

Data: 26/08/2024

Nome: André Luiz Santos

E-mail: and_bas7@hotmail.com

Github: <https://github.com/DehLuizSantos>

Início do teste: 18 : 38

Término do teste: ____ : ____

Prova v20200424 – Tempo total (1 hora e 30 minutos)

Parte I – Questões de múltipla escolha.

	1	2	3	4	5
a					
b	X			X	
c			X		
d					
e		X			X

Parte II – Algoritmos

Parte III – Questão dissertativa

Parte IV – Questões sobre você

Parte I – Questões de múltipla escolha. (sugestão: 30 minutos)

1) Avalie o seguinte código:

```
1. int fn(int v)
2. {
3.   if(v==1 || v==0)
4.     return 1;
5.   if(v%2==0)
6.     return fn(v/2)+2;
7.   else
8.     return fn(v-1)+3;
9. }
```

Qual o resultado esperado de **fn(7)**?

- a) 10
- b) 11
- c) 1
- d) 9
- e) 7

2) Considere as seguintes afirmações sobre expressões regulares:

- I. O padrão **a{3}b?c*** ocorre na sequência de caracteres **aaab**.
- II. O padrão **a*b** ocorre na sequência de caracteres **b**.
- III. O padrão **a{1,3}b?c*** ocorre na sequência de caracteres **aaab**.

Analisando as afirmações se conclui que:

- a) Apenas as afirmações I e II são verdadeiras.
- b) Apenas as afirmações II e III são verdadeiras.
- c) Apenas a afirmação I é verdadeira.
- d) Apenas as afirmações I e III são verdadeiras.
- e) Todas as afirmações são verdadeiras.

3) Qual das seguintes estruturas de dados é, em média, a mais rápida para recuperar um item de dados escolhido aleatoriamente?

- a) Binary Tree
- b) Stack
- c) Hash Table
- d) Queue
- e) Linked List

- 4) Considere um algoritmo que possa ser solucionado em diferentes ordens de complexidade. Qual das opções abaixo representa a correta ordenação do mais eficiente (mais a esquerda) para o menos eficiente (mais a direita):
- a) $O(1)$; $O(n)$; $O(n \log n)$; $O(\log n)$; $O(n^2)$; $O(n!)$; $O(2^n)$
 - b) $O(1)$; $O(\log n)$; $O(n)$; $O(n \log n)$; $O(n^2)$; $O(2^n)$; $O(n!)$
 - c) $O(n!)$; $O(2^n)$; $O(n^2)$; $O(n \log n)$; $O(n)$; $O(\log n)$; $O(1)$
 - d) $O(1)$; $O(n)$; $O(n \log n)$; $O(\log n)$; $O(n^n)$; $O(2^n)$; $O(n!)$
 - e) $O(n!)$; $O(2^n)$; $O(n^2)$; $O(\log n)$; $O(n \log n)$; $O(n)$; $O(1)$
- 5) Considere as tabelas a seguir, criadas em um banco de dados relacional através da linguagem SQL.

```
1) CREATE TABLE Funcionario
2) ( fcod int PRIMARY KEY,
3)   nome varchar (32),
4)   salario number (7,2),
5)   dcod int FOREIGN KEY REFERENCES Diretoria (dcod));
6) CREATE TABLE Diretoria
7) ( dcod int PRIMARY KEY,
8)   dnome varchar (12),
9)   chefe int FOREIGN KEY REFERENCES Funcionario (fcod));
```

Sejam as consultas (C1, C2 e C3) também em SQL, a seguir.

```
C1. SELECT nome, salario FROM Funcionario F, Diretoria D
WHERE F.dcod = D.dcod AND F.fcod = D.chefe;
C2. SELECT nome, salario FROM Funcionario as F INNER JOIN
Diretoria as D ON F.dcod = D.dcod WHERE F.fcod = D.chefe;
C3. SELECT nome, salario FROM F.fcod = D.chefe;
```

- a) Apenas a consulta C1 retorna o nome e o salário dos chefes das diretorias.
- b) Apenas a consulta C2 retorna o nome e o salário dos chefes das diretorias.
- c) Apenas a consulta C3 retorna o nome e o salário dos chefes das diretorias.
- d) As consultas C1, C2 e C3 são equivalentes e retornam o nome e o salário dos chefes das diretorias.
- e) As consultas C1 e C2 são equivalentes e retornam o nome e o salário dos chefes das diretorias.

Parte II – Algoritmos – resolver as duas questões

- 1) Desenvolva um algoritmo que, para cada número inteiro dentro do intervalo fechado entre 0 e 16, imprima:
- "foo" se o número for divisível por 3
 - "bar" se o número for divisível por 5
 - "baz" se o número for divisível por 3 e por 5
 - o próprio número caso contrário

Resposta:

```
function checkNumbers() {  
  for (let i = 0; i <= 16; i++) {  
    let output = "";  
  
    if (i % 3 === 0) output += 'foo';  
    if (i % 5 === 0) output += 'bar';  
  
    console.log(output || i);  
  }  
}  
  
checkNumbers();
```

- 2) Implemente um algoritmo que receba como entrada uma sequência de Strings – que podem ou não apresentar repetições – e imprima, uma única vez para cada termo, a quantidade de vezes em que o mesmo foi encontrado.

Obs: a formatação da saída é livre.

Ex:

Entrada: ["PaTiNeTe", "SKATE", "Patinete", "Bicicleta"]

Saída:

```
skate=1
patinete=2
bicicleta=1
```

Resposta:

```
function countStringValue(strings) {

  if (!Array.isArray(strings) || !strings.every(item => typeof item === 'string')) {
    console.error('Entrada inválida. Por favor, forneça um array de strings.');
```

```
    return;
  }

  const counts = new Map();

  strings.forEach(str => {
    const normalizedStr = str.toLowerCase();
    counts.set(normalizedStr, (counts.get(normalizedStr) || 0) + 1);
  });

  for (const [key, value] of counts.entries()) {
    console.log(` ${key}=${value}`);
  }
}

// Exemplo de uso
const input = ["PaTiNeTe", "SKATE", "Patinete", "Bicicleta"];
countStringValue(input);
```

Parte III – Questão de arquitetura – escolha um item

Faça um diagrama de arquitetura de sistemas que represente **apenas um** dos casos de uso abaixo:

- a. Efetuar uma compra em um site de e-commerce com a opção de boleto.
- b. Gerar um dashboard de número de acessos - em tempo real - por categoria (ex: política, esporte, economia) feitos em um portal de notícias.

Requisitos:

- Especificar os componentes e sistemas externos em alto nível, indicando a funcionalidade de cada um e a maneira (protocolo) como comunicam entre si.
- Indicar e justificar a escolha de infraestrutura da solução (cloud, on premise ou mista).

Obs: caso esteja respondendo a prova no computador, utilize a ferramenta <https://draw.io>

O desenho da arquitetura está em anexo junto com a prova.

Sobre a arquitetura escolhida (a):

Optei por uma arquitetura de micro serviços, que possui ao todo 7 serviços próprios, e 2 serviços de um gateway de pagamentos (IUGU).

O deploy seria na AWS, gerenciando containers em Docker e Kubernetes.

Como APPs, teremos:

1) Site do ecommerce em NextJs, para otimizar o SEO e diminuir a latência com ferramentas de Server-side Rendering. Ele será conectado a um serviço de login e uma API de Produtos.

2) Admin do ecommerce em React, que será conectado a retornos da Gateway de pagamento, e a uma API para preencher os produtos e categorias cadastradas no site. Ele possuirá um dashboard com informações sobre vendas, e um cadastro de produtos/categorias/úsuários.

Como servidores, teremos:

1) API de navegação (conectada com o site), ela retornará dados referente ao ecommerce, e estará conectada com um SQL de produtos/categorias.

2) API de segurança (conectada com o ecommerce e o gateway de pagamento). Ela também terá seu próprio banco de dados, e terá uma segurança reforçada.

3) API Admin: (conectada com o admin, e compartilhando o SQL da API de navegação). Ela faz o crud de serviços para o ecommerce, como produtos e categorias.

Os bancos de dados serão em SQL.

Parte IV – Questões sobre você

1) Como você adquire novos conhecimentos? Quais suas fontes de estudo?

Geralmente, quando preciso implementar alguma funcionalidade ou realizar algo de forma rápida, vou direto para a documentação da tecnologia utilizada. Quando preciso adquirir um conhecimento mais amplo, como aprender uma nova linguagem ou um framework, faço um curso completo na Udemy e, em seguida, tento realizar algum projeto para praticar. Fora do trabalho, estou fazendo uma pós-graduação em Negócios Digitais na USP e tenho interesse em fazer uma pós-graduação em Data Science ou Arquitetura de Software depois desta.

2) Qual livro técnico você recomendaria a alguém? Por quê?

Para um guia técnico, recomendo o Guia Definitivo de JavaScript, que é o mais completo e abrange tanto conceitos avançados quanto frameworks.

Para um livro menos técnico, mas que oferece um conhecimento abrangente sobre a área, sugiro Código Limpo. Este livro ajuda a gerenciar sua experiência e a adquirir bons hábitos na programação.