

DESAFIO 2

O que você precisa fazer:

- Desenvolva um programa que permita o cadastro de diversos livros com as seguintes informações: Nome do livro, Gênero, Autor, Número total de páginas, Número de páginas lidas.
- O programa deve possibilitar a consulta de livros por nome, gênero e autor, além de acompanhar a progressão de leitura de cada livro.

Entradas:

- Nome do livro,
- Gênero,
- Autor,
- Número total de páginas,
- Número de páginas lidas

Saídas:

- Nome do livro,
- Número total de páginas,
- Número de páginas lidas,
- Classificação por gênero, Autor

CÓDIGO:

```
data class Livro(  
    val nome: String,  
    val genero: String,
```

```
    val autor: String,  
    val totalPaginas: Int,  
    var paginasLidas: Int  
)
```

```
fun main() {  
    val livros = mutableListOf<Livro>()
```

// Função para cadastrar um livro

```
fun cadastrarLivro() {  
    print("Nome do livro: ")  
    val nome = readLine()!!  
    print("Gênero: ")  
    val genero = readLine()!!  
    print("Autor: ")  
    val autor = readLine()!!  
    print("Número total de páginas: ")  
    val totalPaginas = readLine()!!.toInt()  
    val livro = Livro(nome, genero, autor, totalPaginas, 0)  
    livros.add(livro)  
    println("Livro cadastrado com sucesso!")  
}
```

// Função para consultar livros por nome, gênero e autor

```
fun consultarPorNome(nome:  
    String):  
    List<Livro> {  
    return livros.filter {it.nome.contains(nome, ignoreCase = true)}  
    }  
}
```

```
fun consultarPorGenero(genero:  
    String):  
    List<Livro> {  
    return livros.filter {it.genero.contains(genero, ignoreCase = true)}  
    }  
}
```

```
fun consultarPorAutor(autor:  
    String):
```



```
        println("Nenhum livro encontrado.")
    } else {
        resultadosAutor.forEach { mostrarLivro(it) }
    }
    } else {
        resultadosGenero.forEach { mostrarLivro(it) }
    }
    } else {
        resultados.forEach { mostrarLivro(it) }
    }
    }
    }
    }
    3 -> break
    else -> println("Opção inválida.")
}
}
```

Script:

```
main.kt x +
main.kt > Livro > Livro > Livro > Livro > Livro > f main > f cadastrarLivro > ...

1  data class Livro(
2      val nome: String,
3      val genero: String,
4      val autor: String,
5      val totalPaginas: Int,
6      var paginasLidas: Int
7  )
8  fun main() {
9      val livros = mutableListOf<Livro>()
10     // Função para cadastrar um livro
11     fun cadastrarLivro() {
12         print("Nome do livro: ")
13         val nome = readLine()!!
14         print("Gênero: ")
15         val genero = readLine()!!
16         print("Autor: ")
17         val autor = readLine()!!
18         print("Número total de páginas: ")
19         val totalPaginas = readLine()!!.toInt()
20         val livro = Livro(nome, genero, autor, totalPaginas, 0)
21         livros.add(livro)
22         println("Livro cadastrado com sucesso!")
23     }
24     // Função para consultar livros por nome, genero e autor
25     fun consultarPorNome(nome:
26         String):
27         List<Livro> {
28         return livros.filter {it.nome.contains(nome, ignoreCase = true)
29         }
30     }
31     fun consultarPorGenero(genero:
32         String):
33         List<Livro> {
34         return livros.filter {it.genero.contains(genero, ignoreCase = true)
35         }
36     }
37     fun consultarPorAutor(autor:
38         String):
```



```
main.kt x +
main.kt > Livro > Livro > Livro > Livro > Livro > f main > f cadastrarLivro > ...
1 -> cadastrarLivro()
61 2 -> {
62     print("Digite o nome, gênero ou autor para buscar: ")
63     val busca = readLine()!!
64
65     // Verifica o tipo de busca e chama a função correspondente
66     when {
67         busca.toIntOrNull() != null -> {
68             println("Busca por ID não implementada ainda.")
69         }
70         else -> {
71             val resultados = consultarPorNome(busca)
72             if (resultados.isEmpty()) {
73                 val resultadosGenero = consultarPorGenero(busca)
74                 if (resultadosGenero.isEmpty()) {
75                     val resultadosAutor = consultarPorAutor(busca)
76                     if (resultadosAutor.isEmpty()) {
77                         println("Nenhum livro encontrado.")
78                     } else {
79                         resultadosAutor.forEach { mostrarLivro(it) }
80                     }
81                 } else {
82                     resultadosGenero.forEach { mostrarLivro(it) }
83                 }
84             } else {
85                 resultados.forEach { mostrarLivro(it) }
86             }
87         }
88     }
89 }
90 3 -> break
91 else -> println("Opção inválida.")
92 }
93 }
94 }
95 }
```

Saída:

1. Cadastrar livro
2. Consultar livro
3. Sair

Opção: 1

Nome do livro: Criptografia para Todos

Gênero: Criptografia e Seguranca

Autor: Jiyan Yari

Número total de páginas: 216

Livro cadastrado com sucesso!

1. Cadastrar livro
2. Consultar livro
3. Sair

Opção: 2

Digite o nome, gênero ou autor para buscar: Criptografia para Todos

Nome: Criptografia para Todos

Gênero: Criptografia e Seguranca

Autor: Jiyan Yari

Total de páginas: 216

Páginas lidas: 0

1. Cadastrar livro
2. Consultar livro
3. Sair

Opção: 2

Digite o nome, gênero ou autor para buscar: Criptografia e Seguranca

Nome: Criptografia para Todos

Gênero: Criptografia e Seguranca

Autor: Jiyan Yari

Total de páginas: 216

Páginas lidas: 0

1. Cadastrar livro
2. Consultar livro
3. Sair

Opção: 2

Digite o nome, gênero ou autor para buscar: Jiyan Yari

Nome: Criptografia para Todos

Gênero: Criptografia e Seguranca

Autor: Jiyan Yari

Total de páginas: 216

Páginas lidas: 0

1. Cadastrar livro
2. Consultar livro
3. Sair

Opção: 3

