

# TESTE TEÓRICO

PROFESSIONAL SERVICES



## Instruções

Preencha as informações com os seus dados e você poderá responder a este teste neste mesmo arquivo.

## Importante

Pedimos a você que responda às questões com o seu verdadeiro conhecimento, a prova pode ser respondida com consulta, porém ao responder com consulta é importante bom senso em suas respostas para expressar o seu entendimento e conhecimento sobre os assuntos propostos.

## O que será avaliado

Todas as questões serão avaliadas, sendo que algumas questões têm maior pontuação do que outras.

**Desejamos uma boa sorte!**

Nome do Candidato(a)	
Telefone	
Linkedin	
Data	

# Javascript

1. Qual é o operador lógico usado para verificar a negação de uma expressão? (Nota: 0,2)

- a) &&
- b) ||
- c) !
- d) ==

2. Qual dos seguintes métodos é usado para adicionar um elemento ao final de um array? (Nota: 0,2)

- a) push()
- b) pop()
- c) shift()
- d) unshift()

3. O que o método "Array.map()" faz? (Nota: 0,2)

- a) Remove o último elemento de um array.
- b) Mapeia os elementos de um array para um novo array com base em uma função de mapeamento.
- c) Filtra os elementos de um array com base em uma função de filtro.
- d) Inverte a ordem dos elementos em um array.

4. Qual é a função do método "Array.filter()"? (Nota: 0,2)

- a) Adicionar elementos ao início do array.
- b) Remover elementos do array com base em uma função de filtro.
- c) Transformar os elementos do array em uma string concatenada.
- d) Ordenar o array em ordem alfabética.

5. O que é async/await em JavaScript? (Nota: 0,2)

- a) Um método para criar funções síncronas em JavaScript.
- b) Uma técnica para manipulação de erros em operações assíncronas.
- c) Um conjunto de palavras-chave que tornam as funções assíncronas mais legíveis e fáceis de usar.
- d) Uma biblioteca JavaScript para criar animações e transições suaves.

6. Qual é a sintaxe correta para definir uma função assíncrona chamada "getData"? (Nota: 0,2)

- a) async getData() { return new Promise({}); }
- b) getData() { return new Promise({}); }
- c) async function getData() { return new Promise({}); }
- d) async function getData() => new Promise({});

7. O que será impresso no código abaixo? (Nota: 0,6)

```
let palavra = "ABC";  
switch (palavra)
```

```
{  
    case "ACB":  
        console.log("C");  
        break;  
    case "BC":  
    case "ABC":  
        console.log("A");  
        break;  
    case "B":  
        console.log("Hello");  
        break;  
    default:  
        console.log("Palavra não encontrada");  
        break;  
}
```

- a) C.
- b) A.
- c) Hello.
- d) Palavra não encontrada

**8. Escreva uma função em JavaScript chamada “somImpares” que recebe um número inteiro positivo “n” como parâmetro e retorna a soma de todos os números ímpares de 1 até n. (Nota: 0,6)**

**Exemplo:**

```
somImpares(5); // Saída esperada: 9 (1 + 3 + 5)  
somImpares(10); // Saída esperada: 25 (1 + 3 + 5 + 7 + 9)
```

9. Escreva uma função chamada "inverterPalavra" que recebe uma string como parâmetro e retorna a string com as letras invertidas. (Nota: 0,6)

```
inverterPalavra("javascript"); // Saída esperada: "tpircsavaj"
```

10. Considere o seguinte trecho de código em JavaScript que tenta realizar a divisão de dois números: (Nota: 0,6)

```
function dividirNumeros(number1, number2) {  
  try  
  {  
    if (number2 === 0)  
    {  
      throw new Error("Divisão por zero não é permitida.");  
    }  
  
    return number1 / number2;  
  }  
  catch (error)  
  {  
    return "Erro: " + error.message;  
  }  
}
```

Escreva abaixo o resultado retornado por cada função:

a)  

```
console.log(dividirNumeros(20, 2));
```

b)  

```
console.log(dividirNumeros(6, 0));
```

c)  

```
console.log(dividirNumeros(21, 3));
```

**11. Como você pode percorrer e mapear um array JSON em JavaScript? Explique como usar métodos como "map", "forEach" ou "for...of" para iterar e manipular os elementos do array. (Nota: 0,7)**

**12. O que são variáveis em JavaScript? Explique como declarar e atribuir valores a uma variável. (Nota: 0,7)**

**13. Em JavaScript, é possível ter múltiplas condições em uma estrutura "if/else"? Descreva como usar operadores lógicos (como "&&" e "||") para combinar condições. (Nota: 0,6)**

**14. Descreva a sintaxe do bloco "try" em JavaScript. Dê um exemplo prático de como usar o "try" para envolver um código suscetível a erros. (Nota: 0,7)**

**15. Como você pode lançar manualmente uma exceção em JavaScript? Explique o uso da palavra-chave "throw" para criar e lançar exceções personalizadas. (Nota: 0,7)**

## SQL

**1. Como você seleciona todas as colunas de uma tabela em SQL? (Nota: 0,2)**

- a) SELECT ALL
- b) SELECT \*
- c) SELECT COLUMNS
- d) SELECT FULL

**2. Qual é o comando SQL utilizado para filtrar resultados em uma consulta? (Nota: 0,2)**

- a) LIMIT
- b) FILTER
- c) ORDER BY
- d) WHERE

**3. Qual é o comando SQL utilizado para ordenar os resultados de uma consulta em ordem ascendente?** (Nota: 0,2)

- a) ORDER ASC
- b) SORT ASC
- c) ASCENDING
- d) ORDER BY

**4. Qual é o comando SQL utilizado para inserir novos dados em uma tabela?** (Nota: 0,2)

- a) ADD
- b) INSERT
- c) UPDATE
- d) CREATE

**5. Qual é o comando SQL utilizado para atualizar dados em uma tabela?** (Nota: 0,2)

- a) MODIFY
- b) UPDATE
- c) ALTER
- d) SET

## Integração de sistemas

**1. O que é integração de sistemas?** (Nota: 0,2)

- a) É um processo de comunicação entre diferentes sistemas de computador para permitir o compartilhamento de dados e funcionalidades.
- b) É um processo de integração de hardware e software em um único sistema de computador.
- c) É um processo de otimização do desempenho de um único sistema de computador.
- d) É um processo de criação de sistemas de computador a partir do zero.

**2. O que significa API (Interface de Programação de Aplicativos) em integração de sistemas?** (Nota: 0,2)

- a) Uma Arquitetura de Programação de Aplicativos que define os padrões de codificação.
- b) Uma linguagem de programação usada para criar aplicativos.
- c) Um conjunto de funções e procedimentos que permitem a comunicação entre sistemas.
- d) Um tipo de sistema de gerenciamento de banco de dados.



**3. O que é um Web Service? (Nota: 0,2)**

- a) É um serviço fornecido por uma empresa de hospedagem de sites.
- b) É um serviço web que permite a interação com um banco de dados online.
- c) É uma solução para conectar sistemas diferentes via web, usando padrões como XML e SOAP.
- d) É um serviço que permite a interação com aplicativos móveis.

**4. O que é um token de acesso em integração de sistemas? (Nota: 0,2)**

- a) Um código secreto usado para acessar uma rede privada.
- b) Um arquivo de configuração usado para conectar sistemas diferentes.
- c) Uma chave de autenticação usada para autorizar o acesso a um serviço.
- d) Um identificador único para um arquivo de dados em um sistema.

**5. O que é um "webhook" na integração de sistemas? (Nota: 0,2)**

- a) É uma ferramenta usada para rastrear o tráfego da web em sistemas corporativos.
- b) É uma interface gráfica usada para projetar páginas da web.
- c) É uma API que permite a integração de diferentes aplicativos.
- d) É uma URL pública fornecida por um sistema para receber notificações automáticas de outro sistema.

**6. O que é JSON? (Nota: 0,2)**

- a) Uma linguagem de programação.
- b) Um protocolo de comunicação entre servidores.
- c) Um formato de dados leve e de fácil leitura usado para trocar informações entre sistemas.
- d) Um método de autenticação e autorização em APIs.

**7. Qual é o código de status HTTP que indica sucesso na solicitação? (Nota: 0,2)**

- a) 200 OK.
- b) 404 Not Found.
- c) 500 Internal Server Error.
- d) 302 Found.

**8. O que são headers HTTP? (Nota: 0,2)**

- a) Conteúdo HTML dos sites.
- b) Informações adicionais enviadas pelo cliente e servidor em uma solicitação ou resposta HTTP.
- c) Arquivos de configuração do servidor web.
- d) Nomes de domínio registrados.

**9. Quais são os delimitadores usados para marcar tags em XML? (Nota: 0,2)**

- a) [ ]
- b) { }
- c) ( )
- d) < >

**10. Qual é a diferença entre integração de sistemas síncrona e assíncrona? (Nota: 0,2)**

- a) Na síncrona, a comunicação ocorre em tempo real com respostas imediatas, enquanto na assíncrona, a resposta pode ser recebida em um momento posterior.
- b) A integração síncrona só pode ocorrer entre sistemas da mesma empresa, enquanto a assíncrona permite a comunicação entre empresas diferentes.
- c) A integração síncrona ocorre apenas em situações de migração de servidores, enquanto a assíncrona é utilizada em situações de integração de dados.
- d) A integração síncrona é mais rápida e eficiente do que a assíncrona.

## Desafio

Deverá desenvolver uma API em node.js usando o javascript.

Seu desafio é criar uma API simples para gerenciar os pedidos. A API deve permitir a criação, leitura, atualização e exclusão de pedidos.

Criar endpoints para as seguintes operações:

- Criar um novo pedido. (Obrigatório).  
URL: `http://localhost:3000/order`
- Obter os dados do pedido passando por parâmetro na URL o número do pedido. (Obrigatório)  
URL: `http://localhost:3000/order/v10089016vdb`
- Listar todos os pedidos. (Opcional)  
URL: `http://localhost:3000/order/list`
- Atualizar o pedido passando por parâmetro na url o número do pedido que será atualizado. (Opcional)  
URL: `http://localhost:3000/order/v10089016vdb`
- Delete o pedido passando por parâmetro na url o número do pedido que será deletado.. (Opcional)  
URL: `http://localhost:3000/order/v10089016vdb`

Armazenar os dados dos pedidos em um banco de dados (Mongodb, SQL ou PostgreSql).

A API que criará o pedido no banco de dados receberá no body o json abaixo:

```
{  
  "numeroPedido": "v10089015vdb-01",  
  "valorTotal": 10000,  
  "dataCriacao": "2023-07-19T12:24:11.5299601+00:00",  
}
```

```
"items": [  
  {  
    "idItem": "2434",  
    "quantidadeItem": 1,  
    "valorItem": 1000  
  }  
]
```

Exemplo request:

```
curl --location 'http://localhost:3000/order' \  
--header 'Content-Type: application/json' \  
--data '{  
  "numeroPedido": "v10089015vdb-01",  
  "valorTotal": 10000,  
  "dataCriacao": "2023-07-19T12:24:11.5299601+00:00",  
  "items": [  
    {  
      "idItem": "2434",  
      "quantidadeItem": 1,  
      "valorItem": 1000  
    }  
  ]  
}'
```

Esta API deverá sofrer uma transformação dos dados, ou seja, deverá fazer o mapping dos campos para salvar no banco de dados. O JSON ficará desta forma:

```
{  
  "orderId": "v10089016vdb",  
  "value": 10000,  
  "creationDate": "2023-07-19T12:24:11.529Z",  
  "items": [  
    {  
      "productId": 2434,  
      "quantity": 1,  
      "price": 1000  
    }  
  ]  
}
```

Caso optar pelo banco de dados SQL ou PostgreSQL, as tabelas deverão ficar desta forma:

- Tabela: Order
  - Coluna: orderId
  - Coluna: value
  - Coluna: creationDate
- Tabela: Items
  - Coluna: orderId
  - Coluna: productId
  - Coluna: quantity
  - Coluna: price

Caso optar pelo banco de dados MongoDB, a collection deverá ficar desta forma:

```
{
  "_id" : ObjectId("64dab8a0f6b7183237d307f6"),
  "orderId" : "v10089016vdb-01",
  "value" : 10000,
  "creationDate" : ISODate("2023-07-19T12:24:11.529Z"),
  "items" : [
    {
      "productId" : 2434,
      "quantity" : 1,
      "price" : 1000,
      "_id" : ObjectId("64daba7d05bcc674899dc5bf")
    }
  ],
  "__v" : 0
}
```

Critérios de Avaliação:

- Funcionalidade completa dos requisitos mínimos.
- Código bem organizado e comentado.
- Utilização adequada das convenções de nomenclatura.
- Tratamento de erros robustos e mensagens de erro compreensíveis.
- Uso correto das respostas HTTP adequadas para cada operação.
- Código hospedado em um repositório público no GitHub, com commits organizados e mensagens claras.

Recursos Adicionais (opcional):

- Implementar autenticação básica (por exemplo, usando tokens JWT).
- Documentar a API usando uma ferramenta como Swagger ou Postman.

**Importante: Enviar o link do GitHub.**

