**CENTRO UNIVERSITÁRIO BELAS ARTES DE SÃO PAULO**

**GRADUAÇÃO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - EAD**

**FELIPE SCHAITEL - 24112424**

**ENGENHARIA DE SOFTWARE - ATIVIDADE 01**

**FLORIANÓPOLIS, SC**

**2025**

TRABALHO – N1

QUESTÕES

1. Valor (0,50 ponto) – Explique a importância dos constructos de programação na criação de software. Dê exemplos de como variáveis, operadores e estruturas de controle (como condicionais e loops) são utilizados para resolver problemas específicos no desenvolvimento de um programa.

Os constructos de programação são importantes na arquitetura de software e na codificação. Os construtos são estruturas essenciais para o desenvolvimento de um programa. São, por exemplo, as variáveis, estrutura de dados, operadores, condicionais, repetições, funções, procedimento etc.

Segundo Robert C. Martin no livro Arquitetura Limpa, define que esses elementos básicos do código (os constructos de programação) são utilizados há muito tempo. Mesmo com as novas tecnologias e paradigmas, essas estruturas essenciais continuam sendo a mesma lógica e forma de codificar. O autor afirma que ele usa esses elementos e declarações desde 1964 e essa lógica é assim até os dias atuais aplicada em diferentes tipos de sistemas, linguagens e paradigmas

Outra característica do software atual em comparação com o anterior: ele continua sendo feito com os mesmos elementos. É formado por declarações if, declarações de atribuição e laços while (...) Quando observa bem de perto a prática de programar computadores, você percebe que muito pouco mudou em 50 anos. As linguagens ficaram um pouco melhores e as ferramentas, fantasticamente melhores. Mas os blocos de construção básicos de um programa de computador não mudaram.

(MARTIN, p.36)

Esse é um dos motivos pelo qual os constructos de programação são tão consistentes e importantes na criação e interoperabilidade de um Software. Nos exemplos de função a seguir, é possível notar os constructos presentes em três diferentes linguagens.

| Constructo de Programação - Função | | |
| --- | --- | --- |
| Python | JavaScript | PHP |
| def soma(a, b):  return a + b  resultado = soma(3, 5)  print("A soma é:", resultado) | function soma(a, b) {  return a + b;  }  const resultado = soma(3, 5);  console.log("A soma é:", resultado); | <?php  function soma($a, $b) {  return $a + $b;  }  $resultado = soma(3, 5);  echo "A soma é: " . $resultado;  ?> |

É possível perceber que mesmo sendo sintaxes diferentes de escrever o código, sua lógica de receber, processar e saída de dados são a mesma.

2. Valor (0,50 ponto) – Pesquise sobre uma linguagem de programação de sua escolha (que não seja Java). Explique brevemente suas principais características, incluindo a classificação da linguagem (compilada ou interpretada), e como ela pode ser utilizada em um ambiente de programação (IDE) específica.

A linguagem escolhida Python, é uma linguagem de programação multiparadigma que suporta diferentes estilos de programação, incluindo programação orientada a objetos, programação imperativa e programação funcional. A sintaxe de Python é simples, limpa e legível, com estruturas de controle intuitivas. Alguns exemplos de programas famosos escritos em Python incluem o backend do Instagram, Netflix, vários projetos do Google, incluindo YouTube e Google Search. Ou seja, Python é uma linguagem interpretada mas também pode ser compilada em bytecode.

O principal ambiente de programação (IDE) é o Pycharm da JetBrains.

Python é empregado no desenvolvimento web, por meio de frameworks como Django e Flask. É amplamente utilizado em ciência de dados e análise de dados e também é aplicado na automação de tarefas, simplificando processos em sistemas de TI e administração de servidores. No desenvolvimento de jogos, Python é utilizado em conjunto com bibliotecas como Pygame e Panda3D. Por fim, na área de inteligência artificial e aprendizado de máquina, frameworks como TensorFlow e PyTorch são populares para desenvolver modelos de IA e ML.

3. Valor (1,50 ponto) – Você foi contratado para desenvolver um sistema de controle de estoque para uma pequena empresa. Utilizando o método ADIT, descreva como você conduziria o processo de desenvolvimento desse software, detalhando as atividades de Análise, Design, Implementação e Testes.

1. Analise: Através da comunicação, reuniões, entrevistas e pesquisas com os stakeholders a fim de compreender as personas e entender suas necessidades, os requisitos, que são importantes para desenvolver as funcionalidades do sistema. Coletar e documentar esses requisitos.
2. Design: Os requisitos que foram coletados serão transformados em uma arquitetura de sistema. A modelagem escolhida para criar e definir a interface do sistema seria wireframe e mockup. E a programação em camadas pois é um sistema simples de pequeno porte. É importante levar em consideração que pessoas de diferentes idades utilizarão o sistema do estoque, por isso, as funções da aplicação precisam se comunicar bem e de forma acessível.
3. Implementação: Codificar em camadas. Na fase de construir utilizaria o Python para gerar o código fonte seguindo o design proposto. O python é uma linguagem de alto nível e versátil por ser multiparadigmas, dinâmica além de ter um bom acervo de bibliotecas.
4. Testes: Teste de integração: testar cada módulo para garantir que todos estejam funcionando corretamente. Os testes seriam feitos com o setor de QA e também com usuários escolhidos especialmente para testar o protótipo.

4. Valor (1,50 ponto) – Uma empresa deseja desenvolver um aplicativo móvel para **gerenciar as tarefas diárias dos seus funcionários**. O aplicativo deve **permitir que os funcionários registrem suas atividades, consultem relatórios, e se comuniquem via chat interno.** Com base nesse cenário, responda:

a) Quais seriam os principais requisitos a serem identificados na fase de Análise?

1. Requisitos funcionais:

* O sistema deve permitir que o usuário faça o login utilizando código de funcionário e senha;
* O aplicativo deve permitir escrever suas atividades diárias;
* O aplicativo deve permitir que os funcionários pesquisar e consultar relatórios;
* O aplicativo deve permitir os funcionários de se comunicar (escrever, enviar e ler) via chat do app;
* O aplicativo deve mostrar na tela notificações de mensagens ou tarefas pendentes;
* O aplicativo deve ter espaço de suporte e relatar erros/bugs.

1. Requisitos não funcionais:

* O sistema deve ser acessível em dispositivos móveis (celular, tablet etc);
* O tempo de carregamento da página deve ser inferior a 1 minuto;
* O aplicativo deve ser capaz de suportar até 1.000 funcionários;

b) Que tipos de aplicações e sistemas computacionais seriam necessários para dar suporte a esse aplicativo?

As aplicações necessárias seriam ambiente de desenvolvimento (IDE) Pycharm para back-end e camada de negócios, API para integrar o back-end ao front, Code SandBox para desenvolver a camada de apresentação Front-end (HTML,CSS e JS).

c) Como a utilização de Inteligência Artificial (IA) e Internet das Coisas (IoT) poderia ser incorporada para melhorar as funcionalidades desse aplicativo?

A Internet das coisas (IoT) pode ajudar a integrar o ambiente e facilitar o registro assim como a consulta. Desenvolver a aplicação para Tablets e nas prateleiras inteligentes.

**Referências de leituras**

**1 MARTIN, Robert C.** Arquitetura limpa: o guia do artesão para estrutura e design de software. Tradução de Alexandre Huchel. São Paulo: Starlin Alta Editora e Consultoria Eireli, 2020. ISBN 978-85-508-1600-5. Disponível em:<https://github.com/KAYOKG/BibliotecaDev/blob/main/.github/img/Arquitetura%20Limpa%20-%20O%20Guia%20do%20Artes%C3%A3o%20para%20Estrutura%20e%20Design%20de%20Software.svg>. Acesso em: [20/02/2025].

2. Consulte os materiais adicionados em cada aula.

3. PRESSMAN, Roger S.; MAXIM, Bruce. Engenharia de software. 9a edição. São Paulo: AMGH, 2021

4. SOMMERVILLE, I. Engenharia de software. 10. ed. São Paulo: Pearson Prentice Hall, 2019.