



Universidade Federal do ABC

Universidade Federal do ABC
Disciplina Engenharia Unificada I

Caixa contadora de presença

Santo André,
3º Quadrimestre de 2019.

1. Introdução

Manter controle da assiduidade dos alunos em sala de aula é uma tarefa árdua quando o meio disponível é a lista de chamada em papel, que não é o método mais prático para contabilizar o número exato de quantos alunos estão efetivamente frequentando as aulas. Esse controle torna-se ainda mais difícil quando o tempo de aula é curto para a discussão e apresentação dos conteúdos curriculares.

Neste contexto, o presente projeto foi desenvolvido com o objetivo de facilitar a contabilização de presença dentro das salas de aula. Na UFABC, as listas de papel utilizadas para checagem de presença resultam em inúmeros problemas, como o desperdício demasiado de papel, a fraude quanto às assinaturas, o enorme trabalho para o professor desde carregar inúmeras listas para todos os lados até fazer a contagem dessas presenças e atribuir a porcentagem de cada aluno, e por fim o fato de precisar interromper os alunos durante a prova para registrar seu comparecimento.

A caixa registradora de presença vem com o objetivo de solucionar todas essas adversidades. De forma primária, seu funcionamento consiste no registro de cada RA pelo próprio aluno no início das aulas de uma dada matéria, esse registro criaria um banco de dados próprio do professor que seria alocado no cartão SD. Deste modo, em cada aula de cada matéria dada pelo professor a quem a caixa pertence, é criado um arquivo com data e todas as pessoas que compareceram, este arquivo também consta no cartão SD.

Logo, é possível perceber quantos problemas nos cercam no dia a dia letivo e como todos eles seriam facilmente resolvidos pela caixa contabilizadora de presença, pensada e desenvolvida para o presente projeto da disciplina de Engenharia Unificada I.

2. Objetivo

O projeto tem como objetivo desenvolver um sistema contabilizador de presença para ser utilizado, a princípio, em universidades e escolas. Esse resolveria problemas como fraudes, desperdício de papel, dificuldade para contabilizar o número de faltas de cada aluno, entre outros.

3. Fundamentação Teórica

- **Arduíno**

O Arduino foi criado no Instituto de Design Interativo Ivrea e, de acordo com o distribuidor, trata-se de uma plataforma eletrônica de hardware e software de uso simples. Suas placas são capazes de ler entradas como a luminosidade, o toque ou até uma mensagem na internet e transformá-las em algum sinal de saída - a rotação de um motor, uma luz, publicações online.

Dessa forma, o instrumento se consolidou como um ótimo ponto de partida para estudantes de eletrônica e computação, mas, dada sua ótima versatilidade, também passou a ser implementado em trabalhos mais complexos.

Figura 1: Arduino



- **Sensor RFID**

O sensor RFID é composto de uma antena, um transceptor e um transponder (ou etiqueta de rádio frequência). A antena transmite a informação, o transceptor faz a leitura do sinal e transfere a informação para um dispositivo leitor e o transponder converte as ondas de rádio do RFID para informações digitais. Essas informações, depois de serem convertidas, podem ser lidas por um computador e assim conseguirem ser, finalmente, analisadas.

Nesse projeto, o sensor RFID será utilizado para ler o código hexadecimal das carteirinhas aproximadas dele.

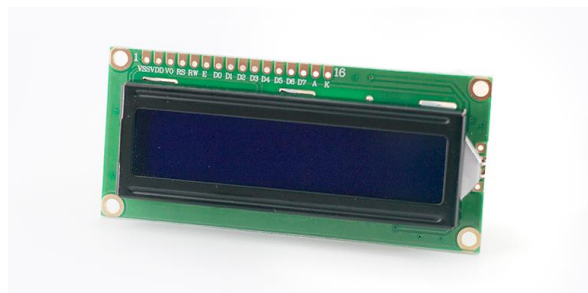
Figura 2: Sensor RFID.



- **Display LCD**

O módulo display LCD atua como uma interface de comunicação visual simples e barata para a nossa eletrônica embarcada. No projeto, o display LCD é responsável por fornecer todas as instruções necessárias para o usuário, e principalmente mostrar o resultado das ações realizadas. O seu funcionamento consiste na tecnologia LCD (Liquid Crystal Display), em que dependendo da energia enviada para a molécula de cristal líquido, ele permite passar uma certa quantia de luz por um pixel, sendo que dependendo da corrente enviada o ponto no display fica aceso ou apagado. Entretanto, esses comandos em baixo nível para controlar a luminosidade do display não são diretamente utilizados, pois seria extremamente complexo, sendo que o seu controle é feito por 4 (ou 8) pinos enviando sinais digitais. No projeto, o seu controle é feito utilizando uma biblioteca do Arduino.

Figura 3: Display LCD.

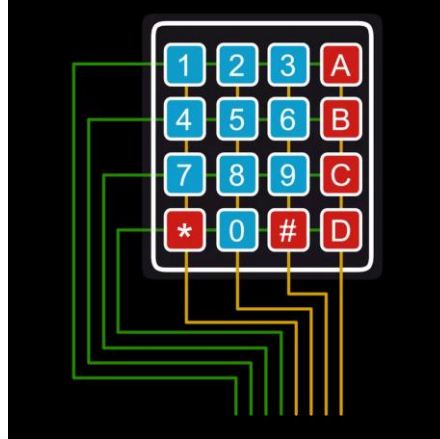


- **Teclado matricial**

Teclados geralmente são utilizados quando é necessário realizar uma interação entre o usuário e o sistema. No nosso projeto em específico, é necessário uma interação com o sistema

para navegar entre o menu, escolher o modo de operação ou mesmo entrar com um RA no caso da ausência de uma carteirinha. O teclado matricial possui um funcionamento similar ao de um botão comum, entretanto com vários deles acoplados.

Figura 4: Teclado Matricial e Suas Colunas e Linhas.



Na imagem acima é possível observar o seu funcionamento: ele possui 16 botões e apenas 8 saídas. Essas 8 saídas são conectadas nos pinos digitais do Arduino para serem lidos e identificados, e o processo para identificá-los consta em analisar as duplas de sinais recebidos pelos pinos do Arduino. Ao apertar o botão “*” por exemplo, a quarta e a quinta saída da esquerda para a direita terão sinais digitais 1 enquanto as outras terão 0. Esse caso só ocorre com o asterisco, sendo então que sempre que o Arduino ler esses dois pinos, considerará a entrada como sendo um “*”. O funcionamento é o mesmo para os outros 15 botões, sendo que cada um deles possui uma dupla de sinais de saída diferentes.

Figura 5: Teclado Matricial.



- Módulo SD

O módulo SD é um componente que realiza operações em um cartão SD através de sinais digitais enviados em seus pinos, que podem ser enviados por um Arduino para realizar o seu controle. O objetivo desse projeto é realizar a contabilização de presenças, e para isso é necessário armazenar esses dados em algum dispositivo. Por ser de fácil acesso e barato, escolhemos armazenar os dados em um cartão SD e para guardá-los nele é necessário o módulo SD para ser a interface entre ele e o Arduino. Outra função do cartão SD no projeto é o fato de servir como um banco de dados, em que registra os RAs cadastrados de cada carteirinha.

Figura 6: Módulo SD.



- **Buzzer**

Buzzer é um alto-falante piezoelétrico que usa o efeito piezo elétrico para geração de som, sendo possível variar a sua frequência de som utilizando os pinos digitais PWM do Arduino, que geram uma constante variação de tensão em seus terminais. No projeto, sua função é a de ser uma segunda interface para o sistema junto com o display LCD, sendo que ele emite um som sempre que a carteirinha for encostada no sistema.

Figura 7: Buzzer.



- **Bateria de lítio**

A bateria de lítio é composta por 4 partes internas: anodo, catodo, separador e camada de íons de lítio. O anodo é composto por grafite, que recebe elétrons; o catodo é composto por óxido de cobalto, que cede elétrons; o separador é geralmente composto por um material

semipermeável, o qual deixa passar apenas um tipo de elemento, nesse caso o lítio, separando este do óxido de cobalto; e a camada de íons de lítio seria esses íons separados do óxido.

A bateria funciona através de processo onde os íons de lítio percorrem do anodo para o catodo, passando pelo separador e se ligando ao óxido de cobalto. Nesse processo, sobra basicamente 1 elétron por íon de lítio, o qual é capturado pelo anodo de grafite, que o leva para fora para fornecer carga. A reação química acaba quando não há mais íons de lítio para serem transportados, fazendo com que a energia da bateria esgote. Quando recarregamos a bateria, o processo inverso ocorre, fazendo os íons de lítio voltarem para seu lugar novamente.

Nesse projeto, as baterias de lítio serão utilizadas para fornecer energia ao circuito.

Figura 7: Bateria de lítio



4. O Projeto

O desenvolvimento do projeto pode-se dividir em duas etapas: o desenvolvimento da eletrônica e o desenvolvimento da programação. Pesquisando na internet, foi possível encontrar esquemáticos de como se ligar individualmente cada um dos componentes no Arduino, sendo que cada componente possui também uma biblioteca própria que auxilia realizar o seu controle pelo código.

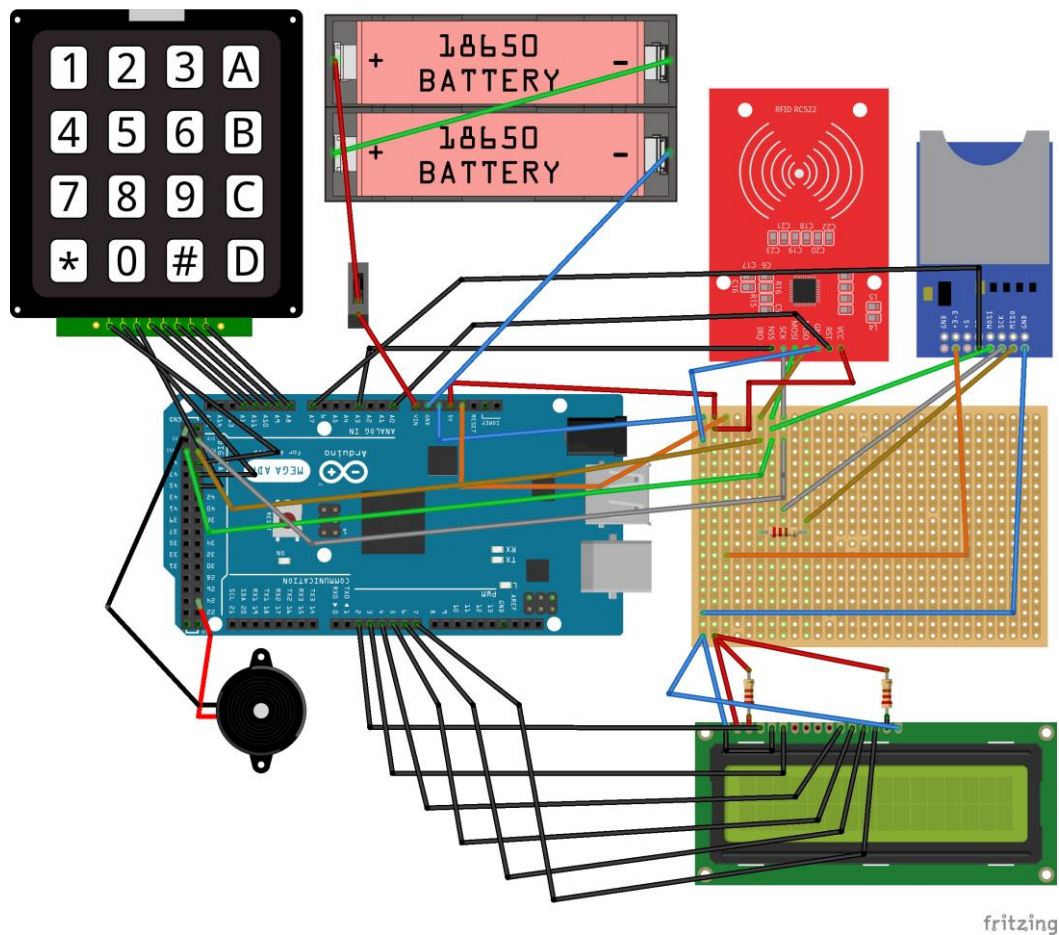
Em [1] há um esquemático de como ligar o display LCD ao Arduino, e de como programa-lo utilizando a biblioteca “LiquidCrystal.h”. Em [2] há um esquemático de como ligar

o teclado matricial no Arduino e alguns exemplos de comandos que podem ser utilizados com o auxílio da biblioteca “Keypad.h”. Em [3] há um esquemático de como realizar a ligação do sensor RFID e alguns exemplos de código utilizando a biblioteca “MFRC522.h”. Em [4] há um esquemático de como realizar a ligação do módulo SD com o Arduino e alguns exemplos de uso com a utilização da biblioteca “SD.h”, nativa da própria IDE do Arduino.

Os exemplos mencionados acima dão uma introdução de como utilizar cada um dos componentes, entretanto a programação desses elementos no projeto difere muito dos exemplos mostrados acima, visto que é uma aplicação completamente diferente. O apêndice contém o código desenvolvido para o registrador de presença.

O esquemático elétrico de todos os elementos juntos, de acordo com a programação feita, ficou da seguinte maneira:

Figura 8 - Esquemático do circuito elétrico.



Um detalhe que deu problemas durante o desenvolvimento do circuito é o fato de que o leitor RFID e o módulo SD usaram o mesmo protocolo de comunicação com o Arduino (SPI) e não funcionarem em conjunto, com suas ligações originais. Em outras palavras,

individualmente cada um dos componentes funcionava perfeitamente, entretanto ao integrar o sistema apenas o módulo SD funciona, enquanto o RFID não fornecia resposta. Após muita pesquisa na internet, foi constatado que outras pessoas também tiveram o mesmo problema de uma solução que funcionou muito bem foi a adição de um resistor de 220Ω no MISO do módulo SD. Dessa forma, o leitor RFID consegue funcionar simultaneamente com ele. Outro detalhe é que as baterias de lítio devem estar ligadas em série, de forma a fornecerem uma tensão maior que a tensão mínima de operação do Arduino (que é por volta de 7V), sendo que as baterias em série fornecem em torno de 7,5V. Por fim, há um switch em série com o VCC da bateria para ser possível ligar e desligar sem precisa remover as baterias do suporte.

Com o sistema elétrico montado, pode-se então iniciar o desenvolvimento do software. Como pode ser visto no apêndice, o código ficou extremamente extenso, por isso vale a pena destacar apenas a estrutura de seu funcionamento e o seu algoritmo. A estrutura do código foi feita utilizando o próprio relógio interno do Arduino, criando um sistema multitarefa, de forma que seja possível realizar qualquer ação quase simultaneamente, ou seja, o display atualiza as informações quase ao mesmo tempo em que o sensor RFID verifica se há algum cartão próximo. Dessa forma o usuário pode executar as ações na ordem que quiser, não sendo necessário um processo engessado, do tipo o usuário ler primeiro o display, depois esperar que aperte algum botão do teclado matricial ou aproxime alguma carteirinha para só então atualizar o display novamente. A grande vantagem disso é que todas as instruções aparecem na tela do display com intervalos de 2s independente da ação que o usuário esteja tomando.

Para entender o funcionamento do código, pode-se analisar o seguinte algoritmo:

Menu A: cadastrar RA

- Aproxima carteirinha

- Se RA já ta cadastrado, dá a opção de digitar o RA e alterar no banco de dados.

- Nesse passo, a pessoa precisa digitar '*' ou 'C', e há uma condição verificadora pra isso pra evitar erro.

- Se RA não está cadastrado, dá a opção de digitar o RA e cadastrar no banco de dados.

- Pressiona C

- sai do loop de cadastrar RA.

Menu B: fazer lista de presença

- Professor coloca a data para criar o nome do arquivo a se armazenar a lista

- Verifica se a data foi informada no formato certo, caso contrário avisa e pede para preencher de novo.

- Aproxima carteirinha

- Compara o ID dela com o banco de dados, e caso esteja cadastrado contabiliza presença. Caso contrário avisa o usuário que a carteirinha não está cadastrada.

- Pressiona D

- Entra em um campo para digitar o RA, e cadastrar na lista de presença. Serve para as pessoas que não estão com carteirinha no dia.

- Pressiona C

- Encerra a lista de presença.

* os arquivos ficam salvos no SD em pastas diferentes, para melhor organização. Caso deseje alterar a lista de presença de um dia (caso algum aluno chegue atrasado, após o professor encerrar) basta começar uma lista com a mesma data utilizada na anterior, que ele irá escrever no final do mesmo arquivo.

*em todos os campos de digitar (seja data ou RA) há a opção de apagar o caractere que foi escrito.

*todos os menus são auto explicativos, mostrando todas as opções a serem pressionadas no teclado para prosseguir e indicando se deu certo ou não.

5. Resultado

A Figura 8 mostra a caixa contabilizadora após ser montada com todos os componentes. Desta maneira, o equipamento está pronto para uso.

Figura 9 - Caixa Contabilizadora de Presença.



Fonte: elaborado pelos autores.

O programa possui, basicamente, duas opções de inicialização: uma responsável por cadastrar os RAs; outra para iniciar a lista de presença. Como, no projeto, não houve vínculo com alguma base de dados da UFABC, é necessário que o professor cadastre todas as carteirinhas dos alunos para que o mesmo possa ter um banco de dados que vincule o código hexadecimal da carteirinha com o RA do aluno. Assim sendo, as etapas discutidas abaixo serão referentes à metodologia de uso do equipamento.

Nas duas etapas iniciais, cadastro do RA e a lista de presença, dois arquivos serão criados no módulo SD, um como banco de dados das carteirinhas registradas e outro como lista de presença.

5.1. Cadastro do RA

Para entrar no modo de cadastro do RA, é necessário pressionar a tecla *A*. Este modo é responsável pela leitura das carteirinhas em um *looping*, sendo que ao final, para finalizar a leitura das carteirinhas, é necessário teclar a tecla *C*.

A leitura é realizada aproximando a carteirinha do sensor RFID (marcado pelo “Encoste Aqui” da Figura 7) na caixa contabilizadora, emitindo um som pela caixa, indicando que foi realizado a leitura do código hexadecimal da carteirinha. Em seguida, o sensor de LCD irá

solicitar para que seja digitado o RA do aluno¹. Assim que for finalizado o RA, basta clicar na tecla **C** e o display LCD irá notificar que o RA do aluno foi cadastrado, voltando ao *looping* inicial de cadastro de carteirinhas.

É possível, também, alterar o RA da carteirinha já cadastrado, caso algum erro tenha ocorrido no momento do registro. Para isso, basta entrar no modo de cadastro do RA e aproximar a carteirinha. O display LCD irá informar que a mesma já havia sido cadastrada, e assim, irá perguntar caso o aluno queira realizar alguma alteração. Para alterar, basta apertar a tecla **C** e cadastrar o novo RA.

5.2. Lista de Presença

A lista de presença é iniciada pressionando a tecla **B** ao finalizar o cadastro dos RAs ou ao ligar a caixa. Assim que entramos no modo da lista de presença, o display solicitará a data do dia em que será contabilizada a presença, devendo ser escrita no formato **dd#mm**².

Neste ponto, basta aproximar as carteirinhas já registradas perto do sensor RFID (“Encoste Aqui”), sendo exibida a mensagem “Carteirinha lida com sucesso” em caso de presença contabilizada. Caso alguma carteirinha não cadastrada seja colocada o display LCD irá informar que a mesma não possui cadastro, sendo necessário retornar à opção A (cadastro dos RAs) para que a mesma seja salva.

Em caso do aluno não possuir carteirinha, podemos apertar a tecla **D** para que o aluno consiga digitar o RA e o mesmo seja cadastrado na lista. Por fim, para finalizar a lista de chamada basta apertar a tecla **C**. Em caso do professor querer alterar alguma coisa, basta escrever uma lista com a mesma data para sobrescrever.

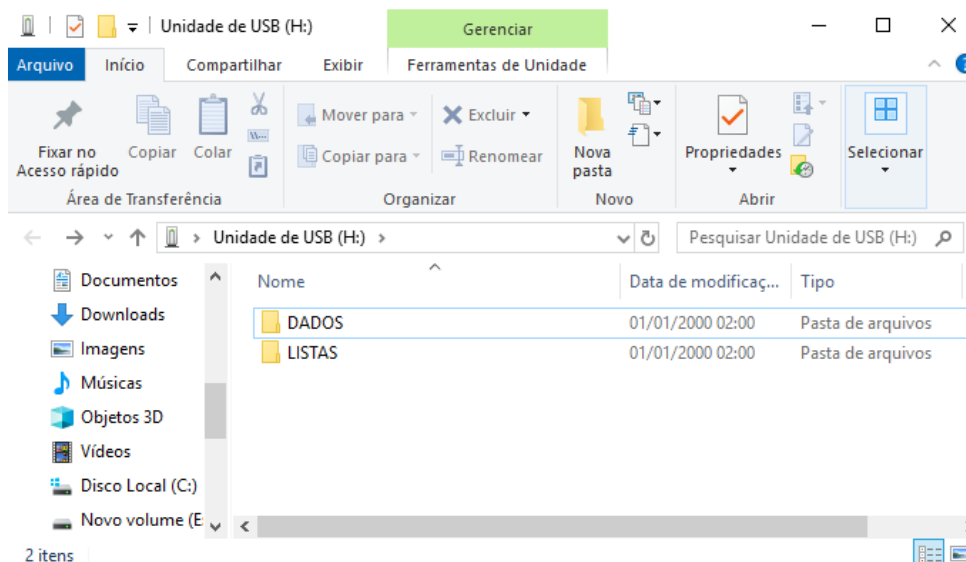
5.3. Arquivos

Duas pastas serão criadas: **DADOS** e **LISTAS**, conforme ilustrado pela Figura 9.

Figura 10 - Pastas criadas.

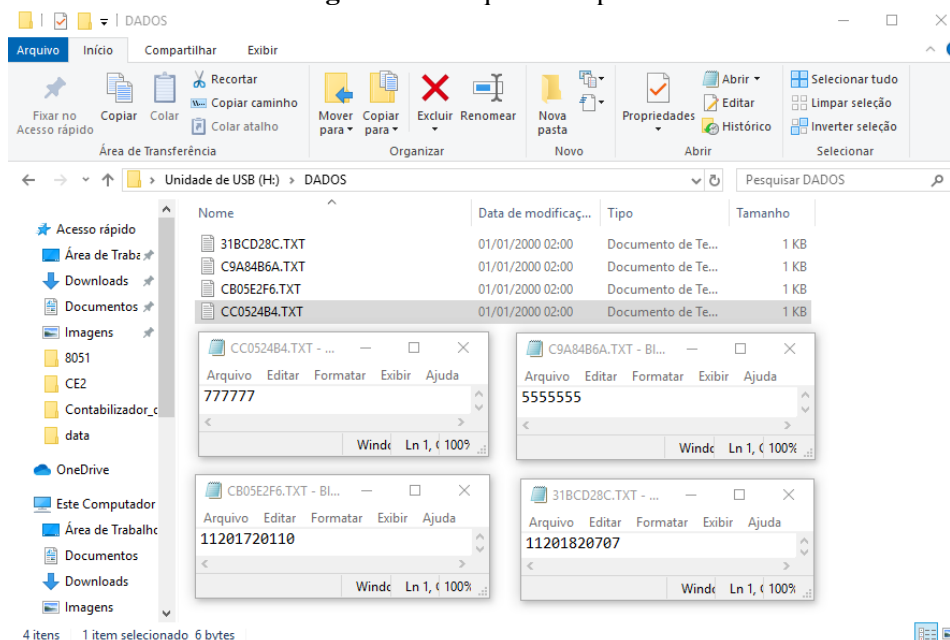
¹ Para apagar o RA digitado basta apertar o * no teclado matricial. Vale notar que o * é o “backspace” para corrigir eventuais erros de digitação no teclado matricial.

² Caso seja escrita de outra maneira, o display também irá notificar.



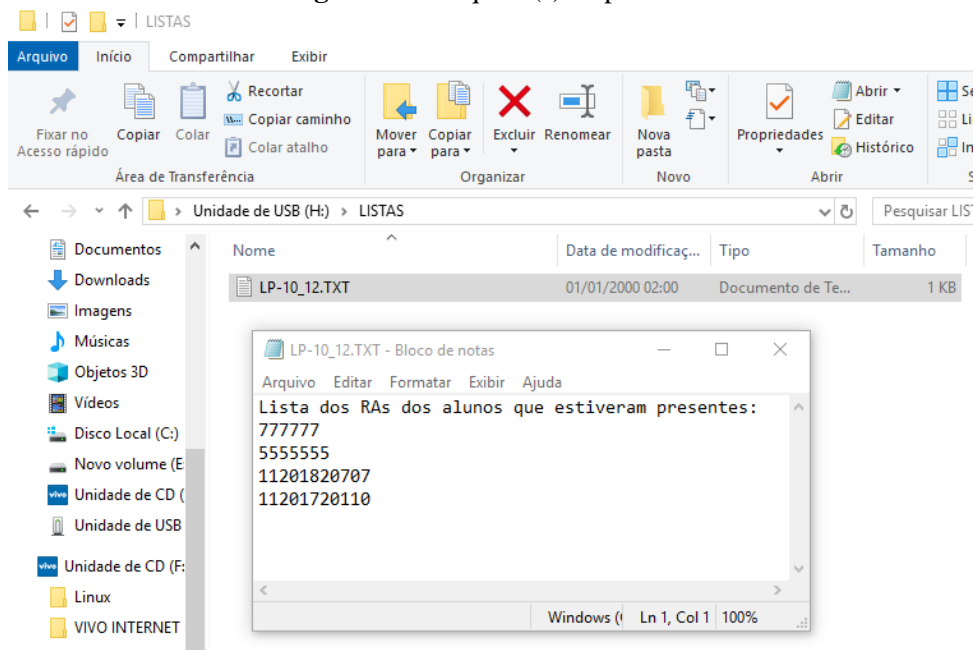
Cada pasta será responsável por conter as respectivas informações, referentes às duas seções anteriores. Na pasta *Dados*, podemos encontrar um arquivo de extensão “.txt” com o título do código hexadecimal referente à carteirinha registrada. Dentro do arquivo, estará o RA do aluno que o aluno digitou. Os arquivos da pasta *Dados* estão ilustrados na Figura 10.

Figura 11 - Arquivos da pasta dados.



A pasta “*Listas*” irá conter um arquivo também com extensão “.txt” com o título no seguinte formato: “LP-DD_MM.txt”, sendo a abreviatura para “Lista de Presença-dia_mês”. A Figura 11 mostra o arquivo dentro da pasta “*Listas*”, mostrando o RA dos alunos que estiveram presente em determinada aula.

Figura 12 - Arquivo(s) da pasta Listas.



6. Conclusão

Deste modo, é possível visualizar que, embora primária, a caixa contabilizadora de presença é uma solução viável para diversos problemas que possuímos no nosso cotidiano letivo, desde o desperdício de papel até a facilidade na contagem dos 75% de presença dos alunos.

É nítido que, para implementação futura, alguns aprimoramentos poderiam ser colocados em prática. Um exemplo é o cadastro não necessário do RA, neste projeto cada professor teria de criar sua própria biblioteca de dados com o cadastro prévio de cada RA pelo aluno a quem pertence, entretanto se possuíssemos acesso ao banco de dados da UFABC este cadastro não seria necessário e pouparia ainda mais trabalho aos envolvidos. Outra melhoria seria a obsolescência do cartão SD, aqui nós precisamos do mesmo para salvar tanto o banco de dados quanto as listas de presença registradas em cada dia, todavia a caixa poderia estar interligada com um sistema wi-fi para exportar estes dados diretamente para o computador, ou até mesmo para uma nuvem (o que necessitaria de um sistema um pouco mais complexo, porém possível). O Arduino inclusive poderia ser substituído por uma placa mais específica para com as necessidades do projeto, como ele possui muitas portas, que aqui a maioria permaneceram inutilizadas, seria mais prático, até mesmo para produção em massa, o desenvolvimento de uma placa específica que acabaria por baratear os custos. Por fim, poderia haver a construção de

uma caixinha mais prática, com tampa para proteger o display LCD e o teclado matricial, de tamanho e material adequado aos componentes do circuito e que comportasse cada um deles com segurança.

Por mais que o projeto possa ser melhorado em aspectos específicos visando praticidade e baixo custo quando pensamos em implementação e produção em massa, ele está 100% apto para uso em suas condições atuais e poderia ser posto em funcionamento a qualquer momento que garantiria com absoluta certeza a solução dos problemas os quais resultaram no desenvolvimento do mesmo. A caixa contabilizadora de presença é um protótipo que facilitará a vida de professores e alunos, e tem sua existência em grande escala garantida num futuro não tão distante.

7. Fontes

[1] <http://blog.baudaeletronica.com.br/conhecendo-biblioteca-liquidcrystal/>

[2] <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-teclado-matricial-de-membrana-4x4/>

[3] <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-kit-rfid-mfrc522/>

[4] <https://www.filipeflop.com/blog/cartao-sd-com-arduino/>

8. Apêndice

Código em Arduino:

```
#include <Keypad.h> //Biblioteca teclado matricial
#include <LiquidCrystal.h> //Biblioteca LCD
#include <MFRC522.h> //Biblioteca para o uso do módulo RFID
#include <SPI.h> //Biblioteca para comunicação SPI (utilizada pelo sensor RFID e módulo SD)
#include <SD.h> //Biblioteca para o uso do módulo SD

const byte qtdLinhas = 4; //Quantidade de linhas do teclado
const byte qtdColunas = 4; //Quantidade de colunas do teclado

//Setagem dos pinos utilizados pelo display LCD
```

```

const int rs = 2;
const int enable = 3;
const int d4 = 4;
const int d5 = 5;
const int d6 = 6;
const int d7 = 7;

//Determina o pino do buzzer
#define BZ 24
#define tempobuzzer 1500

//Determina os pinos utilizados pelo módulo RFID
#define SS_PIN A3
#define RST_PIN A0

//Determina o pino utilizado pelo módulo SD
#define chipselectSD A7

//Construção da matriz de caracteres do teclado matricial
char matriz_teclas[qtdLinhas][qtdColunas] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte PinosqtdLinhas[qtdLinhas] = {A8, A9, A10, A11}; //Pinos utilizados pelas linhas do
teclado matricial
byte PinosqtdColunas[qtdColunas] = {A12, 44, 46, 48}; //Pinos utilizados pelas colunas do
teclado matricial

//Criação do objeto teclado matricial
Keypad meuteclado = Keypad( makeKeymap(matriz_teclas), PinosqtdLinhas,
PinosqtdColunas, qtdLinhas, qtdColunas);

```



```
//Criação do objeto display LCD
```

```
LiquidCrystal lcd(rs, enable, d4, d5, d6, d7);
```

```
//Criação do objeto sensor RFID
```

```
MFRC522 rfid(SS_PIN, RST_PIN);
```

```
//Criação do objeto responsável por ler/escrever utilizando o módulo SD
```

```
File myFile;
```

```
//função que lê o ID do cartão que é aproximado do módulo RFID e retorna uma string
```

```
String ler_cartao()
```

```
{
```

```
  rfid.PICC_ReadCardSerial();
```

```
  String cardID = "";
```

```
  for (byte i = 0; i < rfid.uid.size; i++)
```

```
  {
```

```
    cardID.concat(String(rfid.uid.uidByte[i] < 0x10 ? "0" : ""));
```

```
    cardID.concat(String(rfid.uid.uidByte[i], HEX));
```

```
  }
```

```
  return cardID;
```

```
}
```

```
typedef struct Timer
```

```
{
```

```
  unsigned long start;//Armazena o tempo de quando foi iniciado o timer
```

```
  unsigned long timeout;//Tempo após o start para o estouro
```

```
};
```

```
unsigned long Now (void)
```

```
{
```

```
  return millis();//Retorna os milisegundos depois do reset
```

```
}
```

```
char TimerExpired (struct Timer *timer)
```

```
{  
    //Verifica se o timer estourou  
    if ( Now() > timer->start + timer->timeout )  
        return true;  
  
    return false;  
}
```

```
void TimerStart (struct Timer *timer)
```

```
{  
    timer->start = Now();//Reseta o timer gravando o horário atual  
}
```

```
//Rotinas menu inicial
```

```
Timer MenuInicial1 = {0, 2500}; //Muda o display lcd inicial para opção A
```

```
Timer MenuInicial2 = {0, 5000}; //Muda o display lcd inicial para opção B
```

```
Timer LeTeclado1 = {0, 10}; //Le o teclado a cada 10ms
```

```
//Rotinas menu cadastrar RA
```

```
Timer MenuCadRA1 = {0, 2500};
```

```
Timer MenuCadRA2 = {0, 5000};
```

```
Timer LeCartao2 = {5, 10};
```

```
Timer LeTeclado4 = {0, 10};
```

```
//Rotinas do menu cadastrar RA para entrar com RA
```

```
Timer MenuEntrarCadRA1 = {0, 2500};
```

```
Timer MenuEntrarCadRA2 = {0, 5000};
```

```
Timer MenuEntrarCadRA3 = {0, 7500};
```

```
Timer LeTeclado5 = {0, 10};
```

```
//Rotinas menu chamada
```

```
Timer MenuChamada1 = {0, 2500}; //Muda o display lcd para avisar para encostar a carteirinha
```

```
Timer MenuChamada2 = {0, 5000}; //Muda o display lcd para mostrar como entrar com RA
```

Timer MenuChamada3 = {0, 7500}; //Muda o display lcd para mostrar como finalizar a chamada

Timer LeCartao = {5, 10}; //Le o cartão a cada 10ms

Timer LeTeclado2 = {0, 10}; //Le o teclado a cada 10ms

//Rotinas menu chamada para entrar com RA

Timer MenuEntrarRA1 = {0, 2500};

Timer MenuEntrarRA2 = {0, 5000};

Timer MenuEntrarRA3 = {0, 7500};

Timer LeTeclado3 = {0, 10};

//Observação sobre os menus: para não utilizar delay e os fazer travar, fiz um esquema simples. Eu deixo cada opção como um múltiplo tempo

//anterior, exemplo: se quero exibir 2 opções a cada 2s, deixo a primeira com 2s e a segunda com 4s, se tivesse uma terceira ela teria 6s.

//A sacada pra esse timer é ir resetando as opções a cada chamada, na opção 1 eu reseto o tempo da 1, na opção 2 eu reseto o tempo da 1 e da 2,

//na opção 3 eu reseto o tempo da 1, da 2 e da 3. É complexo de entender, mas assim funciona.

O princípio de funcionamento é uma forma de não deixar

//em regime permanente, em que a primeira opção oscilando a cada 1s ficaria o tempo todo na tela não deixando a com 2s aparecer. Resetar a de 1s faz

//com que a de 2s apareça.

void setup() {

 //Comandos de inicialização:

Serial.begin(9600); //inicia comunicação com a serial do computador (serve para debugar utilizando o monitor serial)

SPI.begin(); //inicia a comunicação SPI (é utilizado pelo módulo SD e RFID)

lcd.begin(16, 2); //inicia o display LCD

rfid.PCD_Init(); //inicia o módulo RFID

 //inicia o módulo SD

SD.begin(chipselectSD);

```

//cria os diretórios caso não existam
if(!SD.exists("/DADOS")) SD.mkdir("/DADOS");
if(!SD.exists("/LISTAS")) SD.mkdir("/LISTAS");
}

void loop() {
  //Menu inicial
  char tecla_pressionada; //tecla utilizada para navegar no primeiro menu
  while (1)
  {

    if (TimerExpired(&MenuInicial1)) //Muda o display lcd inicial para opção A
    {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Selecione o modo");
      lcd.setCursor(0, 1);
      lcd.print("A-cadastrar RA");
      TimerStart(&MenuInicial1);
    }

    if (TimerExpired(&MenuInicial2)) //Muda o display lcd inicial para opção B
    {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Selecione o modo");
      lcd.setCursor(0, 1);
      lcd.print("B-iniciar lista");
      TimerStart(&MenuInicial1);
      TimerStart(&MenuInicial2);
    }

    if (TimerExpired(&LeTeclado1)) //lê a tecla e direciona para a opção desejada
    {

```

```

        tecla_pressionada = meuteclado.getKey();
        TimerStart(&LeTeclado1);
        break;
    }

}

switch (tecla_pressionada)
{
    case 'A': //caso para cadastrar o RA

        while (1)
        {
            if (TimerExpired(&MenuCadRA1))
            {
                lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print("Aproxime a");
                lcd.setCursor(0, 1);
                lcd.print("carteirinha");
                TimerStart(&MenuCadRA1);
            }

            if (TimerExpired(&MenuCadRA2))
            {
                lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print("Digite C para");
                lcd.setCursor(0, 1);
                lcd.print("finalizar");
                TimerStart(&MenuCadRA1);
                TimerStart(&MenuCadRA2);
            }

            if (TimerExpired(&LeTeclado4))

```

```
{
    tecla_pressionada = meuteclado.getKey();
    if (tecla_pressionada == 'C') //se pressionado C, avisa no display e finaliza o cadastro. Se
    não pressionado, nada acontece.
```

```
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Cadastro");
    lcd.setCursor(0, 1);
    lcd.print("finalizado!");
    delay(2500);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Voltando ao");
    lcd.setCursor(0, 1);
    lcd.print("menu inicial");
    delay(2500);
    TimerStart(&LeTeclado4);
    break;
}
TimerStart(&LeTeclado4);
}
```

```
if (TimerExpired(&LeCartao2)) //lê a tecla e direciona para a opção desejada
{
    if (rfid.PICC_IsNewCardPresent())
    {
        tone(BZ,800,tempobuzzer);
        //Mostrando que leu no display LCD e armazenando o ID da carteirinha
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Carteirinha lida");
        lcd.setCursor(0, 1);
        lcd.print("com sucesso!");
    }
}
```

```
String cardID = ler_cartao();  
cardID += ".txt";  
delay(2500);  
String diretorio = "/DADOS/"; //ate este momento o cardID contem o ID da carteirinha  
e a extensão txt (ex: CC02953F.txt). aqui eu acrescento o diretório para ficar  
/DADOS/CC02953F.txt por exemplo.
```

```
diretorio += cardID;  
//Verifica se a carteirinha ja foi cadastrada  
if (SD.exists(diretorio))  
{  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Carteirinha ja");  
    lcd.setCursor(0, 1);  
    lcd.print("possui cadastro");  
    delay(2500);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Deseja alterar?");  
    lcd.setCursor(0, 1);  
    lcd.print("Sim:'C' Nao:'*");  
    char tecladotemp;  
    do  
    {  
        tecladotemp = meuteclado.waitForKey();  
        if ((tecladotemp != 'C') && (tecladotemp != '*'))  
        {  
            lcd.clear();  
            lcd.setCursor(0, 0);  
            lcd.print("Voce deve teclar");  
            lcd.setCursor(0, 1);  
            lcd.print("'C' ou '*");  
            delay(2500);  
            lcd.clear();
```

```

    lcd.setCursor(0, 0);
    lcd.print("Deseja alterar?");
    lcd.setCursor(0, 1);
    lcd.print("Sim:'C' Nao:'*");
}
}
while ((tecladotemp != 'C') && (tecladotemp != '*'));
if (tecladotemp == 'C')
{
    String RA = "";

    while (1)
    {
        if (TimerExpired(&MenuEntrarCadRA1)) //Muda o display lcd para avisar para
entrar com o RA
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Digite seu RA:");
            lcd.setCursor(0, 1);
            lcd.print(RA);
            TimerStart(&MenuEntrarCadRA1);
        }

        if (TimerExpired(&MenuEntrarCadRA2)) //Muda o display lcd para mostrar como
enviar o RA
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("C- enviar");
            lcd.setCursor(0, 1);
            lcd.print(RA);
            TimerStart(&MenuEntrarCadRA1);
            TimerStart(&MenuEntrarCadRA2);

```



```
}
```

`if (TimerExpired(&MenuEntrarCadRA3))` //Muda o display lcd para mostrar como apagar um caractere

```
{  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("*-apaga character");  
    lcd.setCursor(0, 1);  
    lcd.print(RA);  
    TimerStart(&MenuEntrarCadRA1);  
    TimerStart(&MenuEntrarCadRA2);  
    TimerStart(&MenuEntrarCadRA3);  
}
```

`if (TimerExpired(&LeTeclado5))` //Le o teclado matricial, e ao comando da letra C envia o RA para a lista de presença

```
{  
    char digito = meuteclado.getKey();  
    if (digito != 'C')  
    {  
        if (digito == '*') //apaga um caractere quando '*' for pressionado e printa no LCD  
        {  
            int lastindex = RA.length() - 1;  
            RA.setCharAt(lastindex, ' ');  
            lcd.setCursor(0, 1);  
            lcd.print(RA);  
            RA.trim();  
            TimerStart(&LeTeclado5);  
        }  
    }
```

`else if (digito)` //acrescenta um caractere (se pressionado) à variavel RA e printa no LCD

```
{  
    RA += digito;
```

```

        lcd.setCursor(0, 1);
        lcd.print(RA);
        TimerStart(&LeTeclado5);
    }
}
else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("RA da carteirinh");
    lcd.setCursor(0, 1);
    lcd.print("foi alterado");
    delay(2500);
    SD.remove(diretorio);
    myFile = SD.open(diretorio, FILE_WRITE);
    myFile.print(RA);
    myFile.close();
    TimerStart(&LeTeclado5);
    break;
}
}
}
}
else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("RA da carteirinh");
    lcd.setCursor(0, 1);
    lcd.print("nao foi alterado");
    delay(2500);
}
}
else

```

```

{
    String RA = "";

    while (1)
    {
        if (TimerExpired(&MenuEntrarCadRA1)) //Muda o display lcd para avisar para
entrar com o RA
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Digite seu RA:");
            lcd.setCursor(0, 1);
            lcd.print(RA);
            TimerStart(&MenuEntrarCadRA1);
        }

        if (TimerExpired(&MenuEntrarCadRA2)) //Muda o display lcd para mostrar como
enviar o RA
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("C- enviar");
            lcd.setCursor(0, 1);
            lcd.print(RA);
            TimerStart(&MenuEntrarCadRA1);
            TimerStart(&MenuEntrarCadRA2);
        }

        if (TimerExpired(&MenuEntrarCadRA3)) //Muda o display lcd para mostrar como
apagar um caractere
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("*-apaga caractere");

```

```

    lcd.setCursor(0, 1);
    lcd.print(RA);
    TimerStart(&MenuEntrarCadRA1);
    TimerStart(&MenuEntrarCadRA2);
    TimerStart(&MenuEntrarCadRA3);
}

```

if (TimerExpired(&LeTeclado5)) //Le o teclado matricial, e ao comando da letra C envia o RA para a lista de presença

```

{
    char digito = meuteclado.getKey();
    if (digito != 'C')
    {
        if (digito == '*') //apaga um caractere quando '*' for pressionado e printa no LCD
        {
            int lastindex = RA.length() - 1;
            RA.setCharAt(lastindex, ' ');
            lcd.setCursor(0, 1);
            lcd.print(RA);
            RA.trim();
            TimerStart(&LeTeclado5);
        }
        else if (digito) //acrescenta um caractere (se pressionado) à variavel RA e printa no

```

LCD

```

{
    RA += digito;
    lcd.setCursor(0, 1);
    lcd.print(RA);
    TimerStart(&LeTeclado5);
}
}
else
{
    lcd.clear();

```

```

        lcd.setCursor(0, 0);
        lcd.print("RA cadastrado no");
        lcd.setCursor(0, 1);
        lcd.print("banco de dados");
        delay(2500);
        myFile = SD.open(diretorio, FILE_WRITE);
        myFile.print(RA);
        myFile.close();
        TimerStart(&LeTeclado5);
        break;
    }
}
}
}
}
TimerStart(&LeCartao2);
}
}
break;
case 'B': //caso para fazer a lista de presença
    String nomelista = "LP-"; //cria a variável que armazenará a lista de presença
    String data = "";
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Insira a data");
    lcd.setCursor(0, 1);
    lcd.print("de hoje");
    delay(2500);
    while (1)
    {
        if (TimerExpired(&MenuEntrarRA1)) //Muda o display lcd para mostrar o formato de data
a se entrar
        {
            lcd.clear();

```

```

    lcd.setCursor(0, 0);
    lcd.print("Data: DD#MM");
    lcd.setCursor(0, 1);
    lcd.print(data);
    TimerStart(&MenuEntrarRA1);
}

```

if (TimerExpired(&MenuEntrarRA2)) //Muda o display lcd para mostrar como enviar a data

```

{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("C- enviar");
    lcd.setCursor(0, 1);
    lcd.print(data);
    TimerStart(&MenuEntrarRA1);
    TimerStart(&MenuEntrarRA2);
}

```

if (TimerExpired(&MenuEntrarRA3)) //Muda o display lcd para mostrar como apagar um caractere

```

{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("*-apaga caracter");
    lcd.setCursor(0, 1);
    lcd.print(data);
    TimerStart(&MenuEntrarRA1);
    TimerStart(&MenuEntrarRA2);
    TimerStart(&MenuEntrarRA3);
}

```

if (TimerExpired(&LeTeclado3)) //Le o teclado matricial, e ao comando da letra C envia a data

```

{
    char digito = meuteclado.getKey();
    if (digito != 'C')
    {
        if (digito == '*') //apaga um caractere quando '*' for pressionado e printa no LCD
        {
            int lastindex = data.length() - 1;
            data.setCharAt(lastindex, ' ');
            lcd.setCursor(0, 1);
            lcd.print(data);
            data.trim();
            TimerStart(&LeTeclado3);
        }
        else if (digito) //acrescenta um caractere (se pressionado) à variavel data e printa no
LCD
        {
            data += digito;
            lcd.setCursor(0, 1);
            lcd.print(data);
            TimerStart(&LeTeclado3);
        }
    }
    else
    {
        if (data.length() != 5 || isDigit(data.charAt(2))) //verifica se a data esta no formato válido
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Formato invalido");
            lcd.setCursor(0, 1);
            lcd.print("Digite: DD#MM");
            data = "";
            delay(2500);
        }
    }
}

```

```

else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Data inserida!");
    lcd.setCursor(0, 1);
    lcd.print("Iniciando lista");
    delay(2000);
    String dia = "";
    String mes = "";
    dia += data.charAt(0);
    dia += data.charAt(1);
    mes += data.charAt(3);
    mes += data.charAt(4);
    nomelista += dia;
    nomelista += '_';
    nomelista += mes;
    break;
}
}
}
}

```

```
nomelista += ".txt";
```

`String` diretoriolista = `"/LISTAS/";` //ate este momento a 'nomelista' contem a data a extensão txt (ex: LP-01_12.txt). aqui eu acrescento o diretório para ficar `/LISTAS/LP-01_12.txt` por exemplo.

```

diretoriolista += nomelista;
if(!SD.exists(diretoriolista))
{
    myFile = SD.open(diretoriolista, FILE_WRITE);
    myFile.println("Lista dos RAs dos alunos que estiveram presentes:");
    myFile.close();
}

```



```

while (1)
{

    if (TimerExpired(&MenuChamada1)) //Muda o display lcd para avisar para encostar a
carteirinha
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Aproxime a sua");
        lcd.setCursor(0, 1);
        lcd.print("carteirinha");
        TimerStart(&MenuChamada1);
    }

    if (TimerExpired(&MenuChamada2)) //Muda o display lcd para mostrar como entrar com
RA
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Para entrar com");
        lcd.setCursor(0, 1);
        lcd.print("um RA tecle D");
        TimerStart(&MenuChamada1);
        TimerStart(&MenuChamada2);
    }

    if (TimerExpired(&MenuChamada3)) //Muda o display lcd para mostrar como finalizar a
chamada
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Para finalizar");
        lcd.setCursor(0, 1);
        lcd.print("a lista tecle C");
    }
}

```

```

TimerStart(&MenuChamada1);
TimerStart(&MenuChamada2);
TimerStart(&MenuChamada3);
}

```

`if (TimerExpired(&LeCartao))` //rotina que lê o cartão, compara com o banco de dados e contabiliza a presença no SD

```

{
    if (rfid.PICC_IsNewCardPresent()) {
        //Mostrando que leu no display LCD
        tone(BZ,800,tempobuzzer);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Carteirinha lida");
        lcd.setCursor(0, 1);
        lcd.print("com sucesso!");
        String cardID = ler_cartao();
        delay(2500);

        //Verificando se o RA foi cadastrado e lendo o RA dentro do arquivo
        cardID += ".txt";
        cardID.toUpperCase();
        String RA = "";
        String diretorio = "/DADOS/"; //ate este momento o cardID contem o ID da carteirinha
        e a extensão txt (ex: CC02953F.txt). aqui eu acrescento o diretório para ficar
        /DADOS/CC02953F.txt por exemplo.
        diretorio += cardID;
        if (SD.exists(diretorio))
        {
            myFile = SD.open(diretorio);
            while (myFile.available())
            {
                RA += (char)myFile.read();
            }

```

```

myFile.close();
myFile = SD.open(diretoriolista, FILE_WRITE);
myFile.println(RA);
myFile.close();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("RA cadastrado");
lcd.setCursor(0, 1);
lcd.print("na lista");
delay(2500);
}
else //informa que a carteirinha não está cadastrada
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Carteirinha nao");
  lcd.setCursor(0, 1);
  lcd.print("cadastrada");
  delay(2500);
}
}
TimerStart(&LeCartao);
}

```

if (TimerExpired(&LeTeclado2)) //rotina utilizada para finalizar a lista de presença ou entrar com novo RA

```

{
  char tempteclado = meuteclado.getKey();
  if (tempteclado == 'C')
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Lista de chamada");
    lcd.setCursor(0, 1);

```

```

    lcd.print("finalizada!");
    delay(2500);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Voltando ao");
    lcd.setCursor(0, 1);
    lcd.print("menu inicial");
    delay(2500);
    TimerStart(&LeTeclado2);
    break;
}
if (tempteclado == 'D')
{
    String RA = "";

    while (1)
    {

```

o RA

```

    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Digite seu RA:");
        lcd.setCursor(0, 1);
        lcd.print(RA);
        TimerStart(&MenuEntrarRA1);
    }

```

o RA

```

    {
        lcd.clear();
        lcd.setCursor(0, 0);

```

```

    lcd.print("C- enviar");
    lcd.setCursor(0, 1);
    lcd.print(RA);
    TimerStart(&MenuEntrarRA1);
    TimerStart(&MenuEntrarRA2);
}

```

if (TimerExpired(&MenuEntrarRA3)) //Muda o display lcd para mostrar como apagar um caractere

```

{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("*-apaga caracter");
    lcd.setCursor(0, 1);
    lcd.print(RA);
    TimerStart(&MenuEntrarRA1);
    TimerStart(&MenuEntrarRA2);
    TimerStart(&MenuEntrarRA3);
}

```

if (TimerExpired(&LeTeclado3)) //Le o teclado matricial, e ao comando da letra C envia o RA para a lista de presença

```

{
    char digito = meuteclado.getKey();
    if (digito != 'C')
    {
        if (digito == '*') //apaga um caractere quando '*' for pressionado e printa no LCD
        {
            int lastindex = RA.length() - 1;
            RA.setCharAt(lastindex, ' ');
            lcd.setCursor(0, 1);
            lcd.print(RA);
            RA.trim();
            TimerStart(&LeTeclado3);
        }
    }
}

```

```
    }  
    else if (digito) //acrescenta um caractere (se pressionado) à variavel RA e printa no
```

LCD

```
    {  
        RA += digito;  
        lcd.setCursor(0, 1);  
        lcd.print(RA);  
        TimerStart(&LeTeclado3);  
    }  
}  
else  
{  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("RA cadastrado");  
    lcd.setCursor(0, 1);  
    lcd.print("na lista");  
    delay(2500);  
    myFile = SD.open(diretoriolista, FILE_WRITE);  
    myFile.println(RA);  
    myFile.close();  
    TimerStart(&LeTeclado3);  
    break;  
}  
}  
}  
}  
TimerStart(&LeTeclado2);  
}  
}  
break;  
default:  
    //  
    break;
```

}

}