

DESAFIO ESTÁGIO QA – PROBRAIN

Patrícia Helena Milane

REQUISITOS

Criar aplicação simples com as seguintes implementações e requisitos mínimos:

- Verificar se o número é par ou ímpar;
- Verificar se o número é primo ou não;
- Mostrar se o número existe na sequência de Fibonacci;
- Listar números onde os múltiplos de quatro recebem o adicional da palavra 'pin';
- Deverá ser utilizado HTML, CSS e Javascript nas implementações.

IMPLEMENTAÇÃO 1

Após o levantamento dos requisitos, foi realizado uma listagem do passo a passo que cada implementação deveria ter para chegar no resultado esperado.

- **Desenvolvendo o Algoritmo:**

- Criar input;
- Receber um número através do input;
- Guardar o número em uma variável e converter para Number;
- Verificar se o número está entre 1 e 1000;
- Se não, exibir uma mensagem pedindo ao usuário para digitar um numero entre 1 e 1000;
- Se sim, verificar o resto da divisão do numero digitado;
- Se for igual a 0, exibir na tela “O número é par”;
- Se for igual a 1 (ou else), exibir “O número é ímpar”.

- **Testes, falhas encontradas e corrigidas:**

Foi utilizado inicialmente o `console.log()` para verificar se a implementação estava retornando os valores corretos. O algoritmo apresentou falhas de lógica, pois todos os valores digitados eram retornados como “Ímpar”, que se encontra no else da

implementação. O problema apareceu pois não foi feita a conversão do valor recebido no input para Number. Após a correção, o algoritmo retornou o resultado esperado.

IMPLEMENTAÇÃO 2

Na implementação 2, foi necessário realizar uma pesquisa sobre as regras que envolvem um número primo.

- **Desenvolvendo o Algoritmo:**

- Criar input;
- Receber um número através do input;
- Guardar o numero em uma variável, já convertendo para Number;
- No primeiro if, verificar se o número está entre 1 e 1000;
- Se não, exibir uma mensagem pedindo ao usuário para digitar um número entre 1 e 1000;
- Se sim, no segundo if ele deverá verificar se é igual a 1, 2, 3, 5 ou 7;
- Se for igual a 1, retornar 'O número não é primo';
- Se for igual a algum valor do segundo if e que não seja 1, retornar 'O número é primo';
- Saindo do bloco do segundo if, ele deverá verificar se o resto da divisão de 2, 3, 5 e 7 são iguais a 0 e retornar 'O numero não é primo', caso verdadeiro;
- Caso o valor não corresponder com nenhum dos parâmetros passados, retornar 'O número é primo';

- **Testes, falhas encontradas e corrigidas:**

Uma falha encontrada logo nos primeiros testes, foi que os valores divisíveis por 3, 5 e 7 estavam sendo retornados como 'O é número primo'. Nisso, foi necessário inserir no else if que os números em que o resto da divisão por 3, 5 e 7 eram iguais a 0, deveriam ser retornados como 'O número não é primo'. Foi utilizado também o console.log() em blocos menores do código para verificar o que estava sendo retornado.

IMPLEMENTAÇÃO 3

A implementação 3 não foi realizada. Foi encontrado dificuldade de entender a lógica envolvida, e mesmo após algumas tentativas o resultado retornado não era o proposto pelo teste.

IMPLEMENTAÇÃO 4

- **Desenvolvendo o Algoritmo:**

- Criar uma Div para exibir os valores;
- Criar um laço de repetição For, onde o valor 1000 será o fim da condição de contagem e i será 1, valor do início da contagem;
- Dentro do For, criar estrutura de condicionais onde o if irá comparar se o resto da divisão de i por 4 é igual a 0;
- Se sim, incluir pin na variável;
- Se não, exibir somente o número;
- Exibir os resultados na tela.

- **Teste, falhas encontradas e corrigidas:**

Antes de retornar os valores na tela, foi verificado se a lógica estava correta através do `console.log()` no bloco do código. Retornando os valores corretamente, foi criada a tag `` no html e implementado `` para cada i retornado no JavaScript, porém os valores estavam sendo exibidos um abaixo do outro. Como a contagem foi realizada até 1000, a visualização ficou ineficiente. Para resolver de forma mais simples, foi decidido remover os `` e criar uma div, e nela separar os valores por vírgula, tornando a exibição mais agradável.