

Validando CNPJ e CPF

Mário Leite

...

O CPF (Cadastro de **P**essoas **F**ísicas) e o CNPJ (Cadastro Nacional de **P**essoas **J**urídicas) são os dois identificadores mais importantes no Brasil; para cidadãos e empresas, respectivamente. Assim, surgem perguntas: “*como eles podem ser gerados?*” “*Como se faz para validar, ou mesmo criar, CNPJ e CPF?*” Esta é uma questão que deve merecer atenção especial por parte dos programadores; e mesmo que algumas rotinas sobre aplicações desse tipo possam ser encontradas na Internet, aprender como criar ou validar esses dados é importante no aprendizado de programação. E como foi explicado na postagem anterior sobre “Validação de CNPJ”, uma maneira de fazer isto é observar seus dois **dígitos verificadores** (aqueles dois números logo após o traço); são eles que validam, ou não, o CPF ou CNPJ. Por exemplo, assim como foi feito na postagem anterior, onde havíamos suposto que o CNPJ de uma determinada empresa era 12.345.678/0009-57 e chegamos à conclusão que o CNPJ correto era **12.345.678/0009-42**; isto é, os dois dígitos verificadores são **4** e **2**; e não **5** e **7** como considerados inicialmente. Essa verificação foi baseada no algoritmo “**Módulo 11**”, esquematizado na **figura 1a**. A **figura 1b** mostra o esquema para um exemplo de verificação do CPF.

O programa “**PrValida**”, codificado em Delphi, é um exemplo de aplicação muito simples, mas útil para criar e/ou verificar CPF/CNPJ. A **figura 3** mostra um exemplo de IDE do *Delphi 11.2* no qual a aplicação foi criada (a cor do ambiente pode ser alterada, com muitos programadores preferindo o tema “*dark*”). A **figura 4** mostra a interface deste programa em tempo de projeto, com dois *boxes*: o da esquerda gera os dois DVs do CPF/CNPJ: se forem digitados nove números na primeira caixa de texto será considerado CPF, e se forem digitados doze será considerado CNPJ. Nos dois casos, ao clicar no botão rotulado como **[Gerar CPF/CNPJ]** depois da entrada, serão gerados os dois dígitos verificadores (DV) e o resultado é exibido na segunda caixa de texto e formatado adequadamente. Clicando no botão **[Limpar]** deste primeiro *box* as duas caixas de textos serão limpas e o foco volta para a primeira caixa. O *box* da direita verifica o CPF ou CNPJ digitado na primeira caixa de texto, informando se o dado foi digitado corretamente (12 para CPF e 14 para CNPJ) ou se o CPF/CNPJ é válido, ou não, ao confirmar no botão **[Verificar CPF/CNPJ]**. E tal como acontece no primeiro *box*, clicando no botão **[Limpar]** as duas caixas de textos serão limpas e a primeira caixa focalizada, solicitando uma nova entrada de dado. As diversas situações da aplicação são mostradas na sequência de figuras: **3a, 3b, 3c, 3d, 4a, 4b, 4c**.

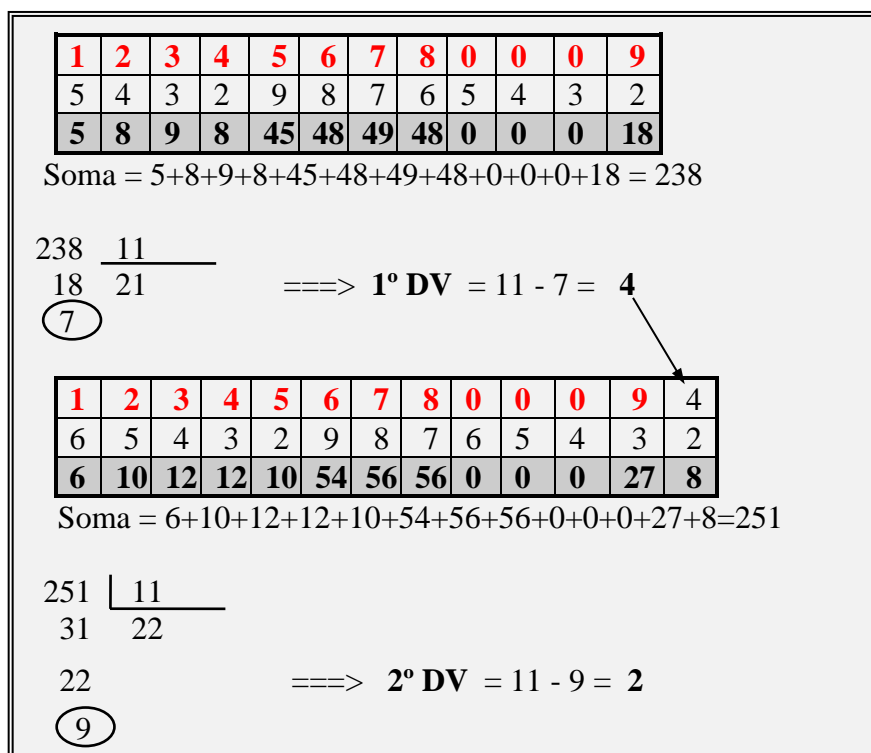


Figura 1a - Algoritmo “Módulo 11” para calcular DV de CNPJ

A **figura 1b** mostra um esquema deste método para validar um suposto **CPF 123.456.789-00**.

CPF	1	2	3	4	5	6	7	8	9	0	0
X	10	9	8	7	6	5	4	3	2		
Produto	10	18	24	28	30	30	28	24	18		

Soma1 = 10 + 18 + 24 + 28 + 30 + 30 + 28 + 24 + 18 = **210**
DV1 = 11 - (210 mod 11) = 11 - 1 = 10 (exceção) ==> DV1 = 0

CPF	1	2	3	4	5	6	7	8	9	0	0
X	11	10	9	8	7	6	5	4	3	2	
Produto	11	20	27	32	35	36	35	32	27	0	

Soma1 = 11 + 20 + 27 + 32 + 35 + 36 + 35 + 32 + 27 = **255**
DV2 = 11 - (255 mod 11) = 11 - 2 = 9 ==> DV2 = 9

Figura 1b - Algoritmo “Módulo 11” para calcular DV de CPF

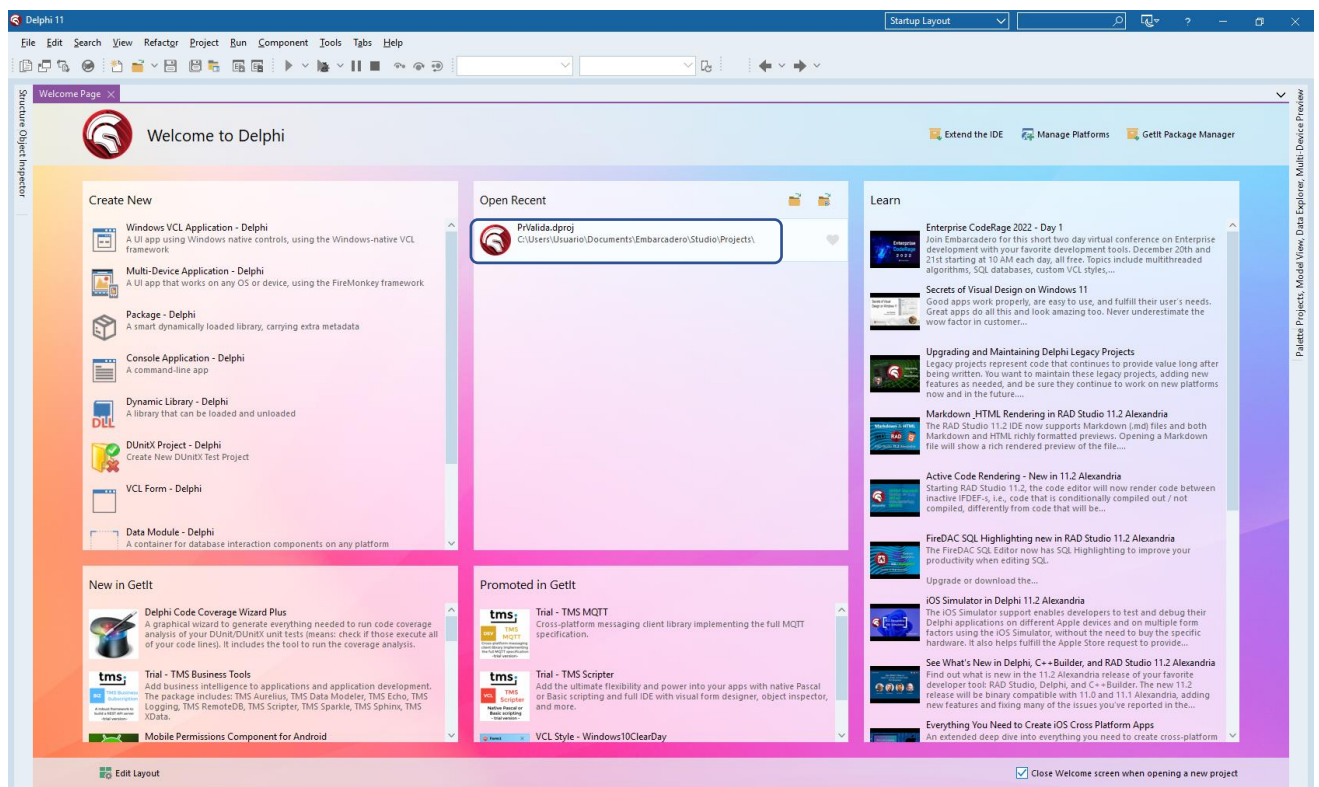


Figura 2 - Ambiente de desenvolvimento integrado do Delphi 11.2

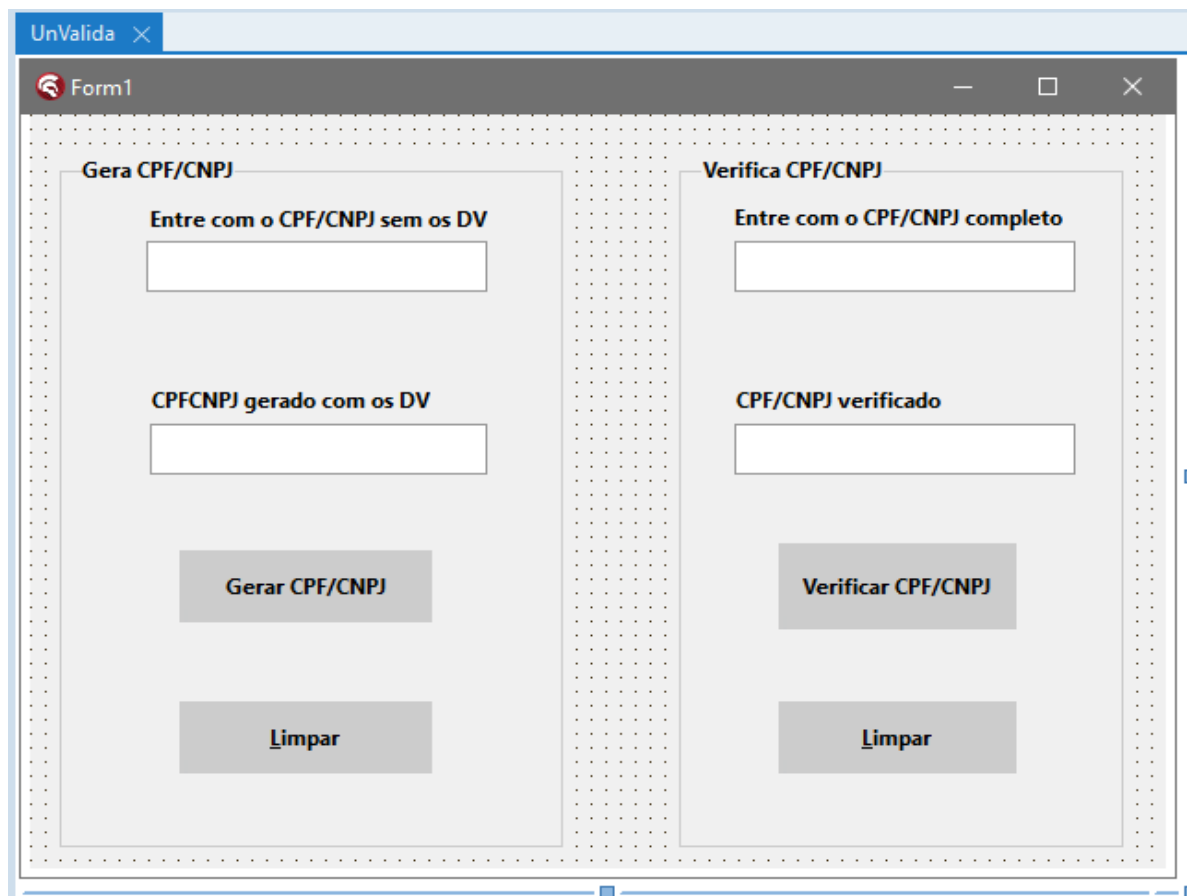


Figura 3 - Interface da aplicação em tempo de projeto

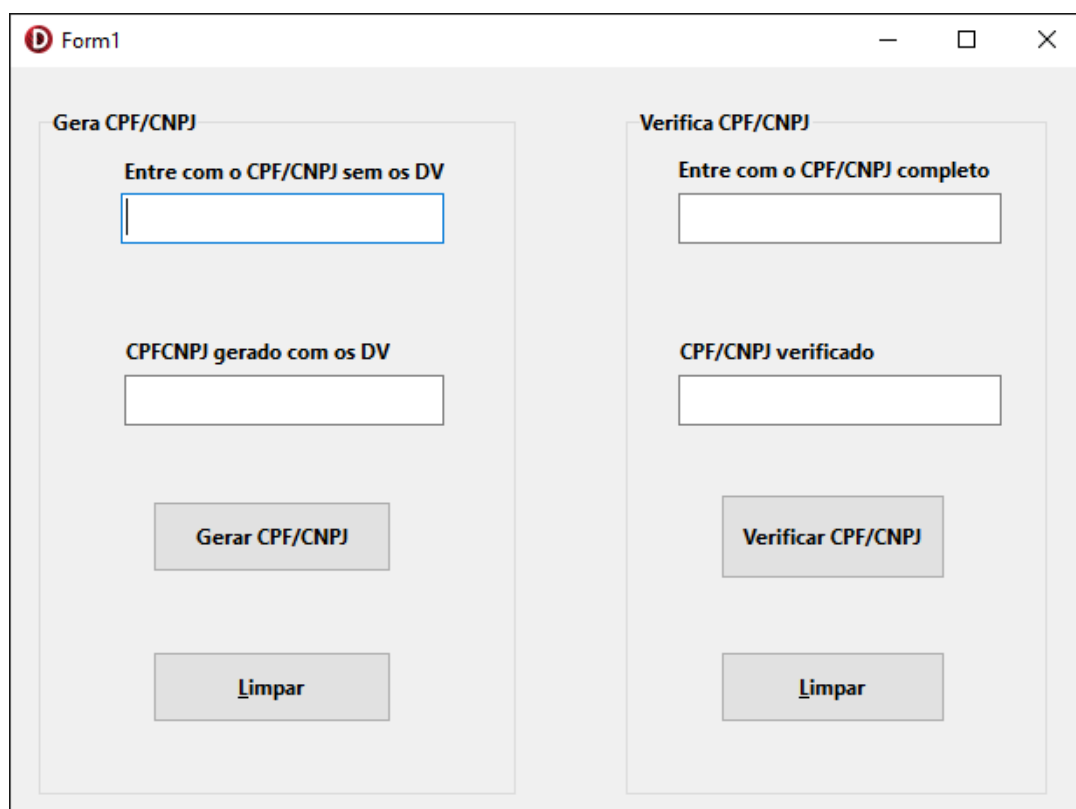


Figura 3a - Interface da aplicação rodando: sem disparos de ações

Form1

Gera CPF/CNPJ

Entre com o CPF/CNPJ sem os DV

123456789

CPF/CNPJ gerado com os DV

123.456.789-09

Gerar CPF/CNPJ

Limpar

Verifica CPF/CNPJ

Entre com o CPF/CNPJ completo

CPF/CNPJ verificado

Verificar CPF/CNPJ

Limpar

Figura 3b - Aplicação rodando: Criando um novo CPF

Form1

Gera CPF/CNPJ

Entre com o CPF/CNPJ sem os DV

CPF/CNPJ gerado com os DV

Gerar CPF/CNPJ

Limpar

Verifica CPF/CNPJ

Entre com o CPF/CNPJ completo

123456789012

CPF/CNPJ verificado

Verificar CPF/CNPJ

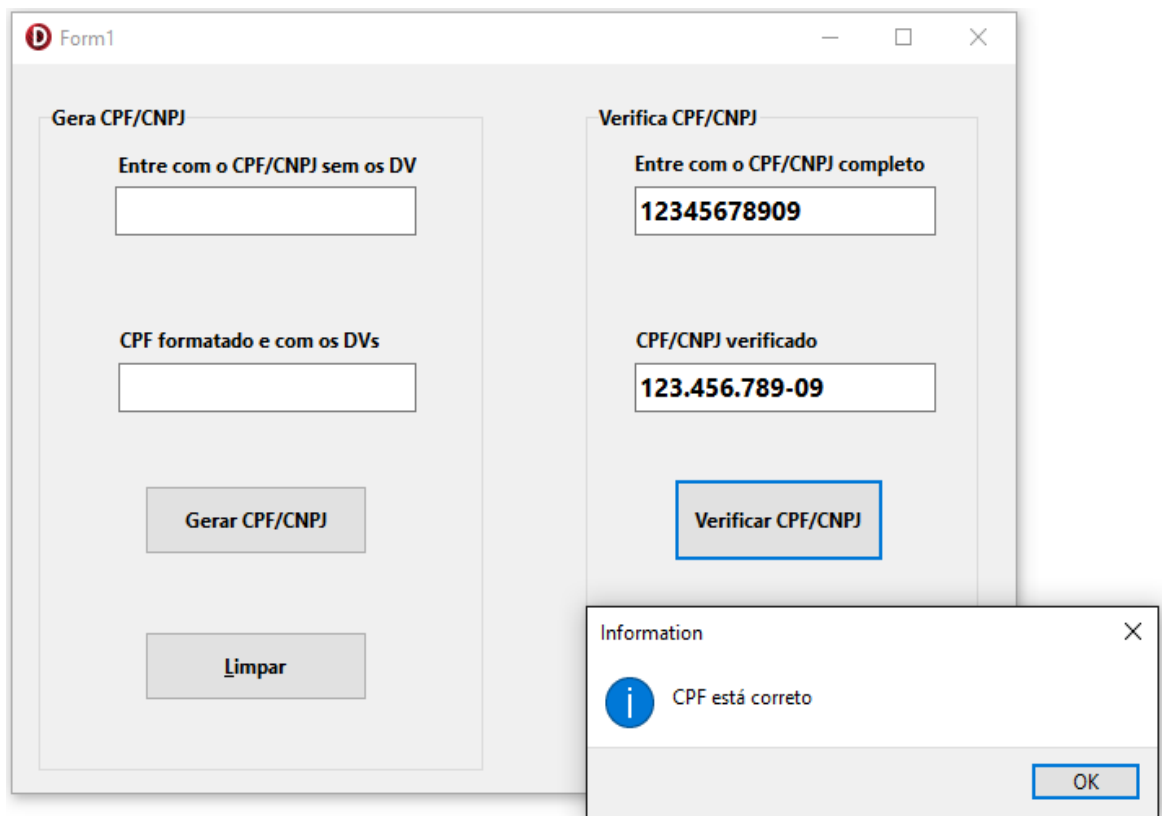
Limpar

Error

Tamanho de CPF ou CNPJ inválido

OK

Figura 3c - Verificando CPF: Inválido



The image shows a software window titled "Form1" with two main panels: "Gera CPF/CNPJ" and "Verifica CPF/CNPJ".

Gera CPF/CNPJ Panel:

- Label: "Entre com o CPF/CNPJ sem os DV"
- Input field: Empty
- Label: "CPF formatado e com os DVs"
- Output field: Empty
- Buttons: "Gerar CPF/CNPJ" and "Limpar"

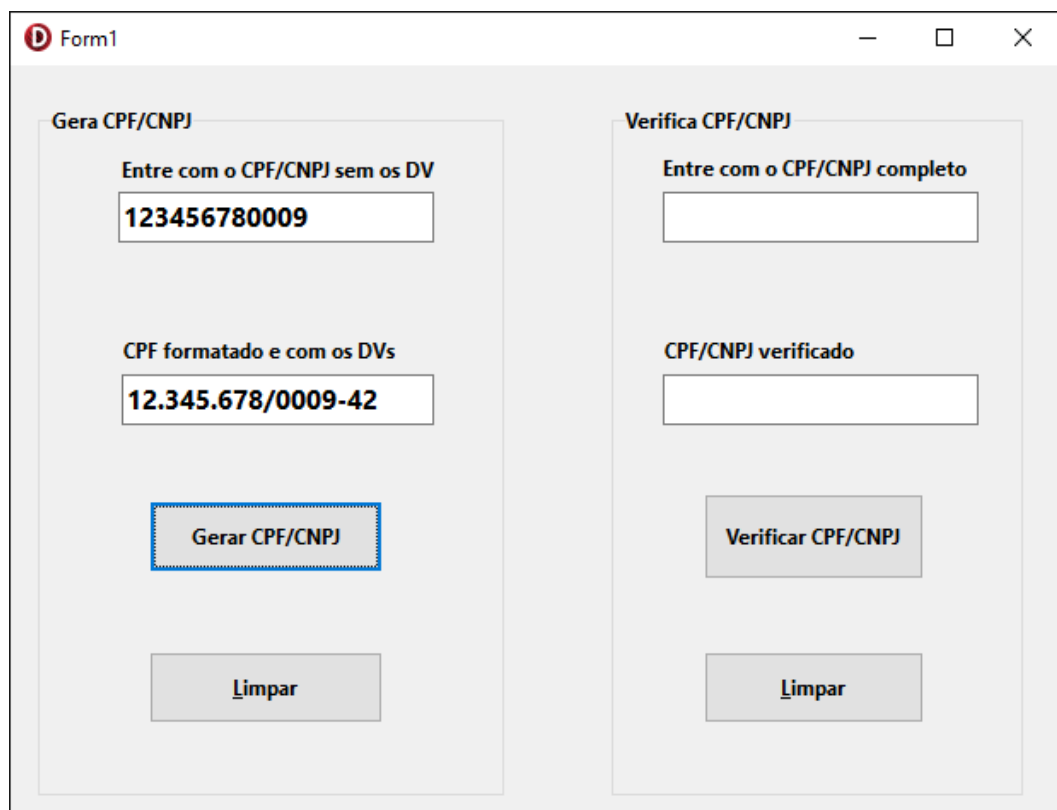
Verifica CPF/CNPJ Panel:

- Label: "Entre com o CPF/CNPJ completo"
- Input field: "12345678909"
- Label: "CPF/CNPJ verificado"
- Output field: "123.456.789-09"
- Buttons: "Verificar CPF/CNPJ" and "Limpar"

Information Dialog Box:

- Title: "Information"
- Icon: Information icon (i)
- Text: "CPF está correto"
- Button: "OK"

Figura 3d - Verificando CPF: Válido



The image shows the same software window "Form1", but with the "Gera CPF/CNPJ" panel active and the "Verifica CPF/CNPJ" panel inactive.

Gera CPF/CNPJ Panel:

- Label: "Entre com o CPF/CNPJ sem os DV"
- Input field: "123456780009"
- Label: "CPF formatado e com os DVs"
- Output field: "12.345.678/0009-42"
- Buttons: "Gerar CPF/CNPJ" and "Limpar"

Verifica CPF/CNPJ Panel:

- Label: "Entre com o CPF/CNPJ completo"
- Input field: Empty
- Label: "CPF/CNPJ verificado"
- Output field: Empty
- Buttons: "Verificar CPF/CNPJ" and "Limpar"

Figura 4a - Criando um novo CNPJ

Form1

Gera CPF/CNPJ

Entre com o CPF/CNPJ sem os DV

CPF formatado e com os DVs

Gerar CPF/CNPJ

Limpar

Verifica CPF/CNPJ

Entre com o CPF/CNPJ completo

CPF/CNPJ verificado

Verificar CPF/CNPJ

Limpar

Error

CPF ou CNPJ inválido

OK

Figura 4b - Verificando CNPJ: Inválido

Form1

Gera CPF/CNPJ

Entre com o CPF/CNPJ sem os DV

CPF formatado e com os DVs

Gerar CPF/CNPJ

Limpar

Verifica CPF/CNPJ

Entre com o CPF/CNPJ completo

CPF/CNPJ verificado

Verificar CPF/CNPJ

Limpar

Information

CNPJ está correto

OK

Figura 4c - Verificando CNPJ: Válido

Nota: Como desafio, deixo para os programadores mais modernos converterem este código em Delphi (não *case sensitive*) na sua linguagem preferida. E para orientar esta nova implementação, eis algumas palavras-chave do Delphi que podem ser desconhecidas do programador; mas, basicamente, é um Pascal no ambiente Windows.

unit até **{ \$R *.DFM }** São definições/importações de bibliotecas; por exemplo, a *unit* **System.UITypes** é onde está definida **MessageDlg()** e deve ser incluída manualmente para que esta *procedure*, que emite mensagens, possa ser usada. As outras bibliotecas são colocadas automaticamente pela ferramenta.

var Declara variáveis.

begin..end; Terminadores que marcam início e fim de um bloco de instruções: tal como { e } no C.

Copy(s,i,n) Pega uma quantidade de caracteres **n**, a partir da posição **i** de uma *string* **s**.

Trim(s) Retira os espaços em branco de uma *string* **s**.

ord(x) Função que retorna o código inteiro de um caractere ASCII **x**.

chr(n) Função que retorna o caractere ASCII do código inteiro **n**.

mod(N,D) Operador que dá o resto da divisão inteira de **N** por **D**.

Length(s) Função que retorna o tamanho de uma *string* **s**.

Form1 Nome (identificador) do formulário da janela da aplicação.

Edit Nome de caixa de texto; tal como uma *TextBox* em outras linguagens “for-Windows”.

Código-fonte na janela de codificação do Delphi

```
//PrValida
//Cria e/ou valida CPF/CNPJ
//Em Delphi 11.2
//Autor: Mário Leite
//Data: 12/02/2023
//-----

unit UnValida; //rotina principal

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms,
  Vcl.Dialogs, Vcl.StdCtrls, System.UITypes;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Edit4: TEdit;
    Button3: TButton;
    Button4: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
//=====

Function CalculaCnpjCpf(Numero: string): string;
{Função que cria/analisa CPF/CNPJ pelo "Módulo 11"
  Entrada: CPF/CNPJ em formato string (sem os dígitos verificadores)
  Saída: CPF/CNPJ em formato string (com os dígitos verificadores)
  Verifica se é CPF ou CNPJ pelo número de dígitos: 9 [CPF] ou 12 [CNPJ]
}
Var //declara variáveis locais à função
  i,j,k : integer;
  Soma : integer;
  Digito : integer;
  CNPJ : boolean;
```



```

Begin
  case Length(Numero) of //pega o tamanho do CPF/CNPJ digitado
    9: CNPJ := False;
    12: CNPJ := True;
  else
    begin
      MessageDlg('Quantidade de dígitos inválida', mtError, [mbOK], 0);
      exit;
    end;
  end;
result := Numero;
for j:=1 to 2 do begin
  k := 2; Soma := 0;
  for i:= Length(result) downto 1 do begin //loop decrementado de 1
    Soma := Soma + (ord(result[i]) - ord('0'))*k;
    inc(k);
    if(k>9) and CNPJ then
      k := 2;
  end;
  Digito := 11 - (Soma mod 11);
  if(Digito>=10) then
    Digito := 0;
  result := result + chr(Digito + ord('0'));
end;
End; //fim da função "CalculaCnpjCpf"
//-----
Procedure TForm1.Button1Click(Sender: TObject);
var //declara variáveis locais à rotina deste evento
  tam: integer;
  str1, str2, str3, str4, str5: string;
Begin
  Edit2.Text := CalculaCnpjCpf(Edit1.Text); //chama função e pega o retorno
  tam := Length(Trim(Edit1.Text)); //pega o tamanho da string (CPF/CNPJ)
  case tam of
    9: //Caso seja CPF
      begin
        str1 := Copy(Edit2.Text, 1, 3);
        str2 := Copy(Edit2.Text, 4, 3);

        str3 := Copy(Edit2.Text, 7, 3);
        str4 := Copy(Edit2.Text, 10, 2);
        Edit2.Text := str1 + '.' + str2 + '.' + str3 + '-' + str4;
        Label2.Caption := 'CPF formatado e com os DVs'
      end;
    12: //Caso seja CNPJ
      begin
        str1 := Copy(Edit2.Text, 1, 2);
        str2 := Copy(Edit2.Text, 3, 3);
        str3 := Copy(Edit2.Text, 6, 3);
        str4 := Copy(Edit2.Text, 9, 4);
        str5 := Copy(Edit2.Text, 13, 2);
        Edit2.Text := str1 + '.' + str2 + '.' + str3 + '/' + str4 + '-' + str5;
        Label2.Caption := 'CNPJ formatado e com os DVs'
      end;
  end; //fim do case
End; //fim da procedure-evento "Button1Click"
//-----
Procedure TForm1.Button2Click(Sender: TObject);
begin
  //limpa caixas de texto do CPF/ CNPJ gerado
  Edit1.Clear;
  Edit2.Clear;
  Edit1.SetFocus;
End;
//-----

```

```

Procedure TForm1.Button3Click(Sender: TObject);
var
    tam1, tam2: integer;
    str1, str2, str3, str4, str5: string;
    texto1, texto2: string;
Begin
    tam1 := Length(Trim(Edit3.Text));
    if ((tam1 <> 11) and (tam1 <> 14)) then
        begin
            MessageDlg('Tamanho de CPF ou CNPJ inválido', mtError, [mbOK], 0);
            Edit3.SetFocus;
            Edit3.SelStart;
            Edit3.SelLength := Length(Edit3.Text);
            exit; //abandona a rotina
        end;
        //Verifica tamanho da entrada para preparar texto
        case tam1 of
            11: texto1 := Copy(Trim(Edit3.Text), 1, 9); //caso seja CPF
            14: texto1 := Copy(Trim(Edit3.Text), 1, 12); //caso seja CNPJ
        end; //fim do case
        //Envia texto para a geração dos dígitos
        tam2 := Length(texto1);
        texto2 := CalculaCnpjCpf(texto1); //chama função e pega o retorno
        if (Trim(Edit3.Text) <> Trim(texto2)) then
            begin
                MessageDlg('CPF ou CNPJ inválido', mtError, [mbOK], 0);
                Edit3.SetFocus;
                Edit3.SelStart;
                Edit3.SelLength := Length(Edit3.Text);
                Exit; //abandona a procedure e retorna
            end;
        //Formata texto de acordo com o tamanho da entrada
        case tam2 of
            9: //caso seja CPF
                begin
                    str1 := Copy(texto2, 1, 3);
                    str2 := Copy(texto2, 4, 3);
                    str3 := Copy(texto2, 7, 3);
                    str4 := Copy(texto2, 10, 2);
                    Edit4.Text := str1 + '.' + str2 + '.' + str3 + '-' + str4;
                    MessageDlg('CPF está correto', mtInformation, [mbOK], 0);
                    Label4.Caption := 'CPF formatado e com os DVs'
                end;
            12: //caso seja CNPJ
                begin
                    str1 := Copy(texto2, 1, 2);
                    str2 := Copy(texto2, 3, 3);
                    str3 := Copy(texto2, 6, 3);
                    str4 := Copy(texto2, 9, 4);
                    str5 := Copy(texto2, 13, 2);
                    Edit4.Text := str1 + '.' + str2 + '.' + str3 + '/' + str4 + '-' + str5;
                    MessageDlg('CNPJ está correto', mtInformation, [mbOK], 0);
                    Label4.Caption := 'CNPJ formatado e com os DVs'
                end;
        end;
    End; //fim da procedure-evento "Button3Click"
    -----
Procedure TForm1.Button4Click(Sender: TObject);
Begin
    Edit3.Clear;
    Edit4.Clear;
    Edit3.SetFocus;
End; //fim da procedure-evento "Button4Click"

end. //===== Fim da Aplicação =====

```