



universidad  
cenfotec\_  
La U de la informática

# Principios de Programación.

Ejemplos resueltos.  
Funciones con listas.

# Ejemplos resueltos de Funciones con listas

A continuación, se presentan unos problema que poseen listas.

## Primer ejemplo de funciones con listas

# Ejercicio de dioses

## Problema:

En un sistema informático se tiene el registro de diferentes dioses de la mitología.

Se poseen tres listas para almacenar la información de los dioses registrados, el sexo de cada dios y la cantidad de hijos que posee cada dios.

1. Análisis.

2. Diseño o modelado.

3. Implementación.

## Listas:

- dioses
- generos
- hijos

# Especificación de las listas

A continuación se muestran las tres listas con sus valores respectivos:

Índices									
lista	0	1	2	3	4	5	6	7	8
dioses	Zeus	Hera	Poseidón	Dionisio	Apolo	Artemisa	Hermes	Atenea	Ares
generos	Masculino	Femenino	Masculino	Masculino	Masculino	Femenino	Masculino	Femenino	Masculino
hijos	26	28	16	34	14	26	38	12	9

# Tamaño de las listas

En este problema las tres listas tienen un mismo tamaño, por lo que dependiendo de lo requerido se pueden utilizar cualquier de las siguientes opciones para obtenerlo.

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

## Tamaño:

- len(dioses)
- len(generos)
- len(hijos)

El valor que se obtiene inicialmente siempre va a ser de **9**, que equivale a la **cantidad de elementos** que existen dentro de las listas.

# ¿Cómo interactúan las listas?

Para este ejemplo particular las tres listas se encuentran enlazadas conceptualmente, esto significa que la tanto el nombre como el género y la cantidad de hijos de un dios específico van a estar en la posición  $i$ .

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

Observe el siguiente cuadro donde se muestra como la información de Hera está en la posición 1 de cada listas:

$i = 1$	
dioses[i]	Hera
generos[i]	Femenino
hijos[i]	28

# Listas: implementación

## mostrar\_info\_dioses()

```
1 dioses = ['Zeus', 'Hera', 'Poseidón', 'Dionisio', 'Apolo', 'Artemisa', 'Hermes',  
2 'Atenea', 'Ares']  
3 generos = ['Masculino', 'Femenino', 'Masculino', 'Masculino', 'Masculino',  
4 'Femenino', 'Masculino', 'Femenino', 'Masculino']  
5 hijos = [26, 28, 16, 34, 14, 26, 38, 12, 9]
```



# Primer ejercicio

Realice una rutina que imprima la información de todos los dioses

Se desea imprimir la información de todos los dioses incluyendo:

- nombre
- sexo
- cantidad de hijos, por lo que es necesario

1. Análisis.

2. Diseño o modelado.

La información de cada dios se encuentra distribuida en las tres listas en la posición  $i$ .

3. Implementación.

Esto permite utilizar un ciclo ***incondicional*** (con contador) para recorrer las diferentes listas y acceder a la información.

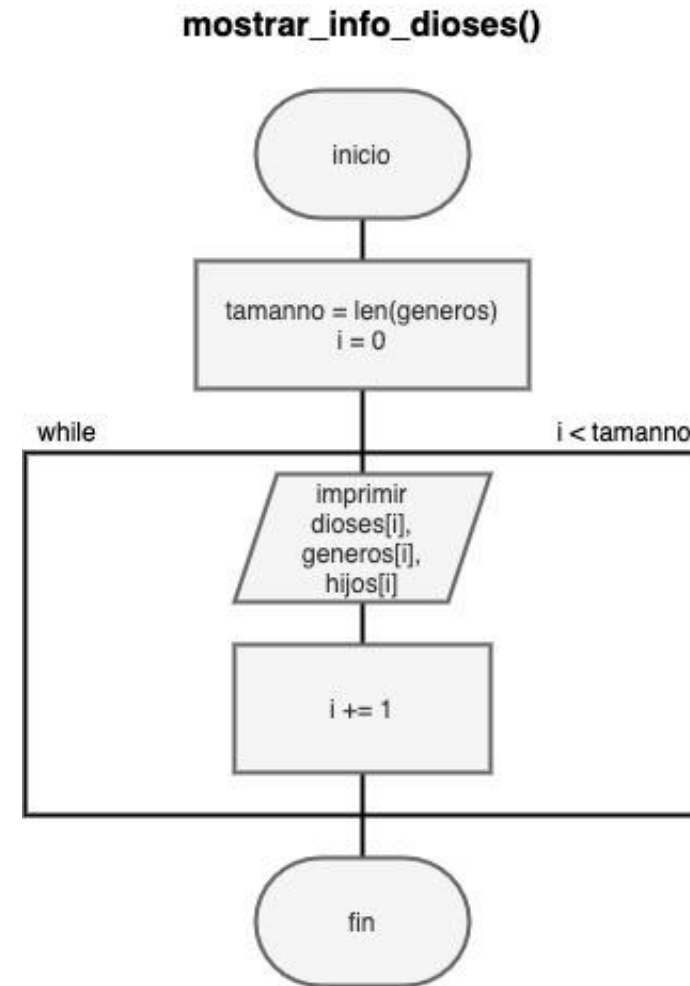
Como las tres listas poseen la misma cantidad de elementos, el tamaño se puede obtener usando cualquiera, como se muestra a continuación:

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

<b>len(dioses)</b>	9
<b>len(generos)</b>	9
<b>len(hijos)</b>	9

# mostrar\_info\_dioses()

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



# mostrar\_info\_dioses()

## Explicación del diagrama:

1. Análisis.
  2. Diseño o modelado.
  3. Implementación.
- Se utiliza un ciclo controlado por el contador *i*
  - Se itera mientras que el contador sea inferior al tamaño de la lista.
    - **$i < \text{tamanno}$**
  - Por cada iteración se imprime la información que se encuentra dentro de la posición *i* de las listas

# mostrar\_info\_dioses() : implementación

## mostrar\_info\_dioses()

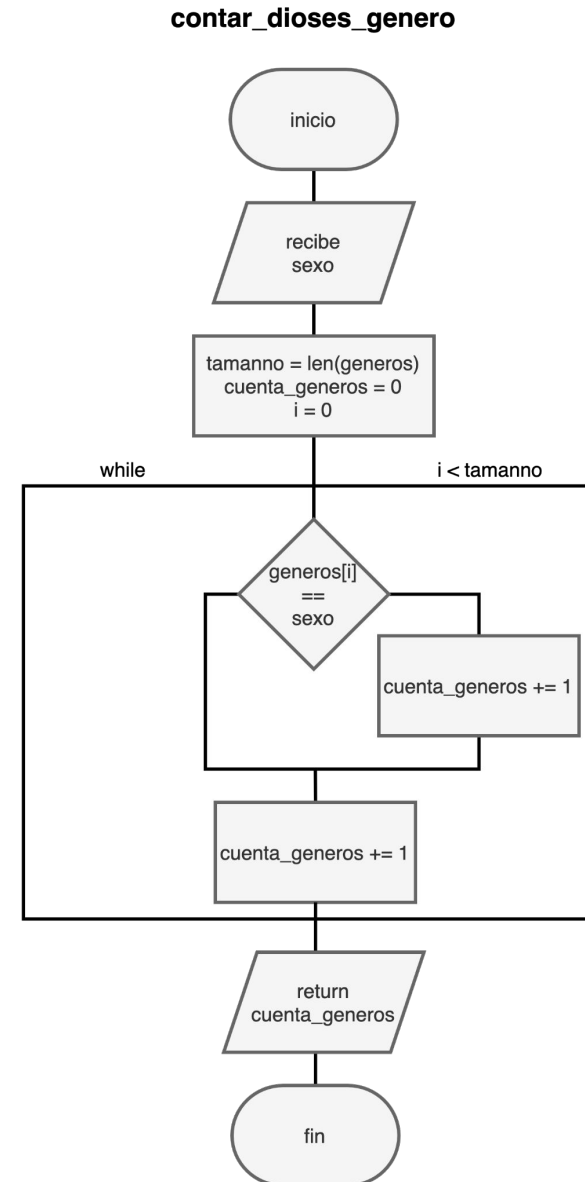
```
1  def mostrar_info_dioses():
2      tamanno = len(generos)
3      i = 0
4      while(i < tamanno):
5          print(f'{dioses[i]} es {generos[i]}')
6          i += 1
7
8  #Prueba
9  mostrar_info_dioses()
```

# Segundo ejercicio

Realice una rutina que reciba el sexo de un dios e indique cuántos dioses con ese sexo existen dentro de la lista

# contar\_dioses\_genero()

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





# contar\_dioses\_genero() : implementación

## contar\_dioses\_genero()

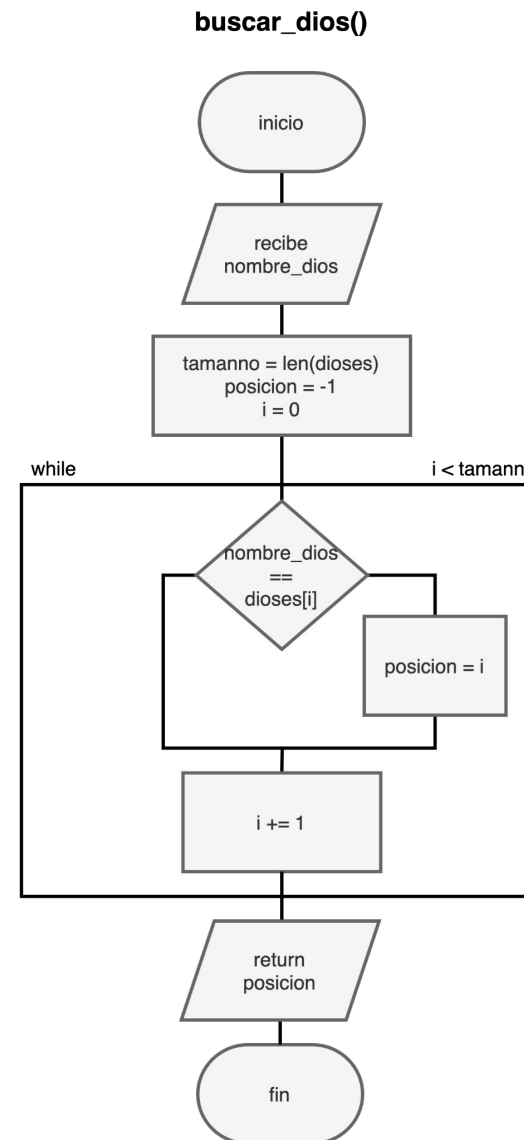
```
1  def contar_dioses_genero(sexo):
2      cuenta_generos = 0
3      tamanno = len(generos)
4      i = 0
5
6      while(i < tamanno):
7          if(generos[i] == sexo):
8              cuenta_generos += 1
9              i += 1
10     return cuenta_generos
11 #Prueba
12 print(f'La cantidad de dioses masculinos es : {contar_dioses_genero("Femenino")})')
```

# Tercer ejercicio

Realice un función que reciba el nombre de un dios, y lo busque dentro de la lista de dioses. Si el dios se encuentra retorna el índice y si el dios no se encuentra en la lista retorna -1

# buscar\_dios()

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



# buscar\_dios() : implementación

## buscar\_dios()

```
1 def buscar_dios(nombre_dios):
2     tamanno = len(dioses)
3     i = 0
4     posicion = -1
5
6     while(i < tamanno):
7         if(nombre_dios.lower() == dioses[i].lower()):
8             posicion = i
9             i += 1
10    return posicion
11
12 #Prueba
13 print(buscar_dios('aPOlo'))
```

### ***\*Importante:***

El método .lower(), permite transformar un String a minúscula.

Al convertir el parámetro nombre\_dios y dioses[i] a minúscula, se mejora la condición.

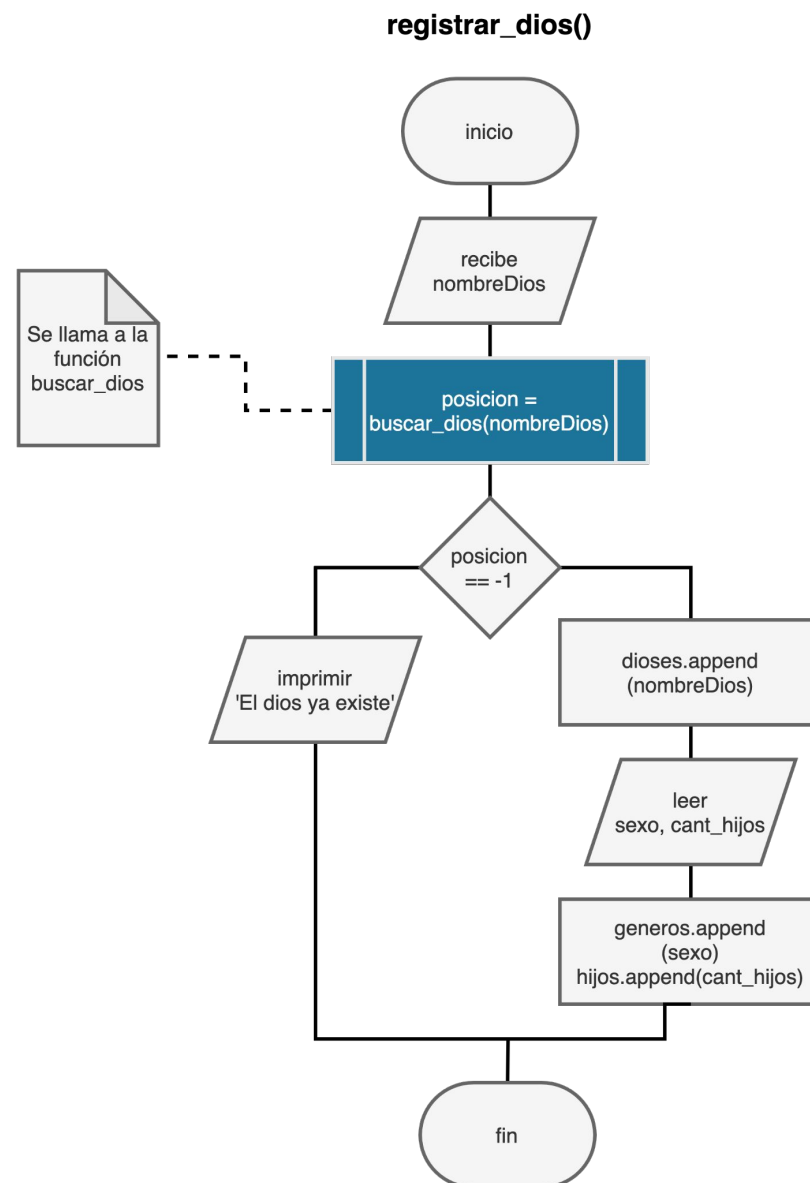
Esto hace que se comparen correctamente los nombres discriminando mayúsculas y minúsculas.

# Cuarto ejercicio

Realice un procedimiento que reciba el nombre de un dios y lo registre junto con su género únicamente si el dios NO existe.

# registrar\_dios()

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



# registrar\_dios() : implementación

## registrar\_dios()

```
1  def registrar_dios(nombreDios):
2      posicion = buscar_dios(nombreDios)
3      if(posicion == -1):
4          dioses.append(nombreDios)
5          sexo = input('Ingrese el género del dios: ')
6          generos.append(sexo)
7          cant_hijos = int(input('Ingrese la cantidad de hijos del dios: '))
8          hijos.append(cant_hijos)
9      else:
10         print('El dios ya existe')
11
12  #Prueba
13  registrar_dios('Vishnu')
```

## Segundo ejemplo de funciones con listas



# Ejemplo 2:

## Problema:

Se quiere saber cuál es la mayor venta que se registró en un almacén entre todas las ventas registradas en un año, en el que se guarda el total de ventas de cada mes.

1. Análisis.

2. Diseño o modelado.

## Definición:

3. Implementación.

Se supone que las ventas del almacén registradas en un año se almacenan en una lista de números reales de 12 posiciones, una para cada mes del año de ventas que se quiere registrar. Así se pueden revisar todos los elementos de la lista para poder hallar la mayor venta.

# Ejemplo 2:

## 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 todas las ventas de los 12 meses.	lista_ventas [ ]	lista	{78000,..., 89500}
Salida			
Variable que el valor asociado a la mayor venta realizada.	mayor_venta	float	96000

# Ejemplo 2:

## Planteamiento

En la lista **ventas** se van a almacenar las ventas del almacén que corresponden a cada mes. Para poder hallar la mayor venta se deben seguir los siguientes pasos:

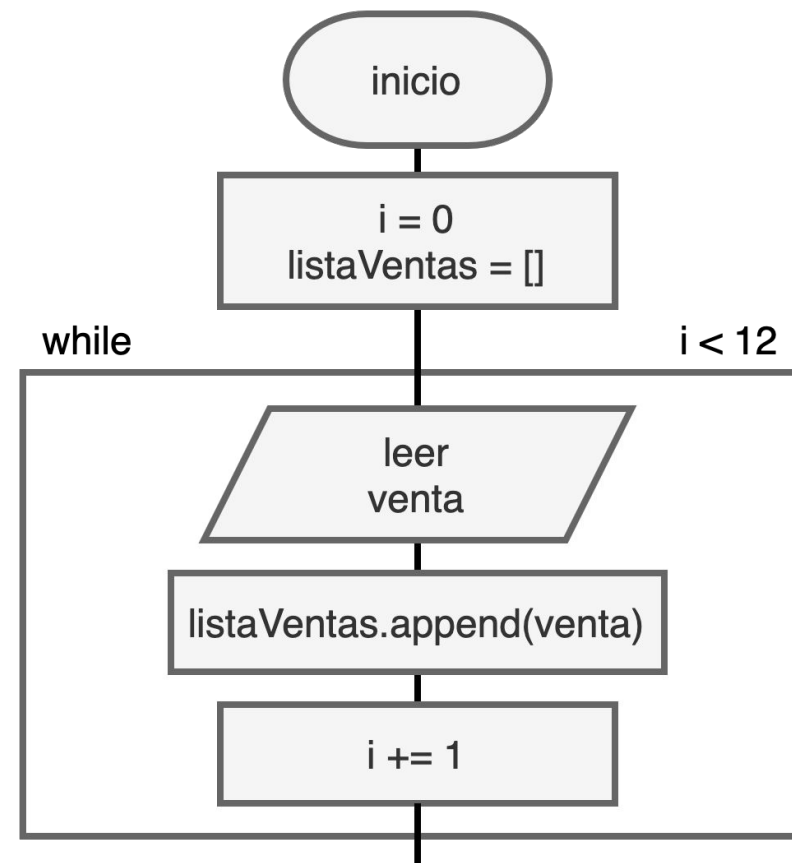
1. Análisis.
  2. Diseño o modelado.
  3. Implementación.
1. Declarar la lista de reales llamada **ventas**.
  2. Leer la lista de **ventas**.
  3. Calcular la mayor venta e imprimirla.

Para leer la lista se usa una **estructura iterativa con contador**, y otra para calcular la mayor venta. Para calcular la mayor venta se supone que al inicio la mayor venta fue en el primer mes, es decir, el primer elemento de la lista. Luego se recorre la lista para comparar con cada elemento de la lista, si se encuentra algún elemento que es mayor que la mayor venta, se cambia su valor por ese elemento. Por último, se imprime la mayor venta.

# Ejemplo 2:

**Diagrama de flujo:**  
Segmento del programa  
para leer las ventas de la lista.

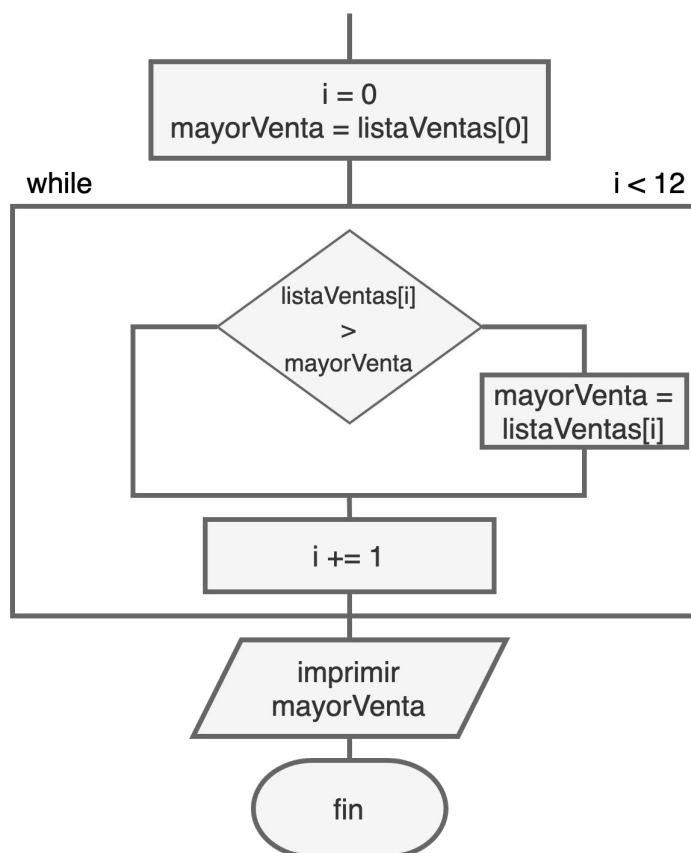
1. Análisis.
2. Diseño o modelado.
3. Implementación.



# Ejemplo 2:

**Diagrama de flujo:**  
Segmento del programa  
para hallar la mayor venta.

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



# Ejemplo 2:

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

```
                                mayor_venta()
1      i = 0
2      lista_ventas = []
3      while(i < 12):
4          venta = float(input('Ingrese la venta del mes '))
5          lista_ventas.append(venta)
6          i += 1
7      i = 0
8      mayor_venta = lista_ventas[0]
9      while(i < 12):
10         if(lista_ventas[i] > mayor_venta):
11             mayor_venta = listaVentas[i]
12         i += 1
13     print(f'La mayor venta en el año fue de: {mayor_venta}')
```



universidad  
cenfotec\_  
La U de la informática