

Code::Blocks

Conhecendo a IDE



Programação de Computadores I

Professor: Edwar Saliba Júnior

Conhecendo a IDE Code::Blocks

Apresentação:

- 1) Abra a IDE (*Integrated Development Environment*), ou seja, o Ambiente Integrado de Desenvolvimento conhecido como “Code::Blocks”. Você verá a IDE, como na imagem a seguir:

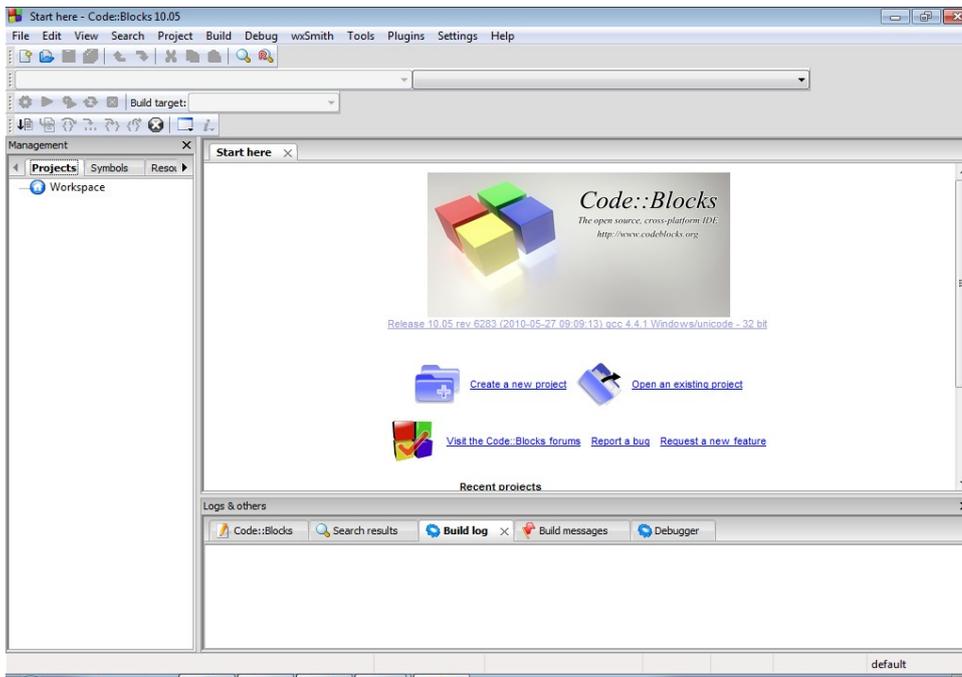


Figura 1: IDE Code::Blocks

Criando Um Novo Projeto de Software:

- 2) Vá em “File | New | Project...” a seguinte tela aparecerá para você:

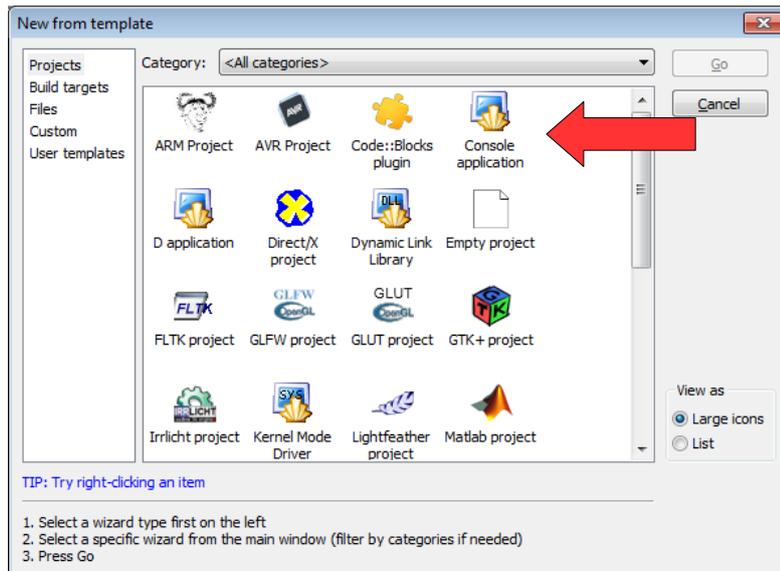


Figura 2: New from template

- 3) Escolha a opção "Console Application" (seta vermelha) (Figura 2), e em seguida aperte o botão GO. A seguinte tela aparecerá para você:

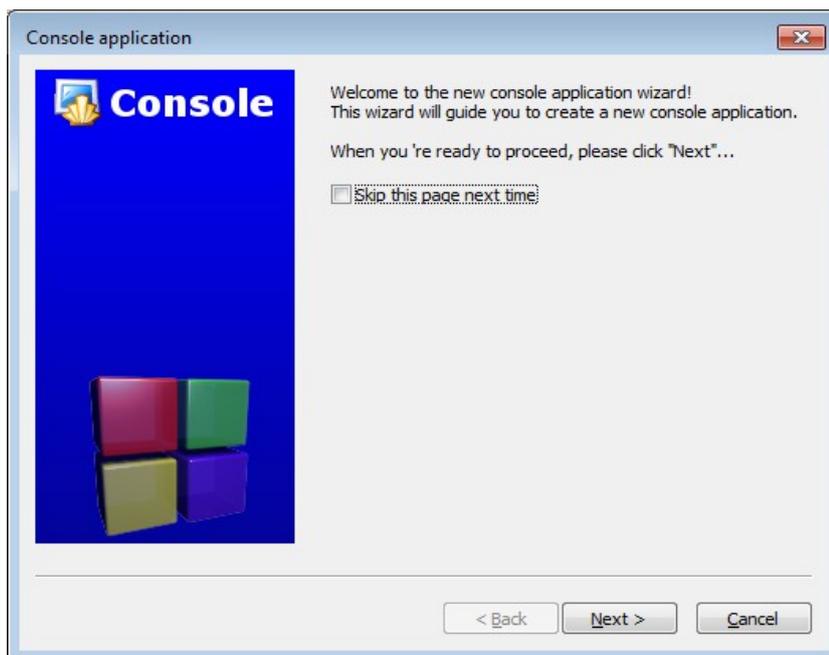


Figura 3: Console Application

- 4) Aperte o botão "Next" (Figura 3).

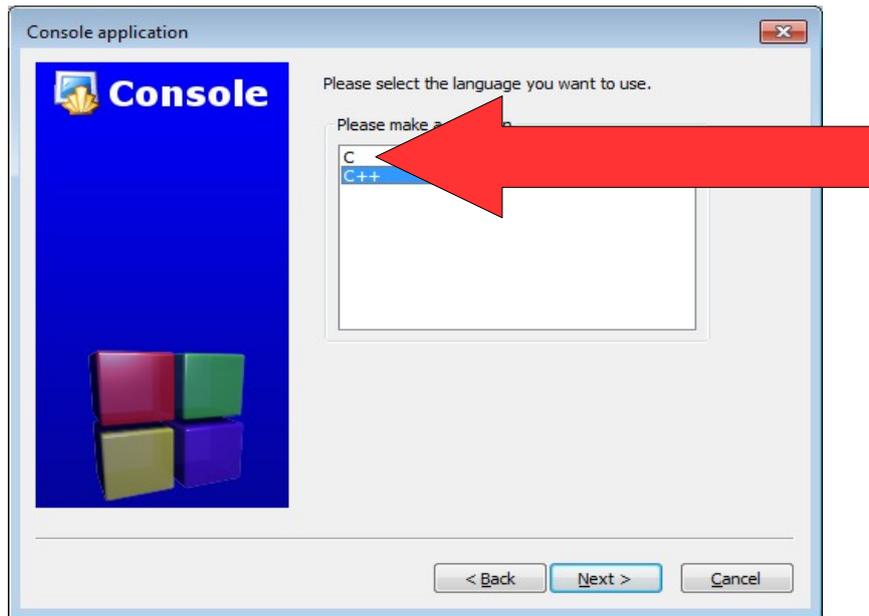


Figura 4: Console Application - Escolha da Linguagem

- 5) Na tela que surgir (Figura 4) você deverá escolher a opção “C”. E em seguida clicar no botão “Next”. Então a seguinte tela aparecerá:

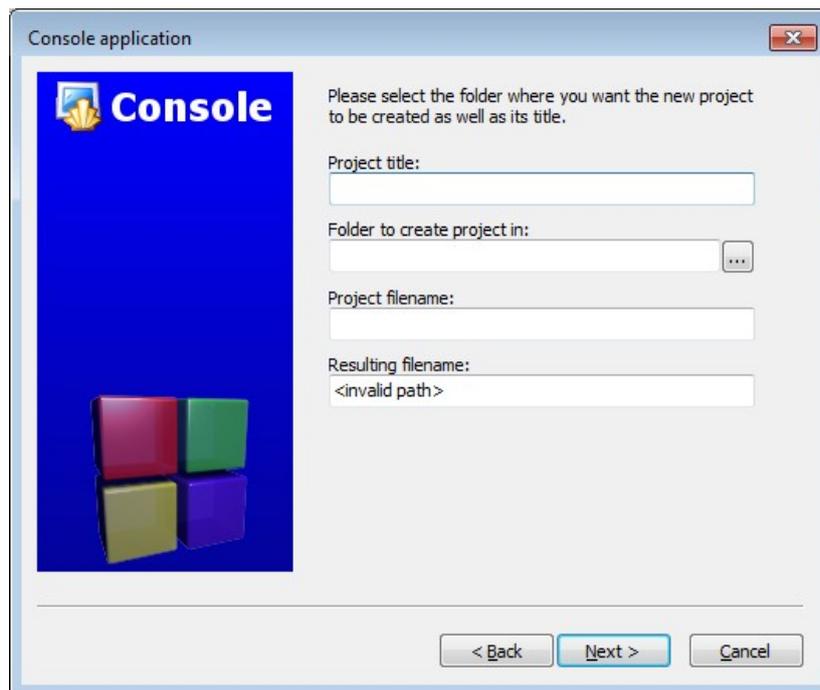


Figura 5: Console Application - Nome e Local do Projeto

- 6) Na tela que aparecer (Figura 5), no campo “*Project title*”, você deverá dar um nome ao projeto. Lembre-se que o nome do projeto deve seguir algumas regras vistas em sala de aula, ou seja, não deve conter espaço em branco, caracteres especiais e/ou acentuados, também não deve começar com número. E lembre-se também que a Linguagem C é *case-sensitive*. Assim sendo, vamos dar o nome de “apresentacao” ao nosso primeiro projeto. Será criada uma pasta com o nome que você deu ao projeto.

- 7) No campo logo abaixo, “Folder to create project in”, aperte o botão “...” e altere a pasta onde seu projeto deverá ser gravado. O ideal é que você grave todo o conteúdo trabalhado, no seu *pen-drive*, para que você possa levá-lo para casa. Caso contrário, salve numa pasta para usuários e depois envie-o para seu *e-mail* particular. Feito isto, aperte o botão “Next” (Figura 5).
- 8) Ao clicar no botão “Next” da tela apresentada na Figura 5, você verá a seguinte tela (Figura 6). Nesta tela você tem a opção de escolher o compilador que será utilizado (caso exista mais de um instalado) e também da a opção de você redirecionar alguns arquivos de saída, neste caso, sugere-se que você deixe como está, apenas aperte o botão “Finish”.

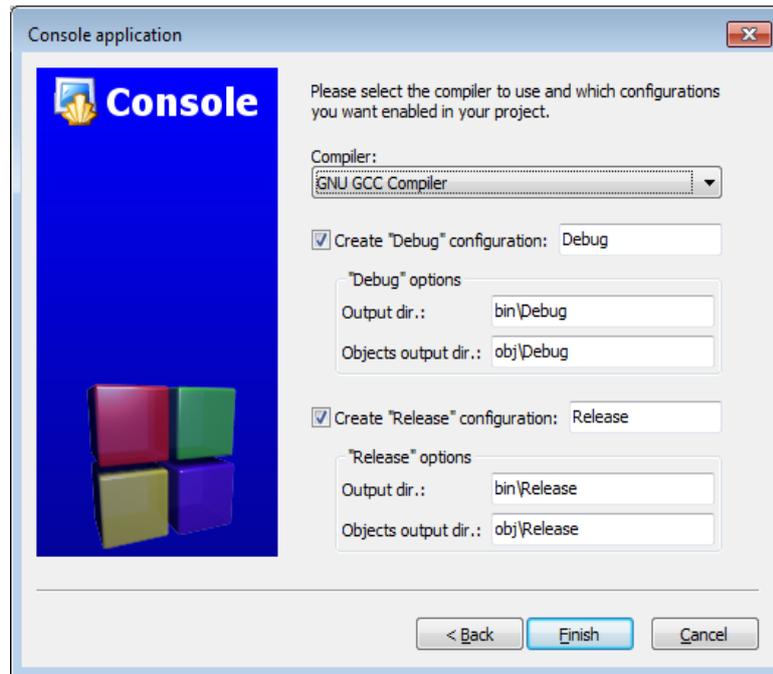


Figura 6: Console Application - Compilador

- 9) Ao apertar o botão “Finish”, você terá a visão da IDE completa. Então, clique no “+” da pasta “source” (seta vermelha) (Figura 7), e um duplo clique no arquivo “main.c” (seta azul). Então você terá a visão da tela abaixo (Figura 7):

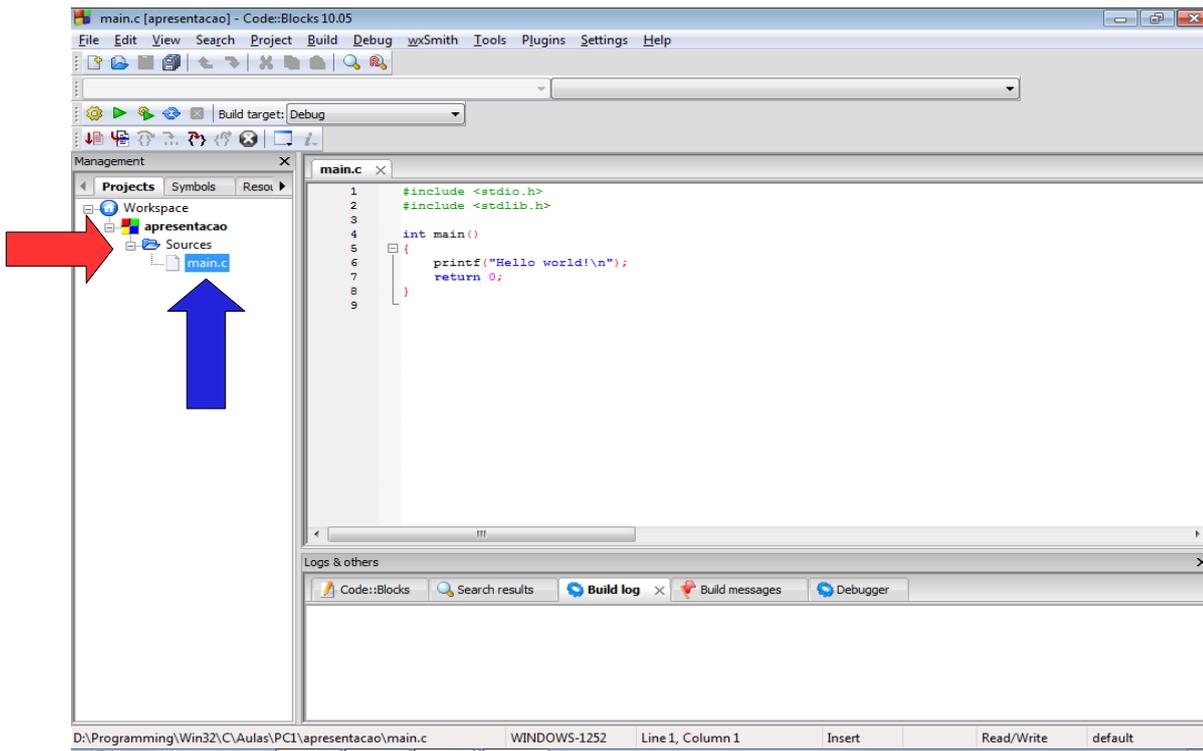


Figura 7: IDE Completa - Program main.c

- 10) Você notará que a IDE, cria um projeto com o nome que você escolheu. E também cria dentro deste projeto, o seu primeiro arquivo de código-fonte, chamado de "main.c" (Figura 7).
- 11) Agora, para continuarmos nossa aula, substitua o código-fonte "Hello World!" que a IDE deixou de exemplo para você, pelo seguinte código-fonte (Figura 8):

```

1  #include <stdio.h>
2  #define P 16
3
4  int main(void) {
5      int n; /* o expoente */
6      int val = 1;
7
8      printf("\npotências de 2^0 até 2^%d\n", P);
9      printf("=====\n");
10
11     for (n=0; n<=P; n++) {
12         printf("\t%d \t %d\n", n, val);
13         val = 2*val;
14     }
15
16     return 0;
17 }

```

Figura 8: Programa Exemplo 01

- 12) Após ter escrito o código-fonte da Figura 8, então compile-o e execute-o. Para compilação e execução em sequencia, basta pressionar a tecla F9.

Depurando Um Código-fonte:

- 13) Após executado o programa, vamos agora aprender a executá-lo passo a passo. Vamos colocar um *break point* na linha 11, ou seja, na linha do comando "for". Para tanto, dê um clique com o ponteiro o *mouse* no lado direito do número da linha (caso este esteja visível), caso não esteja visível, basta clicar próximo à borda da parte cinza onde são apresentados os números de linhas. Você verá uma bola vermelha, algo como mostrado na Figura 9.

```
1  #include <stdio.h>
2  #define P 16
3
4  int main(void) {
5      int n; /* o expoente */
6      int val = 1;
7
8      printf("\npotências de 2^0 até 2^%d\n", P);
9      printf("=====\n");
10
11  for (n=0; n<=P; n++) {
12      printf("\t%d \t %d\n", n, val);
13      val = 2*val;
14  }
15
16  return 0;
17  }
18
```

Figura 9: Break point

- 14) Agora aperte a tecla F8, o compilador deverá parar exatamente na bola vermelha, criando uma seta amarela dentro desta (Figura 10). E apertando a tecla F7, você poderá acompanhar a execução linha a linha do código-fonte.

```
1 #include <stdio.h>
2 #define P 8
3
4 int main(void) {
5     int n; /* o expoente */
6     int val = 1;
7
8     printf("\npotências de 2^0 até 2^%d\n", P);
9     printf("=====\n");
10
11     for (n=0; n<=P; n++) {
12         printf("\t%d \t %d\n", n, val);
13         val = 2*val;
14     }
15
16     return 0;
17 }
18
```

Figura 10: Depuração

- 15) Ao pressionar F7, a setinha amarela vai “andando” linha a linha e você poderá, através da tela de “watches” acompanhar os valores que as variáveis do programa assumirão durante a execução do programa.
- 16) Para visualizar a tela “watches”, acesse o *menu* "Debug | Debugging windows | Watches". Ao marcar a tela “watches”, ela aparecerá na IDE. Para adicionar variáveis na tela de “Watches”, basta clicar com o botão direito do *mouse* dentro da área da tela de *watches*, então aparecerá um *menu pop-up* onde você deverá selecionar a opção “Add watch”. Que abrirá a tela mostrada na Figura 11:

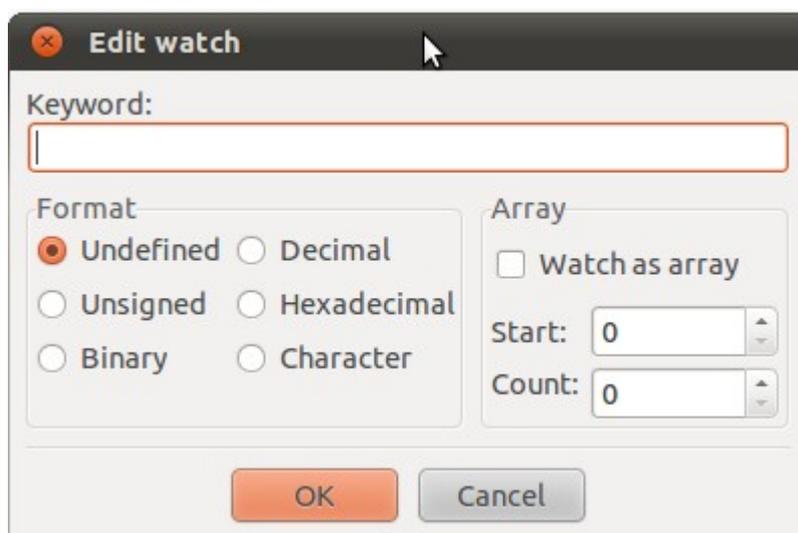
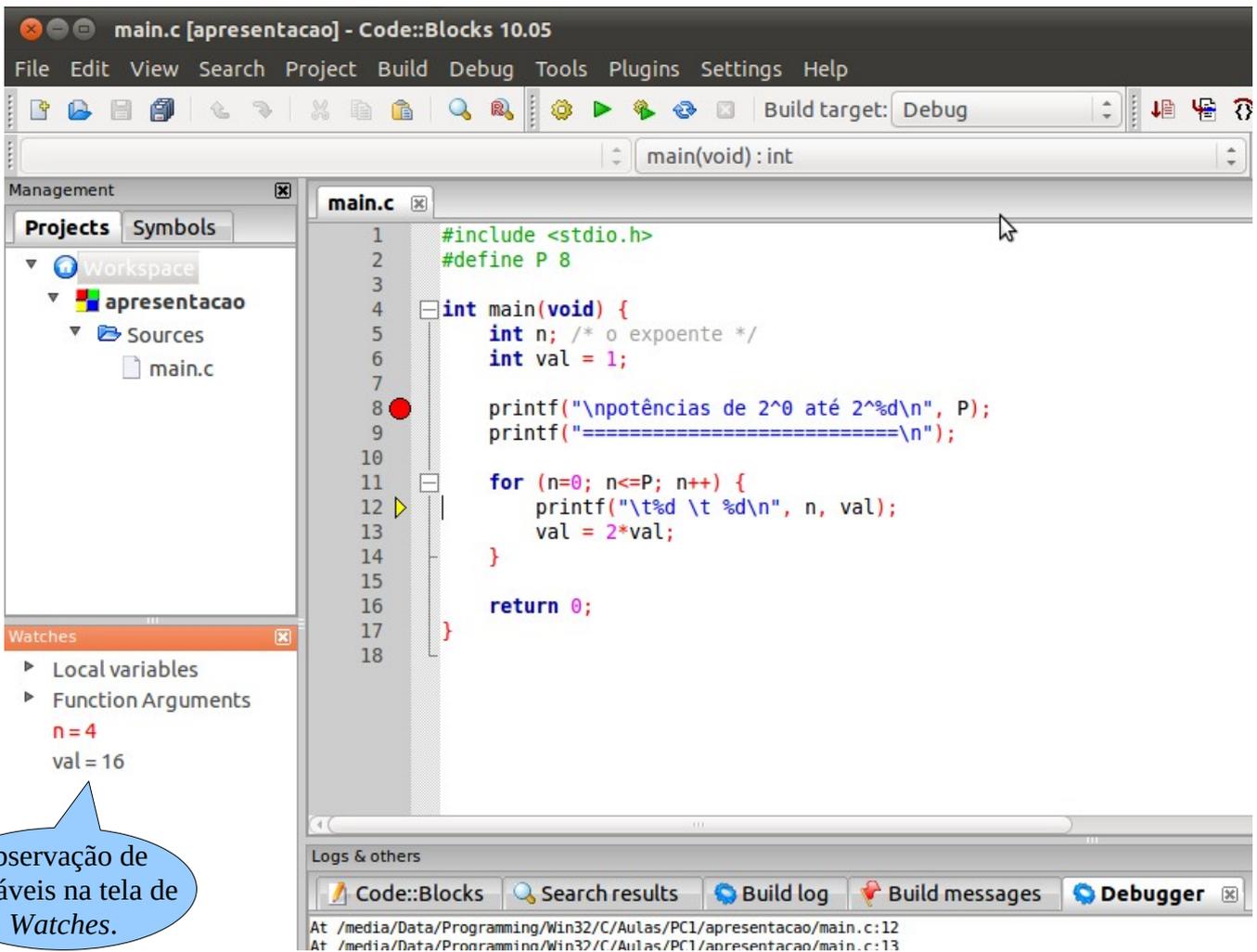


Figura 11: Add watch

- 17) Na tela “Edit watch” (Figura 11), você deverá colocar o nome da variável, ou das variáveis as quais você deseja observar o comportamento (uma por uma).

- 18) A medida que você vai pressionando a tecla F7 e navegando pelo código-fonte, você poderá observar o comportamento das variáveis que adicionou na tela de *watches*. Desta forma, você poderá depurar o seu *software* e eventualmente consertar todos os erros de lógica que encontrar.



Observação de variáveis na tela de *Watches*.

Figura 12: Tela de *Watches* em Tempo de Execução

Entendendo o Código-fonte:

#include – Comando para adição de bibliotecas (de comandos ou funções) no código-fonte.

#define – Declaração de constante em linguagem C.

int – Declaração de tipo. Utilizado na assinatura da função como tipo de retorno e na declaração das variáveis “n” e “val” como tipo da variável.

int main() - Assinatura da função.

{ - Início do corpo da função.

} - Fim do corpo da função.

“**int n**” e “**int val**” - Criação de variáveis do tipo inteiro. Criação de espaços na memória do computador, onde pode-se guardar valores do tipo inteiro.

/* xyz */ - Comentário. O que estiver em comentário não é executado pelo compilador. Utilizado para documentar o *software*.

printf - Comando para impressão na tela do computador.

return – Comando utilizado para retorno de valor pela função.

for – Estrutura de repetição, será tratado em tópicos mais avançados do curso.