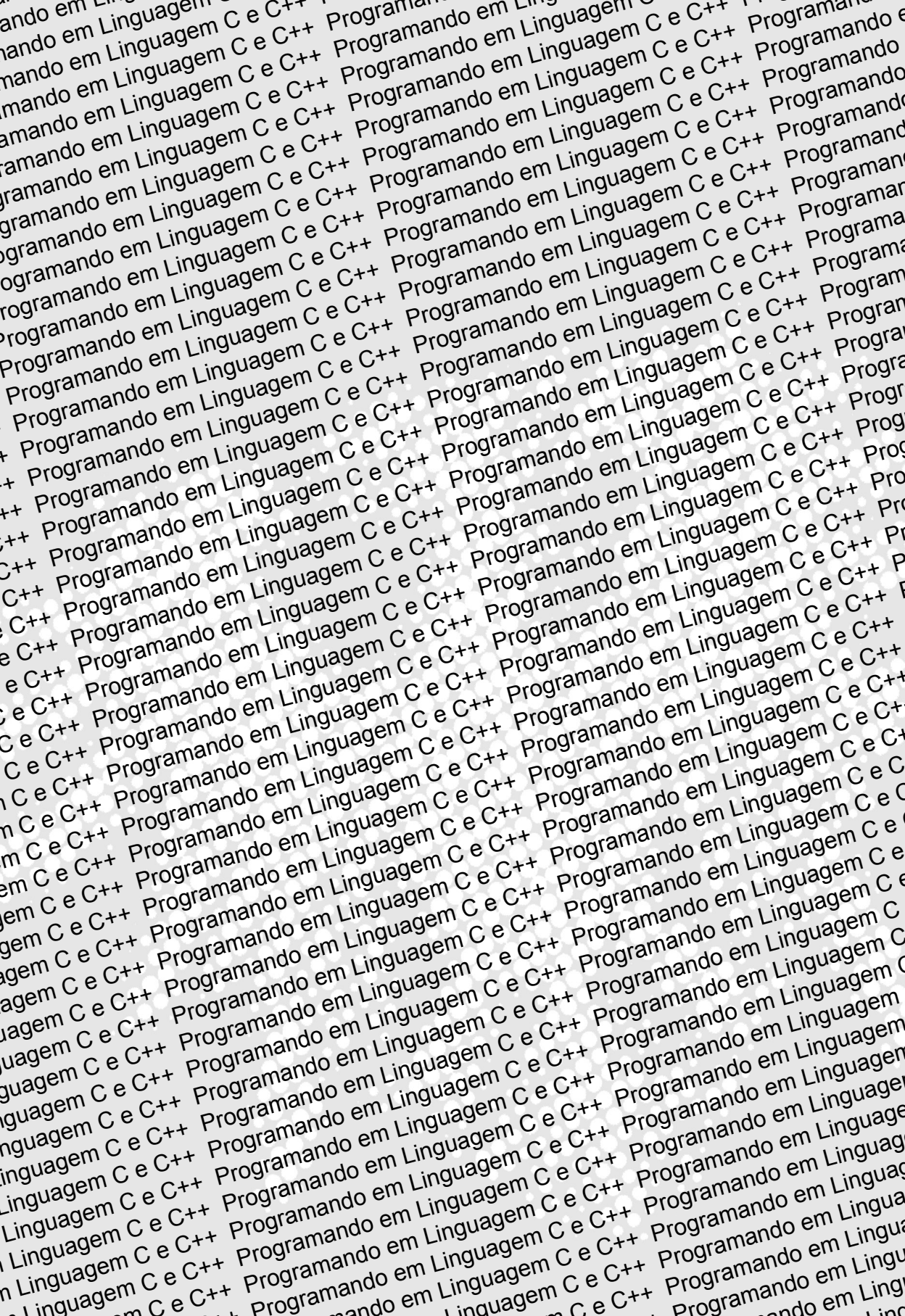

Programando em Linguagem C e C++

Antônio Henrique Reis

Programando em Linguagem C e C++
Com o Microsoft Visual Studio

viena

1ª Edição
Santa Cruz do Rio Pardo/SP
Editora Viena
2015

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Reis, Antônio Henrique
Programando em linguagem C e C++ : com o Microsoft Visual Studio / Antônio Henrique Reis. -- 1. ed. -- Santa Cruz do Rio Pardo, SP : Editora Viena, 2015. -- (Coleção premium)

Bibliografia.
ISBN 978-85-371-0433-0

1. C++ (Linguagem de programação para computadores) 2. Microsoft Visual Studio (Programa de computador) I. Título. II. Série.

15-09163 CDD-005.133

Índices para catálogo sistemático:

1. C++ : Linguagem de programação : Computadores : Processamento de dados 005.133

Autor: Antônio Henrique Reis
Revisão Ortográfica: Graciele Alves Mira
Capa: Luciane Mendonça
Diagramadora: Erika Cristina Bueno
Ilustrações: iStockphoto.com
Revisão de Diagramação: Wellington José dos Reis e Karina de Oliveira
Supervisão Editorial: Karina de Oliveira

ISBN: 978-85-371-0433-0

1ª Edição - 10/2015 - SCRPardo / SP

Impresso no Brasil

Dedico esta obra a Deus, em forma de agradecimento, por me proporcionar uma nova oportunidade de vida e de trabalho junto das pessoas deste planeta Terra.

A.H.R.

"Ninguém sabe o potencial deste sistema poderoso; algum dia poderá executar uma música, compor sinfonias e projetos gráficos complexos."

Ada Lovelace

Prefácio

Este livro foi pensado em proporcionar aos leitores uma visão geral de como programar utilizando as linguagens C e C++. Muitos livros ensinam apenas C ou somente C++, neste procuramos contemplar as duas linguagens. Para quem não tem experiência em programação este livro tem o necessário para introduzi-lo na arte de programar.

Utilizamos um compilador que faz parte do Microsoft Visual Studio, versão 2013. Para estudantes o download deste software poderá ser realizado direto do site da Microsoft.

Nos capítulos 1 e 2 damos as noções básicas de um programa e como compilá-lo transformando-o em linguagem de máquina.

A partir do capítulo 3 ao 5 continuamos a apresentar noções básicas de programação de computadores, como o uso de operadores, variáveis, entrada e saída de dados.

No Capítulo 6 vamos aprender sobre desvios condicionais, acompanhado do capítulo 7 que vai tratar dos operadores lógicos.

Os laços de repetição são tratados no capítulo 8.

Dos capítulos 9 ao 10 vamos tratar de técnicas de programação, como os vetores, matrizes e funções.

No capítulo 11 aprendemos sobre ponteiros, no 12 as bibliotecas de funções da linguagem e no capítulo 13 as estruturas chamada: "struct".

Terminamos no capítulo 14, escrevendo sobre o pré-processador e no 15 sobre os arquivos em disco.

Tenho certeza que o iniciante no mundo da programação terá uma ferramenta poderosa de estudo em suas mãos com este livro.

Desejo a todos bons estudos.

Antônio Henrique Reis

Sumário

Lista de Siglas e Abreviaturas

ABNT	*Associação Brasileira de Normas Técnicas.*
ANSI	*American National Standards Institute.*
ASCII	*American Standard Code for Information Interchange.*
BCPL	*Basic Combined Programming Language.*
IDE	*Integrated Development Environment.*
I/O	*Input / Output.*
OOP	*Object-Oriented Programming - Programação Orientada a Objetos.*
VS	*Visual Studio Express 2013 for Desktop.*

1
Introdução

1. Introdução

Nos laboratórios da empresa americana: "**Bell Labs**", nos anos 70, **Dennis M. Ritchie** e **Ken Thompson** criaram a **linguagem C**. Com origem na linguagem B de Thompson, que foi uma evolução da linguagem BCPL. B era a primeira letra de BCPL e C a segunda, portanto os autores acharam que seria lógico chamar a linguagem de C.

A estrutura modular da **linguagem C** resulta em programas mais legíveis e documentados. Quando as técnicas de OOP (Object-Oriented Programming - Programação Orientada a Objetos) são utilizadas, se faz o uso da **linguagem C++**.

A **linguagem C** é utilizada no desenvolvimento de sistemas operacionais e em diversos tipos de aplicações. Tem influência direta nas linguagens C++, Java, C#, PHP e JavaScript.

Esta linguagem é reconhecida entre os programadores como uma linguagem de "alto/baixo nível", ou seja, é possível trabalhar como uma linguagem próxima da linguagem humana, fazendo operações do tipo: "a=b+c;" e também com operações de baixo nível, como, por exemplo: ligando ou desligando portas no hardware do computador.

***Nota**: Linguagem de baixo nível é a linguagem de programação que utiliza instruções do processador. Portanto, estão diretamente relacionadas a arquitetura do computador. Um exemplo desta linguagem é o Assembly. A linguagem de alto nível distancia-se da arquitetura do computador e fica mais próxima da linguagem humana. Não há necessidade de se conhecer as características do processador.*

Como a utilização da **linguagem C** tornou-se muito popular foi criado um padrão para a linguagem, chamado de: "**Padrão ANSI**". É chamado desta maneira porque "**ANSI**" é uma sigla que representa: "**American National Standards Institute**", algo parecido com a sigla: "**ABNT**", que no Brasil significa: "**Associação Brasileira de Normas Técnicas**".

Antes do "**Padrão ANSI**" a única referência para a **linguagem C** era o livro: "**Linguagem de Programação C"**, de **Brian Kernighan** e **Dennis Ritchie**. Este livro causou algumas divergências entre compiladores da **linguagem C**, então, o comitê ANSI teve que intervir para criar um padrão a ser utilizado por todos eles.

A **linguagem C++** é uma sequência natural da linguagem C.

A **linguagem C** utiliza a técnica da programação estruturada, mas C++ agrega a técnica da programação orientada a objetos.

No início dos anos 80, **Bjarne Stroustrup**, da "**Bell Labs**", foi quem desenvolveu a **linguagem C++**. Esta linguagem teve como objetivo fazer algumas simulações orientadas a eventos, mas a principal meta foi manter a compatibilidade com a **linguagem C**, para manterem protegidas as milhares de linhas de códigos fonte escritos em linguagem C. Esta é a razão principal de poder utilizar os compiladores atuais para **C++** e utilizar a estrutura da **linguagem C** e tudo funcionar corretamente.

As **linguagens C e C++** são uma das melhores alternativas para entrar no mundo da programação de computadores.

Seja bem-vindo e agora o trabalho será iniciado com programas simples para você ir se acostumando com a sintaxe das **linguagens C/C++**.

1.1. O Primeiro Programa em C/C++

Os estudos serão iniciados com um programa que escreve na tela do computador a frase: "**Meu primeiro programa**". Veja, abaixo, como é o código fonte do programa:

```
main( )
{
   printf("Meu primeiro programa");
}
```

Um programa em C consiste de uma ou mais funções que especificam o que deve ser feito. No exemplo acima **main( )** é uma função. Qualquer programa em C começa com a função **main( )** que marca o ponto inicial da execução do programa.

Todas as instruções devem estar dentro de um bloco de comandos envoltos em uma chave que abre "{" e uma chave que fecha "}".

A linha: **printf ("Meu primeiro programa")**; é uma chamada a função **printf**, com o argumento "**Meu primeiro programa**". Esta sequência de caracteres entre aspas é chamada de: **cadeia de caracteres**, ou também **string**. A palavra **printf** é uma função de biblioteca que escreve o argumento na tela do seu computador. Os parênteses devem estar presentes mesmo quando não há argumentos.

As instruções C são sempre encerradas por um **ponto-e-vírgula (;)**.

Em **linguagem C++** o mesmo programa poderia ter sido escrito da seguinte forma:

```
main( )
{
   cout << "Meu primeiro programa";
}
```

A única diferença é que foi utilizada a forma de escrever na tela usando o fluxo de saída: "cout" que escreve na tela a frase que está entre aspas: "Meu primeiro programa".

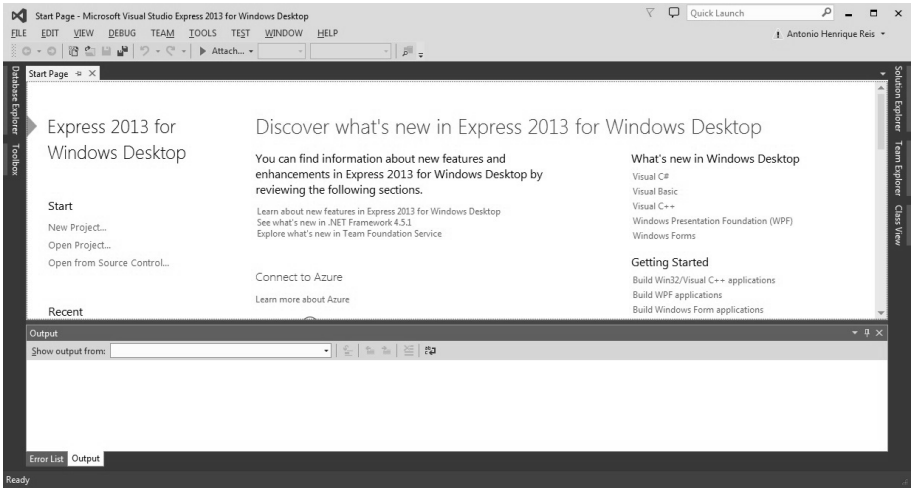
1.2. Ambiente de Desenvolvimento do Microsoft Visual Studio

Para escrever os programas deste livro, foi escolhido o ambiente de desenvolvimento da **Microsoft®**, o **Visual Studio C++ Express 2013 for Desktop**, que pode ser acessado e instalado direto do site da **Microsoft**. Você deve verificar a forma de utilização do mesmo para seus estudos.

Este sistema tem uma interface **IDE** (Integrated Development Environment), ou seja: um ambiente integrado de desenvolvimento de software, bastante amigável na qual é possível escrever o código fonte, compilar, executar e testar o programa no mesmo local.

Para acessar o programa depois de instalado, conforme instruções do site da **Microsoft**, proceda da seguinte forma:

1. Clique no botão **Iniciar** ()do **Windows**;
2. Clique em **Todos os Programas**, localize e clique na pasta: **Visual Studio 2013**, e em seguida, em: **VS Express 2013 for Desktop**;
3. Aguarde a inicialização do programa.



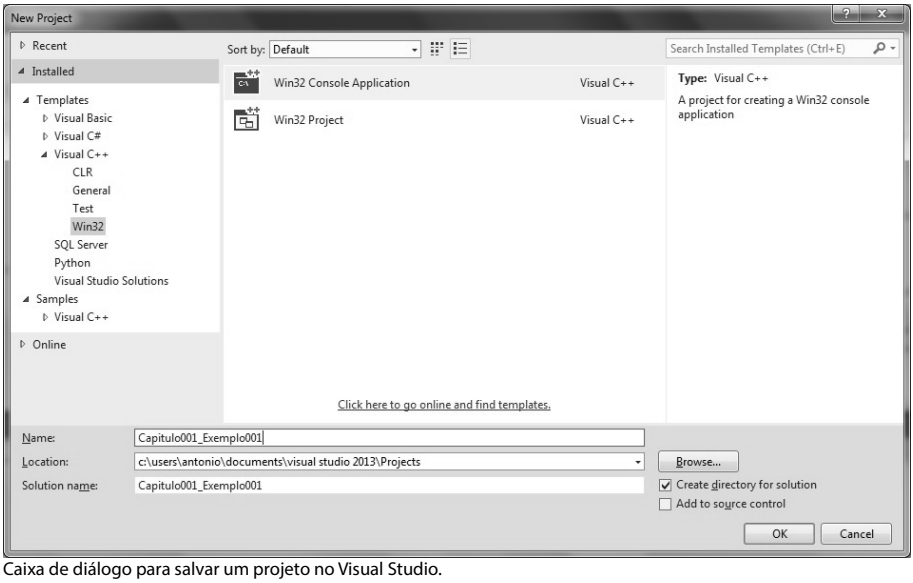
Tela inicial do Microsoft Visual Studio.

Esta é a tela inicial do **Visual Studio**, onde encontra-se o acesso aos menus e a interface de desenvolvimento de software. No ambiente profissional do **Microsoft Visual Studio** é possível desenvolver programas em várias linguagens como C, C++, C# (pronuncia-se C Sharp), Asp.net e também o Visual Basic.

1.3. Editando o Programa

Inicialmente é necessário preparar o ambiente de trabalho do **Visual Studio** para que seja possível desenvolver programas utilizando as **linguagem C/C++**. Para preparar o ambiente de trabalho, proceda da seguinte forma:

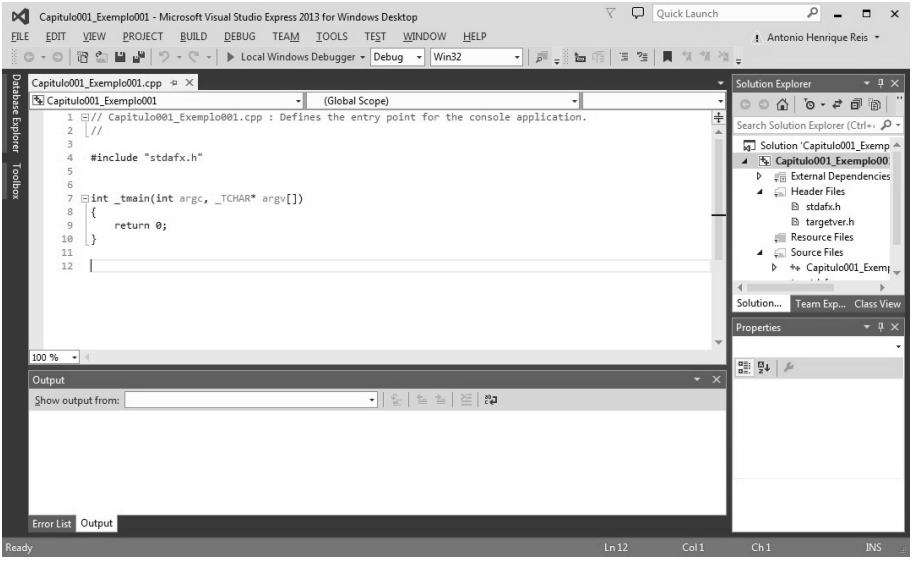
1. Na tela inicial do **Visual Studio**, acesse o menu: **FILE** e, em seguida, clique na opção **New Project (Ctrl+Shift+N)**;
2. A caixa de diálogo **New Project** será exibida:



Caixa de diálogo para salvar um projeto no Visual Studio.

3. Escolha o tipo de projeto que será desenvolvido;
4. No lado esquerdo da caixa de díalogo, escolha a opção **Visual C++** e, em seguida, clique na opção **Win32**;
5. Ao centro da janela, clique em **Win32 Console Application**, que é o tipo de programa que será desenvolvido;
6. Defina um nome para o projeto, na caixa de textos **Name**;
7. Verifique se na caixa de textos **Location** está definido o caminho da pasta padrão do projeto, conforme a imagem anterior;
8. Clique no botão **OK** e na próxima tela, clique no botão **Finish**.

Após realizar estes passos o ambiente de trabalho deverá ficar semelhante à imagem a seguir:



Janela de codificação do Visual Studio.

Note que agora o **Visual Studio** já está pronto para trabalhar com o código fonte do projeto do primeiro programa que será desenvolvido.

A partir deste momento será utilizado apenas o código fonte para referência da **linguagem C/C++**. A partir de agora, o ambiente do **VS Express 2013 for Desktop** será referido apenas como **VS**.

1.4. Verificando o Código Fonte C Utilizando o VS

Observe com maiores detalhes o código apresentado na imagem anterior:

```
// Capitulo001_Exemplo001.cpp : Defines the entry point for the
console application.
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
   return 0;
}
```

O **VS** traz várias linhas já preenchidas, basta digitar os comandos dentro das chaves da função principal **main**, que no **VS** está identificada como **_tmain**.

As linhas que começam pelos caracteres das barras "//", são comentários de final de linha e o compilador não irá conferir estas linhas.

A linha escrita: **#include "stdafx.h"** irá incluir um cabeçalho com informações da biblioteca padrão para os programas em C. É uma diretiva do pré-processamento.

Arquivos com sobrenome: "**.h**", são identificados como "**headers**", ou seja, uma inclusão de um cabeçalho com informações para funções que serão utilizadas no programa.

Note que a esquerda da "**_tmain**" o termo "**int**", informa que haverá um retorno de um valor inteiro, o que é possível verificar pela linha escrita: "**return 0;**".

Dentro dos parênteses na **_tmain**, há uma lista de parâmetros que poderão ser recebidos pela função (**int argc, _TCHAR* argv**[]).

A partir deste momento os comandos serão escritos dentro das chaves da função " **_tmain**", se você estiver utilizando outra interface **IDE**, basta fazer os ajustes necessários para o seu ambiente de desenvolvimento.

O padrão adotado para o desenvolvimento de todo o projeto, é o **ANSI (**American National Standards Institute), portanto independente da **IDE** utilizada os comandos que foram inseridos dentro das chaves, irão funcionar em todas as **IDE's**.

1.5. Apresentando uma String na Tela

Será feita agora a inclusão do comando que vai apresentar na tela a string com a frase "**Meu primeiro programa**".

```
// Capitulo001
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
   printf("Meu primeiro programa");
   return 0;
}
```

O comando que vai fazer esta tarefa é a chamada da função **printf** escrito na linha:

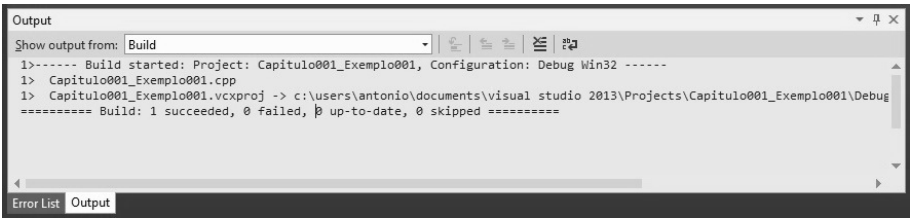
```
   printf("Meu primeiro programa");
```

1.6. Compilando o Programa

Para compilar o programa desenvolvido, proceda da seguinte maneira:

1. Na **Barra de Menus**, acesse o menu **BUILD** e, em seguida, clique em **Build Solution (F7)**;
2. O compilador irá verificar se a sintaxe do código fonte está correta.

É possível verificar se não há erros no código, por meio da janela **Output**, localizada na parte inferior da tela, conforme a imagem a seguir:

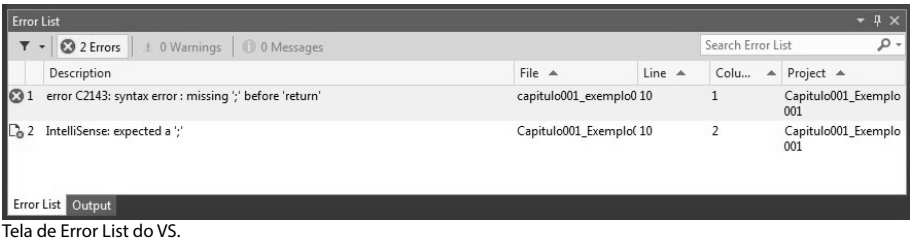


Tela de OutPut do VS.

Repare na última linha a frase: "**Build: 1 succeeded, 0 failed, ...**" onde fica possível identificar que o código foi compilado com sucesso sem nenhuma falha.

Se houver algum erro de sintaxe a janela: **Error List** irá aparecer. Nesse caso, basta dar dois cliques no primeiro erro encontrado, interpretar o que diz na janela sobre este erro e corrigí-lo.

Veja na figura a seguir a janela de erros:



Tela de Error List do VS.

Deve-se sempre corrigir o primeiro erro e depois compilar novamente o programa, pois um erro na **linguagem C/C++** poderá levar a ocorrência de muitos erros.

Nesse caso, foi retirado propositalmente o ponto-e-vírgula (";") de uma das linhas, forçando o aparecimento de erros.

1.7. Executando o Programa

Depois de corrigir todos os erros de sintaxe, o programa deve ser executado. Basta acessar o menu **DEBUG**, e clicar no comando **Start Debugging (F5)**.

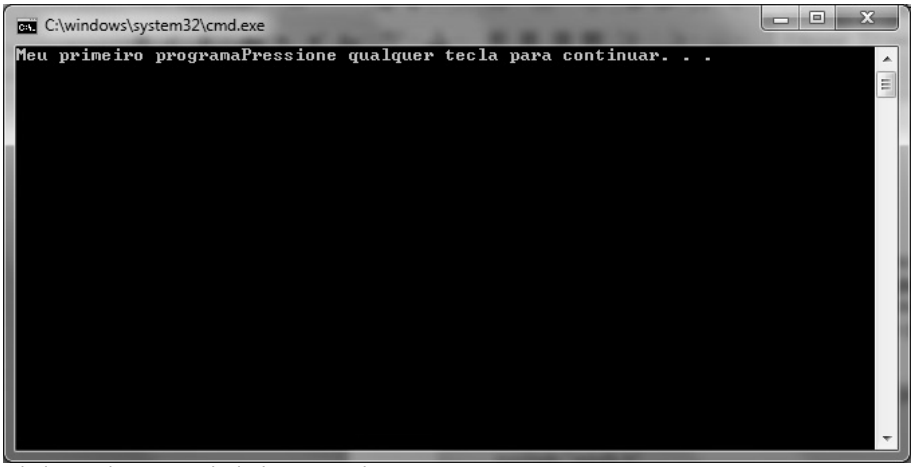
Você vai perceber que o programa irá mostrar uma janela com a mensagem e logo desaparecerá. Para fazer uma pausa na tela até que uma tecla seja pressionada, deve-se colocar uma linha de comando para esta tarefa, conforme o exemplo a seguir:

```
// Capitulo001
//

#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
   printf("Meu primeiro programa");

   system("PAUSE");
   return 0;
}
```

Foi inserida a diretiva: **#include "stdlib.h"**, que instrui o pré-processamento para incluir um arquivo no programa com as definições para o uso do comando: "**system("PAUSE");**" que irá fazer uma pausa na tela e esperar uma tecla ser pressionada.

Depois de compilado, o programa foi novamente executado por meio do comando **Start Debugging (F5)**, localizado no menu **DEBUG**.

Veja o resultado, após a execução do comando:



Tela de console com o resultado da execução de um programa.

Repare que a mensagem "**Pressione qualquer tecla para continuar. . .**" foi apresentada como resposta do comando **system("PAUSE")**, mas ela apareceu junto da mensagem "**Meu primeiro programa**". Para que a mesma fosse apresentada na linha de baixo seria necessario adicionar ao código, um comando para mudar de linha, que é o "**\n**". Veja o exemplo no código fonte a seguir:

```
// Capitulo001
//
#include "stdafx.h"
#include "stdlib.h"

int _tmain(int argc, _TCHAR* argv[])
{
   printf("Meu primeiro programa \n");

   system("PAUSE");
   return 0;
}
```