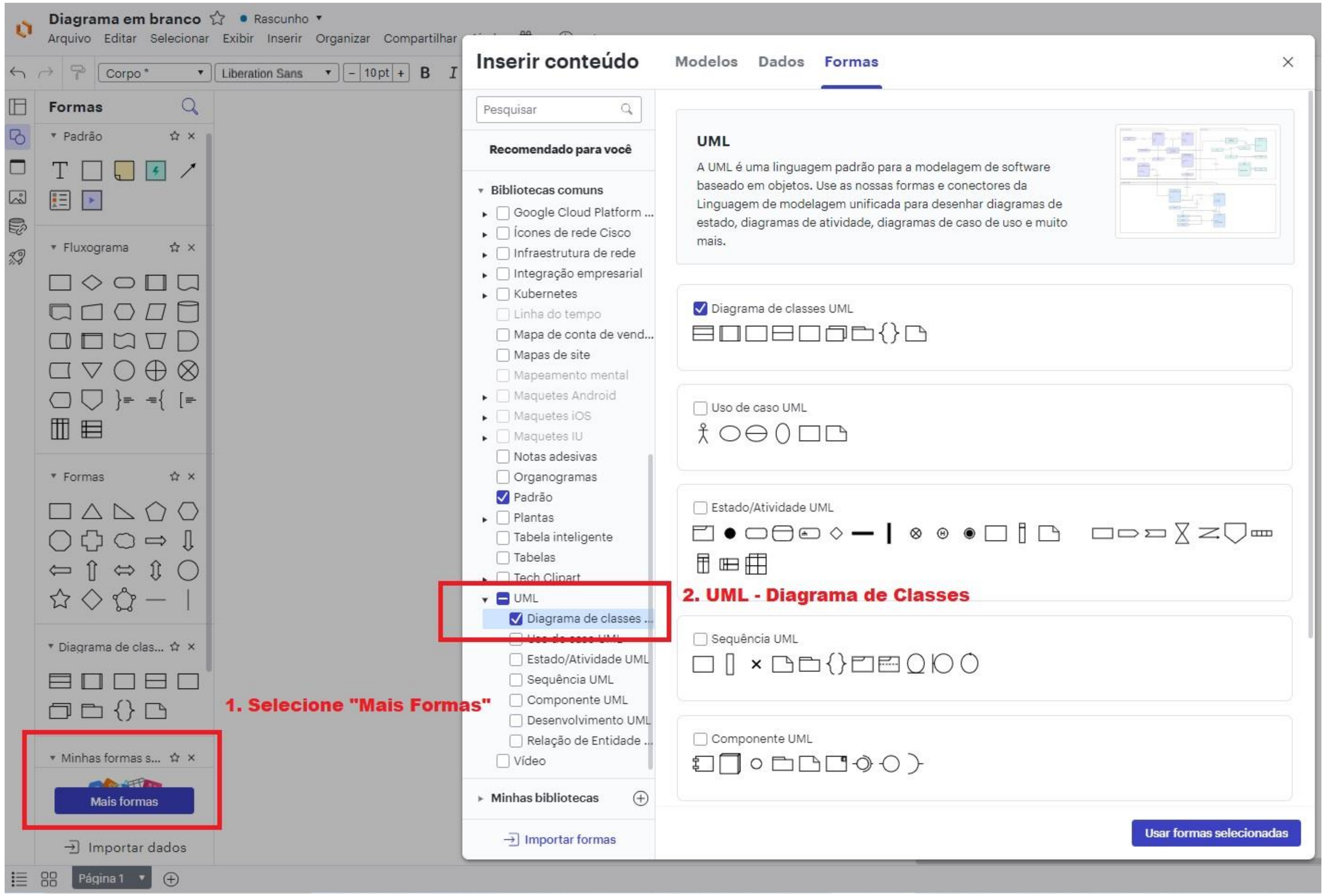
- Inclua o UML – Diagrama de Classes



Representando o 1º Objeto

- Acesse o link do aplicativo LucidChart disponível no site do professor

- Inclua o UML – Diagrama de Classes



Notações

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Existem várias notações para nomes de variáveis, métodos, classes, e outros elementos em programação. Cada uma dessas notações segue regras específicas sobre como as palavras devem ser formatadas e combinadas.

- Benefícios das Notações:

  - Legibilidade: Notações bem escolhidas facilitam a leitura e compreensão do código.
  - Convenção: Seguir uma convenção de nomenclatura ajuda na consistência do código e na colaboração em equipe.
  - Compatibilidade: Algumas linguagens ou ambientes têm convenções de nomenclatura específicas que devem ser seguidas para compatibilidade e boas práticas.

- Em resumo, a escolha da notação de nomenclatura adequada depende das convenções da linguagem de programação, ambiente de desenvolvimento e práticas recomendadas. O importante é manter consistência dentro do projeto ou equipe para garantir um código claro e fácil de entende.

- Acesso a página do professor para conhecer as principais notações.

Exemplo Prático – SI Biblioteca

Antônia é a nova estagiária de Informática da Secretária de Educação de um pequeno município do interior de Minas Gerais. Infelizmente o município não tem orçamento para comprar uma solução de mercado para gestão da Biblioteca, sendo assim o prefeito quer que seja feito um sistema para automatizar as atividades rotineiras da única biblioteca da cidade.

O prefeito comprou vários livros, contudo a conduta rigorosa da bibliotecária Sra. Clotilde tem afastado os leitores da biblioteca. O prefeito quer que o sistema permita qualquer pessoa da cidade pesquise os livros disponíveis pela internet podendo ser pesquisado pelo Nome do Autor, Título ou Assunto. Ele também quer que o sistema agilize o empréstimo e a verificação de quem está com o empréstimo atrasado.

Diante desta demanda, o Secretário de Educação solicitou que a nova estagiária Antônia fosse na biblioteca da cidade para conversar com a bibliotecária Sra. Clotilde para levantar os requisitos do novo sistema.

Exemplo Prático – SI Biblioteca

Assim que a estagiária chegou na biblioteca, a Sra. Clotilde a interpelou com os seguintes dizeres:

"Estou cansada dessa gentalha que vem aqui coloca as mãos sujas nos meus livros fora as pessoas que os devolvem com rabiscos. Aqui nessa biblioteca só eu posso pegar nos livros antes do empréstimo. Senão eles pegam, folheiam, sujam e coloca na estante errada mesmo eu pedindo para eles colocarem no carrinho perto do balcão.

Você é a menina que veio a pedido do prefeito, certo? Bom, acho totalmente desnecessário um sistema já que eu dou conta. Mas vou explicar as regras daqui. Para pegar livro emprestado tem que ser aluno matriculado nas escolas da cidade ou morador da cidade. Podem ficar com até três livros emprestados e o tempo do empréstimo é de 15 (quinze) dias. Só pode renovar o empréstimo uma única vez. Antes de pedir empréstimo, o estudante ou morador deve estar cadastrado aqui. Para o cadastro é necessário a carteirinha de estudante ou comprovante de residência. Só estudante com matrícula ativa na escola pode pegar livro."

Exemplo Prático – SI Biblioteca

A estagiária Antônia anotou tudo que a Sra. Clotilde falou e no fim ela perguntou:

- Você poderia descrever em detalhes as principais atividades da biblioteca?

A Sra. Clotilde respondeu:

- Ora, você nunca esteve numa biblioteca antes? Inicialmente, o leitor pesquisa o livro em uma das 3 (três) gavetas de ficha (Autor, Título e Assunto). Ele anota os dados da ficha do livro e traz aqui no balcão. Eu verifico se o livro está emprestado. Caso o livro esteja aqui, eu peço a identificação do leitor e verifico se o cadastro dele está ok. Se tiver tudo certo, eu pego livro na estante e carimbo a data de devolução na ficha na contracapa. Falo bem alto que só pode renovar por uma vez e se devolver com atraso paga uma multa de R$ 0,50 por dia. Pouca gente paga a multa, mas se tiver devendo não pega mais empréstimo. Eu verifico todos os dias os livros que estão atrasados e ligo para as pessoas cobrando a devolução. Se não tiver sido renovado ainda, eu dou a opção de renovar por uma única vez.

SI Biblioteca – Atividade Prática

A partir da entrevista com a bibliotecária e utilizando o poder da abstração, faça as seguintes atividades:

1) Identifique as entidades envolvidas (atores e objetos) no contexto (domínio) da SI Biblioteca;

2) Identifique os atributos (campos) das entidades do item 1;

3) Identifique as ações (transações) do SI Biblioteca.

SI Biblioteca – Atividade Prática

Como identificar esses elementos (entidades)?

- Alguns são bem óbvios (Leitor e Livro), mas outros estão implícitos que requerem maior maturidade daquele que está fazendo o levantamento de requisitos.

- Após enumerar as entidades e ações óbvias, pode adotar seguinte estratégia:

  1. Listar as etapas do fluxo principal do sistema;
  2. Passo 1 – Isole os substantivos;
  3. Passo 2 – Analise os substantivos;
  4. Passo 3 – Isole e analise os verbos;
  5. Passo 4 – Para cada conceito verifique se ele é composto com outras partes do sistema.

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

1. O **Leitor** chega ao **balcão** de atendimento da **biblioteca** e diz ao **atendente** que deseja emprestar um ou mais **livros** da **biblioteca**.
2. O **Atendente** seleciona a **opção** para adicionar um novo **empréstimo**.
3. O **Atendente** solicita ao **leitor** sua **identidade**, seja de **estudante** ou **morador**.
4. O **Atendente** informa ao **sistema** a **identificação do leitor**.
5. O **Sistema** exibe o **nome do leitor** e sua situação.
6. O **Atendente** solicita os **livros** a serem emprestados.
7. Para cada um deles, informa ao **sistema** o **código de identificação do livro**.
8. O **Sistema** informa a **data de devolução** de cada **livro**.

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

PASSO 1 – Isolar os substantivos

- Leitor
- Balcão
- Biblioteca
- Atendente
- Livros
- Opção
- Empréstimo
- Carteirinha
- Estudante
- Morador
- Sistema
- Identificação de leitor
- Nome do leitor
- Código de identificação do livro
- Data de devolução

SI Biblioteca – Atividade Prática

Passo 2: Análise dos substantivos

Para cada substantivo, verifique se é relacionado a assuntos importantes no domínio do sistema.

Descarte aqueles que:

- fogem do escopo do sistema
- são similares a outros conceitos já identificados
- são propriedades de outros substantivos

Lembre-se:

Conceitos relevantes são aqueles que se referem a entidades que têm que ser lembradas pelo sistema: fazem algo, sabem algo, conhecem algo, ...

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

PASSO 2 – Analise os substantivos

- Leitor
- ~~Balcão~~
- Biblioteca
- Atendente
- Livros
- ~~Opção~~
- Empréstimo
- ~~Carteirinha~~
- Estudante
- Morador
- ~~Sistema~~
- ~~Identificação de leitor~~
- ~~Nome do leitor~~
- ~~Código de identificação do livro~~
- ~~Data de devolução~~

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Passo 3: Isole e analise os verbos

Isole os verbos que poderiam ser transformados em substantivos (possivelmente com a ajuda de outras palavras).

Concentre-se nos verbos que representam ações de interesse para o sistema, ou seja, aqueles relacionados a eventos e transações que possuem informações importantes e que devem ser lembradas pelo sistema.

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

1. O Leitor **chega** ao balcão de atendimento da biblioteca e **diz** ao atendente que deseja **emprestar** um ou mais livros da biblioteca.
2. O Atendente **seleciona** a opção para **adicionar** um novo empréstimo.
3. O Atendente **solicita** ao leitor sua identidade, seja de estudante ou morador.
4. O Atendente **informa** ao sistema a identificação do leitor.
5. O Sistema **exibe** o nome do leitor e sua situação.
6. O Atendente **solicita** os livros a serem emprestados.
7. Para cada um deles, **informa** ao sistema o código de identificação do livro.
8. O Sistema **informa** a data de devolução de cada livro.

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

PASSO 3 – Isole e analise os verbos

- ~~Emprestar~~
- ~~Adicionar~~
- ~~Informar~~

SI Biblioteca – Atividade Prática

Passo 4: Para cada conceito verifique se ele é composto com outras partes do sistema

Para cada candidato a conceito, verifique se ele é composto de outras partes que sejam de interesse do sistema, mesmo que essas não apareçam explicitamente no texto.

Exemplo:

- Empréstimo normalmente refere-se a vários livros emprestados em uma mesma ocasião por um mesmo leitor.
- Linha do Empréstimo refere-se a cada livro emprestado.

(Obs. poderia ser também: Item do Empréstimo)

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Outras possíveis entidades:

- Objetos físicos ou tangíveis:
  - Livro, Leitor
- Especificação de Projetos ou descrição de coisas:
  - EspecificacaoDeLivro, CategoriaDeLivro
- Lugares:
  - Biblioteca, SalaDeAula
- Transações:
  - Emprestimo, Reserva
- Linha de Itens de Transações:
  - LinhaDoEmprestimo

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Outras possíveis entidades:

- Papéis desempenhados por pessoas:
  - Atendente, ChefeDeBiblioteca, Usuario
- Contêineres de outras coisas:
  - Estante, Armario, Sala
- Coisas em um contêiner:
  - CopiaDeLivro, Revista
- Catálogos:
  - CatalogoDeLivros, CatalogoDeRevistas
- etc, etc..

SI Biblioteca – Atividade Prática

Como identificar atributos?

Substantivos podem ser candidatos a atributos de conceitos.

Cautela:

- não torne o modelo conceitual muito complexo desnecessariamente.
- limite-se a adicionar
  - atributos importantes para compreender o conceito
  - atributos que serão importantes para o futuro projeto do sistema

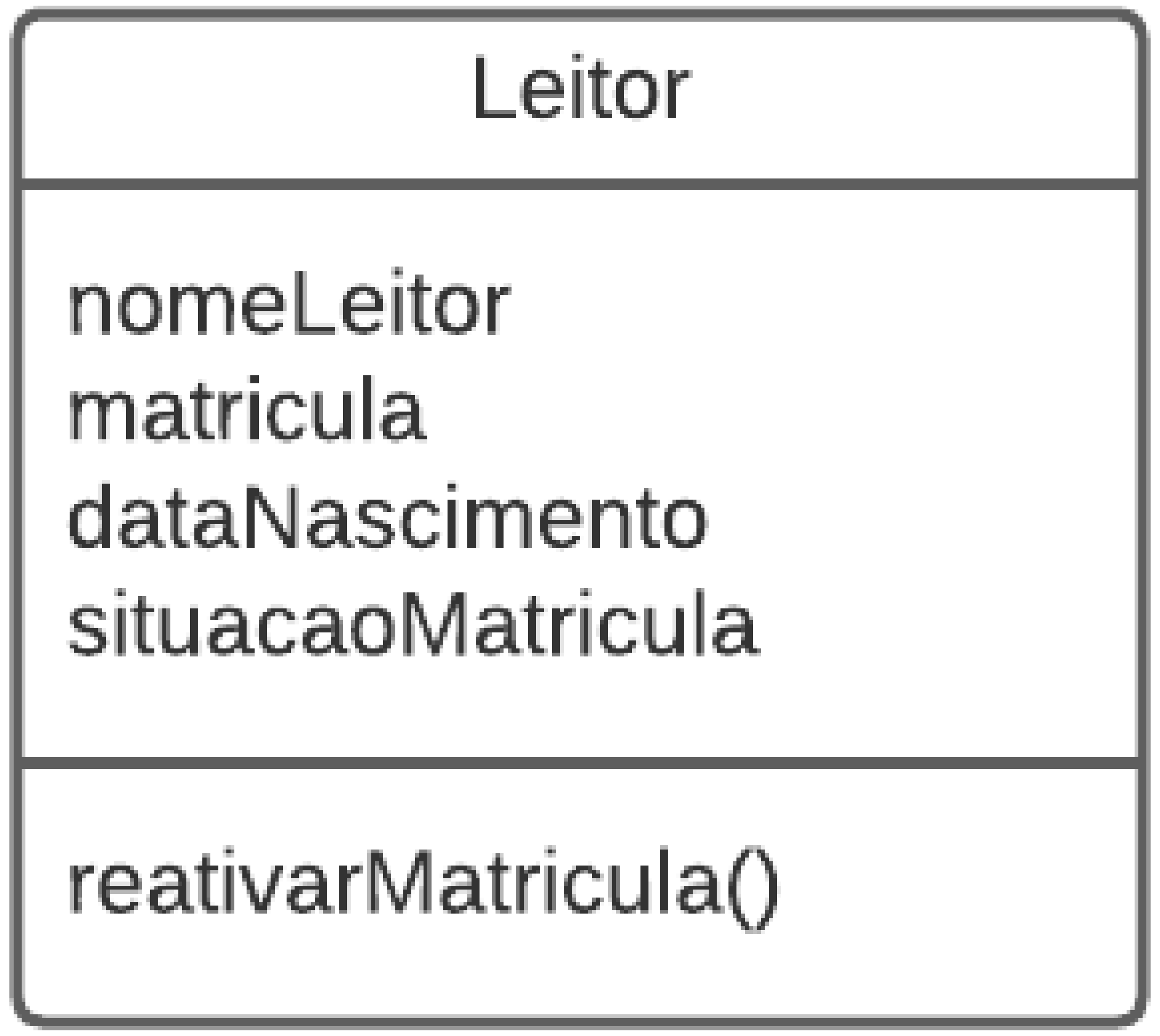
Classes e Objetos

Modelo Conceitual x Diagrama de Classes

Classe

- É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

- É representadas por um retângulo que pode possuir até três divisões:
  - Nome da classe
  - Atributos da classe
  - Métodos da classe



Classes e Objetos

Qual é a diferença entre Classe e Objeto

Classe
- é definição do tipo;
- representa um conjunto de objetos de mesmo tipo;
- Classe = {obj1, obj2, obj3, ..., objN}

Objeto
- cada instância derivada da classe;
- é um elemento do conjunto representado pela classe

Leitor
nomeLeitor matricula dataNascimento situacaoMatricula
reativarMatricula()

nomeLeitor: Maria
matrícula:20230001
dataNascimento: 15/10/2017
Situacao:Ativo

nomeLeitor: João
matrícula:20230002
dataNascimento: 10/08/2017
Situacao:Ativo

nomeLeitor: Lia
matrícula:20230003
dataNascimento: 20/12/2017
Situacao:Ativo

SI Biblioteca – Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

1. Fazer cadastro no site do LucidChart;
2. Criar um novo diagrama utilizado a biblioteca de formas "UML – Diagrama de Classes";
3. Criar as classes do SI Biblioteca (cada entidade vai virar uma classe);
4. Criar os atributos identificados de cada classe;
5. Utilizar a convenção de nomenclatura para Javascript (veja este artigo)

Dúvidas?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Dúvidas?

fernando.alves@ifrj.edu.br

ftamberlini.dev.br