

# Introdução à Computação

Lógica e Ambiente de Programação

Prof.: Julliano Rosa Nascimento  
julliano.rn@gmail.com



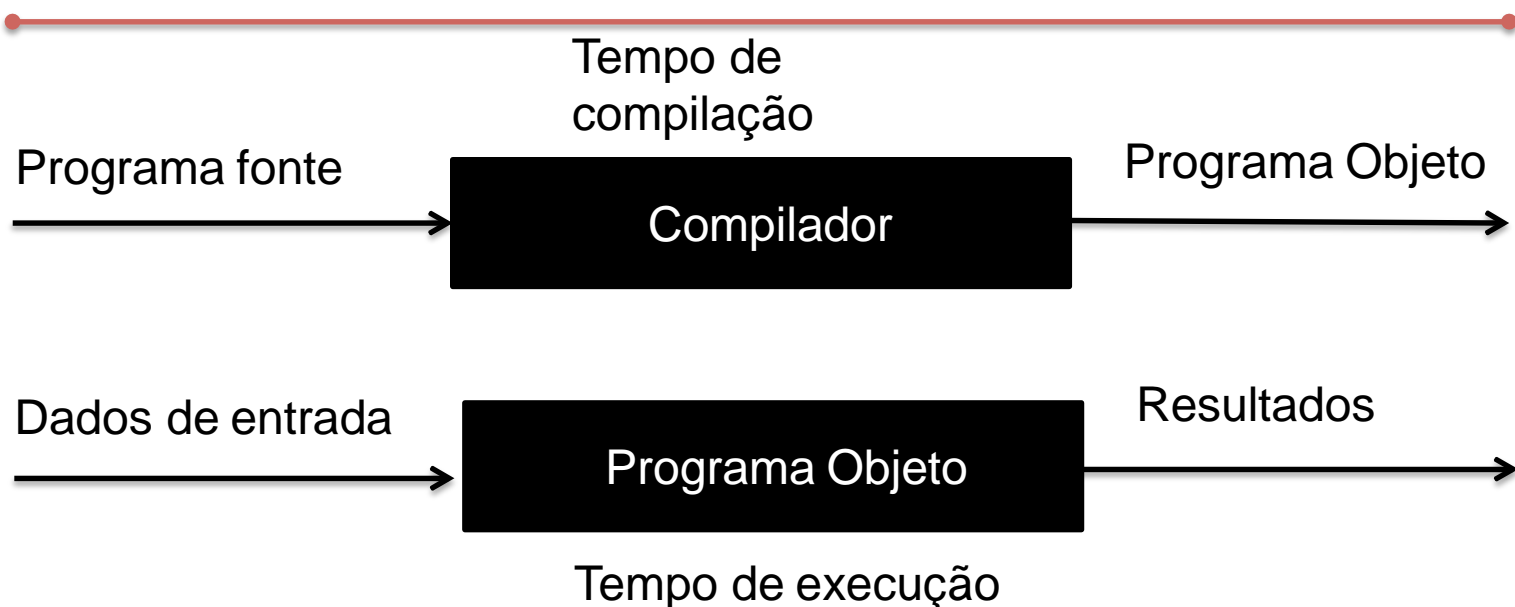
# Linguagem de Programação

- Conjunto de palavras, regras sintáticas e semânticas usadas para a definição de um programa de computador;
- Exemplos de linguagens de alto nível:
  - C
  - Java
  - PHP
  - Ruby



# Linguagem de Programação

- O computador não entende a linguagem de alto nível, então:



# Linguagem de Programação

- C, popular, programação estruturada
- Criada em 1972 – Dennis Ritchie
- Compilada
- De propósito geral
- C++, programação orientada a objetos



# Ambiente de Programação

- Code Bocks
  - Ambiente de desenvolvimento integrado (IDE, do inglês);

- Download Windows

<https://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01mingw-setup.exe/download>



## Code::Blocks



# Ambiente de Programação

- Code Bocks
  - Ambiente de desenvolvimento integrado (IDE, do inglês);

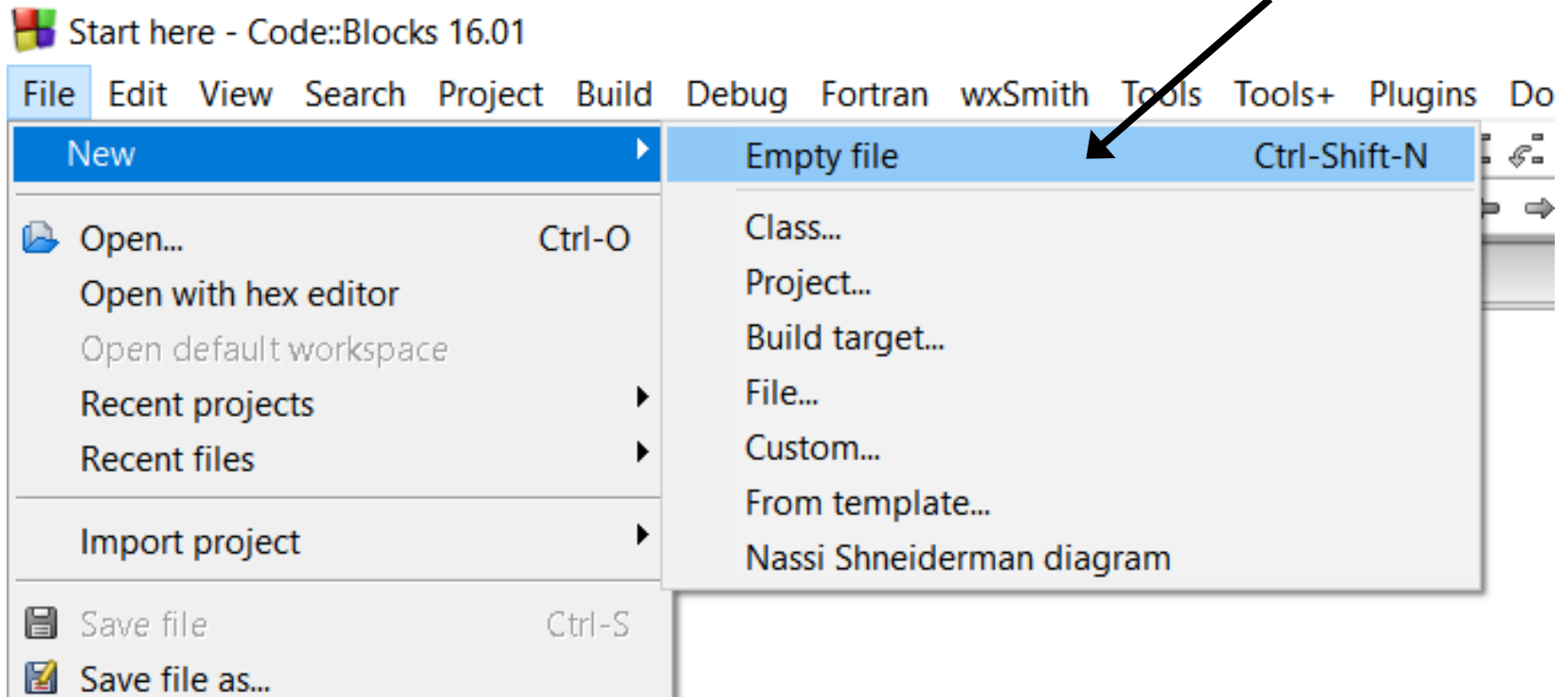
Para outros Sistemas Operacionais Visite  
<http://www.codeblocks.org/>



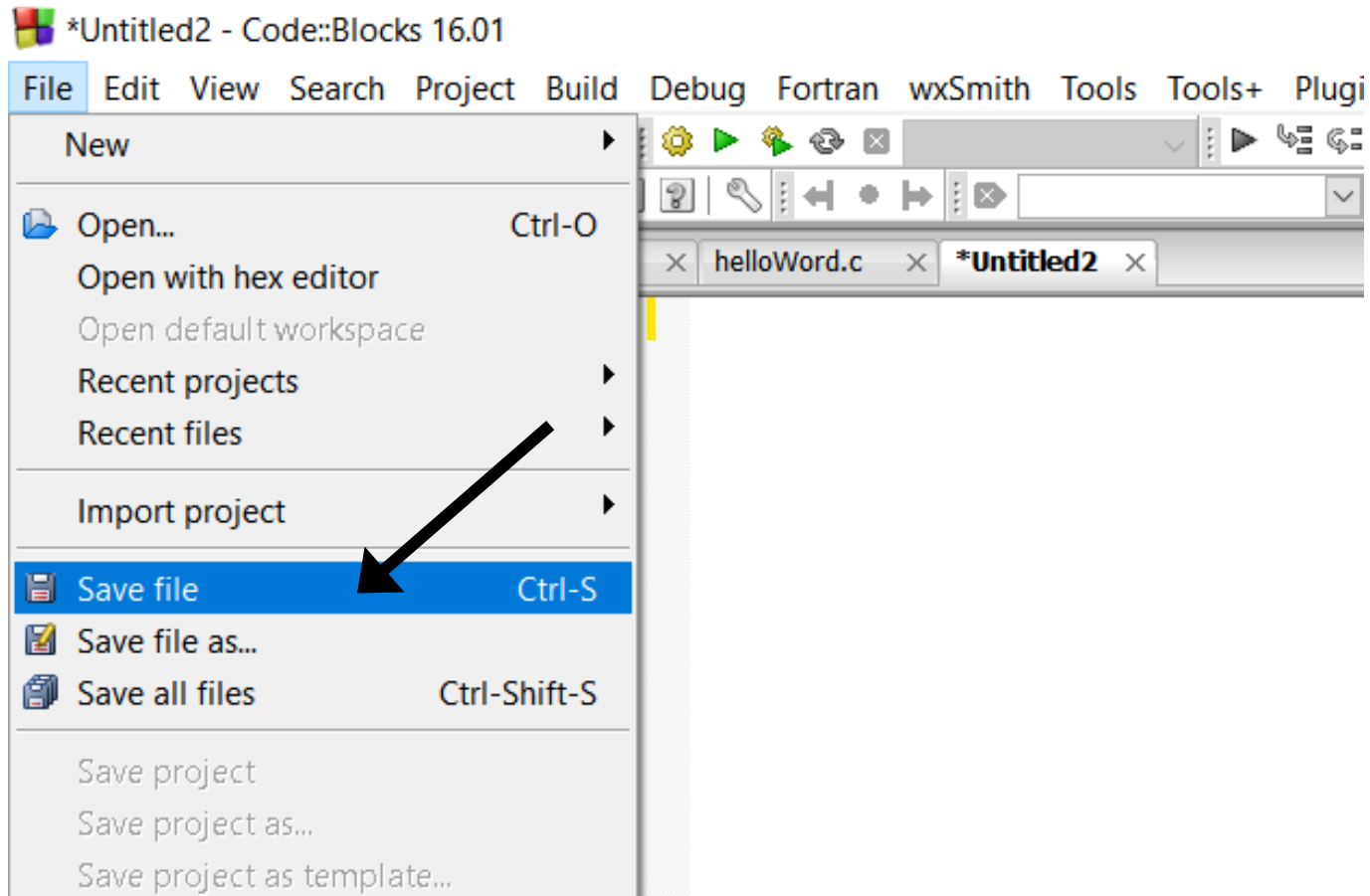
## Code::Blocks



# Ambiente de Programação

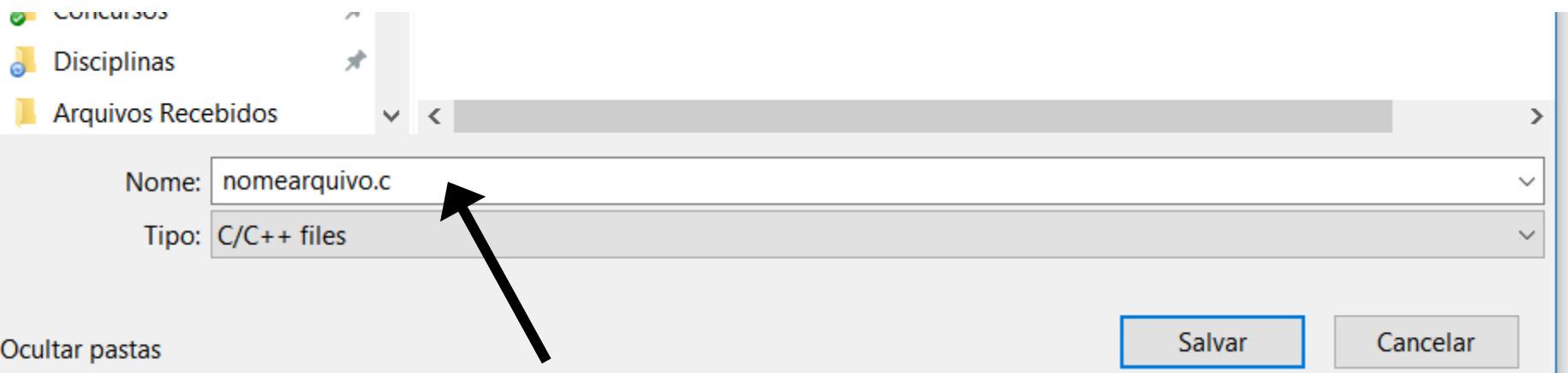


# Ambiente de Programação





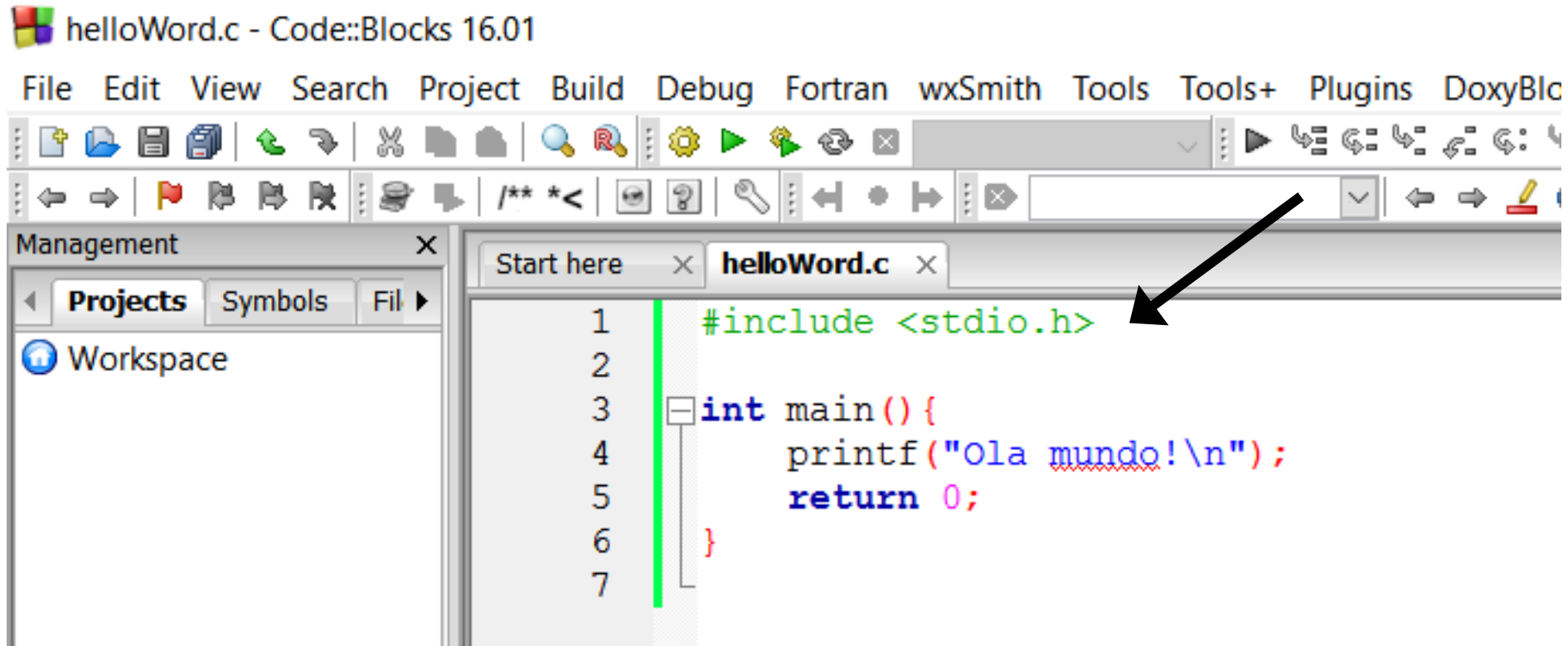
# Ambiente de Programação



Não esquecer de salvar **.c**



# Estrutura de um Programa



The screenshot shows the Code::Blocks IDE interface. The title bar reads "helloWord.c - Code::Blocks 16.01". The menu bar includes "File", "Edit", "View", "Search", "Project", "Build", "Debug", "Fortran", "wxSmith", "Tools", "Tools+", "Plugins", and "DoxyBlo". The toolbar contains various icons for file operations, editing, and execution. The "Management" panel on the left shows "Projects" and "Symbols" tabs, with "Workspace" selected. The main editor window, titled "Start here" and "helloWord.c", displays the following C code:

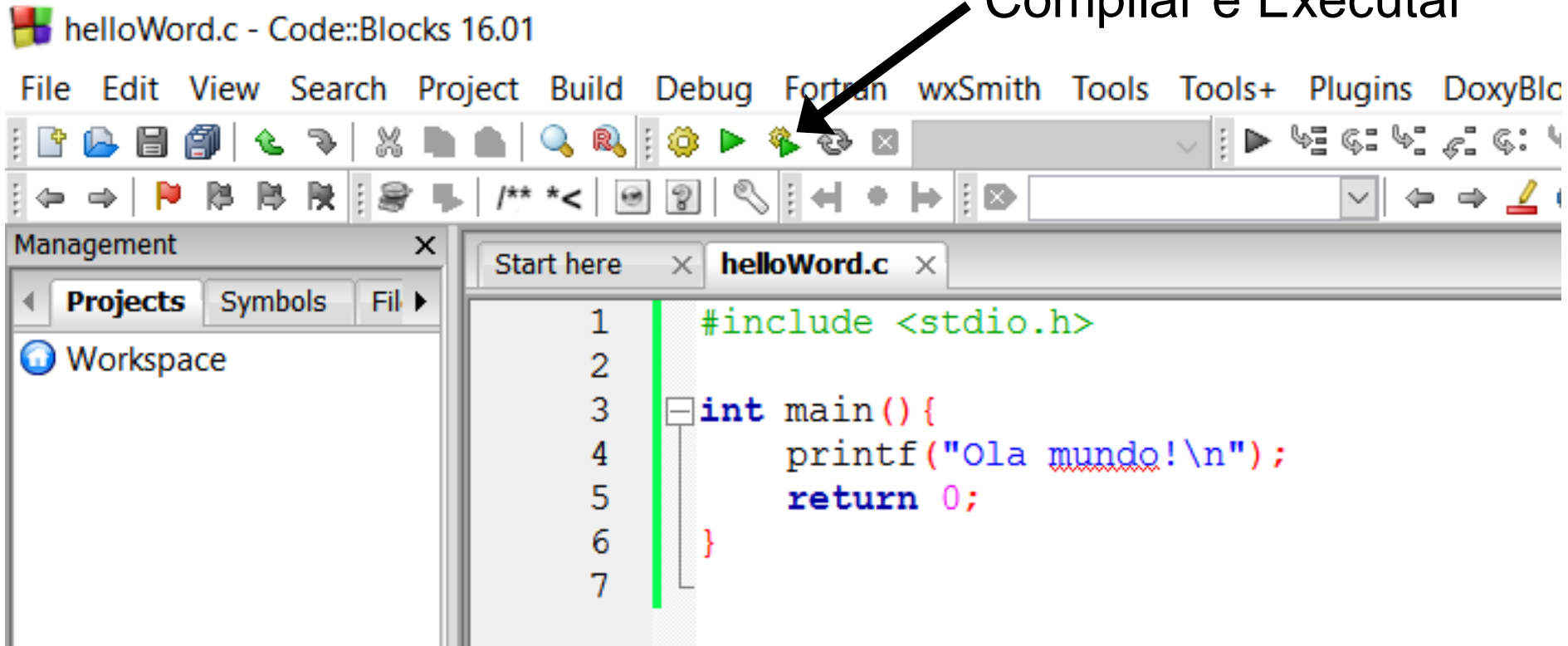
```
1  #include <stdio.h>
2
3  int main() {
4      printf("Ola mundo!\n");
5      return 0;
6  }
7
```

A black arrow points from the right side of the editor window to the first line of code, `#include <stdio.h>`.



# Estrutura de um Programa

Compilar e Executar

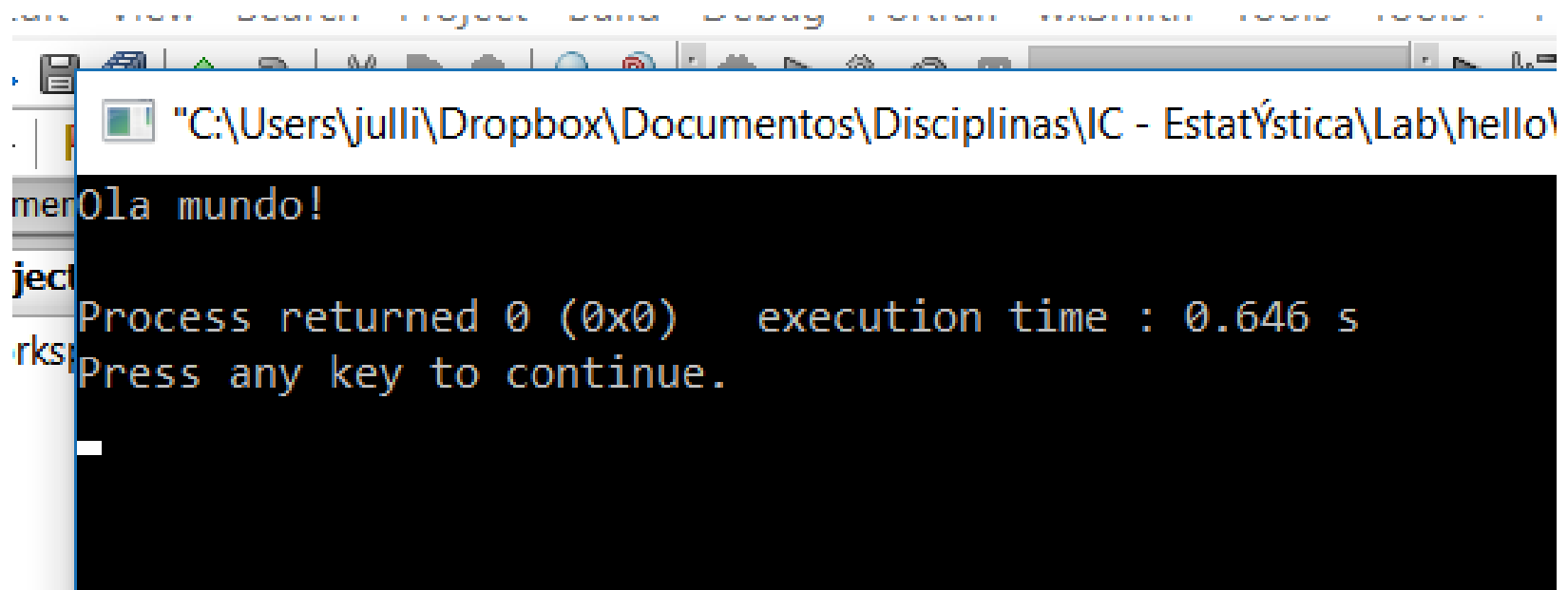


The screenshot shows the Code::Blocks IDE interface. The title bar reads "helloWord.c - Code::Blocks 16.01". The menu bar includes "File", "Edit", "View", "Search", "Project", "Build", "Debug", "Fortran", "wxSmith", "Tools", "Tools+", "Plugins", and "DoxyBlo". The toolbar contains various icons, including a green play button (run) and a green gear icon (compile). A black arrow points from the text "Compilar e Executar" to the green play button. The "Management" panel on the left shows "Projects" and "Symbols" tabs, with "Workspace" selected. The main editor window shows the code for "helloWord.c":

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Ola mundo!\n");
5      return 0;
6  }
7
```



# Estrutura de um Programa



```
"C:\Users\julli\Dropbox\Documentos\Disciplinas\IC - Estatística\Lab\hello"
Ola mundo!
Process returned 0 (0x0)   execution time : 0.646 s
Press any key to continue.
_
```



# Exercícios

1. Crie um programa para somar dois números inteiros quaisquer e exibir a soma para o usuário.



## ALGORITMO – EXEMPLO

Escreva um algoritmo para **somar dois números** quaisquer.

1. Leia o número X
2. Leia o número Y
3. Some X e Y
4. Mostre o resultado da soma

Entrada

Processamento

Saída

Número X  
Número Y

Somar X e Y

Resultado  
da soma



# Exercícios

2. Crie um programa para calcular e imprimir a média aritmética das notas de três provas de um aluno.



## ALGORITMO – Solução mais completa

### Entrada de Dados com Mensagem para o Usuário

**real** nota1, nota2, nota3, soma, media

1. **escreva**("Digite a primeira nota :")  
leia (nota1)

nota1

5.0

2. **escreva**("Digite a segunda nota :")  
leia (nota2)

nota2

6.0

3. **escreva**("Digite a terceira nota :")  
leia (nota3)

nota3

7.0

4. soma = nota1 + nota2 + nota3

5. media = soma / 3

6. **escreva** ("A média é = ", media)

soma

18.0

media

6.0





# Exercícios

3. Crie um programa para calcular a distância entre duas cidades numa mesma rodovia.



# Algoritmo

Calcular a distância entre duas cidades.

**real** cidade1, cidade2, distancia

1. **escreva**("Digite a localização (km) da cidade 1:")  
leia (cidade1)
2. **escreva**("Digite a localização (km) da cidade 2:")  
leia (cidade2)
3. distancia = cidade2 – cidade1
4. **escreva** ("A distancia entre as duas cidade é = ",  
distancia)
5. **escreva** ("ATENÇÃO: desconsidere sinal negativo")

