TESTE TEÓRICO

PROFESSIONAL SERVICES

Jitterbit

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)	Ramsés Messias de Oliveira Carvalho
Telefone	95 991554391
Linkedin	https://www.linkedin.com/in/ramscarv/
Data	01/12/2025

Javascript

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

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

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

x a) push()
b) pop()
c) shift()
d) unshit()

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

a) Remove o último elemento de um array.
x 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.
x 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.
x 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({}); }
x 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.
x b) A.
c) Hello.
d) Palavra não encontrada

8. Escreva uma função em JavaScript chamada “somaImpares” 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:

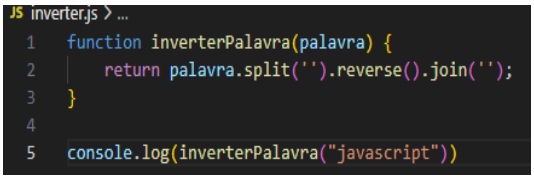
```
somaImpares(5); // Saída esperada: 9 (1 + 3 + 5)
somaImpares(10); // Saída esperada: 25 (1 + 3 + 5 + 7 + 9)
```

```
function somaImpares(n) {
    let soma = 0;
    for (let i = 1; i <= n; i += 2) {
        soma += i;
    }
    return soma;
}

console.log(somaImpares(5))
console.log(somaImpares(10))
```

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));
```

10

b)

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

Erro: Divisão por zero não é permitida.

c)

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

7

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)

map() – É usado quando se quer copiar um array, usando apenas partes dos seus dados, gerando um novo array como resultado. Por exemplo, se você tem um array de produtos e quer extrair apenas os nomes, o map() percorre cada produto e retorna um novo array só com os nomes.

forEach() – É utilizado quando se quer realizar uma ação apenas sobre aalguns elementos do array. Por exemplo, exibir cada item no console. Ele apenas itera e aplica uma função a cada elemento.

for...of - É utilizazdo quando se quer iterar diretamente sobre os valores de um array, acessando o valor de cada elemento ou indíce. É útil quando a lógica de iteração é mais complexa e requer decisões durante o percurso.

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

Variáveis são espaços de memória responsáveis por armazenar dados de determinados tipos. Para criar uma variável, deve-se declarar o seu tipo (let, const ou var), em seguida definir um nome e usar o símbolo "=" para atribuir o valor desejado a ela.

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)

sim, é possível ter múltiplas condições em um estrutura "if/else", para isso é necessário usar os operadores lógicos ou "if aninhados". O uso do operador lógico "&&", pede que todos os valores sejam verdadeiros.

Ex: if(1 > 0 && 2 > 1)

Já para o uso do operador lógico "||" é necessário que ao menos um dos valores seja verdadeiro.

Ex: if(1 < 0 || 2 > 1)

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)

Ele é utilizado para envolver instruções que podem lançar erros durante a execução. O bloco try deve ser acompanhado por pelo menos um catch, que captura e trata o erro, ou por um finally, que executa um código independentemente de ter ocorrido erro ou não. Por exemplo, ao tentar converter uma string inválida em número, você pode usar um bloco try para realizar a operação, e um bloco catch para tratar o erro e exibir uma mensagem apropriada caso a conversão falhe.

```
try {}{}{}{
    // código que pode gerar um erro
{}{}{}
```

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)

Você pode lançar manualmente uma exceção em JavaScript usando a palavra-chave throw, seguida de um valor, normalmente um objeto do tipo Error. Para criar exceções personalizadas, basta instanciar um novo erro, como em throw new Error("Mensagem"). Isso permite gerar e propagar condições de erro específicas, que depois podem ser capturadas e tratadas por um bloco try - catch.

SQL

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

a) SELECT ALL
x 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
x 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
x d) ORDER BY

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

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

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

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

Integração de sistemas

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

x 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.
x 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.
x 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.
x 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.
x 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.
x 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)

x 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.
x 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) ( )
x d) < >

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

x 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
        }
    ]
}'
```

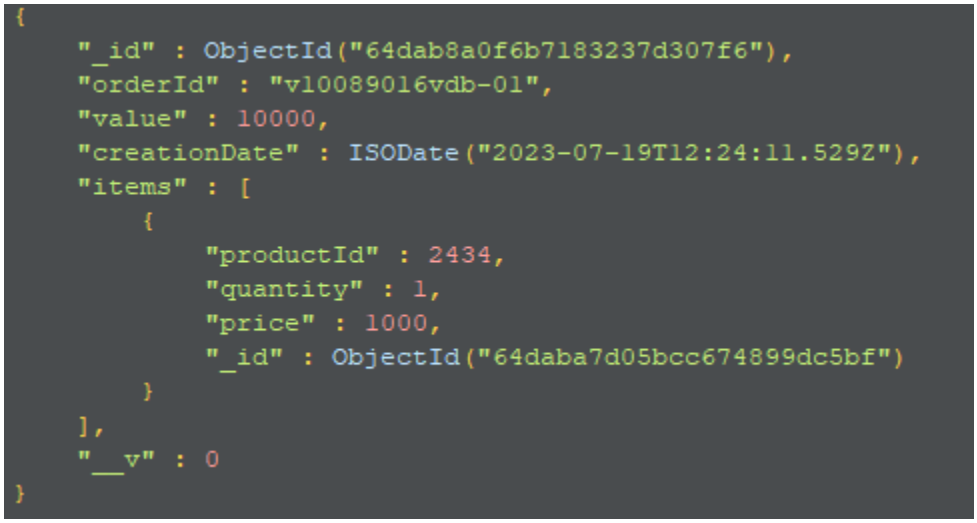
Esta API deverá sofrer uma transformação dos dados, ou seja, deverá fazer o mapping dos campos para salvar no banco de dados. O JSON ficará desta forma:

```
{
  "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. https://github.com/ramscarv/jitter_test