

# SSC 0301 – IC para Engenharia Ambiental

## *Tipos, variáveis e expressões*

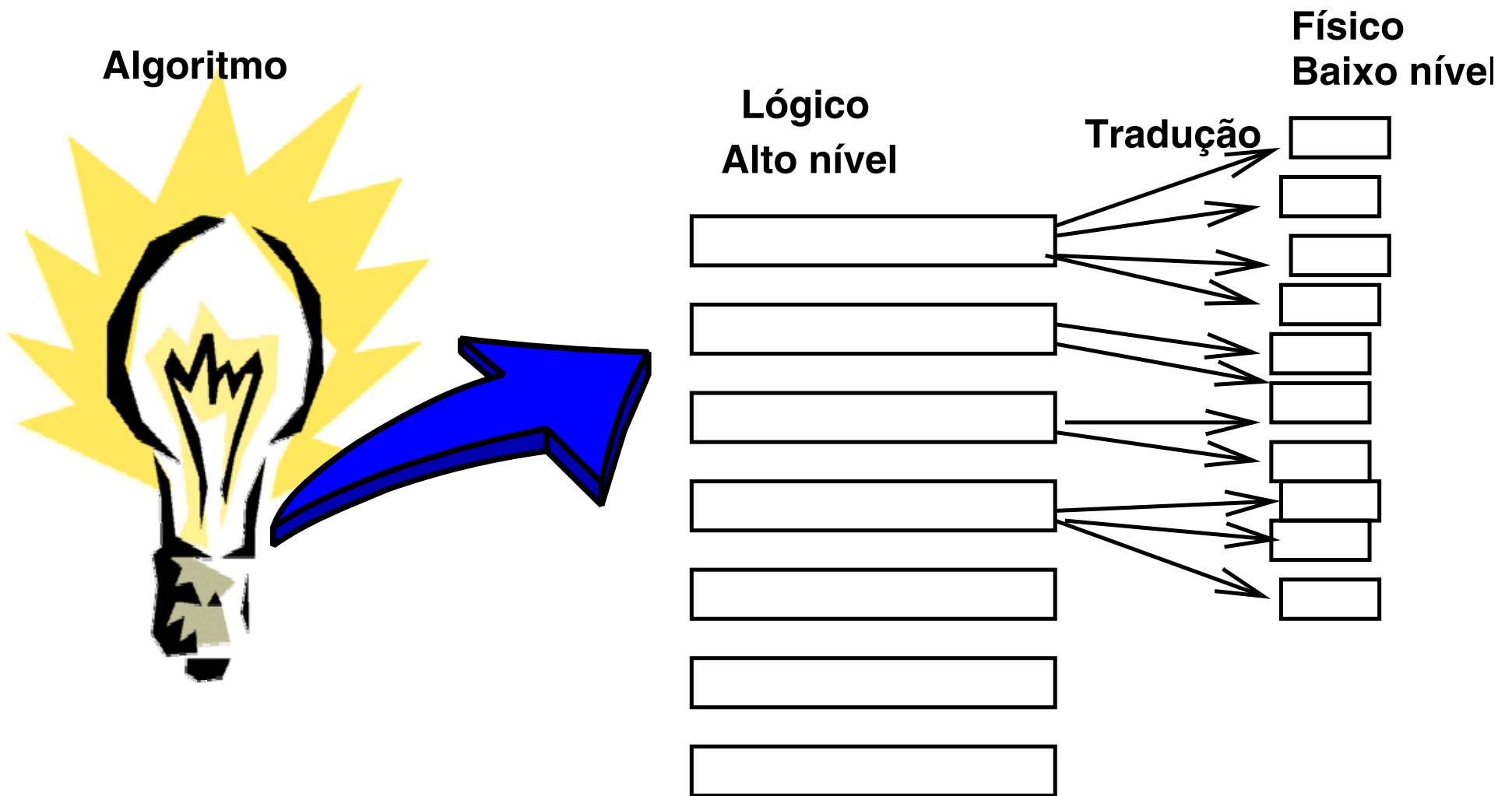
Prof. Márcio Delamaro e Prof. Jorge Luiz e Silva

`delamaro@icmc.usp.br; jsilva@icmc.usp.br`

# Algoritmo

- Um algoritmo é uma sequência de passos para realizar uma tarefa ou resolver um problema. Em nosso dia a dia utilizamos algoritmos para realizar nossas atividades, definindo a sequência de atividades que devemos fazer para atingir um objetivo.

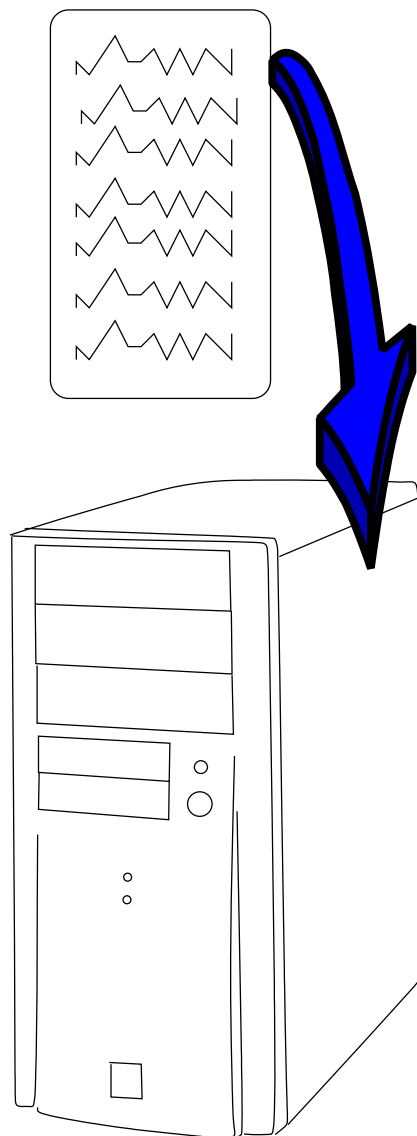
# Resumindo (de novo)



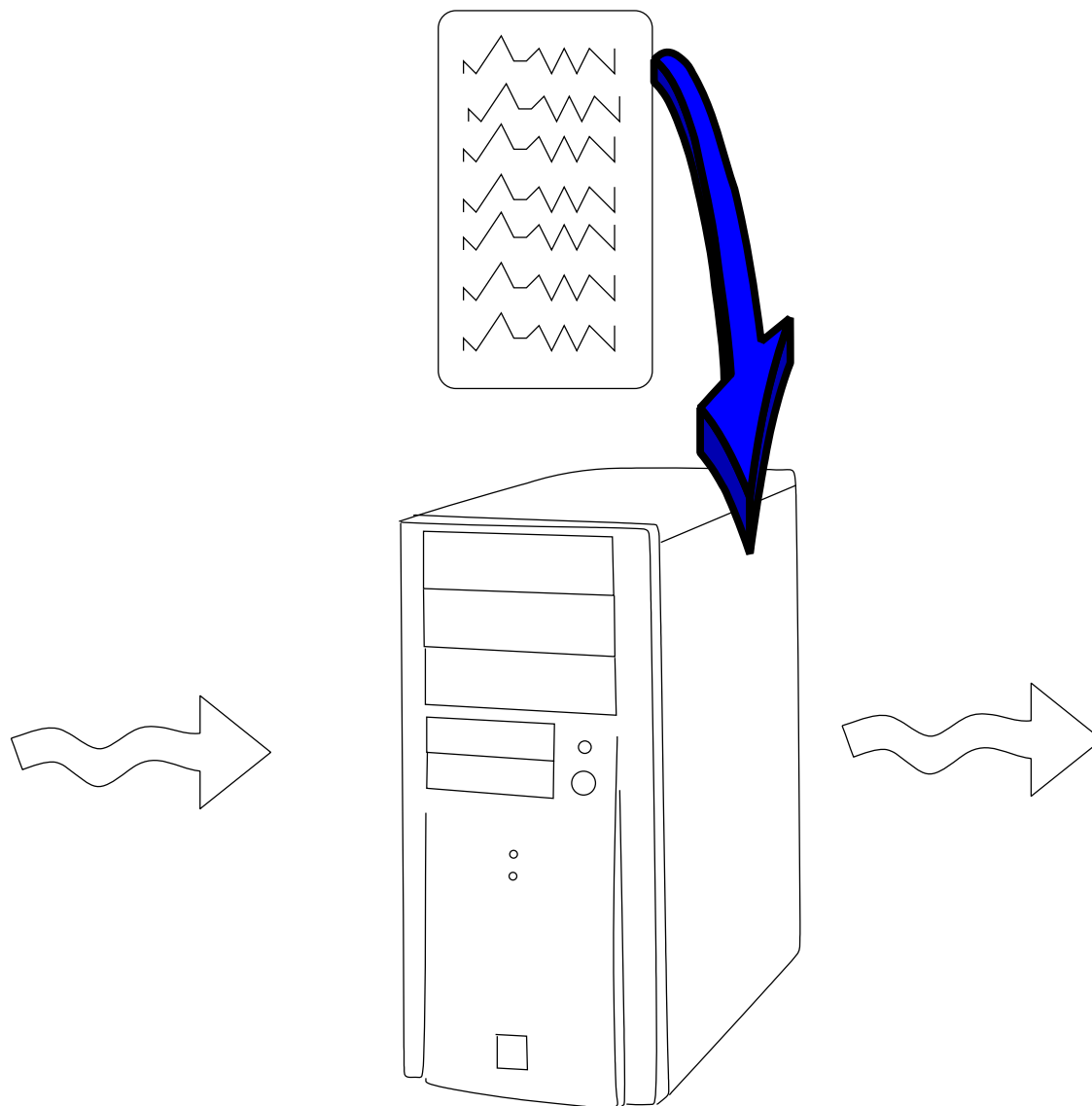
# A linguagem C

- Criada em 1972, por Dennis Ritchie, no AT&T Bell Labs
- Criada para desenvolver o sistema operacional Unix
- Uma das linguagens mais populares
- Linguagem poderosa e complexa
- Possui diversos dialetos

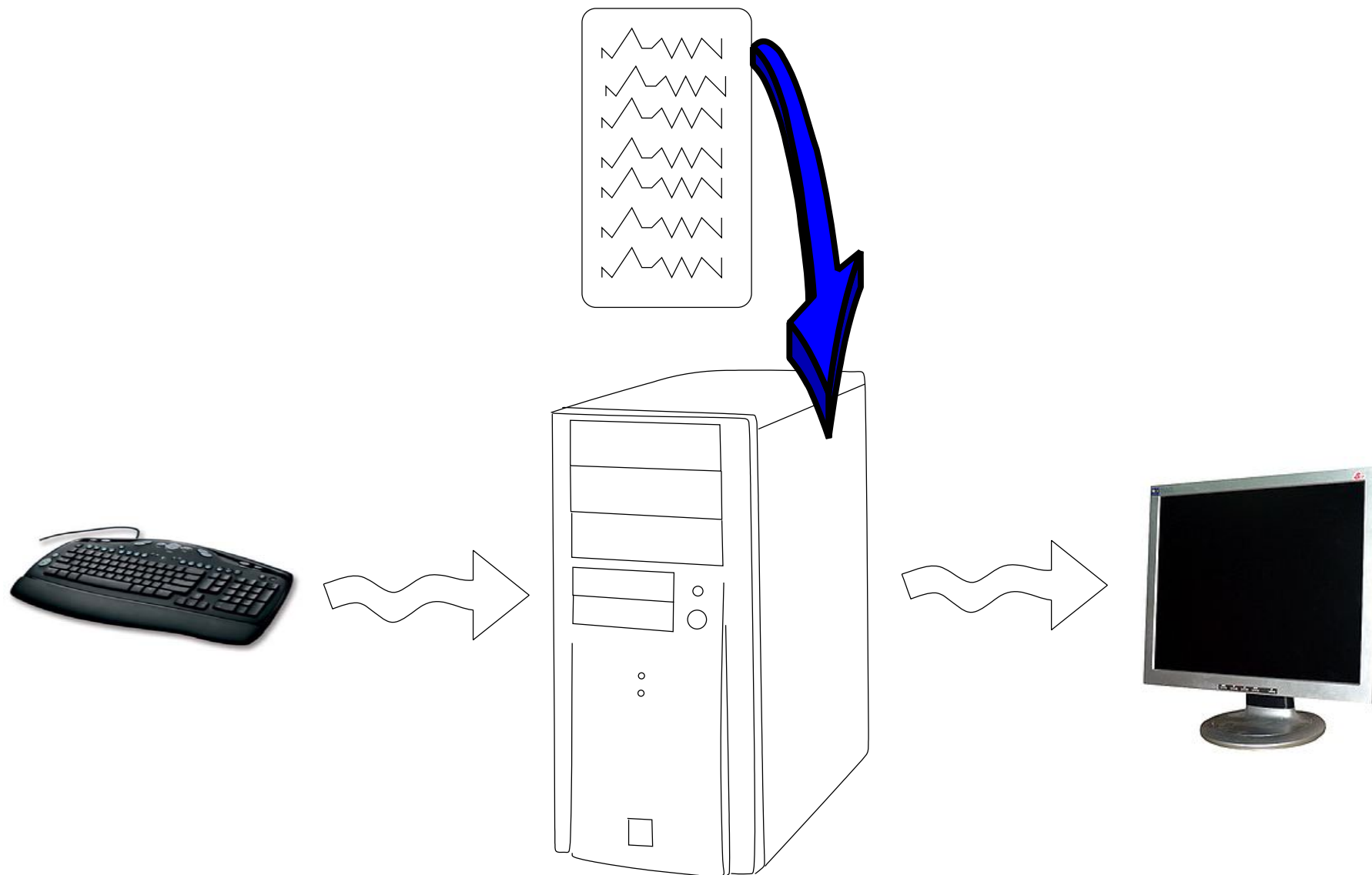
# O ambiente de execução



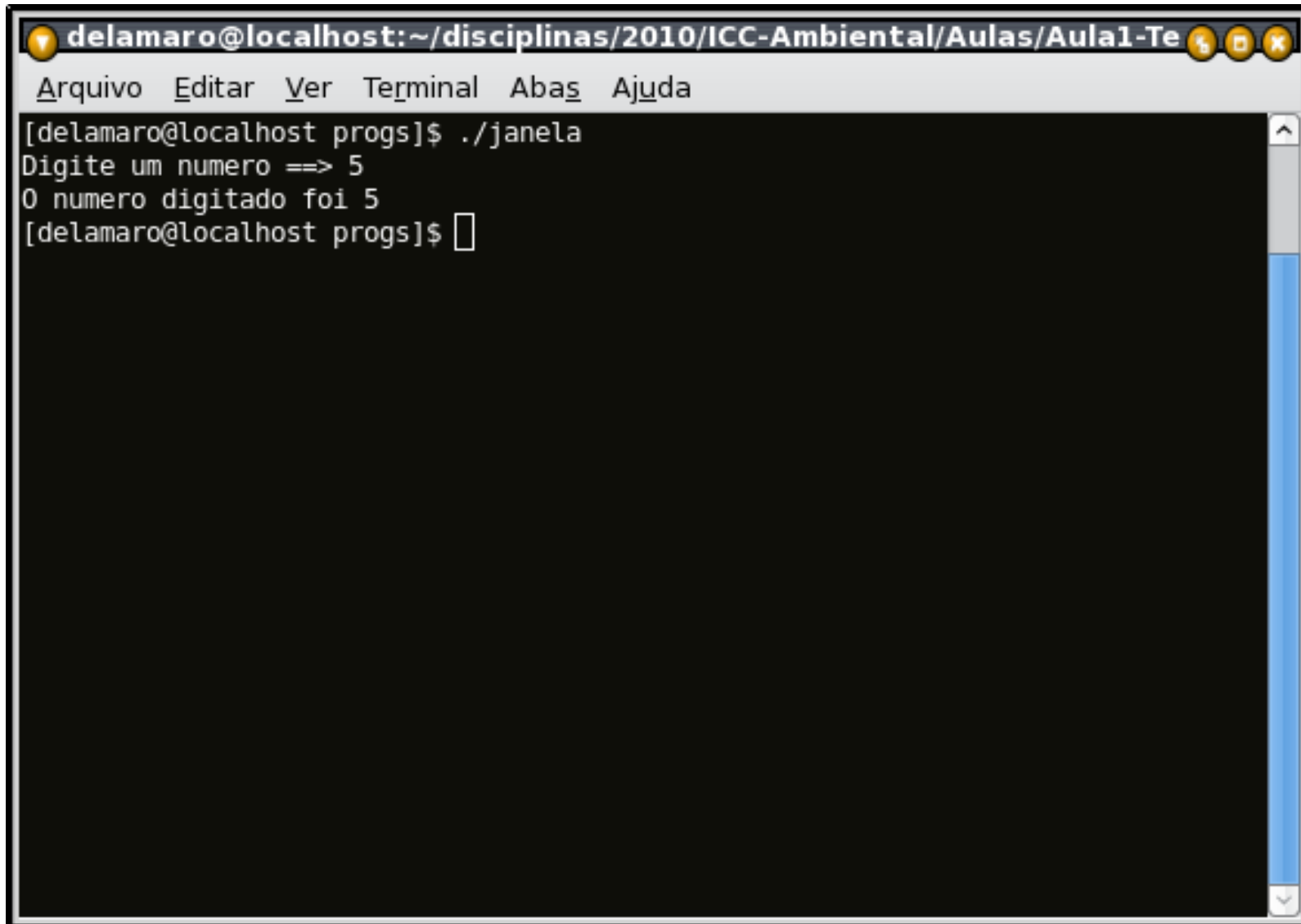
# O ambiente de execução



# O ambiente de execução



# Ambiente textual



The image shows a terminal window with a title bar that reads "delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te". The window has a menu bar with options: "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal content shows a shell prompt "[delamaro@localhost progs]\$ ./janela", followed by the program's output: "Digite um numero ==> 5" and "O numero digitado foi 5". The prompt returns to "[delamaro@localhost progs]\$".

```
delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[delamaro@localhost progs]$ ./janela
Digite um numero ==> 5
O numero digitado foi 5
[delamaro@localhost progs]$
```

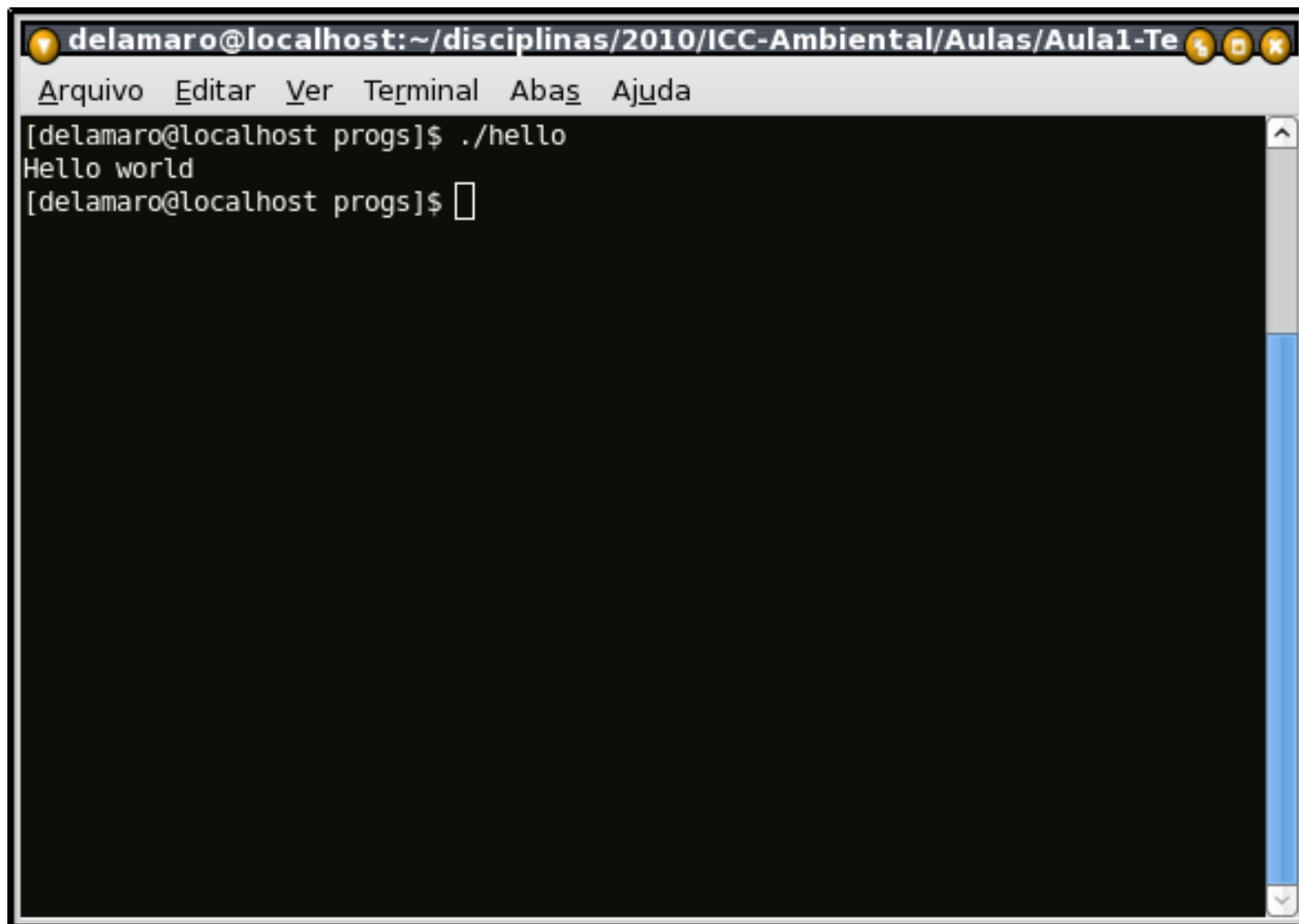


# Hello world

```
#include <stdio.h>

int main()
{
    printf("Hello world\n");
}
```

# O ambiente de execução



A screenshot of a terminal window. The title bar shows the user 'delamaro' at 'localhost' in the directory '~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Terminal', 'Abas', and 'Ajuda'. The terminal content shows the command './hello' being executed, resulting in the output 'Hello world'. The prompt '[delamaro@localhost progs]\$' is visible before and after the command.

```
delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te
Arquivo  Editar  Ver    Terminal  Abas  Ajuda
[delamaro@localhost progs]$ ./hello
Hello world
[delamaro@localhost progs]$
```

# Comando de saída/escrita

- `printf` é uma **função** padrão do C
- A execução desse comando é uma **chamada** de função
- O que vai entre parênteses são **parâmetros**
- O “\n” no final serve para pular uma linha

# Variáveis

- Variáveis são espaços do programa aonde podemos armazenar valores
- Em C podemos ter quantas variáveis quisermos
- Podemos dar qualquer nome às variáveis (resguardadas algumas restrições)
- toda variável precisa ser **declarada** antes de ser usada

# Nomes das variáveis

- Nomes podem ter Letras, Numeros e \_
- Nomes começam por Letra ou \_
- Não deve conter caracteres estranhos
- Não pode usar palavras reservadas
- Minúsculas e Maiúsculas podem ser usadas
- Minúsculas e Maiúsculas são diferentes
  - Podemos ter uma variável `x` e uma `X`
  - Ou ainda `variavel_qquer` e `variaveL_qquer`

# Tipos de dados

- Tudo em C possui um **tipo**
- Uma variável possui um tipo
- O tipo da variável é definido na declaração da variável
- Não é possível mudar o tipo da variável

# Tipos mais comuns

- char: um **byte** que armazena o código de um caractere
- int: um inteiro cujo tamanho depende do processador e do compilador usado, tipicamente 16 ou 32 bits (2 ou 4 bytes)
- float: um número real com precisão simples
- double: um número real com precisão dupla

# Declaração de variáveis

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int i;
```

```
int j, k;
```

```
double x, y, z;
```

```
    printf("Hello world\n");
```

```
}
```

- Devem aparecer antes dos comandos
- Podem ser várias declarações
- Cada declaração pode ter uma ou mais variáveis



# Declaração de variáveis

- É sempre uma boa ideia usar nomes que indiquem para que serve a variável

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int idade;
```

```
int qtidade_alunos;
```

```
double perimetro, velocidade_media;
```

```
}
```

# Comando de entrada/leitura

- A função `scanf` é usada para ler do teclado o valor de uma variável
- A chamada do `scanf` precisa de dois parâmetros
  - o tipo do valor que deve ser lido
  - o nome da variável onde o valor vai ser armazenado

# scanf – exemplo

```
#include <stdio.h>

int main()
{
int idade;
float perimetro;
double velocidade_media;

scanf ("%d", &idade);
scanf ("%f", &perimetro);
scanf ("%lf", &velocidade_media);
scanf ("%d %f %lf", &idade, &perimetro,
        &velocidade_media);
}
```

# scanf – exemplo (2)

```
#include <stdio.h>

int main()
{
int idade;

printf("Forneça a sua idade ==>");
scanf("%d", &idade);

}
```

# Escrever uma variável

- O comando `printf` também pode receber valores como parâmetros
- Com isso é possível escrever na saída o valor armazenado nas variáveis
- Da mesma forma, é preciso indicar qual é o tipo e qual é o valor a ser mostrado
- `printf("%d", idade);`
- `printf("%lf", velocidade_media);`
- O primeiro parâmetro pode ser uma mensagem completa

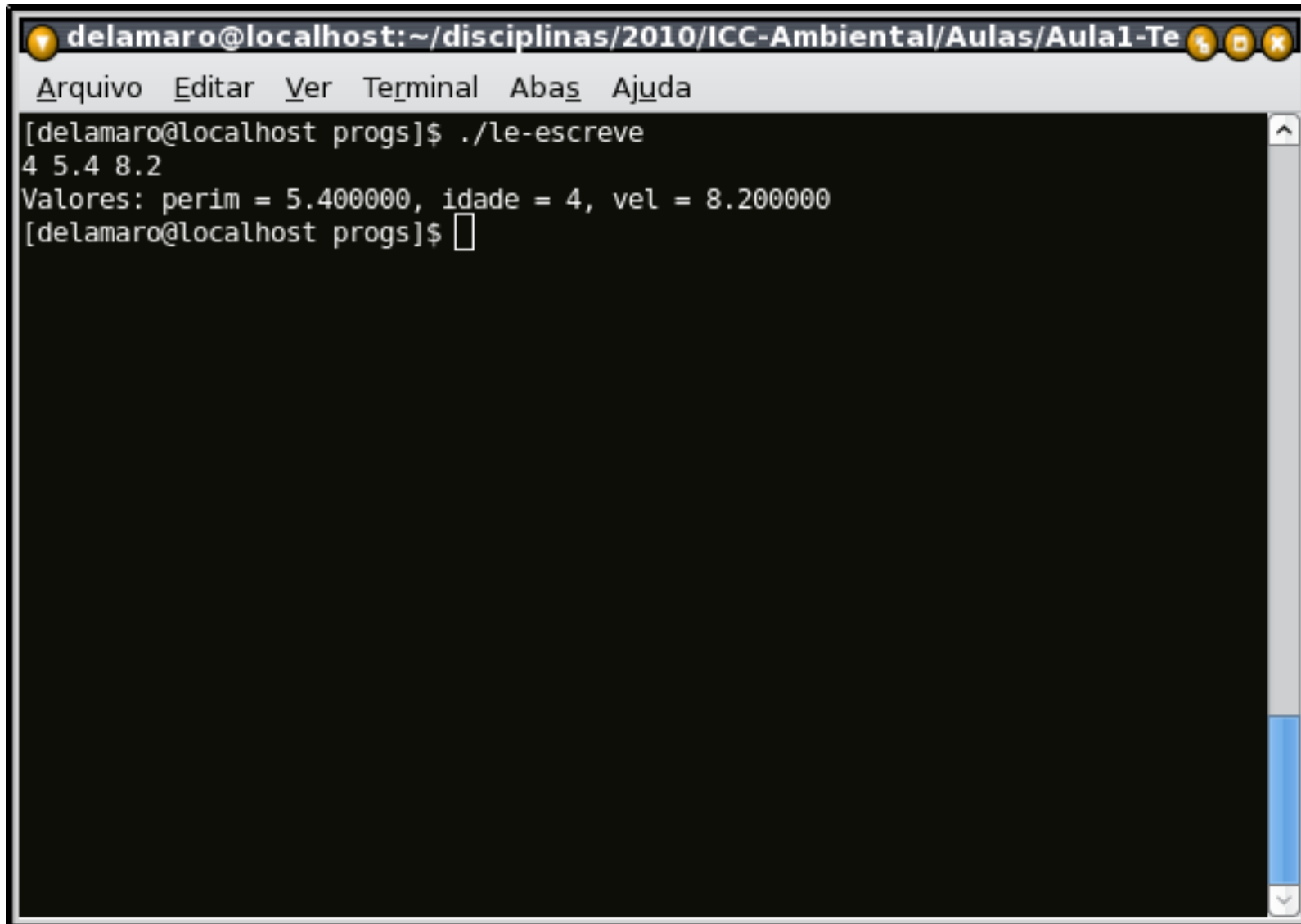
# printf – exemplo

```
#include <stdio.h>

int main()
{
int idade;
float perimetro;
double velocidade_media;

scanf("%d %f %lf", &idade, &perimetro,
      &velocidade_media);
printf("Valores: perim = %f, idade = %d, vel = %lf \n",
      perimetro, idade, velocidade_media);
}
```

# printf – exemplo



A terminal window titled "delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te" with standard window controls. The menu bar includes "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal content shows a shell prompt "[delamaro@localhost progs]\$ ./le-escreve" followed by the input "4 5.4 8.2" and the output "Valores: perim = 5.400000, idade = 4, vel = 8.200000". The prompt returns to "[delamaro@localhost progs]\$" with a cursor.

```
delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[delamaro@localhost progs]$ ./le-escreve
4 5.4 8.2
Valores: perim = 5.400000, idade = 4, vel = 8.200000
[delamaro@localhost progs]$
```

# Mais formatação

- `%5d` `%5f` `%5lf` – escreve o número com tamanho 5
- `%05d` `%05f` `%05lf` – escreve o número com tamanho 5 preenchendo com zeros à esquerda caso necessário
- `%5.2f` `%5.2lf` – tamanho total e numero de casas decimais



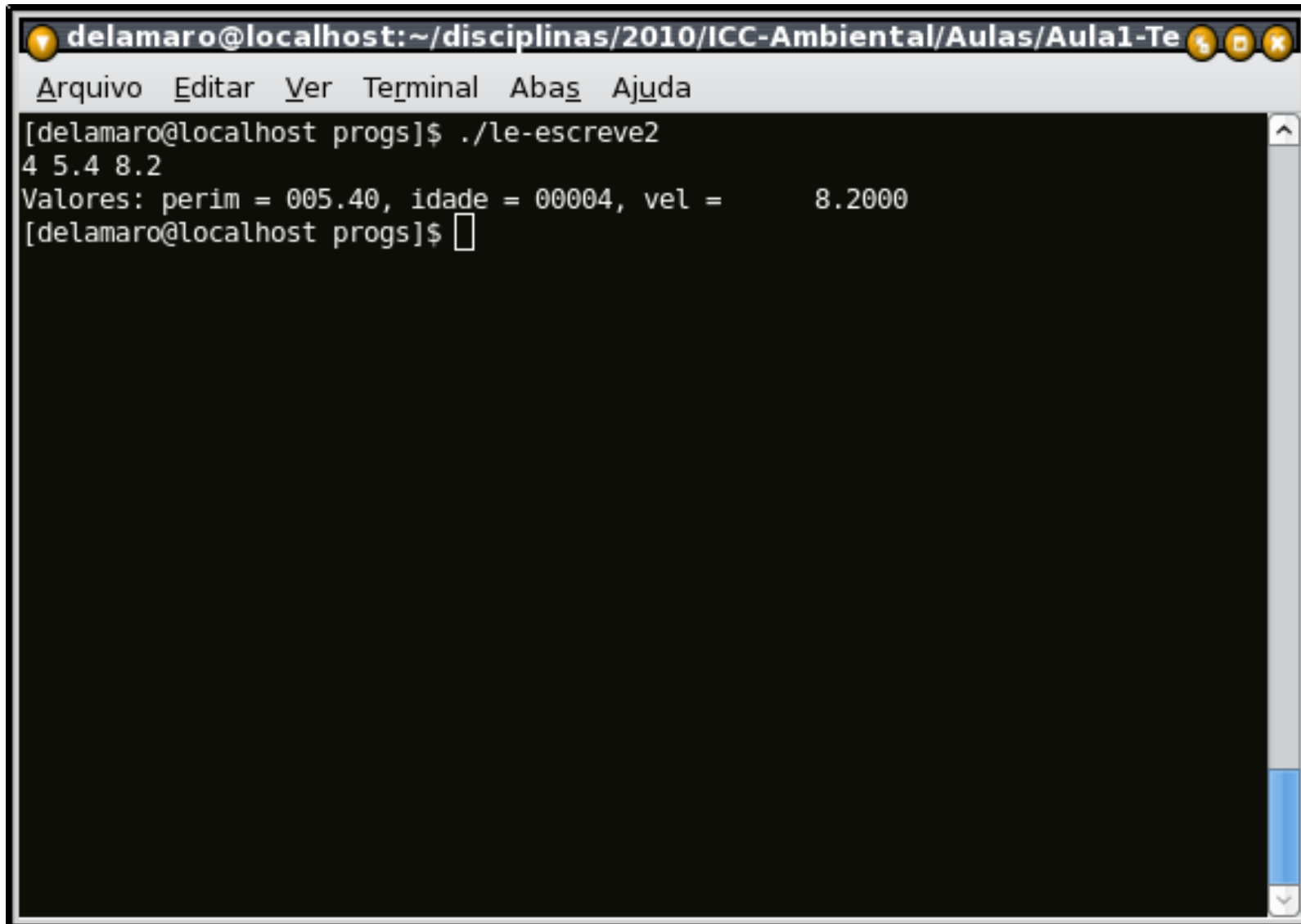
# Exemplo – mais formatação

```
#include <stdio.h>

int main()
{
int idade;
float perimetro;
double velocidade_media;

scanf("%5d %f %lf", &idade, &perimetro,
      &velocidade_media);
printf("Valores: perim = %06.2f, idade = %05d, "\
      "vel = %11.4lf \n",
      perimetro, idade, velocidade_media);
}
```

# printf – exemplo



A terminal window titled "delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te" with standard window controls. The menu bar includes "Arquivo", "Editar", "Ver", "Terminal", "Abas", and "Ajuda". The terminal content shows the execution of a program named "le-escreve2" which outputs the values "4 5.4 8.2" and a formatted string "Valores: perim = 005.40, idade = 00004, vel = 8.2000".

```
delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[delamaro@localhost progs]$ ./le-escreve2
4 5.4 8.2
Valores: perim = 005.40, idade = 00004, vel = 8.2000
[delamaro@localhost progs]$
```

# Expressões

- É possível computar novos valores usando variáveis, constantes e operadores
- Por exemplo  $idade * 5$
- Cada expressão também possui um tipo
- $idade * 5.0$  é double
- Prioridade na execução:  $((a + b) * (c - d)) / 2$

# Outras funções

- Além das funções de entrada e saída existem outras que servem para realizar algum tipo de cálculo
- Essas funções podem ser usadas dentro de expressões
- Por exemplo, funções matemáticas
  - seno e coseno:  $\sin(3.1416) * \cos(3.1416)$
  - tangente:  $\tan(3.1416/2)$
  - raiz quadrada:  $\text{sqrt}(2345678)$

# Expressões – exemplo

Vamos escrever um programa que leia dois números que representam os catetos de um triângulo retângulo e calcule o valor da hipotenusa.

# Expressões – exemplo

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
}
```

# Expressões – exemplo

```
#include <stdio.h>
#include <math.h>

int main()
{

}
```

# Expressões – exemplo

```
#include <stdio.h>
#include <math.h>

int main()
{
double cateto1, cateto2;

}
```



# Expressões – exemplo

```
#include <stdio.h>
#include <math.h>

int main()
{
    double cateto1, cateto2;

    printf("Entre com o tamanho do primeiro lado: ");
    scanf("%lf", &cateto1);

}
```

# Expressões – exemplo

```
#include <stdio.h>
#include <math.h>

int main()
{
double cateto1, cateto2;

    printf("Entre com o tamanho do primeiro lado: ");
    scanf("%lf", &cateto1);
    printf("Entre com o tamanho do segundo lado: ");
    scanf("%lf", &cateto2);

}
```

# Expressões – exemplo

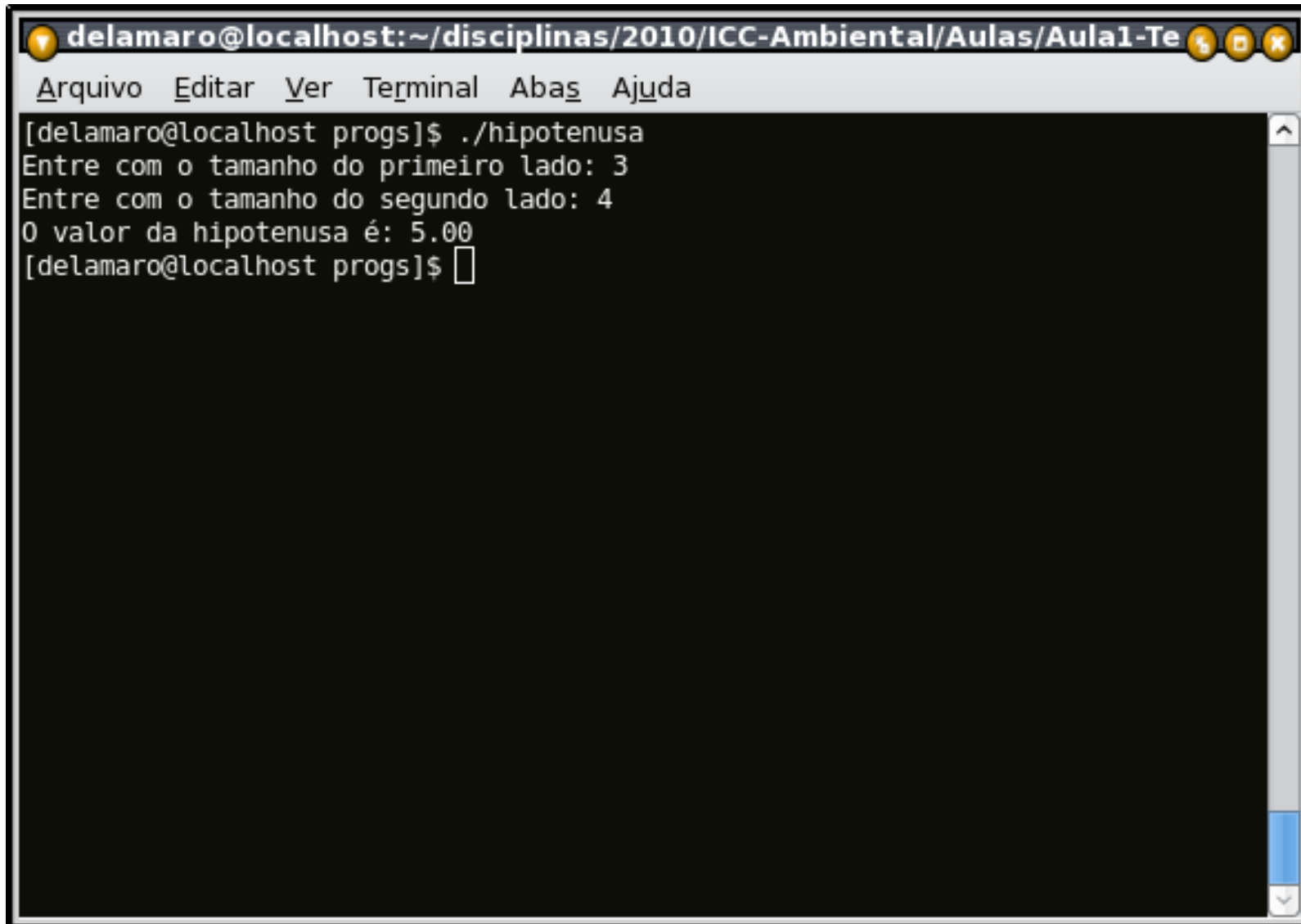
```
#include <stdio.h>
#include <math.h>

int main()
{
double cateto1, cateto2;

    printf("Entre com o tamanho do primeiro lado: ");
    scanf("%lf", &cateto1);
    printf("Entre com o tamanho do segundo lado: ");
    scanf("%lf", &cateto2);

    printf("O valor da hipotenusa é: %-5.2lf\n",
           sqrt(cateto1 * cateto1 + cateto2 * cateto2));
}
```

# Expressões – exemplo



```
delamaro@localhost:~/disciplinas/2010/ICC-Ambiental/Aulas/Aula1-Te
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[delamaro@localhost progs]$ ./hipotenusa
Entre com o tamanho do primeiro lado: 3
Entre com o tamanho do segundo lado: 4
O valor da hipotenusa é: 5.00
[delamaro@localhost progs]$
```

# Operadores aritméticos

- + soma (int, float, double)
- - subtração (int, float, double)
- \* multiplicação (int, float, double)
- / divisão (int, float, double)
- % módulo da divisão (resto da divisão **inteira**)
- - sinal negativo (operador unário)

# Operadores relacionais

- Servem para comparar dois valores
- O resultado é sempre 0 ou 1
  - 0 quer dizer que a relação não se aplica (expressão é falsa)
  - 1 quer dizer que a relação se aplica (expressão é verdadeira)
- Operadores:  $>$  (maior)  $<$  (menor)  $\geq$  (maior ou igual)  $\leq$  (menor ou igual)  $==$  (igual)  $\neq$  (diferente)
- $5 > 10$  produz como resultado 0
- $5 < 10$  produz como resultado 1

# Op relacionais – exemplo

```
#include <stdio.h>
#include <math.h>

int main()
{
double cateto1, cateto2;

printf("Entre com o tamanho do primeiro lado: ");
scanf("%lf", &cateto1);
printf("Entre com o tamanho do segundo lado: ");
scanf("%lf", &cateto2);
printf("O cateto 1 é maior que o 2: %d\n", cateto1 > cateto2);

printf("O valor da hipotenusa é: %-5.2lf\n", sqrt(cateto1 *
}
```

# Comando de atribuição

- Permite que um valor seja armazenado em uma variável
- Lembra do comando “guardar em” ?
- É importante que os tipos dos dados sejam os mesmos
- ```
int idade;  
idade = 32;
```
- ```
int idade;  
idade = 3.1416; /* ERRADO */
```



# Atribuição – exemplo

```
#include <stdio.h>
#include <math.h>
int main()
{
double cateto1, cateto2;
double hipo, cat1_2, cat2_2;

printf("Entre com o tamanho do primeiro lado: ");
scanf("%lf", &cateto1);
printf("Entre com o tamanho do segundo lado: ");
scanf("%lf", &cateto2);
cat1_2 = cateto1 * cateto1;
cat2_2 = cateto2 * cateto2;
hipo = cat1_2 + cat2_2;
hipo = sqrt(hipo);
printf("O valor da hipotenusa é: %-5.2lf\n", hipo);
}
```

# Comentarios

- Comentarios servem para explicar o que os comandos do seu programa fazem
- Utilize sempre comentarios pois eles fazem com que você ou outras pessoas entendam qual o objetivo do seu código
- existem dois tipos de comentarios
  - de uma linha: `//`
  - de múltiplas linhas `/* */`

# Comentários – exemplo

```
// le o primeiro valor
printf("Entre com o tamanho do primeiro lado: ");
scanf("%lf", &cateto1);
// le o segundo valor
printf("Entre com o tamanho do segundo lado: ");
scanf("%lf", &cateto2);
// computa a hipotenusa
cat1_2 = cateto1 * cateto1;
cat2_2 = cateto2 * cateto2;
hipo = cat1_2 + cat2_2;
hipo = sqrt(hipo);
// mostra o valor do resultado
printf("O valor da hipotenusa é: %-5.2lf\n", hipo);
```

# Comentários – exemplo

```
#include <stdio.h>
#include <math.h>

/* Esse programa computa o valor da hipotenusa de um
triangulo retangulo. Os valores dos dois catetos devem
ser fornecidos pelo usuário */

int main()
{
double cateto1, cateto2;
double hipo, cat1_2, cat2_2;

    ...
}
```

# Exercícios

- a. Faça um programa que leia duas notas de um aluno (nota 1 e nota 2) fornecidas pelo usuário que irá entrar as notas digitando pelo teclado. Usando estas 2 notas, calcule a média simples do aluno, e depois mostre na tela o resultado da média calculada.
- b. Faça um programa que leia 3 notas de um aluno, onde a primeira e segunda nota possuem peso 1 e a terceira nota possui peso 2. Calcule a média ponderada destas notas, usando os pesos, e depois mostre na tela o resultado (exibir a média com apenas 2 casas após a vírgula).
- c. Faça um programa que leia uma temperatura fornecida em graus Celsius ( $^{\circ}\text{C}$ ) e converta para graus Fahrenheit ( $^{\circ}\text{F}$ ), exibindo o resultado na tela.
- d. Faça um programa que leia o valor da hora de trabalho (em reais) e número de horas trabalhadas no mês, e exiba na tela o valor a ser pago ao funcionário, adicionando 10% sobre o valor calculado.

# Exercícios

- a. Faça um programa que compute as soluções da equação  $Ax^2 + Bx + C = 0$ . Os valores de A, B e C devem ser fornecidos pelo usuário
- b. O valor do seno de  $x$  pode ser calculado pela série de Taylor, dada por:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots$$

Faça um programa que leia o valor de  $x$  e compute o valor do seno usando os 5 primeiros termos da série de Taylor. Mostre o valor calculado e o valor da função padrão  $\sin(x)$ .