

# wiRegras Semânticas

## Tipos de Variáveis:

- interu - números inteiro
- stringo - conjunto de letras
- float - ponto flutuante

A declaração de variáveis deve começar com o tipo nome da variável e o terminador ou valor.

- Não será permitido a declaração da variável cujo valor for diferente do tipo da variável.

inteirtu age ;

inteiru age2 = 17;

inteiru name = p3dro; O tipo da variável não aceita!

## Início do programa:

O programa deverá sempre ser iniciado com a expressão:

```
main Claud ( kd gaucho ) {  
  
    código ;  
  
};
```

## Formatação dos CódigoCs;

Cada instrução tem um símbolo terminador “ ; ”

## Declaração de Ponteiro:

Para declarar um ponteiro basta informar o tipo seguido por \* e o nome seguido pelo símbolo terminal.

## Declaração de vetor:

Uma declaração de ponteiro é válida se for um tipo seguido por "" e um identificador e um ponto e vírgula.

## Declaração de estrutura condicional:

### Estrutura:

```
SE BAH ( condição ) {
```

```
// instruções
```

```
}
```

```
SE BAH ( condição ) {
```

```
// instruções
```

```
} TCHE {
```

```
// instruções
```

```
}
```

### Tipos de Operadores:

- ">" Maior
- ">=" Maior igual
- "<" Menor
- "<=" Menor igual
- "==" Igual
- "<>" Diferente

### Declaração de Comentário:

Os comentários são escritos da seguinte forma , símbolo seguido do comentário

- @
- #
- /\* \*/

@ e # , usados para linha

/\* \*/ , usados para blocos de comentários

## Declaração de entrada e saída

- **ClaudDiz ( valor )**

Usando para imprimir algo na tela, tem como parâmetro um valor inteiro, string com “ ”, ou uma variável.

Não pode ser usado expressões ou combinações.

- **Receba( tipo nome)**

Usado para receber e armazenar uma variável informada pelo usuário.

Não pode receber uma variável já existente ou palavras chaves.

## Operações Aritméticas

A linguagem possui todas as operações porém não obedece ordem de precedência.

- /
- \*
- +
- -

## Declaração de Constante

A linguagem possui a constante, deve ter o tipo seguido do nome da constante e o valor.

```
const nome = “pedro”;
```

## Declaração de Função

Deve começar com FUNC seguido do nome e ( ) junto com o bloco de código;

```
FUNC nome "(" ")" {
```

```
}
```

## Estrutura de Repetição

- ATE
- FAZ ATÉ

ATE, é um laço simple que executa até a operação for verdadeira.

```
ATE ( operações logica ) {
```

```
}
```

**FAZ AI**, é um laço com parametros que inicializa uma variavel, verifica a condição e executa algo, como incrementar ou decrementar.

```
FAZAI ( Type Ident = VALUE | OPL | INCREMENTO ) {  
  
}
```

**Declaração de Desvio:**

- **GOTO**

Comando para desviar para uma label declarada

**Declaração de Label:**

- **LABEL**

Comando para criar uma label onde o codigo pode ser desviado

**Declaração de Struct:**

uma declaração de uma estrutura capaz de armazenar valores heterogeneos de diferentes tipos.

```
ISTRUCT nome [ Integer ] = { Valores, Valores };
```

```
ISTRUCT inventaio[ 3 ] = { "lapis" , 17 , 2,00 }
```