**Data: 16/08/2023**

**Nome Completo: Ian Pedro Passos de Santana**

**Atividade de Pesquisa sobre Fundamentos do Software**

**Objetivo:** O objetivo desta atividade de pesquisa é aprofundar a compreensão dos alunos sobre os fundamentos do software, incluindo suas definições, evolução, tipos e características.

**Avaliação:** Os alunos serão avaliados com base na qualidade e profundidade de suas pesquisas, bem como na capacidade de comunicar de forma clara e envolvente os conceitos fundamentais do software.

**Instruções:**

1. **Definições e Conceitos Básicos:**
   * Pesquise e explique o que é software em termos de computação.
   * Descreva a diferença entre hardware e software.
   * Defina os termos "programa" e "código-fonte".

**Resposta:**

* É uma sequência de instruções escritas para serem interpretadas pelo computador, para ele executar tarefas específicas. Podendo assim, o software ser considerado como os programas, dados e instruções que comandam o computador
* Hardware são os componentes físicos do computador e o software é o sistema que faz com que o hardware funcione
* Programa: é um conjunto de instruções que são usadas direta ou indiretamente por um computador, com o objetivo de obter um determinado resultado; Código fonte: conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes.

1. **Evolução do Software:**
   * Explore a evolução histórica do desenvolvimento de software, desde os primórdios até os dias atuais.
   * Destaque marcos importantes, como o surgimento das linguagens de programação, sistemas operacionais, etc.

**Resposta:**

* Em 1950 a 1960, os programas eram escritos diretamente em linguagem de máquina e já se tinha o desejo de simplificar e tornar mais eficiente o processo de desenvolvimento de software. Em 1980, o mundo testemunhou a popularização dos microcomputadores e interfaces gráficas, o que levou a ambientes de desenvolvimento mais amigáveis. Em 1990, a programação web ganhou destaque com o advento da World Wide Web e a proliferação de linguagens como JavaScript. A virada do século trouxe a computação em nuvem, permitindo escalabilidade e flexibilidade sem precedentes. Em 2020, a inteligência artificial, aprendizado de máquina e automação se tornaram componentes-chave do desenvolvimento de software

1. **Tipos de Software:**
   * Explique os principais tipos de software: sistema, aplicativo e software embutido.
   * Diferencie software de sistema e software de aplicativo.
   * Forneça exemplos de software embutido em dispositivos comuns.

**Resposta:**

* Software de sistema: é o software projetado para fornecer uma plataforma para outro software; Software de aplicativo: são os programas que nós usamos no computador que permite ao usuário executar algum tipo de tarefa, seja ela qual for; Software embutido: é um software que é especifico pra um hardware que tem algum tipo de restrição de memória e processamento devido aos recursos de computação limitada do dispositivo, além de que também esse software controla máquinas que geralmente não são consideradas um computador
* O software de sistema é aquele que oferece uma plataforma para outros softwares executarem suas próprias tarefas, alguns exemplos de software de sistema são: Windows, macOS e Linux. Software de aplicativo é aquele que executa os programas no computador, por exemplo: Word, Excel, Paint, Bloco de notas e calculadora.
* Alguns exemplos de software embutido são: Smartphones, roteadores, impressoras, veículos automotores etc.

1. **Características do Software:**
   * Liste e explique as características essenciais do software, como portabilidade, usabilidade, manutenibilidade, etc.
   * Discuta a importância da modularidade e da reusabilidade no desenvolvimento de software.

**Resposta:**

* Algumas características essenciais do software são: A portabilidade, que é a capacidade do software de ser executado em diferentes plataformas e sistemas operacionais sem a necessidade de grandes modificações; A usabilidade, se relaciona com a facilidade de uso do software por parte dos usuários finais; A manutenibilidade, que é a facilidade com que o software pode ser mantido e atualizado; A eficiência, se refere ao desempenho do software em relação ao consumo de recursos, como memória e processamento; A Confiabilidade, é a capacidade do software de funcionar de maneira consistente e sem falhas;
* A Modularidade facilita a organização, colaboração e manutenção do código. A reutilização economiza tempo, melhora a qualidade e promove eficiência no desenvolvimento de software. Ou seja, são práticas indispensáveis no desenvolvimento do software Elas permitem que os desenvolvedores construam sistemas mais robustos e flexíveis, ao mesmo tempo em que economizam tempo e recursos valiosos.

1. **Desenvolvimento de Software:**
   * Descreva o ciclo de vida típico do desenvolvimento de software, incluindo etapas como análise, design, codificação, testes e implantação.
   * Introduza a ideia de metodologias ágeis e a abordagem waterfall (cascata).

**Resposta:**

* O ciclo de vida típico do desenvolvimento de software começa pela análise de requisitos, nesse primeira parte os requisitos do software são coletados, no entanto, para isso precisa-se entender as necessidades, funcionalidades desejadas e as restrições do projeto. O próximo passo é o desing, que com base nos requisitos, o projeto de software é criado e Isso inclui o design da arquitetura, estrutura de dados, interfaces de usuário e outros componentes do sistema. Agora a codificação, nesta fase os desenvolvedores escrevem o código fonte do software com base nas especificações de design. Logo após a codificação vem a parte dos testes, onde o software é submetido a testes rigorosos como testes de unidade, testes de integração e testes de sistema. Após os testes bem-sucedidos vem a parte da implantação onde o software é aprimorado em um ambiente de produção que envolve a instalação em servidores, a configuração de ambientes e a disponibilização para os usuários finais.
* Metodologias Ágeis: é desenvolvimento de software que enfatiza a flexibilidade, colaboração e entrega incremental, permitindo ajustes ao longo do projeto com foco na satisfação do cliente. Algumas das metodologias ágeis mais conhecidas são Scrum, Kanban, Extreme Programming (XP) e Lean Development. Abordagem waterfall (cascata): é um método de desenvolvimento sequencial, onde cada fase (como análise, design, codificação, teste) é concluída antes de avançar para a próxima. É mais utilizado em projetos com requisitos estáveis ​​e bem definidos.

1. **Linguagens de Programação:**
   * Explique o que são linguagens de programação e qual é o papel delas no desenvolvimento de software.
   * Liste e explique algumas linguagens de programação populares e suas aplicações.

**Resposta:**

* **Linguagens de programação são conjuntos de instruções que os programadores usam para criar softwares e programas. Elas fornecem uma maneira de comunicar com o computador, definindo como as tarefas devem ser executadas. As linguagens de programação permitem que os desenvolvedores criem algoritmos, estruturas de dados e lógica de software. As linguagens de programação desempenham um papel fundamental no desenvolvimento de software, servindo como a base para criar aplicativos, programas e sistemas que atendem a diversas necessidades, desde tarefas simples até projetos complexos**
* **Java Script: é uma linguagem de programação padrão para desenvolvimento web**
* **aplicações: desenvolvimento web, criação de aplicativos móveis, desenvolvimentos de jogos**

**Java: é conhecido por sua portabilidade e segurança. É usado em uma variedade de aplicações, desde sistemas empresariais até aplicativos móveis**

* **Aplicações: desenvolvimento web, desenvolvimento de aplicativos móveis, sistemas embarcados, aplicações empresariais**

**Python: é conhecido por sua sintaxe legível e facilidade de aprendizado. É usado em uma ampla variedade de campos devido à sua versatilidade e à grande quantidade de bibliotecas disponíveis**

* **Aplicações: Desenvolvimento web, ciência de dados, automação, aprendizado de máquina, desenvolvimento de jogos**

**C++: é valorizado por seu alto desempenho e é frequentemente usado em aplicações que exigem controle de hardware ou recursos de baixo nível**

* **Aplicações: Desenvolvimento de jogos, software de sistema, sistemas operacionais, aplicativos de alto desempenho**

**PHP: uma linguagem popular para desenvolvimento web e é usada para criar páginas da web interativas e dinâmicas**

* **Desenvolvimento web, criação de sites dinâmicos, sistemas de gerenciamento de conteúdo como WordPress**

**C#: é uma linguagem da Microsoft usada principalmente no desenvolvimento de aplicativos Windows e jogos**

* **Aplicações: Desenvolvimento de jogos, desenvolvimento de aplicativos Windows, aplicativos empresariais, aplicativos para dispositivos móveis**

1. **Open Source vs. Proprietário:**
   * Compare o software de código aberto (open source) e o software proprietário.
   * Discuta os benefícios e desafios associados a ambos os modelos.

**Resposta:**

* Software de Código Aberto: O código-fonte do software de código aberto é disponibilizado para o público em geral. Qualquer pessoa pode visualizar, modificar e distribuir o código, desde que siga as licenças de código aberto; Software Proprietário: O código-fonte do software proprietário é mantido em segredo pela empresa ou desenvolvedor que o cria. Os usuários não têm acesso ao código-fonte. A escolha entre software de código aberto e software proprietário depende das necessidades específicas do projeto, dos recursos disponíveis e dos incentivados da organização
* Benefícios do código aberto: o custo, transparência e a flexibilidade; desafios do código aberto: o suporte, complexidade e a compatibilidade. Benefícios do código proprietário: suporte e manutenção garantidos, facilidade do uso, segurança e a integração com tecnologias comerciais; desafios do código proprietário: custo de licenciamento, falta de transparência, falta de flexibilidade e o lok-in do fornecedor.

1. **Desafios do Desenvolvimento de Software:**
   * Pesquise e identifique os principais desafios enfrentados no desenvolvimento de software, como gerenciamento de requisitos, testes, segurança, etc.

**Resposta:**

* O desenvolvimento de software enfrenta diversos desafios, que podem variar dependendo do projeto. Por exemplo: No gerenciamento de requisitos, você tem que lidar com mudanças nos requisitos e garantir que as alterações sejam controladas e bem comunicadas; No teste, você tem que Planejar e executar testes abrangentes para garantir que o software funcione conforme o esperado; Na segurança, você tem que proteger o software contra ameaças de segurança, como ataques de hackers, invasões e vazamento de dados.

1. **Tendências Futuras:**
   * Pesquise sobre as tendências atuais e futuras no desenvolvimento de software, como inteligência artificial, aprendizado de máquina, realidade virtual, etc.

**Resposta:**

* O desenvolvimento de software está em constante evolução devido às rápidas mudanças tecnológicas, alguns exemplos são: Inteligência Artificial (IA) e Aprendizado de Máquina (AM), Desenvolvimento Low-Code/No-Code, Realidade Virtual (RV) e Realidade Aumentada (RA), Computação de borda, Blockchain, Computação Quântica, 5G e Conectividade Avançada.