



TDE 1 - Importância dos Principais Diagramas UML

Ary Felipe Farah e Silva

1. Diagramas

1.1 Pacotes

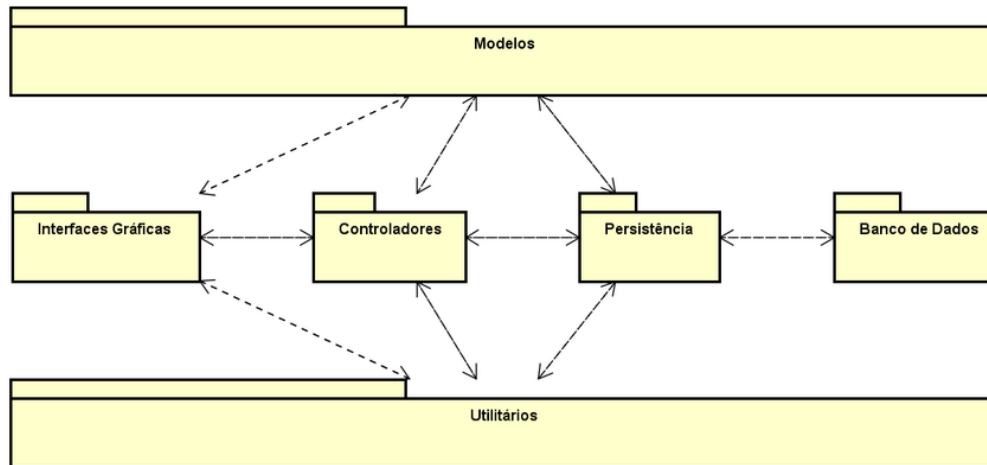
Importância

O **Diagrama de Pacotes** é fundamental para gerenciar a estrutura de um sistema, podendo ser usado para simplificar diagramas mais complexos.

Pontos de destaque:

- Agrupamento de classes, interfaces e elementos em pacotes
- Visão clara da estrutura e interação dos componentes
- Facilitação da separação de responsabilidades
- Representação de módulos ou subsistemas
- Destacamento de dependências entre pacotes
- Identificação de acoplamentos fortes para melhorar independência
- Facilitação da manutenção e evolução do sistema
- Visão de alto nível do sistema para desenvolvedores e gerentes
- Reuso de pacotes em diferentes partes ou projetos
- Visualização de componentes reutilizáveis e sua integração
- Documentação do design do sistema
- Facilitação de entendimento da arquitetura e interações para novos membros

Exemplo



1.2 Componentes

Importância

O **Diagrama de Componentes** mostra o relacionamento entre diferentes componentes de um sistema.

Principais benefícios:

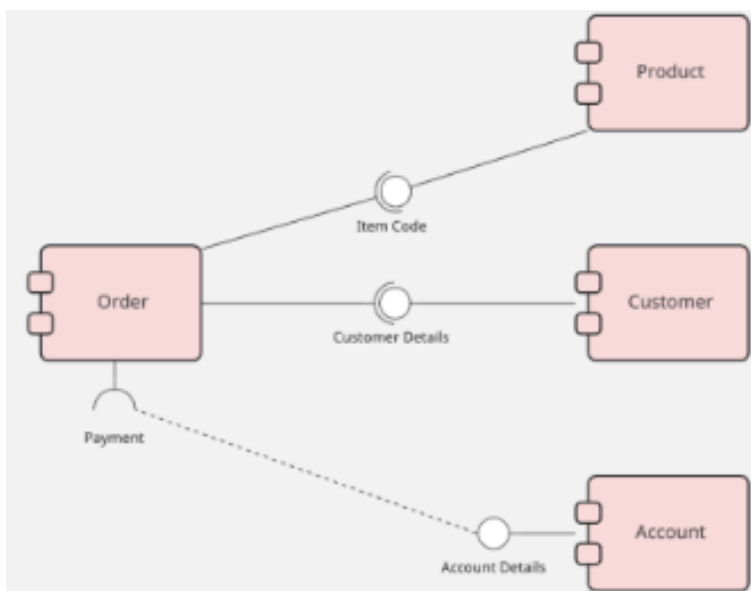
- Visualização da arquitetura física do sistema
- Organização e interligação de módulos, bibliotecas e subsistemas
- Planejamento da implantação do sistema
- Mapeamento dos componentes de software para a infraestrutura física
- Promoção da modularidade e desenvolvimento independente
- Facilitação da reutilização de componentes em outros projetos
- Identificação de dependências entre partes do sistema
- Prevenção de dependências cíclicas e acoplamento excessivo
- Garantia de flexibilidade e facilidade de manutenção
- Documentação para desenvolvedores e arquitetos
- Facilitação da compreensão do sistema e dos papéis dos componentes
- Planejamento de interfaces expostas e consumidas pelos componentes
- Garantia de comunicação eficaz entre componentes

- Identificação e isolamento de problemas
- Simplificação de manutenção e atualizações

Elementos

- **Componentes** → representam partes de um sistema, como uma biblioteca, um serviço, um banco de dados ou interface do usuário.
- **Interfaces** → pontos de interação entre componentes, especificam serviços fornecidos/consumidos.
- **Conexões/Dependências** → linhas que indicam como são as interações/dependências entre componentes.

Exemplo



1.3 Implantação

Importância

O D

Diagrama de Implantação mostra a arquitetura de um sistema e a relação entre componentes de software e a infraestrutura física em que o sistema será executado.

Pontos principais:

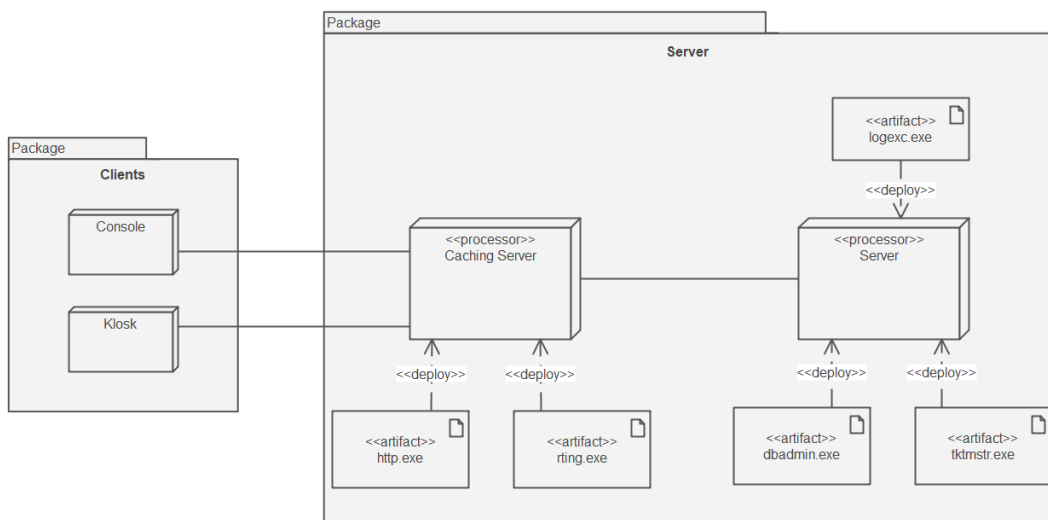
- Representação da estrutura física do sistema

- Visualização de nós (servidores, dispositivos) e a implantação de componentes de software
- Modelagem de ambientes de execução em máquinas ou ambientes distintos
- Exibição de comunicações entre nós (servidores, redes, roteadores, dispositivos)
- Mostra a conectividade física ou lógica entre os elementos
- Facilitação do planejamento da infraestrutura necessária para o sistema
- Identificação de requisitos de hardware, rede e armazenamento
- Definição de distribuição de software entre nós físicos
- Aplicação a sistemas distribuídos, nuvem ou ambientes locais

Elementos

- **Nós** → dispositivos de hardware ou máquinas físicas/virtuais onde o software será implantado (servidores, computadores, dispositivos móveis...).
- **Artefatos** → partes físicas do software (arquivos, bibliotecas, bases de dados) implantadas nos nós.
- **Conexões** → conexões físicas/lógicas entre nós (redes, links de comunicação...)

Exemplo



1.4 Tempo

Importância

O D

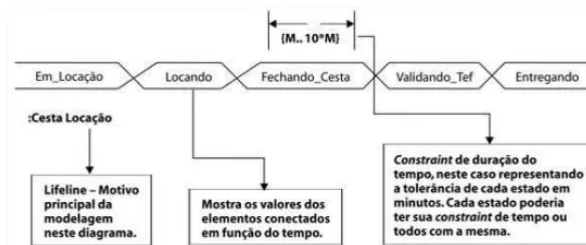
Diagrama de Tempo foca no comportamento temporal dos sistemas, exibindo as interações do sistema em razão do tempo.

- Monitoramento de comportamentos temporais em sistemas embarcados ou em tempo real.
- Análise de protocolos de comunicação, onde eventos precisam acontecer em uma ordem específica dentro de limites de tempo.
- Visualização da sincronização de eventos ou mudanças de estado entre componentes em um sistema.

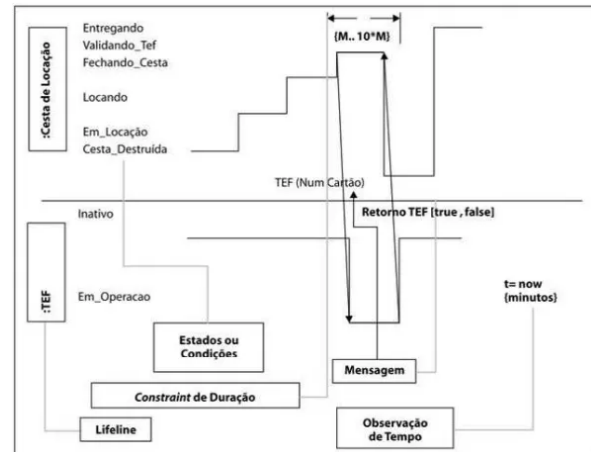
Elementos

- **Eixo do tempo** → eixo horizontal do gráfico, podendo ser especificado (segundos, minutos, etc.).
- **Estados ou condições** → eixo vertical, mostra os diferentes estados/condições de um objeto ou sistema. Cada linha representa uma entidade (objeto, componente, etc.) e suas mudanças de estado ao longo do tempo.
- **Transições** → mudanças de estado representadas por linhas que conectam os diferentes estados ao longo do tempo. Elas indicam os pontos em que ocorre a transição de um estado para outro.
- **Instantes de tempo** → São marcados pontos no eixo horizontal para mostrar eventos importantes ou mudanças de estado que acontecem em momentos específicos.
- **Mensagens/Interações (Opcional)** → mensagens/interações entre os objetos.

Exemplo



Fluxo do Contexto do elemento Foco



Detalhamento do elemento Foco

1.5 Estrutura Composta

Importância

O

Diagrama de Estrutura Composta descreve a estrutura interna de um sistema, mostrando como suas partes estão relacionadas e como interagem entre si. Oferece uma visão geral lógica de todo ou parte de um sistema de software,

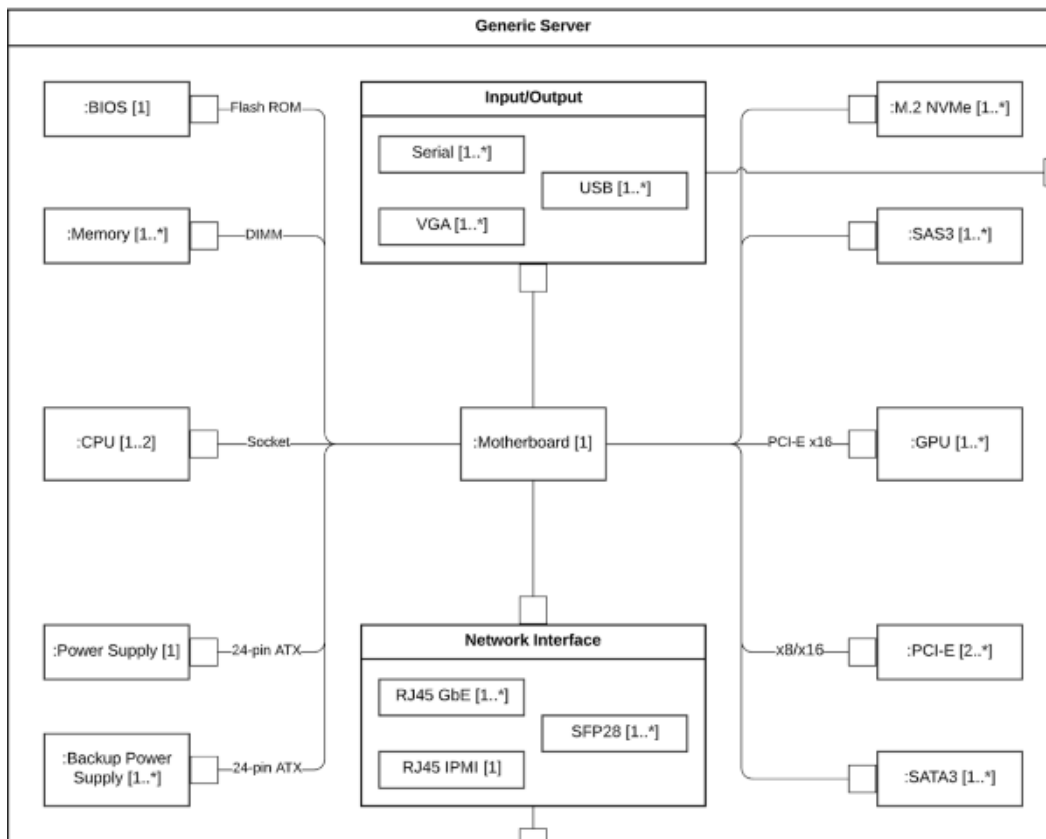
- Descrever a **arquitetura interna** de sistemas complexos.
- Modelar a **colaboração entre partes** de um sistema para realizar uma função.
- Especificar **interações** entre componentes e suas interfaces.
- Detalhar o design interno de uma classe, ajudando na implementação e documentação de um sistema.

Elementos

- **Classes ou Componentes** → entidades principais no sistema. Retângulos contendo partes internas e descrições de suas interações.
- **Portas** → Pontos de interação que permitem a comunicação entre um componente e o ambiente externo. As portas são representadas por pequenos quadrados no contorno do retângulo de uma classe ou componente.
- **Partes** → Objetos internos que constituem a classe/componente. São mostrados dentro do retângulo do componente pai.

- **Conectores** → Linhas que representam as interações/fluxos de comunicação entre as partes internas ou entre portas. Indicam colaborações ou troca de informações.
- **Interfaces** → Descrições dos serviços que uma parte/porta oferece ou requer. Frequentemente associadas às portas para descrever os tipos de comunicação que são permitidos.
- **Conexões entre portas e interfaces** → Mostram como as portas se conectam a outras partes internas/externas por meio de interfaces.

Exemplo



1.6 Visão Geral da Interação

Importância

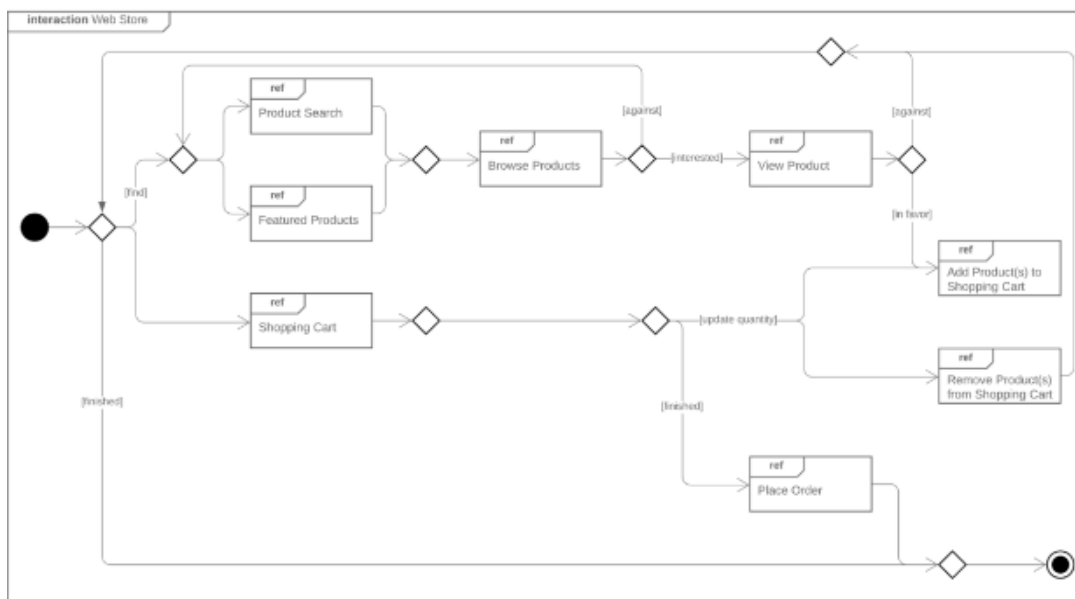
O **Diagrama de Visão Geral da Interação** combina elementos de diagramas de atividades e de sequência, mostrando uma visão panorâmica das interações em um sistema. Foca na

sequência de interações e nos fluxos entre os elementos, fornece um resumo ou um guia para outros diagramas detalhados. Geralmente faz referência a outros diagramas (sequência, comunicação...), detalhando pontos específicos.

Elementos

- **Interações resumidas** → Interações importantes em um sistema sem os detalhes de cada mensagem trocada. Link para outro diagrama de interação (como um diagrama de sequência ou de colaboração) que entra em detalhes.
- **Nós de atividade** → Atividades/tarefas que ocorrem durante a interação. Usados para mostrar um fluxo de atividades de alto nível e indicar quando uma interação ocorre.
- **Fluxo de controle** → Linhas e setas que mostram a ordem em que as interações ocorrem. Fluxos condicionais e loops podem ser representados..
- **Nós de decisão** → Pontos onde o fluxo de controle se divide com base em condições. Utilizados para ilustrar ramificações no processo de interação.
- **Nós de fusão** → Múltiplos fluxos de controle se combinam em um único fluxo.
- **Blocos de interação** → Interações específicas dentro do diagrama, "atalhos" para expandir os detalhes em outros diagramas de interação.

Exemplo



2. Trabalhando com todos esses diagramas de forma relacionada

2.1 É possível?

É possível trabalhar com esses 6 diagramas de forma relacionada da seguinte maneira:


- **Diagrama de Pacotes** define a estrutura lógica e organiza o sistema em pacotes, onde cada pacote pode conter diferentes **Componentes**.
- **Diagrama de Componentes** mostra a estrutura física desses pacotes em termos de como o sistema é composto por módulos de software, como eles interagem e dependem uns dos outros.
- **Diagrama de Implantação** expande para mostrar como esses componentes estão distribuídos em infraestrutura física, como servidores, redes e dispositivos.
- **Diagrama de Visão Geral da Interação** oferece uma visão de alto nível de como os componentes interagem entre si, referenciando os fluxos e comportamentos entre componentes em diferentes pacotes ou nós.
- **Diagrama de Estrutura Composta** detalha como cada componente internamente é organizado, mostrando suas partes e interações entre elas.
- **Diagrama de Tempo** pode complementar mostrando o comportamento ao longo do tempo dessas interações e componentes, especialmente para sistemas distribuídos ou com requisitos temporais.

3. Referências

3.1 Referências

O que é um Diagrama de Pacotes e como cria-lo: Explicação com modelos

Este artigo traz métodos simples e dicas sobre como criar um diagrama de pacotes. Além disso, trazemos algumas informações importantes sobre ele.

 https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjpxujRxealAxV4mZUCHcg2DzYQFnoECB8QAw&url=https://gitmind.com/pt/diagrama-de-pacotes.html#:~:text=Um%20diagrama%20de%20pacotes%20pode,outros%20campos%20em%20pacotes%20simplificados.&usg=AOvVaw0mLK9ED9obmMPc-wY_c7xR&opi=89978449

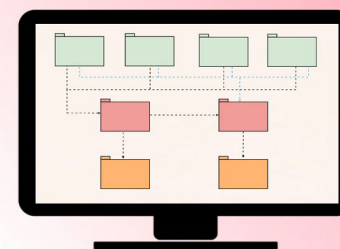


Diagrama de componentes UML: o que é, como fazer e exemplos

Guia definitivo de diagramas de componentes em UML, incluindo uma visão geral, formas e símbolos e até mesmo exemplos para modificar!

 <https://www.lucidchart.com/pages/pt/diagrama-de-componentes-uml>

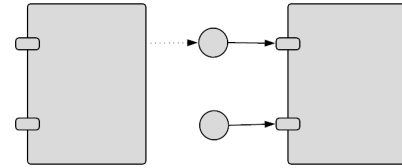

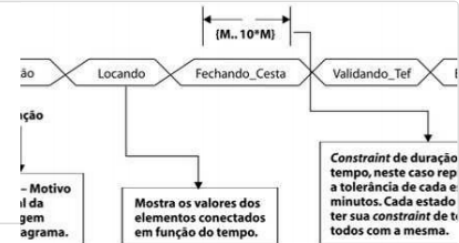


Diagrama de Implantação: [https://creately.com/blog/pt/diagrama/tutorial-do-diagrama-de-implantacao/#:~:text=Um diagrama de implantação é,software físico de um sistema](https://creately.com/blog/pt/diagrama/tutorial-do-diagrama-de-implantacao/#:~:text=Um diagrama de implanta%C3%A7%C3%A3o %C3%A9,software f%C3%ADsico de um sistema)

Diagrama de Tempo


O diagrama de tempo, é um diagrama de interação da UML que tem o propósito de demonstrar as interações do sistema em razão do tempo. Isso...

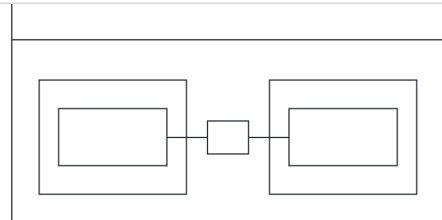
 <https://medium.com/documentacao-uml/diagrama-de-tempo-505aed0e8c6f>



Tutorial de diagrama de estrutura composta


Um guia completo com tudo o que você precisa saber sobre diagramas de estrutura composta em UML. E vem com um modelo pronto para editar!

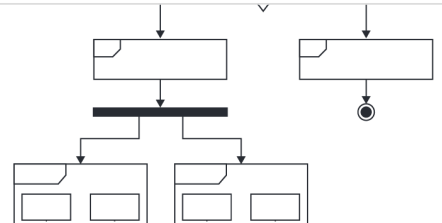
 <https://www.lucidchart.com/pages/pt/diagrama-de-estrutura-composta-uml>



Tutorial sobre diagramas de interação


Um guia completo com tudo o que você precisa saber sobre diagramas de interação em UML. E vem com um modelo pronto para editar!

 <https://www.lucidchart.com/pages/pt/diagrama-de-interacao-uml>



ChatGPT

A conversational AI system that listens, learns, and challenges

 <https://chatgpt.com/>

ChatGPT ●