



universidad  
cenfotec\_  
La U de la informática

# Principios de Programación.

Introducción a las listas.

# Objetivos

- Comprender el concepto de listas y usarlos para resolver problemas pequeños.
- Representar en un diagrama de flujo problemas que incluyan el uso de listas.
- Representar y usar en el lenguaje de programación Python listas unidimensionales.

# Introducción a listas (Ejercicio IMN)



Imagine que usted trabaja en el **Instituto Meteorológico Nacional de Costa Rica**, donde cada día necesita registrar la temperatura de San José al medio día y obtener el promedio al final de la semana.

Al analizar el problema usted decide que puede construir un programa en Python para solucionarlo. Se da a la tarea de crear un algoritmo que mediante una **estructura iterativa** obtenga la temperatura de cada día de la semana, la guarde en una variable **acumuladora** y calcule el promedio a partir de los datos obtenidos.

Como resultado, genera una tabla de entradas, el diagrama final del algoritmo y la implementación en código como se muestra a continuación:

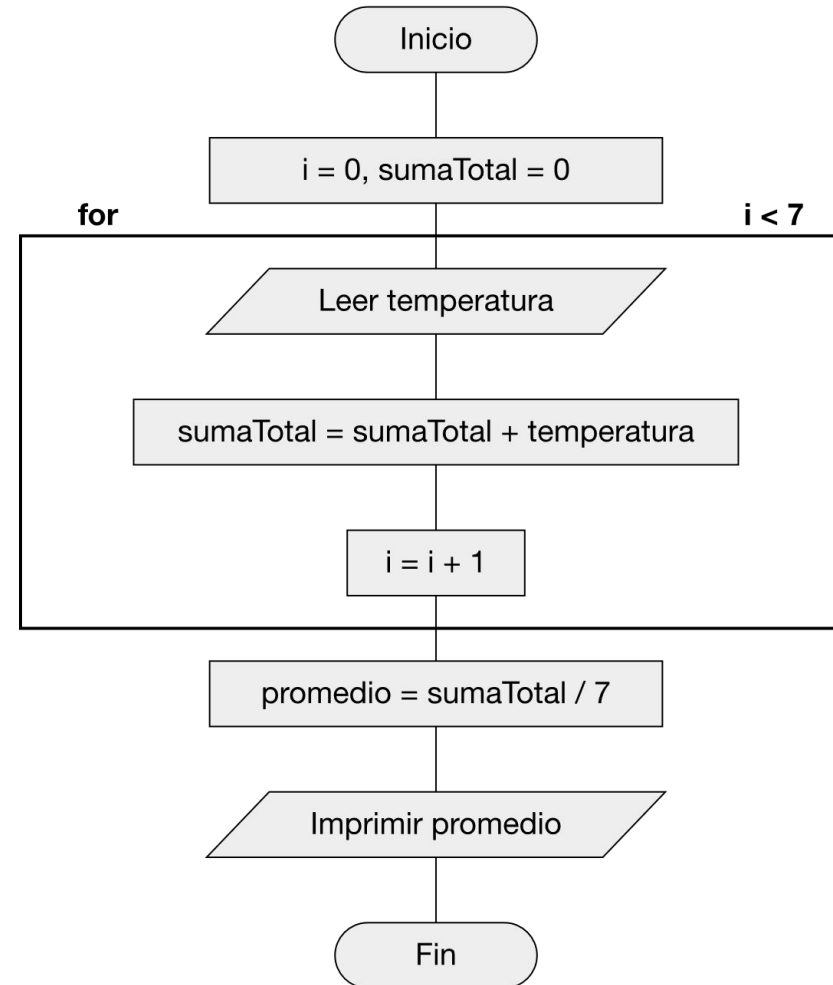
# Introducción a listas (Ejercicio IMN)

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

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
Variable que almacena la temperatura del día.	temperatura	float	23.6
Intermedias			
Variable que acumula la suma de las temperaturas.	suma_total	float	158.8
Salida			
Variable que almacena el promedio de las temperaturas.	promedio	float	22.57

# Introducción a listas (Ejercicio IMN)

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



# Introducción a listas (Ejercicio IMN)

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

## promedio\_temperatura.py

```
1  temperatura = 0
2  suma_total = 0
3  promedio = 0
4  i = 0
5  while(i < 7):
6      temperatura = float(input(f"Digite la temperatura del día {i + 1} "))
7      suma_total += temperatura
8      i += 1
9  promedio = suma_total / 7
10 print(f"La temperatura promedio de esta semana fue de: {promedio}")
```

# Introducción a listas (Ejercicio IMN)

Esta solución le permite obtener el **promedio** de 7 días, ¿qué sucede si en lugar de solo el promedio, necesita guardar cada una de las temperaturas de esos 7 días? ¿Será necesario declarar una variable por cada día?

Esta solución podría funcionar si implementa algunos condicionales dentro del ciclo, pero ¿qué sucede si en lugar de 7 días, necesita guardar la temperatura de todo el mes? ¿O de todo un año? Para implementar esta solución necesitaría 365 variables para almacenar la temperatura de cada día del año.

# Introducción a listas (Definición)

Para la construcción de algoritmos que requieran almacenar grupos de datos de un mismo tipo se utiliza la **estructura de datos** de tipo **Lista**.

- Los **listas** son variables que consisten en una **colección de valores**, representados bajo un mismo nombre e identificados por un índice.
- Los elementos dentro de una lista deben ser todos del **mismo tipo de dato** (números enteros, caracteres, números con punto flotante, valores booleanos, etc.).



# Introducción a listas (Definición)

Los listas son de gran utilidad cuando se requiere procesar **múltiples datos** que tienen **características comunes** y **están relacionados entre sí**.

*Ejemplo:* una lista que guarde las notas de los alumnos de un curso, una lista que guarde los nombres de un grupo de usuarios, una lista que guarde los precios de los artículos de un supermercado, etc.

# Introducción a listas (Ejercicio IMN)

Para el ejercicio anterior del **IMN**, es posible implementar una lista que almacene las temperaturas de cada día de la semana. Para lograrlo se creará una lista llamado *temperaturas*, el cual tendrá 7 posiciones, una para cada día de la semana.

Diariamente se irá agregando un nuevo valor al lista, por lo que al final de la semana, el lista contendrá todas las temperaturas de la semana. Como se muestra a continuación:

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
18.5	22	20.8	23	21.5	18.8	20.5

*temperaturas*

# Introducción a listas (Ejercicio IMN)

El lista **temperaturas** está representado con este nombre, pero es únicamente su identificador. Para almacenar y obtener datos de cada una de las posiciones es necesario utilizar un **índice** en conjunto con el **identificador** de la lista. El **índice** indica una posición o “casilla” dentro de la lista.

El valor del **índice** representa el desplazamiento que hay que hacer para acceder a determinada posición dentro de la lista, comenzando por el inicio. Es por esto que para acceder la primera “casilla” o primer elemento de la lista hay que desplazarse 0 posiciones, por lo tanto, el **primer elemento** se accede con el **índice 0**.

Por ejemplo, para obtener la temperatura del día lunes, es necesario el identificador y el índice dentro de paréntesis cuadrados así: **temperaturas[0]**. Para el día martes **temperaturas[1]**, para el día miércoles **temperaturas[2]**, así sucesivamente hasta **temperaturas[6]** que representa el día domingo.

	temperaturas[0]	temperaturas[1]	temperaturas[2]	temperaturas[3]	temperaturas[4]	temperaturas[5]	temperaturas[6]
<b>temperaturas</b>	18.5	22	20.8	23	21.5	18.8	20.5

# Introducción a listas (Ejercicio IMN)

Para recorrer una lista desde el inicio hasta el final, se puede usar una estructura iterativa con contador, en la cual el contador usualmente es el mismo **índice**. En estas estructuras iterativas en las que se usa el contador como **índice** de la lista, el contador debe inicializarse con el valor 0, ya que los listas **siempre** inician en la posición 0.

Es por esto que, aunque el lista *temperaturas* fue creado con 7 posiciones, el **índice** va de 0 a 6, como se mostró anteriormente.

	temperaturas[0]	temperaturas[1]	temperaturas[2]	temperaturas[3]	temperaturas[4]	temperaturas[5]	temperaturas[6]
<i>temperaturas</i>	18.5	22	20.8	23	21.5	18.8	20.5

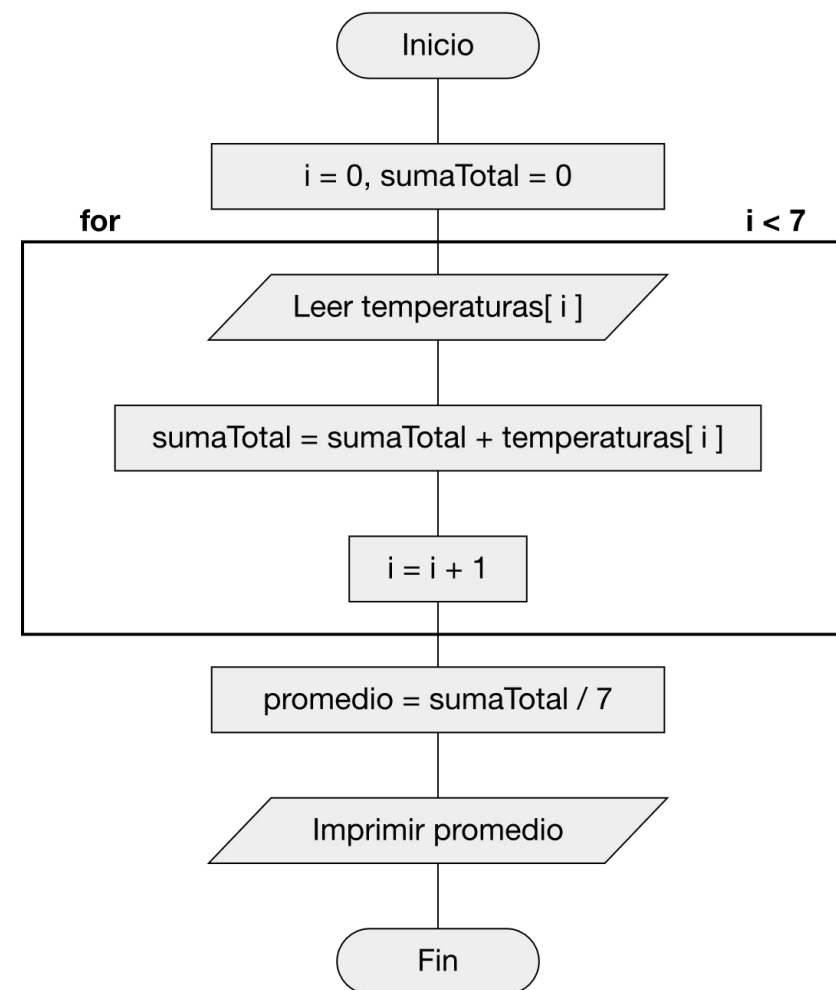
Usar el contador de una estructura iterativa como el **índice** de una lista, es tan comúnmente usado, que de ahí se deriva el uso de la letra **i** para denotar al contador en muchos lenguajes de programación.

# Introducción a listas (diagrama)

## Implementación

Una vez diseñado el lista que se va a utilizar, es necesario modificar el algoritmo para implementarlo.

Ahora el lista ***temperaturas*** se encargará de almacenar el dato de cada día en lugar de hacerlo temporalmente en la variable *temperatura*, como se puede ver en el siguiente diagrama de flujo:



# Introducción a listas (declaración)

Cuando se quiere **declarar una lista** en Python se inicializa la variable que representa al lista con dos paréntesis cuadrados ( `[]` ).

Los listas pueden ser de cualquier **tipo de dato**.

Ej. `temperaturas = []`

# Introducción a listas (inicialización)

Una vez declarado y reservado el espacio en memoria para el lista, se puede inicializar cada elemento de la lista con valores, mediante el **nombre** de la lista y la **posición** del elemento.

El proceso es similar a asignar un valor a una variable, la diferencia yace en que se está almacenando el valor en una **posición de la lista**.

Puede observar que, para cada uno de los elementos de la lista, el **índice** aumenta de uno en uno, similar a un contador. Es por esto que al trabajar con listas se utilizan estructuras iterativas.

```
temperatura = []  
  
temperatura.append(18.5) # Lunes  
temperatura.append(22) # Martes  
temperatura.append(20.8) # Miércoles  
temperatura.append(23) # Jueves  
temperatura.append(21.5) # Viernes  
temperatura.append(18.8) # Sábado  
temperatura.append(20.5) # Domingo
```

# Introducción a listas (estructuras iterativas)

El código anterior se puede modificar para que en lugar de asignar cada una de las temperaturas de la lista línea por línea, realizarlo mediante una **estructura iterativa**, como se muestra a continuación:

## lista\_promedio\_temperaturas.py

```
1  # lista para almacenar las temperaturas de la semana
2  temperaturas = [23.0, 21.0, 17.0, 29.0, 22.5, 18.0, 19.0]
3  i = 0
4  suma_total = 0
5  promedio = 0
6  while(i < 7):
7      suma_total += temperaturas[i]
8      i += 1
9  promedio = suma_total / 7
10 print(promedio)
```



# Introducción a listas (estructuras iterativas)

Dentro del ciclo, se puede, además, implementar las instrucciones necesarias para calcular el **promedio de temperaturas** durante la semana que se solicitó en el enunciado de la siguiente manera:

```
# lista para almacenar las temperaturas de la semana
temperaturas = [23.0, 21.0, 17.0, 29.0, 22.5, 18.0, 19.0]
i = 0
suma_total = 0
promedio = 0
while(i < 7):
    suma_total += temperaturas[i]
    i += 1
promedio = promedio = suma_total / 7
print(promedio)
```

# Introducción a listas (estructuras iterativas)

Se puede definir el tamaño de una lista en memoria de acuerdo con un valor dado por el usuario. Además, es posible obtener el valor del tamaño en cualquier momento, utilizando el atributo **length**. Este es de gran utilidad en las estructuras iterativas, ya que se puede utilizar en la condición del ciclo, como valor límite.

También puede ser utilizado al hacer otros cálculos como el promedio. En lugar de utilizar el valor 7, se utiliza **len(temperaturas)**.

```
# lista para almacenar las temperaturas de la semana
temperaturas = [23.0, 21.0, 17.0, 29.0, 22.5, 18.0, 19.0]
i = 0
suma_total = 0
promedio = 0
while(i < len(temperaturas)):
    suma_total += temperaturas[i]
    i += 1
promedio = suma_total / len(temperaturas)
print(f'El promedio de las temperaturas de la semana es: {promedio}')
```



universidad  
cenfotec\_  
La U de la informática