

CENTEC

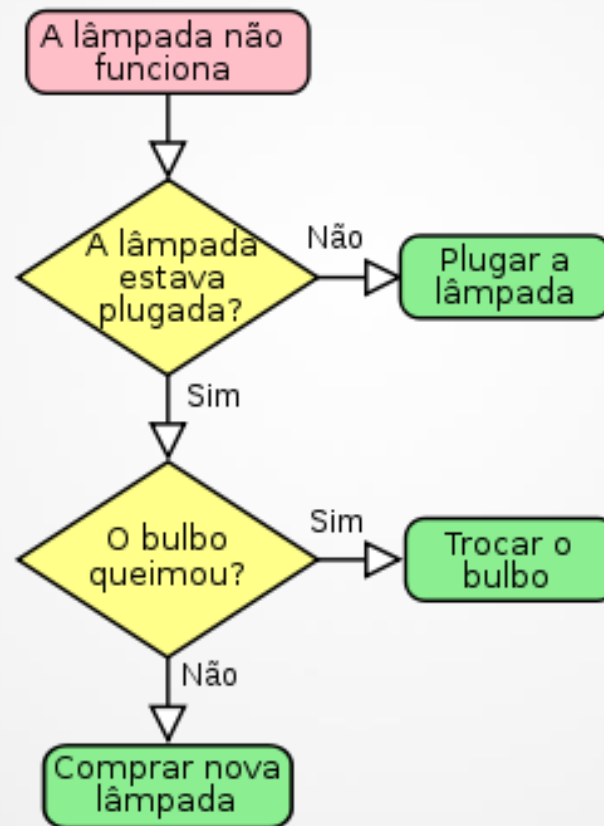
INSTITUTO CENTRO DE ENSINO TECNOLÓGICO

Linguagem de Programação
Microcontroladores e Microprocessadores
Especialização em Automação Industrial

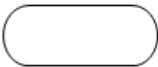


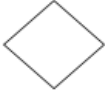




- Computadores surgiram inicialmente da necessidade de automatizar tarefas repetitivas. Com o passar do tempo as máquinas foram desenvolvidas para atender a tarefas cada vez mais complexas.



•Com a intenção de facilitar a descrição do problema e como ele deve ser resolvido, é viável utilizar algumas técnicas. O fluxograma é um método gráfico de estabelecer o sequenciamento das ações com suas respectivas condições.



•Com a intenção de facilitar a descrição do problema e como ele deve ser resolvido, é viável utilizar algumas técnicas. O fluxograma é um método gráfico de estabelecer o sequenciamento das ações com suas respectivas condições.

SIMBOLO	FUNÇÃO
	TERMINAL Usado no início e no fim de um processo.
	PROCESSAMENTO Usado para processamento de um modo geral.
	ENTRADA / SAIDA Usado para operações de entrada e saída de dados.
	DECISÃO Usado para tomada de decisões.
	DESVIO Usado para desviar o código para um ponto do programa.
	ENTRADA MANUAL Usado para indicar entrada de dados através do teclado.
	EXIBIR Usado para mostrar informações e resultados.
	RELATÓRIO Usado para exibir relatórios.

- Estabelece a relação entre os resultados esperados e suas respectivas combinações de entradas. Assim como o fluxograma, auxilia na construção de código, especificamente nas estruturas de decisão

A	B	C	D	S
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

- Para sistemas digitais, o número de linhas numa tabela verdade é 2^n . A base é 2 por se tratar de sistemas em que as entradas e saídas atendem a apenas dois estados (0 ou 1) e “n” é o número de entradas.
- O preenchimento das colunas das entradas ocorre alternadamente de um em um para a primeira coluna da direita, de dois em dois para a coluna seguinte à esquerda, de quatro em quatro, oito em oito e assim em diante.

E3	E2	E1	Saída
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Baixo nível (Assembly)

```
; Hello World for Intel Assembler (MSDOS)

mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h

Hello:
  db "Hello World!",13,10,"$"
```

Alto nível (Portugol)

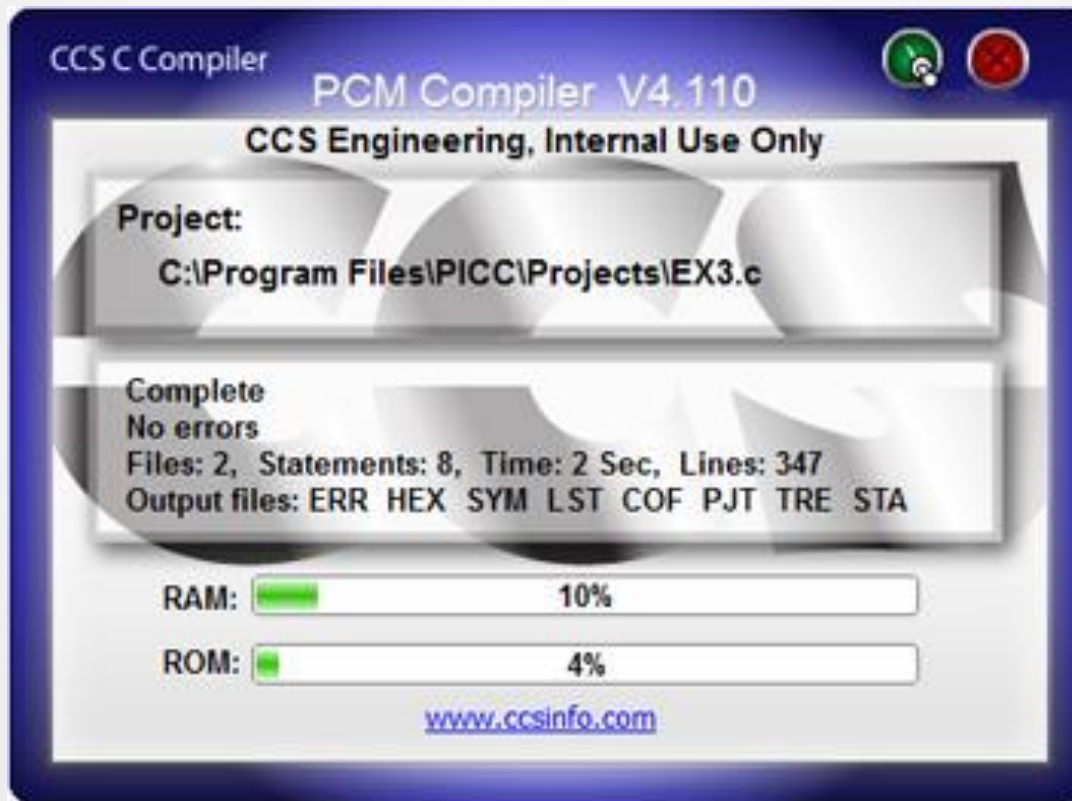
```
programa Olamundo;
inicio
    Escreva ("Olá Mundo");
fim
```

- Programação Estruturada (C);
- Programação Orientada a Objetos (JAVA);
- Programação Orientada Fluxo de Dados (Com diagramas);
- Programação Orientada a Regras (Mathematica).

Portugol	C
var	var
inicio	begin
escreva ("Olá mundo!");	print ("Olá mundo!");
fim	end

- Converte uma linguagem de nível maior para linguagem de máquina, para que então o processador possa interpretar e executar.

Compilador para PIC



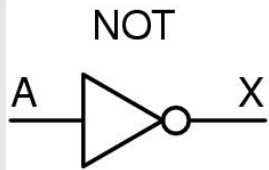
- **Tradutor:** Converte uma linguagem de nível maior para uma linguagem de nível intermediário.
- **Interpretador:** Faz a leitura da linguagem de máquina e verifica se ela está correta. Em caso positivo, executa instrução por instrução até o fim do código.

Tipo	Descrição
int	Valores numéricos inteiros
char	Caracteres
float	Valores numéricos reais
boolean	Verdadeiro ou falso

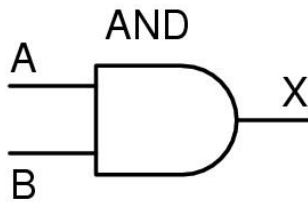
Tipo	Num de bits	Formato para leitura com scanf	Intervalo	
			Início	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto de divisão

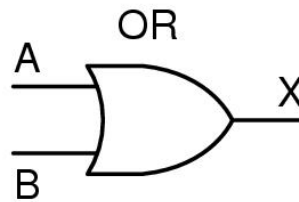
Operador	Descrição
>	Maior que
>=	Maior ou igual
<	Menor que
<=	Menor ou igual
==	Igualdade
!=	Diferença



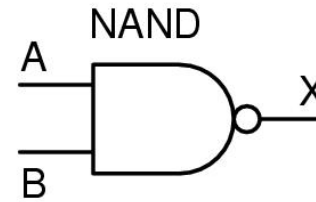
A	X
0	1
1	0



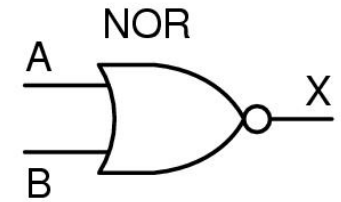
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Operador	Descrição
&&	AND
	OR
!	NOT

Operador	Descrição
&	AND
	OR
~	NOT
^	XOR

Operador	Descrição
=	Simples
+=	Aditiva
-=	Subtrativa
*=	Multiplicativa
/=	Divisão
%=	Módulo

- Seleção simples;
- Seleção dupla;
- Seleção aninhada.

SIMPLES:

```
if (<expressão lógica>){  
<sequência de comandos>  
}
```

DUPLA:

```
if (<expressão lógica>){  
<sequência de comandos 1>  
}else{  
<sequência de comandos 2>  
}
```

EXEMPLO 1:

```
If (<expressão lógica>){  
    If (<expressão lógica>){  
        <sequência de comandos>  
    }  
}
```

EXEMPLO 2:

```
if (<expressão lógica>){  
    <sequência de comandos 1>  
}else{  
    If (<expressão lógica>){  
        <sequência de comandos 2>  
    }  
}
```

- Com precondição;
- Condensado.

COM PRECONDIÇÃO:

```
while (<expressão lógica>){  
<sequência de comandos>  
}
```

CONDENSADO:

```
for (<inicialização>; <expressão lógica>; <incremento>){  
<sequência de comandos>  
}
```


1. Diferencie operadores lógicos de operadores lógicos bit a bit.
2. Que diferença existe entre “=” e “==”?
3. Quando se deve usar “while”, “if” e “for”?
4. Diferencie semântica e sintaxe em linguagens de programação.
5. Diferencie IDE, Compilador, Tradutor e Interpretador.