



Paradigmas de Linguagens de Programação

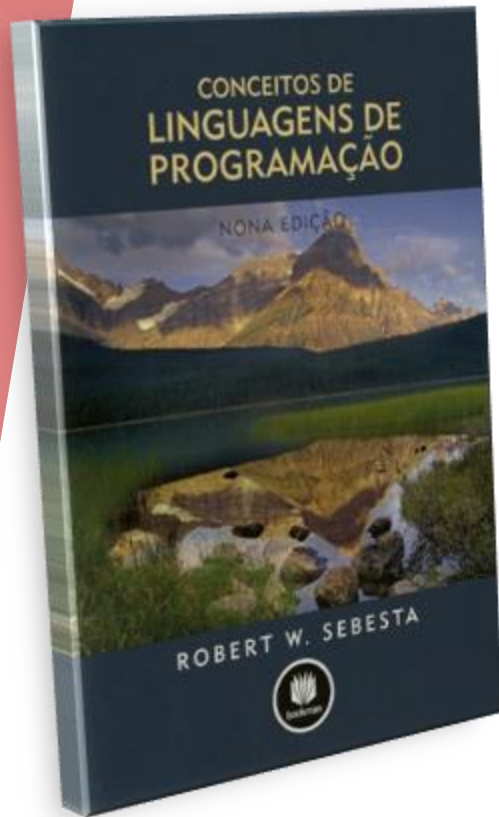
Introdução



Prof. Ausberto S. Castro V.

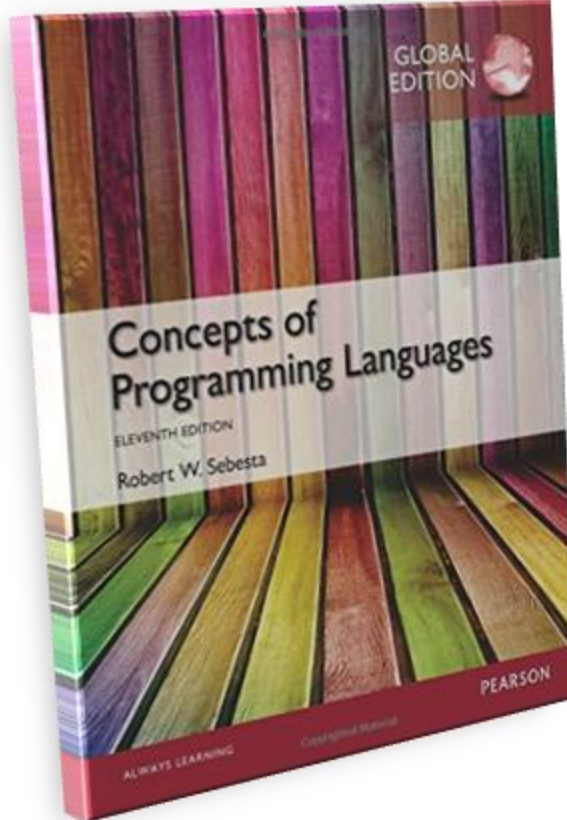
ascv@uenf.br

Bibliografia Básica

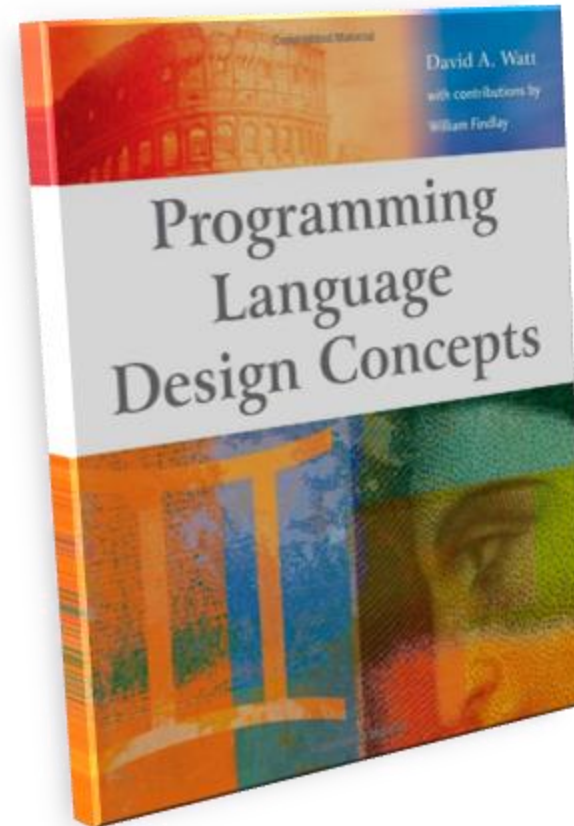


Bibliografia Complementar

Pearson Education Limited; 11 edition
2016



Wiley; 1 edition (May 21, 2004)



<http://www.levenez.com/lang/>

Linguagens de Programação?

❖ **Mais de 5.000 linguagens de programação (LP)**

❖ **Perguntas:**

- Por que há tantas linguagens de programação diferentes?
- Como e por que elas foram desenvolvidas?
- De que maneira elas são semelhantes?
- Quais são suas diferenças?
- Quais tipos de linguagens podem ser desenvolvidas e usadas no futuro?
- Por que simplesmente não continuaríamos a usar o que temos agora?
- Quem inventou as linguagens?
- O que é uma Linguagem de Programação?

Linguagem de Programação

❖ Linguagem

- Um sistema de sinais utilizados para se comunicar
 - Sistema (conjunto, componentes, complexo)
 - Sinais (símbolos, texto)
 - Comunicar (transmissor, meio, receptor)

❖ Programar

- Escrever um programa
 - Análise (problema)
 - Algoritmo (Lógica)
 - Codificação (Linguagem de Programação)

❖ Programação

- Processo de escrita, teste e manutenção de um programa de computador
 - Escrever (Linguagem)
 - Testar (ferramentas)
 - Manutenção (erros)

Fatorial

```
int fact(int n) {  
    intsofar = 1;  
    while (n>0) sofar *= n--;  
    return sofar;  
}
```

```
fact(X,1) :-  
    X ::= 1.  
fact(X,Fact) :-  
    X > 1,  
    NewX is X - 1,  
    fact(NewX,NF),  
    Fact is X * NF.
```

```
(defun fact (x)  
  (if (<= x 0) 1 (* x (fact (- x 1)))))
```

```
fun fact x =  
  if x <= 0 then 1 else x * fact(x-1);
```

```
public class MyInt {  
    private int value;  
    public MyInt(int value) {  
        this.value = value;  
    }  
    public int getValue() {  
        return value;  
    }  
    public MyInt getFact() {  
        return new MyInt(fact(value));  
    }  
    private int fact(int n) {  
        intsofar = 1;  
        while (n > 1) sofar *= n--;  
        return sofar;  
    }  
}
```

Três conceitos relacionados

1. Linguagem

- Um sistema de sinais utilizados para se comunicar

2. Linguagem de programação

- Um sistema de sinais utilizados por uma pessoa para comunicar uma tarefa/algoritmo a um computador, de modo que a tarefa seja executada

3. Computador

- Algo que computa, calcula
- Computador digital automático de alta velocidade

Linguagem de Programação?

- ❖ Uma ferramenta para instruir máquinas
- ❖ Um meio de comunicação entre programadores
- ❖ Um veículo para expressar projetos de alto nível
- ❖ Uma notação para algoritmos
- ❖ Uma maneira de expressar relacionamentos entre conceitos
- ❖ Uma ferramenta para experimentar
- ❖ Um meio para controlar dispositivos computarizados

Cada linguagem de programação é um artefato e como tal tem sido cuidadosamente projetada.

Criando uma linguagem



Uma pessoa
C++, BASIC



Pequenos grupos
C, FORTRAN



Grandes grupos
ADA

ADA
ALGOL60
ALGOL68
C
C++
C#
COBOL
FORTRAN
JAVA
LISP
HASKELL
ML
MODULA
PASCAL
PERL
PL/I
PROLOG
PYTHON
SIMULA
SMALLTALK

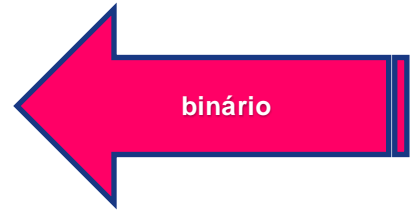
ISO/IEC (1995); www.ada-auth.org/~acats/arm.html
Naur (1963)
van Wijngaarden et al. (1976)
Kernighan and Ritchie (1989); ISO/IEC (1999)
Stroustrup (1997); ISO/IEC (1998)
Drayton et al. (2002)
ISO/IEC (2002)
ISO/IEC (1997)
Joy et al. (2000); Flanagan (2002)
McCarthy et al. (1965); ANSI (1994)
Thompson (1999)
Milner et al. (1997)
Wirth (1977)
ISO (1990)
Wall et al. (2000)
ISO (1979)
Bratko (1990)
Beazley (2001); www.python.org/doc/current/ref/
Birtwhistle et al. (1979)
Goldberg and Robson (1989)

Níveis de Ling. de Programação

- ❖ Uma grosseira distinção entre linguagens de programação
- ❖ **Baseado:**
 - Na leitura
 - Na independência
 - No propósito (geral ... Específico)
- ❖ **Nível**
 - Máquina?
 - Assembler?
 - Baixo nível, Alto nível (3GL)
 - Quarta Geração (4GL): Especificação Formal
 - Quinta Geração (5GL linguagens lógicas), Prolog, OPS5, Mercury

Linguagem de Máquina

00000010101111001010
00000010101111001000
00000011001110101000



Linguagem de
programação
DE
máquina

❖ Utilizado por qualquer máquina (computador)

- Instruções: 1 (ligado) 0 (desligado)
- Somente antes de ter linguagens de alto nível (1940)

❖ Significado

- Alguém pode dizer o significado do código 0011001110101?

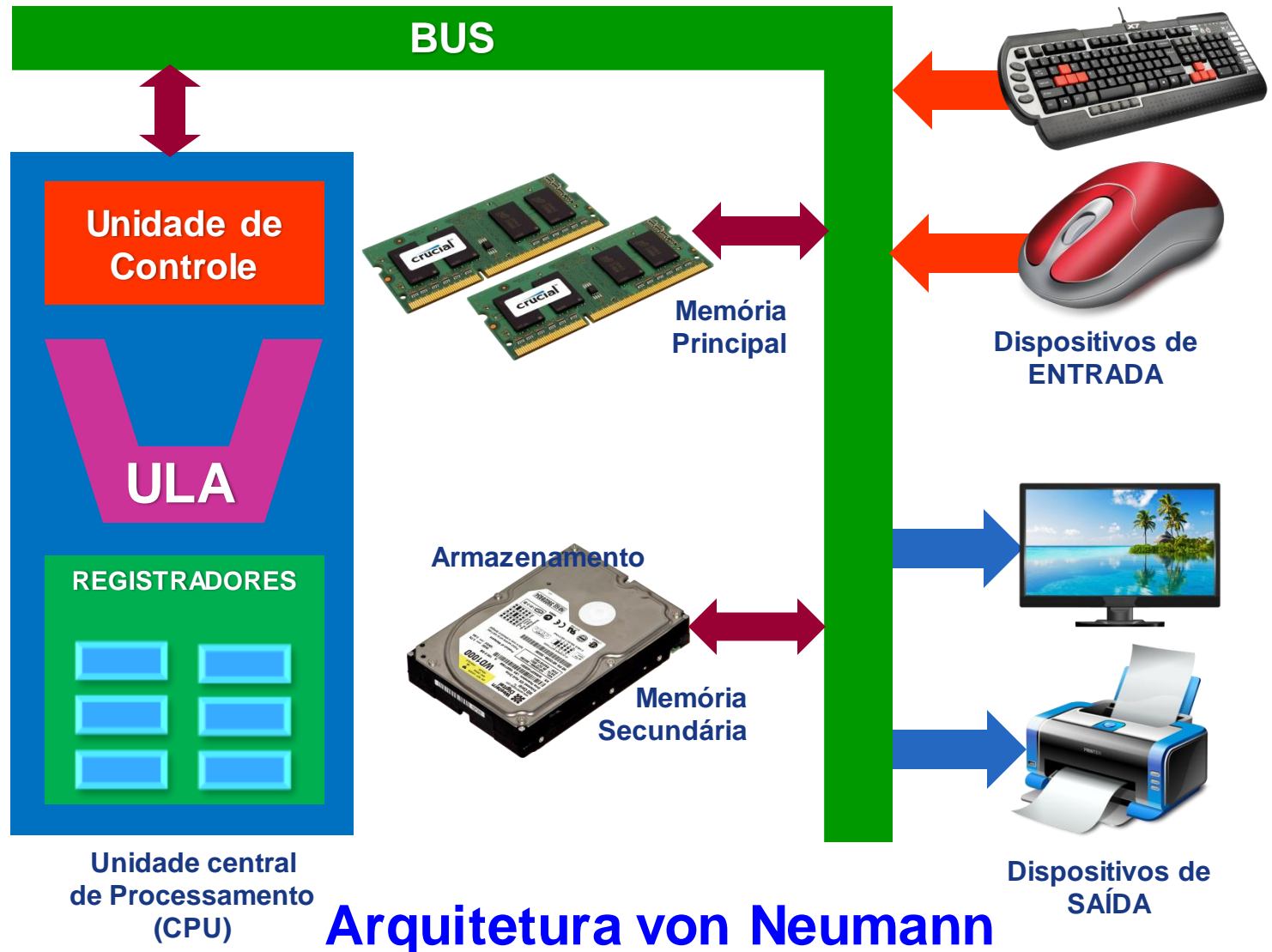
❖ Execução

- Pode ser executado em qualquer máquina?

❖ Uso

- Serve para qualquer propósito (uso geral)?

Linguagem (Software) e Arquitetura (Hardware)



Arquitetura von Neumann

John von Neumann, 1945
"First Draft of a Report on the EDVAC"

Von Neumann - Pólya

John von Neumann, nascido Margittai Neumann János Lajos (Budapeste, 28 de dezembro de 1903 — Washington, D.C., 8 de fevereiro de 1957) foi um matemático húngaro de origem judaica, naturalizado estadunidense.

Contribuiu na teoria dos conjuntos, análise funcional, teoria ergódica, mecânica quântica, **ciência da computação**, economia, teoria dos jogos, análise numérica, hidrodinâmica das explosões, estatística e muitas outras áreas da matemática. De fato é considerado um dos mais importantes matemáticos do século XX. Foi professor na Universidade de Princeton e um dos construtores do ENIAC.

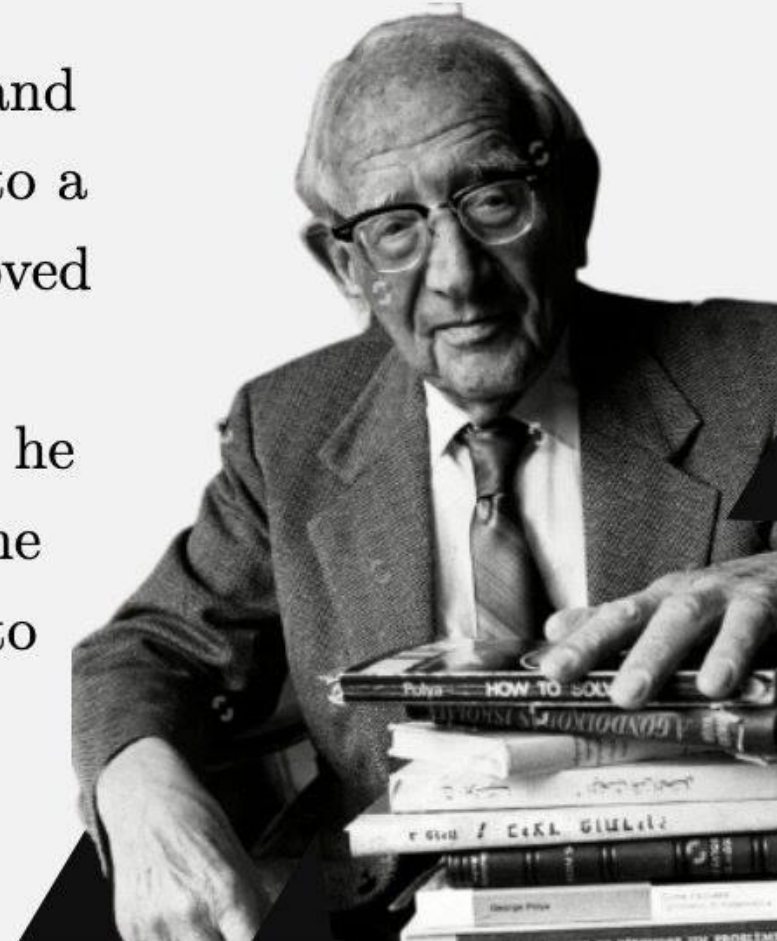
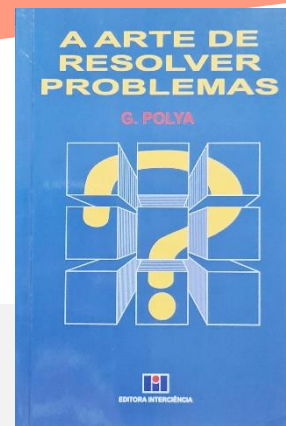
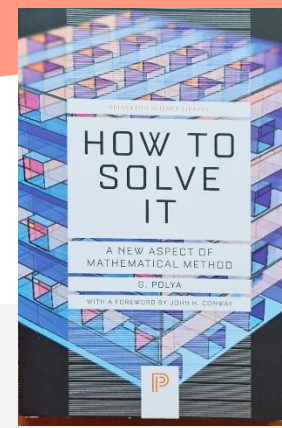


George Pólya, Budapeste, 13 de dezembro de 1887 — Palo Alto, 7 de setembro de 1985 foi um matemático húngaro e professor de matemática de 1914 a 1940 no ETH Zürich na Suíça, e de 1940 a 1953 na Stanford University. Pólya permaneceu como professor emérito de Stanford o resto de sua vida e carreira. Trabalhou com uma variedade de tópicos matemáticos, incluindo séries, teoria dos números, análise matemática, geometria, álgebra, combinatória e probabilidade. Também é notável sua contribuição para a heurística em educação matemática

Von Neumann

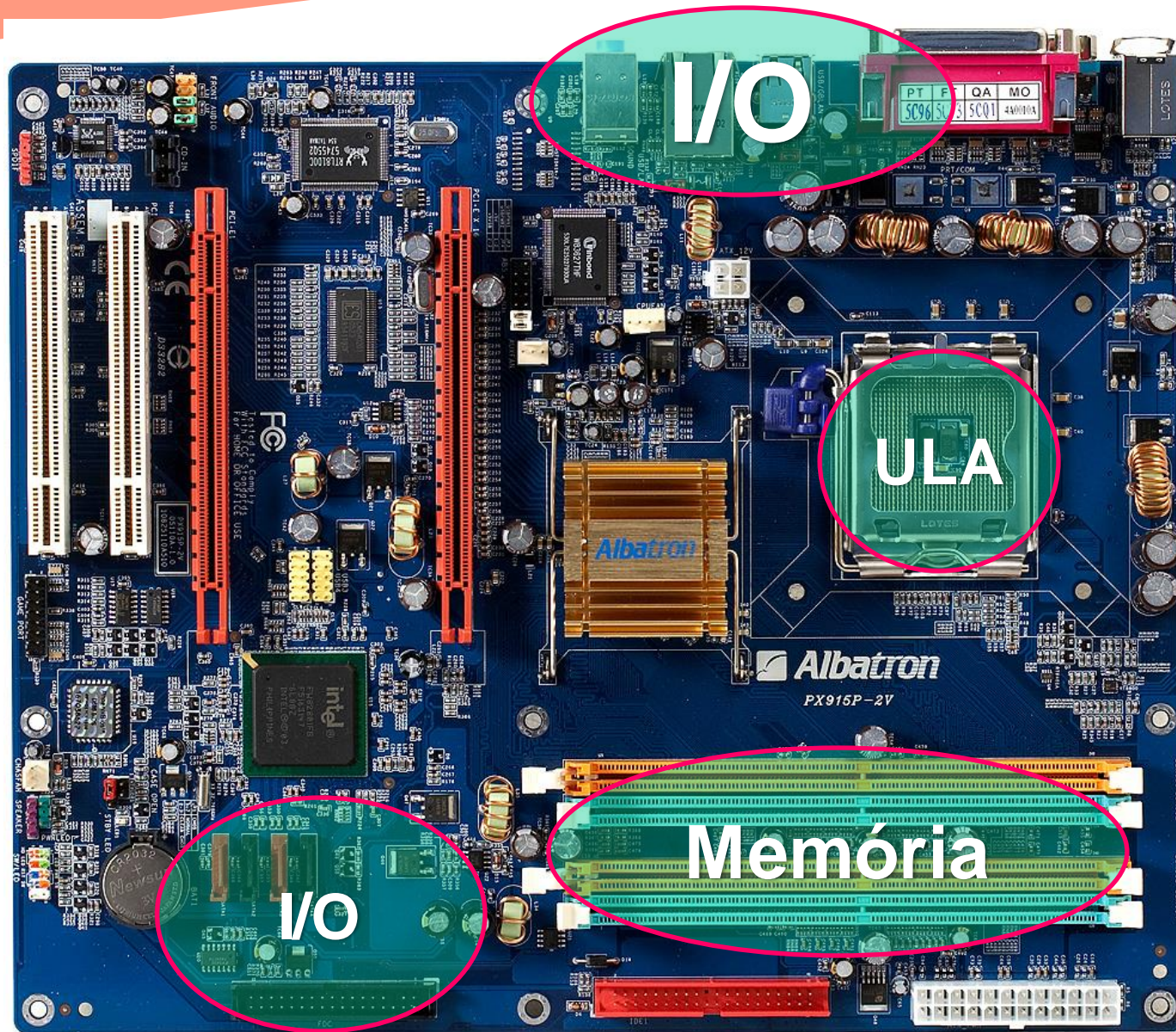
George Pólya on teaching John von Neumann
Fermat's Library @fermatslibrary

"There was a seminar for advanced students in Zürich that I was teaching and von Neumann was in the class. I came to a certain theorem, and I said it is not proved and it may be difficult. Von Neumann didn't say anything but after 5 minutes he raised his hand. When I called on him he went to the blackboard and proceeded to write down the proof. After that I was afraid of von Neumann"

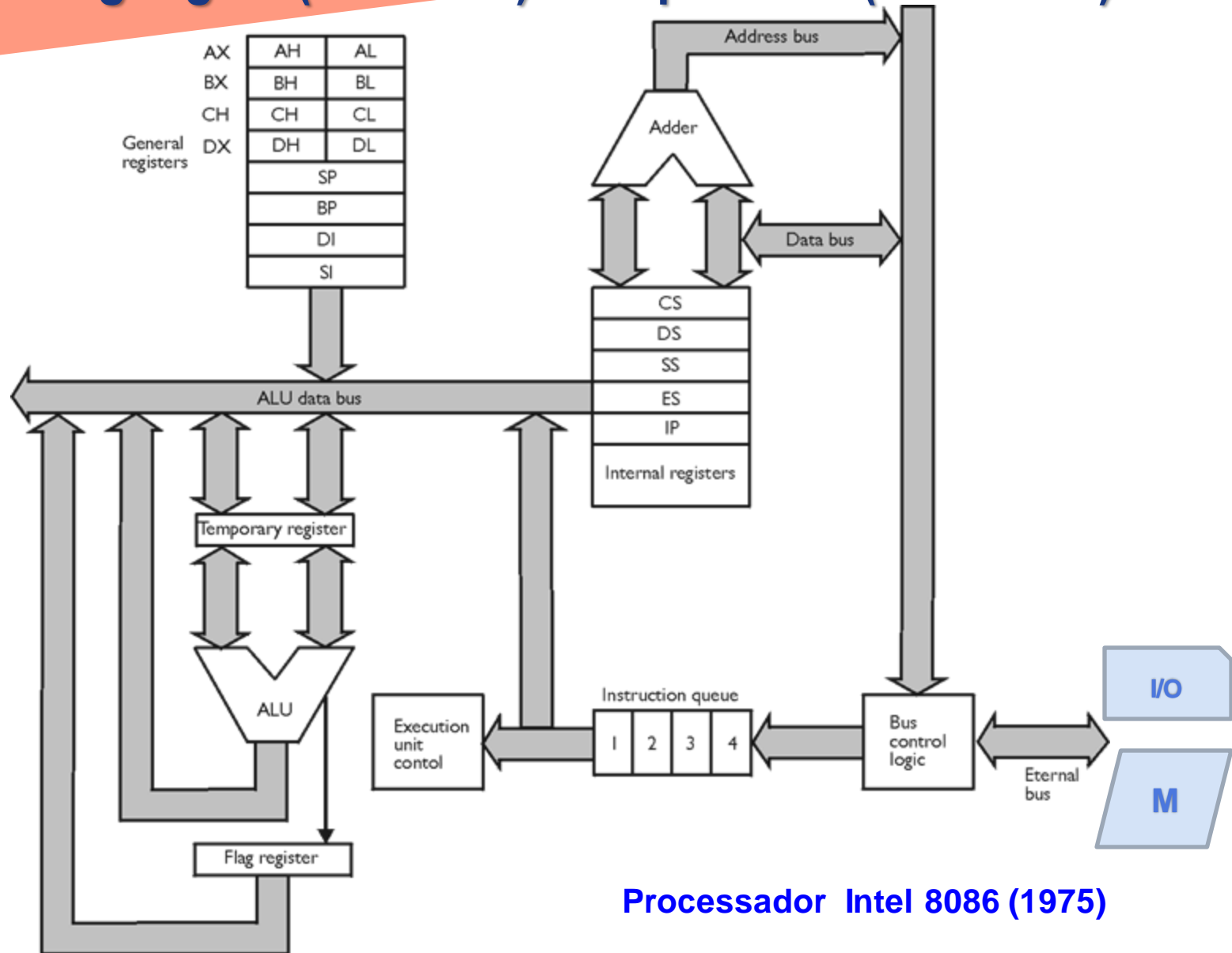


Linguagem (Software) e Arquitetura (Hardware)

Arquitetura von Neumann nos PCs modernos



Linguagem (Software) e Arquitetura (Hardware)



Linguagem Assembly

```
title Programa Hello World          (hello.asm)
; Este programa mostra "Hello, world!"
.model small
.stack 100h
.data
message db "Hello, world!",0dh,0ah,'$'
.code
main proc
    mov ax,@data
    mov ds,ax
    mov ah,9
    mov dx,offset message
    int 21h

    mov ax,4C00h
    int 21h
main endp
end main
```

Linguagem de programação
PARA máquina

```
.MODEL SMALL ;modelo de memória
.STACK       ;espaço de memória para instruções do programa na pilha
.CODE        ;as linhas seguintes são instruções do programa
    mov ah,01h ;move o valor 01h para o registrador ah
    mov cx,07h ;move o valor 07h para o registrador cx
    int 10h    ;interrupção 10h
    mov ah,4ch ;move o valor 4ch para o registrador ah
    int 21h    ;interrupção 21h
.DATA
x db 1
END           ;finaliza o código do programa
```

Linguagem vs. Máquina

DE

PARA

NA



0110101
1101101
0001001

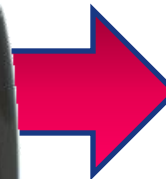


Assembly

Fortran
Delphi
C, C++
Scilab

Máquina (computador)

Linguagem PARA Máquina



Máquina (computador)

Linguagem Baixo-Alto Nível

**Alto
Nível**

Difícil para a máquina
Fácil para o programador



**Baixo
Nível**



Fácil para a máquina
Difícil para o programador

Linguagem Baixo-Alto Nível

```
real, dimension (10) :: a=(/(j,j=1,10)/), b, c
data b/3*1.,4*2.,3*3./
c(2:9)=a(2:9)+b(2:9)
cs=s.(c(2:9))
```

```
.MODEL SMALL
.STACK
.CODE
    move ah,01h
    move cx,07h
    int 10h
    move ah, 4ch
    int 21h
.DATA
x db 1
END
```

Linguagem Alto Nível

- ❖ **Leitura de notações familiares**
 - READ, WRITE, OPEN
- ❖ **Independência de máquina**
 - Programa = texto ASCII
- ❖ **Disponibilidade de bibliotecas de programas**
 - Bibliotecas, toolbox, DLL, etc.
- ❖ **Verificação de consistência**
 - Verificação de tipos de dados
- ❖ **Exemplos de linguagens de alto nível:**
 - Assembly, Fortran, Pascal, Delphi, C, Java,

4GL – Especificação Formal

ARRAY (Elem: [Undefined → Elem])

sort Array
imports INTEGER

Arrays are collections of elements of generic type Elem. They have a lower and upper bound (discovered by the operations First and Last). Individual elements are accessed via their numeric index.

Create takes the array bounds as parameters and creates the array, initializing its values to Undefined. Assign creates a new array which is the same as its input with the specified element assigned the given value. Eval reveals the value of a specified element. If an attempt is made to access a value outside the bounds of the array, the value is undefined.

Create (Integer, Integer) → Array
Assign (Array, Integer, Elem) → Array
First (Array) → Integer
Last (Array) → Integer
Eval (Array, Integer) → Elem

First (Create (x, y)) = x
First (Assign (a, n, v)) = First (a)
Last (Create (x, y)) = y
Last (Assign (a, n, v)) = Last (a)
Eval (Create (x, y), n) = Undefined
Eval (Assign (a, n, v), m) =
 if m < First (a) **or** m > Last (a) **then** Undefined **else**
 if m = n **then** v **else** Eval (a, m)

Pontos de Vista

❖ O Projetista

- Os inventores das linguagens de programação

❖ O Implementador

- O compilador ou interpretador para uma linguagem

❖ O Usuário

- Quem escreve os programas na linguagem

O sucesso de uma linguagem de programação depende do sucesso destes três pontos de vista

Programação: 5 Aspectos Básicos

- ❖ **O Processo da Programação (ciclo de vida)**
 - Etapas, atividades, arquivos
- ❖ **Ferramentas da Programação**
 - Linguagem, Compilador
 - Manual, Guia de Referência
 - Editores, compiladores, interpretadores, ligadores, bibliotecas
- ❖ **Ambientes de Programação**
 - Shells, DOS-Windows-UNIX
- ❖ **Elementos da Programação**
 - Sintaxe-semântica da linguagem
 - Comandos, sub-programas, bibliotecas
- ❖ **Paradigmas**
 - Imperativo, Funcional, Lógico, Objetos, Concorrente, Marcas, híbridos

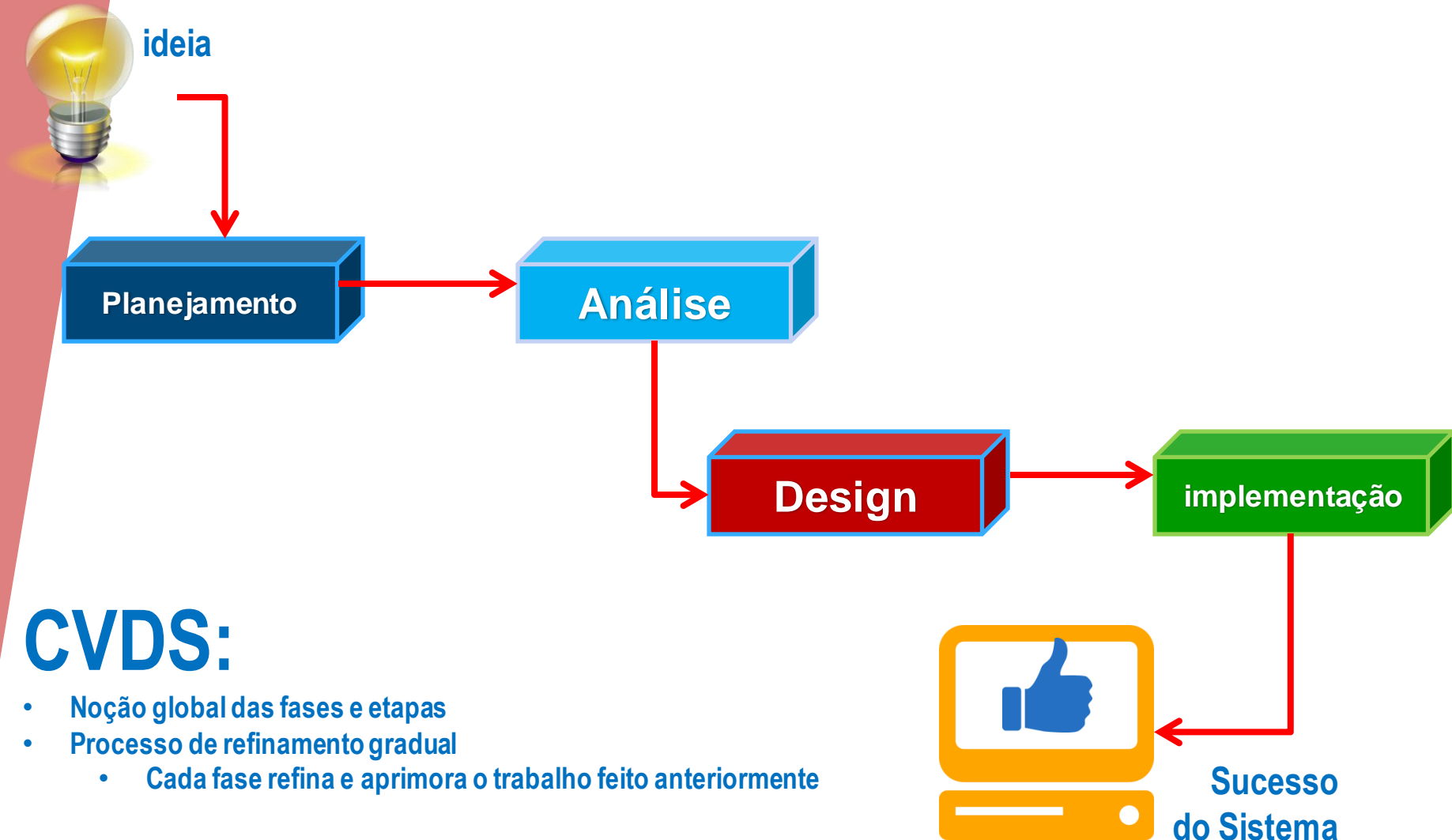
Ciclo de Vida da Programação

Problema: Somar 2 números

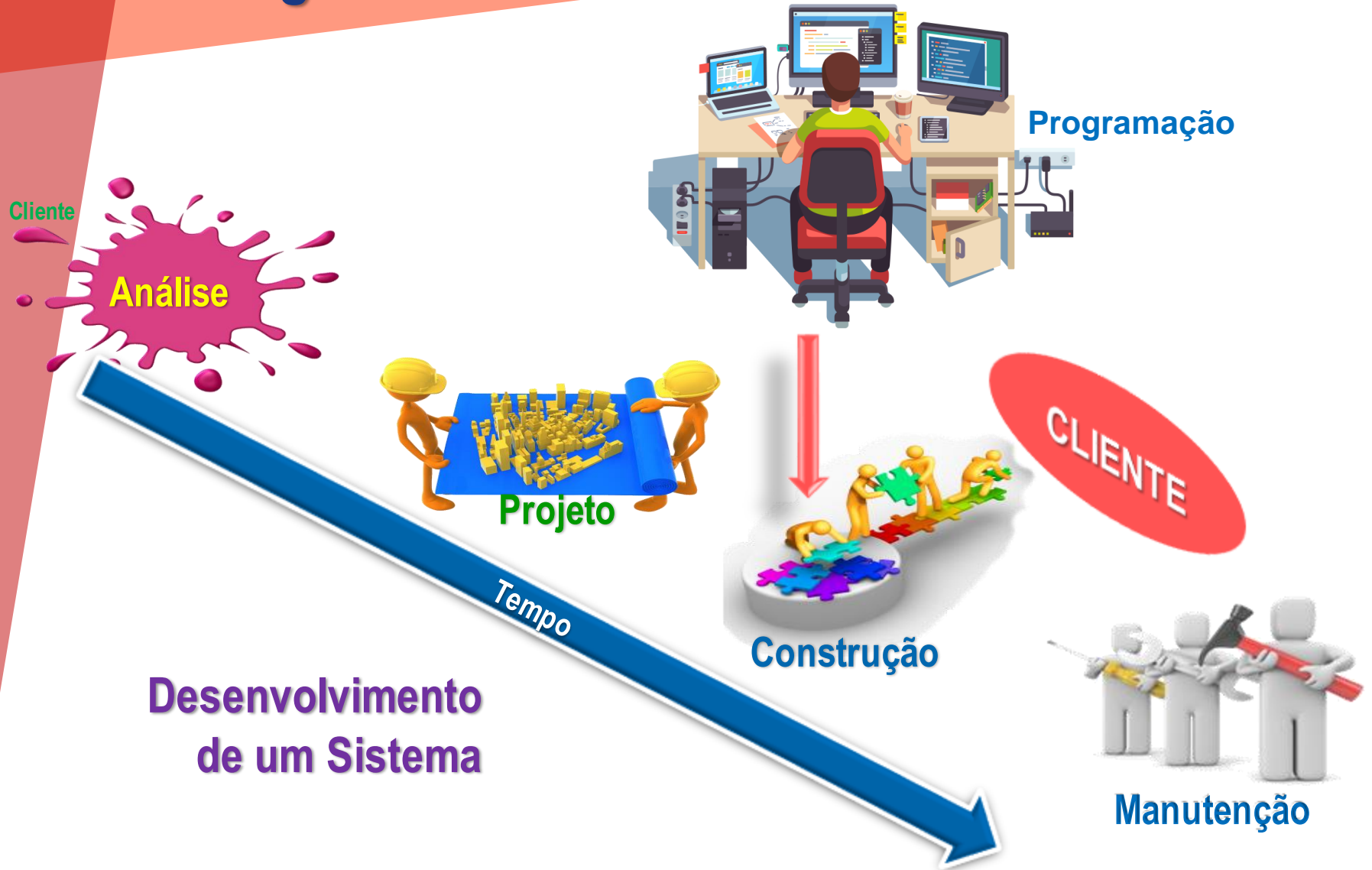


Solução: Programa

CVDS - Ciclo de Vida de Desenvolvimento de Sistemas



Paradigma Genérico de Desenvolvimento: FASES



Ciclo de Vida da Programação

1. Fase de Resolução de problemas

- Análise (problema, domínio) e Especificação (QUE)
- Solução Geral (Algoritmo)
- Verificação (solução vs. problema)

2. Fase de Implementação

- Solução concreta (algoritmo em Programa: codificação)
- Teste (executar no computador: resultados corretos e erros)

3. Fase de manutenção

- Uso
- Manutenção (modificação, corrigir erros)

O Processo da Programação

❖ Programa (Código) Fonte

- Conjunto de instruções em uma determinada linguagem de Programação: FORTRAN, C, Pascal, MATLAB, Delphi
 - atribuição `x:= 23;`
 - Laços FOR, WHILE, REPEAT
 - Funções e procedimentos
 - Mecanismos de entradas e saídas de dados
- Fácil (de entender) para o homem, difícil para o computador
- Possui uma sintaxe e uma semântica.
- Antes de ser processado e executado, deve ser traduzido para a linguagem do S.O., utilizando um COMPILADOR.
- Tipo de arquivo: *.c, *.pas, *.for, *.m, *.cpp (ASCII, texto)

O Processo da Programação

❖ Programa (Código) Objeto

- É o resultado do processo de compilação de um programa fonte.
- O código usado é formado por símbolos entendidos unicamente pelo compilador. %\$785s#axc\$b°æêα
- Tempo de Compilação: é tempo utilizado para traduzir o programa fonte em programa objeto.
- Tipo de arquivo: *.OBJ

❖ Programa (Código) Executável

- É o programa que ao ser executado, produz os resultados esperados, como solução ao problema original.
- Tipo de arquivo: *.EXE (*.com *.bat)

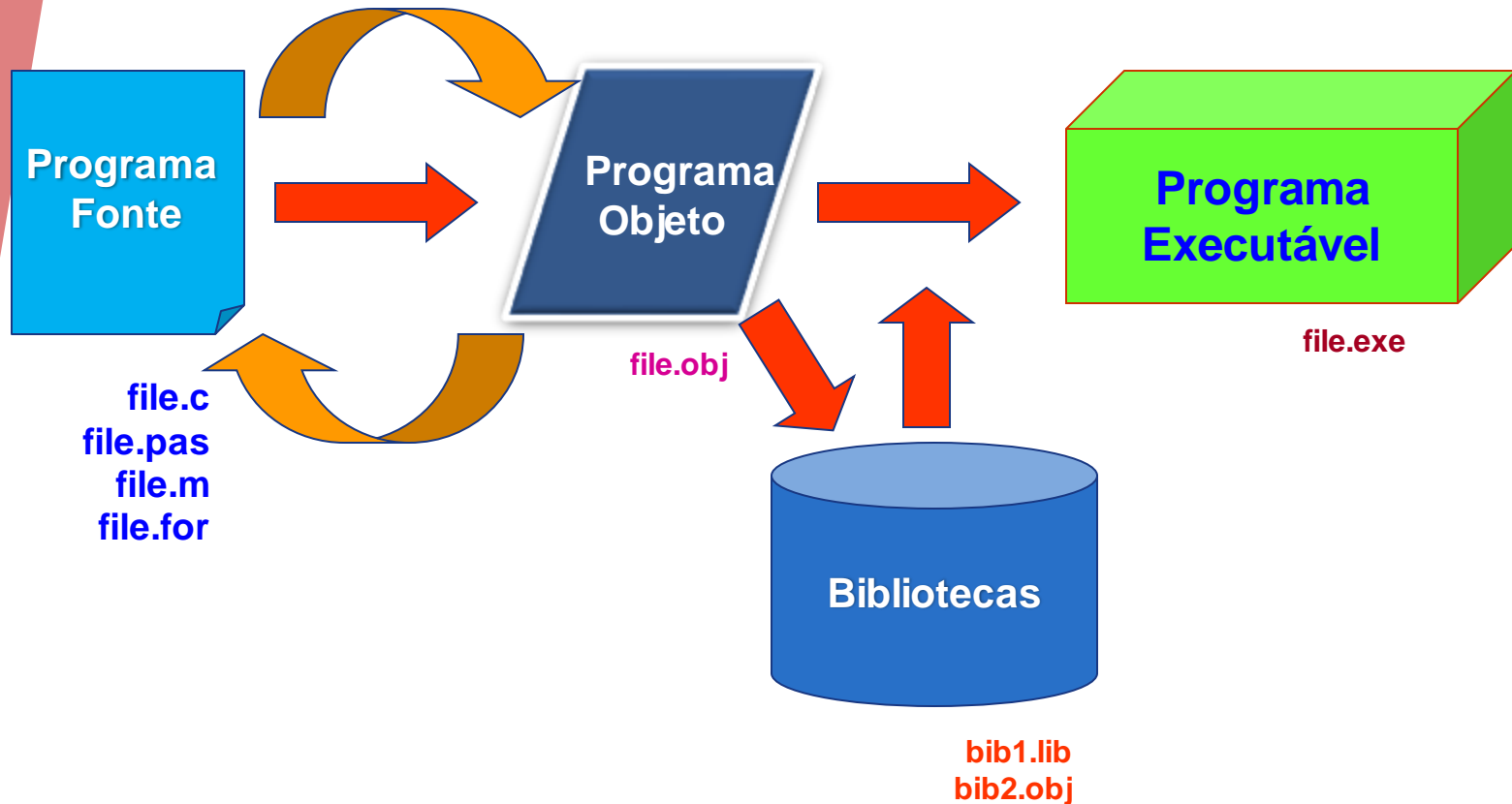
O Processo da Programação

❖ Tipos

- Por compilação
 - Compilador - Linker
 - Códigos: Fonte, objeto, executável, resultados
- Por interpretação
 - Interpretador
 - Códigos: Fonte, resultados
- .NET
 - Microsoft (Plataforma .NET)
 - Códigos: Fonte, MSIL, Nativo, resultados

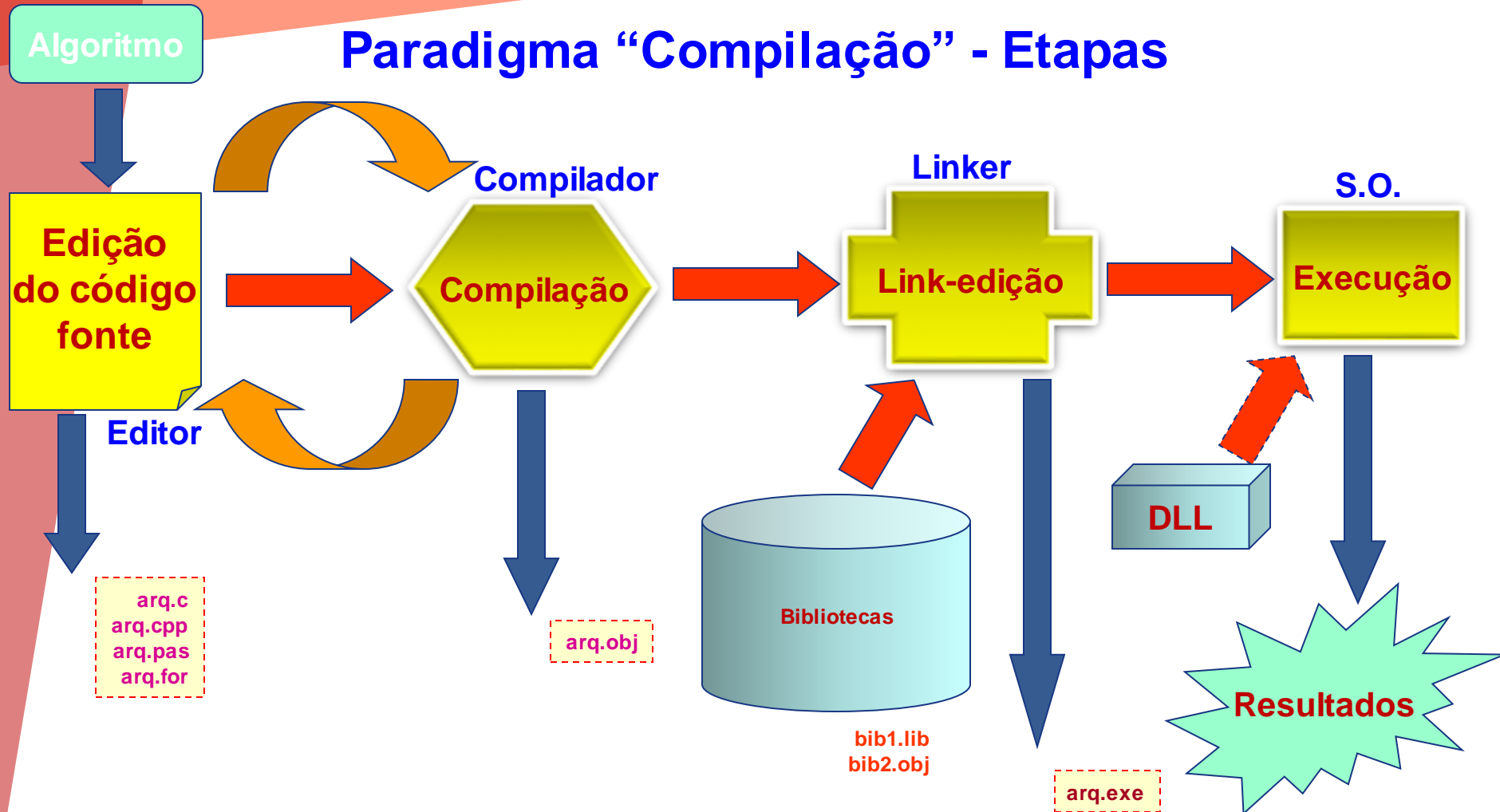
O Processo da Programação

Paradigma “Compilação” - objetos



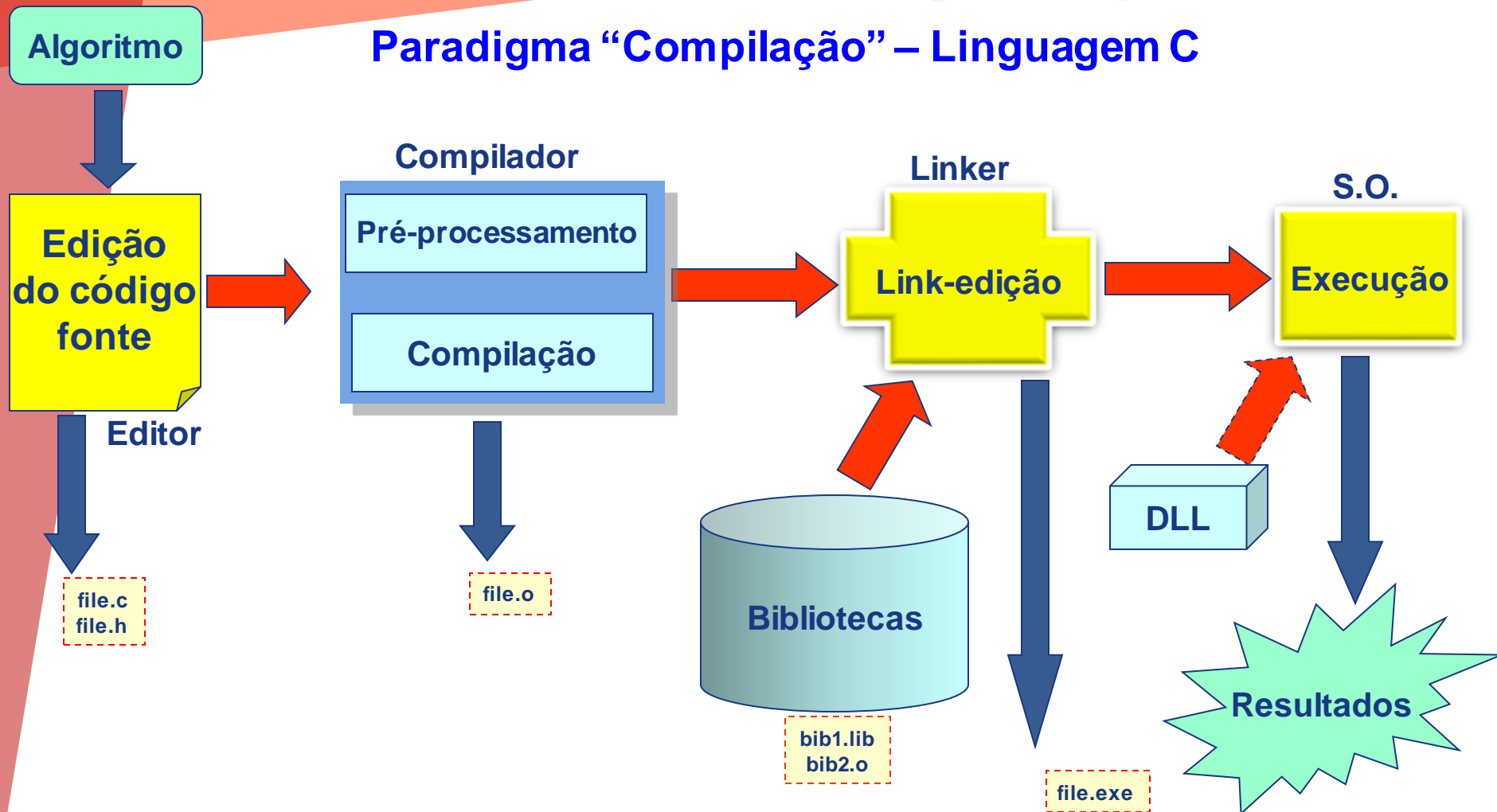
O Processo da Programação

Paradigma “Compilação” - Etapas



O Processo da Programação - C

Paradigma “Compilação” – Linguagem C



O Processo da Programação

Paradigma “Interpretação”



- Interpretação da instrução (texto) “linha por linha”
- Interpretação seqüencial
- Não é necessário compilador – Não existe código objeto
- Não existe programa executável
- Interpretadores: MATLAB, Haskell, Lisp, Prolog, Python

O Processo da Programação

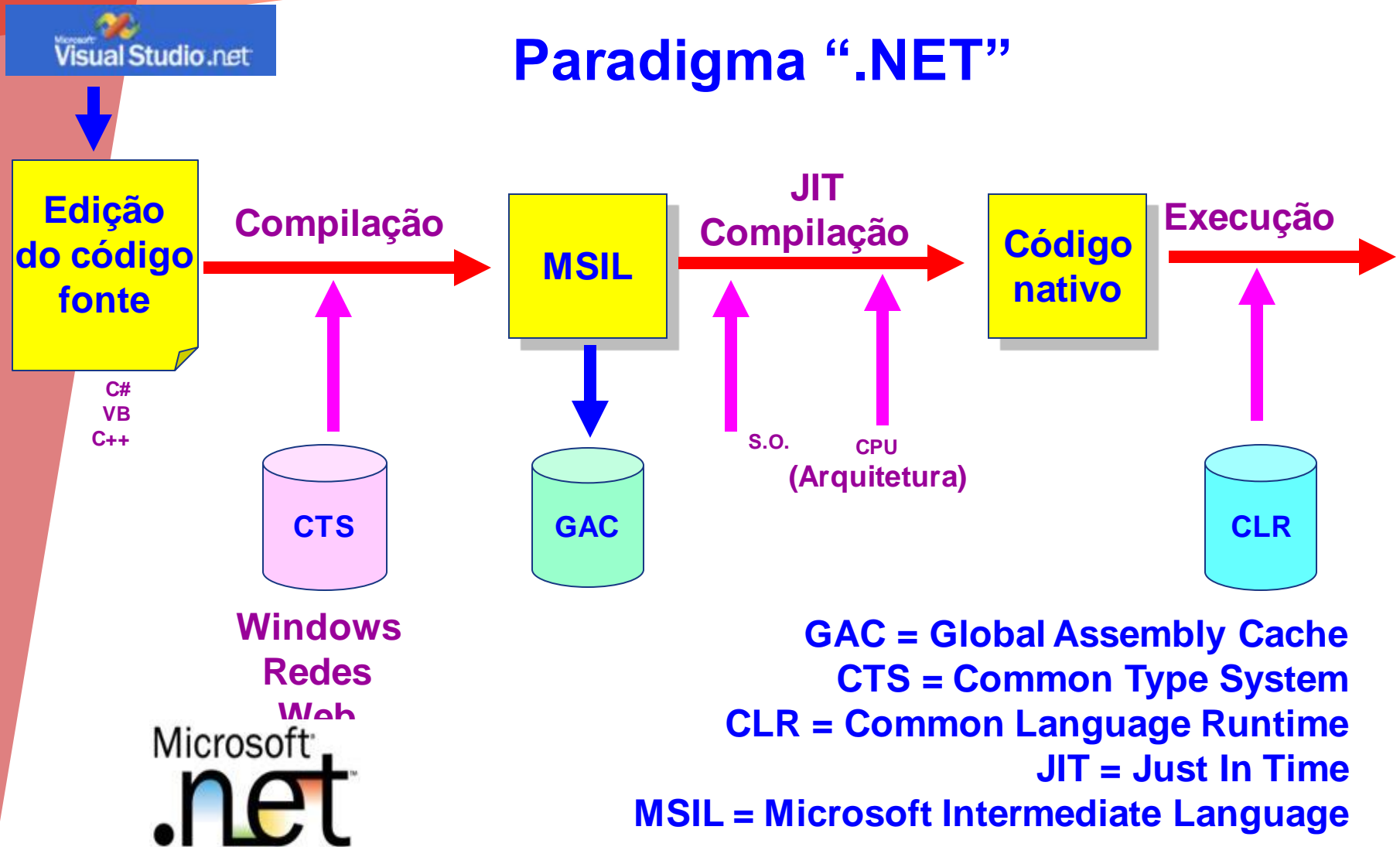
Paradigma “.NET”

- ❖ **Proposta pela Microsoft Corporation**
- ❖ **Conecta**
 - Informação + Pessoas + Sistemas + dispositivos
- ❖ **.NET Framework**
 - Componente do windows integral para construir e executar aplicações e serviços web.
 - Suporte para mais de 20 linguagens de programação diferentes
- ❖ **Visual Studio .NET**
 - Ambiente de Desenvolvimento Integrado



O Processo da Programação

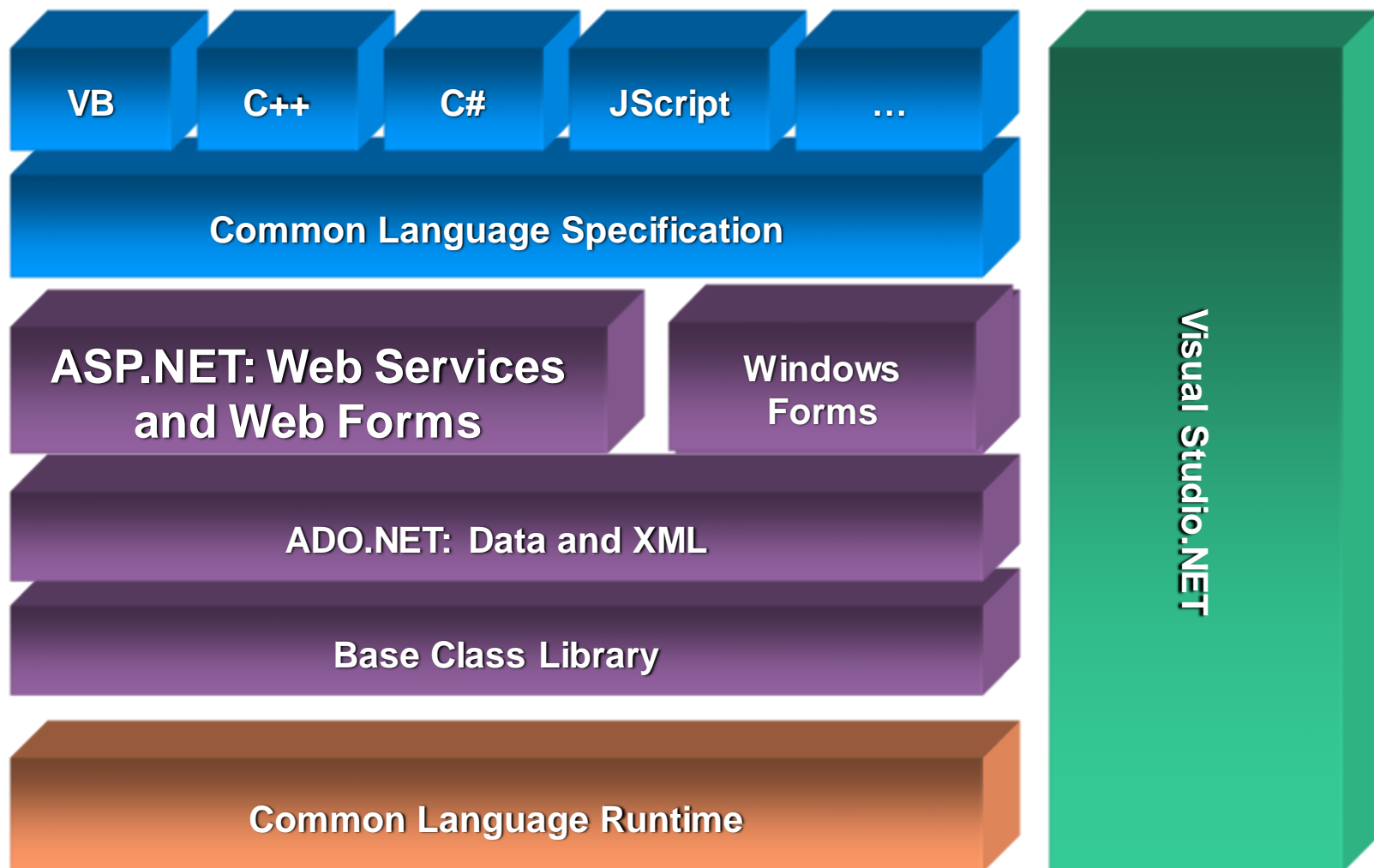
Paradigma “.NET”



O Processo da Programação



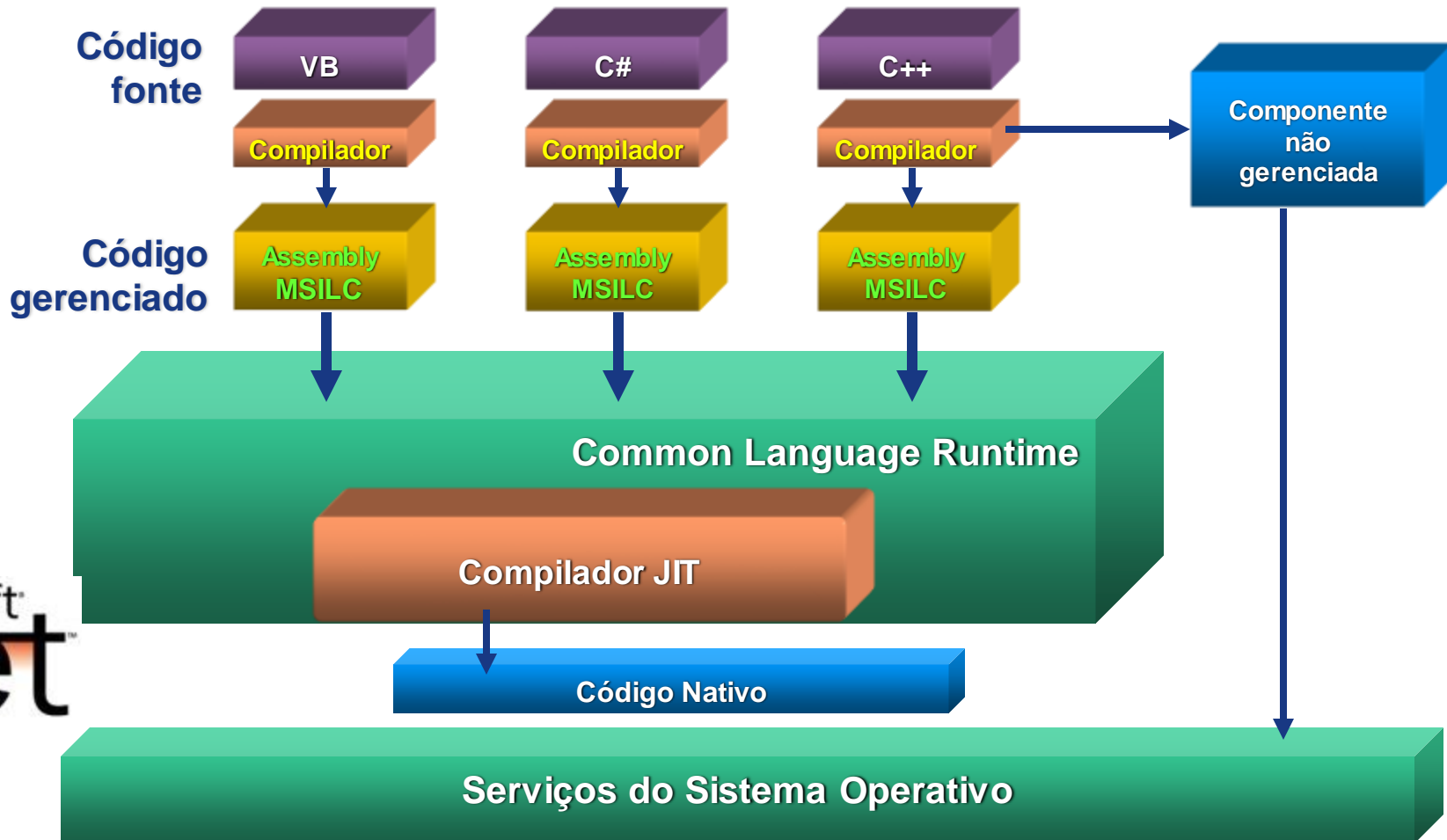
.NET Framework



O Processo da Programação

Paradigma “.NET”

CLR: Modelo de Execução

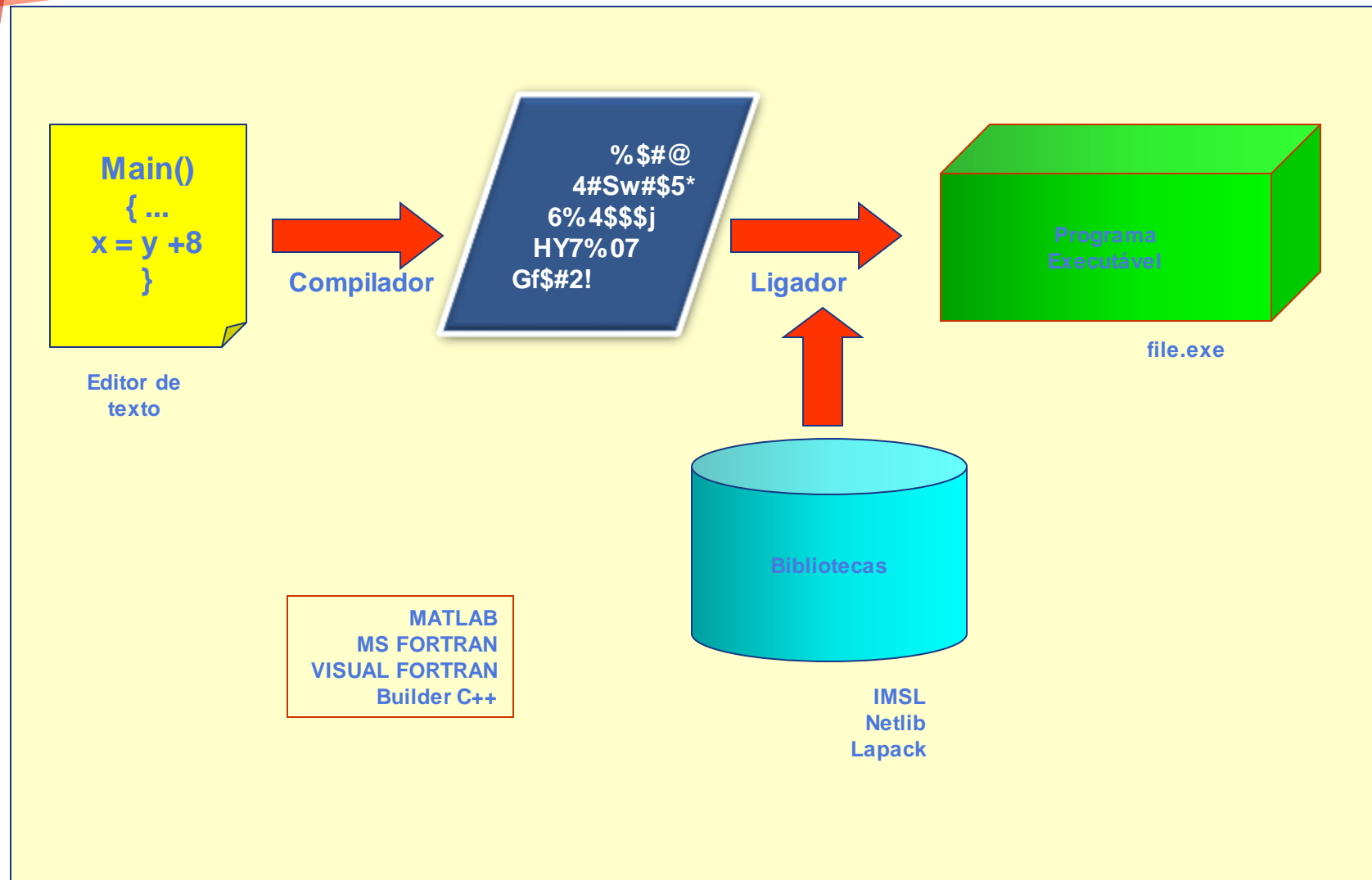


Microsoft
.net

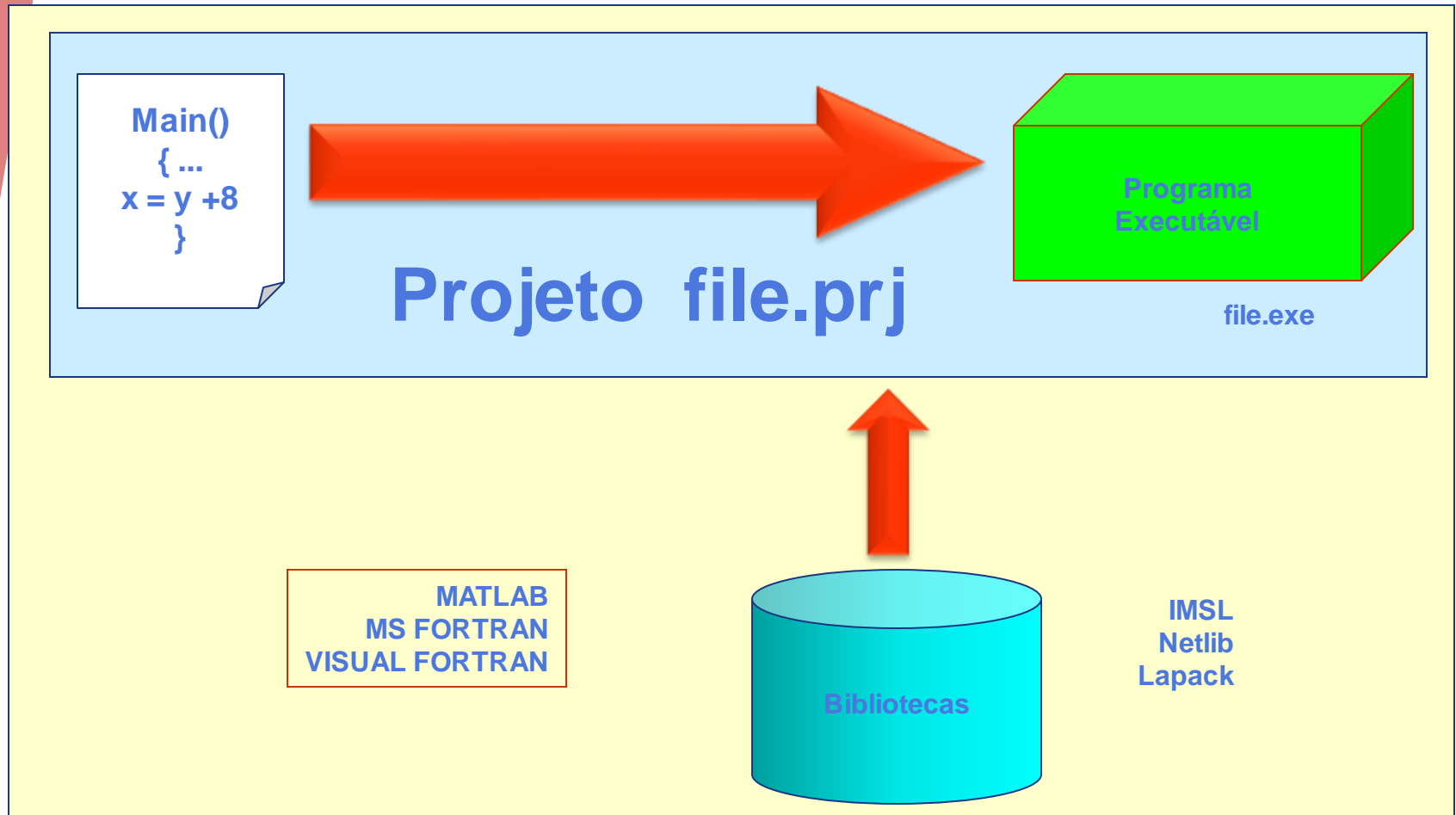
Ambientes de Programação

- ❖ **É o conjunto de elementos e ferramentas integradas, necessários para construir um programa executável, que resolva um problema dado.**
- ❖ **Elementos:**
 - Editor de programas fonte
 - Compilador
 - Ligador (linker)
 - Depurador
- ❖ **Ambientes**
 - UNIX: o mais antigo
 - Smalltalk e MATLAB: linguagem + ambiente
 - Borland C++Builder, Delphi, Microsoft Visual C++, Compaq Digital FORTRAN, Java Development Kit, Visual Studio.net

Ambientes de Programação

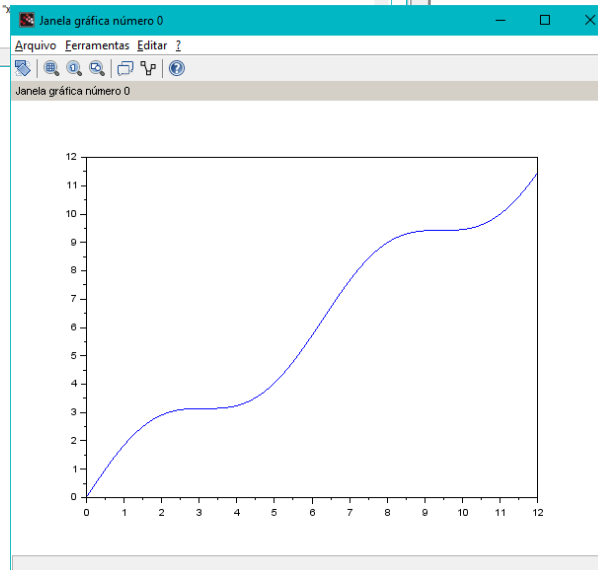
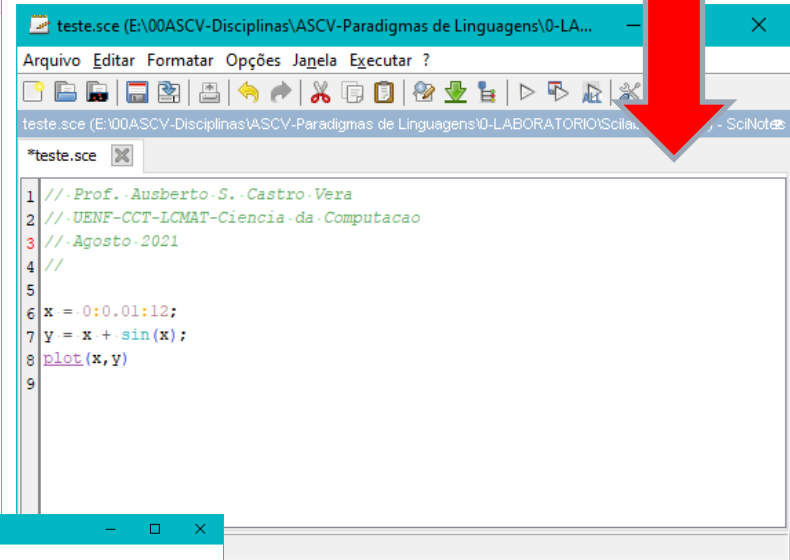


Ambientes de Programação

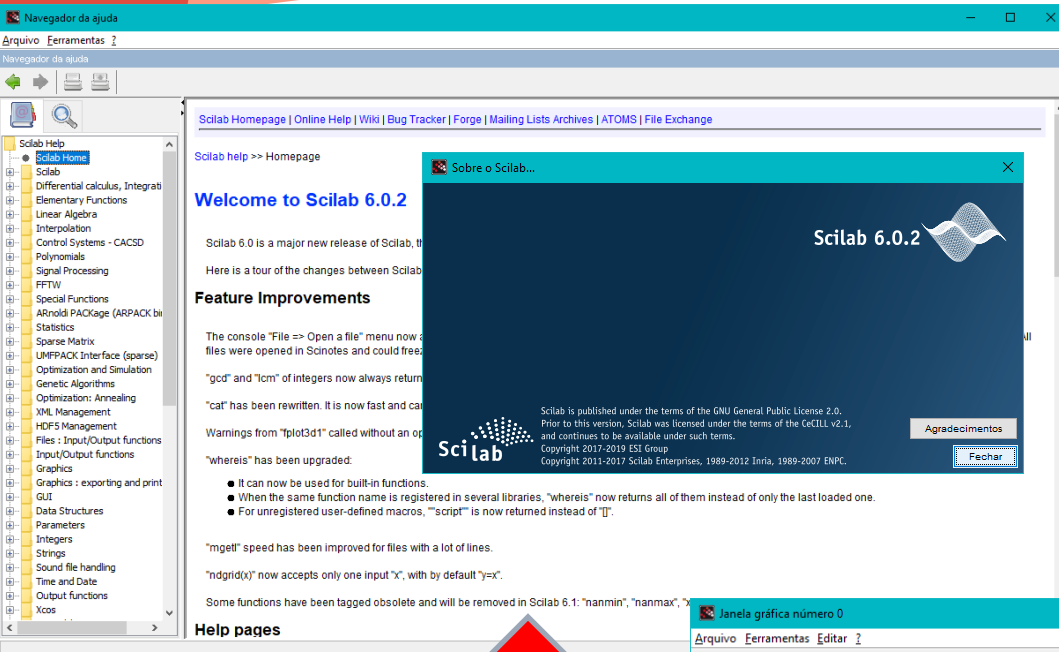


Ambiente Scilab 6.0.2

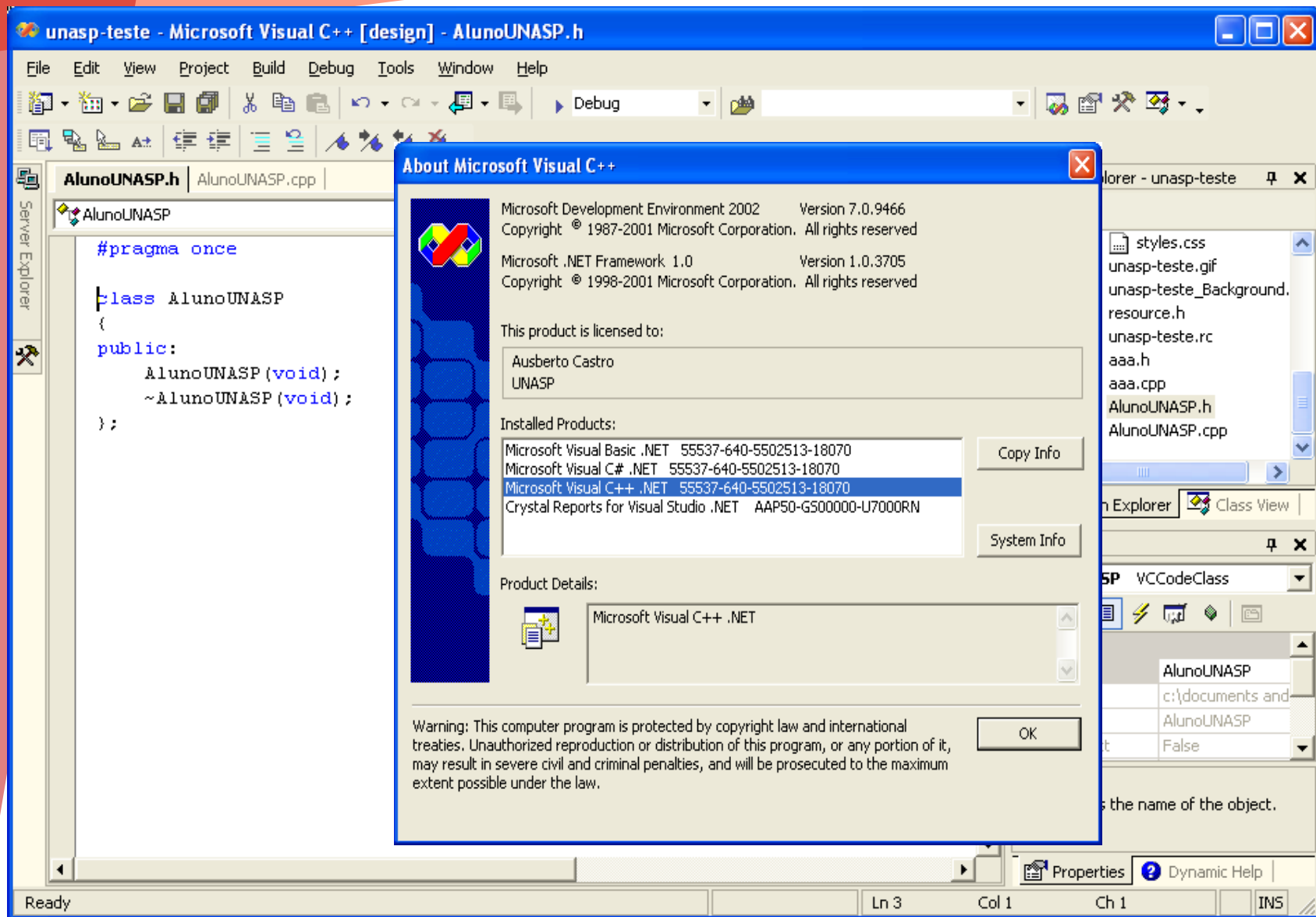
Editor SciNotes



Janela de Comandos Console



MS Visual Studio .NET



Visual Studio Express 2012

The screenshot displays the Visual Studio Express 2012 for Windows 8 interface. The main window shows a C++ project named "WindowsRuntimeComponent2" with a file "Class1.h" open. The code in "Class1.h" is as follows:

```
#pragma once

namespace WindowsRuntimeComponent2
{
    public ref class Class1 sealed
    {
    public:
        Class1();
    };
}
```

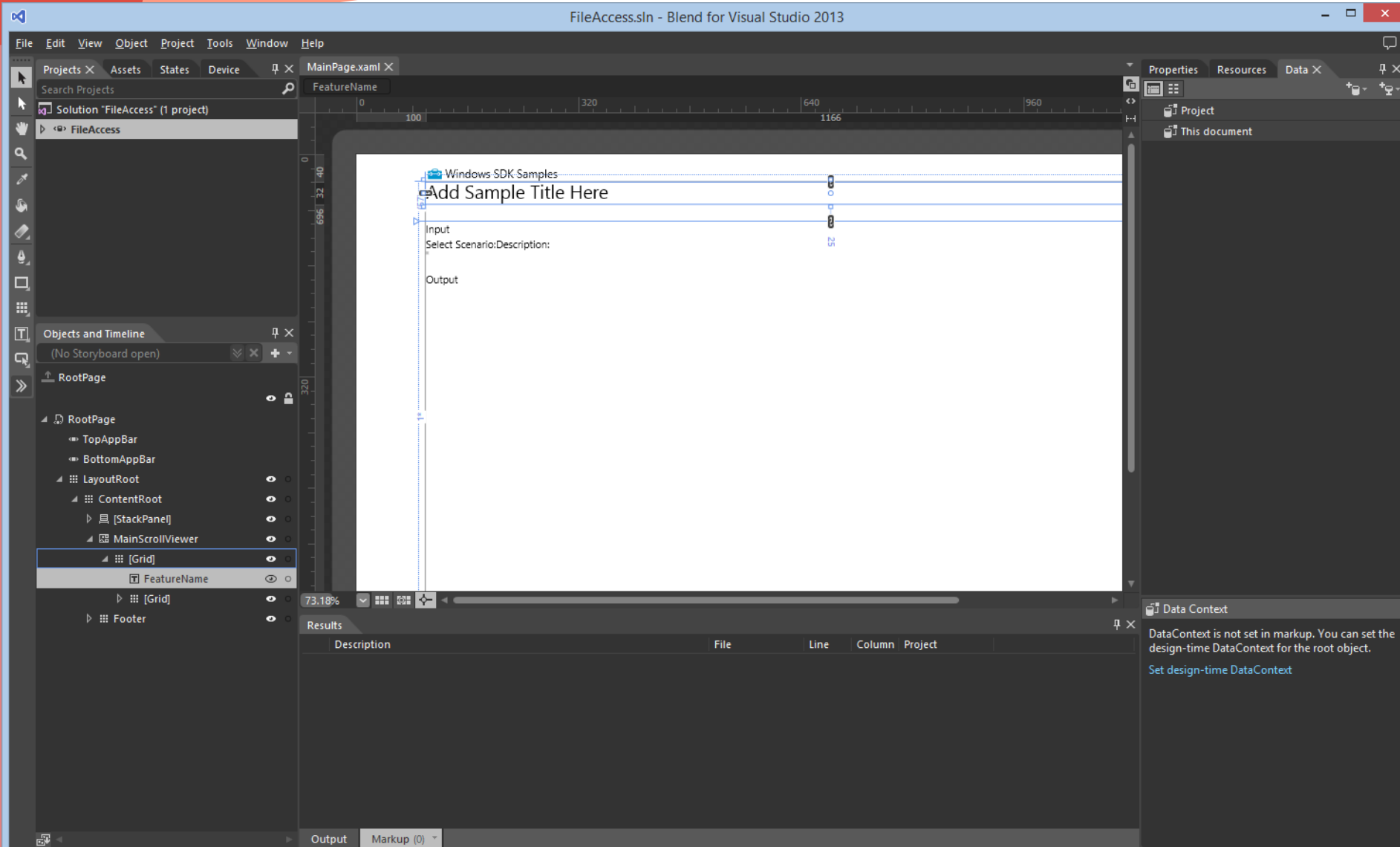
An "About Microsoft Visual Studio Express 2012 for Windows 8" dialog box is open in the foreground. It provides the following information:

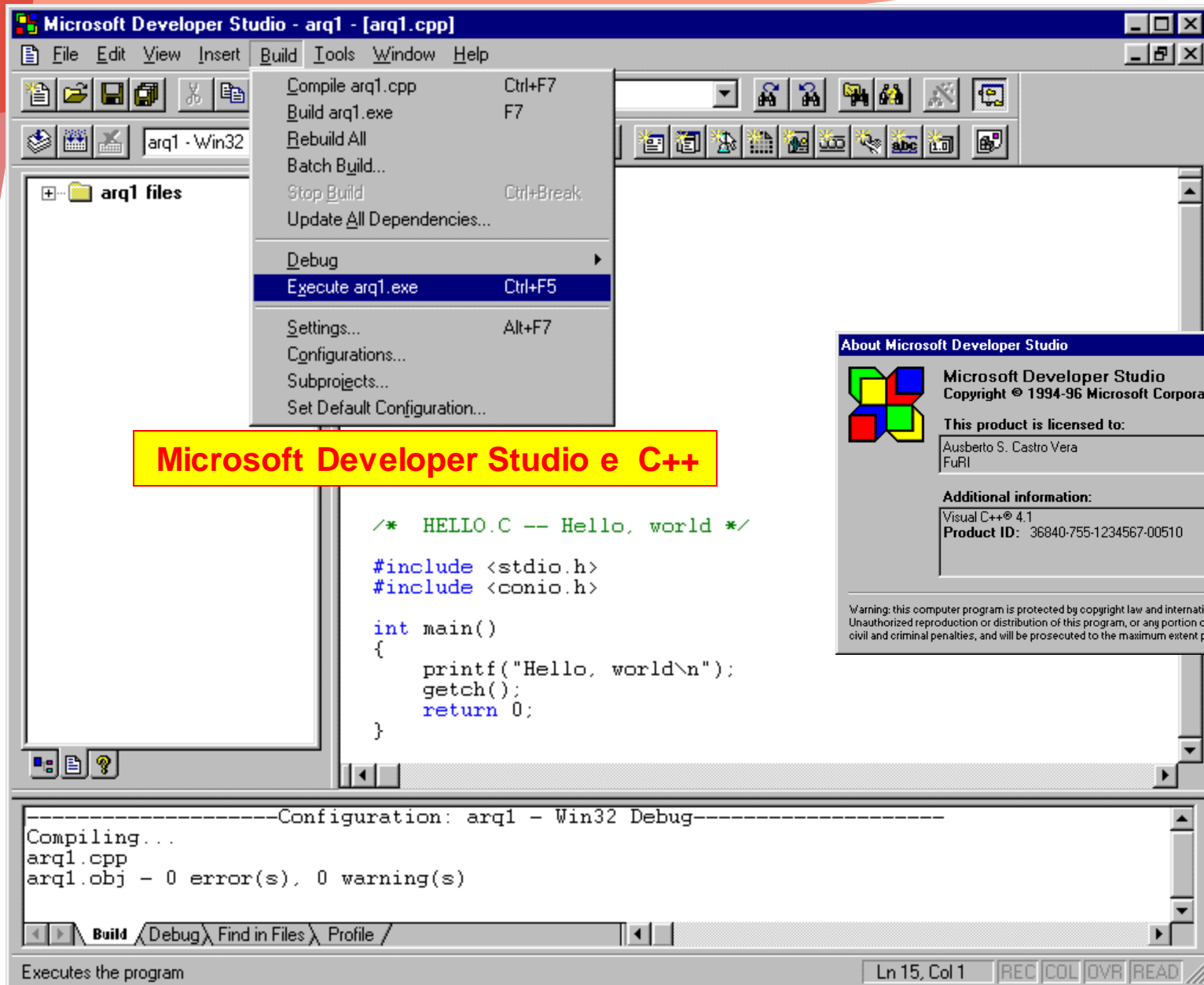
- Visual Studio** logo and "Licensed to: User"
- Microsoft Visual Studio Express 2012 for Windows 8, Version 11.0.50727.1 RTMREL, © 2012 Microsoft Corporation. All rights reserved.
- Microsoft .NET Framework, Version 4.5.50709, © 2012 Microsoft Corporation. All rights reserved.
- Installed products:**
 - Team Explorer for Visual Studio 2012 04933-004-0021004-02851
 - Visual Basic 2012 04933-004-0021004-02851
 - Visual C# 2012 04933-004-0021004-02851
 - Visual C++ 2012 04933-004-0021004-02851
 - Visual Studio 2012 Code Analysis Spell Checker 04933-004-0021004-02851
 - NuGet Package Manager 2.0.30625.9003
- Product details:** Microsoft Team Explorer for Visual Studio 2012
- Warning:** This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

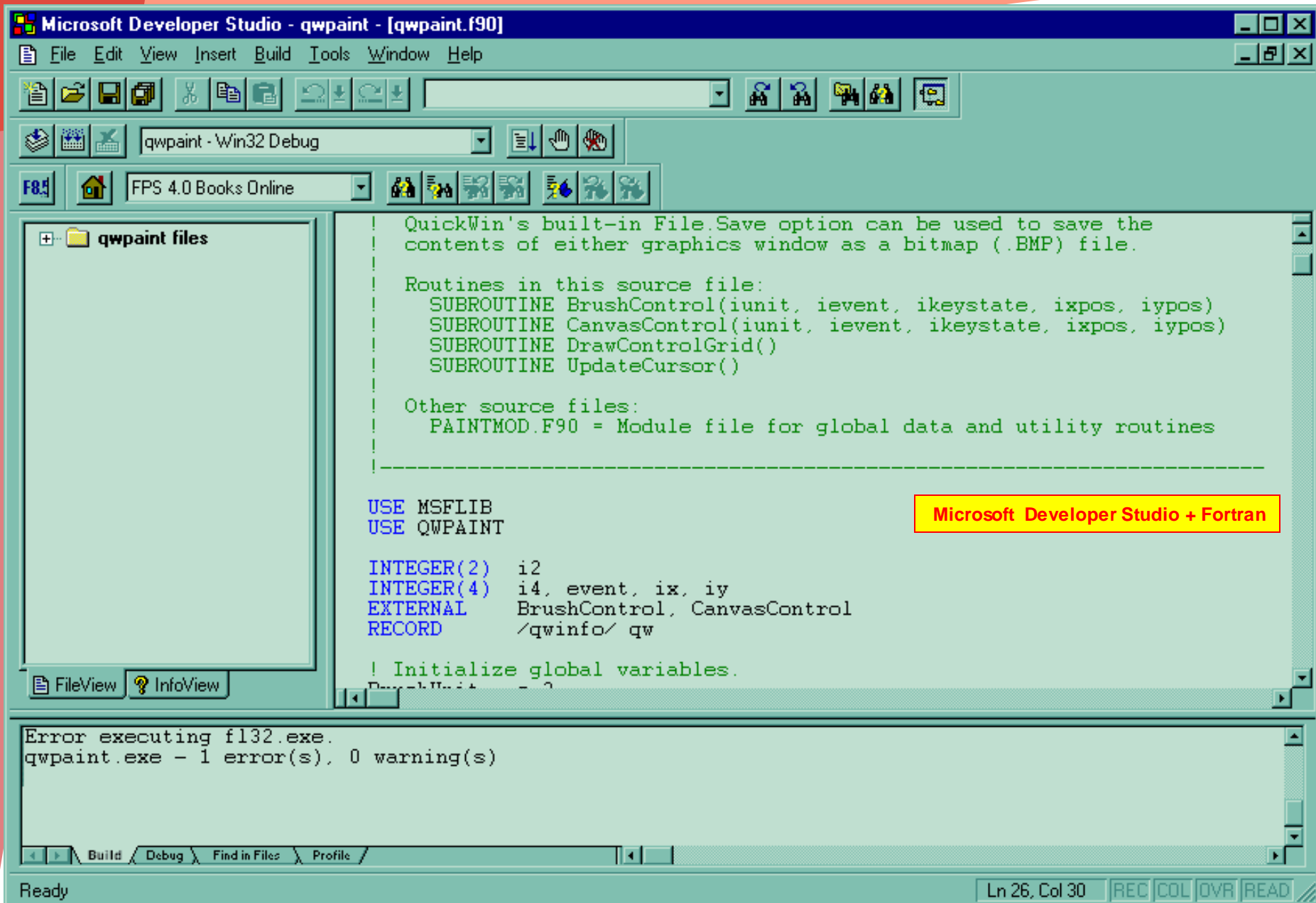
The background shows the Solution Explorer with the project "WindowsRuntimeComponent2" containing files: External Dependencies, Resources, Class1.cpp, Class1.h, pch.cpp, and pch.h. The Properties window shows the "Class1 VCCodeClass" with the following properties:

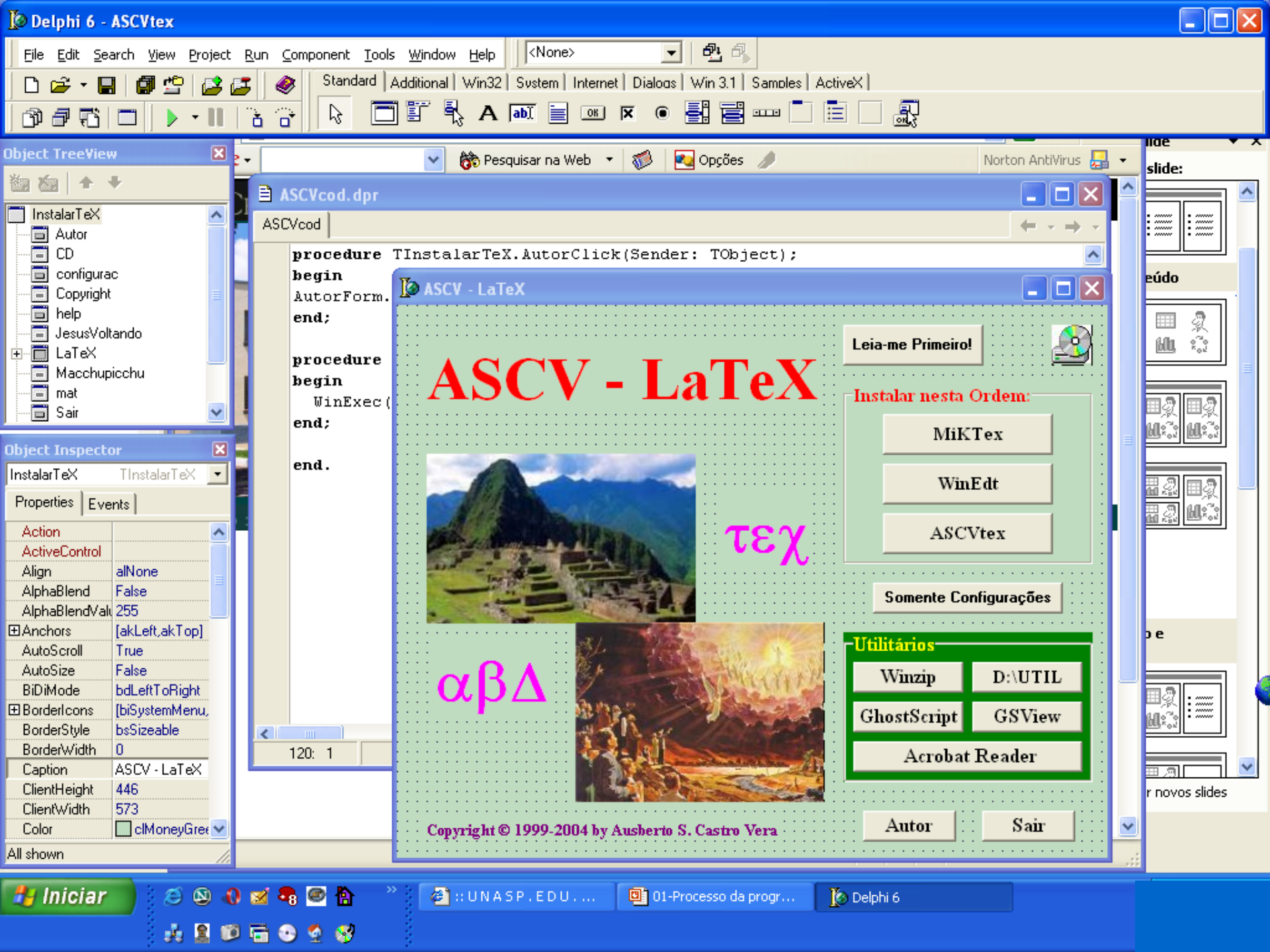
Property	Value
(Name)	Class1
Access	public
File	c:\users\user\documents\visual
FullName	WindowsRuntimeComponent2
IsAbstract	False
IsInjected	False
IsManaged	True
IsSealed	True
IsTemplate	False
IsValue	False

Visual Studio 2013

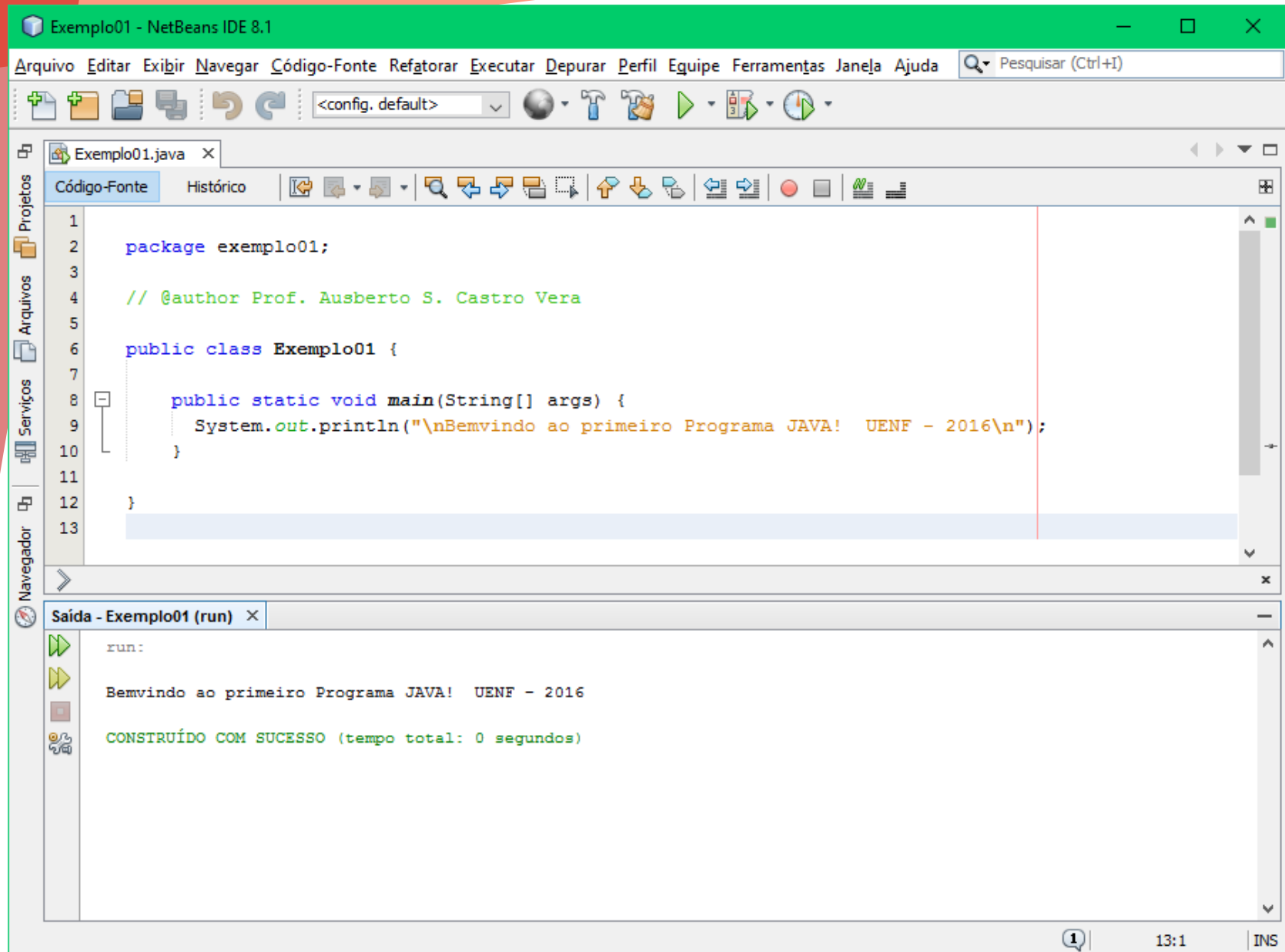








Oracle NetBeans IDE 8.1



Apache NetBeans 12.4

The screenshot displays the Apache NetBeans IDE 12.4 interface. The main window shows a Java project named '01FirstJava'. The 'Projects' pane on the left lists the project structure, including 'Source Packages', 'Dependencies', 'Java Dependencies', and 'Project Files'. The 'main - Navigator' pane shows the 'A' package with a 'main(String[] args)' method. The 'Source' editor displays the following code:

```
1  ...5 lines
6
7  /**
8   *
9   * @author Prof. Ausberto S. Castro Vera (ascv@uenf.br)
10  */
11  public class A {
12      public static void main(String args[]){
13          System.out.println("\nOi, Brasil!\n");
14          System.out.println("Bemvindo ao primeiro Programa JAVA! UENF 2021\n");
15      }
16  }
17
```

The 'Output' pane at the bottom shows the execution results:

```
DoisVeiculos (run) x Run (01FirstJava) x
--- exec-maven-plugin:3.0.0:exec (default-cli) @ 01FirstJava ---
Oi, Brasil!
Bemvindo ao primeiro Programa JAVA! UENF 2021
BUILD SUCCESS
```

An 'About' dialog box is open in the foreground, displaying the Apache NetBeans logo and the following information:

NetBeans IDE and NetBeans Platform are based on Apache NetBeans from the Apache Software Foundation and are licensed under [Apache License Version 2.0](https://www.apache.org/licenses/LICENSE-2.0). For more information, please visit netbeans.apache.org.

Product Version: Apache NetBeans IDE 12.4
Java: 16.0.2; Java HotSpot(TM) 64-Bit Server VM 16.0.2+7-67
Runtime: Java(TM) SE Runtime Environment 16.0.2+7-67
System: Windows 10 version 10.0 running on amd64; Cp1252; pt_BR (nb)
User directory: C:\Users\ASCVI78G\AppData\Roaming\NetBeans\12.4
Cache directory: C:\Users\ASCVI78G\AppData\Local\NetBeans\Cache\12.4

The status bar at the bottom indicates the current file is 'INS/Windows (CRLF)' and the time is 14:26.



Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

ascv@uenf.br
ausberto.castro@gmail.com

