****

**ESTI – ESCOLA SUPERIOR DA TECNOLOGIA DA INFORMAÇÃO**

**GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**BLOCO DESENVOLVIMENTO FRONT-END**

**JAVASCRIPT**

**TESTE DE PERFORMANCE 1**

Simon Assagra Barbosa Araújo Teixeira

Prof: Daniel de Oliveira

Goiânia - GO, 21 de Fevereiro de 2022.

**SUMÁRIO**

1. Desenvolvimento.....................................................................................................3
2. Referências Bibliográficas.......................................................................................9

**DESENVOLVIMENTO**

**QUESTÃO 1**

***Qual a diferença de uma linguagem script server-side e uma linguagem cliente-side. Cite alguns exemplos de cada.***

|  |  |
| --- | --- |
| **CLIENT-SIDE / FRONT-END** | **SERVER-SIDE / BACK-END** |
| São as linguagens que processam as informações pelo navegador. | São as linguagens que processam as informações pelo servidor e enviam uma resposta ao navegador. |
| O responsável pelo processo de carregar e renderizar o front-end do website, é o navegador. | Implementa arquiteturas robustas que realizam uma comunicação com o banco de dados de maneira segura ao usuário. |
| É a parte visual de um site. | É a parte não visual de um site. |
| O front-end pode ser desde imagens e textos até animações complexas. | O back-end pode ser desde a segurança e a estrutura até o gerenciamento de conteúdo. |
| É composto pelas principais linguagens HTML, CSS e JavaScript. | É composto pelas principais linguagens Python, C, C#, C++, Ruby, Java e PHP |

**QUESTÃO 2**

***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.***

A arquitetura de 3 camadas é a mais predominante para aplicações tradicionais de cliente-side, ou seja, ela divide aplicativos em 3 camadas, através, de computação lógica e física. Tais como:

|  |  |  |
| --- | --- | --- |
| **CAMADA DE APRESENTAÇÃO** | **CAMADA DO APLICATIVO** | **CAMADA DE DADOS** |
| Interação entre a interface e o usuário final. | Coração da aplicação. | Processamento de dados. |
| Coleta informações do usuário, sendo executada via web, aplicação desktop ou interface. | Processa as informações da camada de apresentação, por meio, de regras de negócio. Além disso, pode ter a função de executar um CRUD ou consumir uma API, com a camada de dados. | Processa, armazena e gerencia os dados recebidos pelas camadas anteriores. |
| Desenvolvida em HTML, CSS e JavaScript. | Desenvolvida em Python, C, C#, C++, Ruby, Java e PHP. | Gerenciada por bancos de dados relacionais (PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix ou Microsoft SQL Server) ou por bancos de dados não relacionais (Cassandra, CouchDB ou MongoDB). |

**QUESTÃO 3**

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

É importante estudá-la, pois ela possibilita que possamos executar tarefas simples e básicas no desenvolvimento de algum software, tal como os algoritmos. Para o desenvolvimento de software, ela acaba por nos beneficiar, tornando-nos melhores em:

* **Organização** – ter ambientes mais organizados e que auxiliam na produtividade.
* **Raciocínio lógico** – ter códigos mais limpos e estruturados.
* **Resolução de problemas** – dividir um problema grande e complexo em partes menores.
* **Concentração** – ter mais clareza, ordenar e executar suas tarefas de forma mais concentrada.
* **Compreensão das tecnologias** – ter mais facilidade ao aprender uma nova tecnologia, pois a lógica é a mesma, mudando, apenas sua sintaxe.

**QUESTÃO 4**

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

|  |  |
| --- | --- |
| **ESTRUTURAS DE REPETIÇÃO** | **ESTRUTURAS CONDICIONAIS** |
| Usadas para repetir um trecho de código até que satisfaça uma condição. | Usadas para tomadas de decisões, onde caso uma condição seja atendida será realizada uma ação no código. |
| * **WHILE:**   INICIE o número como 1;  ENQUANTO, (o número for menor ou igual a 10) o repita;  IMPRIMA (o número);  INCREMENTE ao número mais um; | * **IF e ELSE IF:**   SE (média for maior ou igual a 70);  IMPRIMA (“Aluno aprovado”);  SE NÃO (média for maior ou igual a 40 e média for menor ou igual a 69);  IMPRIMA (“Aluno em recuperação”); |
| * **FOR:**   PARA (INICIE o número como 1; CASO SEJA menor ou igual a 10; INCREMENTE ao número mais um);  IMPRIMA (o número); | * **SWITCH CASE:**   SE dia é igual a 3;  INTERROMPA (o dia);  PARA CASO 1: IMPRIMA (“Domingo”) e PARE;  PARA CASO 2: IMPRIMA (“Segunda-feira”) e PARE;  PARA CASO 3: IMPRIMA (“Terça-feira”) e PARE;  PADRÃO: IMPRIMA (“Outro dia”) e PARE; |

**QUESTÃO 5**

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

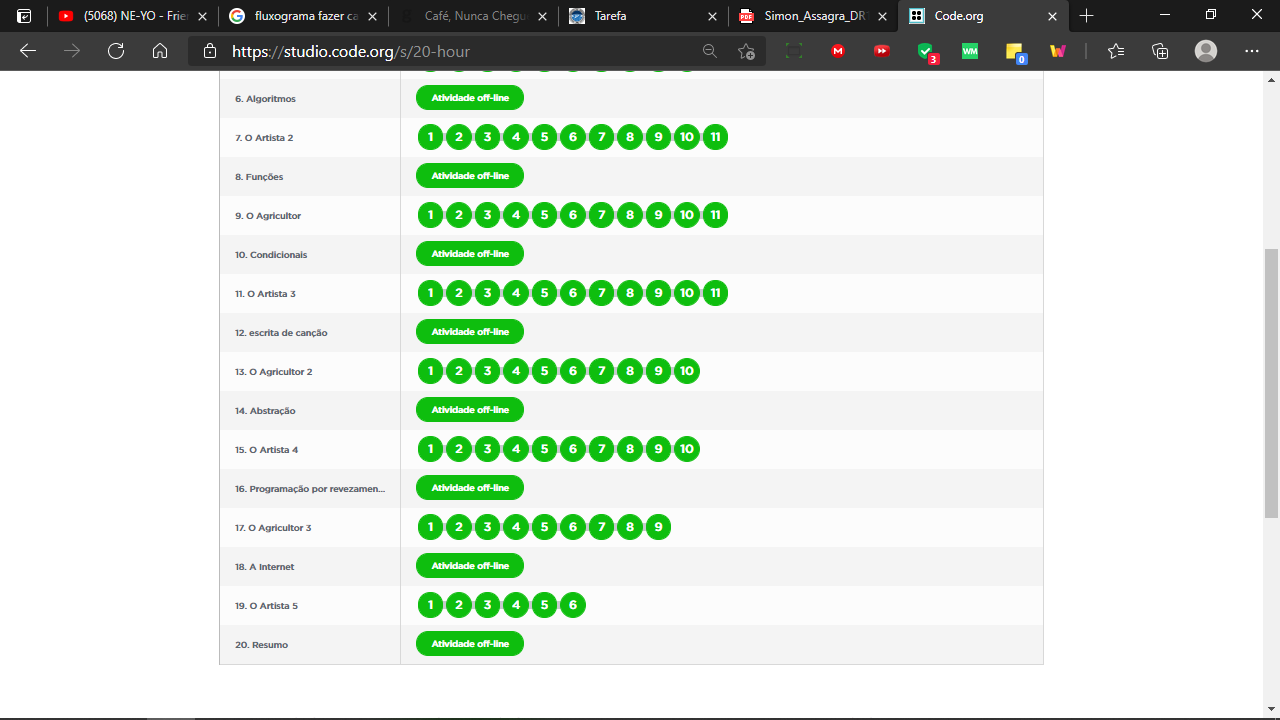
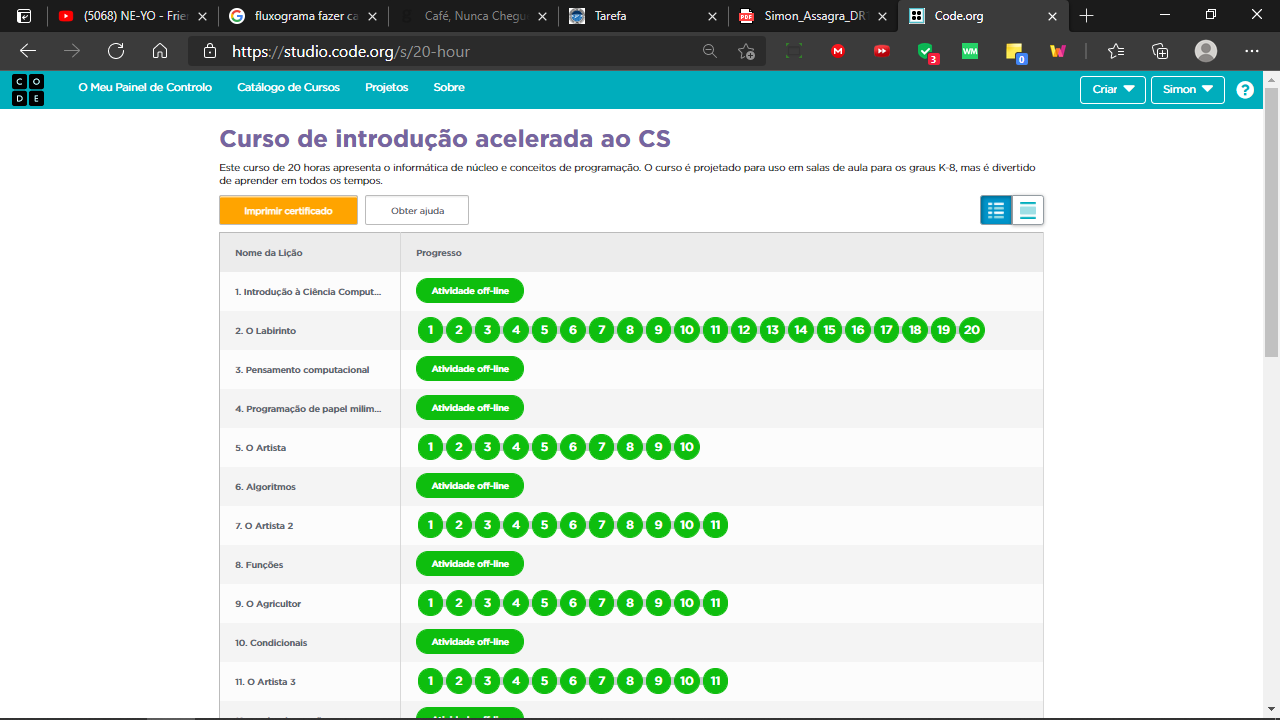
1. **Compreensão do problema** – conhecê-lo, identificar com clareza, reunir o que se sabe e quebrá-lo em parte menores para sua resolução.
2. **Conhecer os dados do problema** –analisar o que foi fornecido e complementar com mais alguns dados.
3. **Criar um método** –desenvolver um algoritmo para solucioná-lo.
4. **Otimizar o método** – deixar o código mais limpo e simples;
5. **Escrever o código** –transferir o algoritmo para alguma linguagem de programação e assim, criando o código fonte.
6. **Compilar** – usar alguma IDE para compilar a linguagem de máquina traduzida.
7. **Testar** – executar o programa para entender as possíveis soluções e erros, caso haja algum erro, retomar os passos acima.

**QUESTÃO 6**

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

São conhecidas como sub-rotinas, na qual deixam o código menos repetitivo e a leitura mais intuitiva. Logo, possuem um papel importante por:

* Simplificar o desenvolvimento (implementando um trecho menor de código).
* Organizar (implementando um código mais limpo e bem estruturado).
* Reaproveitar (poder reutilizar um código já implementado).

****

****

**REFERÊNCIAS BIBLIOGRÁFICAS**

TABLELESS – **O que é client-side e server-side? Diferenças entre linguagem client-side e linguagem server-side**. [Internet] Disponível em: [**https://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html**](https://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html)**.**

Acesso em: 21/02/2022

IBM Cloud Education – **Arquitetura de três camadas (tiers), 2020**. [Internet] Disponível em: [**https://www.ibm.com/br-pt/cloud/learn/three-tier-architecture#toc-o-que--arq-erz4oSL\_**](https://www.ibm.com/br-pt/cloud/learn/three-tier-architecture#toc-o-que--arq-erz4oSL_)**.**

Acesso em: 21/02/2022

Kenzie – **Lógica de programação: o que é e por que é importante**. [Internet] Disponível em: [**https://kenzie.com.br/blog/logica-de-programacao/**](https://kenzie.com.br/blog/logica-de-programacao/)**.**

Acesso em: 21/02/2022

Zero e Umas – **Estruturas de repetição para iniciantes, 2020**. [Internet] Disponível em: [**https://medium.com/zeroeumas/estruturas-de-repeticao-8e07d23a92c2**](https://medium.com/zeroeumas/estruturas-de-repeticao-8e07d23a92c2)**.**

Acesso em: 21/02/2022

Zero e Umas – **Estruturas de condição, 2020**. [Internet] Disponível em: [**https://medium.com/zeroeumas/estruturas-de-condicao-2edd300d54ac**](https://medium.com/zeroeumas/estruturas-de-condicao-2edd300d54ac)**.**

Acesso em: 21/02/2022

Dicas de Programação – **O que são Funções e Procedimentos?**. [Internet] Disponível em: [**https://dicasdeprogramacao.com.br/o-que-sao-funcoes-e-procedimentos/**](https://dicasdeprogramacao.com.br/o-que-sao-funcoes-e-procedimentos/)**.**

Acesso em: 21/02/2022