

Software a Utilizar nas Aulas de AP

Caso: IDE DevC++

Introdução

Segue-se uma breve apresentação do software a utilizar nas aulas de Algoritmos e Programação. Propõe-se este IDE, mas há muitos outros que poderão ser utilizados.

Para quem for o feliz possuidor de um Mac, pode usar o nativo Xcode (embora, claro, seja muito mais “poderoso” do que aqui se necessita) ou criar uma máquina virtual (com a VirtualBox) em Windows 7, p. ex., e instalar lá o DevC++, passando a utilizá-la nas aulas, como se estivesse a usar um PC Windows.

O DevC++ é bastante prático, de instalação fácil e é parco em utilização de recursos.

Para instalar o DevC++ para Windows, basta procurar no Google por DevC++ e depois fazer o *download* do ficheiro de instalação e proceder da forma usual para instalar (ir respondendo às solicitações feitas pelo instalador). No final da sua instalação iniciar o DevC++.

Na Figura 1, mostra-se o ambiente de programação e, neste caso, já com opções seleccionadas quando se pretende adicionar um novo ficheiro, para escrever um programa em C.

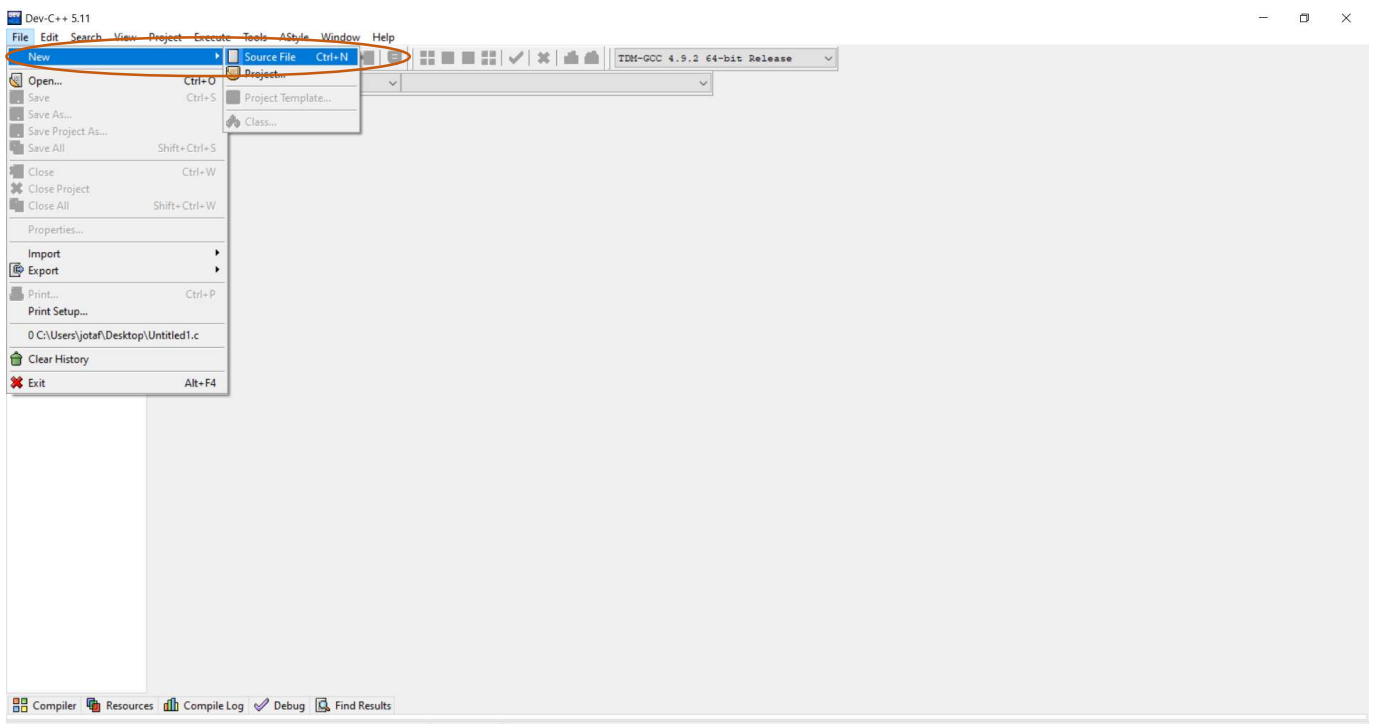


Figura 1 – Form do DevC++, já com opções seleccionadas quando se pretende criar um novo ficheiro.

Iniciar a Utilização do DevC++ para a criar e executar programas

Quando surge a janela onde podemos escrever o programa, insiram o código seguinte:

```
#include<stdio.h>//biblioteca base da linguagem C
#include <locale.h>

int main()
{
    setlocale(LC_ALL, "Portuguese");
    printf("***Bom dia!!***\n\n");
    return 0;
}
```

Os programas devem guardar-se com a extensão “.c” ou seleccionar o tipo, tal como especificado na Figura 2.

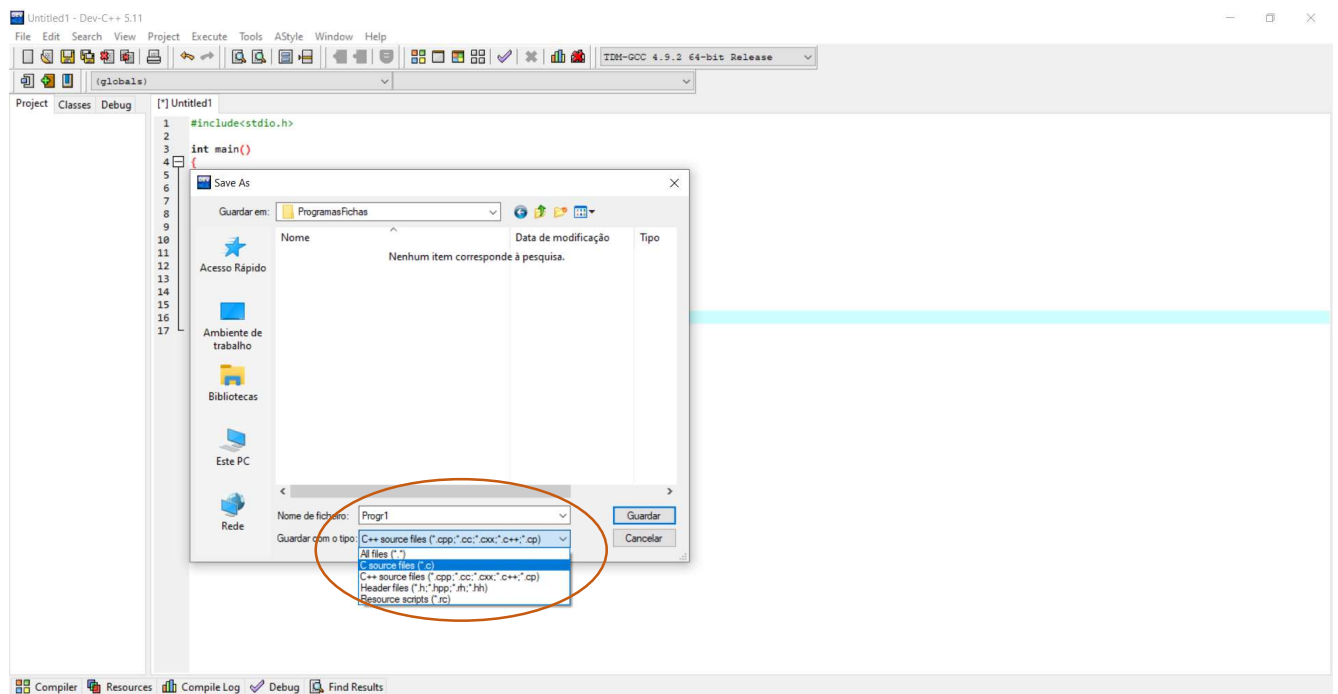


Figura 2 - Guardar o ficheiro em C.

Para compilar e executar o programa, digitar F11 ou selecciona-se *Execute->Compile and Run* (ver Figura 3).

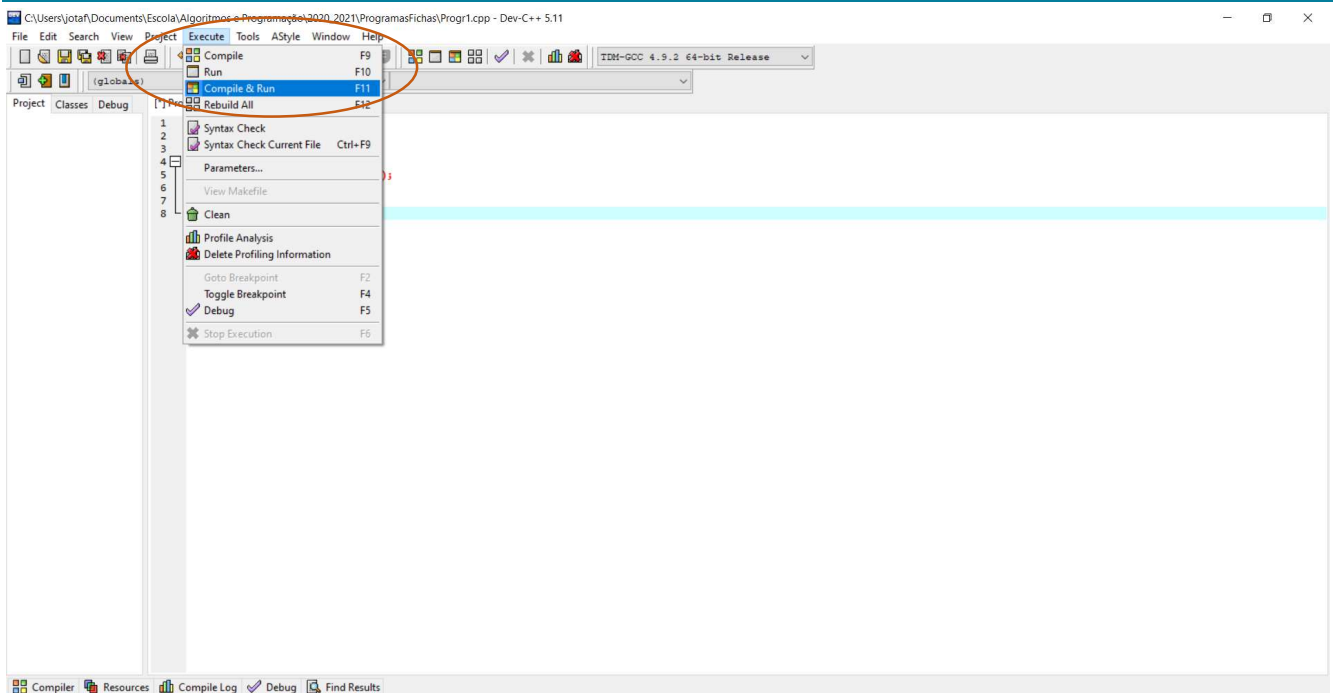


Figura 3 - Compilar e executar.

Se não houver erros, o programa é executado na janela de comandos, tal como é mostrado na Figura 4.

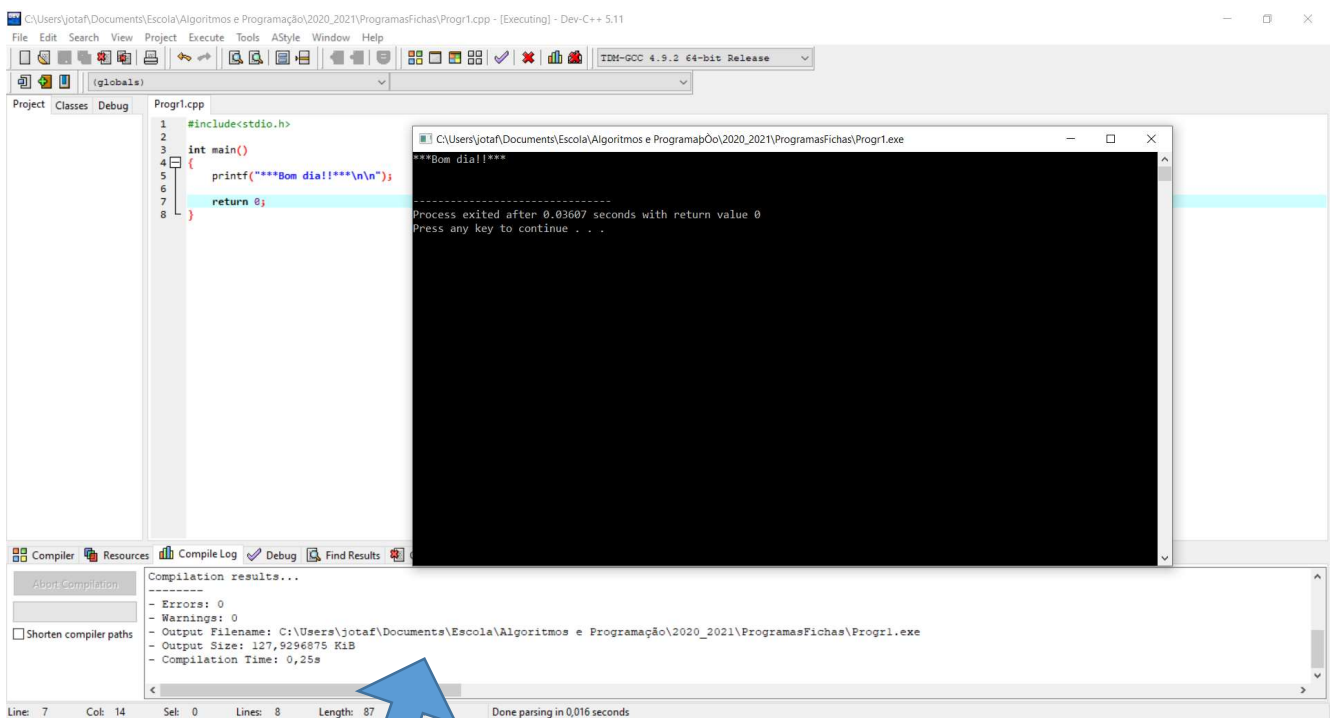


Figura 4 - Execução do programa.

Caso o compilador detectar erros, eles aparecem nesta janela.

Podem agora alterar o programa e acrescentar o código infra, antes da linha `return 0;`

```
int idade;  
float x;  
char nome[20];  
printf("Qual a sua idade? ");  
scanf("%d", &idade);  
printf("Qual o seu nome? ");  
scanf("%s", nome);  
printf("A pessoa tem %d anos e chama-se %s!", idade, nome);
```

Compilem e executem o programa. Interpretem como surge o que é mostrado no ecrã, em resultado da execução do programa, fazendo a correspondência entre cada linha que surge no ecrã e a(s) linha(s) do programa que forma executadas para obter cada linha. Fiquem já com a ideia de que tudo o que vai surgir no ecrã em resultado da execução de um programa se denomina genericamente como “output” ou “saídas” (em Português), embora o termo em inglês seja o mais comumente utilizado .

Acrescentem ainda as seguintes linhas:

```
x=2/3;  
printf("1.ª vez: x=%.2f\n\n", x);  
x=2.0/3;  
printf("2.ª vez: x=%.2f", x);
```

Compilem e executem o programa. Interpretem o que é mostrado no ecrã, em resultado do novo código inserido. O que acham estranho? Justifiquem.

Criação de Projeto com o DevC++

Pode, também, criar-se um projeto (vazio ou Consola), englobando um conjunto de ficheiros com funções¹ e/ou outros componentes².

A criação de projetos é útil para a criação de aplicações mais complexas (englobando muitas funcionalidades), em que é necessário dividir o programa em várias partes. Esta divisão pode ajudar na organização do projeto e/ou quando existem várias pessoas a trabalhar na mesma solução (caso típico em que um grupo de estudantes está a desenvolver, de forma colaborativa, o MiniProjecto Final da UC). A janela correspondente à criação de um novo Projecto apresenta-se na Figura 5.

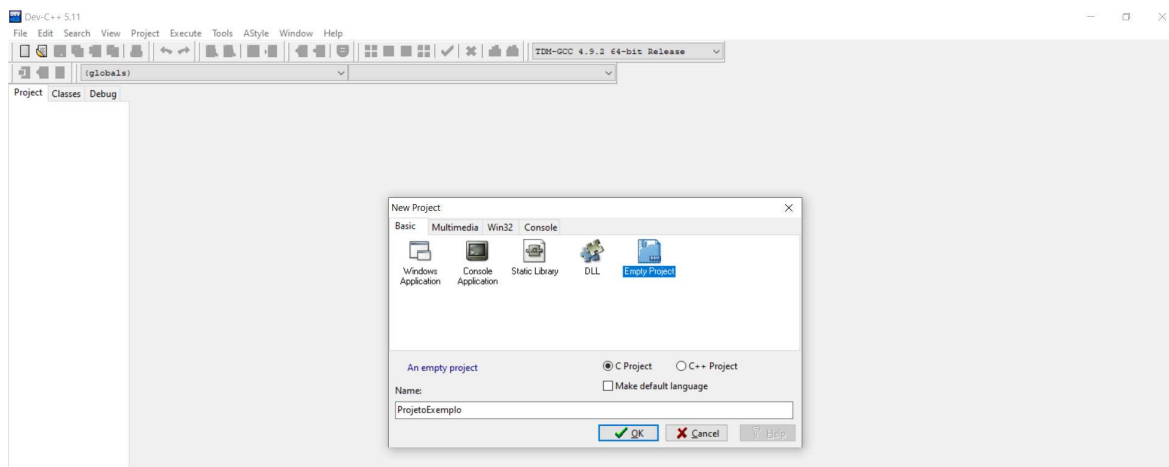


Figura 5 - Criação de novo projeto

Convém criar uma pasta para guardar o conjunto de ficheiros do projeto, tal como é mostrado na Figura 6, para que os ficheiros correspondentes a informação ou trabalhos realizados em cada e de outros quaisquer assuntos, fiquem agrupados convenientemente e possam ser facilmente encontrados.

¹ Uma função, em termos gerais:

- pode ser vista como um conjunto de comandos que realiza uma tarefa específica. Em outras palavras, pode-se dizer que é um pequeno "programa" utilizado por outros programas;
- é referenciada (chamada) pelo programa principal através de um nome atribuído a ela.
- visa subdividir um programa em partes (módulos) menores que realizam uma tarefa bem definida, algo de utilização muito comum em programação estruturada;
- Benefícios da utilização de funções:
 - permite o reaproveitamento de código já construído (por você ou por outros programadores);
 - evita que um mesmo trecho de código seja repetido várias vezes dentro de um mesmo programa e, com isso, qualquer alteração é feita apenas nesse trecho e de forma simples;
 - possibilita limitar o tamanho dos blocos do programa e, por consequência, mais fáceis de entender e manipular;
 - facilita a leitura do programa de maneira que os blocos de código possam ser logicamente compreendidos de forma isolada.

Adaptado de: <https://www.codingame.com/playgrounds/24988/programacao-c/funcoes-em-c>

² Obs.:

Possivelmente, ao terem lido a nota de rodapé anterior, houve muitos conceitos e termos que não compreenderam “bem”. Talvez isso só não terá acontecido aos estudantes que já tenham conhecimentos de uma qualquer linguagem de programação, pois, as funções estão disponibilizadas em qualquer linguagem de alto nível. Não sendo este o seu caso, e tendo, desde curiosidade sobre este tema, pode consultar o conteúdo do site de onde foi adaptado o texto aqui existente e, inclusivamente, procurar muitos outros, para alargar os s/ “horizonte”, desde já, neste domínio. Entretanto, como será dito um pouco mais abaixo, todos esses conceitos e nomes que motivaram a v/ pequena “frustração”, foram 1) talvez benéficos pois motivaram em alguns a curiosidade e a posterior pesquisa para “ordenar as ideias”; 2) serão devidamente apresentados daqui a algum tempo na UC e, aí, passarão a fazer todo o sentido, encaixando-se as diversas “peças” do puzzle.

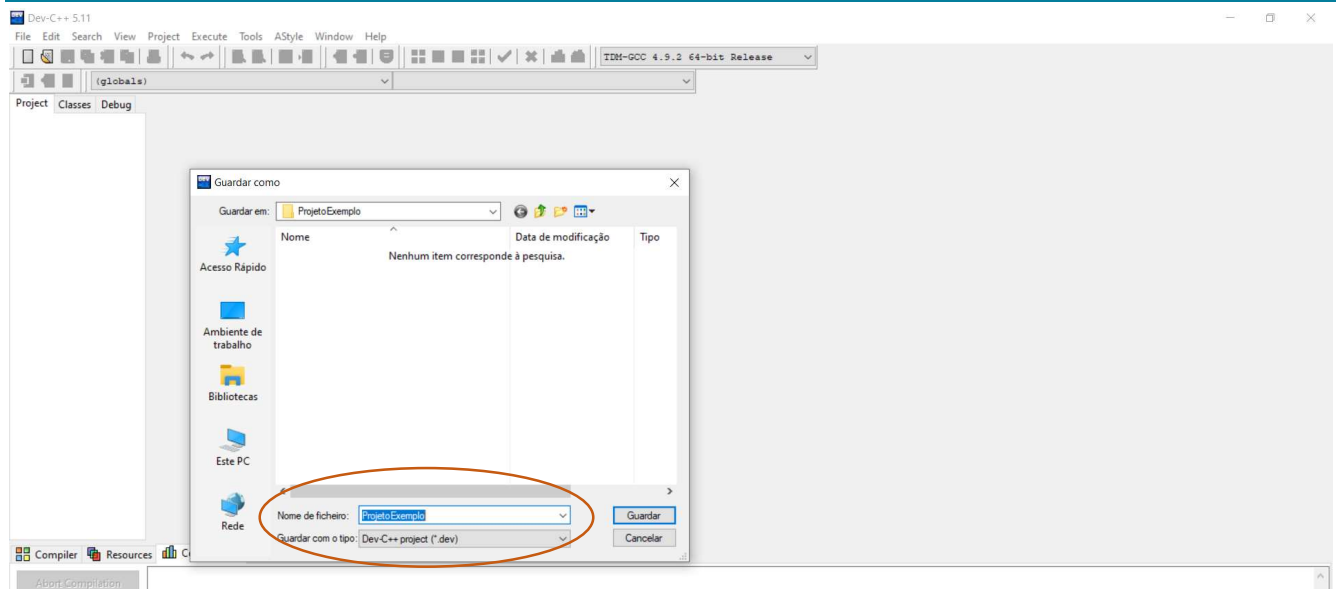


Figura 6 - Guardar novo projeto numa pasta criada especificamente para o efeito

Podem ir-se acrescentando novos ficheiros ao projeto, sempre que necessário.

Um exemplo pode observar-se nas Figura 7 e Figura 8, onde é acrescentado um ficheiro de nome `FuncoesXPTO`, onde se poderão guardar as funções (cujo conceito e utilidade, como foi informado na nota de rodapé 2 da página anterior, daqui a poucas semanas já farão sentido) que serão depois invocadas no `main()`, que não é mais do que a função “principal”, como o próprio nome indica e que é o “programa” de arranque da aplicação em C, sendo, por isso, de existência obrigatória em qualquer projecto.

Com a criação de vários ficheiros, de forma planeada, o programa, mesmo que muito complexo (como virão a perceber mais tarde, pode incluir dezenas ou centenas de componentes) fica mais organizado, tornando-se mais “manejável” e “partilhável”.

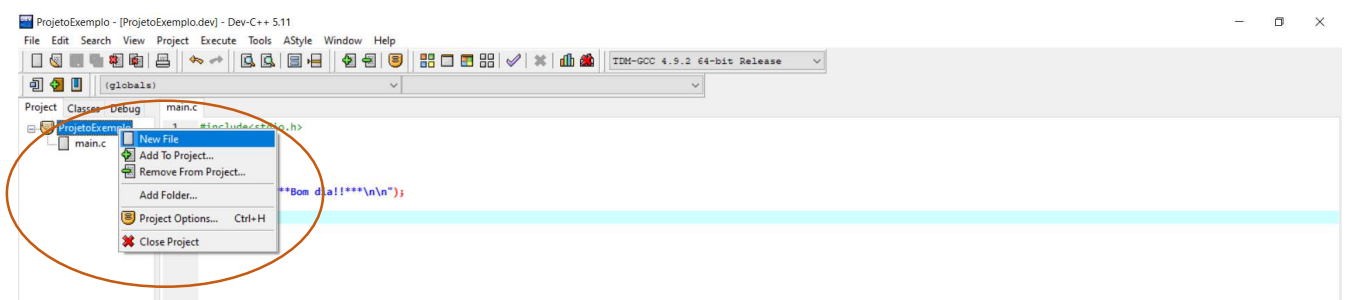


Figura 7 - Acrescentar ficheiro ao projeto.

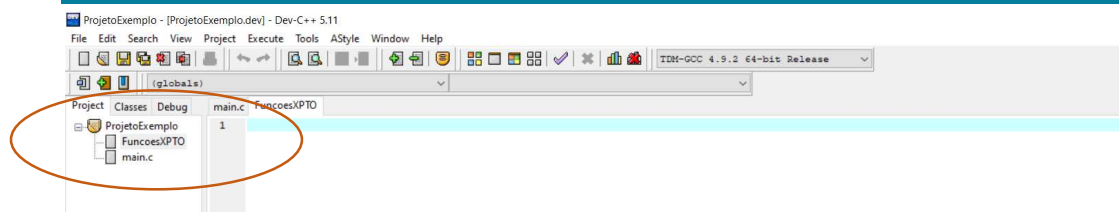


Figura 8 - Visualização dos ficheiros do projeto.