

Todas as boas práticas de desenvolvimento serão consideradas na prova:

- Clean code;
- Padrões de projeto;
- P.O.O.;
- Herança, encapsulamento, polimorfismo;
- Projeto 2 ou 3 camadas (recomendado fazer 3 camadas);
- Tratamento adequado em criar e destruir objetos;
- Utilização de recursos mais nos do Delphi (Generics, Threads)

1. Orientações gerais

- Arquivo de orientação e a avaliação estão na pasta C:\TesteDelphi\
- Salvar o projeto teste em C:\TesteDelphi\Nome do Candidato
- DLL's necessárias para conexão com banco de dados estão pasta: c:\TesteDelphi\Dlls
- Dados para acesso banco (Local)
 - Tipo conexão: PG
 - DataBase: Teste
 - Server: localhost
 - Porta: 5432
 - Usuário: postgres
 - Senha: postgres
- O candidato pode definir qualquer arquitetura para aplicação, segue abaixo alguns exemplos:
 - DataSnap: c:\TesteDelphi\Arquitetura DataSnap
 - Horse: c:\TesteDelphi\Arquitetura Horse

2. Criar banco de dados com estrutura do arquivo abaixo (preferencialmente em PostgreSQL)

```
CREATE TABLE pessoa (  
    idpessoa bigserial NOT NULL,  
    flnatureza int2 NOT NULL,  
    dsdocumento varchar(20) NOT NULL,  
    nmprimeiro varchar(100) NOT NULL,  
    nmsegundo varchar(100) NOT NULL,  
    dtregistro date NULL,  
    CONSTRAINT pessoa_pk PRIMARY KEY (idpessoa)  
);  
  
CREATE TABLE endereco (  
    idendereco bigserial NOT NULL,  
    idpessoa int8 NOT NULL,  
    dscep varchar(15) NULL,  
    CONSTRAINT endereco_pk PRIMARY KEY (idendereco),  
    CONSTRAINT endereco_fk_pessoa FOREIGN KEY (idpessoa) REFERENCES pessoa(idpessoa) ON DELETE  
cascade  
);  
  
CREATE INDEX endereco_idpessoa ON endereco (idpessoa);  
  
CREATE TABLE endereco_integracao (  
    idendereco bigint NOT null,
```

```

dsuf varchar(50) NULL,
nmcidade varchar(100) NULL,
nmbairro varchar(50) NULL,
nmlogradouro varchar(100) NULL,
dscomplemento varchar(100) NULL,
CONSTRAINT enderecointegracao_pk PRIMARY KEY (idendereco),
CONSTRAINT enderecointegracao_fk_endereco FOREIGN KEY (idendereco) REFERENCES
endereco(idendereco) ON DELETE cascade
);

```

3. Definir arquitetura do sistema em três camadas

- Comunicação Rest com JSON entre aplicação Cliente / Servidor;
- Aplicar Clean Code;
- Orientação a objetos;
- Padrões de projeto;
- Garantir integridade entre registros (não ter pessoa sem endereço);
- Camada de persistência, utilizar Firedac.

2.1 Desenvolver um cadastro de pessoas

Objetivo é fazer cadastro simplificado com os dados da pessoa e o CEP (no item 2.2 a tabela `endereco_integracao` será atualizada com base no CEP informado)

- Tabelas: Pessoa e Endereco
- Métodos:
 - Insert
 - Update
 - Delete
 - Insert em lote (novo método): recebe uma lista de pessoas (considerando que essa lista poderá ter 50.000 registros. Adotar uma estratégia para que a inserção desses registros seja performática).

2.2 Desenvolver nova rotina utilizando Threads

- Objetivo é atualizar os endereços das pessoas cadastradas no item 2.1
- Para cada registro da tabela `endereco`, ler campo CEP e fazer a integração com a “API via cep” através da URL `viacep.com.br/ws/_numero_CEP/json/`
 - Utilizar campo CEP da tabela `endereco`
- Atualizar os campos da tabela `endereco_integracao` com os dados do JSON de retorno