Introdução ao Go

Resumo da Aula 30 - Exercício: "if" Statements em Go

Nesta aula, foi proposto um exercício para implementar a **terceira opção** do programa bancário: **o saque de dinheiro**. Após a pausa para que os alunos tentassem por conta própria, a solução foi explicada passo a passo.

---

1. Implementando a Opção de Saque

A lógica do saque é semelhante à do depósito, mas agora o valor inserido pelo usuário **deve ser subtraído do saldo**.

Passos para Implementação:

1. **Adicionar um novo bloco "else if" para verificar se a escolha foi 3:**

```go
else if choice == 3 {
```

2. **Criar uma variável para armazenar o valor do saque:**

```go
var withdrawalAmount float64
```

3. **Solicitar ao usuário o valor que deseja sacar:**

```go
fmt.Print("Valor do saque: ")
fmt.Scan(&withdrawalAmount)
```

4. **Atualizar o saldo, subtraindo o valor sacado:**
   - Utilizando o operador de atribuição composta (`-=`):

```go
accountBalance -= withdrawalAmount
```

5. **Exibir o saldo atualizado:**

```go
fmt.Println("Saque realizado! Novo saldo:", accountBalance)
```

---

2. Testando o Código

Após a implementação, a funcionalidade foi testada:

- Ao escolher a opção `3` e inserir um valor de `200`, o saldo foi atualizado corretamente para `800`.

Observação Importante:

- Como o programa **não armazena dados entre execuções**, ao reiniciá-lo, o saldo **retorna ao valor inicial (1000)**.
- Para manter os dados entre execuções, seria necessário **salvá-los em um arquivo ou banco de dados** – um conceito que será abordado em aulas futuras.

---

3. Próximos Passos

Na **próxima aula**, será implementada a última opção do menu, completando o fluxo de operações disponíveis no programa.

Com essa funcionalidade de saque, o programa agora permite que o usuário **visualize o saldo, deposite e saque dinheiro**, tornando-se um sistema bancário funcional básico.