Relatório

Objetivo do projeto

O objetivo do projeto é desenvolver uma aplicação em C de um caixa eletrônico, onde o usuário cria a senha do seu cartão e pode: consultar o seu saldo, realizar saques, depositar valores e ver o histórico de suas operações realizadas dentro desse "caixa eletrônico".

Descrição dos módulos

Módulo 1 Main -Feito por Italo:

O módulo principal (main) é responsável por iniciar e controlar todo o fluxo do sistema de caixa eletrônico. Ele começa com a geração de um número de cartão aleatório usando rand() e solicita que o usuário crie uma senha. Após a criação da conta, o sistema realiza a etapa de login, onde o número do cartão e a senha digitados são comparados com os armazenados.

Uma vez autenticado com sucesso, o usuário tem acesso a um menu com as seguintes opções:

1-Consultar saldo

2-Realizar depósito

3-Realizar saque

4-Sair

A navegação entre essas opções é feita com switch-case, onde cada opção chama a função correspondente dos módulos auxiliares. O saldo da conta é manipulado por ponteiro, garantindo que as alterações feitas em realizarDeposito() e realizarSaque() atualizem o valor corretamente.

Ao final, após o usuário escolher a opção de sair, a função exibirTotalOperacoes() é chamada, exibindo a quantidade total de transações realizadas. O main integra todos os módulos do sistema e garante que a experiência do usuário seja sequencial, segura e funcional do início ao fim.

Módulo 2 Saldo - Feito por Guilherme Kalil:

Esse módulo consiste em mostrar ao usuário seu saldo atual com base na função ‘void visualizarSaldo’ que carrega como parâmetro o ‘float saldo’, declarada no código principal (main) desse projeto. Após isso, ao usuário acessar a opção “Consultar saldo” no menu, é exibido o nome do usuário através do ‘printf’, seguido da exibição do seu saldo atual: ‘Seu saldo atual é: %,2f.\n, saldo’. Por fim, é feito a

estrutura de decisão, que mostra algumas informações conforme o saldo desse usuário é alterado. Ao utilizar o ‘if (saldo >= 100)’, é mostrado ao usuário que ele possui saldo suficiente para uma compra de valor alto. Se essa informação não proceder, o ‘else if (saldo >= 50)’ é acionado e mostra ao usuário que ele pode realizar uma compra em um valor considerável. Caso contrário, o ‘else’ entra em ação e mostra que o usuário possui um saldo baixo.

Modulo 3 Saque -Feito por Jullie:

A funcionalidade **Saque**, implementada na função *void realizarSaque(Usuario* u)*, permite que o usuário retire um valor da sua conta bancária. Através de uma interface no terminal, a função utiliza *scanf* para capturar o valor informado pelo usuário, e em seguida realiza duas validações principais:

- Verifica se o valor é maior que zero.
- Confirma se o usuário possui saldo suficiente para o saque.

Se ambas as condições forem satisfeitas, o valor é subtraído do campo *saldo* da *struct Usuario*, e a função *registrarOperacao("Saque", valor)* é chamada para registrar a transação. Em caso de erro, são exibidas mensagens apropriadas indicando o problema.

Essa função é essencial para simular transações reais de uma conta bancária, garantindo segurança e controle no fluxo financeiro do usuário.

Modulo 4 Deposito -Feito por Maria Clara:

Foi utilizado o extern void para chamar a variável externa declarada no Modulo 1, void realizarDeposito, foi utilizado para salvar as alterações que vão ser feitas na variável *saldo (que está declarado no Modulo 1) ele modifica diretamente o saldo original da conta, float deposito, cria uma variável para guardar o valor do deposito, printf e scanf, para perguntar o valor ao usuário e guardar o valor, um IF e Else para verificar se o valor depositado e positivo e por fim um printf para exibir o valor do deposisto e o saldo atualizado.

Modulo 5 Registro -Feito por Pedro:

O módulo de registro é responsável por armazenar e exibir o histórico das operações realizadas pelo usuário durante o uso do sistema. Ele utiliza uma struct chamada Operacao, que possui dois campos: tipo (string que indica o tipo da operação, como “Saque” ou “Depósito”) e valor (quantia correspondente à operação).

Foi definido um vetor global chamado historico[MAX_OPERACOES], que armazena até 100 operações. A variável totalOperacoes controla a quantidade de registros efetuados.

A função registrarOperacao(const char *tipo, float valor) é chamada sempre que uma operação de saque ou depósito é concluída. Ela copia os dados da operação

para a próxima posição livre no vetor historico e incrementa o contador totalOperacoes.

Ao final do uso do sistema, a função exibirTotalOperacoes() é chamada para mostrar um resumo final no terminal, informando ao usuário quantas operações foram realizadas no total. Esse módulo é essencial para fornecer rastreabilidade e controle das ações feitas dentro do sistema bancário simulado.

Dificuldades enfrentadas

A maior dificuldade enfrentada no grupo foi a união dos códigos e fazê-los funcionar, fizemos várias mudanças na estrutura para eles se completarem e tivemos que adicionar funções que não dominamos muito bem ainda como o “Void”.

Como foi feita a integração?

Utilizamos o GitHub para que todos tenham acesso aos códigos feitos e assim conseguir implementar um ao outro, um grupo no WhatsApp foi feito para o trabalho ser discutido e decidir como seria feito o código.