

Alunos: Bruno Luis Carletto e Willian Alves dos Santos

Disciplina: Compiladores

## Documentação Linguagem izi-lang

**Comandos ignorados pela linguagem:** \t, \n, \r, sua utilização não irá causar nada no programa e não irá alterar seu funcionamento.

**Final de linha:** O caractere utilizado para encerrar uma linha é o ponto e vírgula (;).

**Letras e dígitos:** As letras que poderão ser utilizadas para a criação das variáveis e que serão reconhecidas pela linguagem é o alfabeto comum de A até Z. Os dígitos aceitados pela linguagem são de 0 a 9.

### Operadores:

Operadores Aritméticos	Descrição
+	Adição: utilizado para a realização de somas aritméticas.
-	Subtração: utilizado para a realização de subtrações aritméticas.
*	Multiplicação: Utilizado para realizar multiplicações.
/	Divisão: Utilizado para realizar divisões.
%	Módulo: Utilizado para calcular o modulo de um valor.
Operadores Lógicos	Descrição
>	Maior: Utilizado para a verificação a comparação de dois ou mais valores. (Ex: a>5)
<	Menor: Utilizado para a verificação a comparação de dois ou mais valores. (Ex: a<5)
==	Igualdade: Utilizado para verificar a igualdade de valores. (Ex: a==5)
!=	Diferença: Utilizado para verificar a diferença de valores. (Ex: a!=5)

**Estrutura:** Os caracteres utilizados para início e final de instruções são as chaves ({}), e os caracteres de interação são os parênteses ( ).

## Variáveis:

Tipo de Variável	Descrição
<b>int</b>	Inteiro: números inteiros. (32 bits)
<b>string</b>	String: letras e dígitos.
<b>byte</b>	Byte:
<b>short</b>	Short:
<b>long</b>	Long: números decimais. (64 bits)
<b>float</b>	Float: números decimais. (32 bits)
<b>double</b>	Double: números decimais.
<b>char</b>	Char:
<b>boolean</b>	Boolean: estados de decisão (true ou false).

Valores Booleanos: Os dados do tipo booleano podem assumir dois estados somente, True para verdadeiro e False para falso.

**Declaração e atribuição de variáveis:** Para uma variável ser declarada ela deverá conter um o identificador var, tipo, nome, e um valor poderá ser atribuído. O nome da variável deverá conter pelo menos uma letra, poderão também ser incluídos números ou o caractere underline (\_). A atribuição de valores deverá ser feita através do caractere <-, o final da criação deverá ser feito através de ponto e vírgula (;).

Exemplo de declaração e atribuição de variável: `int a<-5;`

**Operações aritméticas:** As operações aritméticas deverão conter no mínimo um valor (letra ou dígito) e um operador aritmético, também poderá ser atribuído uma expressão a uma variável utilizando o caractere <-. (OBS: Deve-se sempre começar com o caractere (\$))

**Operações booleanas:** As operações booleanas deverão conter dois valores (letra ou número) e um operador que será utilizado para a comparação dos valores e determinar o valor de saída (true ou false).

**Comando FOR:** o comando FOR é utilizado como estrutura de repetição e sua declaração será feita da seguinte forma:

Comando for: for, abertura de parênteses, utilização ou declaração de uma variável de controle, expressão booleana para o controle do número de expressões, expressão aritmética para o controle do ritmo de repetição, fechamento de parênteses, abertura de chave, instruções para serem realizadas (podendo conter outras funções como estruturas de decisão ou expressões aritméticas ou booleanas), fechamento de chave.

Exemplo comando for:

```
for(var int i <- 1; i < 5, $i <- i + 1;){  
  &i <- i + 1;  
}
```

**Comando WHILE:** o comando WHILE é utilizado como estrutura de repetição e sua declaração será feita da seguinte forma:

Comando while: while, abertura de parênteses, expressão booleana para o controle do número de repetições, fechamento de parênteses, abertura de chave, instruções para serem realizadas (podendo conter outras funções como estruturas de decisão ou expressões aritméticas ou booleanas), fechamento de chave.

Exemplo comando while:

```
while(a === 1){  
    $a <- a + 1;  
}
```

**Comando IF/ELSE IF/ELSE:** O comando é IF/ELSE IF/ELSE utilizado como estrutura de decisão e sua declaração será da seguinte forma:

Comando if: if, abertura de parênteses, expressão booleana para decisão da próxima etapa (se a expressão booleana for verdadeira prosseguir o if normalmente, senão ir para o comando else if ou else), fecha parênteses, abre chave, instruções a serem realizadas (podendo conter outras funções como estruturas de repetição ou expressões aritméticas ou booleanas), fecha chave.

Comando else if: else if, abertura de parênteses, expressão booleana para decisão da próxima etapa (se a expressão booleana for verdadeira prosseguir o if normalmente, senão ir para o comando else if ou else), fecha parênteses, abre chave, instruções a serem realizadas (podendo conter outras funções como estruturas de repetição ou expressões aritméticas ou booleanas), fecha chave.

Comando else: else, abertura de chave, instruções a serem realizadas (podendo conter outras funções como estruturas de repetição ou expressões aritméticas ou booleanas), fecha chave.

Exemplo IF/ELSE IF/ELSE:

```
if(a != 6){  
    $a <- a + 1;  
}  
elif(b === 5){  
    $b <- a + 1;  
}  
else{  
    $c <- a + 1;  
}
```

**Comando fopen():** Comando para a abertura de arquivos de texto para leitura. Comando começa com fopen, abre parênteses, caminho, fecha parênteses, ponto e vírgula.

Exemplo:

```
fopen(caminho);
```

**Comando SWITCH/CASE:** O comando é SWITCH/CASE utilizado como estrutura de decisão e sua declaração será da seguinte forma:

Comando switch/case: switch, abre parênteses, variável utilizada, fecha parênteses, abre chave, case, número do case, abre chaves, instrução do case, break, ponto e vírgula, fecha chave duas vezes.

O comando switch irá escolher o case com número correspondente ao da variável e executará o comando presente nele.

Exemplo:

```
switch(aa){  
  case 1 {  
    $a <- a + 1;  
    break;  
  }  
}
```

**Entrada e Saída de dados:** a entrada e saída de dados é feita através dos comandos in e out. O comando IN irá atribuir um dado para uma determinada variável enquanto o dado out irá apresentar o valor exato desta variável.

Exemplo:

Entrada

```
&i <- get(int);
```

Saída

```
**show(i);
```

## RELATO DAS DIFICULDADES

A Principal dificuldade que passamos foi integrar todas as instruções para serem realizadas em qualquer lugar. Assim como o switch/case que acabou atrasando o projeto. Tivemos dificuldades em relação aos comentários com o special\_token ou skip, por exemplo.