

# Conceptos de Algoritmos Datos y Programas

# CADP – TEMAS

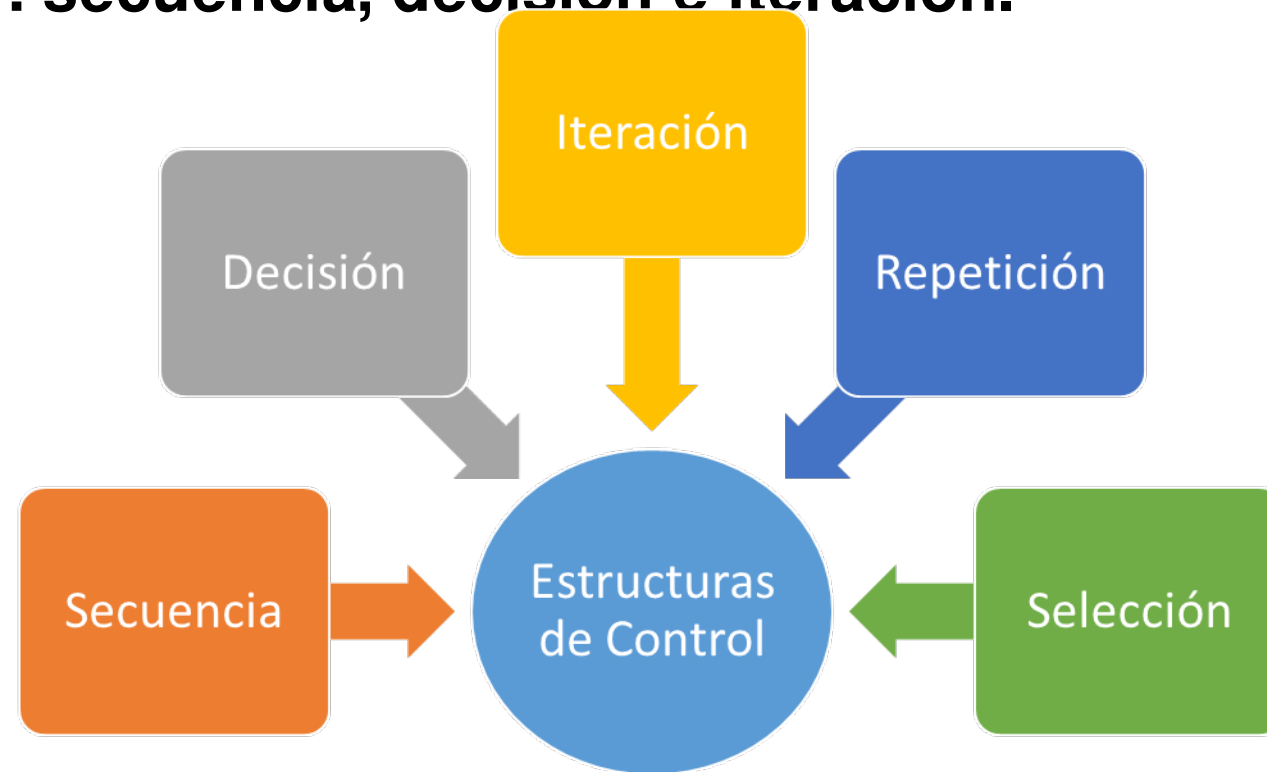
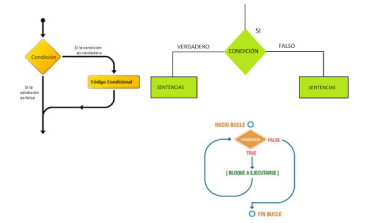


- Estructura de control
- Estructura de secuencia
- Estructura de control de decisión IF
- Estructura de control de selección  
CASE

# 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.



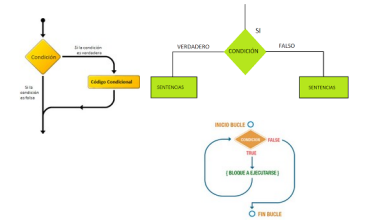
# CADP – ESTRUCTURAS DE CONTROL



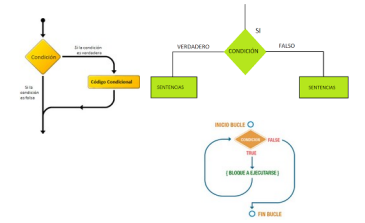
## SECUENC

**LA** estructura de control más simple, está representada por una sucesión de operaciones (por ej. asignaciones), en la que el orden de ejecución coincide con el orden físico de aparición de las instrucciones.

```
Program uno;  
  
...  
  
var  
    num:integer;  
begin  
    read (num);  
    write (num);  
end.
```

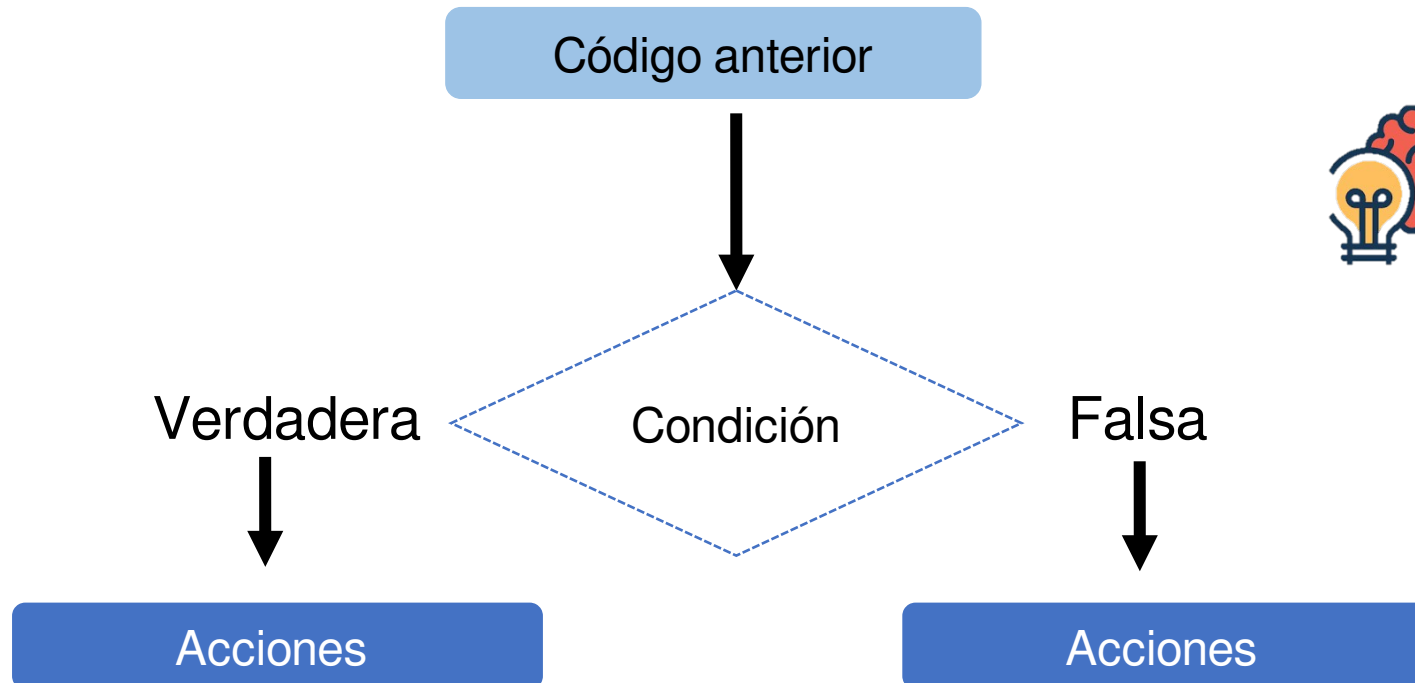


# CADP – ESTRUCTURAS DE CONTROL



## DECISION

En un algoritmo representativo de un problema real es necesario tomar decisiones en función de los datos del problema. La estructura básica de decisión entre dos alternativas es la que se representa simbólicamente:




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

*Cómo es la sintaxis?*

# CADP – ESTRUCTURAS DE CONTROL

```
if (condición) then  
    acción;
```

  
más de  
una acción

```
if (condición) then  
    acción 1  
else  
    acción 2;
```

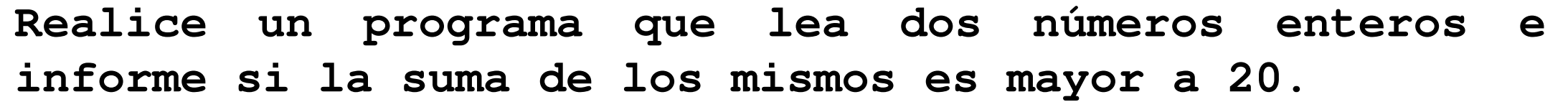
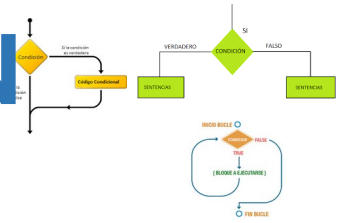
```
if      (condición)  
then  
    begin  
        acción 1;  
        acción 2;  
    end  
else  
    acción 3;
```

```
if (condición) then  
    begin  
        acción 1;  
        acción 2;  
    end;
```

```
if (condición) then  
    begin  
        acción 1;  
        acción 2;  
    end  
else  
    begin  
        acción 3;  
        acción 4;  
    end;
```



# DECISION



- Cómo leo un número
- Cómo veo si la suma es  $> 20$
- Cómo muestro el resultado

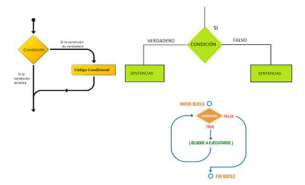
# CADP – ESTRUCTURAS DE CONTROL



Realice un programa que lea dos números enteros e informe si la suma de los mismos es mayor a 20.

```
Program uno;  
var  
    num1, num2, suma: integer;  
  
begin  
    read (num1);  
    read (num2);  
    suma := num1 + num2;  
    if (suma > 20)  
    then  
        write ("La suma supera 20")  
    else  
        write ("La suma NO supera 20");  
end.
```

## DECISION



Otra  
forma  
?

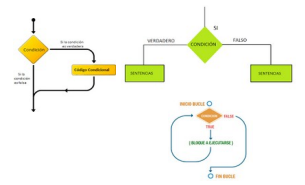


# CADP – ESTRUCTURAS DE CONTROL



Realice un programa que lea dos números enteros e informe si la suma de los mismos es mayor a 20.

## DECISION



```
Program uno;  
var  
    num1,num2:integer;  
  
begin  
    read (num1);  
    read (num2);  
    if ((num1+num2) > 20)  
    then  
        write ("La suma supera 20")  
    else  
        write ("La suma NO supera 20");  
end.
```

# CADP – ESTRUCTURAS DE CONTROL



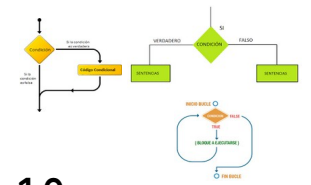
Realice un programa que lea un número (suponga  $> 0$ ) y asigne el valor 10 a una variable si el número es menor a 10; asigne 50 a la misma variable si el número es mayor a 10 pero menor que 50; y 100 si el número es mayor a 50.

- Cómo leo un número
- Cómo verifico en que rango está
- Cómo muestro el resultado

# CADP – ESTRUCTURAS DE CONTROL



## DECISION



Realice un programa que lea un número (suponga  $> 0$ ) y asigne el valor 10 a una variable si el número es menor a 10; asigne 50 a la misma variable si el número es mayor a 10 pero menor que 50; y 100 si el número es mayor a 50.

```
Program uno;
```

```
var
```

```
    num1, resultado: integer;
```

```
begin
```

```
    read (num1);
```

```
    if (num1 <= 10) then resultado:= 10;
```

```
    if (num1 > 10) and (num1 <= 50) then resultado:=  
        50;
```

```
    if (num1 > 50) then resultado:= 100;
```

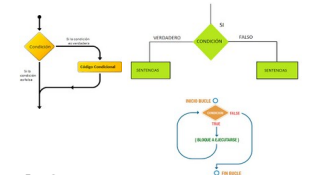
```
    write (resultado);
```

Es la  
mejor  
solución  
?

# CADP – ESTRUCTURAS DE CONTROL



## DECISION



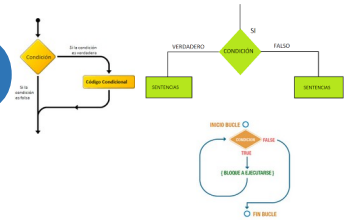
Realice un programa que lea un número (suponga  $> 0$ ) y asigne el valor 10 a una variable si el número es menor a 10; asigne 50 a la misma variable si el número es mayor a 10 pero menor que 50; y 100 si el número es mayor a 50.

```
Program uno;
var
    num1,resultado:integer;

begin
    read (num1);
    if (num1 <= 10) then resultado:= 10
    else
        if (num1 > 10) and (num1 <= 50) then resultado:=
50
        else
            resultado:= 100;
    write (resultado);
end
```

# CADP – ESTRUCTURAS DE CONTROL

## SELECCION



Realizar un programa que lea un caracter y al finalizar informe si se leyó un caracter mayúscula, minúscula, dígito, y ó especiales ha leído.



Qué tipo de datos leo?

'A'

'0'

'B'

'x'

'!'

'\$'

'='



informa

Mayúscula

Dígito

Mayúscula

Minúscula

Especial

Especial

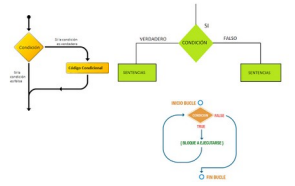
Especial



Cómo identifico que carácter es?

# CADP – ESTRUCTURAS DE CONTROL

## SELECCION



```
Program uno;  
var  
    car:char;  
  
begin  
    read (car);
```

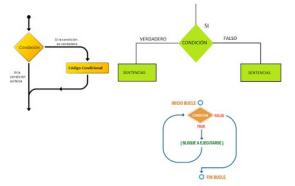
Se puede  
mejorar?

Si sólo  
existe  
el if

```
    if (car = 'a')    or (car = 'b')... or (car = 'z')  
then  
    write ("minúscula")  
    if (car = 'A')    or (car = 'B')... or (car = 'Z')  
then  
        write ("mayúscula");  
    if (car = '0')    or (car = '1')... or (car = '9')  
then  
        write ("numero");  
    if (car = '@')    or (car = '!')... or (car = '*')  
then
```

# CADP – ESTRUCTURAS DE CONTROL

## SELECCION

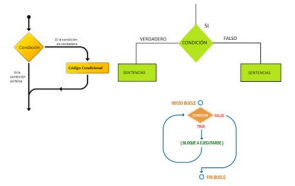


```
Program uno;  
var  
    car:char;  
begin  
    read (car);  
    if (car = 'a')    or (car = 'b')... or (car = 'z')  
then  
    write ("minúscula");  
    else  
    if (car = 'A')    or (car = 'B')... or (car = 'Z')  
then  
    write ("mayúscula");  
    else  
    if (car = '0')    or (car = '1')... or (car = '9')  
then  
    write ("digito");  
    else if (car = '@')    or (car = '!')... or (car =  
    '*' ) then
```

Si sólo  
existe  
el if

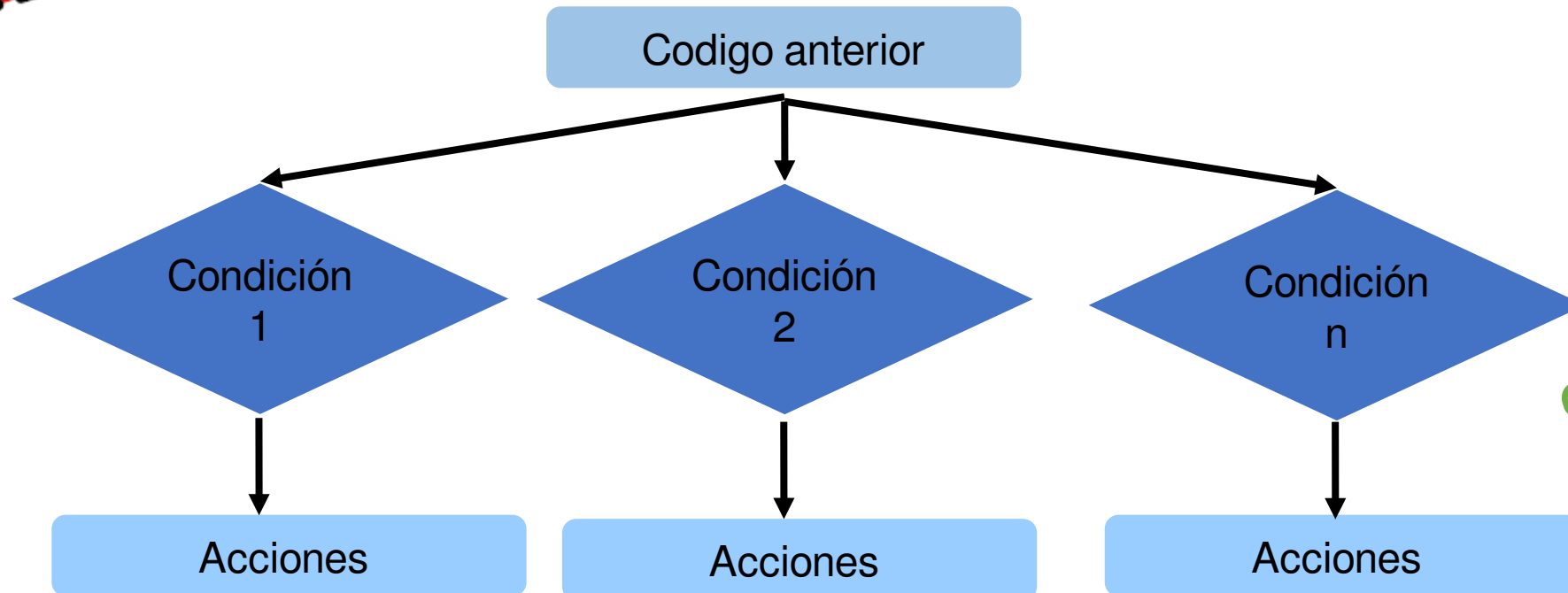
Se puede  
mejorar?

# CADP – ESTRUCTURAS DE CONTROL



## SELECCION

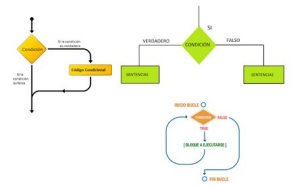
Permite realizar distintas acciones dependiendo del valor de una variable de tipo ordinal.



*Cómo es la sintaxis?*



# CADP – ESTRUCTURAS DE CONTROL SELECCION



```
case (variable) of  
  condicion1:  accion1;  
  condición 2: acción2;  
  ....  
  condición n: acción  
n;  
end;
```



más de una  
acción

```
case (variable) of  
  condicion1:  
    accion1;  
  condición 2: begin  
    acción2;  
    accion3;  
end;
```

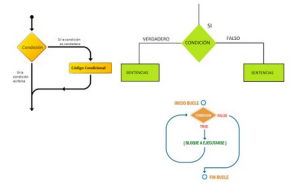
....

```
  condición n:
```

acción;

# CADP – ESTRUCTURAS DE CONTROL

# SELECCION

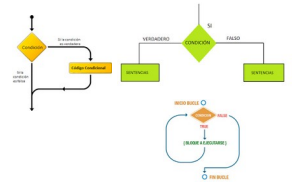


```
Program uno;  
var  
    car:char;  
begin  
    read (car);  
    case car of  
        car = 'a': write("minúscula");  
        ...  
        car = 'z': write("minúscula");  
  
        car = 'A': write("mayúscula");  
        ...  
        car = 'Z': write("mayúscula");  
  
        car = '0': write("dígito");  
        ...  
        car = '9': write("dígito");  
        else write("especial");  
    end;  
End.
```

Se puede  
mejorar?

# CADP – ESTRUCTURAS DE CONTROL

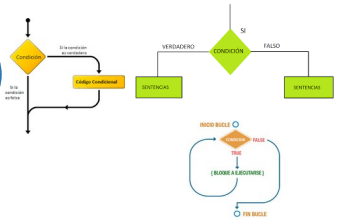
## SELECCION



```
Program uno;  
var  
    car:char;  
begin  
    read (car);  
    case car of  
        'a'.. 'z': write ("minúscula");  
        'A'.. 'Z': ("mayúscula");  
        '0'.. '9': ("dígito");  
        else ("especial");  
    end;  
End.
```

# CADP – ESTRUCTURAS DE CONTROL

## SELECCION



- La variable del case debe ser de tipo ordinal
- Las opciones deben ser disjuntas

# CADP – Estructuras de control



**Problema:** se leen valores de alturas de personas, hasta leer la altura 1.59. Informar la cantidad de personas que miden entre 1.00 y 1.30; la cantidad de personas que miden entre 1.31 y 1.50; la cantidad de personas que miden entre 1.51 y 1.89 y las que miden más de 1.89

**El alumno 1:** utiliza una variable real para leer las alturas y cuatro contadores para contar la cantidad de personas en cada rango. Además utiliza un while como estructura de control principal y adentro utiliza un case que incluye los rangos de alturas para saber cual contador sumar. Al final informa los valores de los contadores.

**El alumno 2:** utiliza una variable real para leer las alturas y cuatro contadores para contar la cantidad de personas en cada rango. Además utiliza un while como estructura de control principal y adentro utiliza un if con else para saber cual contador sumar. Al final informa los valores de los contadores.

**Ambas soluciones son correctas?**