



# Curso de criação de componentes em Delphi



## Unidade 3. Um componente não-visual: TNif

[Voltar ao índice](#)

Por Luis Roche



**NIF** Chegou o momento de entrar em ação. Nós criaremos nosso primeiro componente: TNif. Mas antes de começar a escrever código isto serve para analisar o propósito do componente e como nós implementaremos isto.

### • Objetivo do componente

O propósito do componente é muito simples: a partir de um dado de entrada, o número de DNI, nosso componente deverá calcular a letra do NIF correspondente. O método de calcular esta letra é por meio de uma operação matemática muito simples. Esta operação é:  $DNI - ((DNI \div 23) * 23)$ . O resultado desta operação será um número entendido entre 1 e 23. Por meio de um quadro nós nomearemos uma certa letra a cada número, a letra que corresponde o NIF pedido. Os detalhes completos desta letra eu tenho no código fonte.

### • O projeto do componente

Baseado em nosso objetivo, está claro que nosso componente será do tipo não-visual e, então, nós derivaremos isto a partir de TComponent que eu como já foi visto; (unidade 2), é a base para a criação de componentes não-visuais.

A forma de introduzir o número de DNI será por meio de uma propriedade (propriedade DNI) de leitura. O valor desta propriedade será armazenada em um campo (FDNI) do tipo LongInt . Esta propriedade será publicada (published) de forma que ela aparece deste modo no Object Inspector . A leitura e escrita de valores nesta propriedade serão diretamente no campo FDNI, quer dizer, nós deixaremos que o próprio Object Inspector se encarrega de verificar se o valor introduzido orresponde ao tipo LongInt.

Nós teremos outra propriedade (NIF) no que a letra calculada será armazenada (de tipo Char). Mas esta propriedade será somente leitura ja que o usuário do componente não poderá inserir a letra manualmente, pois o proprio componente que calcula isto. Esta propriedade deveria ser pública (Public) desde que seja somente leitura. Este aspecto serve para se salientar isto: *as propriedades somente-leitura deverão ser declaradas na parte pública*. A função que se encarrega de calcular a letra do NIF se

chamará GetNIF.

Sempre serão declarados os campos que armazenam o valor de propriedades na parte privada (Private) desde que nós tenhamos certeza de que todos tenham acesso a ele, mas o usuário do componente, desde que se conscientize pela propriedade e não pelo campo (isso representa o armazenamento interno da propriedade).

Relativo aos nomes usados, são continuadas as convenções seguintes:

- Os tipos que nós definimos T começará com a letra (de tipo). P.e. TNif
- Os campos que armazenam os valores de propriedades começam com a carta F seguido pelo nome da propriedade que armazena. Deste modo o campo FDNI está claro que armazena o valor da propriedade DNI.
- Os nomes dos métodos de leitura e dos valores de uma propriedade, eles serão denominados pelo prefixo Get (para ler) ou Set (para escrever) seguido pelo nome da propriedade. P.e. o método GetNIF.

Uma vez o método de empregado de desígnio de acordo é hora de começo para digitar.

## ● Codifique fonte do componente

```
unit Nif;           { (c)1996 by Luis Roche }

interface

uses
  Classes;

type
  TNif = class(TComponent)
  private
    FDNI : LongInt;
    function GetNIF : char;
  protected
  public
    property NIF: char read GetNIF;
  published
    property DNI: LongInt read FDNI write FDNI;
  end;

procedure Register;

implementation

function TNIF.GetNIF : char;
Var aux1 : integer;
Const letras : string = 'TRWAGMYFPDXBNJZSQVHLCKE';
begin
  aux1:=FDNI - ((FDNI div 23) * 23);
  result:=letras[aux1+1];
end;

procedure register;
begin
  registercomponents('curso', [tnif]);
```

end ;

end .

## ● Comentários para o código fonte

O código fonte tem que ser criado através dos seguintes passos:

Nós usamos o Expert de componentes de forma que nós geramos a unidade na qual nós escrevemos nosso componente. Na Caixa de diálogo que o expert nos mostrou, nós introduzimos TNif como Nome da Classe Name, TComponent como Tipo de Ascendente e Curso como Nome da Paleta. Quando clicamos OK, o expert de componentes cria todo o esqueleto básico de nosso componente, inclusive o Registro de procedimento.

De todas as unidades que aparecem na cláusula uses nós só deixaremos a unidade Classes, pois nós não usaremos qualquer procedimento dos outros.

Nós declaramos o campo FDNI (longInt) na seção privada. Neste campo, como já foi dito, nós armazenaremos o DNI. Na seção published nós escrevemos a linha seguinte:

**propriedade DNI: LongInt read FDNI write FDNI**

Deste modo nós declaramos a propriedade DNI e nós especificamos que a leitura e escrita de valores no mesmo são feitos diretamente com o campo FDNI que usa o Object Inspector.

Nós declaramos a propriedade NIF somente-leitura na parte publica. Ele serve para se lembrar que as propriedades de leitura só deveriam ir declaradas na parte pública e não no published. Nós especificamos isso para ler o valor da propriedade: nós usaremos a função GetNIF, o qual nós declaramos na seção privada.

Nós escrevemos a função que calcula a letra do NIF na parte de implementação da unidade.


Nós mantemos a unidade com o nome nif.pas. Veste que todos os componentes que nós vamos criar durante o curso separados em diretório (delphi\componen)

## ● Criando um bitmap para o componente

Cada componente precisa de um bitmap para representar ao componente na paleta de componentes. Se nenhum for especificado, o Delphi usará um como padrão.

O bitmap não é incluído no código fonte do componente, mas é incluído separado em um arquivo com a extensão .DCR (Dynamic Component Resource). Este arquivo pode ser criado com o próprio editor de imagens que vem com o Delphi.

O nome do arquivo que .DCR deveria coincidir com o nome da unidade do seu componente. O nome do bitmap de imagem (em letra maiúscula) devera coincidir com o nome do componente. Os dois arquivos (o da unidade \* .pas e que do bitmap \* .dcr) devem residir no mesmo diretório.

Em nosso componente, se nós salvamos a unidade com o nome nif.pas o arquivo de recursos se chamará nif.dcr. Dentro deste arquivo estarão o bitmap para os quais nós colocaremos o nome TNIF. O bitmap que nós criamos:  O tamanho do bitmap devera ser de 24x24 pixels.

Como último detalhe, se você quer usar este mesmo bitmap, você pode usar um programa de tratamento de imagens para recortar isto e colar no editor de imagens do Delphi.

## ● **TNif instalando na paleta de componentes.**

Para instalar nosso componente na paleta de componentes é muito simples. Siga os passos:

- Escolher o menu Componentes de Options|Install .
- Clicar no botão Add, selecionar o local que contém o código fonte do componente. Em nosso caso, nif.pas. Uma vez selecionado a unidade, clicar OK.
- O Delphi compilará a unidade e reconstruirá o arquivo COMPLIB.DCL. Se ele gerar erros durante a compilação, nós seremos informados disto.
- Se a compilação tem êxito, nosso componente fara parte da paleta de componentes.

---

*Luis Roche* [revueltaroche@redestb.es](mailto:revueltaroche@redestb.es)

Ultima modificación 5.12.1996