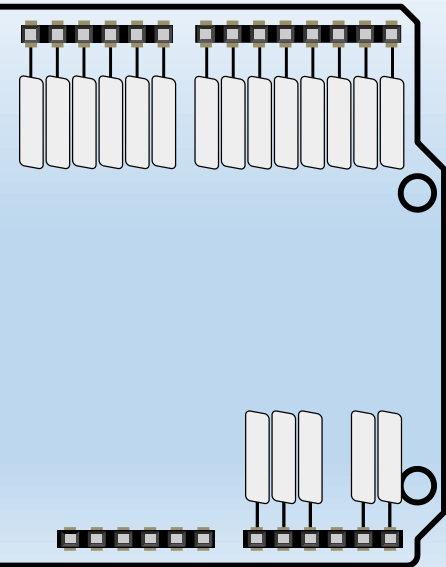


Estruturas de repetição

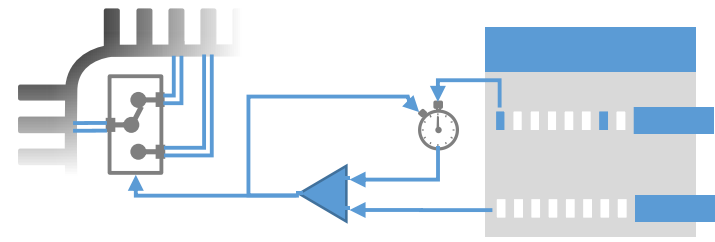


Para atingir a excelência em qualquer coisa na vida, é preciso repetir e treinar. Treinar e repetir, aprender a técnica de tal maneira que ela se torne intuitiva.

Paulo Coelho, Aleph

Estruturas de repetição

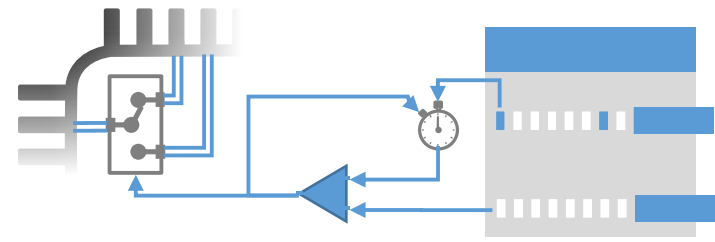
- Laços de repetição são estruturas computacionais que permitem a repetição de um trecho de código N vezes ou enquanto uma condição for verdadeira.
- Todo loop deve possuir uma condição que indique quando este deve terminar. Uma condição mal feita pode prender o programa dentro do loop
- Esta é uma das causas mais comuns para o "travamento" dos aplicativos, comumente chamada de loop infinito



Estruturas de repetição

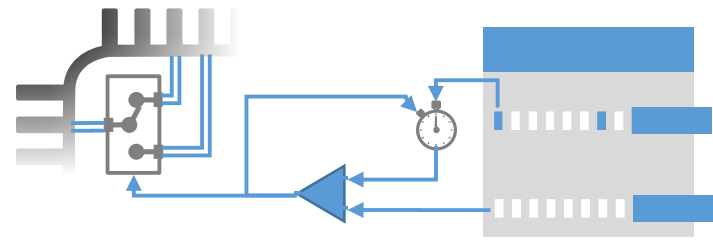
- A repetição com teste no início do loop é usada para repetir N vezes uma ou mais instruções.
- Não é necessário conhecer com antecedência o número de repetições.
- O controle do loop é feito através de uma condição.
- Para que o sistema **não** entre em "loop infinito" a condição **tem** que ser alterada em algum momento **dentro** do loop

```
while( condicao ){  
    comando1;  
    comando2;  
    alteracao da condicao;  
}//end while
```



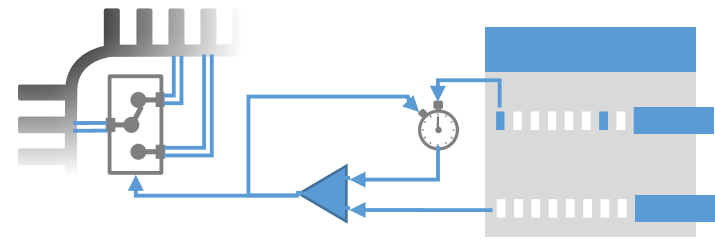
Exemplo

- Fazer um programa que:
 - Leia o valor do salário dos funcionários de uma empresa.
 - Ao terminar de ler os valores, deve imprimir a soma dos salários.
 - A quantidade de funcionários não é conhecida.



Exemplo

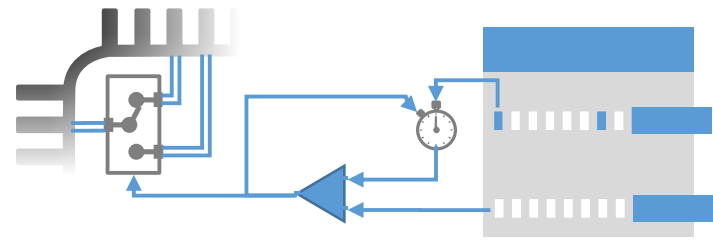
```
#include<stdio.h>
int main(int argc, char *argv[]){
    float total = 0.0, salario=1;
    while( salario > 0 ){ //0 - sai do programa
        printf("Digite o valor de salario = ");
        scanf("%f", &salario);
        total = total + salario;
    }//end while
    printf("Somatório = %f", total);
    return 0;
}//end main
```



Exemplo

- Fazer um programa que:
 - Fique aguardando alguma tecla ser pressionada

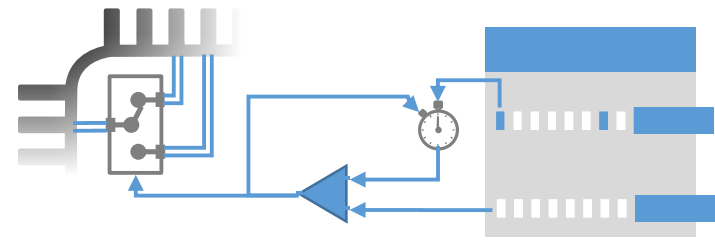
```
x = kpRead();  
//0 indica que não houve botão pressionado  
while(x == 0){  
    x = kpRead();  
}
```



Estruturas de repetição

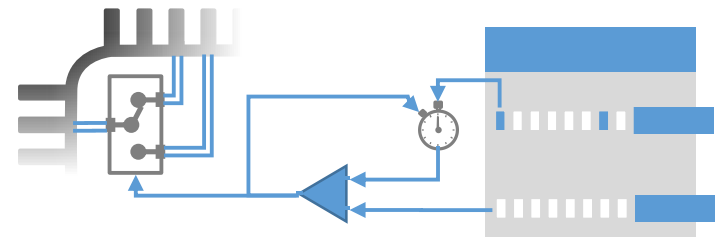
- Assim como a instrução while a instrução do..while é utilizada para repetirmos um bloco do algoritmo diversas vezes.
- A diferença é o ponto onde a verificação da condição é realizada.
- Mesmo que a condição seja falsa desde o início, na estrutura do...while o bloco é executado pelo menos uma vez.

```
do{  
    comando1;  
    comando2;  
    alteracao da condicao  
}while( condicao )
```



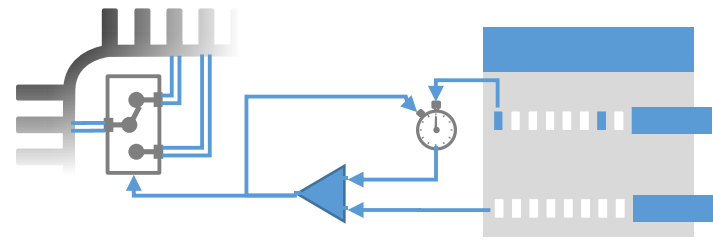
Exemplo

- Fazer um algoritmo que:
 - Leia o valor do salário dos funcionários de uma empresa.
 - Ao terminar de ler os valores deve imprimir a soma dos salários.
 - A quantidade de funcionários não é conhecida.



Exemplo

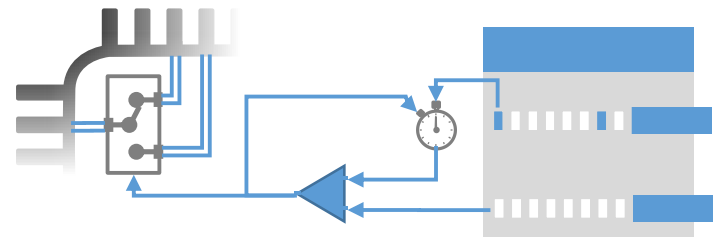
```
#include<stdio.h>
int main(int argc, char *argv[]){
    float total = 0.0, salario;
    do{
        printf("Digite o valor de salario = ");
        scanf("%f", &salario);
        total = total + salario;
    } while(salario > 0) //0 - sai do programa
    printf("Somatório = %f\n", total);
    return 0;
} //end main
```



Estruturas de repetição

- Diferentemente das duas formas de loop apresentadas anteriormente a repetição com variável de controle for, é utilizada para repetir um bloco de instruções com uma quantidade de repetições pré-estabelecida.
- Para atingir este objetivo utilizamos dentro desta estrutura uma variável que trabalha como um contador. Esta indicará a quantidade de vezes que o bloco de instruções será repetido.

```
for(inicializacao; teste; incremento){  
    comandos;  
}//end for
```



Exemplo

- Enviar para o LCD uma mensagem, um caracter por vez

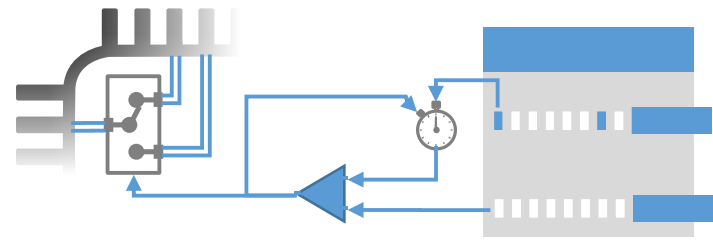
```
//o texto "Hello World" possui 11 caracteres
```

```
//a variável message possui 12 posições para armazenar  
também o caracter de terminação '\0'
```

```
char message[] = "Hello World";
```

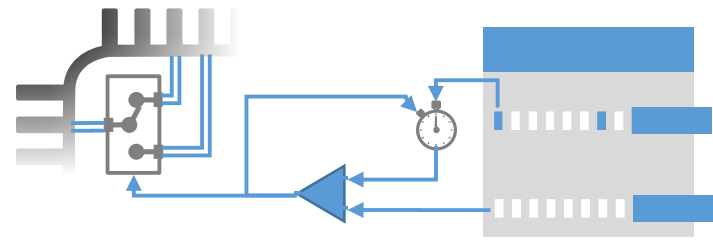
```
//como não é necessário imprimir o '\0', paramos a  
contagem no décimo primeiro elemento
```

```
for (i=0; i<11; i++) {  
    lcdChar(message[i]);  
}
```



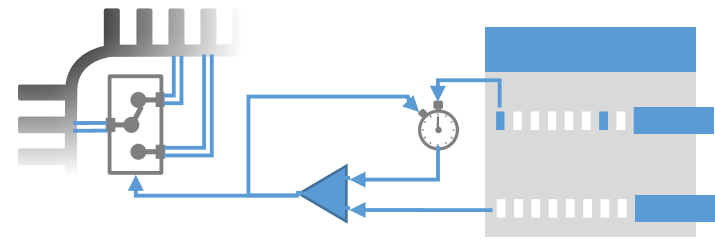
Exemplo

- Fazer um programa em C que:
 - Leia cinco valores dados pelo usuário
 - Some o triplo de cada valor
 - Imprima o somatório na tela



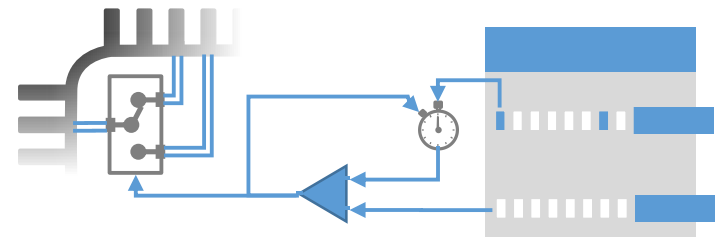
Exemplo

```
#include<stdio.h>
int main(int argc, char *argv[]){
    int total = 0, cont, num;
    for(cont = 0; cont < 5; cont++){
        printf("Digite um numero = ");
        scanf("%d", &numero);
        total = total + (numero * 3);
    }//end for
    printf("Somatório = %d", total);
    return 0;
} //end main
```



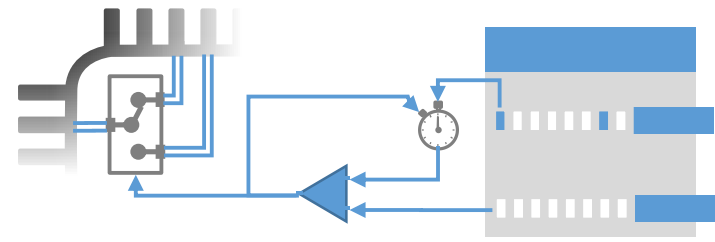
Alteração de fluxo do Loop

- Os comandos break e continue permitem ao programador alterar o fluxo do programa dentro de um loop.
- break
 - Ao utilizar o comando break, o loop é parado imediatamente, independente das condições.
 - O programa tem sequencia no primeiro comando depois do loop.
 - Também é utilizado em conjunto com o comando switch
- continue
 - Ao utilizar o comando continue, a iteração atual do loop para de ser executada e o loop reinicia.
 - Se for usada dentro de um loop do tipo for o bloco de incremento é executado.



Alteração de fluxo do Loop

- Fazer um algoritmo que:
 - Leia o valor do salário dos funcionários de uma empresa.
 - Ao terminar de ler os valores deve imprimir a soma dos salários.
 - A quantidade de funcionários não é conhecida.



Exemplo

```
#include<stdio.h>
typedef enum {false,true} bool;
int main(int argc, char *argv[]){
    float total = 0.0, salario;
    while(true){
        printf("Digite o salario = ");
        scanf("%f", &salario);
        if(salario==0){
            break;
        }
        //se o salario = 0, sai do loop
    }
    total = total + salario;
    printf("Total salario = %f", total);
    return 0;
}
```

