



DIAGRAMA DE CLASSES

SUMÁRIO

Apresentação	3
Relacionamento entre classes	6
Ferramentas de criação de diagramas de classe	8
UML – Linguagem de Modelagem Unificada	11
Diagrama de Classes Melhores práticas.....	14
Como desenhar um Diagrama de Classes.....	15
Referências.....	22



APRESENTAÇÃO

Diagrama de classes, no paradigma de programação orientada a objetos, representa a estrutura e relações das classes utilizadas como base na criação de objetos. No desenvolvimento de softwares, essa modelagem é muito utilizada, pois define todas as classes que o sistema deve possuir, independentemente da linguagem de programação dele.

Por sua vez, um objeto representa uma entidade (classe), contendo seus atributos, operações, comportamentos e relação entre objetos.

Na Linguagem de Modelagem Unificada (UML), o diagrama de classes representa um tipo de diagrama da estrutura do sistema (software) que será modelado, pois descreve o que deve estar presente nesse sistema e a ser modelado.



A representação do diagrama de classes é composta por três partes:

- • • • •
- 1. **Nome da classe:** a classe é um elemento que contém seus atributos e métodos que serão representados em um objeto;
- 2. **Atributos:** define as características da classe, por exemplo: a classe Aluno possui os atributos nome, CPF, RG, endereço, eMail, dataNascimento. Quando criamos um objeto da classe Aluno, esses atributos estarão disponíveis para ser acessados e prontos para receber valores, de acordo com o tipo do atributo. É importante ressaltar que, para cada objeto da classe, os valores dos atributos são únicos. Na UML, sugere-se que a nomenclatura do atributo esteja em conformidade com a definição camelCase;
- 3. **Métodos:** são ações que definem uma determinada operação em uma classe. Não há nenhuma regra para o número máximo de métodos na classe, e o mesmo nome de método pode ser utilizado por classes distintas.
- • • • •

Para saber mais sobre camelCase, clique no link a seguir ou acesse através do código QR:

<https://techlib.wiki/definition/camelcase.html>



A imagem a seguir apresenta um exemplo de representação de uma classe em um diagrama de classes:

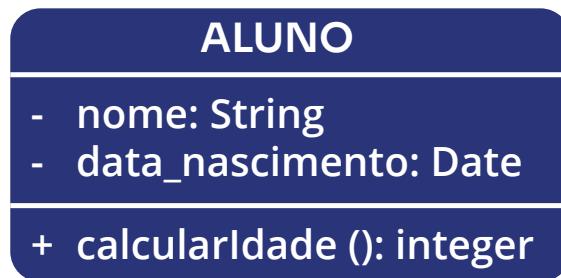


Figura 1 - Representação de uma classe em um diagrama de classes

Fonte: Do autor (2022)

Os diagramas de classes ilustram os atributos e métodos de uma classe, demonstrando a visibilidade que esses elementos possuem.

Para representar a visibilidade dos atributos e métodos, eles estão definidos em:

TIPO	SÍMBOLO	DESCRIÇÃO
público	+	visível em qualquer classe
protegido	#	qualquer descendente pode usar
privado	-	visível somente dentro da classe
pacote	~	visível para todas as classes do mesmo pacote

Quadro 1 - Representação da visibilidade dos atributos e métodos

Fonte: Do autor (2022)

Modelo para a criação de diagrama de classes:

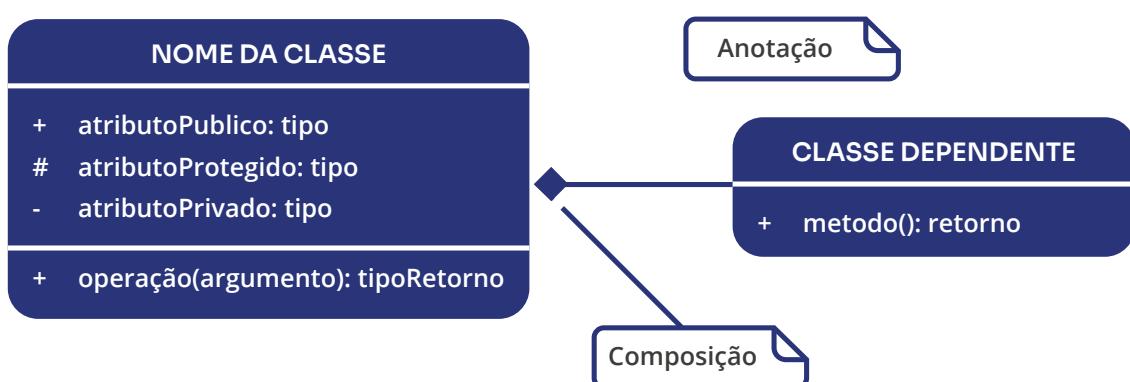


Figura 2 - Modelo para a criação de diagrama de classes

Fonte: Do autor (2022)

Exemplo de diagrama de classes do sistema Moradia:

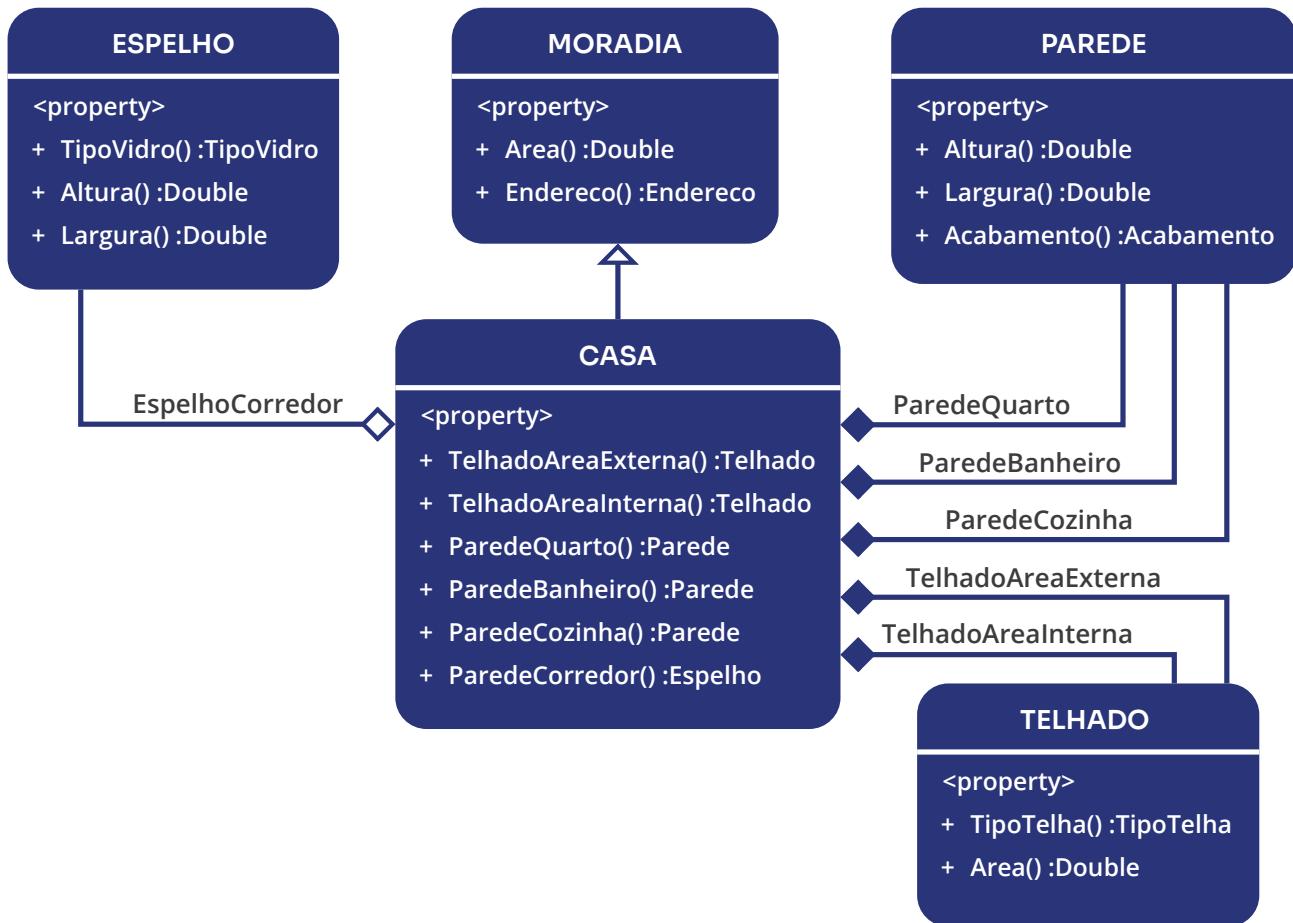


Figura 3 - Diagrama de classes do sistema Moradia

Fonte: Do autor (2022)

RELACIONAMENTO ENTRE CLASSES

Quando criamos instâncias de objetos de classes, elas representam os comportamentos, propriedades e métodos que estão definidos nas classes.

Dessa forma, é possível estabelecer relacionamento entre classes, permitindo que objetos de classes distintas possam interagir entre si.

A Linguagem de Modelagem Unificada (UML) reconhece três tipos de relacionamentos mais importantes: dependência, associação e generalização.

O exemplo de diagrama de classes do sistema Moradia demonstra relacionamentos estabelecidos entre classes:

- › A classe Espelho **é parte** da classe Casa;
- › A classe Telhado **é composição** da classe Casa;
- › A classe Parede **é composição** da classe Casa;
- › A classe Casa **possui todas as características** que a classe Moradia possui, além das suas específicas.

Vamos entender detalhadamente quais são os relacionamentos e as suas representações:

TIPO	REPRESENTAÇÃO	DESCRIÇÃO
Associação	Associação	É um tipo de relacionamento utilizado entre classes. É uma aplicação entre classes que não possuem dependência uma das outras; mas, durante o ciclo de vida do software, poderá ocorrer algum relacionamento conceitual.
Generalização	Herança →	É um tipo de relacionamento chamado de herança, em que a classe generalizada fornece recursos para a classe especializada (herdeira). Nesse relacionamento, tudo o que a classe generalizada possui, a classe herdada terá.
Dependência	Dependência →	É um tipo de relacionamento em que a mudança na especificação de uma classe pode alterar a especificação da classe que possui dependência direta dela.
Agregação	Agregação ◊	É um tipo de relacionamento em que a classe agregada usa outra classe para “existir”, porém, pode viver sem ela.
Composição	Composição ◊	É um tipo de relacionamento em que a classe composta possui dependência de outras classes para poder “existir”.

Quadro 2 - Relacionamentos e suas representações

Fonte: Do autor (2022)

Agora que já aprendemos quais os tipos de relacionamentos entre classes, vamos especificar os tipos existentes no diagrama de classes Casa:

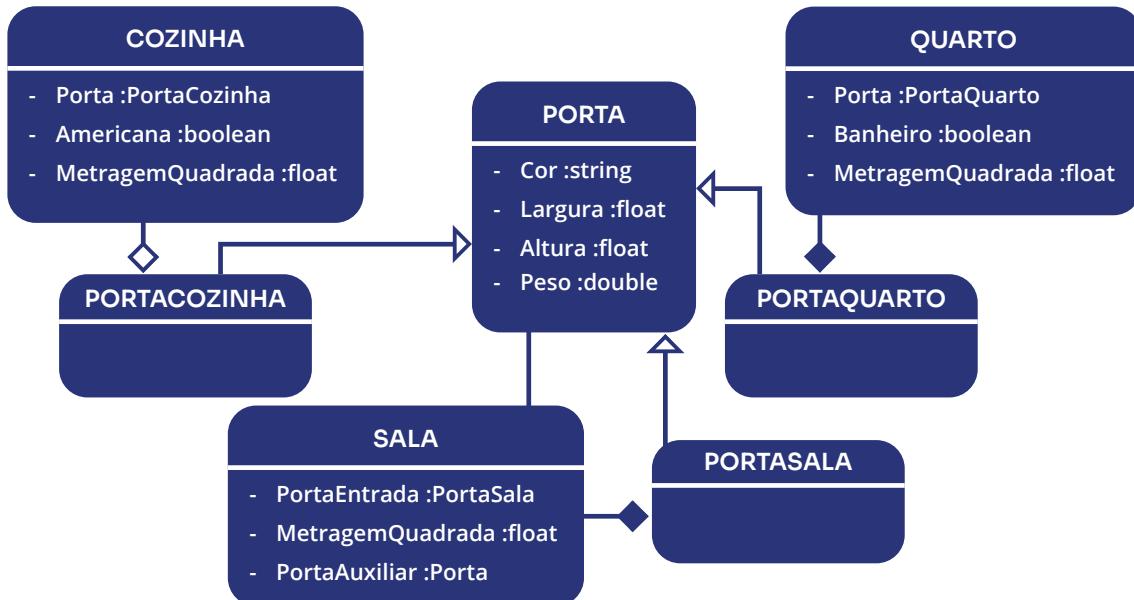


Figura 4 - Diagrama de classes do sistema Casa

Fonte: Do autor (2022)

A partir dos elementos da imagem anterior, podemos entender que:

- › Objeto Cozinha pode ter ou não um objeto PortaCozinha – (**Agregação**);
- › Objeto PortaCozinha generaliza o objeto Porta, ou seja, PortaCozinha possui todas as características de Porta, além das suas específicas – (**Generalização**);
- › Objeto Sala deve ter objeto PortaSala, não podendo existir se não o tiver – (**Composição**);
- › Objeto PortaSala generaliza Porta – (**Generalização**);
- › Objeto PortaQuarto generaliza o objeto Porta – (**Generalização**);
- › Objeto Sala pode existir sem o objeto Porta – (**Associação**);
- › Objeto Quarto deve ter o objeto PortaQuarto, não podendo existir se não o tiver – (**Composição**).

Ferramentas de criação de diagramas de classe

A seguir, são apresentados alguns exemplos de ferramentas muito utilizadas na criação de diagramas de classes:

1. **Edraw Max:** ferramenta online que oferece uma versão gratuita para a criação de diagramas UML;
2. **Lucidchart:** uma das principais plataformas para a criação de diagramas de diversos tipos, inclusive UML. Algumas funcionalidades estão disponíveis somente para assinantes;
3. **Draw.io:** ferramenta totalmente gratuita, com várias opções de diagramas, inclusive UML;
4. **Gliffy:** ferramenta de uso apenas para assinantes;
5. **yUML:** ferramenta que possibilita a criação de diagramas UML, de forma totalmente diferente;
6. **Createley:** ferramenta gratuita e completa para a criação de diagramas UML;
7. **Cacoo:** ferramenta de uso apenas para assinantes.

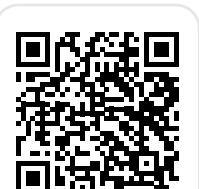
Para acessar a ferramenta Edraw Max, clique no link a seguir ou acesse através do código QR:

<https://www.edrawmax.com/>



Para acessar a ferramenta Lucidchart, clique no link a seguir ou acesse através do código QR:

<https://www.lucidchart.com/pages/pt/exemplos/uml-online>



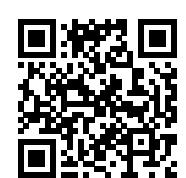
Para acessar a ferramenta Gliffy, clique no link a seguir ou acesse através do código QR:

<https://www.gliffy.com/>



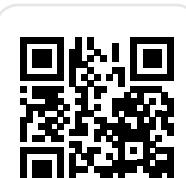
Para acessar a ferramenta Draw.io, clique no link a seguir ou
acesse através do código QR:

<https://app.diagrams.net/>



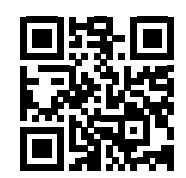
Para acessar a ferramenta yUML, clique no link a seguir ou
acesse através do código QR:

<https://yuml.me/>



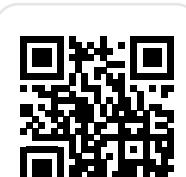
Para acessar a ferramenta Creately, clique no link a seguir ou
acesse através do código QR:

<https://creately.com/>



Para acessar a ferramenta Cacoo, clique no link a seguir ou
acesse através do código QR:

<https://cacoo.com/>



A seguir, você confere um exemplo de diagrama na ferramenta draw.io:

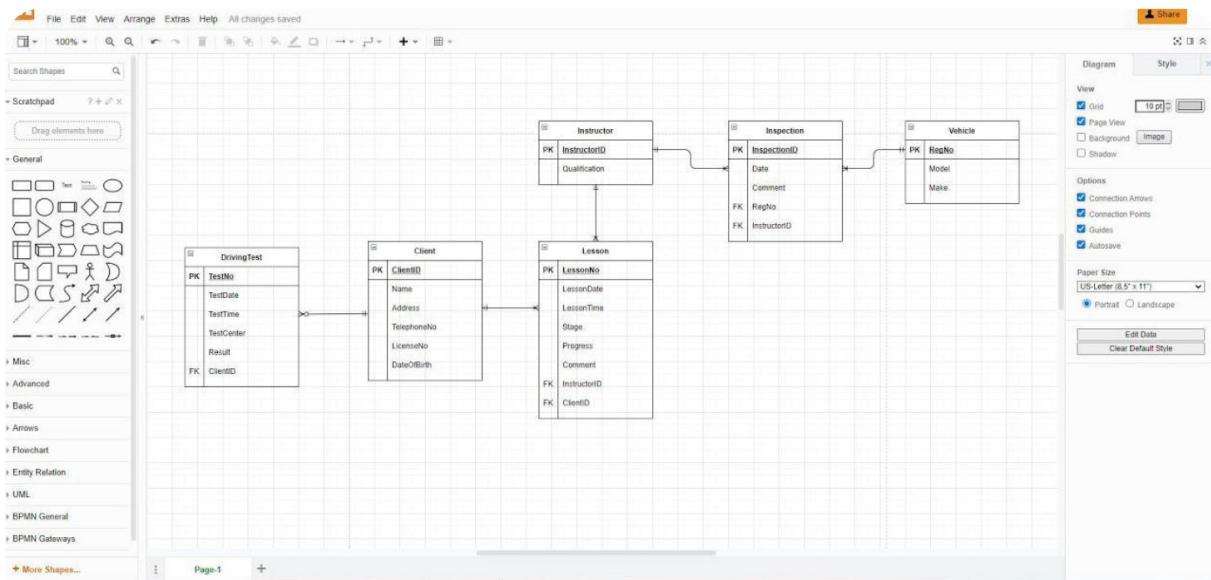


Figura 5 - Exemplo de diagrama criado na ferramenta draw.io

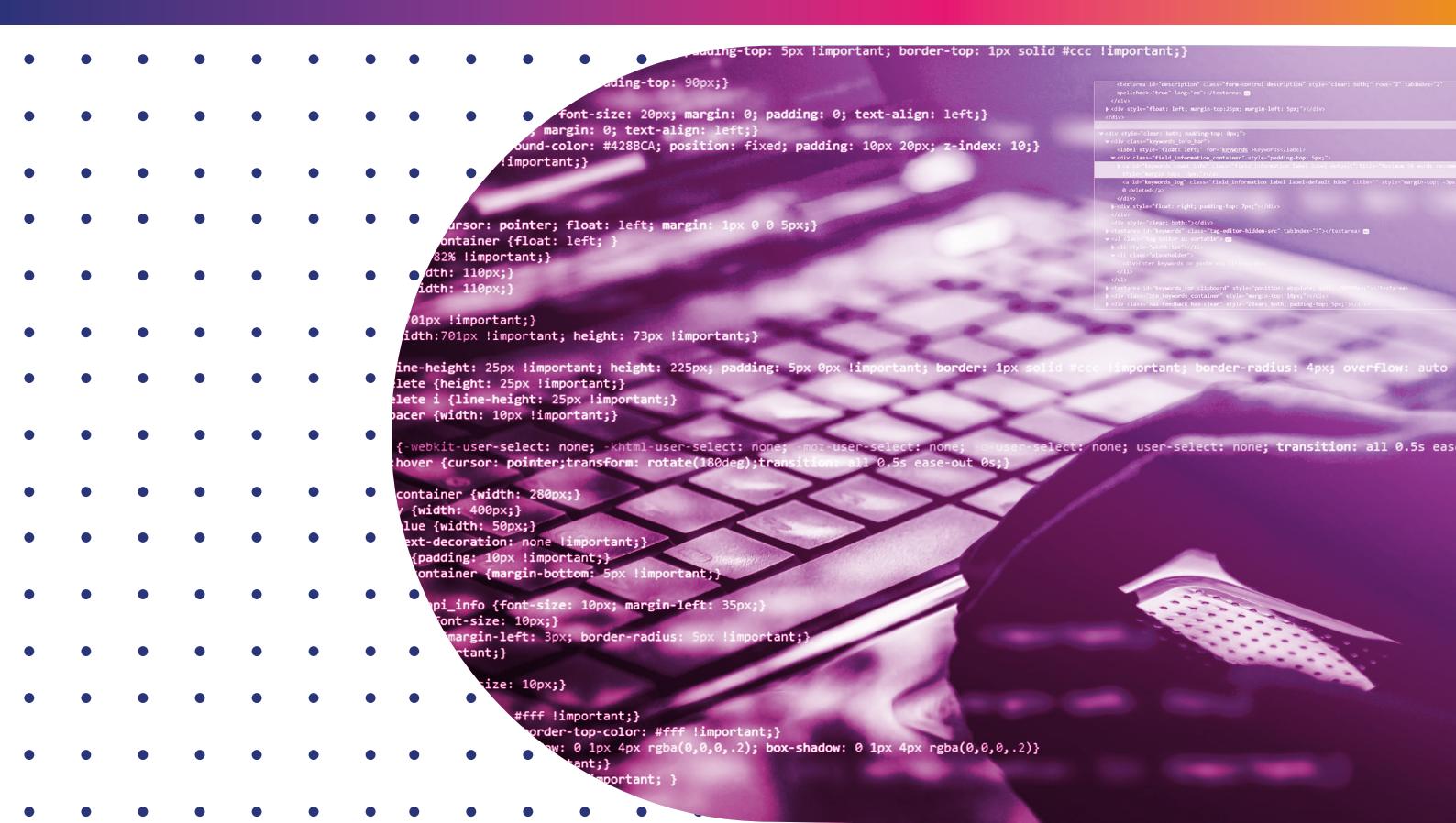
Fonte: Do autor (2022)

UML – LINGUAGEM DE MODELAGEM UNIFICADA

A sigla **UML** (*Unified Modeling Language*) – em português, Linguagem de Modelagem Unificada – é uma linguagem-padrão utilizada para a elaboração da estrutura de projetos de software.

Durante o desenvolvimento de softwares, é importante garantir a qualidade da aplicação e reduzir as chances de problemas durante esse processo. Portanto, a criação de diagramas de classes compõe a fase de documentação e modelagem das especificações do sistema.

A linguagem de notação UML modela e documenta as diversas fases do desenvolvimento de sistemas orientados a objetos, define uma série de elementos gráficos que representam os componentes de uma aplicação, suas interações e mudanças de estado.



A criação de diagramas UML é essencial para evitar erros nas fases de especificação do projeto e comunicação desencontrada entre as partes interessadas na entrega final do software. Com o uso desses diagramas, é possível obter uma visão clara e única do software, deixando todas as entidades envolvidas no projeto no mesmo nível de entendimento do que será desenvolvido e concluído.

Os diagramas UML estão divididos da seguinte forma:

1) Diagrama de estruturas:

- a. Diagrama de classes
- b. Diagrama de componentes
- c. Diagrama de objetos
- d. Diagrama de perfil
- e. Diagrama de estruturas compostas
- f. Diagrama de implantação
- g. Diagrama de pacotes

2) Diagrama de comportamentos:

- a. Diagrama de atividades
- b. Diagrama de casos de uso
- c. Diagrama de máquina de estados
- d. Diagrama de interação
 - i. Diagrama de sequência
 - ii. Diagrama de comunicação
 - iii. Diagrama de visão geral de interação
 - iv. Diagrama de tempo

Exemplo da representação hierárquica de diagramas UML:

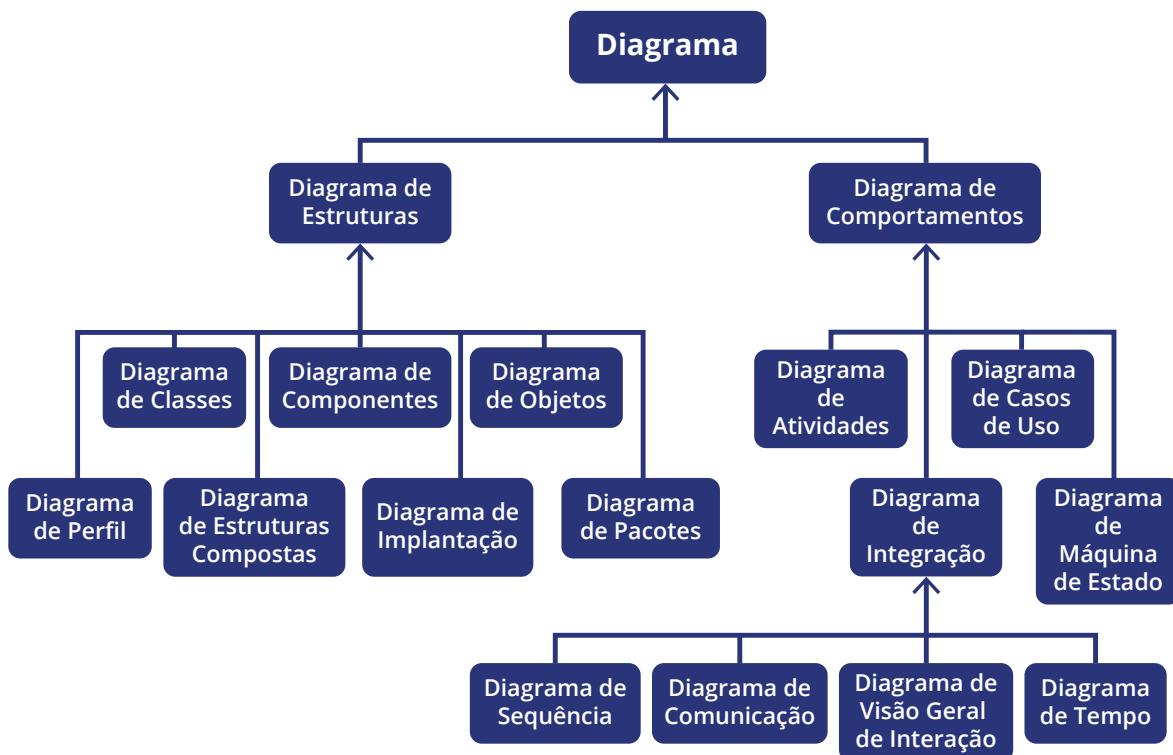


Figura 6 - Representação hierárquica de diagramas UML

Fonte: Adaptado de OMG (2007)

Exemplo de diagrama de classes:

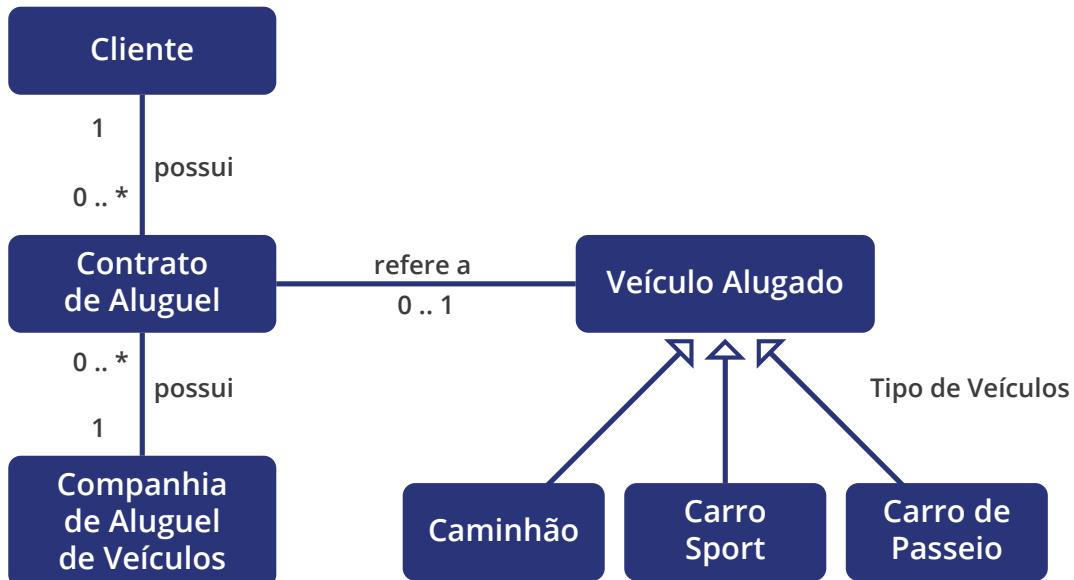


Figura 7 - Diagrama de classes

Fonte: Do autor (2022)

DIAGRAMA DE CLASSES - MELHORES PRÁTICAS

Elaborar, de forma criteriosa, diagramas de classes é um fator de sucesso nos projetos de software, principalmente se neles os problemas do usuário são transformados em aplicações que serão utilizadas como pontes entre requisitos e codificação. A seguir, outras melhores práticas:

• • • • • • •

- Facilidade no mapeamento e visualização do fluxo entre as diversas classes, atributos, operações e relacionamentos entre objetos;
- Utilização de cores para agrupar módulos comuns, ajudando o usuário a diferenciar os diversos grupos existentes no diagrama;
- Melhoria no entendimento da visão geral dos esquemas de uma aplicação.
- Melhor compreensão dos requisitos de domínio do sistema e melhor identificação dos seus componentes.

• • • • • • •

COMO DESENHAR UM DIAGRAMA DE CLASSES

Chegou o momento de colocarmos a mão na massa.

Apresentamos, anteriormente, ferramentas para a criação de diagramas UML e exemplos de diagramas de classes.

Nesse tutorial, vamos utilizar a ferramenta gratuita draw.io e o exemplo do diagrama de classes Moradia:

1. Acesse qualquer browser de internet e digite draw.io.

Serão apresentadas duas opções – criar um diagrama ou abrir um já existente:

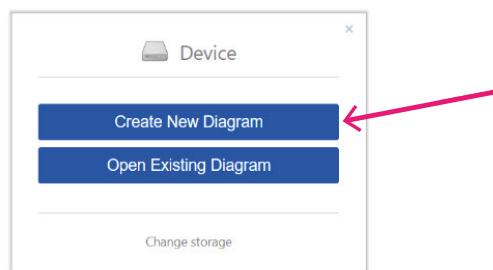


Figura 8 - Criando um novo diagrama

Fonte: Do autor (2022)

2. Na opção “Criar um diagrama”, escolha “**Blank Diagram**”. Digite o nome do diagrama “**moradia.drawio**” e clique no botão “**Create**”:

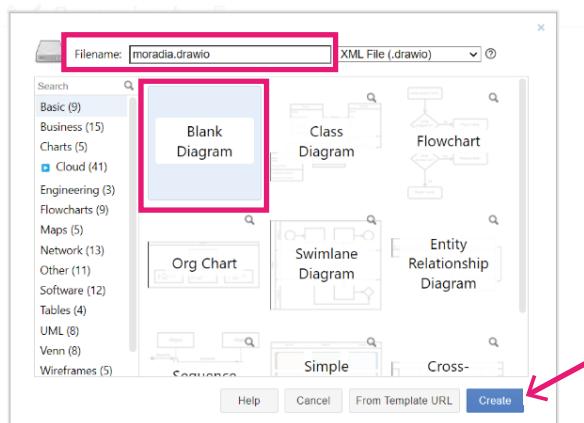


Figura 9 - Criando um diagrama com template em branco

Fonte: Do autor (2022)



3. Na janela “**Salvar Como**”, escolha o caminho físico para salvar o diagrama. Insira o nome “**moradia**” e clique no botão “**Salvar**”:

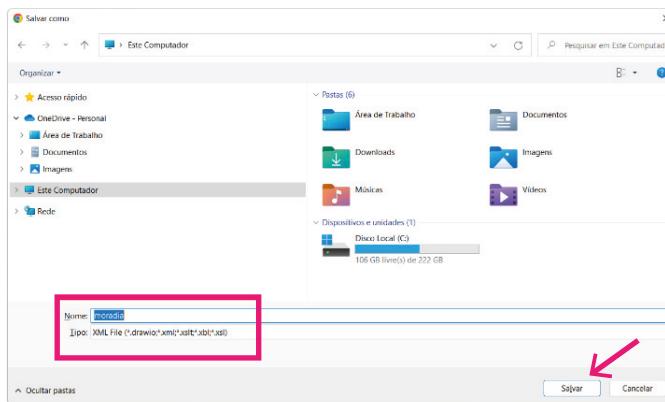


Figura 10 - Salvando o diagrama

Fonte: Do autor (2022)

4. O resultado será uma tela pronta para você criar o diagrama **Moradia**:

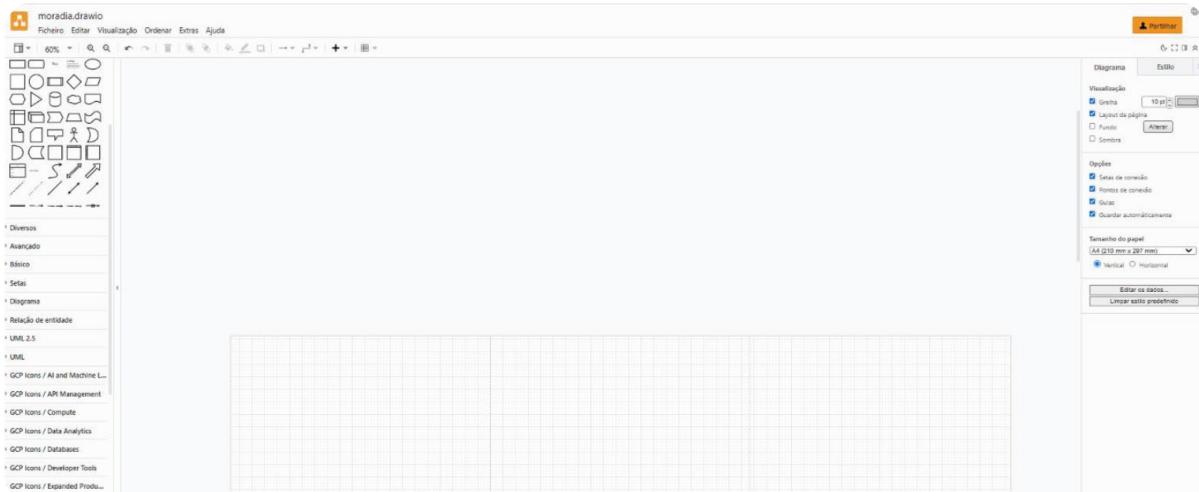


Figura 11 - Ambiente de criação

Fonte: Do autor (2022)

5. No lado esquerdo da tela, clique na opção “UML”:

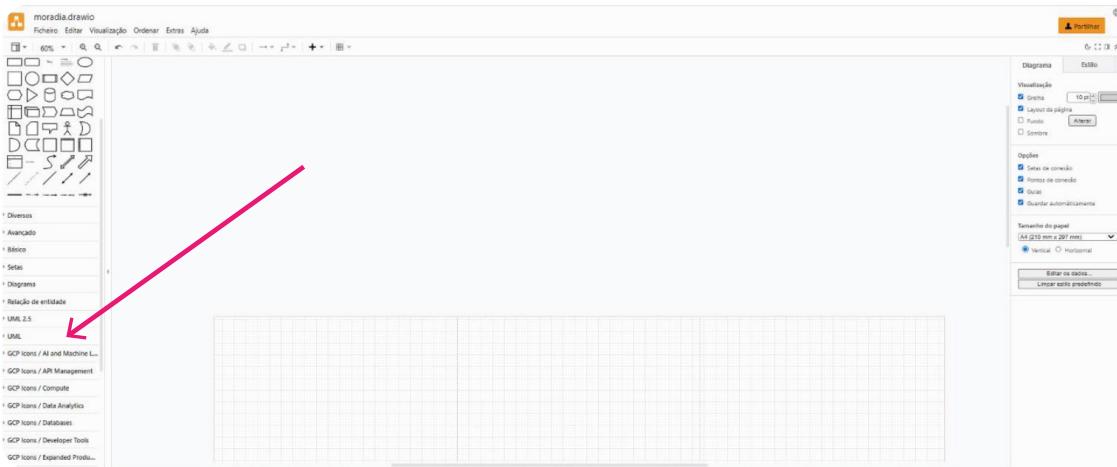


Figura 12 - Opção UML

Fonte: Do autor (2022)

6. Abrir o menu UML e escolher a opção “**Classname**”:

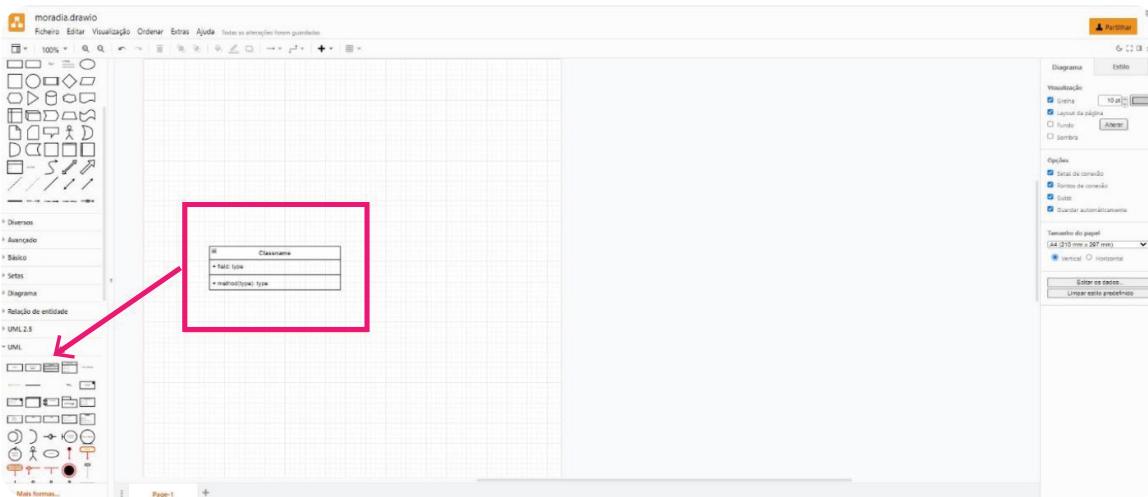


Figura 13 - Opção **Classname**

Fonte: Do autor (2022)

7. Vamos adequar as informações nome da classe, atributos e métodos para os valores que se encontram no diagrama **Moradia**:

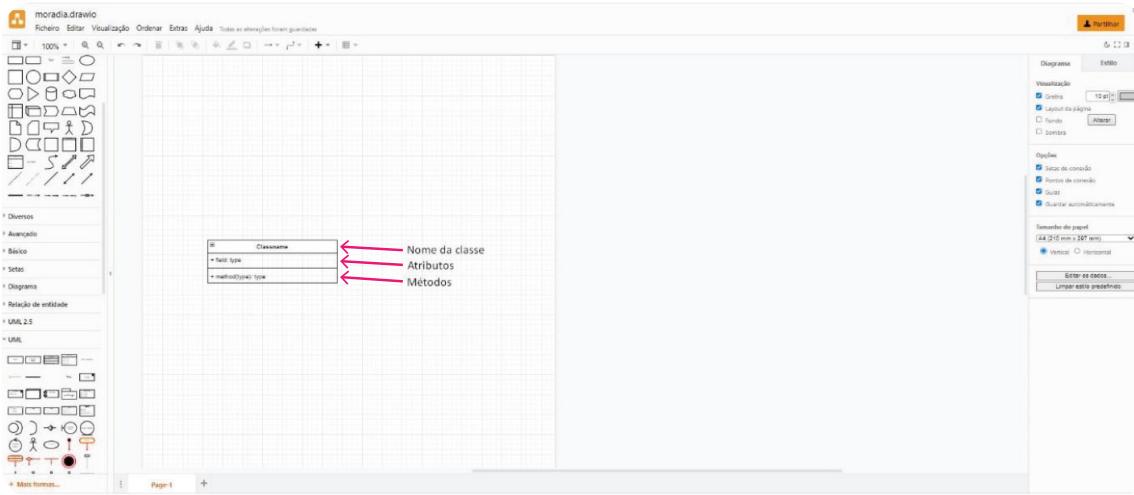


Figura 14 - Classes do diagrama Moradia

Fonte: Do autor (2022)

8. Criamos todas as classes do diagrama Moradia, agora vamos estabelecer os relacionamentos entre elas. Para isso, utilize as opções que se encontram no menu “UML”:

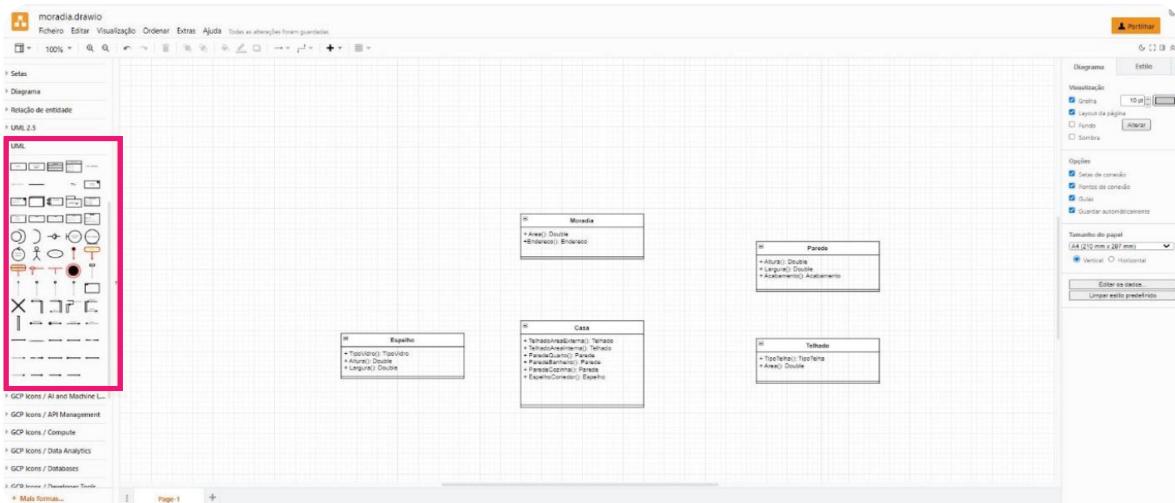


Figura 15 - Opções que se encontram no menu UML

Fonte: Do autor (2022)

9. Estas são as opções de relacionamentos para esse diagrama:

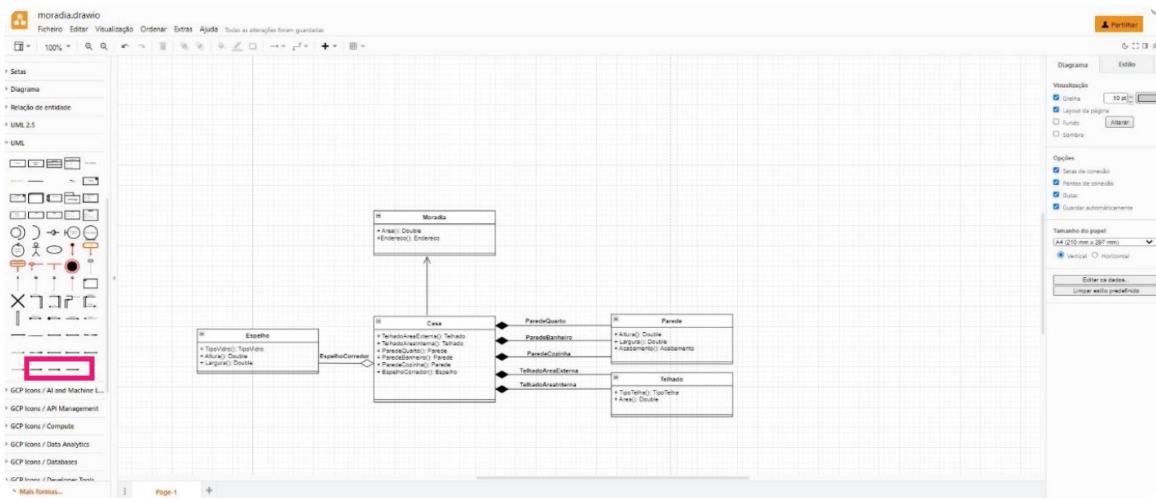


Figura 16 - Opções de relacionamento do diagrama de classes

Fonte: Do autor (2022)

10. Este deverá ser o resultado:

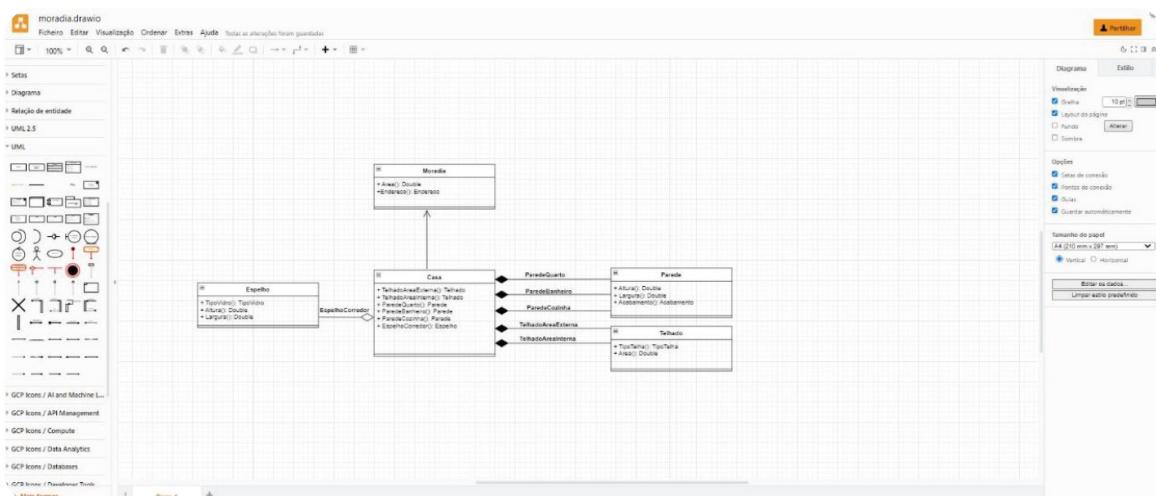


Figura 17 - Resultado da criação do diagrama

Fonte: Do autor (2022)

11. Para salvar o diagrama, no menu superior, escolha a opção “File” ou “Ficheiro”:

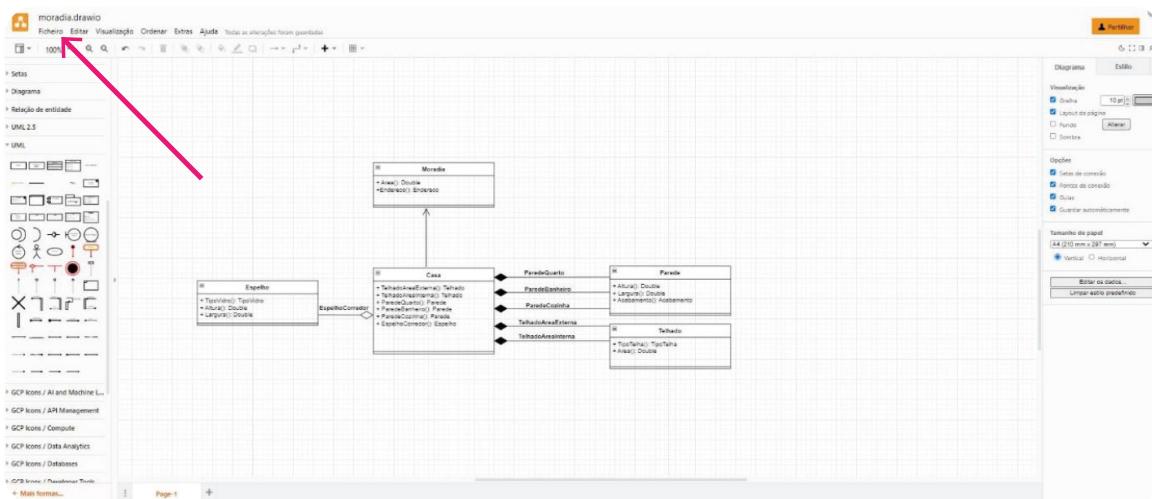


Figura 18 - Salvando o diagrama

Fonte: Do autor (2022)

12. Em seguida, clique na opção “Salvar” ou “Guardar”:

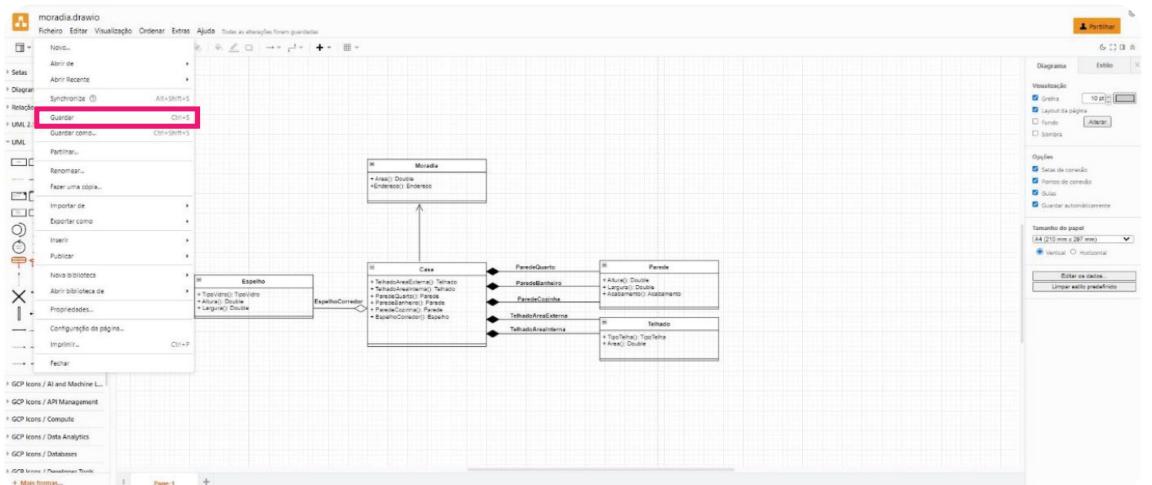


Figura 19 - Salvando o diagrama – passo dois

Fonte: Do autor (2022)

13. Você pode exportar o diagrama para vários tipos de extensões:

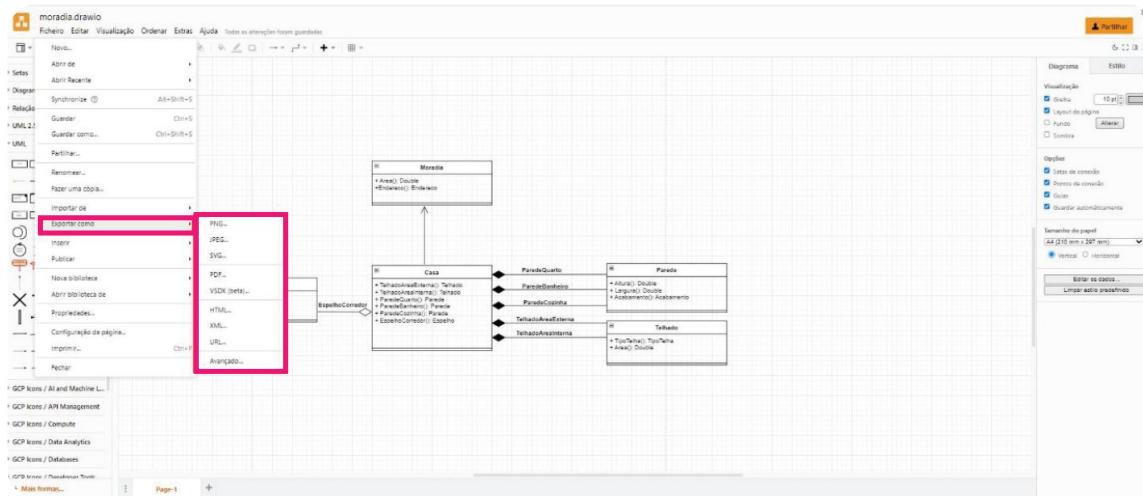


Figura 20 - Exportando o Diagrama de Classes

Fonte: Do autor (2022)

REFERÊNCIAS

LUCIDCHART. **O que é um diagrama de classe UML?** Lucid Software Inc., s.d. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 28 ago. 2022.

VENTURA, Plínio. **Entendendo o Diagrama de Classes da UML:** Modelo de Classes com UML. Portal Até o Momento - Engenharia de Software, 16 jul. 2018. Disponível em: <https://www.ateomomento.com.br/uml-diagrama-de-classes>. Acesso em: 28 ago. 2022.

OMG - OBJECT MANAGEMENT GROUP. **Unified Modeling Language:** Superstructure, version 2.1.1 (with change bars) formal/2007-02-03. Needham, Massachusetts: OMG, 2007 [online]. Disponível em: <https://www.omg.org/spec/UML/2.1.1/Superstructure/PDF/changebar>. Acesso em: 28 ago. 2022.



Alessandra Santos

Experiência em análise e desenvolvimento de software, desenho de arquitetura de sistemas e palestrante sobre tecnologias Microsoft. Atualmente, é Arquiteta de Software no Banco Votorantim e Mentora Educacional no LAB365 do SENAI/SC.

