

Configure TeraLib 3.3 no Visual Studio 2003

João Tácio Corrêa da Silva
joatacio@yahoo.com.br

Ouro Preto - 2010

O tutorial apresentado é um complemento da documentação disponibilizada pelo INPE¹, que ensina como configurar a biblioteca TerraLib 3.3.1 no Visual Studio 2003.

1 Requisitos para executar os exemplos

- Visual Studio C++ 2003;
- MySQL Administrator 5.0;
- Driver MySQL para Windows;
- Banco de dados MySQL versão 5.1;
- Biblioteca TerraLib para Windows versão 3.3 ou superior.

2 Instruções gerais de instalação

O primeiro software a instalar é o Visual Studio 2003. Utilize a instalação personalizada e desmarque o suporte a todas as outras linguagens diferentes do C++, por exemplo: C# e VB. A seguir, instale o MySQL, Driver do MySQL e o MySQL Administrator.

Após instalar com sucesso os softwares descritos acima, crie uma pasta no disco C, por exemplo C:/TerraDir e descompacte todo o conteúdo do arquivo TerraLib_win_v_3_3_1.zip. Após descompactar, a árvore de pastas deve seguir o formato da figura 1.

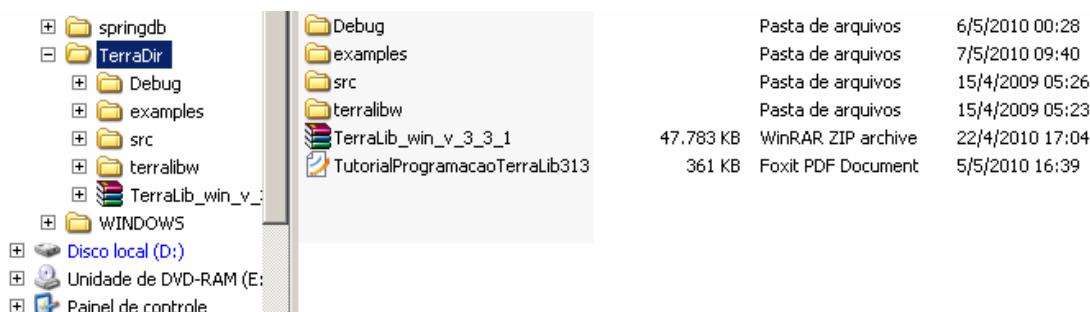


Figura 1 – Árvore de pastas do TerraLib

3 Compilando a biblioteca TerraLib

O primeiro passo é compilar toda biblioteca TerraLib. Para isso, abra o Visual Studio, clique em File/Open Project / e escolha a solution, referente à biblioteca que será executada. No exemplo abaixo será compilado a solution createDataBase.sln.

1) File/Open project/ ou Ctrl+Shift+O

C:\TerraDir\examples\createDatabase\createDatabase.sln

Certifique-se que o arquivo escolhido possui a extensão.sln, porque na pasta createDatabase, também possui um arquivo chamado createDatabase.dsw.

¹ <http://www.terralib.org/docs/v313/TutorialProgramacaoTerraLib313.pdf>

2) Build/Build Solution ou Ctrl+Shift+B

Build Sucessfull -> Essa mensagem é indício que está indo tudo muito bem, parabéns!

Faça este mesmo procedimento para todas as bibliotecas abaixo na ordem que estão listadas.

1. createLayer;
2. createTable;
3. importMIDMIF;
4. importShape;
5. importDBF;
6. importJPEG;
7. copyLayer;
8. importGeoTab;
9. convertCoordinates;
10. databaseQuery;
11. databaseSQLQuery;
12. spatialQuery;
13. addGeomRepresentation;
14. createTheme;
15. themeGrouping;
16. createSTElementSet;
17. mosaicTIFFImages;
18. importCSV;
19. importGridData;
20. rasterSlicing;
21. querierFromTheme;
22. proxMatrixAndSpatialStatistics;
23. spatialQueryAndBuffer;
24. querierFromLayer;
25. createSTElementSetFromLayer;
26. image_processing.

Após compilar todas as bibliotecas acima sem erros, pode fechar o Visual Studio. A próxima etapa consiste em codificar alguns trechos de códigos para iniciar o aprendizado na biblioteca TerraLib. No tutorial do INPE ² existem diversas páginas com trechos de código.

4 Configurar o projeto

Este tutorial ficará limitado em explicar detalhadamente apenas os passos necessários para criar um projeto.

O primeiro passo é criar um novo projeto, para isso abra o Visual Studio. Escolha File/New/Project, conforme a figura 2 e coloque o nome no projeto de Primeiro Exemplo.

² <http://www.terralib.org/docs/v313/TutorialProgramacaoTerraLib313.pdf>

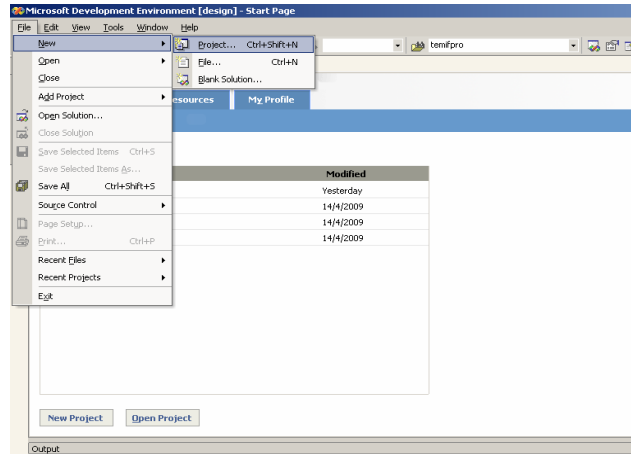


Figura 2 – Criar um novo projeto

Escolha o template Console Application (.NET), conforme imagem 3.

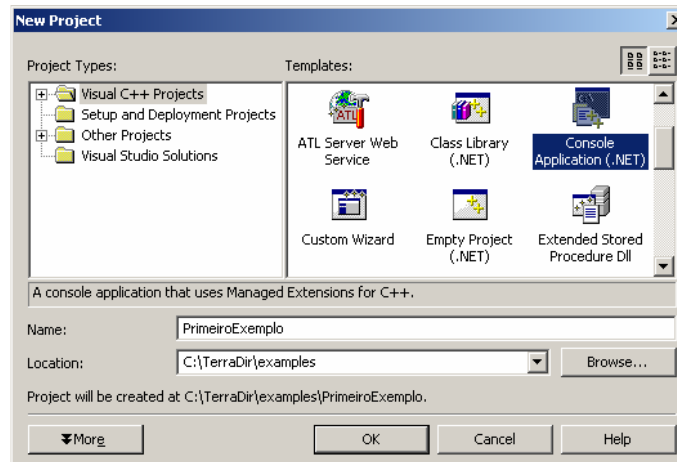


Figura 3 – Escolher template

Após criar o projeto PrimeiroExemplo, é necessário várias configurações. A primeira é adicionar os projetos existentes do TerraLib. Para isso clique na solution e escolha add e selecione Existing Project. Veja nas figuras 4, 5 e 6 como adicionamos o projeto terralib.

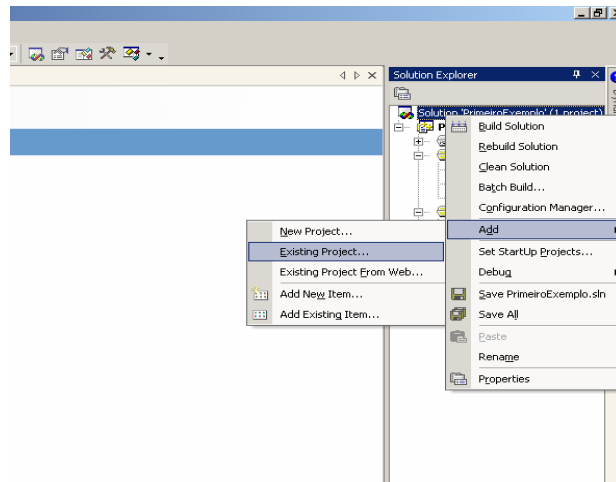


Figura 4 – Adicionar projeto existente 1º passo

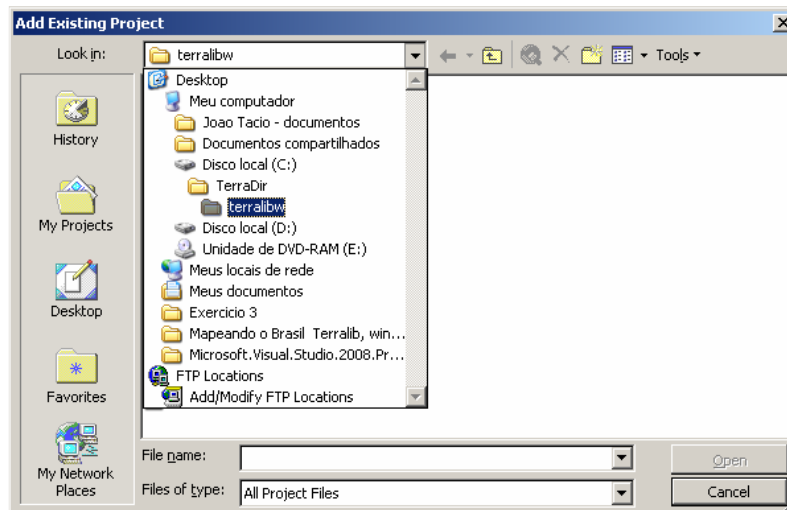


Figura 5 – Adicionar projeto terralib 2º passo

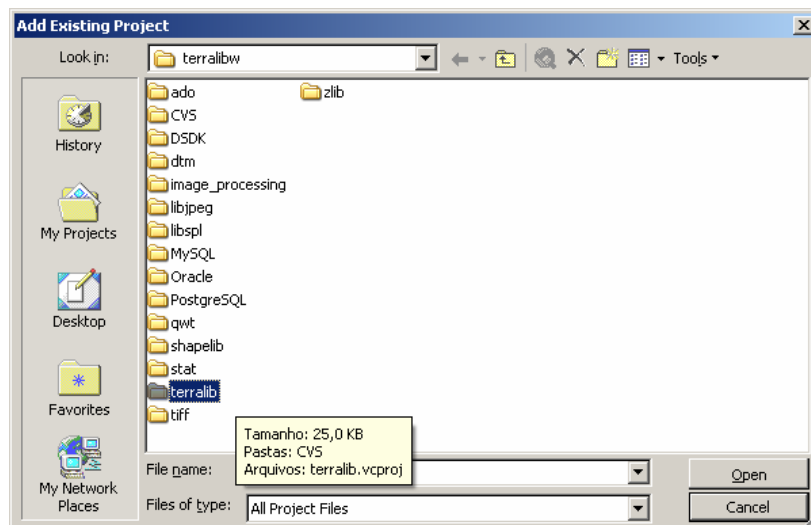


Figura 5 – Adicionar projeto terralib 3º passo

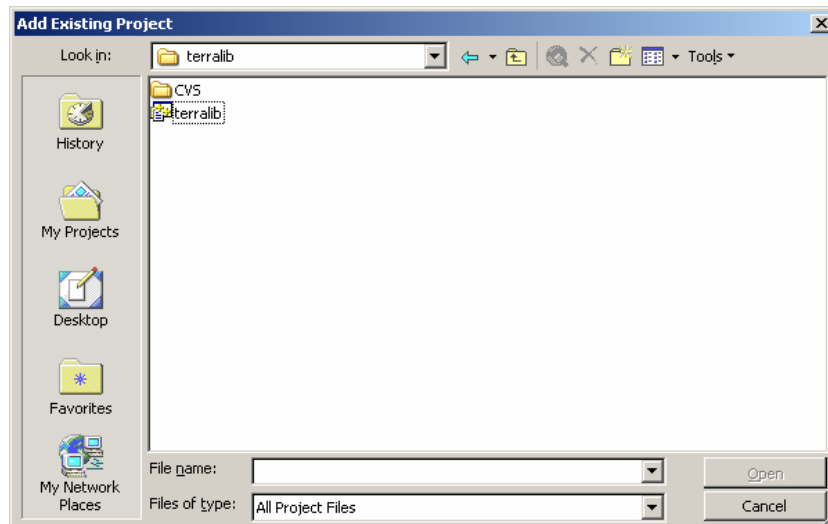


Figura 6 – Adicionar projeto terralib 4º passo

Além do projeto terralib, é necessário adicionar os projetos libjpeg e tiff, que estão respectivamente nos caminhos C:\TerraDir\terralibw\libjpeg e C:\TerraDir\terralibw\tiff. Após adicionar os três projetos, faça um Build na solution.

Além desses três projetos que foram incorporados ao projeto PrimeiroExemplo, outros projetos do TerraLib poderão ser agregados conforme a necessidade.

O procedimento a seguir objetiva configurar o projeto PrimeiroExemplo. Click nas propriedades do projeto, conforme na figura 7.

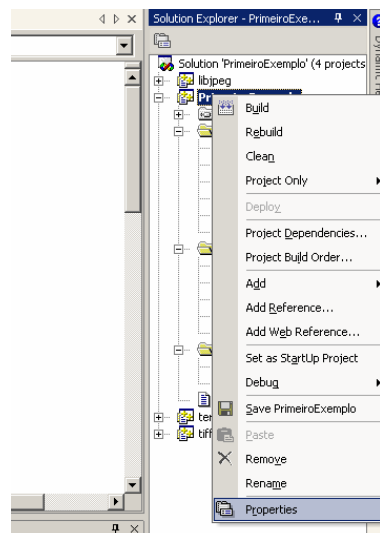


Figura 7 – Propriedades do projeto

Insira o caminho para os diversos diretórios adicionais do projeto. Para realizar essa operação, copie e cole toda a linha a seguir no campo Additional Include Directories. A imagem 8 detalha o que precisa ser feito.

..\..\src\terralib\kernel;..\..\src\terralib\drivers\MySQL;..\..\src\terralib\drivers\MySQL\include

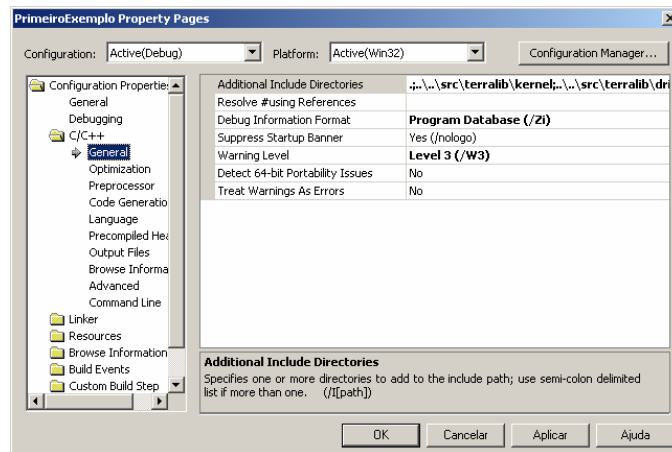


Figura 8 – Adicionar diretórios adicionais

A figura 9 explica como desabilitar o Precompiled Header.

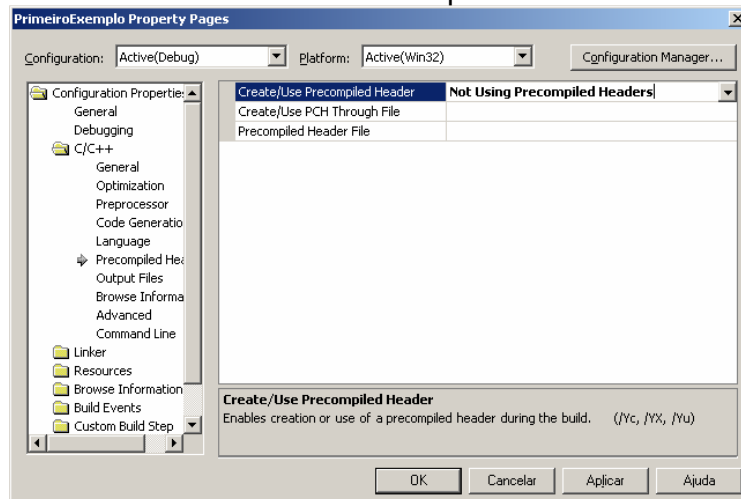


Figura 9 – Precompiled Header

Não feche a janela de propriedades. Configure o Linker do projeto PrimeiroExemplo. Esse procedimento é necessário para que o projeto criado consiga referenciar as classes do TerraLib. Inclua as dependências adicionais conforme ilustra a figura 10. Para isso copie as linhas abaixo e cole em Additional Dependencies.

```

..\..\Debug\terralib\terralib.lib ....\Debug\libjpeg\libjpeg.lib ....\Debug\tiff\tiff.lib
..\..\terralibw\zlib\zlibdll.lib ....\terralibw\MySQL\libMySQL.lib kernel32.lib
user32.lib gdi32.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib imm32.lib winmm.lib wsock32.lib winspool.lib delayimp.lib

```

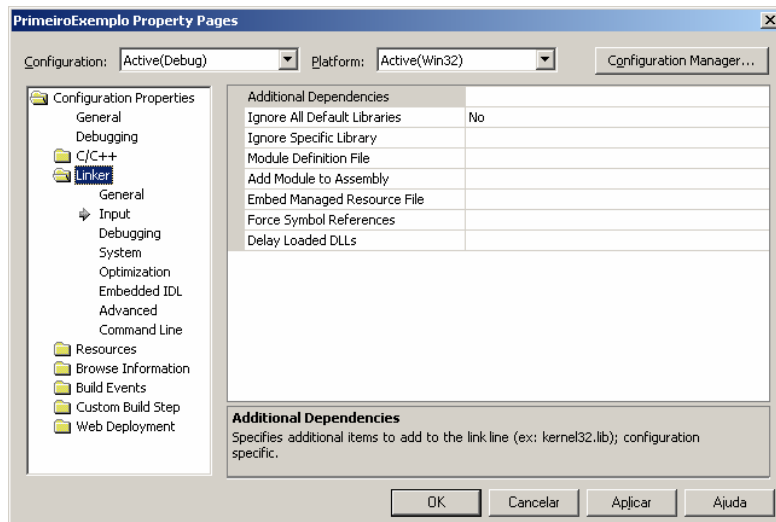


Figura 10 – Additional Dependencies

Após realizar todos os procedimentos acima, faça um build do projeto. Caso ocorrer algum erro, tente remover e copiar novamente as linhas que foram inseridas nas propriedades do projeto. Atenção ao copiar essas linhas, espaços não podem ser retirados ou inseridos.

Os procedimentos abaixo consistem em adicionar itens existentes ao projeto. Para o projeto PrimeiroProjeto, será adicionado as bibliotecas TeMySQL.h e TeMySQL.cpp. As figuras de 11 a 13, demonstram como adicionar o item TeMySQL.cpp ao Source File do Projeto.

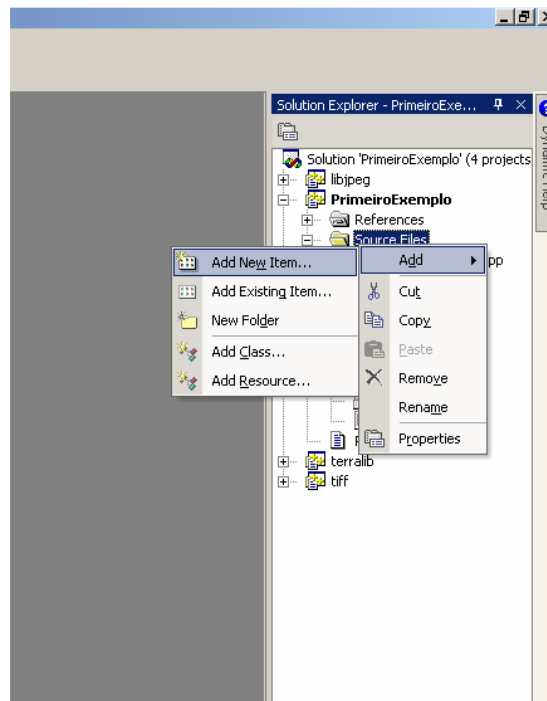


Figura 11 – Acionar itens existentes 1º passo

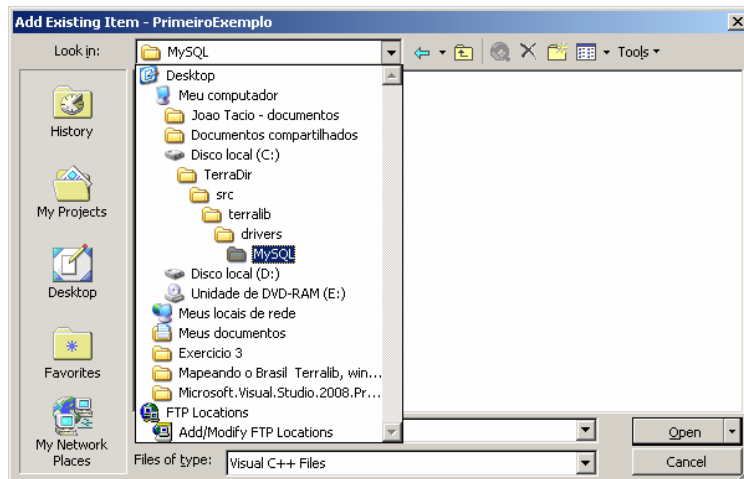


Figura 12 – Adicionar itens existentes 2º passo

A figura 13 apresenta os dois arquivos da biblioteca TerraLib que deverão ser inseridos ao projeto. Inicialmente escolha apenas TeMySQL.cpp a seleccione Open.

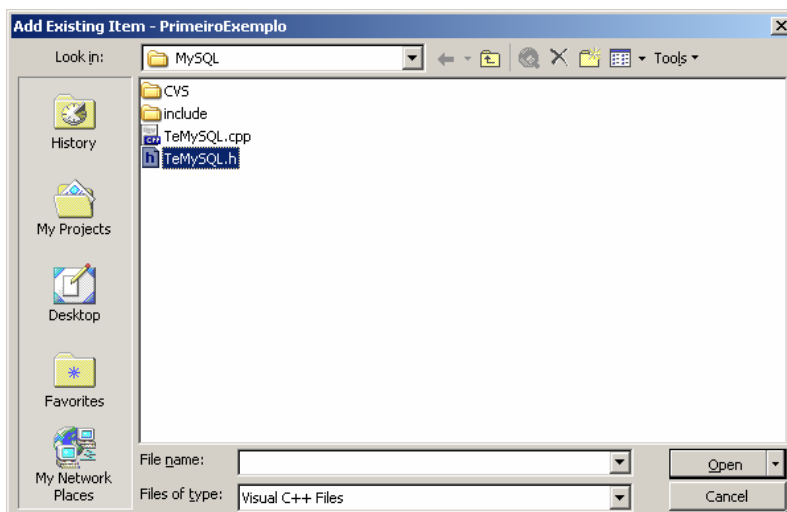


Figura 13 – Arquivos da biblioteca

Para adicionar o arquivo TeMySQL.h ao projeto, click sobre Header Files do projeto PrimeiroProjeto. A figura 14 demonstra como fazer isso.

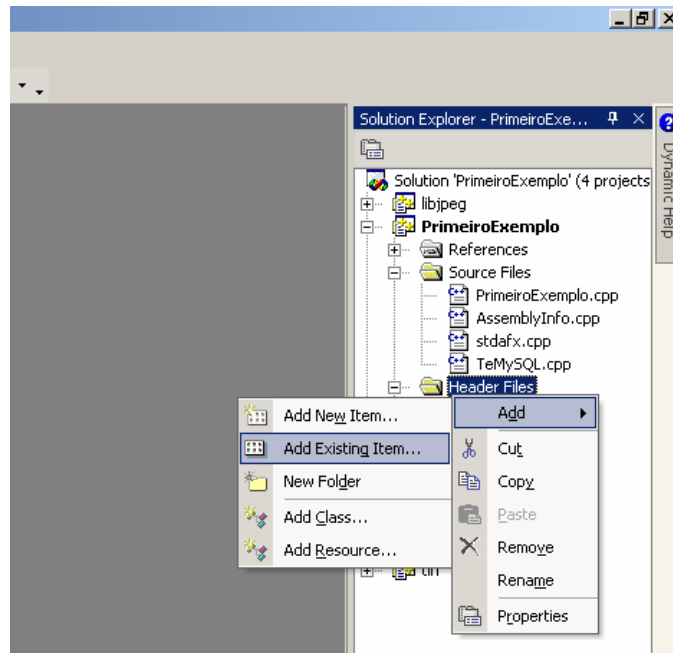


Figura 14 – Header Files

Os próximos passos para adicionar TeMySQL.h são os mesmos das figuras 12 e 13, porém escolha TeMySQL.h e pressione open.

Ao final de todos esses procedimentos, realize um build de todo o projeto PrimeiroProjeto para certificarmos que todos as configurações estão corretas.

5 – Codificar o primeiro exemplo

Para iniciar a codificação do primeiro exemplo, click duas vezes sobre a classe PrimeiroExemplo.cpp, conforme a imagem 15 e adicione as bibliotecas e os métodos criarBancoDeDados e conectar. Após a figura 15 tem o código completo da classe PrimeiroExemplo.cpp. Após codificar a classe, realize um build no projeto e execute-o.

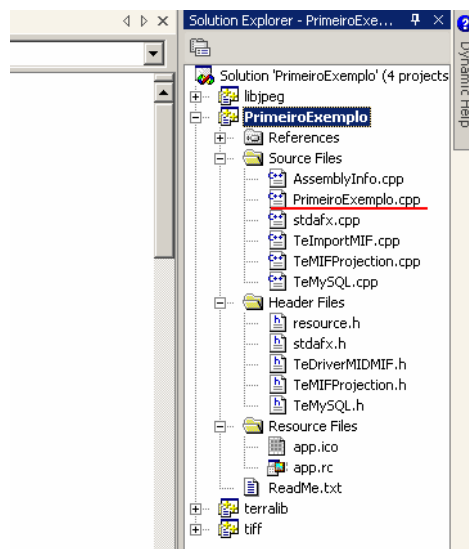


Figura 15 – Alterando a classe PrimeiroExemplo.cpp

```
#include "stdafx.h"
#include "conio.h"

#include "TeMySQL.h"
#include "TeDriverMIDMIF.h"

#using <microsoft.dll>

using namespace System;

//Declaração dos métodos
int conectar();
int criarBancoDeDados();

int _tmain()
{
    // TODO: Please replace the sample code below with your own.
    Console::WriteLine(S"Hello World");
    conectar();
    criarBancoDeDados();
    getch();
    return 0;
}

int criarBancoDeDados ()
{
    // Parâmetros do servidor de bancos de dados
    string host = "localhost";
    string dbname = "TerraTeste";
    string user = "root";
    string password = "";

    // Cria um novo banco
    TeDatabase* db = new TeMySQL();
    if (!db->newDatabase(dbname,user, password, host))
    {
        cout << "Erro: " << db->errorMessage() << endl;
        return 1;
    }

    cout << "O banco de dados \"" << dbname;
    cout <<"\" foi criado com sucesso no servidor MySQL localizado em\"
        << host;
    cout << "\" para o usuario \"" << user << "\"!" << endl;
    // Fecha o banco de dados
    db->close();
    delete db;
    return 0;
}

int conectar()
{
    // Cria uma conexão a um servidor de banco de dados
    TeDatabase* db = new TeMySQL();
    // Parametros do servidor de bancos de dados
    string host ="localhost";
    string dbname = "TerraTeste";
    string user = "root";
```

```
string password = "";  
if (!db->connect(host,user, password,dbname,0))  
{  
    cout<< "Erro: " << db->errorMessage() << endl;  
    return 0;  
}  
cout << "Aberta conexão ao banco: " << dbname;  
db->close();  
delete db;  
}
```