📌

Engenharia de Software💻

👩🏾‍🏫**Professor:** Salatiel Luz Marinho

📩**e-mail:** salatiel.marinho@docente.unip.br

📜Referencial Teórico da Aula

✅ O que é Engenharia de Software?

Engenharia de software é a disciplina que envolve o desenvolvimento, operação e manutenção de software de forma sistemática e disciplinada. Ela abrange uma variedade de práticas e metodologias focadas em garantir a qualidade, eficiência e escalabilidade de sistemas de software, desde a concepção até a implementação e manutenção. A engenharia de software busca aplicar princípios de engenharia ao processo de criação de software, incluindo o uso de ferramentas, técnicas de gerenciamento de projetos, e melhores práticas para garantir que o software atenda aos requisitos dos usuários e seja confiável e seguro.

A engenharia de software lida com várias etapas do ciclo de vida do desenvolvimento de software, que geralmente incluem:

1. **Análise de Requisitos:** Coleta e definição das necessidades e expectativas dos usuários em relação ao software.
2. **Projeto de Software:** Planejamento arquitetônico e detalhado do sistema, especificando como o software será estruturado e como seus componentes irão interagir.
3. **Implementação (Codificação):** Tradução do projeto em código executável usando linguagens de programação adequadas.
4. **Testes:** Verificação e validação do software para garantir que ele funcione corretamente e atenda aos requisitos especificados.
5. **Manutenção:** Atualizações e correções no software após sua liberação para corrigir falhas, melhorar o desempenho ou adaptar-se a novas

necessidades.

6. **Gerenciamento de Configuração e Controle de Versão:** Monitoramento e controle de alterações no software para garantir integridade e rastreabilidade.
7. **Garantia de Qualidade:** Processos para garantir que o software atende aos padrões de qualidade exigidos.

Além dessas etapas, a engenharia de software também pode envolver práticas como gerenciamento de riscos, avaliação de desempenho, segurança, usabilidade e documentação. Os engenheiros de software utilizam diversas metodologias e práticas, como Agile, Scrum, DevOps, entre outras, para otimizar o processo de desenvolvimento e entrega de software.

📚 Exemplos Prático de Análise de Requisitos

Um exemplo de análise de requisitos pode ser ilustrado através do desenvolvimento de um sistema de gerenciamento de biblioteca. Neste exemplo, o processo de análise de requisitos envolve a coleta e definição das necessidades dos usuários, que podem incluir:

1. **Empréstimo de Livros:** O sistema deve permitir que os usuários façam o empréstimo de livros, registrando o usuário, a data de empréstimo e a data de devolução prevista.
2. **Devolução de Livros:** Deve ser possível registrar a devolução de livros no sistema, atualizando o status do livro para disponível.
3. **Catálogo de Livros:** O sistema deve possuir um catálogo onde os usuários possam pesquisar por título, autor ou gênero, verificando a disponibilidade dos livros.
4. **Cadastro de Usuários:** Deve existir um módulo para cadastro de novos usuários, que inclua informações como nome, endereço e contato.
5. **Notificações de Atraso:** O sistema deve enviar notificações automáticas para usuários que tenham livros em atraso, informando sobre as penalidades aplicáveis.
6. **Relatórios de Utilização:** Deve ser possível gerar relatórios sobre a utilização da biblioteca, como livros mais emprestados e usuários mais ativos.

Durante a análise de requisitos, os engenheiros de software colaboram com os stakeholders para assegurar que todos os requisitos sejam capturados com precisão e que o sistema final atenda às expectativas dos usuários.

📚 Exemplo Prático de Projeto de Software

No projeto de software, a fase de projeto envolve o planejamento arquitetônico e detalhado do sistema, especificando como o software será estruturado e como seus componentes irão interagir. Um exemplo prático pode ser o desenvolvimento de um sistema de gerenciamento de biblioteca.

1. **Arquitetura do Sistema:** Decida a arquitetura geral, por exemplo, uma arquitetura cliente-servidor, onde o servidor gerencia o banco de dados de livros e usuários, e o cliente é uma interface de usuário que interage com o servidor.
2. **Design de Componentes:** Identifique os principais componentes do sistema, como a interface do usuário, o módulo de gerenciamento de empréstimos, o módulo de cadastro de usuários e o módulo de notificações.
3. **Modelagem de Dados:** Crie diagramas de entidade-relacionamento para definir como os dados serão armazenados no banco de dados, incluindo tabelas para livros, usuários, empréstimos, etc.
4. **Interação entre Componentes:** Defina como os componentes irão se comunicar. Por exemplo, especificar APIs ou protocolos que o cliente usará para solicitar dados do servidor ou atualizar registros no banco de dados.
5. **Interface do Usuário:** Projete wireframes ou protótipos de baixa fidelidade para visualizar como a interface do usuário será, garantindo que seja intuitiva e atenda às necessidades dos usuários.
6. **Especificações Funcionais:** Detalhe as funcionalidades específicas que cada módulo do sistema deve ter, como a capacidade de buscar livros no catálogo ou enviar notificações automáticas para usuários com livros em atraso.

Este exemplo ilustra a importância de um projeto bem estruturado para garantir que o sistema final seja eficiente, escalável e atenda aos requisitos dos usuários.