universidad cenfotec_
La U de la informática

Principios de Programación.

Ejemplos de estructuras iterativas con bandera.

Ejemplos resueltos de estructuras iterativas con bandera

universidad cenfotec
La U de la informática

A continuación, se presentan algunos de los ejemplos de estructuras iterativas con bandera.

Primer ejemplo de estructuras iterativas con bandera

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Problema:

Se quiere hacer un programa que calcule la suma de varios números enteros, hasta que la suma total sea mayor o igual a 1000.

Diagrama general:

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Tabla de variables de entrada, intermedia y salida

Variables del programa			
Descripción	**Notación**		**Ejemplo**
	Nombre	**Tipo**	
Entrada			
Número entero de entrada	numero	entero	500
Intermedia			
---	---	---	---
Salida			
Valor de la suma de los números	suma_total	entero	1200

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Qué acciones se repiten? (Cuerpo del ciclo)

Se requiere calcular la suma de un conjunto de números dados por el usuario, este número será almacenado en una variable llamada *numero*.

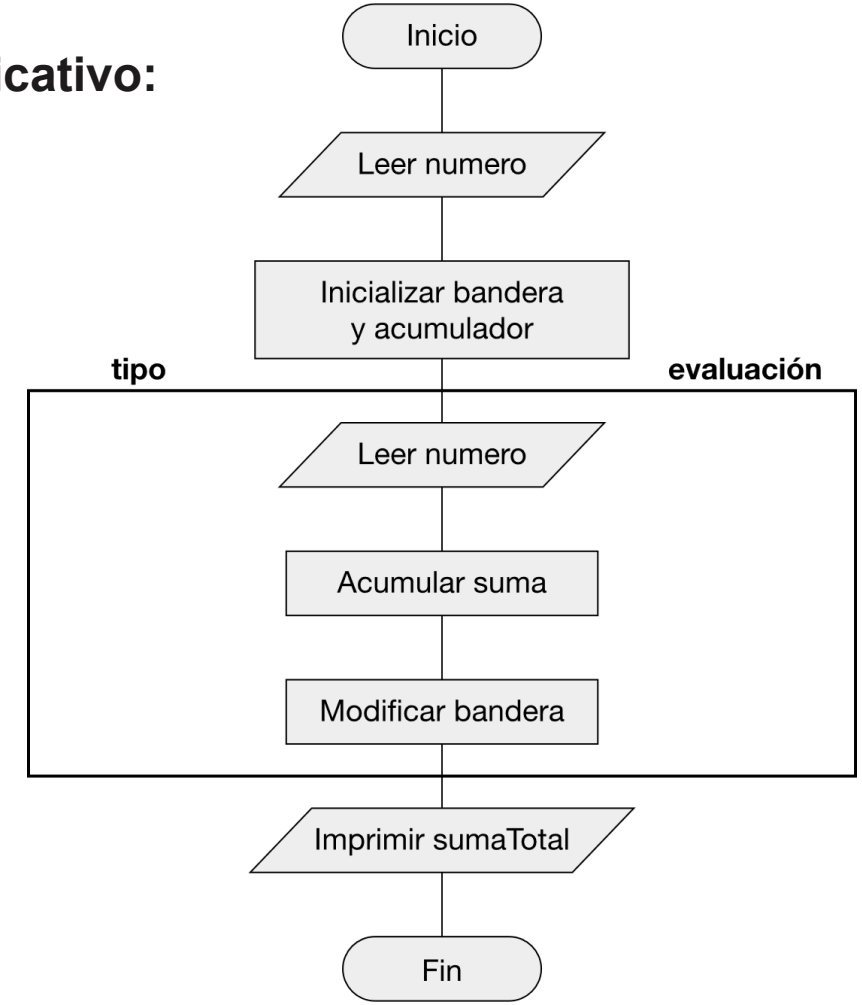
Para calcular la suma, se debe solicitar un valor para la variable *numero* y **acumular** este valor en la variable *suma_total*. Este proceso se debe repetir hasta que la variable *suma_total* sea menor a 1000.

```
Leer numero
suma_total = suma_total + numero
```

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama explicativo:



Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Cuántas veces se repetirá el ciclo? (evaluación)

El ciclo se repetirá mientras que la variable acumuladora suma_total, sea menor a 1000. Debido a que no se conoce en cuantas ejecuciones se cumplirá esta condición, se utilizará un ciclo while.

suma_total < 1000

¿Cómo se modificará la bandera? (modificación)

Al acumular el valor ingresado por teclado en cada iteración de la variable numero en la variable suma_total.

suma_total = suma_total + numero

¿Con qué valor se inicializa la bandera? (inicialización)

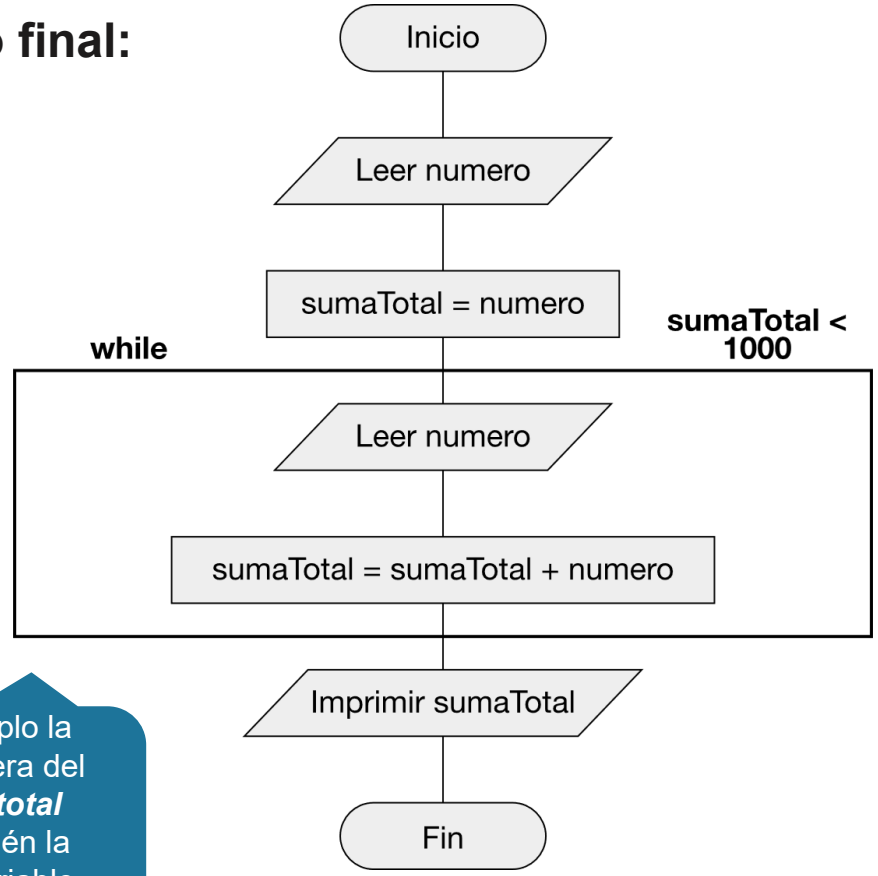
Con el primer valor dado a la variable numero, ingresado por el usuario por teclado.

suma_total = numero

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama de flujo final:



En este ejemplo la variable bandera del ciclo ***suma_total*** cumple también la función de variable acumuladora.

Ejemplo 1:

universidad cenfotec_
La U de la informática

calcular_sumatoria.py

```
1    suma_total = 0
2    numero = 0
3
4    numero = int(input('Por favor teclee un número: '))
5    suma_total = numero
6
7    while (suma_total < 1000):
8
9       numero = int(input('Por favor ingrese un número: '))
10      suma_total += numero;
11      #Es lo mismo que suma_total = suma_total + numero
12
13   print(f'El valor total de la suma es: {suma_total}' )
```

Segundo ejemplo de estructuras iterativas con bandera

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Problema:

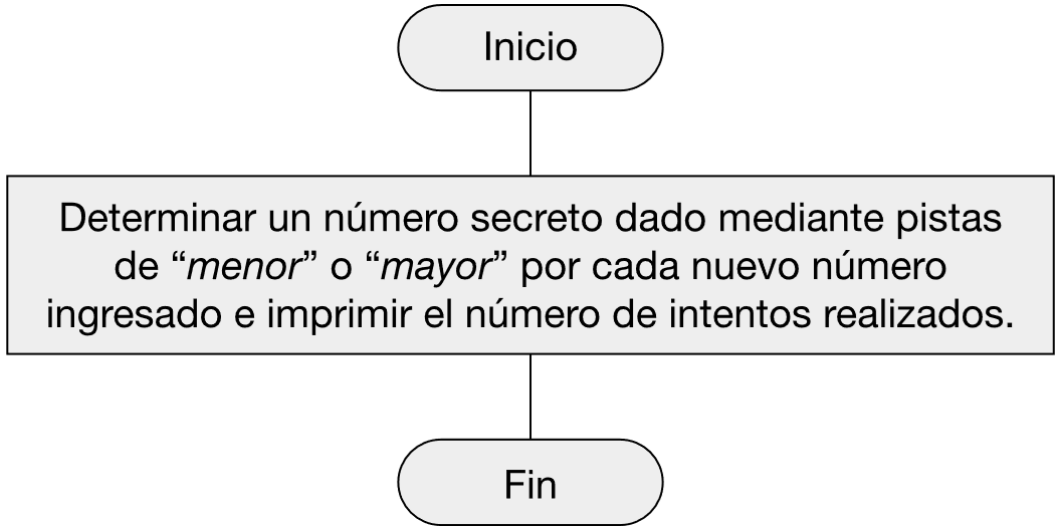
Se quiere crear un juego para adivinar un número secreto. Para ello debe pedir al usuario un número entero secreto. Luego se solicitan nuevos números, para los cuales se muestra el mensaje "*mayor*" o "*menor*" según sea mayor o menor con respecto al número secreto.

Es necesario llevar el control de la cantidad de intentos del usuario. El programa termina cuando el usuario acierta el número secreto.

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama general:



Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Tabla de variables de entrada, intermedia y salida

Variables del programa			
Descripción	**Notación**		**Ejemplo**
	Nombre	**Tipo**	
Entrada			
Número entero que debe ser adivinado por el usuario.	numero_secreto	entero	14
Número entero ingresado por el usuario aproximado al número secreto.	numero	entero	21
Intermedia			
---	---	---	---
Salida			
Cantidad de intentos del usuario para adivinar el número secreto.	cantidad_intentos	entero	5

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Qué acciones se repiten? (Cuerpo del ciclo)

En cada una de las iteraciones es necesario obtener un nuevo número de parte del usuario, con el cual se debe validar si es menor, mayor o igual al número secreto y se deben contar la cantidad de intentos del usuario en la variable *cantidad_intentos* de uno en uno.

Leer numero

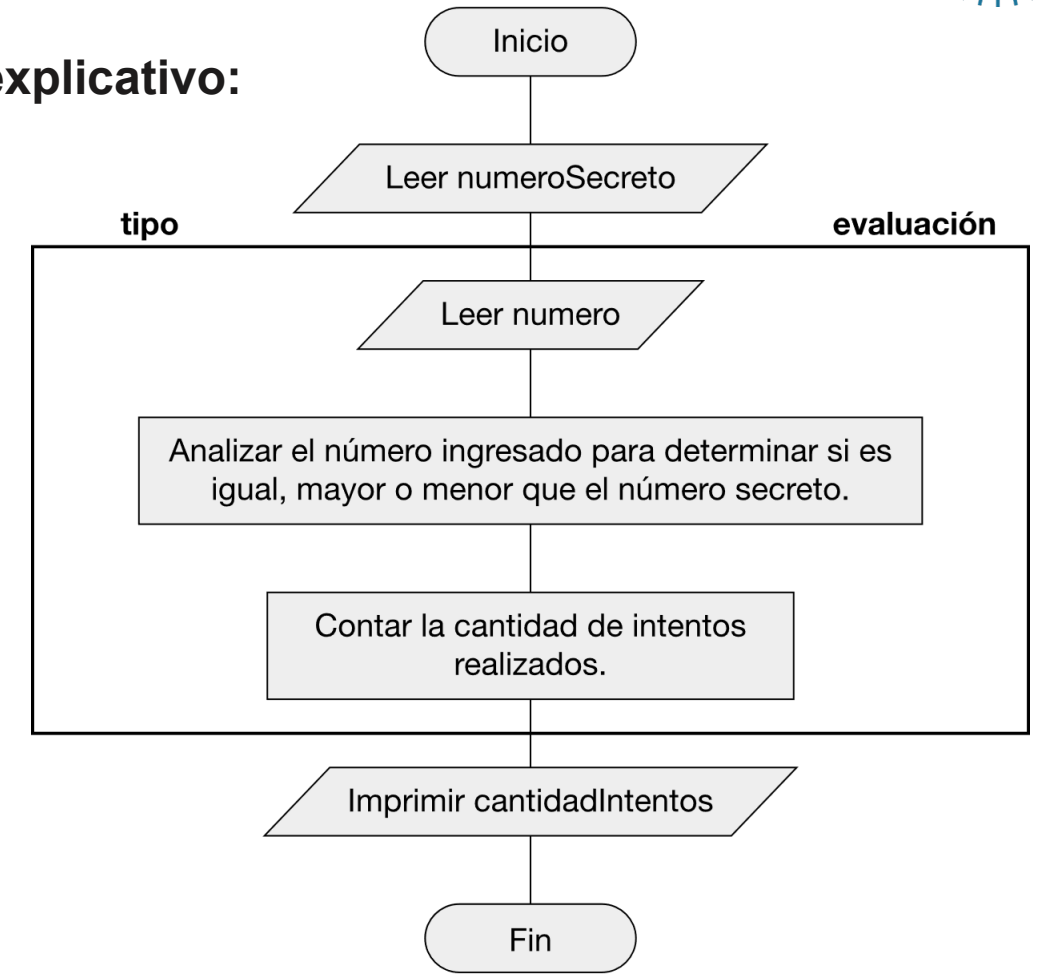
Analizar el número ingresado para determinar si es igual, mayor o menor que el número secreto

```
cantidad_intentos = cantidad_intentos + 1
```

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama explicativo:



Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Cuántas veces se repetirá el ciclo? (evaluación)

El ciclo se repetirá mientras la variable *numero* sea diferente a la variable *numero_secreto*. Debido a que no se conoce la cantidad intentos que le tomará al usuario adivinar el número y se quiere ejecutar las acciones del ciclo antes de realización la evaluación se utilizará un ciclo ***do – while***.

numero != numero_secreto

¿Cómo se modificará la bandera? (modificación)

En cada iteración se obtendrá un nuevo valor por teclado para la variable *numero*.

Leer numero

¿Con qué valor se inicializa la bandera? (inicialización)

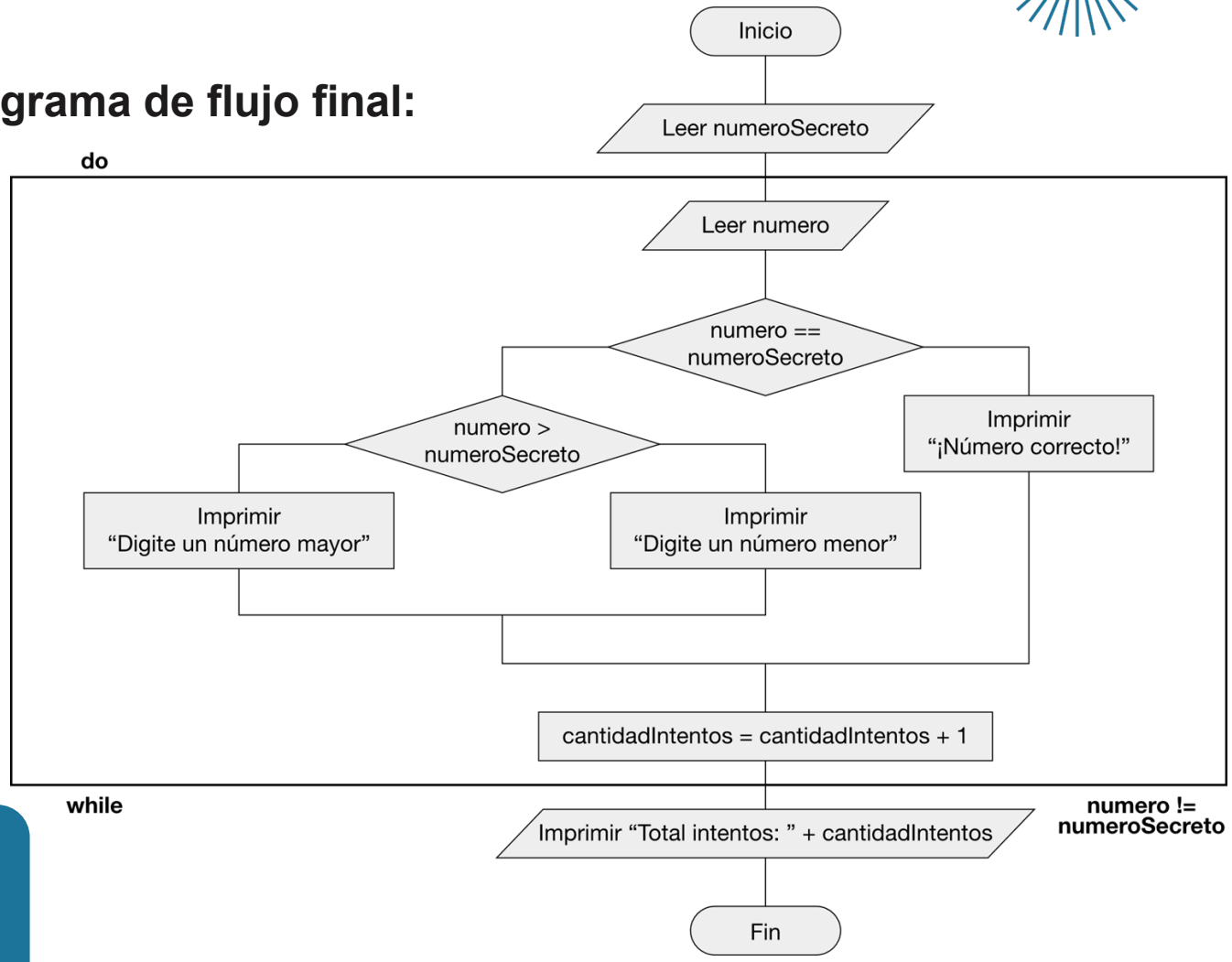
Con el valor dado por el usuario por teclado para la variable *numero*.

Leer numero

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama de flujo final:



En este ejemplo la inicialización y la modificación del ciclo, corresponden a la misma instrucción ***Leer numero***.

Ejemplo 2:

universidad cenfotec_
La U de la informática

numero_secreto.py

```
1    numero_secreto = 0
2    numero = 0
3    cantidad_intentos = 0;
4    numero_secreto = int(input('Digite el número secreto: '))
5    while (numero != numero_secreto):
6       numero = int(input('Digite un número: '))
7       if (numero == numero_secreto):
8           print('¡Número correcto! ')
9       else:
10          if (numero > numero_secreto):
11              print('Digite un número menor ')
12          else:
13              print('Digite un número mayor ')
14      cantidad_intentos += 1
15   print(f'Total de intentos realizados: {cantidad_intentos}')
```


universidad
cenfotec_
La U de la informática