

Record

É um tipo de dado do Object Pascal (hoje não muito utilizado devido ao uso de Orientação a Objetos), funciona como um Array, mas a vantagem é que pode conter diferentes tipos de dados (String, Integer, Record, etc).

O Record trabalha com tamanho fixo de acordo com a quantidade de dados que você determinar ao Record. A aplicação do Record pode ser de inúmeras formas, abaixo veremos alguns exemplos:

```
type
```

```
// em type declare Record
```

```
TPessoa = record
```

```
nome: string[30];
```

```
idade: byte;
```

```
end;
```

```
var
```

```
// Em var declare a variável do tipo Record para utilizar na aplicação
```

```
Pessoa : TPessoa;
```

{Repare que depois de “begin” podemos utilizar um dado contido dentro de uma Estrutura (Record), para isso basta digitar a variável do tipo da estrutura seguido de ponto e o nome do campo declarado em Record. O delphi sugere o nome do campo quando se usa o recurso de auto completar - CTRL + Barra de Espaço.}

```
begin
```

```
// Informe os dados para os campos do Record (Estrutura)
```

```
Pessoa.name := 'João Marcos';
```

```
Pessoa.idade:= 25;
```

```
end;
```

Também é possível evitar a digitação repetitiva da variável pessoa caso sua estrutura seja mais complexa, para isso basta utilizar o recurso “with”, abaixo temos o exemplo de utilização desse recurso disponível no Delphi.

```
type
```

```
// Declare Record logo abaixo da cláusula type
```

```
TPessoa = record
```

```
nome: string[30];
```

```
idade: byte;
```

```
end;
```

```
var
```

```
// Declare a variável do tipo Record para utilizar na aplicação
```

```
Pessoa : TPessoa;
```

```
begin
```

```
// Abaixo, com o recurso with é possível digitar somente o nome dos campos
```

```
// que estão contidos na estrutura Pessoa, não sendo necessário fazer como
```

```
// no exemplo anterior Pessoa.nome := 'João Marcos';
```

```
with Pessoa do
```

```
begin
nome := 'João Marcos';
idade := 25;
end;
```

Até aqui tudo está fácil e maravilhoso, mas vamos complicar um pouco mais para que você possa saber até onde se pode ir com o tipo Record.

```
// Código completo com array de records
// -----
// Você pode copiar esse código e colar numa unit do Delphi
// e utiliza-lo fazendo chamadas pelo Evento OnCreate do Form.
```

```
unit uPrincipal;
```

```
interface
```

```
uses
Forms, Dialogs;
```

```
type
TFrmRecords = class(TForm)
procedure FormCreate(Sender: TObject);
// Procedimento Para mostrar os dados da estrutura Pessoa
procedure MostrarPessoa(const campos: array of string);
end;
```

```
var
FrmRecords: TFrmRecords;
```

```
implementation
{$R *.dfm}
```

```
procedure TFrmRecords.FormCreate(Sender: TObject);
type
// Outro modo que podemos utilizar para criar a estrutura
// é declara-la dentro de uma procedure. Nesse exemplo
// declaramos dentro do Evento OnCreate do Form.
TPessoa = Record
PrimeiroNome : string[20];
UltimoNome : string[20];
Endereco : string[100];
Bairro : string[100];
Cidade : string[50];
CodigoPostal : string[8];
end;
```

```
var
// Em seguida declaramos um array do tipo da estrutura TPessoa.
// Assim trabalharemos com 3 "Pessoas".
Pessoas: array[1..3] of TPessoa;
i : Integer;
```

```

begin
// Agora vamos informar os dados de cada pessoa.
// Lembrando que esse é um exemplo, é possível trabalhar com a
// estrutura de repetição "for" normalmente.

with Pessoas[1] do
begin
PrimeiroNome := 'João';
UltimoNome := 'Sakalauska';
Endereco := 'Av. Brasil, 1000';
Bairro := 'Guanabara';
Cidade := 'Campinas';
CodigoPostal := '1000';
end;

// Agora copiamos os dados informados acima para a posição
// 2 e 3 da estrutura TPessoa.
Pessoas[2] := Pessoas[1];
Pessoas[3] := Pessoas[1];

// E aproveitamos para mudar o Primeiro Nome nos registros de
// posição 2 e 3 da estrutura.
Pessoas[2].PrimeiroNome:= 'Sara';
Pessoas[3].PrimeiroNome:= 'Henrique';

// Agora mostramos os detalhes das pessoas
// contidas na estrutura
for i := 1 to 3 do
with Pessoas[i] do
MostrarPessoa(
[PrimeiroNome,
UltimoNome,
Endereco,
Bairro,
Cidade,
CodigoPostal]);
// Após mostrar todos os registros o programa é encerrado
Application.Terminate;
end;

// Procedure para mostrar um array com os dados das pessoas
// inseridas na estrutura TPessoa.
procedure TFrmRecords.MostrarPessoa(const campos: array of string);
var
i : Integer;
Texto : String;
begin
// Mostra todos os campos começando no registro 0
for i := 0 to Length(campos)-1 do
Texto := Texto + campos[i] + #10#13;
ShowMessage(Texto);

```

end;

end.

O resultado do código acima é:

'João'
'Sakalauska'
'Av. Brasil, 1000'
'Guanabara'
'Campinas'
'1000'

'Sara'
'Sakalauska'
'Av. Brasil, 1000'
'Guanabara'
'Campinas'
'1000'

'Henrique'
'Sakalauska'
'Av. Brasil, 1000'
'Guanabara'
'Campinas'
'1000'

Considerações: Quando se trabalha com estruturas (record) é necessário alguns cuidados como por exemplo limpar todas as variáveis antes de utilizar e limpar após o uso ou finalização do sistema. Isso pode ser resolvido facilmente com um laço de repetição atribuindo valor a cada variável da estrutura, exemplo:

```
// Lembre-se que nossa estrutura tem 3 posições
// São elas 0, 1, 2, não cometa o amadorismo de colocar o 3 no for
for i := 0 to 2 do
begin
Pessoas[i].PrimeiroNome := "";
Pessoas[i].UltimoNome := "";
Pessoas[i].Endereco := "";
Pessoas[i].Bairro := "";
Pessoas[i].Cidade := "";
Pessoas[i].CodigoPostal := "";
end;
```

É isso, boa sorte com o Record.

Ricardo Alves de Souza.