

# BIBLIOTECA LIBVAGAS



## EXEMPLO DE USO

Tutorial de utilização da Biblioteca LibVagas para comunicação com painéis de chamada Wyma no sistema operacional Windows.

Rev.0 – 19/09/2018 – Revisão inicial

# Índice

1. Introdução .....	2
2. Requisitos para iniciar .....	2
3. Instalando .....	3
4. Software aplicativo exemplo .....	6
5. Biblioteca LibVagas .....	7
6. Código exemplo em C#.NET .....	8
7. Registrando o arquivo typeLib (tlb), no Windows.....	9
8. Criando os componentes no Delphi .....	11
9. Criando um projeto no Delphi.....	12

# 1. Introdução

A Biblioteca de Comunicação LibVagas é um arquivo de extensão de aplicativo (DLL) com diversas funções prontas, que embutem e simplificam a implementação de um protocolo de comunicação.

A biblioteca foi concebida usando a tecnologia Microsoft .NET 4, portanto, para sua utilização será necessário possuir no mínimo a mesma versão, além das ferramentas de desenvolvimento. O framework pode ser adquirido no link abaixo:

<https://www.microsoft.com/pt-br/download/details.aspx?id=24872>

Para este projeto, utilizou-se o Visual Studio 2017, que pode ser instalado gratuitamente a partir do site da Microsoft, disponível no link:

<http://www.visualstudio.com/downloads/download-visual-studio-vs>

Selecione uma versão com linguagem C#.

# 2. Requisitos para iniciar

Antes de iniciar o projeto, verifique se recebeu todos os arquivos necessários. Você deve ter os seguintes arquivos:

**1-setup.exe** = Instalador dos arquivos de código em C#.NET para testar a comunicação, através de aplicativo exemplo e arquivos de biblioteca “DLL” e “TLB”.

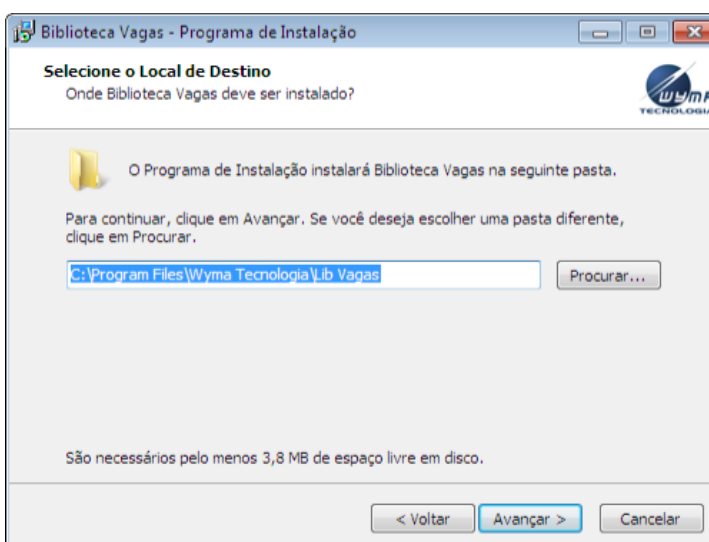
Também é necessário que o painel esteja ligado, e conectado à rede. Também deve ter seu endereço IP configurado (consulte o manual de instalação do painel).

### 3. Instalando

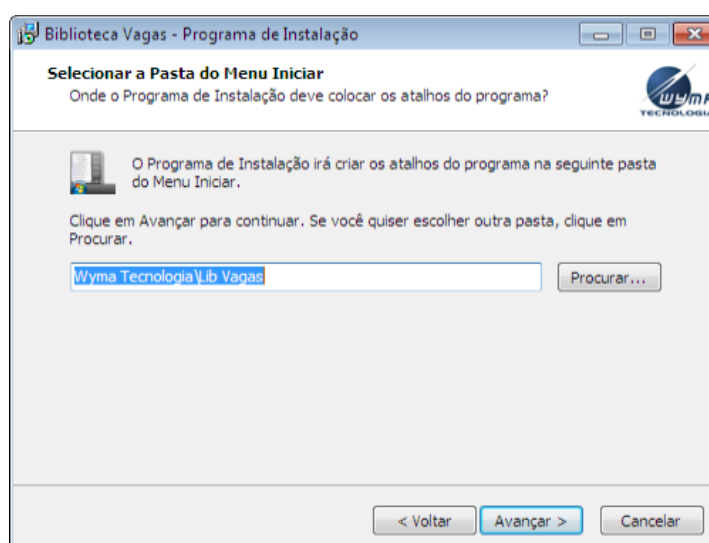
Localize o arquivo “**setup.exe**” e com dois cliques abra o instalador. Clique no botão “**Avançar**” para iniciar a instalação.



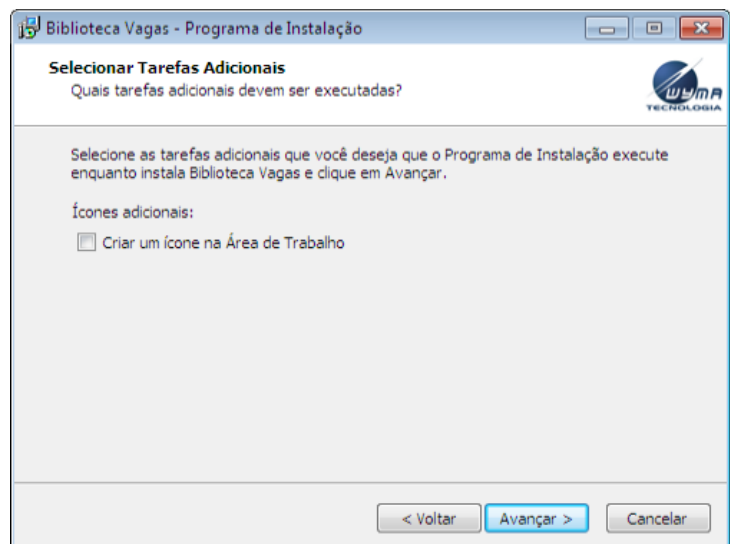
Sugerimos deixar a mesma pasta de instalação que será usada como referência neste documento.



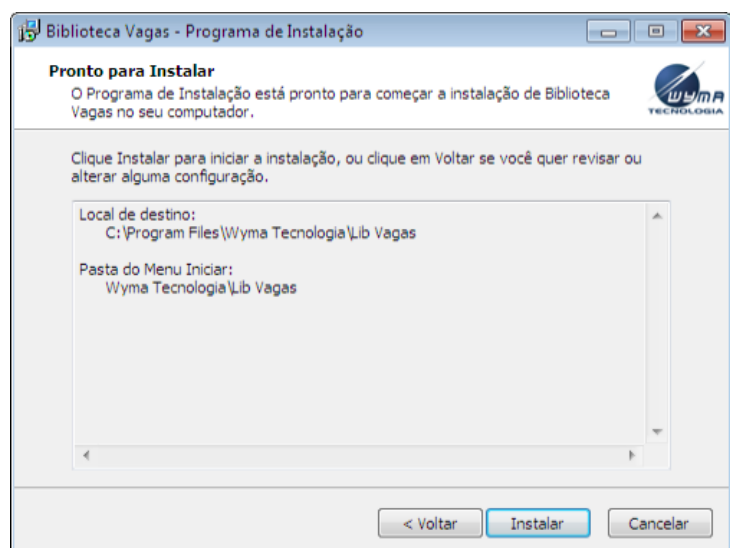
Esta é a pasta de atalhos para o aplicativo de teste.



Se desejar criar um atalho na área de trabalho marque a opção ao lado.



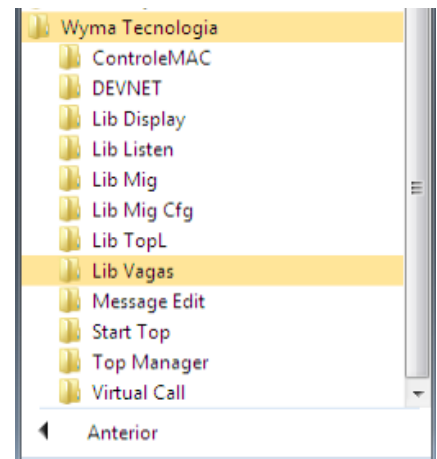
Este é o resumo das opções escolhidas para a instalação, clique em instalar para finalizar.



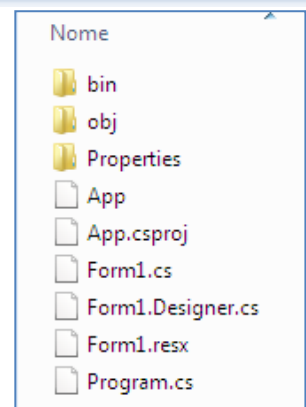
Marque a opção “Executar” para iniciar o aplicativo assim que concluir a instalação.



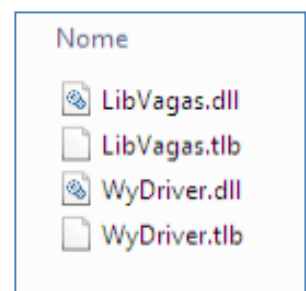
Após a instalação alguns atalhos serão criados dentro da pasta “Wyma Tecnologia” o nome do atalho para o aplicativo será “Lib Vagas”. Note que existe um atalho para a pasta “**Código fonte**” com o código de um projeto exemplo e também um atalho para uma pasta “**Arquivos biblioteca**”, onde estão os arquivos dll e tlb.



Clique no atalho “**Código Fonte**” para abrir a pasta com os arquivos do projeto.



Clique no atalho “**Arquivos biblioteca**” para abrir a pasta com os arquivos “dll” e “tlb”.



## 4. Software Aplicativo exemplo

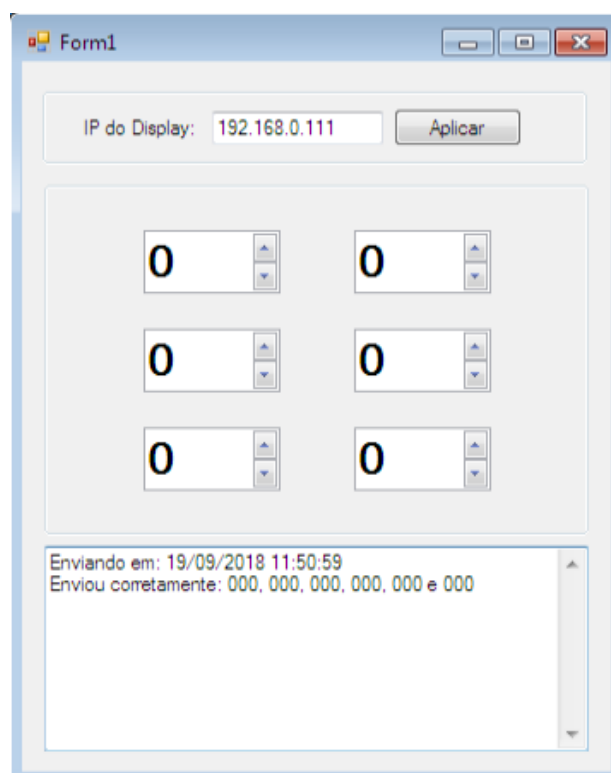
O software aplicativo exemplo deve ser usado para testar a comunicação com o painel e verificar se toda a infraestrutura de rede esta funcionando.

O uso do software é bem simples:

1º Abra o software aplicativo: **“Biblioteca Vagas”**.

2º Preencha o campo IP com o endereço IP do painel, e clique em **“Aplicar”**.

3º Alterando os valores do aplicativo o painel também devem mudar.



## 5. Biblioteca LibVagas

### Propriedades:

String **\_dislp** – Endereço IP do painel (servidor), remoto (RW).

String **\_status** – Informativo sobre status do último comando chamado (R).

### Métodos:

#### Nome:

Show

**Descrição:**

Envia comando para o painel com as informações de vagas.

**Cabeçalho:**

bool Show(string value1, string value2, string value3, string value4, string value5, string value6)

**Parâmetros:**

value1 – Valor do display 1.

Value2 – Valor do display 2.

Value3 – Valor do display 3.

Value4 – Valor do display 4.

Value5 – Valor do display 5.

Value6 – Valor do display 6.



## 6. Código exemplo em C#.NET

Iniciaremos a análise do código abrindo o Visual Studio e clicando na sequência de menus FILE → OPEN PROJECT, abra o projeto instalado em “c:\Program Files\Wyma Tecnologia\Lib Vagas\source\App\App.csproj”.

Com o projeto aberto localize “**Solution Explorer**” e expanda o projeto “**App**” e em seguida expanda a pasta “**References**” note que a dll “**LibVagas**” aparece nas referências do projeto.

Agora clique em **Form1.cs** e vamos abrir o código para ver como é feito o uso da DLL.

Note que é criada uma nova instância do objeto com a linha de código abaixo, onde “**Display**” é o construtor da DLL “**LibVagas**”:

```
private IDisplay disp = new Display();
```

No carregamento do form1 preencha as seguintes propriedades:

Precisamos informar o endereço do servidor remoto, altere e preencha com o IP do seu painel.

```
disp._dispIp = "192.168.0.111";
```



O exemplo utiliza de um timer para atualizar o display a cada 1 segundo usando a DLL “**LibVagas**” que tem um método chamado “Show” para simplificar a comunicação com o painel.

```
disp.Show(  
string.Format("{0:D3}",(int)numericUpDown1.value),  
string.Format("{0:D3}",(int)numericUpDown2.value),  
string.Format("{0:D3}",(int)numericUpDown3.value),  
string.Format("{0:D3}",(int)numericUpDown4.value),  
string.Format("{0:D3}",(int)numericUpDown5.value),  
string.Format("{0:D3}",(int)numericUpDown6.value));
```

As mensagens de resultados dos comandos da biblioteca você obtém a partir do atributo:

```
disp._status
```

## 7. Registrando o arquivo typeLib (tlb), no Windows

Para usar a DLL em sistemas não Microsoft .net, é necessário registrar os arquivos TLB que são distribuídos na instalação junto com os arquivos DLL, verifique na pasta de atalhos do programa o atalho para “**LibVagas → Arquivos biblioteca**”, este é um atalho para a pasta onde são copiadas as DLLs e TLBs necessários para o funcionamento do software.

Para registrar os arquivos TLB, você precisa abrir o **Prompt de comandos** do Windows em modo **administrador**, clicando o botão direito do mouse sobre o atalho esta opção estará disponível.

Certifique-se de ter instalado o Framework indicado no começo do manual, ou será necessário mudar o path abaixo para o caminho de um framework instalado em seu computador.

Digite o seguinte comando:

```
set path="%path%";C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319
```

A linha de comando acima apenas vai facilitar o uso do aplicativo “**regasm**”, assim não teremos que digitar todo o caminho dele para que o aplicativo funcione, basta digitar “regasm”.

Agora vamos mudar o diretório do prompt de comandos para a pasta onde estão os arquivos TLB para executar o comando regasm.

Para registrar o WyDriver.TLB execute o comando:

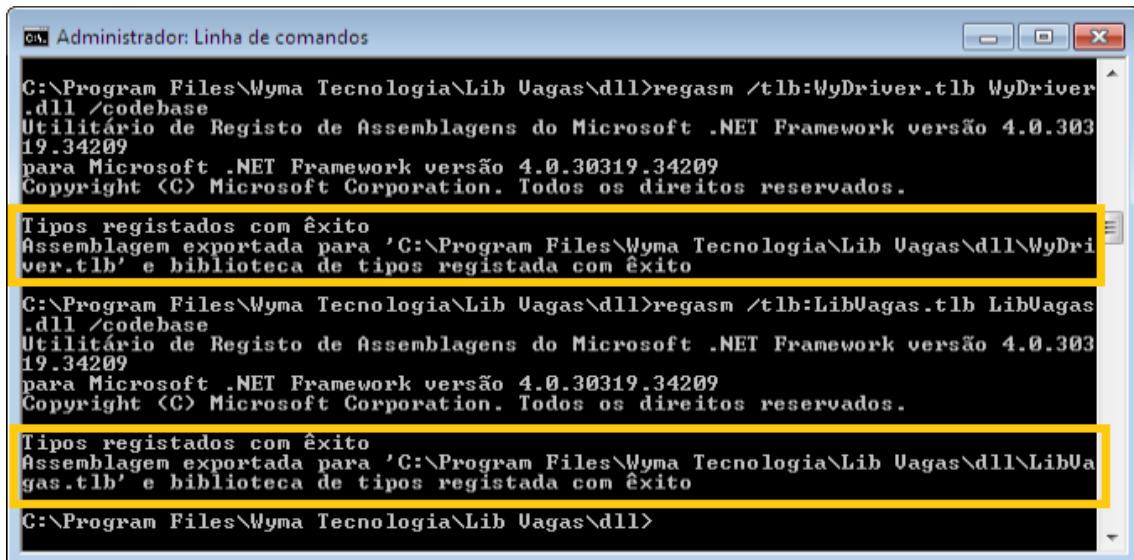
```
regasm /tlb:WyDriver.tlb WyDriver.dll /codebase
```

Para registrar o LibVagas.TLB execute o comando:

```
regasm /tlb:LibVagas.tlb LibVagas.dll /codebase
```

Após este comando tudo vai estar registrado no Windows para começar a usar como um componente.

A tela abaixo exemplifica alguns dos comandos digitados no prompt, note que ao final do registro de cada um dos TLB é indicado se o registro obteve sucesso.



```
Administrator: Linha de comandos

C:\Program Files\Wyma Tecnologia\Lib Vagas\dll>regasm /tlb:WyDriver.tlb WyDriver.dll /codebase
Utilitário de Registro de Assemblagens do Microsoft .NET Framework versão 4.0.30319.34209
para Microsoft .NET Framework versão 4.0.30319.34209
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Tipos registrados com êxito
Assemblagem exportada para 'C:\Program Files\Wyma Tecnologia\Lib Vagas\dll\WyDriver.tlb' e biblioteca de tipos registrada com êxito

C:\Program Files\Wyma Tecnologia\Lib Vagas\dll>regasm /tlb:LibVagas.tlb LibVagas.dll /codebase
Utilitário de Registro de Assemblagens do Microsoft .NET Framework versão 4.0.30319.34209
para Microsoft .NET Framework versão 4.0.30319.34209
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

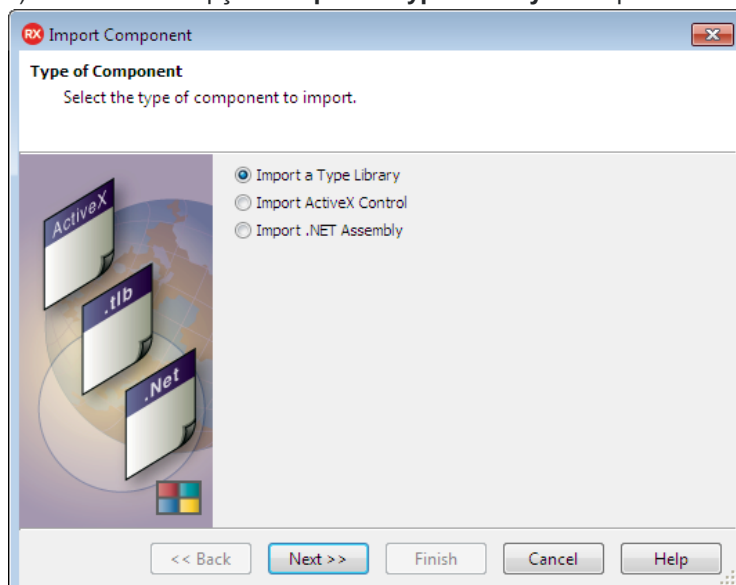
Tipos registrados com êxito
Assemblagem exportada para 'C:\Program Files\Wyma Tecnologia\Lib Vagas\dll\LibVagas.tlb' e biblioteca de tipos registrada com êxito

C:\Program Files\Wyma Tecnologia\Lib Vagas\dll>
```

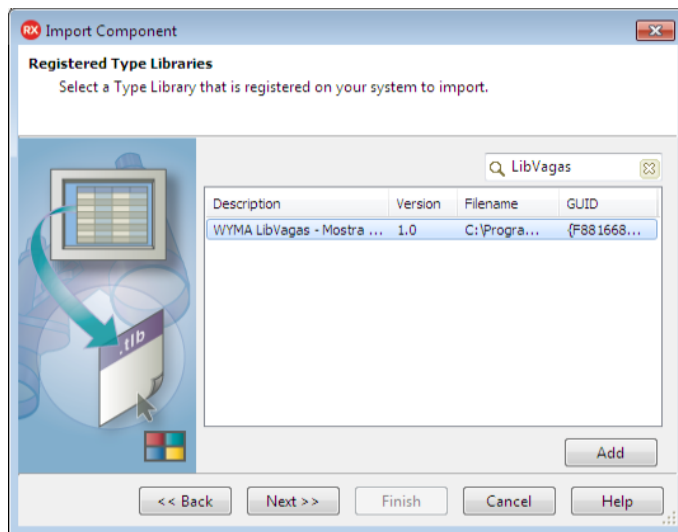
## 8. Criando os componentes no Delphi

O procedimento para usar a DLL como um componente no Delphi é bem simples, o exemplo abaixo foi feito no RAD Studio 10.1 Berlin.

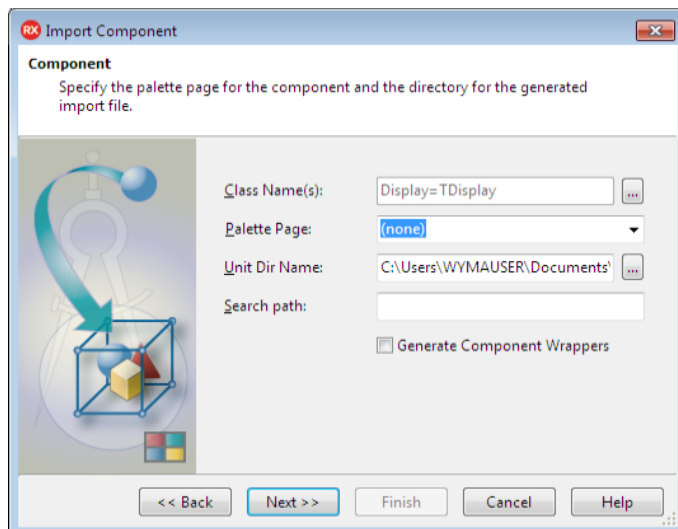
- Para criar os componentes vá até o menu em: **“Component”**, depois em **“Import Component”**.
- Selecione a opção: **“Import a Type Library”** e clique em **“Next”**:



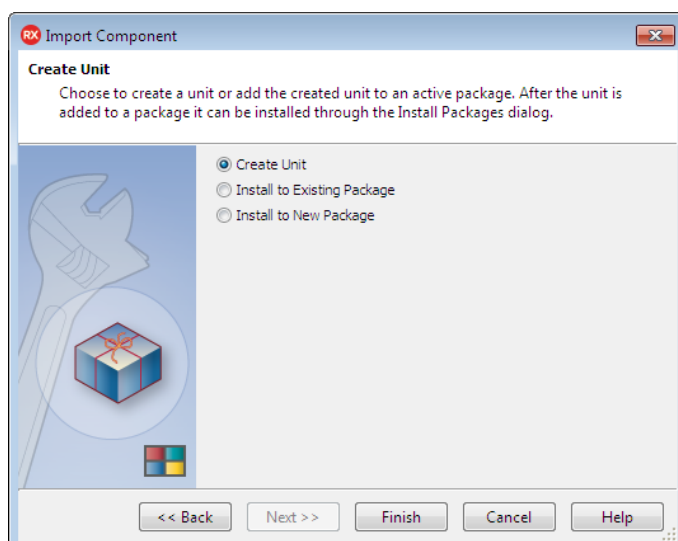
c) Digite “**LibVagas.tlb**” na pesquisa escolha o item WYMA LibVagas e clique em “**Next**”:



d) Clique em “**Next**”:



e) Por fim escolha “**Create Unit**” e clique em “**Finish**”:



- f) Este componente **“LibVagas.tlb”** depende de outro componente chamado **“WyDriver.tlb”**, é necessário seguir as mesmas etapas acima (passos “a” até “e”), para importar esta outra biblioteca também.

## 9. Criando um projeto no Delphi

Agora podemos criar um projeto para usarmos estes novos componentes em nosso projeto, siga as etapas abaixo para começar um projeto novo:

- 1) Clique em **“File”**, depois em **“New”** e finalmente em **“VCL Forms application – Delphi”** para criar um projeto novo.
- 2) Localize a caixa de ferramentas **“Tool Palette”** e digite no campo de busca o texto: **“TButton”**, com isto deverá aparecer o componente **“TButton”** logo abaixo, clique duas vezes sobre ele para que ele seja acrescentado ao formulário do projeto.
- 3) Verifique que agora o formulário **“Form1”** tem um botão, clique duas vezes sobre ele para irmos até o código.
- 4) Localize mais acima no código a palavra chave **“uses”**. Em **“uses”**, acrescente as bibliotecas: **ActiveX**, **WyDriver\_TLB** e **LibVagas\_TLB**, estas bibliotecas são necessárias para o nosso projeto.
- 5) Agora digite o código abaixo para o componente TButton.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  iLibrary: IDisplay;
begin
  iLibrary:=CoDisplay.Create();           // Cria instância da dll
  iLibrary._displp:='192.168.0.111';      // Configura o Ip do servidor Display

  iLibrary.Show('001','002','003','004','005','006'); // Envia requisição para o Display.
end;
```

- 6) Antes de rodar o exemplo verifique qual o IP do servidor Display e ajuste no código a linha abaixo com o IP correto:

```
iLibrary._displp:='192.168.0.111';      // Configura o Ip do servidor Display
```

- 7) Agora basta adequar ao seu programa..