

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 todos os exercícios da lista
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

Atividade 01 – Fila

1) Escreva um programa contendo uma **Collection Queue** (Fila) de Objetos do tipo **string**, para organizar uma fila por ordem de chegada dos Clientes de um Banco. O programa deverá ter um Menu que aceitará as opções 0, 1, 2 e 3:

- **1:** Adicionar um novo Cliente na fila. Deve solicitar o nome do Cliente.
- **2:** Listar todos os Clientes na fila
- **3:** Chamar (retirar) uma pessoa da fila
- **0:** O programa deve ser finalizado.

Caso a fila esteja vazia, o programa deverá informar que a fila está vazia ao tentar retirar (chamar) um cliente da fila.

* Para resolver este exercício, não se esqueça de criar a Classe Queue

Menu:

```
*****  
1 - Adicionar Cliente na Fila  
2 - Listar todos os Clientes  
3 - Retirar Cliente da Fila  
0 - Sair  
*****
```

Entre com a opção desejada:

Veja os exemplos abaixo:

| ENTRADA | SAÍDA |
|---|---|
| Digite uma opção: 3 | A Fila está vazia! |
| Digite uma opção: 1 Digite o nome: João | Fila: João Cliente Adicionado! |
| Digite uma opção: 1 Digite o nome: Maria | Fila: João Maria Cliente Adicionado! |
| Digite uma opção: 1 Digite o nome: Ana | Fila: João Maria Ana Cliente Adicionado! |
| Digite uma opção: 2 | Lista de Clientes na Fila: João Maria Ana |
| Digite uma opção: 3 | Fila: Maria Ana |

| | |
|----------------------------|-------------------------------|
| | O Cliente foi Chamado! |
| Digite uma opção: 0 | Programa Finalizado! |

Atividade 02 – Pilha

2) Escreva um programa contendo uma **Collection Stack** (Pilha) de Objetos **string**, para organizar a retirada de livros de uma pilha. O programa deverá ter um Menu que aceitará as opções 0, 1, 2 e 3:

- **1:** Adicionar um novo livro na pilha. Deve solicitar o nome do livro.
- **2:** Listar todos os livros da Pilha
- **3:** Retirar um livro da pilha
- **0:** O programa deve ser finalizado.

Caso a pilha esteja vazia, ao tentar retirar um livro da pilha, o programa deverá informar que a pilha está vazia.

* Para resolver este exercício, não se esqueça de criar a Classe Stack

Menu:

```
*****  
1 - Adicionar Livro na pilha  
2 - Listar todos os Livros  
3 - Retirar Livro da pilha  
0 - Sair  
*****
```

Entre com a opção desejada:

Veja os exemplos abaixo:

| ENTRADA | SAÍDA |
|---|--|
| Digite uma opção: 3 | A Pilha está vazia! |
| Digite uma opção: 1 Digite o nome: O Auto da Compadecida | Pilha: O Auto da Compadecida Livro adicionado! |
| Digite uma opção: 1 Digite o nome: O Alquimista | Pilha: O Auto da Compadecida O Alquimista Livro adicionado! |
| Digite uma opção: 1 Digite o nome: Um Estudo em Vermelho | Pilha: O Auto da Compadecida O Alquimista Um Estudo em Vermelho Livro adicionado! |
| Digite uma opção: 2 | Lista de Livros na Pilha: O Auto da Compadecida O Alquimista Um Estudo em Vermelho |
| Digite uma opção: 3 | Pilha: O Auto da Compadecida O Alquimista Um Livro foi retirado da pilha! |

Digite uma opção: 0

Programa Finalizado!