

Atividade Avaliativa

Alunos:	Arthur Henrique Dai	Conceito:
Disciplina:	Engenharia de Software Turma: TADS-1A	
Professor:	Nelson Bellincanta Filho	

1 - Pesquise e liste cinco exemplos de softwares para cada uma das categorias abaixo.

Software de Sistema:

- a. Microsoft Windows;** Sistema operacional da Microsoft, amplamente utilizado em desktops e laptops.
- b. Linux (Ubuntu, Fedora, Debian);** Sistema operacional de código aberto, popular em servidores e dispositivos embarcados.
- c. macOS;** Sistema operacional da Apple, utilizado em computadores Mac.
- d. Android;** Sistema operacional baseado em Linux, utilizado em dispositivos móveis.
- e. iOS.** Sistema operacional da Apple para dispositivos móveis, como iPhones e iPads.

Software de Aplicação:

- a. Microsoft Office (Word, Excel, PowerPoint);** Pacote de aplicativos para criação de documentos, planilhas e apresentações.
- b. Adobe Photoshop;** Software de edição de imagens e design gráfico.
- c. Google Chrome;** Navegador de internet desenvolvido pelo Google.
- d. Slack;** Plataforma de comunicação e colaboração para equipes.
- e. Spotify.** Serviço de streaming de música.

Software de Engenharia/Científico:

- a. MATLAB;** Ambiente de desenvolvimento para cálculos numéricos e visualização de dados.
- b. AutoCAD;** Software de desenho assistido por computador (CAD) para projetos de engenharia e arquitetura.
- c. SolidWorks;** Software CAD para modelagem 3D e engenharia mecânica.
- d. ANSYS;** Software de simulação para análise de elementos finitos.
- e. LabVIEW.** Plataforma para desenvolvimento de sistemas de medição e controle.

Software Embarcado:

- a. FreeRTOS;** Sistema operacional em tempo real para microcontroladores.
- b. Zephyr;** Sistema operacional de código aberto para dispositivos embarcados.
- c. VxWorks;** Sistema operacional em tempo real utilizado em sistemas críticos.
- d. RIOT OS;** Sistema operacional para dispositivos da Internet das Coisas (IoT).
- e. ChibiOS.** Sistema operacional em tempo real para microcontroladores.

Software para Linha de Produtos:

- a. SAP ERP;** Sistema de planejamento de recursos empresariais para gestão de processos de negócios.
- b. Oracle JD Edwards;** Software ERP para gestão de manufatura e cadeia de suprimentos.
- c. Infor LN;** Solução ERP para empresas de manufatura complexa.
- d. Microsoft Dynamics 365;** Plataforma de aplicativos empresariais para gestão de operações.
- e. Epicor ERP.** Solução ERP para empresas de manufatura e distribuição.

Aplicações Web/Aplicativos Móveis:

- a. **WhatsApp**; Aplicativo de mensagens instantâneas.
- b. **Instagram**; Plataforma de compartilhamento de fotos e vídeos.
- c. **TikTok**; Aplicativo de compartilhamento de vídeos curtos.
- d. **Uber**; Aplicativo de transporte urbano.
- e. **Netflix**. Serviço de streaming de filmes e séries.

Software de Inteligência Artificial:

- a. **TensorFlow**; Biblioteca de código aberto para aprendizado de máquina.
- b. **PyTorch**; Biblioteca para aprendizado profundo e redes neurais.
- c. **IBM Watson**; Plataforma de IA para análise de dados e automação.
- d. **Google Cloud AI**; Serviços de IA e aprendizado de máquina na nuvem.
- e. **ChatGPT**. Plataforma de IA Online de uso geral.

2 - Neste exercício, você irá explorar e descrever o processo de software, bem como suas principais etapas.

a) Definição do Processo de Software:

R: O processo de software é um conjunto estruturado de atividades e tarefas que guiam o desenvolvimento de sistemas de software. É importante porque proporciona uma abordagem sistemática, garantindo a qualidade, eficiência e previsibilidade no desenvolvimento.

b) Especifique cada uma das quatro principais etapas do processo de software:

- Especificação de Software;
 - Define os requisitos e funcionalidades do sistema.
Exemplo: Uma empresa de e-commerce especifica que o sistema deve permitir aos usuários filtrar produtos por categoria e faixa de preço.
- Projeto e Implementação de Software;
 - Desenvolve a arquitetura e o código do sistema.
Exemplo: A equipe de desenvolvimento cria a interface de usuário e implementa a lógica de filtragem de produtos.
- Validação de Software;
 - Verifica se o sistema atende aos requisitos e funciona corretamente.
Exemplo: Os testadores executam testes de usabilidade e verificam se o login com autenticação de dois fatores está funcionando como esperado.
- Evolução de Software.
 - Realiza manutenção e melhorias no sistema após sua entrega.
Exemplo: Após o lançamento, o cliente solicita a adição de um gráfico com relatórios no painel, e a equipe atualiza o sistema para incluir essa nova funcionalidade.

c) Exemplos ou Cenários:

R: Uma empresa de redes sociais decide criar uma nova funcionalidade de "recomendações de amigos". A fase de especificação inclui reuniões com as partes interessadas, como os gerentes de produto e os usuários finais, para definir os requisitos, como a integração com a lista de amigos do usuário, algoritmos de recomendação, e a interface para exibição de sugestões.

d) Importância das Etapas:

R: A equipe de desenvolvimento começa a projetar a arquitetura da funcionalidade de "recomendações de amigos", definindo como o algoritmo de recomendação será implementado e como ele irá interagir com a base de dados. Eles criam wireframes para a interface de usuário e implementam o código para gerar as sugestões com base nas interações anteriores do usuário.

e) Desafios e Soluções:

R: Após a implementação, a equipe de testes começa a verificar se a funcionalidade de recomendação de amigos funciona corretamente. Eles testam diferentes cenários, como adicionar novos amigos, remover amigos e testar a exibição das sugestões. Também realizam testes de usabilidade para garantir que a funcionalidade seja fácil de entender e de utilização para os usuários.

3 - Neste exercício, você irá investigar e compreender os modelos de processo de software, que são estruturas fundamentais para organizar e gerenciar o desenvolvimento de software.

a) Definição dos Modelos de Processo de Software:

R: Modelos de processo de software são abordagens estruturadas que guiam o desenvolvimento de sistemas. Eles ajudam a organizar e gerenciar as atividades de desenvolvimento de forma eficiente, assegurando a qualidade do software e o cumprimento de prazos. Cada modelo possui características distintas que o tornam mais adequado a diferentes tipos de projetos.

b) Identificação de Exemplos:

R:

1. **Modelo Cascata (Waterfall):**

1.1. **Descrição:** O modelo Cascata é um dos modelos de desenvolvimento de software mais tradicionais, seguindo uma abordagem sequencial e linear. Em cada fase do desenvolvimento, uma etapa precisa ser concluída antes de passar para a próxima. As fases incluem: levantamento de requisitos, projeto, implementação, testes e manutenção.

1.2. **Exemplo:** O desenvolvimento de um sistema bancário, onde os requisitos são bem definidos no início e não mudam frequentemente ao longo do projeto.

2. **Modelo Incremental:**

2.1. **Descrição:** O modelo incremental divide o desenvolvimento em pequenas partes ou incrementos, onde cada incremento oferece uma funcionalidade funcional do sistema. A cada iteração, mais funcionalidades são adicionadas, e o sistema vai evoluindo progressivamente.

- 2.2. **Exemplo:** O desenvolvimento de um sistema de gestão de clientes (CRM) com entrega de funcionalidades incrementais, como cadastro de clientes, histórico de interações, relatórios, etc.

3. Modelo Ágil (Agile):

- 3.1. **Descrição:** O modelo Ágil enfatiza entregas rápidas, colaboração constante com o cliente e flexibilidade para mudanças nos requisitos. Ele usa ciclos curtos de desenvolvimento chamados "sprints", onde funcionalidades específicas são entregues em cada iteração.
- 3.2. **Exemplo:** Uma startup que está desenvolvendo um aplicativo de compartilhamento de fotos, onde os recursos como upload de fotos, filtros e integração com redes sociais são entregues em ciclos curtos e com feedback constante dos usuários.

4. Modelo Spiral:

- 4.1. **Descrição:** O modelo Spiral combina elementos do modelo Cascata e Incremental com uma forte ênfase na análise de riscos. O desenvolvimento é realizado em ciclos, e cada ciclo envolve o planejamento, a análise de riscos, a engenharia e a avaliação.
- 4.2. **Exemplo:** O desenvolvimento de um sistema de segurança bancária que requer constante avaliação de riscos de segurança e testes rigorosos de cada nova funcionalidade.

5. Modelo DevOps:

- 5.1. **Descrição:** O modelo DevOps é focado na integração contínua e entrega contínua, promovendo a colaboração entre as equipes de desenvolvimento e operações. O objetivo é acelerar o ciclo de vida do software e permitir que os recursos sejam entregues rapidamente e com alta qualidade.
- 5.2. **Exemplo:** Desenvolvimento de uma plataforma de e-commerce, onde as atualizações são feitas de forma contínua e os erros são corrigidos rapidamente com a colaboração entre desenvolvedores e a equipe de operações.

c) Comparação entre Modelos:

R:

1. Cascata (Waterfall)

- 1.1. **Vantagens:** Fácil de entender e gerenciar; cada fase é claramente definida.
- 1.2. **Desvantagens:** Pouca flexibilidade para mudanças; difícil de adaptar durante o desenvolvimento.

2. Incremental

- 2.1. **Vantagens:** Permite entregas rápidas e feedback contínuo; adapta-se bem a mudanças.
- 2.2. **Desvantagens:** Pode ser difícil de integrar cada incremento sem planejamento adequado.

3. **Ágil**

3.1. **Vantagens:** Flexibilidade para mudanças; foco no cliente e melhoria contínua.

3.2. **Desvantagens:** Pode ser difícil de gerenciar em grandes equipes ou projetos complexos.

4. **Spiral**

4.1. **Vantagens:** Gerenciamento eficaz de riscos; flexível e adaptável.

4.2. **Desvantagens:** Requer mais tempo e esforço devido aos ciclos de análise de riscos.

5. **DevOps**

5.1. **Vantagens:** Integração e entrega contínua; acelera o ciclo de vida do software.

5.2. **Desvantagens:** Requer ferramentas e práticas de automação avançadas; exige colaboração constante entre equipes.

d) Aplicação Prática:

R:

1. **Modelo Cascata:**

1.1. Ideal para projetos com requisitos bem definidos e sem mudanças constantes. Um exemplo seria o desenvolvimento de um sistema de controle de inventário para uma pequena empresa. Como as funcionalidades e requisitos são estáveis e bem definidos desde o início, o modelo Cascata é eficiente e direto, sem a necessidade de ciclos iterativos.

2. **Modelo Incremental:**

2.1. Um exemplo prático seria o desenvolvimento de um sistema de CRM (Customer Relationship Management). As primeiras versões podem incluir funcionalidades básicas de cadastro de clientes, enquanto os incrementos subsequentes adicionam recursos como relatórios, automação de marketing, e integração com redes sociais.

3. **Modelo Ágil:**

3.1. No desenvolvimento de um aplicativo móvel de finanças pessoais, a equipe pode trabalhar de forma ágil, entregando funcionalidades em sprints curtos de duas semanas. Cada sprint pode incluir novas funcionalidades, como notificações de gastos, categorização de despesas, e geração de relatórios, com base no feedback dos usuários.

4. **Modelo Spiral:**

4.1. O modelo Spiral seria ideal para o desenvolvimento de software para um sistema bancário. Como esse tipo de software lida com questões de segurança e conformidade regulatória, a análise de riscos contínuos e os ciclos de desenvolvimento iterativos ajudam a garantir que cada nova funcionalidade seja segura e eficiente.

5. **Modelo DevOps:**

5.1. O modelo DevOps seria perfeito para o desenvolvimento de um site de e-commerce, onde é necessário garantir alta disponibilidade e implementar atualizações frequentes sem interrupções. O ciclo contínuo de integração e entrega contínua permite atualizações rápidas e seguras, com a colaboração constante entre as equipes de desenvolvimento e operações.

e) Reflexão:

R: A escolha do modelo de processo de software adequado tem um grande impacto no sucesso do projeto. Ele determina como as equipes irão abordar o desenvolvimento, como as funcionalidades serão entregues e como os problemas serão resolvidos ao longo do caminho. A seleção do modelo ideal deve levar em consideração fatores como a estabilidade dos requisitos, a necessidade de flexibilidade, o tamanho da equipe e os recursos disponíveis. Por exemplo, um projeto com requisitos bem definidos e mudanças mínimas pode se beneficiar de um modelo Cascata, enquanto um projeto que exige flexibilidade e constante adaptação (como o desenvolvimento de um aplicativo de consumo) pode ser mais adequado ao modelo Ágil.

Ao refletir sobre diferentes projetos de software, fica claro que a escolha do modelo certo pode facilitar a comunicação entre as equipes, aumentar a satisfação do cliente e reduzir o risco de falhas no projeto. A habilidade de adaptar e combinar elementos de diferentes modelos de processo também pode ser uma vantagem em cenários mais complexos.

f) Exemplos de Modelos de Processo de Software:

1. Modelo Cascata (Waterfall)

1.1. Como descrito, esse modelo segue uma abordagem linear e sequencial, adequada para projetos com requisitos claros e estáveis.

2. Modelo incremental

2.1. Foca no desenvolvimento de versões parciais do sistema, permitindo entregas incrementais e a adaptação conforme o progresso.

3. Modelo Ágil (Agile):

3.1. Baseado em ciclos curtos de desenvolvimento e interação constante com o cliente, promove flexibilidade e adaptação rápida a mudanças.

4. Modelo Spiral:

4.1. Enfatiza a análise de riscos contínuos e a iteração, adequado para projetos de alto risco e complexidade.

5. Modelo DevOps:

5.1. Foca na integração contínua e entrega contínua, permitindo lançamentos rápidos e frequentes.