

```

[section]
keyname=valor ;comment
variavel=valor ;comentario
[secao]
variavel=valor ;comentario
[dados]
janela1=1
janela2=0
[[janela1pos]
x=100 ;posição x da janela 1
y=210 ;posição y da janela 1
[[janela1size]
x=250
y=250
[[janela2pos]
x=0 ;posição x da janela 2
y=0 ;posição y da janela 2
[[janela2size]
x=250
y=250
[section]
keyname=valor ;comment
variavel=valor ;comentario
[secao]
variavel=valor ;comentario
[dados]
janela1=1
janela2=0
[[janela1pos]
x=100 ;posição x da janela 1
y=210 ;posição y da janela 1
[[janela1size]
x=250
y=250
[[janela2pos]
x=0 ;posição x da janela 2
y=0 ;posição y da janela 2
[[janela2size]

```

TRABALHANDO COM ARQUIVOS INI NO DELPHI

POR: [DANILLO RODRIGUES](#)

➔ INTRODUÇÃO

Os arquivos INI são arquivos de configuração, que podem ser facilmente visualizados, modificados ou apagados.

Antes de utilizarmos este tipo de arquivo com o Delphi® iremos entender o funcionamento dele.

Observe o exemplo abaixo:

```
[seção1]
variavel=valor      ;comentario
nome=arquivoini    ;comentario

[seção2]
nome=outronome
```

A seção deve estar sempre no começo da linha e entre colchetes. Os nome de seção são case-sensitive (ou seja, “Seção” é diferente de “seção”). As seções podem conter variáveis com nomes iguais.

A variavel, assim como no delphi, pode conter valores do tipo texto (string), inteiros (integer), booleanos (bool), etc...

Caso exista duas seções com o mesmo nome, ou duas variáveis com o mesmo nome dentro de uma mesma seção então a ultima ocorrência prevalece.

Uma das desvantagens do arquivo INI é que ele não é muito seguro pois a pessoa pode modifica-lo facilmente com um editor de textos (Exemplo: Bloco de Notas).

➔ DELPHI

O Delphi tem uma classe chamada TiniFile que contém procedimentos e funções para trabalhar com arquivos INI. A classe TiniFile pode ser utilizada declarando-se a unit inifiles.pas na clausula **uses**.

```
uses inifiles;
```

Para utilizar um arquivo ini, primeiro devemos definir uma variavel do tipo TiniFile. (O codigo abaixo pode ser colocado em uma function ou em um procedure (como no clique de um botão)).

```
var
arquivo : TiniFile;      //variavel do tipo TiniFile
```

```
arquivo := Tinifile.Create('arquivoini.ini');  
//Inicializa a variavel arquivo especificando o arquivo  
"arquivoini.ini" como o arquivo INI que vai ser  
utilizado.
```

No código acima, caso o arquivo “arquivoini.ini” não exista ele será criado. Neste caso, o arquivo será criado/procurado na pasta Windows, mas se você quiser você pode definir um diretório.

```
arquivo := Tinifile.Create('c:\arquivoini.ini');
```

Ou, para facilitar mais, colocar o arquivo INI no mesmo diretório do executavel.

```
var  
dir : string;  
begin  
dir := extractfilepath(application.exename);  
arquivo := Tinifile.Create(dir + 'arquivoini.ini');  
end;
```

➔ FUNÇÕES DE ESCRITA

As funções para escrever são simples, basta usar a variavel antes definida (no nosso exemplo, a variavel “arquivo”) e após ela colocar um ponto (.) e a função desejada.

Exemplo de funções:

WriteString:

Escreve uma variavel com valor string, exemplo abaixo:

```
Arquivo.writeString('seção', 'variavel', 'valor');
```

WriteInteger:

Escreve uma variavel com valor integer, exemplo abaixo:

```
Arquivo.writeInteger('seção', 'variavel', 123456);  
//Aonde 123456 é o valor a ser gravado ( valor numérico)
```

WriteBool:

Escreve uma variavel com valor booleano (**true** ou **false**), exemplo abaixo:

```
Arquivo.writeBool('seção', 'variavel', true);  
//Aonde true é o valor a ser gravado ( valor booleano)
```

Existem também outros formatos de variaveis que podem ser gravadas no arquivo INI (Float, Time, Date, DateTime).

➔ FUNÇÕES DE LEITURA

Assim como as funções de escrita, as funções de leitura são fáceis de usar, e tem a utilidade de pegar um valor escrito em uma variável do arquivo INI. Antes vamos declarar algumas variáveis que serão utilizadas para receber o valor lido do arquivo INI.

```
var
rstr : string;           //variavel que será utilizada
                           //para receber um valor string
rint : Integer;         //variavel que será utilizada
                           //para receber um valor integer
rbool : bool;           //variavel que será utilizada
                           //para receber um valor bool
```

ReadString:

Lê uma variável de valor string, caso você use o ReadString para ler uma variável que contenha um valor Integer, este valor será retornado como um texto. Exemplo abaixo:

```
rstr := arquivo.ReadString('seção','variavel', 'valor
padrão');
//passa para variavel "rstr" o valor da variavel
//"variavel" contida na seção "seção".
//Caso a variavel lida não exista ela retorna o "valor
//padrão"
```

ReadInteger:

Lê uma variável de valor integer. Exemplo abaixo:

```
rint := arquivo.ReadString('seção','variavel', 54321);
//passa para variavel "rint" o valor da variavel
//"variavel" contida na seção "seção".
//Caso a variavel lida não exista ela retorna o valor
//padrão definido acima como 54321 ( você pode definir o
//valor que quiser)
```

ReadBool:

Lê uma variável de valor bool (**true** ou **false**). Exemplo abaixo:

```
rbool := arquivo.ReadString('seção','variavel', false);
//passa para variavel "rint" o valor da variavel
//"variavel" contida na seção "seção".
//Caso a variavel lida não exista ela retorna o valor
//padrão definido acima como false ( você pode definir
//true ou o valor que quiser)
```

➔ OUTRAS FUNÇÕES

Também existem outras funções que nos ajudam bastante a lidar com um arquivo INI.

EraseSection

Deleta uma seção inteira (todas as suas variaveis)

```
arquivo.eraseSection( 'exemplo' );  
//deleta a seção "exemplo"
```

SectionExists

Retorna True caso a seção exista e False caso ela não exista.

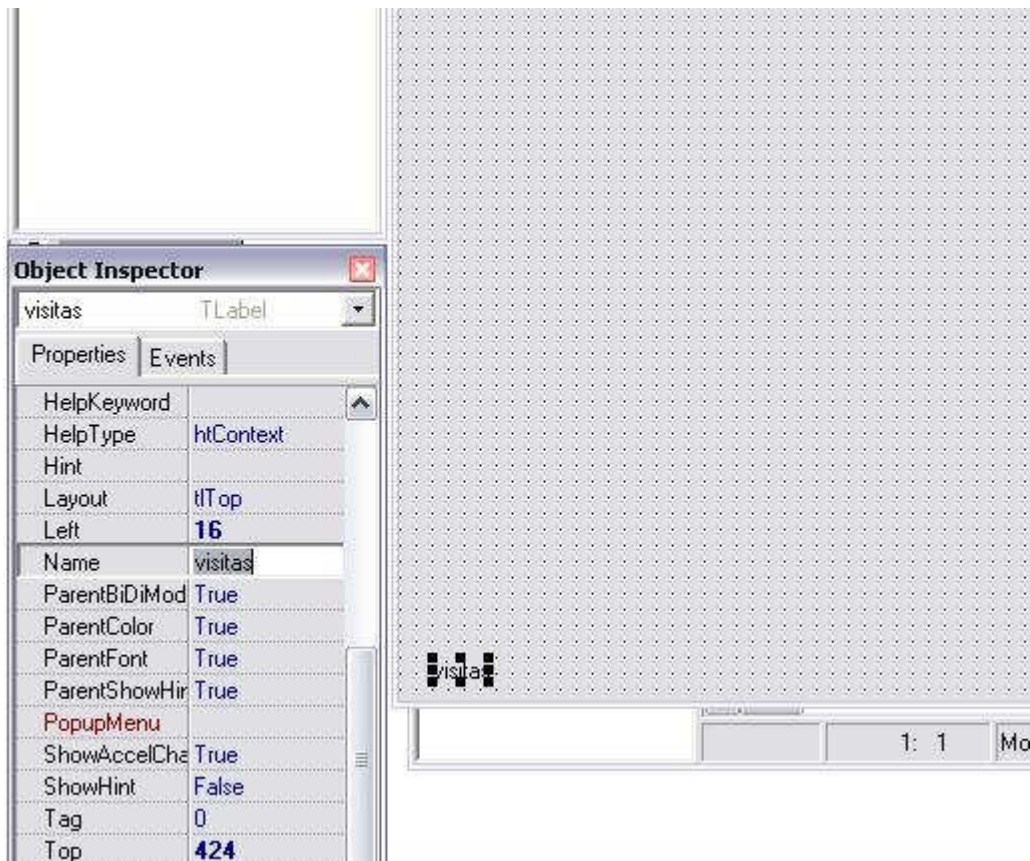
```
If arquivo.SectionExists( 'exemplo' ) then  
showmessage( 'Existe' );  
//verifica a existencia da seção "exemplo"
```

Existem também outras funções que não foram faladas aqui.

➔ EXEMPLOS

Exemplo 1:

Exemplo de contador de visitas(conta quantas vezes você entro no programa). No formulário coloque uma Label e nomeie ela como visitas.(fig1)



A lógica é a seguinte, quando a pessoa abrir o programa, o programa consultara no arquivo visitas.ini o número de visitas, se a pessoa visitou já 1 vez, então esta é a 2ª visita.

Coloque o código abaixo no onShow do formulário.

Obs.: Não esqueça de declarar a unit Inifiles na clausula **Uses** do formulário.

```
var
arquivo : Tinifile;
Nvisitas: integer;
begin
arquivo := Tinifile.Create('visitas.ini');
Nvisitas := arquivo.ReadInteger('exemplo1','visitas',0);
//A variavel Nvisitas recebe o valor integer lido do arquivo ini
//da variavel "visitas" contida na seção [exemplo1], sendo que o valor
//padrão definido ( o valor que é passado para a variavel caso não seja
//encontrada a variavel "visitas" na seção "exemplo1") é 0

//Como o arquivo guarda o número de vezes que a pessoa ja visitou o programa
//O numero dessa visita seria, o numero de visitas +1;
//abaixo o codigo!
Nvisitas := Nvisitas +1;
visitas.Caption := 'Esta é a '+inttostr(nvisitas)+'ª vez que você usa o
programa';

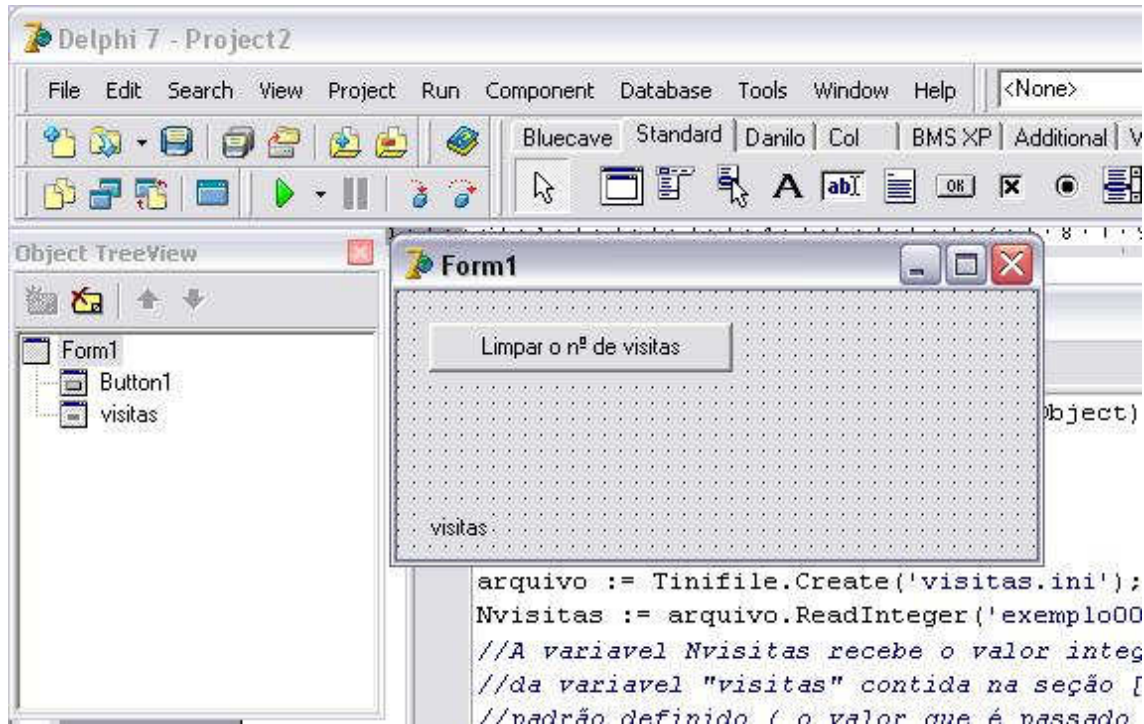
//agora vamos salvar a informação o nº de visitas até agora
arquivo.WriteInteger('exemplo1','visitas',Nvisitas); //guarda o numero
contido na variavel NVisitas
arquivo.Free; //Libera a variavel arquivo da memória!
```

Compile o programa e veja o resultado:



Feche o programa e abra ele denovo (mesmo que você recompile ele, observe o contador de visitas continuara funcionando).

Agora vamos fazer um botao para limpar o número de visitas. Adicione um botão e mude o caption dele para “Limpar o nº de visitas”.(Na figura abaixo eu mudei um pouco o layout para deixa-lo so com o espaço necessário para o botão e o label)



No evento OnClick deste botão, coloque o código abaixo:

```
var
arquivo : TiniFile;
begin
arquivo := TiniFile.Create('visitas.ini');
//inicializa a variavel arquivo com o arquivo visitas.ini
arquivo.WriteInteger('exemplo1', 'visitas', 0);
//escreve 0 na variavel visitas dentro da seção exemplo1,
//limpando assim o nº de visitas
arquivo.Free;
//libera a variavel arquivo da memória
end;
```

Rode o programa e clique no botão. Agora feche e abra denovo! Pronto, agora apareceu que esta é a primeira vez que você entra. O arquivo visitas.ini foi criado no diretório Windows (pois não definimos nenhum diretório)

Fim do Exemplo.

O código de fonte deste exemplo pode ser encontrado no link abaixo:

<http://www.bestsharing.com/files/ms00171936/exemplo1.zip.html>

Exemplo 2:

Exemplo de como salvar as posições top e left, e o tamanho width e height do formulário. Este exemplo pode ser encontrado no link abaixo

<http://www.bestsharing.com/files/ms00171957/exemplo2.zip.html>

➔ NOTAS FINAIS

Observação:

Sempre que terminar de usar a variável que tem acesso ao arquivo INI libere ela da memória. Exemplo abaixo:

```
var
arquivo : Tinifile;
begin
arquivo := Tinifile.Create('teste.ini');
//inicializa a variavel arquivo com o arquivo teste.ini
arquivo.Free;
//libera a variavel arquivo da memória
end;
```

A classe TRegIniFile foi introduzida para um fácil acesso ao Registro do sistema sem ter que conhecer a estrutura do registro. TRegIniFile tem funcionalidade de acesso ao Registro igual a TIniFile tem aos arquivos .INI.

Limitações: A classe TIniFile usa a API do Windows que impõe um limite de 64kb nos arquivos INI. Se você precisar armazenar mais de 64kb de dados, você precisará usar o TMemIniFile ao invés da TIniFile, que não possui o limite de 64kb.

➔ SOBRE

Tutorial desenvolvido por Danilo Rodrigues.

Este tutorial não pode ser comercializado de nenhuma forma, pois o seu conteúdo é livre e de livre acesso a qualquer usuário.

Se você tiver alguma critica, duvida, comentario, problema com o tutorial ou com os exemplos, problema com links quebrados envie um e-mail para: danielod100@terra.com.br