

## ▼ Asignación

1. Convertir en coordenadas polares los datos de `de` y `d1`
2. Graficar un cardioide en coordenadas polares en Python

```
import pandas as pd
import numpy as np
np.random.seed(123)

df1 = pd.DataFrame({
    'de': np.sort(np.random.normal(loc = 4, scale = 1, size=96)),
    'd1': np.sort(np.random.normal(loc=4.5, scale=1.2, size=96)),
    'ddd': np.repeat(np.random.randn(1), 4)
})

df1['localidad'] = np.repeat(['l1', 'l2']*24, 2)
df1.head()
```

## ▼ Convirtiendo coordenadas polares

```
import numpy as np

def cart2pol(x, y):
    rho = np.sqrt(x**2 + y**2)
    theta = np.degrees(np.arctan2(y, x))
    return(rho, theta)

coordpde = cart2pol(df1['ddd'], df1['de'])
#print(coordpde)
df3 = pd.DataFrame({
    'rho': coordpde[0],
    'theta': coordpde[1],
})
print(df3)

coordpdl = cart2pol(df1['ddd'], df1['d1'])
#print(coordpdl)
df4 = pd.DataFrame({
    'rho': coordpdl[0],
    'theta': coordpdl[1],
})
print(df4)
```

```
coordpdle = cart2pol(df1['d1'], df1['de'])
#print(coordpdle)
```

```
df5 = pd.DataFrame({
    'rho':coordpdle[0],
    'theta': coordpdle[1],
})
print(df3)
```

## ▼ Elaborando cardioides con las coordenadas convertidas

```
from matplotlib import pyplot as plt
```

```
plt.polar(df3["theta"],df3["rho"] , 'r')
plt.title("Cardioide para diametro ecuatorial vs ddd")
plt.show()
```

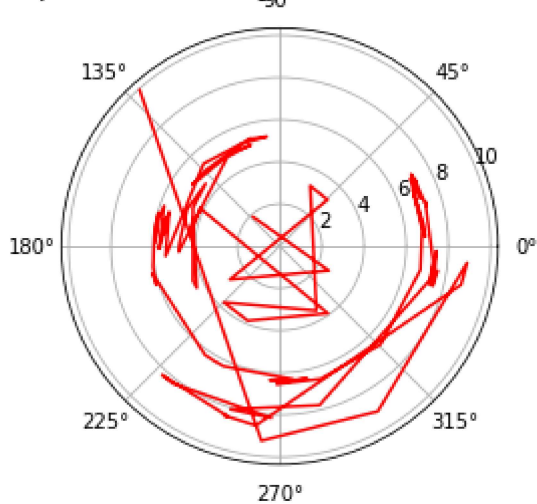


```
plt.polar(df4["theta"],df4["rho"] , 'r')
plt.title("Cardioide para diametro longitudinal vs ddd")
plt.show()
```

Cardioide para diametro longitudinal vs ddd

```
plt.polar(df5["theta"],df5["rho"] , 'r')
plt.title("Cardioide para diametro longitudinal vs diametro ecuatorial")
plt.show()
```

Cardioide para diametro longitudinal vs diametro ecuatorial



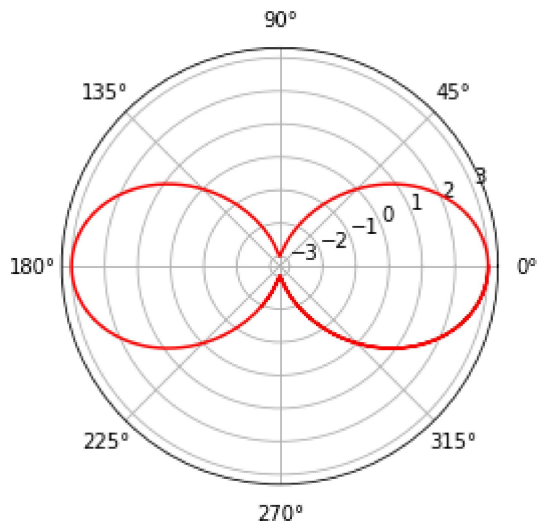
## ▼ Elaborando cardioide estándar

```
import numpy as np
from matplotlib import pyplot as plt

theta = np.linspace(0, 2*np.pi, 1000)
r = 3 * (np.cos(2*theta))

plt.polar(theta, r, 'r')

plt.show()
```



---

✓ 0 s completado a las 15:54 ● ✕