Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação I
AD2 2^o semestre de 2013.
Professores: Mario Benevides e Paulo Roma

---

1 Questão 1 (5 pontos)

Um usuário gostaria de fazer uma aplicação com menus do tipo popup para abrir arquivos existentes que contenham informações sobre funcionários de uma empresa. Cada arquivo contém informações sobre funcionários de um departamento da empresa. Para cada funcionário temos os seguintes dados: nome, cpf, salário e idade. A aplicação deve ser capaz de procurar um dado funcionário e atualizar seus dados. No menu principal, ele gostaria de ter as opções de $abrir - arquivo$, $salvar - arquivo$, e $fechar - aplicacao$. Num outro item de menu de $achar - funcionario$, $alterar - nome - funcionario$ e $alterar - salario$ e $alterar - idade$. As informações a serem procuradas ou substituídas devem ser perguntadas ao usuário em tempo de execução.

- Faça o projeto desta aplicação e implemente esta em Lazarus.

2 Questão 2 (5 pontos)

Na aplicação da questão anterior, refaça o projeto de maneira a usar um banco de dados da sua escolha para armazenar as informações contidas nos arquivos.

1. Faça o projeto desta aplicação e implemente esta em Lazarus.
2. **O aluno deverá enviar o código fonte que execute no Lazarus para ambiente Linux**.

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação I
AD2 - 2^o semestre de 2013.

GABARITO

---

1. Solução

A questão 1 pede para desenvolver um programa que armazene informações sobre os funcionários de uma empresa. A Figura 1 é composta por dois menus que possuem todos as funções disponíveis nesse programa. O Menu Arquivo possui as funções:

- Abrir Arquivo - Essa opção utiliza o componente TOpenDialog para selecionar o arquivo que armazena as informações dos funcionários.
- Salvar Arquivo - Utiliza o componente TSaveDialog para salvar as informações da memória do computador, para um arquivo.
- Fechar Aplicação - Utiliza o comando Application.Termination para encerrar o programa.

O Menu Funcionário é composto por:

- Achar Funcionário - Essa opção abre o formulário apresentado na Figura 2, que possui o componente TRadioGroup para o usuário escolher o campo que deseja pesquisar (CPF ou Nome). Ao clicar no botão "Buscar"o funcionário será buscado e prencherá os componentes TEdits não editáveis do formulário principal. Caso não encontre o funcionário pesquisado, os campos ficarão vazios.
- Alterar Nome - O formulário Altera Nome, Figura 3, serve para alterar o nome do funcionário selecionado no formulário principal. Caso não exista funcionário, será retornada uma mensagem informando que nenhum funcionário está selecionado.
- Alterar Salário - Esse menu, apresentado na Figura 4, permite a alteração do salário do funcionário corrente.
- Alterar Idade - Essa opção abre o formulário da Figura 5, que permite a edição do salário do funcionário corrente.

Figura 1: Formulário Principal.

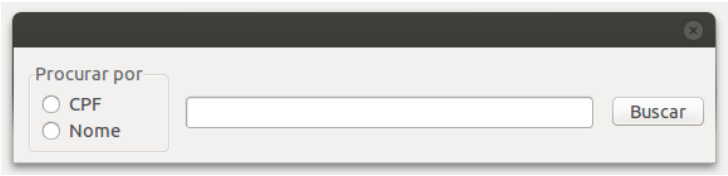

Figura 2: Formulário de Busca.

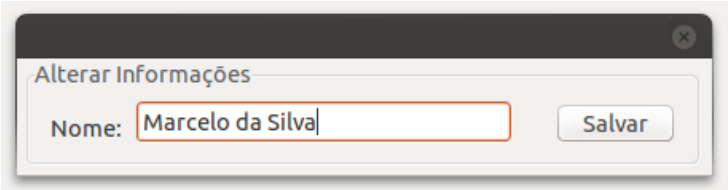

Figura 3: Formulário Altera Nome.

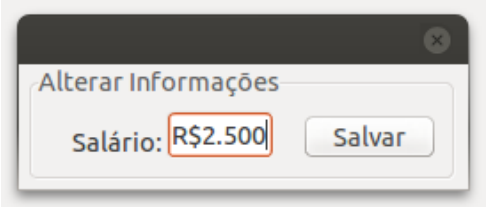

Figura 4: Formulário Altera Salário.

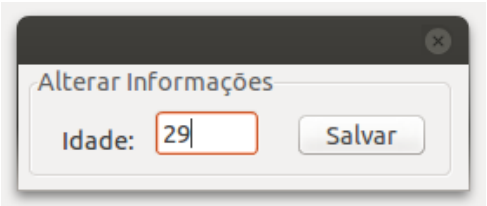

Figura 5: Formulário Altera Idade.

Armazenamento

0.1 Arquivo

Os dados deverão ser armazenados em um ou mais arquivos. Uma solução é utilizar um único arquivo de texto para armazenar todos os registros. Para isso, cada linha representa um único registro e os campos desse registro podem ser separados por um simbolo especial, por exemplo, nessa solução, utiliza-se o caracter "|"para separar os campos. Portanto, os campos de cada registro podem facilmente ser obtidos com o uso da função GetToken.

A função GetToken, apresentada abaixo, recebe três parametros, o primeiro é a string, o segundo é o caracter que separa os valores e o terceiro é um inteiro que indica qual o campo desejado, onde 0 retorna o primeiro campo da string (cpf), 1 retorna o Nome, 2 retorna o Salário e 3 retorna a Idade.

Implementação da função GetToken.

```
1  function TFormPrinc.GetToken (a:String; Sep: Char; dev: integer):String;
2    var
3      Token : String;
4      TEnd : Byte;
5      contador : integer;
6    begin
7      contador:=0;
8      while(contador<=dev) do
9      begin
10       TEnd := Length(a);em meu caso
11       Result := '';
12       TEnd := Pos(Sep, a);
13       if TEnd <> 0 then
14         begin
15           Token := Copy( a, 1, TEnd-1);
16           Delete( a, 1, TEnd);
17         end
18         else
19         begin
20           Token := a;
21           a := '';
22         end;
23         Result := Token;
24       contador:= contador +1;
25     end;
26   end;
```

Uma estratégia para trabalhar com arquivo e não ficar o tempo todo acessando o disco é carregar o arquivo texto para uma variável do tipo *TStringList*.

Exemplo de como declarar, carregar e salvar arquivos para TStringList:

```
var
  List : TStringList;
begin
  List := TStringList.Create;
  List.LoadFromFile('dados.txt');

  List.SaveToFile('dados.txt');
end
```

Outros três comandos importantes para o desenvolvimento são:

- Apagar uma linha (registro), por exemplo, apagar a primeira linha:
  List.Delete(0);
- Inserir um registro na linha 5:
  List.Insert(4, novoRegistro);
- Obter todo o conteúdo da linha 4:
  registro := List.Strings[3];

0.2 Banco de Dados

A segunda questão da AD implica em utilizar a mesma interface já apresentada nas Figuras 2, 3, 4 e 5, porém utiliza um SGBD para armazenar os dados. No caso dessa solução, foi utilizado o MySql conforme as aulas 14 e 15 dessa disciplina.

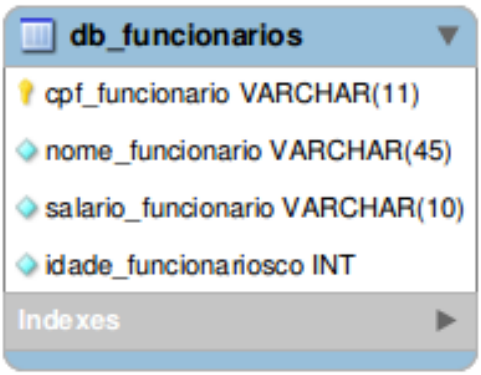

Figura 6: Tabela de Funcionários do Banco de Dados.

A tabela da Figura 6 foi criada com o seguinte comando:

```
CREATE  TABLE IF NOT EXISTS `mydb`.`db_funcionarios` (
  `cpf_funcionario` VARCHAR(11) NOT NULL ,
  `nome_funcionario` VARCHAR(45) NOT NULL ,
  `salario_funcionario` VARCHAR(10) NOT NULL ,
  `idade_funcionariosco` INT NOT NULL ,
  PRIMARY KEY (`cpf_funcionario`) )
```

Demais comandos utilizados durante o projeto:

- Aloca ou inicia um objeto MYSQL adequado à chamada da função mysql_real_connect().

  Comando: mysql_init();

- Estabelece uma conexão com um banco de dados MySQL.

  Comando: mysql_real_connect();

- Consultas.

  *Comando: select * from db_funcionarios where cpf_funcionario = "25192101212";*

- Atualizando dados.

  Comando: update db_funcionarios set nome_funcionario = 'Marcelo Soares' where cpf_funcionario = "25192101212";

- Fecha uma conexão.

  Comando: mysql_close();