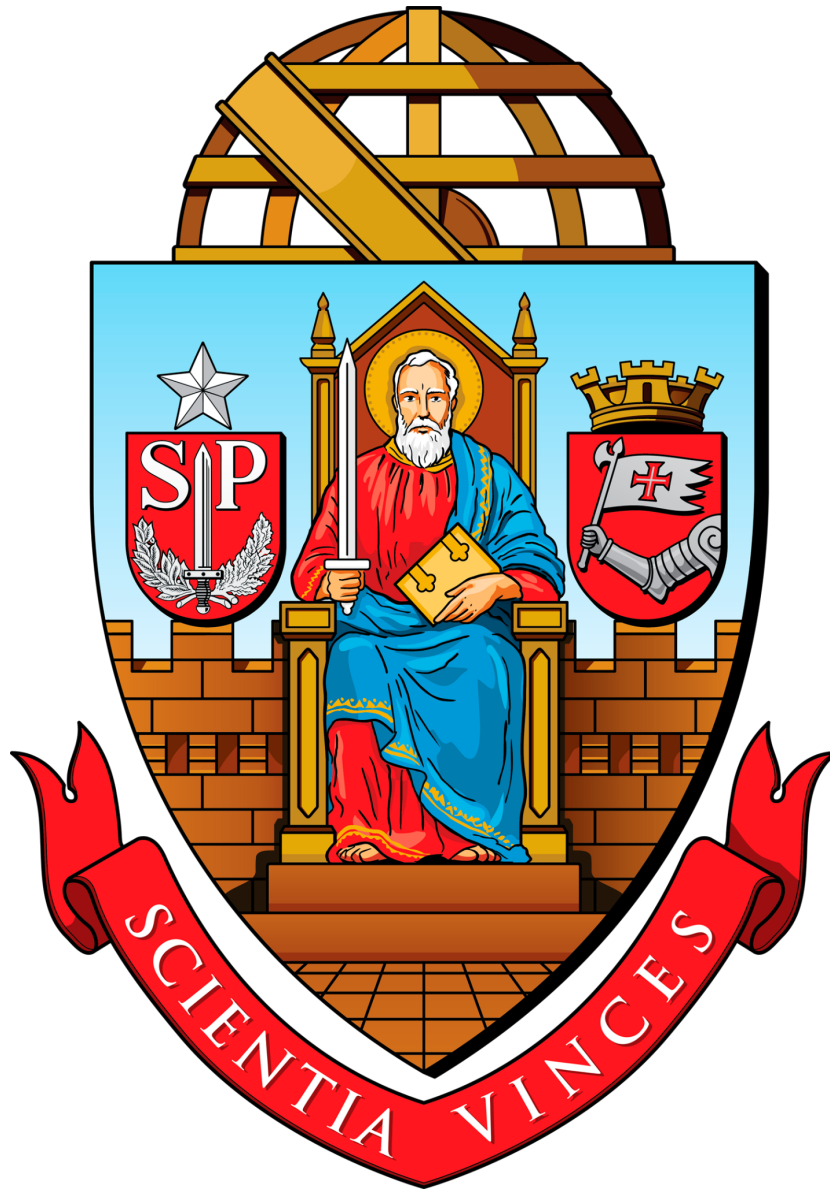


**UNIVERSIDADE DE SÃO PAULO INSTITUTO DE CIÊNCIAS MATEMÁTICAS E
COMPUTAÇÃO**

**Disciplina SSC0502 – Laboratório de Introdução à Ciência de Computação
Professor: Danilo Spatti**

Pseudocódigo e Fluxograma do UaiBank



Grupo:

Bruno Giacomini Volpe - 14651980

Enzo Vasconcellos Pagotto - 13632844

Vitor Marçal Brasil - 12822653

Início algoritmo UaiBank

```
Função imprimir_opcoes() {  
    Imprimir todas as opções disponíveis:  
    0. Fechar  
    1. Adicionar usuário  
    2. Adicionar vários usuários  
    3. Busca de usuário por ID  
    4. Transferência entre usuários  
    5. Remover usuário  
}
```

```
Função adicionar_usuario() {  
    Declare caractere nome, int idade, float saldo, struct novo_usuario, int id  
    Receba nome, idade, saldo  
    novo_usuario = Struct{nome, idade, saldo}  
    id = valor crescente começando em 0  
    Adicionar a struct desse usuários ao array de usuários  
    Atualizar o arquivo de texto com a lista atualizada de usuários  
    escreva "usuário adicionado"  
    return 0  
}
```

```
Função adicionarVariosUsuarios() {  
    declare int quantidade_usuários  
    Receba quantidade_usuários  
    repetir a quantidade de vezes informada {  
        chamar adicionarUsuario()  
        escreva "usuários adicionados"  
    }  
    return 0  
}
```

```

Função buscarUsuarioPorID(){
    declare int id
    Receba o id do usuário a ser buscado
    Procurar o usuário na array pelo ID
    Se o usuário for encontrado {
        escreva as informações do usuário na tela
    }
    Senão {
        escreva "usuário não encontrado"
    }
    return 0 }

```

```

Função transferenciaEntreUsuarios():{
    declare int id_origem, int id_destino, float valor_transferido
    Receba o id_origem
    Receba o id_destino
    Receba o valor_transferido
    Procurar os usuários na array pelos respectivos IDs
    se os usuários foram encontrados{
        se o saldo do usuário de origem é suficiente {
            usuarioDeOrigem.saldo = usuarioDeOrigem.saldo - valor_transferido
            usuarioDeDestino.saldo = usuarioDeDestino.saldo + valor_transferido
            escreva "Transferência realizada entre usuarioDeOrigem e
usuarioDeDestino"}
        senao {
            escreva "saldo insuficiente" }
    }
    senao{
        escreva "usuário não encontrado"}

    Atualizar array de usuários
    Atualizar o arquivo de texto
    return 0
}

```

```

Função removerUsuario() {
    declare int id_usuario_removido
    Ler o id_usuario_removido
    Procurar o usuário na array pelo ID
    Remover o usuário da array
    Atualizar o arquivo de texto com a lista atualizada de usuários
    escreva "usuário removido"
    return 0
}

```

```

Função main() {
    Struct Usuarios = {nome, saldo, idade}
    Defina int opcao
    Defina vetor dadosUsuarios Tipo: Usuarios
    Carregar os usuários do arquivo de texto para a array
    Imprima "Bem-Vindo ao UaiBank"
    Repetir até que o usuário escolha sair:
        imprimirOpcoes()
        receba opcao
        Executar a ação correspondente à opção escolhida usando switch

```

case:

```

    Case 0. Sair:
        Finalizar o programa
    Case 1. Adicionar usuário:
        chamar adicionarUsuario()
        break
    Case 2. Adicionar vários usuários:
        chamar adicionarVariosUsuarios()
        Break
    Case 3. Busca de usuário por ID:
        chamar buscarUsuarioPorID()
        Break
    Case 4. Transferência entre usuários:
        chamar transferenciaEntreUsuarios()
        Break
    Case 5. Remover usuário:
        chamar removerUsuario()
        Break
    Default
        escreva "opcao invalida"

```

```

}

```

Fim algoritmo

