

## ATIVIDADE PRÁTICA – JAVASCRIPT

JS-07 – Collections

### Instruções gerais:

1. Utilize o VS Code para desenvolver os algoritmos.
2. Ao concluir os exercícios, envie todos os códigos criados no VS Code para o Repositório criado na sua conta pessoal do Github, em uma pasta identificada com o tema da sessão
3. Caso seja solicitado, adicione o **link do Repositório do Github** e os **Prints** dos códigos desenvolvidos na Plataforma Canvas.

**Mantenha as entregas das Atividades em dia na Plataforma Canvas**

## EXERCÍCIOS

### Boas práticas:

1. Resolva **1 exercício da lista 01 - Collection Array** e **1 exercício da lista 02 - Collection Set**.
2. Leia o enunciado do exercício com atenção
3. Observe as indicações de Entrada e Saída esperadas em cada exercício
4. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
5. Utilize o Cookbook, os Vídeos da Plataforma e os Códigos guia como referências para a resolução dos exercícios
6. Na entrega das atividades na Plataforma Canvas, efetue o **envio do Print do código de cada exercício resolvido e o link do repositório do Github**.
7. Caso ainda fique alguma dúvida, consulte os instrutores da sua turma pelo Discord

## **Lista 01 – Collection Array (Resolva 1 Exercício)**

- 1) Escreva um programa para criar uma **Collection Array** de Objetos do tipo **string**. O programa deverá solicitar ao usuário, que ele digite via teclado 5 cores e deverá adicioná-las individualmente no Array. Em seguida, faça o que se pede:
- Mostre na tela todas as cores adicionadas.
  - Mostre na tela todas as cores adicionadas, ordenadas em ordem crescente.

Veja o exemplo abaixo:

ENTRADA	SAÍDA
Azul Verde Amarelo Branco Laranja	<b>Listar todas as cores:</b>  Azul Verde Amarelo Branco Laranja  <b>Ordenar as cores:</b>  Amarelo Azul Branco Laranja Verde

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Entrada e Saída de dados
- Laços de Repetição
- **Collection Array**

- 2) Escreva um programa para criar uma **Collection Array** do tipo **number**, inicializada com 10 valores inteiros. O programa deverá solicitar ao usuário, que ele digite via teclado 1 número inteiro e caso ele seja encontrado no Array, exiba na tela a **posição deste número** na Collection. Caso o número não seja encontrado, o programa deverá exibir na tela a mensagem: **O número NN não foi encontrado!**

Veja o exemplo abaixo:

ENTRADA	SAÍDA										
<b>0 1 2 3 4 5 6 7 8 9</b> <b>Array</b> <table border="1" style="display: inline-table;"> <tr> <td>2</td><td>5</td><td>1</td><td>3</td><td>4</td><td>9</td><td>7</td><td>8</td><td>10</td><td>6</td> </tr> </table> Digite o número que você deseja encontrar: 7	2	5	1	3	4	9	7	8	10	6	<b>O número 7 está localizado na posição: 6</b>
2	5	1	3	4	9	7	8	10	6		
<b>0 1 2 3 4 5 6 7 8 9</b> <b>Array</b> <table border="1" style="display: inline-table;"> <tr> <td>2</td><td>5</td><td>1</td><td>3</td><td>4</td><td>9</td><td>7</td><td>8</td><td>10</td><td>6</td> </tr> </table> Digite o número que você deseja encontrar: 40	2	5	1	3	4	9	7	8	10	6	<b>O número 40 não foi encontrado!</b>
2	5	1	3	4	9	7	8	10	6		

\* Os números em vermelho representam o índice (posição) na Collection Array.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- **Collection Array**

## **Lista 02 – Collection Set (Resolva 1 Exercício)**

- 3) Escreva um programa para criar uma **Collection Set** do tipo **number**. O programa deverá solicitar ao usuário, que ele digite via teclado 10 valores inteiros não repetidos e adicione-os individualmente na Collection Set. Em seguida, faça o que se pede:
- Mostre na tela todos os elementos da Collection Set.

Veja o exemplo abaixo:

ENTRADA	SAÍDA										
<b>Set</b> <table border="1"><tr><td>2</td><td>5</td><td>1</td><td>3</td><td>4</td><td>9</td><td>7</td><td>8</td><td>10</td><td>6</td></tr></table>	2	5	1	3	4	9	7	8	10	6	<b>Listar dados do Set:</b> 1 2 3 4 5 6 7 8 9 10
2	5	1	3	4	9	7	8	10	6		
<b>Set</b> <table border="1"><tr><td>2</td><td>5</td><td>10</td><td>3</td><td>4</td><td>2</td><td>2</td><td>3</td><td>10</td><td>5</td></tr></table>	2	5	10	3	4	2	2	3	10	5	<b>Listar dados do Set:</b> 2 3 4 5 10
2	5	10	3	4	2	2	3	10	5		

\* *Lembre-se que a Collection Set não aceita valores repetidos.*

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Entrada e Saída de dados
- **Collection Set**

- 4) Escreva um programa para criar uma **Collection Set** do tipo **number**, inicializada com 10 valores inteiros. O programa deverá solicitar ao usuário, que ele digite via teclado 1 número inteiro e caso ele seja encontrado na Collection Set, exiba na tela a mensagem: **O Número NN foi encontrado!** Caso o número não seja encontrado, o programa deverá exibir na tela a mensagem: **O número NN não foi encontrado!**

Veja o exemplo abaixo:

ENTRADA	SAÍDA
<b>ENTRADA:</b> List [ 2   5   1   3   4   9   7   8   10   6 ]  Digite o número que você deseja encontrar: 7	<b>SAÍDA:</b> O número 7 foi encontrado!
<b>ENTRADA:</b> List [ 2   5   1   3   4   9   7   8   10   6 ]  Digite o número que você deseja encontrar: 40	<b>SAÍDA:</b> O número 40 não foi encontrado!

\* Lembre-se que a **Collection Set** não aceita valores repetidos.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- **Collection Set**