

# Estructuras de Datos y Algoritmos

## Programación en Python y C++

Rafael De Luna Loredó

Facultad de Ciencias, UASLP



# Índice

- |   |                   |    |                      |    |              |
|---|-------------------|----|----------------------|----|--------------|
| 1 | Introducción      | 6  | Arreglos y Matrices  | 11 | Ordenamiento |
| 2 | Conceptos Básicos | 7  | Funciones            | 12 | Búsqueda     |
| 3 | Operaciones       | 8  | Apuntadores          | 13 | Grafos       |
| 4 | Condicionales     | 9  | Estructuras de Datos | 14 | Cadenas      |
| 5 | Ciclos            | 10 | Objetos              | 15 | Referencias  |

**1** Introducción

## 2 Conceptos Básicos

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

# ¿Qué es un algoritmo?

Es una serie de pasos a seguir para solucionar un problema.  
Pero no solo eso, incluso en nuestra vida diaria usamos algoritmos sin saberlo, por ejemplo al ponernos los zapatos o vestirnos seguimos una serie de pasos para llegar a un resultado final

# ¿Como programar algoritmos?

Antes de empezar a tirar código hay que hacernos unas cuántas preguntas

- ¿sigue una serie de pasos?
- ¿son consecutivos?
- ¿que resultados puedo esperar?

## ¿Que lenguajes de programación usar?

Podemos hacer uso de cualquier lenguaje, todo dependerá de que tan cómodos nos sintamos con el lenguaje que vayamos a usar o estemos utilizando.

En este curso veremos los ejemplos en Python y C++ , C++ por ser el más utilizado en programación competitiva y Python por su sintaxis sencilla y su amplia utilización en la industria.

## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres  
Tipos de Datos

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

## ① Introducción

## ② Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres  
Tipos de Datos

## ③ Operaciones

## ④ Condicionales

## ⑤ Ciclos

## ⑥ Arreglos y Matrices

## ⑦ Funciones

## ⑧ Apuntadores

## ⑨ Estructuras de Datos

## ⑩ Objetos

## ⑪ Ordenamiento

## ⑫ Búsqueda

## ⑬ Grafos

## ⑭ Cadenas

## ⑮ Referencias



# Cabeceras

Son archivos que contienen las declaraciones de funciones y/o clases por lo que suelen tener código

# Bibliotecas

Una biblioteca o librería podríamos considerarla una colección de cabeceras, además de cabeceras incluye archivos de enlazado dinámico o estático.

# Espacio de nombres

Es un contenedor donde existen una o más identificadores para clases, funciones ó métodos contenidos en las cabeceras.

## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres

### Tipos de Datos

Númericos

Caracteres

Lógicos

Contenedores y  
colecciones

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres

### Tipos de Datos

Númericos

Caracteres

Lógicos

Contenedores y  
colecciones

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

# Enteros

- No aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 4 bytes de memoria y un valor máximo de  $\pm 2147483647$
- En Python varía el tamaño en memoria pero puede ser desde 28 bytes hasta 408 bytes o más

# Reales

- Aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 4 bytes de memoria y valores entre  $1.17549e-38$  y  $3.40282e+38$
- En Python varía el tamaño en memoria pero puede ser desde 24 bytes hasta 408 bytes o más

# Reales de doble precisión

- Aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 8 bytes de memoria y valores entre  $2.22507e-308$  y  $1.79769e+308$
- En Python el tipo float hace una implementación a bajo nivel del tipo **double** de C.



## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres

### Tipos de Datos

Númericos

Caracteres

Lógicos

Contenedores y  
colecciones

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

# Caracter

- Representan letras, simbolos o caracteres
- En C++ ocupan 1 byte de memoria y valores de  $\pm 127$
- En Python como tal no existe.

# Cadena de caracteres

- Representan letras, simbolos, caracteres y/o palabras
- Tanto en C++ como en Python el tamaño varía según el tamaño de la cadena
- En el caso de C++ existen dos tipos, un arreglo de tipo **char** y el tipo **str** a través de la biblioteca **cstring**

## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres

### Tipos de Datos

Númericos

Caracteres

Lógicos

Contenedores y  
colecciones

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

# Booleanos

- Representan solamente **True** o **False**, **1** o **0**
- Ocupan 1 byte de almacenamiento, por lo que se desperdicia mucho espacio de memoria

## 1 Introducción

## 2 Conceptos Básicos

Bibliotecas,  
cabeceras,  
espacio de  
nombres

### Tipos de Datos

Númericos  
Caracteres  
Lógicos

Contenedores y  
colecciones

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

# Diccionarios

- Se componen de una llave o clave y un valor
- Son elementos ordenados
- Pueden existir múltiples llaves
- Las llaves no se pueden repetir
- En C++ se llaman mapas, para usarlos hay que importar la cabecera *map*

# Listas

- Son una secuencia de elementos
- En Python pueden contener múltiples tipos de datos
- Para usarlos en C++ hay que importar la cabecera *list*



# Tuplas

- Son una secuencia de elementos
- Pueden contener múltiples tipos de datos
- En Python son inmutables, es decir no se pueden modificar o eliminar elementos
- Para usarlos en C++ hay que importar la cabecera *tuple*

# Conjuntos

- Son una secuencia de elementos no repetidos
- Hacen alusión a la definición matemática de conjuntos
- En C++ existen dos tipos, ordenados y no ordenados
- Para usarlos en C++ hay que importar las cabeceras `set` y `unordered_set` respectivamente

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

# Operaciones Matemáticas

- Tienen orden de precedencia, siendo el mismo que conocemos en matemáticas
- Van de izquierda a derecha
- Se realizan en pares
- Se pueden realizar entre diferentes tipos, todo dependerá de si se guardan o no en una variable
- Son las mismas que en matemáticas, suma, resta, multiplicación, división y módulo o residuo
- Para uso de funciones mas avanzadas habrá que hacer uso de cabeceras o bibliotecas creadas para dicho propósito

# Operaciones Lógicas

- Devuelven un valor booleano
- Van de izquierda a derecha
- Se realizan en pares
- Son algunas que ya conocemos en matemáticas, mayor que, menor que, igual a, diferente de, menor o igual, mayor o igual, conjunción y disyunción
- La conjunción y disyunción dependen de las otras

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

if else elif

Switch

Operador

ternario

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias

# Introducción

- Son un tipo de estructura
- Nos ayudan a tener control según valores o resultados esperados
- Evalúan operaciones lógicas
- Pueden contener o concatenar múltiples estructuras

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

if else elif

Switch

Operador

ternario

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias



## if

Nos ayuda a evaluar si una condición existe, en caso de que se cumpla, se ejecuta el código que contiene

# else

Es la contraparte de **if**, en caso de que la condición de **if** no se cumpla, ejecuta el código que contiene

## elif

Es una combinación de **else** con **if**, hace una segunda evaluación de otro posible resultado esperado

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

if else elif

Switch

Operador

ternario

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias

# Switch

- En Python no existe pero se puede hacer con diccionarios o ecandenando múltiples **if else**
- Es una estructura que evalúa múltiples casos posibles en base al posible valor de una variable
- También puede contener otras estructuras

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

if else elif

Switch

Operador

ternario

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias

# Operador ternario

- El funcionamiento es como el de un **if**, en Python de hecho es un **if**
- Sirve para comparaciones simples, donde no necesitamos ejecutar mucho código

- 1 Introducción
- 2 Conceptos Básicos
- 3 Operaciones
- 4 Condicionales
- 5 Ciclos
- 6 Arreglos y Matrices
- 7 Funciones
- 8 Apuntadores
- 9 Estructuras de Datos
- 10 Objetos
- 11 Ordenamiento
- 12 Búsqueda
- 13 Grafos
- 14 Cadenas
- 15 Referencias



# Introducción

- Son un tipo de estructura
- Nos ayudan a ejecutar un código múltiples veces
- Ayudan a reducir código

## Estructuras

## 10 Objetos

## 11 Ordenamiento

## 12 Búsqueda

## 13 Grafos

## 14 Cadenas

## 15 Referencias

## 1 Introducción

## 2 Conceptos Básicos

## 3 Operaciones

## 4 Condicionales

## 5 Ciclos

## 6 Arreglos y Matrices

## 7 Funciones

## 8 Apuntadores

## 9 Estructuras de Datos

# for

- Se define un contador
- El contador cambia sin necesidad de declarar dicho cambio
- Hay que definir un tope o límite
- Hay que definir como avanza el contador

# while

- Se ejecuta mientras dada una condición, dicha condición exista o se cumpla
- No se define un límite, mientras la condición exista, se seguirá ejecutando
- Necesitamos crear una manera de salir, por si la condición sigue existiendo
- Podría no ejecutarse si la condición no existe antes de hacer la evaluación

# do while

- Es muy parecido a **while** con la diferencia de que primero ejecuta y después comprueba si dicha condición existe
- No se define un límite, mientras la condición exista, se seguirá ejecutando
- Necesitamos crear una manera de salir, por si la condición sigue existiendo
- Se ejecuta al menos una vez
- En Python no existe

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

Arreglos  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y  
Matrices

Arreglos

Funcionamiento

Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias

# Introducción

- Son de un solo tipo
- Puede contener múltiples valores
- Es unidimensional
- Su espacio en memoria varía según el tipo de dato y los valores que pueda tener
- En Python como tal no existen, se usan las listas en su lugar
- El índice siempre empieza en cero, puede variar en algunos lenguajes, pero suele ser una regla general



① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

Arreglos  
Funcionamiento  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

# Almacenamiento en memoria

- El valor nos indica el contenido del arreglo en determinada posición
- El índice nos indica la posición en el arreglo
- La dirección de memoria donde se encuentra almacenado dicho valor

Valor	Índice	Dirección
325	0	0x61fee8
400	1	0x61feec
78	2	0x61fef0

# Inserción en C++

- Hay que conocer el índice donde queremos el valor
- No es necesario insertarlos en orden
- No se puede tener un índice mayor al tamaño del arreglo

1

---

<sup>1</sup>Aplica solo para C++

# Inserción en Python

- Las listas en Python tienen un método para agregar valores
- No es necesario conocer el índice
- No es necesario conocer el tamaño

2

---

<sup>2</sup>Aplica solo para C++

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

Arreglos  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones  
Introducción

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones  
Introducción

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntes  
Introducción

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias



① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntes  
Introducción

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos  
Introducción

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos  
Introducción

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

Introducción

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

Introducción

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y  
Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de  
Datos

⑩ Objetos

⑪ Ordenamiento  
Introducción

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de Datos

⑩ Objetos

⑪ Ordenamiento  
Introducción

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas

⑮ Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y Matrices

7 Funciones

8 Apuntadores

9 Estructuras de Datos

10 Objetos

11 Ordenamiento

12 Búsqueda  
Introducción

13 Grafos

14 Cadenas

15 Referencias



1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y Matrices

7 Funciones

8 Apuntadores

9 Estructuras de Datos

10 Objetos

11 Ordenamiento

12 Búsqueda  
Introducción

13 Grafos

14 Cadenas

15 Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos  
Introducción

14 Cadenas

15 Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y  
Matrices

7 Funciones

8 Apuntadores

9 Estructuras de  
Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos  
Introducción

14 Cadenas

15 Referencias

① Introducción

② Conceptos Básicos

③ Operaciones

④ Condicionales

⑤ Ciclos

⑥ Arreglos y Matrices

⑦ Funciones

⑧ Apuntadores

⑨ Estructuras de Datos

⑩ Objetos

⑪ Ordenamiento

⑫ Búsqueda

⑬ Grafos

⑭ Cadenas  
Introducción

⑮ Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y Matrices

7 Funciones

8 Apuntadores

9 Estructuras de Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas  
Introducción

15 Referencias

1 Introducción

2 Conceptos Básicos

3 Operaciones

4 Condicionales

5 Ciclos

6 Arreglos y Matrices

7 Funciones

8 Apuntadores

9 Estructuras de Datos

10 Objetos

11 Ordenamiento

12 Búsqueda

13 Grafos

14 Cadenas

15 Referencias

# Referencias I

- [1] Bhasin, H.  
*Python Basics*, 3 ed.  
David Pallai, Mercury Learning and Information, 2019.
- [2] CAIRÓ, O., and Guardati, S.  
*Algorithms*, 3 ed.  
McGraw-Hill Interamericana, 2006.
- [3] Deitel, P., and Deitel, H.  
*C++ Como Programar*, 9 ed.  
Pearson Educación de México, 2014.
- [4] Downey, B., A.  
*Think Python*, 2 ed.  
O'Reilly, 2016.

## Referencias II

- [5] Jaworski, M., and Ziadé, T.  
*Expert Python Programming*, 3 ed.  
Packt Publishing, 2019.
- [6] Laakmann, M., G.  
*Cracking the Coding Interview*, 6 ed.  
CarrerCup, 2016.
- [7] Matthes, E.  
*Python Crash Course*, 2 ed.  
No Star Press, 2019.
- [8] Ramalho, L.  
*Fluent Python*, 1 ed.  
O'Reilly, 2014.



## Referencias III

- [9] Reek, K.  
*Pointers On C*.  
Addison-Wesley Longman, 1997.
- [10] Sedgewick, R., and Wayne, K.  
*Algorithms*, 4 ed.  
Pearson Education, 2011.
- [11] Stroustrup, B.  
*The C++ Programming Language*, 4 ed.  
Pearson Education, 2013.
- [12] Stroustrup, B.  
*Principles and Practice Using C++*, 2 ed.  
Pearson Education, 2014.

## Referencias IV

[13] unknown.  
Cplusplus.

*Danke!*