

Organização de programas em Pascal

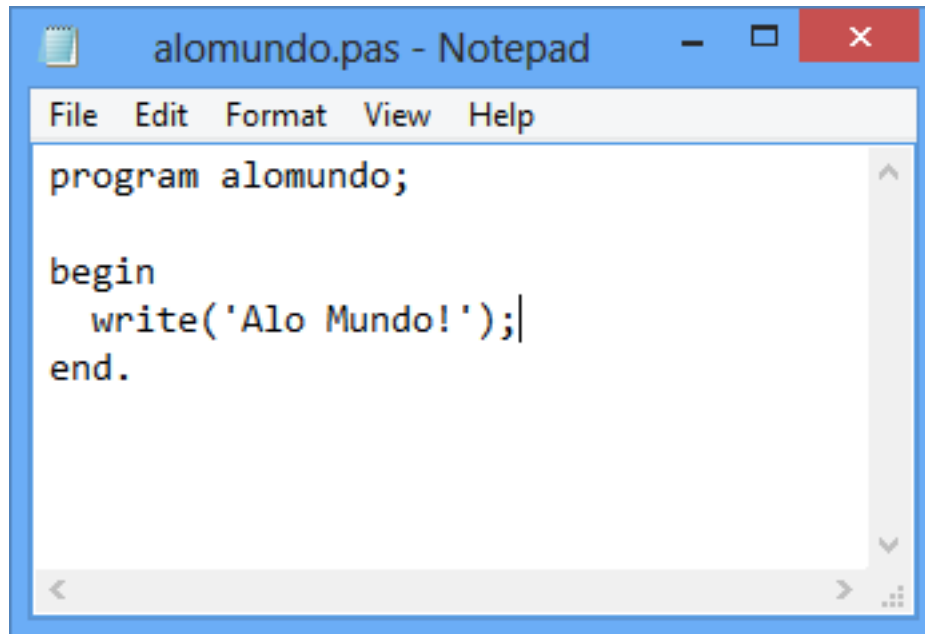
Leonardo Gresta Paulino Murta
leomurta@ic.uff.br

Vamos programar em Pascal! Mas...

- Como um programa é organizado?
- Quais são os tipos de dados disponíveis?
- Como variáveis podem ser declaradas?
- Como atribuir valores às variáveis?
- Como entrada e saída básica de dados podem ser feitas?

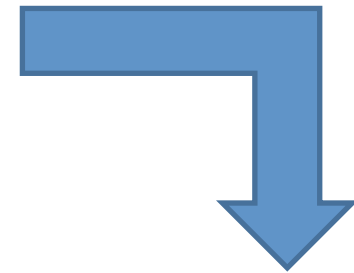
Vamos começar com um exemplo...

Primeiro passo: escrever o programa!



```
alomundo.pas - Notepad
File Edit Format View Help
program alomundo;

begin
  write('Alo Mundo!');
end.
```



alomundo.pas

Mas o computador não conhece Pascal!!!

- O computador só entende binário
 - Linguagem de zeros e uns
 - 010010011101010101001010101, entendeu?
- Precisamos traduzir o programa de Pascal para binário
- Esse processo é conhecido como **compilação**

Instalação do Compilador

- Download do compilador Free Pascal
 - <http://www.freepascal.org>
 - Versão mais recente
- Programa principal
 - fpc (compilador)

Compilação



```
program alomundo;  
  
Begin  
  write('Alo Mundo!');  
end.
```



fpc



Binário

```
00101110011110  
10010110000110  
10011100011010
```

Compilação

```

C:\Users\Leonardo\prog1>dir
Volume in drive C has no label.
Volume Serial Number is 06C8-507B

Directory of C:\Users\Leonardo\prog1

02/03/2014  18:55    <DIR>          .
02/03/2014  18:55    <DIR>          ..
02/03/2014  18:05                57 alomundo.pas
               1 File(s)              57 bytes
               2 Dir(s)    4.430.966.784 bytes free

C:\Users\Leonardo\prog1>fpc alomundo.pas
Free Pascal Compiler version 2.6.2 [2013/02/12] for i386
Copyright (c) 1993-2012 by Florian Klaempfl and others
Target OS: Win32 for i386
Compiling alomundo.pas
Linking alomundo.exe
4 lines compiled, 0.1 sec , 25536 bytes code, 1628 bytes data

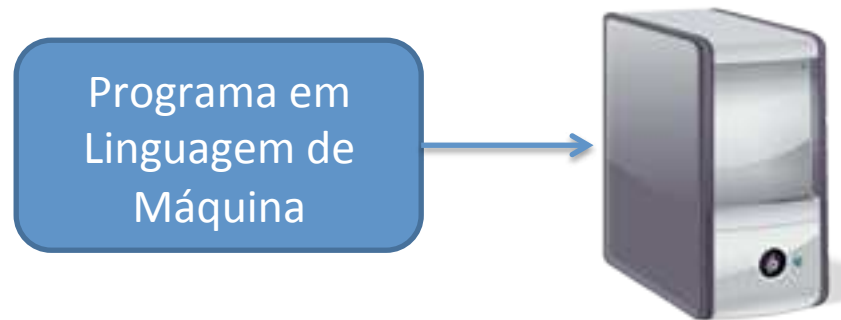
C:\Users\Leonardo\prog1>dir
Volume in drive C has no label.
Volume Serial Number is 06C8-507B

Directory of C:\Users\Leonardo\prog1

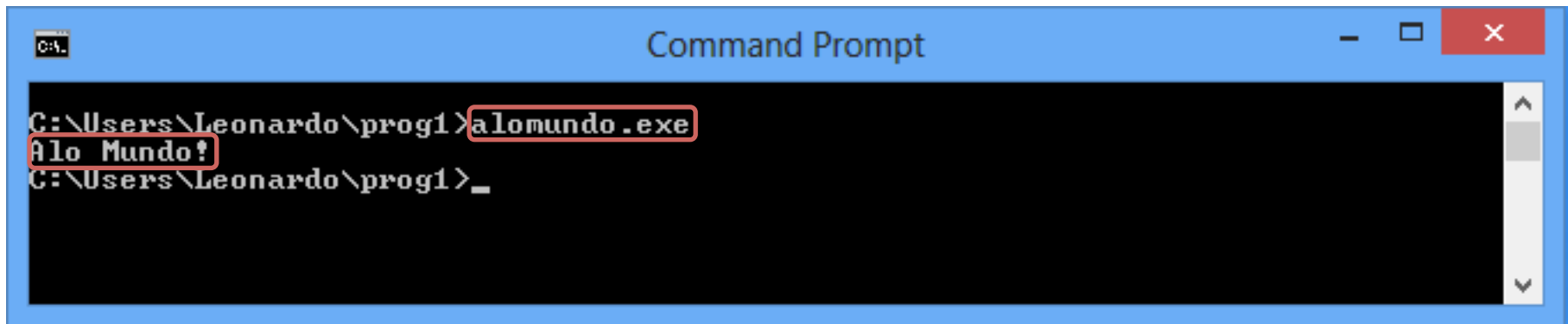
02/03/2014  18:55    <DIR>          .
02/03/2014  18:55    <DIR>          ..
02/03/2014  18:55            31.232 alomundo.exe
02/03/2014  18:55             2.668 alomundo.o
02/03/2014  18:05                57 alomundo.pas
               3 File(s)              33.957 bytes
               2 Dir(s)    4.430.917.632 bytes free

```

Execução



Execução



The image shows a Windows Command Prompt window with a blue title bar and standard window controls. The command prompt is open in the directory C:\Users\Leonardo\prog1. The user has entered the command 'alomundo.exe', which is highlighted with a red box. The output of the program is 'Alo Mundo!', also highlighted with a red box. The prompt is now waiting for the next command.

```
C:\Users\Leonardo\prog1>alomundo.exe
Alo Mundo!
C:\Users\Leonardo\prog1>_
```

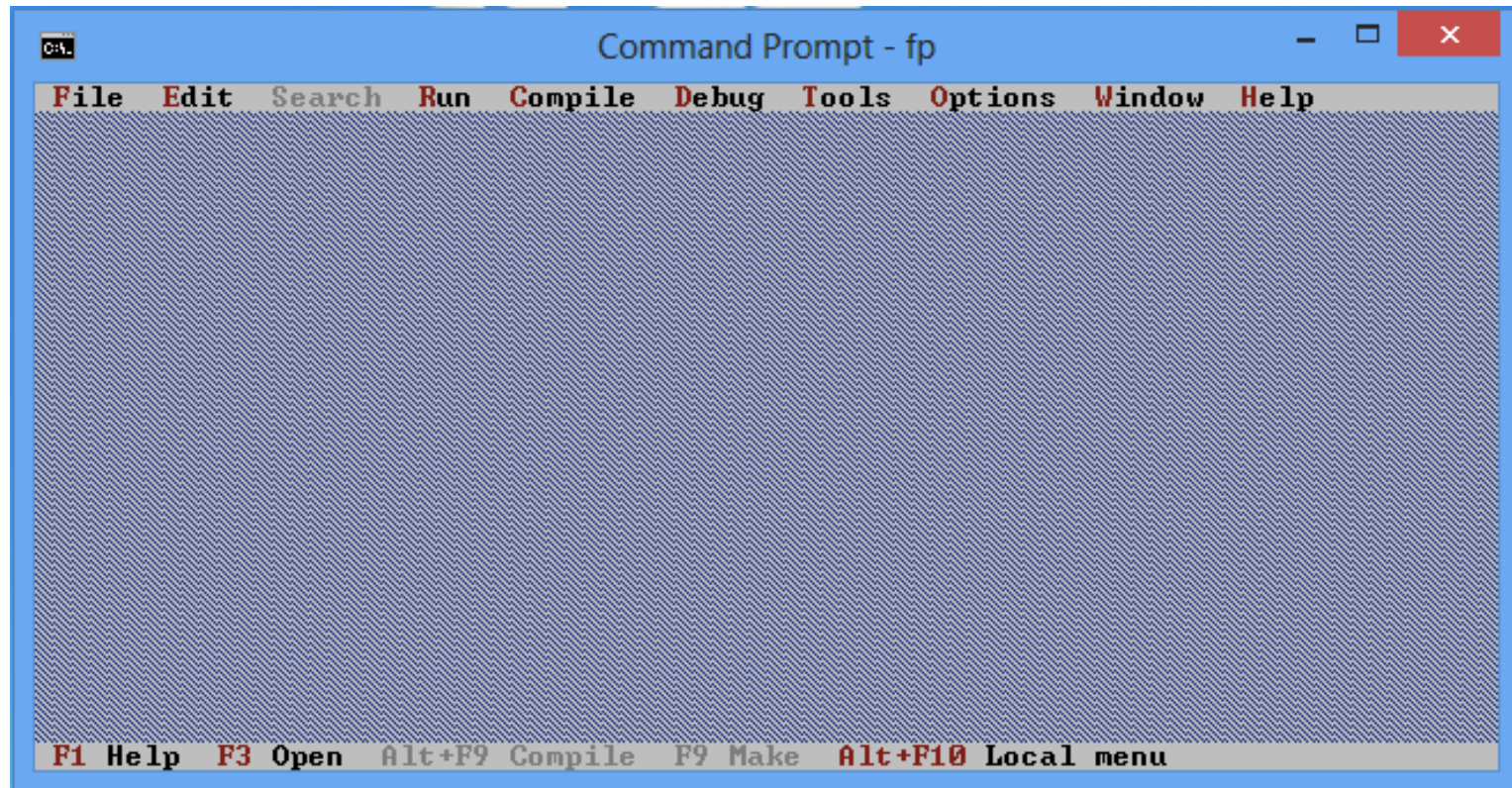
VAMOS FAZER JUNTOS?

Notepad x IDE

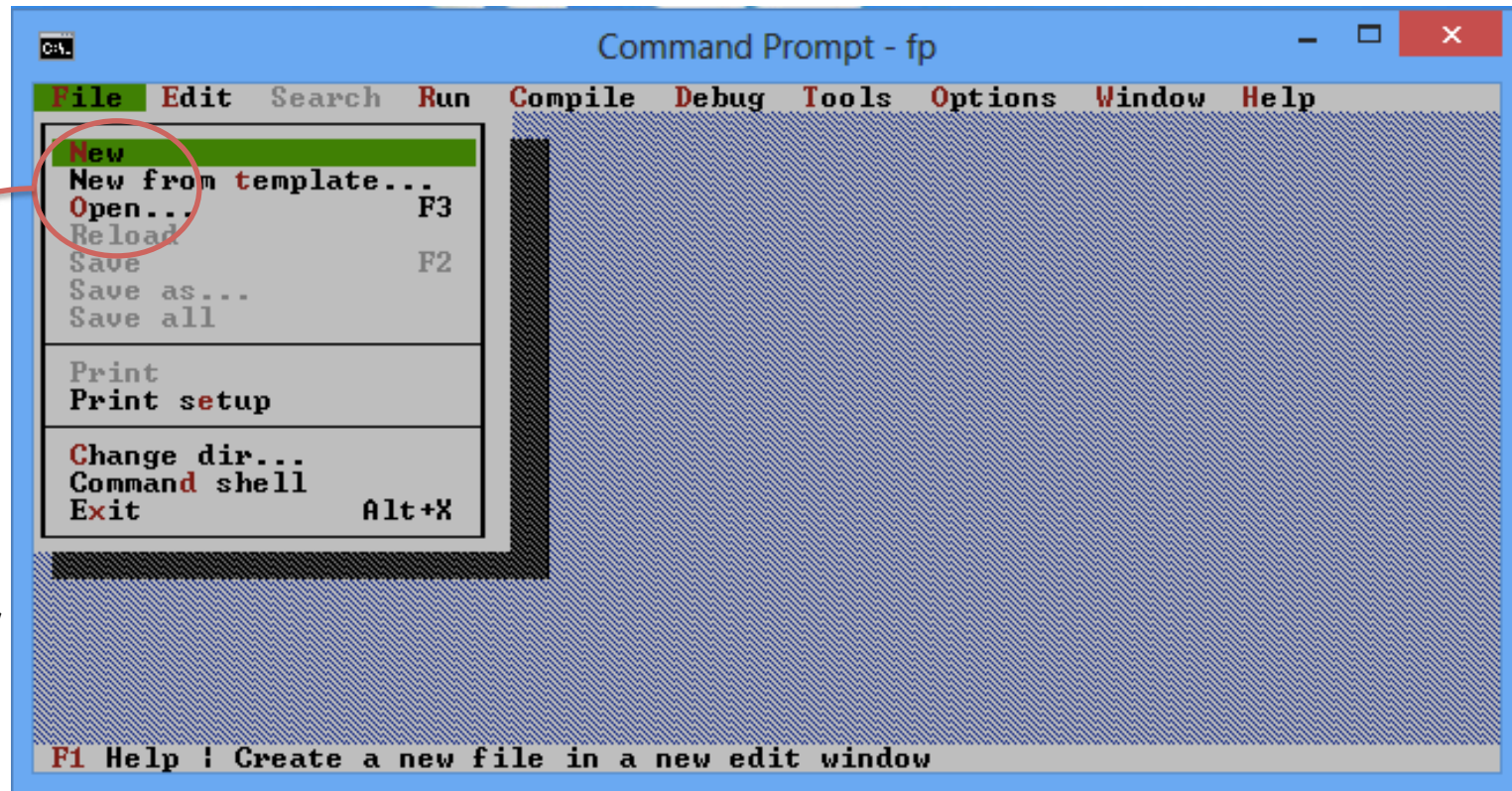
- Dificuldades do Notepad
 - Editor básico, sem ajuda para programar
 - Compilação externa
 - Execução externa
- *Integrated Development Environment (IDE)*
 - Editor
 - Compilador
 - Executor
 - Depurador

IDE Free Pascal Modo Texto

- Já vem com o Free Pascal
- Basta rodar o comando “fp”

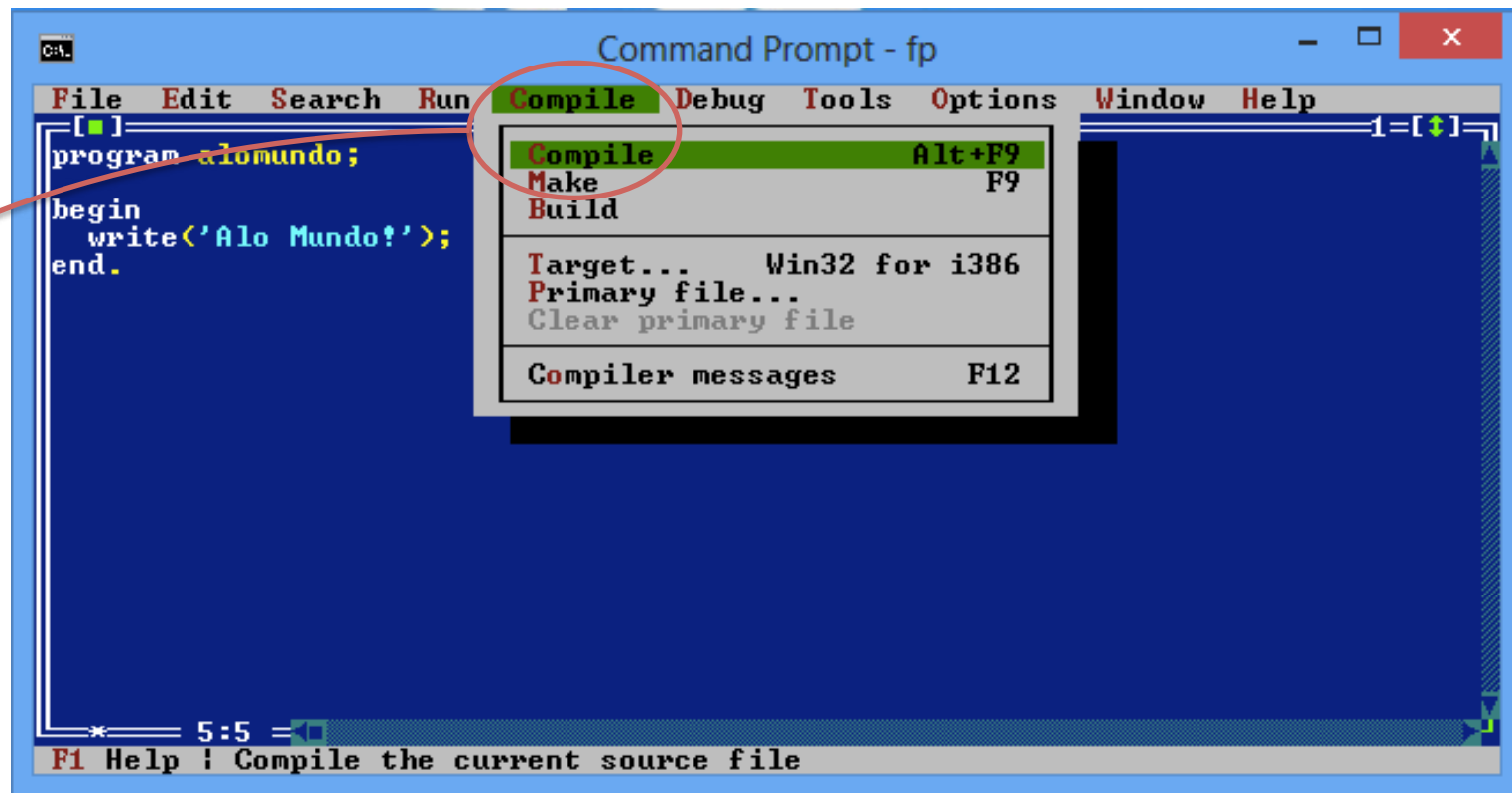


IDE Free Pascal Modo Texto



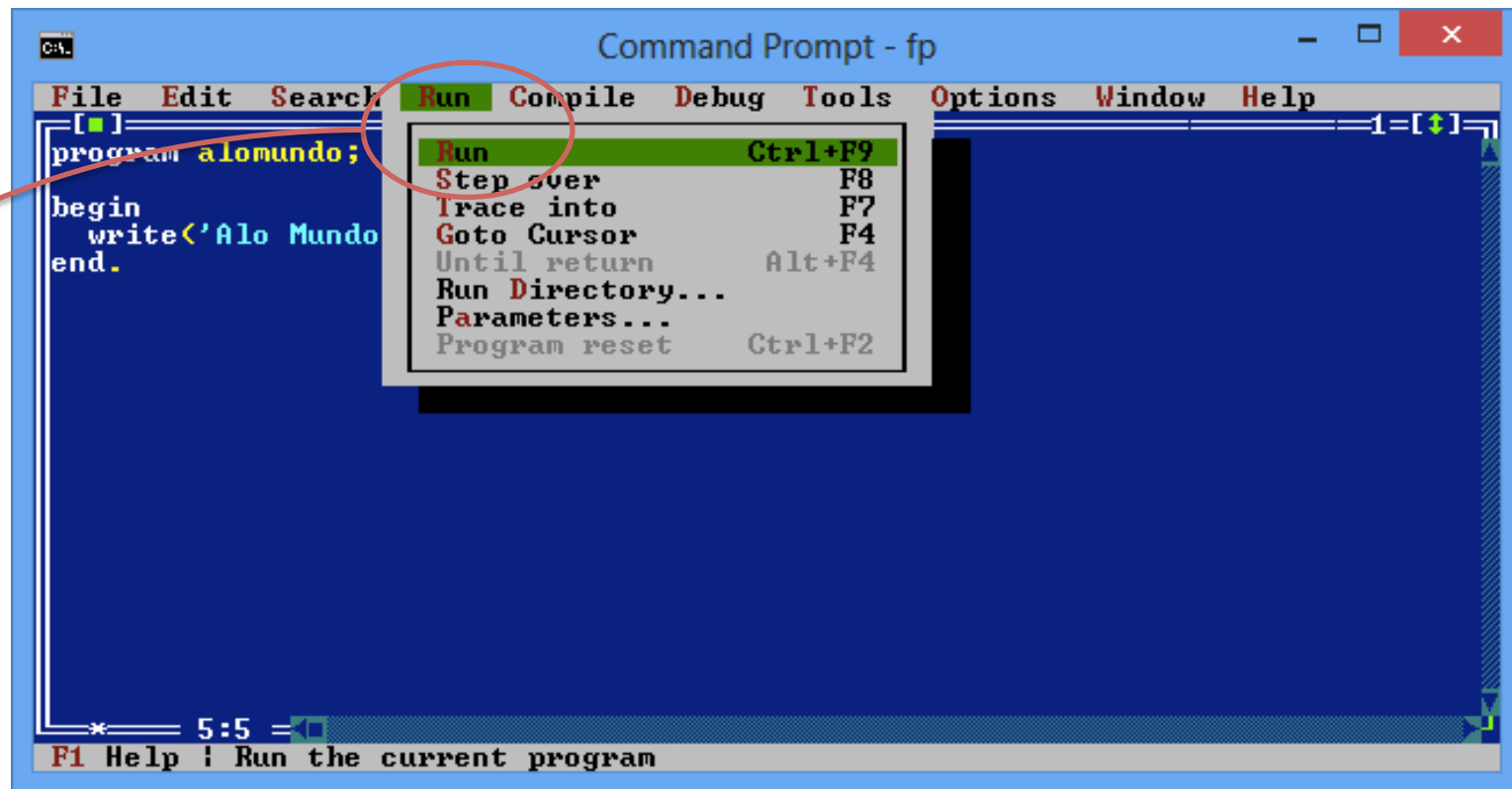
Selecionar *New*
para **criar um
novo
programa** ou
Open para **abrir
um programa
existente**

IDE Free Pascal Modo Texto



Selecionar
Compile para
compilar o
programa

IDE Free Pascal Modo Texto



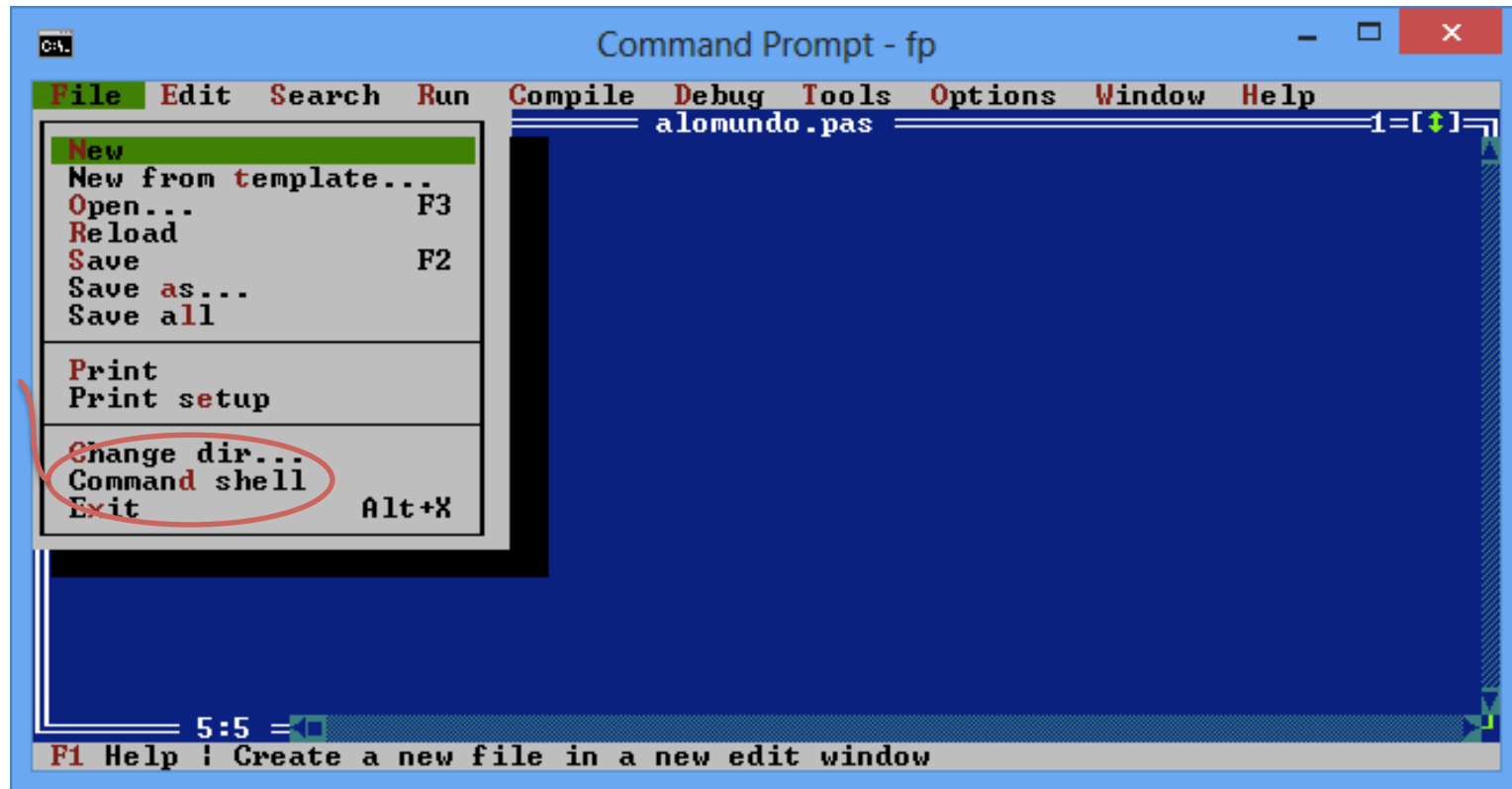
Selecionar *Run*
para **executar**
o programa

IDE Free Pascal Modo Texto

Selecionar
Command shell
para ver a
saída do
programa
(escreva *exit*
para retornar)

OU

colocar *readln*
no final do
programa para
ver a saída
antes de fechar
a janela

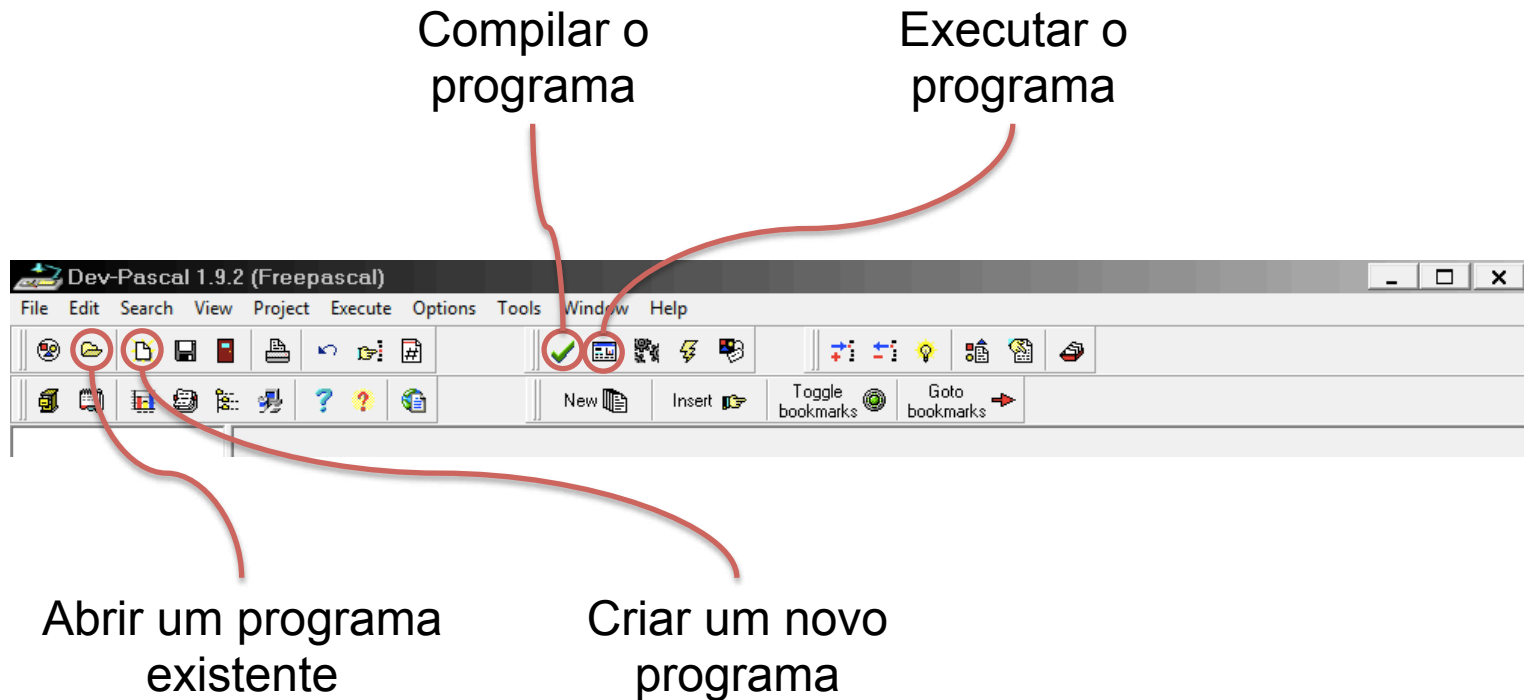


VAMOS FAZER JUNTOS?

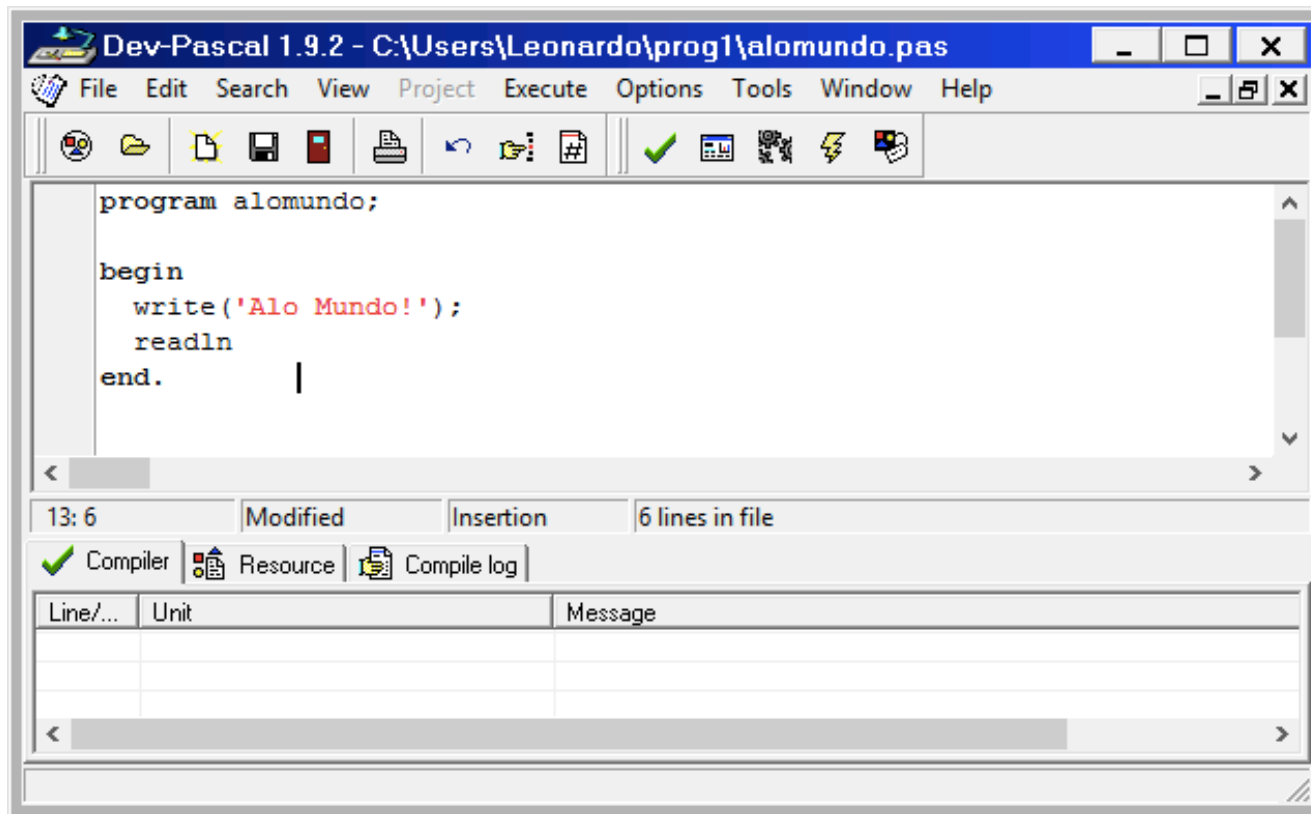
IDE Dev-Pascal

- Usaremos o Dev-Pascal nas aulas, mas os alunos podem optar por qualquer outra IDE ou editor
- Download do Dev-Pascal
 - <http://www.bloodshed.net/devpascal.html>
 - Versão mais recente

IDE Dev-Pascal



Escrevendo, compilando e executando o programa no Dev-Pascal...



VAMOS FAZER JUNTOS?

Organização de um programa básico em Pascal

```

program NOME DO PROGRAMA;

const
  NOME = VALOR;
  ...

var
  NOME 1, ..., NOME N : TIPO;
  ...

begin
  CÓDIGO DO PROGRAMA
end.

```

Regras básicas

; no final dos comandos!
(opcional para último
comando do programa)

begin e end
delimitam blocos!

último end
termina com ponto!

Comentários

- Comentários são trechos do programa voltados para a leitura por humanos, e ignorados pelo interpretador
- **{ COMENTÁRIO }**
 - Tudo entre { e } é ignorado pelo compilador
- **(* COMENTÁRIO *)**
 - Tudo entre (* e *) é ignorado pelo compilador

Exemplo de programa em Pascal

```
(* Este programa calcula a área
   de um triângulo retângulo *)
program triangulo;

var
  altura, base : integer; { dados de entrada }
  area : real; { dados de saída }

begin
  write('Informe a altura: ');
  readln(altura);
  write('Informe a base: ');
  readln(base);
  area := 0.5 * altura * base;
  writeln('Área: ', area);
end.
```

Quais são os tipos de dados disponíveis?

- Em Pascal, toda variável tem que ter um tipo
- Com isso, o computador pode **reservar o espaço correto de memória**
- Os tipos básicos podem ser divididos em dois grupos
 - Tipos numéricos (inteiro e real)
 - Tipos textuais (caractere e texto)
 - Tipo lógico (booleano)

Tipos Numéricos

- integer
 - Número inteiro
 - Precisão dependente da implementação da linguagem
 - Para 32 bits, de -2.147.483.648 a 2.147.483.647
- real
 - Número real
 - Precisão dependente da implementação da linguagem
 - Para 32-bits, 7 casas decimais com magnitude de 10^{38}
 - Pode fazer uso de notação científica: $5.3E2 = 5.3 \times 10^2 = 530$

Exemplos de números inteiros

- integer

- -12312312
- 10
- 345092834

- real

- -21.4
- 0.0000034
- 123456.0
- 0.6023e24
- 0.4e-3

Tipos Textuais

- char
 - Caractere (letra, número, símbolos)
- string
 - Texto de até 255 letras
 - Pode informar o tamanho máximo entre colchetes (ex. `string[5]`)

Exemplos de tipos textuais

- char
 - ‘A’
 - ‘b’
 - ‘4’
- string
 - “
 - ‘Olá mundo!’
 - ‘4’

Tipos Lógicos

- boolean
 - Tipo lógico, com valores *true* ou *false*

Valores padrão

- Algumas linguagens não limpam o espaço de memória ao alocar uma nova variável
- Pascal usualmente toma esse cuidado para nós
 - Tipos numéricos são inicializados com 0
 - Tipos textuais são inicializados com “
 - Tipo lógico é inicializado com *false*
- De qualquer forma, sempre inicialize as suas variáveis por precaução

Declaração de variáveis

- Para serem usadas, as variáveis precisam ser declaradas (criadas)
- Toda variável é declarada da seguinte forma:

```
var
  NOME : TIPO;
```

OU

```
var
  NOME1, NOME2, ... : TIPO;
```

Declaração de variáveis

- Os tipos são os que já vimos
- Os nomes devem respeitar algumas regras
 - São insensíveis a caixa
 - Podem ter tamanho ilimitado (mas evite abusos)
 - Somente os primeiros caracteres são considerados como identificadores (ex.: 64 caracteres, dependendo da implementação da linguagem)
 - Devem começar com letra
 - Não podem ter espaço nem acentos
 - Não podem ser uma palavra reservada da linguagem

Declaração de variáveis

- Um caso especial é referente a variáveis que nunca trocam de valor
 - Mais conhecidas como **constante**
- Em Pascal, constantes são declaradas da seguinte forma:

```
const
  NOME = VALOR;
  . . .
```


Atribuição de valores

- Em Pascal, o operador `:=` é usado para atribuir valores às variáveis
- É equivalente ao símbolo de atribuição (\leftarrow) que usávamos no pseudocódigo
- Sempre na forma: **variável := valor ou expressão**
 - A expressão do lado direito é processada
 - O valor gerado é atribuído à variável

Como constantes e variáveis podem ser declaradas ou inicializadas?

- idade := 15;
- minutos := horas * 60;
- const ACELERACAO_GRAVIDADE = 9.80665;
- const PI = 3.14159265358979;
- const MELHOR_TIME = 'Flamengo';
- gosto_prog1 := true;
- var nome, endereco, telefone : string;
- var ano, mes, dia : integer;

Saída de dados

- Para saída de dados, é necessário usar uma função
 - write(EXPRESSÕES)
 - writeln(EXPRESSÕES)
- Write
 - Escreve o valor de expressões no monitor na sequência informada (separadas por vírgula)
- Writeln
 - Equivalente ao write, e desloca para a linha de baixo ao final
- Exemplo
 - write('Fla é o...');
 - write('melhor!!!');
 - writeln('Você tem ', x, ' vidas');
 - write('Recomeçar (S/N)? ');

Entrada de dados

- Para entrada de dados, é necessário usar uma função
 - read(VARIÁVEIS)
 - readln(VARIÁVEIS)
- Read
 - Le valores do teclado e atribui às variáveis informadas em sequência (separadas por vírgula)
- Readln
 - Equivalente ao read, mas despreza o restante da linha assim que completar as variáveis
 - Atende à maioria das necessidades
- A leitura só ocorre de fato após o usuário teclar *Enter*
- Exemplos:
 - readln(nome);
 - readln(endereco);

Entrada de dados

- Suponha a seguinte entrada pelo teclado:

1 2 3

4 5 6

7 8 0

- Três chamadas a `read(i)` atribuiriam respectivamente os valores 1, 2 e 3 à variável `i`
- Três chamadas a `readln(i)` atribuiriam respectivamente os valores 1, 4 e 7 à variável `i`

Exercícios

- Qual a saída do programa abaixo?

```

program atribuicoes;

var x, y, z : real;

begin
  x := 1.0;
  y := 2.0;
  z := 3.0;

  x := -x;
  y := y - 1;
  z := z + x;
  z := z + x - y;

  writeln('x = ', x, ', y = ', y, ', z = ', z);
end.

```

Exercícios

- Faça um programa que leia o nome, a idade, a altura, o peso e a nacionalidade do usuário e escreva essas informações na forma de um parágrafo de apresentação
- Faça um programa que exiba o perímetro de uma circunferência a partir do seu raio
- Faça um programa que leia dois pontos num espaço bidimensional e calcule a distância entre esses pontos

Exercícios

- Faça um programa que informe a distância em quilômetros de um raio para o observador
 - O observador deve informar o tempo transcorrido entre ver o raio e ouvir o trovão
 - Assuma que a velocidade do som seja 340 m/s
 - Assuma que a velocidade da luz tende a infinito

Exercícios

- Faça um programa para, a partir de um valor informado em centavos, indicar a menor quantidade de moedas que representa esse valor
 - Considere moedas de 1, 5, 10, 25 e 50 centavos, e 1 real
 - Exemplo: para o valor 290 centavos, a menor quantidade de moedas é 2 moedas de 1 real, 1 moeda de 50 centavos, 1 moeda de 25 centavos, 1 moeda de 10 centavos e 1 moeda de 5 centavos

Organização de programas em Pascal

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br