

Conceptos de Algoritmos Datos y Programas



CADP – TEMAS

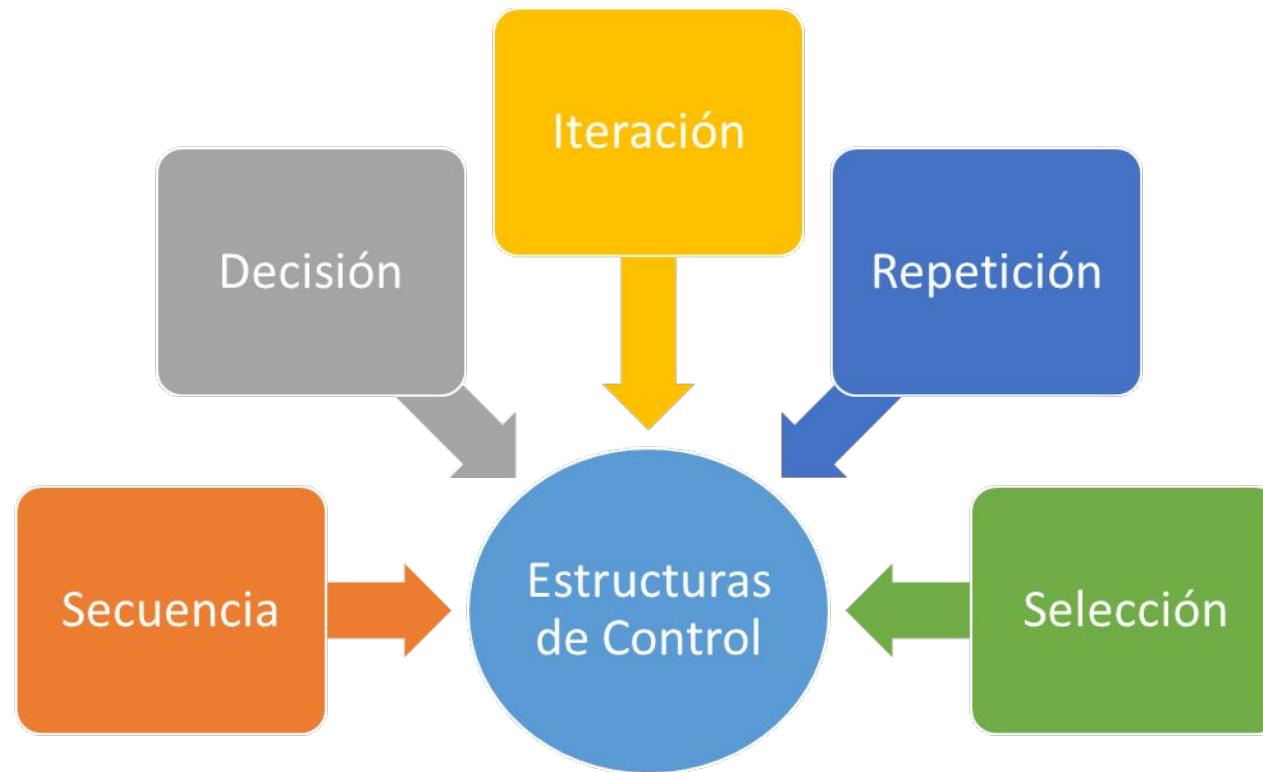
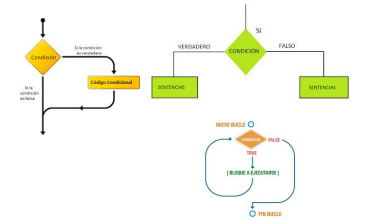


- Estructura de control
- Estructura de iteración
- Estructura de control WHILE y REPEAT UNTIL

CADP – ESTRUCTURAS DE CONTROL



Todos los lenguajes de programación tienen un conjunto mínimo de instrucciones que permiten especificar el control del algoritmo que se quiere implementar. Como mínimo deben contener: secuencia, decisión e iteración.



CADP – ESTRUCTURAS DE CONTROL



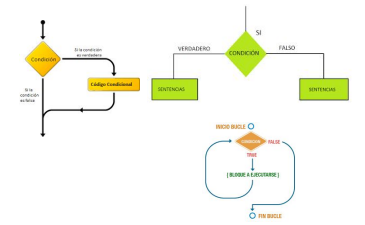
ITERACION

Puede ocurrir que se desee ejecutar un bloque de instrucciones desconociendo el número exacto de veces que se ejecutan.

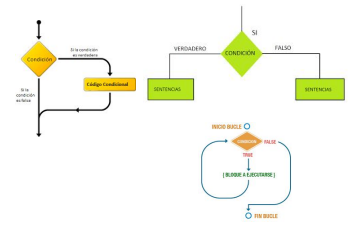
Para estos casos existen en la mayoría de los lenguajes de programación estructurada las estructuras de control iterativas condicionales.

Como su nombre lo indica las acciones se ejecutan dependiendo de la evaluación de la condición.

Estas estructuras se clasifican en **pre-condicionales y post-condicionales**.



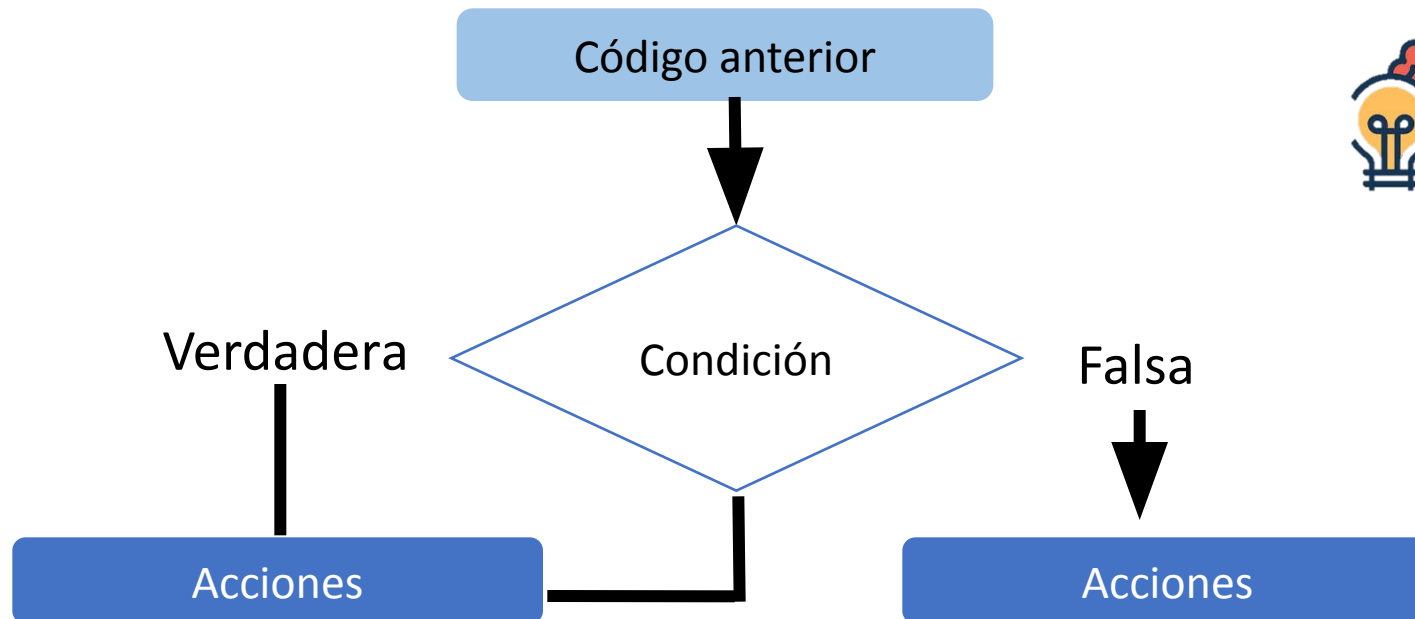
CADP – ESTRUCTURAS DE CONTROL



ITERACION - PRECONDICIONAL

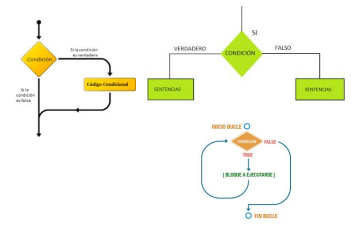
Evalúan la condición y si es verdadera se ejecuta el bloque de acciones. Dicho bloque se pueda ejecutar 0, 1 ó más veces.

Importante: el valor inicial de la condición debe ser conocido o evaluable antes de la evaluación de la condición.



A qué estructura de control vista en el entorno del robot se parece?.

Cómo es la sintaxis?



ITERACION - PRECONDICIONAL

```
while (condición) do  
    accion;
```

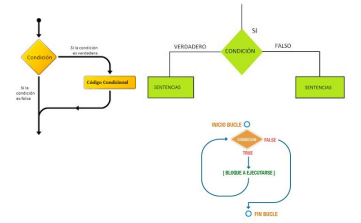


más de una acción

```
while(condición) do  
    begin  
        acción 1;  
        acción 2;  
    end;
```

CADP – ESTRUCTURAS DE CONTROL

ITERACION



Realizar un programa que lea códigos de productos hasta leer un código igual a 30. Al finalizar informe la cantidad de productos con código par.

- Cómo leo un código
- Cómo veo si es par
- Cuál es la condición de fin
- Cómo muestro el resultado

35
70
32
5
30

informa 2

CADP – ESTRUCTURAS DE CONTROL ITERACION



```
Program uno;  
var  
    resto, prod: integer;  
    total: integer;
```

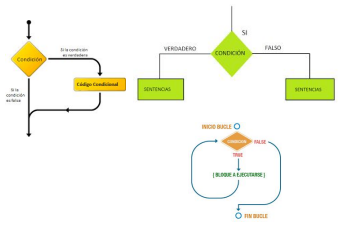
```
begin  
    total:=0;  
    while (prod <> 30)do  
        begin  
            read(prod);  
            resto:= prod MOD 2;  
            if (resto = 0)then  
                total:= total + 1;  
            end;  
            write (total);  
        end.  
end.
```

Cuál es el error?

```
Program dos;  
var  
    prod, resto: integer;  
    total: integer;
```

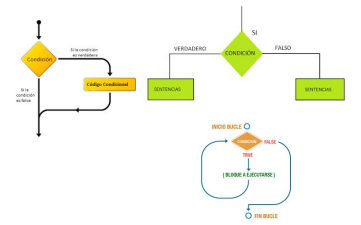
```
begin  
    total:=0;  
    read (prod);  
    while (prod <> 30)do  
        begin  
            resto:= prod MOD 2;  
            if (resto = 0)then  
                total:= total + 1;  
            read (prod);  
        end;  
        write (total);  
    end.
```

Otra alternativa?



CADP – ESTRUCTURAS DE CONTROL

ITERACION

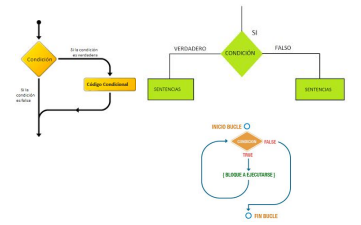


```
Program dos;  
var  
    prod:integer;  
    total:integer;
```

```
begin  
    total:=0;  
    read (prod);  
    while (prod <> 30)do  
        begin  
            if (prod MOD 2 = 0)then  
                total:= total + 1;  
            read (prod);  
        end;  
        write (total);  
    end.
```

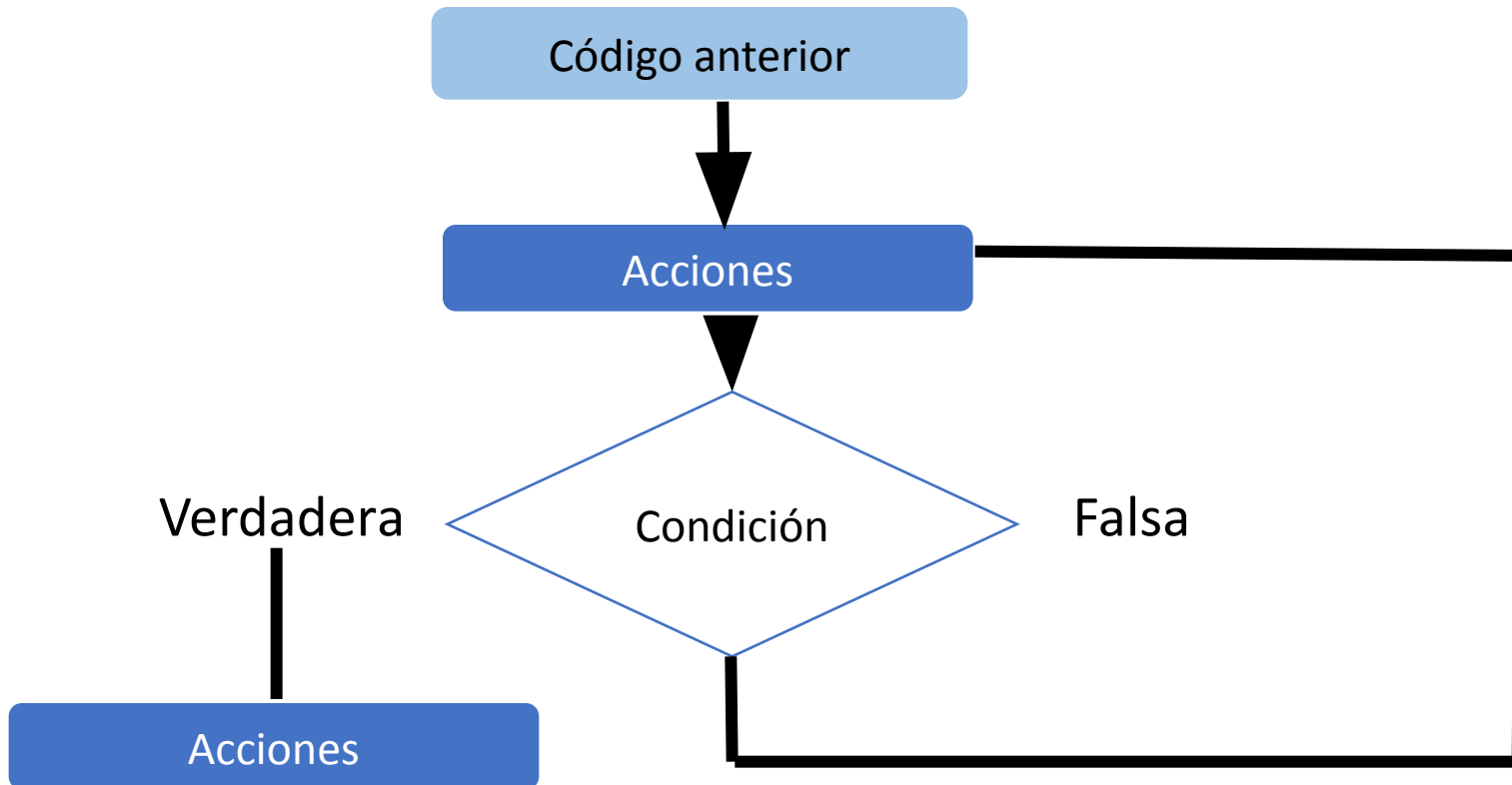
No se utiliza
la variable
resto

CADP – ESTRUCTURAS DE CONTROL



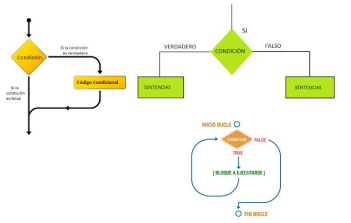
ITERACION - POSTCONDICIONAL

Ejecutan las acciones luego evalúan la condición y ejecutan las acciones mientras la condición es falsa. Dicho bloque se pueda ejecutar 1 ó más veces.



A qué estructura de control vista en el entorno del robot se parece?.

Cómo es la sintaxis?



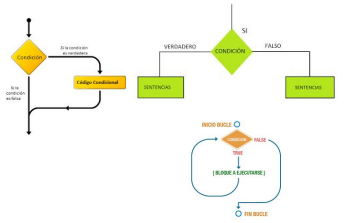
ITERACION - POSTCONDICIONAL

repeat
 accion;
until (condición);


más de una acción

repeat
 acción 1;
 acción 2;
until (condicion);

CADP – ESTRUCTURAS DE CONTROL ITERACION



Realizar un programa que lea códigos de productos hasta leer un código igual a 30. Al finalizar informe la cantidad de productos con código par. El último producto debe procesarse.

- Cómo leo un producto
- Cómo veo si es par
- Cuál es la condición de fin
- Cómo muestro el resultado

35
70
32
5
30

informa 3

CADP – ESTRUCTURAS DE CONTROL

ITERACION

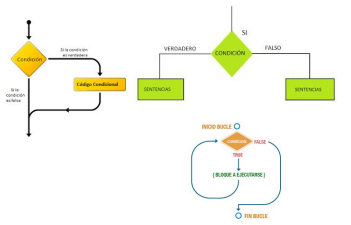


Sino existiera el
repeat until

```
Program uno;  
var  
    prod:integer;  
    total:integer;  
begin  
    total:=0;  
    read(prod);  
    while (prod <> 30)do  
        begin  
            if (prod MOD 2 = 0)then  
                total:= total + 1;  
            read(prod);  
        end;  
        if (prod MOD 2 = 0)then  
            total:= total + 1;  
        write (total);  
    end.
```

Cuál es el problema de
este tipo de soluciones?

Todo el procesamiento
sobre la variable **prod** se
debe repetir dentro y
fuera del while



CADP – ESTRUCTURAS DE CONTROL

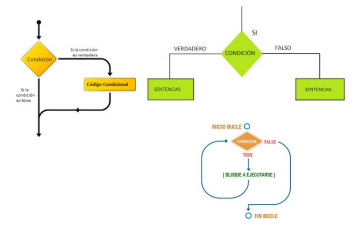
ITERACION

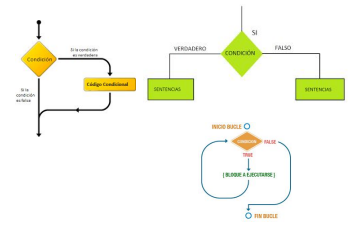


```
Program correcto;  
var  
    prod:integer;  
    total:integer;
```

```
begin  
    total:=0;  
    repeat  
        read (prod);  
        if (prod MOD 2 = 0)then  
            total:= total + 1;  
    until (prod = 30)  
    write (total);  
end.
```

Se ejecuta cuando
la condición es
falsa





PRE CONDICIONALES

Evalúa la condición y en caso de ser verdadera, ejecuta las acciones.

Se repite mientras la condición es verdadera.

Puede ejecutarse 0, 1 o más veces.



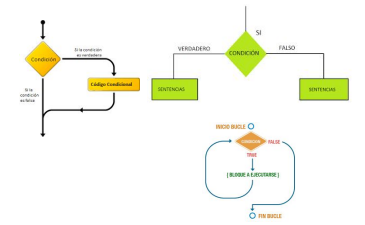
POST CONDICIONALES

Ejecuta las acciones y luego evalúa la condición.

Se repite mientras la condición es falsa.

Puede ejecutarse 1 o más veces.

CADP – ESTRUCTURAS DE CONTROL



Mirando estos enunciados que estructuras de control usarías?

- Realizar un programa que lea un número e informe si el número es par o impar
- Realizar un programa que lea un letras hasta leer la letra “@” la cual debe procesarse e informe la cantidad de letras ‘á’ leídas.
- Realizar un programa que lea un letras hasta leer la letra “@” e informe la cantidad de letras ‘á’ leídas.