(https://colab.research.google.com/github/GabrielChuchuca/SIMULACION/blob/main/DEBERES_1INTERCICLO/DeberDatosFIFA.ipynb)

Tutorial de Matplot para graficas y de Reportes pagermill

A continuación se detalla un pequeño tutorial de como utilizar matplotlib y pagermill para la generacion de graficas y reportes respectivamente, este tutorial se basa en tres librerias:

- Matplotlib
- Numpy
- Pandas

Al finalizar el estudiante estará en la capacidad de generar graficas y enviar parametros para la realización de reportes utilizando Notebook. Además permite la lectura de archivos .csv y de diferentes tipos de graficos.

In [ ]:
```
#importar las librerias necesarias
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import math

# Cómo conectar el cuaderno a la la Unidad de Google Drive
import os

# Importamos el paquete propio de Google Colab para acceder a la Unidad
from google.colab import drive
drive.mount('/content/drive/')
```

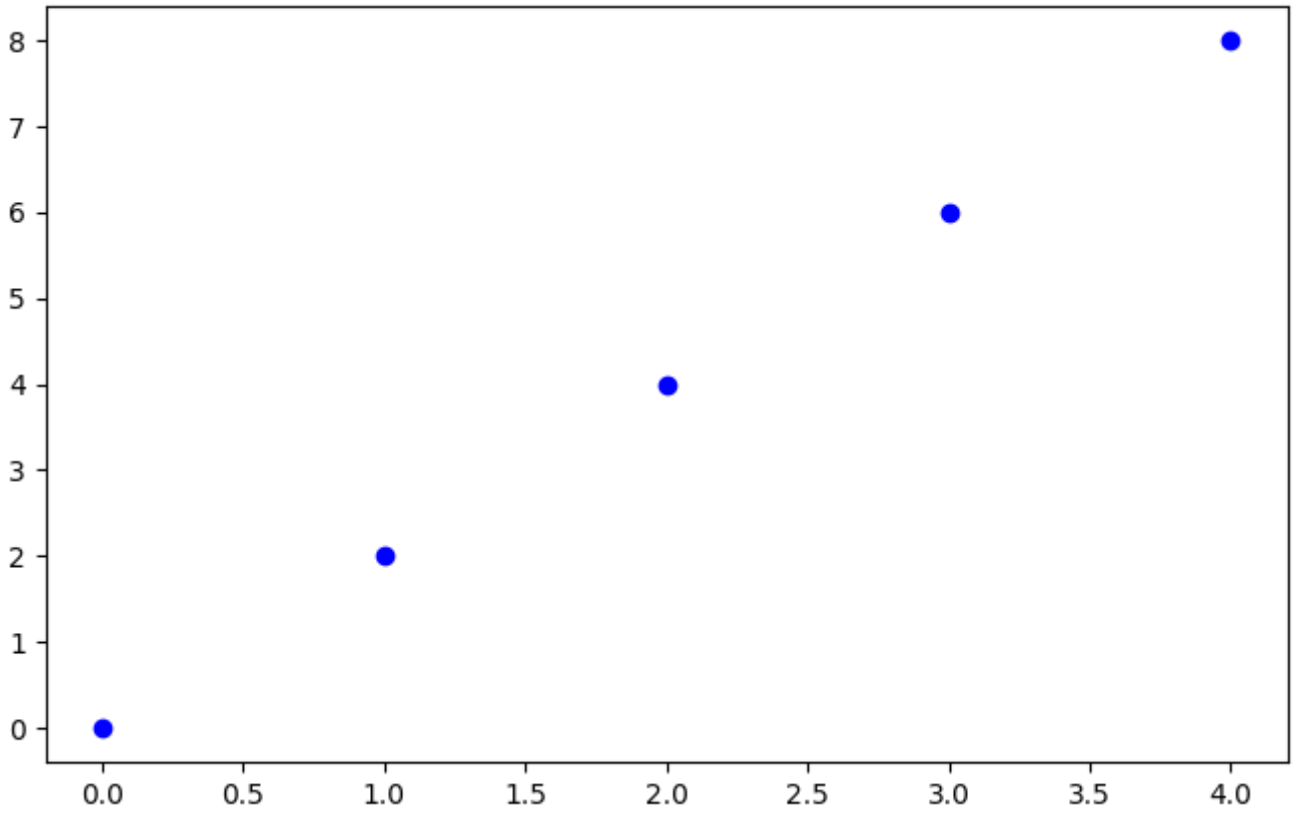
```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive.mount("/content/drive/",
force_remount=True).
```

Construccion de un grafico basico

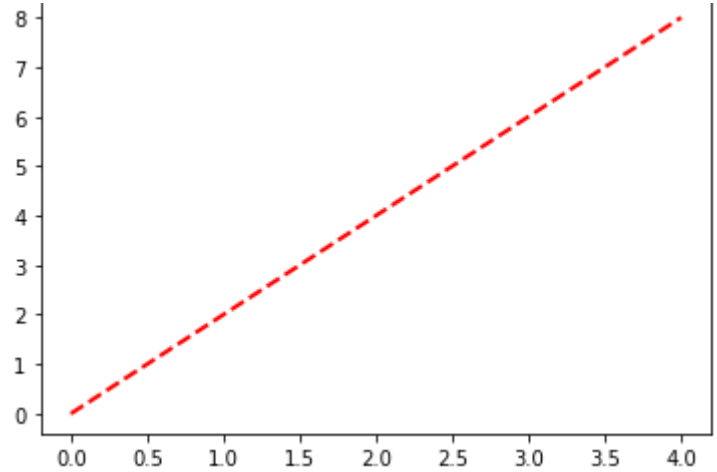
Para la contriccion de este primer grafico utilizaremos dos arreglos

In [ ]:
```
x = [0,1,2,3,4]
y = [0,2,4,6,8]

# Asignamos un tamaño y el numero de pixeles por pulgada.
plt.figure(figsize=(8,5), dpi=100)

# Graficos en formato punto
plt.plot(x,y,'bo')
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7f1876d16dd0>]
```

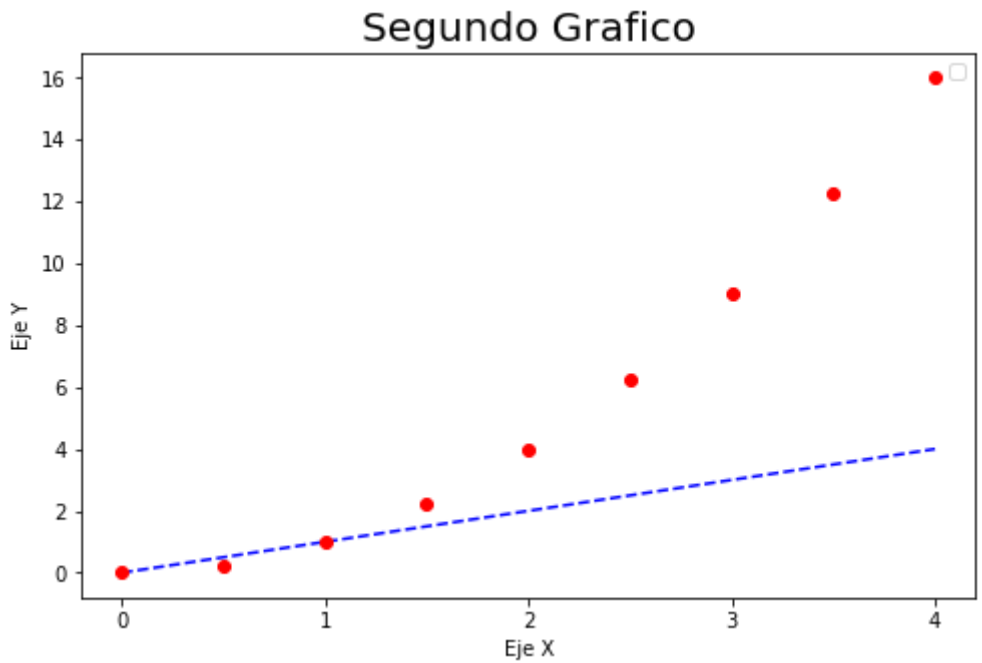


In [ ]:
```
#Para graficar en un formato linea utilizamos el siguiente comando

#Graficamos linea
plt.plot(x,y, color='red', linewidth=2, linestyle='--', label='Linea')
plt.show()
```

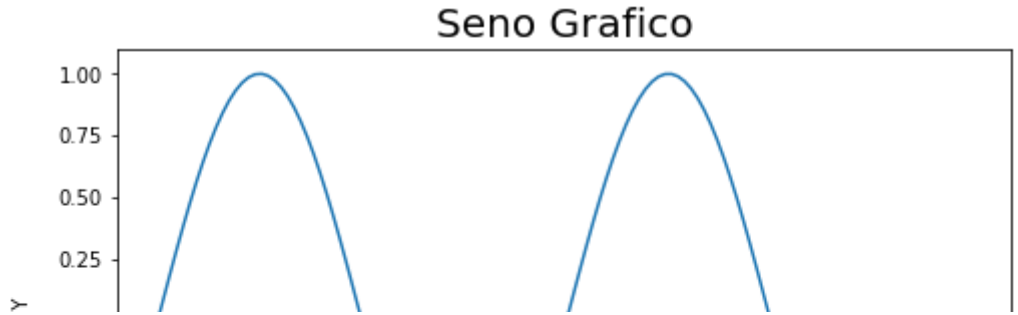


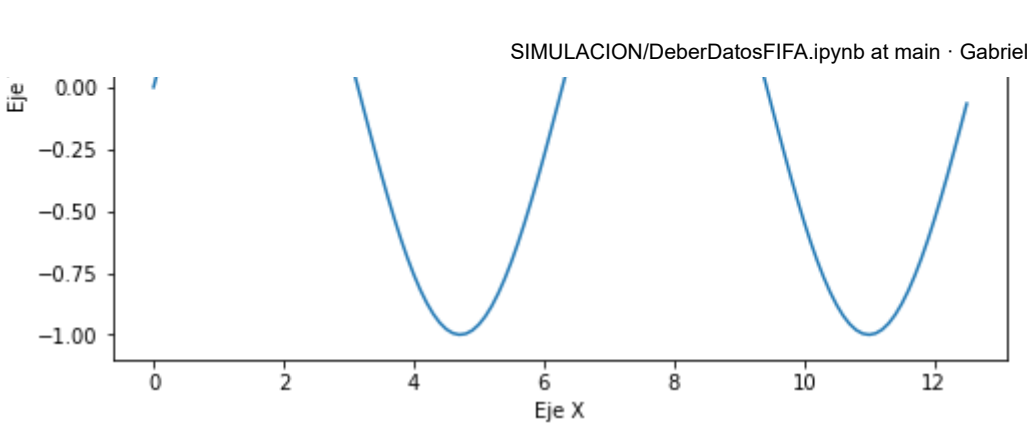
In [ ]:
```
# Generar un rango de puntos para ello vamos a utilizar np.arange(inicio, fin, paso)

x2 = np.arange(0,4.5,0.5)
y2 = np.arange(0,4.5,0.5)


plt.figure(figsize=(8,5))
plt.title('Segundo Grafico', fontdict={'fontname':'Comic Sans MS', 'fontsize': 20})
plt.xlabel('Eje X')
plt.ylabel('Eje Y')

#Graficar
plt.plot(x2,y2,'b--')
#Segunda linea con puntos y elevado al cuadrado el arreglo x2
plt.plot(x2,x2**2,'ro')
# Asignar Escala de eje en X
plt.xticks([0,1,2,3,4])
#Agregar la leyenda al grafico
plt.legend()

#Guardar la imagen en un archivo
plt.savefig('segundografico.png',dpi=300)

plt.show()
```

```
No handles with labels found to put in legend.
findfont: Font family ['Comic Sans MS'] not found. Falling back to DejaVu Sans.
```

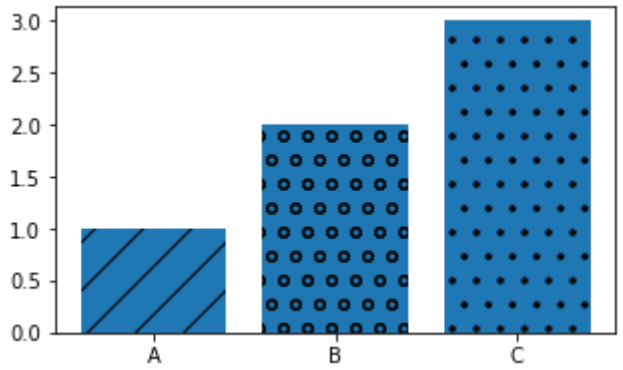


In [ ]:
```
# Graficar funciones matematicas

x3 = np.arange(0, 4*np.pi,0.1)
y3 = np.sin(x3)

plt.figure(figsize=(8,5))
plt.title('Seno Grafico', fontdict={'fontname':'Comic Sans MS', 'fontsize': 20})
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.plot(x3,y3)
plt.show()
```





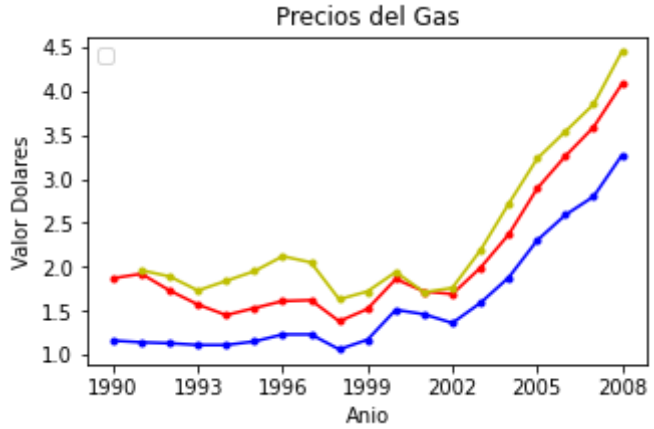
In [ ]:
```
#Grafico de Barras
etiquetas = ['A', 'B', 'C']
valores = np.arange(1,4,1)
plt.figure(figsize=(5,3))
barras = plt.bar(etiquetas, valores)

#Patrones dentro de las barras
patrones = ['/', 'o', '.']
for bar in barras:
    bar.set_hatch(patrones.pop(0))

plt.show()
```



In [ ]:
```
#Trabajar con datos en formato .csv

#datos = pd.read_csv('gas_precios.csv')
datos = pd.read_csv('/content/drive/MyDrive/gas_precios.csv')
```

```
plt.figure(figsize=(5,3))

plt.title('Precios del Gas')

plt.plot(datos.Year, datos.USA, 'b.-')
plt.plot(datos.Year, datos.Canada, 'r.-')
plt.plot(datos.Year, datos.Australia, 'y.-')

plt.xticks(datos.Year[::3].tolist())

plt.xlabel('Anio')
plt.ylabel('Valor Dolares')
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



In [ ]:
```
#Trabajar con otro tipo de datos Fifa
#fifa = pd.read_csv('fifa_datos.csv')
fifa = pd.read_csv('/content/drive/MyDrive/fifa_datos.csv')
#imprimir los primeros 5 datos del archivo
fifa.head(5)
```

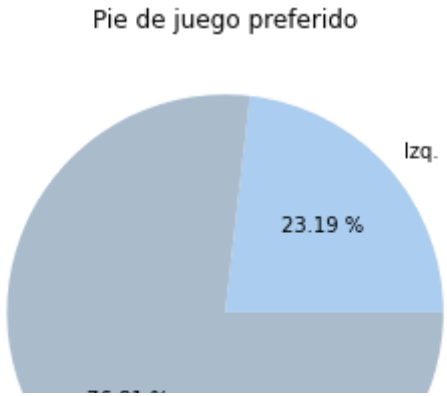
Out[ ]:

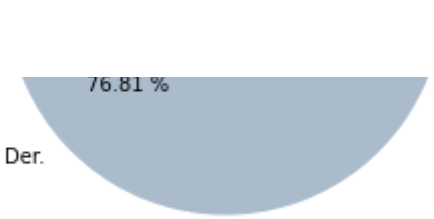
	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png

1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png

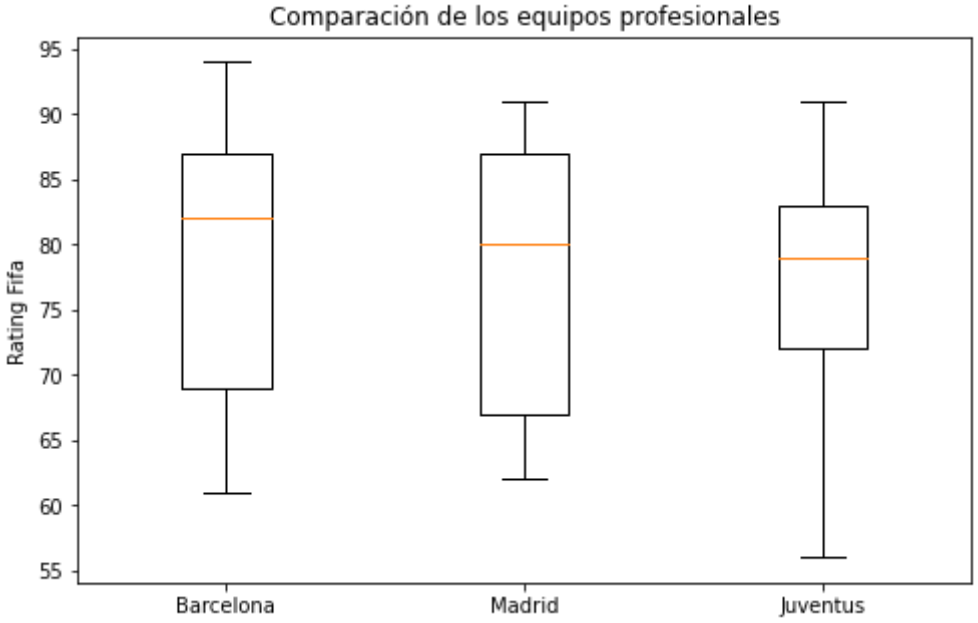
5 rows × 89 columns

In [ ]:
```
# Generar un grafico de cual es su pie diestro

izquierdo = fifa.loc[fifa['Preferred Foot'] == 'Left'].count()[0]
derecho = fifa.loc[fifa['Preferred Foot'] == 'Right'].count()[0]

plt.figure(figsize=(8,5))

etiquetas = ['Izq.', 'Der.']
colores = ['#abcdef', '#aabbcc']
plt.pie([izquierdo, derecho], labels=etiquetas, colors=colores, autopct='%.2f %%')
plt.title('Pie de juego preferido')
plt.show()
```





In [ ]:
```
# Cuadro de barras y bigotes

plt.figure(figsize=(8,5))

barcelona = fifa.loc[fifa.Club == 'FC Barcelona']['Overall']
madrid = fifa.loc[fifa.Club == 'Real Madrid']['Overall']
juventus = fifa.loc[fifa.Club == 'Chelsea']['Overall']
bp = plt.boxplot([barcelona, madrid, juventus], labels=['Barcelona', 'Madrid', 'Juventus'])
plt.title('Comparación de los equipos profesionales')
plt.ylabel('Rating Fifa')
plt.show()
```



DEBER FIFA

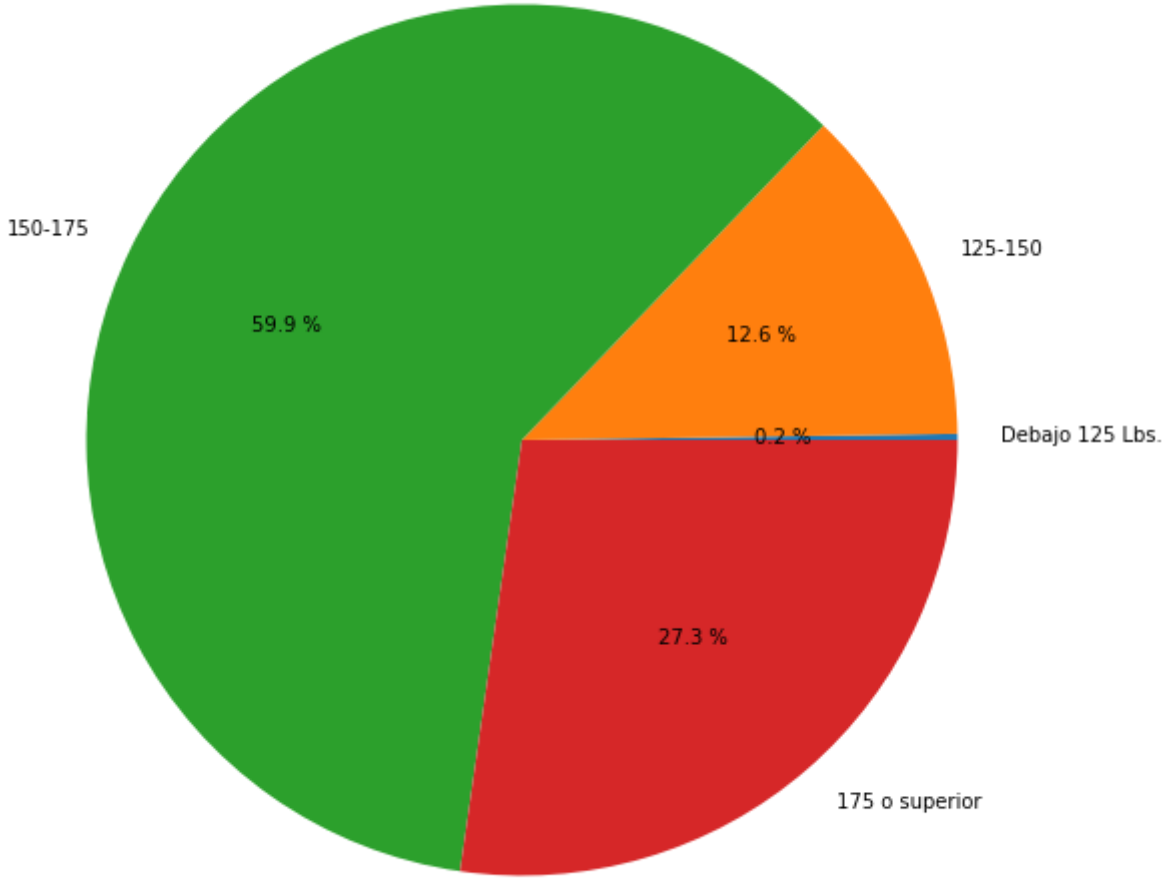
1 Con los datos de Fifa, organizar a los jugadores de acuerdo al peso en las siguientes escalas y generar un cuadro tipo PIE

- Debajo 125 Lbs.
- 125-150
- 150-175
- 175 o superior

In [ ]:
```
pri = 0
seg = 0
ter = 0
cua = 0
n = 0
try:
  for i in fifa['Weight']:
    if (pd.isna(i) == True):
      n = n + 1
    elif (float(i[:3]) < 125.0):
      pri = pri + 1
    elif (float(i[:3]) >= 125.0 and int(i[:3]) < 150.0):
      seg = seg + 1
    elif (float(i[:3]) >= 150.0 and int(i[:3]) < 175.0):
      ter = ter + 1
    elif (float(i[:3]) >= 175.0):
      cua = cua + 1
except TypeError:
  pass
b = np.array([pri, seg, ter, cua])
a = np.array(["Debajo 125 Lbs.", "125-150", "150-175", "175 o superior"])
#print(n)
print(pri)
print(seg)
print(ter)
print(cua)
#print(n+pri+seg +ter+cua)
plt.pie(b, labels=a, autopct="%0.1f %%")
plt.gcf().set_size_inches(50, 10)
plt.show()
```

```
41
2290
10876
```

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

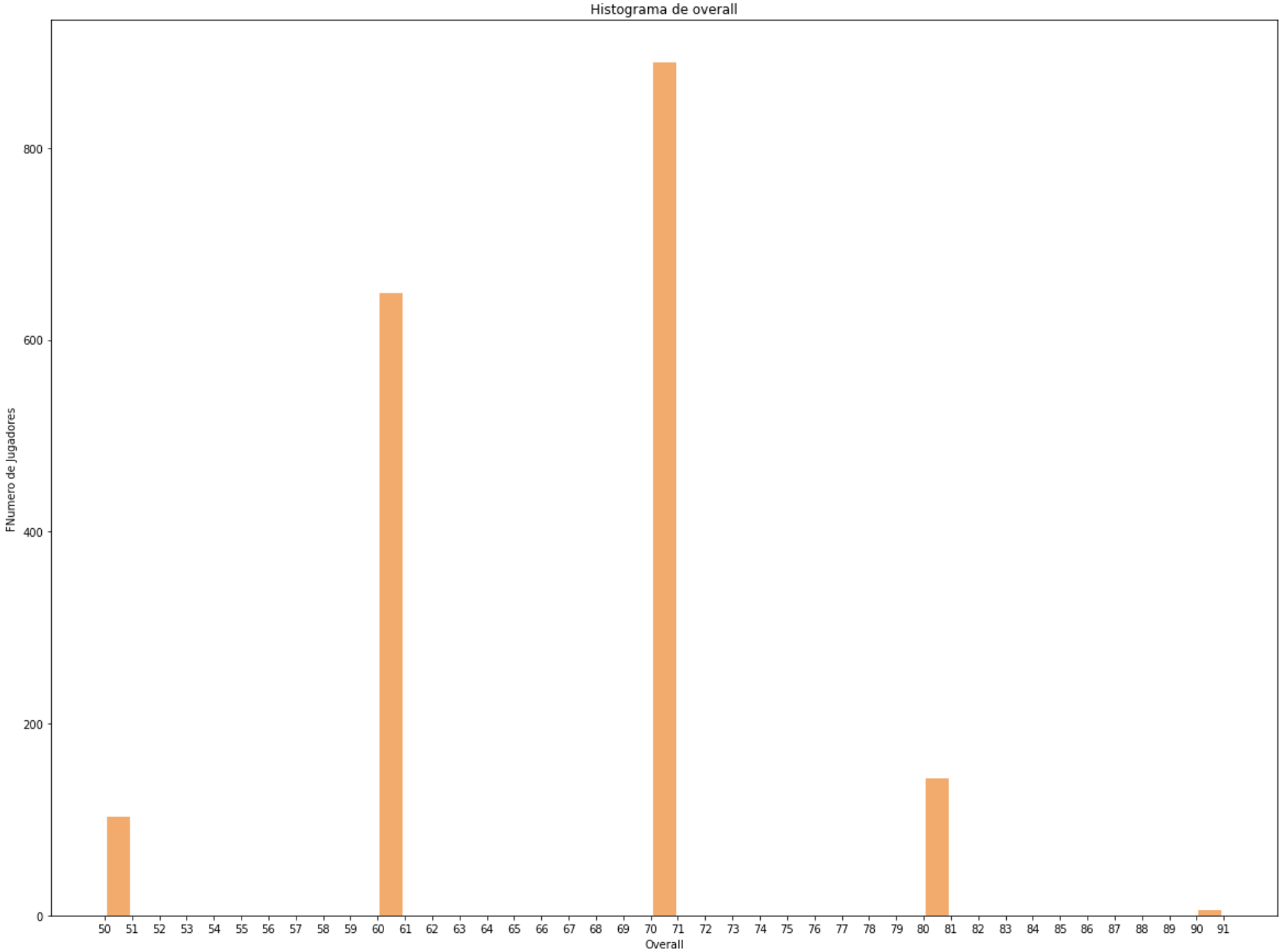
- 40
- 50
- 60

- 70
- 80
- 90
- 100

In [ ]:
```
l = []
for i in fifa['Overall']:
  if i == 40:
    l.append(i)
  elif i == 50:
    l.append(i)
  elif i == 60:
    l.append(i)
  elif i == 70:
    l.append(i)
  elif i == 80:
    l.append(i)
  elif i == 90:
    l.append(i)
  elif i == 100:
    l.append(i)
intervalos = range(min(l), max(l) + 2)
print(l.count(40))
print(l.count(50))
print(l.count(60))
print(l.count(70))
print(l.count(80))
print(l.count(90))
print(l.count(100))
plt.hist(x=l, bins=intervalos, color='#F2AB6D', rwidth=0.85)
plt.gcf().set_size_inches(20, 15)
plt.title('Histograma de overall')
plt.xlabel('Overall')
plt.ylabel('FNumero de Jugadores')
plt.xticks(intervalos)
plt.show() #dibujamos el histograma
```

```
0
103
649
889
143
```

```
143
5
0
```



3 Investigar como pasar parametros y generar reportes utilizando Notebook, una de las formas es utilizar papermill

5 Investigar como pasar parametros y general reportes utilizando Notebook, una de las formas es utilizar papermill