



UNIVERSIDADE FEDERAL DE LAVRAS

Kaio Vinícius Moraes, Otávio Resende, Fábio Júnior Oliveira

TRABALHO DE ESTRUTURA DE DADOS

**Lavras - MG
2018**

Fábio Júnio Rolin, Kaio Vinícius de Moraes, Otávio Resende

TRABALHO DE ESTRUTURA DE DADOS

Monografia apresentada à
Universidade Federal de Lavras,
como parte das exigências do
Curso de Ciência da
Computação,
para a obtenção do título de
Bacharel.

Prof (a). Dr (a). Joaquim Quinteiro Uchoa

Lavras - 2018
2018

SUMÁRIO

1	INTRODUÇÃO.....	4
2	DESENVOLVIMENTO.....	5
2.1	DESCRIÇÃO DO USO DO ARDUINO.....	6
2.2	DESCRIÇÃO DAS ESTRUTURAS UTILIZADAS.....	8
2.3	DESCRIÇÃO DO FUNCIONAMENTO DO CÓDIGO	10
3	CONCLUSÃO.....	11

INTRODUÇÃO

Com base nos conhecimentos adquiridos durante as aulas da disciplina de Estrutura de Dados, a proposta do trabalho envolve a criação de um sistema de cadastros utilizando determinadas estruturas de dados já vistas durante o curso, com o respectivo sistema sendo implementado em um dispositivo embarcado (Arduino).

O tema idealizado e desenvolvido pelo grupo é focado na simulação de um sistema de cadastro bancário, com funções de busca, remoção e inserção de dados, realizadas através do fornecimento do ID de identificação do usuário.

Com o desenrolar do relatório, será explicado, com maior profundidade, todo o processo de desenvolvimento do projeto prático, as estruturas de dados utilizadas, a construção e execução do código, além de considerações e conclusões com relação ao funcionamento e resultados obtidos pela implementação do sistema de cadastro proposto pelo grupo.

DESENVOLVIMENTO

A proposta do projeto prático envolve a simulação simples de um sistema de cadastro bancário, onde o usuário utilizará de seu ID de identificação para ter um acesso às funções específicas fornecidas pelo sistema construído. Os serviços disponibilizados pelo programa elaborado estão organizados em 4 opções:

- Consulta de saldo.
- Remoção de usuário do cadastro de usuários.
- Escrita de todos os usuários cadastrados, com seus respectivos saldos bancários descritos ao lado de seus ID's de identificação.
- Inserção de um novo usuário no cadastro.

Com relação a estrutura do código construído, o mesmo é executado em toda sua extensão no Arduino, e dentro dele são utilizadas duas estruturas de dados simultâneas:

- Tabela Hash (com a utilização de uma lista encadeada para tratamento de colisão de dados).
- Fila Encadeada.

A Tabela Hash é utilizada para armazenar os dados de cada usuário que será cadastrado no sistema, com os ID's de identificações e seus respectivos saldos bancários fornecidos para realizar esse processo de inserção. As informações do usuário (ID e Saldo) estão salvas em um arquivo texto, cujo o mesmo se encontra anexado em um cartão de memória SD. O conteúdo do cartão é lido pelo Arduino através de um módulo SD, consequentemente todos os dados fornecidos pelo arquivo serão salvos na Tabela Hash, cuja utilidade já foi destacada anteriormente.

Após a inserção nessa estrutura, o programa fará a identificação (conforme instruído) dos dez maiores saldos bancários registrados e executará o cadastro desses valores em uma Fila Encadeada. Posteriormente, a relevância dessa estrutura de dados se refletirá em uma espécie de bonificação ao cliente, onde cada usuário, cujo saldo esteja entre os dez maiores saldos de conta bancária salvos no sistema, receberá um bônus financeiro equivalente a 10% do valor armazenado em sua conta. Durante a execução do programa, mais precisamente após a opção de busca ser solicitada, será possível perceber o aumento de 10% do saldo relativo ao usuário em questão (considerando que o mesmo esteja qualificado para receber a bonificação destacada). Vale ressaltar que essa operação de gratificação será realizada somente uma vez por usuário.

DESCRIÇÃO DO USO DO ARDUINO

Primeiramente, segue na descrição os componentes utilizados em conjunto ao funcionamento do Arduino:

- 4 Leds.
- 4 Chaves Tácteis.
- 4 Resistores (220 Ohms).
- Jumpers.
- 1 Módulo SD.
- 1 Protoboard.

Partindo do princípio do seu funcionamento, o Arduino lê os dados dos usuários salvos no arquivo texto fornecido pelo cartão de memória SD (como já descrito nas primeiras conclusões do tópico de Desenvolvimento). e faz o registro de todas as informações em sua memória interna (conforme instruído pelo código produzido no projeto, que será comentado mais adiante). Logo após essa operação de cadastro, o Arduino estará apto a executar no terminal as respectivas funções programadas previamente.

As quatro chaves tácteis (inseridas na protoboard) serão utilizadas para controlar qual função será solicitada pelo usuário (a opção por cada ferramenta gerará uma saída de informações no terminal da IDE do Arduino) , conseqüentemente logo após a escolha, cada led conectado (que seja relacionado à chave pressionada) será aceso para identificar qual ação foi tomada pelo usufrutuário do sistema, e somente apagará quando o processo for finalizado.

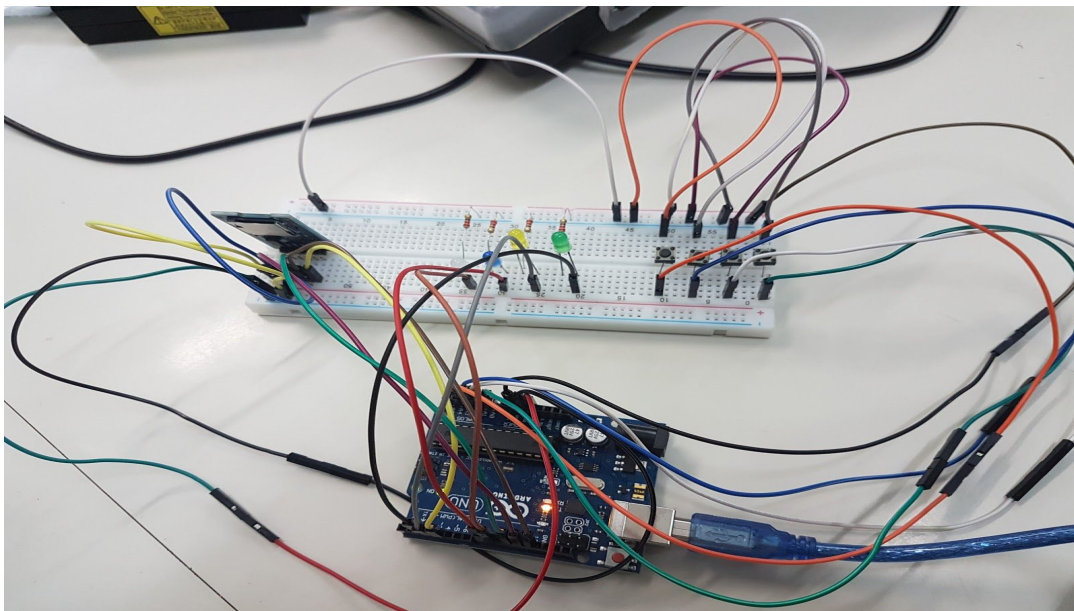
Quando solicitada, a opção de busca usará o ID do utilizador do programa para buscar dentro do banco de dados (criado na memória do Arduino) o seu saldo bancário respectivo. Conseqüentemente, a saída de dados do terminal retornará a informação requerida pelo usuário, e será possível perceber um leve aumento no valor armazenado em sua conta, através do sistema de bonificação financeira já citado anteriormente (essa alteração só será realizada caso a quantia armazenada na conta do cliente esteja salva entre os maiores saldos cadastrados).

A função de inserção remete a criação de um novo usufruidor do sistema, sendo necessário a entrada do valor do novo ID no terminal para ser armazenado no banco de dados do Arduino. É válido ressaltar que todo usuário criado através dessa opção tem um saldo inicial equivalente a 500 reais salvo em sua conta criada, não sendo possível modificar essas condições iniciais. Após isso, uma mensagem será imprimida na saída do terminal, indicando o sucesso do procedimento. Deve ser considerado que não haverá inserção de um mesmo ID já existente.

O mecanismo de escrita também é acionado por um dos botões. Logo após ser selecionado, o terminal fará a impressão de todos os ID's dos clientes cadastrados, em suas respectivas posições na Tabela Hash, com seus saldos pessoais impressos em seguida.

Por último, existe a funcionalidade da remoção, onde será solicitada a entrada do ID do usuário em questão para que o Arduino execute a busca pela memória interna, por consequência a remoção de seu código de identificação e de seu saldo. Caso não seja encontrado o cliente pesquisado, o terminal imprimirá uma mensagem de erro na saída de dados.

Todas essas funcionalidades são manipuladas pelo Arduino, já que o mesmo abriga toda a execução do código construído no projeto prático. A entrada de dados depende da conexão serial de um computador com o dispositivo embarcado.



-Projeto montado e em execução.

DESCRIÇÃO DAS ESTRUTURAS UTILIZADAS

Citadas em tópicos anteriores, as duas principais estruturas de dados utilizadas são uma Tabela Hash e uma Fila Encadeada. Iniciando a análise da primeira estrutura, percebe-se que a construção da mesma depende de uma classe `nohHash` e uma Struct com tipo “Dados”. Os dados guardados pela Struct são:

- Uma variável do tipo `Int` (inteira) para o ID de identificação.
- Uma variável do tipo `Long Int` para o saldo armazenado.
- Uma variável do tipo `Bool` (booleana) para verificar se um saldo já foi alterado (se um usuário tem um valor de saldo entre os maiores cadastrados, essa variável informa se o cliente em questão já teve seu saldo modificado ou não).

A classe `nohHash` contém atributos e métodos como:

- Um variável do tipo `nohHash` que aponta para um próximo endereço de uma outra variável do mesmo tipo.
- Uma Struct do tipo `Dados` (armazena as informações de um usuário).
- Método de construção da classe `nohHash`.
- Método de destruição da classe `nohHash`.

Considerando essas classes citadas, a construção da Tabela Hash (nomeada “`tabelaHash`” no código) envolve a criação dos atributos e métodos em questão:

- Um vetor de ponteiros da classe `nohHash`.
- Uma estrutura de dados do tipo Fila, que armazena os dez maiores saldos cadastrados.
- Uma variável do tipo inteira chamada “capacidade”, que representa o tamanho da Tabela Hash (tamanho do vetor de ponteiros).
- Método de construção da Hash.
- Método de destruição da Hash.
- Método de inserção da Hash.
- Método de percorrimento da Hash (imprime toda a estrutura).
- Método de remoção de um usuário da Hash.
- Método de busca de um determinado usuário (faz a busca pelo ID do cliente, e imprime o saldo).

Usando adequadamente os métodos de cada classe apresentada acima, a inserção das informações de um cliente começam sendo armazenadas em uma Struct do tipo “`Dados`”, logo o programa cria um novo nó do tipo `nohHash` (onde será depositada a Struct), em seguida irá fazer um cálculo usando uma função nomeada “`funcaoHash`” (retorna o resto da divisão do valor do ID pela capacidade do vetor de ponteiros da Tabela Hash), e usará o valor resultante como uma posição do vetor de ponteiros do tipo `nohHash` (atributo da Hash),

onde o novo nó poderá ser inserido (caso a posição do vetor já esteja sendo usada, o mesmo será inserido como o próximo elemento do último nó inserido anteriormente). É válido ressaltar que para uma estrutura de dados utilizar métodos e atributos de uma outra, é necessário declarar as respectivas classes como “amigas” (“friend class TipoDaClasse”).

Dando seguimento ao tópico, a existência da Fila Encadeada (nomeada “MaioresSaldos”, salva como atributo da tabelaHash) depende de uma classe denominada nohFila, que possui os seguintes atributos e métodos:

- Variável do tipo inteira que armazena o respectivo saldo.
- Um ponteiro do tipo da classe nohFila (aponta para o endereço do próximo nó).
- Método de construção do nó.
- Método de destruição do nó;

Já a Fila Encadeada possui a seguinte estrutura:

- Ponteiro do tipo nohFila para o primeiro nó da Fila (nomeado “primeiroFila”).
- Ponteiro do tipo nohFila para o último nó da Fila (nomeado “ultimoFila”).
- Método de construção da Fila.
- Método de destruição da Fila.
- Método de inserção na Fila.
- Método de remoção na Fila.
- Método “VerificaReajuste” (verifica se o ID buscado tem seu respectivo saldo entre os dez maiores da Fila MaioresSaldos).

A inserção de um saldo na Fila (caso o mesmo esteja entre os maiores saldos cadastrados no sistema) será feita após a criação de um nó do tipo nohFila. Se o nó criado for o primeiro a ser inserido na Fila, ele será reconhecido como o “primeiroFila”, caso contrário será conhecido como “ultimoFila” e terá seu ponteiro para um próximo nó ajustado. Os dados dos dez maiores saldos serão lidos de um arquivo texto (esse arquivo já contém todos os valores bancários ordenados) e serão inseridos na Fila atribuída na classe tabelaHash, configurando por final toda a estrutura do “MaioresSaldos”, assim finalizando a criação da Tabela Hash.

DESCRIÇÃO DO FUNCIONAMENTO DO CÓDIGO

Utilizando as definições de tópicos anteriores, a execução do código proposto se inicia com a inclusão das bibliotecas SdFat.h (usada para manipulação de arquivos utilizando-se do módulo SD) e da biblioteca SPI.h (usada para trocar informações via serial com os periféricos do Arduino).

Inicialmente, ao entrar no Void Setup() (função que executa suas instruções somente uma vez, quando o Arduino for iniciado) os dados dos usuários (ID e Saldo) são lidos por um arquivo texto denominado “cadastros.txt” para serem inseridos na Tabela Hash (processo descrito em tópicos anteriores). Após essa inserção, o arquivo é fechado, consequentemente será feita a leitura dos maiores saldos cadastrados de outro arquivo texto nomeado “maiores.txt”, para assim serem inseridos na Fila MaioresSaldos (atributo da classe tabelaHash do código).

Logo em seguida, é feita a manipulação das entradas do Arduino, com cada pino sendo setado para as funções descritas no programa. Feito isso, a execução caminhará para o Void Loop() (todas as suas instruções são executadas “infinitamente” até a intervenção do usuário), onde as operações de escolha do utilizador serão definidas. Como relatado no tópico “Descrição do uso do Arduino”, as funções disponíveis se resumem a busca de um saldo através do ID do usuário, remoção de um determinado cliente cadastrado no banco de dados do Arduino, percorrimento para imprimir todos os ID’s de identificação seguidos de seus respectivos saldos bancários e a inserção de um novo usufruidor no sistema bancário. A escolha por cada mecanismo do programa será selecionado por cada chave tátil correspondente, (ressaltando o fato de que cada led associado às funções permanecerá aceso até a finalização do processo), por consequência o programa continuará a ser executado inúmeras vezes até o usuário decidir finalizar as operações e fechar o terminal de saída de dados.

CONCLUSÃO

Com a finalização do projeto prático, foi possível ter uma experiência de manipulação de estruturas de dados com fins mais específicos e limitados a dispositivos embarcados. Sendo assim, lidando com certas restrições com relação ao Arduino, adquiriu-se conhecimento para elaborar otimizações na construção do código e também para criação e organização de um raciocínio lógico eficiente e adequado aos requisitos solicitados pela proposta do projeto prático.

Levantamos também considerações acerca da experiência adquirida com o trabalho exigido na manipulação de diversos componentes utilizados (juntamente com o Arduino) para construir um circuito que auxiliasse o Arduino na leitura e escrita de dados, usando recursos adquiridos em disciplinas em andamento ou já cursadas.

Para finalizar, também foi possível visualizar a evolução e a utilidade do conhecimento adquirido durante a disciplina de Estrutura de Dados, cujas ferramentas de programação fornecidas acabam por ter inúmeras implementações práticas vitais em inúmeras situações, sendo essa visão confirmada através de todo o processo de construção desse projeto prático proposto por esse grupo, assim como pelos resultados obtidos.

