

BIBLIOTECA LIBMIG



EXEMPLO DE USO REV.2

Tutorial de utilização da Biblioteca LibMig para comunicação com servidor Mig Wyma para chamada de tickets de atendimento.

Rev.0 – 05/10/2016 – Revisão inicial

Rev.1 – 14/10/2016 – Incluídos atributos da última chamada no painel, permitindo recuperar as informações da última chamada em caso de queda do sistema.

Rev.2 – 19/10/2016 – Corrigido recuperação de informações do último ticket.
Alterados atributos de última chamada para somente leitura. Excluído o método “RepeatCall()” e incluído “GetLastTicket()”

Índice

1. Introdução	3
2. Requisitos para iniciar	3
3. Instalando	4
4. Software aplicativo exemplo	7
5. Biblioteca LibMig	8
6. Código exemplo em C#.NET	10
7. Registrando os arquivos typeLib (tlb), no Windows.....	12
8. Criando os componentes no Delphi.....	13
9. Criando um projeto no Delphi	15

1. Introdução

A Biblioteca de Comunicação LibMig 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 2015, 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, aplicativo exemplo e arquivos de biblioteca “DLL” e “TLB”.

2-LibMig – Exemplo de uso.pdf = este manual de utilização

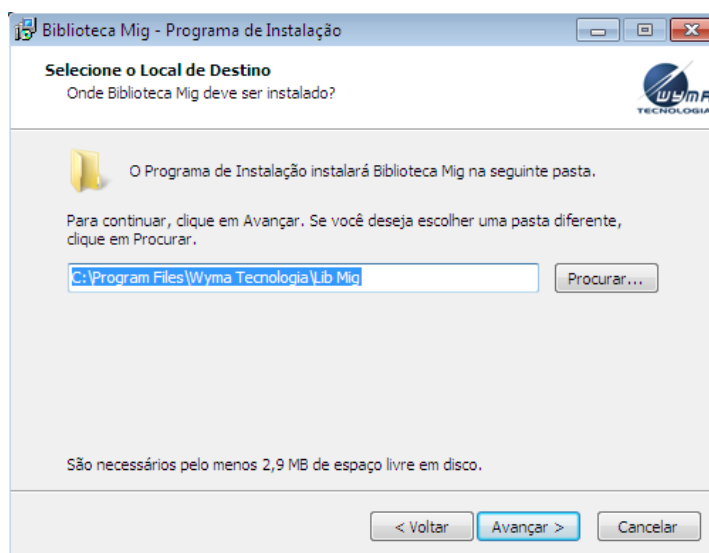
Também é necessário que a impressora MIG e o painel estejam ligados, e conectados à rede. Também devem ter o endereço IP configurado (consulte o manual de instalação da impressora e do painel).

3. Instalando

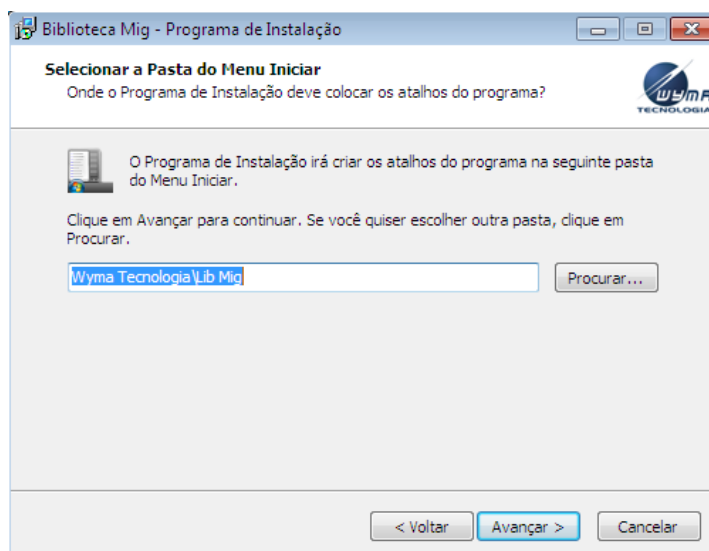
Localize o arquivo “**setup.exe**” e com dois cliques abra o instalador. Clique no botão “**Avançar**” para iniciar a instalação.



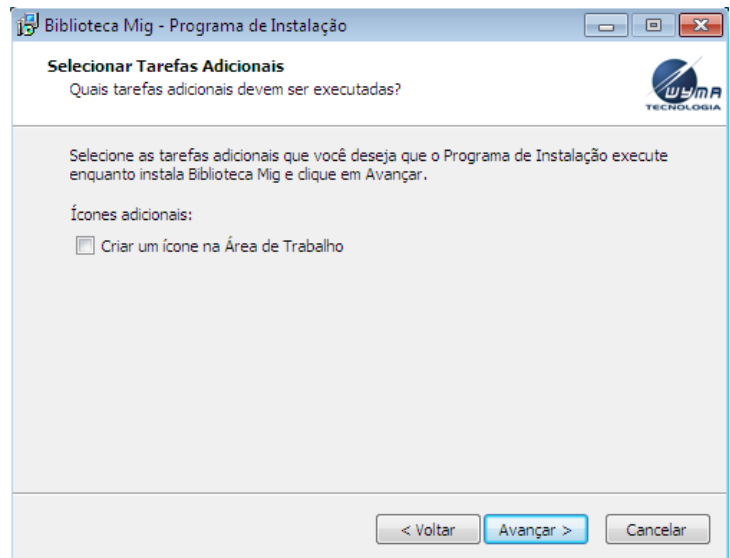
Sugerimos deixar a mesma pasta de instalação que será usada como referência neste documento.



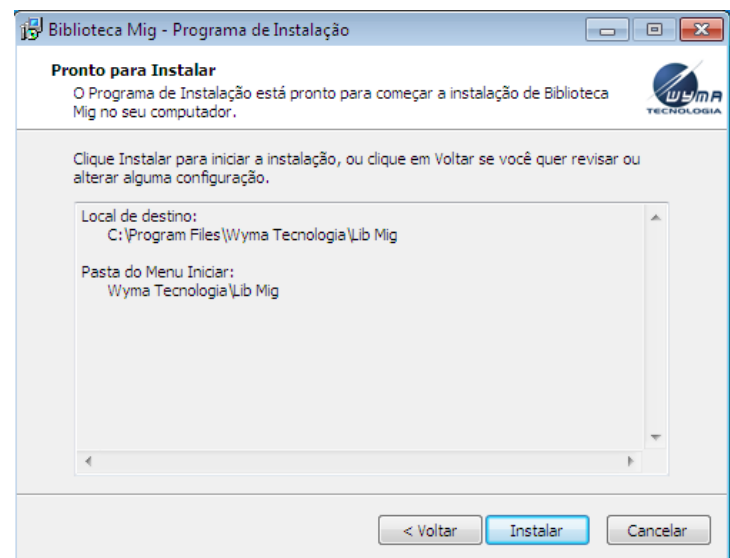
Esta é a pasta de atalhos para o aplicativo de teste.



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



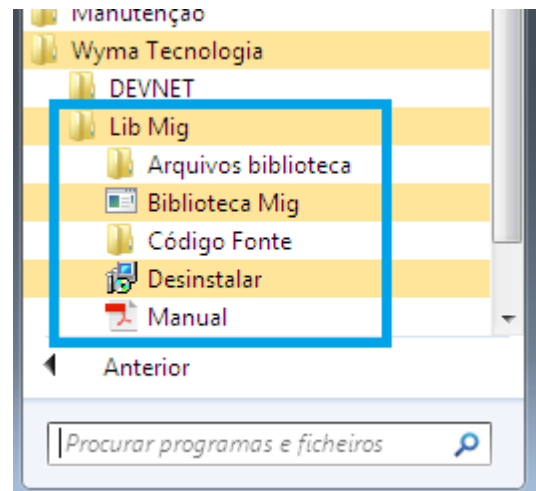
Este é o resumo das opções escolhidas para a instalação, clique em instalar para finalizar.



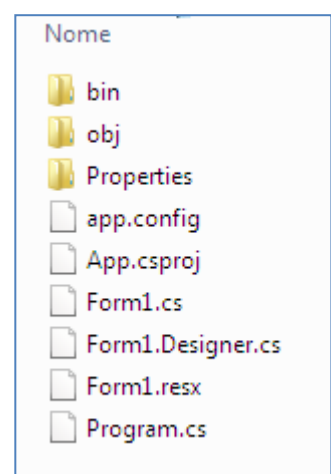
Marque a opção "Executar" para iniciar o aplicativo assim que concluir a instalação.



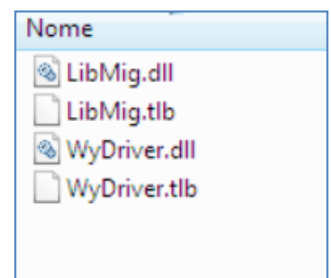
Após a instalação alguns atalhos serão criados dentro da pasta “Wyma Tecnologia” o nome do atalho para o aplicativo será “Lib Mig”. Note que existe um atalho para a pasta “**Código fonte**” com o código de um projeto exemplo e também um atalho para uma pasta “**Arquivos biblioteca**”, onde estão os arquivos 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 conectividade com os dispositivos e verificar se toda a infraestrutura de rede esta funcionando.

O uso do software é bem simples:

1º Cadastre algumas senhas no dispositivo MIG tocando a área touch do primeiro atendimento no dispositivo.

2º Abra o software aplicativo: **“Biblioteca Mig”**.

3º A tela deste software é repleta de comandos que controlam a chamada de tickets no servidor MIG. Em Informações básicas preencha com o IP do servidor e com o box que irá chamar, logo em seguida clique em “Aplicar” para salvar o IP e Box. Cada um dos 7 botões abaixo realizam uma tarefa diferente de atendimento, escolha um e clique para verificar o que acontece, a parte inferior do software vai informar o resultado da operação. No canto direito você pode consultar as informações da chamada no painel.

The screenshot shows a Windows application window titled 'Form1'. It is divided into two main panels. The left panel, titled 'Informações básicas', contains fields for 'IP do servidor MIG:' (192.168.0.114) and 'Usuário:' (1), with an 'Aplicar' button. Below these are seven buttons: 'Clique para chamar o próximo ticket na fila do servidor', 'Clique para chamar ticket específico no servidor:' (with a dropdown set to 1), 'Clique para repetir a chamada', 'Clique para encerrar o atendimento', 'Clique para cancelar o atendimento', 'Clique para encaminhar o ticket para departamento:' (with a dropdown set to 1), and 'Clique para habilitar a pesquisa no teclado de opinião'. At the bottom of this panel is a text box displaying 'Chamando próximo Ticket da fila: N563 no Box: 1'. The right panel, titled 'DADOS DA ÚLTIMA CHAMADA NO DISPLAY', contains a list of fields with their current values: 'Linha do display' (1), 'Letra no display' (N), 'Box no display' (1), 'Ticket no display' (563), 'Tipo de Ticket no display' (Normal), 'Número Depto' (1), 'Nome Depto' (NORMAL), 'IP do display' (192.168.0.112), 'Interface Display' (ETHERNET), 'IP do TOP' (192.168.0.117), and 'Interface TOP' (ETHERNET). A 'Recupera' button is located at the bottom of this panel.

Informações básicas	
IP do servidor MIG:	192.168.0.114
Usuário:	1
<input type="button" value="Aplicar"/>	
<input type="button" value="Clique para chamar o próximo ticket na fila do servidor"/>	
Clique para chamar ticket específico no servidor:	1
<input type="button" value="Clique para repetir a chamada"/>	
<input type="button" value="Clique para encerrar o atendimento"/>	
<input type="button" value="Clique para cancelar o atendimento"/>	
Clique para encaminhar o ticket para departamento:	1
<input type="button" value="Clique para habilitar a pesquisa no teclado de opinião"/>	
Chamando próximo Ticket da fila: N563 no Box: 1	

DADOS DA ÚLTIMA CHAMADA NO DISPLAY	
Linha do display	1
Letra no display	N
Box no display	1
Ticket no display	563
Tipo de Ticket no display	Normal
Número Depto	1
Nome Depto	NORMAL
IP do display	192.168.0.112
Interface Display	ETHERNET
IP do TOP	192.168.0.117
Interface TOP	ETHERNET
<input type="button" value="Recupera"/>	

5. Biblioteca LibMig

Propriedades:

string **_serverIp** – Endereço IP do servidor remoto (RW).
byte **_userNum** – Número do usuário em operação (RW).
string **_status** – Informativo sobre status do último comando chamado (R).
byte **_activeRowDisp** – Número da linha do painel na última chamada (R).
byte **_activeAlphaTicket** – Letra do ticket na última chamada (R).
byte **_activeBoxUser** – Número do box na última chamada (R).
ushort **_activeTicket** – Número do ticket na última chamada (R).
byte **_activeTypeTicket** – Tipo do ticket na última chamada (R).
byte **_activeDepNum** – Número do departamento na última chamada (R).
string **_activeTopIp** – Endereço IP teclado opinião na última chamada (R).
byte **_activeTopInterf** – Interface do TOP na última chamada (R).
string **_activeDepName** – Nome do departamento na última chamada (R).
byte **_activeDispInterf** – Interface do display na última chamada (R).
string **_activeDisplayIp** – Endereço IP do display na última chamada (R).

Métodos:

Nome:

GetNextTicket

Descrição:

Solicita ao servidor MIG o próximo ticket.

Cabeçalho:

bool GetNextTicket()

Parâmetros:

Nenhum

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

SetCallTicket

Descrição:

Solicita ao servidor MIG uma chamada de um ticket específico.

Cabeçalho:

bool SetCallTicket(byte userNum, ushort ticketNumSpec)

Parâmetros:

userNum – Número do usuário em operação

ticketNumSpec – Número do ticket a ser exibido no painel

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

GetLastTicket

Descrição:

Solicita ao servidor MIG uma chamada ao último ticket do usuário ativo.

Cabeçalho:

bool GetLastTicket()

Parâmetros:

Nenhum

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

CloseCall

Descrição:

Solicita ao servidor MIG o encerramento do último ticket.

Cabeçalho:

bool CloseCall()

Parâmetros:

Nenhum

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

CancelCall

Descrição:

Solicita ao servidor MIG o cancelamento do último ticket.

Cabeçalho:

bool CancelCall()

Parâmetros:

Nenhum

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

RouteTicket

Descrição:

Solicita ao servidor MIG o encaminhamento do último ticket para outro departamento.

Cabeçalho:

bool RouteTicket(byte depNum)

Parâmetros:

depNum – Número do departamento para encaminhar o ticket

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

Nome:

StartResearch

Descrição:

Solicita ao servidor MIG o início de uma pesquisa de satisfação através de um teclado de opinião TOPM.

Cabeçalho:

bool StartResearch(byte userNum)

Parâmetros:

userNum – Número do usuário em operação.

Retorno:

true=Comunicação bem sucedida ; false=Erro de comunicação

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 Mig\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 "**LibMig**" é um item necessário ao 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 "**ServerMig**" é o construtor da DLL "**LibMig**":

```
private IServerMig mig = new ServerMig();
```

No carregamento do form1 preencha as seguintes propriedades:

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

```
mig._serverIp = "192.168.0.114";
```

Precisamos informar também o número do usuário em operação, esta informação deve ser alterada de acordo com a configuração de usuários feita no servidor MIG.

```
mig._userNum = 1;
```

O restante são os códigos dos botões, onde cada botão executa um comando diferente no servidor MIG e aqui entra a DLL **“LibMig”** que tem um comando para simplificar cada uma das funções de atendimento do servidor MIG.

Chamada de próximo ticket:

```
mig.GetNextTicket();
```

Chamada do ticket específico “005”, com usuário número “1”:

```
mig.SetCallTicket(1, 005);
```

Repete chamada do ticket:

```
mig.GetLastTicket();
```

Fecha o atendimento do ticket:

```
mig.CloseCall();
```

Cancela a chamada do ticket:

```
mig.CancelCall();
```

Encaminha o ticket para o departamento de número 5:

```
mig.RouteTicket(5);
```

Requisita o inicio da pesquisa de satisfação do ticket para o usuário 1:

```
mig.StartResearch(1);
```

Para auxiliar na compreensão do que ocorre internamente na DLL, você deve ler o parâmetro:

```
mig._status
```

Isto é feito no método abaixo, todas as informações são mostradas na caixa de texto na tela do aplicativo de exemplo.

```
textBox1.Text += mig._status + Convert.ToChar(13) + Convert.ToChar(10);
```

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 “**LibMig → 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 LibMig.TLB execute o comando:

```
regasm /tlb:LibMig.tlb LibMig.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.

```
CA: Administrador: Linha de comandos
Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

M:\>c:

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

C:\>cd C:\Program Files\Wyma Tecnologia\Lib Mig\dll

C:\Program Files\Wyma Tecnologia\Lib Mig\dll>regasm /tlb:LibMig.tlb LibMig.dll /
codebase
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.18408
for Microsoft .NET Framework version 4.0.30319.18408
Copyright (C) Microsoft Corporation. All rights reserved.

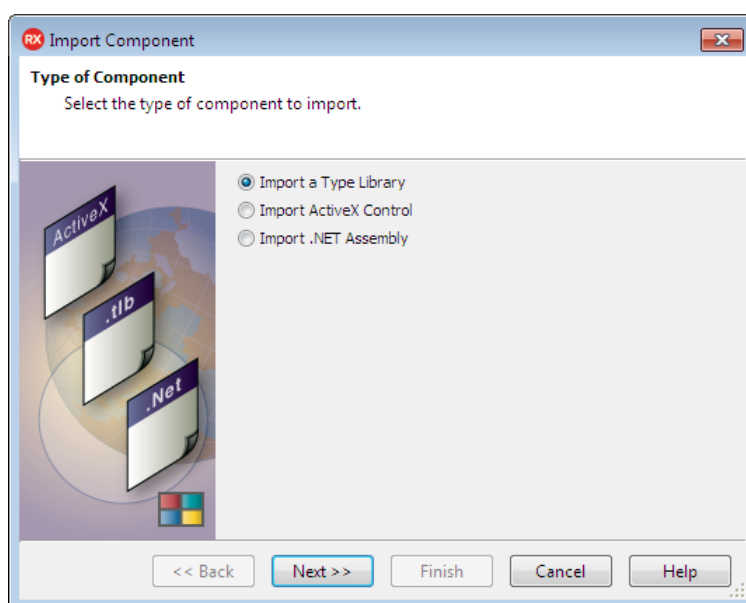
Assembly exported to 'C:\Program Files\Wyma Tecnologia\Lib Mig\dll\LibMig.tlb',
and the type library was registered successfully

C:\Program Files\Wyma Tecnologia\Lib Mig\dll>_
```

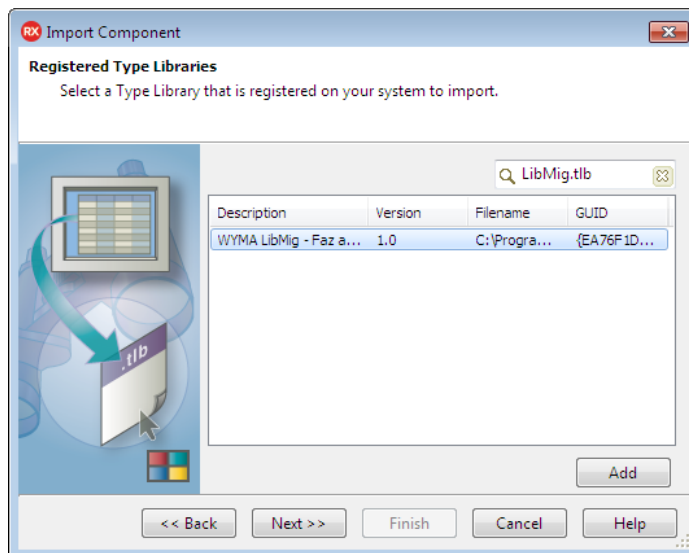
8. Criando os componentes no Delphi

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.

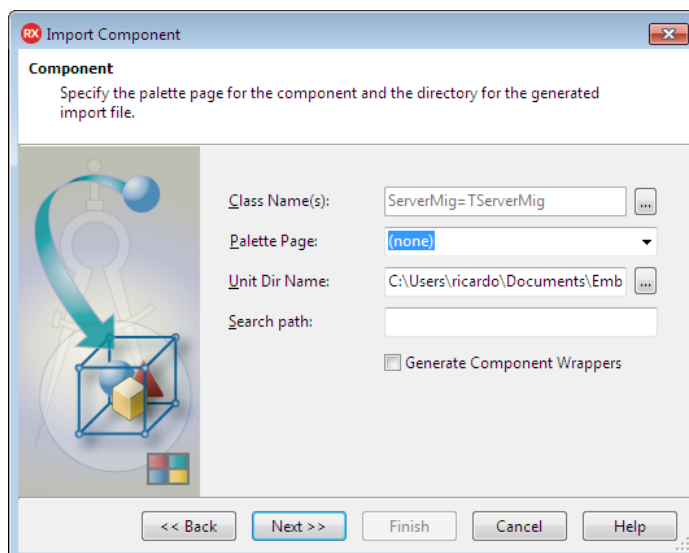
- Para criar os componentes vá até o menu em: **“Component”**, depois em **“Import Component”**.
- Selecione a opção: **“Import a Type Library”** e clique em **“Next”**:



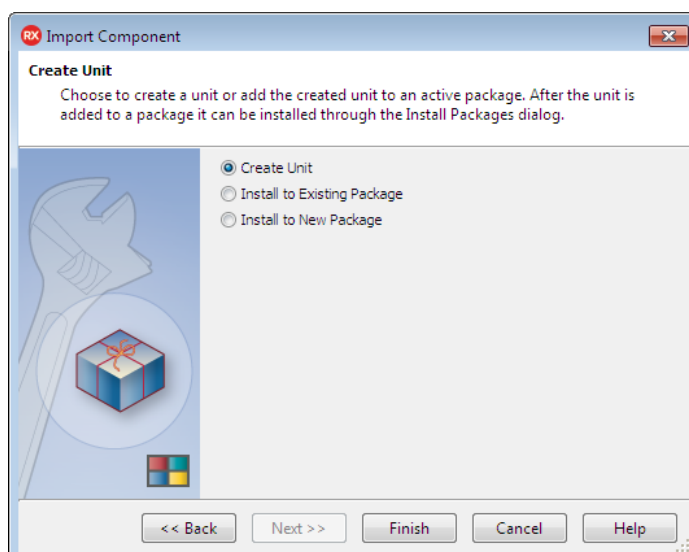
- Digite **“LibMig.tlb”** na pesquisa escolha o item WYMA LibMig e clique em **“Next”**:



d) Clique em **“Next”**:



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



- f) Este componente **“LibMig.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:

- a) Clique em **“File”**, depois em **“New”** e finalmente em **“VCL Forms application – Delphi”** para criar um projeto novo.
- b) 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.
- c) Verifique que agora o formulário **“Form1”** tem um botão, clique duas vezes sobre ele para irmos até o código.
- d) Localize mais acima no código a palavra chave **“uses”**. Em **“uses”**, acrescente as bibliotecas: **ActiveX**, **WyDriver_TLB** e **LibMig_TLB**, estas bibliotecas são necessárias para o nosso projeto.
- e) Agora digite o código abaixo para o componente TButton.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  iLibrary: IServerMig;
begin
  iLibrary:=CoServerMig.Create();      // Cria instância da dll
  iLibrary._serverIp:='192.168.0.114'; // Configura o Ip do servidor MIG
  iLibrary._userNum:=1;                // Configura o número do usuário

  iLibrary.GetNextTicket();            // Envia requisição para o servidor MIG.
end;
```

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

```
iLibrary.serverIp:='192.168.0.114'; // Configura o Ip do servidor MIG
```

- g) Certifique-se de ter cadastrado alguma senha no servidor MIG e de ter configurado no servidor MIG o IP do painel de chamada, para que ocorra a chamada.
- h) Agora basta adequar ao seu programa..