

**CARRERA: COMPUTACIÓN**

**PRÁCTICA: 03**

**EQUIPO: COMPUTADOR**

**CÁTEDRA O MATERIA  
RELACIONADA**

**REVISIÓN N°: 1**

**Responsable /Equipo.:**

Simulación

**EDICIÓN: 1**

**Accesorios:**

COMPUTADOR / INTERNET

**DOCENTE:**

Ing. Diego Quisi

**NÚMERO DE ESTUDIANTES  
POR EQUIPO O PRÁCTICA:**

**Fecha:** 14/11/2021

**Tema:**

Simulación

**Realizado por:**

Christian Zhirzhan

**Objetivo:**

**Enunciado**

Utilizar Matplot y Plotly para realizar las graficas.

1. Con los datos de Fifa, organizar a los jugadores de acuerdo al peso en las siguientes escalas y generar un cuadro tipo PIE  
Debajo 125 Lbs.  
125-150  
150-175  
175 o superior
2. Generar un grafico de barras (histograma) de acuerdo a su habilidad (Overall) en base a los siguientes segmentos contando el número de jugadores  
40  
50  
60  
70  
80  
90

3. Investigar alguna herramienta o software que permita generar reportes con Python utilizando CodeLabs.

### Desarrollo/Procedimiento:

Utilizar Matplot y Plotly para realizar las graficas.

Librerias necesarias

```
1 #importar las librerias necesarias
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pandas as pd
5 import io
6 from google.colab import files
7 import plotly.graph_objects as go
```

Carga del dataset

```
1 #Trabajar con otro tipo de datos Fifa
2 fifa = pd.read_csv(io.StringIO(archivo_fifa['fifa_datos.csv'].decode('utf-8')))
3 #imprimir los primeros 5 datos del archivo
4 fifa.describe()
```

1. Con los datos de Fifa, organizar a los jugadores de acuerdo al peso en las siguientes escalas y generar un cuadro tipo PIE  
Debajo 125 Lbs.  
125-150  
150-175  
175 o superior

```
1 # Filtrar la columna de los pesos
2
3 pesos = fifa['Weight']
4
5 overall.head(40)
```

```
# convertir a numericos los datos

pesos = pesos.replace('lbs', '', regex=True).astype(float)
pesos.head(5)
```

```
2
3 menos125 = [] # la lista que contendrá los elementos filtrados <125
4
5 for i in range(len(pesos)):
6     if pesos[i] < 125.0:
7         menos125.append(int(pesos[i]))
8
9
10 print(menos125) # mostramos el resultado
11
12
13 to125150 = [] # la lista que contendrá los elementos filtrados entre 125 y 150
14
15 for i in range(len(pesos)):
16     if pesos[i] >= 125.0 and pesos[i] <= 150.0:
17         to125150.append(int(pesos[i]))
18
19 print(to125150) # mostramos el resultado
20
21
22 to150175 = [] # la lista que contendrá los elementos filtrados entre 150 a 175
23
24 for i in range(len(pesos)):
25     if pesos[i] >= 150.0 and pesos[i] <= 175.0:
26         to150175.append(int(pesos[i]))
27
28 print(to150175) # mostramos el resultado
29
30
31 mayor175 = [] # la lista que contendrá los elementos filtrados > 175
32
33 for i in range(len(pesos)):
34     if pesos[i] >= 175.0:
35         mayor175.append(int(pesos[i]))
36
37 print(mayor175) # mostramos el resultado
```

```

1 menos125= len(menos125)
2 to125150 =len(to125150)
3 to150175 = len(to150175)
4 mayor175 = len(mayor175)

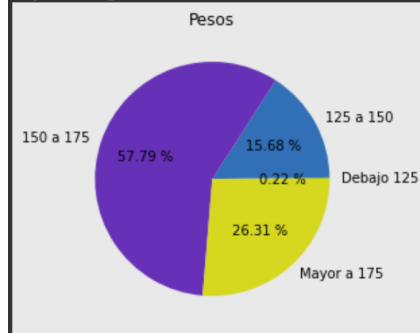
1 # Generar un grafico de cual es su pie diestro
2 print('Debajo de 125 lbs = ',menos125)
3 print('Entre 125 y 150 lbs = ',to125150)
4 print('Entre 150 y 175 lbs = ',to150175)
5 print('Mayor o igual a 175 lbs = ',mayor175)
6
7 plt.figure()
8
9 etiquetas = ['Debajo 125', '125 a 150', '150 a 175', 'Mayor a 175']
10 colores = ['#A2C736', '#367BC7', '#7036C7','#E9EC21']
11 plt.pie([menos125, to125150, to150175, mayor175], labels=etiquetas, colors=colores, autopct='%.2f %%')
12 plt.title('Pesos')
13 plt.show()

```

```

Debajo de 125 lbs = 41
Entre 125 y 150 lbs = 2950
Entre 150 y 175 lbs = 10876
Mayor o igual a 175 lbs = 4952

```



2. Generar un grafico de barras (histograma) de acuerdo a su habilidad (Overall) en base a los siguientes segmentos contando el número de jugadores

40  
50  
60  
70  
80  
90

```
1 overall = fifa['Overall']
```

```
overs=[]
j=1
for i in overall:
    if i==40:
        overs.append(i)

    if i==50:
        overs.append(i)

    if i==60:
        overs.append(i)

    if i==70:
        overs.append(i)

    if i==80:
        overs.append(i)

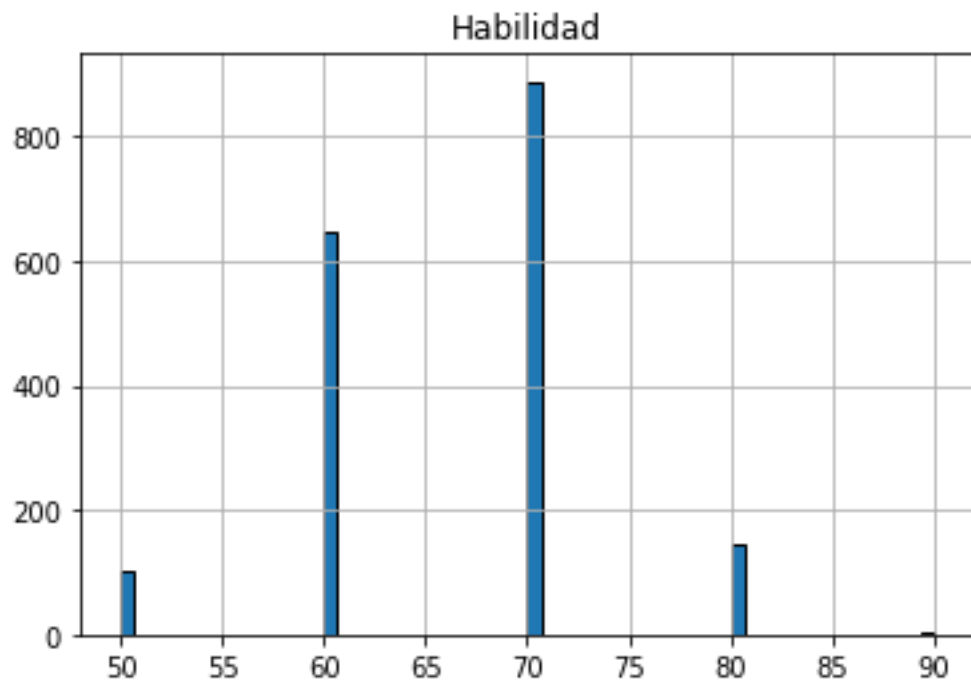
    if i==90:
        overs.append(i)
```

```
from random import random
import matplotlib.pyplot as plt

#Datos aleatorios para el ejemplo

plt.title('Habilidad')
plt.hist(overs, bins=60, alpha=1, edgecolor = 'black', linewidth=1)
plt.grid(True)
plt.show()
plt.clf()
```

No existen jugadores con una habilidad igual a 40.



3. Investigar alguna herramienta o software que permita generar reportes con Python utilizando CodeLabs.

### **Xmlgraphics**

Apache™ FOP (Procesador de objetos de formato) es un formateador de impresión impulsado por objetos de formato XSL (XSL-FO) y un formateador independiente de salida. Es una aplicación Java que lee un árbol de objetos de formato (FO) y procesa las páginas resultantes en una salida especificada. Los formatos de salida admitidos actualmente incluyen PDF, PS, PCL, AFP, XML (representación de árbol de área), Print, AWT y PNG y, en menor medida, RTF y TXT. El objetivo de salida principal es PDF. (The Apache™ FOP Project, n.d.)

### **Bibliografía**

*The Apache™ FOP Project.* (s.f.). Obtenido de The Apache™ FOP Project:  
<http://xmlgraphics.apache.org/fop/>