



# PBL 02

## ALGORITMOS E PROGRAMAÇÃO

INICIAR

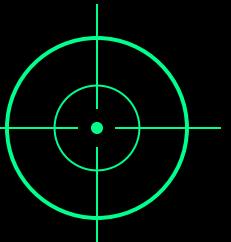




# Integrantes do grupo

- Davi Chaves
- Davi Castro
- Kalil Ramos

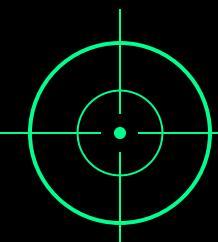
- Kauã Carneiro
- Lucas Gomes
- Gabriel Araújo



# Refatoração Baseada em Casos de Uso

---

- A função `emprestar_livro()` representa o caso de uso de realizar um empréstimo.
- A função `devolver_livro()` lida com a devolução de livros.
- A função `listar_livros()` permite visualizar o estado atual da biblioteca, seja com livros disponíveis ou emprestados.
- Já `exibir_menu()` e `main()` compõem o fluxo de interação com o usuário.



# Substituição do while



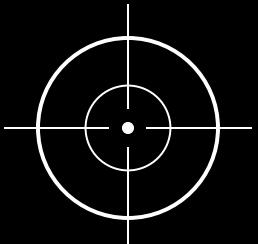
```
while True:  
    print("\n1 - Empréstimo\n2 - Devolução\n3 - Listar Livros\n4 - Sair")  
    opcao = input("Escolha uma opção: ")
```

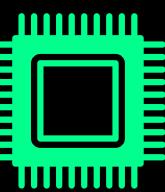
**Em vez de um while True, que pode parecer “rude” ou “infinito sem controle”, a recursividade dá uma sensação de continuidade mais controlada.**



```
def main():  
    opcao = exibir_menu()  
    if opcao == "1":  
        emprestar_livro()  
    elif opcao == "2":  
        devolver_livro()  
    elif opcao == "3":  
        listar_livros(disponivel=True)  
        listar_livros(disponivel=False)  
    elif opcao == "4":  
        print("Saindo...")  
        return  
    else:  
        print("Opção inválida!")  
  
    main() # Chamada recursiva para repetir o menu  
  
# Execução do programa  
main()
```

**A recursão pode deixar o código visualmente mais limpo, com o fluxo de execução mais linear e fácil de ler (especialmente se você estiver começando a programar).**





# Lista x Dicionário

```
# Sistema de Biblioteca (Versão Bagunçada)
biblioteca = ["Dom Casmurro", "1984", "O Senhor dos Anéis", "Harry Potter"]
emprestados = []
```



```
biblioteca = {
    1: {"titulo": "Dom Casmurro", "emprestado": False},
    2: {"titulo": "1984", "emprestado": True}, # já emprestado
    3: {"titulo": "O Senhor dos Anéis", "emprestado": False},
    4: {"titulo": "Harry Potter", "emprestado": True} # já emprestado
}
```

**A mudança para dicionário torna o sistema mais robusto e flexível, preparando-o para evoluir com novos requisitos (como salvar em banco, adicionar filtros, etc).**

# Organização do Fluxo Principal

```
def main():
    opcao = exibir_menu()
    if opcao == "1":
        emprestar_livro()
    elif opcao == "2":
        devolver_livro()
    elif opcao == "3":
        listar_livros(disponivel=True)
        listar_livros(disponivel=False)
    elif opcao == "4":
        print("Saindo...")
        return
    else:
        print("Opção inválida!")

main() # Chamada recursiva para repetir o menu

# Execução do programa
main()
```

- O `main()` controla o menu e as opções.
- O código principal fica limpo, centralizado e mais fácil de entender.
- Código mais modular, escalável e fácil de expandir.





MUITO  
Obrigado!

-  Davi chaves
-  Davi Castro
-  Kalil Ramos
-  Lucas Gomes
-  Gabriel Araújo
-  Kauã Carneiro

