Cadastro de Clientes

**Link GitHub: https://github.com/wgcostta/DevsON/tree/master/Delphi/CloudPessoa**

**Histórico de Revisões**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 19/06/2020 | 1.0 | Criação do Documento | Wagner |
| 20/06/2020 | 1.1 | Conclusão do Projeto | Wagner |

**Sumário**

[1. Introdução 2](#_heading=h.30j0zll)

2[.](#_heading=h.2s8eyo1) Roteiro de atividade/ tarefas de desenvolvimento 2

# Introdução

A seguir detalhamos as informações para a execução do teste. Ressaltamos que a organização do código e as boas práticas também serão critérios de avaliação.

1. Utilizar a linguagem Delphi (Qualquer versão);
2. Criar uma tela de cadastro de clientes, com os seguintes campos:

Dados do Cadastro:

* + Nome
  + Identidade
  + CPF
  + Telefone
  + Email
  + Endereço
    - Cep
    - Logradouro
    - Numero
    - Complemento
    - Bairro
    - Cidade
    - Estado
    - Pais

1. Ao informar um Cep o sistema deve realizar a busca dos dados relacionados ao mesmo no seguinte endereço: <https://viacep.com.br/>;
2. A forma de consumo da API do via Cep, deverá ser utiliza JSON;
3. Ao termino do cadastro o usuário deverá enviar um e-mail contendo as informações cadastrais e anexar um arquivo no formato XML com o mesmo conteúdo;
4. Os registros devem ficar salvo em memória, não é necessário criar um banco de dados ou arquivo para o armazenamento dos dados;
5. Disponibilizar o código fonte do projeto no github;

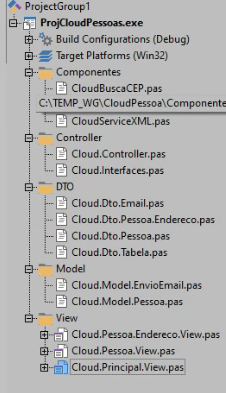
# Roteiro de atividade/ tarefas de desenvolvimento

Para este projeto, como forma de demonstrar o conhecimento que possuo foi utilizado no desenvolvimento conceitos de Padrões de Projeto, Rtti, POO, Interfaces, MVC, Generics<T>, Collections, Clean Code.

Estrutura do Projeto Desenvolvido:

Units:

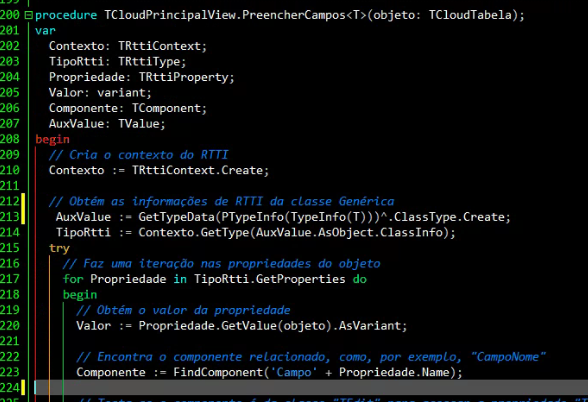




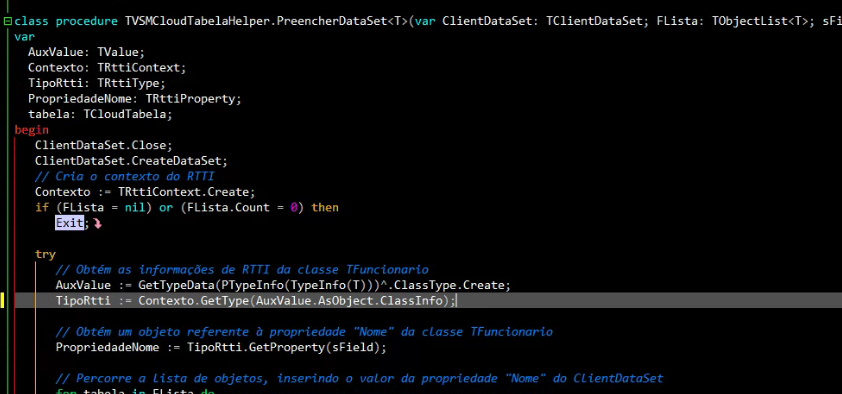
Tela Principal com preenchimento em RTTI através de um ClientDataSet e uma lista genérica, onde é preenchido as propriedades da tela:



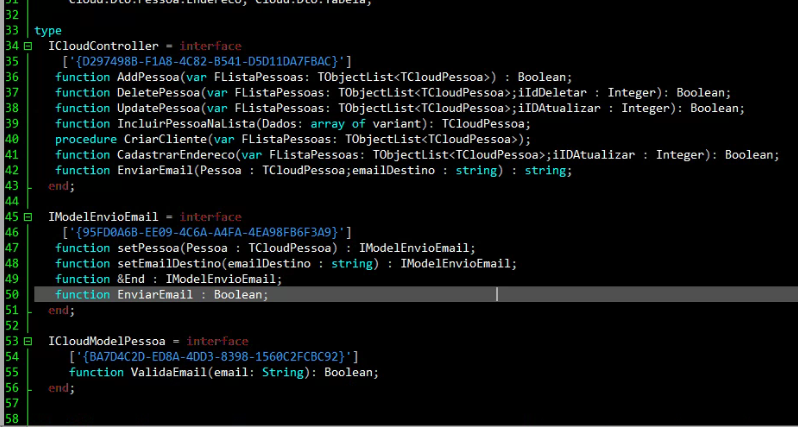
A função a seguir percorre os componentes da tela e associa aos menos os objetos encontrados no registro (Tabela) selecionada. No gerenciador foi usado apenas TEdit, como componente nativo, para que não fosse necessário realizar instalação de componentes para utilização:



Na procedure TVSMCloudTabelaHelper.PreencherDataSet é percorrido a lista atual dando AppendRecord no DataSet, sempre que é atualizado alguma lista.



Foi utilizado ainda uma Interface para o Controller, Model para utilização dos métodos dos objetos.



Para as Consultas ViaCep, envio de e-mail e criação do XML foram criados como Units de Componentes:

