

```

import numpy as np
import sympy as sym
import matplotlib.pyplot as plt

# INGRESO , Datos de prueba
deltat=0.05
desplaza=6
tiempo=np.arange((desplaza)*deltat,deltat*36+desplaza*deltat,deltat)
pulso = np.array([507.0, 509.0, 508.0, 508.0, 509.0,
524.0, 530.0, 517.0, 507.0, 504.0, 504.0, 509.0, 512.0,
509.0, 505.0, 505.0, 507.0, 507.0, 509.0, 508.0, 507.0,
508.0, 518.0, 534.0, 523.0, 508.0, 503.0, 500.0, 505.0,
510.0, 509.0, 508.0, 507.0, 507.0, 508.0, 509.0, 508.0,
505.0, 505.0, 509.0, 526.0, 532.0, 518.0, 505.0, 504.0,
505.0, 510.0, 512.0, 507.0, 507.0])

pulsoa=pulso[desplaza:36+desplaza]#este usamos para un pulso

def crearpol(xi, fi):
    # PROCEDIMIENTO
    n = len(xi)
    x = sym.Symbol('x')
    # Polinomio
    polinomio = 0
    for i in range(0,n,1):
        # Termino de Lagrange
        termino = 1
        for j in range(0,n,1):
            if (j!=i):
                termino =
termino*(x-xi[j])/(xi[i]-xi[j])

```

```

        if fi[i]==fi[0] and fi[0]==fi[1]:
            fi[i]= fi[i]+0.0000000000001
            polinomio = polinomio + termino*fi[i]
        else:
            polinomio = polinomio + termino*fi[i]
# Expande el polinomio
px = polinomio.expand()
# para evaluacion numérica
pxn = sym.lambdify(x,polinomio)
return (px)

```

"""

Codigo previo

```
polinomios=[]
```

```
pxns=[]
```

```
tn1=[]
```

```
pn1=[]
```

```
tiempoi = tiempo[i:pospico+1]
```

```
tiempoinuevo=tiempo[pospico:]
```

```
pulsoi = pulsob[i:pospico+1]
```

```
pulsoinuevo=pulsob[pospico:]
```

```
tn1.append(tiempoi)
```

```
pn1.append(pulsoi)
```

```
pol=crearpol(tiempoi,pulsoi)
```

```
polinomios.append(pol[0])
```

```
pxns.append(pol[1])
```

```
tn1.append(tiempoinuevo)
```

```
pn1.append(pulsoinuevo)
```

```
pol2=crearpol(tiempoinuevo,pulsoinuevo)
```

```
polinomios.append(pol2[0])
```

```
pxns.append(pol2[1])
```

```

# Salida
print('Polinomios: ')
for i in range(len(polinomios)):
    print(i+1, ':', polinomios[i])
    a = np.min(tn1[i])
    b = np.max(tn1[i])
    # muestras = 51
    muestras = 41
    xi_p = np.linspace(a,b,muestras)
    fi_p = pxns[i](xi_p)

    plt.title('Gráfica de un pulso')

    plt.plot(xi_p,fi_p)"""
#Puntos usados para los polinomios
xia=np.copy(tiempo[:9])
fia=np.copy(pulsoa[:9])
xib=np.copy(tiempo[8:15])
fib=np.copy(pulsoa[8:15])
xic=np.copy(tiempo[14:18])
fic=np.copy(pulsoa[14:18])
xid=np.copy(tiempo[17:22])
fid=np.copy(pulsoa[17:22])
xie=np.copy(tiempo[21:31])
fie=np.copy(pulsoa[21:31])
xif=np.copy(tiempo[30:37])
fif=np.copy(pulsoa[30:37])
#PROCEDIMIENTO
x=sym.Symbol('x')
pxa=crearpol(xia,fia)

```

```
pa=sym.lambdify(x,pxa)
pxb=crearpol(xib,fib)
pb=sym.lambdify(x,pxb)
pxc=crearpol(xic,fic)
pc=sym.lambdify(x,pxc)
pxd=crearpol(xid,fid)
pd=sym.lambdify(x,pxd)
pxe=crearpol(xie,fie)
pe=sym.lambdify(x,pxe)
pxf=crearpol(xif,fif)
pf=sym.lambdify(x,pxf)
```

```
#Puntos Grafica
```

```
a=np.min(xia)
b=np.max(xia)
c=np.min(xib)
d=np.max(xib)
e=np.min(xic)
f=np.max(xic)
g=np.min(xid)
h=np.max(xid)
i=np.min(xie)
j=np.max(xie)
k=np.min(xif)
l=np.max(xif)
```

```
muestras=61
xnia=np.linspace(a,b,muestras)
pia=pa(xnia)
xnib=np.linspace(c,d,muestras)
pib=pb(xnib)
xnic=np.linspace(e,f,muestras)
pic=pc(xnic)
xnid=np.linspace(g,h,muestras)
```

```

pid=pd(xnid)
xnie=np.linspace(i,j,muestras)
pie=pe(xnie)
xnif=np.linspace(k,l,muestras)
pif=pf(xnif)

#SALIDA
frecuencia=round(60/(((tiempo[18]-tiempo[0])/2)+((tiempo
[-1]-tiempo[18])/2)))
print("La frecuencia cardiaca es:",frecuencia,"ppm")
print("Se aproximó el Pulso con los siguientes
polinomios")
print("Polinomio 1:")
print(pxa)
print("Rango de existencia:["a","b"]")
print("Polinomio 2:")
print(pxb)
print("Rango de existencia:["c","d"]")
print("Polinomio 3:")
print(pxc)
print("Rango de existencia:["e","f"]")
print("Polinomio 4:")
print(pxd)
print("Rango de existencia:["g","h"]")
print("Polinomio 5:")
print(pxe)
print("Rango de existencia:["i","j"]")
print("Polinomio 6:")
print(pxf)
print("Rango de existencia:["k","l"]")
plt.plot(xnia,pia)
plt.plot(xnib,pib)
plt.plot(xnic,pic)
plt.plot(xnid,pid)

```

```
plt.plot(xnie, pie)
plt.plot(xnif, pif)
plt.plot(tiempo, pulsoa, 'o')
plt.show()
```