**Documentação do Projeto: Sistema de Cadastro de Agências e Contas Bancárias**

**Sumário**

1. [Introdução](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#introdu%C3%A7%C3%A3o)
2. [Ferramentas Utilizadas](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#ferramentas-utilizadas)
3. [Arquitetura do Projeto](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#arquitetura-do-projeto)
4. [Configuração do Ambiente](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#configura%C3%A7%C3%A3o-do-ambiente)
5. [Descrição das Funcionalidades](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#descri%C3%A7%C3%A3o-das-funcionalidades)
6. [Detalhamento dos Módulos](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#detalhamento-dos-m%C3%B3dulos)
7. [Banco de Dados](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#banco-de-dados)
8. [Considerações Finais](https://chatgpt.com/c/d526fe7c-d1b3-4a81-82f6-02db3bf4bdff#considera%C3%A7%C3%B5es-finais)

**Introdução**

Este documento apresenta o projeto de um Sistema de Cadastro de Agências e Contas Bancárias, desenvolvido como parte de um processo seletivo. O sistema foi construído usando Delphi 12 e SQL Server, utilizando a arquitetura Model-View-Controller (MVC) para garantir modularidade e manutenibilidade.

**Ferramentas Utilizadas**

* **Delphi 12**: Ambiente de desenvolvimento integrado (IDE) para a criação de aplicações.
* **SQL Server 2014 ou superior**: Sistema de gerenciamento de banco de dados relacional.
* **FireDAC**: Biblioteca de acesso a dados usada para conexão com o SQL Server.
* **FastReport**: Ferramenta para geração de relatórios.

**Arquitetura do Projeto**

O projeto segue o padrão de arquitetura MVC, dividido em três camadas principais:

* **Model**: Responsável pela lógica de negócios e comunicação com o banco de dados.
* **View**: Interface do usuário, onde os dados são apresentados e as interações são capturadas.
* **Controller**: Faz a mediação entre a View e o Model, processando as entradas do usuário e atualizando a interface.

**Estrutura de Pastas**

sql

MVCProject

│

├── Controller

│ ├── ControllerAgencia.pas

│ └── ControllerConta.pas

│

├── DataModule

│ └── DataModule.pas

│

├── Model

│ ├── ModelAgencia.pas

│ └── ModelConta.pas

│

├── View

│ ├── MainForm.pas

│ ├── MainForm.dfm

│ ├── CadastroPaiForm.pas

│ ├── CadastroPaiForm.dfm

│ ├── CadastroFilhoForm.pas

│ ├── CadastroFilhoForm.dfm

│ ├── ConsultaForm.pas

│ ├── ConsultaForm.dfm

│ └── RelatorioForm.pas

│ └── RelatorioForm.dfm

│

├── MVCProject.dpr

└── MVCProject.dproj

**Configuração do Ambiente**

**Requisitos**

* Delphi 10 Seattle ou superior
* SQL Server 2014 ou superior
* Componente FireDAC instalado
* Componente FastReport instalado

**Configuração do SQL Server**

1. **Habilitar TCP/IP**:
   * Abra o SQL Server Configuration Manager.
   * Expanda SQL Server Network Configuration e selecione Protocols for SQLEXPRESS.
   * Habilite o TCP/IP.
   * Configure a porta TCP como 1433 na aba IP Addresses.
2. **Configurar Firewall**:
   * Permita a porta 1433 no firewall do Windows.
3. **Iniciar Serviços**:
   * Certifique-se de que os serviços SQL Server (SQLEXPRESS) e SQL Server Browser estão em execução.

**Configuração do Projeto no Delphi**

1. **Adicione o FireDAC ao projeto**:
   * Vá para o menu Project -> Options -> Packages e certifique-se de que o FireDAC está habilitado.
2. **Configure a Conexão**:
   * No DataModule.pas, configure a conexão com o SQL Server conforme abaixo:

pascal

procedure TdmMain.DataModuleCreate(Sender: TObject);

begin

FDConnection.Params.Clear;

FDConnection.DriverName := 'MSSQL';

FDConnection.Params.Add('Server=127.0.0.1,1433'); // IP e porta do servidor SQL Server

FDConnection.Params.Add('Database=NomeDoBancoDeDados'); // Nome do banco de dados

FDConnection.Params.Add('OSAuthent=Yes');

FDConnection.Params.Add('ApplicationName=MVCProject');

FDConnection.LoginPrompt := False;

try

FDConnection.Connected := True;

except

on E: Exception do

ShowMessage('Erro ao conectar ao banco de dados: ' + E.Message);

end;

end;

**Descrição das Funcionalidades**

**Tela Principal**

* **Botões de Navegação**:
  + **Cadastro Pai**: Navega para a tela de cadastro de agências.
  + **Cadastro Filho**: Navega para a tela de cadastro de contas bancárias.
  + **Relatório**: Navega para a tela de geração de relatórios.

**Tela de Cadastro de Agências (CadastroPaiForm)**

* **Campos de Cadastro**:
  + Código da Agência
  + Número da Agência
  + Nome da Agência
  + Nome da Cidade
  + Estado (ComboBox)
  + Data de Abertura
  + Gerente Responsável
* **Funcionalidades**:
  + Incluir, alterar, excluir e consultar registros de agências.

**Tela de Cadastro de Contas (CadastroFilhoForm)**

* **Campos de Cadastro**:
  + Código da Conta
  + Número da Conta
  + Dígito da Conta
  + Código da Agência
  + Número da Agência
  + Titular da Conta
  + Gerente da Conta
* **Funcionalidades**:
  + Incluir, alterar, excluir e consultar registros de contas bancárias.

**Tela de Consulta (ConsultaForm)**

* **Busca**:
  + Permite a busca de registros com base em parâmetros específicos.
* **Grid de Resultados**:
  + Exibe os resultados da busca em um grid.
  + Permite a seleção de um registro para edição.

**Tela de Relatório (RelatorioForm)**

* **Geração de Relatórios**:
  + Permite a geração de relatórios completos ou filtrados por agência ou conta.
  + Utiliza o componente FastReport para exibição dos relatórios.

**Detalhamento dos Módulos**

**Controller**

* **ControllerAgencia.pas**:
  + Contém a lógica de negócios para manipulação dos dados das agências.
* **ControllerConta.pas**:
  + Contém a lógica de negócios para manipulação dos dados das contas bancárias.

**DataModule**

* **DataModule.pas**:
  + Configura a conexão com o banco de dados usando FireDAC.
  + Contém os componentes de conexão e consulta ao banco de dados.

**Model**

* **ModelAgencia.pas**:
  + Define a estrutura dos dados das agências.
* **ModelConta.pas**:
  + Define a estrutura dos dados das contas bancárias.

**View**

* **MainForm.pas e MainForm.dfm**:
  + Tela principal com botões de navegação para outras telas.
* **CadastroPaiForm.pas e CadastroPaiForm.dfm**:
  + Tela de cadastro de agências.
* **CadastroFilhoForm.pas e CadastroFilhoForm.dfm**:
  + Tela de cadastro de contas bancárias.
* **ConsultaForm.pas e ConsultaForm.dfm**:
  + Tela de consulta com grid de resultados.
* **RelatorioForm.pas e RelatorioForm.dfm**:
  + Tela de geração de relatórios.

**Banco de Dados**

**Estrutura das Tabelas**

sql

CREATE TABLE Agencia (

CodigoAgencia VARCHAR(10) PRIMARY KEY,

NumeroAgencia VARCHAR(10),

NomeAgencia VARCHAR(100),

NomeCidade VARCHAR(100),

Estado CHAR(2),

DataAbertura DATE,

GerenteResponsavel VARCHAR(100)

);

CREATE TABLE Conta (

CodigoConta VARCHAR(10) PRIMARY KEY,

NumeroConta VARCHAR(10),

DigitoConta CHAR(1),

CodigoAgencia VARCHAR(10),

NumeroAgencia VARCHAR(10),

TitularConta VARCHAR(100),

GerenteConta VARCHAR(100),

FOREIGN KEY (CodigoAgencia) REFERENCES Agencia(CodigoAgencia)

);

**Scripts de Criação**

Execute os scripts SQL acima no SQL Server Management Studio (SSMS) para criar as tabelas necessárias para o funcionamento do sistema.

**Considerações Finais**

Este projeto foi desenvolvido seguindo as melhores práticas de desenvolvimento, utilizando a arquitetura MVC para garantir a modularidade e facilidade de manutenção. O uso de componentes como FireDAC e FastReport garante a eficiência na conexão com o banco de dados e na geração de relatórios. Este sistema está pronto para ser utilizado em um ambiente de produção, oferecendo uma solução robusta para o gerenciamento de agências e contas bancárias.

Se houver qualquer dúvida ou necessidade de suporte adicional, por favor, não hesite em entrar em contato.

Este documento foi elaborado para fornecer uma visão geral clara e detalhada do projeto, facilitando a compreensão de sua estrutura e funcionalidades.