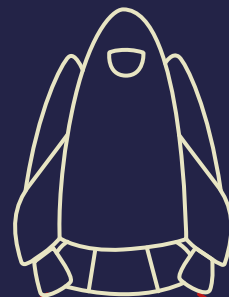


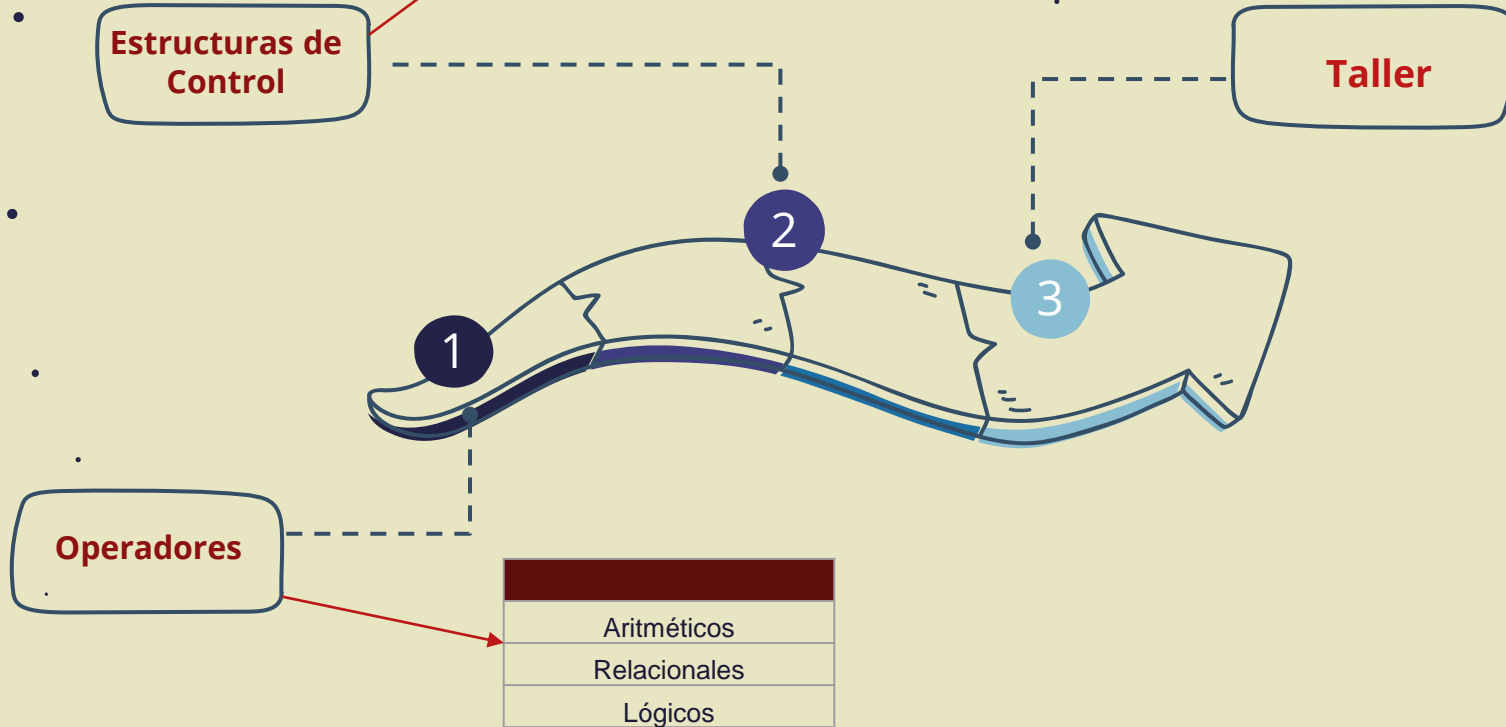
# Programa académico CAMPUS



Ciclo 1:  
Fundamentos de  
Programación



# Agenda



# Estructuras de Control

## Ciclos - Iterativas - Conceptualización



En la vida diaria se manejan con mucha frecuencia las estructuras iterativas, en donde se encuentra **procesos que se repiten una cantidad de veces**, por ejemplo:

**Ejemplo 1:** En un entrenamiento con el equipo de baloncesto, el profesor – entrenador nos invita a realizar 5 vueltas a la cancha como calentamiento. En este ejemplo, el proceso que se repite es la vuelta y se debe realizar una cantidad de veces determinada (5). En términos de programación, se le denomina la estructura iterativa controlada por cantidad – FOR.

**Ejemplo 2:** En una clase de educación física, el profesor es llamado desde la coordinación y nos invita a realizar vueltas a la cancha mientras regresa . En este ejemplo, el proceso que se repite es la vuelta y no se conoce la cantidad a realizar, depende de la condición del regreso del docente, es decir mientras el docente no regrese, debo dar vuelta. En términos de programación, se le denomina la estructura iterativa controlada por condición – WHILE.

# Estructuras de Control

## Iteración o ciclo controlado por cantidad - FOR



**Vble** = Encargada de tomar valores de acuerdo con cada ocurrencia de la repetición. Puede darse cualquier nombre a la variable (tener en cuenta las reglas para identificadores), aunque se utiliza con mucha frecuencia, los nombres de I, J, K entre otros. Las iteraciones estarán controladas por esta variable.

**Valor\_Inicial** = Constante o variable con el valor inicial del ciclo, esta asignación solo se realiza una vez, cuando se inicializa el ciclo, es la primera ocurrencia.

**Valor\_Final** = Constante o variable con el valor final del ciclo, esta determina la condición para terminar el ciclo, es la última ocurrencia.

**Incremento** = Constante o variable que representa el intervalo de aumento de la variable del ciclo, cuando el incremento es 1 se puede omitir, si es un decremento se representa con un valor negativo.

# Estructuras de Control

Iteración o ciclo controlado por cantidad



En pseudocódigo, la estructura sería:

```
PARA Vble= Valor_Inicial HASTA Valor_Final INCREMENTO INCR  
    Proceso que se Repite  
FIN PARA
```

# Estructuras de Control – range()

Iteración o ciclo controlado por cantidad



La función `range()` proporciona una secuencia de enteros basada en los argumentos de la función.

# Estructuras de Control – range()

Iteración o ciclo controlado por cantidad

```
range(parar)  
range(incio, parar[, paso])
```

El argumento **inicio** es el primer valor del rango. Si se llama a range() con un solo argumento, Python asume **inicio = 0**.

El argumento **parar** es el límite superior del rango. Es importante tener en cuenta que este valor superior **no está incluido** en el rango.

El valor de **paso opcional** controla el incremento entre los valores del rango. De forma predeterminada, **paso = 1**.

# Estructuras de Control – range()

Iteración o ciclo controlado por cantidad



```
# Ejemplo con un argumento
for i in range(5):
    print(i, end=" ", " ") # prints: 0, 1, 2, 3, 4,
```



# Estructuras de Control – range()

Iteración o ciclo controlado por cantidad

```
# Ejemplo con dos argumentos
for i in range(-1, 5):
    print(i, end=", ") # prints: -1, 0, 1, 2, 3, 4,
```

# Estructuras de Control – range()

Iteración o ciclo controlado por cantidad



# Ejemplo con tres argumentos

```
for i in range(-1, 5, 2):
```

```
    print(i, end=" ", " ") # prints: -1, 1, 3,
```

# Estructuras de Control



## Iteración o ciclo controlado por cantidad

Se tiene la siguiente información sobre los N (N es suministrado) usuarios del servicio de AGUA:

- Código
- Nombre
- Estado: Puede ser V=Vigente o S=Suspendido
- Estrato: Puede ser 1,2,3,4 5 o 6
- Consumo del mes (en cm<sup>3</sup>)

Se pide calcular el valor a pagar por concepto de servicio de AGUA de cada usuario, teniendo en cuenta que este valor es la suma del valor de tarifa más el valor del consumo. También nos indican que el valor de la tarifa básica depende del estrato así:

(ESTRATO - TARIFA BÁSICA): ( 1 - \$10.000, 2 - \$20.000, 3 - \$30.000, 4 - \$45.000, 5 - \$60.000, 6 - \$70.000)

Además el valor del consumo es el consumo del mes por el valor de 1 cm<sup>3</sup> que de \$200

Se debe imprimir el nombre del usuario, el valor de la tarifa básica, el valor del consumo y el valor a pagar por concepto del servicio de AGUA

NOTA: Se liquida servicio de AGUA a los usuarios con estado V (Vigente)



# Estructuras de Control

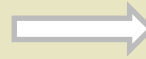
Iteración o ciclo controlado por cantidad

Metodología -> Pensamiento lógico estructurado

Análisis



Diseño



Construcción



Método  
Entrada – Proceso - Salida



Algoritmo  
Diagrama de Flujo



Programa

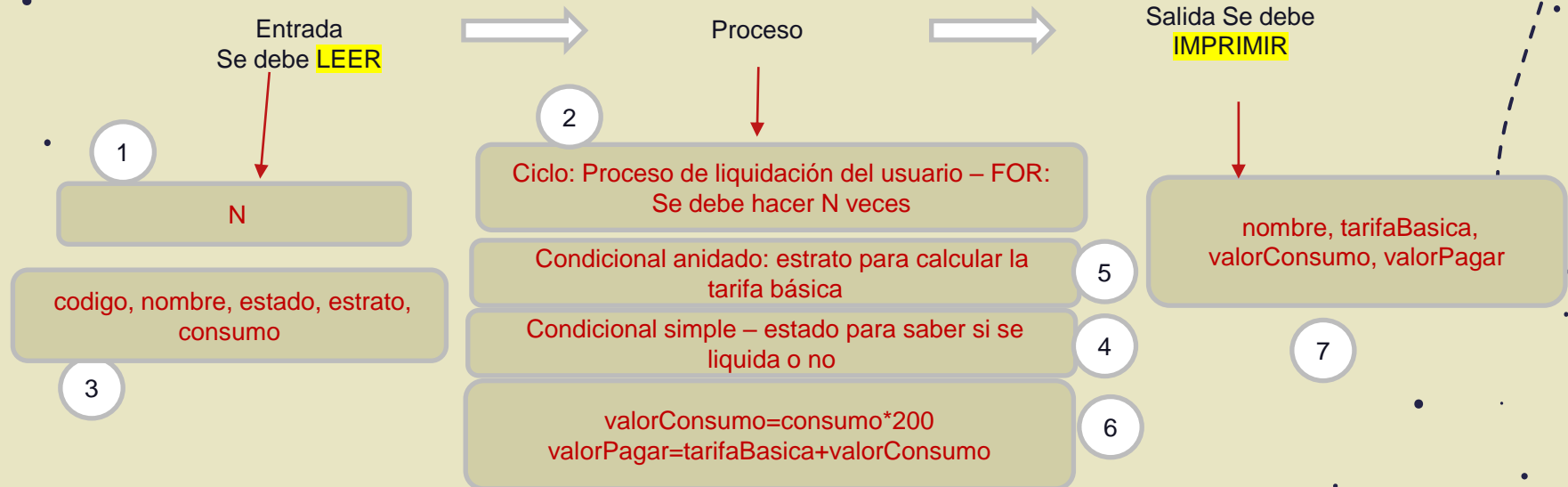


# Estructuras de Control

Iteración o ciclo controlado por cantidad

Análisis → Método Entrada-Proceso-Salida

Operaciones, cálculos, **estructuras de control**



# Estructuras de Control

Iteración o ciclo controlado por cantidad

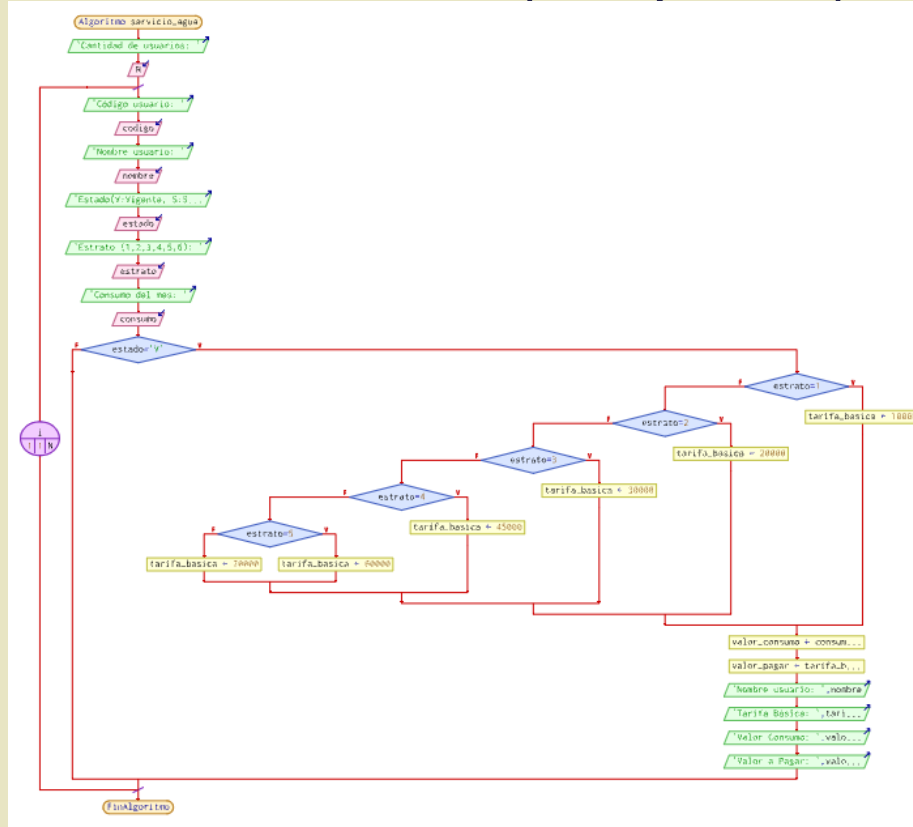
Diseño → Algoritmo

```
Algoritmo servicio_agua
Escribir "Cantidad de usuarios: "
Leer N
Para i=1 Hasta N Con Paso 1 Hacer
    Escribir "Código usuario: "
    Leer codigo
    Escribir "Nombre usuario: "
    Leer nombre
    Escribir "Estado(V:Vigente, S:Suspendido): "
    Leer estado
    Escribir "Estrato (1,2,3,4,5,6): "
    Leer estrato
    Escribir "Consumo del mes: "
    Leer consumo
    Si estado="V" Entonces
        Si estrato=1 Entonces
            tarifa_basica=10000
        SiNo
            Si estrato=2 Entonces
                tarifa_basica=20000
            SiNo
                Si estrato=3 Entonces
                    tarifa_basica=30000
                SiNo
                    Si estrato=4 Entonces
                        tarifa_basica=45000
                    SiNo
                        Si estrato=5 Entonces
                            tarifa_basica=60000
                        SiNo
                            tarifa_basica=70000
                    Fin Si
                Fin Si
            Fin Si
        Fin Si
        valor_consumo=consumo*200
        valor_pagar=tarifa_basica+valor_consumo
        Escribir "Nombre usuario: ",nombre
        Escribir "Tarifa Básica: ",tarifa_basica
        Escribir "Valor Consumo: ",valor_consumo
        Escribir "Valor a Pagar: ",valor_pagar
    Fin Si
Fin Para
FinAlgoritmo
```

# Estructuras de Control

Iteración o ciclo controlado  
por cantidad

Diseño → Diagrama de flujo



# Estructuras de Control

Iteración o ciclo controlado por cantidad



Programación en Python

```
for i in range(3):  
    print(i)
```

```
# 0  
# 1  
# 2
```



```
for i in range(5, 8):  
    print(i, i ** 2)  
    print('end of loop')
```

```
# 5 25  
# 6 36  
# 7 49  
# fin del ciclo
```



```
for i in range(10, 0, -2):  
    print(i)
```

```
# 10  
# 8  
# 6  
# 4  
# 2
```





# Estructuras de Control

Iteración o ciclo  
controlado por cantidad

Construcción → Programa

```
# fecha: 01/02/2023
N=int(input("Cantidad de usuarios: "))
for i in range(N):
    codigo=int(input("Código: "))
    nombre=input("Nombre: ")
    estado=input("Estado(v=Vigente, S=Suspendido): ")
    estrato=int(input("Estrato(1,2,3,4,5,6): "))
    consumo=float(input("Consumo del mes: "))
    if estado=="V":
        if estrato==1:
            tarifaBasica=10000
        elif estrato==2:
            tarifaBasica=20000
        elif estrato==3:
            tarifaBasica=30000
        elif estrato==4:
            tarifaBasica=45000
        elif estrato==5:
            tarifaBasica=60000
        else:
            tarifaBasica=70000
    valorConsumo=consumo*200
    valorPagar=tarifaBasica+valorConsumo
    print("Nombre: ",nombre)
    print("Tarifa Básica: ", "{:,.2f}".format(tarifaBasica))
    print("Valor Consumo: ", "{:,.2f}".format(valorConsumo))
    print("Valor a Pagar: ", "{:,.2f}".format(valorPagar))
```



# Estructuras de Control

Variables de Testigo - Control - Acumuladores



## Testigos, contadores y acumuladores

- En muchos programas se necesitan variables que indiquen si simplemente ha ocurrido algo (**testigos**), o que cuenten cuántas veces ha ocurrido algo (**contadores**), o que acumulen valores (**acumuladores**). Las situaciones pueden ser muy diversas, por lo que en este apartado simplemente se ofrecen unos ejemplos para mostrar la idea.

# Estructuras de Control

Variables de Testigo - Control - Acumuladores



## Testigo

- Se entiende por **testigo** una variable que indica simplemente *si una condición se ha cumplido o no*. Es un caso particular de contador, pero se suele hacer con variables lógicas en vez de numéricas

### Ejemplo

Lanzar un dado y e indicar cuando cae en cinco (para este caso la variable que hace de **testigo** es la variable **sacaste\_cinco**):

# Estructuras de Control

## Variables Testigos

```
import random

print("Comienzo")

sacaste_cinco = False
for i in range(3):
    dado = random.randrange(1, 7)
    print(f"Tirada {i + 1}: {dado}")
    if dado == 5:
        sacaste_cinco = True
if sacaste_cinco:
    print("Ha salido al menos un 5.")
else:
    print("No ha salido ningún 5.")
print("Final")
```

Comienzo

Tirada 1: 3

Tirada 2: 5

Tirada 3: 1

Ha salido al menos un 5.

Final

## Detalles importantes

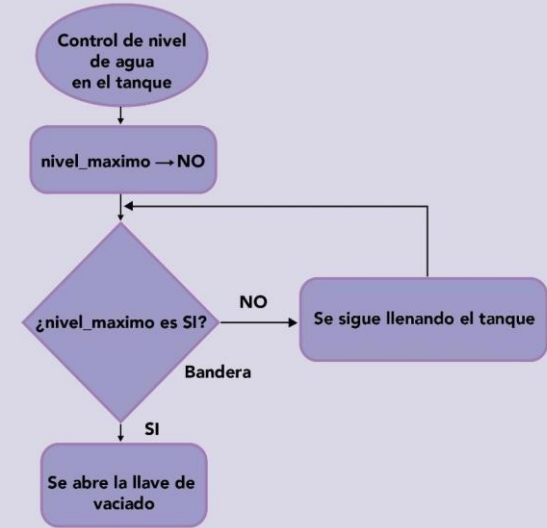
- Valor inicial al testigo (en este caso, False)
- En cada iteración, el programa comprueba
- El testigo se modifica cuando  $i=5$
- El testigo no cambia a False una vez ha cambiado True.
- Se podría haber utilizado un `range(1, 4)` para escribir la salida `{i}` en vez de `{i + 1}`.

# Estructuras de Control

## Variables de Control - Banderas

Las variables **testigo** también se les conoce como variable **Banderas** hacen referencia a variables que toman un valor, preferiblemente **binario**, booleano e indican un estado; su valor y el cambio del mismo, definen el estado en el que se encuentra el programa.

Por ejemplo, se requiere llevar el **control del nivel de agua de un tanque**.



# Estructuras de Control

## Variables de Control - Acumuladores



### ACUMULADORES

Son variables de memoria que se incrementan durante la ejecución de un proceso. Este incremento puede ser de dos tipos:

- A través de un valor fijo o constante, normalmente 1, y se denomina **CONTADOR**. Su incremento sería por ejemplo  $CONT = CONT + 1$ , en donde la variable **CONT**, incrementa en 1 el valor que tiene almacenado.
- A través de otra variable, con lo cual recibe el nombre de **SUMADOR**. Por ejemplo, el total de netos de nómina es la suma de los netos de cada empleado. Sería así:  
 $TOTAL\_NETOS = TOTAL\_NETOS + NETO\_EMPLEADO$ , en donde el **TOTAL\_NETOS** se incrementa con el valor de la variable **NETO\_EMPLEADO**, que es un valor variable.

# Estructuras de Control

## Variables de Control - Acumuladores



Estas variables ACUMULADORAS siempre se definen sobre procesos que manejen estructuras iterativas, como el caso del FOR, WHILE. Además, durante la ejecución de un proceso manejan tres (3) actividades importantes, que son:

1. Se les debe asignar un valor inicial, normalmente cero (0), pero ello depende del problema a resolver
2. Se les debe incrementar, con el valor fijo, en caso del CONTADOR, o con un valor variable en el caso del SUMADOR.
3. En alguna parte del proceso se conoce su valor final.

Si esquematizamos las estructuras de repetición, mencionadas anteriormente, encontramos tres zonas básicas, que son: Antes, dentro y después del ciclo.

Estas zonas, combinadas con las actividades que maneja un ACUMULADOR, nos permiten concluir:

1. Antes del ciclo, se debe dar valores iniciales a los acumuladores
2. Dentro del ciclo, se debe incrementar los acumuladores.
3. Después del ciclo, se conoce el valor final de un acumulador.

# Estructuras de Control



## Iteración o ciclo controlado por cantidad

Se tiene la siguiente información sobre los N (N es suministrado) usuarios del servicio de AGUA:

- Código
- Nombre
- Estado: Puede ser V=Vigente o S=Suspendido
- Estrato: Puede ser 1,2,3,4 5 o 6
- Consumo del mes (en cm3)

Se pide calcular el valor a pagar por concepto de servicio de AGUA de cada usuario, teniendo en cuenta que este valor es la suma del valor de tarifa más el valor del consumo. También nos indican que el valor de la tarifa básica depende del estrato así:

(ESTRATO - TARIFA BÁSICA): ( 1 - \$10.000, 2 - \$20.000, 3 - \$30.000, 4 - \$45.000, 5 - \$60.000, 6 -\$70.000)

Además el valor del consumo es el consumo del mes por el valor de 1 cm3 que de \$200

Se debe imprimir el nombre del usuario, el valor de la tarifa básica, el valor del consumo y el valor a pagar por concepto del servicio de AGUA

**NUEVO:** Además se pide calcular e imprimir el valor total del servicio de agua (TODOS LOS USUARIOS)

**NOTA:** Se liquida servicio de AGUA a los usuarios con estado V (Vigente)





# Estructuras de Control

Iteración o ciclo controlado por cantidad



Análisis → Método Entrada-Proceso-Salida

Operaciones, cálculos, **estructuras de control**



# Estructuras de Control

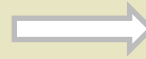
Iteración o ciclo controlado por cantidad

Metodología -> Pensamiento lógico estructurado

Análisis



Diseño



Construcción



Método  
Entrada – Proceso - Salida



Algoritmo  
Diagrama de Flujo



Programa



# Estructuras de Control

Iteración o ciclo controlado por cantidad

Diseño → Algoritmo

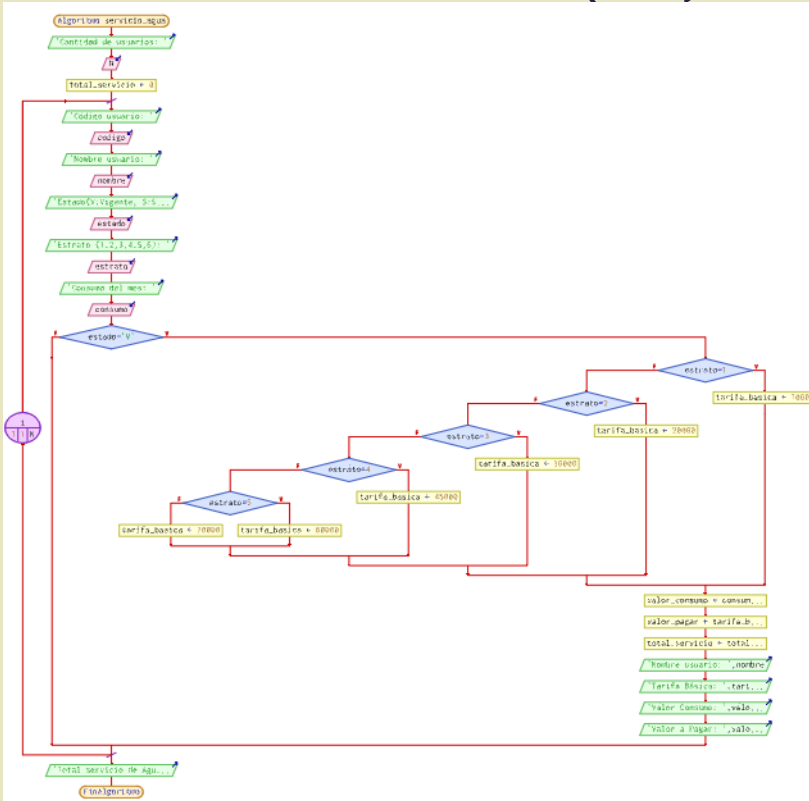


```
Algoritmo servicio_agua
  Escribir "Cantidad de usuarios: "
  Leer N
  total_servicio=0
  Para i=1 Hasta N Con Paso 1 Hacer
    Escribir "Código usuario: "
    Leer codigo
    Escribir "Nombre usuario: "
    Leer nombre
    Escribir "Estado(V:Vigente, S:Suspendido): "
    Leer estado
    Escribir "Estrato (1,2,3,4,5,6): "
    Leer estrato
    Escribir "Consumo del mes: "
    Leer consumo
    Si estado="V" Entonces
      Si estrato=1 Entonces
        tarifa_basica=10000
      SiNo
        Si estrato=2 Entonces
          tarifa_basica=20000
        SiNo
          Si estrato=3 Entonces
            tarifa_basica=30000
          SiNo
            Si estrato=4 Entonces
              tarifa_basica=45000
            SiNo
              Si estrato=5 Entonces
                tarifa_basica=60000
              SiNo
                tarifa_basica=70000
            Fin Si
          Fin Si
        Fin Si
      Fin Si
    Fin Si
    valor_consumo=consumo*200
    valor_pagar=tarifa_basica+valor_consumo
    total_servicio=total_servicio+valor_pagar
    Escribir "Nombre usuario: ",nombre
    Escribir "Tarifa Básica: ",tarifa_basica
    Escribir "Valor Consumo: ",valor_consumo
    Escribir "Valor a Pagar: ",valor_pagar
  Fin Si
Fin Para
Escribir "Total servicio de Agua: ",total_servicio
FinAlgoritmo
```

# Estructuras de Control

## Iteración o ciclo controlado por cantidad

Diseño -> Diagrama de flujo



# Estructuras de Control

Iteración o ciclo  
controlado por cantidad

Construcción → Programa

```
# Programa para liquidar servicio de Agua de N usuarios
# Autor: Sergio Medina
# Fecha: 02/06/2022

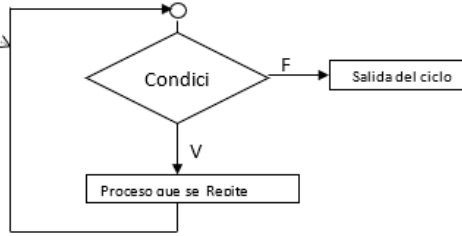
N=int(input("Cantidad de usuarios: "))
total_servicio=0
for i in range(N):
    codigo=int(input("Código Usuario: "))
    nombre=input("Nombre usuario: ")
    estado=input("Estado (V=Vigente,S=Suspendido): ")
    estrato=int(input("Estrato (1,2,3,4,5,6): "))
    consumo=float(input("Consumo del mes: "))
    if estado=="V":
        if estrato==1:
            tarifa_basica=10000
        elif estrato==2:
            tarifa_basica=20000
        elif estrato==3:
            tarifa_basica=30000
        elif estrato==4:
            tarifa_basica=45000
        elif estrato==5:
            tarifa_basica=60000
        else:
            tarifa_basica=70000
    valor_consumo=consumo*200
    valor_pagar=tarifa_basica+valor_consumo
    total_servicio=total_servicio+valor_pagar
    print("Nombre Usuario: ",nombre)
    print("Tarifa Básica: ", "{:,.2f}".format(tarifa_basica))
    print("Valor Consumo: ", "{:,.2f}".format(valor_consumo))
    print("Valor a Pagar: ", "{:,.2f}".format(valor_pagar))
print("Total servicio Agua: ", "{:,.2f}".format(total_servicio))
```

# Estructuras de Control

## Estructura iterativa o ciclo controlados por una condición WHILE

Situaciones o problemas de manejo de información que incluyen iteraciones o ciclos que se repiten dependiendo de una condición

El proceso se repite hasta cuando no se cumpla la condición



Una característica importante de esta estructura es que está conformada de dos elementos fundamentales:

- Una Condición
- Un Proceso que se repite

Y además se tiene como norma que PRIMERO se realiza la condición y SI SE CUMPLE entonces se realiza como SEGUNDA medida el proceso. Por ello esta estructura se debe utilizar cuando se tengan las siguientes características:

1. Debe existir un proceso que se repita (CICLO).
2. La repetición del proceso debe depender de una condición.

# Estructuras de Control

Estructura iterativa o ciclo controlados por una condición WHILE



En pseudocódigo:

**MIENTRAS** Condición **HAGA**

Proceso que se repite

**FIN MIENTRAS**

# Estructuras de Control

Estructura iterativa o ciclo controlados por una condición WHILE



## Guía del WHILE

### GUIA del WHILE (Iteración manejada con condición)

En la estructura de control WHILE se presenta una DOBLE lectura de información, por las siguientes razones:

- La primera lectura se encuentra ANTES del ciclo, debido a que, al ingreso al ciclo WHILE, se encuentra una condición que necesita tener un valor en la variable de control bandera.
- La segunda lectura se encuentra DENTRO del ciclo, al final del proceso que se repite, debido a que es necesario, una vez procesada una ocurrencia de la repetición, pasar a la siguiente información almacenada en la variable bandera, es decir, pasar a leer el siguiente elemento. Si no se realiza esta lectura del siguiente elemento se podría incurrir en un ciclo SIN FIN, por no tener variación en la variable de control necesaria para el ingreso o salida del ciclo



# Estructuras de Control

Estructura iterativa o ciclo controlados por una condición WHILE



Dada unos números enteros, que se ingresan uno a uno se pide calcular e imprimir:

- Cuáles y cuántos números son pares
- Cuáles y cuántos números son impares

El ingreso de números se termina cuando el número ingresado es -1 (**Bandera o testigo**)

# Estructuras de Control

Iteración o ciclo controlado por cantidad

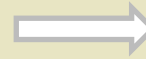


Metodología -> Pensamiento lógico estructurado

Análisis



Diseño



Construcción



Método  
Entrada – Proceso - Salida



Algoritmo  
Diagrama de Flujo



Programa

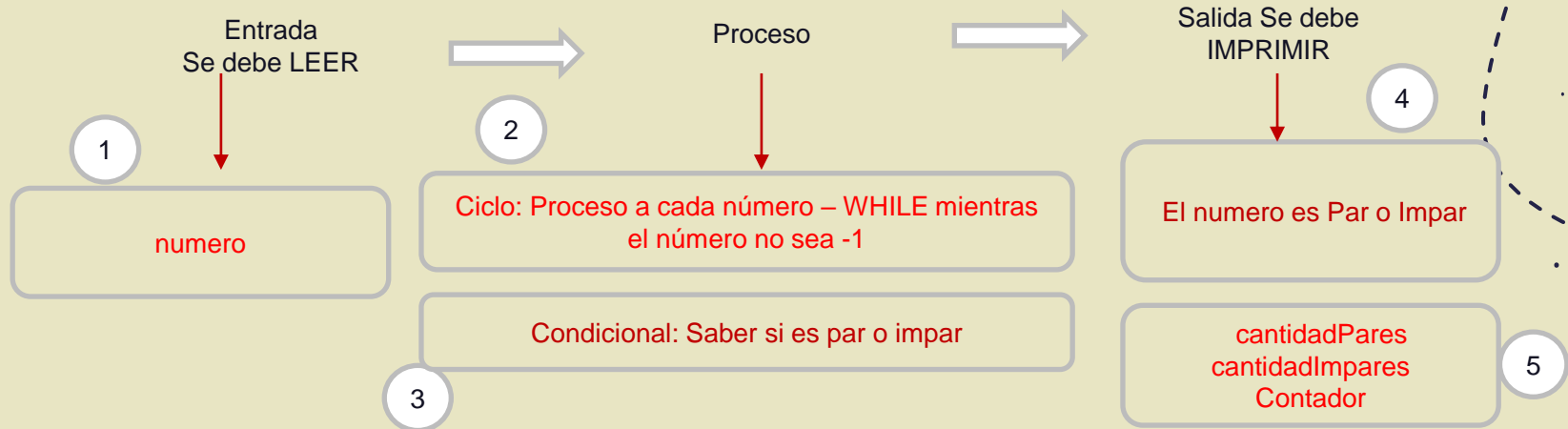


# Estructuras de Control

Estructura iterativa o ciclo controlados por una condición WHILE

Análisis -> Método Entrada-Proceso-Salida

Operaciones, cálculos, **estructuras de control**



# Estructuras de Control

## Estructura iterativa o ciclo controlados por una condición WHILE

```
Algoritmo pares_impares
    can_pares=0
    can_impares=0
    Escribir "Número entero: "
    Leer num
    Mientras num≠-1 Hacer
        Si num%2=0 Entonces
            Escribir num," es PAR"
            can_pares=can_pares+1
        SiNo
            Escribir num," es IMPAR"
            can_impares=can_impares+1
        Fin Si
        Escribir "Número entero: "
        Leer num
    Fin Mientras
    Escribir "Cantidad de pares: ",can_pares
    Escribir "Cantidad de impares: ",can_impares
FinAlgoritmo
```

Lectura del primero, para ingresar al ciclo while

Diseño → Algoritmo

Lectura del siguiente número para que no se quede en un ciclo sin fin.

# Estructuras de Control

Estructura iterativa o ciclo controlados por una condición WHILE

Construcción → Programa



```
11 canPares = 0
12 canImpares = 0
13
14 # Se lee el primer numero
15 num = int(input("Numero entero: "))
16 while num != -1:
17     # Si es diferente a -1 HAGA
18     if num % 2 == 0:
19         print("→ El numero es PAR.")
20         canPares += 1
21     else:
22         print("→ El numero es IMPAR.")
23         canImpares = +1
24
25     # Leer el siguiente numero
26     num = int(input("\nNumero entero: "))
27
28 print("\n", "-" * 30)
29 print("Cantidad de numero PARES:", canPares)
30 print("Cantidad de numero IMPARES:", canImpares)
31 print("-", " * 30, "\n")
32
```

# Talleres

# Estructuras de Control

## Ejercicio



Dado el nombre y estrato (1,2,3,4,5) de un usuario del servicio de energía eléctrica, calcular lo que pagaría de tarifa básica del servicio de energía eléctrica, que depende del estrato, así



Estrato	Tarifa Básica
1	\$10.000
2	\$15.000
3	\$30.000
4	\$50.000
5	\$65.000

Se pide visualizar el nombre y tarifa básica

Se pregunta si desea continuar (S/N). Termina cuando no desee continuar.

# Estructuras de Control

## Ejercicio



### Situación problema Liquidación servicio de matrícula

Se tiene una la información sobre 1 estudiante de una institución de educación para el trabajo, que realizará su proceso de matrícula financiera. La información que se conoce del estudiante es la siguiente:

- Código
- Nombre
- Programa a académico al cual pertenece, que puede ser
  - 1: Técnico en Sistemas
  - 2: Técnico en Desarrollo de videojuegos
  - 3: Técnico en Animación Digital
- Indicador de Beca, puede ser:
  - 1: Beca por rendimiento académico. Descuento del 50% sobre el valor matrícula.
  - 2: Beca Cultural – Deportes. Descuento del 40% sobre el valor matrícula
  - 3: Sin Beca.

También nos suministran el cuadro de valores de matrícula que depende del programa académico que cursa el estudiante, así:

Programa académico	Valor Matrícula
Técnico en Sistemas	\$800.000
Técnico en Desarrollo de videojuegos	\$1.000.000
Técnico en Animación Digital	\$1.200.000

Se pide calcular el valor neto a pagar de matrícula para el estudiante e imprimir el nombre y el valor a pagar por matrícula y valor total matriculas