



# PROGRAMAÇÃO 101

## 4 - VARIÁVEIS



# O QUE É UMA VARIÁVEL?

**É UMA POSIÇÃO DE MEMÓRIA ONDE PODEMOS  
GUARDAR UM DADO OU VALOR E MODIFICÁ-LO  
AO LONGO DA EXECUÇÃO DO PROGRAMA**

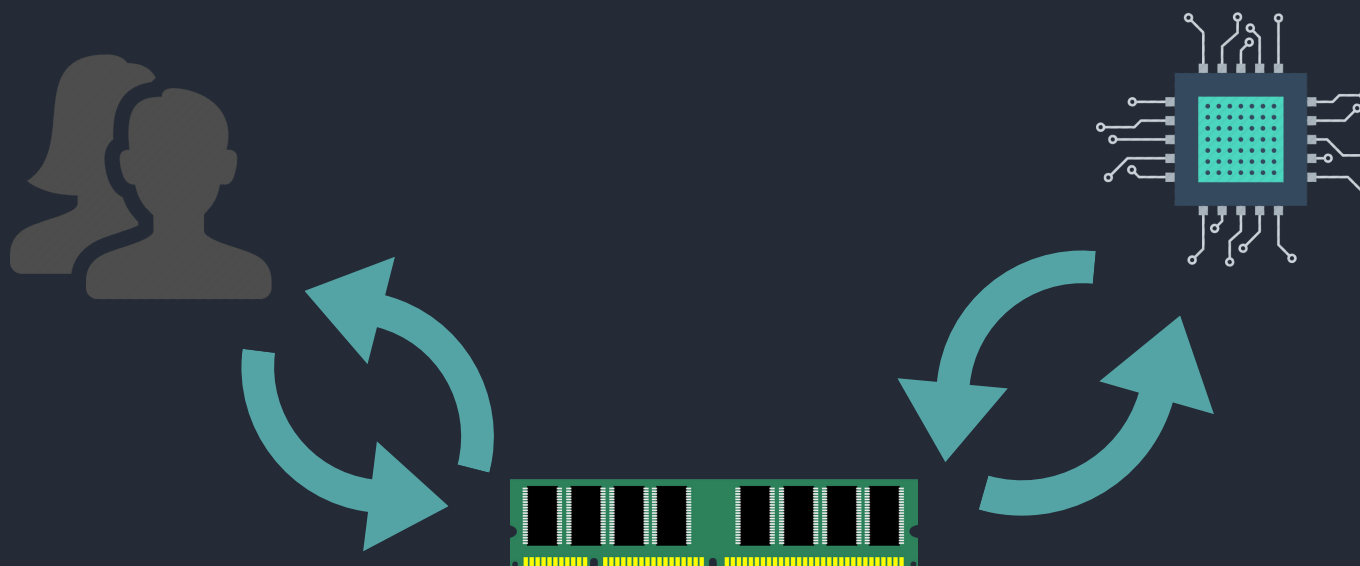
**A MEMÓRIA USADA É A PRINCIPAL (RAM)**





# O QUE É UMA VARIÁVEL?

ATUA COMO ARMAZENAMENTO INTERMEDIÁRIO  
DE DADOS DURANTE A EXECUÇÃO DO  
PROGRAMA



# DECLARANDO E NOMEANDO VARIÁVEIS





# DECLARAÇÃO

## FORMA GERAL:

**Tipo\_da\_variável** nome\_da\_variável;

O TIPO DA VARIÁVEL DETERMINA O CONJUNTO DE VALORES E OPERAÇÕES QUE UMA VARIÁVEL ACEITA.

O NOME DA VARIÁVEL É COMO O PROGRAMADOR IDENTIFICA ESSA VARIÁVEL DENTRO DO PROGRAMA





## IMPORTANTE

É POSSÍVEL NOMEAR MAIS DE UMA VARIÁVEL NO MESMO TIPO, BASTA SEPARAR POR UMA VÍRGULA

UMA VARIÁVEL DEVE SEMPRE SER DECLARADA ANTES DE SER USADA NO PROGRAMA

ANTES DE USAR O CONTEÚDO DE UMA VARIÁVEL CRIADA, TENHA CERTEZA DE QUE ELE FOI DEFINIDO ANTES



# REGRAS DE NOMEAÇÃO

**O NOME DE UMA VARIÁVEL PODE SER UMA MISTURA DE**

- LETRAS,
- NÚMEROS
- OU UNDERSCORES (\_)

**NÃO PODE COMEÇAR COM NÚMERO**

**PALAVRAS CHAVE NÃO PODEM SER USADAS COMO VARIÁVEIS**





# IMPORTANTE

NA LINGUAGEM C, LETRAS MAIÚSCULAS E MINÚSCULAS  
SÃO CONSIDERADAS DIFERENTES



# DEFININDO E MODIFICANDO OS TIPOS





# EXISTEM 4 TIPOS

- **char**
  - **int**
  - **float**
  - **double**
- 



# INT

PERMITE ARMAZENAR NÚMEROS INTEIROS

INTERVALO DE VALORES

-2.147.483.648 A 2.147.483.647

SEU TAMANHO DEPENDE DO PROCESSADOR E É TIPICAMENTE 16  
OU 32 BITS

```
int n = 1459;
```

```
int i = 1254854;
```





# CHAR

ARMAZENA O VALOR NUMÉRICO DE UM CARACTERE DA TABELA ASCII

## INTERVALO DE VALORES

-128 A 127

PODE SER UTILIZADO PARA ARMAZENAR NÚMEROS PEQUENOS EM UM ÚNICO BYTE (8 BITS) OCUPANDO POUCA MEMÓRIA

```
char c = 'a';
```

```
char s = 97;
```



Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL



# FLOAT

ARMAZENA VALORES REAIS (COM PARTE FRACIONÁRIA)

## INTERVALO DE VALORES

1,175494E-038 A 3,402823E+038

É CHAMADA DE FLOAT POIS A REPRESENTAÇÃO DO PONTO  
"FLUTUA" DE POSIÇÃO

```
float f = 5.25;
```





# DOUBLE

## INTERVALO DE VALORES

2,225074E-308 A 1,79693E+308


PERMITE ARMAZENAR UM VALOR REAL DE PRECISÃO DUPLA

```
double d = 15.673;
```





# EXISTEM 4 MODIFICADORES

- **signed**
  - **unsigned**
  - **short**
  - **long**
- 





# O QUE SÃO MODIFICADORES?

**PERMITEM FAZER PEQUENAS ALTERAÇÕES  
NO TIPO DE VARIÁVEL ESCOLHIDA, DE MODO A  
ADEQUÁ-LO ÀS NECESSIDADES DO NOSSO  
PROGRAMA**





# SIGNED

**DETERMINA QUE UMA VARIÁVEL DECLARADA DOS TIPOS CHAR OU INT PODERÁ TER VALORES POSITIVOS OU NEGATIVOS**

**TRATA-SE DO MODO PADRÃO DE DEFINIÇÃO E POR ISSO É RARAMENTE USADO**

```
signed char x;  
signed int y;
```





# UNSIGNED

**DETERMINA QUE UMA VARIÁVEL DOS TIPOS CHAR OU INT  
PODERÁ TER VALORES APENAS POSITIVOS OU 0**

**O BIT DE SINAL SERÁ USADO PARA GUARDAR NÚMEROS,  
AUMENTANDO A SUA CAPACIDADE DE ARMAZENAMENTO**

```
unsigned char x;  
unsigned int y;
```





# SHORT

**DETERMINA QUE UMA VARIÁVEL DO TIPO INT TERÁ APENAS 16 BITS INDEPENDENTE DO PROCESSADOR**

**ÚTIL EM PROGRAMAS QUE RODARÃO EM APARELHOS COM RESTRIÇÃO DE MEMÓRIA COMO IOT, POR EXEMPLO**

```
short int i;
```





# LONG



**DETERMINA QUE UMA VARIÁVEL DO TIPO INT TERÁ 32 BITS  
INDEPENDENTE DO PROCESSADOR**

**DETERMINA QUE O DOUBLE POSSUA MAIOR PRECISÃO**

**long int n;**

**long double d;**





## IMPORTANTE

A LINGUAGEM C PERMITE QUE SE UTILIZE MAIS DE UM MODIFICADOR NO MESMO TIPO

```
unsigned long int m;
```