

Manual e Informações do Software

Lactec CRUD v1.0

Desenvolvedor: Eliel Marcos Rocha Romancini

e-mail: elielmarcos@hotmail.com GitHub: github.com/elielmarcos

Lactec CRUD (*Create, Read, Update and Delete*) é uma aplicação voltada ao cadastro de clientes, requisitado pelo Instituto LACTEC, setor de Desenvolvimento de Software & UX, em Curitiba - PR, com a motivação e proposta de avaliação para vaga de estágio.

Requisitos

Os requisitos funcionais do software são apresentados:

1. *Escrever um programa em uma linguagem a escolha que seja capaz de cadastrar, buscar e excluir Clientes;*
2. *O programa deve possuir uma interface gráfica amigável;*
3. *Cada Cliente possui ID, Nome, Idade e Telefone;*
4. *Quando optar por buscar ou excluir um cliente, deve ser solicitado o ID ou Nome para que a ação seja realizada;*
5. *Se não existir o Cliente informado para busca ou exclusão, deve ser escrita a mensagem "Cliente não encontrado" e retornar para o "menu" principal;*
6. *Ao excluir um Cliente, deve-se dar um feedback ao usuário com a mensagem: "Cliente excluído com sucesso!";*
7. *Ao buscar um Cliente, deve-se printar todas as suas informações:*
Ex: "ID: 5, Nome: José Santos, Idade: 21, Telefone: 41-3355-8800"
8. *O programa deve possuir dois tipos de listagem:*
 - 8.1. *"Listar clientes". Esta opção lista todos os clientes. Apenas id e nome devem ser printados:*
Ex: "ID: 1, Nome: João Silva"
 - 8.2. *"Listar clientes com ordenação". Esta opção lista os clientes ordenando-os de acordo com sua idade:*
Ex: "ID: 1, Nome: João Silva, Idade: 18"
"ID: 20, Nome: Maria, Idade: 20"
"ID: 5, Nome: José Santos, Idade: 21"
9. *O programa deve ter um botão de saída.*

Além dos requisitos ainda foram solicitadas as metas:

1. *Separar em 3 camadas e salvar dados em DB;*
2. *Documentar classes e métodos públicos;*
3. *Usar binding na UI;*
4. *Hospedar código no GitHub.*

Ferramentas utilizadas

Com base nos tópicos requeridos, foi construído uma aplicação utilizando como ferramenta de desenvolvimento o *RAD Studio 10.3*, fabricado pela companhia *Embarcadero Technologies*. Esta ferramenta permite programar em linguagem de programação C++ e *Delphi/Pascal*, sendo de fácil implementação e desenvolvimento da interface gráfica / interface com usuário.

Para maiores informações sobre o *RAD Studio 10.3*, acesse: <https://www.embarcadero.com/br/products/rad-studio/whats-new-in-10-3-rio>

Como meio de guardar e ter acesso os dados cadastrados no Lactec CRUD, foi utilizado um *Database* (base de dados) que trabalha com linguagem SQL, o *SQLite3*, pois é leve, pequeno, de fácil utilização e portátil para dispositivos móveis por ser multiplataforma. Além do mais, é uma ferramenta livre/grátis, *OpenSource*, excelente para pequenas aplicações de necessitam de um banco de dados local.

Para maiores informações sobre o *SQLite*, acesse: <https://www.sqlite.org/index.html>

Modelo do Banco de Dados

Modelo Conceitual:



Figura 01 - Modelo Conceitual da Tabela Cliente

O modelo da figura 01, representa o modelo conceitual adotado para construção da principal e única tabela do banco de dados, chama *Cliente*. Esta possui os seguintes atributos:

ID: Identificador; Armazena o Id do Cliente; Tipo Integer; Auto Incrementável.

Nome: Armazena o Nome do Cliente; Tipo varchar(100).

Idade: Armazena a Idade do Cliente; Tipo Integer.

Telefone: Armazena o Número de Telefone do Cliente; Tipo varchar(15).

Modelo Lógico:


Cliente	
	ID: Integer
	Telefone: varchar(15)
	Idade: integer
	Nome: varchar(100)

Figura 02 - Modelo Lógico da Tabela Cliente

O modelo da figura 02, representa o modelo lógico adotado da tabela *Cliente* do banco de dados. Tem como chave primária (*PRIMARY KEY*) o identificador ID.

Para gerar a construção da tabela em linguagem SQL, foi utilizado o código:

```
CREATE TABLE IF NOT EXISTS Cliente(  
    ID Integer PRIMARY KEY AUTOINCREMENT,  
    Telefone varchar(15),  
    Idade integer,  
    Nome varchar(100)  
);
```

Implementação

A construção do software se deu em linguagem *Delphi/Pascal*, sendo desenvolvido e compilado para S.O. Windows 32 bits, gerando um arquivo executável de nome *Lactec_CRUD.exe*. O arquivo de projeto gerado pela ferramenta *RAD Studio* é nomeado *.dproj, interface *.dfm e código *.pas.

Classes utilizadas:

```
TForm // Responsável pela criação do Formulário Principal  
  
TLabel // Responsável pela criação do objeto Label (Texto estático no  
formulário)  
  
TEdit // Responsável pela criação do objeto Edit (Caixa de texto  
editável no formulário)  
  
TSpeedButton // Responsável pela criação do objeto SpeedButton (Botão  
com imagem no formulário)  
  
TRadioGroup // Responsável pela criação do objeto RadioGroup (Botões de  
opções selecionáveis no formulário)
```

TComboBox // Responsável pela criação do objeto ComboBox (Caixa de seleção no formulário)

TPanel // Responsável pela criação do objeto Panel (Painel no formulário)

TStatusBar // Responsável pela criação do objeto StatusBar (Caixa de status no formulário)

TImageCollection // Responsável pela criação do objeto ImageCollection (Armazena a coleção das imagens)

TFDConnection // Responsável pela criação do objeto FireDac FDCConnection (Realiza a conexão com Banco de Dados)

TFDGUIxWaitCursor // Responsável pela criação do objeto FireDac FGUIxWaitCursor (Responsável por controlar o cursor de espera)

TFDPhysSQLiteDriverLink // Responsável pela criação do objeto FireDac FPhysSQLiteDriverLink (Responsável pelo Driver de conexão do SQLite com a aplicação)

TFDQuery // Responsável pela criação do objeto FireDac FDQuery (Responsável pela tabela do Banco de Dados e comandos SQL)

TDataSource // Responsável pela criação do objeto DataSource (Realiza a conexão da tabela com outros componentes);

TDBEdit // Responsável pela criação do objeto DBEdit (Caixa de texto editável, conectada a tabela do Banco de Dados no formulário)

TDBGrid // Responsável pela criação do objeto DBGrid (Tabela/Lista editável, conectada a tabela do Banco de Dados no formulário)

Principais Métodos utilizados:

```
procedure FormCreate(Sender: TObject);  
// Realiza ações quando o formulário principal é criado, neste método é  
// aberta a conexão com o banco de dados.  
  
procedure FormClose(Sender: TObject; var Action: TCloseAction);  
// Realiza ações quando o formulário principal é fechado / finalizado.  
  
procedure SpeedButtonListarClick(Sender: TObject);  
// Realiza ações quando SpeedButtonListar é clicado.
```

```
procedure SpeedButtonCadastrarClick(Sender: TObject);
// Realiza ações quando SpeedButtonCadastrar é clicado.

procedure SpeedButtonNovoClick(Sender: TObject);
// Realiza ações quando SpeedButtonNovo é clicado.

procedure SpeedButtonSalvarClick(Sender: TObject);
// Realiza ações quando SpeedButtonSalvar é clicado.

procedure SpeedButtonCadastrarVoltarClick(Sender: TObject);
// Realiza ações quando SpeedButtonCadastrarVoltar é clicado.

procedure SpeedButtonBuscarClick(Sender: TObject);
// Realiza ações quando SpeedButtonBuscar é clicado.

procedure SpeedButtonBuscarExcluirVoltarClick(Sender: TObject);
// Realiza ações quando SpeedButtonBuscarExcluirVoltar é clicado.

procedure SpeedButtonSairClick(Sender: TObject);
// Realiza ações quando SpeedButtonSair é clicado, neste método é
chamado o método FormClose para finalizar a aplicação.

procedure SpeedButtonListarVoltarClick(Sender: TObject);
// Realiza ações quando SpeedButtonListarVoltar é clicado.

procedure SpeedButtonBuscarBuscarClick(Sender: TObject);
// Realiza ações quando SpeedButtonBuscarBuscar é clicado.

procedure SpeedButtonExcluirClick(Sender: TObject);
// Realiza ações quando SpeedButtonExcluir é clicado.

procedure SpeedButtonBuscarExcluirClick(Sender: TObject);
// Realiza ações quando SpeedButtonBuscarExcluir é clicado.

procedure SpeedButtonSobreClick(Sender: TObject);
// Realiza ações quando SpeedButtonSobre é clicado.

procedure SpeedButtonSobreVoltarClick(Sender: TObject);
// Realiza ações quando SpeedButtonSobreVoltar é clicado.

procedure SpeedButtonGithubClick(Sender: TObject);
// Realiza ações quando SpeedButtonGithub é clicado.

procedure SpeedButtonIconesClick(Sender: TObject);
```

```
// Realiza ações quando SpeedButtonIcones é clicado.
procedure SpeedButtonEmailClick(Sender: TObject);
// Realiza ações quando SpeedButtonEmail é clicado.

procedure RadioGroupListarClick(Sender: TObject);
// Realiza ações quando RadioGroupListar é clicado.

procedure EditBuscarKeyPress(Sender: TObject; var Key: Char);
// Verifica se a tecla ENTER foi pressionada.

procedure ShowHint(Sender: TObject);
// Associa os Hint's dos objetos a serem apresentados no StatusBar.

procedure LoadIcons();
// Carrega as imagens nos SpeedButtons.

procedure IconPng(Btn: TSpeedButton; Ind: Integer; W: Integer; H:
Integer);
// Carrega uma imagem do ImageCollection ao SpeedButton que o invocou.
```

Utilização

Ao executar o arquivo *Lactec_CRUD.exe* (no sistema operacional Windows 32 ou 64 bits), aplicação verifica a existência do arquivo de banco de dados *.db. Caso o mesmo não exista, ele será criado e algumas mensagens de avisos serão apresentadas. Após, o programa ficará aberto no *Menu* principal:

As opções do *Menu* são:



Cadastrar: Realiza o cadastro de Clientes na base de dados;

Buscar: Realiza a busca de Clientes na base de dados;

Listar: Lista todos os Clientes armazenados na base de dados;

Excluir: Realiza a busca e remoção de Clientes da base de dados;

Sobre: Informações sobre o app e configuração de ícones;

Sair: Finaliza a aplicação.

Figura 03 - Menu Principal

Cadastrar Clientes:

Através do botão Cadastrar é possível adicionar novos Clientes, sendo direcionado ao cadastro, clique no botão Novo, informe o nome, idade e telefone, após clique no botão Salvar ou Voltar para cancelar e retornar ao *Menu* principal.



Figura 04 - Cadastrar

Buscar Clientes:

Através do botão Buscar é possível encontrar os Clientes cadastrados, sendo direcionado a busca, escolha o método de busca (por ID ou Nome do Cliente), informe o dado do cliente que deseje encontrar (campo *Digite*) e clique no botão Busca. Os Clientes encontrados serão listados na tabela. Se nenhum Cliente for localizado será retornado ao *Menu* principal ou clicando no botão Voltar.



Figura 05 - Buscar

Listar Clientes:

Através do botão Listar é possível listar todos os Clientes cadastrados, sendo direcionado ao listar, escolha a order de listagem (por ID ou Idade dos Clientes). Serão listados todos os ID's e Nomes dos Clientes de forma crescente se a opção *Ordenar por ID* for escolhida ou serão listados todos os ID's, Nomes, Idades e Telefones dos Clientes de forma crescente se a opção *Ordenar por Idade* for escolhida. Para voltar ao *Menu* principal clique no botão Voltar.



Figura 06 - Listar

Excluir Clientes:

Através do botão Excluir é possível encontrar e remover Clientes cadastrados, sendo direcionado a excluir, escolha o método de busca (por ID ou Nome do Cliente), informe o dado do cliente que deseja encontrar (campo *Digite*) e clique no botão Busca. Os Clientes encontrados serão listados na tabela. Selecione, na tabela, o Cliente que deseja remover e clique no botão Excluir. Se nenhum Cliente for localizado será retornado ao *Menu* principal ou clicando no botão Voltar.

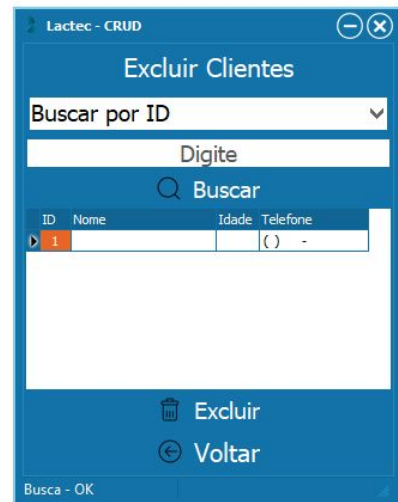


Figura 07 - Excluir

Sobre:

Através do botão Sobre serão apresentadas as informações do app, entrar em contato com o desenvolvedor e alterar o tema (ícones) da aplicação. Para retornar ao *Menu* principal clique no botão Voltar.



Figura 08 - Sobre