



Nome do aluno: _____

Matrícula: _____

Data: _____

02/11/2022

Disciplina: DEC7560 Professor: Fábio Rodrigues de la Rocha

Turma(s): 08655

1 Projeto: Agenda telefônica

1.1 Introdução

O objetivo deste trabalho é desenvolver uma agenda telefônica embarcada utilizando o ESP8266 que deve aceitar solicitações de operações (inserção, consulta, remoção, etc.) via porta serial (utilizando o terminal cutecom ou equivalente). A agenda embarcada armazena os registros num chip de memória serial AT24C512¹.

A agenda telefônica utiliza registros compostos por apenas 3 campos: Nome (string de 20 caracteres), telefone (string de 14 caracteres), endereço (string de 30 caracteres). Cada registro possui um tamanho fixo de 64 bytes.

O software embarcado deve apresentar uma interface para o usuário (que usa o sistema via terminal), tal como abaixo:

MENU

- [1] - Lista todos os registros
- [2] - pesquisa registro por nome
- [3] - pesquisa registro por telefone
- [4] - remove registro baseado no telefone
- [5] - mostra a quantidade de registros atualmente armazenados
- [6] - Inicializa o banco de dados (todas as informações armazenadas serão perdidas)

Depois de ler e executar a operação solicitada o software apresenta novamente o menu para o usuário e aguarda uma nova seleção de opção.

A funcionalidade das opções é evidente. Opção 1 varre todos os registros armazenados e mostra para o usuário, opção 2 solicita que seja entrado um nome e em posse do nome é feita a uma pesquisa nos dados. Se for encontrado um registro com os dados iguais ele é apresentado para o usuário ou caso não exista uma mensagem de erro é apresentada. A opção 3 faz o mesmo mas para o telefone, a opção 4 solicita um numero de telefone e em posse deste varre o banco de dados para encontrar o registro e o remove. A opção 5 mostra quantos são os dados armazenados atualmente na base de dados. Finalmente a opção 6 inicializa o banco de dados (precisa ser chamada na primeira vez para inicializar a memória).

1.2 Armazenamento dos dados

Para facilitar o armazenamento e recuperação dos dados é possível criar um banco de dados tal como apresentado na figura 1.2. Como os registros possuem tamanho fixo de 64 bytes temos uma quantidade máxima de registros conhecida. Se utilizarmos um cabeçalho logo no inicio da memória podemos manter a contabilização de quantos registros temos armazenados. Vamos a um exemplo:

Qtd de registros	Qtd max	
	4 bytes	
REG1	64 bytes	
REG2	64 bytes	
REG3	64 bytes	
⋮	⋮	
REG'n'	64 bytes	

Figure 1: Arquitetura simples para representar a base de dados

Digamos que inicialmente o cabeçalho mostre que existem 0 registros. Ao inserir um registro novo, ele será inserido na posição REG1 que será iniciada no byte 4 (o cabeçalho vai de 0 até 3). A posição de cada registro

¹<https://pdf1.alldatasheet.com/datasheet-pdf/view/56068/ATMEL/AT24C512.html>

pode ser calculada facilmente e como o registro ocupa 64 bytes podemos ter uma função `le_registro` que lê os dados do registro e `salva_registro` que grava um registro baseado no seu número. Para realizar uma operação de remoção, devemos encontrar o alvo a ser removido e sobrescrever este com o último registro do banco de dados, logo depois, atualiza-se o cabeçalho. A figura 1.2 exemplifica a operação de um registro (registro da Ana). Chamamos isso de apagamento lógico do registro. A operação de inicialização do banco de dados deve preencher o cabeçalho com valores iniciais (zero registros armazenados). No campo máximo de registros, deve-se contabilizar quantos registros podem ser inseridos na memória de 64KB utilizada, descontando o cabeçalho. Se for de interesse o cabeçalho pode ser construído tal forma a ocupar o mesmo tamanho de um registro de dados.

```
typedef struct {
    char nome[20];
    char telefone[14];
    char endereco[30];
}tipo_registro;

void le_registro (uint16_t numero_do_registro, tipo_registro &R);

void salva_registro(uint16_t numero_do_registro, tipo_registro R);
```

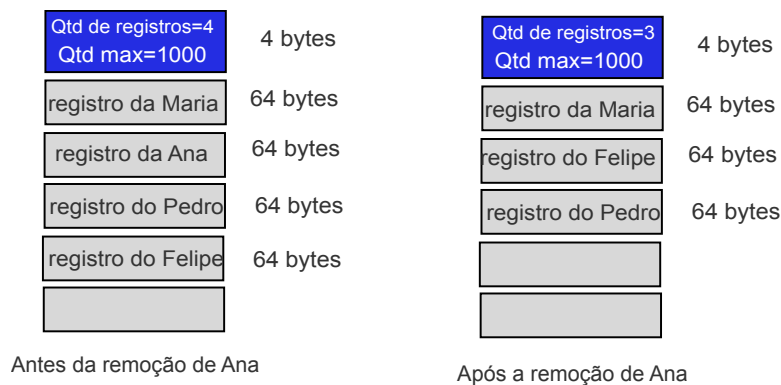


Figure 2: Exemplo de remoção de Ana da base de dados

1.3 Dicas:

Utiliza os exemplos de código para ler e escrever na porta serial. Utilize os exemplos de código que grava e lê dados da memória, crie operações que trabalham sobre registros (tal como exemplificado) e sobre o cabeçalho (`le_cabecalho`, `salva_cabecalho`)

1.4 Participantes:

O trabalho pode ser realizado individualmente ou em equipes de até 3 estudantes.

1.5 Data limite e apresentação:

A data limite para apresentação é dia 16/11. A apresentação será agendada com o professor e será realizada pessoalmente pela equipe no lab ou na sala do professor. Os códigos devem ser enviados pelo moodle num arquivo .zip