

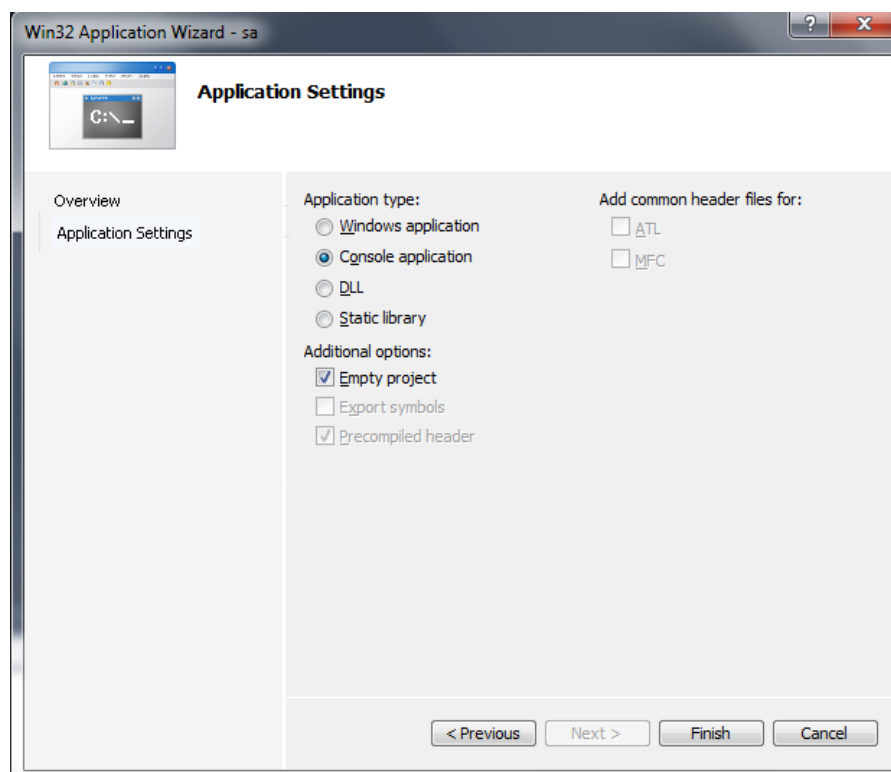
# Introdução ao Visual Studio como Plataforma de Desenvolvimento

## Introdução

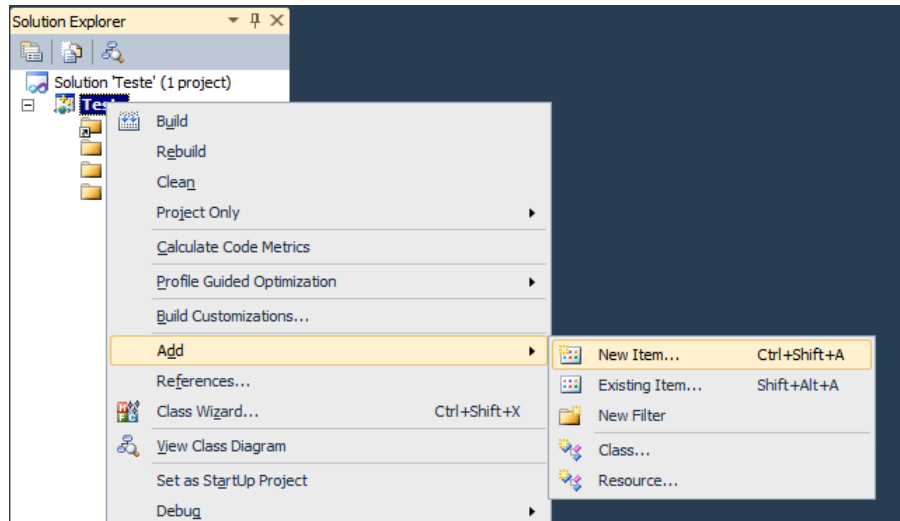
O presente documento pretende introduzir um dos possíveis ambientes de desenvolvimento (*IDE – Integrated Development Environment*) a utilizar na unidade curricular de Programação, do MIEIC, o Visual Studio (versão de 2010) para o sistema operativo Microsoft Windows. As seguintes secções descrevem o processo de criação de um pequeno projecto, execução e depuração do programa resultante, utilizando este ambiente.

## 1. Criação de um projecto

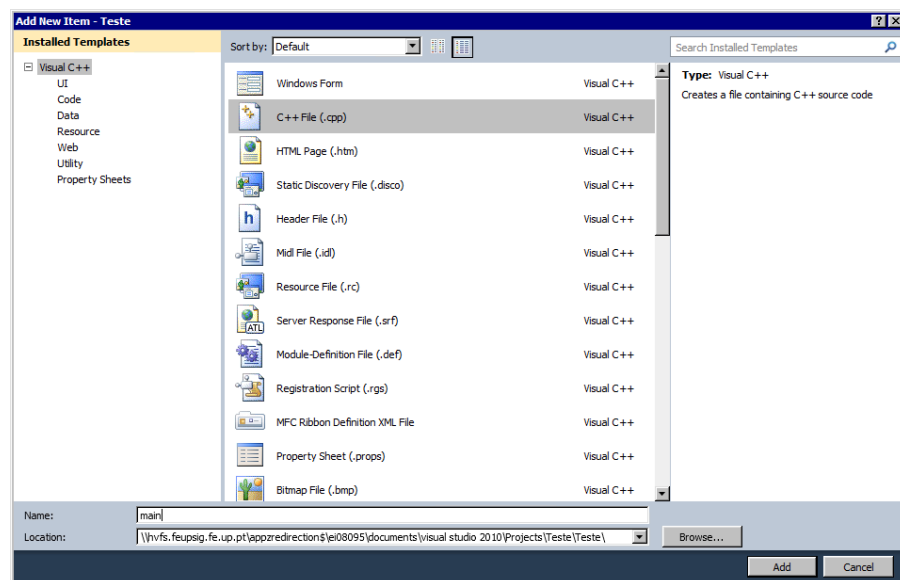
1. Arranque a plataforma Visual Studio e invoque o comando do menu **File > New > Project**.
2. Escolha o *template* Visual C++ e um projecto do tipo **Win32 > Win32 Console Application**.
3. No campo **Name**, introduza o nome do projecto, e em **Location** escolha o directório onde este será guardado.
4. De seguida aparecerá um *Wizard* para finalizar a criação do projecto, no qual deverá seleccionar a opção **Empty Project**, na secção **Additional Options**, e clicar no botão **Finish**.



5. Após a criação do projecto, crie um ficheiro para a escrita do código do programa. Use o botão direito do rato sobre o nome do projecto e seleccione o comando **Add > New Item**, de acordo com o representado na figura. Se, em vez de criar um novo ficheiro, pretender importar um já existente deve escolher a opção **Add > Existing Item** e seleccionar o ficheiro em questão.



6. Escolha o tipo de ficheiros que pretende inserir, neste caso **C++ File (.cpp)** e atribua um nome ao novo ficheiro, por exemplo, **main.cpp**. Depois de adicionado, este ficheiro passará a estar disponível na pasta **Source Resources** do projecto.



7. No ficheiro **main.cpp** escreva o seguinte código:

```
#include <iostream>
using namespace std;

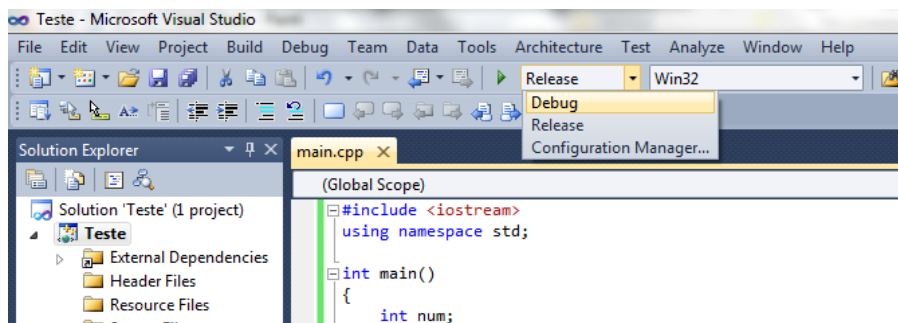
int main()
{
    int num;
    cout << "Insira um número entre (2-10): ";
```

```
cin >> num;
cout << "Tabuada dos " << num << endl;
for (int i = 1; i <= 10; i++)
{
    cout << num << " x " << i << " = " << num * i << endl;
}
return 0;
}
```

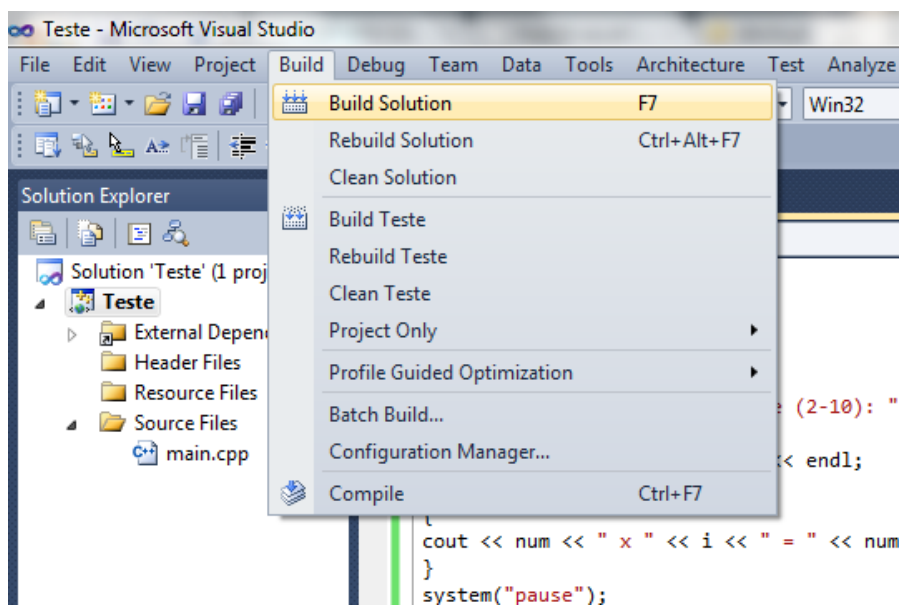
Este programa mostra a tabuada da multiplicação de um número fornecido pelo utilizador. Grave o ficheiro, teclando **Ctrl-S** ou seleccionando **File > Save**.

## 2. Compilação e execução do programa

1. O programa pode ser compilado em dois modos, *Debug* ou *Release*; os executáveis gerados no modo *Release* ocupam menos espaço que os gerados no modo *Debug*.



2. O programa pode ser compilado invocando o comando de menu **Build > Build Solution** ou premindo a tecla **F7**.



3. O programa tanto pode ser executado a partir do *IDE* como a partir do *Command Prompt* do Windows. Para executar a partir do *IDE* basta teclar **Ctrl-F5** ou invocar o comando de menu **Debug > Start Without Debugging**. Neste caso, o programa também é compilado antes de ser executado.

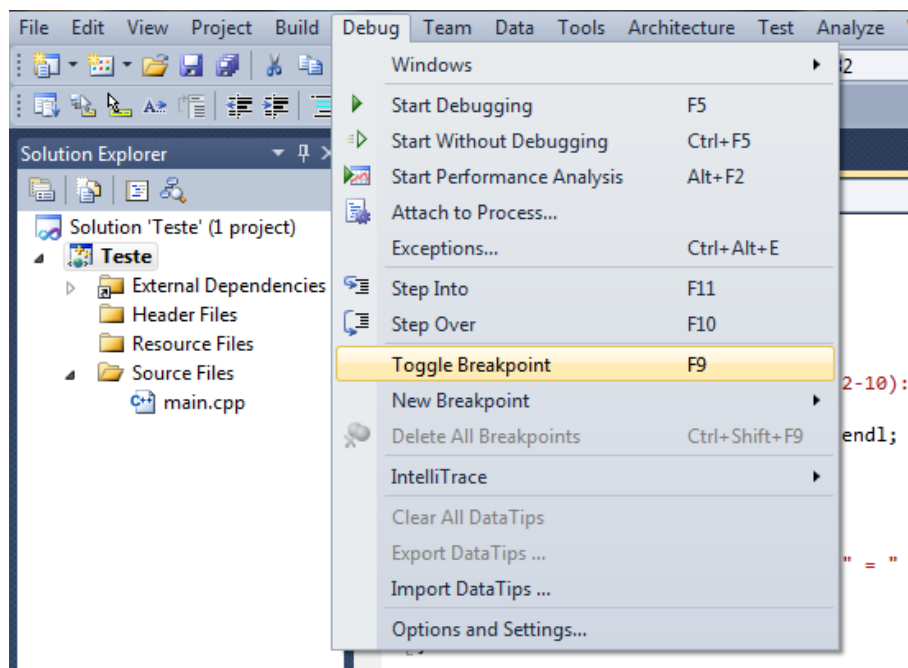
Para executar a partir do *Command Prompt*, é necessário, depois de abrir o *Command Prompt*, mudar o directório corrente para aquele onde está o executável, através de um comando **cd C:<directório\_doprojecto>\<Modo\_de\_compilação>**, por exemplo, “cd C:\Pastal\ProjectoTeste\Release”, dando de seguida o comando de execução que consiste em escrever o nome do ficheiro executável (ProjectoTeste, no caso do exemplo).

4. Por vezes, quando se executa o programa a partir do *IDE* a janela da consola fecha automaticamente, sem que seja possível verificar qual a saída que o programa gerou. Para evitar esta situação, faça o seguinte: no menu seleccione **Project > project\_name properties ... > Configuration Properties > Linker > System** e configure a propriedade **SubSystem** para **Console (/SUBSYSTEM:CONSOLE)**.

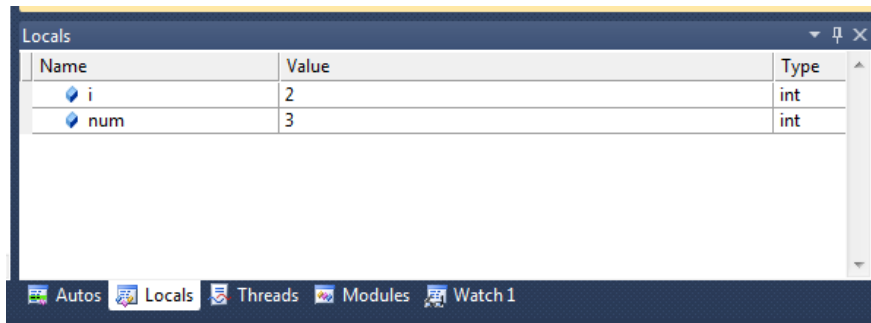
### 3. Depuração do programa

Depurar um programa consiste em controlar e acompanhar a sua execução, investigando possíveis casos de erros.

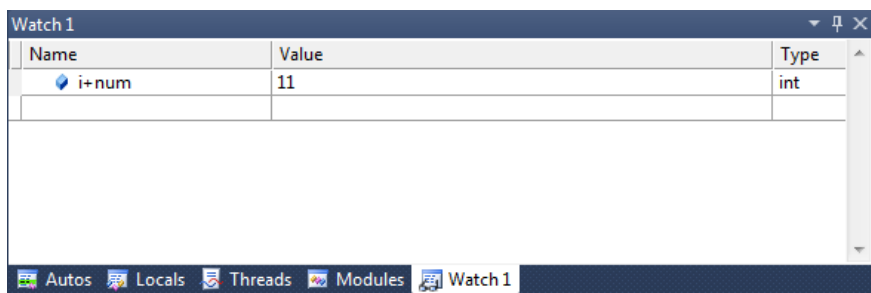
1. Para se poder controlar a execução do programa deve-se adicionar um *Breakpoint*, colocando o cursor na linha em que se pretende interromper temporariamente a execução do programa e seleccionando o comando **Debug > Toggle Breakpoint** (em alternativa, premir **F9**).



2. Seleccione o comando **Debug > Start Debugging** (em alternativa, premir a tecla **F5**), para executar o programa no modo de depuração.
3. Poderá acompanhar o progresso dos valores das variáveis locais no separador **Locals**, presente na base da janela (caso este separador não esteja visível, pode-se torná-lo visível utilizando o comando **Debug > Windows > Locals**). Ao longo da execução no modo de depuração, poderá analisar e alterar o valor das variáveis.



4. No separador **Watch** (caso não apareça, executar o comando **Debug > Windows > Watch > Watch1**, por exemplo) poderá acompanhar o progresso de expressões que não tenham sido explicitamente criadas como variáveis locais do programa. Para criar uma expressão deste tipo basta clicar num campo disponível e escrever a expressão que pretende avaliar.



5. Poderá controlar a execução do programa com as opções presentes no menu **Debug**:

**Step Into** – permite acompanhar a depuração das funções chamadas;  
**Step Over** – não faz a depuração das funções chamadas, apenas as executa e passa imediatamente para a instrução seguinte;  
**Step Out** – caso se tenha feito *Step Into* a uma função e não se pretenda acompanhar o resto da depuração, esta opção permite correr o que sobra da função e voltar para o procedimento que a chamou;  
**Continue** – executa o que resta da função;  
**Stop Debugging** – termina a depuração do programa.

*Joel Ferreira, Rui Fernandes*  
HLC, JAS, RCS