

ESTRUCTURAS ITERATIVAS

- PERMITEN REPETIR UNA INSTRUCCIÓN O UNA SECUENCIA DE INSTRUCCIONES
- EXISTEN TRES Y LAS USAMOS:
 - WHILE.- SI EL BLOQUE DE INSTRUCCIONES PUEDE QUE NO SE EJECUTE NI UNA SOLA VEZ.
 - DO.- SI ES NECESARIO QUE EL BLOQUE O INSTRUCCIÓN SE EJECUTE AL MENOS EN UNA OCASIÓN.
 - FOR .- NORMALMENTE CUANDO CONOCEMOS EL NUMERO EXACTO DE REPETICIONES.

while

- LAS INSTRUCCIONES SE EJECUTAN 0 O MAS VECES.
- ENTRA AL BLOQUE DE INSTRUCCIONES CUANDO LAS EXPRESIÓN ES EVALUADA COMO VERDADERA
- RECUERDE QUE SI ES MÁS DE UNA INSTRUCCIÓN DEBEN DE IR ENTRE LLAVES.
- SINTAXIS
while(exp)
Instrucción(es);

NOTA: DESPUÉS DE LA EXPRESIÓN NO LLEVA “;” DE HACERLO SE EVALÚA LA EXPRESIÓN PERO NO SE REALIZA NADA LO QUE LO CONVERTIRÍA EN UN CICLO INFINITO

do...while

- SUS INSTRUCCIONES SE EJECUTAN UNA O MÁS VECES
- REGRESA AL BLOQUE DE INSTRUCCIONES CUANDO LA EXPRESIÓN ES VERDADERA
- SINTAXIS

```
do{  
    instrucción(es);}  
while (exp);
```

for

- UTILIZADA FRECUENTEMENTE EN EL RECORRIDO DE COLECCIONES
- INGRESA CUANDO LA CONDICIÓN ES EVALUADA COMO VERDADERO
- SINTAXIS

```
for(exp1; exp2; exp3){  
    instrucción(es);} 
```

DONDE

EXP 1 = INICIALIZACIÓN, SOLO SE EJECUTA UNA VEZ AL INICIO DEL CICLO.

EXP2 = CONDICIÓN

EXP3 = INCREMENTO/DECREMENTO

NOTA

- ES IMPORTANTE INDICAR QUE LA ESTRUCTURA `for` ES FLEXIBLE POR LO TANTO PUEDE
 - CONTENER MAS DE UNA VARIABLE PARA CONTROLAR EL CICLO. (REQUIERE COMAS PARA ENCADENAR EXPRESIONES)
 - OMITIR ALGUNA O TODAS LAS EXPRESIONES DENTRO DEL CICLO (PARA EVITAR EL CICLO INFINITO SERÁ NECESARIO UTILIZAR UN `BREAK`, O INCLUIR LA VARIABLE DE CONTROL DENTRO DEL CICLO)

FOR SIN INSTRUCCIONES?

- UN CICLO `FOR` SIN INSTRUCCIONES NO NECESARIAMENTE ES UN CICLO INFINITO, EN MUCHAS OCASIONES ESTOS FUNCIONAN COMO RETARDOS.

```
for(t=0;t<tiempo;t++);
```