

## Explicação das Funções “temNumero” e “alugarLivros”

### 1. 🚀 Resumo rápido:

- 📌 A função `temNumero()` faz uma varredura na string e retorna se há números.
  - 📌 A função `alugarLivro()` permite que o usuário escolha um livro, verifica se está disponível e solicita um nome válido (sem números) para efetuar o aluguel.
- 

### 2. 🔍 Função `temNumero()`

```
int temNumero(const char texto[]) {  
    for (int i = 0; texto[i] != '\0'; i++) {  
        if (texto[i] >= '0' && texto[i] <= '9') {  
            return 1; // Tem número  
        }  
    }  
    return 0; // Não tem número  
}
```

### 🧠 Explicação detalhada:

- `int temNumero(const char texto[])`  
👉 Função que recebe uma string (`texto[]`) e retorna **1 (verdadeiro)** se houver qualquer número na string ou **0 (falso)** se não tiver número.
- `for (int i = 0; texto[i] != '\0'; i++)`  
👉 Loop que percorre cada caractere da string até encontrar o final (`'\0'` é o caractere nulo que indica o fim de uma string em C).
- `if (texto[i] >= '0' && texto[i] <= '9')`  
👉 Verifica se o caractere atual está no intervalo dos dígitos de '0' a '9'.

■ Isso significa que o caractere é um número.

- **return 1;**  
👉 Se encontrou um número, retorna **1** imediatamente e para a execução da função.
  - **return 0;**  
👉 Se percorreu toda a string e não encontrou números, retorna **0**, indicando que **não há números** na string.
- 

### 3. 📖 Função **alugarLivro()**

🔗 Código da função:

```
void alugarLivro() {  
    int escolha;  
    listar_livros();  
  
    printf("Escolha o número do livro que deseja alugar: ");  
    scanf("%d", &escolha);  
  
    char nomeLocador[50];  
  
    switch (escolha) {
```

#### 3.1 🧠 Explicação do começo:

- **void alugarLivro()**  
👉 Função que permite ao usuário alugar um livro disponível.
- **int escolha;**  
👉 Variável que armazena o número do livro escolhido (1 a 5).
- **listar\_livros();**  
👉 Chama a função que lista os livros cadastrados.
- **printf(...); scanf("%d", &escolha);**  
👉 Mostra uma mensagem solicitando o número do livro e lê esse número.

- **char nomeLocador[50];**  
👉 Vetor que armazenará o nome da pessoa que está alugando o livro.
  - **switch (escolha) {**  
👉 Estrutura de decisão que escolhe qual livro será alugado, de acordo com a escolha do usuário (1 a 5).
- 

#### 4. Dentro de cada **case**:

Exemplo para o livro 1 (**estrutura se repete para os outros livros**):

**case 1:**

```
if(strlen(titulo1) == 0) {  
    printf("Livro não cadastrado!\n");  
    break;  
}
```

- Verifica se o título do livro 1 está vazio (**livro não cadastrado**).  
Se estiver vazio, exibe mensagem e sai desse **case**.

```
if(locado1 == 1) {  
    printf("Esse livro já está alugado!\n");  
    break;  
}
```

- Verifica se o livro 1 já está alugado (**locado1 == 1**).  
Se sim, exibe mensagem e não permite alugar novamente.

```
getchar();
```

- Limpa o buffer do teclado, retirando o **\n** que sobra do **scanf**.
-

## 5. Recebendo o nome do locador:

```
do {  
  
    printf("Digite o nome do locador (sem números): ");  
  
    fgets(nomeLocador, 50, stdin);  
  
    size_t len1 = strlen(nomeLocador);  
  
    if (len1 > 0 && nomeLocador[len1-1] == '\n')  
nomeLocador[len1-1] = '\0';
```

- **fgets()** lê o nome do locador (permite espaços).
- Remove o `\n` do final da string, se existir.

```
    if (temNumero(nomeLocador)) {  
  
        printf("Nome inválido! Não pode conter números.\n");  
  
    }  
  
} while (temNumero(nomeLocador));
```

- Chama a função **temNumero()** para verificar se o nome contém números.  
👉 Se tiver, mostra mensagem de erro e pede novamente.
-

## 6. 📁 Finalizando o aluguel:

```
strcpy(locador1, nomeLocador);

locado1 = 1;

printf("Livro '%s' alugado por '%s'. Prazo de %d dias.\n",
titulo1, locador1, diasRestantes1);

break;
```

- **strcpy(locador1, nomeLocador);** → Copia o nome do locador para a variável do livro 1.
  - **locado1 = 1;** → Marca que o livro está alugado.
  - Mostra uma mensagem de confirmação com o título do livro, nome do locador e prazo.
- 

## 7. ✅ Final da função:

default:

```
printf("Opção inválida!\n");
```

- Caso o usuário digite um número fora do intervalo (1 a 5), exibe que a opção é inválida.

```
printf("\nPRESSIONE ENTER PARA VOLTAR AO MENU PRINCIPAL...");
```

```
getchar();
```

```
getchar();
```

- Pausa (2 - getchar) para o usuário apertar ENTER antes de voltar para o menu principal