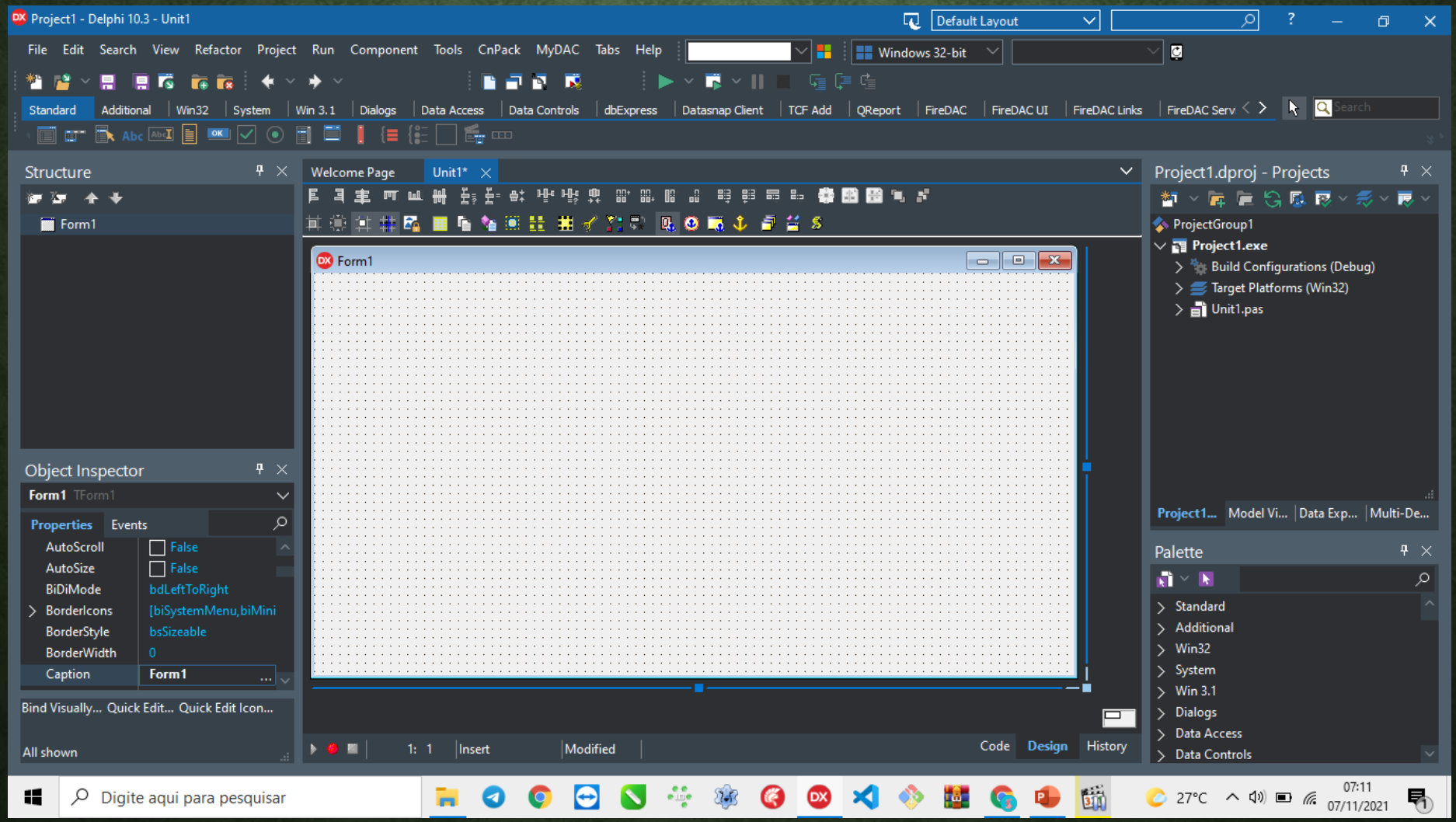
MVC – GETS E SETS

Programação Orientada e Objetos

INICIAREMOS UM NOVO PROJETO



DevTeam

ENTENDIMENTO ENTRE GET E SET

- Property Nome read GetNome -> Leitura Write SetNome -> Escrita
- Os Get usamos quando queremos da funcionalidade a regras de entrada no Edit.
  - Não é obrigatório usar sempre....Isso depende do projeto a ser desenvolvido.
- Read – Leitura - Get será sempre para obter a leitura do método.
- Write – Escrita - Set será sempre para obter a Escrita do método.
- Get : Retorna sempre um function que retorna um valor.
- Set : Retorne sempre um procedure sem que haja retorno de nada.

DevTeam

CRIAREMOS UM NOVA CLASSE

```
Tpessoa = class  //classe Pessoa
  private

  protected

  public

  property nome:string;
  property endereco:string;
  property salario:Currency;
end;
```

Ctrl + Shift + C

Selecione as 3 property antes do comando.

Criado as property na sessão public

DevTeam

OBSERVE A SEGUINTE QUESTÃO!

```
{ Tpessoa }

procedure Tpessoa.Setendereco(const Value: string);
begin
  Fendereco := Value;
end;

procedure Tpessoa.Setnome(const Value: string);
begin
  Fnome := Value;
end;

procedure Tpessoa.Setsalario(const Value: Currency);
begin
  Fsalario := Value;
end;
```

Criou a penas os Sets

DevTeam

OBSERVE TAMBÉM QUE ELE CRIOU PROCEDURES E FIELDS.

```
16
   Tpessoa = class  //classe Pessoa
     private

     Fsalario: Currency;
     Fnome: string;
     Fendereco: string;

     procedure Setendereco(const Value: string);
     procedure Setnome(const Value: string);
     procedure Setsalario(const Value: Currency);

     protected

     public
     property nome:string read Fnome write Setnome;
     property endereco:string read Fendereco write Setendereco;
     property salario:Currency read Fsalario write Setsalario;
   end;
```

Fields(campos)

Procedures com Sets

Nós sabemos que as procedures não retorna nenhum valor (sets) leituras, ou seja apenas damos funcionalidade(métodos) para a mesmo.

DevTeam

VAMOS CRIAR OS GETS, ALTERE AS PROPERTY COMO ESTÁ NA IMAGEM

```
  public
  property nome:string read Getnome write Setnome;
  property endereco:string read Getendereco write Setendereco;
  property salario:Currency read Getsalario write Setsalario;
end;
```

Apenas mude o texto Fnome para GetNome

Ctrl + Shift + C

DevTeam

```
    Fsalario: Currency;
    Fnome: string;
    Fendereco: string;

    procedure Setendereco(const Value: string);
    procedure Setnome(const Value: string);
    procedure Setsalario(const Value: Currency);

    function Getendereco: string;
    function Getnome: string;
    function Getsalario: Currency;

    protected
```

Function Get

DevTeam

AS FUNCTIONS SEMPRE RETORNA UM VALOR

```
function Tpessoa.Getendereco: string;
begin
   result:=Fendereco;
end;

function Tpessoa.Getnome: string;
begin
   result:=Fnome;
end;

function Tpessoa.Getsalario: Currency;
begin
  result:=Fsalario;
end;
```

Retorna o field(Campo)

Sem regras!!!

DevTeam

PROCEDURE DE ESCRITA ESTA PEDINDO UM VALOR (TEXTO)

```
procedure Tpessoa.Setnome(const Value: string);
begin
    if Value = '' then
    raise Exception.Create('Não pode ser um valor vazio');
    Fnome := Value;
end;
```

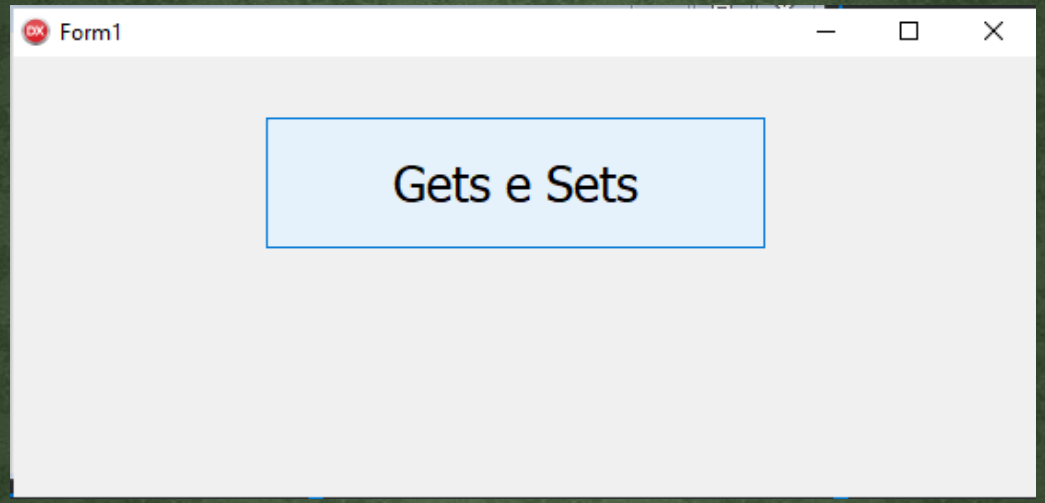
Veja que criamos uma funcionalidade no formulário com um botão pata que o mesmo Seja acionado para representar o método

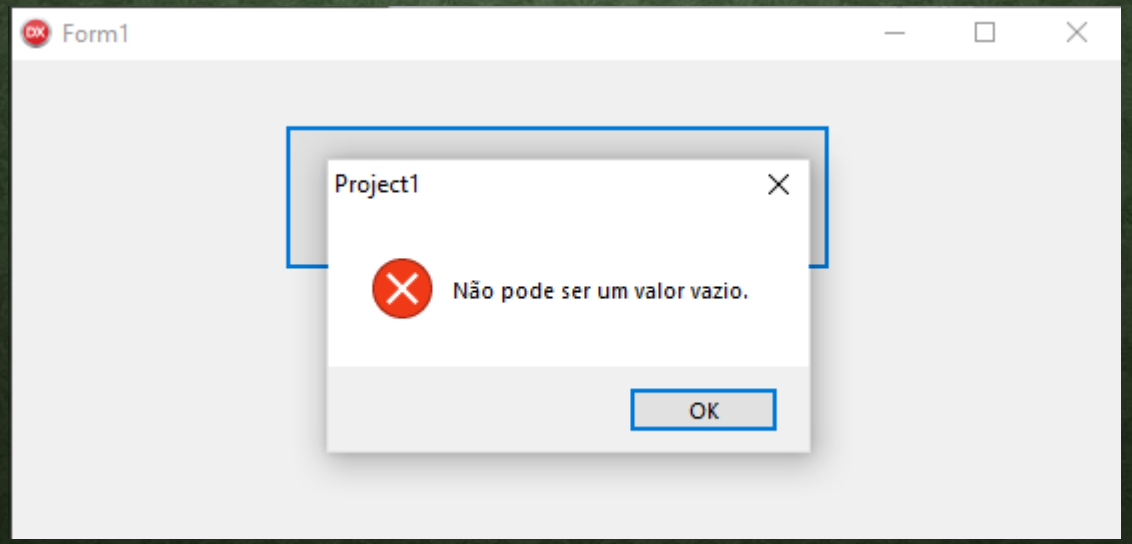
```
procedure TForm1.btn1Click(Sender: TObject);
var
pessoa:Tpessoa;
begin
  pessoa:=Tpessoa.Create;
  try
    pessoa.nome:='';
    ShowMessage(pessoa.nome);
  finally
   FreeAndNil(pessoa);
  end;
```

Veja que o pessoa.nome está vazio.
Agora vamos para o resultado.

DevTeam

PRESSIONE O BOTÃO





Tudo certinho!

DevTeam