

Ejercicio en Clase

1. Con los datos de Fifa, organizar a los jugadores de acuerdo al peso en las siguientes escalas y generar un cuadro tipo PIE

```
import matplotlib.pyplot as plt
import pandas as pd
fifa = pd.read_csv("fifa_datos.csv")

menor = fifa.loc[fifa['Weight'].between(0, 125)].count()[0]

veintecinco = fifa.loc[fifa['Weight'].between(125,150)].count()[0]

cincuenta = fifa.loc[fifa['Weight'].between(151,175)].count()[0]

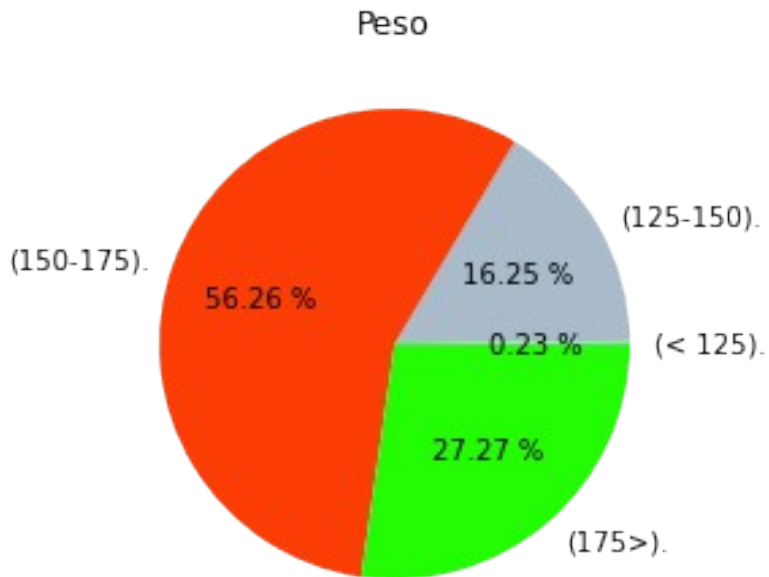
setentaycinco = fifa.loc[fifa['Weight'].between(175,500) ].count()[0]

print("< 125) = ", menor)
print("(125-150) = ", veintecinco)
print("(150-175) = ", cincuenta)
print("> 175) = ", setentaycinco)

plt.figure()

etiquetas = ['(< 125).', '(125-150).', '(150-175).', '(175>).']
colores = ['#abcdef', '#aabbcc', '#FC3D05', '#24FC05']
plt.pie([menor, veintecinco, cincuenta, setentaycinco],
labels=etiquetas, colors=colores, autopct='%.2f %%')
plt.title('Peso')
plt.show()

(< 125) = 41
(125-150) = 2950
(150-175) = 10216
(> 175) = 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

```
count = 0
count1 = 0
count2 = 0
count3 = 0
count4 = 0
count5 = 0

resultado = []
datos = []
aux = []
array = []
array = fifa['Overall']

for i in array:
    if i == 40:
        aux.append(i)
        count += 1

    if i == 50:
        aux.append(i)
        count1 += 1

    if i == 60:
        aux.append(i)
        count2 += 1
```

```

if i == 70:
    aux.append(i)
    count3 += 1

if i == 80:
    aux.append(i)
    count4 += 1

if i == 90:
    aux.append(i)
    count5 += 1

resultado = ['Variable (40) = ', count, 'Variable (50) = ', count1,
'Variable (60) = ', count2, 'Variable (70) = ', count3, 'Variable (80) = ',
count4, 'Variable (90) = ', count5]

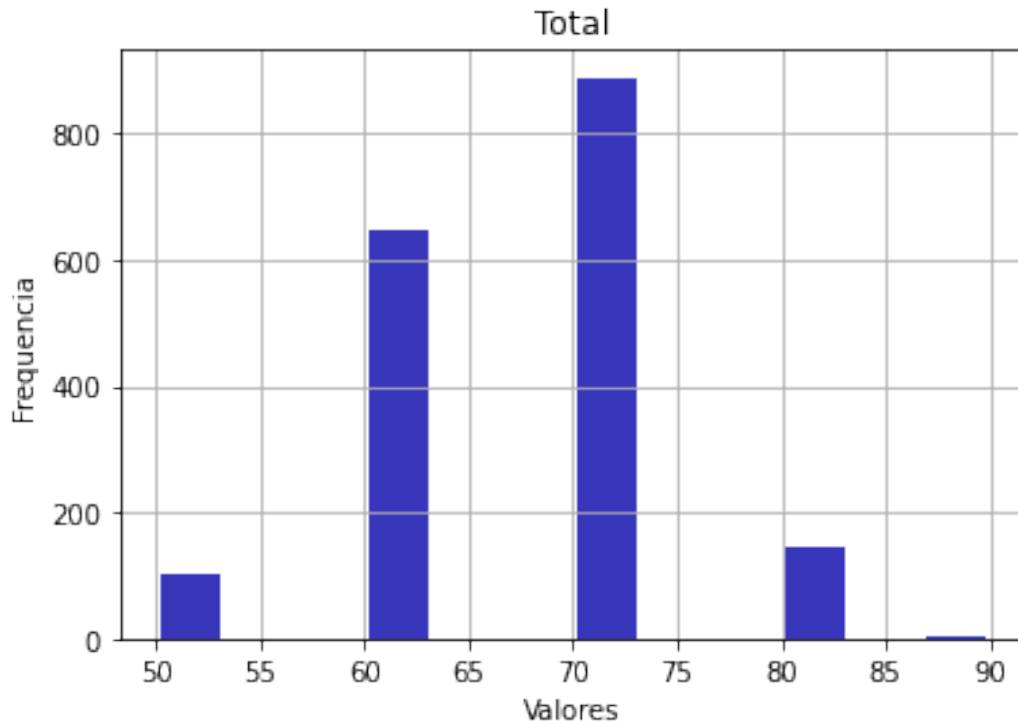
print (resultado)
print ("")

n, bins, patches = plt.hist(x=aux, bins=12, color='#0504aa',
                           alpha=0.8, rwidth=0.85)
plt.grid(alpha=1)
plt.xlabel('Valores')
plt.ylabel('Frequencia')
plt.title('Total')

['Variable (40) = ', 0, 'Variable (50) = ', 103, 'Variable (60) = ',
649, 'Variable (70) = ', 889, 'Variable (80) = ', 143, 'Variable (90)
= ', 5]

Text(0.5, 1.0, 'Total')

```



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

CodeLab

cuyo objetivo es ayudar a los usuarios de una plataforma informática sin servidor de Google Cloud a modernizarse a productos en la nube más nuevos u otras plataformas informáticas sin servidor. Cada migración de modernización tiene como objetivo presentar un video, un laboratorio de código (tutorial práctico y a su propio ritmo) y ejemplos de código. El contenido inicialmente se enfoca en App Engine y los primeros usuarios de la nube de Google; de esta manera mostrar a los desarrolladores cómo realizar migraciones individuales que pueden aplicar para modernizar sus aplicaciones para el últimos tiempos de ejecución, con diferentes repositorios administrando las muestras de esos codelabs

1. Migrar de un servicio de App Engine heredado a un producto equivalente en la nube
2. Migrar a un nuevo producto, servicio o biblioteca de cliente de API en la nube
3. Migrar a otra plataforma informática sin servidor de Google Cloud o más reciente

Requisitos:

- Una cuenta de Google Workspace/G Suite
- Un proyecto de Google Cloud (GCP) con una cuenta de facturación activa
- Familiaridad con los comandos de terminal/shell del sistema operativo
- Familiaridad con el desarrollo y la implementación de aplicaciones de Python 2 en App Engine
- Habilidades generales en Python 2 y 3

App Engine, Cloud Functions y Cloud Run no son servicios gratuitos. Si bien es posible que no haya necesitado habilitar la facturación en los primeros días de App Engine, ahora todas las aplicaciones requieren una cuenta de facturación activa respaldada por un instrumento financiero (generalmente una tarjeta de crédito). No se preocupe, App Engine (y otros productos de GCP) todavía tienen un nivel "Siempre gratuito" y, siempre que se mantenga dentro de esos límites, no incurrirá en ningún cargo.

Además, la implementación en plataformas sin servidor de GCP genera costos menores de compilación y almacenamiento. Cloud Build tiene su propia cuota gratuita al igual que Cloud Storage. Para una mayor transparencia, Cloud Build crea la imagen de su aplicación que luego se envía a Cloud Container Registry, o Artifact Registry, su sucesor; el almacenamiento de esa imagen utiliza parte de esa cuota (almacenamiento en la nube), al igual que la salida de la red al transferir esa imagen al servicio que está implementando. Sin embargo, es posible que viva en una región que no tenga un nivel tan gratuito, así que tenga en cuenta su uso de almacenamiento para minimizar los costos potenciales.