



**AS RESPOSTAS QUE ESTÃO EM BRANCO, FORAM RESPONDIDAS UTILIZANDO CÓDIGO, LOCALIZADO NO GITHUB, SETADO NO CORPO DO EMAIL!**

**Abaixo segue um teste para compreendermos um pouco dos seus conhecimentos. A interpretação faz parte do processo de avaliação. BOA SORTE!**

### Questão 1

Observe o algoritmo JS abaixo:

```
var valor_unitario = 10.00
```

```
var desconto = "1";
```

```
var total = valor_unitario - desconto;
```

```
total *= quantidade;
```

```
console.log(total);
```

**Onde está o erro?**

A variável quantidade não foi declarada e nem atribuída. A variável desconto está sendo atribuída como string, e não como inteiro/real (NÃO SENDO OBRIGATÓRIO ALTERAR, mas por boa prática, deve ser tratada com o seu devido tipo).

**Como ele deve ser tratado ?**

Devemos declarar a variável quantidade e atribuir um valor a mesma, para que a soma total resulte em um valor maior que zero, a variável desconto também deve ser atribuída como um inteiro ou real para que seja utilizada como boa prática ( visto que não é obrigatório em JS).

### Questão 2

**Dado o seguinte array de empresas, responda:**

```
const empresas = [  
  { nome:'Samsung',valorDeMercado: 50, CEO: 'Kim Hyun Suk', anoDeCriacao: 1938},  
  { nome: 'Microsoft',valorDeMercado: 415, CEO: 'Satya Nadella', anoDeCriacao: 1975 },  
  { nome: 'Intel',valorDeMercado: 117, CEO:'Brian Krzanich', anoDeCriacao: 1968},  
  { nome: 'Facebook',valorDeMercado: 383, CEO:'Mark Zuckerberg', anoDeCriacao: 2004},  
  { nome: 'Spotify',valorDeMercado: 30, CEO:'Daniel Ek', anoDeCriacao: 2006 },  
  {nome: 'Apple', valorDeMercado: 845, CEO: 'Tim Cook', anoDeCriacao: 1976}  
];
```

a. Faça com que seja formada pelo menos uma frase como a de exemplo:

**A empresa Apple, foi fundada por Tim Cook, no ano de 1976**

b. Retorne o valor de mercado somado de todas as empresas.

c. Retorne todas as empresas que foram fundadas após o ano de 2000.

**Questão 3**

Desenvolva uma função em PHP para realizar a operação de "Delete" em um banco de dados SQL Server, considerando a conexão já estabelecida. Explique brevemente o propósito de cada linha de código em sua função.

**Questão 4**

Em um projeto React um componente tem seu ciclo de vida bem definido. Em quais momentos ele será reconstruído (atualizado)? [A atualização de um componente pode ocorrer em resposta a mudanças em seu estado \(state\) ou em suas propriedades \(props\). Podemos utilizar de vários métodos nativos do react \(vue e angular\) para trabalhar com Life Cycle Hooks em estados de componentes.](#)

**Questão 5**

Crie uma função em PHP que receba duas datas e retorne um array com todas as datas nesse intervalo. (Não utilizar a função "range" nativa do php)

**Questão 6**

O que será impresso na tela ao rodar o código abaixo? Justifique.

Ao invocar a função criada, o retorno será nulo (o que ocasionará um erro não tratado), pois último índice da lista, é o número 1. Como o valor dado como parâmetro foi o 2, não será possível encontrar nenhum dado nesta posição de Array.

```
<?php

function getCustomerNameByIndex(array $customers, int $index):string {
    return $customers[$index]["name"] ?? null;
}

$customers = array(
    ["name" => "John Doe"],
    ["name" => "Ana Smith"]
);

$customerName = getCustomerNameByIndex($customers, 2);

print($customerName);
```

Marque as linhas de código referentes à linguagem PHP:

- `const nPeso = 70;`
- `console.log("Aglets Soluções");`



- `print_r("Hello world!");`
- `alert(`Hoje é ${new Date()}`);`
- `$date = date("Y/m/d");`

Marque uma ou mais opções corretas que representam um comando, do sistema de controle de versões git, que baixa as alterações do repositório remoto:

- STASH
- PUSH
- **PULL** Podemos utilizar o `git clone +linkDoRepositório`, para clonar o repositório pela primeira vez, ou quando não há segmento do repositório na nossa máquina.
- ENVIATE
- COMMIT

Marque uma ou mais opções corretas que representam um comando, do sistema de controle de versões git, que envia as alterações para o repositório remoto.

- STASH
- **PUSH** 2 - Realizamos o push, seguido de parâmetros (opcionais ou não.)
- PULL
- ENVIATE
- **COMMIT** 1 - Primeiramente realizamos o commit, com descrição intuitiva do que foi feito.

### Questão 7

O que sera printado no `var_dump`?

Será printado no `var_dump` a string "azul"

```
<?php
```

```
$cor = 'azul';
```

```
$azul = 'cor';
```

```
var_dump($$azul);
```

### Questão 8

Crie um algoritmo que faça a validação se um número informado pelo usuário é múltiplo de 3.

### Questão 9

Dado o seguinte array de objetos representando livros, responda:



```
const livros = [  
  { titulo: 'JavaScript: The Good Parts', autor: 'Douglas Crockford', ano: 2008 },  
  { titulo: 'Eloquent JavaScript', autor: 'Marijn Haverbeke', ano: 2011 },  
  { titulo: 'You Don\'t Know JS', autor: 'Kyle Simpson', ano: 2014 }  
];
```

- Escreva uma função chamada **“formatarLivro”** que receba um objeto livro e retorne uma string no formato “Título, escrito por Autor (Ano)”.
- Dada a seguinte lista de números, crie uma função que retorne todos os números, com todos os valores pares dobrados:

```
const numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
```

- Dado o seguinte array de nomes, retorne um novo array sem que os valores estejam duplicados:

```
var arr = ['Guilherme', 'Joao', 'Guilherme', 'Vitoria'];
```

### Questão 10

Qual é a abordagem correta para lidar com grandes conjuntos de dados retornados por uma API? Explique sua resposta.

- Retornar todos os dados de uma vez para melhorar o desempenho
- Implementar uma paginação eficiente, limitando o número de resultados por solicitação
- Não fornecer mecanismos de paginação
- Utilizar exclusivamente cabeçalhos HTTP para controle de paginação

Trabalhar com paginação é o ideal para realizar amostragem de dados de forma prática e eficiente, realizando requisições para API somente quando for solicitado uma nova amostragem de dados.

### Questão 11

Em um projeto React utilizando contexto, crie uma aplicação que simule um carrinho de compras. Onde o usuário poderá adicionar produto no carrinho, remover produto do carrinho e no mesmo devesse realizar a somatória do carrinho em tempo de execução, usando as boas praticas.



### Questão 12

Explique o conceito de Server-Side Rendering (SSR). Cite algum framework com esse propósito.

O que é renderização progressiva?

Explique a diferença entre funções síncronas e assíncronas.

**SSR** refere-se a uma técnica onde o servidor fica responsável pela renderização das páginas web, antes de enviá-las ao cliente. Diferente do CSR, onde a página é renderizada no client, no CSR o servidor gera a página solicitada e envia ao Browser.

**Renderização progressiva:** Envia uma versão básica da página ao navegador, permitindo ao usuário visualizar o conteúdo inicial antes da conclusão total do carregamento. A página é aprimorada gradualmente à medida que mais recursos são carregados, proporcionando uma experiência mais rápida.

**Síncronas:**

Em funções síncronas, as instruções são executadas uma após a outra, em ordem sequencial. Cada instrução espera a conclusão da anterior antes de ser executada.

**Assíncronas**

Em funções assíncronas, certas operações podem ser executadas em segundo plano, sem bloquear a execução do restante do código. Geralmente, são usadas para operações que podem levar algum tempo, como leitura de arquivos, chamadas de API, etc.

### Questão 13

De que maneira faria uma união de dois SQL's em que seriam recuperados dados de tabelas distintas e que poderiam estar tanto em uma quanto em outra, contudo essa união não pode utilizar o UNION para auxílio?

Dependendo da lógica a ser implementada, eu optaria por utilizar a cláusula JOIN, seguida ou não do tipo da junção (inner, left, right..

SELECT \*

FROM tabela1

JOIN tabela2 ON tabela1.coluna\_comum = tabela2.coluna\_comum;

### Questão 14

Crie um algoritmo que receba como entrada o valor unitário do produto, a porcentagem de desconto em cima do valor total calculado, a quantidade comprada e a porcentagem de imposto que será aplicada no valor total. Deverá retornar ao usuário o valor total e o valor do imposto.

### Questão 15

Em uma API construída em PHP com seu ambiente já totalmente configurado e conectado a um banco de dados, utilizando Eloquent como ORM, crie um serviço que realizará um POST, em uma tabela Usuários (ID, NOME, IDADE, SEXO, ALTURA), dentro de um Service.

Descreva qual processo deve ser seguido para que se tenha uma rota funcional dentro da API que irá utilizar esse novo serviço.

Criação da controller responsável por lidar com as requisições relacionadas aos usuários;

Utilizar o serviço de usuarios na controller com os métodos necessários;

Definir na rota o verbo HTTP do tipo POST e a classe/método a ser utilizado;

Testar a rota utilizando CURL, Insomnia ou Postman e seguir com implementações se necessário.

### Questão 16



**Situação:**

Suponha que você está construindo do zero uma aplicação web usando Node.js e Express para lidar com muitas requisições simultâneas. Seu objetivo é garantir que a aplicação seja eficiente e escalável.

Como você otimizaria o desempenho de uma aplicação Node.js que lida com um grande volume de requisições simultâneas? Explique algumas práticas e estratégias que você consideraria para garantir uma execução eficiente e escalabilidade da aplicação.

O uso de um framework eficiente como Express já lida muito bem com o problema de altas demandas de requisições;  
Gerenciar processos com ferramentas como pm2;  
Utilizar pools de conexões com o bancos que possuem suporte (postgres, sql)

**Questão 17**

Descreva estratégias avançadas para otimizar o tamanho do bundle em um aplicativo React

Ter controle do Bundle, fazer o uso do Webpacks , realizar code splitting para fazer carregamento mínimo em tempo de execução

**Questão 18**

Considere o seguinte cenário: um sistema ERP web construído com as seguintes tecnologias PHP, HTML, javascript/Jquery, SqlServer(banco de dados) e armazena algumas informações na sessão do PHP.

O usuário utilizando esse ERP percebe uma lentidão ao abrir uma das rotinas desse sistema. Enquanto aguarda o término do processamento dessa rotina, ele decide abrir uma segunda página no mesmo browser para realizar outras operações. Ao abrir percebe que essa página também está com lentidão. Ele decide então abrir uma terceira página desse mesmo ERP, mas em outro browser, e percebe que ela não está com lentidão.

Porque isso ocorreu ?

Esta situação pode estar relacionada na forma em como o browser utilizado se comporta com sessions, memória cache ou até mesmo limite de conexão do mesmo.



### Questão 19

Explique um caso de uso do Docker.

Facilita a execução do projeto/sistema em diversos ambientes de desenvolvimento, compactando o projeto e acelerando a preparação do ambiente.

Porque você criaria classes estáticas?

Criar classes estáticas para armazenar métodos ou propriedades que são relevantes para a classe em si, em vez de instâncias individuais, além de economizar memória por não poder ser instanciada

Quais são as diferenças entre continuous integration, continuous delivery e continuous deployment?

CI: Integrar automaticamente o código produzido pelos desenvolvedores em um repositório compartilhado várias vezes ao dia.

CD: Garantir que o software esteja sempre pronto para ser implantado em produção.

CDeployment: Automatizar completamente o processo de implantação, liberando automaticamente as versões do software que passam nos testes.

### Questão 20

Em um processamento assíncrono que faz a transmissão de dados através de vários arquivos para um servidor externo, defina a melhor maneira, na sua concepção, como garantir a transmissão completa e correta, tratando quedas de conexão durante a execução do processamento dos dados.

Dividir os dados em partes menores (Paginação) para evitar timeouts e permitir tentativas específicas de partes falhadas. ( Já vi bastante esse tipo de implementação em cursos/ambientes de trabalho

Implementar tentativas automáticas para falhas temporárias, agendando novas tentativas após intervalos definidos.

Sistema externo conectado a API para monitoramento de erros e possível solução automática, mas focado no log de falhas

Definir timeouts apropriados para operações de transmissão, encerrando tentativas demoradas e considerando tentativas.

Um sistemas de filas de mensagens para transmissão confiável, lidando com falhas e garantindo a entrega consistente de mensagens.