



INSTITUTO INFNET
ESCOLA SUPERIOR DA TECNOLOGIA DA INFORMAÇÃO
CURSO DE ENGENHARIA DE SOFTWARES

Francisco Alves Camello Neto

TP 1 - Fundamentos do Desenvolvimento Javascript

Prof. Daniel de Oliveira

**Brasília
2022**

SUMÁRIO

1ª - QUESTÃO.....	3
2ª - QUESTÃO.....	3
3ª - QUESTÃO.....	4
4ª - QUESTÃO.....	4
5ª - QUESTÃO.....	5
6ª - QUESTÃO.....	6
PRATICA DE LÓGICA	6
REFERÊNCIAS	8

1ª - QUESTÃO

Qual a diferença de uma linguagem script server-side e uma linguagem client-side. Cite um exemplo de cada.

Resposta: A linguagem de cliente ou client-side scripting, é uma linguagem que é executada no lado cliente, ou seja, no computador do próprio usuário, e por isso é usada nas situações em que a linguagem server-side não tem alcance. Entre as linguagens client-side, há o JavaScript, que é a única que realmente roda no navegador do usuário.

A linguagem de servidor, ou Server-side scripting, é a linguagem que vai rodar "por trás dos panos", fornecendo a lógica principal da aplicação. Funciona da seguinte maneira: sempre que o usuário faz um HTTP request (entra numa página, clica num link, etc), o pedido é enviado para o servidor. A linguagem server-side recebe o requerimento (Request) e faz o processamento. Depois, transforma o resultado final em um XHTML e envia para o navegador. É a linguagem server-side que vai verificar se o usuário está logado, vai buscar informações no banco de dados etc.

2ª - QUESTÃO

Como funciona a arquitetura web de 3 camadas? Explique o funcionamento de cada uma das camadas. Cite, pelo menos, um exemplo de tecnologia ou aplicação que se utiliza em cada camada.?

Resposta: A arquitetura de três camadas é uma arquitetura de aplicativo de software bem estabelecida que organiza aplicativos em três camadas de computação lógica e física: a camada de apresentação ou interface do usuário; a camada do aplicativo, onde os dados são processados; e a camada de dados, em que os dados associados ao aplicativo são armazenados e gerenciados.

Camada de apresentação

A camada de apresentação é a interface do usuário e a camada de comunicação do aplicativo, em que o usuário final interage com o aplicativo. Sua principal finalidade é exibir e coletar informações do usuário. Essa camada de nível superior pode ser executada em um navegador da web, como um aplicativo para desktop ou em uma interface gráfica com o usuário (GUI), por exemplo. As camadas de apresentação da web são geralmente desenvolvidas usando HTML, CSS e JavaScript. Os aplicativos para desktop podem ser escritos em uma variedade de linguagens, dependendo da plataforma.

Camada do aplicativo

A camada do aplicativo, também conhecida como a camada lógica ou a camada do meio, é o coração do aplicativo. Nesta camada, as informações coletadas na camada de apresentação são processadas - por vezes em relação a outras informações na camada de dados - usando a lógica de negócios, um conjunto específico de regras de negócio. A camada do aplicativo também pode incluir, excluir ou modificar dados na camada de dados.

A camada do aplicativo é tipicamente desenvolvida usando Python, Java, Perl, PHP ou Ruby e se comunica com a camada de dados usando chamadas API.

Camada de dados

A camada de dados, por vezes chamada de camada de banco de dados, camada de acesso a dados ou back-end, é onde as informações processadas pelo aplicativo são armazenadas e gerenciadas. Este pode ser um sistema de gerenciamento de banco de dados relacional, como PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix ou Microsoft SQL Server ou em um servidor de banco de dados NoSQL, como Cassandra, CouchDB ou MongoDB.

3ª - QUESTÃO

Porque estudar lógica de programação? Qual sua influência no desenvolvimento de software?

Resposta: A lógica é a ciência que expõe o conhecimento científico e nos fornece ferramentas para validar fatos e teorias, possibilitando assim distinguir o raciocínio correto do incorreto. Sendo assim, quando aplicada à programação manifesta sua importância ao disponibilizar recursos para executar o mais básico dos processos no desenvolvimento de uma aplicação: o algoritmo.

4ª - QUESTÃO

Explique usando 2 exemplos diferentes de pseudocódigo o conceito de repetições e de condicionais. Qual o propósito de cada uma?

Resposta: Trata-se de uma forma de executar blocos de comandos somente sob determinadas condições, mas com a opção de repetir o mesmo bloco quantas vezes for necessário. As estruturas de repetição são úteis, por exemplo, para repetir uma série de operações semelhantes que são executadas para todos os elementos de uma lista ou de uma tabela de dados, ou simplesmente para repetir um mesmo processamento até que uma certa condição seja satisfeita.

*PARA $i < \text{valorInicial}$ ATE valorFinal FAÇA
primeiroComando*

5ª - QUESTÃO

Cite e explique os 4 passos para resolução de problemas computáveis.

Resposta:

Decomposição

É o processo que divide os problemas em partes menores para facilitar a resolução, desenvolvimento e gerenciamento. Analisa os problemas para identificar as partes que podem ser separadas e formas como podem ser reconstituídas para solucionar um problema como um todo. Possibilita resolver problemas complexos de forma mais simples e permite projetar sistemas de grande porte.

Reconhecimento de Padrões

Os padrões são similaridades ou características que alguns problemas compartilham. O hábito de identificar padrões nos acompanha desde a infância, é uma construção continuada e o nosso repertório de padrões não para de crescer e de se reconstruir. O processo de reconhecimento de padrões permite encontrar similaridades ou padrões entre pequenos problemas decompostos. Quanto mais padrões encontrarmos, mais fácil e rápida será a nossa tarefa geral de solução de problemas.

Abstração

Consiste na filtragem e classificação dos dados, criando mecanismos que permitam separar apenas os elementos essenciais em determinado problema, ignorando detalhes irrelevantes. Permite criar uma representação, uma ideia do que está se tentando resolver. Neste processo, essencial é escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido sem perder nenhuma informação que seja importante para tal.

Algoritmo

É uma sequência finita de etapas (passos), cada qual executável em um tempo finito, por um agente computacional, natural (humano) ou sintético (computador). Um algoritmo é um plano, uma estratégia ou um conjunto de instruções ordenadas para a solução de um problema ou execução de uma tarefa. A formulação de um algoritmo passa pelo processo de decomposição, reconhecimento de padrões e abstração. Na execução seguirão os passos pré-definidos, não havendo a necessidade de criar um novo algoritmo para cada uma de suas execuções posteriores. É o pilar que agrega os demais pilares.

6ª - QUESTÃO

Explique a importância das funções para a lógica de programação e a escrita de algoritmos.

Resposta: Caso um algoritmo tenha um objetivo muito bem definido é melhor encapsulá-lo em um bloco de modo a poder invocá-lo sempre que precisar desse algoritmo. Geralmente todas as linguagens de programação possibilitam esse encapsulamento e isso é feito na forma de uma função ou de um procedimento. A grande vantagem dessa abordagem é deixar o seu código melhor organizado, além de permitir que a função seja utilizada em outros trechos do código sem a necessidade de codificá-la novamente.

PRÁTICA DE LÓGICA

Exercícios

<https://studio.code.org/s/20-hour>

Certificado

<https://code.org/v2/hoc/certificate/eyJhbnRnJhbmNpc2NvIEFsdmVzIENhbWVsbG8gTmV0byIsImNvdXJzZSI6IkFjY2VsZXJhdGVkIENvdXJzZSJ9.jpg>

Certificate of Completion

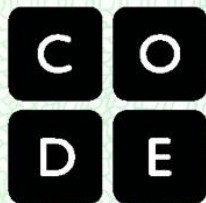
This certificate is awarded to

Francisco Alves Camello Neto

for successful completion of

Accelerated Course

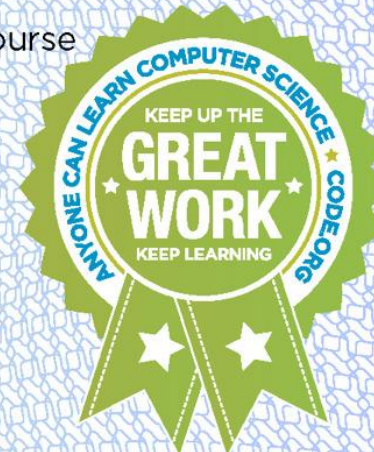
To see the computer science concepts learned in this course
and for more learning opportunities, please visit
<http://studio.code.org>



www.code.org

Hadi Partovi

Hadi Partovi, Co-founder and Chief Executive Officer, Code.org



Ballmer Group made the generous gift to sponsor your learning.

REFERÊNCIAS

EDUCATION, Ibm Cloud. **Arquitetura de três camadas (tiers)**. 2020. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/three-tier-architecture>. Acesso em: 10 maio 2022.

WIKIPEDIA. **Linguagem client-side**. 2018. Disponível em: https://pt.wikipedia.org/wiki/Linguagem_client-side. Acesso em: 10 maio 2022.

WIKIPEDIA. **Linguagem server-side**. 2018. Disponível em: https://pt.wikipedia.org/wiki/Linguagem_server-side. Acesso em: 10 maio 2022.

WIKIVERSIDADE. **Pensamento Computacional/Conceitos e Pilares do Pensamento Computacional**. 2020. Disponível em: https://pt.wikiversity.org/wiki/Pensamento_Computacional/Conceitos_e_Pilares_do_Pensamento_Computacional#Pilares_do_pensamento_computacional. Acesso em: 10 maio 2022.