Substitua os vermelhos pelos registros

Linguagem: **DELPHI**

Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cidade/UF Residência: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Onde viu a vaga: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mudança de cidade: () Sim / () Não (marque com X a opção)

CLT pretensão salarial: R$ \_\_\_\_\_\_\_\_\_ (se não oferecer esse tipo de contrato deixar em branco)

PJ pretensão salarial: R$ \_\_\_\_\_\_\_\_\_ (se não oferecer esse tipo de contrato deixar em branco)

Último salário e tipo de contrato: R$ \_\_\_\_\_\_\_\_\_ / () CLT / () PJ / () Freelance

Tempo de experiencia (Ano/Mês): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Empresas do ramo e tempo: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Empresa X[2 anos e 3 meses])

Disponibilidade de início: () Imediato / () Outros \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (descreva)

Conhecimentos gerais

|  |  |  |
| --- | --- | --- |
| () Git | () Metodologia scrum | () Project Manager |

Conhecimentos banco de dados MySQL

|  |  |  |
| --- | --- | --- |
| () Views | () Procedures | () Joins(left, right...) |

Conhecimentos Delphi

Versões: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| () Classe | () Herança | () Frame |
| () Firemonkey | () ACBRNfe | () ACBRSat |
| () ACBRSped | () ACBRNFse | () ACBRBoleto |
| () TEF | () Biometria | () OpenCV |
| () Zeos (banco de dados) |  |  |

Montar e enviar Código fonte e banco de dados (MySQL)

|  |  |  |
| --- | --- | --- |
| **Tabela operadora** | **Tabela cliente** | **Tabela cliente\_telefone** |
| Data\_Hora  Operadora | Data\_Hora  Nome  CPF  Data\_Nascimento  Email  Removido (cliente excluído ou não) | Data\_Hora  ClienteID  OperadoraID  Telefone  Removido (cliente excluído ou não) |

**Funções do banco de dados**

* Deve ter uma função chamada “F\_TemCPF” onde é mandado o cpf e ela retorna se já existe o cpf cadastrado
* Deve ter uma procedure chamada “P\_Telefone” onde envio o Código do cliente, código da operadora e o telefone e ela faz o cadastro do telefone no banco de dados
* Deve ter uma view “VW\_Clientes” onde me exibe todos os clientes ativos

**Classe Clientes**

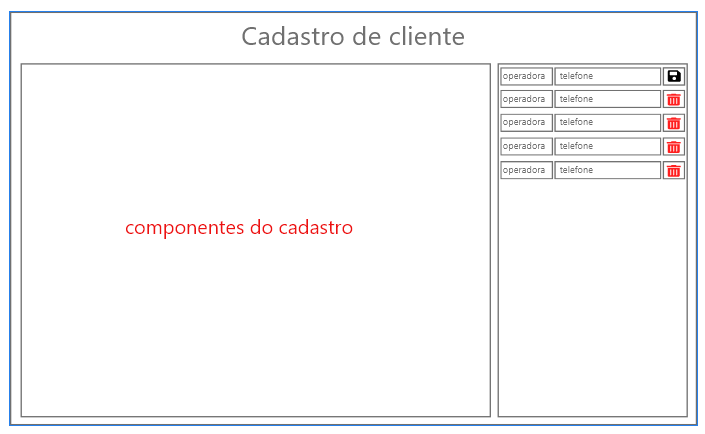
* deve ser montado em uma única unit
* cada classe deve ter um constructor
* cada classe deve ter um destructor
* deve conter uma classe "TCliente" onde vai armazenar todos os dados do cliente com a função salvar e a própria classe deve saber se vai salvar e editar o registro
* não pode cadastrar 2 clientes ativos com o mesmo CPF, por isso usar a Função do Banco “FTemCPF”
* na classe "TCliente" deve ter uma função "Pode\_Salvar" que retorna uma booleana para saber se todos os dados estão validados
* no Constructor devo passar o código do cliente que vou editar (passar o código zero caso for um novo cliente)
* toda vez que editar um cliente deve ser atualizado o campo data hora
* Dentro da Classe “TCliente” deve ter a classe “TTelefones” onde ficam guardado a lista de telefones do cliente, os telefones de vem ser cadastrados usando a função “P\_Telefone”
* deve ter uma classe “TClientes”, onde armazena a lista de clientes cadastrados.

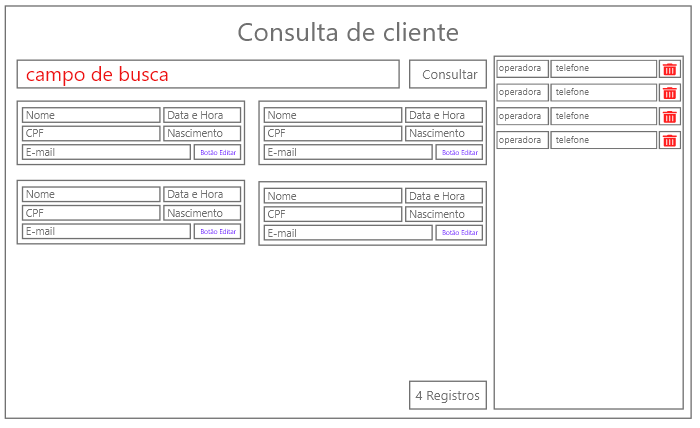
**Tela cadastro/edição**

* deve ter todos os campos do cliente
* deve ter um painel onde eu seleciono a operadora e faço o cadastro de novos telefones ou exclusão (telefones devem ser exibidos em cards construídos em run-time)

**Tela de consulta**

* deve ter 1 único edit onde consulto os clientes pelo Nome ou CPF
* todos clientes devem ser exibidos em cards construídos em run-time
* ao clicar em um card deve ser exibidos todos os telefones com a operadora (exibidos em cards construídos em run-time)
* deve usar a View “VW\_Clientes” para fazer a consulta no banco de dados





**Conteúdo de estudo**

Banco de dados (MySQL)

1. Procedures
2. Functions
3. Views
4. Joins
5. Ifs
6. Select
7. Update
8. Delete
9. Insert

Delphi

1. Procedures
2. Functions
3. Classes com property, constructor e destroy.
4. Thread com Parallel for, synchronize
5. Construção de objetos em Run-Time
6. Herança