



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Adrian Ulises Mercado Martinez

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 13

*No de Práctica(s):* 10

*Integrante(s):* Monroy Salazar Diego Gustavo

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* Brigada 9

*Semestre:* 2020-2

*Fecha de entrega:* 07/06/2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Introducción

Python es un lenguaje interpretado y ofrece muchas facilidades al momento de programar, en ésta práctica vemos varias de ellas.

## Desarrollo

1. Vimos como importar bibliotecas y parte de estas

```
1 #import math
2 #x=math.cos(math.pi)
3 ##from math import *#Para importar absoltuamente todo
4 from math import cos, pi##Para importar solo una parte
5 x=cos(pi)
6 print(x)
7
8
```

2. Vimos como las estructura while funciona exactamente igual a como la vimos en C

```
def fact(n):
    i=2
    temp=1
    while i<=n:
        temp=temp*i
        i=i+1
    return temp

if __name__=="__main__":
    a=int(input("Ingresa un número: "))
    print(fact(a))
```

3. Hicimos ciclos for que cambian bastante a como los vimos en C, también vimos cómo funcionan en listas y diccionarios

```

...
For de listas
"""

def forlist():
    for x in[1, 2, 3, 4, 5]:
        print(x)
    for x in["Uno", "Dos", "Tres", "Cuatro", "Cinco"]:
        print(x)

"""
For para rangos
"""

def forrange():
    for x in range(5):
        print(x)

    for y in range(-3, 3):
        print(y)

    for z in range(-4, 2, 2):
        print(z)

    for i in range(5, 0, -1):
        print(i)

```

```

...
For para diccionarios
Un diccionario es va a tener
asociados valores y claves
"""

def fordic():
    diccionario={'manzana':1, 'pera':3, "uva":10}#izq clave der valor
    for clave, valor in diccionario.items():
        print(clave, "=", valor)

    for clave in diccionario.keys():
        print(clave)
    for valor in diccionario.values():
        print(valor)
    for idx, x in enumerate(diccionario):
        print("El indice {} del elemento {}".format(idx, x))

```

Esto es probablemente una de las cosas que pueden ser confusas al principio en comparación con C.

4. Creamos una pila la cual es más sencilla de tratar que en C

```
def insertar(pila, dato):
    pila.append(dato)#append es agregar al final
    print(pila)
def borrar(pila):
    dato= pila.pop()#pop elimina al final
    return dato
def imprimir_pila(pila):
    for x in range(len(pila)-1, -1, -1):
        print(pila[x])

def main():
    pila=[0]
    insertar(pila, 1)
    insertar(pila, 2)
    imprimir_pila(pila)
    borrar(pila)
    imprimir_pila(pila)

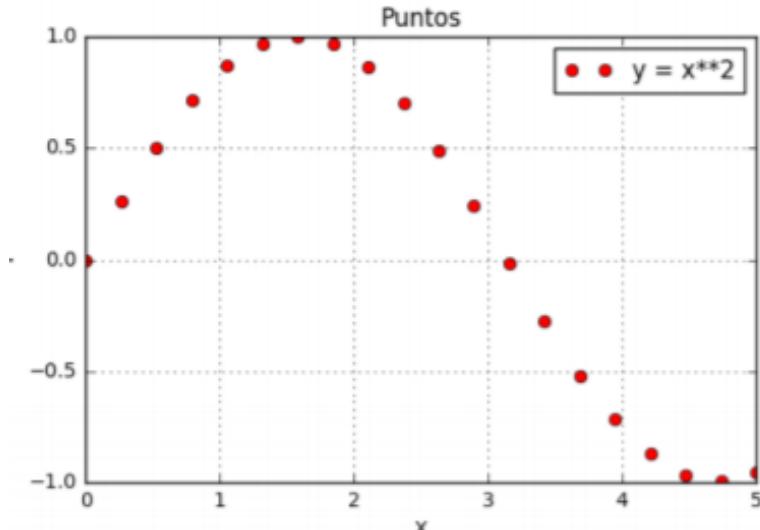
if __name__=="__main__":
    main()
```

En ella usamos funciones que ya vienen predefinidas en Python y que facilitan bastante el trabajo.

5. Finalmente aprendimos como hacer gráficas con un ejemplo de la función seno. Una de las herramientas más importantes de Python.

```
import matplotlib.pyplot as plt
from math import sin
from mpl_toolkits.mplot3d import Axes3D
x=linspace(0, 5, 20)
fig, ax=plt.subplots(facecolor='w', edgecolor='k')
ax.plot(x, sin(x), marker="o", color="r", linestyle='None')
ax.grid(True)
ax.set_xlabel('X')#Etiqueta eje x
ax.set_ylabel('Y')#Etiqueta eje y
ax.grid(True)
ax.legend(["y=x**2"])
plt.title('Puntos')
plt.show()

fig.savefig("grafica.png") ##guarda la grafica
```



## Conclusiones

Python definitivamente es un lenguaje de programación bastante práctico, ya que su relativa facilidad hace que se puedan hacer aplicaciones que de otra forma tardarían mucho tiempo en desarrollarse en C.