



universidad
cenfotec_
La U de la informática

Principios de Programación.

Ejemplos resueltos.
Funciones y procedimientos con listas.

Ejemplos resueltos de Funciones

A continuación, se presentan algunos de los ejemplos de rutinas (funciones y procedimientos) con listas.

Primer ejemplo de rutinas con listas

Ejemplo 1:

Problema:

Se quiere saber cuál es la mayor nota de un grupo de estudiantes de Fundamentos de Programación.

1. Análisis.

2. Diseño o modelado.

3. Implementación.

Definición:

Se supone que las notas del grupo de estudiantes de Fundamentos de Programación se almacenan en un lista de números reales, así se puede leer la lista y luego recorrerlo para calcular cuál es la mayor nota. El número de estudiantes del grupo es dado por el usuario.

Ejemplo 1:

Variables de entrada y salida:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
Variable para guardar el número de estudiantes del grupo.	num_estudiantes	entero	3
Variable para guardar todas las notas del grupo de estudiantes	notas	lista	[78, 90.5, 89.4]
Salida			
Variable que almacena la mayor nota del grupo.	mayor_nota	float	90.5

Ejemplo 1:

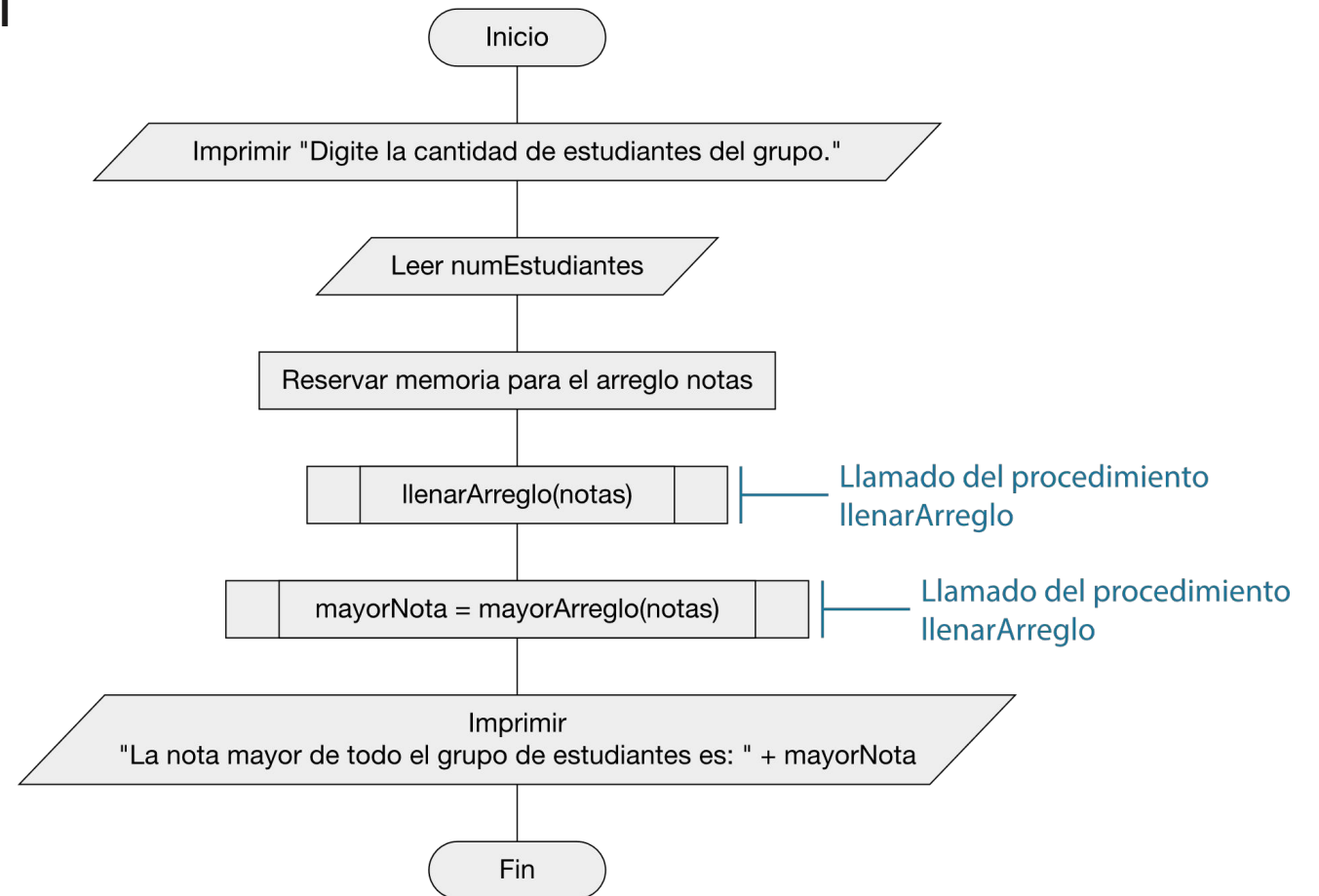
Planteamiento

En la lista **notas** se van a almacenar las notas de todos los estudiantes del grupo de Principios de Programación, pero no se sabe con anterioridad cuántos estudiantes hay en el grupo. Es por esto que se deben seguir los siguientes pasos:

1. Análisis.
 2. Diseño o modelado.
 3. Implementación.
1. Declarar la lista de reales llamado **notas**.
 2. Preguntarle al usuario cuántos estudiantes tiene el grupo y reservar el espacio en memoria para ello.
 3. Usar una rutina para leer la lista **notas**.
 4. Usar una rutina para calcular la mayor de las notas del curso
 5. Imprimir la nota mayor.

Ejemplo 1:

Diagrama de flujo del *main*:



1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 1:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

mayor_nota_grupo.py

```
1  notas = []
2  mayor_nota = 0
3  tamanno_lista = 0
4
5  #Se llama al procedimiento que lee el lista.
6  llenar_lista(notas)
7  #Se llama a la función que calcula el mayor
8  mayor_nota = mayor_lista(notas)
9  print(f'La mayor nota del grupo es: {mayor_nota}')
```


Ejemplo 1:

Rutina *llenar_lista*:

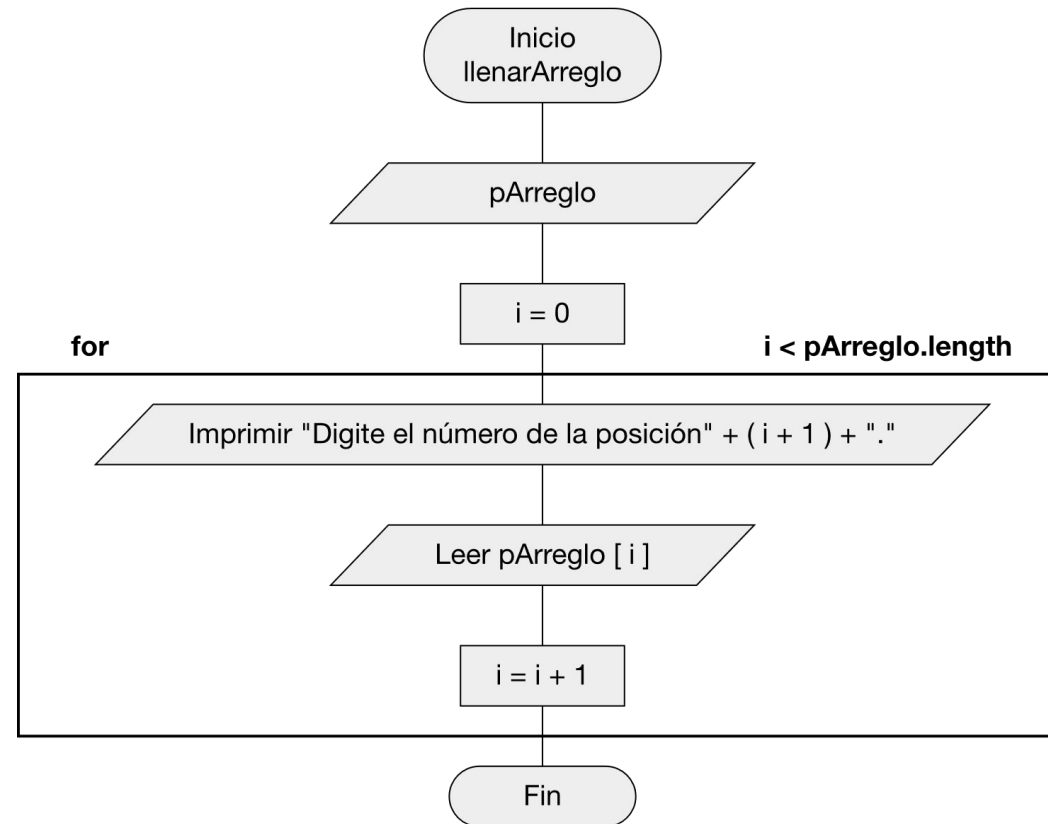
1. Análisis.
2. Diseño o modelado.
3. Implementación.

Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de reales que se recibe en parámetros.	plista	lista	[]
Salida			
lista de reales que se recibe como parámetro y se lee o llena en el procedimiento.	plista	lista	[89.2, 90.5, 76, 80]

Ejemplo 1:

Diagrama de flujo de *llenar_lista*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.



Ejemplo 1:

procedimiento: *llenar_lista*

1. Análisis.
2. Diseño o modelado.
3. Implementación.

```
1  def llenar_lista(plista):
2      i = 0
3      cantidad = int(input('Ingrese la cantidad de notas que desea agregar: '))
4      while (i < cantidad):
5          print();
6          nota = float(input(f'Digite la nota {(i + 1)} : '))
7          plista.append(nota)
8          i += 1
```

Ejemplo 1:

Rutina *mayor_lista*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

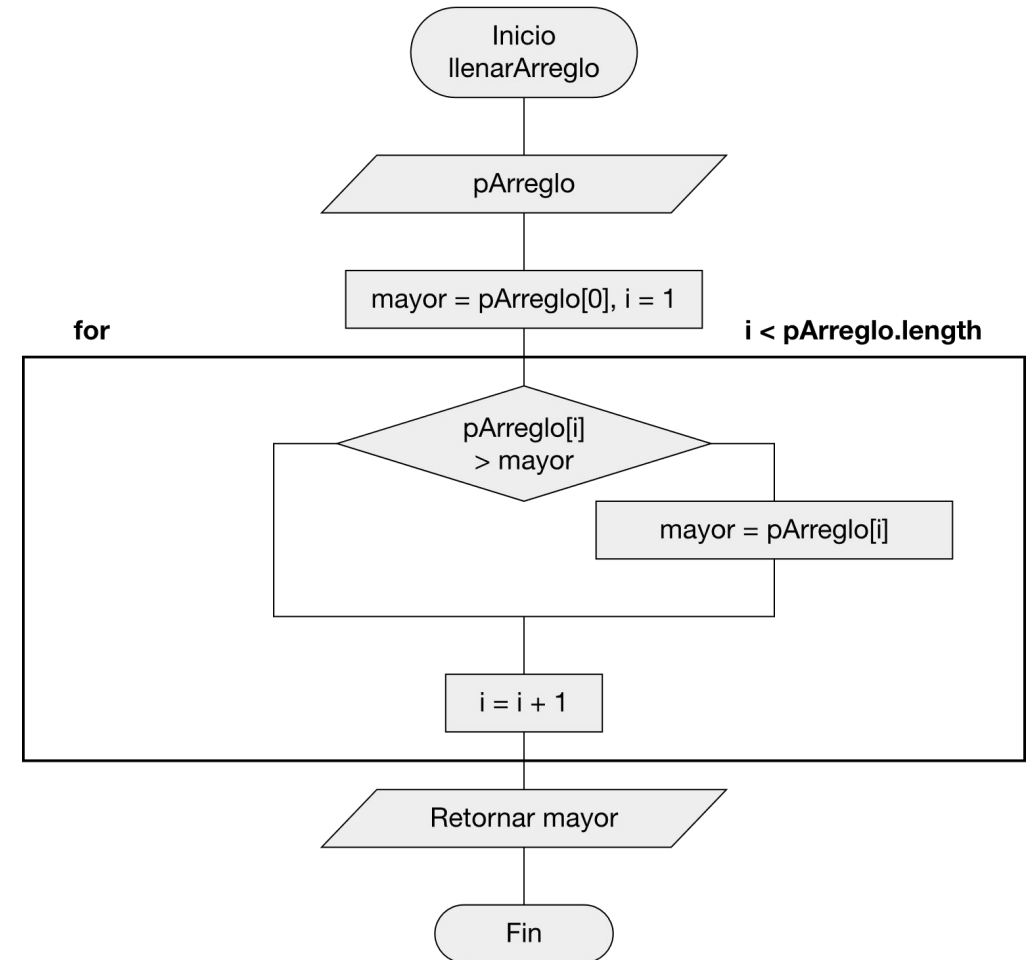
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de enteros que se recibe por parámetros.	plista	lista	[89, 90, 78, 56]
Salida			
Variable que almacena el mayor de la lista	mayor	float	90

Ejemplo 1:

Diagrama de flujo de *mayor_lista*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Note que la variable **mayor** se inicializa con el valor **plista[0]**, esto es para inicializar la variable con un valor que se encuentre dentro de la lista. Ya que **mayor** se inicializa con este valor, el índice o contador **i** se inicializa con el valor **1**.



Ejemplo 1:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

función: *mayor_lista*

```
1  def mayor_lista(plista):
2      #Se inicializa la variable mayor con el valor plista[0] en i con 1
3      i = 1
4      mayor = plista[0]
5      while (i < len(plista)):
6          if (plista[i] > mayor):
7              mayor = plista[i]
8              i += 1
9
10     return mayor
```

Segundo ejemplo de rutinas con listas

Ejemplo 2:

Problema:

En un almacén se quieren registrar los precios de dos listas de productos. Una lista de productos se quiere poner en promoción dada las ventas de fin de año. Y la otra lista de productos se le quiere agregar el IVA debido a que son productos que tienen impuesto.

1. Análisis.

2. Diseño o modelado.

3. Implementación.

Definición:

Se supone que las listas de los productos del almacén se guardan en dos listas de números reales, una lista para cada lista. El usuario indicará el número de productos que se quiere registrar en cada lista. Así, se llenan las dos listas con valores dados por el usuario, y luego se modifica cada lista, una para disminuir los precios de acuerdo a la promoción indicada por el usuario y otra se modifica para aumentar los precios de acuerdo al IVA. Por último se imprimen las dos listas de precios modificadas.

Ejemplo 2:

Variables de entrada:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
Variable para guardar los precios de la lista de productos que se quiere poner en promoción.	precios_promocion	lista	[78000,..., 89500]
Variable para guardar los precios de la lista de productos que se le quiere aplicar el IVA.	precios_iva	lista	[28500,..., 59000]
Variable que guarda el porcentaje de promoción o descuento que se le va a aplicar a la primera lista de productos.	porcentaje_promocion	float	25
Constante que guarda el valor del porcentaje de IVA que se le va a aplicar como impuesto a la segunda lista de productos .	IVA	float	0.13

Ejemplo 2:

Variables de salida:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Salida			
Lista de precios modificada después de haberle aplicado el descuento o promoción definida por el usuario.	precios_promocion	lista	[58500,..., 67125\
Lista de precios modificada después de haberle aplicado el impuesto del IVA.	precios_iva	lista	[32205,..., 66670]

Ejemplo 2:

Planteamiento

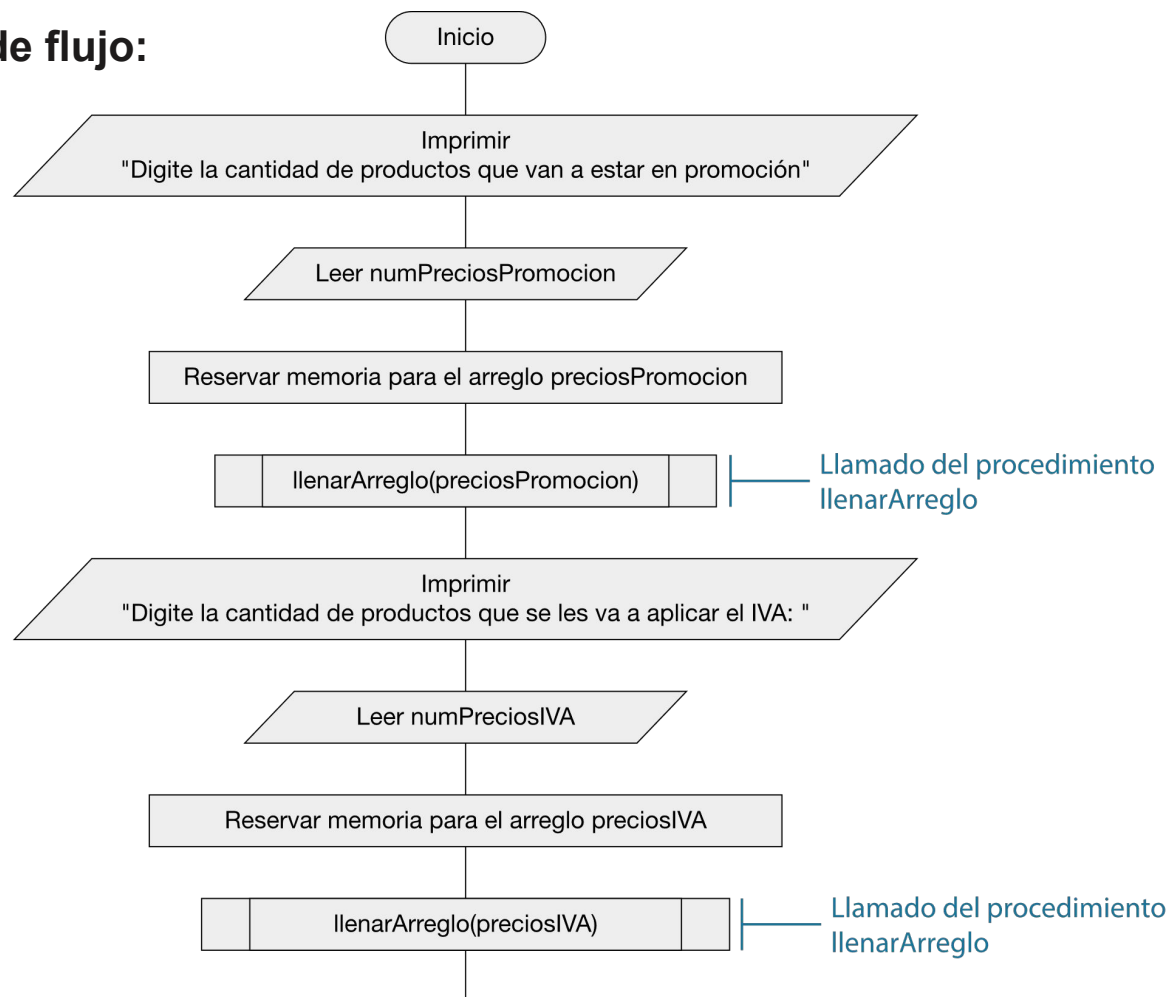
En la lista ***precios_promocion*** se van a almacenar los precios de los productos a los que se les va a aplicar la promoción o descuento. En la lista ***precios_iva*** se van a almacenar los precios de los productos a los que se les va a aplicar el impuesto o IVA. Para resolver el problema se deben seguir los siguientes pasos:

1. Análisis.
 2. Diseño o modelado.
 3. Implementación.
1. Declarar los dos listas de reales y reservar el espacio en memoria para ellos.
 2. Leer o llenar los dos listas.
 3. Leer el descuento o promoción que quiere aplicar el usuario y modificar al lista ***precios_promocion*** para aplicarle este descuento.
 4. Modificar al lista ***precios_iva*** para aplicarle el impuesto del IVA.

Para leer los dos listas se usa el procedimiento ***llenar_lista*** que se llama primero para leer a ***precios_promocion*** y luego se llama para leer a ***precios_iva***. Luego se usa un procedimiento ***aplicar_promocion*** para modificar la lista aplicando la promoción o descuento. Luego se usa otro procedimiento ***aplicar_iva*** para modificar la lista aplicando el impuesto del IVA. Por último, se usa el procedimiento ***imprimir_lista*** que se llama primero para imprimir a ***precios_promocion*** después de haber sido modificado, y luego se llama para imprimir a ***precios_iva*** después de haber sido modificado.

Ejemplo 2:

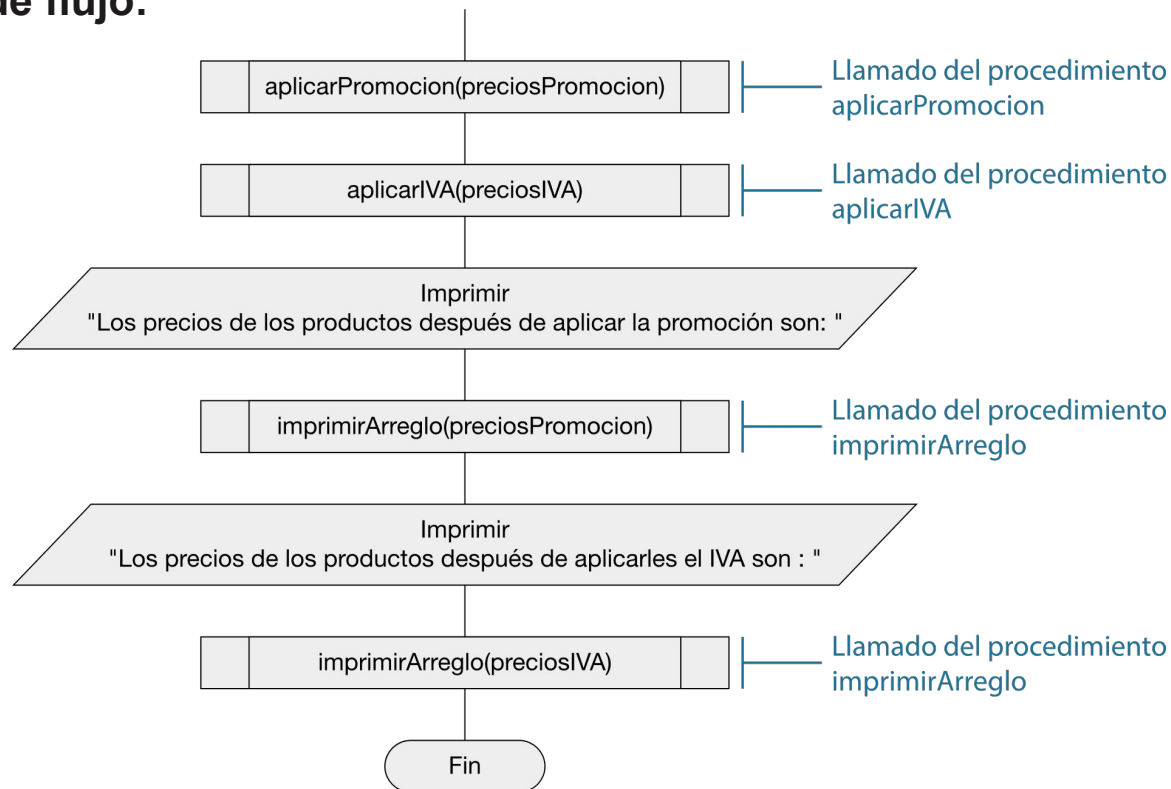
Diagrama de flujo:



1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 2:

Diagrama de flujo:



1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 2:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 1:

precios_iva.py (flujo principal)

```
1  numprecios_promocion = int(input('Por favor indique el número de productos en promoción: '))
2  precios_promocion = []
3  llenar_lista(precios_promocion, numprecios_promocion)
4  numprecios_iva = int(input('Por favor indique el número de productos que se les debe aplicar el IVA: '))
5  precios_iva = []
6  llenar_lista(precios_iva, numprecios_iva)
7  descuento = float(input('¿Cuál descuento o promoción le quiere aplicar a los artículos?: '))
8
9  aplicar_promocion(precios_promocion, descuento)
10 aplicar_iva(precios_iva)
11
12 print('_____')
13 print('Los precios de los artículos en promoción son ahora los siguientes: ')
14 imprimir_lista(precios_promocion)
15
16 print('_____')
17 print('Los precios de los artículos con IVA son ahora los siguientes: ')
18 imprimir_lista(precios_iva)
```

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 2:

Rutina *llenar_lista*:

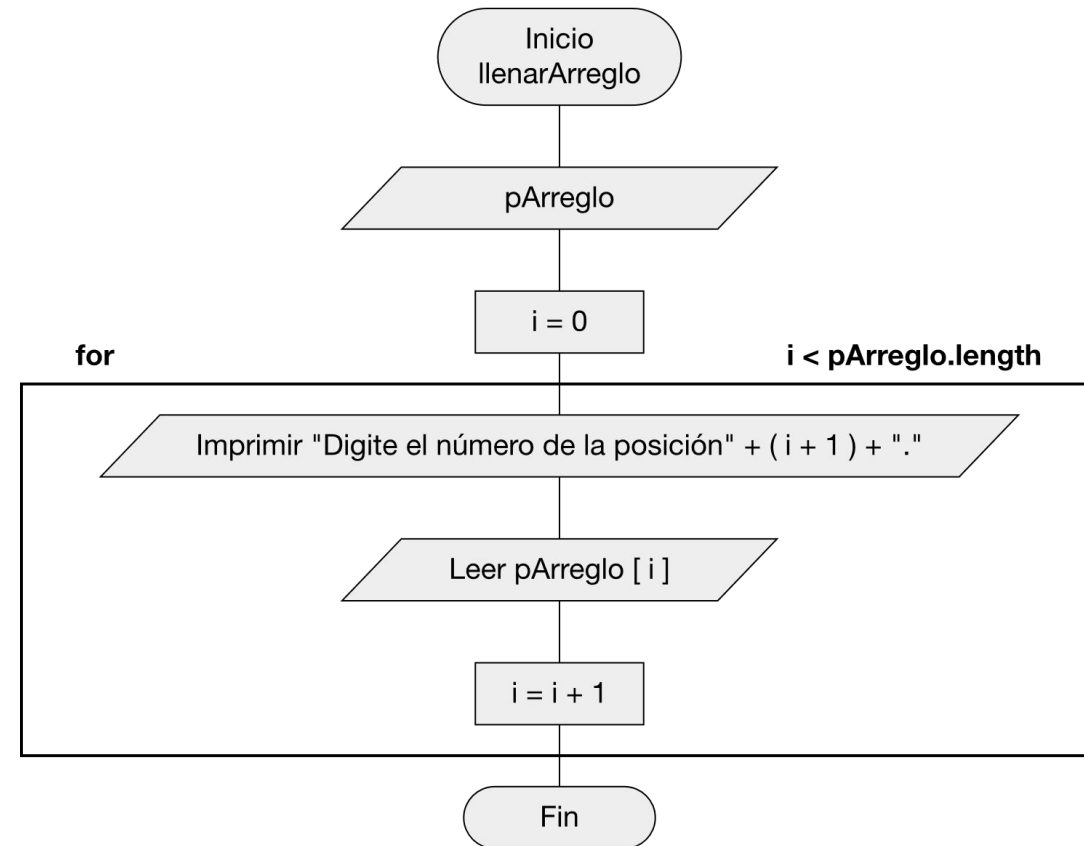
1. Análisis.
2. Diseño o modelado.
3. Implementación.

Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de reales que se recibe por parámetros.	plista	lista	[]
Salida			
lista de reales que se recibe como parámetro y se lee o llena en el procedimiento.	plista	lista	[78000,...,89500]

Ejemplo 2:

Diagrama de flujo de *llenar_lista*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.



Ejemplo 2:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

función: *llenar_lista*

```
1  def llenar_lista(plista, pcantidad):  
2      i = 0  
3      while (i < pcantidad):  
4          plista.append(int(input(f'Digite el número de la posición {(i + 1)} ')))  
5          i += 1
```

Ejemplo 2:

Rutina *aplicar_promocion*:

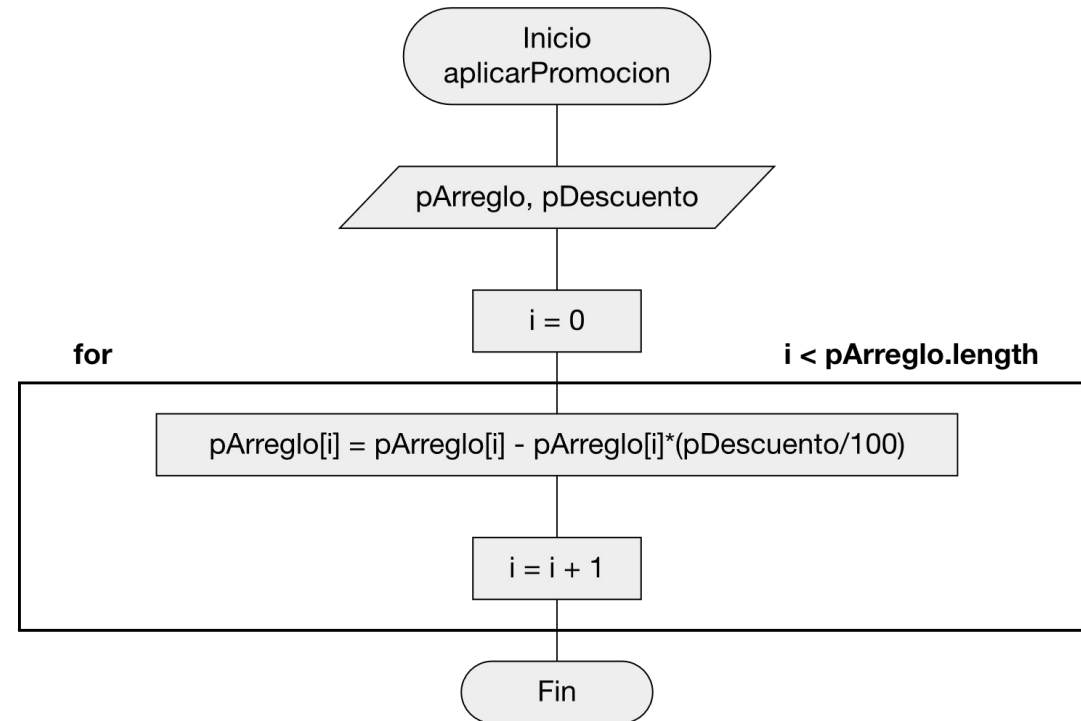
1. Análisis.
2. Diseño o modelado.
3. Implementación.

Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de reales que se recibe por parámetros.	plista	lista	[78000,...,89500]
Variable que guarda el descuento que se quiere aplicar.	pdescuento	real	25
Salida			
lista de reales que se recibe como parámetro y se lee o llena en el procedimiento.	plista	lista	[58500,...,67125]

Ejemplo 2:

Diagrama de flujo de *aplicar_promocion*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.



Ejemplo 2:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

función: *aplicar_promocion*

```
1  def aplicar_promocion(plista, pdescuento):  
2      i = 0  
3      while (i < len(plista)):  
4          plista[i] = plista[i] - plista[i]*(pdescuento/100.0)  
5          i += 1
```

Ejemplo 2:

Rutina *aplicar_iva*:

1. Análisis.

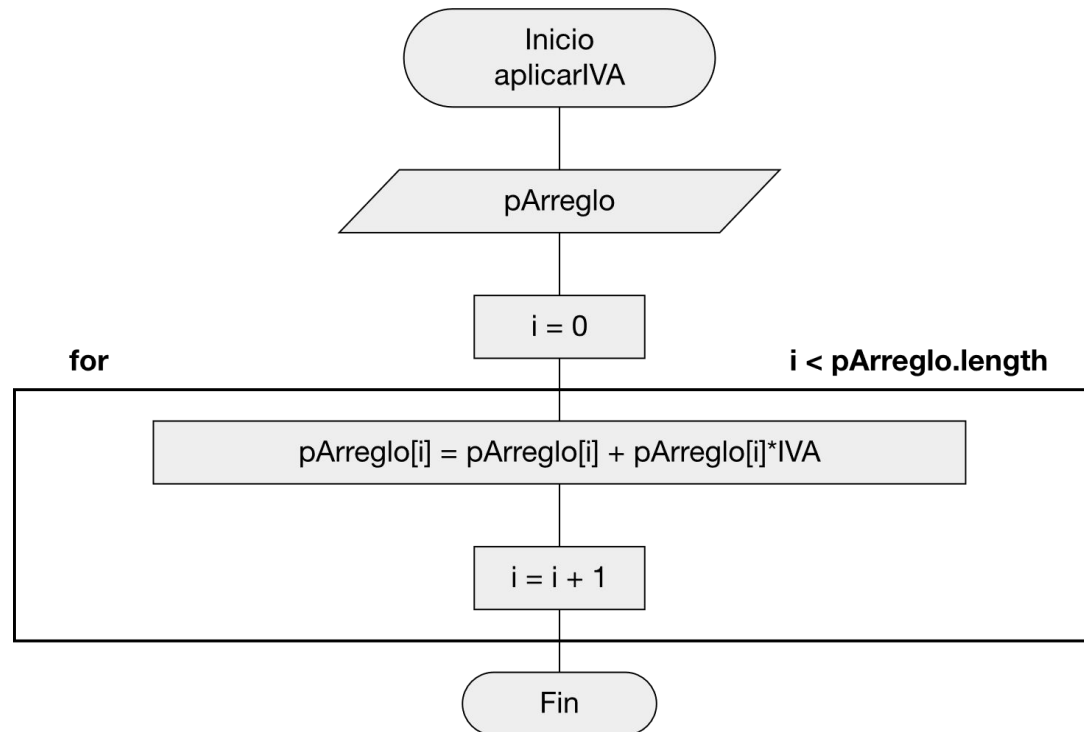
2. Diseño o modelado.

3. Implementación.

Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de reales que se recibe por parámetros.	plista	lista	[28000,...,59000]
Salida			
lista de reales que se recibe como parámetro y se lee o llena en el procedimiento.	plista	lista	[32205,...,66670]

Ejemplo 2:

Diagrama de flujo de
aplicar_iva:



1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 2:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

función: *aplicar_iva*

```
1  def aplicar_iva(plista):
2      i = 0
3      while (i < len(plista)):
4          plista[i] = plista[i] + plista[i] * IVA
5          i += 1
```


Ejemplo 2:

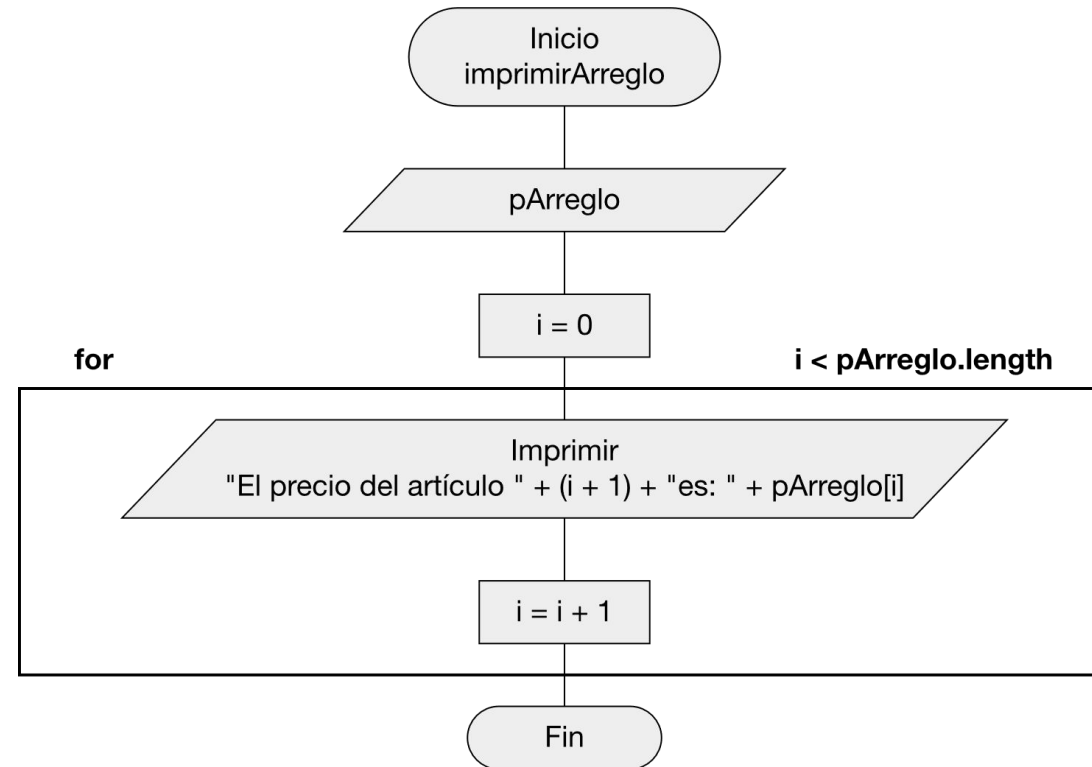
Rutina *imprimir_lista*:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
lista de reales que se recibe por parámetros.	plista	lista	[78000,...,89000]

Ejemplo 2:

Diagrama de flujo de
imprimir_lista:



1. Análisis.
2. Diseño o modelado.
3. Implementación.

Ejemplo 2:

1. Análisis.
2. Diseño o modelado.
3. Implementación.

función: *imprimir_lista*

```
1  def imprimir_lista(plista):
2      i = 0
3      while (i < len(plista)):
4          print(f'El precio del artículo {(i + 1)} es: {plista[i]}')
5          i += 1
```



universidad
cenfotec_
La U de la informática