

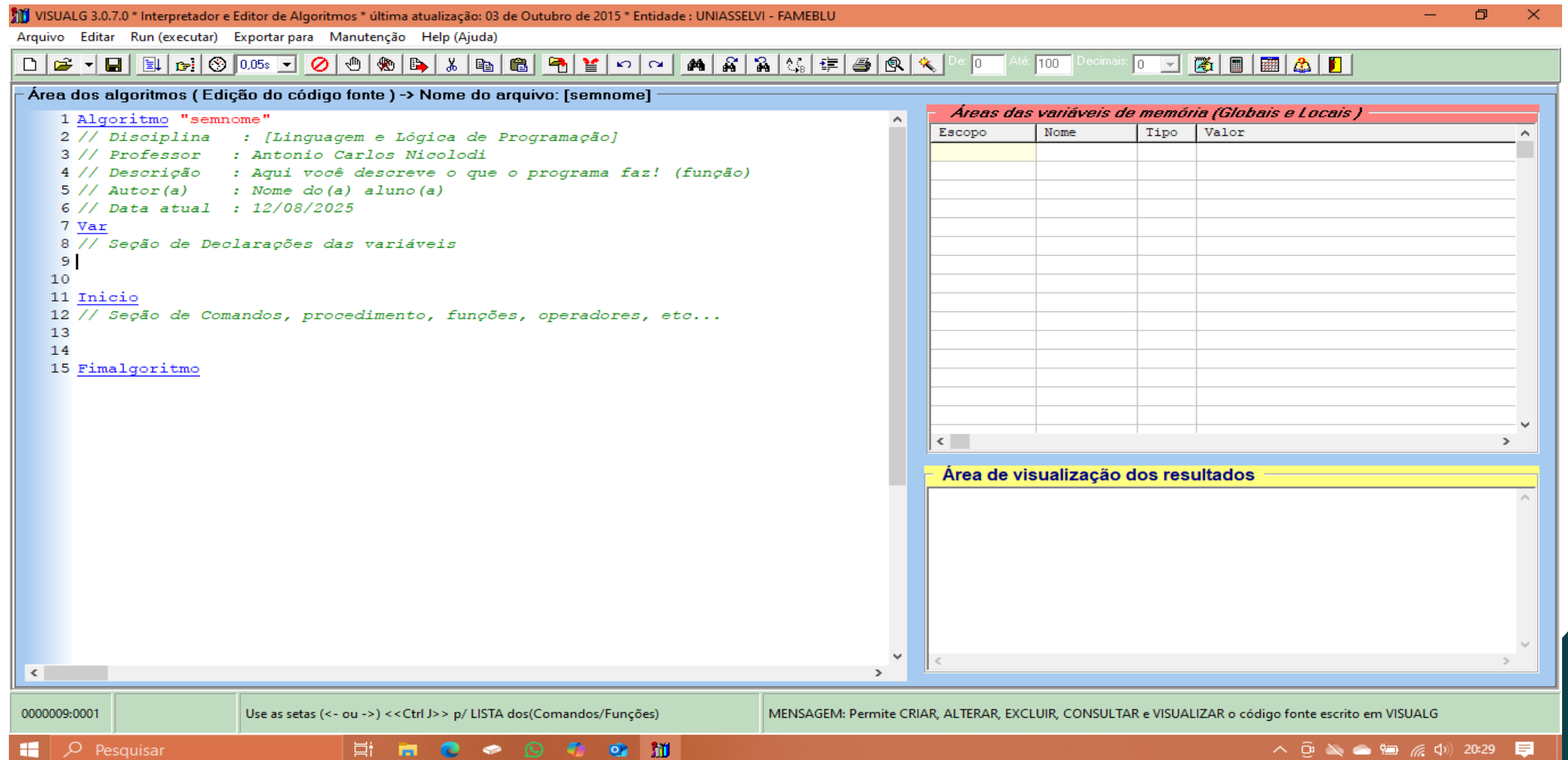
# Lógica de Programação com o Visualg e Portugol

Prof: Patrícia Lima Felismino  
Bacharel em Ciência da Computação - UNICSUL

# Visualg

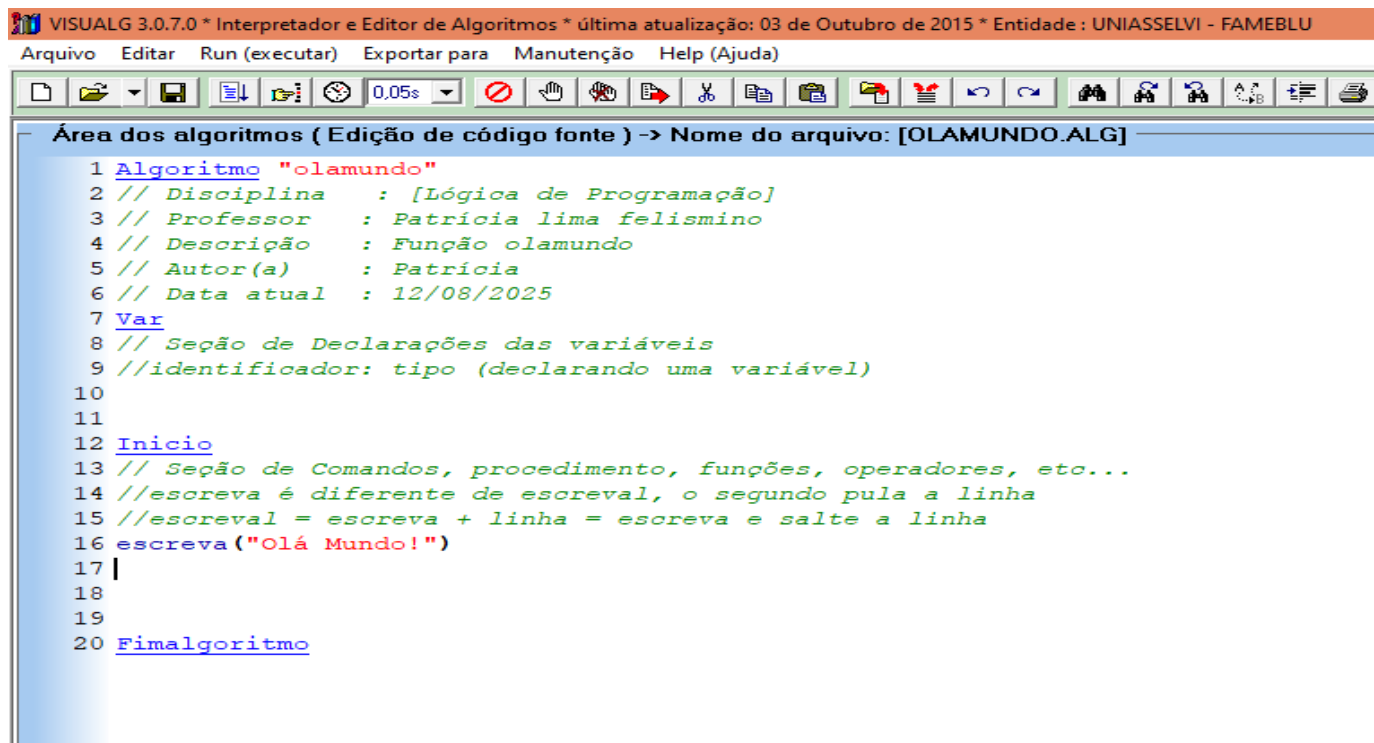
- **O VisuALG (Visualizador de algoritmo), criado pelo professor Cláudio Morgado de Souza**, é um programa que edita, interpreta e executa algoritmos escritos em pseudocódigo (Portugol), como um programa normal de computador. É um programa de livre uso e distribuição, empregado no ensino de programação em várias escolas e universidades no Brasil e no exterior.
- **Portugol** é uma linguagem de programação didática que utiliza uma sintaxe baseada na língua portuguesa. Seu principal objetivo é ensinar lógica de programação para iniciantes, eliminando a complexidade de linguagens como Python, Java ou C.
- **Obs!** Portugol é uma linguagem didática usada para auxiliar os estudantes de programação no entendimento da lógica de programação, facilitando a criação de algoritmos por se tratar de uma linguagem que utiliza do português.

# Conhecendo o ambiente no visualg



# Nosso primeiro programa no visualg

- Vamos criar um script simples que imprime uma frase (Olá! Mundo).



The screenshot shows the VISUALG 3.0.7.0 application window. The title bar reads "VISUALG 3.0.7.0 \* Interpretador e Editor de Algoritmos \* última atualização: 03 de Outubro de 2015 \* Entidade : UNIASSELVI - FAMEBLU". The menu bar includes "Arquivo", "Editar", "Run (executar)", "Exportar para", "Manutenção", and "Help (Ajuda)". The toolbar contains various icons for file operations, execution, and editing. The main area is titled "Área dos algoritmos ( Edição de código fonte ) -> Nome do arquivo: [OLAMUNDO.ALG]". The code is as follows:

```
1 Algoritmo "olamundo"  
2 // Disciplina   : [Lógica de Programação]  
3 // Professor    : Patrícia lima felismino  
4 // Descrição    : Função olamundo  
5 // Autor(a)     : Patrícia  
6 // Data atual   : 12/08/2025  
7 Var  
8 // Seção de Declarações das variáveis  
9 //identificador: tipo (declarando uma variável)  
10  
11  
12 Inicio  
13 // Seção de Comandos, procedimento, funções, operadores, etc...  
14 //escreva é diferente de escreval, o segundo pula a linha  
15 //escreval = escreva + linha = escreva e salte a linha  
16 escreva("Olá Mundo!")  
17 |  
18  
19  
20 Fimalgoritmo
```

# Conhecendo alguns pontos importantes

- Vamos conhecer as palavras reservadas
- Os tipos primitivos
- Vamos lembrar como funcionam as variáveis
- Hora de vermos como acontece o funcionamento das variáveis
- O que é escopo?

# Escopo, do que se trata?

- OBS! Escopo é tudo aquilo que está dentro dos limites de um projeto de software. Ele define o que será feito (funcionalidades, tarefas, entregas) e o que não será feito. É como desenhar uma cerca em volta do projeto para saber exatamente o que está incluído.
- Se o escopo não estiver bem definido, o projeto pode sair do controle — com prazos estourados, custos extras e confusão entre a equipe e o cliente.

# Exemplo de um escopo de um projeto

- Escopo definido:
- O usuário pode adicionar itens à lista ✓
- O usuário pode marcar itens como comprados ✓
- O usuário pode apagar itens da lista ✓
- Fora do escopo:
- Sincronizar com outros dispositivos ✗
- Criar gráficos de gastos mensais ✗
- Compartilhar a lista com outras pessoas ✗
- Essas funcionalidades extras não fazem parte do escopo atual mas podem vir a fazer parte em outro momento. Se o cliente quiser incluir mais coisas, o escopo precisa ser revisado, e isso pode impactar o tempo e o custo do projeto.

# Tipos primitivos no visualg

- INTEIRO – armazena números inteiros (ex: 10, -3, 0)
  - REAL – armazena números com casas decimais (ex: 3.14, -0,5, 2.0)
  - CARACTERE – armazena um único símbolo ou letra (ex: 'A', '%', '9')
  - LITERAL – armazena textos ,ou seja, sequência de caracteres (ex: "olá" , "123")
  - LÓGICO - armazena uma valor lógico (ex: verdadeiro, falso)
- 
- OBS! Vamos ao final criar um algoritmo que use todos os tipos primitivos, para ver como eles são armazenados e separados.



# Entendendo alguns pontos

- 📄 Explicação rápida:
- `var`: declara as variáveis.
- `leia()`: recebe dados do usuário.
- `escreva()`: exibe mensagens na tela.
- `caractere`: tipo de dado para textos.
- `inteiro`: tipo de dado para números inteiros.

# Testando todos os tipos primitivos usados no visualg

- Use inteiro para contar coisas.
- Use real para medidas com decimais.
- Use caractere para uma letra ou símbolo.
- Use literal para frases ou palavras.
- Use lógico para decisões (sim ou não).

# Algoritmo em português para ver o funcionamento dos tipos primitivos

- algoritmo "SolicitaNomeEIdade"
- var
  - nome: caractere
  - idade: inteiro
- inicio
  - escreva("Digite seu nome: ")
  - leia(nome)
- escreva("Digite sua idade: ")
- leia(idade)
- escreva("Olá, ", nome, "! Você tem ", idade, " anos.")
- fimalgoritmo

# O que o algoritmo a seguir vai fazer?

- Pede ao usuário que digite dados de diferentes tipos.
- Usa variáveis de todos os tipos primitivos.
- Exibe tudo de forma organizada no final

OBS! Vamos usar todos os tipos primitivos no algoritmo a seguir

# Algoritmo com todos os tipos primitivos

```
Área dos programas ( Edição do código fonte ) -> Nome do arquivo: [OLAMUNDO.ALG]
3  var
4      idade: inteiro
5      altura: real
6      letraInicial: caractere
7      nome: literal
8      aprovado: logico
9
10 inicio
11     escreva("Digite seu nome: ")
12     leia(nome)
13
14     escreva("Digite sua idade: ")
15     leia(idade)
16
17     escreva("Digite sua altura (ex: 1.75): ")
18     leia(altura)
19
20     escreva("Digite a letra inicial do seu nome: ")
21     leia(letraInicial)
22
23     escreva("Você foi aprovado? (verdadeiro ou falso): ")
24     leia(aprovado)
25
26     escreval()
27     escreval("Resumo das informações:")
28     escreval("Nome: ", nome)
29     escreval("Idade: ", idade)
30     escreval("Altura: ", altura)
31     escreval("Letra inicial: ", letraInicial)
32     escreval("Aprovado: ", aprovado)
33 finalgoritmo
```