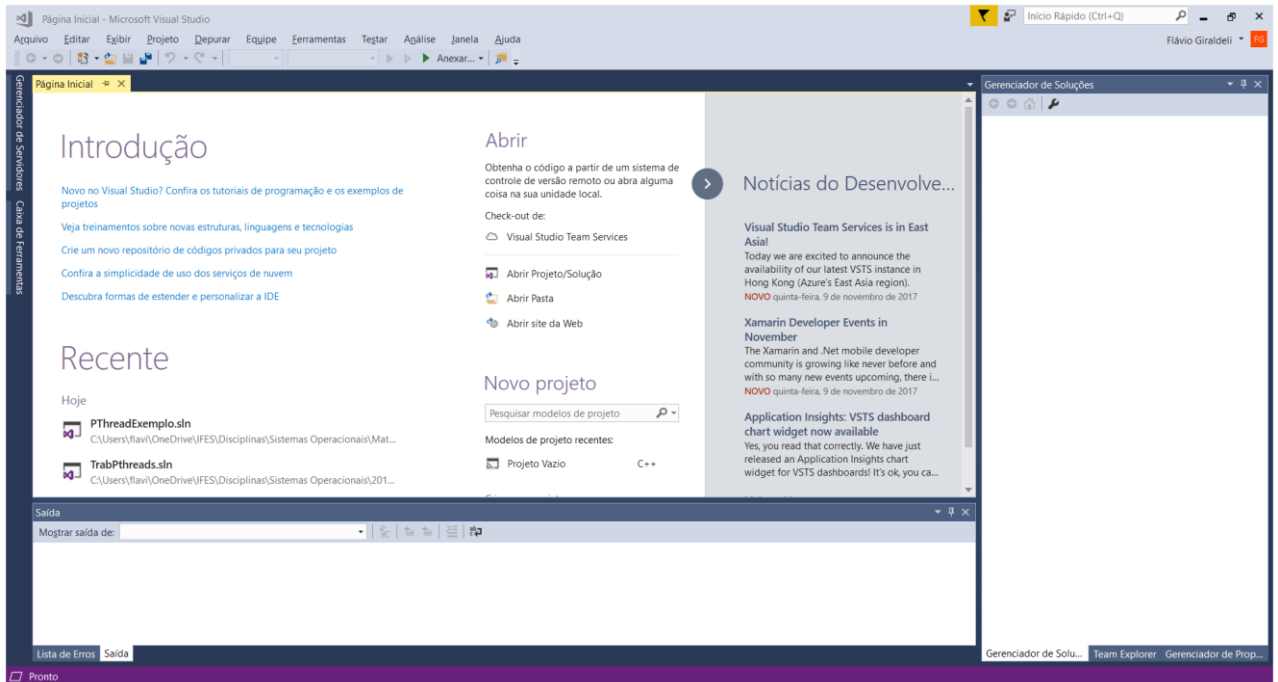


MICROSOFT VISUAL STUDIO 2017

ROTEIRO DE USO

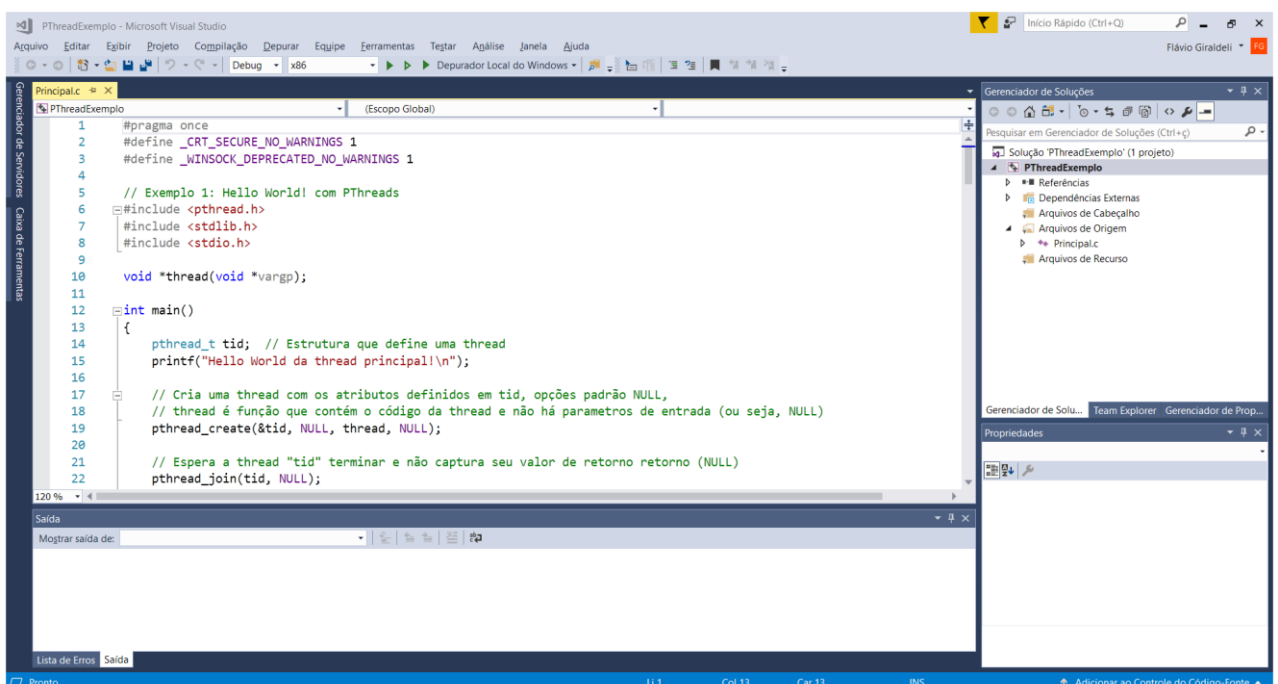
INICIANDO E CONFIGURANDO O AMBIENTE

Após instalar e iniciar o VS 2017 pela primeira vez, você deverá ver um ambiente semelhante ao abaixo (caso você tenha instalado a versão em Português, claro):



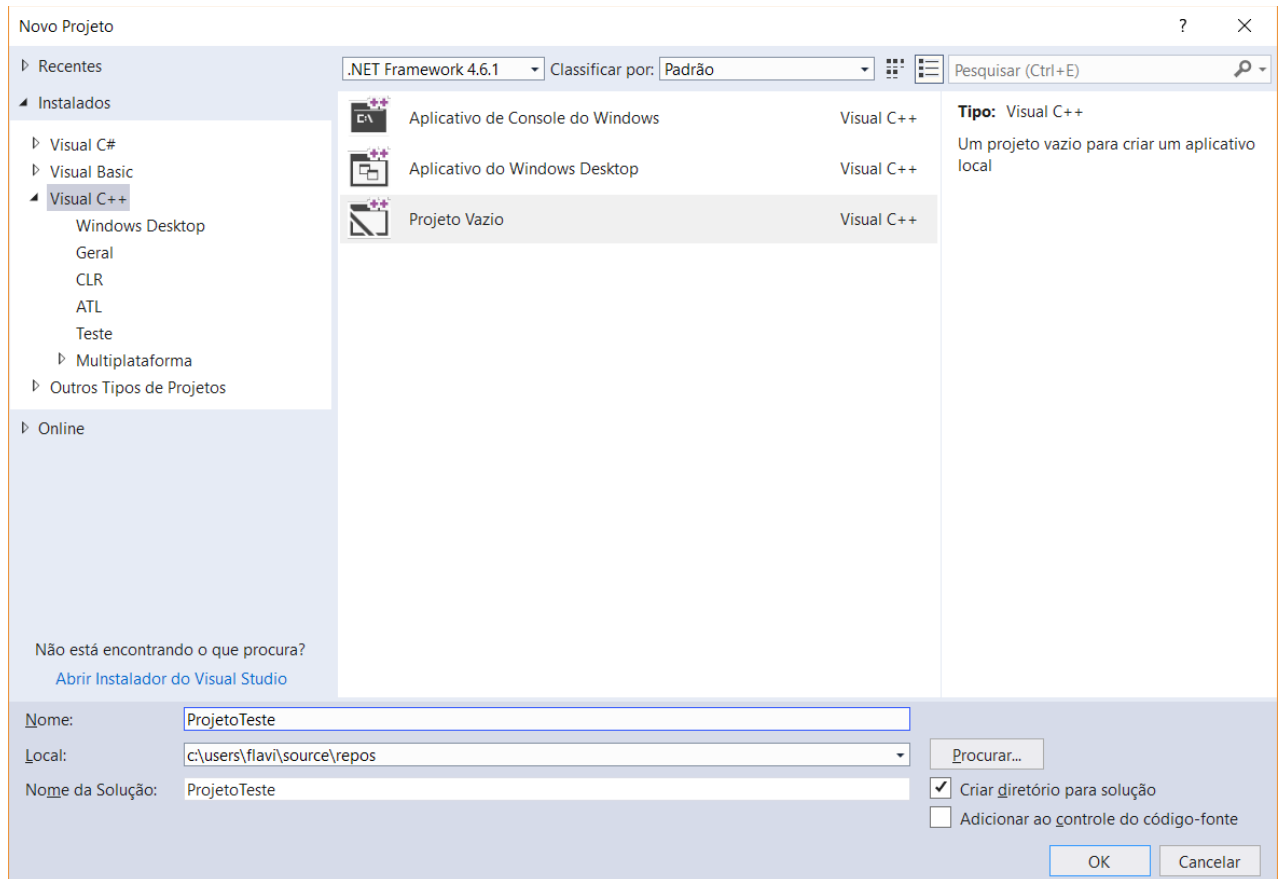
Ao clicar no X sobre a guia da aba **Página Inicial**, você pode “limpar” o ambiente.

Observe se na área à esquerda (ou à esquerda, dependendo de como está o ambiente) você consegue visualizar o **Gerenciador de Soluções**, pois você precisará dele. Caso não esteja sendo exibido, clique no menu **Exibir > Gerenciador de Soluções**.



CRIANDO UM PROJETO

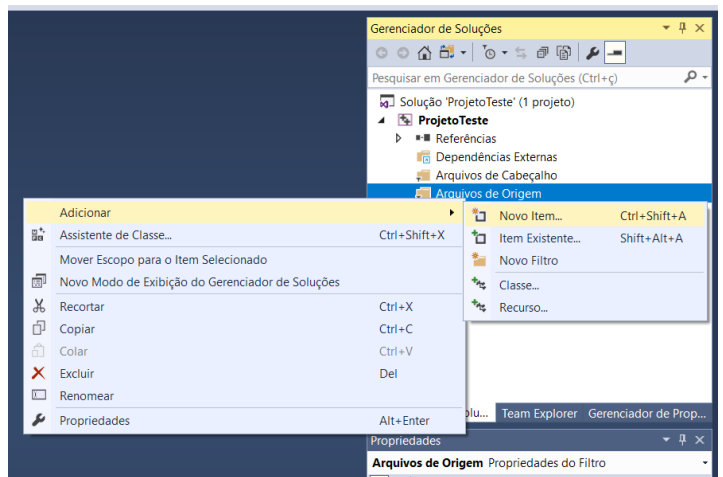
Você deverá criar corretamente um projeto antes de escrever qualquer programa em C. Para tanto, vá ao menu **Arquivo > Novo > Projeto**. Clique o termo **Visual C++** à esquerda e você verá a seguinte tela:



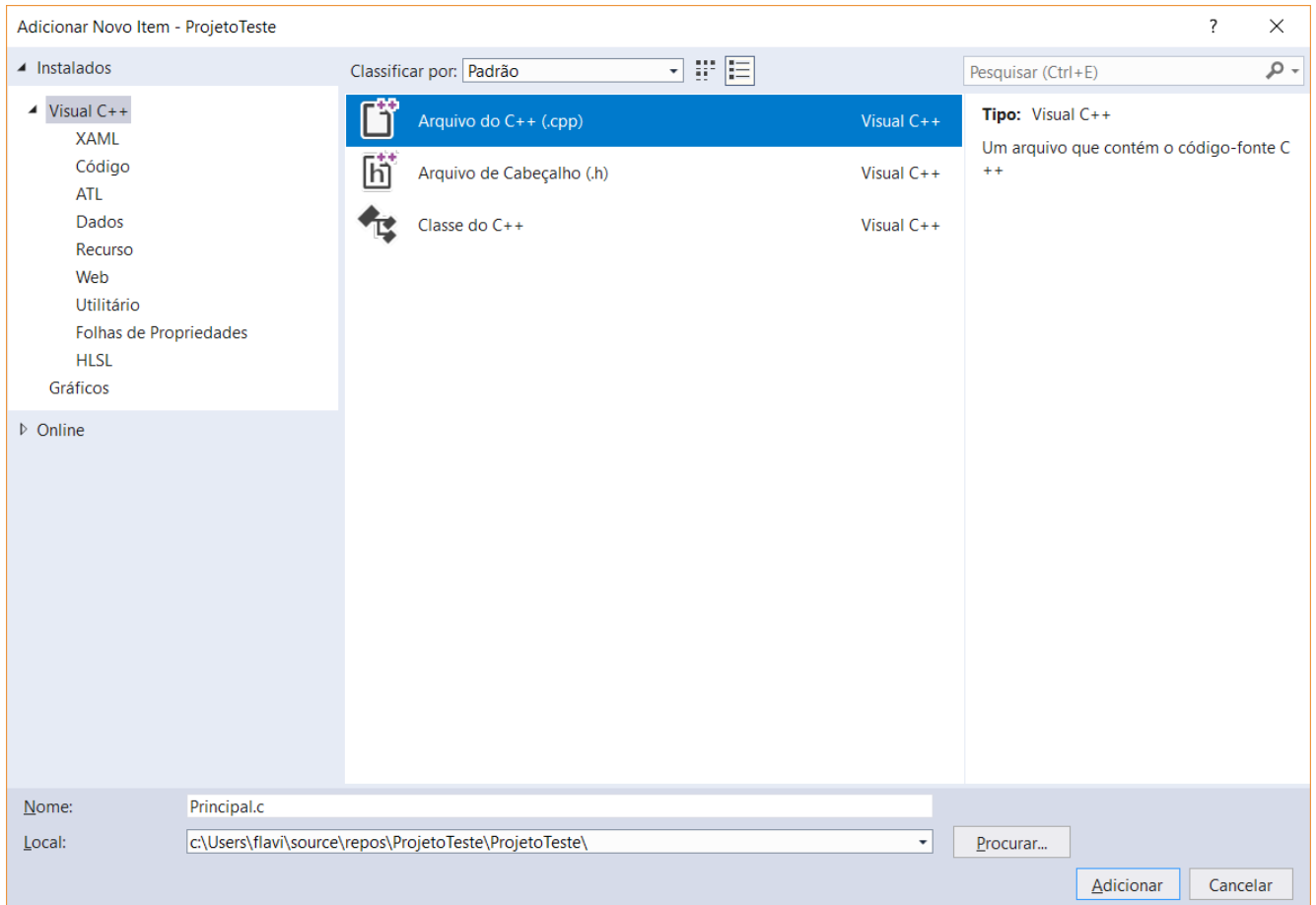
Marque **Projeto Vazio**, à direita, em seguida preencha o campo **Nome**, com o nome do projeto (este nome será o nome do seu executável, ao final), conforme exemplo acima.

Observe a localização dos arquivos do projeto. Caso não lhe agrade, basta clicar em **Procurar...** e selecionar outra pasta. Em seguida, dê **OK**.

Observe que em **Gerenciador de Soluções**, agora tem-se algumas pastas vinculadas ao projeto. Neste momento, interessa-lhe apenas a pasta **Arquivos de Origem**, onde ficarão localizados seus arquivos-fonte. Clique com o botão direito sobre esta pasta e selecione **Adicionar > Novo Item...**



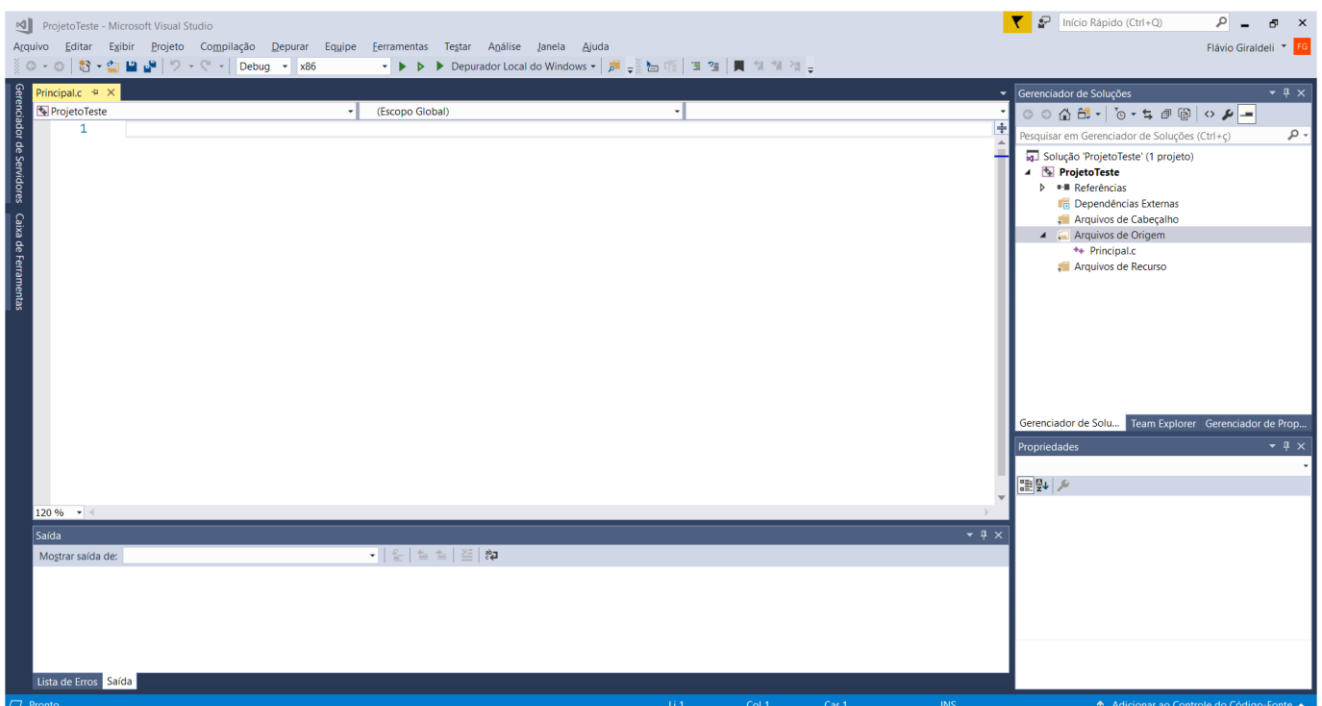
Ao clicar sobre o termo **Visual C++** à esquerda e em seguida sobre **Arquivo do C++ (.cpp)** à direita, você verá a seguinte tela:



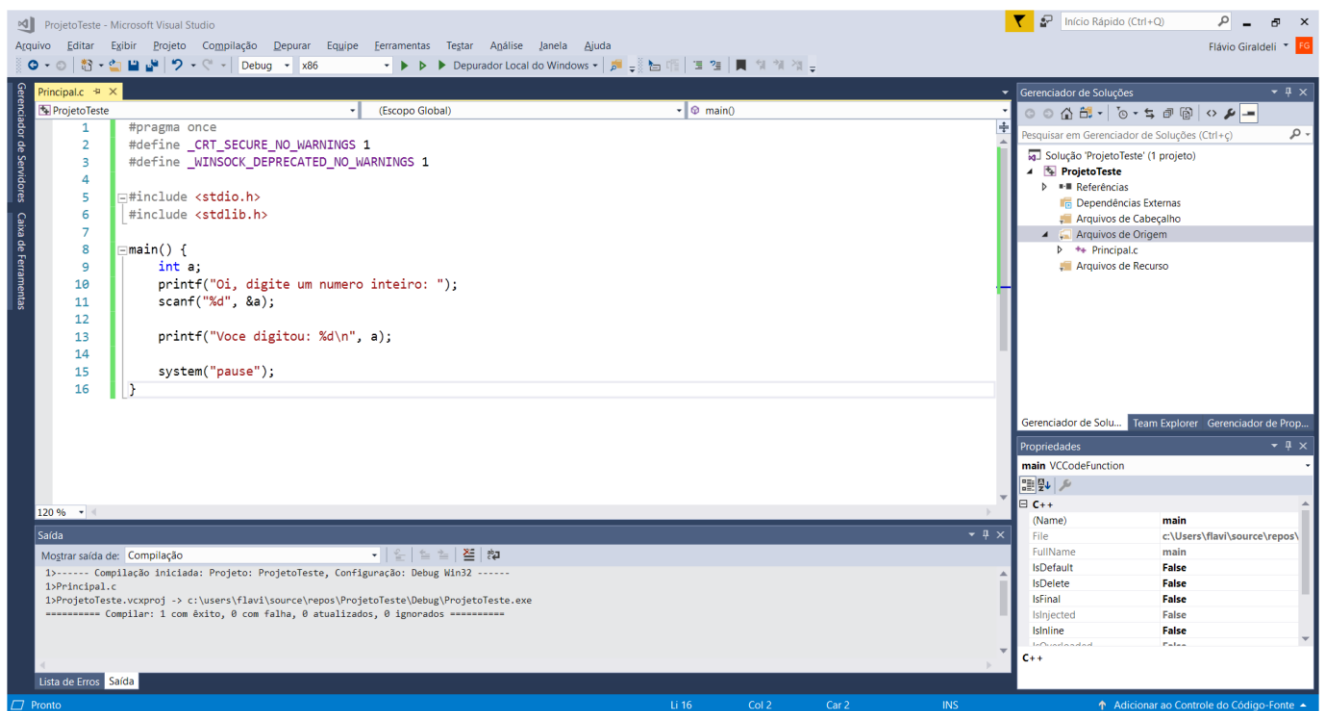
Agora, dê um nome para o arquivo **lembrando-se de acrescentar manualmente a extensão .c ao final do nome**, caso contrário, o ambiente poderá reportar erros na compilação, conforme exemplo acima.

Agora, clique em **Adicionar**.

Aparecerá, portanto, a seguinte tela, com o editor pronto para receber o código em C que você criará:



Como um exemplo, abaixo, criei um pequeno exemplo abaixo.



COMPILANDO E RODANDO O PROGRAMA

Nosso próximo passo é compilar o programa. Para tanto, clique em **Compilação > Compilar Solução**. Observe a sub-janela **Saída** (parte inferior do ambiente), ela exibe o relatório do processo de compilação. Se esta janela não estiver habilitada, basta habilitá-la em **Exibir > Saída**. Se tudo der certo, você terá na linha de resumo **Compilar: 1 com êxito, ...** conforme imagem anterior.

A princípio, ignore eventuais **Warnings (Avisos)**, eles não impedirão que seu projeto compile. Mas **erros (erros)** não podem ser ignorados, pois eles impedem a compilação do projeto. Erros serão mostrados na janela de Saída e farão referência a linha em que ocorreram, entre parênteses após o nome do arquivo, e em seguida a mensagem de erro, indicando o problema (Geralmente, falhas na sintaxe). É importante que você aprenda a interpretar as mensagens de erro do compilador. O compilador da Microsoft é um dos que exibe as mensagens mais claramente.

Erros em *tempo de compilação* são geralmente fáceis de serem diagnosticados e corrigidos, ao contrário dos erros que podem vir a ocorrer em *tempo de execução*. Estes últimos demandam um bom domínio da ferramenta de **debug (depuração)**, que futuramente poderemos aprender a usar. No entanto, nada impede que desde o princípio você comece a aprender a usá-la. Contudo, esta não será abordada neste pequeno roteiro. Mas nada impede que você aprenda sozinho... é muito simples.

Evite, neste primeiro momento, o uso de atalhos. Eles poderão te confundir em algumas operações que podem parecer parecidas, mas não são.

Caso a compilação tenha ocorrido com sucesso, chegou a vez de rodar o programa.

Para tanto, clique no menu **Depurar > Iniciar sem depurar...**

Se a janela apenas piscar e você não ver nada, basta adicionar o clássico **system("pause");** no final do programa (lembrando de incluir a biblioteca `stdlib.h`).

Se aparecer uma janela solicitando uma nova compilação após o arquivo ter sido alterado, marque a caixa de seleção para que ele evite perguntar a todo momento. Assim ele vai sempre compilar novamente antes de rodar.

Pronto, você pode enfim conferir seu programa rodando!

Maiores informações podem encontradas com o professor... ou no Google.

Bons estudos!

Prof. Flávio Giraldele