

# Compilador de Código PCode

Baseado no projeto do Professor Dr. Daniel Lucrédio da UFSCar, disponibilizado no [GitHub](#)

## Como usar

Para utilizar o compilador PCode, execute o JAR da seguinte forma:

```
java -jar pcode.jar -Input=<FILE> -Debug=<BOOLEAN> -WaitTime=<INT>
```

- **-Input | -i:** Caminho para o arquivo contendo o código P-code.
- **-Debug | -d:** (*Opcional*) true para habilitar o modo de depuração, false para desabilitar (padrão: true).
- **-WaitTime | -w:** (*Opcional*) somente para *debug*, define quanto tempo vai ser esperado até executar o próximo comando (padrão: 500ms).

## Instruções

As seguintes instruções são aceitas pela P-Code Machine:

Instrução	Descrição	Tipos de Token Usados
Entrada/Saída		
in	Lê um valor <b>X</b> da entrada e armazena <b>X</b> no topo da pilha.	DATA_TYPE
out	Retira um valor <b>X</b> do topo da pilha e escreve <b>X</b> na saída.	NUMBER, STRING, BOOL
Manipulação de Dados		
push C	Insere <b>C</b> no topo da pilha.	NUMBER, STRING, BOOL, ADDRESS, CHAR
pop	Remove o valor do topo da pilha.	ANY
dup	Duplica o valor no topo da pilha.	ANY
swap	Troca os dois valores do topo da pilha.	ANY
Aritmética		
to T	Retira <b>X</b> do topo da pilha, e insere <b>X</b> no tipo T (DATA_TYPE) no topo da pilha.	NUMBER
add	Retira <b>X</b> e <b>Y</b> do topo da pilha, e insere <b>X + Y</b> .	NUMBER
sub	Retira <b>X</b> e <b>Y</b> do topo da pilha, e insere <b>Y - X</b> .	NUMBER
mul	Retira <b>X</b> do topo da pilha, retira <b>Y</b> , e insere <b>X * Y</b> .	NUMBER
div	Retira <b>X</b> e <b>Y</b> do topo da pilha, e insere <b>Y / X</b> .	NUMBER
mod	Retira <b>X</b> e <b>Y</b> do topo da pilha, e insere <b>Y % X</b> (RESTO DA DIVISÃO).	NUMBER

Instrução	Descrição	Tipos de Token Usados
Lógicos		
grt	Retira <b>X</b> e <b>Y</b> , e armazena <b>Y &gt; X</b> (booleano) no topo da pilha.	NUMBER
let	Retira <b>X</b> e <b>Y</b> , e armazena <b>Y &lt; X</b> (booleano) no topo da pilha.	NUMBER
gte	Retira <b>X</b> e <b>Y</b> , e armazena <b>Y &gt;= X</b> (booleano) no topo da pilha.	NUMBER
lte	Retira <b>X</b> e <b>Y</b> , e armazena <b>Y &lt;= x&lt; code&gt; (booleano) no topo da pilha.</b>	NUMBER
equ	Retira <b>X</b> e <b>Y</b> , e armazena <b>X == Y</b> (booleano) no topo da pilha.	NUMBER, STRING
neq	Retira <b>X</b> e <b>Y</b> , e armazena <b>X != Y</b> (booleano) no topo da pilha.	NUMBER, STRING
and	Retira booleanos <b>X</b> e <b>Y</b> , e armazena <b>X &amp;&amp; Y</b> no topo da pilha.	BOOL
or	Retira booleanos <b>X</b> e <b>Y</b> , e armazena <b>X    Y</b> no topo da pilha.	BOOL
not	Inverte o valor booleano no topo da pilha.	BOOL
xor	Realiza a operação XOR entre dois valores booleanos que estão no topo da pilha.	BOOL
Memória		
lod	Troca o endereço presente no topo da pilha pelo valor salvo nesse mesmo endereço.	ADDRESS
sto	Retira <b>X</b> e <b>A</b> , e armazena <b>X</b> na memória no endereço <b>A</b> .	( <b>X</b> = NUMBER, STRING, BOOL, ADDRESS, CHAR), ( <b>A</b> = ADDRESS)
Controle de Fluxo		
L:	Marca uma posição de código com um rótulo <b>L</b> (LABELS).	LABEL
ujp L	Salta para a instrução marcada com <b>L</b> .	LABEL
fjp L	Retira booleano <b>X</b> , e se falso, salta para a instrução marcada com <b>L</b> .	BOOL, LABEL
tjp L	Retira booleano <b>X</b> , e se positivo, salta para a instrução marcada com <b>L</b> .	BOOL, LABEL
call	Chama uma função especificada por um rótulo.	LABEL

Instrução	Descrição	Tipos de Token Usados
ret	Retorna de uma função para a instrução anterior que está no topo da <b>CALL STACK</b> .	-
Bitwise		
shl	Retira um valor <b>X</b> do topo da pilha, um valor <b>Y</b> do topo da pilha e realiza um deslocamento à esquerda em <b>Y</b> na quantidade de <b>X</b> .	NUMBER
shr	Retira um valor <b>X</b> do topo da pilha, um valor <b>Y</b> do topo da pilha e realiza um deslocamento à direita em <b>Y</b> na quantidade de <b>X</b> .	NUMBER
Strings		
sln	Calcula o comprimento de uma string no topo da pilha.	STRING
sct	Concatena duas string/char no topo da pilha.	STRING, CHAR
Finalização		
hlt	Interrompe a execução.	-

Observações

- **NUMBER**: Representa valores inteiros ou de ponto flutuante.
- **FLOAT**: Representa números de ponto flutuante especificamente.
- **BOOL**: Representa valores booleanos (**true** ou **false**).
- **ADDRESS**: Representa endereços de memória.
- **STRING**: Representa cadeias de caracteres.
- **CHAR**: Representa um caractere único.
- **DATA\_TYPE**: Representa os tipos de dados, como **int**, **float**, **long**, **double**.
- **LABEL**: Representa rótulos usados nas instruções de salto.