



Jornada Bank

Alisson Manica e Iago Cappeletti

Esta é a primeira tela que aparecerá para o usuário. Nela, o usuário deve escolher entre 3 opções, tais como Logar, Cadastrar e sair do aplicativo

```
System.out.println("\n==== Jornada Bank =====");
System.out.println("#1 - Entrar");
System.out.println("#2 - Cadastrar usuário");
System.out.println("#3 - Sair do aplicativo");
System.out.println("=====");
System.out.print("Qual opção você deseja acessar? ");
int opcao = sc.nextInt();
System.out.println();

switch (opcao) {
    case 1:
        logar();
        break;
    case 2:
        registrarUsuario();
        break;
    case 3:
        System.out.println("Obrigado por acessar o aplicativo do Jornada Bank!");
        return;
    default:
        System.out.println("Opção inválida! Tente novamente.");
}
}
```

Caso seja escolhido para se cadastrar, vai para a tela de cadastro.

```
public static void registrarUsuario() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Digite seu nome completo: ");
    String nome = sc.nextLine();
    System.out.print("Digite seu CPF: ");
    String cpf = sc.next();
    System.out.print("Digite seu e-mail: ");
    String email = sc.next();
    System.out.print("Digite uma senha: ");
    String senha = sc.next();

    if (numUsers > 1) {
        if (validaRegistro(cpf, email)) {
            return;
        }
    }

    usuario[numUsers][0] = nome;
    usuario[numUsers][1] = cpf;
    usuario[numUsers][2] = email;
    usuario[numUsers][3] = senha;
    usuario[numUsers][4] = String.valueOf(new Random().nextInt( bound: 9999));
    usuario[numUsers][5] = "0";
    usuario[numUsers][6] = String.valueOf(numUsers);
    usuario[numUsers][7] = "0";
    numUsers++;

    System.out.println("\nUsuário cadastrado com sucesso!");
}
```

"Solicitamos que o usuário preencha os dados na tela, os quais serão guardados em uma matriz .

- Após isso, fazemos uma validação utilizando o CPF e o e-mail digitados. Caso já tenhamos os dados armazenados na matriz, retornamos com uma mensagem informando que essas informações já foram cadastradas em nosso "Banco de dados".

```
public static boolean validaRegistro(String cpf, String email) {
    for (int i = 1; i < numUsers; i++) {
        if (usuario[i][1].equals(cpf) || usuario[i][2].equals(email)) {
            System.out.println("Usuário já cadastrado!");
            return true;
        }
    }
    return false;
}
```

Caso opte por fazer o login no App e passe pela validação, aparecerá essa tela onde o usuário poderá fazer depósito, saque, transferência, criar uma reserva de emergência, alterar dados como e-mail e senha, e realizar a exclusão da conta.

```
public static void logado() {  
    Locale.setDefault(Locale.US);  
    Scanner input = new Scanner(System.in);  
    System.out.println("\n==== Jornada Bank ====");  
    System.out.println("Conta: " + usuario[userId][4] + "-" + usuario[userId][6] + "\nSaldo: R$ " + usuario[userId][5]  
        + "\nReserva de Emergência: R$ " + usuario[userId][7] + "\n");  
    System.out.println("#1 - Fazer um depósito");  
    System.out.println("#2 - Fazer um saque");  
    System.out.println("#3 - Fazer uma transferência");  
    System.out.println("#4 - Fazer uma reserva de emergência");  
    System.out.println("#5 - Alterar cadastro");  
    System.out.println("#6 - Excluir usuário");  
    System.out.println("#7 - Sair");  
    System.out.println("=====");  
    System.out.print("Qual opção você deseja acessar? ");  
    int opcao = input.nextInt();  
    System.out.println();  
}
```

O usuário selecionou que gostaria de fazer um depósito em sua conta. Será direcionado para a tela de depósito, onde é possível adicionar um valor com valor maior que zero. Quando finalizado o depósito, o usuário receberá uma confirmação e será mostrado(a) a ele(a) o seu novo saldo.

```
private static void fazerDeposito() {  
    Locale.setDefault(Locale.US);  
    System.out.println("Saldo inicial: R$ " + String.format("%.1f", Double.parseDouble(usuario[userId][5])));  
    System.out.print("Digite o valor do depósito: R$ ");  
    double valorDeposito = sc.nextDouble();  
  
    if (valorDeposito <= 0) {  
        System.out.println("Valor inválido");  
    } else {  
        usuario[userId][5] = String.valueOf(d: Double.parseDouble(usuario[userId][5]) + valorDeposito);  
        System.out.println("Depósitos realizado com sucesso.");  
        System.out.println("Saldo atual: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));  
        System.out.println();  
    }  
    logado();  
}
```

"O usuário selecionou que deseja fazer um saque em sua conta. Será direcionado(a) para a tela de saque onde o usuário poderá retirar qualquer valor disponível em sua conta. Caso ele tente retirar um valor maior do que o saldo disponível em sua conta, não será possível e voltará para a tela de seleção das operações que ele deseja

```
public static void fazerSaque() {
    Locale.setDefault(Locale.US);
    System.out.println("Saldo disponível em conta: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));
    System.out.print("Quanto você quer sacar? R$ ");
    double valorSaque = sc.nextDouble();

    if (valorSaque <= 0) {
        System.out.println("Valor inválido!\n");
    } else if (valorSaque > Double.parseDouble(usuario[userId][5])) {
        System.out.println("O valor selecionado é maior que o saldo disponível!\n");
    } else {
        double novoSaldo = Double.parseDouble(usuario[userId][5]) - valorSaque;
        usuario[userId][5] = Double.toString(novoSaldo);
        System.out.println("Saque realizado com sucesso.");
        System.out.println("Novo saldo disponível: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));
        System.out.println();
    }
    logado();
}
```

Código de transferencia

Caso o usuário deseje fazer uma transferência para outra pessoa que esteja cadastrada em nosso sistema, poderá realizar facilmente. Basta que o usuário informe o número da conta do destinatário e o dígito da conta, e em seguida informar o valor que deseja enviar.

Lembrando que não será possível fazer a transferência quando o valor informado for zero, quando o valor for superior ao saldo disponível ou quando for digitado um número de conta inválido.

```
public static void fazerTransferencia() {
    Locale.setDefault(Locale.US);
    System.out.println("Saldo disponível em conta: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));
    System.out.print("Qual é o valor da transferência? ");
    double valorTransferencia = sc.nextDouble();
    System.out.print("Para qual conta você quer transferir? ");
    String conta = sc.next();
    System.out.println("Digite o dígito único: ");
    String id = sc.next();

    if (Double.parseDouble(usuario[userId][5]) <= 0) {
        System.out.println("Você não tem saldo suficiente para essa transferência. Saldo disponível: " + usuario[userId][5]);
    } else if (valorTransferencia > Double.parseDouble(usuario[userId][5])) {
        System.out.println("O valor selecionado é maior que o saldo disponível!");
    } else if (valorTransferencia <= 0) {
        System.out.println("Valor inválido, tente novamente!");
    } else {
        for (int i = 0; i < numUsers; i++) {
            if (usuario[i][4].equals(conta) && usuario[i][6].equals(id)) {
                if (confirmaTransferencia(i)) {
                    double novoSaldo = Double.parseDouble(usuario[userId][5]) - valorTransferencia;
                    usuario[userId][5] = Double.toString(novoSaldo);
                    double novoSaldoDestino = Double.parseDouble(usuario[i][5]) + valorTransferencia;
                    usuario[i][5] = Double.toString(novoSaldoDestino);
                    System.out.println("Transferência realizada com sucesso!");
                }
            } else {
                if (i == numUsers - 1) {
                    System.out.println("Conta não encontrada, tente novamente!");
                }
            }
        }
    }
}
```


Aqui, o usuário tem a opção de confirmar as informações do destinatário e de confirmar ou cancelar a operação de transferência.

```
public static boolean confirmaTransferencia(int idUsuario) {  
    Locale.setDefault(Locale.US);  
    System.out.println("\n==== Jornada Bank ====");  
    System.out.println("Nome: " + usuario[idUsuario][0] + "\nConta: " + usuario[idUsuario][4] + "-" + usuario[idUsuario][6]);  
    System.out.println("#1 - Confirmar");  
    System.out.println("#2 - Cancelar");  
    System.out.println("=====");  
    int option = sc.nextInt();  
    switch (option) {  
        case 1:  
            return true;  
        case 2:  
            return false;  
        default:  
            System.out.println("Opção inválida!!");  
            return false;  
    }  
}
```


Caso o usuário queira realizar uma reserva de emergência, onde ele possa sacar a qualquer momento, temos como inspiração as caixinhas do Banco Nubank, que têm como objetivo ajudar o usuário a juntar um dinheirinho para realizar um sonho, como, por exemplo, uma viagem, comprar um carro ou até mesmo uma casa.

Acessando a reserva de emergência, o usuário poderá escolher entre três opções: fazer um depósito, sacar o valor disponível em sua reserva ou excluí-la. Mas não se preocupe, mesmo que o usuário exclua sua reserva, o dinheiro que tiver depositado lá voltará para o saldo de sua conta

```
public static void reservaDeEmergencia() {  
    Locale.setDefault(Locale.US);  
    System.out.println("\n==== Jornada Bank =====");  
    System.out.println("#1 - Depositar saldo de reservas de emergência");  
    System.out.println("#2 - Sacar saldo de reservas de emergência");  
    System.out.println("#3 - Excluir reservas de emergência");  
    System.out.println("=====");  
    System.out.print("Qual opção você deseja acessar? ");  
    int input = sc.nextInt();  
}
```

```

switch (input) {
    case 1:
        System.out.println("Saldo disponível em conta: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));
        System.out.print("Qual valor você quer colocar na sua reserva de emergência? R$ ");
        double valorCaixinha = sc.nextDouble();

        if (valorCaixinha <= 0) {
            System.out.println("Valor inválido!\n");
        } else if (valorCaixinha > Double.parseDouble(usuario[userId][5])) {
            System.out.println("O valor selecionado é maior que o saldo disponível!\n");
        } else {
            usuario[userId][5] = Double.toString(d: Double.parseDouble(usuario[userId][5]) - valorCaixinha);
            usuario[userId][7] = Double.toString(d: Double.parseDouble(usuario[userId][7]) + valorCaixinha);
            System.out.println("Reserva de Emergência criada com sucesso!\n");
        }
        break;

```

Depositar na reserva
de emergência

Sacar valor da reserva
de emergência

```

case 2:
    System.out.println("Saldo da reserva de Emergência: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][7])));
    System.out.print("Qual valor você quer sacar da sua reserva de emergência? R$ ");
    double valorDoSaqueCaixinha = sc.nextDouble();
    if (valorDoSaqueCaixinha <= 0) {
        System.out.println("Valor inválido!\n");
    } else if (valorDoSaqueCaixinha > Double.parseDouble(usuario[userId][7])) {
        System.out.println("O valor selecionado é maior que o saldo disponível!\n");
    } else {
        double updateSaldoCaixinha = Double.parseDouble(usuario[userId][7]) - valorDoSaqueCaixinha;
        usuario[userId][7] = Double.toString(updateSaldoCaixinha);
        double updateSaldoConta = Double.parseDouble(usuario[userId][5]) + valorDoSaqueCaixinha;
        usuario[userId][5] = Double.toString(updateSaldoConta);
        System.out.println("Saque da reserva de emergência efetuada com sucesso!\n");
        System.out.println("Novo saldo em conta: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][5])));
        System.out.println("Reserva de Emergência: R$ " + String.format("%.2f", Double.parseDouble(usuario[userId][7])));
    }

```

Excluir o reserva de emergência

```
case 3:
    System.out.println("\n==== Jornada Bank =====");
    System.out.println("Deseja excluir o a reserva de emergência? ");
    System.out.println("#1 - Sim");
    System.out.println("#2 - Não");
    System.out.println("=====");
    System.out.print("Qual opção você deseja acessar? ");
    input = sc.nextInt();

    switch (input) {
        case 1:
            double updateSaldoConta = Double.parseDouble(usuario[userId][5]) + Double.parseDouble(usuario[userId][7]);
            usuario[userId][5] = Double.toString(updateSaldoConta);
            usuario[userId][7] = "0";
            System.out.println("Caixinha reservas de emergência deletada com sucesso!");
            break;
        case 2:
            logado();
            break;
        default:
            System.out.println("Opção inválida!!");
    }
default:
    System.out.println("Opção inválida!!");
```

Finalização

Trabalhamos juntos neste projeto com muita vontade e determinação. E os nossos horários semelhantes ajudou bastante na hora de trabalhar no código.

Conseguimos desenvolver o que planejamos desde o início, utilizamos a plataforma Trello para nos manter organizados. Tivemos algumas dificuldades ao manipular matrizes, mas com pesquisas em sites e vídeos no YouTube, conseguimos entender como funcionava a manipulação dos dados. Tirando esse problema, conseguimos realizar o restante do projeto sem maiores dificuldades.

