



Teste Técnico Vaga de Desenvolvedor Pleno

1. Manipulação Intermediária do DOM

Cenário:

Você precisa capturar cliques em todos os botões com a classe `.adicionar-carrinho` e enviar para o console um objeto contendo:

- ID do produto (armazenado no atributo `data-id` do botão)
- Nome do produto (armazenado no atributo `data-nome`)
- Hora do clique no formato `YYYY-MM-DD HH:mm:ss`

Tarefa:

Implemente um código JavaScript puro que faça essa captura e exiba os dados no console.

2. Manipulação de Strings e Objetos

Cenário:

Você recebeu um objeto com informações de um pedido, mas precisa extrair apenas alguns dados para enviar ao GA4.

```
const pedido = {  
  id: 987,  
  cliente: { nome: "João Silva", email: "joao@email.com" },  
  itens: [  
    { nome: "Camiseta", preco: 50 },  
    { nome: "Calça", preco: 120 }  
  ],  
  status: "pago"  
};
```

Tarefa:

Escreva uma função `resumirPedido` que retorne um novo objeto contendo:

```
{  
  id: 987,  
  total: 170,  
  quantidadeItens: 2  
}
```

3. Funções Reutilizáveis

Cenário:

Você precisa criar uma função que formate preços para o padrão brasileiro (R\$ 0,00).

Tarefa:

Implemente uma função `formatarPreco` que:

- Receba um número e retorne uma string formatada no padrão brasileiro.
- Utilize `Intl.NumberFormat`.
- Depois, use essa função para formatar todos os preços do array abaixo:

```
const precos = [50, 120, 250];
```

4. Assíncrono e APIs

Cenário:

Você precisa buscar a cotação atual do dólar em uma API pública antes de enviar um evento para o GA4.

Tarefa:

Implemente uma função assíncrona que:

1. Faça uma requisição para <https://dolarapi.com/v1/cotacao/usd>
2. Retorne apenas o valor de **compra** do dólar em relação ao real (campo "compra").
3. Trate erros de forma adequada, exibindo uma mensagem amigável caso a requisição falhe. Simule um erro de execução.

5. Programação Funcional

Cenário:

Você recebeu um array de eventos de usuários e precisa saber quantos eventos de cada tipo ocorreram.

```
const eventos = [  
  { tipo: "click" },  
  { tipo: "scroll" },  
  { tipo: "click" },  
  { tipo: "page_view" },  
  { tipo: "click" }  
];
```

Tarefa:

Escreva uma função que retorne: `{ click: 3, scroll: 1, page_view: 1 }`

6. Google Tag Manager – Configuração Intermediária

Explique como você configuraria no GTM um evento `add_to_cart` que:

- É disparado quando o usuário clica em um botão "Adicionar ao Carrinho".
- Recebe do Data Layer as informações de ID, nome e preço do produto.
- Envia os dados para o GA4.

7. Google Tag Manager – Data Layer

1. Explique o que é o Data Layer e por que ele é importante.
2. Mostre como adicionar manualmente um evento login_sucesso ao Data Layer, incluindo o ID do usuário.

8. Resolução de Problemas

Cenário:

Você implementou um evento purchase no GTM, mas percebeu que em alguns casos o valor total está sendo enviado como ``.

- Quais seriam as possíveis causas?
- Como você investigaria e corrigiria o problema?

9. Boas Práticas

Liste **3 boas práticas** que você considera essenciais ao implementar códigos de trackeamento em um site.