



Curso Técnico em Desenvolvimento de Sistemas Online

PLANEJAMENTO DE TRABALHO DE CONCLUSÃO DE CURSO

GEEaD - Grupo de Estudo de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Expediente

*GEEaD – CETEC GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
PLANEJAMENTO DO TRABALHO DE CONCLUSÃO DE CURSO - PTCC*

Autor: Eliana Cristina Nogueira Barion

Revisão Técnica: Lilian Aparecida Bertini

Revisão Gramatical: Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação: Flávio Biazim

APRESENTAÇÃO

Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.

O material foi elaborado para servir de apoio aos estudos dos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.

Esperamos que este livro possa contribuir para uma melhor formação e aperfeiçoamento dos futuros Técnicos.

AGENDA 5

UML - DIAGRAMA
DE CLASSE COMO
INSTRUMENTO PARA
DESENVOLVIMENTO
DE PROJETOS





MERGULHANDO NO TEMA...

Por meio de análise e abstração definimos as entidades do mundo real a serem representadas na aplicação, a uma das representações gráficas desenvolvida a partir dessa análise damos o nome de Diagrama de Classes, onde são definidas as classes de objetos, suas operações, atributos e as relações entre elas.

A dificuldade é que não há somente um método ou receita para escolher as classes de um sistema. Essa é uma tarefa que depende da experiência do desenvolvedor, por isso nas fases iniciais do projeto, as classes são chamadas de classes candidatas ou de análise, pois há grande probabilidade que mudem no decorrer do projeto.

De qualquer forma o desenvolvimento do Diagrama de Classes é muito importante, ele auxilia na construção do banco de dados, uma vez que cada classe identificada pode representar uma tabela no banco.

Mas o diagrama de classes representa muito mais do que isso, não é mesmo? Com certeza você se lembra que em UML (Unified Modeling Language), no Diagrama de Classes, uma classe possui três divisões, o nome da classe, seus atributos e funcionalidades ou métodos:

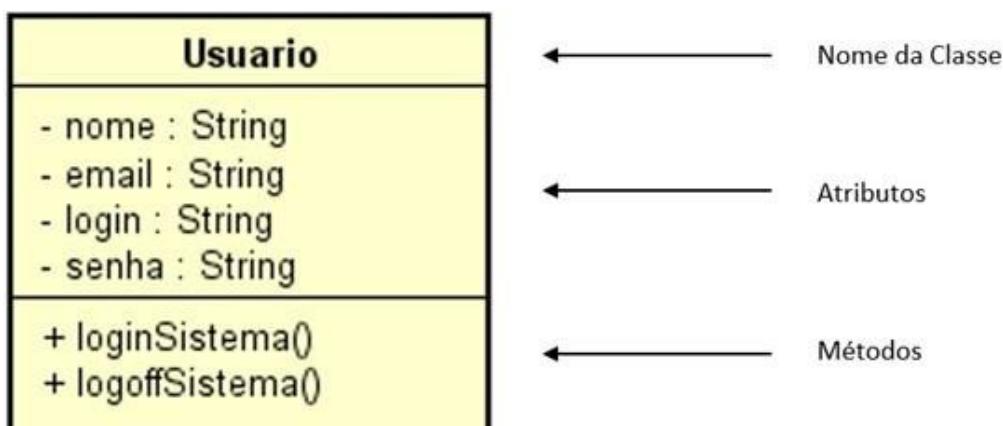


Imagen 03 - Notação de Classe em UML

Lembretes e dicas de boas práticas:

- Classes devem representar uma só abstração do mundo real. Por exemplo, a classe Usuario contém atributos e métodos. Os atributos são os dados que são necessários armazenar e os métodos são as ações executadas pelos objetos, como por exemplo, o login e logoff no sistema. Portanto, se precisar armazenar os logs de acesso desses usuários é melhor criar uma outra classe, Histórico, por exemplo, contendo os atributos data e hora que entrou e saiu do sistema.
- Utilize substantivos para nomear as classes e inicie com uma letra maiúscula. Quando a classe possuir um nome composto, como Menu Principal, o primeiro caractere de cada palavra deve ser sempre maiúsculo.
- Nas fases iniciais de desenvolvimento, pode-se suprimir os atributos e os métodos deixando somente o nome da Classe.

O roteiro a seguir, apresenta uma ótima alternativa para um bom começo de desenvolvimento de um Diagrama de Classes:

1) Identificar os objetos tangíveis

O objeto é algo tangível, você consegue identificá-lo no mundo real, como se ele estivesse na sua frente. Exemplos de objetos que podemos identificar em um consultório: Médico, Funcionário, Paciente etc.

2) Identificar os objetos por seus atributos

Podemos também encontrar objetos através de suas características ou atributos. Um objeto pode não ter seu nome informado, dificultando sua identificação, mas em uma situação pode ter sido mencionado um atributo, por exemplo CRM, onde poderíamos supor que esse dado é de um Médico, identificando o objeto Médico mesmo que o seu nome não tenha sido pronunciado.

3) Fazer uma lista completa com todos os objetos encontrados

Utilizada somente para facilitar a visualização de todos os objetos encontrados durante o processo de abstração. Exemplo:

- | | |
|---|---|
| <ul style="list-style-type: none"> • Médico • Funcionário • Paciente • Agenda • Consulta | <ul style="list-style-type: none"> • Histórico do Paciente • Solicitação de Exames • Receita de Medicamentos • Contas à Pagar • Contas à Receber |
|---|---|

4) Agrupar os objetos por semelhança

Esse passo tem o objetivo de identificar possíveis situações em que podemos aplicar os conceitos de generalização e especificação, ou simplesmente eliminar classes, agrupando os objetos encontrados por características ou atributos semelhantes. Exemplo:

- | |
|---|
| <ul style="list-style-type: none"> • Médico, Funcionário e Paciente = Pessoas • Solicitação de Exames, Receita de Medicamentos = Documentos • Contas à Pagar, Contas à Receber = Caixa |
|---|

Observação: pode ser que alguns objetos da lista não necessitem ser agrupados, o que significa que eles já estão classificados corretamente.

5) Eliminar classes desnecessárias ou repetidas

Observe que contas a pagar e a receber na essência são situações onde há a movimentação de dinheiro, com apenas uma diferença, uma é de entrada e a outra de saída, neste caso podemos deixar somente uma classe.

Adaptado de Orientações básicas na elaboração de um diagrama de classes.

<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>.

Acessado em 20/10/2025.

Percebeu como o conhecimento do contexto é fundamental? O Diagrama de Casos de Uso também pode auxiliá-lo neste momento, mesmo que não seja um artefato obrigatório, as equipes que decidem por desenvolvê-lo acabam reconhecendo a sua importância, principalmente se o nível de detalhamento ultrapassar os limites da representação gráfica por meio de especificações ricas em detalhes sobre os processos do sistema.

Assista o vídeo que o auxiliará na compreensão de um Diagrama de Classes.



Imagen 04 – Diagrama de Classe - UML

<https://www.youtube.com/watch?v=niGtsDAagzc> - Acessado em 20/10/2025

Para aplicar todos esses conceitos você pode utilizar a ferramenta Drawio.com (<https://www.drawio.com/blog/uml-class-diagrams>):

Criar diagramas de classes UML



Os diagramas de classes UML são usados para ilustrar a estrutura de um programa de computador. Eles detalham os tipos de dados ou atributos armazenados em cada "classe", os métodos (operações ou funções) que cada classe fornece e os relacionamentos entre as classes.

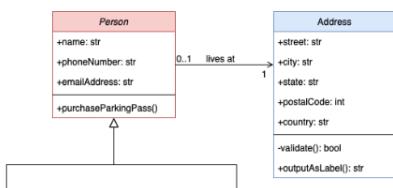


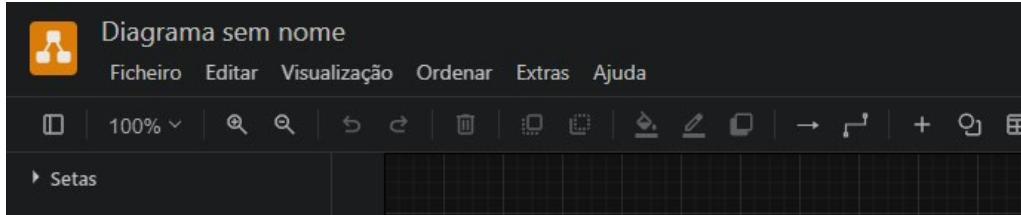
Imagen 05 - Ferramenta Draw.io.

Draw.io (Diagrams.net)

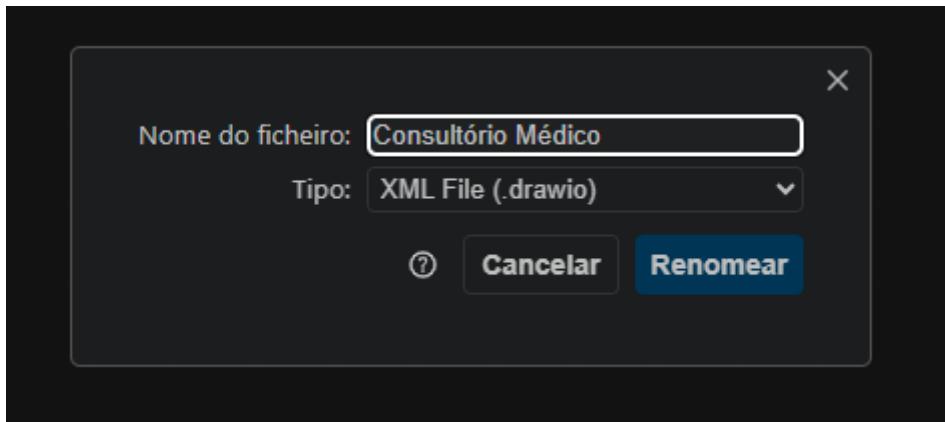
O Draw.io é uma ferramenta gratuita e online para criação de diagramas UML e fluxogramas.

Acesse a ferramenta através do site: <https://app.diagrams.net/>. Você pode armazenar as informações diretamente no navegador, Google Drive, ou dispositivo local.

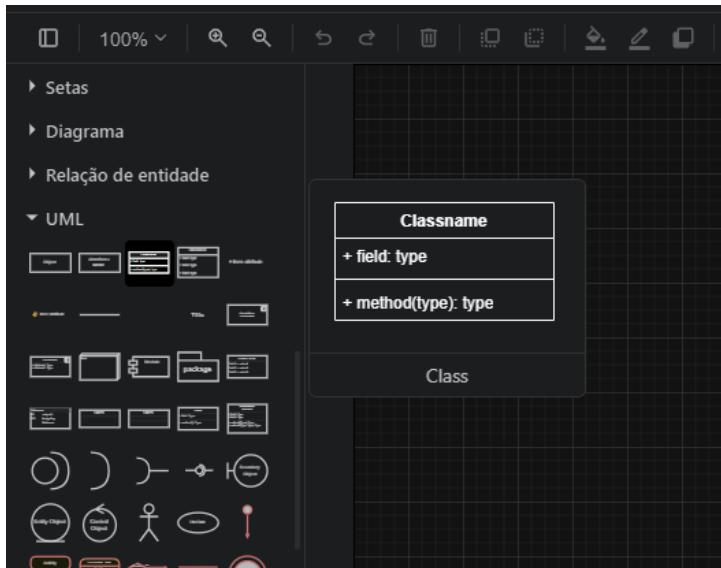
Adicione um nome para o seu diagrama:



Ao clicar sobre o título é possível renomear:

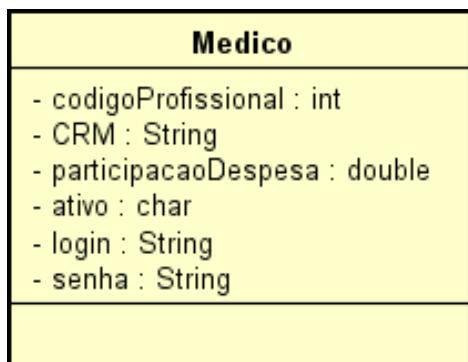


Vamos agora criar a classe Médico, selecione a opção Classe (Class) no menu lateral:

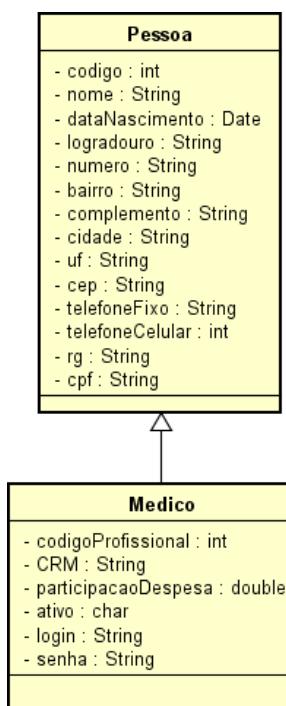
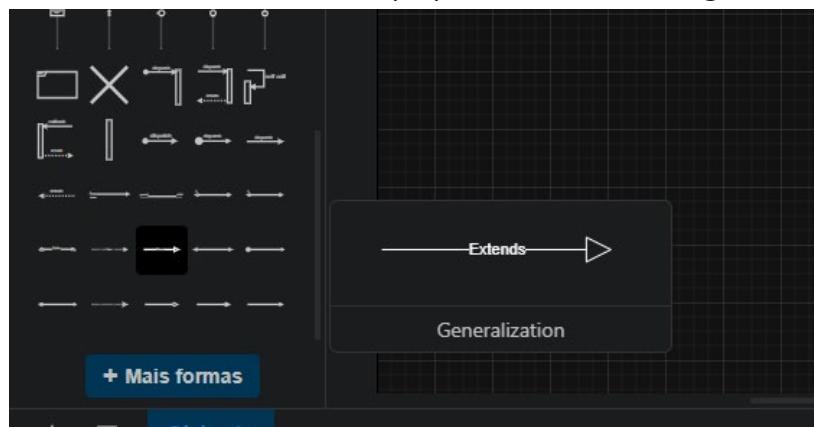


O próximo passo é definir os atributos da classe, que são as características ou dados que deverão ser armazenados de cada uma delas.

Em UML:



Não podemos deixar de aplicar o conceito de herança ou generalização entre as classes Médico e Pessoa. Crie a Classe Pessoa e aplique o conceito no Diagrama:



Se achar necessário você poderá complementar qualquer um dos documentos caso entenda que alguma informação importante para sua continuidade esteja faltando.



VOCÊ NO COMANDO

Agora é com você!

Utilize a ferramenta Draw.io ou outra ferramenta de sua escolha e termine de criar as outras Classes do projeto Consultório aplicando o conceito de generalização onde achar que é possível.

Achou complicado? Realmente não é simples, mas uma boa documentação de levantamento e especificação de requisitos nesse momento faz toda diferença.

O Diagrama de Casos de Uso também poderá auxiliá-lo nessa tarefa, ao especificar os casos de uso representados no diagrama, além de entender melhor como o processo deverá ser executado por meio da descrição dos fluxos principal e alternativos, ele também acaba se tornando uma fonte rica em detalhes sobre o contexto, e melhor, sistematicamente organizada.

Veja a seguir como ficaria o caso de uso Logar no Sistema.

Logar no Sistema

| | |
|---------------------|--|
| Caso de Uso | Logar no Sistema |
| Requisito | RF1. O usuário deverá possuir um login e senha para acessar ao sistema |
| Atores | Médico e Funcionário |
| Pré-Condição | O sistema estar instalado. |
| Condição de Entrada | O Ator acessa o sistema |
| Fluxo Principal | <ol style="list-style-type: none"> 1. O Sistema apresenta a interface de login. 2. O Ator informa login e senha. 3. O Sistema valida as credenciais do usuário. [A1] 4. O Sistema apresenta a interface principal do sistema. 5. Caso de Uso é encerrado |
| Fluxos Alternativos | <p>A1. O Sistema não conseguiu validar as credenciais do usuário.</p> <ol style="list-style-type: none"> 1. O Sistema apresenta uma mensagem ao usuário. 2. O Sistema retorna ao passo 2 do fluxo principal. [A2] <p>A2. O Sistema não conseguiu validar as credenciais do usuário por 3 vezes.</p> <ol style="list-style-type: none"> 1. O Sistema apresenta uma mensagem ao usuário. 2. O Sistema é encerrado. |

Quanto mais detalhado forem os requisitos, melhor será a abstração para qualquer tipo de Diagrama que a equipe decidir utilizar.

No exemplo, podemos identificar Médico, Funcionário como possíveis classes do sistema, login e senha com alguns de seus atributos e logar no sistema como uma das ações que eles poderão executar no sistema.

| Funcionario |
|----------------------------|
| - matricula : int |
| - cargo : String |
| - salario : double |
| - dataAdmissao : Date |
| - dataRecisao : Date |
| - login : String |
| - senha : String |
| + logarSistema() : boolean |

| Medico |
|--------------------------------|
| - codigoProfissional : int |
| - CRM : String |
| - participacaoDespesa : double |
| - ativo : char |
| - login : String |
| - senha : String |
| + logarSistema() : boolean |

