



universidad
cenfotec_
La U de la informática

Principios de Programación.

Estructuras condicionales simples y dobles.

Objetivos

- Identificar las estructuras de selección y sus variantes.
- Comprender el significado de las estructuras de selección y los elementos que las componen.
- Identificar cuándo usar una estructura de selección y usar la adecuada.
- Comprender la implementación de las estructuras de selección en Python.

Estructuras de selección simples y dobles

Estructura de selección

La **estructura de selección** es una *estructura de control* que permite redirigir la ejecución de ciertas acciones según la evaluación de una condición. Si la evaluación de la condición es **verdadera**, se ejecutarán ciertas acciones, de lo contrario, si la evaluación de la condición es **falsa**, se ejecutarán otras acciones.

En otras palabras, la **estructura de selección** permite controlar la ejecución de aquellas acciones que requieren de ciertas condiciones para su realización.

Gracias a dichas condiciones se decide si las acciones se ejecutan o no.

Estructura de selección

La selección presenta esencialmente 3 formas:

- Simple.
- Doble.
- Múltiple.

Esta clasificación se basa en si hay: una, dos o más de dos alternativas de acción.

La estructura de selección se puede clasificar en varias formas:

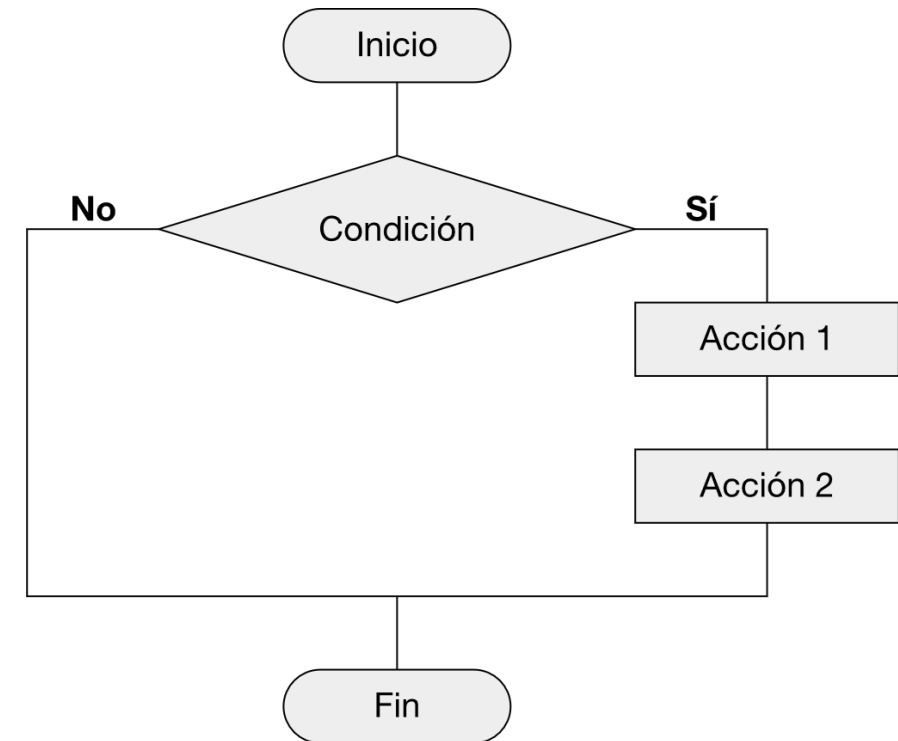
- Simple.
- Doble.
- Anidada.
- Múltiple.

Esta clasificación se basa en si hay una, dos o más de dos alternativas de acción, o si las alternativas de acción se anidan una dentro de la otra.

Estructuras de selección simple

La **estructura de selección simple** permite controlar la ejecución de una o más acciones cuando sólo existe una opción, la de ejecutar las acciones cuando la evaluación de la condición es **verdadera**. Cuando la evaluación de la condición es **falsa**, no existe ninguna acción a ejecutar.

Se utiliza cuando una o varias acciones se encuentran condicionadas para que se lleven a cabo, pero no se tiene una opción alterna.



Estructuras de selección simple en Python

En Python la estructura de selección simple se implementa con la instrucción `if`, de la siguiente manera:

**Los paréntesis son opcionales, pero ayudan a comprender el orden de las instrucciones*

```
if (condición):  
    acción 1  
    acción 2
```

Estructuras de selección simple en Python

En Python para definir que las acciones 1 y 2 se ejecutan únicamente cuando la condición se cumple se usa una indentación (con la tecla tabuladora del teclado).

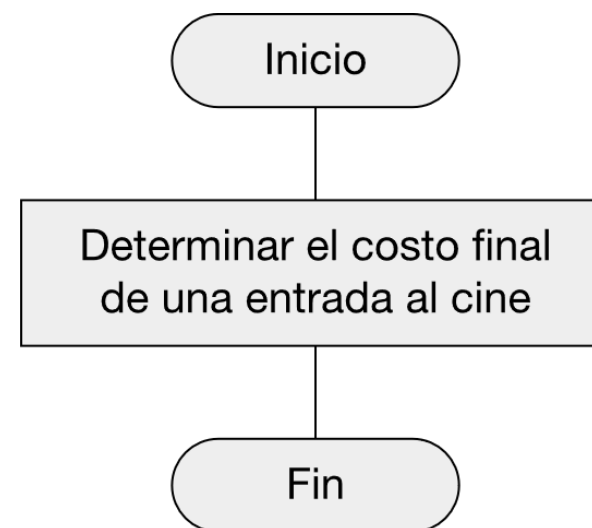
```
if (condición):  
    — acción 1  
    — acción 2
```


Ejemplo 1: Imprimir el costo de una entrada

Enunciado

En un cine se determina que el costo de la entrada para los días miércoles equivale a la mitad del precio normal. Dado el costo de la entrada y el día de la semana determine el costo final.

Diagrama general



Ejemplo 1: Imprimir el costo de una entrada

Tablas de variables y constantes (Análisis)

ENTRADAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Costo de la entrada al cine	costo	int	3000
Día de la semana	dia	String	'miércoles'

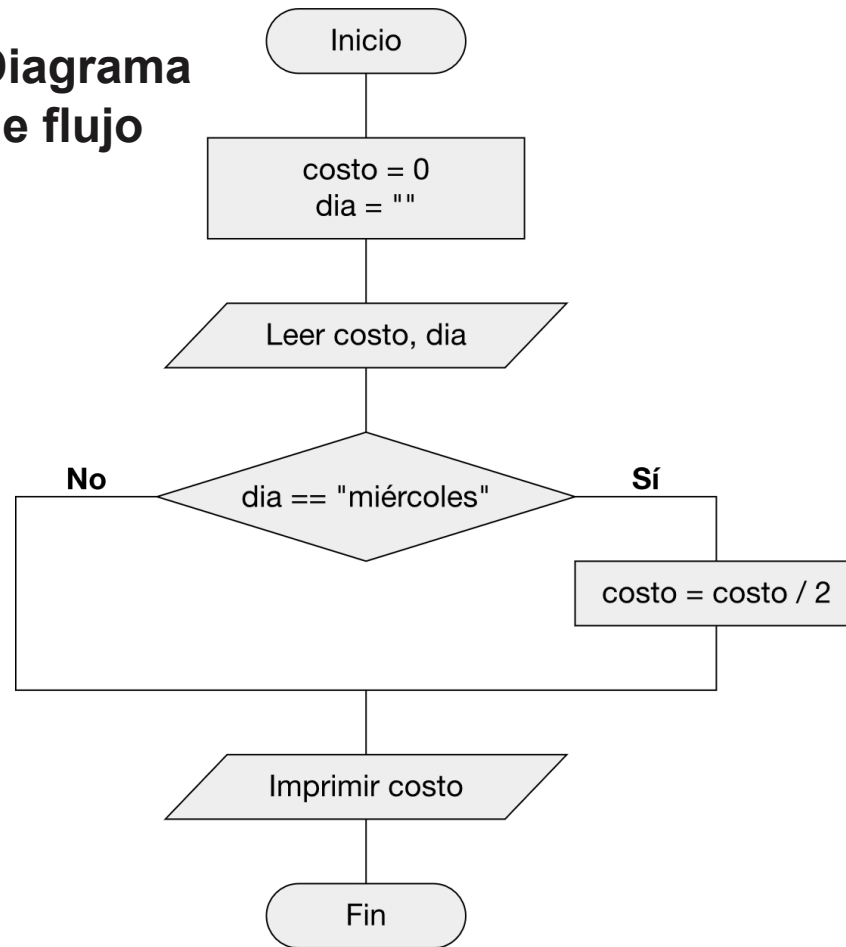
Ejemplo 1: Imprimir el costo de una entrada

Tablas de variables (Análisis)

SALIDAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Costo final dela entrada	costo	int	3000

Ejemplo 1: Imprimir el costo de una entrada

Diagrama
de flujo



calcular_salario.py

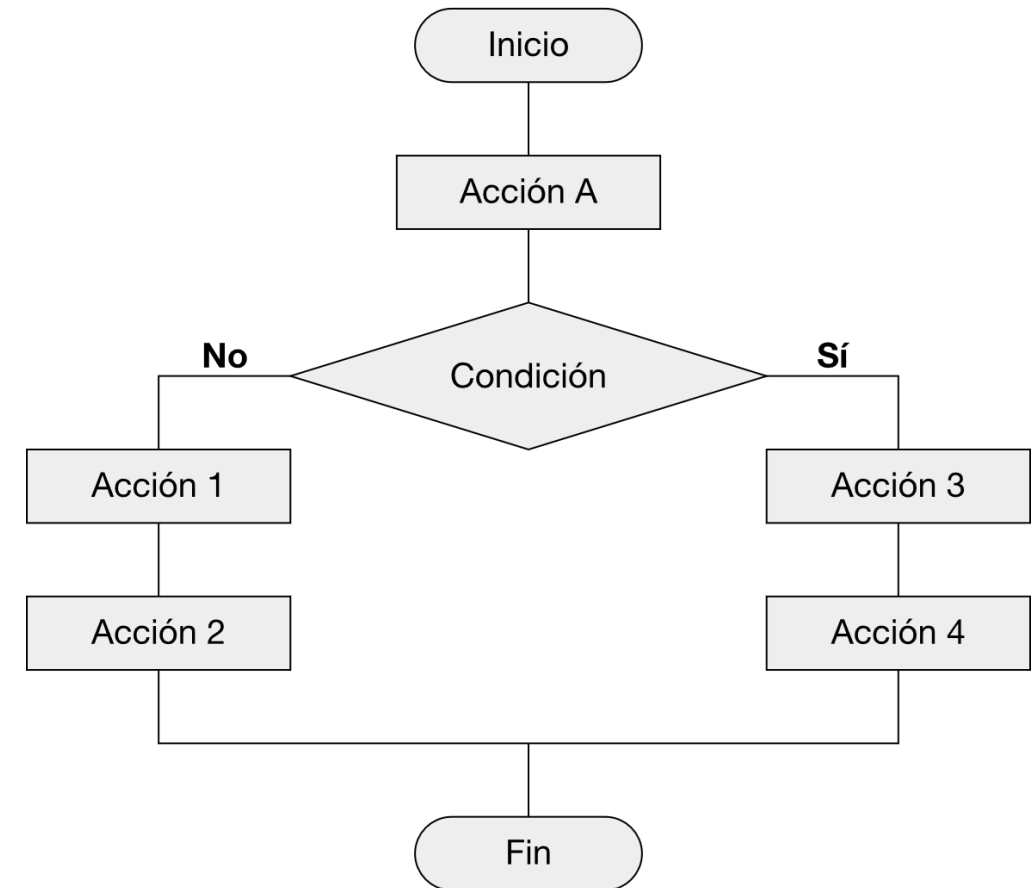
```
1  costo = 0
2  dia = ''
3
4  costo = int(input('Ingrese el costo de la entrada: '))
5  dia = input('Ingrese el nombre del día de la semana: ')
6  if(dia == 'miércoles'):
7      costo = costo / 2
8  print(costo)
```

Estructura de selección doble

La **estructura de selección doble** permite controlar la ejecución de acciones cuando se presentan **dos opciones diferentes**.

Cuando la evaluación de la condición es **verdadera**, se ejecutarán las acciones del lado del “Sí” y cuando la evaluación de la condición es **falsa**, se ejecutarán las acciones del lado del “No”.

Por la naturaleza de estas estructuras, se ejecuta sólo un grupo de acciones, pero no ambos, es decir, son mutuamente excluyentes.

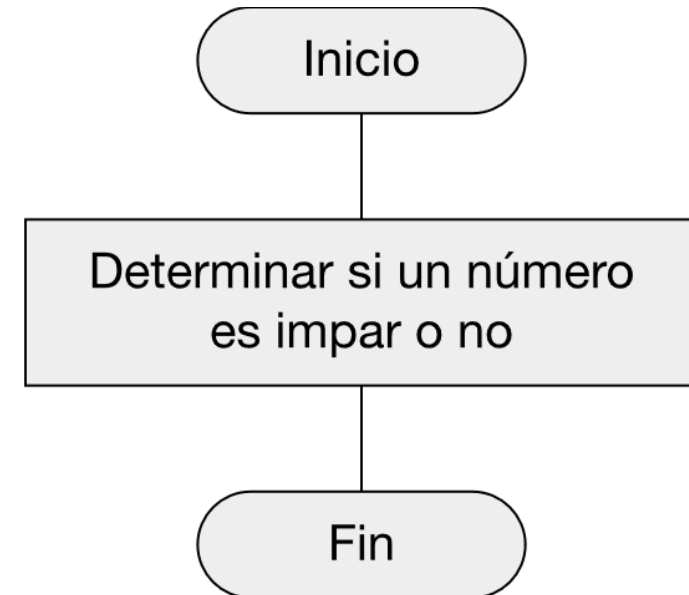


Ejemplo 1: Imprimir si un número es impar o par

Enunciado

Se quiere hacer un programa que imprima el mensaje “el número es impar” si el número dado es impar o imprima “el número es par”, de lo contrario.

Diagrama general



Ejemplo 1: Imprimir si un número es impar o par

Tablas de variables y constantes (Análisis)

ENTRADAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Número entero	numero	int	48

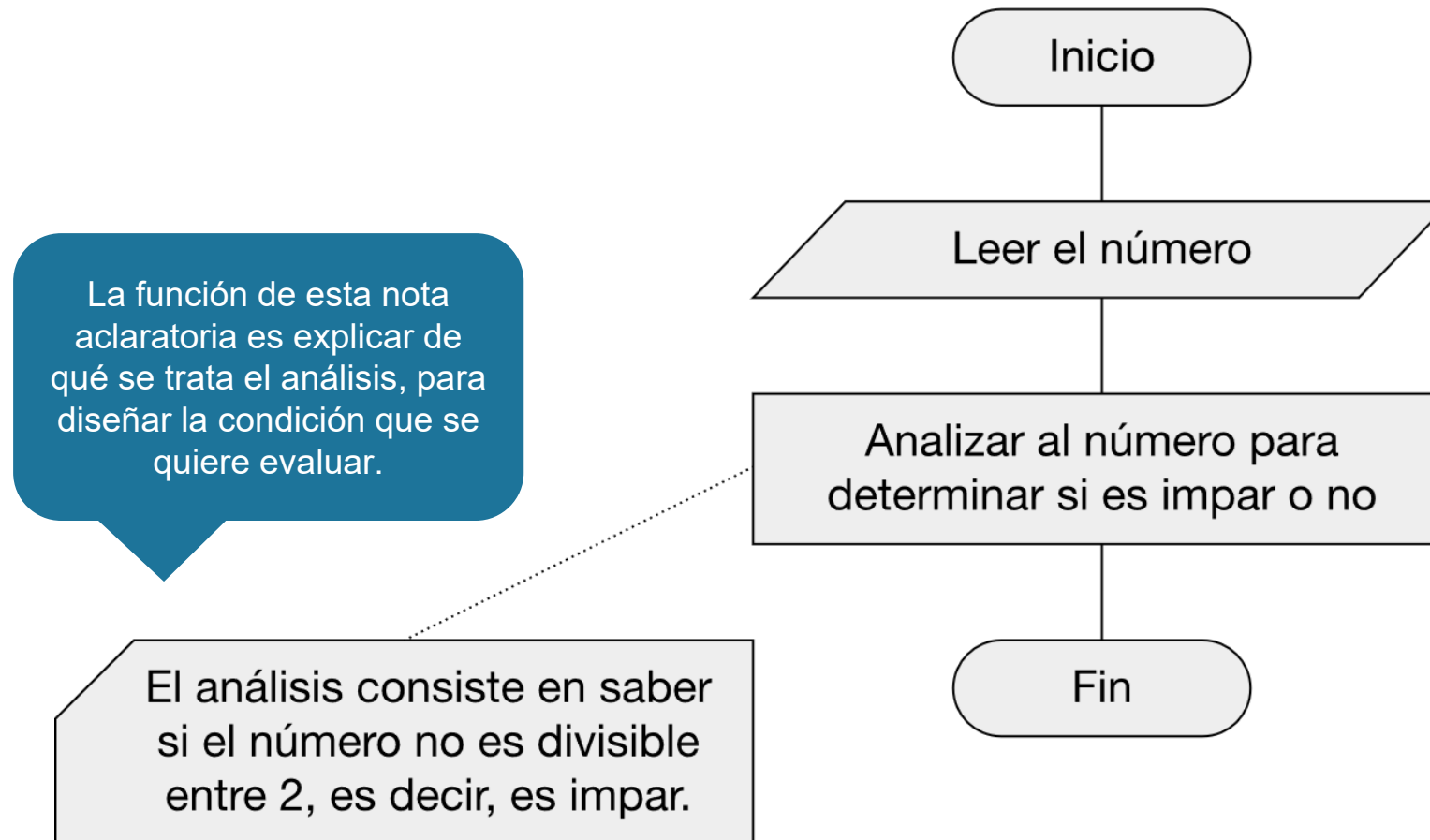
Ejemplo 1: Imprimir si un número es impar o par

Tablas de variables (Análisis)

SALIDAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Se imprime el mensaje “el número es impar” si el número es impar, o “el número es par” de lo contrario.	msj	String	'Es par'

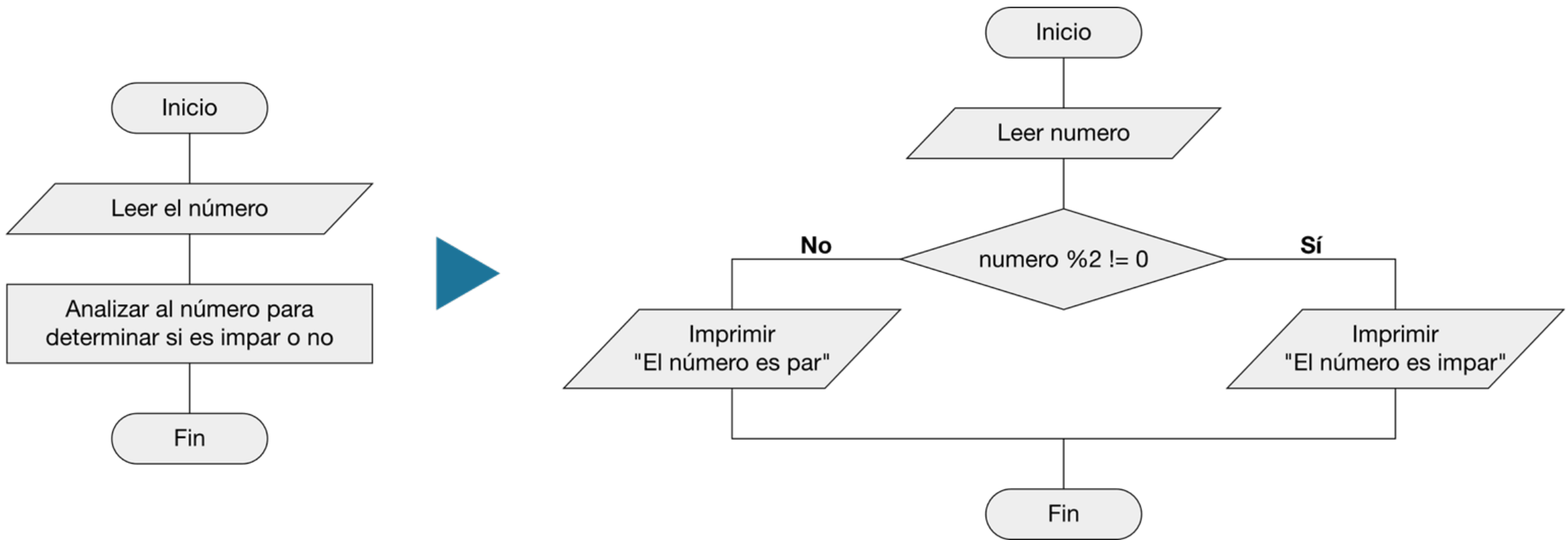
Ejemplo 1: Imprimir si un número es impar o par

Diagrama explicativo



Ejemplo 1: Imprimir si un número es impar o par

Algoritmo o Diagrama de Flujo



Ejemplo 1: Imprimir si un número es impar o par

es_par.py

```
1  numero = 0
2
3  numero = int(input('Por favor teclee un número para saber si es impar: '))
4
5  if (numero % 2 != 0):
6      print('El número: ', numero, 'es impar')
7  else:
8      print('El número: ', numero, 'es par')
```

Ejemplo 2: Calcular el costo de hospedaje

Problema:

Haga un programa que calcule la cuenta de un cliente que se hospeda una cierta cantidad de noches en un hotel, el costo por noche es definido por el usuario:

- Si el cliente se hospeda más de 4 noches se le hace un descuento del 15%
- Esta responsabilidad la podemos expresar como:

Analizar las noches de estadía **para determinar** el coste de hospedaje.

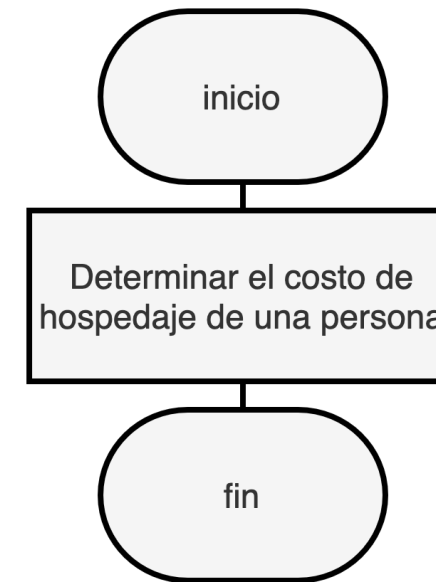
Ejemplo 2: Calcular el costo de hospedaje

Enunciado

Haga un programa que calcule la cuenta de un cliente que se hospeda una cierta cantidad de noches en un hotel, el costo por noche es definido por el usuario:

- Si el cliente se hospeda más de 4 noches se le hace un descuento del 15%
- En caso contrario el descuento es de 5%

Diagrama general



Ejemplo 2: Calcular el costo de hospedaje

Tablas de variables y entradas (Análisis)

ENTRADAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Cantidad de noches a hospedarse	cant_noches	int	6
Costo de estadía de cada noche	costo_noche	float	100.5

Ejemplo 2: Calcular el costo de hospedaje



Tablas de variables (Análisis)

INTERMEDIAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Monto del descuento	descuento	float	30
Costo subtotal del hospedaje	subtotal_hospedaje	float	600

Ejemplo 2: Calcular el costo de hospedaje

Tablas de variables (Análisis)

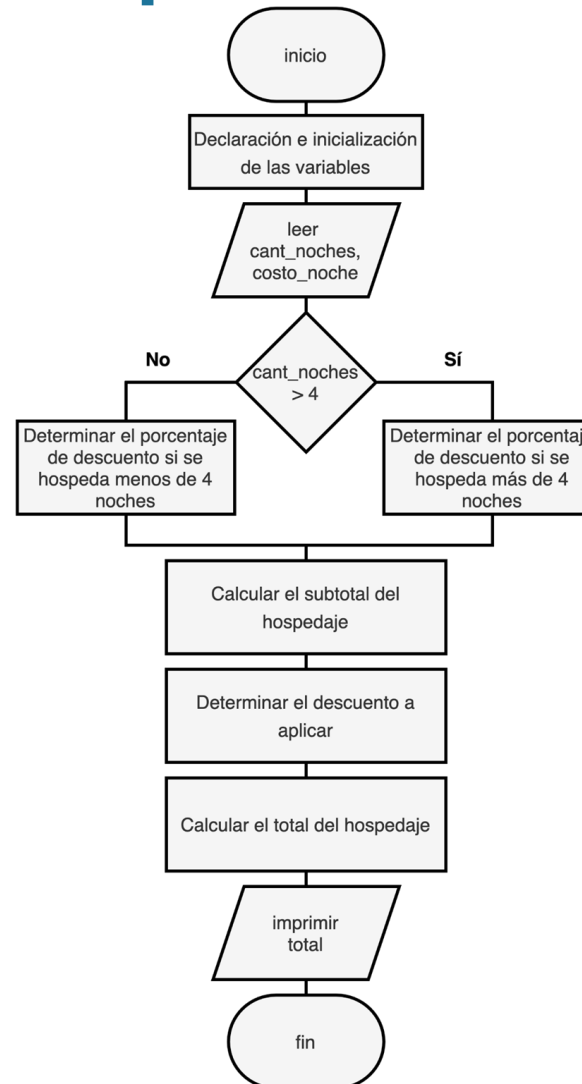
DATOS ESTABLECIDOS O CONSTANTES			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Monto del porcentaje del descuento	porc_desc	float	0.05

Ejemplo 2: Calcular el costo de hospedaje

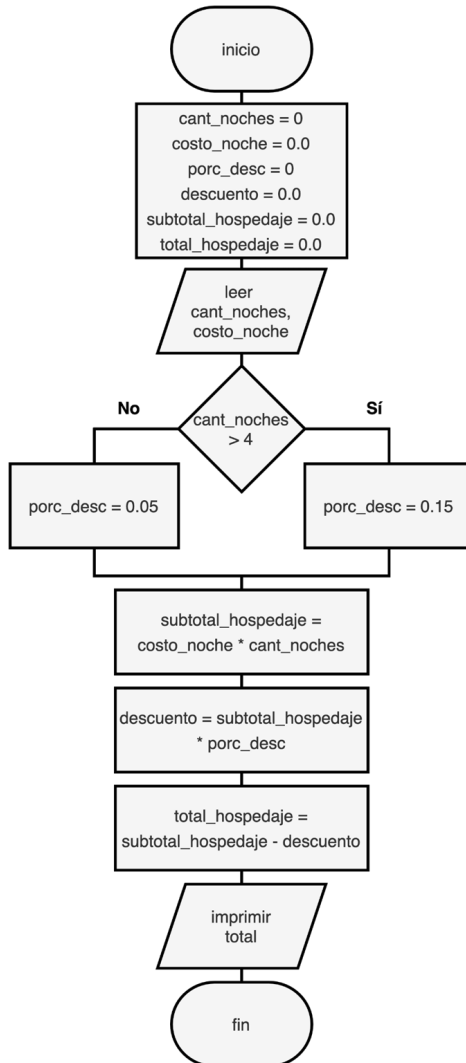
Tablas de salidas (Análisis)

SALIDAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Monto total del hospedaje	total_hospedaje	float	570

Ejemplo 2: Diagrama explicativo



Ejemplo 2: Diagrama de flujo e implementación



hospedaje.py

```
1 cant_noches = 0
2 costo_noche = 0.0
3 porc_desc = 0
4 descuento = 0.0
5 subtotal_hospedaje = 0.0
6 total_hospedaje = 0.0
7 cant_noches = int(input('Ingrese la cantidad de noches a hospedarse: '))
8 costo_noche = float(input('Ingrese el costo de la estadía por noche: '))
9 if (cant_noches > 4):
10     porc_desc = 0.05
11 else:
12     porc_desc = 0
13 subtotal_hospedaje = costo_noche * cant_noches
14 descuento = subtotal_hospedaje * porc_desc
15 total_hospedaje = subtotal_hospedaje - descuento
16 print(f'El costo total del hospedaje es: {total_hospedaje}')
```



universidad
cenfotec_
La U de la informática