

Relatório sobre o minissistema bancário.

Ana Júlia Cruz - 14755096

Lucas Gonzales - 14675524

Rodrigo Kurosawa - 14606321

Nosso programa obteve sucesso no que havia sido planejado fazer. Os mecanismos de segurança dos dados do cliente em arquivo binário e, simultaneamente, criptografados com uma cifra de codificação possibilitou a confiabilidade no armazenamento das informações, visto que a natureza do arquivo binário já carrega consigo a dificuldade na visualização de informações importantes, que, em nosso caso, são a senha, o número de cadastro e o saldo dos nossos clientes.

Além disso, também tivemos a necessidade de elaborar a implementação do extrato bancário, que armazenaria todas as movimentações na conta do usuário, de modo que havia a necessidade de que fosse armazenado de forma individual e como arquivo texto para ser visível e legível. A primeira parte foi especialmente difícil, pois nosso grupo encontrou dificuldade em encontrar uma forma de tornar o extrato bancário como visualização exclusiva do cliente logado, em vez de um extrato único para todos. No entanto, o problema foi contornado quando tivemos a ideia de oferecer ao cliente a opção de imprimir o extrato apertando um dígito.

Outrossim, também tentamos ao máximo modularizar o programa e dividi-lo em sub-rotinas, as quais somaram juntas 14. Dessa forma o programa ficou bastante legível devido a secção das funcionalidades em blocos específicos, de modo a simplificar e otimizar o papel da função "main".

É válido citar também a nossa preocupação por indentar e comentar o código, sempre visando a facilidade de entendimento.

Ademais, também optamos por elaborar uma struct para armazenar os dados do cliente, no qual foi definido um tipo, por meio do typedef. Seguindo nessa lógica, criamos um ponteiro para uma variável cliente com o intuito de criar um vetor de struct com alocação dinâmica para salvar os dados dos clientes na sequência do array. Consequentemente, à medida que adicionávamos mais um usuário no cadastro, utilizamos da função realloc para aumentar em um tamanho de tipo_cliente o comprimento do vetor.

Dito isso, é importante dizer que no momento em que salvamos o arquivo binário, também definimos um índice inteiro sendo o início do código. Isso foi feito para que na hora da leitura fosse possível descobrir a quantidade de clientes que já haviam sido registrados. Dessa forma, pudemos ler o arquivo binário que continha as informações e transferi-las para nosso programa toda vez que compilávamos o código, havendo, portanto um salvamento das informações.