

Aula 01 - Engenharia de Software II

Prof. Me. Deivison S. Takatu

deivison.takatu@fatec.sp.gov.br

Apresentação da Turma

- O que acham do curso de ADS?
- Experiências profissionais na área?
- Expectativas após a conclusão do curso?
- Um hobby ou passatempo.



Histórico Acadêmico e Profissional

- Mestre em Ciência da Computação (2021);
 - Especialização em Psicopedagogia (2021);
 - Especialização em Informática Aplicada à Educação (2018);
 - Graduação em Análise e Desenvolvimento de Sistemas (2016).
-
- Gerente de Projetos (2023 - Atual);
 - Professor Universitário (2022 - Atual);
 - Coordenador Acadêmico (2019 - 2021);
 - Professor de Informática (2017 - 2023).

Áreas de Atuação

- Pesquisa em Educação Digital e Gamificação;
- Projetos em Desenvolvimento Web;
- Projetos de Educação Financeira.



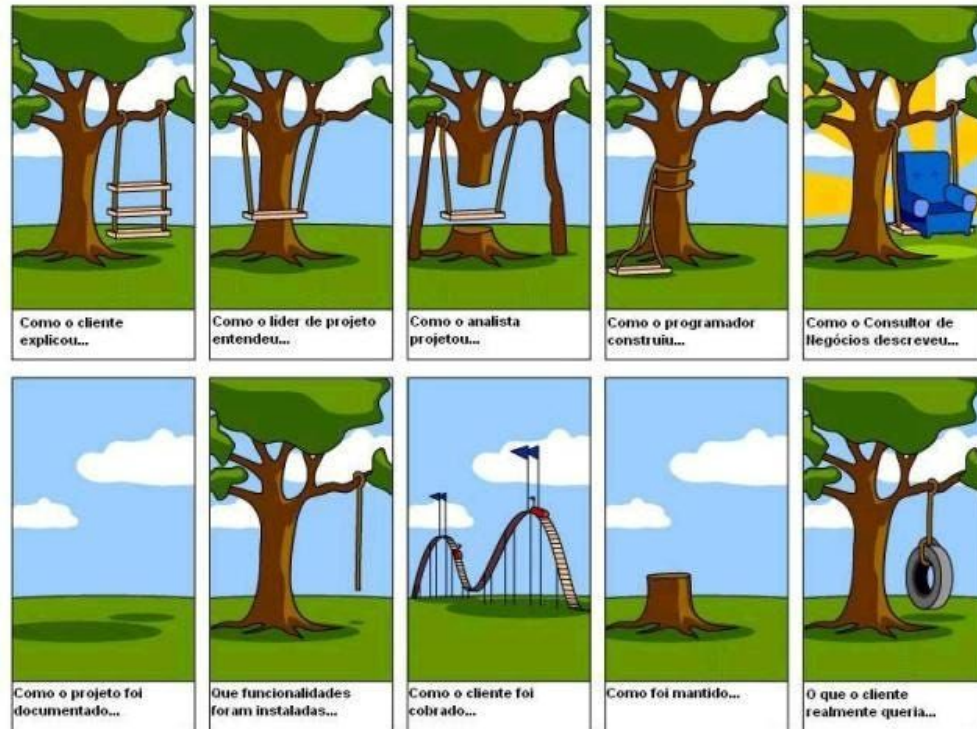
Plano de Ensino

- Ementa;
- Objetivo de aprendizagem;
- Metodologia de ensino;
- Critérios de Avaliação;
- Bibliografia;
- **Cronograma.**

Introdução a Engenharia de Software II

- O que é Engenharia de Software?
- Importância da Disciplina.
- Exemplos no dia a dia.
- Objetivo é produzir software de alta qualidade, dentro do prazo e do orçamento, atendendo às necessidades do cliente.

Introdução a Engenharia de Software II



Importância da Engenharia de Software

- Garante a qualidade e confiabilidade dos sistemas que sustentam operações críticas;
- Reduz custos de desenvolvimento e manutenção através de processos estruturados;
- Permite escalabilidade e adaptação dos sistemas às mudanças de requisitos.

Referência: PRESSMAN e MAXIM (2016).

Ciclo de Vida do Software

- **Análise de Requisitos;**
- **Planejamento e Design;**
- **Desenvolvimento;**
- **Testes;**
- **Sustentação.**



Fonte: [SoftDesign](#).

Engenharia de Requisitos

- São as descrições das características, funcionalidades e restrições que um sistema deve possuir para atender às necessidades do usuário.
- Representam a ponte entre o cliente e a equipe de desenvolvimento.
- A fase de engenharia de requisitos é crítica para o sucesso de um projeto.

Referência: KOTONYA e SOMMERVILLE (1998).

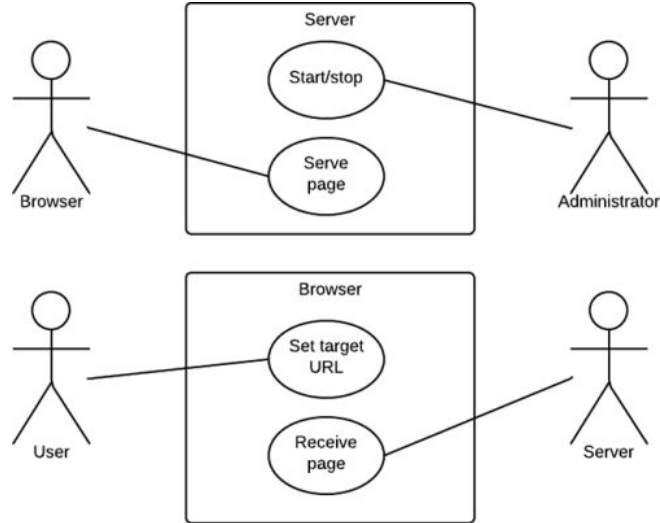
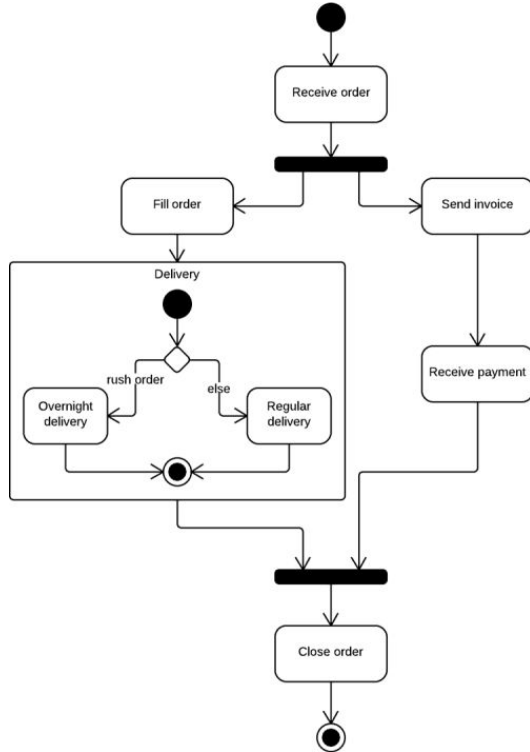
Engenharia de Requisitos

- **Requisitos Funcionais:** Descrevem o que o sistema deve fazer. São as funcionalidades do sistema, as tarefas que ele precisa executar.
 - Exemplos: "O sistema deve permitir o cadastro de novos usuários." ou "O sistema deve calcular o valor total do pedido."
- **Requisitos Não Funcionais:** Descrevem como o sistema deve se comportar. São as restrições e qualidades do sistema.
 - Exemplos: **Usabilidade:** "O sistema deve ter uma interface intuitiva."; **Desempenho:** "O sistema deve responder a uma requisição de busca em no máximo 3 segundos."

Linguagem de Modelagem Unificada (UML)

- UML (Unified Modeling Language) é uma linguagem gráfica padronizada para visualizar, especificar, construir e documentar artefatos de um sistema de software.
- Não é uma metodologia de desenvolvimento, mas uma ferramenta que pode ser usada em conjunto com diferentes processos.
- É a linguagem de fato para a modelagem orientada a objetos.

Linguagem de Modelagem Unificada (UML)



Linguagem de Modelagem Unificada (UML)

- **Diagramas Estruturais:** Mostram a arquitetura estática do sistema, os elementos que o compõem e suas relações.
- **Diagramas Comportamentais:** Descrevem o comportamento dinâmico do sistema, ou seja, como os objetos interagem e como o sistema responde a eventos.

Documentação e Padronização de Projetos (ABNT)

- A utilização de documentação garante que o conhecimento sobre o projeto não se perca.
- Serve como referência para a manutenção, evolução e futuras equipes de desenvolvimento.
- Reduz a ambiguidade e alinha a compreensão entre as partes interessadas (cliente, equipe, gerentes).

Critérios de Avaliação

- P1 – Prova 1;
- P2 – Prova 2;
- PJ – Projeto;
- AT – Atividades;
- **Nota Final** = $(P1 \times 0,25) + (P2 \times 0,25) + ((PJ + AT) \times 0,50)$.

Combinado Aula Sexta-Feira

- As aulas de Engenharia de Software II acontecem semanalmente às quartas-feiras e sextas-feiras.
- Quarta-feira no Laboratório 7 das 20:25 às 23:05
- Sexta-feira no Laboratório 3 das 22:15 às 23:05
- **Sugestão e combinado para as aulas de sexta.**

Atividade

- Formem grupos de 3 a 5 integrantes. **Importante:** Todas as atividades semanais deverão ser desenvolvidas e entregues em grupo, utilizando essa mesma composição. Além disso, o Projeto também será realizado em grupo, com os mesmos integrantes definidos nesta etapa.
- Após formarem o grupo, utilizem o tempo restante da aula para criar as contas nas ferramentas que serão utilizadas.

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 01 - Engenharia de Software II

Prof. Me. Deivison S. Takatu
deivison.takatu@fatec.sp.gov.br