

## Material de apoyo – Planteamiento 7

### Sesión #7 Uso y bibliotecas en Python (LIBRERIAS PANDAS, NUMPY)

**los siguientes temas son:** Uso de funciones y bibliotecas en Python (pandas y Numpy).

#### Series

```
import pandas as pd
s = pd.Series([1,2,3,5,np.nan,6,8])
print(s)
```

#### Dataframe con lista nombres, edad y ciudad.

```
import pandas as pd
data = {'Nombre': ['Fabian', 'Ana', 'Patricia', 'Linda'],
        'Edad': [28, 24, 35, 32],
        'Ciudad': ['Bogota', 'Pereira', 'Bucaramanga', 'Cali']}
df = pd.DataFrame(data)
print(df)
```

#### Se realiza la muestra de un rango de fechas

```
import pandas as pd
fechas = pd.date_range("2024-01-01", periods=6, freq="M")
print(fechas)
```

- **Metodo to\_string()**

```
import pandas as pd
df = pd.read_csv('datos.csv')
print(df.to_string())
```

- **Crear el DataFrame con valores aleatorios**

```
import pandas as pd
df = pd.DataFrame(np.random.rand(6,5),index=fechas, columns=list("ABCDE"))
df
```

DataFrame con 100 filas y 8 columnas generadas aleatoriamente:

```
df1= pd.DataFrame(np.random.rand(100,8))
df1
```

- **Mostrar las primeras filas del DataFrame**

```
import pandas as pd
df1= pd.DataFrame(np.random.rand(100,8))
df1.head(10)
```

Para mostrar solo 3 ultimas filas

```
df1.tail(3)
```

- **Describir el DataFrame:**

```
import pandas as pd
df1= pd.DataFrame(np.random.rand(100,8))
df1.describe()
```

- **Filas máximas**

```
import pandas as pd
print(pd.options.display.max_rows)
```

### **Aumente el número máximo de filas para mostrar todo el DataFrame:**

```
import pandas as pd
pd.options.display.max_rows = 9999
df = pd.read_csv('data.csv')
print(df)
```

- **Leer archivos CSV**

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df)
```

- **Los pandas leen JSON**

```
import pandas as pd
df = pd.read_json('data.json')
print(df.to_string())
```

- **Diccionario en formato JSON**

```
import pandas as pd
data = {
    "Duracion":{
        "0":60,
        "1":60,
        "2":60,
        "3":45,
        "4":45,
        "5":60
    },
}
```

```
"Pulso":{
  "0":110,
  "1":117,
  "2":103,
  "3":109,
  "4":117,
  "5":102
},
"Maxpulso":{
  "0":130,
  "1":145,
  "2":135,
  "3":175,
  "4":148,
  "5":127
},
"Calorias":{
  "0":409,
  "1":479,
  "2":340,
  "3":282,
  "4":406,
  "5":300
}
}
```

```
df = pd.DataFrame(data)
print(df)
```

### Visualización de los datos

- **Método head**

```
import pandas as pd
df= pd.read_csv('data.csv')
print(df.head(10))
```

### Librería NumPy

- **Creación de Arrays desde lista.**

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr)
```

- **Crear un array con ceros**

```
import numpy as np
arr = np.zeros((3, 4))
print(arr)
```

- **Crear un array con unos**

```
import numpy as np
arr = np.ones((2, 3))
print(arr)
```

- **Crear un array con un rango de valores**

```
import numpy as np
arr = np.arange(10)
print(arr)
```

- **Crear un array con valores espaciados uniformemente:**

```
import numpy as np
arr = np.linspace(0, 1, 5)
print(arr)
```

- **Operaciones básicas con Arrays**

```
import numpy as np
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
print(a + b)
print(a - b)
print(a * b)
print(a / b)
```

- **Operaciones con escalares:**

```
import numpy as np
print(a * 2) # Multiplicación por un escalar
print(b + 3) # Suma de un escalar
```

- **Operaciones de agregación**

```
import numpy as np
print(a.sum())
print(b.mean())
print(a.max())
print(a.min())
```

- **Manipulación de Arrays**

```
import numpy as np
arr = np.arange(6)
arr = arr.reshape((2, 3))
print(arr)
```

- **Concatenación de arrays**

```
import numpy as np
a = np.array([1, 2, 3])
```

```
b = np.array([4, 5, 6])  
arr = np.concatenate((a, b))  
print(arr)
```

```
import numpy as np  
a = np.arange(15).reshape(3,5).dtype.name  
print(a)
```

```
import numpy as np  
a = np.arange(15).reshape(3,5).ndim  
print(a)
```

```
import numpy as np  
a = np.arange(15).reshape(3, 5)  
print(type(a))
```