

# **Aula 05 - Engenharia de Software II**

**Prof. Me. Deivison S. Takatu**

**[deivison.takatu@fatec.sp.gov.br](mailto:deivison.takatu@fatec.sp.gov.br)**

# Sumário

- Diagramas Estruturais – UML;
- Situação-Problema;
- Diagrama de Pacotes;
- Diagrama de Componentes;
- Diagrama de Implantação;
- Atividade.

# Diagramas Estruturais – UML

1. **Diagrama de Classes:** mostra atributos, métodos e relacionamentos das classes.
2. **Diagrama de Objetos:** representa instâncias concretas de classes em um momento específico.
3. **Diagrama de Perfil:** permite personalizar UML com estereótipos e restrições.
4. **Diagrama de Pacotes:** organiza classes e elementos em grupos lógicos.
5. **Diagrama de Componentes:** ilustra a estrutura física do sistema em módulos reutilizáveis.
6. **Diagrama de Implantação:** descreve a distribuição física de software e hardware.

## Situação-Problema

O cliente deseja acompanhar o progresso do planejamento do desenvolvimento do software. No entanto, não possui visibilidade clara das etapas já concluídas, das atividades em andamento e dos próximos passos. Diante disso, o gestor solicita que os desenvolvedores preparem e realizem uma apresentação para o cliente, com o objetivo de demonstrar o andamento do planejamento e alinhar expectativas.

# Modelagem Colaborativa com ClickUp

O ClickUp Brain é uma plataforma baseada em IA para centralização e gestão do conhecimento. Ele utiliza uma estrutura simples e integrada ao fluxo de trabalho, permitindo organizar informações, conectar documentos, automatizar tarefas e acessar dados de forma rápida. Além disso, possui o recurso de gerar diagramas para modelagem de Software.



Fonte: [Clickup Brain](#).

# Diagramas Estruturais – Pacote

O diagrama de pacotes organiza elementos de um sistema em grupos chamados pacotes, mostrando suas dependências. Esses pacotes podem conter classes, casos de uso ou até outros pacotes, permitindo estruturar o sistema de forma clara.

Funciona como pastas que agrupam partes relacionadas, facilitando o gerenciamento. Dessa forma, torna-se mais simples compreender a arquitetura geral e manter a organização do projeto durante o desenvolvimento de software.

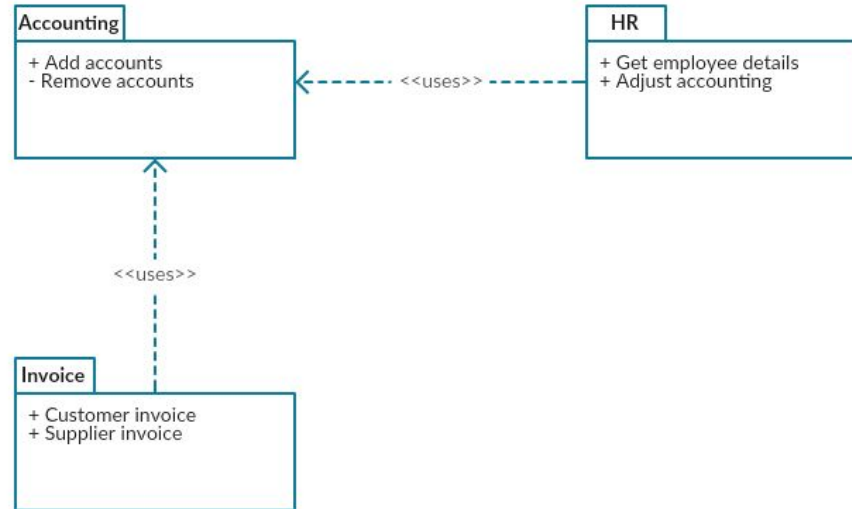
# Diagramas Estruturais – Pacote

**Sistema Bancário:** Pacotes agrupam módulos de contas, clientes e transações, organizando dependências e simplificando manutenção da arquitetura geral do software.

**E-commerce:** Pacotes organizam produtos, pagamentos e entregas, facilitando visão modular do sistema, reduzindo complexidade e melhorando colaboração entre desenvolvedores.

**Aplicativo Educacional:** Pacotes reúnem funcionalidades de cursos, usuários e avaliações, permitindo clara separação lógica e auxiliando evolução organizada do sistema.

# Diagramas Estruturais – Pacote





# Diagramas Estruturais – Componentes

Os diagramas de componentes representam a estrutura interna de uma classe, incluindo partes, interfaces e colaborações relevantes. Eles permitem observar como cada elemento se conecta e se comunica, oferecendo uma visão mais clara da composição interna e do papel de cada componente dentro do sistema.

Além disso, esses diagramas ajudam a identificar dependências entre diferentes elementos de software, destacando interações essenciais para o funcionamento. Essa visualização torna mais simples compreender a arquitetura geral, facilitando o planejamento, a documentação e a manutenção do sistema, além de apoiar equipes de desenvolvimento no trabalho colaborativo.

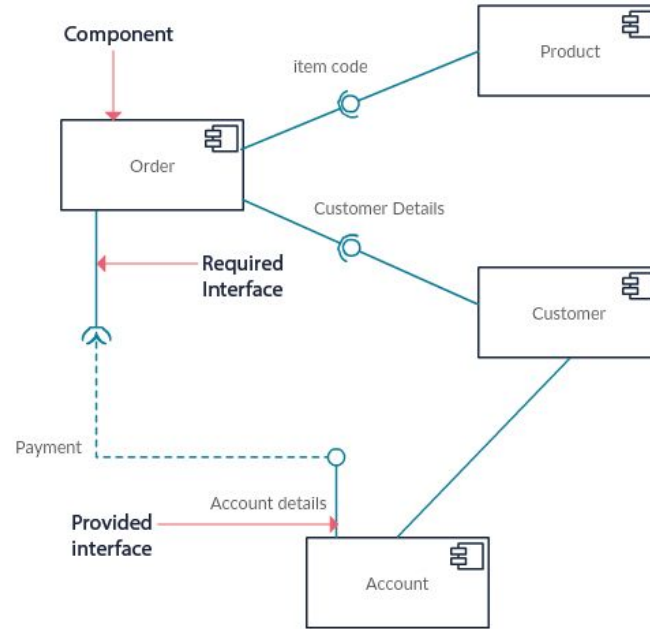
# Diagramas Estruturais – Componentes

**Sistema Bancário:** Componentes mostram módulos de autenticação, contas e relatórios, evidenciando interfaces usadas e dependências, facilitando integração e manutenção do sistema.

**Plataforma de Streaming:** Diagrama organiza componentes de cadastro, reprodução e recomendação, permitindo identificar conexões, otimizando escalabilidade e comunicação entre serviços.

**Sistema Hospitalar:** Componentes representam módulos de prontuário, agendamento e faturamento, destacando interações, apoiando integração segura e reduzindo erros em processos clínicos.

# Diagramas Estruturais – Componentes



# Diagramas Estruturais – Implantação

Os diagramas de implantação representam a distribuição física de componentes de software em diferentes nós de um sistema. Eles destacam a forma como aplicações, servidores e dispositivos se relacionam, permitindo entender claramente a infraestrutura necessária para suportar o funcionamento de um sistema distribuído de maneira eficiente e organizada.

Esses diagramas também mostram a interação entre software e hardware, fornecendo informações essenciais para planejar implantações em variados ambientes.

# Diagramas Estruturais – Implantação

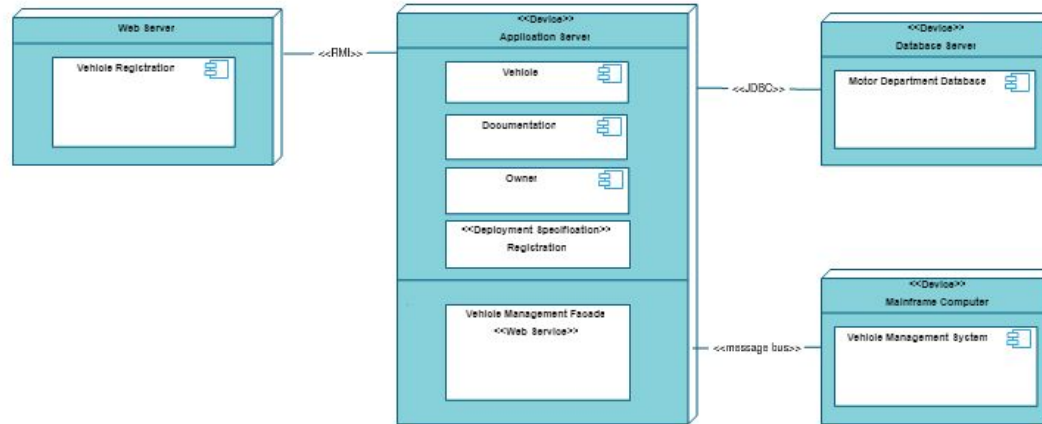
**E-commerce:** Diagrama mostra servidores web, banco de dados e dispositivos clientes, auxiliando planejamento da infraestrutura e garantindo escalabilidade para altas demandas.

**Sistema Hospitalar:** Implantação representa servidores de prontuário, terminais médicos e dispositivos móveis, assegurando comunicação segura e disponibilidade contínua dos serviços clínicos.

**Banco Digital:** Diagrama organiza nós de autenticação, processamento e relatórios, facilitando visualização da distribuição física e otimização do desempenho em operações.

# Diagramas Estruturais – Implantação

Deployment Diagram For a Vehicle Registration System



# Atividade

Nesta atividade, cada grupo deverá utilizar os requisitos ou User Stories definidos na atividade anterior e realizar a modelagem colaborativa utilizando a Mermaid ou ClickUp Brain. Cada grupo deverá elaborar 3 Diagramas de Pacotes, 3 Diagramas de Componentes e 3 Diagramas de Implantação, assegurando clareza, coerência e aplicação prática dos conceitos de modelagem UML.

# Referências

1. PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 8. ed. McGraw-Hill, 2016.
2. SOMMERVILLE, Ian. Engenharia de Software. 10. ed. Pearson, 2019.
3. LARMAN, Craig. Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos. 3. ed. Bookman, 2007.
4. PFLEEGER, Shari Lawrence; ATLEE, Joanne M. Engenharia de Software: teoria e prática. 4. ed. Pearson, 2010.
5. JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. UML: Guia do Usuário. 2. ed. Bookman, 2000.
6. KOTONYA, Gerald; SOMMERVILLE, Ian. Engenharia de Requisitos. 1. ed. LTC, 1998.
7. IEEE Computer Society. Guide to the Software Engineering Body of Knowledge (SWEBOK V3.0). IEEE, 2014.
8. ISO/IEC/IEEE. 29148:2018 – Systems and software engineering — Life cycle processes — Requirements engineering. ISO/IEC/IEEE, 2018.
9. Object Management Group (OMG). Unified Modeling Language (UML) Specification. Versão 2.5.1, 2017.



# **Aula 05 - Engenharia de Software II**

**Prof. Me. Deivison S. Takatu**

**[deivison.takatu@fatec.sp.gov.br](mailto:deivison.takatu@fatec.sp.gov.br)**