



CRIANDO **COMPONENTES** VISUAIS COM **DELPHI**

CRIE COMPONENTES QUE
VÃO TE AUXILIAR NA CRIAÇÃO
DE LAYOUTS INCRÍVEIS



COMO CRIAR COMPONENTES VISUAIS NO DELPHI

Fala Radizeiros e Radizeiras;

É muito legal quando temos as funcionalidades disponíveis dentro do Delphi, não é verdade?

Ainda mais componentes visuais que nos auxiliam e muito na criação de novas telas para os nossos softwares.

Mas quando você precisa de algo para ter um visual mais agradável, ou até mesmo uma funcionalidade que você viu em uma plataforma ou software e quer adicionar no seu software mas não tem no Delphi, isso acaba fazendo com que você busque em tudo que é vídeos e componentes para instalar no seu Delphi para resolver isso, mas mesmo assim não consegue solucionar o que está precisando.

Por que não desenvolver o seu próprio componente visual?

Isso mesmo, você viu aquela tela, ou uma imagem de um sistema e gostaria de colocar no seu software e não encontra compatibilidades ou prontos para o ele, então vamos desenvolver o nosso próprio.

Neste e-book, irei mostrar como criar componentes visuais que irão te auxiliar na criação de layouts incríveis para encantar seus clientes.

Mas como muitos desenvolvedores sabem, desenvolver componentes no Delphi é muito chato e difícil quando seguimos todo um passo a passo, trabalhar com TGraphics, WinControl, dentre outros.

Não é simples você desenvolver um componente visual.

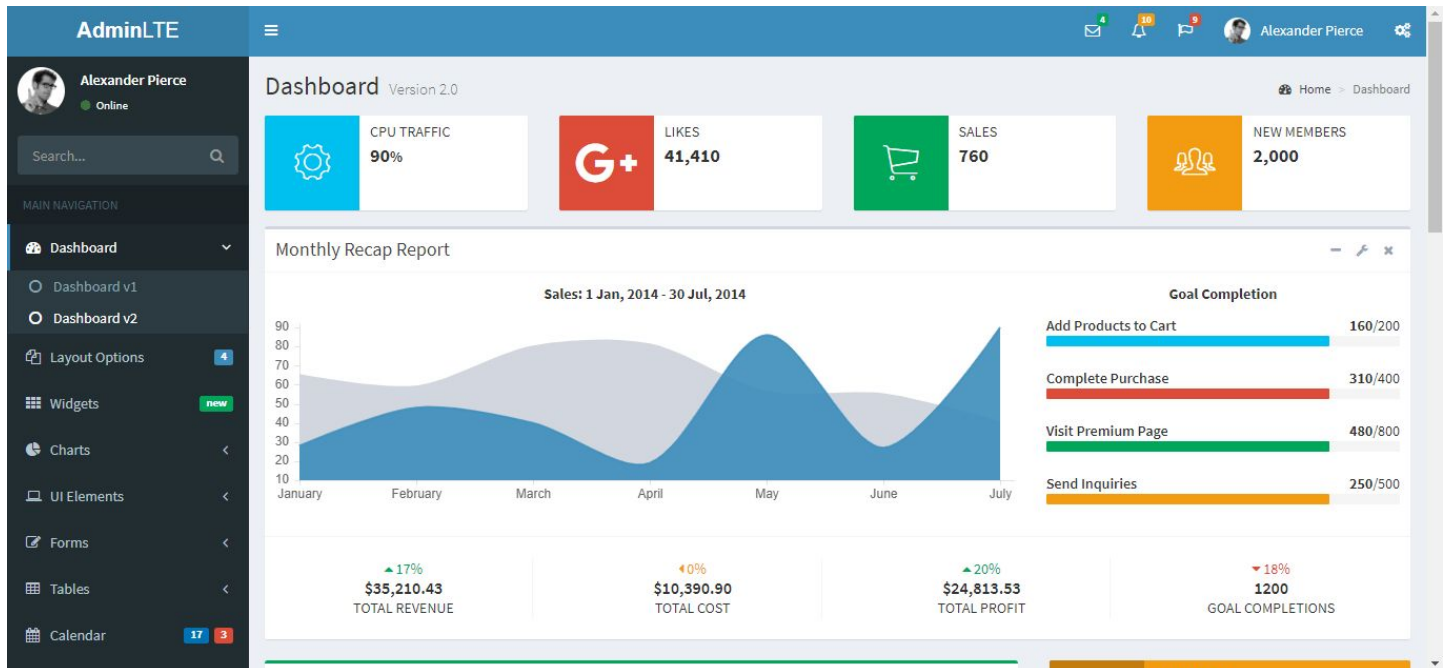
Mas muitas das vezes você quer um nível de personalização no seu software que não é possível fazer com os componentes que temos dentro do Delphi.

Mas todas as vezes que você precisa reutilizar isso , se torna complexo.

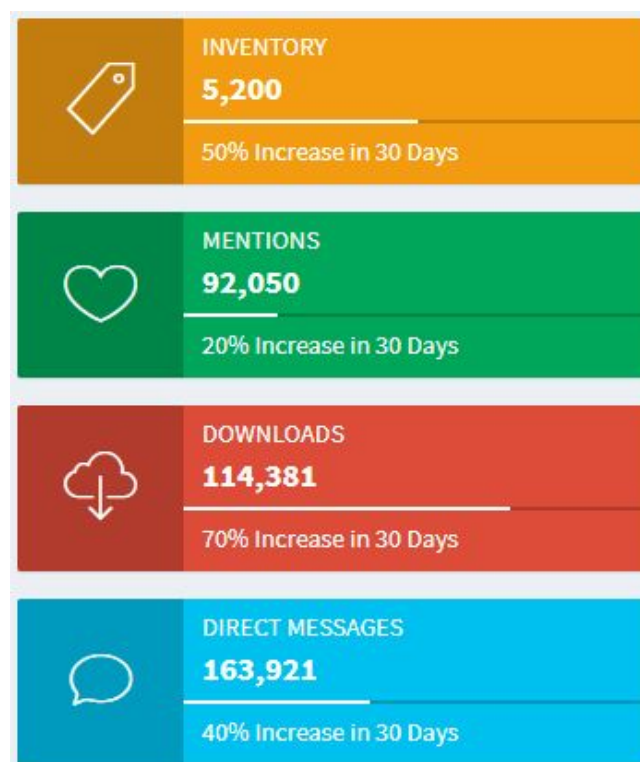
Então o que iremos fazer no decorrer deste e-book, é criar alguns componentes visuais que possamos reutilizar em nossos softwares.

Por exemplo, tem um site onde mostra um Dashboard que foi muito usado, e ainda é por muitos.

É o adminLTE <https://adminlte.io/themes/AdminLTE/index2.html>



Onde iremos reproduzir um desses cards que encontramos nesse template.



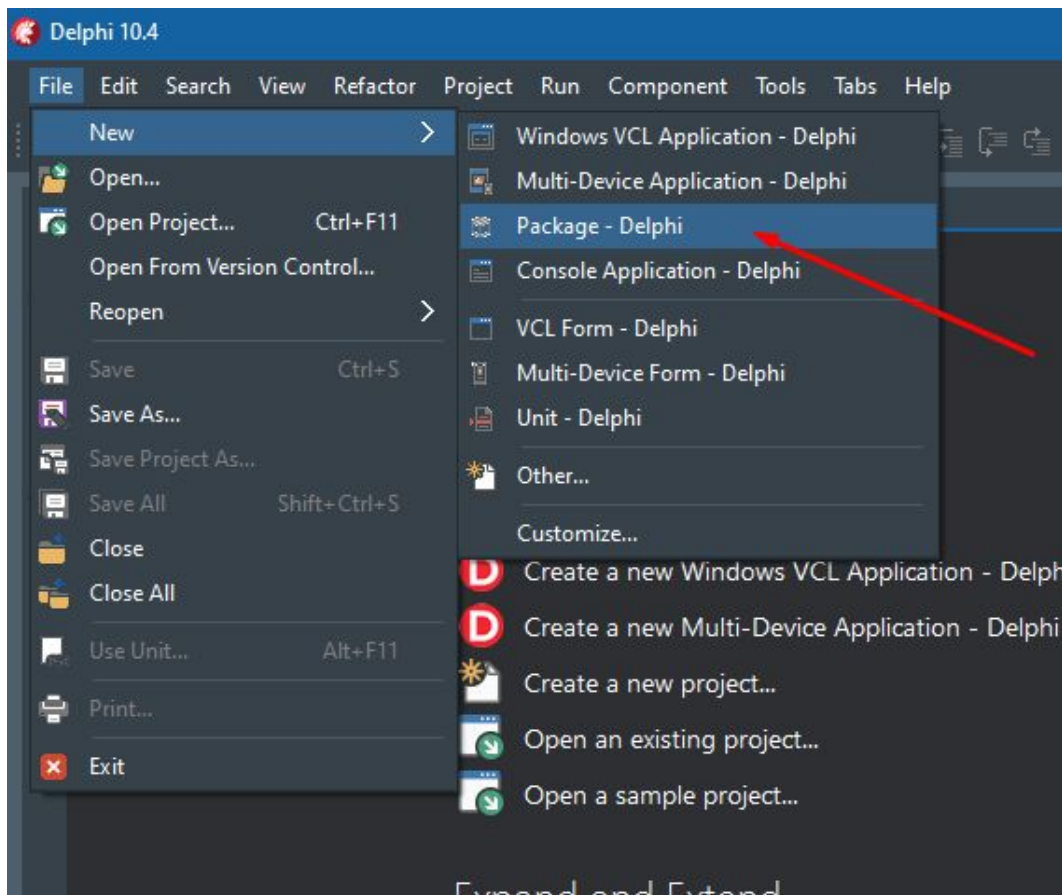
Iremos reproduzir esses cards dentro do Delphi.

E você deve estar se perguntando, como é que eu faço isso, como é que crio isso dentro do Delphi?

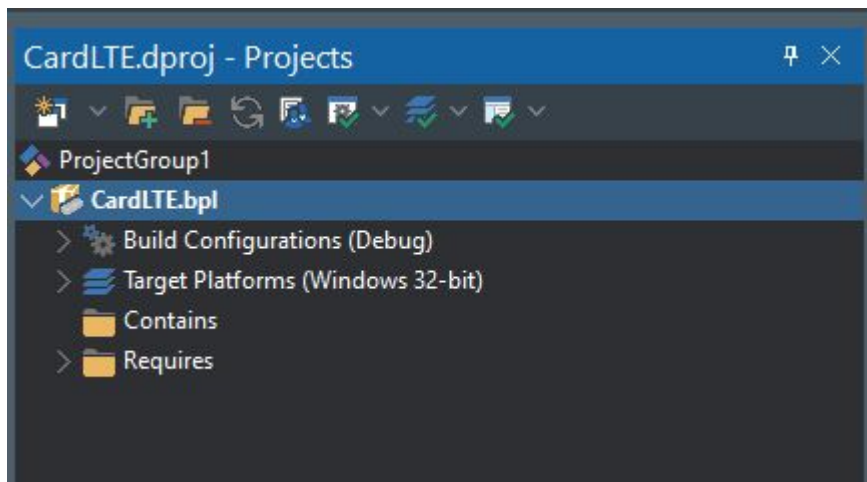
É bem simples, porque quando vamos para o estudo da criação de componentes, acabamos indo para muitas coisas que só servem mesmo a nível de conhecimento.

No dia a dia isso não é prático, e para o nosso dia a dia precisamos de mais praticidade.

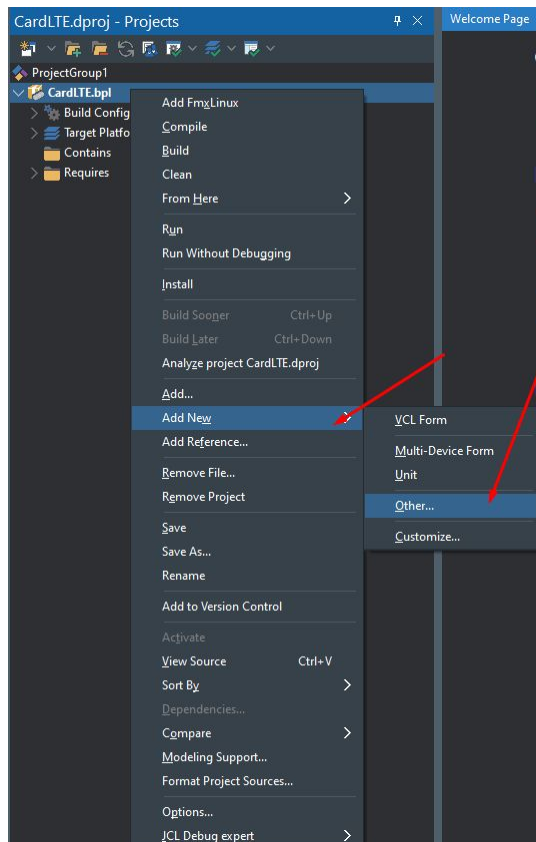
O que precisamos primeiro é criar um package, bastando ir no menu do Delphi e criar o pacote.

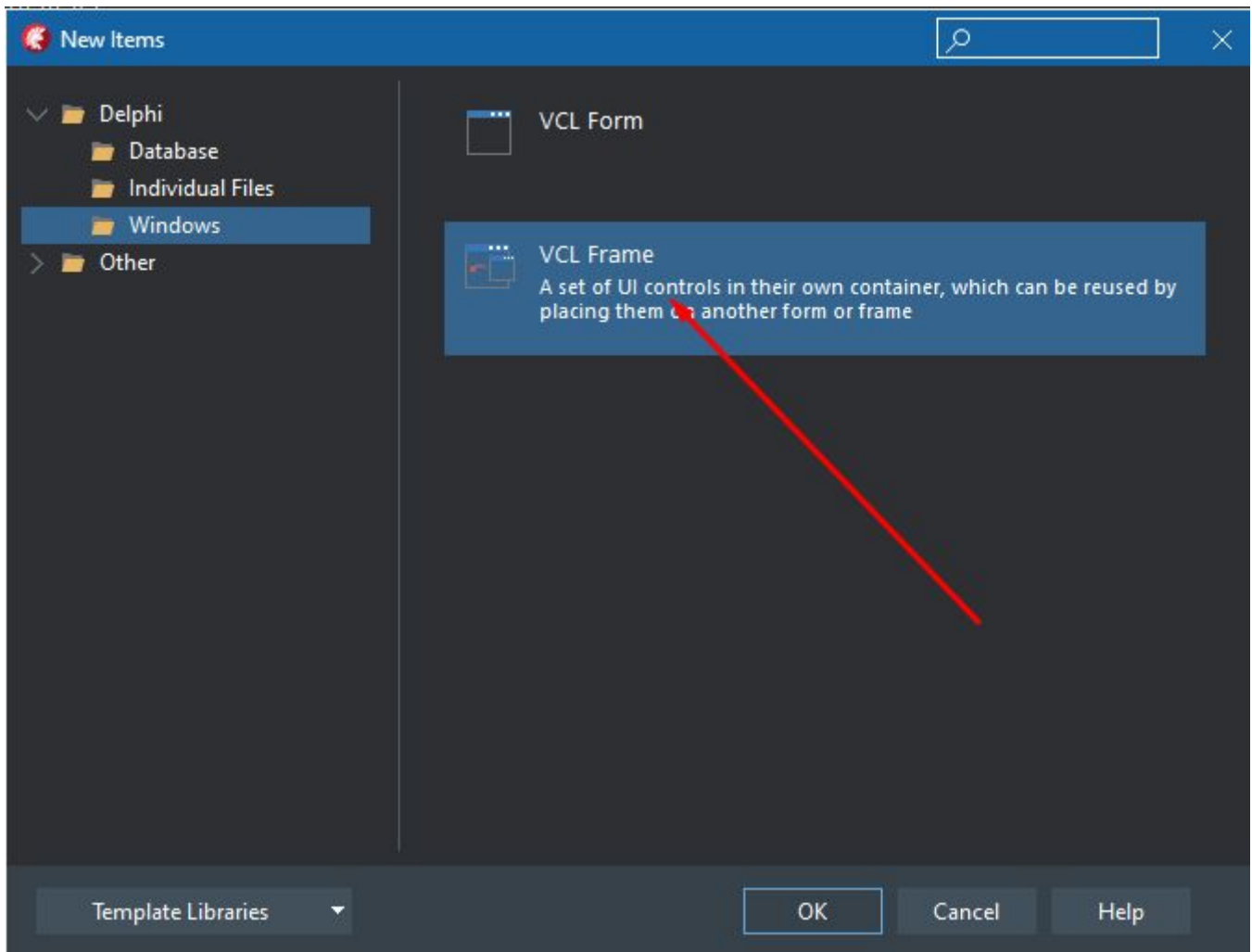


Simplesmente criei esse pacote e dei um nome para ele, *CadLTE*



Com esse nosso pacote criado, ao invés de utilizar Wincontrol e dentre outras classes, iremos simplesmente criar um frame.





Simplesmente eu clico com o botão direito do mouse no nosso pacote e mando adicionar um frame a ele.

Não esquecendo que estamos criando esse componente para VCL e não FMX, mas pode seguir os mesmo procedimento, só realizar algumas alterações.

Iremos criar nosso componente visual utilizando outros componentes do próprio Delphi.

Esse card que mostrei para vocês é muito simples para criar eles utilizando outros componentes nativos do Delphi.

Para que possamos seguir você precisa entender que devemos trabalhar a separação do layout utilizando containers, e nesse caso iremos utilizar panels, para separar as camadas.

Iremos adicionar um panel `alClient`, ou seja, ocupando todo o nosso frame, onde ele será o nosso container principal, o que irá compor todo o corpo do componente.

Para que possamos criar layouts bonitos e organizados, iremos utilizar muito esses panels.

A utilização dos panels fará toda a diferença.

Muitas pessoas se perdem nisso, não conseguem trabalhar layouts responsivos, porque pecam na hora de fazer as organizações dos layouts.

É muito legal você visualizar os frames, os quadros do seu layout.

Se pegarmos o nosso layout do AdminLTE, já podemos ver que no card ele já possui um layout principal, que seria o quadrado dele todo.

E aí nós temos algumas separações.

Temos um *panel* alinhado à esquerda, e um outro *panel* alinhado *alCliente*, que irá compor as informações e o progresso, como você pode revê-lo na imagem logo abaixo.



Seguindo todas essas informações você já consegue criar um novo layout, ou até mesmo componentes para serem utilizados na sua aplicação.

Com essas técnicas você tem a possibilidade de ter um layout responsivo e de fácil utilização.

E caso você tenha interesse de conhecer mais sobre Aplicar Design Criativo no Delphi acesse o nosso portal do CLUBE DE PROGRAMADORES EM DELPHI, onde você não terá só conteúdos relacionados ao Design , mas uma quantidade enorme de conteúdos que poderá lhe ajudar muito no seu dia a dia.

CONHEÇA O CLUBE DE PROGRAMADORES DELPHI

O que é o Clube?

Um clube de programadores com amplo conhecimento, tutelados por um dos MVP's de maior destaque no Brasil, compartilhando conhecimento por um valor super justo e acessível.



O Clube de Programadores Delphi tem a missão de fazer parte da sua vida profissional, lhe ajudando e capacitando nos mais diversos temas no mundo Delphi, para levar a sua carreira para um outro patamar.

[CLIQUE E CONHEÇA O CLUBE](#)

O que você encontrará no Clube?

Escolhemos a dedo 8 competências que consideramos essenciais para todo desenvolvedor Delphi



Preparatório para
Certificação Delphi



Componentes



Inovação



Recursos da
Linguagem



Ferramentas de
Terceiros



Firemonkey



Automação
Comercial



Conteúdos
Exclusivo





CRIANDO TEMPLATES DE COMPONENTES

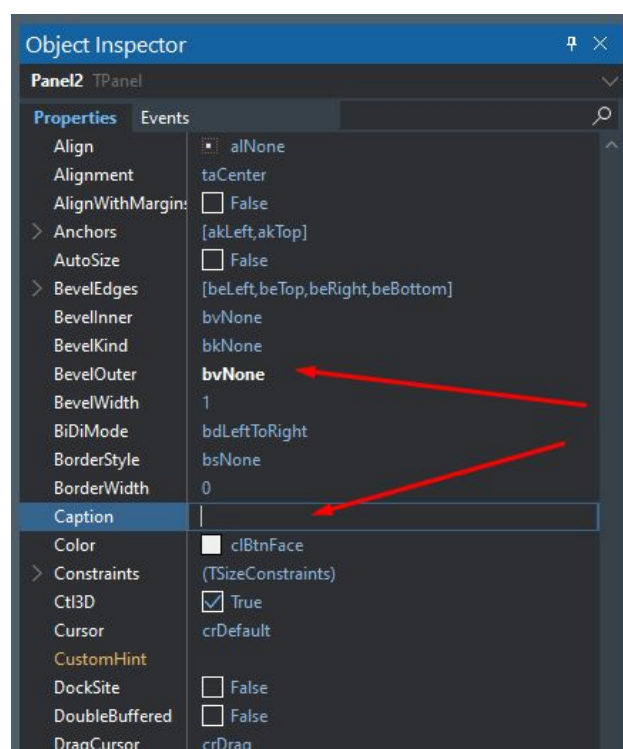
Neste capítulo, iremos de fato, criar nosso componente.

Criar seus próprios componentes lhe dará maior liberdade de prepará-los conforme as suas necessidades.

Nesse caso, com o nosso package aberto iremos começar a adicionar os demais panels para trabalhar o layout do nosso componente.

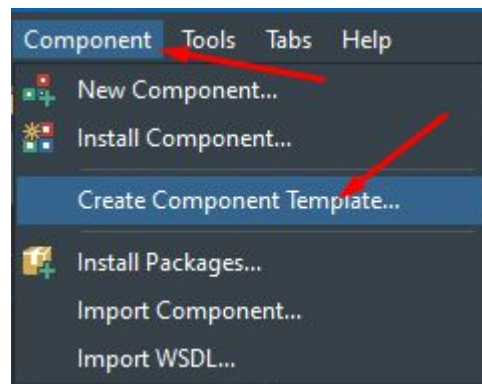
Iremos adicionar mais um panel.

E vai uma dica: todas as vezes que colocamos um panel na tela, temos que fazer algumas alterações, onde temos que limpar o *BevelOuter*, e tirar o *Caption*.

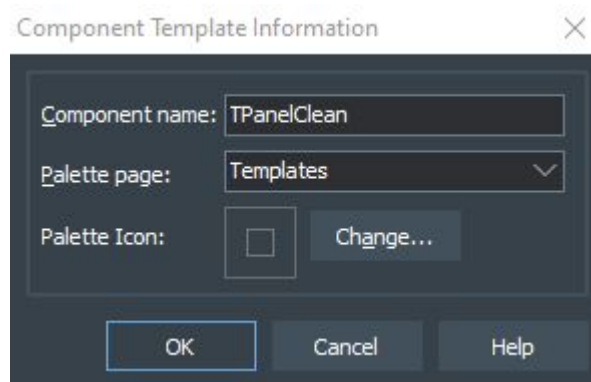


Isso se torna custoso pois iremos trabalhar com ele constantemente.

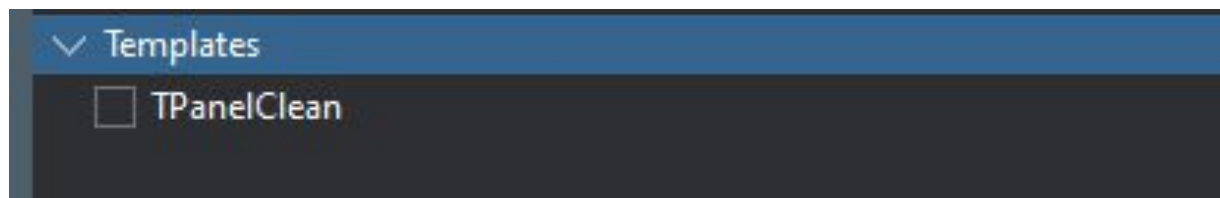
Para isso eu selecionei esse panel alterado, vou no menu *Component* da IDE do Delphi, e escolho a opção *Create Component Template*.



Na tela que irá abrir iremos dar um nome para esse componente e de qual seção irá aparecer esse nosso componente.



Quando procurarmos por panel dentro da nossa paleta de componentes do Delphi, iremos encontrar esse nosso panel criado no Delphi.



Dessa forma já temos o nosso panel clean, do jeito que iremos precisar constantemente para trabalhar nossos layouts.

E você pode estar se perguntando, existe essa funcionalidade em versões mais antigas do Delphi?

Sim, essa funcionalidade está disponível no Delphi há muito tempo.



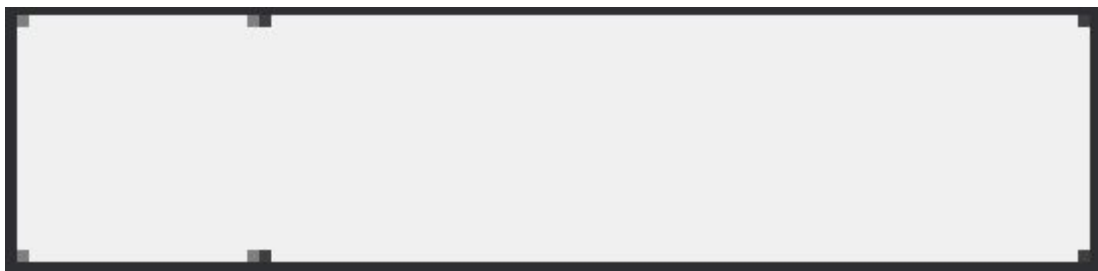
TRABALHANDO COM CORES HEXADECIMAIS NO DELPHI

Quando trabalhamos com layouts, ou até mesmo criação de componentes visuais, devemos nos atentar às cores. Acredite você ou não, essa tem sido uma grande dificuldade na maioria dos desenvolvedores Delphi.

Afinal, quem nunca viu aquelas telas cinza e azul em diversos lugares?

Para lhe ajudar, irei mostrar como aplicar o uso de cores hexadecimais dentro da VCL do Delphi.

Com o nosso componente aberto iremos adicionar um novo panel, aquele painel clean que criamos anteriormente.

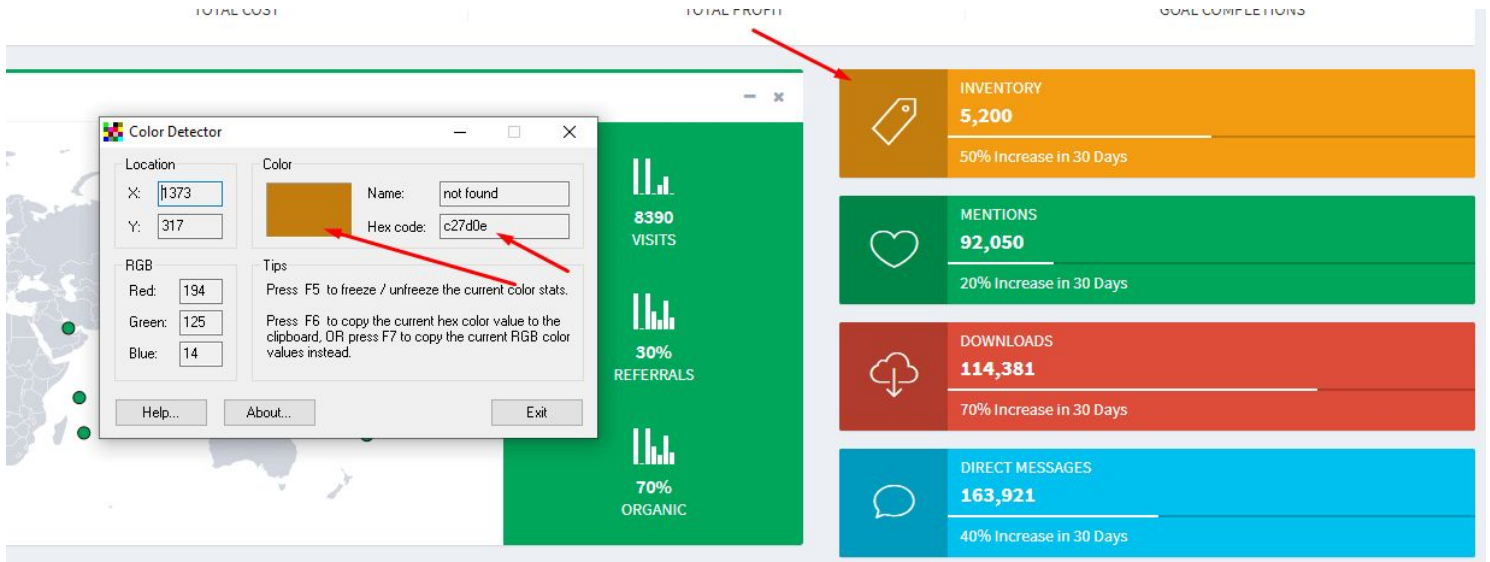


E esse nosso panel estará alinhado à esquerda, e um outro alclient, como o nosso card no AdminLTE.

E para auxiliar no uso das cores, iremos utilizar um programa muito legal para pegar os valores hexadecimais,, o *ColorDetector*:

<https://color-detector.informer.com/2.0/>

Com ele basta colocar o ponteiro do mouse sobre a cor que deseja pegar o valor hexadecimal e apertar o botão F6 do teclado.



Com a cor já transferida para a área de transferência do Windows, isso aconteceu por causa da ação do botão F6 do color detector.

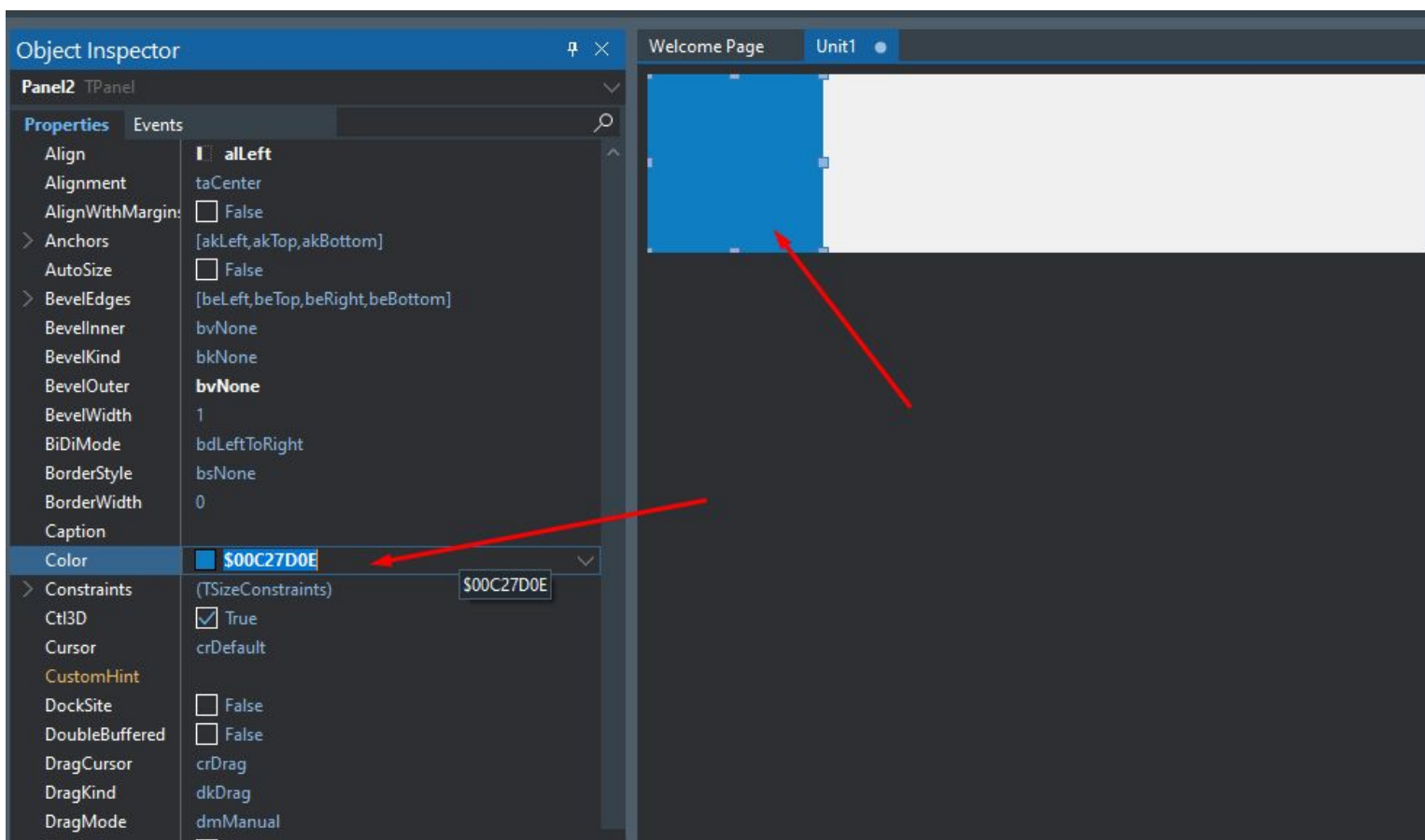
Basta irmos no Delphi, e no *panel* que iremos alterar a cor e adicionar esse hexadecimal.

E uma dica muito valiosa, quem não fez o DDE desconhece essa funcionalidade.

Se colocarmos diretamente o valor que o color detecto nos disponibiliza na propriedade color do *panel*, ele não irá assumir a sua real cor.

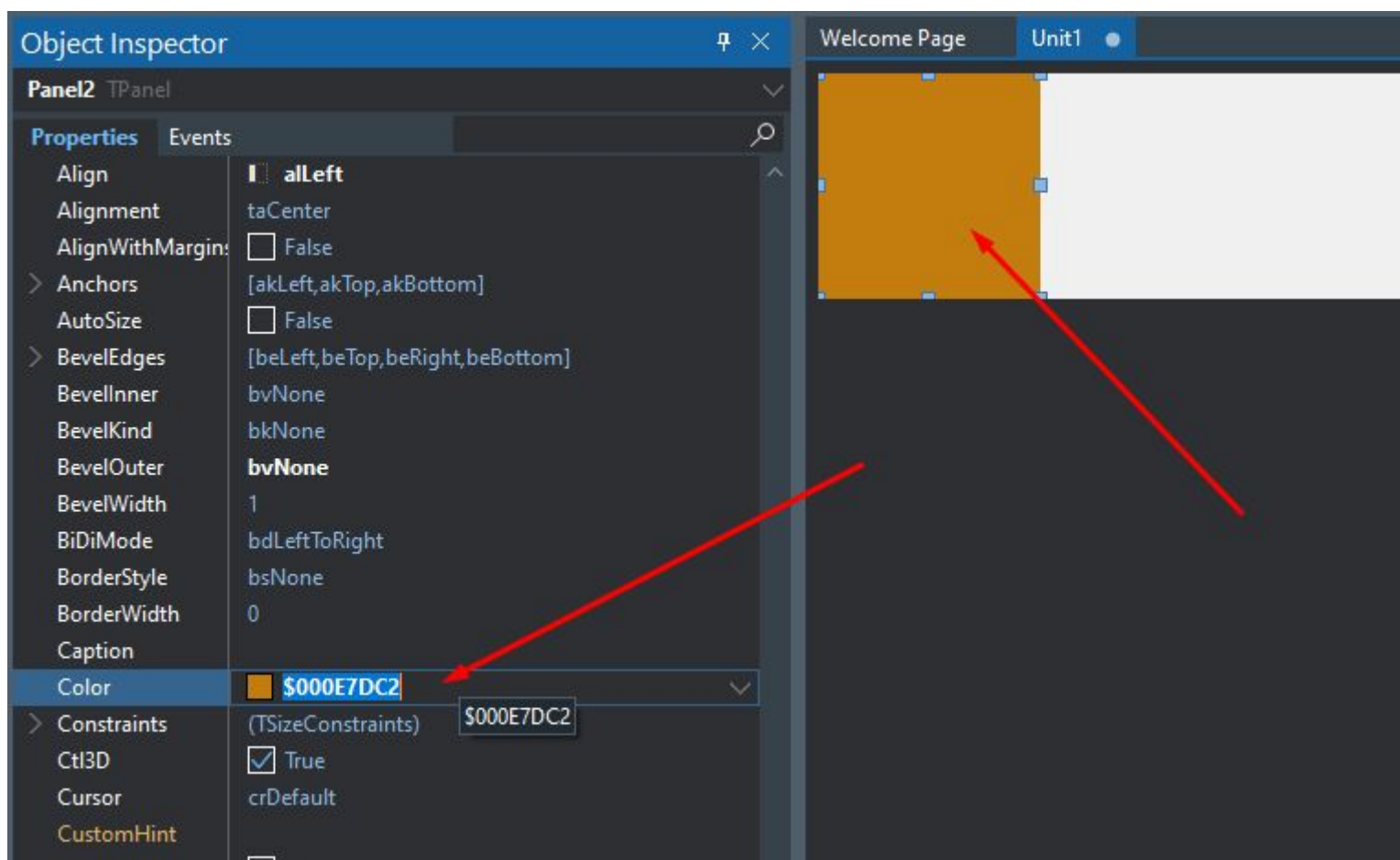
Isso ocorre porque o Delphi entende o RGB de forma contrária, ou seja, ele entende como BGR.

Por exemplo, observe se adicionarmos o valor direto e como a cor irá se apresentar.



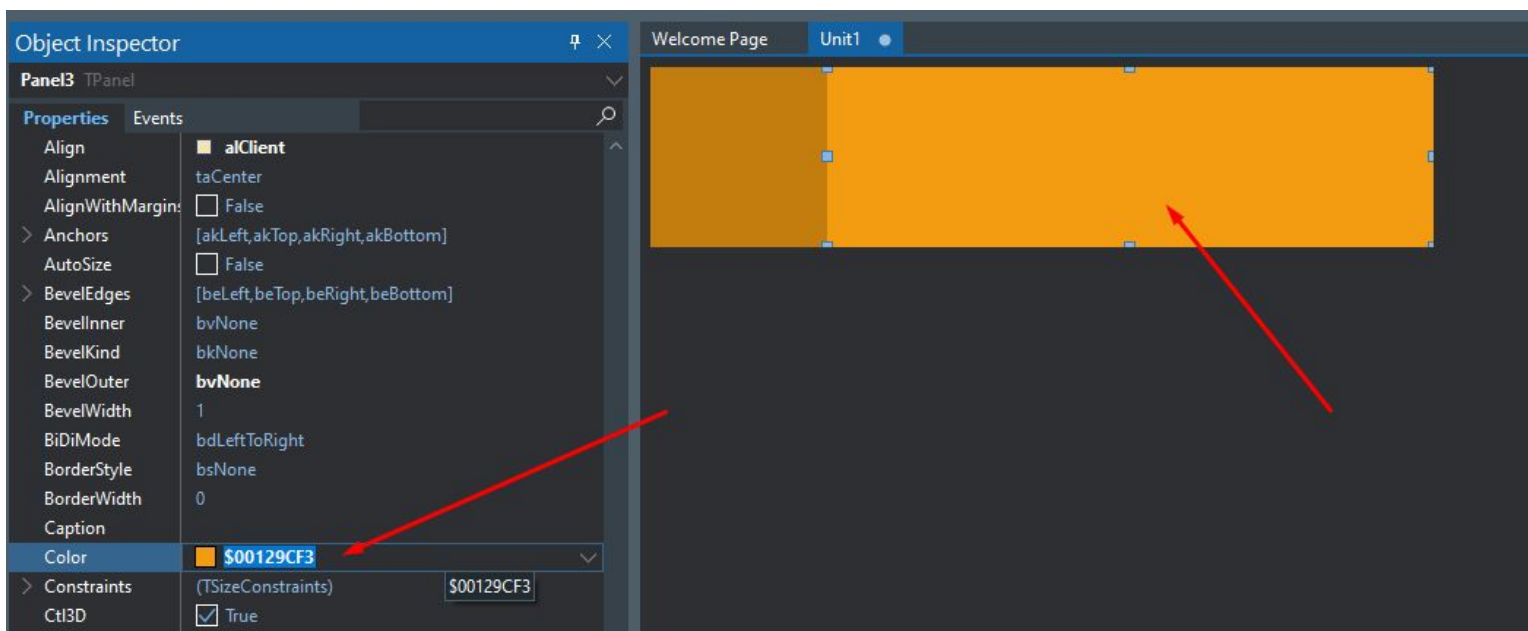
Agora se alteramos as posições seguindo o padrão BGR, onde os dois primeiros números vão para os dois últimos e os dois últimos para o lugar dos dois primeiros números.

Lembrando que para funcionar a cor dentro do VCL é preciso adicionar no início \$00.



Viu como agora as cores ficaram corretas?

Agora já vamos alterar também a cor do outro panel para que possa ficar bem parecido com o nosso card do AdminLTE.



Repare como pequenos detalhes fazem uma grande diferença.

Conteúdos como esse e muitos outros, você consegue encontrar dentro do **Clube dos programadores Delphi**.

CONHEÇA O CLUBE DE PROGRAMADORES DELPHI

O que é o Clube?

Um clube de programadores com amplo conhecimento, tutelados por um dos MVP's de maior destaque no Brasil, compartilhando conhecimento por um valor super justo e acessível.



O Clube de Programadores Delphi tem a missão de fazer parte da sua vida profissional, lhe ajudando e capacitando nos mais diversos temas no mundo Delphi, para levar a sua carreira para um outro patamar.

[CLIQUE E CONHEÇA O CLUBE](#)

O que você encontrará no Clube?

Escolhemos a dedo 8 competências que consideramos essenciais para todo desenvolvedor Delphi



Preparatório para
Certificação Delphi



Componentes



Inovação



Recursos da
Linguagem



Ferramentas de
Terceiros



Firemonkey



Automação
Comercial



Conteúdos
Exclusivo





CRIANDO O LAYOUT DO COMPONENTE

Dando continuidade a nossa criação de componentes visuais, neste capítulo iremos criar todo o Layout do nosso componente.

Até o momento, criamos somente uma estrutura simples para termos uma melhor base de como irá compor o nosso componente.

Então o que iremos ter agora?

Iremos adicionar os textos, onde no card de exemplo do nosso Dashboard tem um *Inventory*, e demais informações.

Então vamos adicioná-los.



E uma dica legal, se você quer criar layouts bonitos, a primeira coisa que deve ser feita é copiar.

Existem diversos layouts na internet que você pode utilizar como inspiração e começar a copiar, para aprender a ter layouts mais bonitos.

É copiando que vai aprimorando as técnicas para desenvolver layouts bonitos.

Agora como iremos adicionar essa barra de progresso?

Lembra que eu comentei que devemos utilizar componentes nativos e que possam se aproximar ao máximo do que é proposto no layout?

Então iremos adicionar um shape, isso mesmo iremos desenhar o progresso deste nosso layout.



Observe que coloquei um shape ao fundo e um outro acima dele, só mudei as cores, e claro seguindo o nosso layout.

Agora para que possamos ter aquela image do lado esquerdo do nosso card, iremos adicionar um image, isso mesmo, o componente TImage do Delphi.

Só alterei a propriedade Padding do TImage, para que a imagem não fique “agarrada” nos cantos.

Desta forma, ele ficará sempre responsivo sem que a imagem perca o espaçamento nos cantos.

E a nossa barra de progresso, eu ancore ele tanto a esquerda quanto a direita, para ele se redimensionar junto com o componente.



Pode-se observar que o componente agora está tomando forma, e ficando muito mais interessante.



REGISTRANDO O COMPONENTE VISUAL

Até agora criamos somente os layouts do nosso componente visual, mas agora iremos para a programação em si.

A primeira coisa que iremos fazer, é trabalhar com esse componente que acabamos de criar. Para isso, precisamos registrá-lo como um componente.

Devemos ir no código do nosso componente, antes do *implementation* adicionar uma procedure chamada *Register*, que será a procedure que irá registrar esse nosso frame como componente.

```
1  ...
2  procedure Register;
3  implementation
4  procedure Register;
5  begin
6  RegisterComponents('LTE Components', [TLTECard]);
7  end;
```

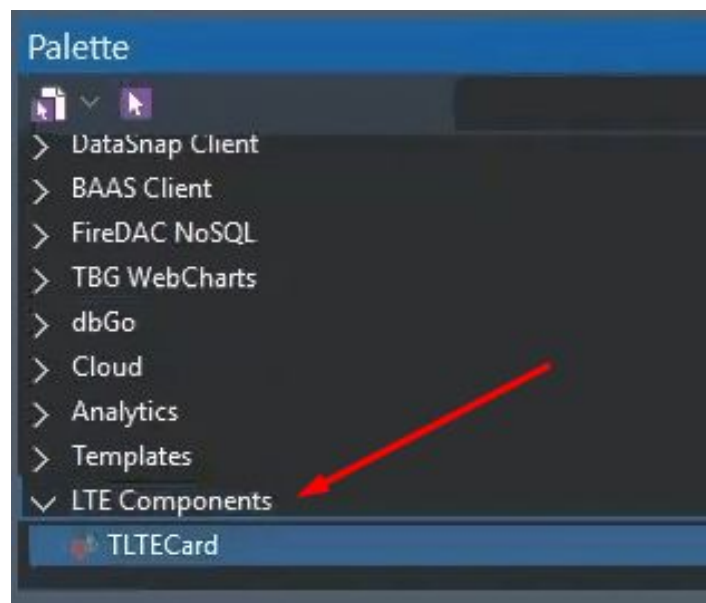
Observe que na implementação da nossa procedure *Register* chamamos o método *RegisterComponents*, onde dentro dele não definimos em qual seção da paleta de componentes ele será registrado.

Deixei essa nomenclatura de *LTE Components*, vamos dizer que eu queria criar todos os componentes do *Adminlte*, e criar eles no Delphi.

Isso é completamente possível, podemos criar um por um, só um trabalhinho de desenhar eles no Delphi, mas é totalmente possível.

E logo em seguida eu passei o componente, onde entre colchete eu passei o nome da classe desse nosso componente.

Só fazendo isso o nosso componente já encontra-se registrado e presente na paleta de componentes do Delphi.



Muito legal, não é pessoal?

Agora temos um componente visual personalizado para a nossa necessidade.

Basta criarmos um projeto VCL e adicionar esse novo componente.



Agora já temos um componente publicado e pronto para uso na IDE do Delphi.



CRIANDO AS PROPRIEDADES DO COMPONENTE

Lendo este e-book sobre a **Criação de Componentes Visuais**, você deve estar se perguntando como pode trabalhar dentro desse componente.

Até o momento só criamos todo o layout desse nosso componente, mas e agora?

Bem, agora precisamos alterar os textos e a progressão que desenhamos no componente, e é aí que iremos colocar a mão no código.

Mas precisamos ter uma observação.

Quando criamos um componente utilizando frame ele vem com as propriedades do frame.

Com isso eu não irei conseguir mexer nos textos, na barra de progresso, na imagem, não consigo mexer em nada.

Quando eu adiciono o componente em uma formulário e mexo ele em tempo de design, eu estou mexendo nele todo, ou seja, em todo o frame.

Ele está todo engessado, não consigo mexer em nada.

Para que possamos mexer no nosso componente iremos ter que criar propriedades, ou seja, *propertys*.

Essas *propertys* são necessárias para que possamos mexer nos atributos do nosso componente.

Então vamos em nosso componente e criar essas propriedades.

A primeira property a ser criada é do texto superior.



Esse nosso label está nomeado como `lbTextTitle`, é o label de títulos.

E dentro da classe do nosso componente iremos adicionar a sessão *published*.

Esta sessão *published* é responsável por tudo que aparece no *Object Inspector*.

Então se você quer criar uma nova propriedade para que apareça o *Object Inspector*, tem que criar uma *property* na sessão *published*.

```
1  private
2  FTextTitle : String;
3  procedure SetTextTitle(const Value : String);
4  ...
5  published
6  property TextTitle : String read FTextTitle write SetTextTitle;
7  end;
8  ...
9  procedure TLTECard.SetTextTitle(const Value: String);
10 begin
11 FTextTitle := Value;
12 end;
```

Observe que já criamos a propriedade para o texto do título.

Já foi criado a variável privada, que será trabalhada dentro da classe, e o método que recebe o valor dessa propriedade.

Esta sessão *published* é responsável por tudo que aparece no *Object Inspector*.

Então se você quer criar uma nova propriedade para que apareça o *Object Inspector*, tem que criar uma *property* na sessão *published*.

```
1  private
2  FTextTitle : String;
3  procedure SetTextTitle(const Value : String);
4  ...
5  published
6  property TextTitle : String read FTextTitle write SetTextTitle;
7  end;
8  ...
9  procedure TLTECard.SetTextTitle(const Value: String);
10 begin
11 FTextTitle := Value;
12 end;
```

Observe que já criamos a propriedade para o texto do título.

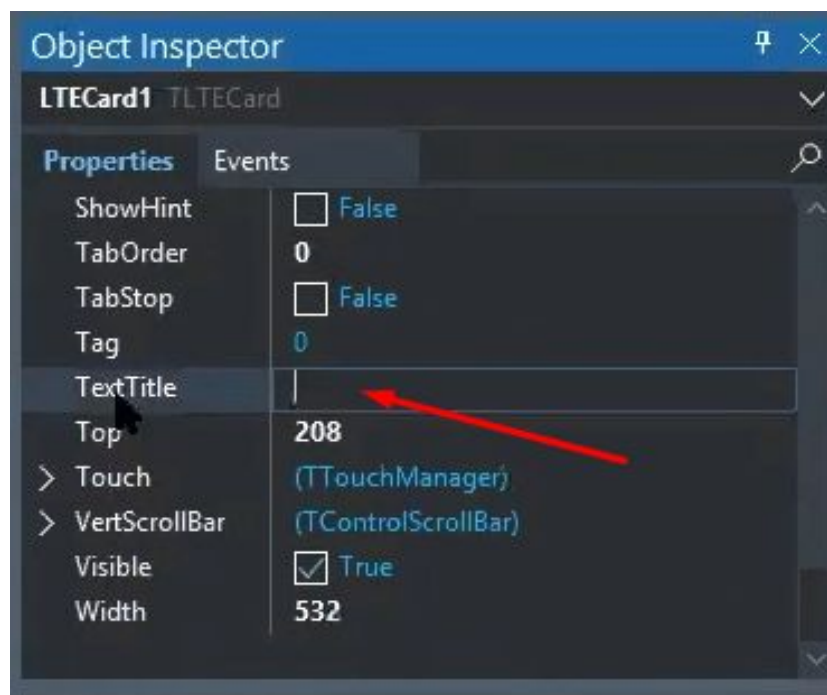
Já foi criado a variável privada, que será trabalhada dentro da classe, e o método que recebe o valor dessa propriedade.

E para que apareça no texto o que foi passado para essa property basta adicionarmos dentro do método `SetTextTitle` o label do título recebendo esse valor passado.

```
1 procedure TLTECard.SetTextTitle(const Value: String);  
2 begin  
3   FTextTitle := Value;  
4   lbTextTitle.Caption := FTextTitle;  
5 end;
```

Ao passar um valor para essa propriedade, ele irá receber esse valor passar para a variável e atribuir o valor dessa variável para o label de título.

Desta forma compilamos novamente esse nosso componente, e pronto, a propriedade está lá no *Object Inspector*.



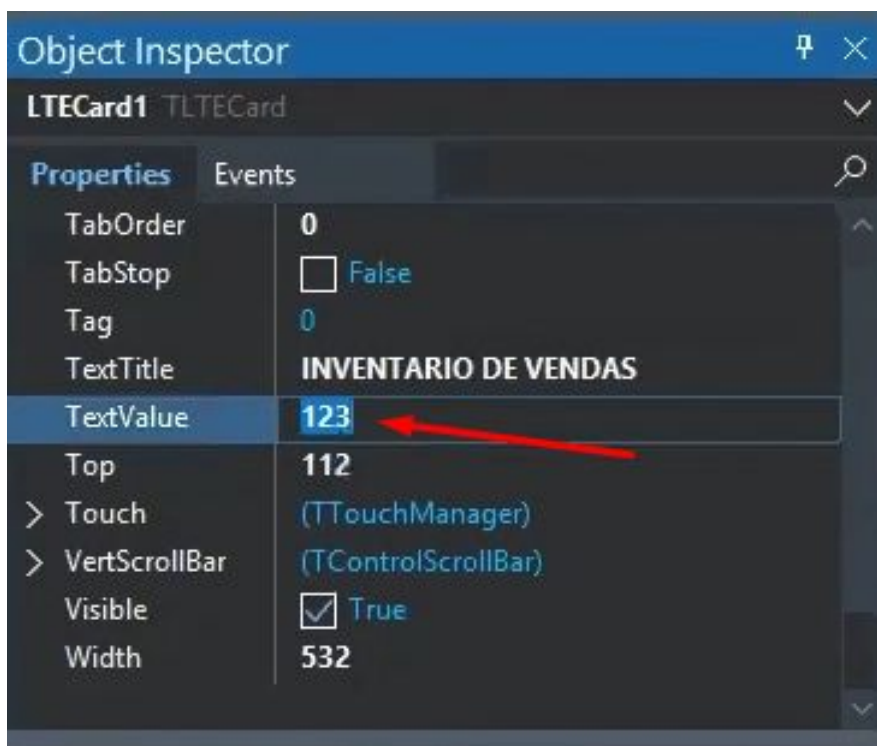
E agora ao adicionarmos qualquer valor nessa propriedade, como por exemplo, *INVENTARIO DE VENDAS*, iremos ver a alteração dentro do texto do nosso componente.



Agora iremos criar outras propriedades para esse nosso componente, onde iremos criar um *TextValue* do tipo *currency*, para o texto monetário que possuímos dentro desse nosso card.

```
1  private
2  FTextValue : Currency;
3  procedure SetTextValue(const Value : Currency);
4  published
5  ...
6  property TextValue : Currency read FTextValue write SetTextValue;
7  ...
8  procedure TLTECard.SetTextValue(const Value : Currency);
9  begin
10 FTextValue := Value;
11 lbTextValue.Caption := FormatCurr('#,##0.00', FTextValue);
12 end;
```

Dessa forma temos também uma outra propriedade dentro do *Object Inspector*.



E após atribuirmos um valor para essa nova propriedade, veremos que no visual do componente é alterado também.

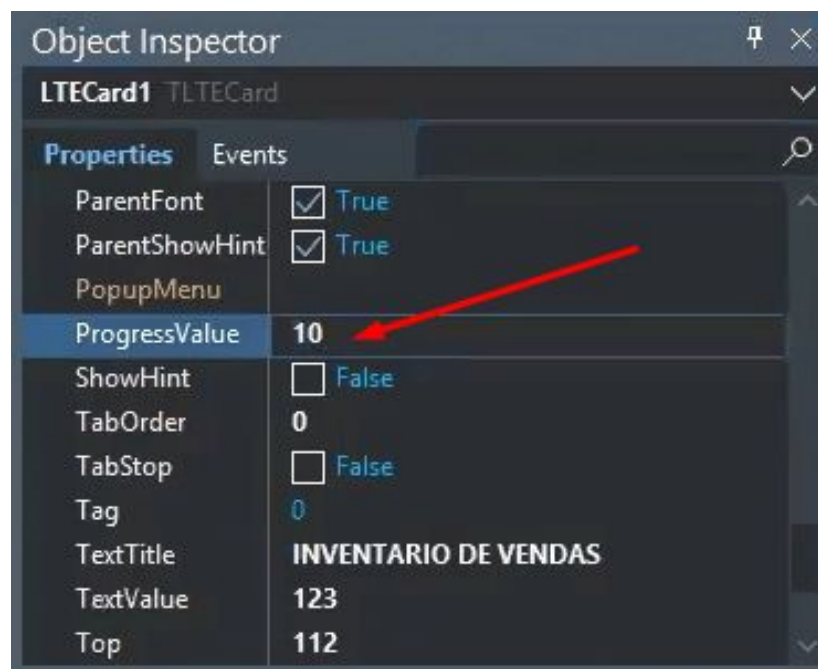


E agora vamos criar uma outra propriedade chamada *ProgressValue*, para que possamos mexer na nossa barra de progresso.

```
private
FProgressValue : Integer;
procedure SetProgressValue(const Value : Integer);
published
...
property ProgressValue : Integer read FProgressValue write SetProgressValue;
...
procedure TLTECard.SetProgressValue(const Value : Currency);
begin
FProgressValue := Value;
Shape2.Width := Round(Value * (Shape1.Width / 100));
end;
```

Observe que nessa nossa barra, pegamos o shape que está sobre o shape escuro, pois é ele que irá se mover, ou seja, terá uma progressão.

E para isso eu tive que realizar um cálculo, onde eu pego o tamanho do shape principal, que é o shape1, e cálculo para que ele possa chegar a 100%.



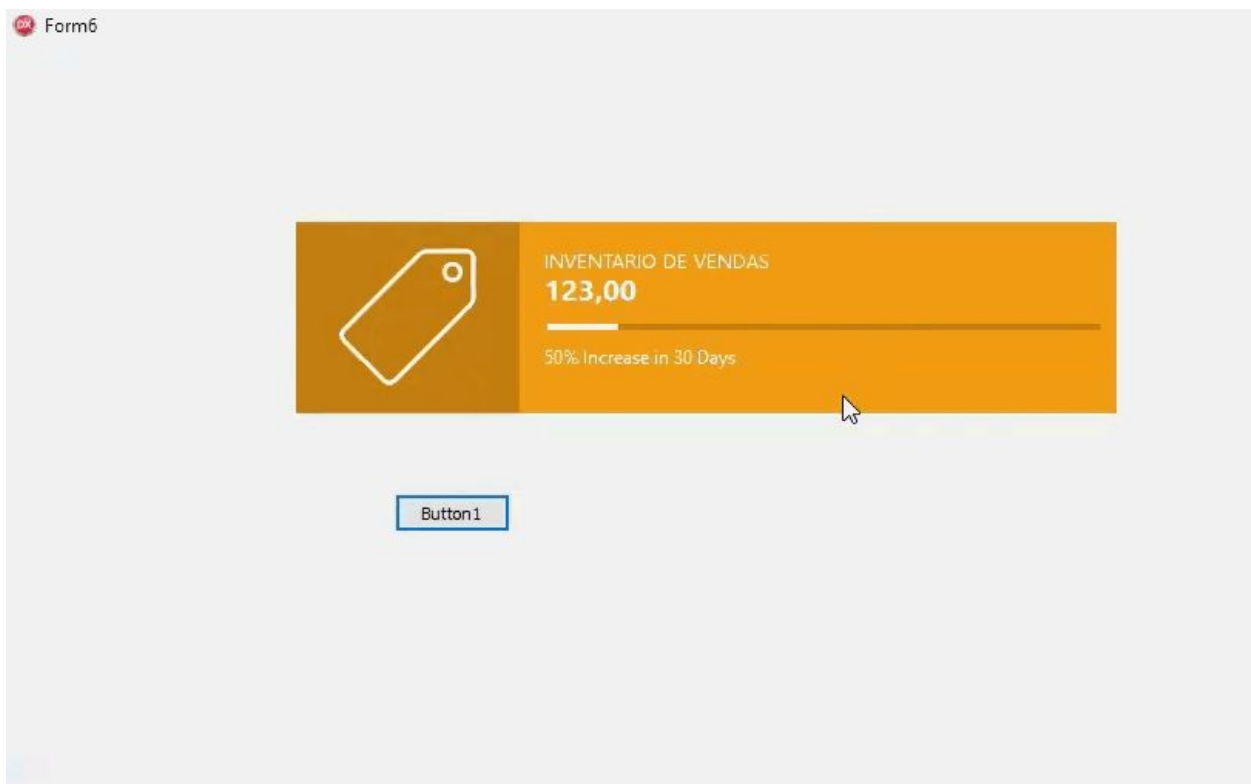
E a cada alteração que formos fazendo o nosso progress altera de tamanho, onde agora ele está com 10% do seu tamanho principal.



E para ficar melhor essa visualização, irei adicionar um botão na tela e programar essa progressão.

```
1  procedure TForm1.ButtonClick(Sender: TObject);  
2  var  
3  i:integer;  
4  begin  
5  for i := 0 to 100 do  
6  begin  
7  LTECard1.ProgressValue := I;  
8  Application.ProgressMessages;  
9  sleep(200);  
10 end;  
11 end;
```

Com isso teremos uma visualização melhor de como o nosso progress está sendo processado.



De uma forma simples já temos um componente visual bonito, e com ações.

Agora você pode criar seus próprios componentes visuais e não ficar tendo retrabalho.

O componente template do Delphi só cria a casca para você, ele não guarda código. Para que você possa guardar código, tem que fazer isso o que fizemos aqui neste e-book.

Nós criamos um componente através de um frame, daí você cria as propriedades que precisar para trabalhar esses seus componentes.

Muito legal, não é?

Está aí, uma dica para que você possa criar seus componentes visuais e personalizar sua aplicação.



E caso você tenha interesse de conhecer mais sobre Componentes Visuais acesse o nosso portal do **CLUBE DE PROGRAMADORES EM DELPHI**, onde você não terá só conteúdos relacionados ao Design , mas uma quantidade enorme de conteúdos que poderá lhe ajudar muito no seu dia a dia, é uma verdadeira NETFLIX para os programadores Delphi.

CONHEÇA O CLUBE DE PROGRAMADORES DELPHI

O que é o Clube?

Um clube de programadores com amplo conhecimento, tutelados por um dos MVP's de maior destaque no Brasil, compartilhando conhecimento por um valor super justo e acessível.



O Clube de Programadores Delphi tem a missão de fazer parte da sua vida profissional, lhe ajudando e capacitando nos mais diversos temas no mundo Delphi, para levar a sua carreira para um outro patamar.

[CLIQUE E CONHEÇA O CLUBE](#)

O que você encontrará no Clube?

Escolhemos a dedo 8 competências que consideramos essenciais para todo desenvolvedor Delphi



Preparatório para
Certificação Delphi



Componentes



Inovação



Recursos da
Linguagem



Ferramentas de
Terceiros



Firemonkey



Automação
Comercial



Conteúdos
Exclusivo

