



# Utilização de Ferramentas UML

## **F**erramentas UML

Embora seja temerário fazer generalizações, podemos observar que o processo de desenvolvimento de um software não é trivial, embora não seja difícil, é altamente complexo. E não há nada de errado com isso, pois o processo criativo é multifacetado. Por exemplo, o projeto de arquitetura de software é uma construção tecnológica e cultural, cujas bases foram surgindo como uma abstração para gerenciar um âmbito complexo e criar mecanismos de comunicação entre entidades. A circunstância não implica nem de longe, porém, que o projeto de arquitetura de software deve envolver apenas fatores relacionados a codificação e seus relacionamentos (estruturas); mas descrever como os elementos de um sistema se relacionam e trabalham juntos para atender aos requisitos do projeto de software. Na verdade, caberia pensar além e perguntar o que há entre ideação e codificação, ideia e produto. Nesse sentido, almejando elaborar um juízo crítico sobre as principais dimensões da arquitetura e o design de software, crê-se possível, nesta aula, proceder uma reflexão sobre ferramentas para modelagem orientada a objetos.

Portanto, nesta aula, continuaremos nosso diálogo sobre programas de diagramação comumente usados na indústria em desenvolvimento de software. Além disso, vamos relembrar o propósito da existência da UML, conceitos de modelagem dentro da UML e objetivos da arquitetura e design de software.

Vamos lá?!



O processo de desenvolvimento de software moderno abrange a documentação, programação, teste, projeto, manutenção e um conjunto de atividades parcialmente ordenadas focadas no ciclo de vida de um produto de software. Portanto, do ponto de vista do processo de desenvolvimento, umas das principais dificuldades da esfera criativa é, sem dúvida, gerenciar o ciclo de vida de um sistema de informação. De fato, no âmbito da indústria 4.0, os desenvolvedores de software se deparam com a tarefa de reorientar os processos de produção; mas, infelizmente, para reorientar ou transformar tal projeto em um ciclo organizado, muitas vezes, devemos reescrever tudo do zero.

Como assim? Na maioria dos projetos de software, o ciclo do produto acontece quase da mesma maneira: ideias e planos. Novamente, não há nada de errado com planos. No entanto, lembre-se: existem muitas etapas entre ideação e codificação. Você tem uma ideia interessante de produto? Não basta apenas validar as ideias através de um protótipo, é necessário compreender os principais componentes do produto, suas estruturas e como eles interagem entre si. Na prática, significa que, conscientemente ou não, cada uma das decisões da etapa criativa pode ter um impacto significativo no desempenho e sucesso geral do produto de software. Teoricamente, trata-se de ponto relevante, já que o desenvolvedor deve:

- Alinhar os requisitos do projeto e definir as necessidades do cliente;
- Projetar soluções estruturais alinhadas aos objetivos do projeto;
- Implementar casos de uso e cenários;
- Projetar a estrutura do sistema, bem como os requisitos funcionais e de qualidade;
- Construir diagramas UML para auxiliar na modelagem da arquitetura do projeto;




Silva e Videira (2001, p. 237) comentam que “os diagramas de arquitetura (ou diagramas de implementação) descrevem aspectos da fase de implementação e instalação de um sistema de software, designadamente a estrutura e dependências de código fonte e de módulos executáveis tal como a sua respectiva instalação nas diferentes plataformas computacionais subjacentes”. Podemos observar que a UML, como sabemos, é uma linguagem de modelagem que serve como meio de comunicação em um projeto; consiste em um conjunto integrado de soluções projetadas para auxiliar os desenvolvedores a construir e visualizar artefatos de sistemas, expressar o design de projetos de software, validar a arquitetura do projeto, e assim por diante.

Na modelagem de processos e arquitetura de projetos, via de regra, existem três modelos principais: Objeto, Funcional e Dinâmico. Para descrever esses modelos na produção de um software, dois tipos de diagramas são usados: estruturais e comportamentais. E quais são os programas de diagramação comumente usados?

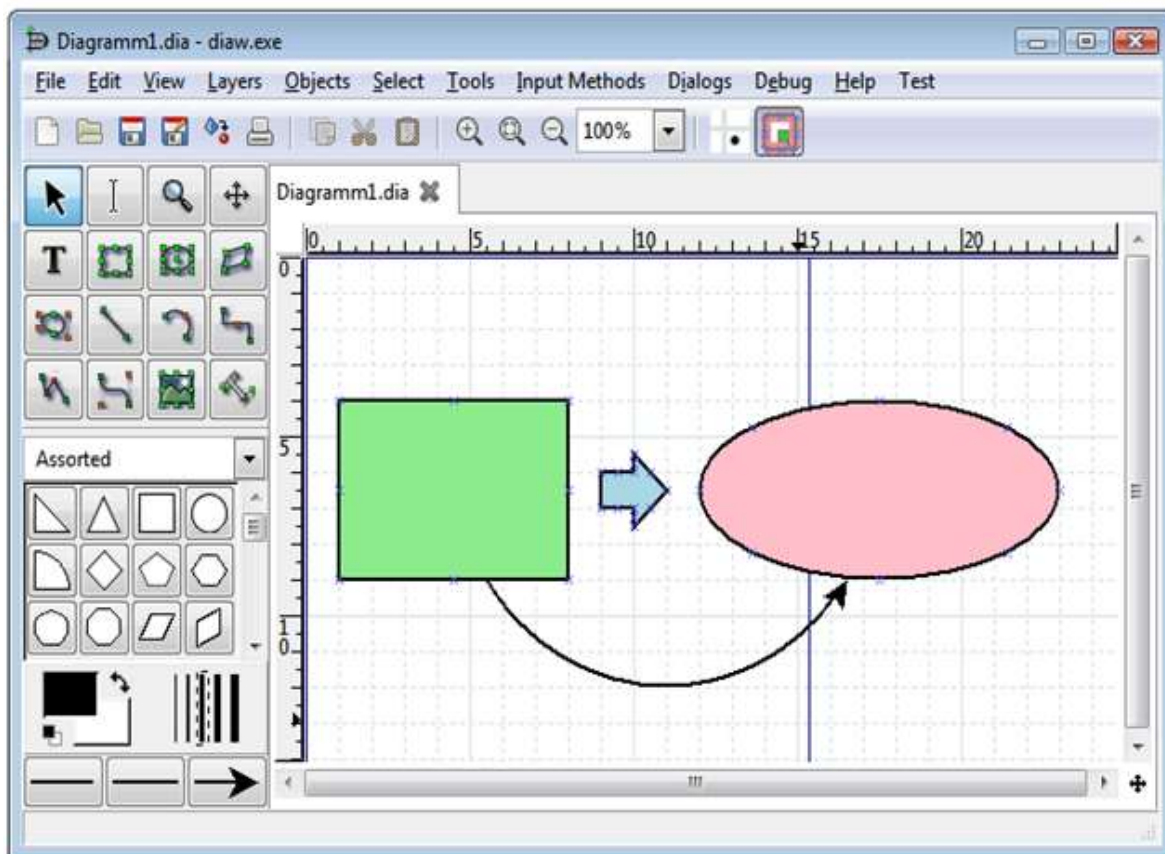
Existem muitas ferramentas de modelagem UML, mas vamos focar em três: Dia (Gnome), ArgoUml e Umbrello. As ferramentas supracitadas foram escolhidas pois são gratuitas e estão disponíveis para download nos sites dos desenvolvedores. A seguir uma descrição de cada ferramenta será fornecida.

### **Dia Diagram Editor (Gnome)**

Dia é uma ferramenta de criação de diagramas baseado no projeto GTK (uma aplicação multiplataforma para criar interfaces gráficas de usuário) para GNU/Linux, MacOS X, Unix e Windows, e é lançado sob a licença GPL. O editor de diagramas é uma ferramenta versátil, embora seja uma solução inspirada no Microsoft Visio (uma aplicação para projetar uma grande variedade de diagramas). De acordo com o site da desenvolvedora, o Dia " pode ser usado para desenhar muitos tipos diferentes de diagramas. Atualmente, possui objetos especiais para ajudar a desenhar diagramas de relacionamento de entidade, diagramas UML, fluxogramas,

diagramas de rede e muitos outros diagramas. Também é possível adicionar suporte para novas formas escrevendo arquivos XML simples, usando um subconjunto  SVG para desenhar a forma” (DIA, 2020, documento online).

Visite a página de download para obter sua cópia do Dia Diagram Editor: <https://wiki.gnome.org/Apps/Dia/Download>. (acesso em 30/09/2022)



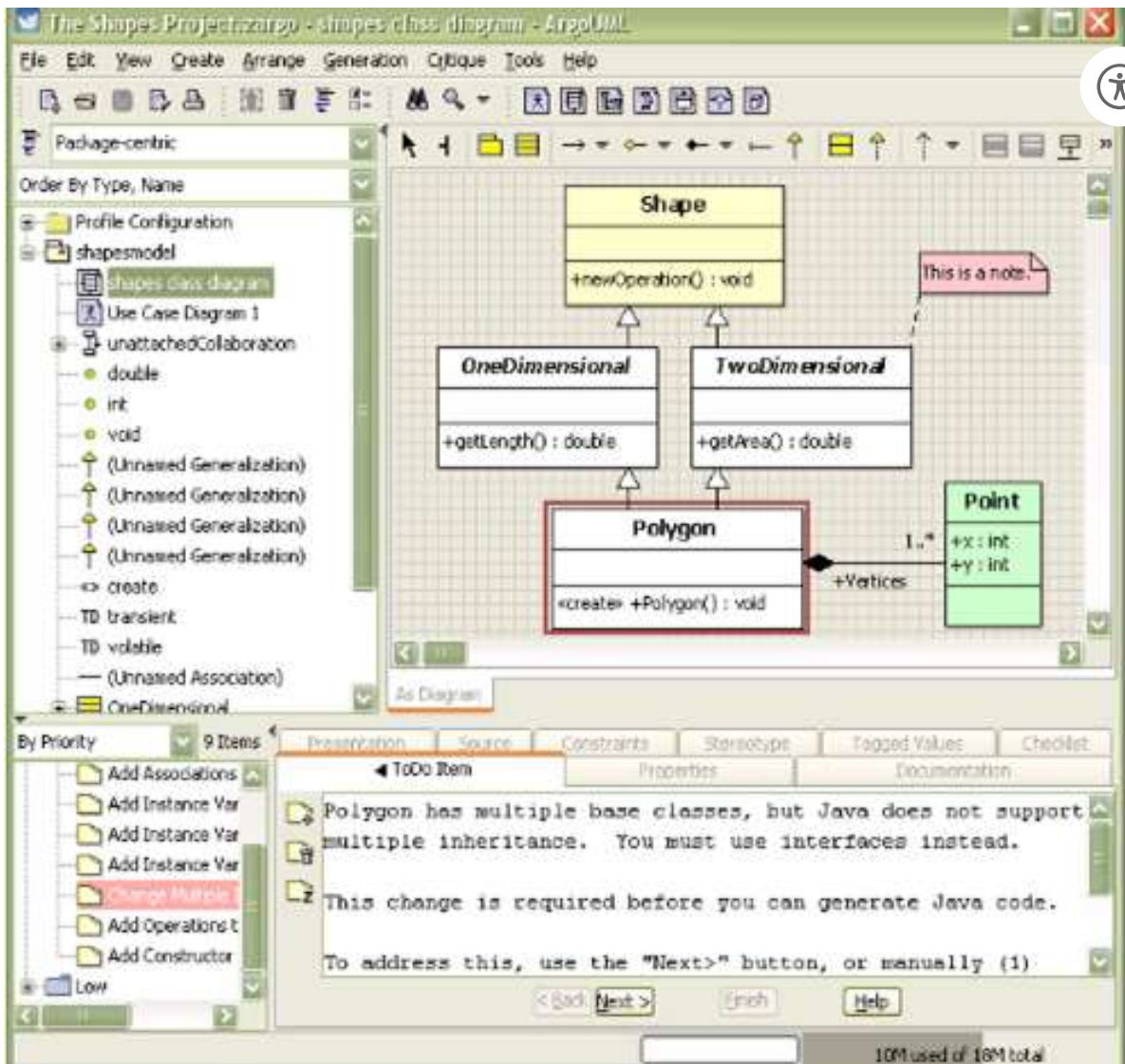
## ArgoUML

O ArgoUML é uma solução inovadora e flexível que permite aos desenvolvedores projetarem o design, a documentação e até mesmo o projeto estrutural de software orientados a objetos. De acordo com o site da ferramenta, é a principal ferramenta de modelagem UML de código aberto e inclui suporte para todos os diagramas UML 1.4 padrão. Ele roda em qualquer plataforma Java e está disponível em dez idiomas" (ARGOUML, 2022, documento online). Principais características do ArgoUML:



- Padrão aberto (Open standard): XMI, SVG and PGML;
- 100% Java;
- OpenSource: permite que a comunidade personalize a experiência do produto;
- Focando nos diagramas de casos de uso e diagramas de classes;
- Simplicidade comparativa de diagramas UML;
- É possível reutilizar um projeto existente.

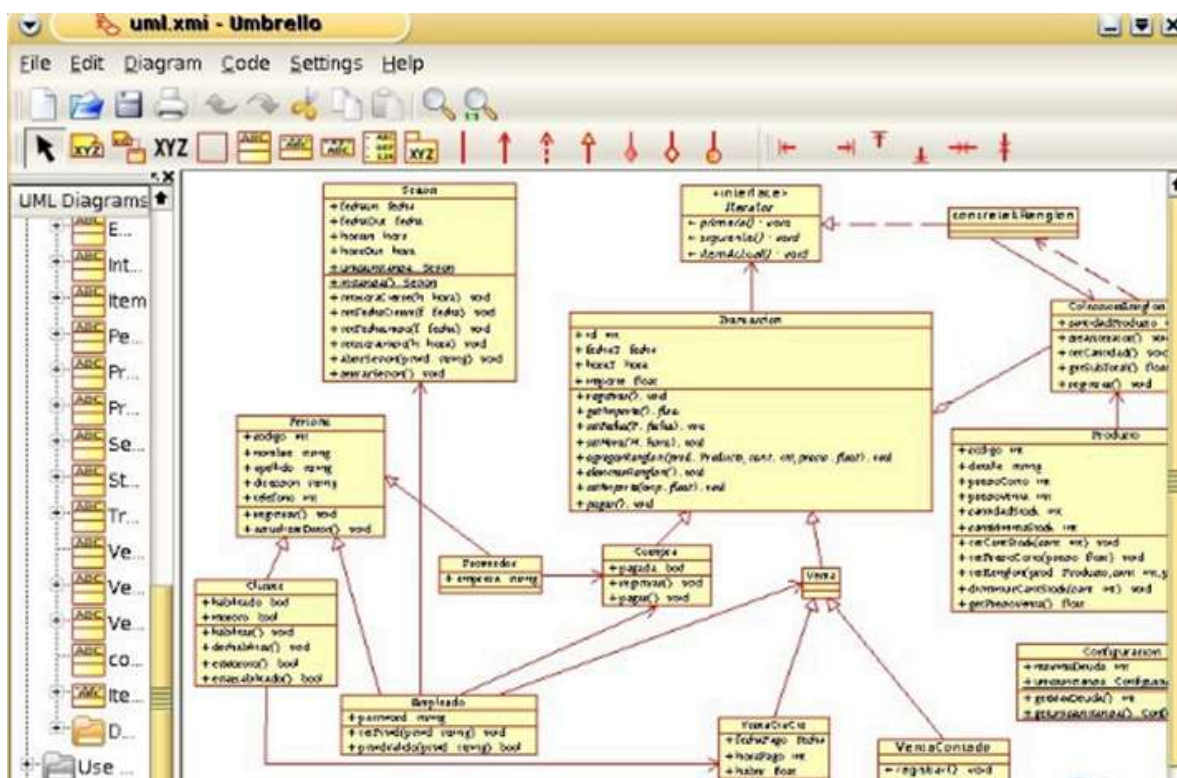
Visite a página de download para obter sua cópia do ArgoUML: <https://argouml-tigris-org.github.io/tigris/argouml/> (acesso em 30/09/2022)



## Umbrello

Umbrello é uma aplicação UML que permite criar diagramas de software orientados a objetos. A ferramenta permite criar novas classes com base nas já existentes, além da possibilidade de trabalhar com abstrações, isto é, descrever como um objeto real se comporta. De acordo com o site do programa, “Umbrello vem com versões do KDE, incluídas em todas as distribuições Linux e disponíveis através do seu gerenciador de pacotes e como instalador para Windows e Mac OS X”, além de permitir que o usuário “crie diagramas de software e outros sistemas em um formato padrão para documentar ou projetar a estrutura de seus programas” (UMBRELLO, 2020, documento online).





Se observamos, de fato, as ferramentas e aplicações apresentadas são soluções que utilizam notação UML para descrever a lógica do projeto. No entanto, ainda existem muitos bons serviços que permitem criar diagramas UML, logicamente, é difícil escolher os melhores, porque depende do contexto e há um grande número deles. Além dos supracitados, podemos considerar: StarUML, Microsoft Visio, [Diagrams.net](#) e LucidChart. Lembre-se que não existe a melhor ferramenta, pois depende do contexto. Na modelagem de processos e arquitetura de projetos, você deve levar em consideração os tipos de diagramas UML: estruturais e comportamentais.



### Atividade extra

Para se aprofundar no assunto desta aula, no site e-publicacoes/uerj da UERJ – Universidade Federal do Estado do Rio de Janeiro leia o artigo “Avaliação de Ferramentas para Desenvolvimento Orientado a Objetos com UML” dos autores Alexandre Nunes Costa, Vera M. B. Werneck e Marcio Francisco Campos.

### Referências Bibliográficas

ArgoUML, 2022. Página inicial. Disponível em: <https://argouml-tigris-org.github.io/tigris/argouml/>. Acesso em: 27 de set. de 2022.

Dia Diagram Editor, 2020. Página inicial. Disponível em: <https://wiki.gnome.org/Apps/Dia/Download>. Acesso em: 27 de set. de 2022.

BOOCH, Grady. **Uml - Guia do Usuário**. Rio de Janeiro: Editora Campus, 2018.

GUEDES, Gilleanes T. A. **UML 2 - Uma Abordagem Prática**. São Paulo: NovaTec, 2018.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**. Porto Alegre: Bookman, 2010.

SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Pearson, 2015.

SILVA, Alberto Manuel Rodrigues; VIDEIRA, Carlos Alberto Escaleira. **UML, Metodologias e Ferramentas Case**. Edições Centro Atlântico, Portugal, 2001.



**Ir para exercício**

