

# Vectores con pLas81-YFP

February 13, 2018

```
In [57]: import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib import colors
from scipy.interpolate import UnivariateSpline
from scipy.optimize import curve_fit
from scipy import stats
import seaborn as sns
```

```
In [99]: tt=np.fromfile('t', sep=',')
```

```
#arrays replicas glucosa
cfp15271=np.fromfile('p1527gCFP1', sep=',')
rfp15271=np.fromfile('p1527gRFP1', sep=',')
yfp15271=np.fromfile('p1527gYFP1', sep=',')
od15271=np.fromfile('p1527gOD1', sep=',')
cfp15272=np.fromfile('p1527gCFP2', sep=',')
rfp15272=np.fromfile('p1527gRFP2', sep=',')
yfp15272=np.fromfile('p1527gYFP2', sep=',')
od15272=np.fromfile('p1527gOD2', sep=',')
cfp15273=np.fromfile('p1527gCFP3', sep=',')
rfp15273=np.fromfile('p1527gRFP3', sep=',')
yfp15273=np.fromfile('p1527gYFP3', sep=',')
od15273=np.fromfile('p1527gOD3', sep=',')
'''

print(cfp15271.shape)
print(rfp15271.shape)
print(yfp15271.shape)
print(od15271.shape)
print(cfp15272.shape)
print(rfp15272.shape)
print(yfp15272.shape)
print(od15272.shape)
print(cfp15273.shape)
print(rfp15273.shape)
print(yfp15273.shape)
```

```
print(od15273.shape)'''
```

```
cfp18271=np.fromfile('p1827gCFP1', sep=',')  
rfp18271=np.fromfile('p1827gRFP1', sep=',')  
yfp18271=np.fromfile('p1827gYFP1', sep=',')  
od18271=np.fromfile('p1827gOD1', sep=',')  
cfp18272=np.fromfile('p1827gCFP2', sep=',')  
rfp18272=np.fromfile('p1827gRFP2', sep=',')  
yfp18272=np.fromfile('p1827gYFP2', sep=',')  
od18272=np.fromfile('p1827gOD2', sep=',')  
cfp18273=np.fromfile('p1827gCFP3', sep=',')  
rfp18273=np.fromfile('p1827gRFP3', sep=',')  
yfp18273=np.fromfile('p1827gYFP3', sep=',')  
od18273=np.fromfile('p1827gOD3', sep=',')
```

```
'''  
print(cfp18271.shape)  
print(rfp18271.shape)  
print(yfp18271.shape)  
print(od18271.shape)  
print(cfp18272.shape)  
print(rfp18272.shape)  
print(yfp18272.shape)  
print(od18272.shape)  
print(cfp18273.shape)  
print(rfp18273.shape)  
print(yfp18273.shape)  
print(od18273.shape)'''
```

```
cfp12271=np.fromfile('p1227gCFP1', sep=',')  
rfp12271=np.fromfile('p1227gRFP1', sep=',')  
yfp12271=np.fromfile('p1227gYFP1', sep=',')  
od12271=np.fromfile('p1227gOD1', sep=',')  
cfp12272=np.fromfile('p1227gCFP2', sep=',')  
rfp12272=np.fromfile('p1227gRFP2', sep=',')  
yfp12272=np.fromfile('p1227gYFP2', sep=',')  
od12272=np.fromfile('p1227gOD2', sep=',')  
cfp12273=np.fromfile('p1227gCFP3', sep=',')  
rfp12273=np.fromfile('p1227gRFP3', sep=',')  
yfp12273=np.fromfile('p1227gYFP3', sep=',')  
od12273=np.fromfile('p1227gOD3', sep=',')
```

```
'''  
print(cfp12271.shape)  
print(rfp12271.shape)  
print(yfp12271.shape)  
print(od12271.shape)
```

```

print(cfp12272.shape)
print(rfp12272.shape)
print(yfp12272.shape)
print(od12272.shape)
print(cfp12273.shape)
print(rfp12273.shape)
print(yfp12273.shape)
print(od12273.shape)'''

#Controles
#Promedios controles glucosa
cfpcg151=np.fromfile('15pcgCFP1', sep=',')
rfpcg151=np.fromfile('15pcgRFP1', sep=',')
yfpcg151=np.fromfile('15pcgYFP1', sep=',')
odcg151=np.fromfile('15pcgOD1', sep=',')
cfpcg152=np.fromfile('15pcgCFP2', sep=',')
rfpcg152=np.fromfile('15pcgRFP2', sep=',')
yfpcg152=np.fromfile('15pcgYFP2', sep=',')
odcg152=np.fromfile ('15pcgOD2',sep=',')
cfpcg153=np.fromfile('15pcgCFP3', sep=',')
rfpcg153=np.fromfile('15pcgRFP3', sep=',')
yfpcg153=np.fromfile('15pcgYFP3', sep=',')
odcg153=np.fromfile('15pcgOD3', sep=',')

'''

print(cfp151.shape)
print(rfp151.shape)
print(yfp151.shape)
print(odcg151.shape)
print(cfp151.shape)
print(rfp151.shape)
print(yfp151.shape)
print(odcg151.shape)
print(cfp151.shape)
print(rfp151.shape)
print(yfp151.shape)
print(odcg151.shape)'''

cfpcg181=np.fromfile('18pcgCFP1', sep=',')
rfpcg181=np.fromfile('18pcgRFP1', sep=',')
yfpcg181=np.fromfile('18pcgYFP1', sep=',')
odcg181=np.fromfile('18pcgOD1', sep=',')
cfpcg182=np.fromfile('18pcgCFP2', sep=',')
rfpcg182=np.fromfile('18pcgRFP2', sep=',')
yfpcg182=np.fromfile('18pcgYFP2', sep=',')
odcg182=np.fromfile ('18pcgOD2',sep=',')
cfpcg183=np.fromfile('18pcgCFP3', sep=',')
rfpcg183=np.fromfile('18pcgRFP3', sep=',')

```

```
yfpcg183=np.fromfile('18pcgYFP3', sep=',')
odcg183=np.fromfile('18pcgOD3', sep=',')
```

```
'''
print(cfpcg181.shape)
print(rfpcg181.shape)
print(yfpcg181.shape)
print(odcg181.shape)
print(cfpcg181.shape)
print(rfpcg181.shape)
print(yfpcg181.shape)
print(odcg181.shape)
print(cfpcg181.shape)
print(rfpcg181.shape)
print(yfpcg181.shape)
print(odcg181.shape)'''
```

```
cfpcg121=np.fromfile('12pcgCFP1', sep=',')
rfpcg121=np.fromfile('12pcgRFP1', sep=',')
yfpcg121=np.fromfile('12pcgYFP1', sep=',')
odcg121=np.fromfile('12pcgOD1', sep=',')
cfpcg122=np.fromfile('12pcgCFP2', sep=',')
rfpcg122=np.fromfile('12pcgRFP2', sep=',')
yfpcg122=np.fromfile('12pcgYFP2', sep=',')
odcg122=np.fromfile('12pcgOD2', sep=',')
cfpcg123=np.fromfile('12pcgCFP3', sep=',')
rfpcg123=np.fromfile('12pcgRFP3', sep=',')
yfpcg123=np.fromfile('12pcgYFP3', sep=',')
odcg123=np.fromfile('12pcgOD3', sep=',')
```

```
'''
print(cfpcg121.shape)
print(rfpcg121.shape)
print(yfpcg121.shape)
print(odcg121.shape)
print(cfpcg121.shape)
print(rfpcg121.shape)
print(yfpcg121.shape)
print(odcg121.shape)
print(cfpcg121.shape)
print(rfpcg121.shape)
print(yfpcg121.shape)
print(odcg121.shape)'''
```

```
Out[99]: '\nprint(cfpcg121.shape)\nprint(rfpcg121.shape)\nprint(yfpcg121.shape)\nprint(odcg121.s
```

```
In [100]: #Promedios glicerol
#arrays replicas glicerol
cfp1527g1=np.fromfile('p1527g1CFP1', sep=',')
```

```

rfp1527g1=np.fromfile('p1527g1RFP1', sep=',')
yfp1527g1=np.fromfile('p1527g1YFP1', sep=',')
od1527g1=np.fromfile('p1527g1OD1', sep=',')
cfp1527g2=np.fromfile('p1527g1CFP2', sep=',')
rfp1527g2=np.fromfile('p1527g1RFP2', sep=',')
yfp1527g2=np.fromfile('p1527g1YFP2', sep=',')
od1527g2=np.fromfile('p1527g1OD2', sep=',')
cfp1527g3=np.fromfile('p1527g1CFP3', sep=',')
rfp1527g3=np.fromfile('p1527g1RFP3', sep=',')
yfp1527g3=np.fromfile('p1527g1YFP3', sep=',')
od1527g3=np.fromfile('p1527g1OD3', sep=',')
'''

print(cfp1527g1.shape)
print(rfp1527g1.shape)
print(yfp1527g1.shape)
print(od1527g1.shape)
print(cfp1527g2.shape)
print(rfp1527g2.shape)
print(yfp1527g2.shape)
print(od1527g2.shape)
print(cfp1527g3.shape)
print(rfp1527g3.shape)
print(yfp1527g3.shape)
print(od1527g3.shape)'''

cfp1827g1=np.fromfile('p1827g1CFP1', sep=',')
rfp1827g1=np.fromfile('p1827g1RFP1', sep=',')
yfp1827g1=np.fromfile('p1827g1YFP1', sep=',')
od1827g1=np.fromfile('p1827g1OD1', sep=',')
cfp1827g2=np.fromfile('p1827g1CFP2', sep=',')
rfp1827g2=np.fromfile('p1827g1RFP2', sep=',')
yfp1827g2=np.fromfile('p1827g1YFP2', sep=',')
od1827g2=np.fromfile('p1827g1OD2', sep=',')
cfp1827g3=np.fromfile('p1827g1CFP3', sep=',')
rfp1827g3=np.fromfile('p1827g1RFP3', sep=',')
yfp1827g3=np.fromfile('p1827g1YFP3', sep=',')
od1827g3=np.fromfile('p1827g1OD3', sep=',')
'''

print(cfp1827g1.shape)
print(rfp1827g1.shape)
print(yfp1827g1.shape)
print(od1827g1.shape)
print(cfp1827g2.shape)
print(rfp1827g2.shape)
print(yfp1827g2.shape)
print(od1827g2.shape)
print(cfp1827g3.shape)
print(rfp1827g3.shape)

```

```

print(yfp1827g3.shape)
print(od1827g3.shape)'''

cfp1227g1=np.fromfile('p1227glCFP1', sep=',')
rfp1227g1=np.fromfile('p1227glRFP1', sep=',')
yfp1227g1=np.fromfile('p1227glYFP1', sep=',')
od1227g1=np.fromfile('p1227glOD1', sep=',')
cfp1227g2=np.fromfile('p1227glCFP2', sep=',')
rfp1227g2=np.fromfile('p1227glRFP2', sep=',')
yfp1227g2=np.fromfile('p1227glYFP2', sep=',')
od1227g2=np.fromfile('p1227glOD2', sep=',')
cfp1227g3=np.fromfile('p1227glCFP3', sep=',')
rfp1227g3=np.fromfile('p1227glRFP3', sep=',')
yfp1227g3=np.fromfile('p1227glYFP3', sep=',')
od1227g3=np.fromfile('p1227glOD3', sep=',')
'''

print(cfp1227g1.shape)
print(rfp1227g1.shape)
print(yfp1227g1.shape)
print(od1227g1.shape)
print(cfp1227g2.shape)
print(rfp1227g2.shape)
print(yfp1227g2.shape)
print(od1227g2.shape)
print(cfp1227g3.shape)
print(rfp1227g3.shape)
print(yfp1227g3.shape)
print(od1227g3.shape)'''

#Promedios controles glicerol
cfpcgl151=np.fromfile('15pcglCFP1', sep=',')
rfpcgl151=np.fromfile('15pcglRFP1', sep=',')
yfpcgl151=np.fromfile('15pcglYFP1', sep=',')
odcgl151=np.fromfile('15pcglOD1', sep=',')
cfpcgl152=np.fromfile('15pcglCFP2', sep=',')
rfpcgl152=np.fromfile('15pcglRFP2', sep=',')
yfpcgl152=np.fromfile('15pcglYFP2', sep=',')
odcgl152=np.fromfile('15pcglOD2', sep=',')
cfpcgl153=np.fromfile('15pcglCFP3', sep=',')
rfpcgl153=np.fromfile('15pcglRFP3', sep=',')
yfpcgl153=np.fromfile('15pcglYFP3', sep=',')
odcgl153=np.fromfile('15pcglOD3', sep=',')
'''

print(cfpcgl151.shape)
print(rfpcgl151.shape)
print(yfpcgl151.shape)
print(odcgl151.shape)
print(cfpcgl151.shape)

```

```

print(rfpcgl151.shape)
print(yfpcgl151.shape)
print(odcgl151.shape)
print(cfpcgl151.shape)
print(rfpcgl151.shape)
print(yfpcgl151.shape)
print(odcgl151.shape)'''

cfpcgl181=np.fromfile('18pcglCFP1', sep=',')
rfpcgl181=np.fromfile('18pcglRFP1', sep=',')
yfpcgl181=np.fromfile('18pcglYFP1', sep=',')
odcgl181=np.fromfile('18pcglOD1', sep=',')
cfpcgl182=np.fromfile('18pcglCFP2', sep=',')
rfpcgl182=np.fromfile('18pcglRFP2', sep=',')
yfpcgl182=np.fromfile('18pcglYFP2', sep=',')
odcgl182=np.fromfile('18pcglOD2', sep=',')
cfpcgl183=np.fromfile('18pcglCFP3', sep=',')
rfpcgl183=np.fromfile('18pcglRFP3', sep=',')
yfpcgl183=np.fromfile('18pcglYFP3', sep=',')
odcgl183=np.fromfile('18pcglOD3', sep=',')
'''

print(cfpcgl181.shape)
print(rfpcgl181.shape)
print(yfpcgl181.shape)
print(odcgl181.shape)
print(cfpcgl181.shape)
print(rfpcgl181.shape)
print(yfpcgl181.shape)
print(odcgl181.shape)
print(cfpcgl181.shape)
print(rfpcgl181.shape)
print(yfpcgl181.shape)
print(odcgl181.shape)'''

cfpcgl121=np.fromfile('12pcglCFP1', sep=',')
rfpcgl121=np.fromfile('12pcglRFP1', sep=',')
yfpcgl121=np.fromfile('12pcglYFP1', sep=',')
odcgl121=np.fromfile('12pcglOD1', sep=',')
cfpcgl122=np.fromfile('12pcglCFP2', sep=',')
rfpcgl122=np.fromfile('12pcglRFP2', sep=',')
yfpcgl122=np.fromfile('12pcglYFP2', sep=',')
odcgl122=np.fromfile('12pcglOD2', sep=',')
cfpcgl123=np.fromfile('12pcglCFP3', sep=',')
rfpcgl123=np.fromfile('12pcglRFP3', sep=',')
yfpcgl123=np.fromfile('12pcglYFP3', sep=',')
odcgl123=np.fromfile('12pcglOD3', sep=',')
'''

print(cfpcgl121.shape)

```

```

print(rfpcgl121.shape)
print(yfpcgl121.shape)
print(odcgl121.shape)
print(cfpcgl121.shape)
print(rfpcgl121.shape)
print(yfpcgl121.shape)
print(odcgl121.shape)
print(cfpcgl121.shape)
print(rfpcgl121.shape)
print(yfpcgl121.shape)
print(odcgl121.shape)'''

```

```
Out[100]: '\nprint(cfpcgl121.shape)\nprint(rfpcgl121.shape)\nprint(yfpcgl121.shape)\nprint(odcgl121.shape)\n'
```

```
In [60]: #Funciones para ajuste Gompertz
```

```

def F_sigma(t, A, um,l):
    return ((A*np.exp(-np.exp(((um*np.exp(1))/A)*(1-t))+1))))

def Function_fit(xdata,ydata,init,end,func=F_sigma,ParamBounds=([0,0,0],[3,1,300]), title='Ajuste Gompertz',
                 Y_fit={})

    z,_=curve_fit(func,xdata[init:end], ydata[init:end], bounds=ParamBounds)

    print(z)

    evalF=func(xdata,z[0],z[1],z[2])

    plt.figure()
    plt.plot(xdata, ydata, '.',label='OD')
    plt.plot(xdata, evalF, '-',label='Ajuste')
    plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
    plt.title(title)
    plt.ylabel('Abs(nm)')
    plt.xlabel('Tiempo(min)')
    lgd=plt.legend(loc='lower right')
    plt.show()

    Y_fit=evalF,z

    return(Y_fit)

```

```

In [61]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control 15 glucosa rep 1
y1 = np.log(odcg151)-np.log(np.min(odcg151))
print('Min OD = %e'%((np.min(odcg151))))
evaly, params=Function_fit(tt,y1,0,-1,title = 'Ajuste control glucosa 0,4% E2R1')

```



```

A1 = params[0]
um1=params[1]
l1=params[2]
print('A=%e'%(A1))
print('um=%e'%(um1))
print('l=%e'%(l1))

#Cálculo datos para determinar extensión de la fase exponencial
tm1=((A1/(np.exp(1)*um1))+l1)
print('Tm=%e'%(tm1))
t21=((np.log(2))/um1)
print('doubpe=%e'%(t21))
extdp1=2*t21
print('ext=%e'%extdp1)
ttot1=tm1+extdp1
print('Tfinal=%e'%ttot1)

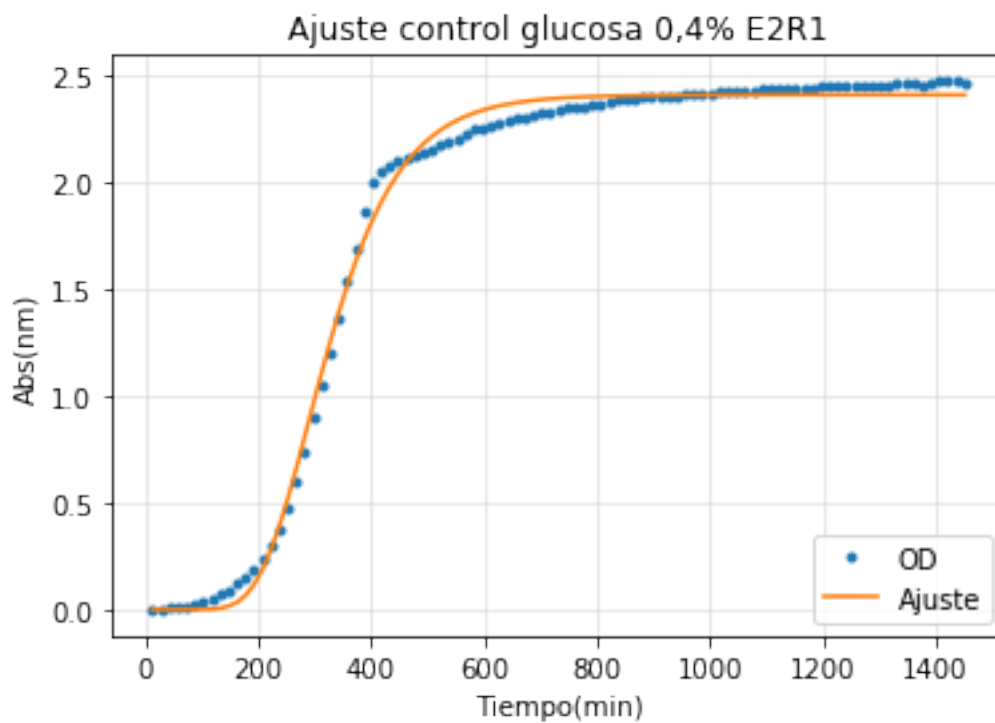
#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[19]
y2=tt[29]
plt.figure()
plt.title('Control Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg151,label='OD control E2R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odcg151[19:30],label='OD control E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

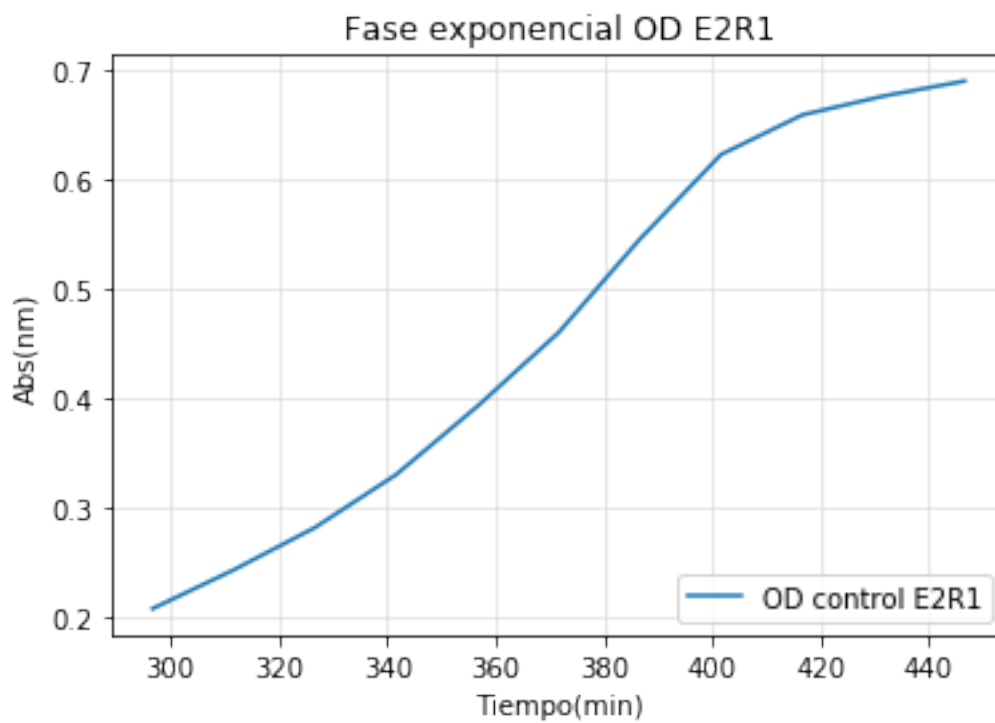
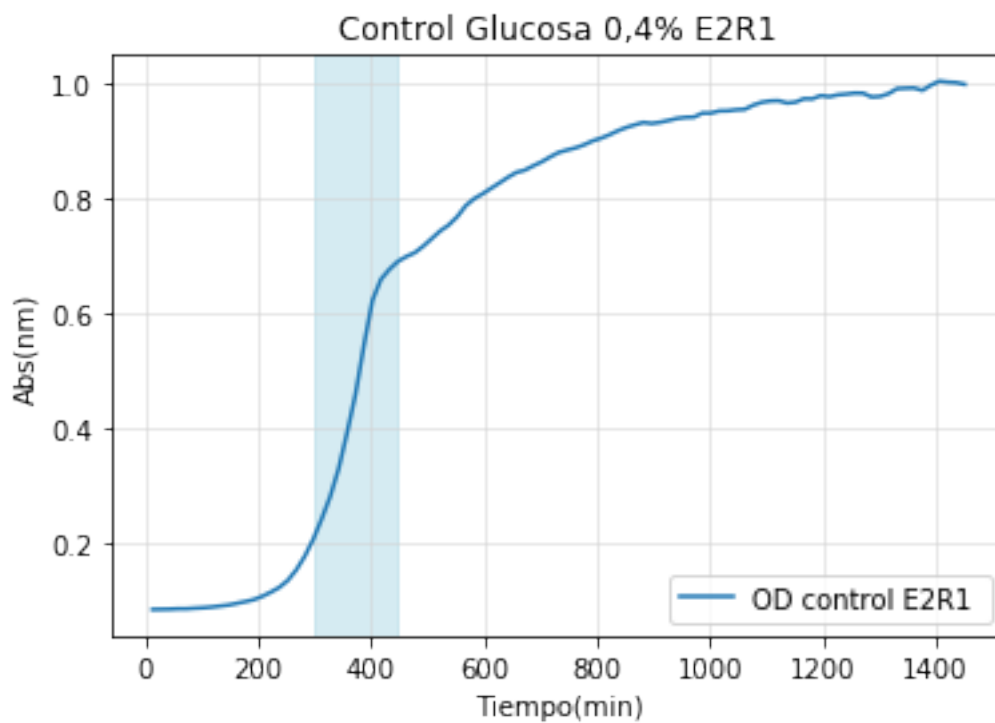
Min OD = 8.475000e-02

[ 2.40766598e+00 1.00663667e-02 2.01163416e+02]



```
A=2.407666e+00  
um=1.006637e-02  
l=2.011634e+02  
Tm=2.891525e+02  
doubpe=6.885773e+01  
ext=1.377155e+02  
Tfinal=4.268680e+02
```

```
Out[61]: <matplotlib.legend.Legend at 0x178911cce48>
```



```

In [62]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y2= np.log(odcg152)-np.log(np.min(odcg152))
print('Min OD = %e'%((np.min(odcg152))))
evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E2R2')
A2 = params[0]
um2=params[1]
l2=params[2]
print('A=%e'%(A2))
print('um=%e'%(um2))
print('l=%e'%(l2))

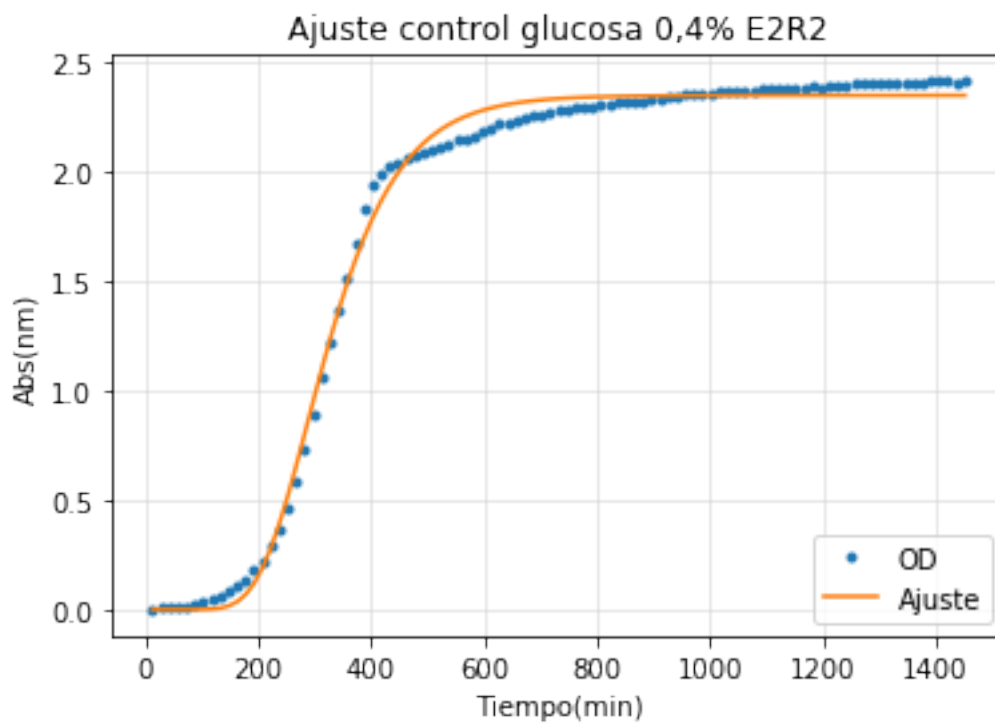
#Cálculo datos para determinar extensión de la fase exponencial
tm2=((A2/(np.exp(1)*um2))+l2)
print('Tm=%e'%(tm2))
t22=((np.log(2))/um2)
print('doubpe=%e'%(t22))
extdp2=2*t22
print('ext=%e'%extdp2)
ttot2=tm2+extdp2
print('Tfinal=%e'%ttot2)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[19]
y2=tt[29]
plt.figure()
plt.title('Control Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg152,label='OD control E2R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odcg152[19:30],label='OD control E2R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

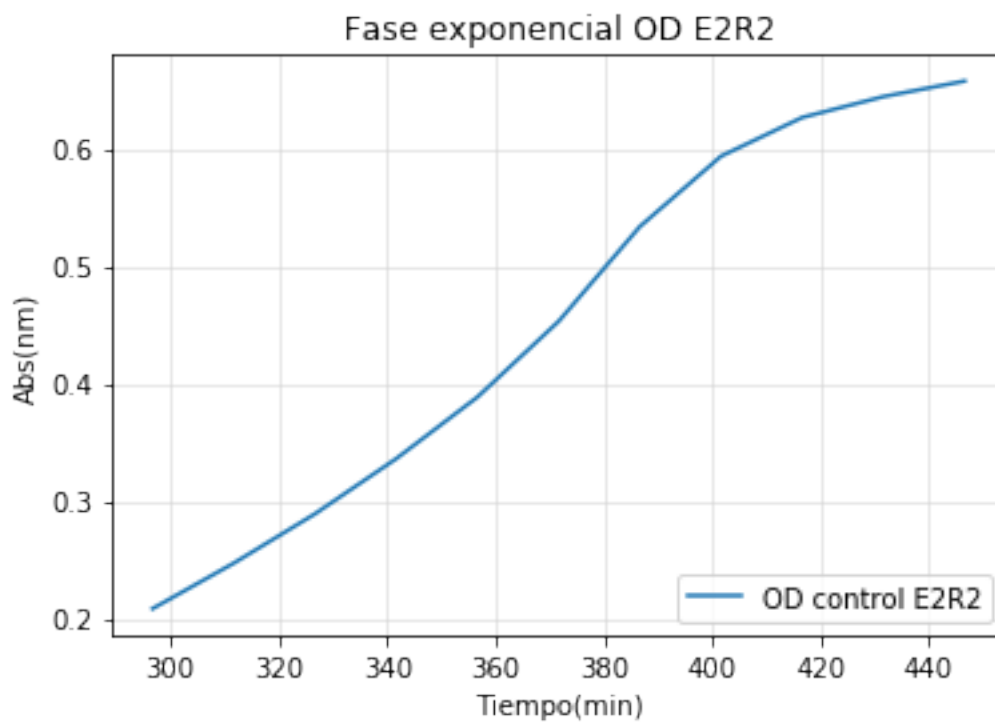
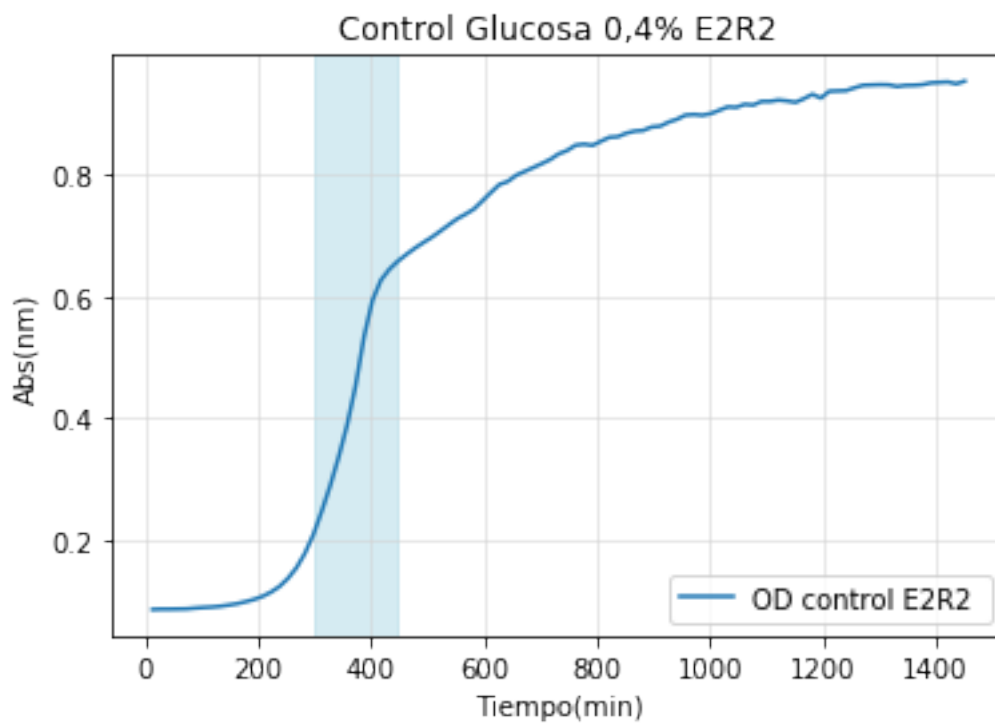
Min OD = 8.550000e-02
[ 2.34701846e+00  9.83004768e-03  1.99784486e+02]

```



```
A=2.347018e+00  
um=9.830048e-03  
l=1.997845e+02  
Tm=2.876192e+02  
doubpe=7.051310e+01  
ext=1.410262e+02  
Tfinal=4.286455e+02
```

```
Out[62]: <matplotlib.legend.Legend at 0x178934a3cf8>
```



```

In [63]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y3= np.log(odcg153)-np.log(np.min(odcg153))
print('Min OD = %e'%((np.min(odcg153))))
evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E2R3')
A3= params[0]
um3=params[1]
l3=params[2]
print('A=%e'%(A3))
print('um=%e'%(um3))
print('l=%e'%(l3))

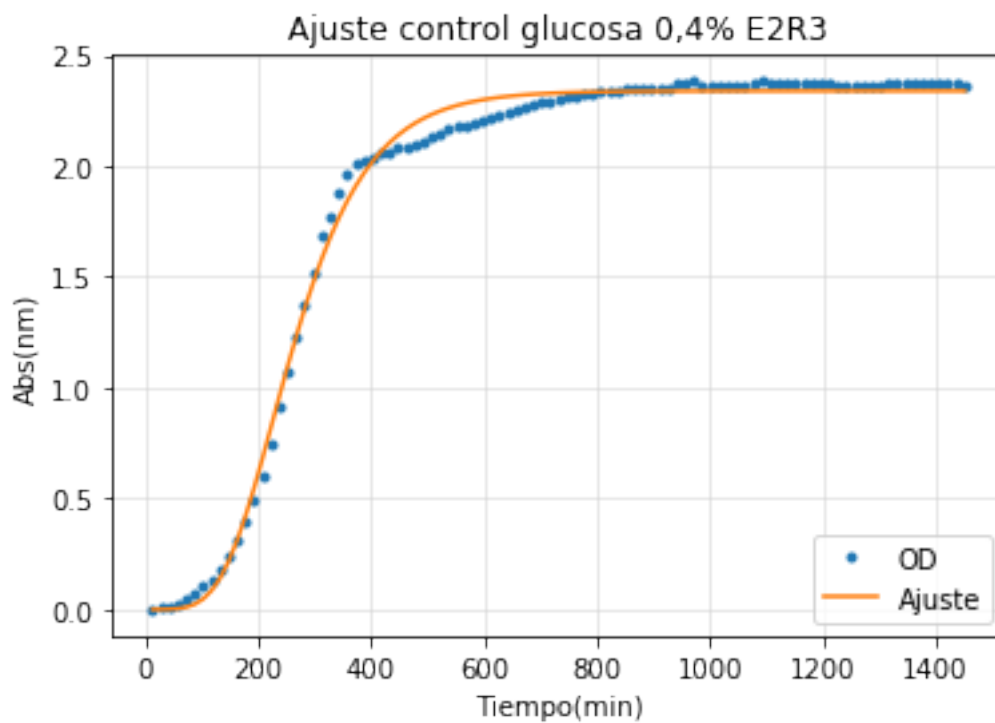
#Cálculo datos para determinar extensión de la fase exponencial
tm3=((A3/(np.exp(1)*um3))+l3)
print('Tm=%e'%(tm3))
t23=((np.log(2))/um3)
print('doubpe=%e'%(t23))
extdp3=2*t23
print('ext=%e'%extdp3)
ttot3=tm3+extdp3
print('Tfinal=%e'%ttot3)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[15]
y2=tt[26]
plt.figure()
plt.title('Control Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg153,label='OD control E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[15:27],odcg153[15:27],label='OD control E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

Min OD = 8.375000e-02
[ 2.33929203e+00  9.40512328e-03  1.34262384e+02]

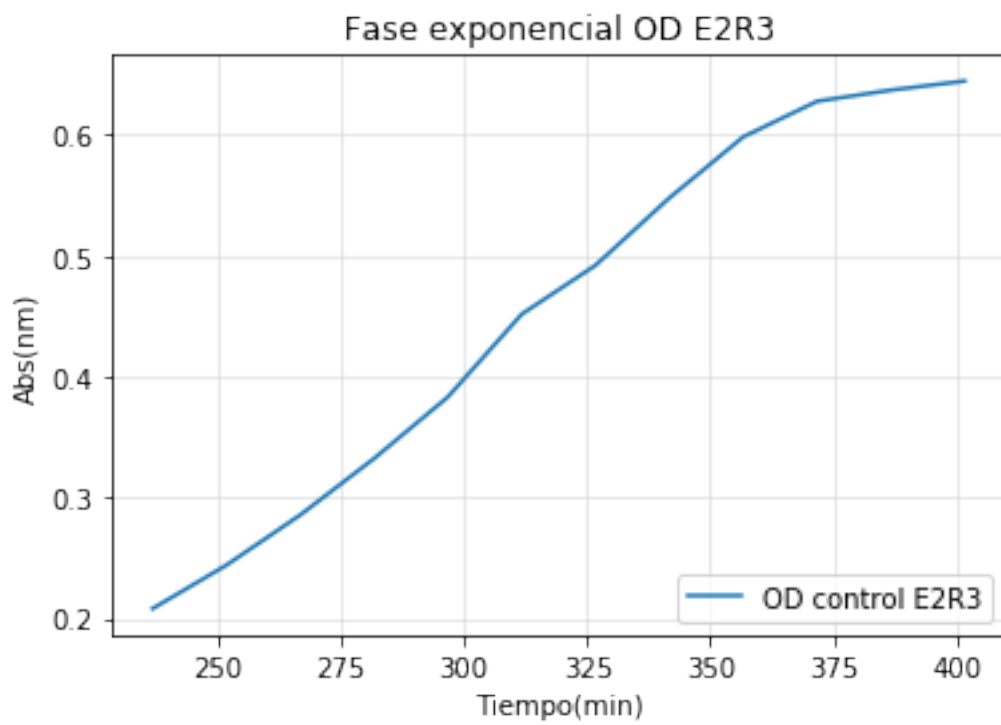
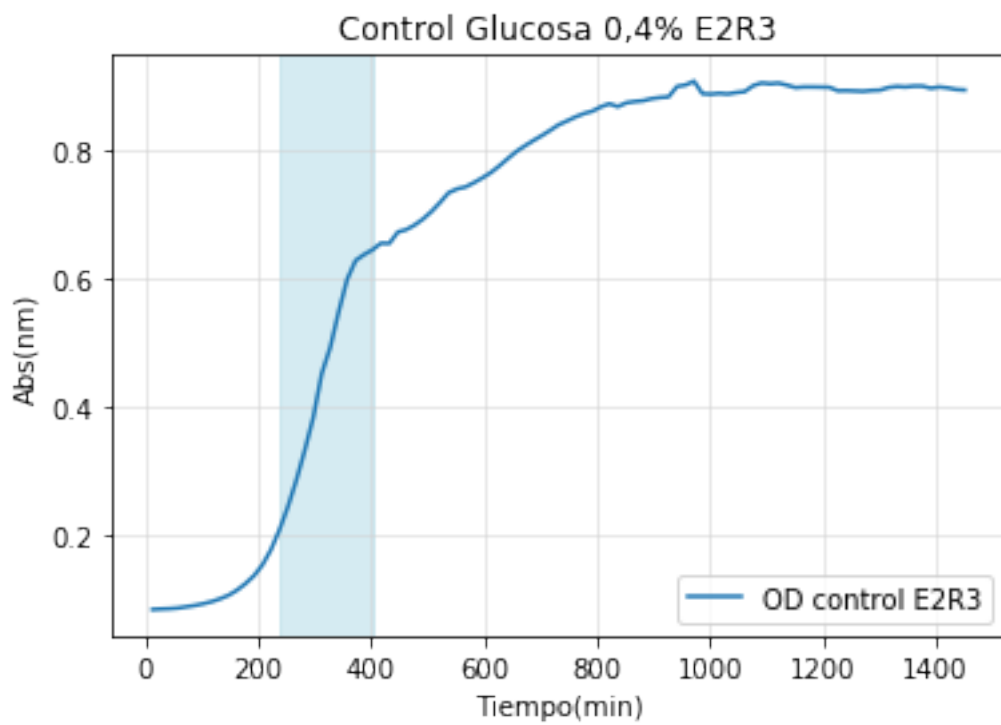
```



```
A=2.339292e+00  
um=9.405123e-03  
l=1.342624e+02  
Tm=2.257633e+02  
doubpe=7.369889e+01  
ext=1.473978e+02  
Tfinal=3.731611e+02
```

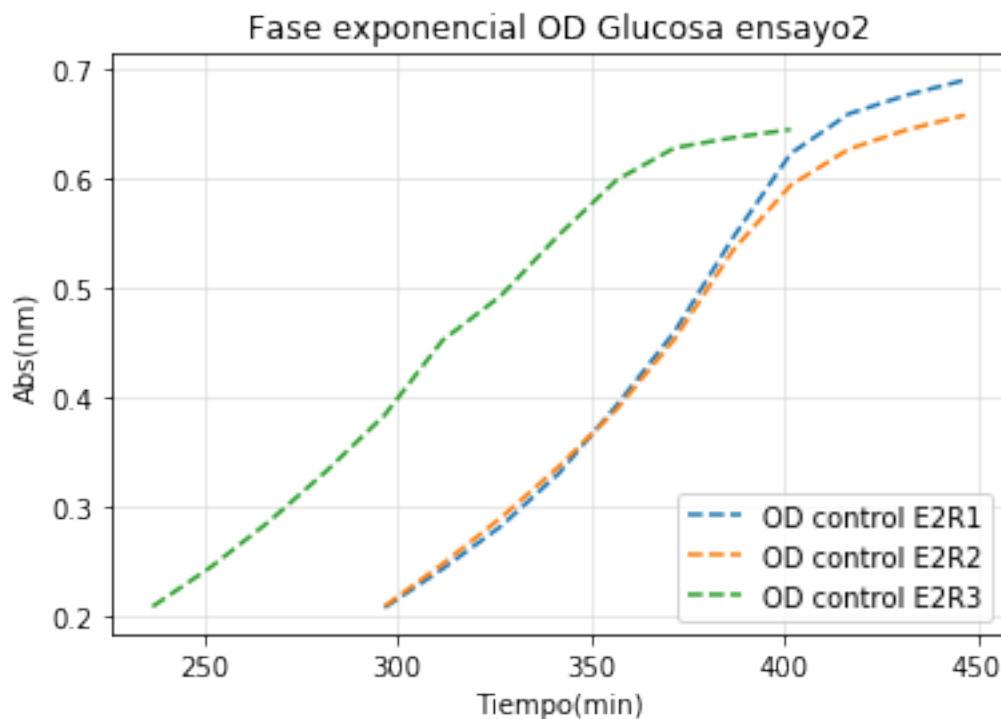
```
Out[63]: <matplotlib.legend.Legend at 0x17893624240>
```





```
In [64]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glucosa ensayo2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odcg151[19:30], '--',label='OD control E2R1')
plt.plot(tt[19:30],odcg152[19:30], '--',label='OD control E2R2')
plt.plot(tt[15:27],odcg153[15:27], '--',label='OD control E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[64]: <matplotlib.legend.Legend at 0x17893734fd0>



```
In [65]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 1
y4= np.log(odcg151)-np.log(np.min(odcg151))
print('Min OD = %e'%((np.min(odcg151))))
evaly, params=Function_fit(tt,y4,0,-1, title = 'Ajuste control glicerol 0,2% E2R1')
A4= params[0]
um4=params[1]
l4=params[2]
print('A=%e'%(A4))
print('um=%e'%(um4))
print('l=%e'%(l4))
```

```

#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+14)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

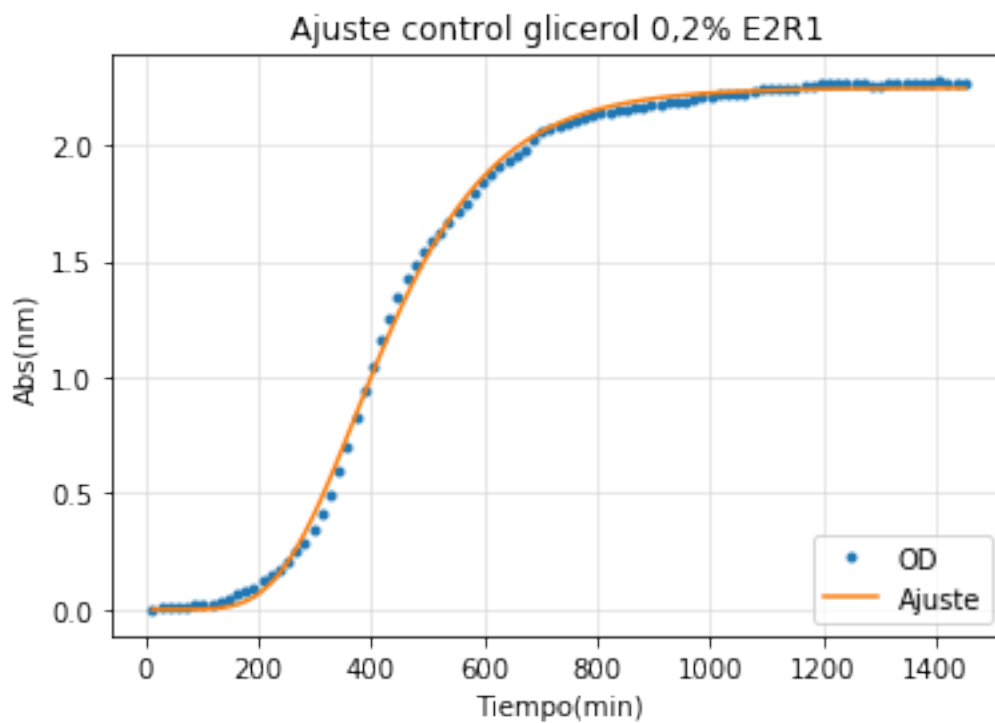
#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[24]
y2=tt[44]
plt.figure()
plt.title('Control Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl151,label='OD control E2R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:45],odcgl151[24:45],label='OD control E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

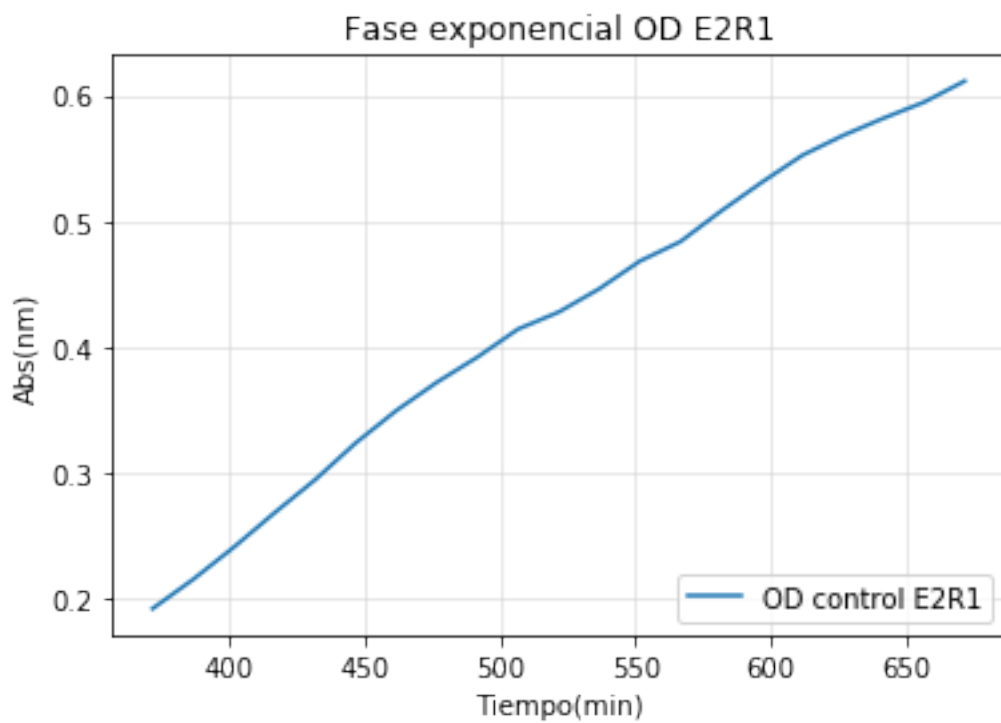
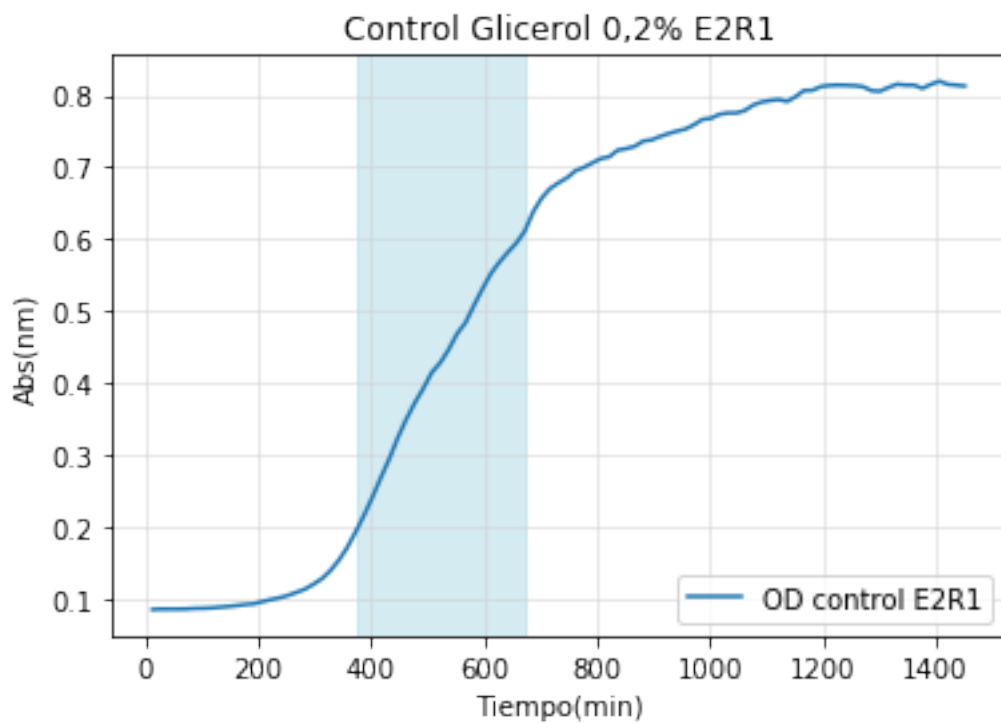
Min OD = 8.425000e-02

[ 2.24595733e+00 6.08336131e-03 2.34248586e+02]



```
A=2.245957e+00  
um=6.083361e-03  
l=2.342486e+02  
Tm=3.700685e+02  
doubpe=1.139415e+02  
ext=2.848537e+02  
Tfinal=6.549222e+02
```

```
Out[65]: <matplotlib.legend.Legend at 0x178938a8748>
```



```

In [66]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 2
y5= np.log(odcgl152)-np.log(np.min(odcgl152))
print('Min OD = %e'%((np.min(odcgl152))))
evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 0,2% E2R2')
A5= params[0]
um5=params[1]
l5=params[2]
print('A=%e'%(A5))
print('um=%e'%(um5))
print('l=%e'%(l5))

#Cálculo datos para determinar extensión de la fase exponencial
tm5=((A5/(np.exp(1)*um5))+l5)
print('Tm=%e'%(tm5))
t25=((np.log(2))/um5)
print('doubpe=%e'%(t25))
extdp5=2.5*t25
print('ext=%e'%extdp5)
ttot5=tm5+extdp5
print('Tfinal=%e'%ttot5)

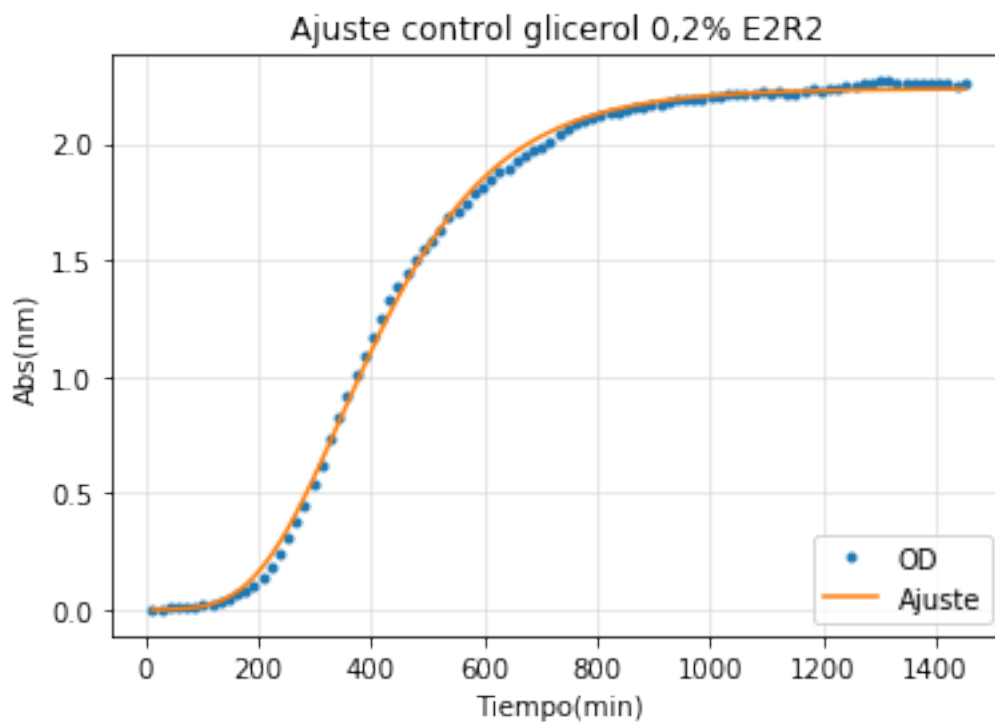
#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[23]
y2=tt[45]
plt.figure()
plt.title('Control Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl152,label='OD control E2R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[23:46],odcgl152[23:46],label='OD control E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

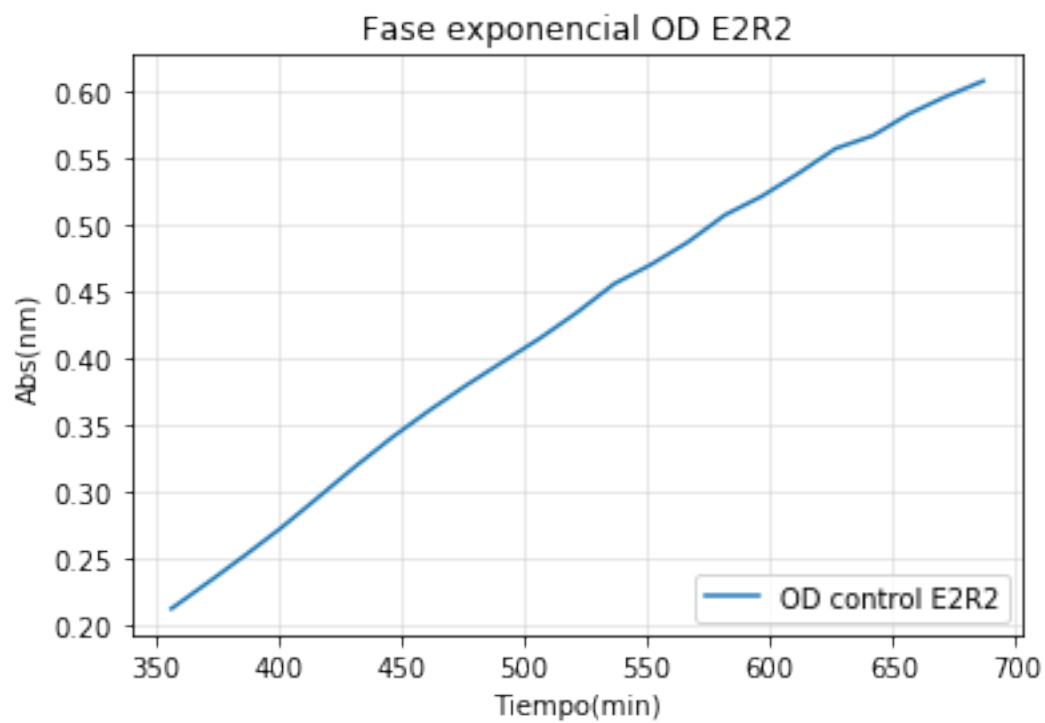
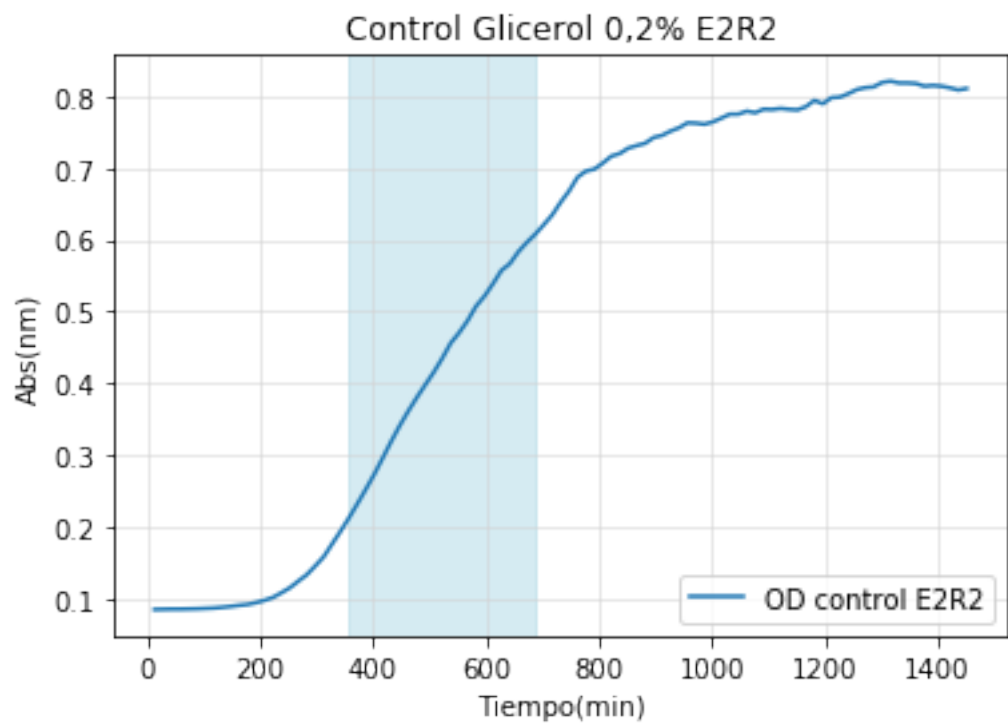
Min OD = 8.500000e-02

[ 2.23521178e+00 5.43425313e-03 1.94280385e+02]



```
A=2.235212e+00  
um=5.434253e-03  
l=1.942804e+02  
Tm=3.455962e+02  
doubpe=1.275515e+02  
ext=3.188788e+02  
Tfinal=6.644750e+02
```

```
Out[66]: <matplotlib.legend.Legend at 0x17892f649b0>
```





```

In [67]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 3
y6= np.log(odcgl153)-np.log(np.min(odcgl153))
print('Min OD = %e'%((np.min(odcgl153))))
evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 0,2% E2R3')
A6= params[0]
um6=params[1]
l6=params[2]
print('A=%e'%(A6))
print('um=%e'%(um6))
print('l=%e'%(l6))

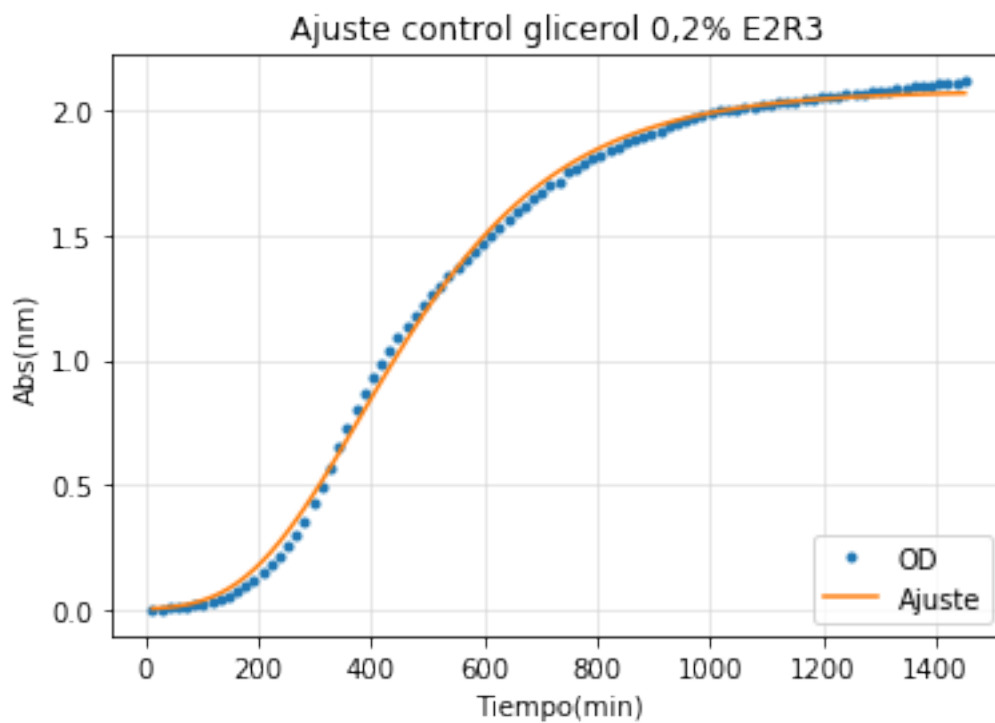
#Cálculo datos para determinar extensión de la fase exponencial
tm6=((A6/(np.exp(1)*um6))+l6)
print('Tm=%e'%(tm6))
t26=((np.log(2))/um6)
print('doubpe=%e'%(t26))
extdp6=2.5*t26
print('ext=%e'%extdp6)
ttot6=tm6+extdp6
print('Tfinal=%e'%ttot6)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
y2=tt[56]
plt.figure()
plt.title('Control Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl153,label='OD control E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:57],odcgl153[25:57],label='OD control E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

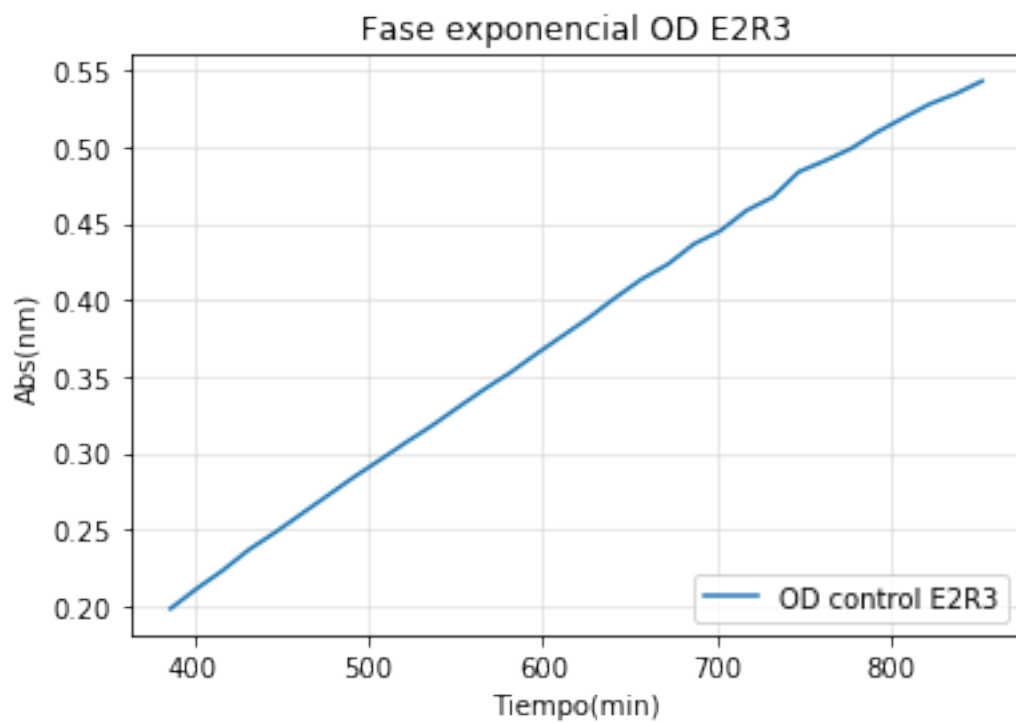
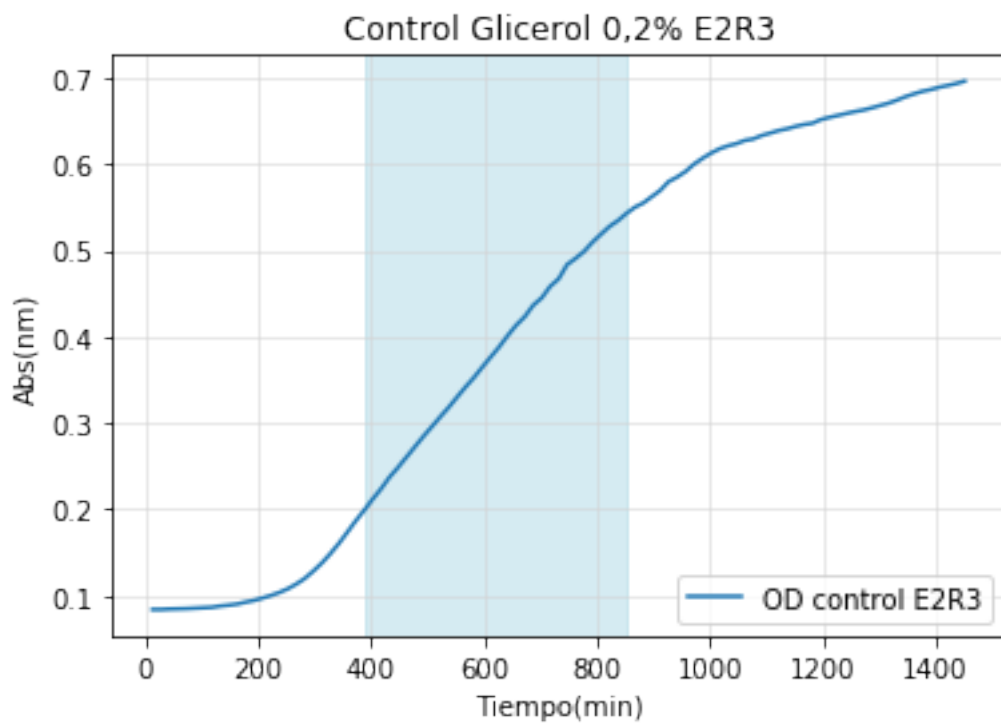
Min OD = 8.375000e-02
[ 2.07943812e+00  3.85150055e-03  1.79490460e+02]

```



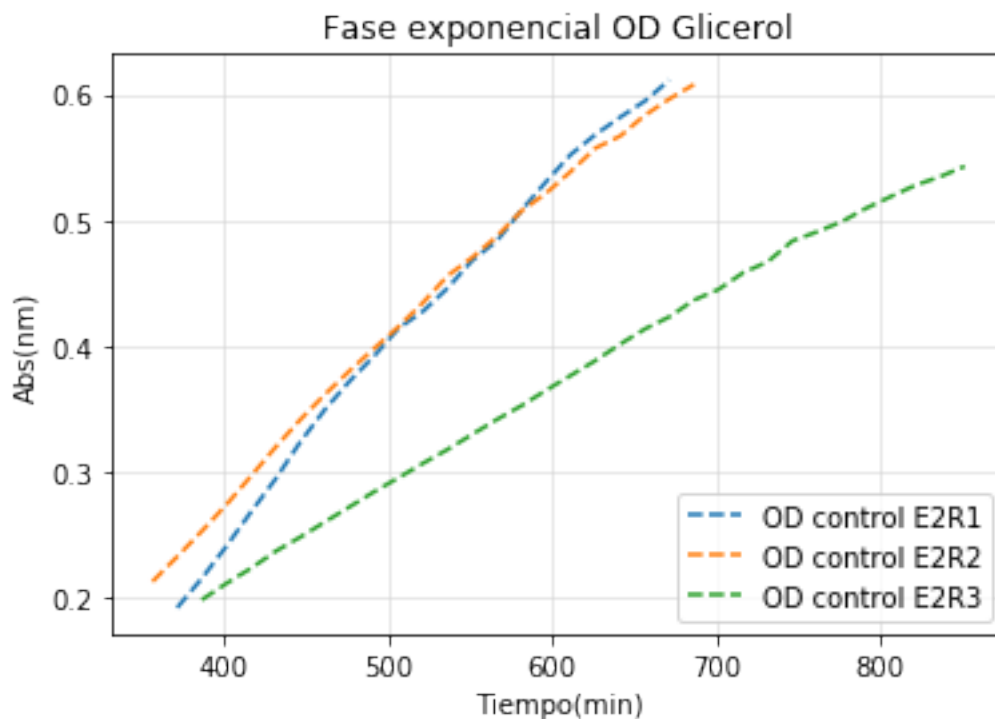
```
A=2.079438e+00  
um=3.851501e-03  
l=1.794905e+02  
Tm=3.781098e+02  
doubpe=1.799681e+02  
ext=4.499202e+02  
Tfinal=8.280300e+02
```

```
Out[67]: <matplotlib.legend.Legend at 0x1788fe17f60>
```



```
In [68]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glicerol')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:45],odcg1151[24:45], '--',label='OD control E2R1')
plt.plot(tt[23:46],odcg1152[23:46], '--',label='OD control E2R2')
plt.plot(tt[25:57],odcg1153[25:57], '--',label='OD control E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

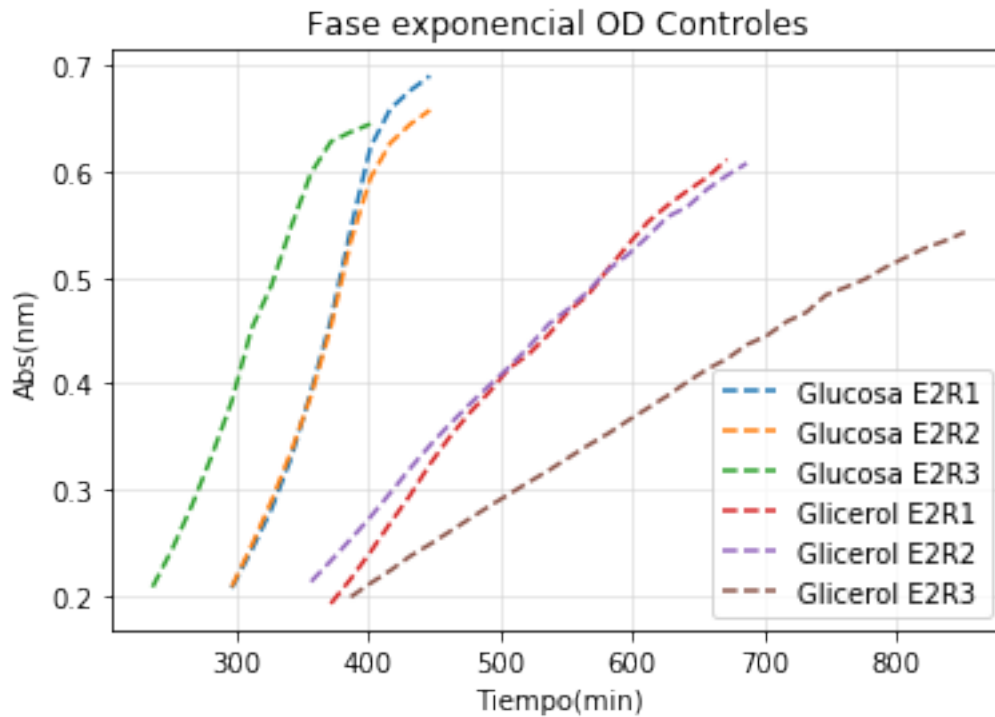
Out[68]: <matplotlib.legend.Legend at 0x178932a8da0>



```
In [69]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Controles')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odcg151[19:30], '--',label='Glucosa E2R1')
plt.plot(tt[19:30],odcg152[19:30], '--',label='Glucosa E2R2')
plt.plot(tt[15:27],odcg153[15:27], '--',label='Glucosa E2R3')
plt.plot(tt[24:45],odcg1151[24:45], '--',label='Glicerol E2R1')
plt.plot(tt[23:46],odcg1152[23:46], '--',label='Glicerol E2R2')
plt.plot(tt[25:57],odcg1153[25:57], '--',label='Glicerol E2R3')
```

```
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[69]: <matplotlib.legend.Legend at 0x178931cb390>



```
In [70]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 1
y7 = np.log(odcg181)-np.log(np.min(odcg181))
print('Min OD = %e'%((np.min(odcg181))))
evaly, params=Function_fit(tt,y7,0,-1,title = 'Ajuste control glucosa 0,4% E3R1')
A7 = params[0]
um7=params[1]
l7=params[2]
print('A=%e'%(A7))
print('um=%e'%(um7))
print('l=%e'%(l7))

#Cálculo datos para determinar extensión de la fase exponencial
tm7=((A7/(np.exp(1)*um7))+l7)
print('Tm=%e'%(tm7))
t27=((np.log(2))/um7)
print('doubpe=%e'%(t27))
extdp7=2*t27
print('ext=%e'%extdp7)
```

```

ttot7=tm7+extdp7
print('Tfinal=%e'%ttot7)

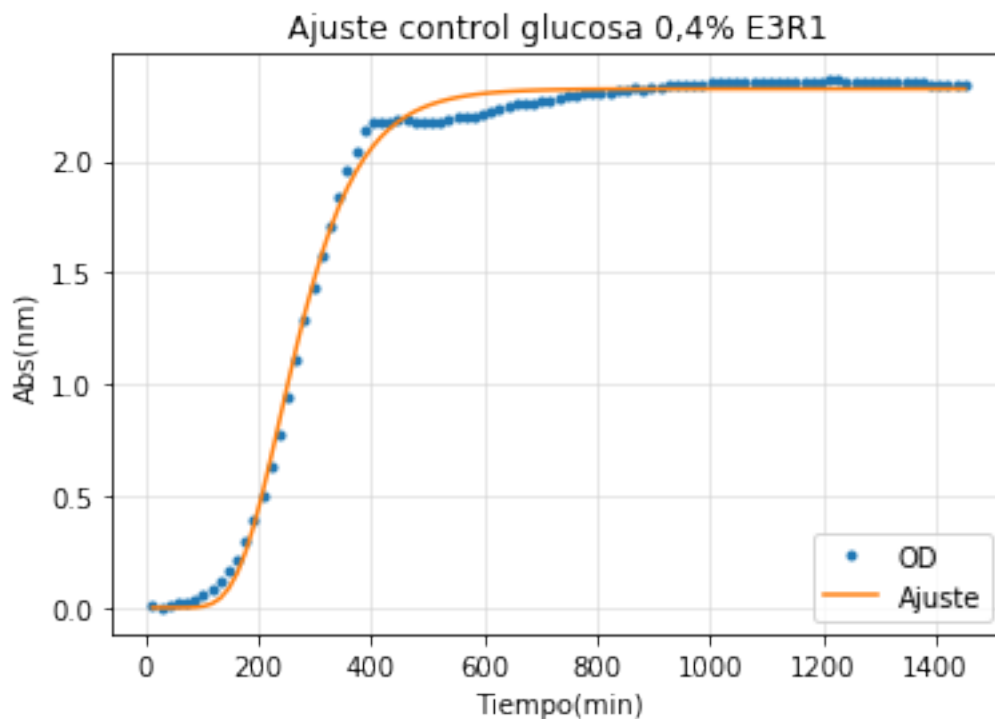
#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
y2=tt[25]
plt.figure()
plt.title('Control Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg181,label='OD control E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],odcg181[16:26],label='OD control E3R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

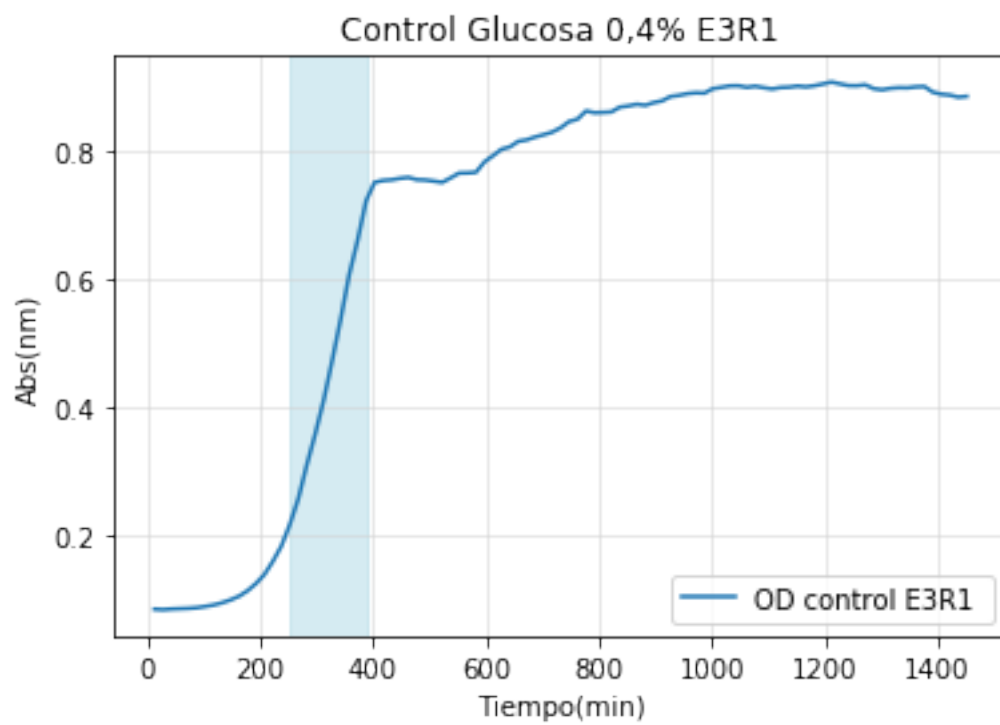
Min OD = 8.550000e-02

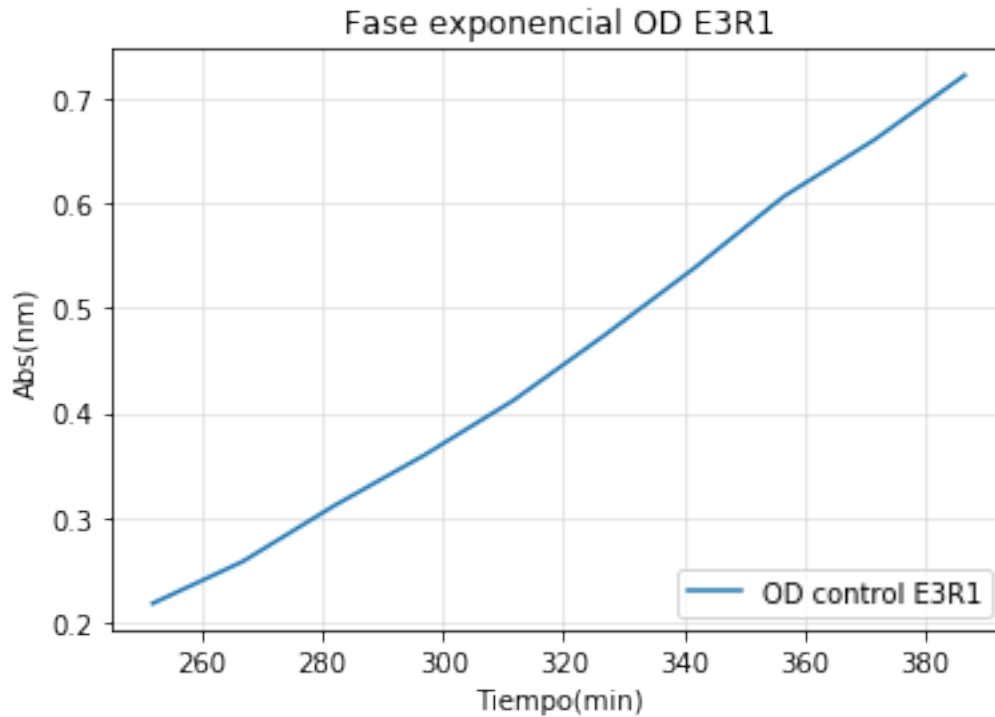
[ 2.32234986e+00 1.11487857e-02 1.61538943e+02]



A=2.322350e+00  
um=1.114879e-02  
l=1.615389e+02  
Tm=2.381701e+02  
doubpe=6.217244e+01  
ext=1.243449e+02  
Tfinal=3.625150e+02

Out[70]: <matplotlib.legend.Legend at 0x1789399ce48>





```
In [71]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y8= np.log(odcg182)-np.log(np.min(odcg182))
print('Min OD = %e'%((np.min(odcg182))))
evaly, params=Function_fit(tt,y8,0,-1, title = 'Ajuste control glucosa 0,4% E3R2')
A8 = params[0]
um8=params[1]
l8=params[2]
print('A=%e'%(A8))
print('um=%e'%(um8))
print('l=%e'%(l8))

#Cálculo datos para determinar extensión de la fase exponencial
tm8=((A8/(np.exp(1)*um8))+l8)
print('Tm=%e'%(tm8))
t28=((np.log(2))/um8)
print('doubpe=%e'%(t28))
extdp8=2*t28
print('ext=%e'%extdp8)
ttot8=tm8+extdp8
print('Tfinal=%e'%ttot8)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```



```

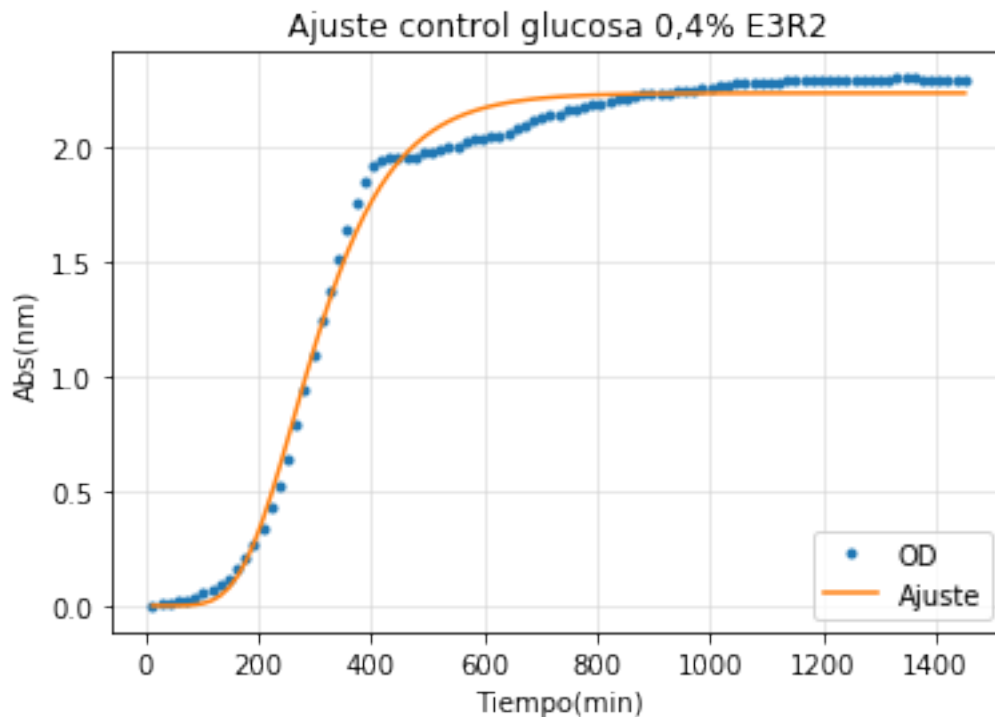
y2=tt[29]
plt.figure()
plt.title('Control Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg182,label='OD control E3R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:30],odcg182[17:30],label='OD control E3R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.375000e-02

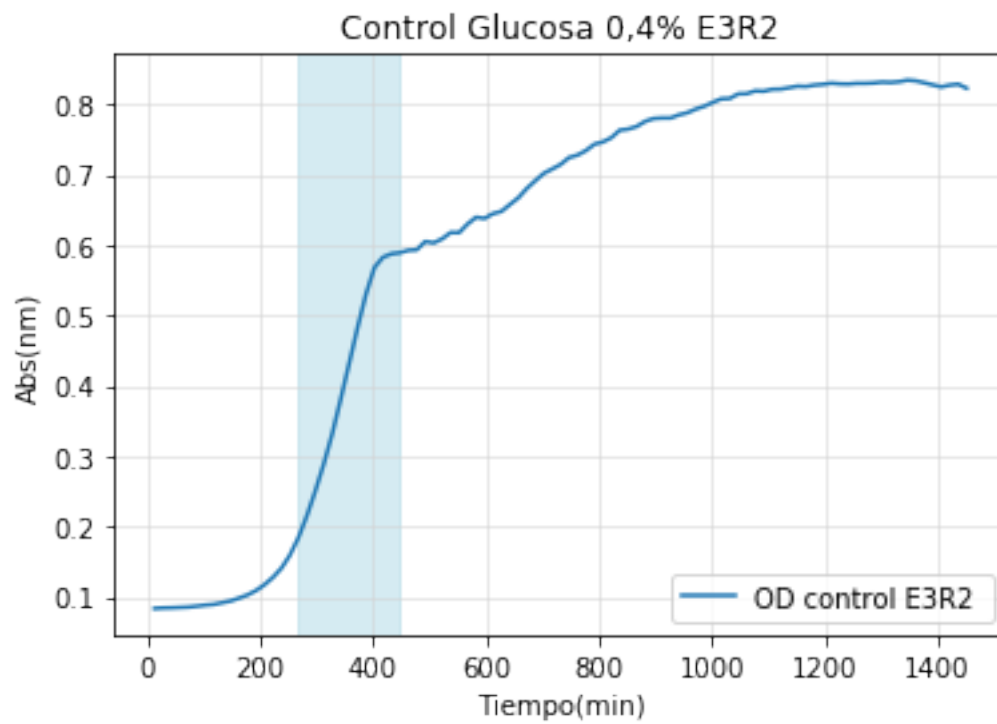
[ 2.23365370e+00 8.60171922e-03 1.67065878e+02]

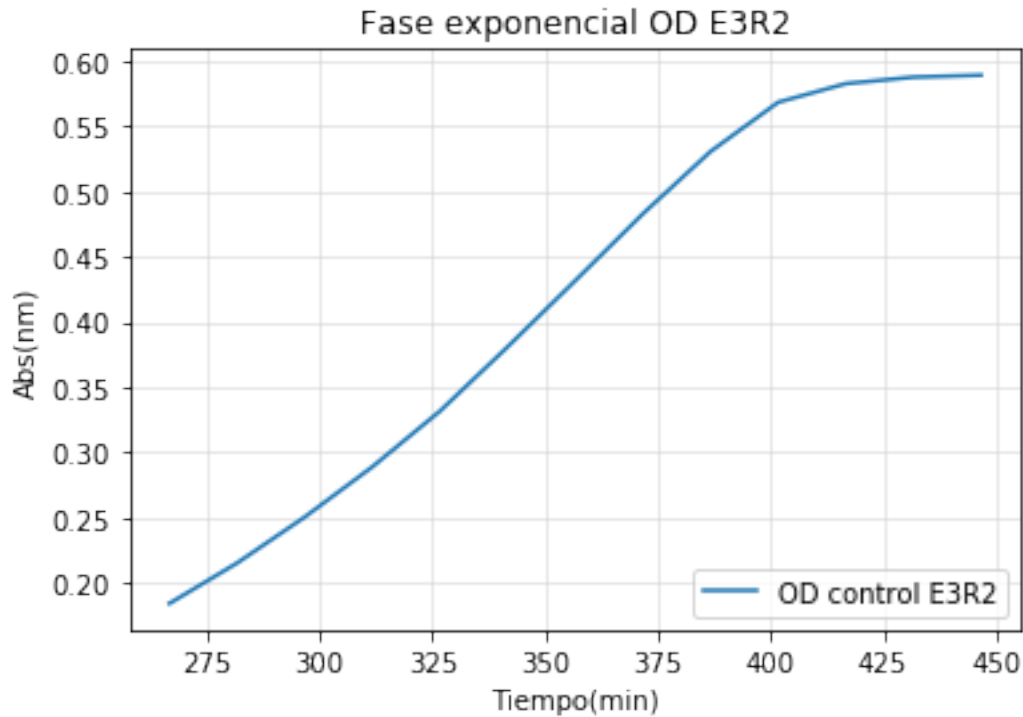


A=2.233654e+00  
um=8.601719e-03

$l=1.670659e+02$   
 $T_m=2.625951e+02$   
 $doubpe=8.058240e+01$   
 $ext=1.611648e+02$   
 $T_{final}=4.237599e+02$

Out[71]: <matplotlib.legend.Legend at 0x17894b5d898>





```
In [72]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y9= np.log(odcg183)-np.log(np.min(odcg183))
print('Min OD = %e'%((np.min(odcg183))))
evaly, params=Function_fit(tt,y9,0,-1, title = 'Ajuste control glucosa 0,4% E3R3')
A9= params[0]
um9=params[1]
l9=params[2]
print('A=%e'%(A9))
print('um=%e'%(um9))
print('l=%e'%(l9))

#Cálculo datos para determinar extensión de la fase exponencial
tm9=((A9/(np.exp(1)*um9))+l9)
print('Tm=%e'%(tm9))
t29=((np.log(2))/um9)
print('doubpe=%e'%(t29))
extdp9=2*t29
print('ext=%e'%extdp9)
ttot9=tm9+extdp9
print('Tfinal=%e'%ttot9)

#Delimitación fase exponencial en grafico con OD/tiempo
```

```

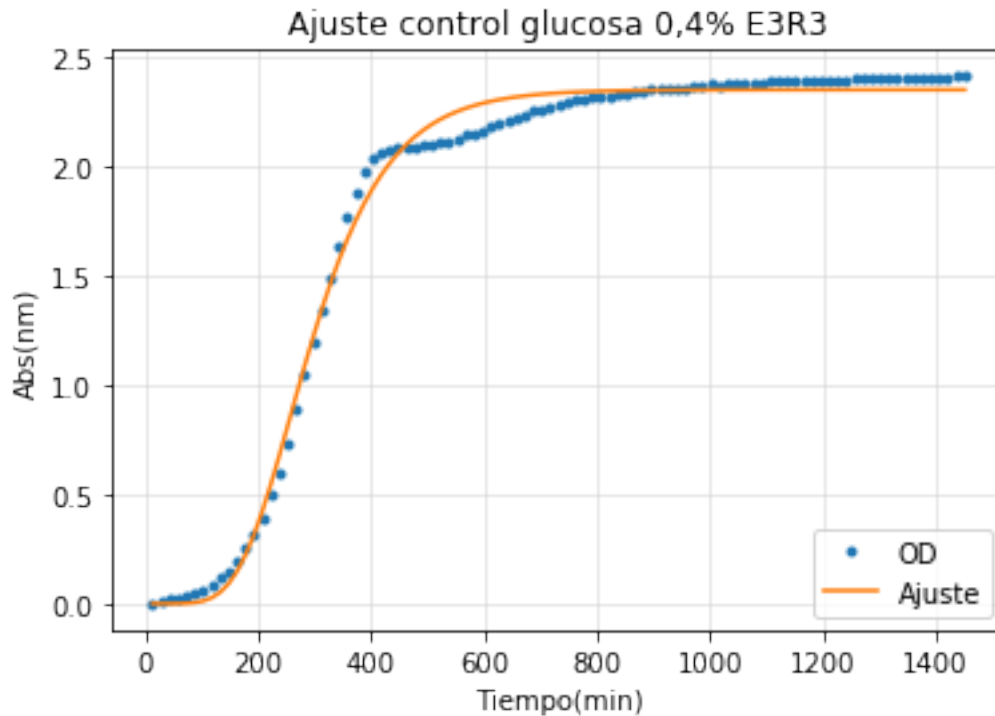
y1=tt[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg183,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],odcg183[17:29],label='OD control E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

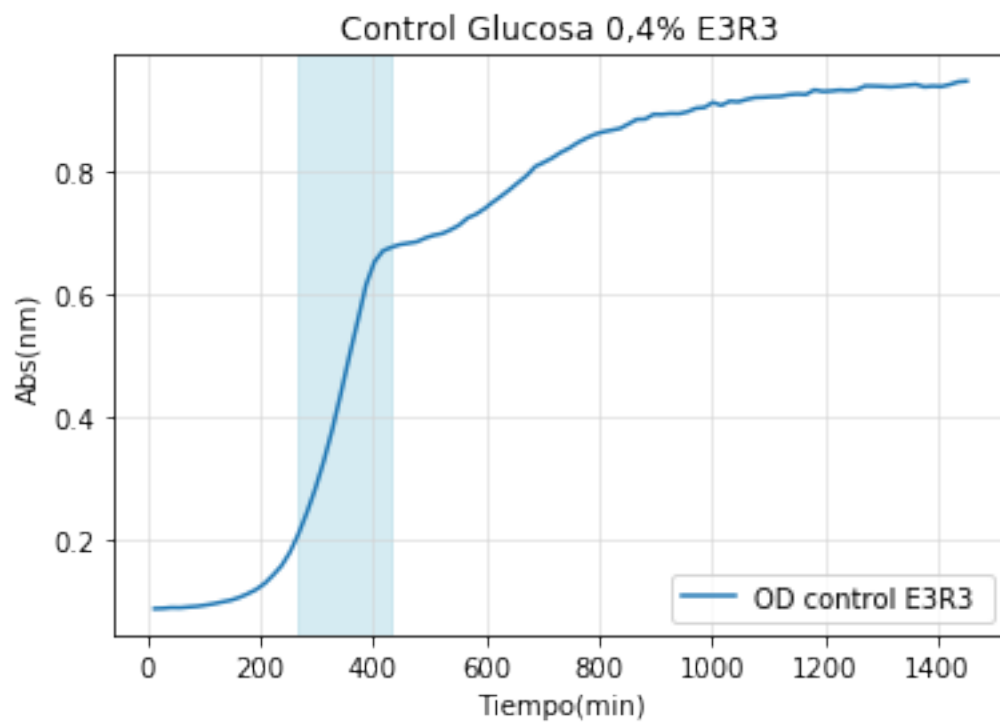
Min OD = 8.525000e-02

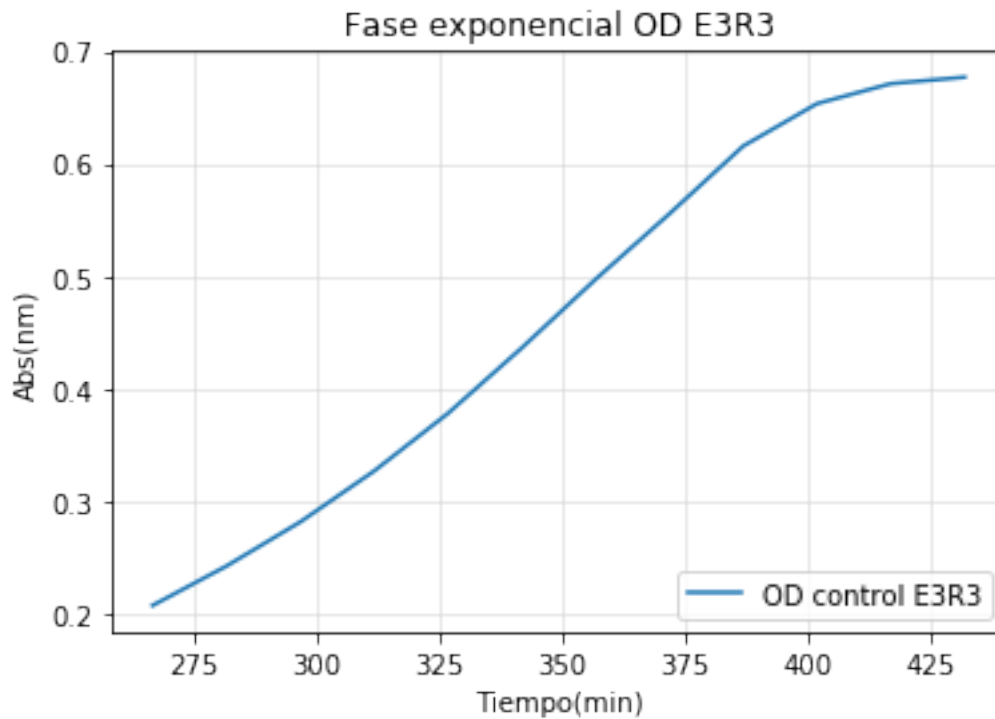
[ 2.34758891e+00 9.20947423e-03 1.63131184e+02]



A=2.347589e+00  
um=9.209474e-03  
l=1.631312e+02  
Tm=2.569074e+02  
doubpe=7.526458e+01  
ext=1.505292e+02  
Tfinal=4.074366e+02

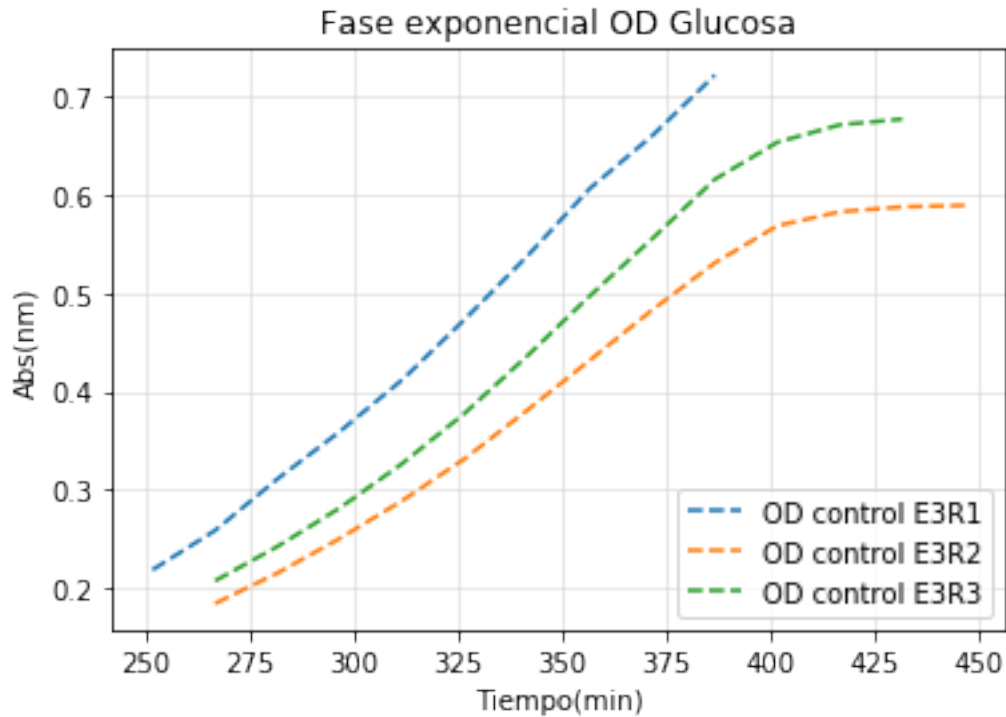
Out[72]: <matplotlib.legend.Legend at 0x17894bb4b00>





```
In [73]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glucosa')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],odcg181[16:26], '--', label='OD control E3R1')
plt.plot(tt[17:30],odcg182[17:30], '--', label='OD control E3R2')
plt.plot(tt[17:29],odcg183[17:29], '--', label='OD control E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[73]: <matplotlib.legend.Legend at 0x17893a2a2e8>
```



```
In [74]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 1
y10= np.log(odcgl181)-np.log(np.min(odcgl181))
print('Min OD = %e'%((np.min(odcgl181))))
evaly, params=Function_fit(tt,y10,0,-1, title = 'Ajuste control glicerol 0,2% E3R1')
A10= params[0]
um10=params[1]
l10=params[2]
print('A=%e'%(A10))
print('um=%e'%(um10))
print('l=%e'%(l10))

#Cálculo datos para determinar extensión de la fase exponencial
tm10=((A10/(np.exp(1)*um10))+l10)
print('Tm=%e'%(tm10))
t210=((np.log(2))/um10)
print('doubpe=%e'%(t210))
extdp10=2.5*t210
print('ext=%e'%extdp10)
ttot10=tm10+extdp10
print('Tfinal=%e'%ttot10)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

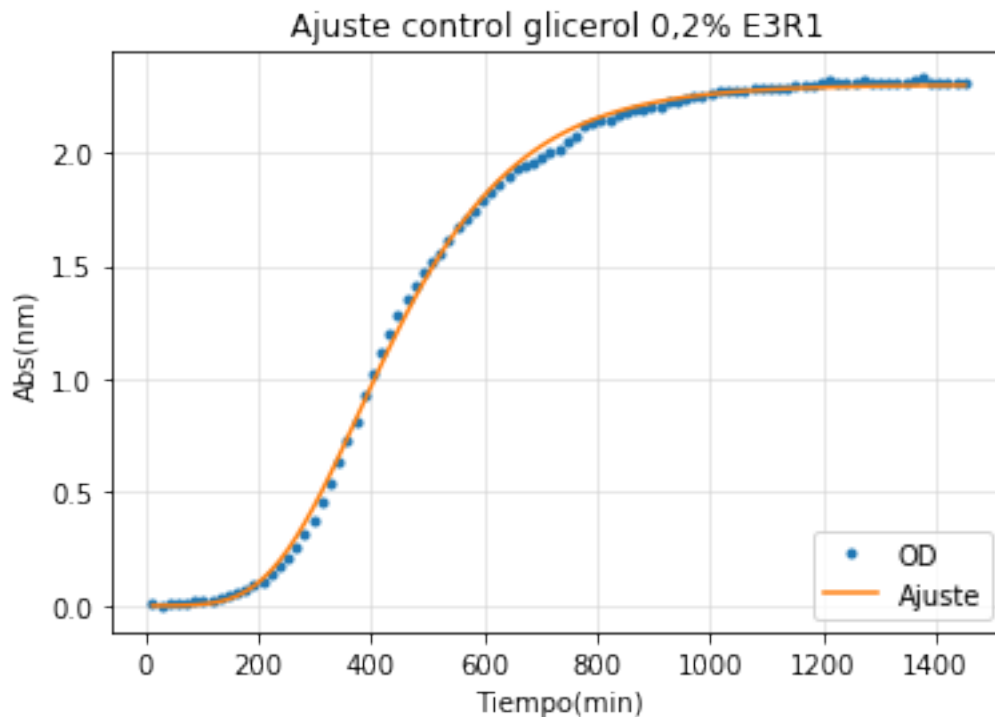
y2=tt[47]
plt.figure()
plt.title('Control Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1181,label='OD control E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:48],odcg1181[25:48],label='OD control E3R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.525000e-02

[ 2.29993135e+00 5.42425453e-03 2.20607778e+02]

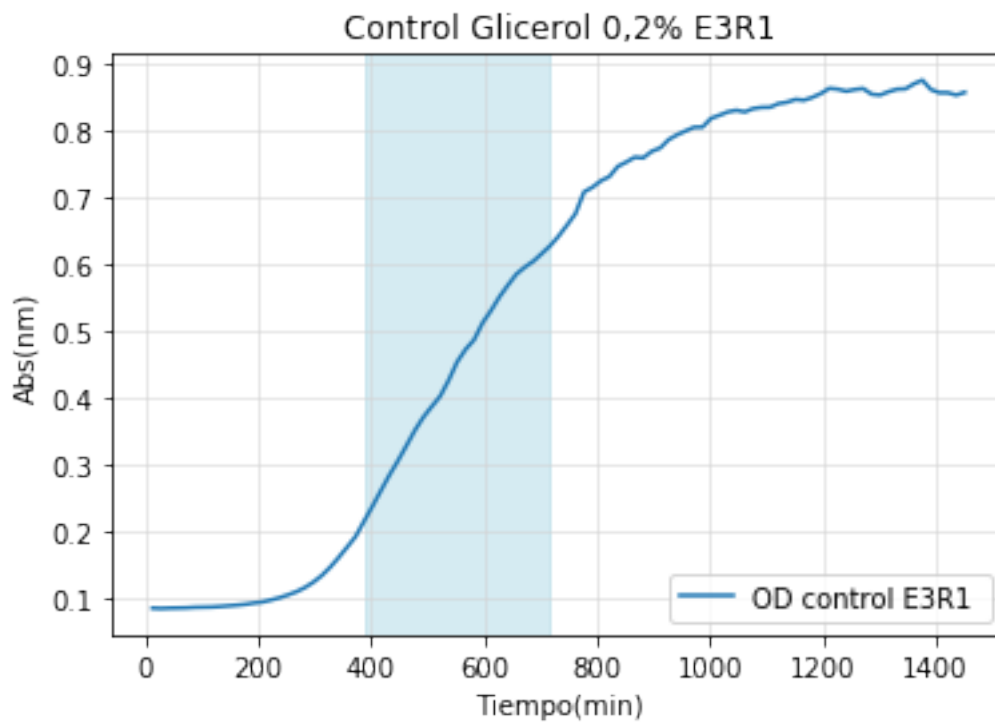


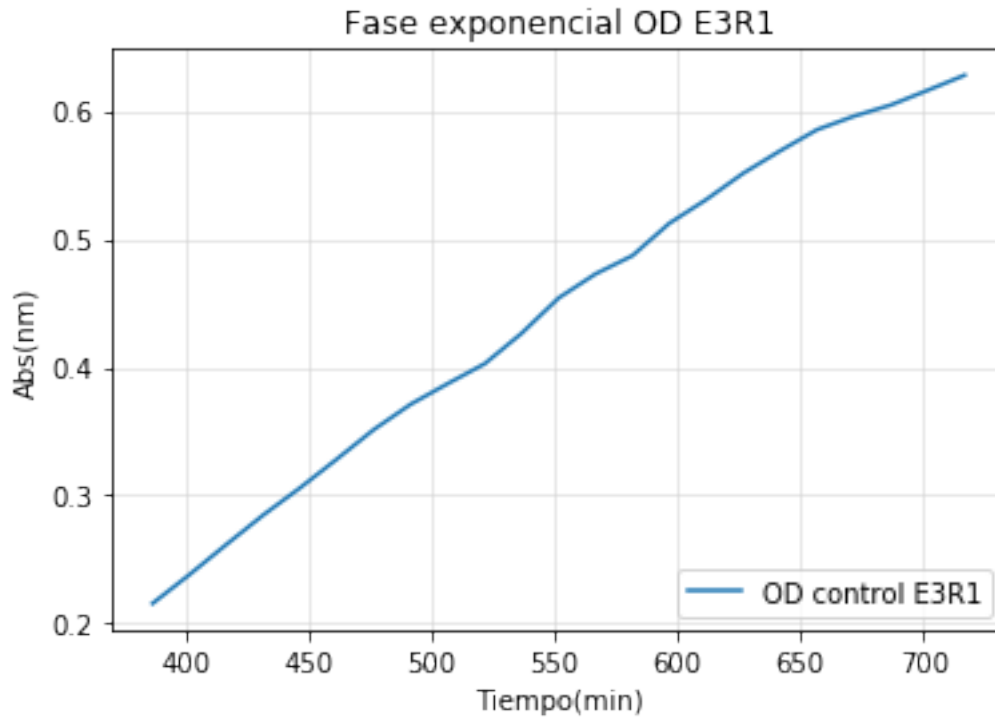
A=2.299931e+00  
um=5.424255e-03



l=2.206078e+02  
Tm=3.765919e+02  
doubpe=1.277866e+02  
ext=3.194666e+02  
Tfinal=6.960584e+02

Out[74]: <matplotlib.legend.Legend at 0x1789196f588>





```
In [75]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 2
y11= np.log(odcgl182)-np.log(np.min(odcgl182))
print('Min OD = %e'%((np.min(odcgl182))))
evaly, params=Function_fit(tt,y11,0,-1, title = 'Ajuste control glicerol 0,2% E3R2')
A11= params[0]
um11=params[1]
l11=params[2]
print('A=%e'%(A11))
print('um=%e'%(um11))
print('l=%e'%(l11))

#Cálculo datos para determinar extensión de la fase exponencial
tm11=((A11/(np.exp(1)*um11))+l11)
print('Tm=%e'%(tm11))
t211=((np.log(2))/um11)
print('doubpe=%e'%(t211))
extdp11=2.5*t211
print('ext=%e'%extdp11)
ttot11=tm11+extdp11
print('Tfinal=%e'%ttot11)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

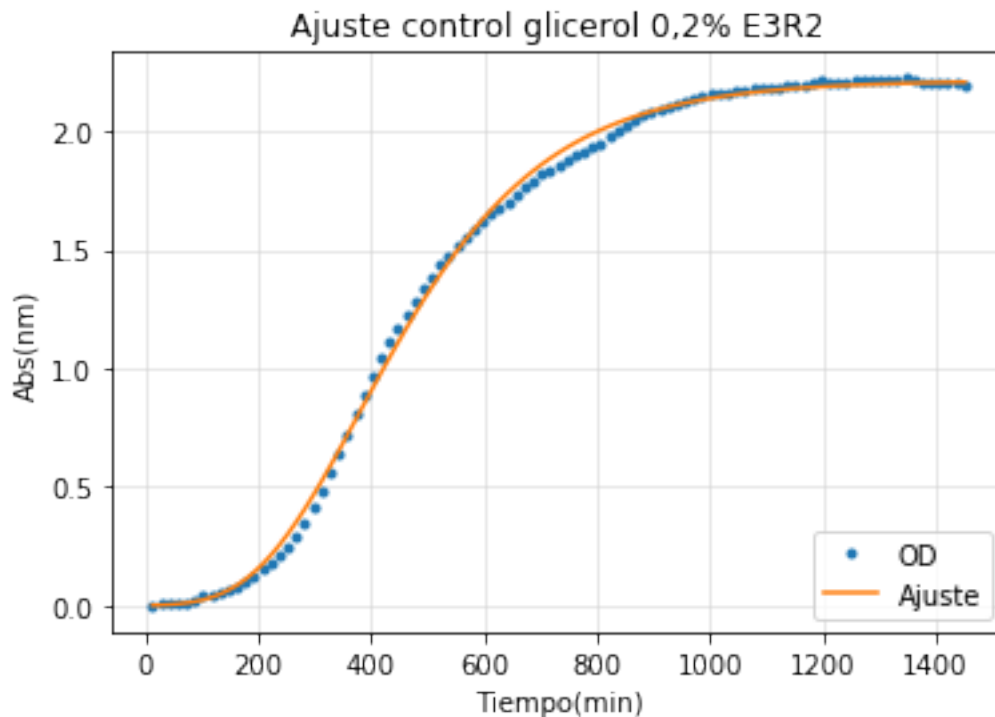
y2=tt[52]
plt.figure()
plt.title('Control Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1182,label='OD control E3R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],odcg1182[25:53],label='OD control E3R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.250000e-02

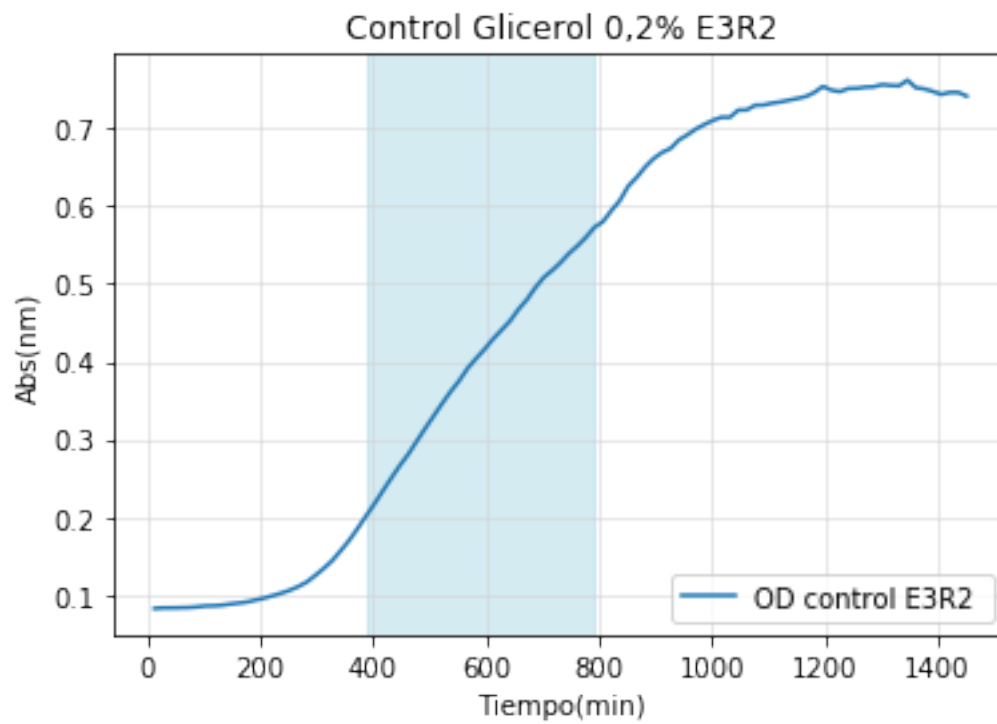
[ 2.21283483e+00 4.40822880e-03 1.94027274e+02]

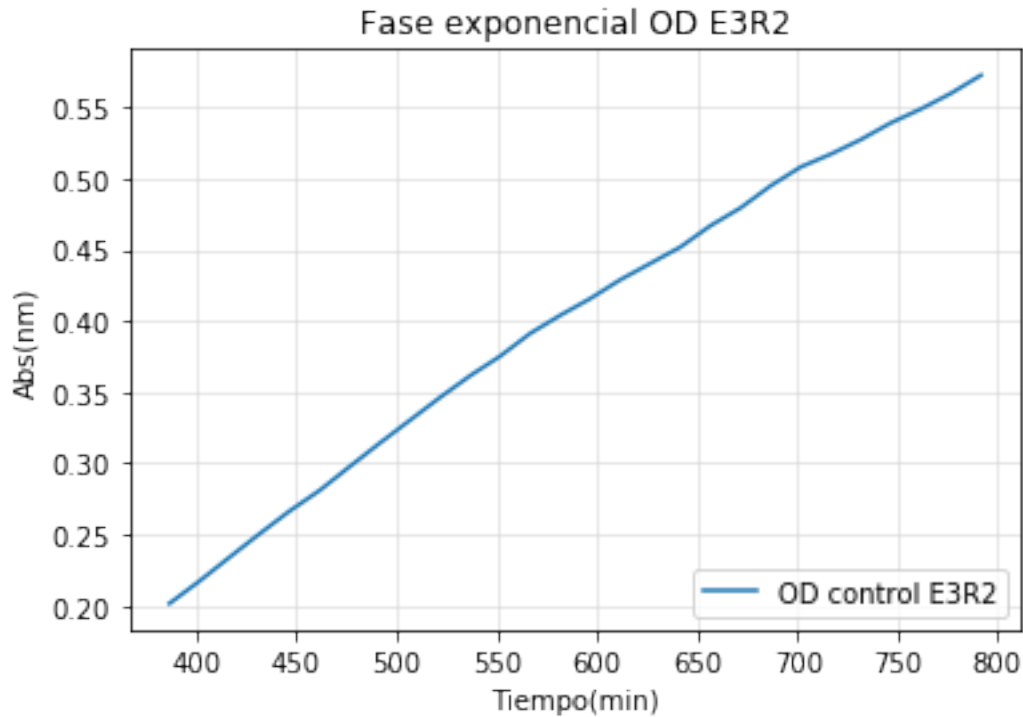


A=2.212835e+00  
um=4.408229e-03

$l=1.940273e+02$   
 $T_m=3.786947e+02$   
 $doubpe=1.572394e+02$   
 $ext=3.930985e+02$   
 $T_{final}=7.717932e+02$

Out[75]: <matplotlib.legend.Legend at 0x17893886c18>





```
In [76]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 3
y12= np.log(odcgl183)-np.log(np.min(odcgl183))
print('Min OD = %e'%((np.min(odcgl183))))
evaly, params=Function_fit(tt,y12,0,-1, title = 'Ajuste control glicerol 0,2% E3R3')
A12= params[0]
um12=params[1]
l12=params[2]
print('A=%e'%(A12))
print('um=%e'%(um12))
print('l=%e'%(l12))

#Cálculo datos para determinar extensión de la fase exponencial
tm12=((A12/(np.exp(1)*um12))+l12)
print('Tm=%e'%(tm12))
t212=((np.log(2))/um12)
print('doubpe=%e'%(t212))
extdp12=2.5*t212
print('ext=%e'%extdp12)
ttot12=tm12+extdp12
print('Tfinal=%e'%ttot12)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[24]
```

```

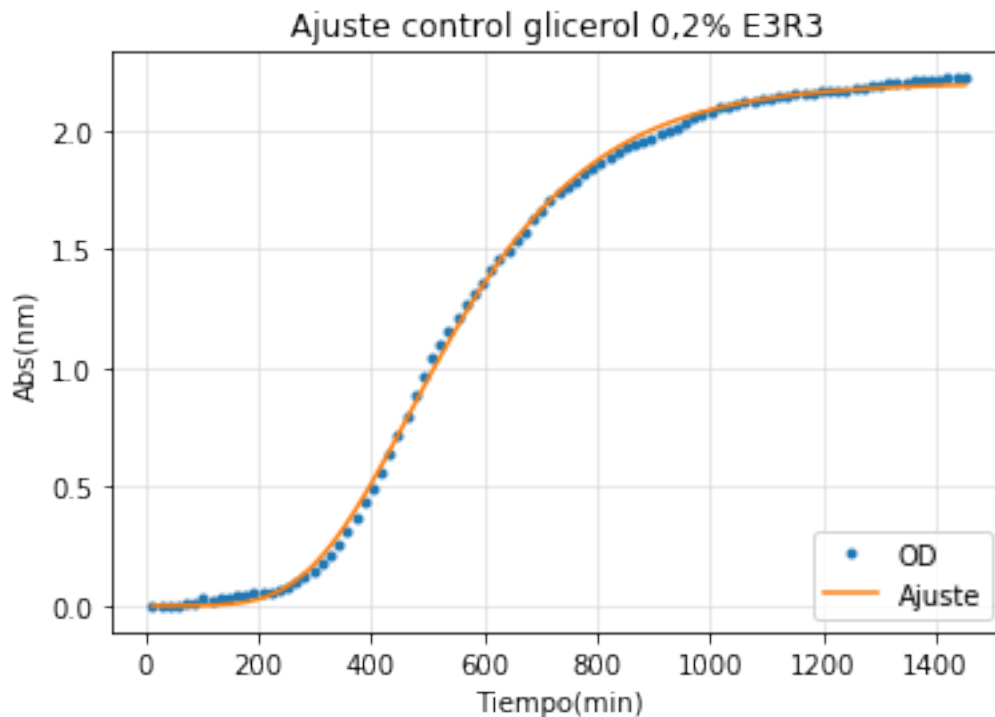
y2=tt[50]
plt.figure()
plt.title('Control Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1183,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:51],odcg1183[24:51],label='OD control E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.275000e-02

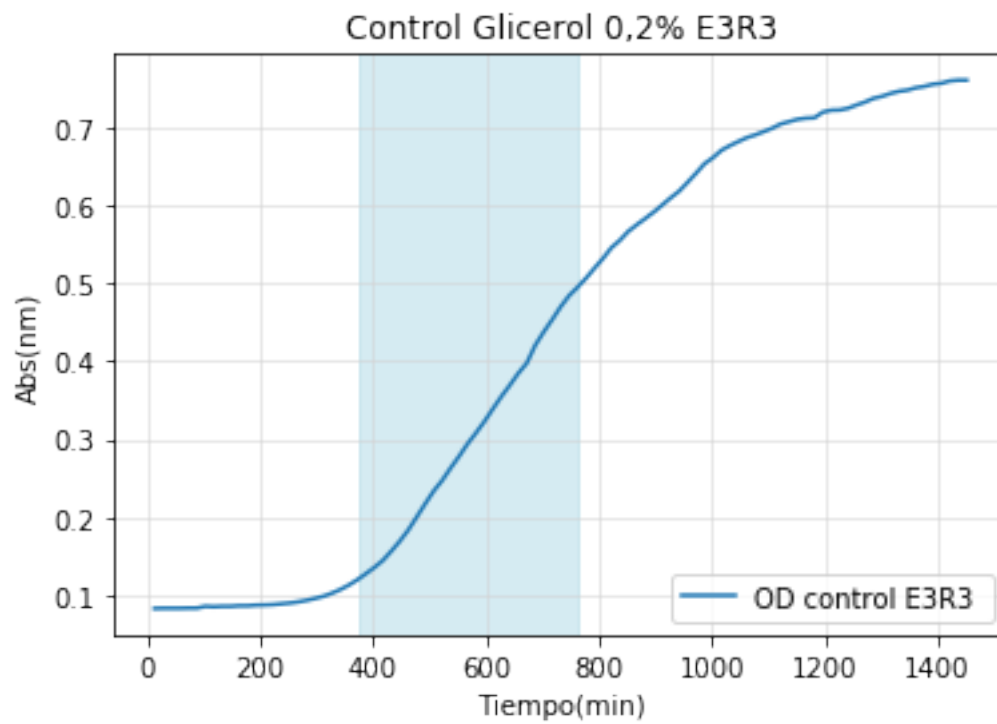
[ 2.19895432e+00 4.44585642e-03 2.85665525e+02]

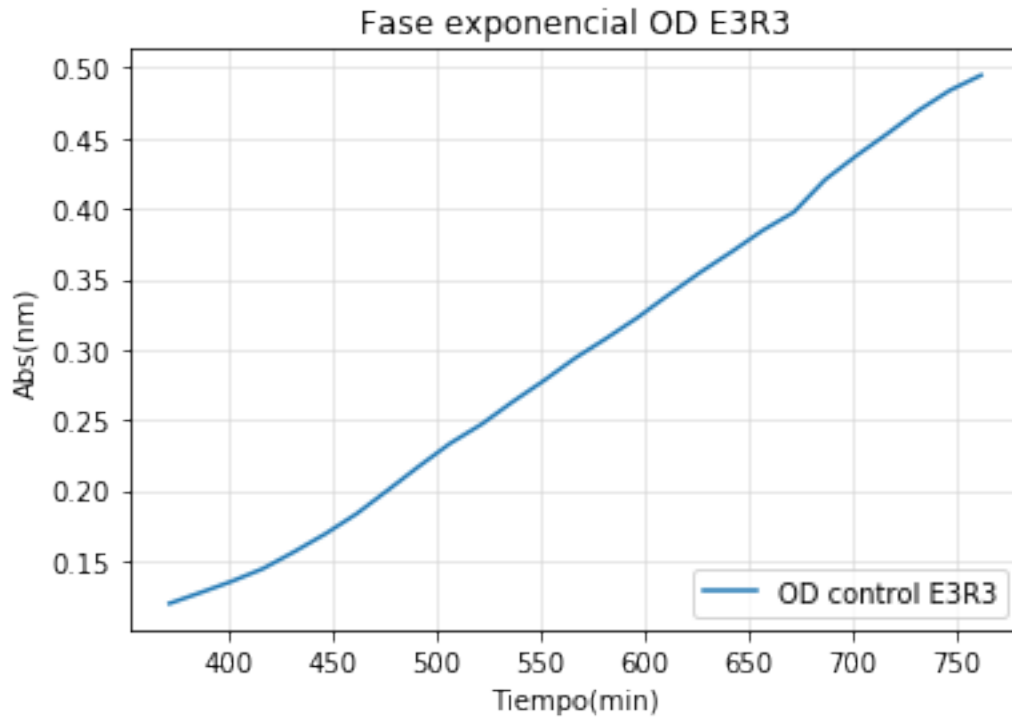


A=2.198954e+00  
um=4.445856e-03

```
l=2.856655e+02  
Tm=4.676215e+02  
doubpe=1.559086e+02  
ext=3.897715e+02  
Tfinal=8.573929e+02
```

Out[76]: <matplotlib.legend.Legend at 0x17894db28d0>

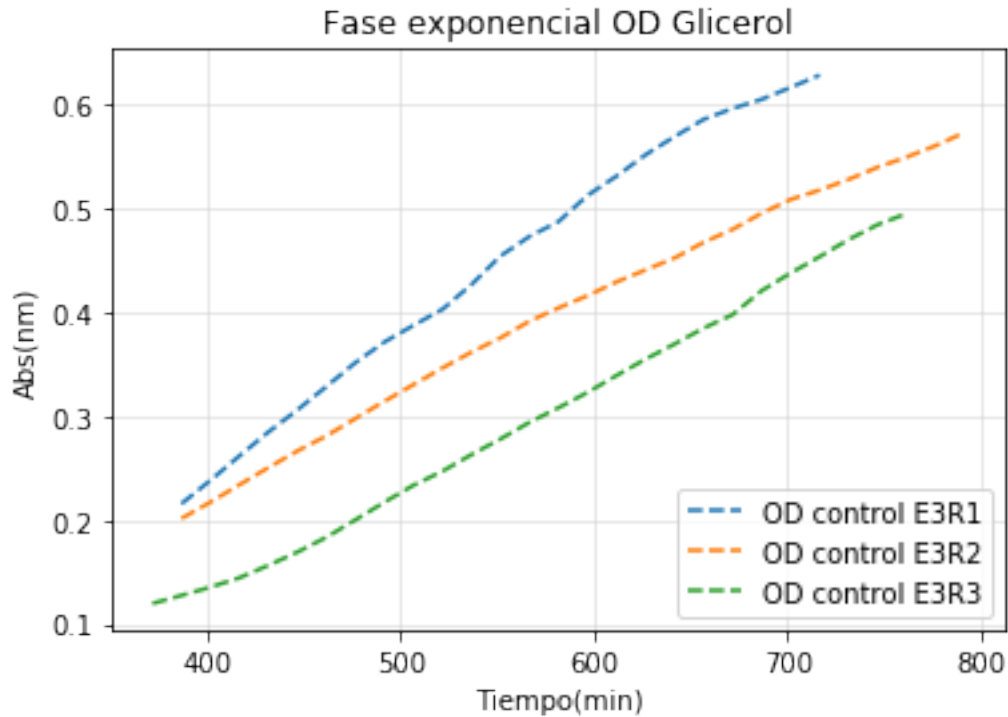




```
In [77]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glicerol')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:48],odcg1181[25:48], '--',label='OD control E3R1')
plt.plot(tt[25:53],odcg1182[25:53], '--',label='OD control E3R2')
plt.plot(tt[24:51],odcg1183[24:51], '--',label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

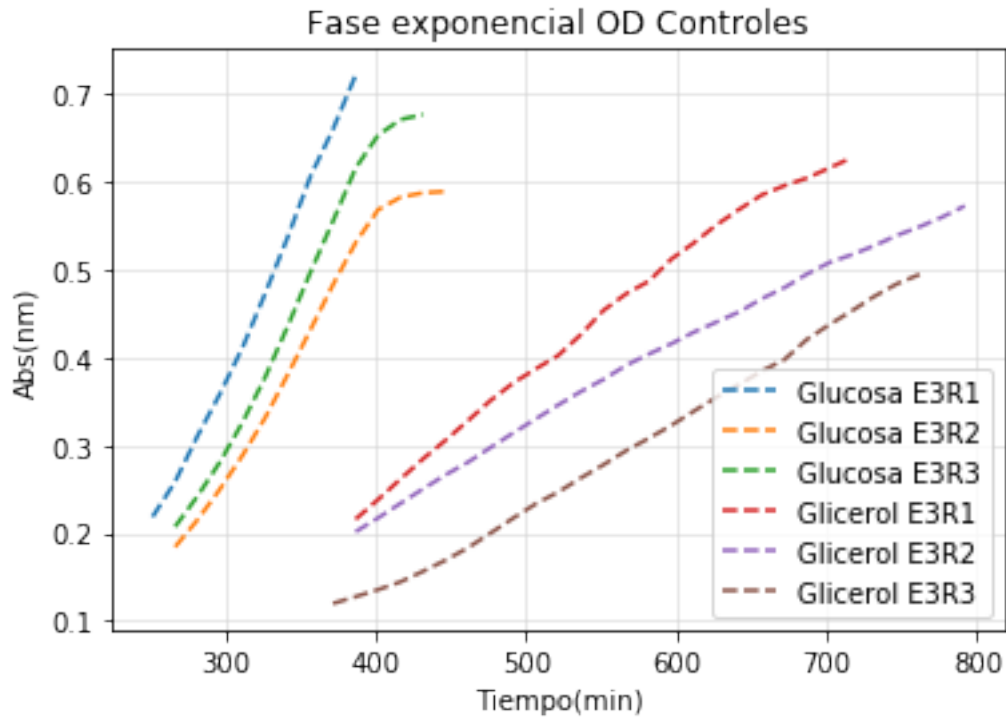
```
Out[77]: <matplotlib.legend.Legend at 0x17894e98a58>
```





```
In [78]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Controles')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],odcg181[16:26], '--', label='Glucosa E3R1')
plt.plot(tt[17:30],odcg182[17:30], '--', label='Glucosa E3R2')
plt.plot(tt[17:29],odcg183[17:29], '--', label='Glucosa E3R3')
plt.plot(tt[25:48],odcg1181[25:48], '--', label='Glicerol E3R1')
plt.plot(tt[25:53],odcg1182[25:53], '--', label='Glicerol E3R2')
plt.plot(tt[24:51],odcg1183[24:51], '--', label='Glicerol E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[78]: <matplotlib.legend.Legend at 0x17894f59c18>
```



```
In [79]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 1
y13 = np.log(odcg121)-np.log(np.min(odcg121))
print('Min OD = %e'%((np.min(odcg121))))
evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste control glucosa 0,4% E4R1')
A13 = params[0]
um13=params[1]
l13=params[2]
print('A=%e'%(A13))
print('um=%e'%(um13))
print('l=%e'%(l13))

#Cálculo datos para determinar extensión de la fase exponencial
tm13=((A13/(np.exp(1)*um13))+l13)
print('Tm=%e'%(tm13))
t213=((np.log(2))/um13)
print('doubpe=%e'%(t213))
extdp13=2*t213
print('ext=%e'%extdp13)
ttot13=tm13+extdp13
print('Tfinal=%e'%ttot13)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```

```

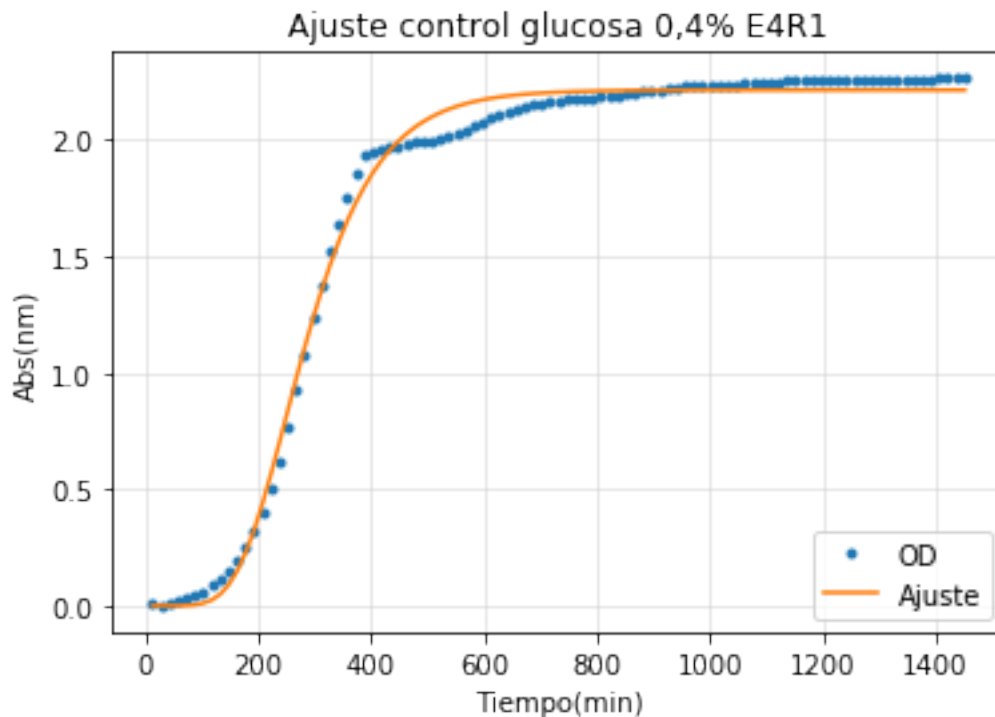
y2=tt[27]
plt.figure()
plt.title('Control Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg121,label='OD control E4R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],odcg121[17:28],label='OD control E4R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.525000e-02

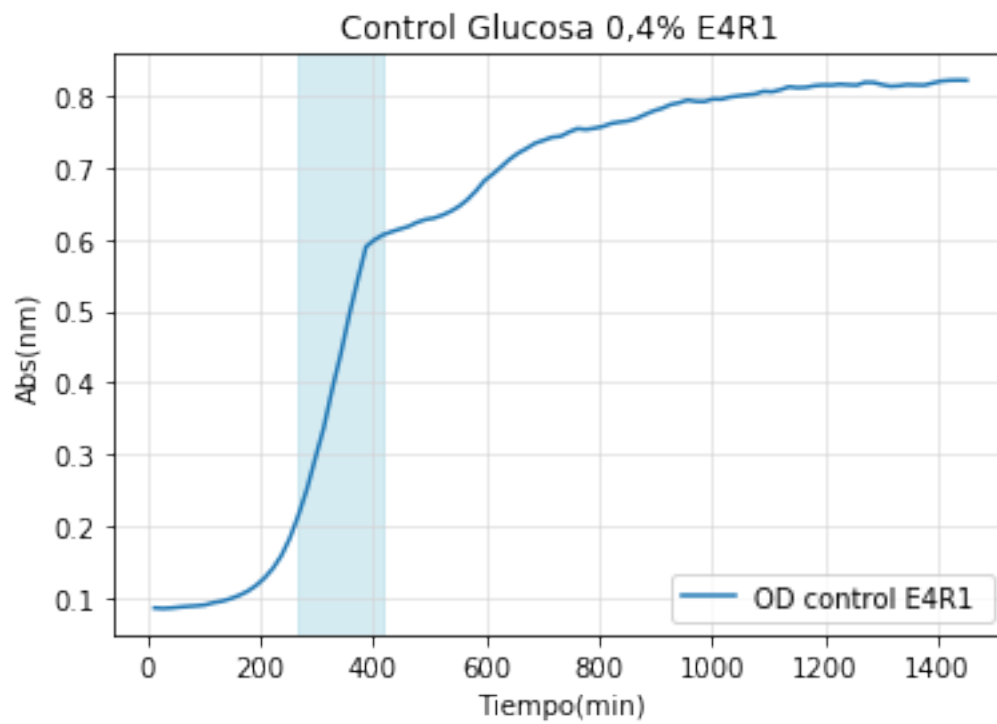
[ 2.21400213e+00 9.27035018e-03 1.61117814e+02]

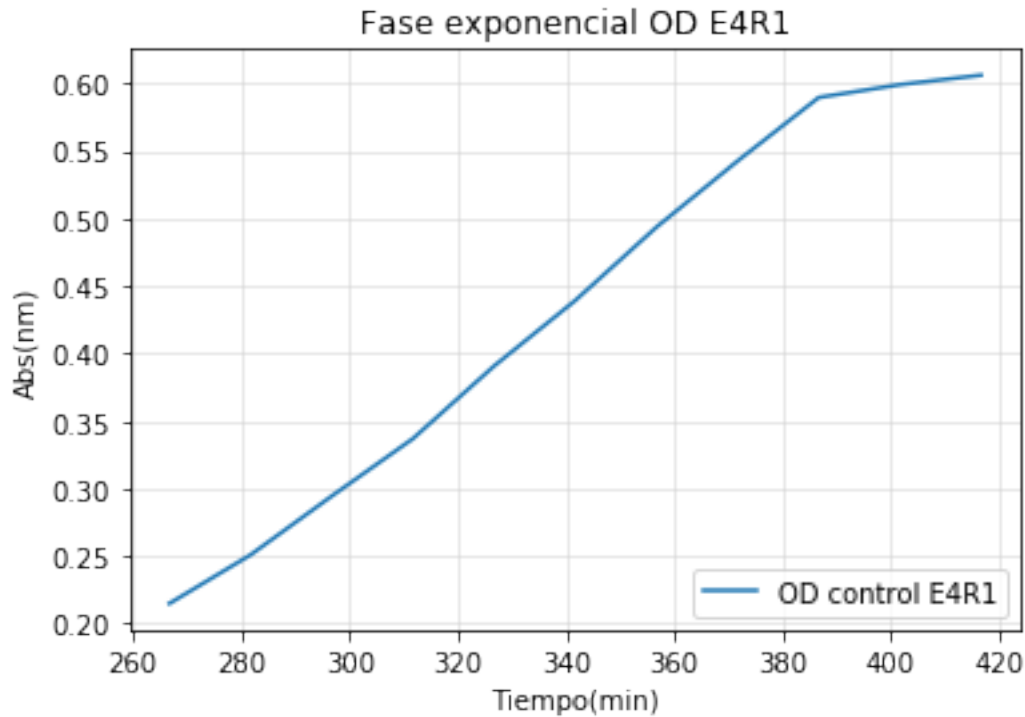


A=2.214002e+00  
um=9.270350e-03

$l=1.611178e+02$   
 $T_m=2.489770e+02$   
 $doubpe=7.477033e+01$   
 $ext=1.495407e+02$   
 $T_{final}=3.985177e+02$

Out[79]: <matplotlib.legend.Legend at 0x178950fc518>





```
In [80]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y14= np.log(odcg122)-np.log(np.min(odcg122))
print('Min OD = %e'%((np.min(odcg122))))
evaly, params=Function_fit(tt,y14,0,-1, title = 'Ajuste control glucosa 0,4% E4R2')
A14= params[0]
um14=params[1]
l14=params[2]
print('A=%e'%(A14))
print('um=%e'%(um14))
print('l=%e'%(l14))

#Cálculo datos para determinar extensión de la fase exponencial
tm14=((A14/(np.exp(1)*um14))+l14)
print('Tm=%e'%(tm14))
t214=((np.log(2))/um14)
print('doubpe=%e'%(t214))
extdp14=2*t214
print('ext=%e'%extdp14)
ttot14=tm14+extdp14
print('Tfinal=%e'%ttot14)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[14]
```

```

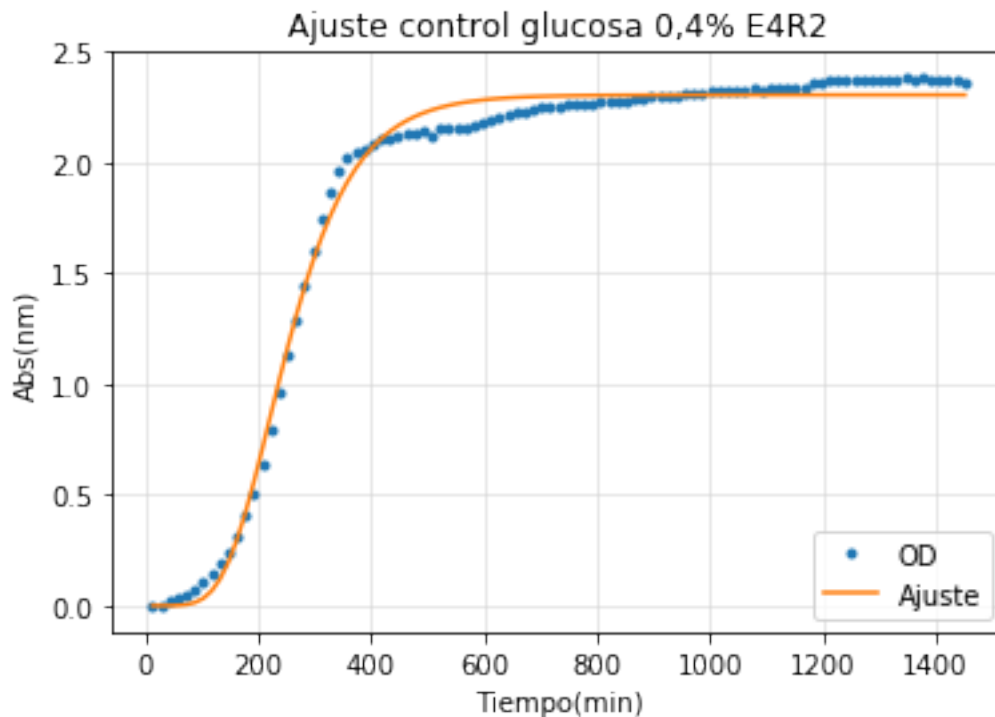
y2=tt[24]
plt.figure()
plt.title('Control Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg122,label='OD control E4R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[14:25],odcg122[14:25],label='OD control E4R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.675000e-02

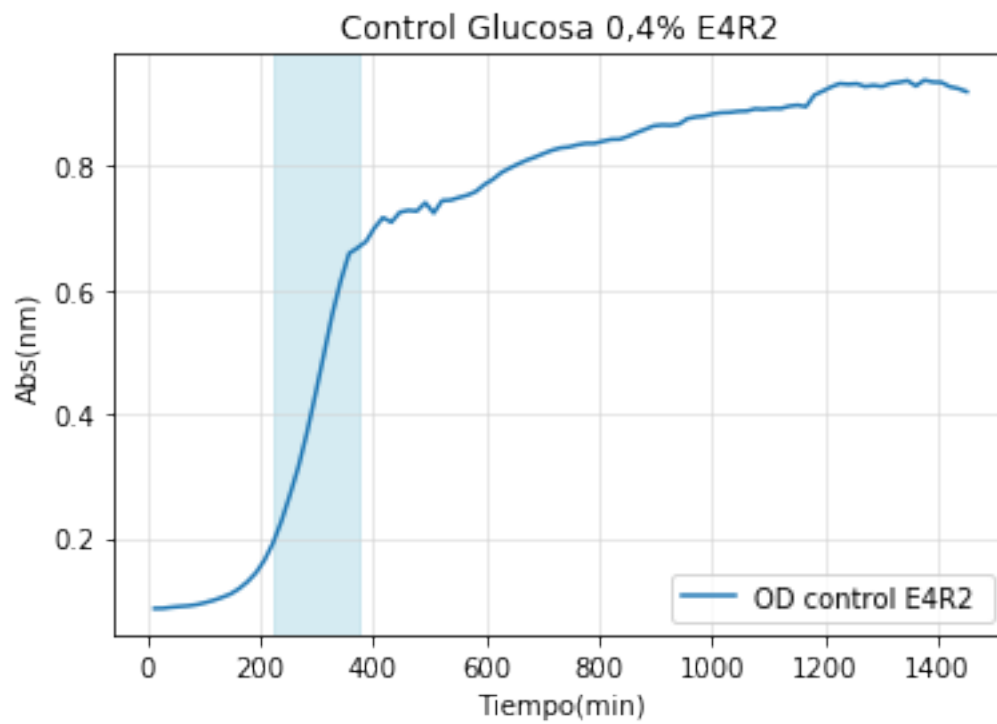
[ 2.30555639e+00 1.03788860e-02 1.38136011e+02]

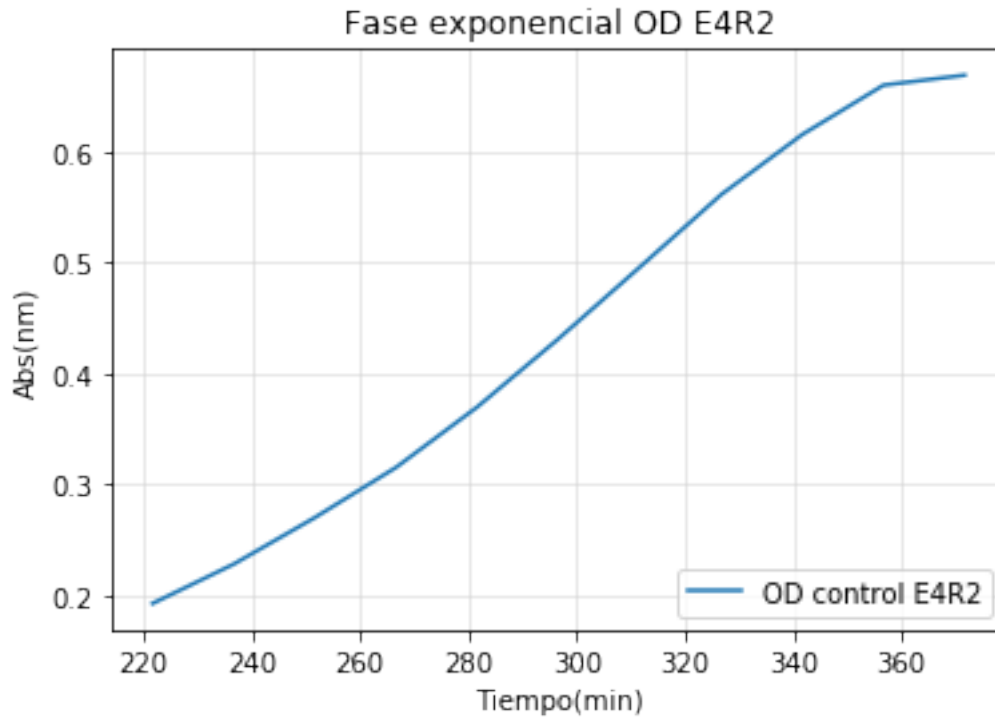


A=2.305556e+00  
um=1.037889e-02

$l=1.381360e+02$   
 $T_m=2.198564e+02$   
 $doubpe=6.678435e+01$   
 $ext=1.335687e+02$   
 $T_{final}=3.534251e+02$

Out[80]: <matplotlib.legend.Legend at 0x17894ce0438>





```
In [81]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y15= np.log(odcg123)-np.log(np.min(odcg123))
print('Min OD = %e'%((np.min(odcg123))))
evaly, params=Function_fit(tt,y15,0,-1, title = 'Ajuste control glucosa 0,4% E4R3')
A15= params[0]
um15=params[1]
l15=params[2]
print('A=%e'%(A15))
print('um=%e'%(um15))
print('l=%e'%(l15))

#Cálculo datos para determinar extensión de la fase exponencial
tm15=((A15/(np.exp(1)*um15))+l15)
print('Tm=%e'%(tm15))
t215=((np.log(2))/um15)
print('doubpe=%e'%(t215))
extdp15=2*t215
print('ext=%e'%extdp15)
ttot15=tm15+extdp15
print('Tfinal=%e'%ttot15)

#Delimitación fase exponencial en grafico con OD/tiempo
```



```

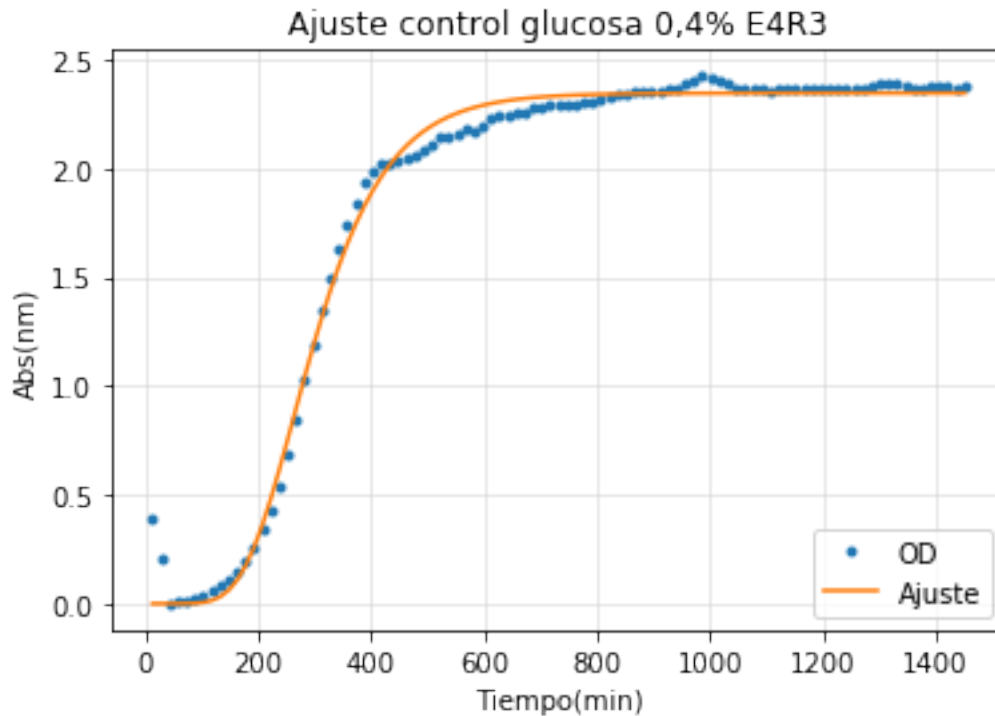
y1=tt[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg123,label='OD control E4R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],odcg123[17:29],label='OD control E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

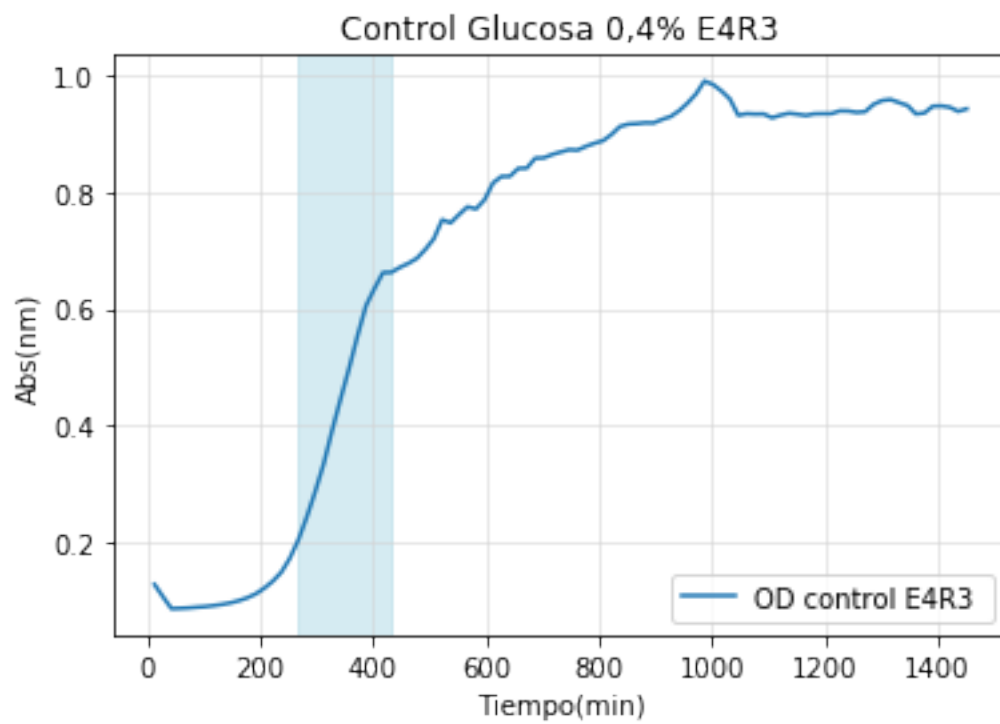
Min OD = 8.775000e-02

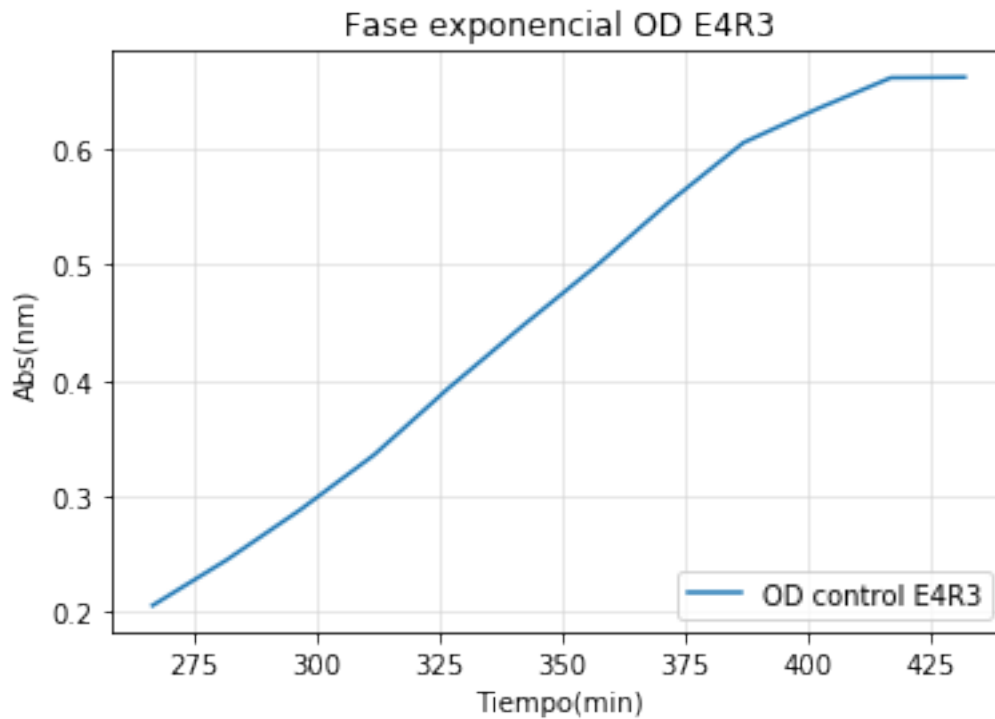
[ 2.34400664e+00 9.59456675e-03 1.72702880e+02]



A=2.344007e+00  
um=9.594567e-03  
l=1.727029e+02  
Tm=2.625779e+02  
doubpe=7.224372e+01  
ext=1.444874e+02  
Tfinal=4.070653e+02

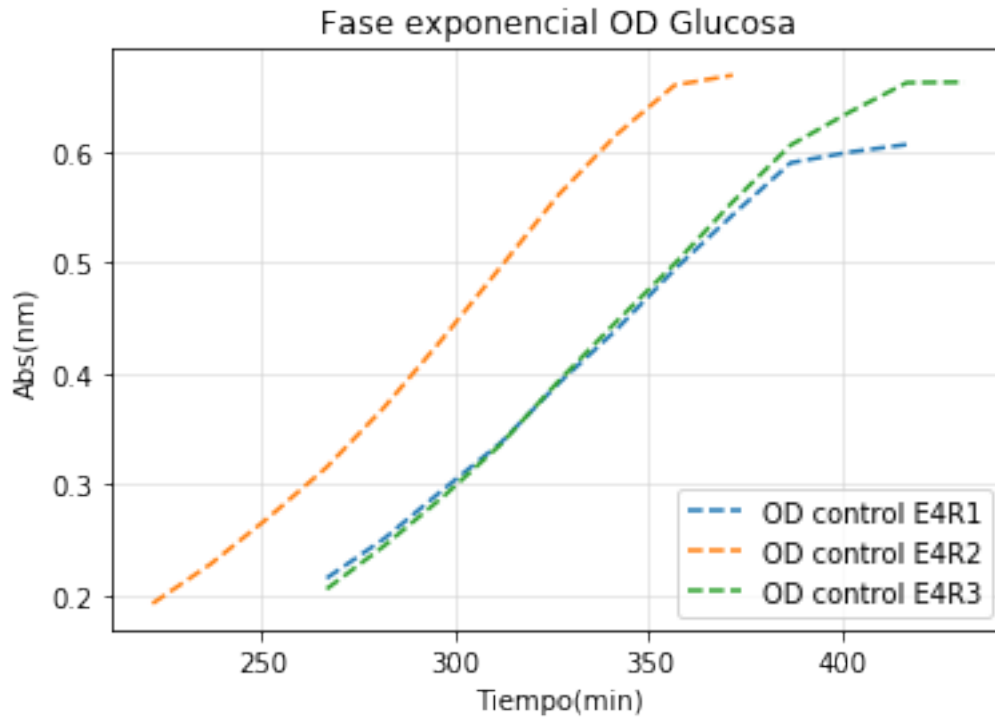
Out[81]: <matplotlib.legend.Legend at 0x1789196a1d0>





```
In [82]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glucosa')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],odcg121[17:28], '--',label='OD control E4R1')
plt.plot(tt[14:25],odcg122[14:25], '--',label='OD control E4R2')
plt.plot(tt[17:29],odcg123[17:29], '--',label='OD control E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[82]: <matplotlib.legend.Legend at 0x17890e9a6a0>
```



```
In [83]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 1
y16= np.log(odcgl121)-np.log(np.min(odcgl121))
print('Min OD = %e'%((np.min(odcgl121))))
evaly, params=Function_fit(tt,y16,0,-1, title = 'Ajuste control glicerol 0,2% E4R1')
A16= params[0]
um16=params[1]
l16=params[2]
print('A=%e'%(A16))
print('um=%e'%(um16))
print('l=%e'%(l16))

#Cálculo datos para determinar extensión de la fase exponencial
tm16=((A16/(np.exp(1)*um16))+l16)
print('Tm=%e'%(tm16))
t216=((np.log(2))/um16)
print('doubpe=%e'%(t216))
extdp16=2.5*t216
print('ext=%e'%extdp16)
ttot16=tm16+extdp16
print('Tfinal=%e'%ttot16)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[24]
```

```

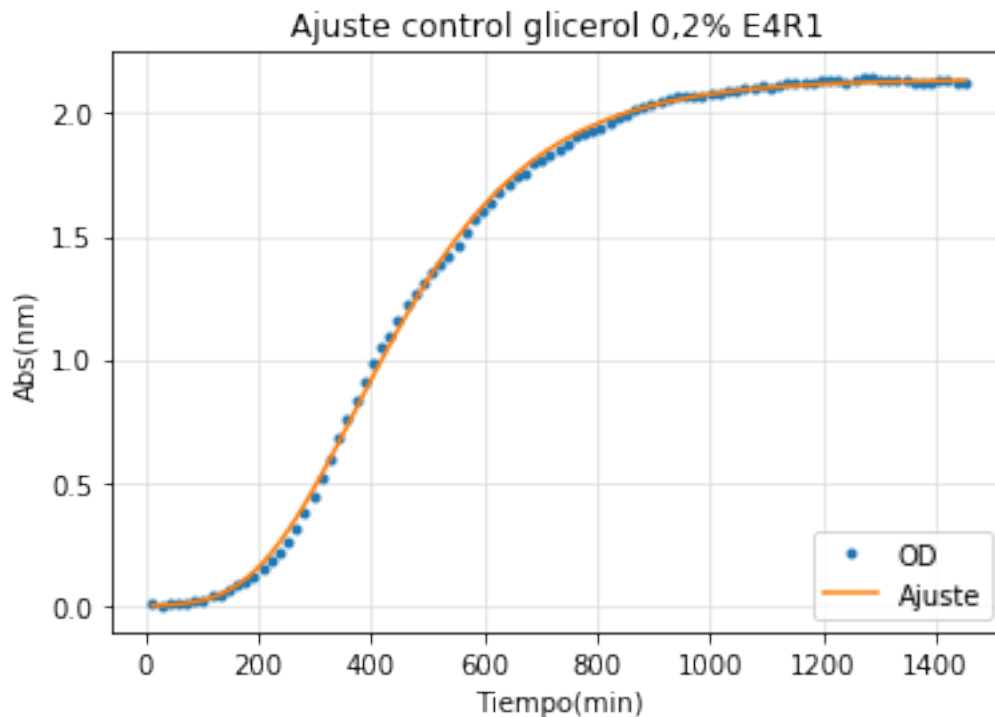
y2=tt[51]
plt.figure()
plt.title('Control Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1121,label='OD control E4R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:52],odcg1121[24:52],label='OD control E4R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.500000e-02

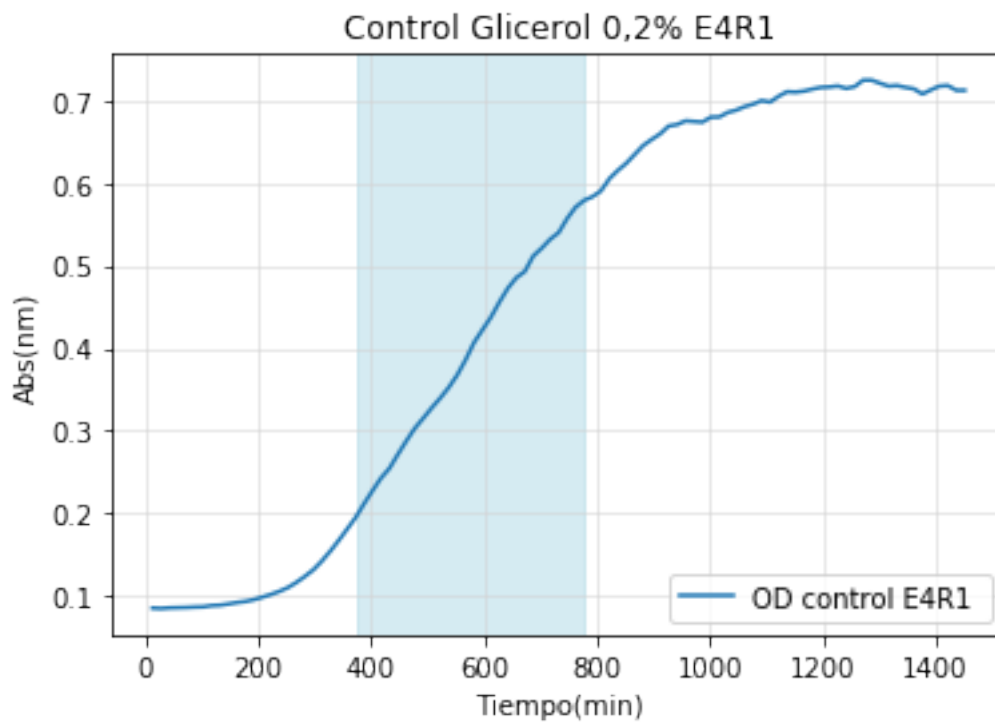
[ 2.13735411e+00 4.44419084e-03 1.92719677e+02]

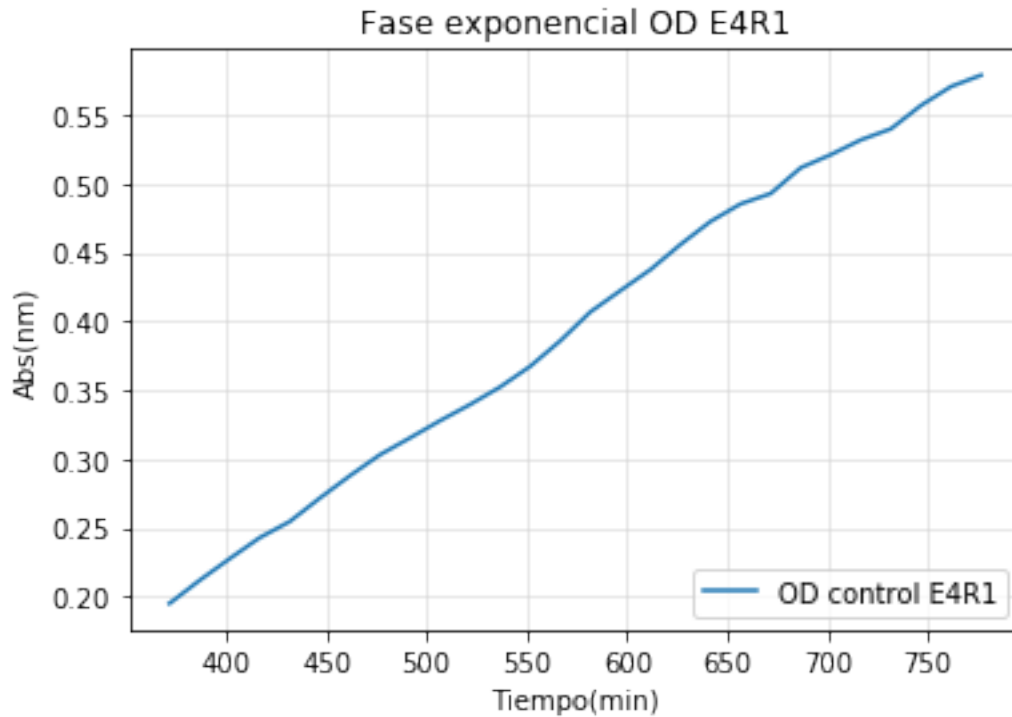


A=2.137354e+00  
um=4.444191e-03

l=1.927197e+02  
Tm=3.696447e+02  
doubpe=1.559670e+02  
ext=3.899175e+02  
Tfinal=7.595623e+02

Out [83]: <matplotlib.legend.Legend at 0x1789517dc18>





```
In [84]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 2
y17= np.log(odcgl122)-np.log(np.min(odcgl122))
print('Min OD = %e'%((np.min(odcgl122))))
evaly, params=Function_fit(tt,y17,0,-1, title = 'Ajuste control glicerol 0,2% E4R2')
A17= params[0]
um17=params[1]
l17=params[2]
print('A=%e'%(A17))
print('um=%e'%(um17))
print('l=%e'%(l17))

#Cálculo datos para determinar extensión de la fase exponencial
tm17=((A17/(np.exp(1)*um17))+l17)
print('Tm=%e'%(tm17))
t217=((np.log(2))/um17)
print('doubpe=%e'%(t217))
extdp17=2.5*t217
print('ext=%e'%extdp17)
ttot17=tm17+extdp17
print('Tfinal=%e'%ttot17)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

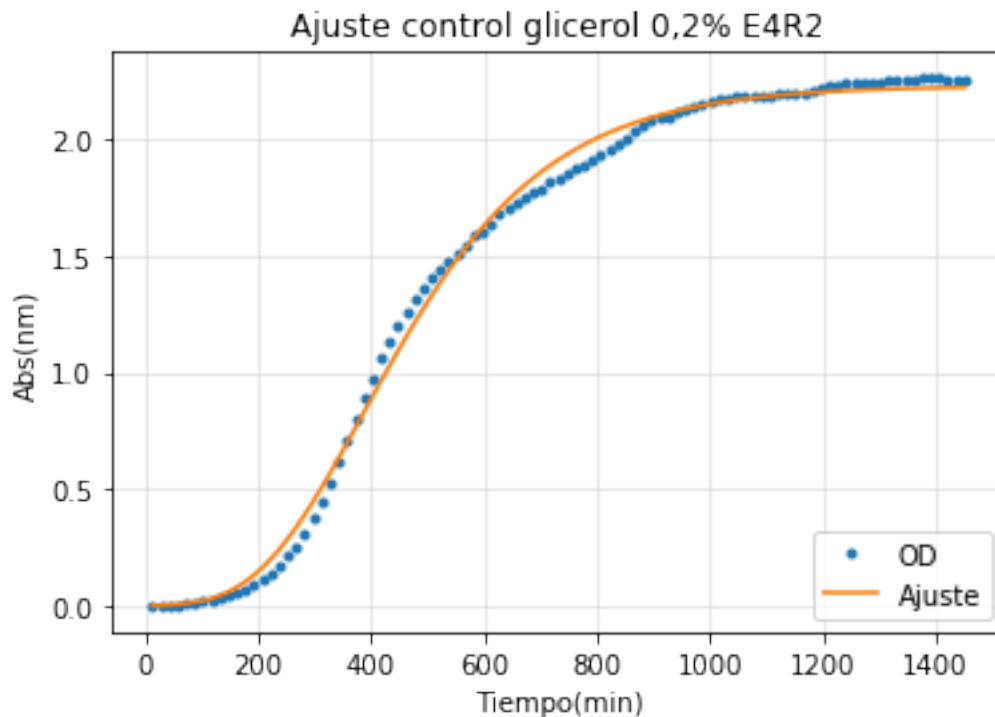
y2=tt[52]
plt.figure()
plt.title('Control Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1122,label='OD control E4R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],odcg1122[25:53],label='OD control E4R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.500000e-02

[ 2.22944589e+00 4.42704053e-03 1.98978142e+02]

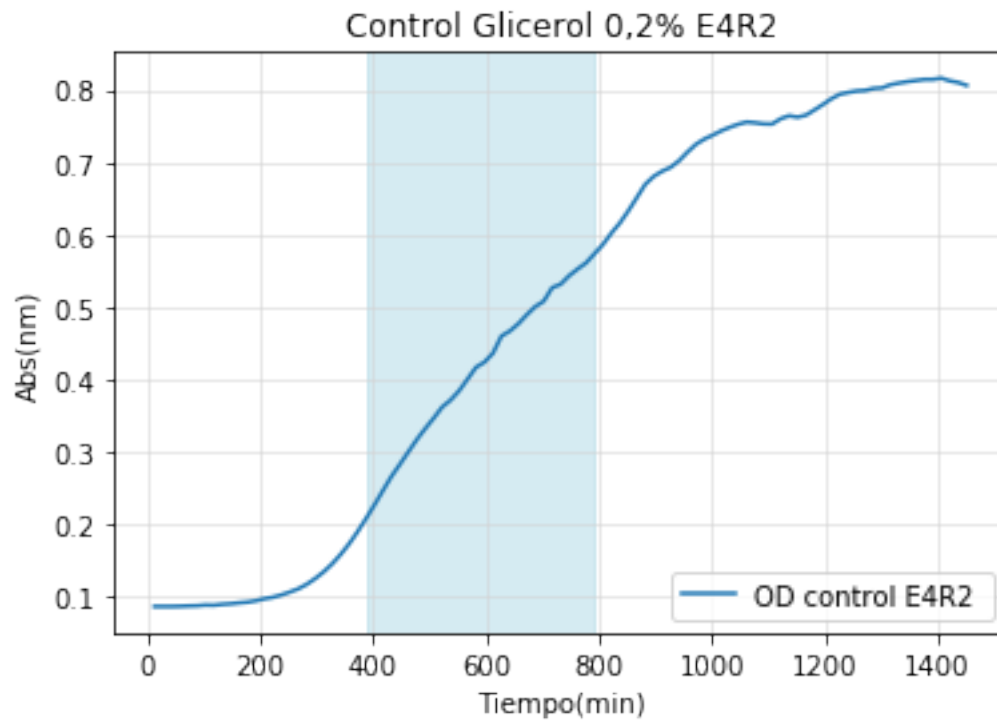


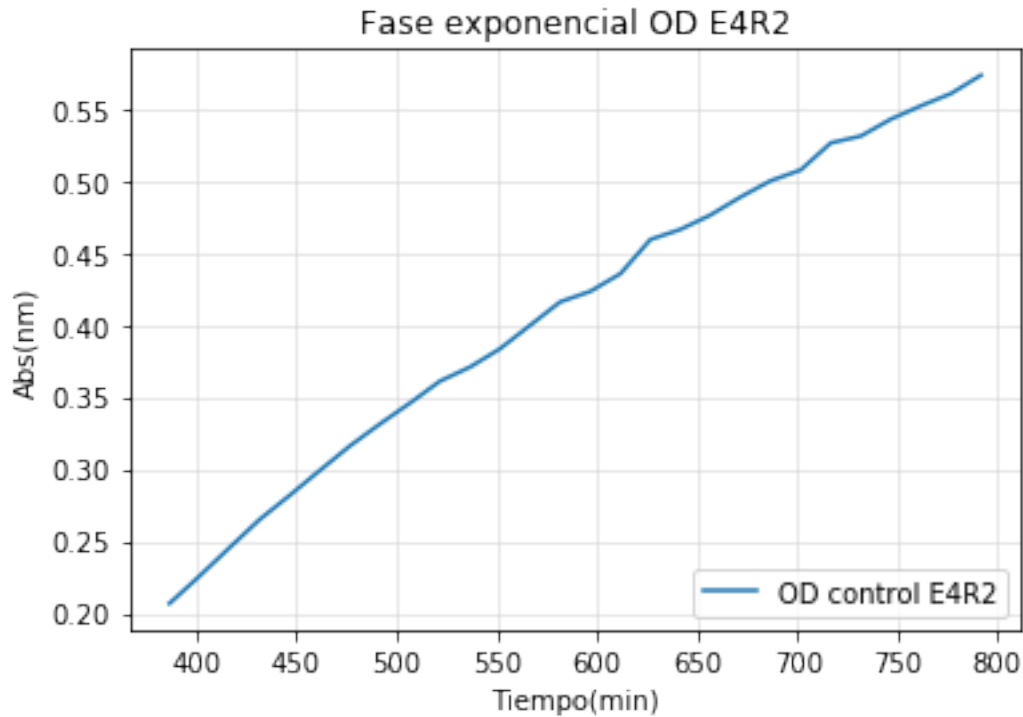
A=2.229446e+00  
um=4.427041e-03



```
l=1.989781e+02  
Tm=3.842413e+02  
doubpe=1.565712e+02  
ext=3.914281e+02  
Tfinal=7.756693e+02
```

Out[84]: <matplotlib.legend.Legend at 0x17895395978>





```
In [85]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 3
y18= np.log(odcgl123)-np.log(np.min(odcgl123))
print('Min OD = %e'%((np.min(odcgl123))))
evaly, params=Function_fit(tt,y18,0,-1, title = 'Ajuste control glicerol 0,2% E4R3')
A18= params[0]
um18=params[1]
l18=params[2]
print('A=%e'%(A18))
print('um=%e'%(um18))
print('l=%e'%(l18))

#Cálculo datos para determinar extensión de la fase exponencial
tm18=((A18/(np.exp(1)*um18))+l18)
print('Tm=%e'%(tm18))
t218=((np.log(2))/um18)
print('doubpe=%e'%(t218))
extdp18=2.5*t218
print('ext=%e'%extdp18)
ttot18=tm18+extdp18
print('Tfinal=%e'%ttot18)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

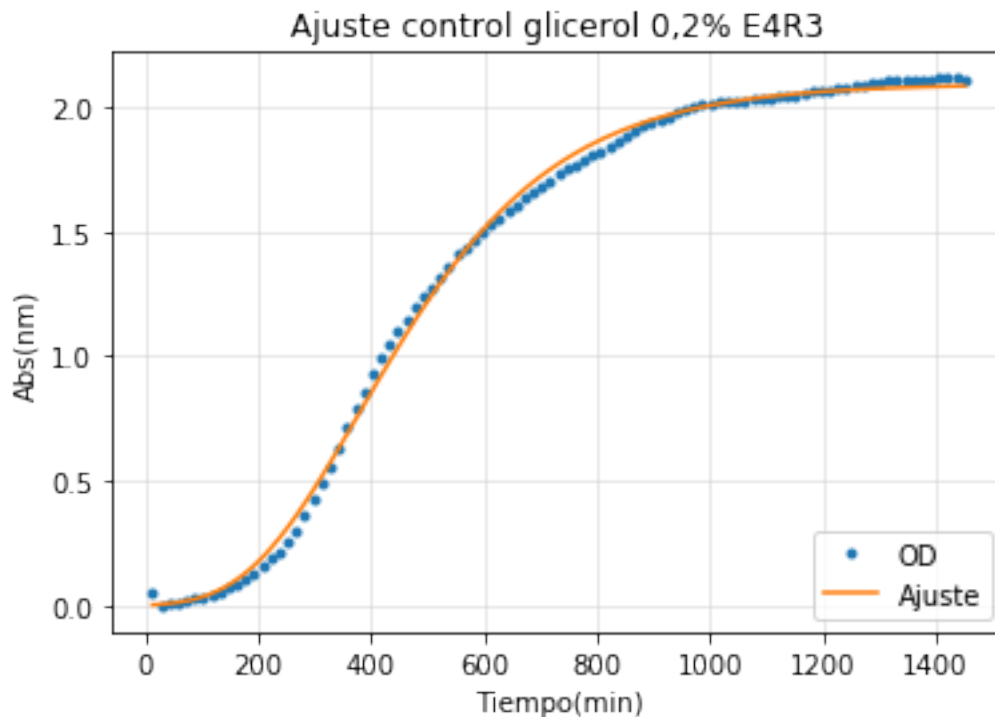
y2=tt[56]
plt.figure()
plt.title('Control Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1123,label='OD control E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:57],odcg1123[25:57],label='OD control E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.525000e-02

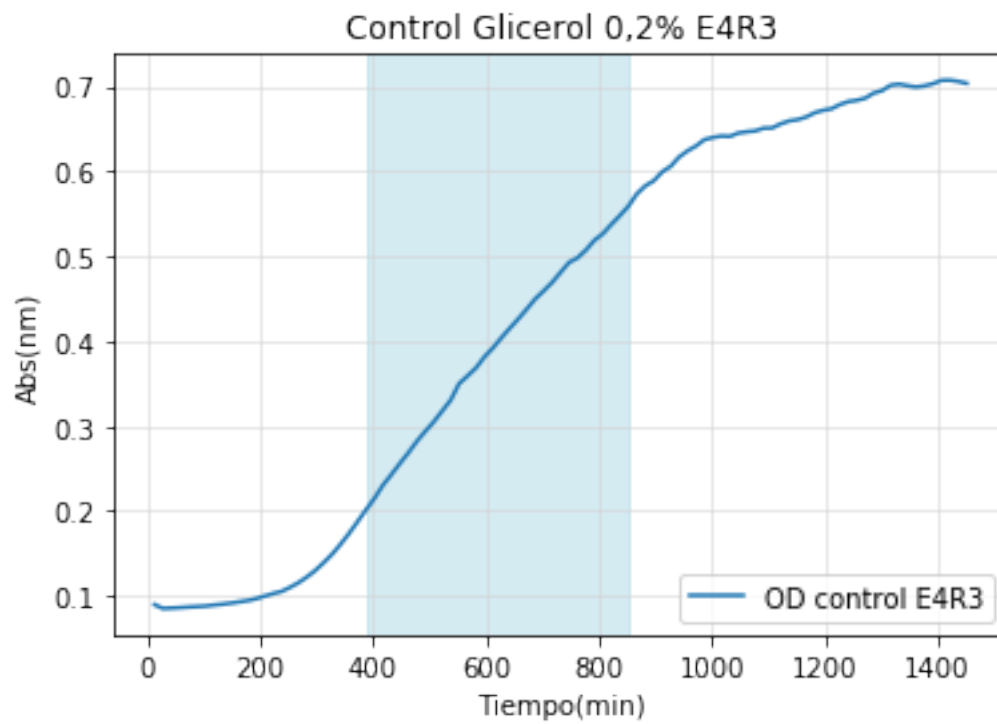
[ 2.09342076e+00 3.90264068e-03 1.80268803e+02]

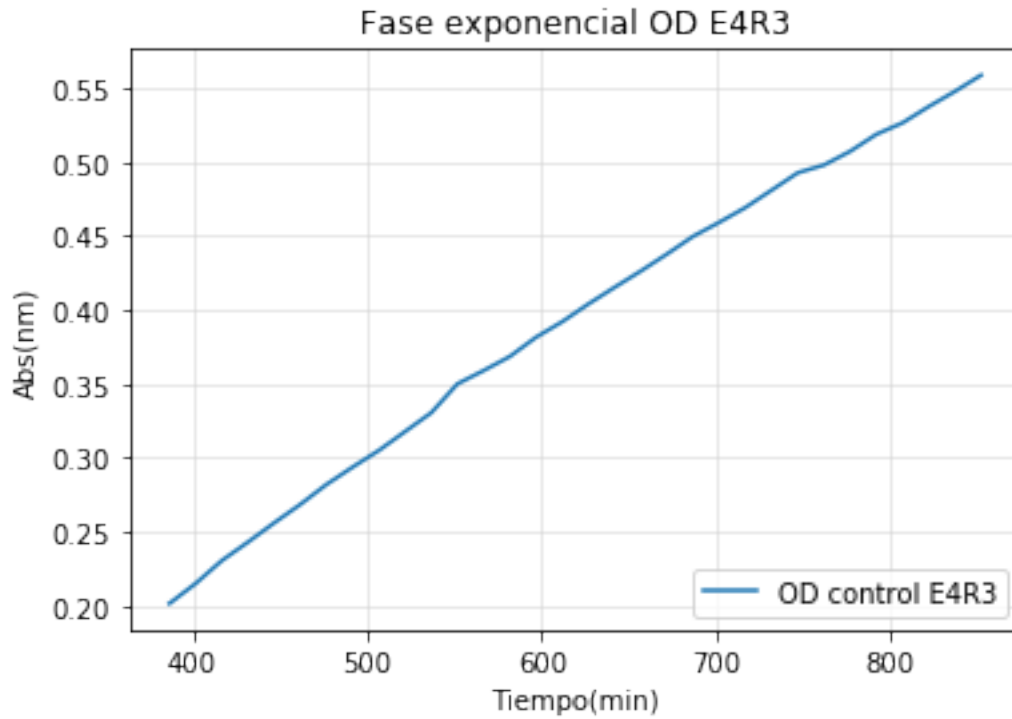


A=2.093421e+00  
um=3.902641e-03

```
l=1.802688e+02  
Tm=3.776035e+02  
doubpe=1.776098e+02  
ext=4.440245e+02  
Tfinal=8.216280e+02
```

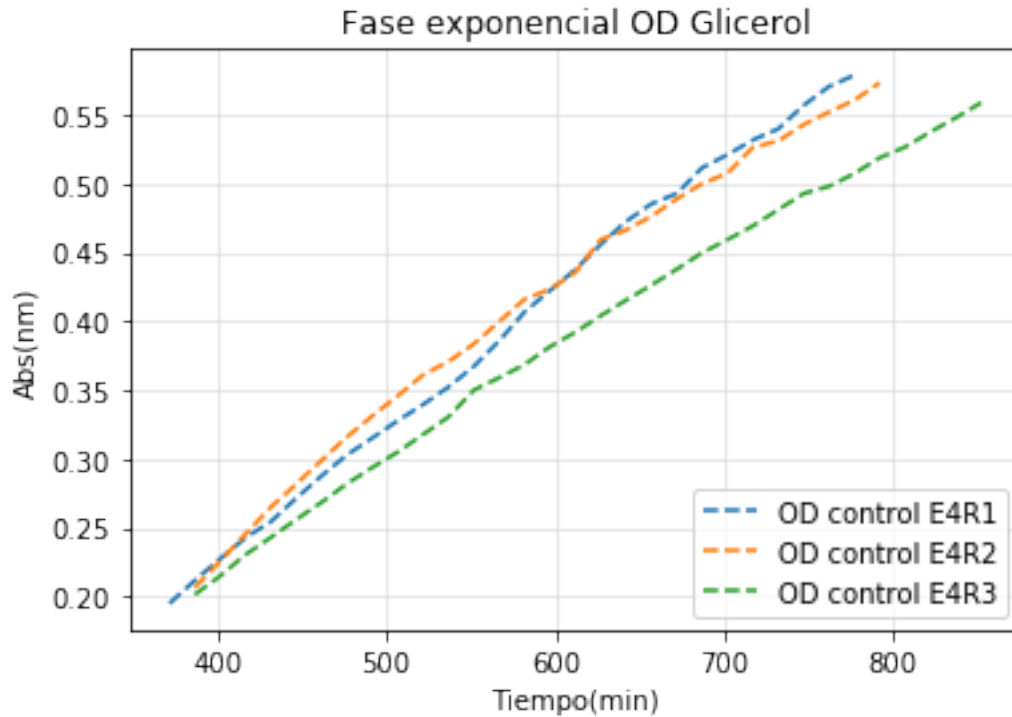
Out[85]: <matplotlib.legend.Legend at 0x178955a7748>





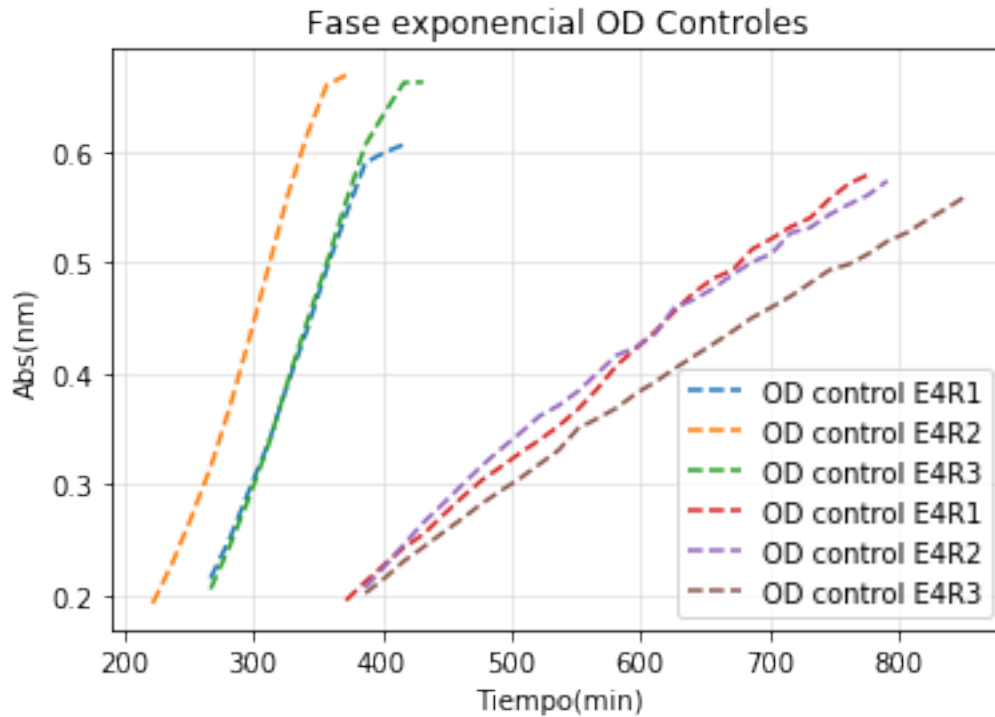
```
In [86]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glicerol')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:52],odcg1121[24:52], '--',label='OD control E4R1')
plt.plot(tt[25:53],odcg1122[25:53], '--',label='OD control E4R2')
plt.plot(tt[25:57],odcg1123[25:57], '--',label='OD control E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[86]: <matplotlib.legend.Legend at 0x178956d0748>
```



```
In [87]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Controles')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],odcg121[17:28], '--',label='OD control E4R1')
plt.plot(tt[14:25],odcg122[14:25], '--',label='OD control E4R2')
plt.plot(tt[17:29],odcg123[17:29], '--',label='OD control E4R3')
plt.plot(tt[24:52],odcg1121[24:52], '--',label='OD control E4R1')
plt.plot(tt[25:53],odcg1122[25:53], '--',label='OD control E4R2')
plt.plot(tt[25:57],odcg1123[25:57], '--',label='OD control E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[87]: <matplotlib.legend.Legend at 0x17896760f28>
```



```
In [88]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-J23101-std glucosa rep 1
y19= np.log(od15271)-np.log(np.min(od15271))
print('Min OD = %e'%((np.min(od15271))))
evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste pTet-pLas81-J23101 glucosa 0,4%')
A19 = params[0]
um19=params[1]
l19=params[2]
print('A=%e'%(A19))
print('um=%e'%(um19))
print('l=%e'%(l19))

#Cálculo datos para determinar extensión de la fase exponencial
tm19=((A19/(np.exp(1)*um19))+l19)
print('Tm=%e'%(tm19))
t219=((np.log(2))/um19)
print('doubpe=%e'%(t219))
extdp19=2*t219
print('ext=%e'%extdp19)
ttot19=tm19+extdp19
print('Tfinal=%e'%ttot19)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[19]
```

```

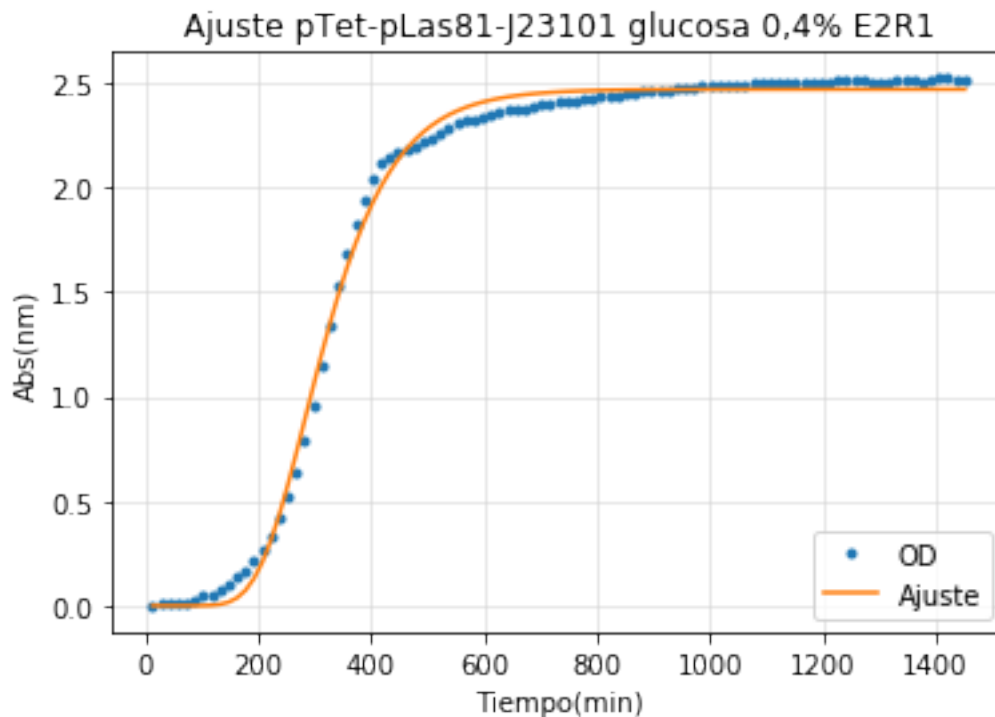
y2=tt[28]
plt.figure()
plt.title('pTet-pLas81-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15271,label='OD pTet-pLas81-J23101 E2R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:29],od15271[19:29],label='OD pTet-pLas81-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.510000e-02

[ 2.46838538e+00 1.06746902e-02 1.98321599e+02]

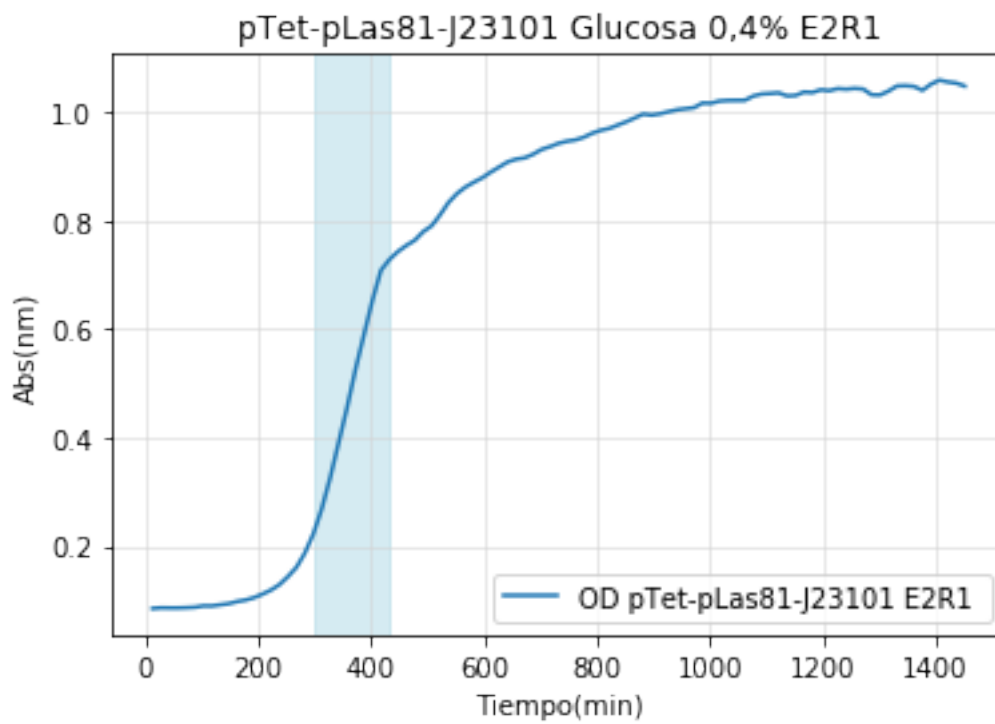


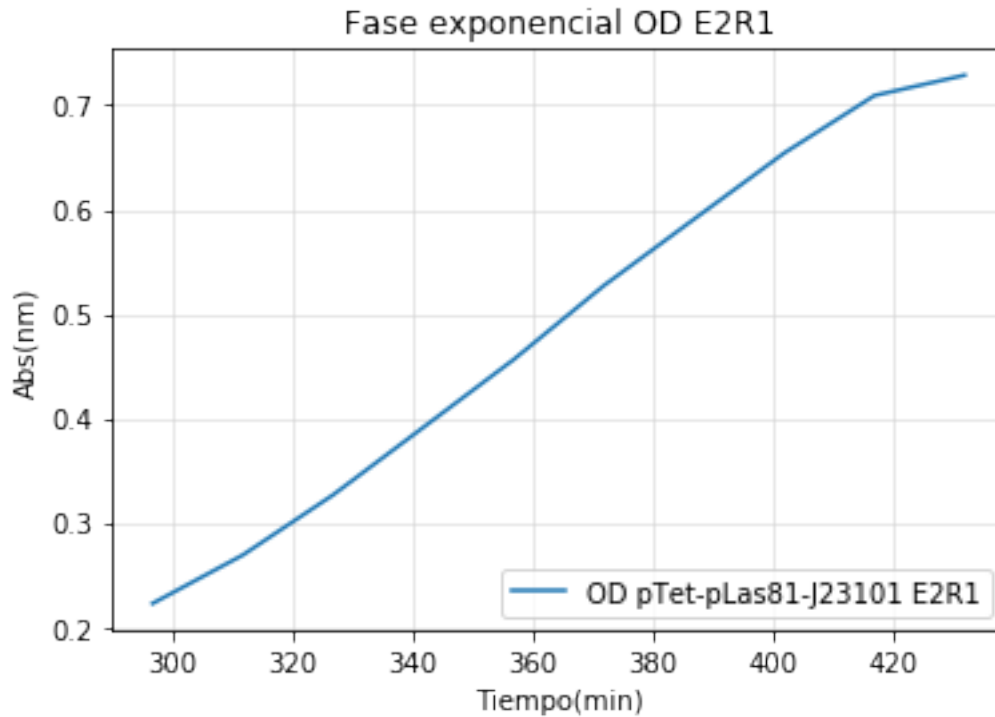
A=2.468385e+00  
um=1.067469e-02



l=1.983216e+02  
Tm=2.833890e+02  
doubpe=6.493370e+01  
ext=1.298674e+02  
Tfinal=4.132564e+02

Out[88]: <matplotlib.legend.Legend at 0x178955a0d68>





```
In [89]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 2
y20= np.log(od15272)-np.log(np.min(od15272))
print('Min OD = %e'%((np.min(od15272))))
evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste pTet-pLas81-J23101 glucosa 0,4%')
A20= params[0]
um20=params[1]
l20=params[2]
print('A=%e'%(A20))
print('um=%e'%(um20))
print('l=%e'%(l20))

#Cálculo datos para determinar extensión de la fase exponencial
tm20=((A20/(np.exp(1)*um20))+l20)
print('Tm=%e'%(tm20))
t220=((np.log(2))/um20)
print('doubpe=%e'%(t220))
extdp20=2*t220
print('ext=%e'%extdp20)
ttot20=tm20+extdp20
print('Tfinal=%e'%ttot20)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
```

```

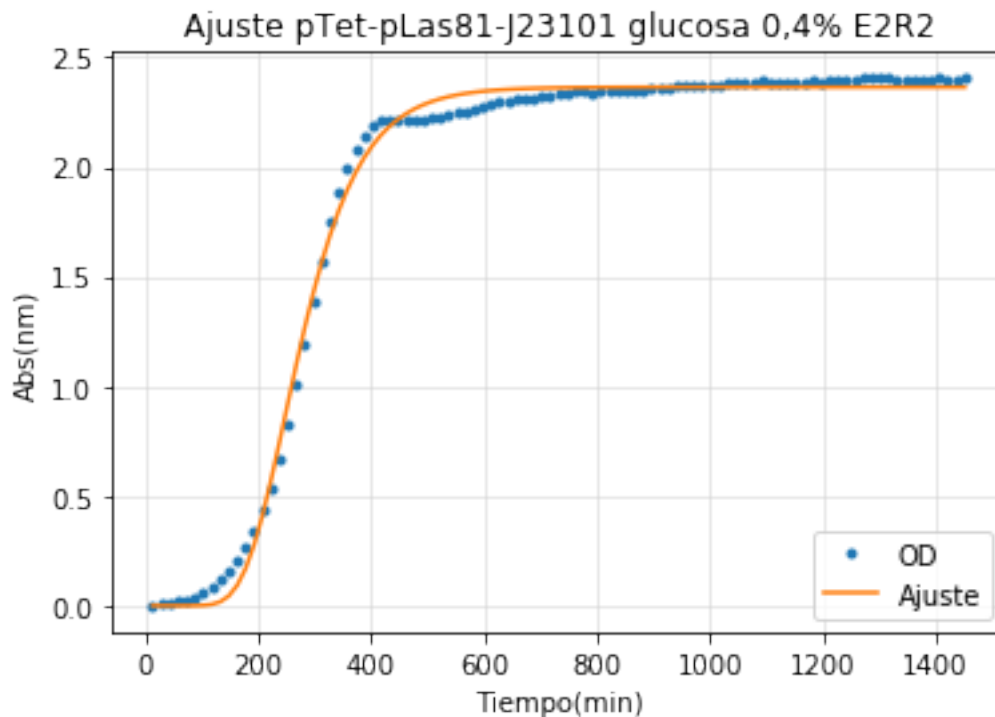
y2=tt[25]
plt.figure()
plt.title('pTet-pLas81-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15272,label='OD pTet-pLas81-J23101 E2R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od15272[16:26],label='OD pTet-pLas81-J23101 E2R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.580000e-02

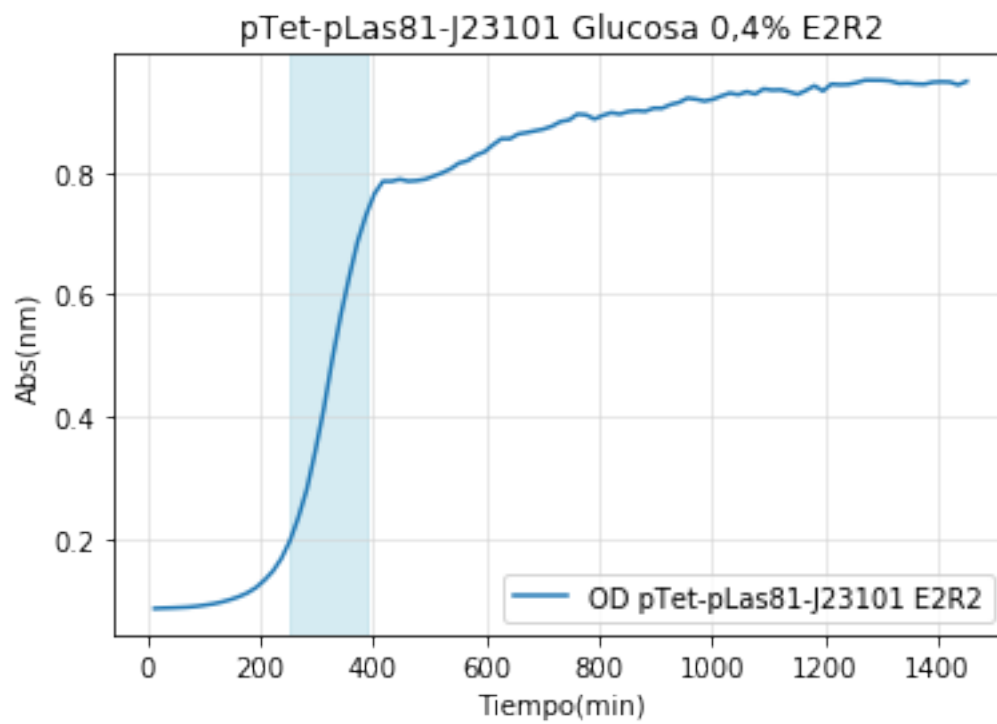
[ 2.36424970e+00 1.19691673e-02 1.74474154e+02]

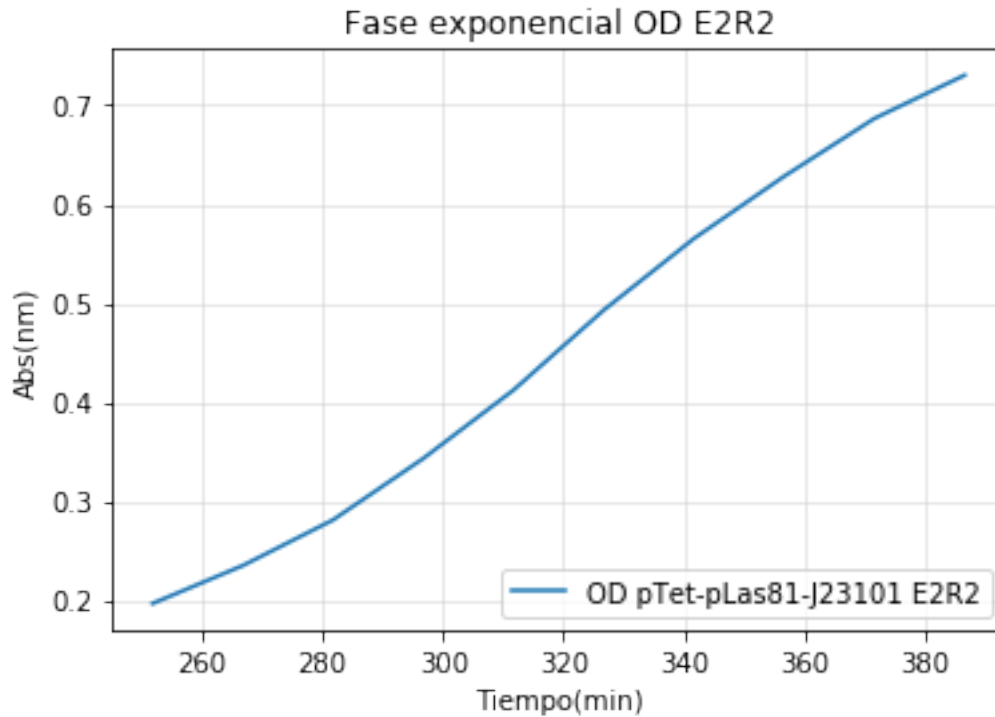


A=2.364250e+00  
um=1.196917e-02

```
l=1.744742e+02  
Tm=2.471408e+02  
doubpe=5.791106e+01  
ext=1.158221e+02  
Tfinal=3.629629e+02
```

Out[89]: <matplotlib.legend.Legend at 0x1789507b160>





```
In [90]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 3
y21= np.log(od15273)-np.log(np.min(od15273))
print('Min OD = %e'%((np.min(od15273))))
evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste pTet-pLas81-J23101 glucosa 0,4%')
A21= params[0]
um21=params[1]
l21=params[2]
print('A=%e'%(A21))
print('um=%e'%(um21))
print('l=%e'%(l21))

#Cálculo datos para determinar extensión de la fase exponencial
tm21=((A21/(np.exp(1)*um21))+l21)
print('Tm=%e'%(tm21))
t221=((np.log(2))/um21)
print('doubpe=%e'%(t221))
extdp21=2*t221
print('ext=%e'%extdp21)
ttot21=tm21+extdp21
print('Tfinal=%e'%ttot21)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
```

```

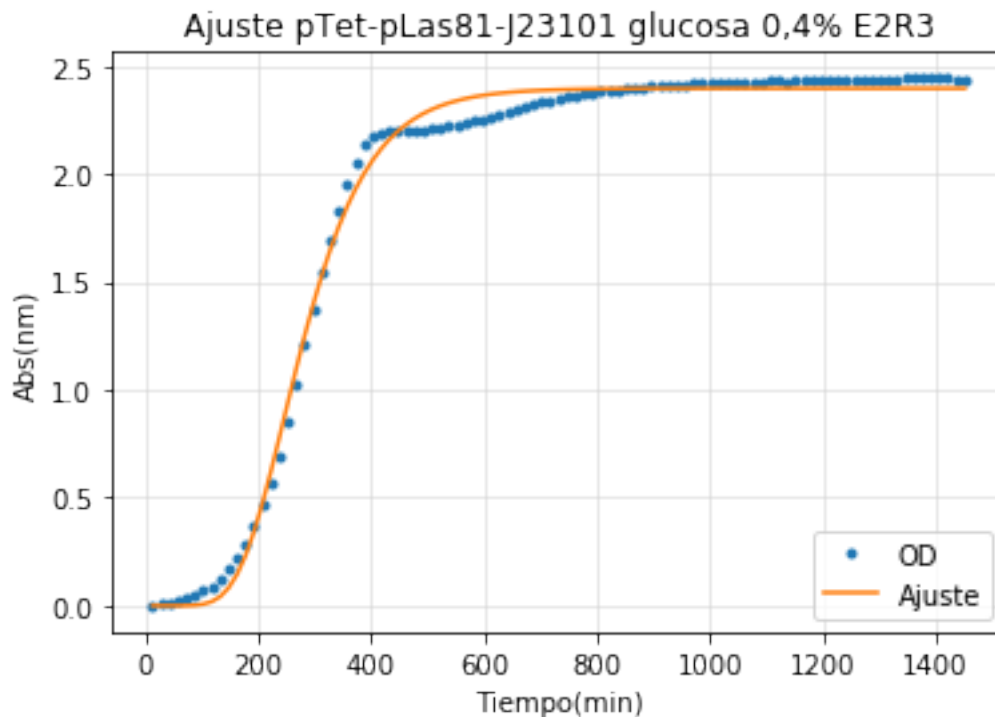
y2=tt[26]
plt.figure()
plt.title('pTet-pLas81-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15273,label='OD pTet-pLas81-J23101 E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:27],od15273[16:27],label='OD pTet-pLas81-J23101 E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.390000e-02

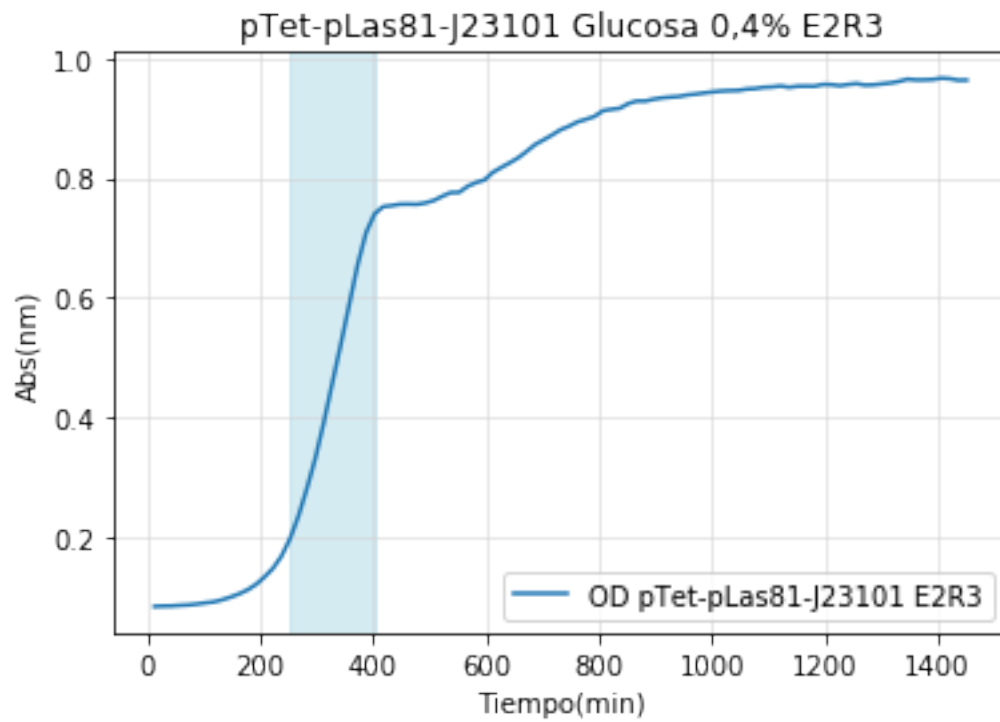
[ 2.39690020e+00 1.07573329e-02 1.64117750e+02]

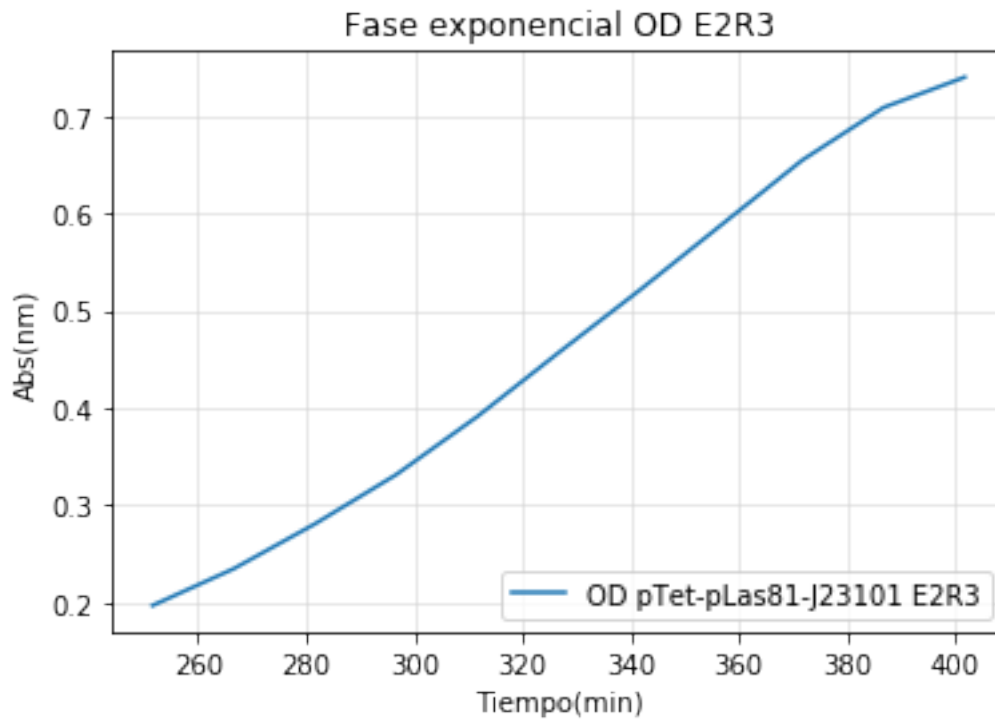


A=2.396900e+00  
um=1.075733e-02

l=1.641177e+02  
Tm=2.460870e+02  
doubpe=6.443485e+01  
ext=1.288697e+02  
Tfinal=3.749567e+02

Out[90]: <matplotlib.legend.Legend at 0x178936840f0>

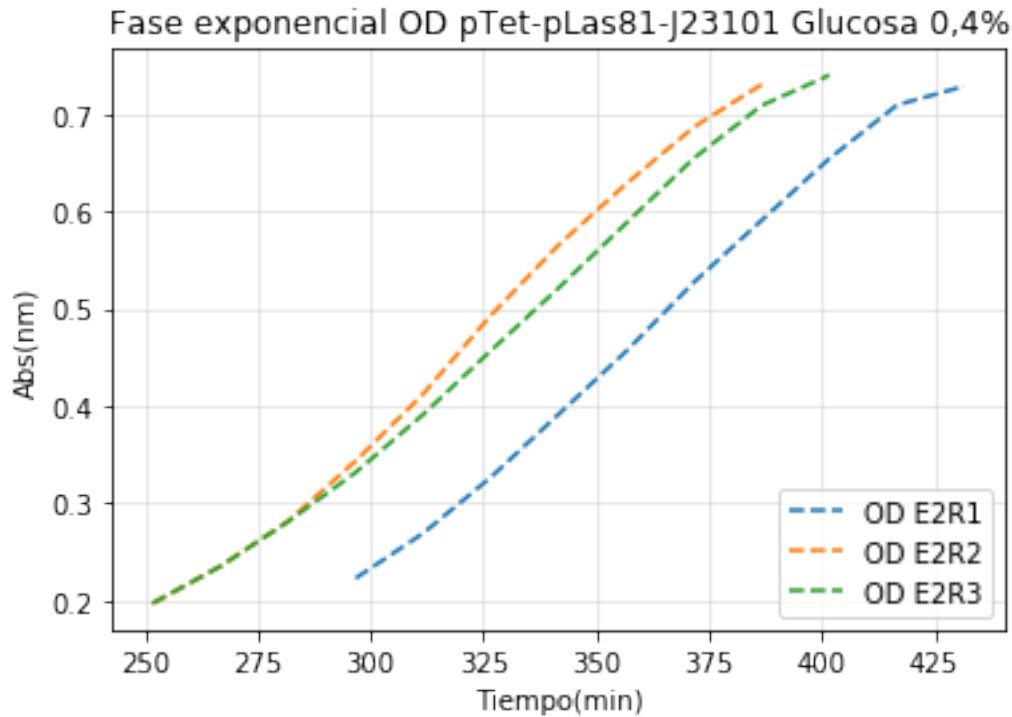




```
In [91]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pTet-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:29],od15271[19:29], '--',label='OD E2R1')
plt.plot(tt[16:26],od15272[16:26], '--',label='OD E2R2')
plt.plot(tt[16:27],od15273[16:27], '--',label='OD E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[91]: <matplotlib.legend.Legend at 0x17895111198>
```





```
In [92]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 1
y22= np.log(od1527g1)-np.log(np.min(od1527g1))
print('Min OD = %e'%((np.min(od1527g1))))
evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%')
A22= params[0]
um22=params[1]
l22=params[2]
print('A=%e'%(A22))
print('um=%e'%(um22))
print('l=%e'%(l22))

#Cálculo datos para determinar extensión de la fase exponencial
tm22=((A22/(np.exp(1)*um22))+l22)
print('Tm=%e'%(tm22))
t222=((np.log(2))/um22)
print('doubpe=%e'%(t222))
extdp22=2.5*t222
print('ext=%e'%extdp22)
ttot22=tm22+extdp22
print('Tfinal=%e'%ttot22)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[29]
```

```

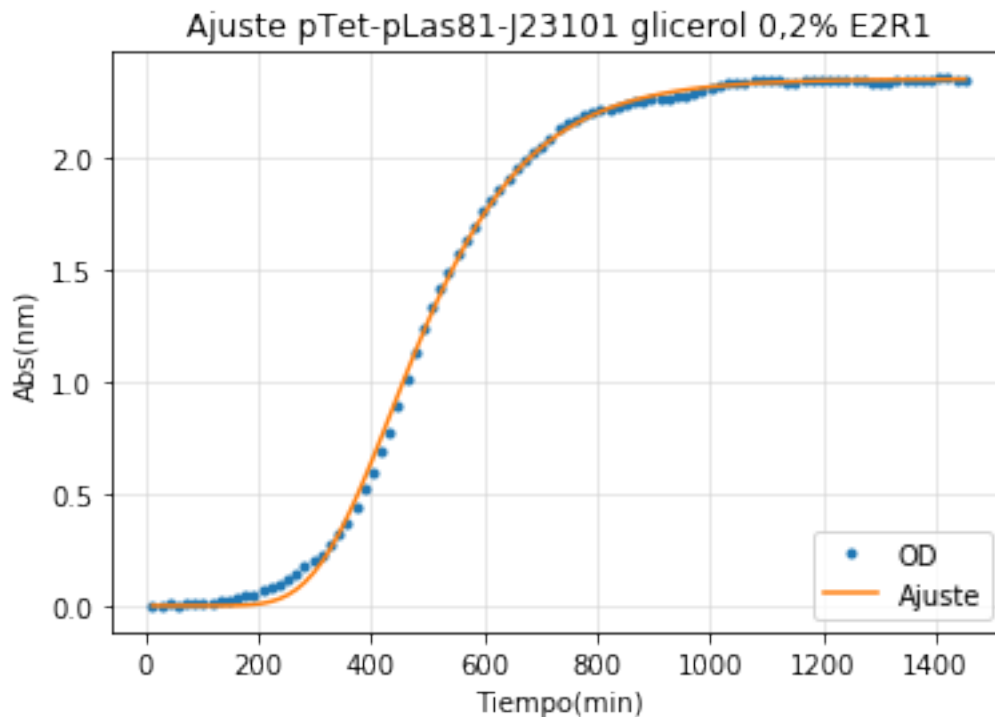
y2=tt[48]
plt.figure()
plt.title('pTet-pLas81-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1527g1,label='OD pTet-pLas81-J23101 E2R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[29:49],od1527g1[29:49],label='OD pTet-pLas81-J23101 E2R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.620000e-02

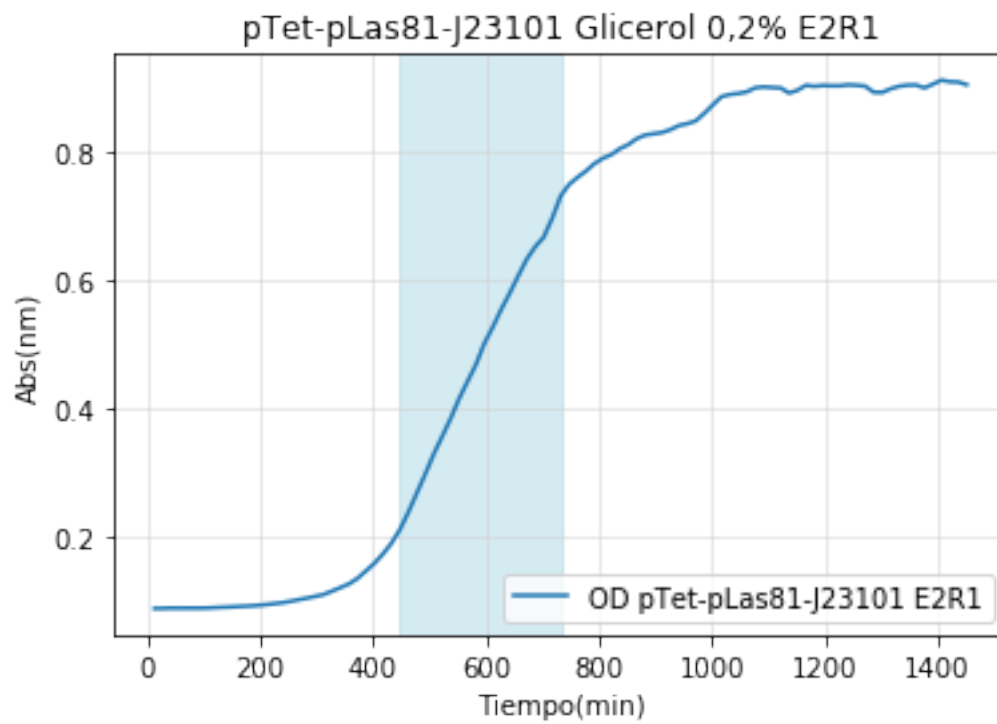
[ 2.35645905e+00 6.42374075e-03 3.00000000e+02]

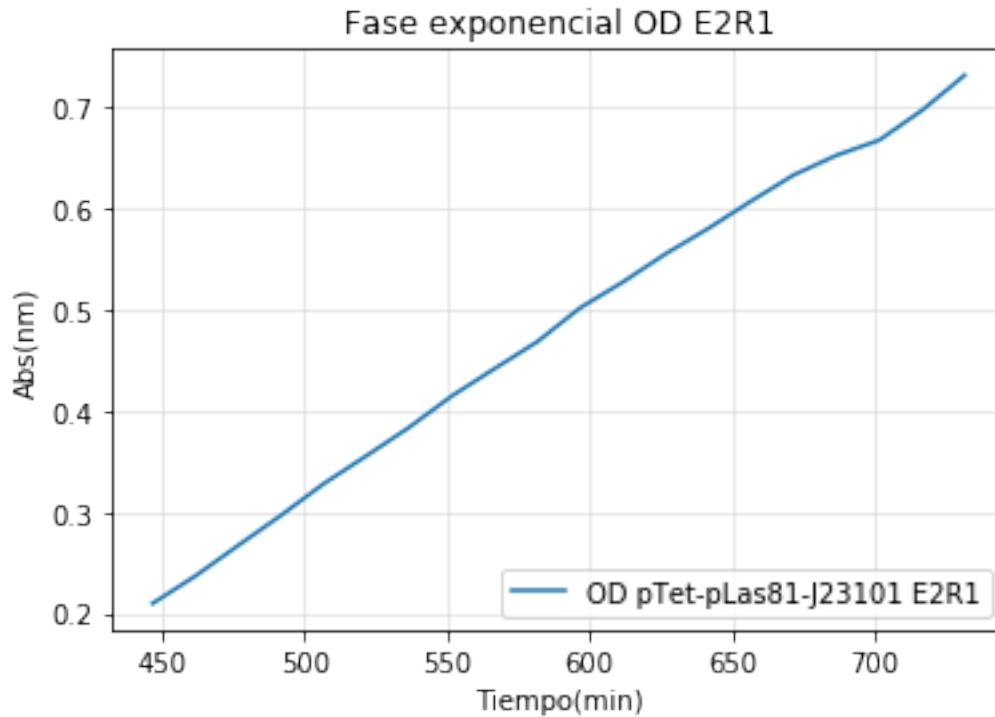


A=2.356459e+00  
um=6.423741e-03

```
l=3.000000e+02  
Tm=4.349514e+02  
doubpe=1.079040e+02  
ext=2.697599e+02  
Tfinal=7.047114e+02
```

Out[92]: <matplotlib.legend.Legend at 0x178954b0a20>





```
In [93]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 2
y23= np.log(od1527g2)-np.log(np.min(od1527g2))
print('Min OD = %e'%((np.min(od1527g2))))
evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%')
A23= params[0]
um23=params[1]
l23=params[2]
print('A=%e'%(A23))
print('um=%e'%(um23))
print('l=%e'%(l23))

#Cálculo datos para determinar extensión de la fase exponencial
tm23=((A23/(np.exp(1)*um23))+l23)
print('Tm=%e'%(tm23))
t223=((np.log(2))/um23)
print('doubpe=%e'%(t223))
extdp23=2.5*t223
print('ext=%e'%extdp23)
ttot23=tm23+extdp23
print('Tfinal=%e'%ttot23)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[26]
```

```

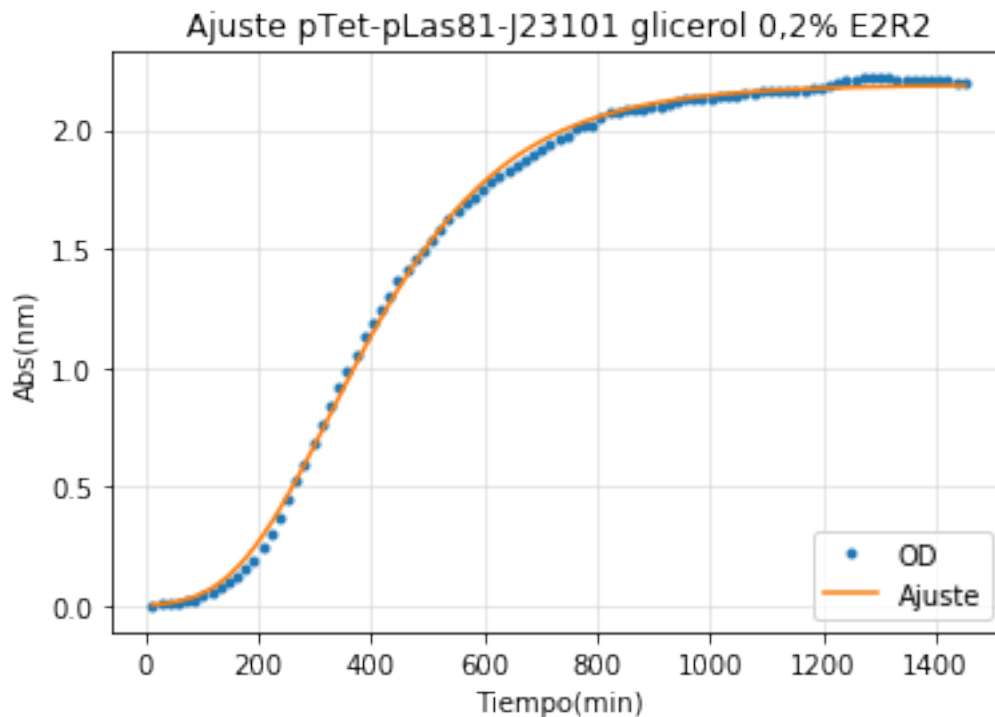
y2=tt[57]
plt.figure()
plt.title('pTet-pLas81-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1527g2,label='OD pTet-pLas81-J23101 E2R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:58],od1527g2[26:58],label='OD pTet-pLas81-J23101 E2R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.590000e-02

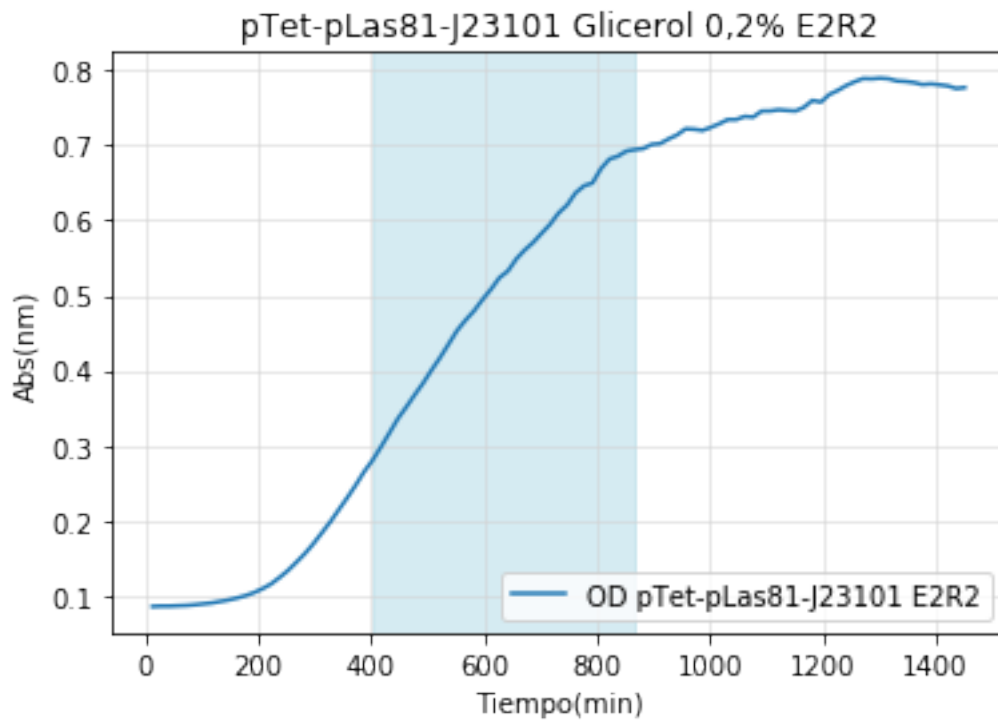
[ 2.18898193e+00 4.67358353e-03 1.54692670e+02]

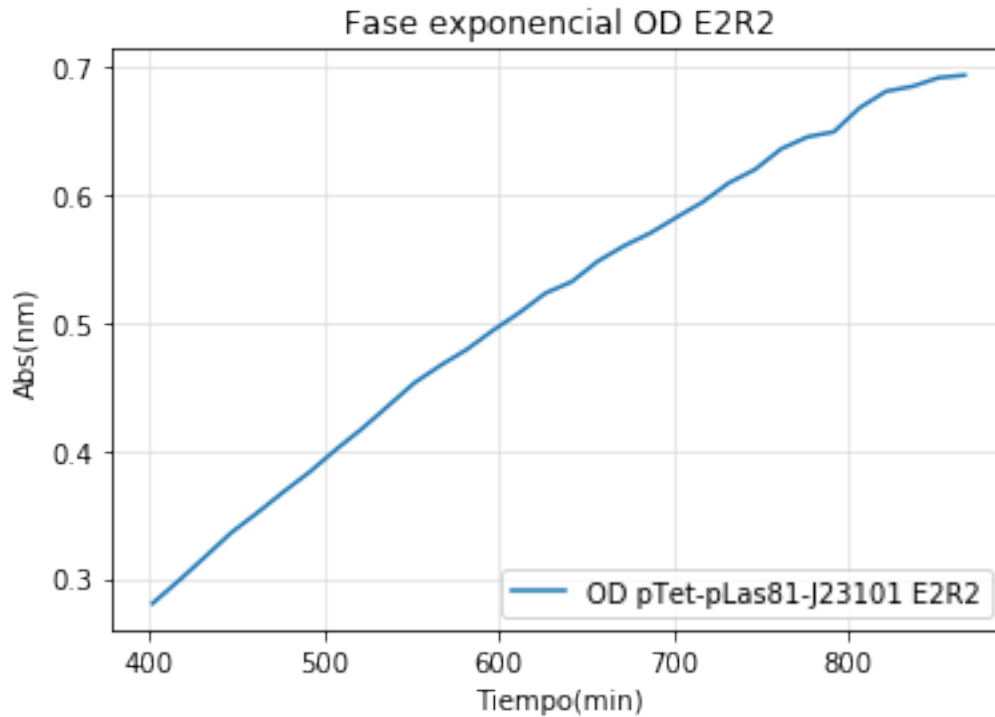


A=2.188982e+00  
um=4.673584e-03

l=1.546927e+02  
Tm=3.269976e+02  
doubpe=1.483117e+02  
ext=3.707793e+02  
Tfinal=6.977769e+02

Out[93]: <matplotlib.legend.Legend at 0x17896a366a0>





```
In [94]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 3
y24= np.log(od1527g3)-np.log(np.min(od1527g3))
print('Min OD = %e'%((np.min(od1527g3))))
evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%')
A24= params[0]
um24=params[1]
l24=params[2]
print('A=%e'%(A24))
print('um=%e'%(um24))
print('l=%e'%(l24))

#Cálculo datos para determinar extensión de la fase exponencial
tm24=((A24/(np.exp(1)*um24))+l24)
print('Tm=%e'%(tm24))
t224=((np.log(2))/um24)
print('doubpe=%e'%(t224))
extdp24=2*t224
print('ext=%e'%extdp24)
ttot24=tm24+extdp24
print('Tfinal=%e'%ttot24)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

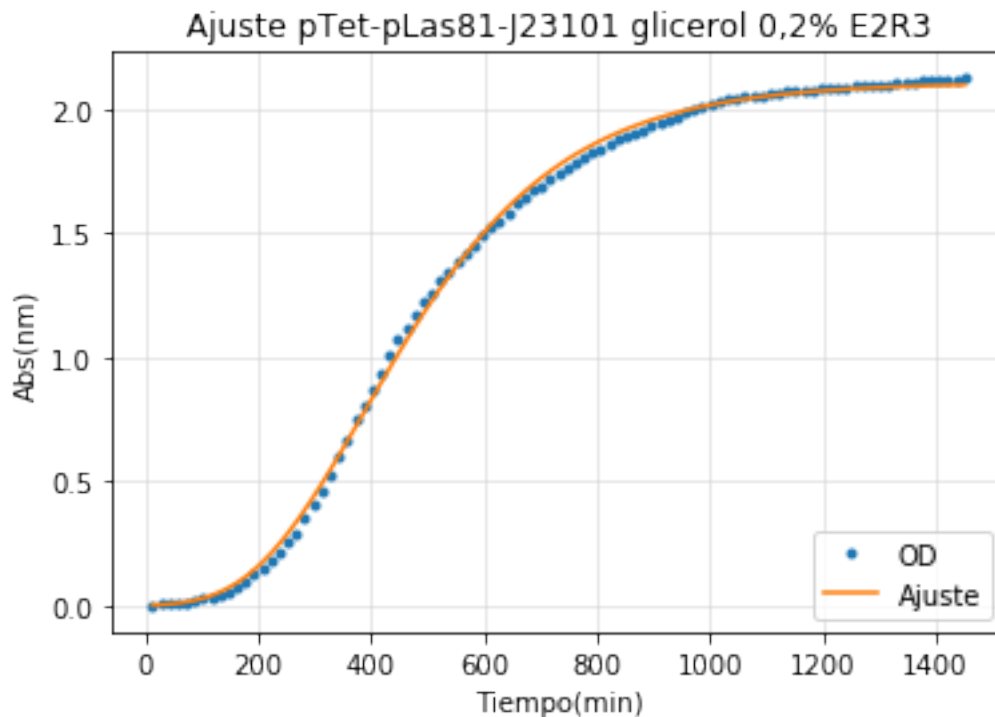
y2=tt[56]
plt.figure()
plt.title('pTet-pLas81-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1527g3,label='OD pTet-pLas81-J23101 E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:57],od1527g3[25:57],label='OD pTet-pLas81-J23101 E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.350000e-02

[ 2.10762223e+00 3.94236767e-03 1.88946108e+02]

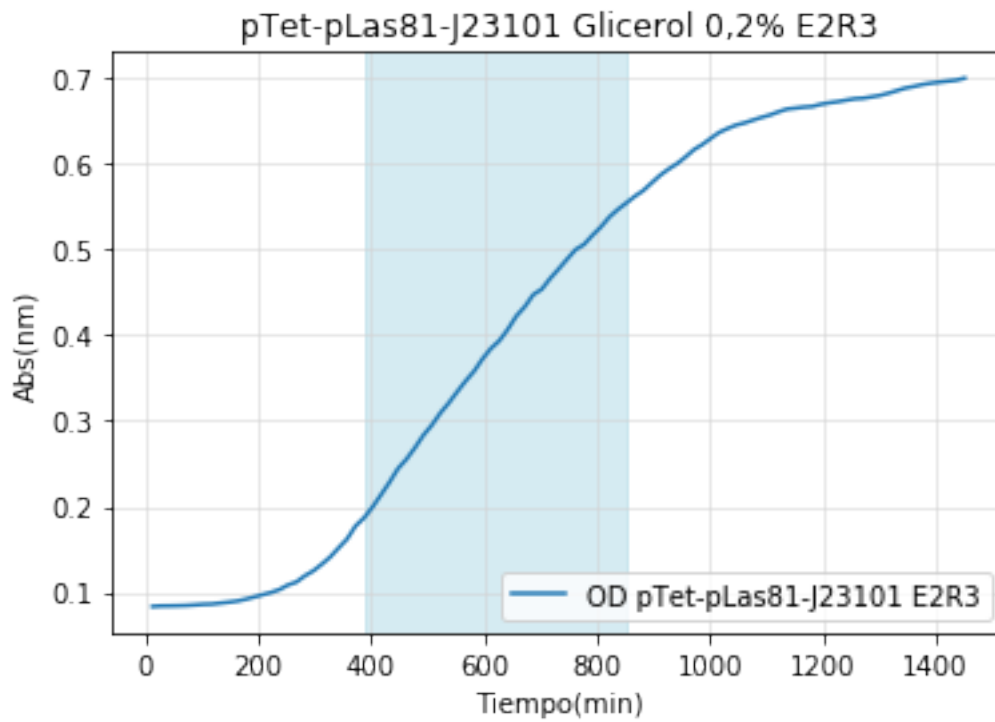


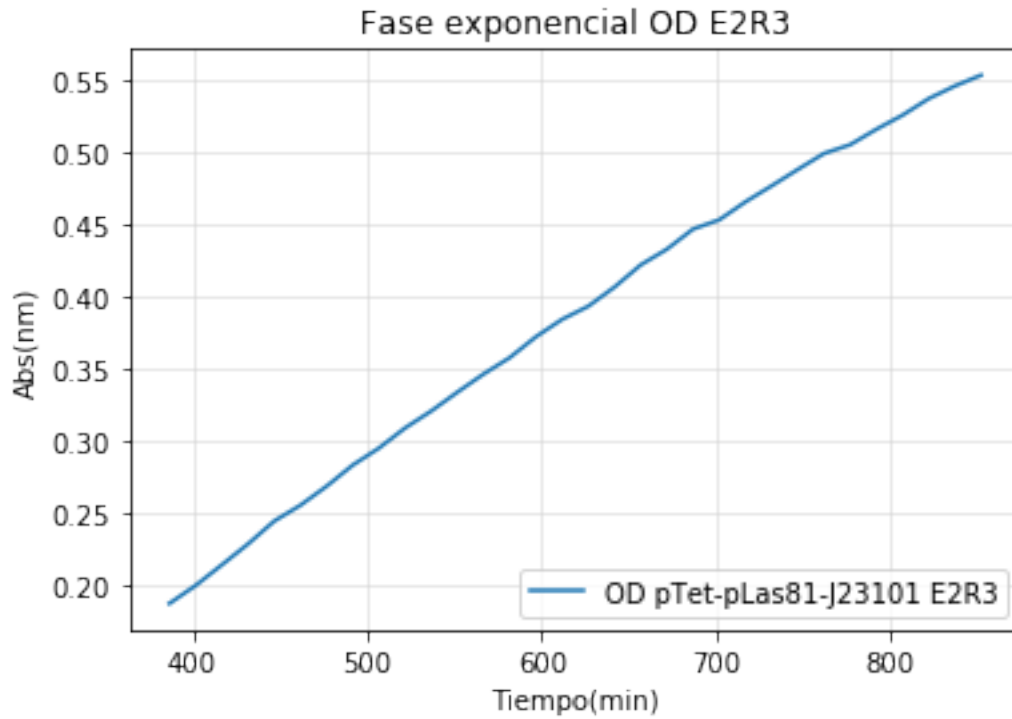
A=2.107622e+00  
um=3.942368e-03



```
l=1.889461e+02  
Tm=3.856175e+02  
doubpe=1.758200e+02  
ext=3.516400e+02  
Tfinal=7.372575e+02
```

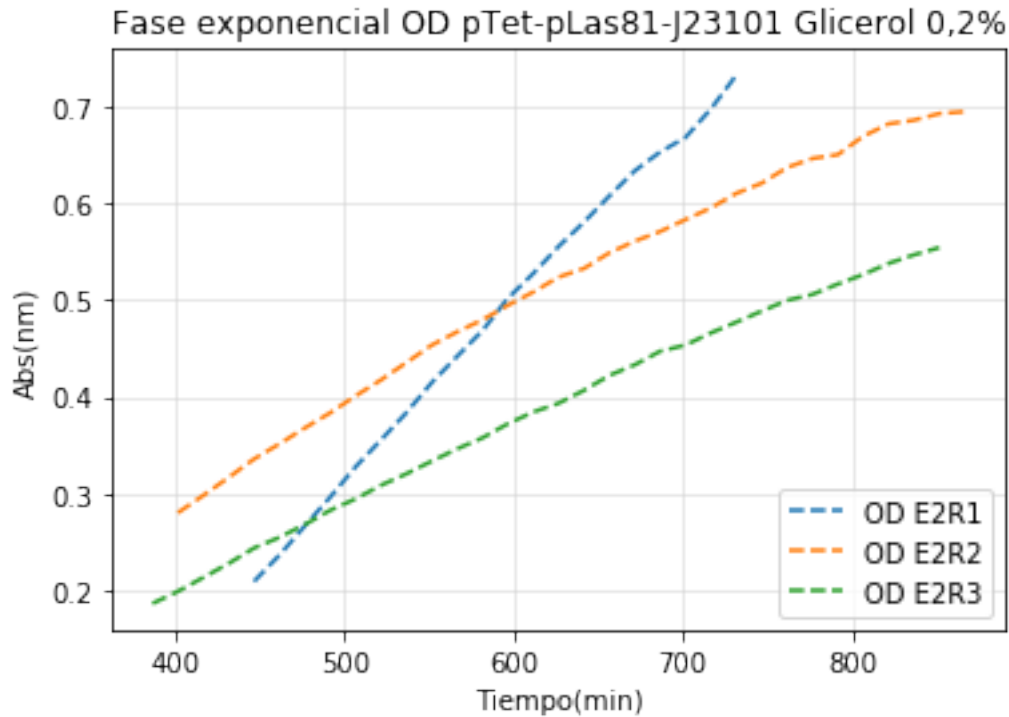
Out[94]: <matplotlib.legend.Legend at 0x17896c0f9e8>





```
In [95]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pTet-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[29:49],od1527g1[29:49], '--',label='OD E2R1')
plt.plot(tt[26:58],od1527g2[26:58], '--',label='OD E2R2')
plt.plot(tt[25:57],od1527g3[25:57], '--',label='OD E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

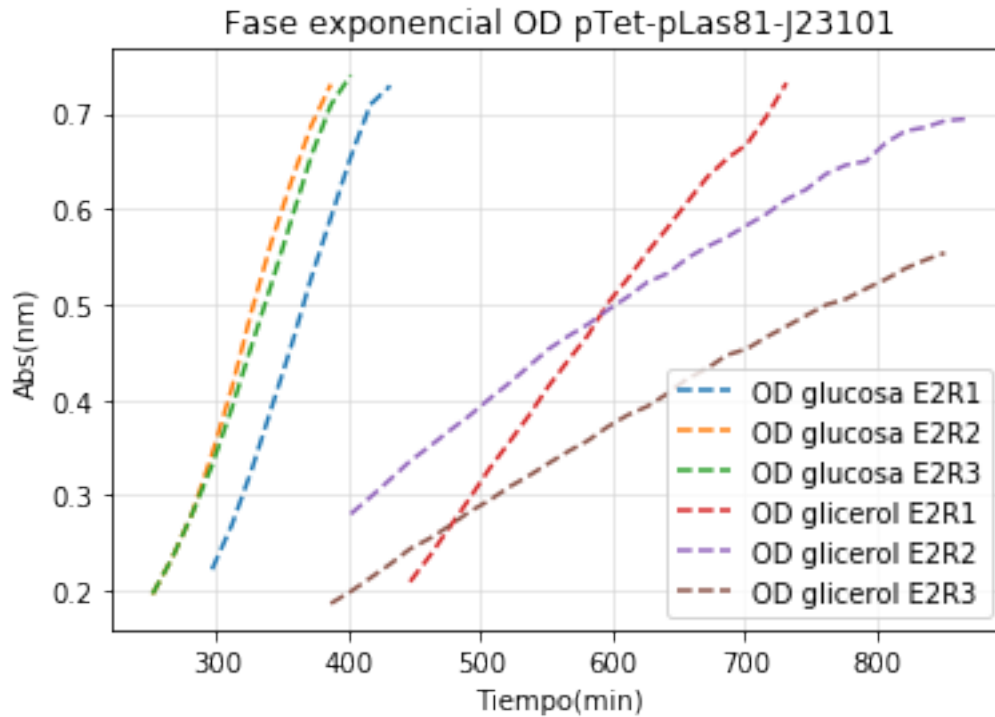
```
Out[95]: <matplotlib.legend.Legend at 0x17896d4cf28>
```



In [96]: *#Fase exponencial OD/tiempo*

```
plt.figure()
plt.title('Fase exponencial OD pTet-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:29],od15271[19:29], '--',label='OD glucosa E2R1')
plt.plot(tt[16:26],od15272[16:26], '--',label='OD glucosa E2R2')
plt.plot(tt[16:27],od15273[16:27], '--',label='OD glucosa E2R3')
plt.plot(tt[29:49],od1527g1[29:49], '--',label='OD glicerol E2R1')
plt.plot(tt[26:58],od1527g2[26:58], '--',label='OD glicerol E2R2')
plt.plot(tt[25:57],od1527g3[25:57], '--',label='OD glicerol E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[96]: <matplotlib.legend.Legend at 0x17896e11c18>



```
In [97]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#lux-std-std glucosa rep 1
y25= np.log(od18271)-np.log(np.min(od18271))
print('Min OD = %e'%((np.min(od18271))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,4
A25= params[0]
um25=params[1]
l25=params[2]
print('A=%e'%(A25))
print('um=%e'%(um25))
print('l=%e'%(l25))

#Cálculo datos para determinar extensión de la fase exponencial
tm25=((A25/(np.exp(1)*um25))+l25)
print('Tm=%e'%(tm25))
t225=((np.log(2))/um25)
print('doubpe=%e'%(t225))
extdp25=2*t225
print('ext=%e'%extdp25)
ttot25=tm25+extdp25
print('Tfinal=%e'%ttot25)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```

```

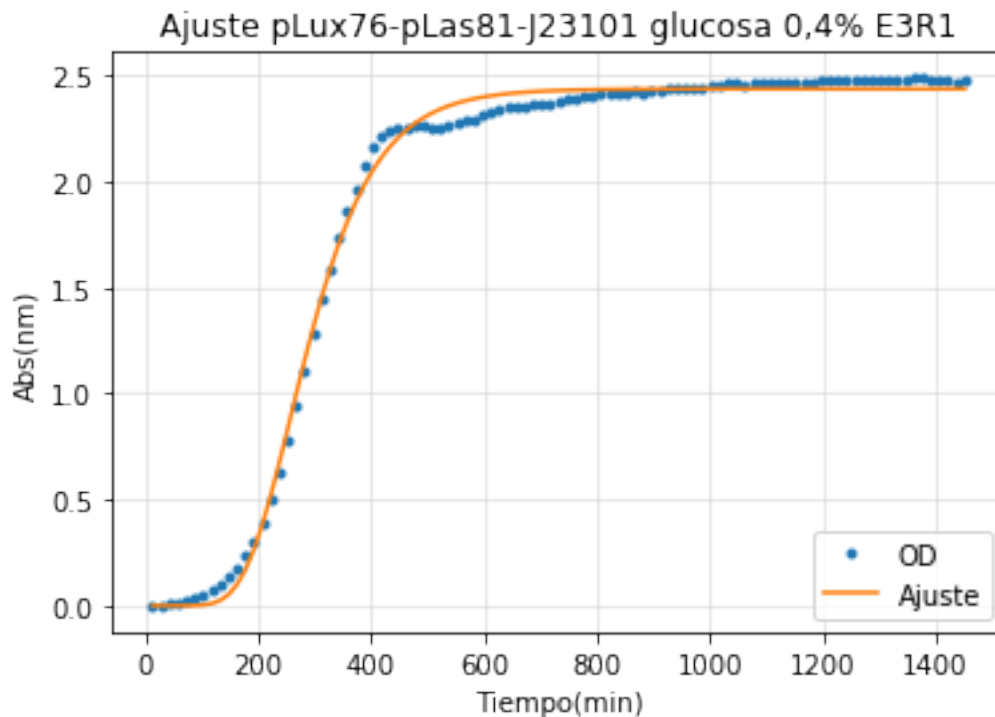
y2=tt[26]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18271,label='OD pLux76-pLas81-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:27],od18271[17:27],label='OD pLux76-pLas81-J23101 E3R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.540000e-02

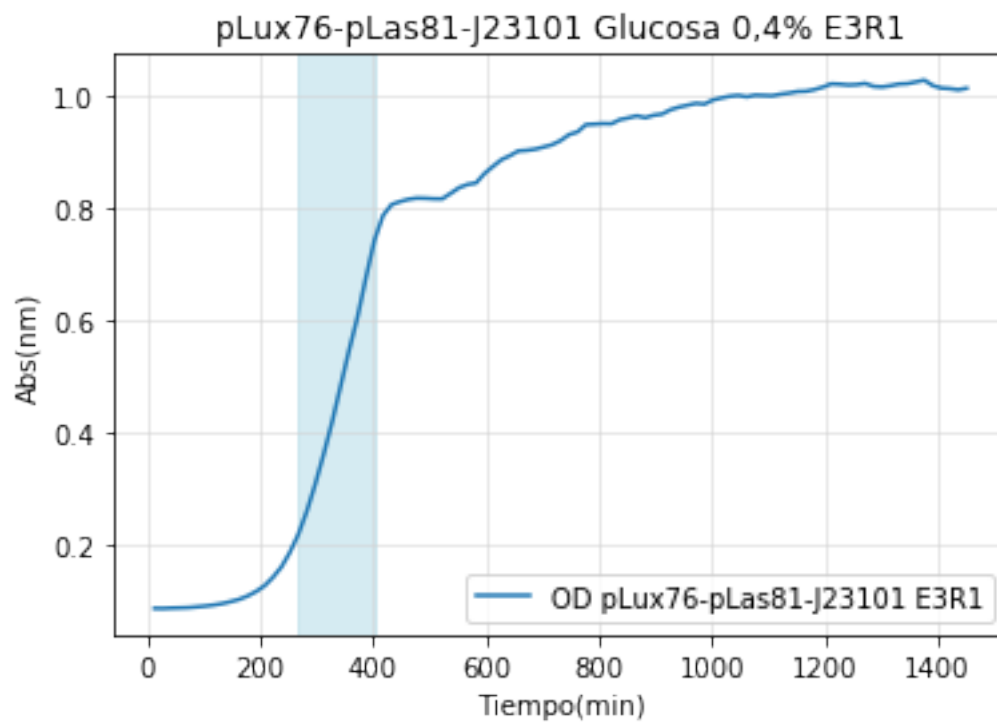
[ 2.43767957e+00 1.09002530e-02 1.74738266e+02]

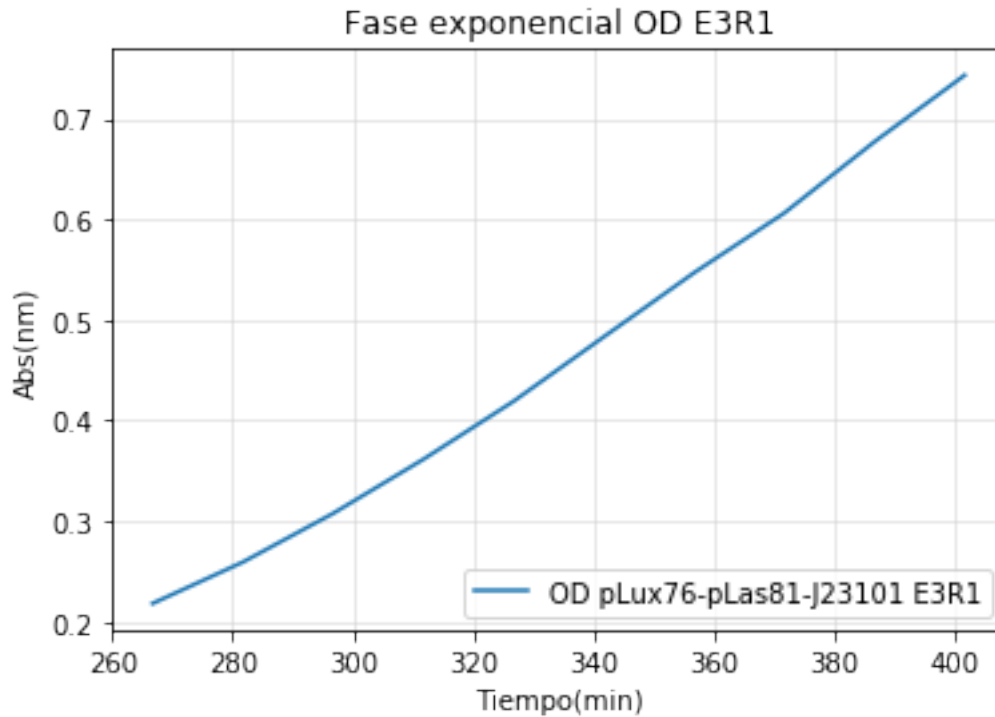


A=2.437680e+00  
um=1.090025e-02

l=1.747383e+02  
Tm=2.570090e+02  
doubpe=6.359001e+01  
ext=1.271800e+02  
Tfinal=3.841891e+02

Out[97]: <matplotlib.legend.Legend at 0x17896cc6fd0>





```
In [98]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#lux-std-std glucosa rep 2
y26= np.log(od18272)-np.log(np.min(od18272))
print('Min OD = %e'%((np.min(od18272))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,4
A26= params[0]
um26=params[1]
l26=params[2]
print('A=%e'%(A26))
print('um=%e'%(um26))
print('l=%e'%(l26))

#Cálculo datos para determinar extensión de la fase exponencial
tm26=((A26/(np.exp(1)*um26))+l26)
print('Tm=%e'%(tm26))
t226=((np.log(2))/um26)
print('doubpe=%e'%(t226))
extdp26=2*t226
print('ext=%e'%extdp26)
ttot26=tm26+extdp26
print('Tfinal=%e'%ttot26)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```

```

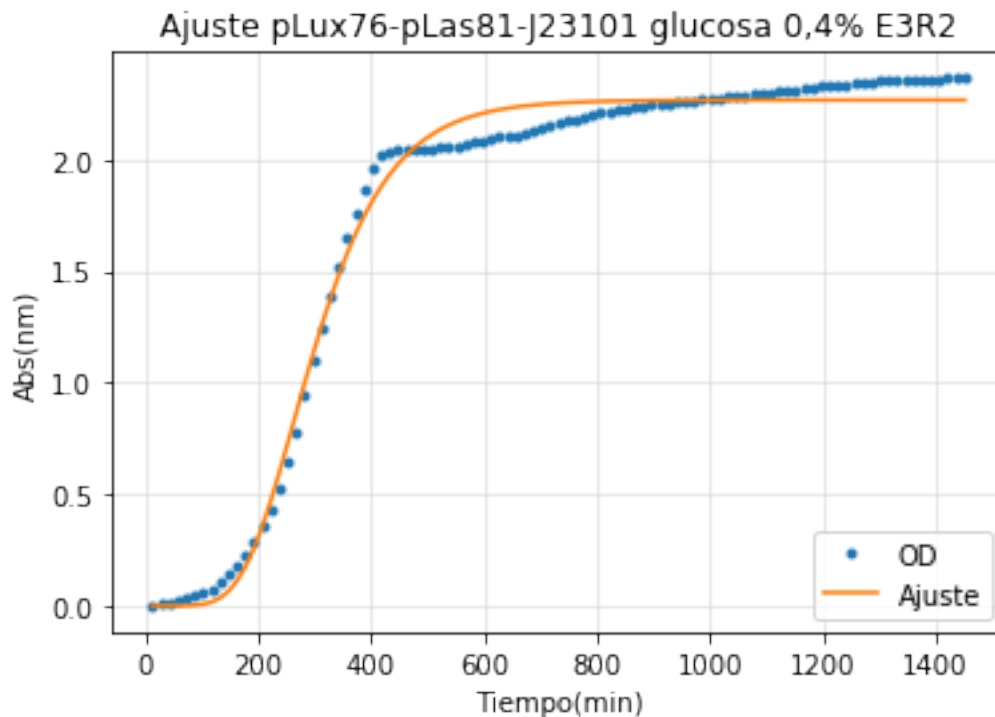
y2=tt[28]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18272,label='OD pLux76-pLas81-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],od18272[17:29],label='OD pLux76-pLas81-J23101 E3R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.220000e-02

[ 2.26837994e+00 9.06999393e-03 1.71041461e+02]

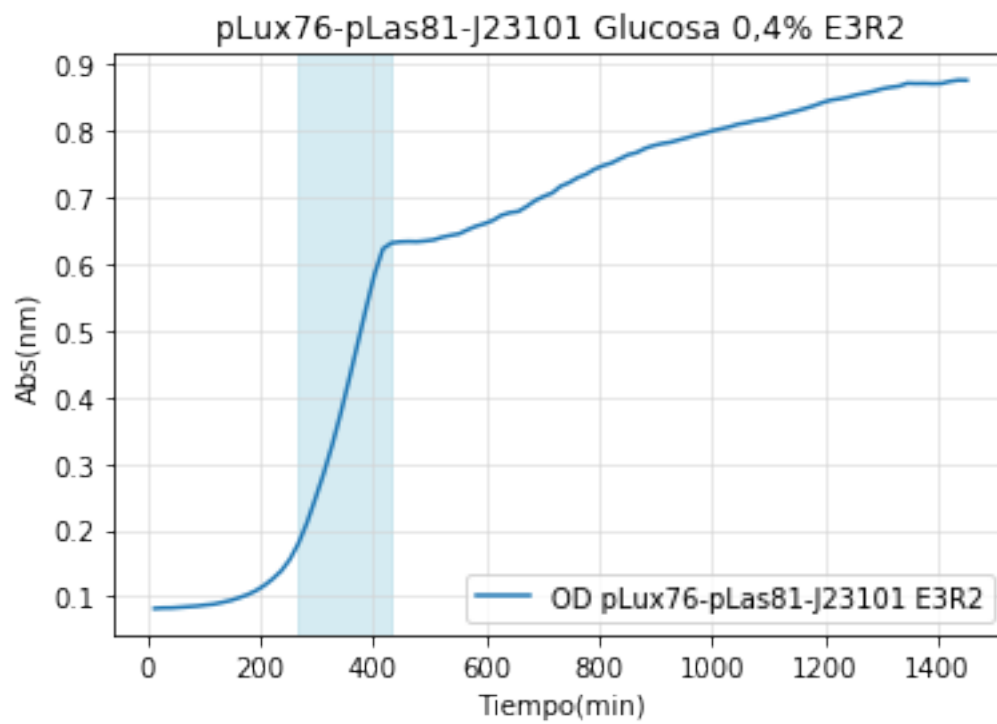


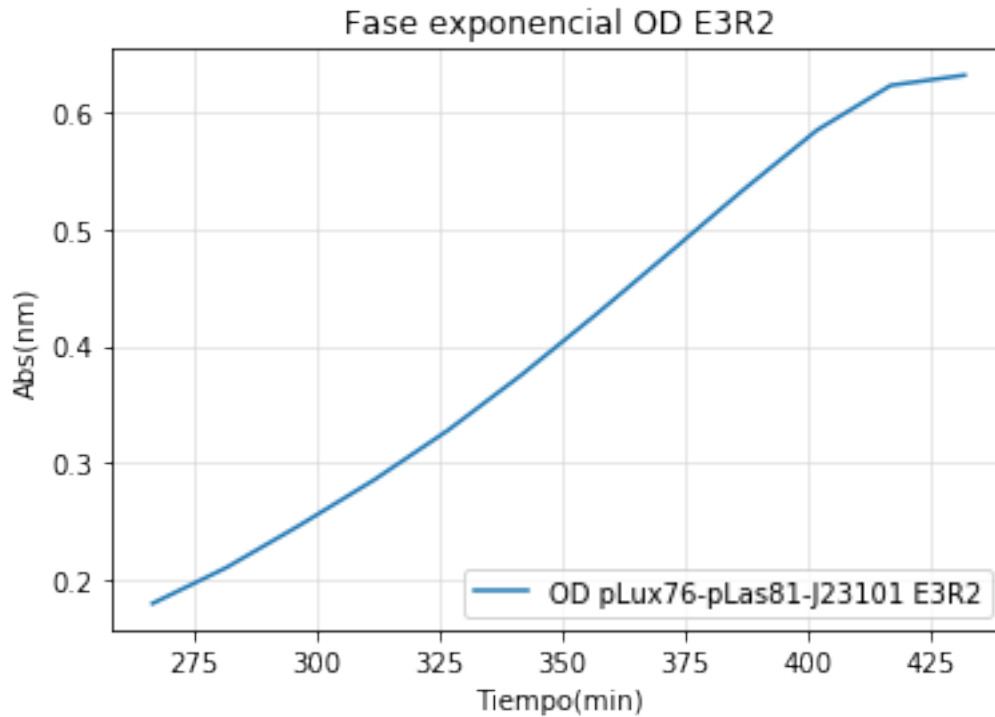
A=2.268380e+00  
um=9.069994e-03



l=1.710415e+02  
Tm=2.630471e+02  
doubpe=7.642201e+01  
ext=1.528440e+02  
Tfinal=4.158911e+02

Out[98]: <matplotlib.legend.Legend at 0x178968945c0>





```
In [102]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 3
y27= np.log(od18273)-np.log(np.min(od18273))
print('Min OD = %e'%((np.min(od18273))))
evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,
A27= params[0]
um27=params[1]
l27=params[2]
print('A=%e'%(A27))
print('um=%e'%(um27))
print('l=%e'%(l27))

#Cálculo datos para determinar extensión de la fase exponencial
tm27=((A27/(np.exp(1)*um27))+l27)
print('Tm=%e'%(tm27))
t227=((np.log(2))/um27)
print('doubpe=%e'%(t227))
extdp27=2*t227
print('ext=%e'%extdp27)
ttot27=tm27+extdp27
print('Tfinal=%e'%ttot27)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```

```

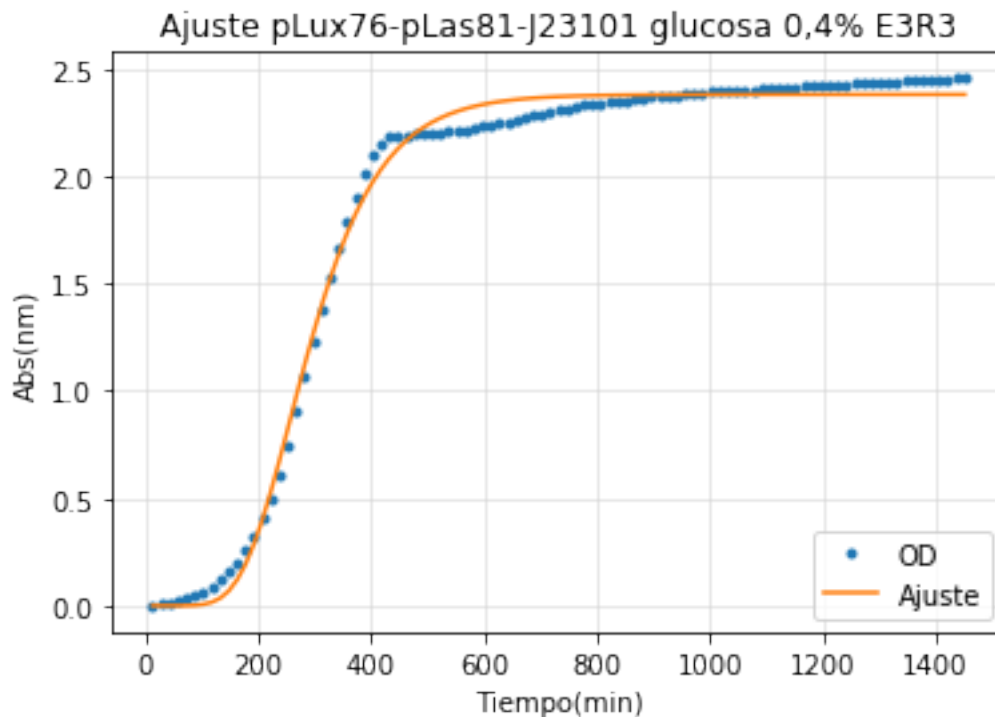
y2=tt[27]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18273,label='OD pLux76-pLas81-J23101 E3R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],od18273[17:28],label='OD pLux76-pLas81-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.500000e-02

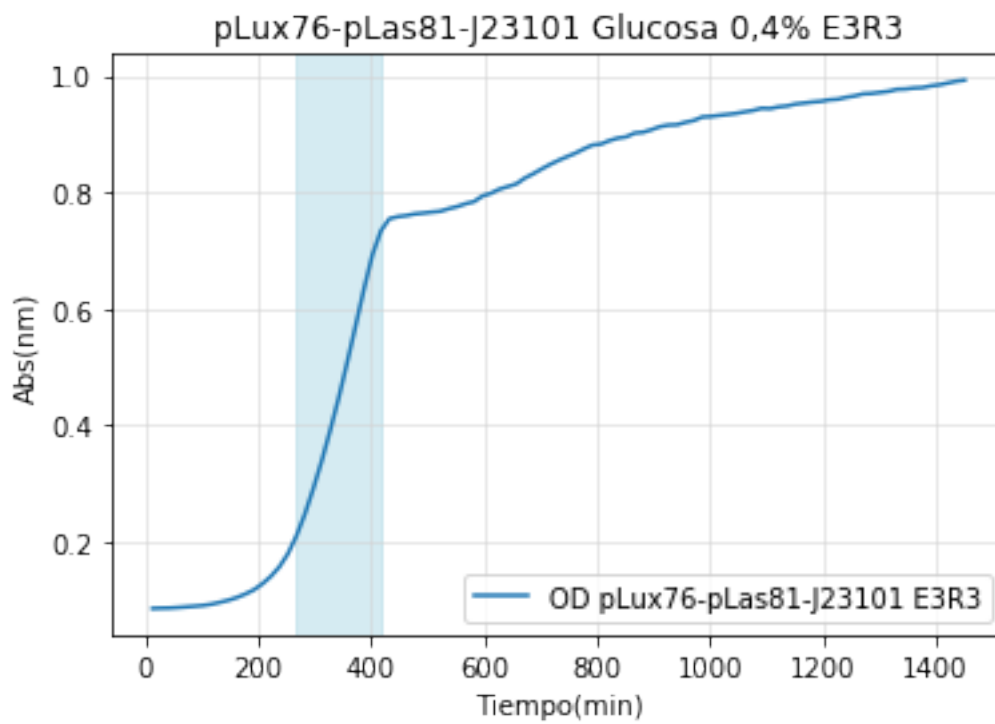
[ 2.38059703e+00 1.00857409e-02 1.69775385e+02]

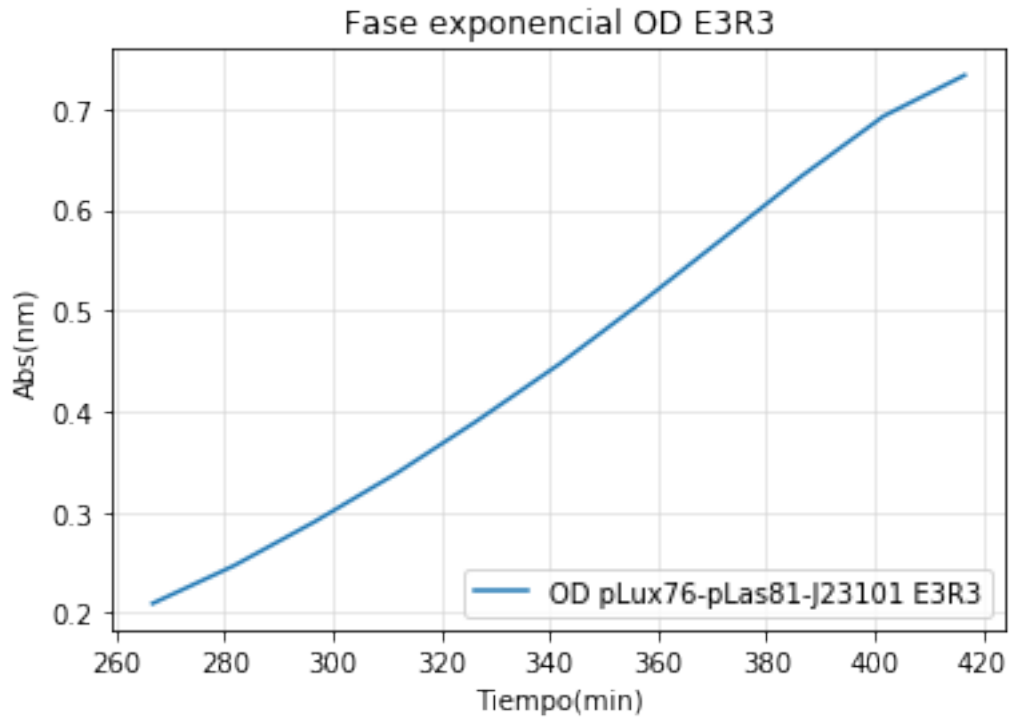


A=2.380597e+00  
um=1.008574e-02

```
l=1.697754e+02  
Tm=2.566081e+02  
doubpe=6.872546e+01  
ext=1.374509e+02  
Tfinal=3.940591e+02
```

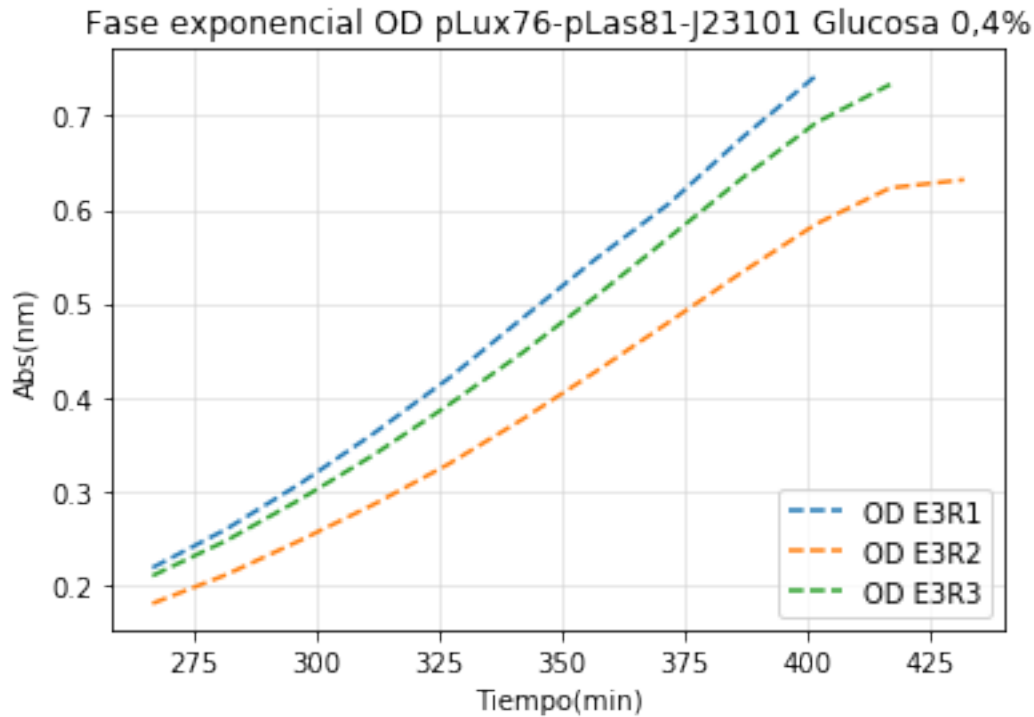
Out[102]: <matplotlib.legend.Legend at 0x1789552e7b8>





```
In [103]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:27],od18271[17:27], '--',label='OD E3R1')
plt.plot(tt[17:29],od18272[17:29], '--',label='OD E3R2')
plt.plot(tt[17:28],od18273[17:28], '--',label='OD E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[103]: <matplotlib.legend.Legend at 0x17896f4def0>
```



```
In [104]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#lux-std-std glicerol rep 1
y28= np.log(od1827g1)-np.log(np.min(od1827g1))
print('Min OD = %e'%((np.min(od1827g1))))
evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0
A28= params[0]
um28=params[1]
l28=params[2]
print('A=%e'%(A28))
print('um=%e'%(um28))
print('l=%e'%(l28))

#Cálculo datos para determinar extensión de la fase exponencial
tm28=((A28/(np.exp(1)*um28))+l28)
print('Tm=%e'%(tm28))
t228=((np.log(2))/um28)
print('doubpe=%e'%(t228))
extdp28=2.5*t228
print('ext=%e'%extdp28)
ttot28=tm28+extdp28
print('Tfinal=%e'%ttot28)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[27]
```

```

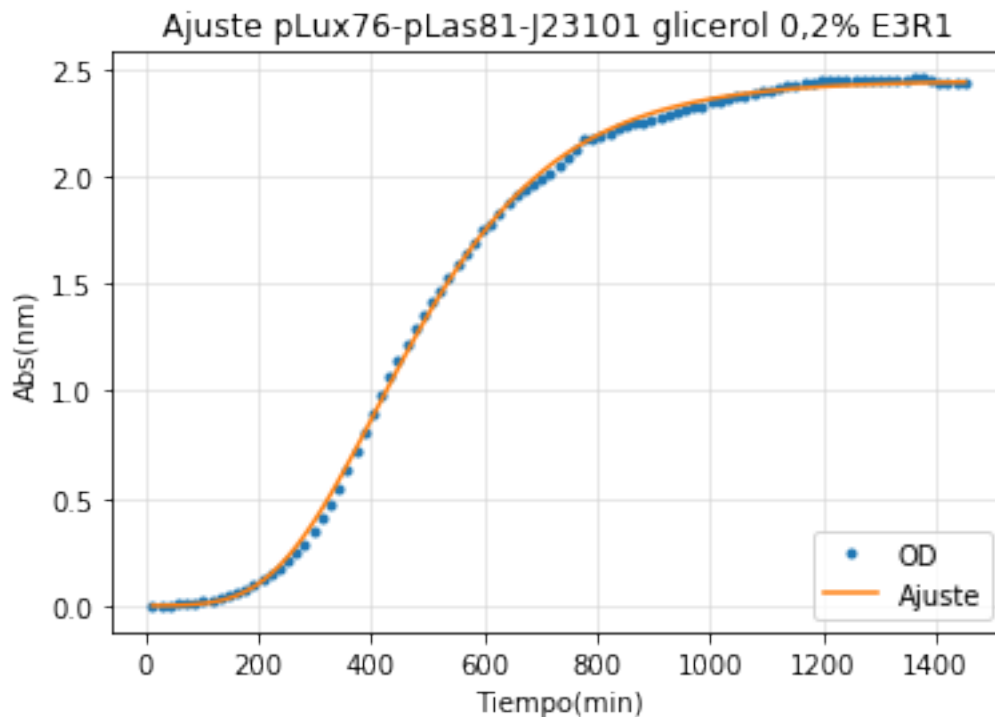
y2=tt[51]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1827g1,label='OD pLux76-pLas81-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:52],od1827g1[27:52],label='OD pLux76-pLas81-J23101 E3R1')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.650000e-02

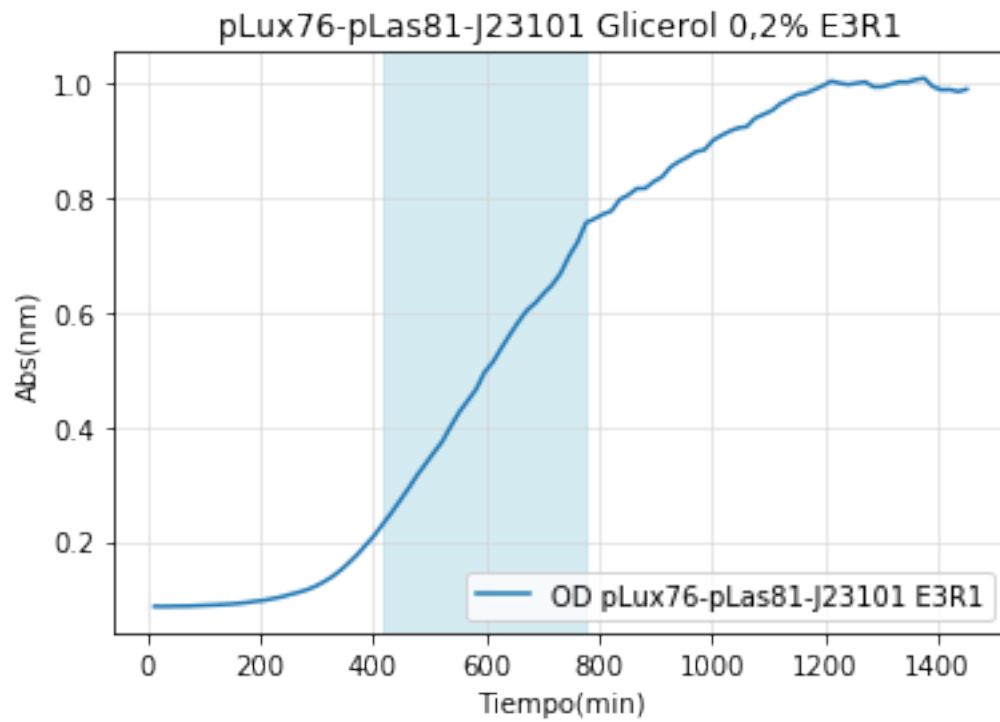
[ 2.44638066e+00 5.03429015e-03 2.27140894e+02]



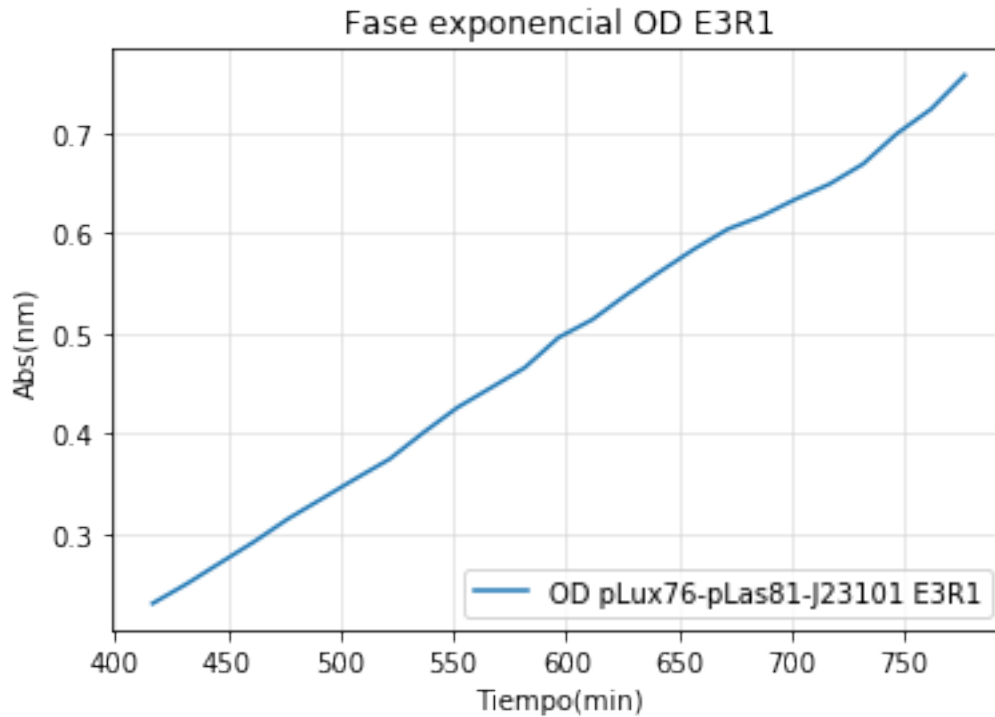
A=2.446381e+00  
um=5.034290e-03

```
l=2.271409e+02
Tm=4.059095e+02
doubpe=1.376852e+02
ext=3.442130e+02
Tfinal=7.501225e+02
```

Out[104]: <matplotlib.legend.Legend at 0x1788cc94278>







```
In [105]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#lux-std-std glicerol rep 2
y29= np.log(od1827g2)-np.log(np.min(od1827g2))
print('Min OD = %e'%((np.min(od1827g2))))
evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0
A29= params[0]
um29=params[1]
l29=params[2]
print('A=%e'%(A29))
print('um=%e'%(um29))
print('l=%e'%(l29))

#Cálculo datos para determinar extensión de la fase exponencial
tm29=((A29/(np.exp(1)*um29))+l29)
print('Tm=%e'%(tm29))
t229=((np.log(2))/um29)
print('doubpe=%e'%(t229))
extdp29=2.5*t229
print('ext=%e'%extdp29)
ttot29=tm29+extdp29
print('Tfinal=%e'%ttot29)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[26]
```

```

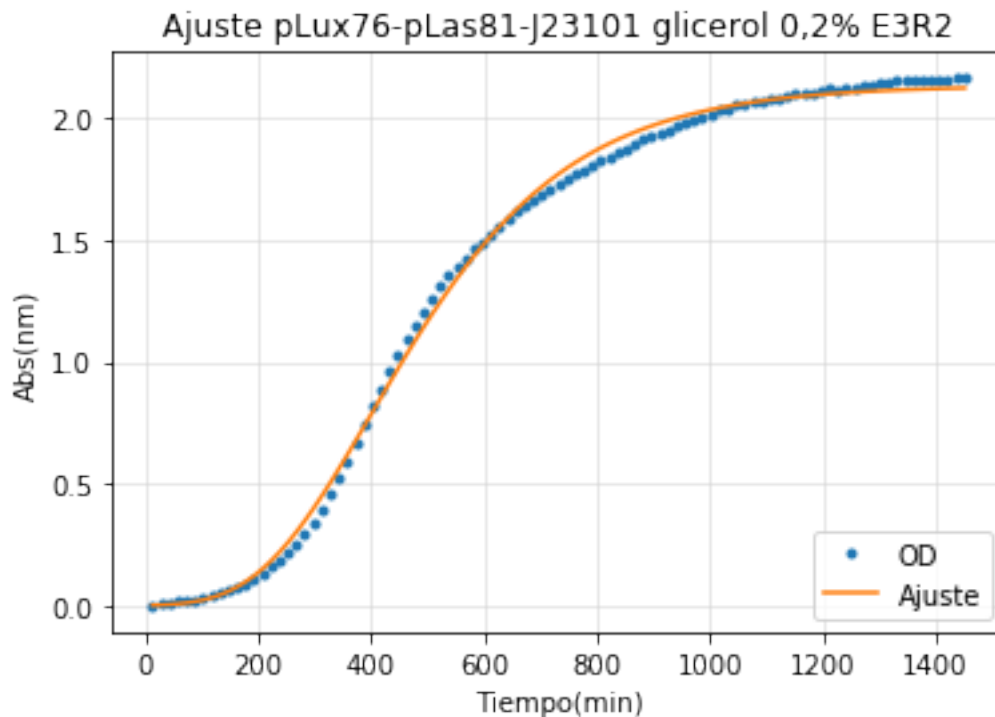
y2=tt[58]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1827g2,label='OD pLux76-pLas81-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:59],od1827g2[26:59],label='OD pLux76-pLas81-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.210000e-02

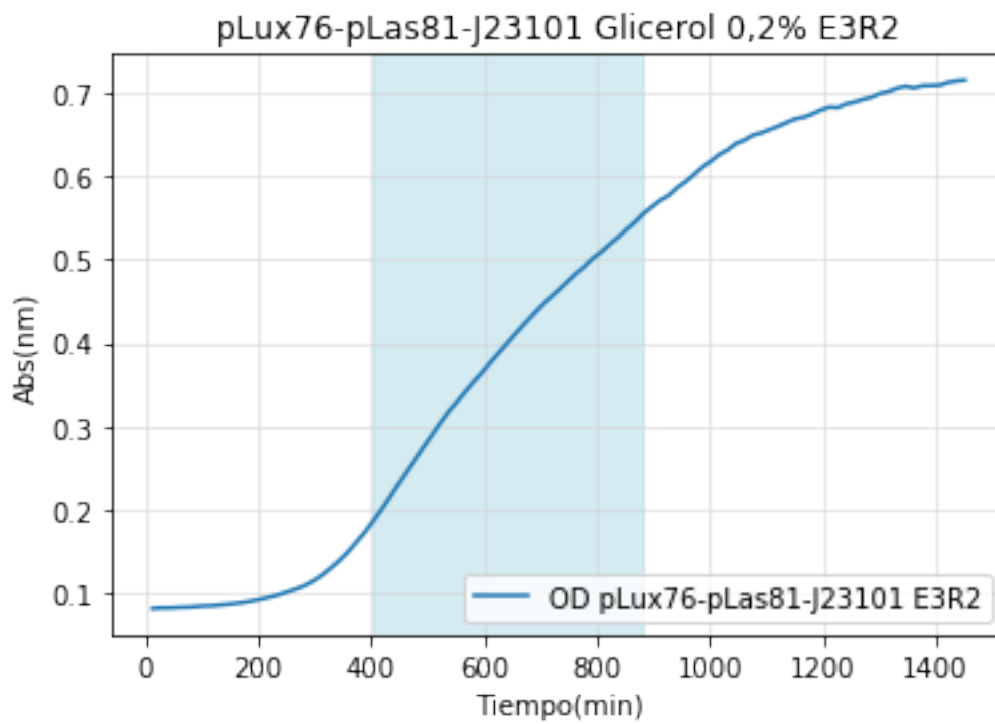
[ 2.13299146e+00 3.96657723e-03 2.01445904e+02]

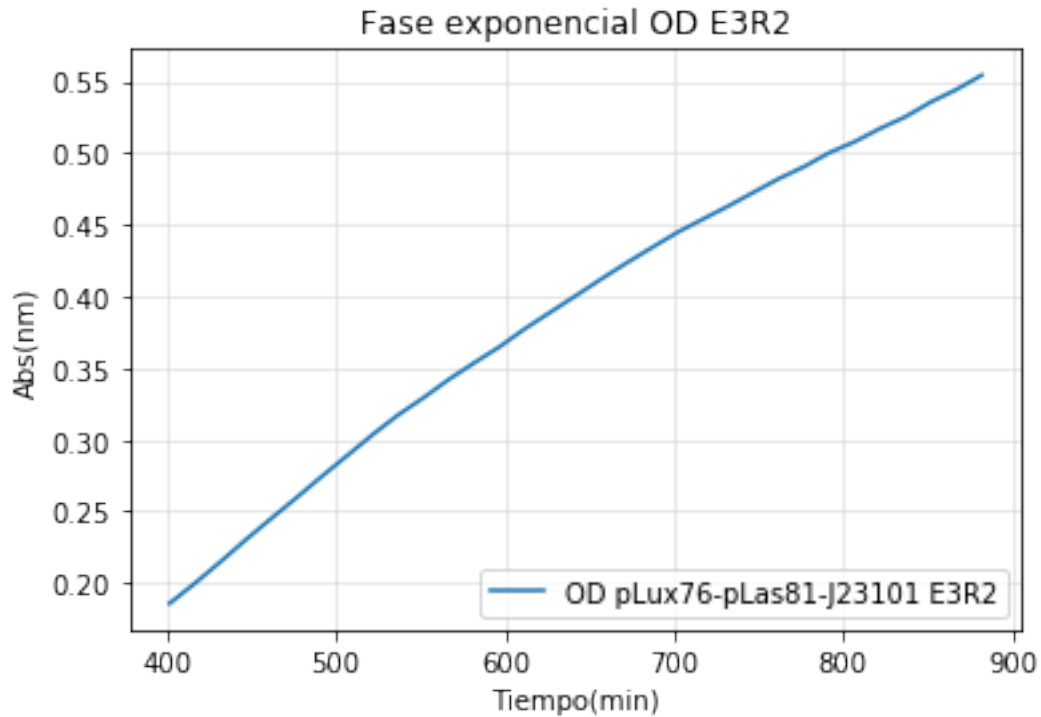


A=2.132991e+00  
um=3.966577e-03

```
l=2.014459e+02  
Tm=3.992698e+02  
doubpe=1.747469e+02  
ext=4.368673e+02  
Tfinal=8.361371e+02
```

Out[105]: <matplotlib.legend.Legend at 0x178974fe710>





```
In [106]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#lux-std-std glicerol rep 3
y30= np.log(od1827g3)-np.log(np.min(od1827g3))
print('Min OD = %e'%((np.min(od1827g3))))
evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0
A30= params[0]
um30=params[1]
l30=params[2]
print('A=%e'%(A30))
print('um=%e'%(um30))
print('l=%e'%(l30))

#Cálculo datos para determinar extensión de la fase exponencial
tm30=((A30/(np.exp(1)*um30))+l30)
print('Tm=%e'%(tm30))
t230=((np.log(2))/um30)
print('doubpe=%e'%(t230))
extdp30=2*t230
print('ext=%e'%extdp30)
ttot30=tm30+extdp30
print('Tfinal=%e'%ttot30)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
```

```

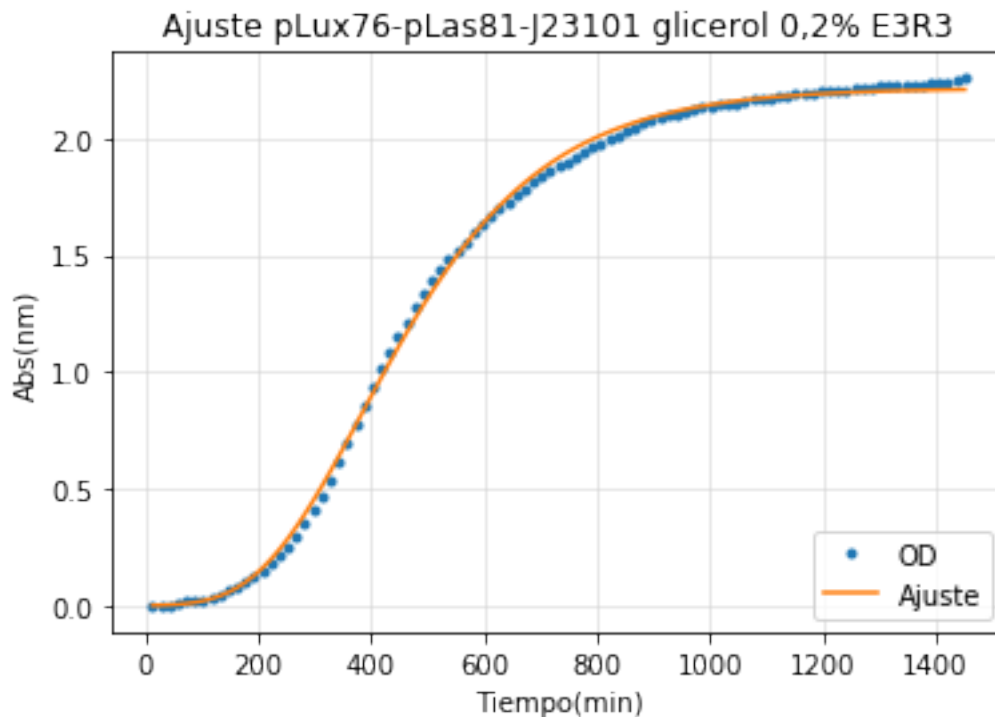
y2=tt[52]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1827g3,label='OD pLux76-pLas81-J23101 E3R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],od1827g3[25:53],label='OD pLux76-pLas81-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.420000e-02

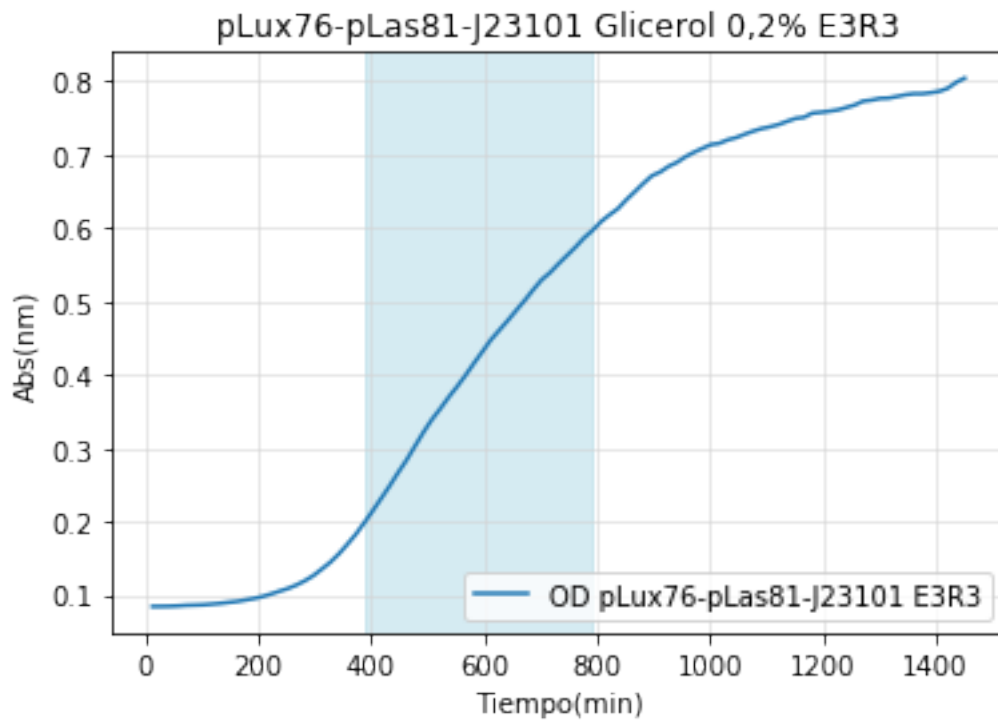
[ 2.21392685e+00 4.49243847e-03 1.99905763e+02]

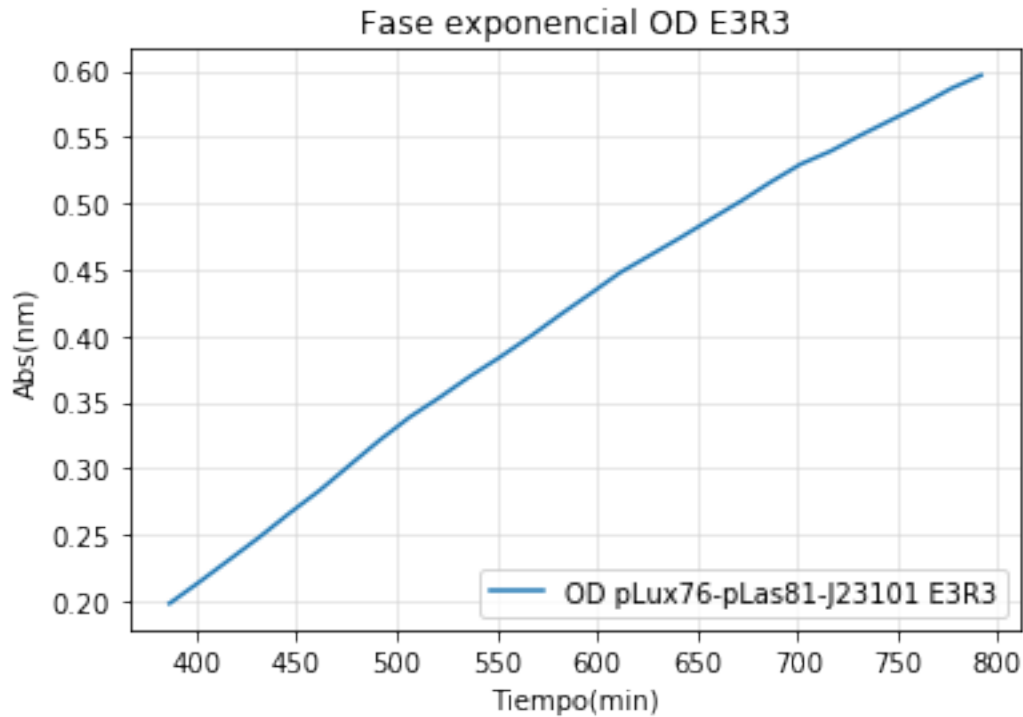


A=2.213927e+00  
um=4.492438e-03

```
l=1.999058e+02  
Tm=3.812011e+02  
doubpe=1.542920e+02  
ext=3.085839e+02  
Tfinal=6.897850e+02
```

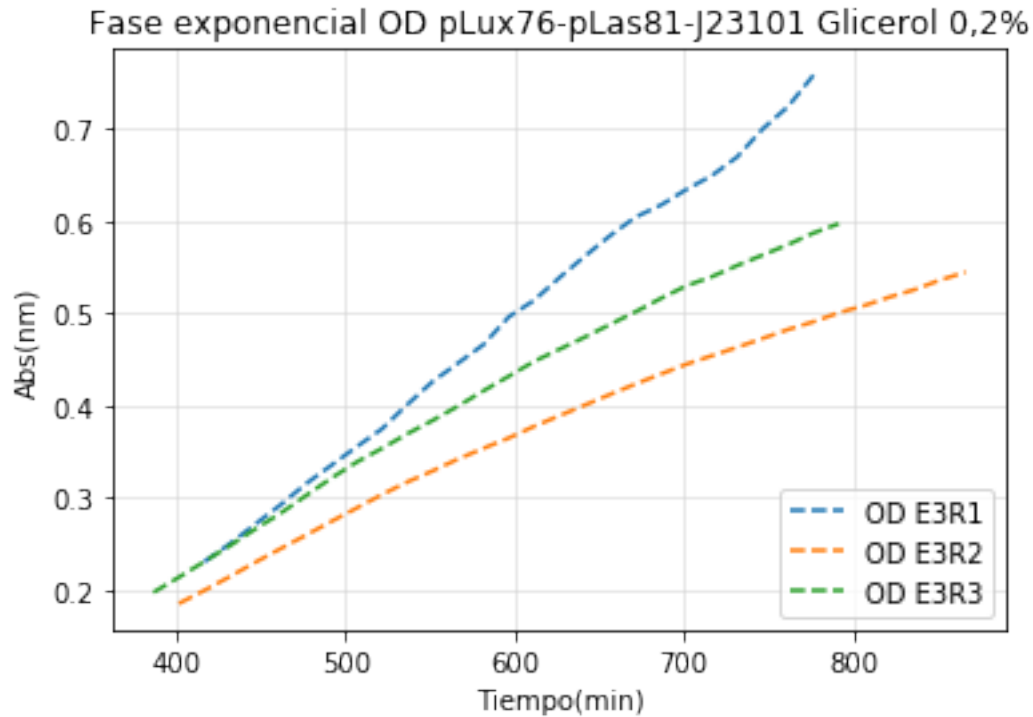
Out[106]: <matplotlib.legend.Legend at 0x178976e94a8>





```
In [107]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:52],od1827g1[27:52], '--',label='OD E3R1')
plt.plot(tt[26:58],od1827g2[26:58], '--',label='OD E3R2')
plt.plot(tt[25:53],od1827g3[25:53], '--',label='OD E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')
```

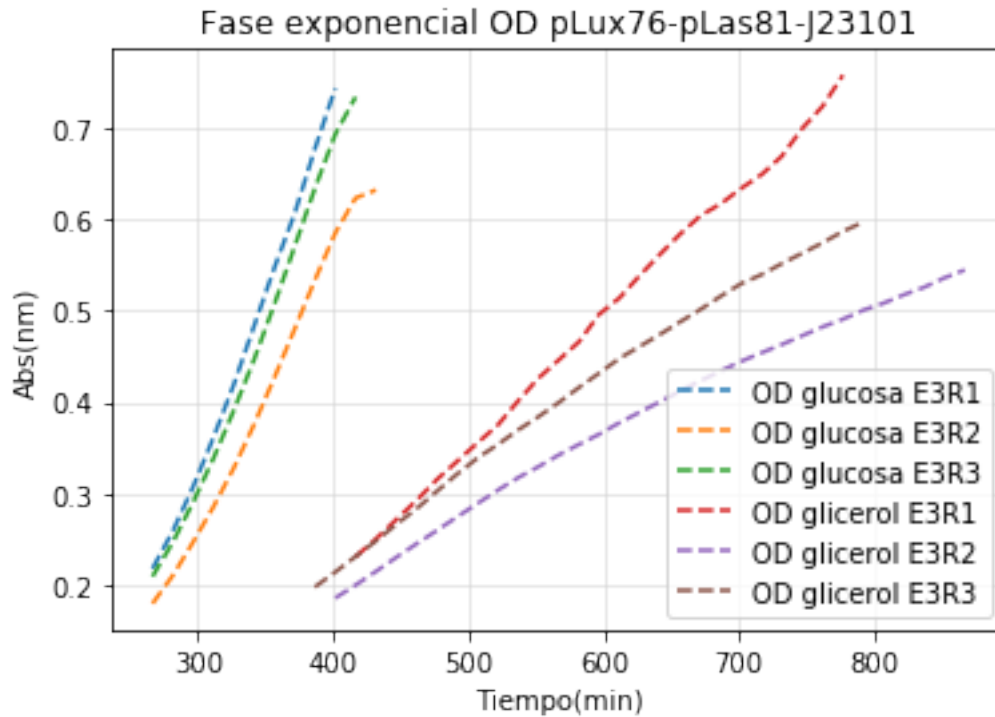
```
Out[107]: <matplotlib.legend.Legend at 0x1789880bcf8>
```



```
In [109]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:27],od18271[17:27], '--',label='OD glucosa E3R1')
plt.plot(tt[17:29],od18272[17:29], '--',label='OD glucosa E3R2')
plt.plot(tt[17:28],od18273[17:28], '--',label='OD glucosa E3R3')
plt.plot(tt[27:52],od1827g1[27:52], '--',label='OD glicerol E3R1')
plt.plot(tt[26:58],od1827g2[26:58], '--',label='OD glicerol E3R2')
plt.plot(tt[25:53],od1827g3[25:53], '--',label='OD glicerol E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[109]: <matplotlib.legend.Legend at 0x178976194a8>
```





```
In [110]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 1
y31= np.log(od12271)-np.log(np.min(od12271))
print('Min OD = %e'%((np.min(od12271))))
evaly, params=Function_fit(tt,y31,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,
A31 = params[0]
um31=params[1]
l31=params[2]
print('A=%e'%(A31))
print('um=%e'%(um31))
print('l=%e'%(l31))

#Cálculo datos para determinar extensión de la fase exponencial
tm31=((A31/(np.exp(1)*um31))+l31)
print('Tm=%e'%(tm31))
t231=((np.log(2))/um31)
print('doubpe=%e'%(t231))
extdp31=2*t231
print('ext=%e'%extdp31)
ttot31=tm31+extdp31
print('Tfinal=%e'%ttot31)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
```

```

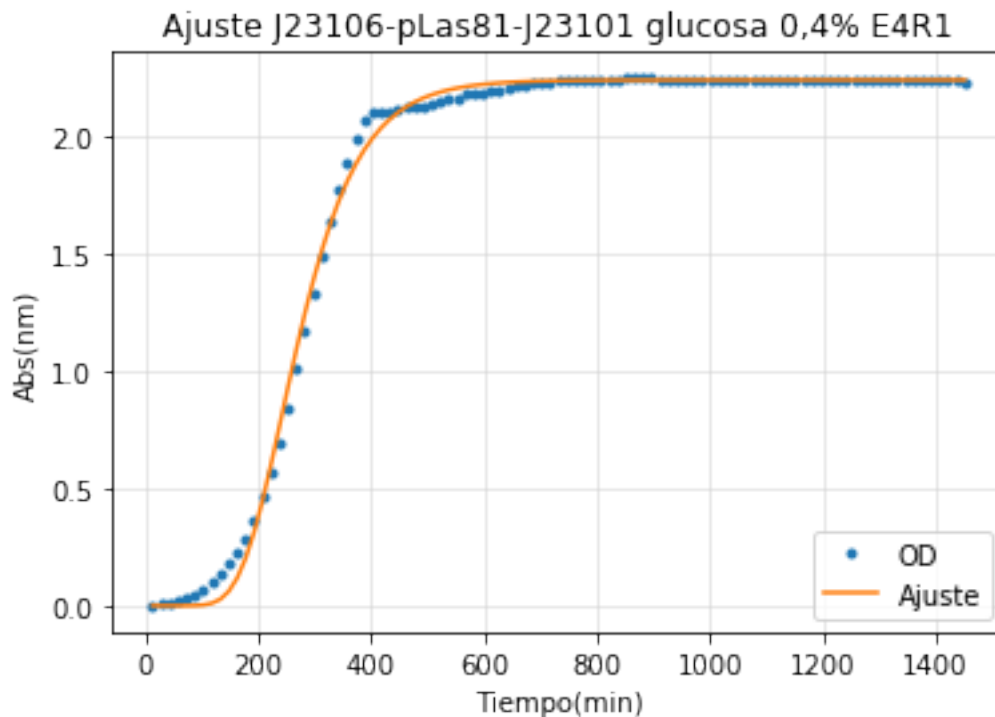
y2=tt[25]
plt.figure()
plt.title('J23106-pLas81-J23101 Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12271,label='OD J23106-pLas81-J23101 E4R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12271[16:26],label='OD J23106-pLas81-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.350000e-02

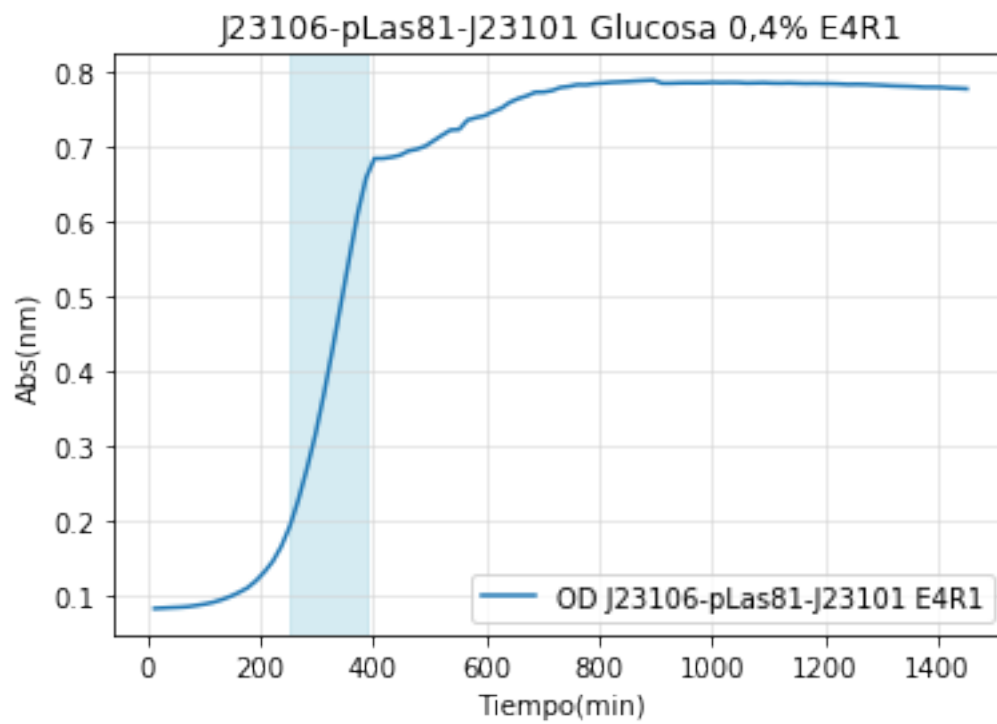
[ 2.23907480e+00 1.10691843e-02 1.67672817e+02]

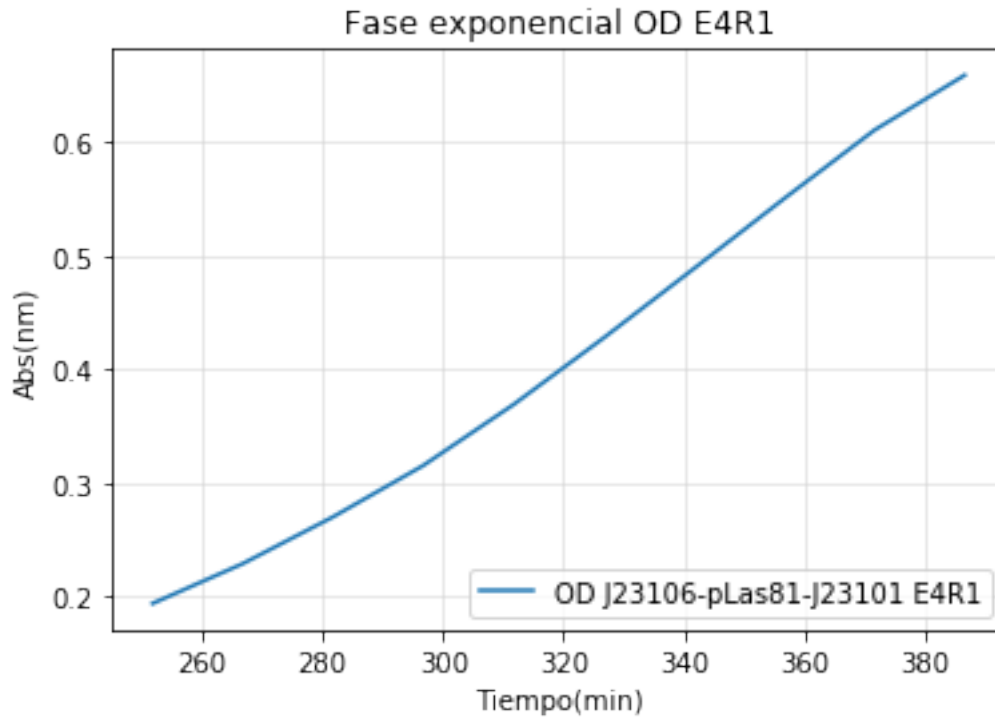


A=2.239075e+00  
um=1.106918e-02

l=1.676728e+02  
Tm=2.420875e+02  
doubpe=6.261954e+01  
ext=1.252391e+02  
Tfinal=3.673265e+02

Out[110]: <matplotlib.legend.Legend at 0x1788cc762b0>





```
In [111]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 2
y32= np.log(od12272)-np.log(np.min(od12272))
print('Min OD = %e'%((np.min(od12272))))
evaly, params=Function_fit(tt,y32,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,
A32= params[0]
um32=params[1]
l32=params[2]
print('A=%e'%(A32))
print('um=%e'%(um32))
print('l=%e'%(l32))

#Cálculo datos para determinar extensión de la fase exponencial
tm32=((A32/(np.exp(1)*um32))+l32)
print('Tm=%e'%(tm32))
t232=((np.log(2))/um32)
print('doubpe=%e'%(t232))
extdp32=2*t232
print('ext=%e'%extdp32)
ttot32=tm32+extdp32
print('Tfinal=%e'%ttot32)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
```

```

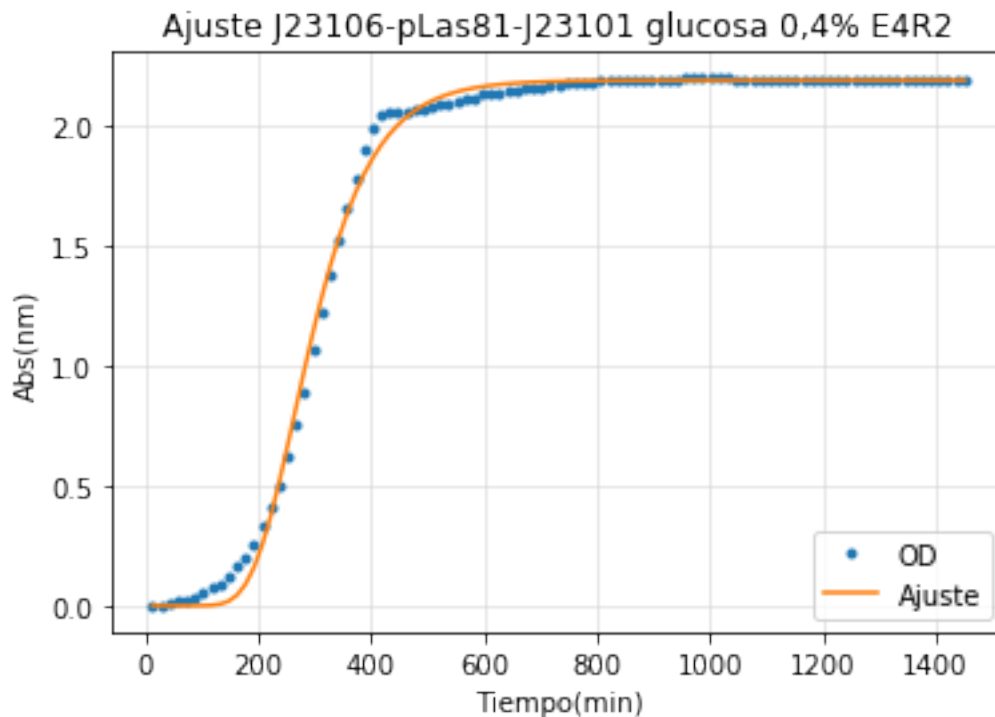
y2=tt[27]
plt.figure()
plt.title('J23106-pLas81-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12272,label='OD J23106-pLas81-J23101 E4R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],od12272[17:28],label='OD pLux76-pLas81-J23101 E4R2')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.360000e-02

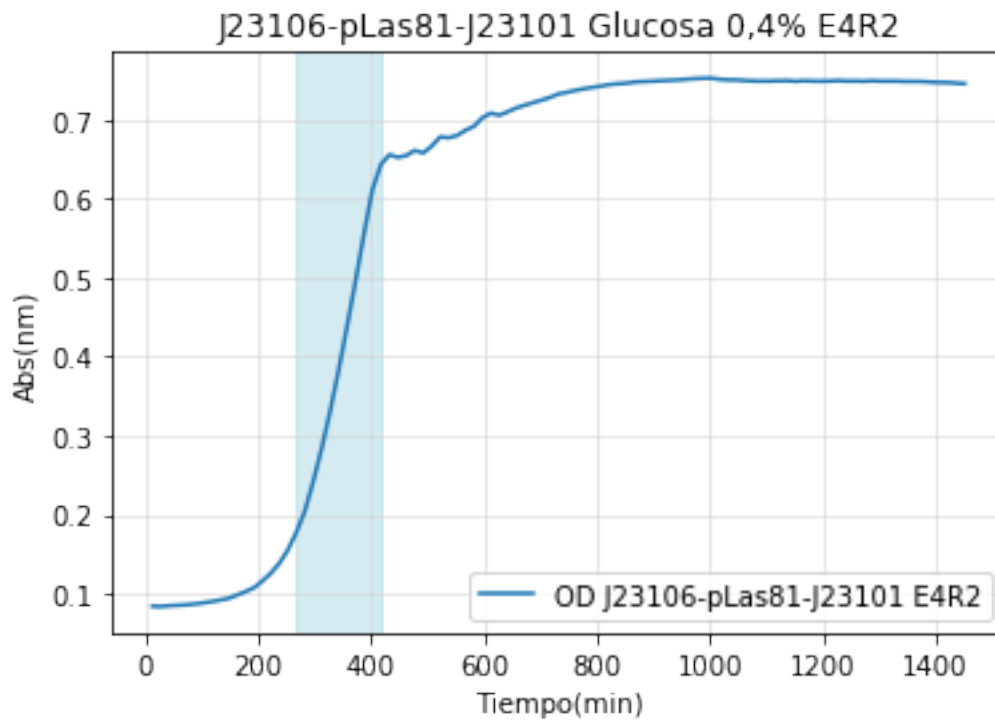
[ 2.18901562e+00 1.06857798e-02 1.88770693e+02]

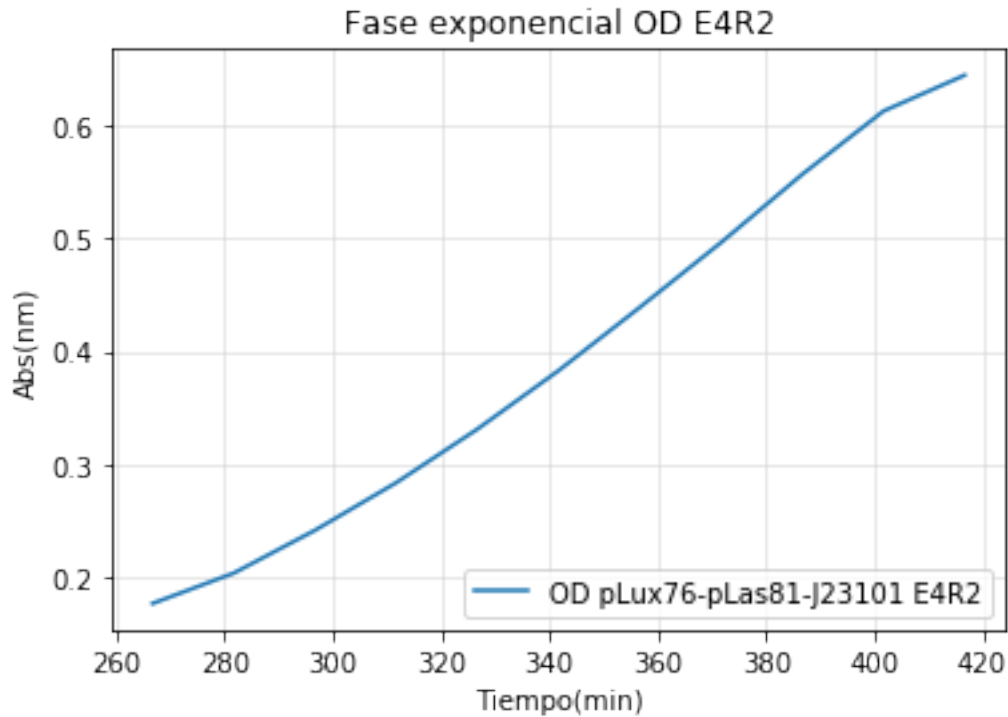


A=2.189016e+00  
um=1.068578e-02

```
l=1.887707e+02  
Tm=2.641320e+02  
doubpe=6.486632e+01  
ext=1.297326e+02  
Tfinal=3.938646e+02
```

Out[111]: <matplotlib.legend.Legend at 0x178952e9e80>





```
In [112]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 3
y33= np.log(od12273)-np.log(np.min(od12273))
print('Min OD = %e'%((np.min(od12273))))
evaly, params=Function_fit(tt,y33,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,
A33= params[0]
um33=params[1]
l33=params[2]
print('A=%e'%(A33))
print('um=%e'%(um33))
print('l=%e'%(l33))

#Cálculo datos para determinar extensión de la fase exponencial
tm33=((A33/(np.exp(1)*um33))+l33)
print('Tm=%e'%(tm33))
t233=((np.log(2))/um33)
print('doubpe=%e'%(t233))
extdp33=2*t233
print('ext=%e'%extdp33)
ttot33=tm33+extdp33
print('Tfinal=%e'%ttot33)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
```

```

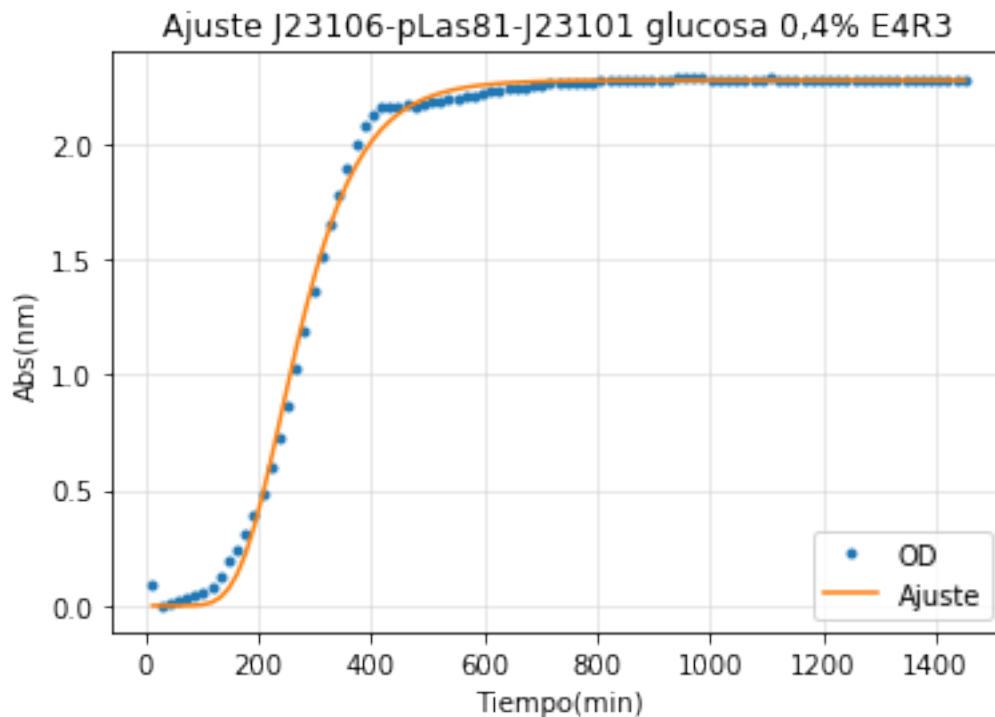
y2=tt[25]
plt.figure()
plt.title('J23106-pLas81-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12273,label='OD J23106-pLas81-J23101 E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12273[16:26],label='OD J23106-pLas81-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.640000e-02

[ 2.27424241e+00 1.09679411e-02 1.64283900e+02]

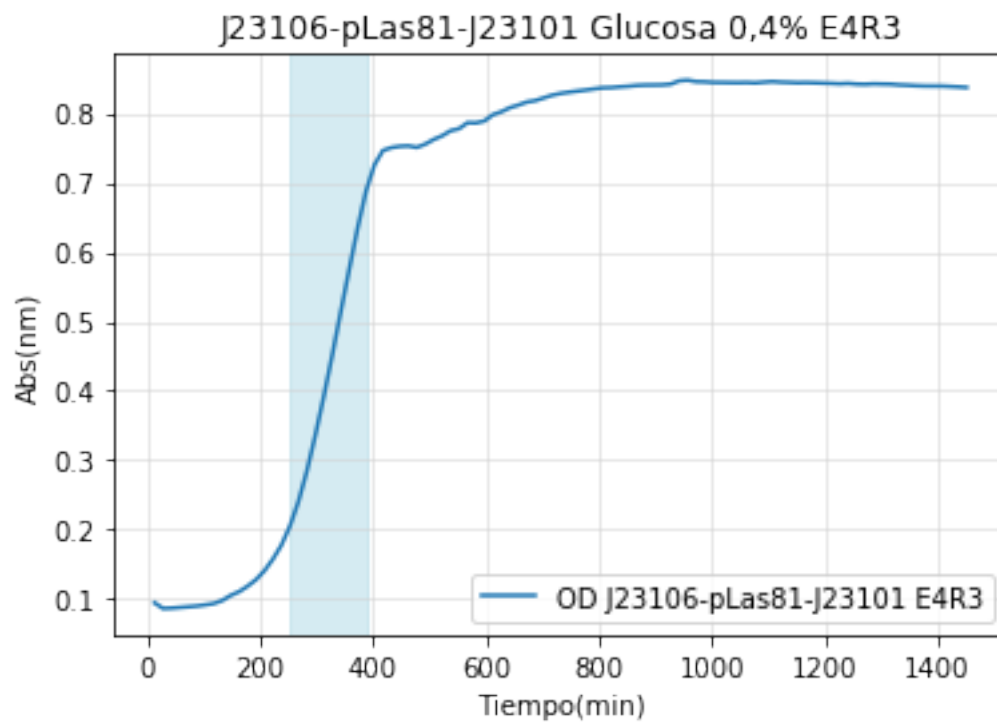


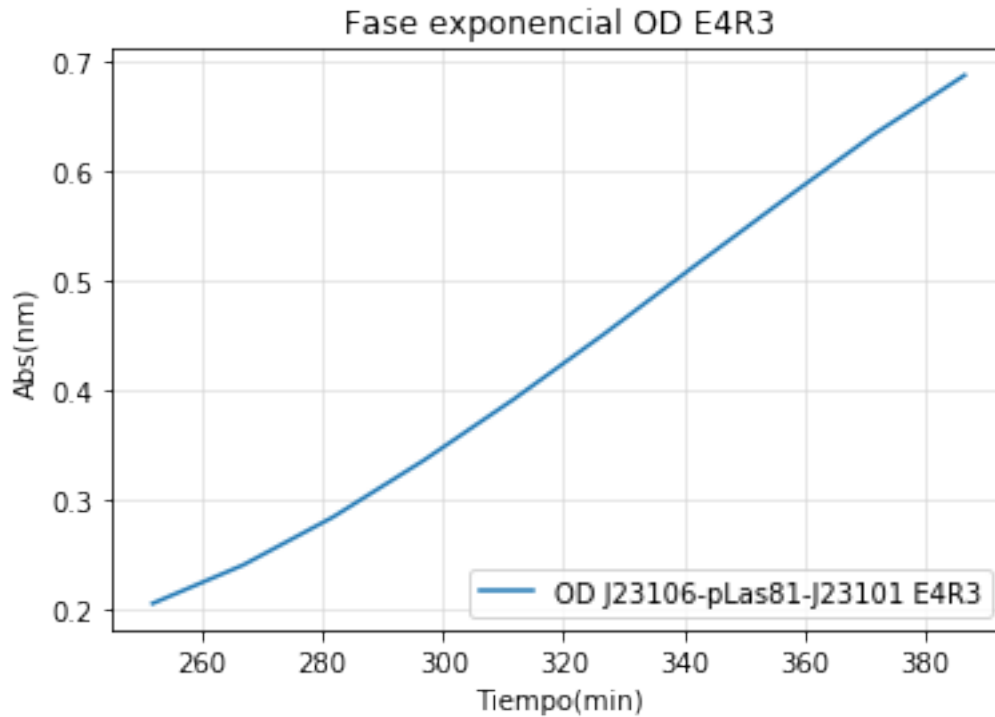
A=2.274242e+00  
um=1.096794e-02



```
l=1.642839e+02  
Tm=2.405650e+02  
doubpe=6.319757e+01  
ext=1.263951e+02  
Tfinal=3.669602e+02
```

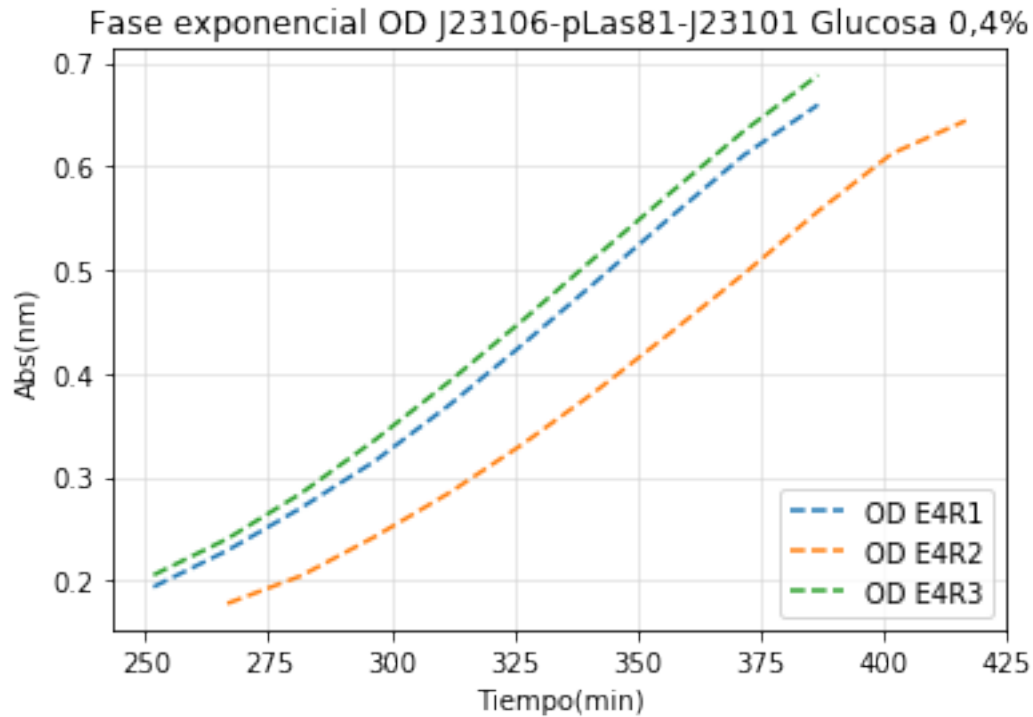
Out[112]: <matplotlib.legend.Legend at 0x178950376d8>





```
In [113]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23106-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12271[16:26], '--',label='OD E4R1')
plt.plot(tt[17:28],od12272[17:28], '--',label='OD E4R2')
plt.plot(tt[16:26],od12273[16:26], '--',label='OD E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[113]: <matplotlib.legend.Legend at 0x17896e93ef0>
```



```
In [114]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 1
y34= np.log(od1227g1)-np.log(np.min(od1227g1))
print('Min OD = %e'%((np.min(od1227g1))))
evaly, params=Function_fit(tt,y34,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0
A34= params[0]
um34=params[1]
l34=params[2]
print('A=%e'%(A34))
print('um=%e'%(um34))
print('l=%e'%(l34))

#Cálculo datos para determinar extensión de la fase exponencial
tm34=((A34/(np.exp(1)*um34))+l34)
print('Tm=%e'%(tm34))
t234=((np.log(2))/um34)
print('doubpe=%e'%(t234))
extdp34=2.5*t234
print('ext=%e'%extdp34)
ttot34=tm34+extdp34
print('Tfinal=%e'%ttot34)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[21]
```

```

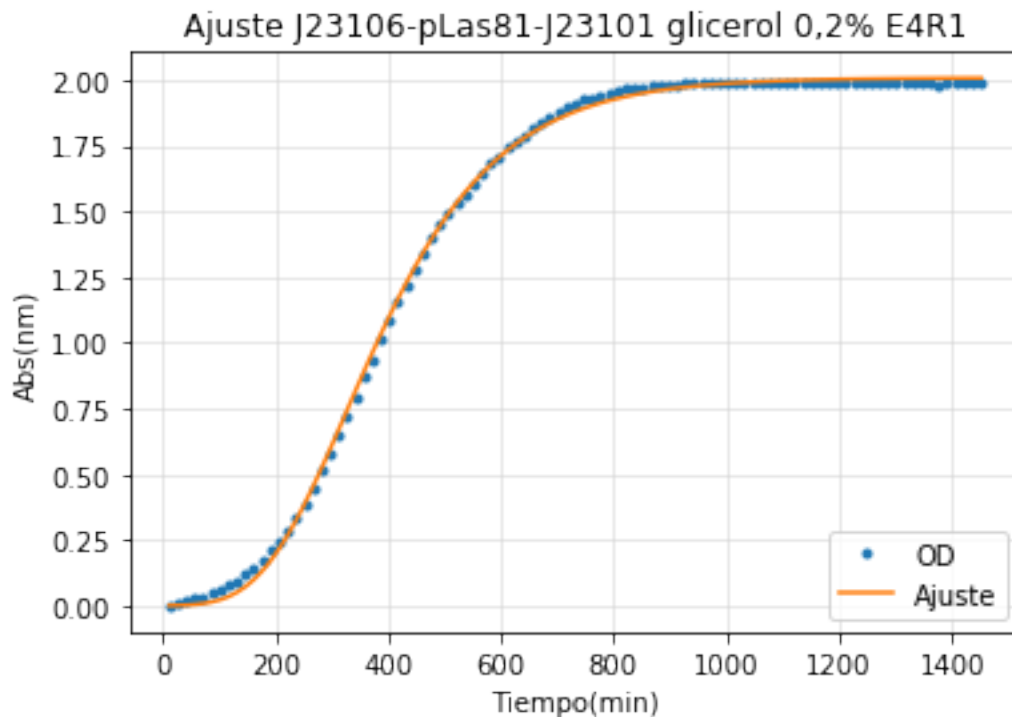
y2=tt[46]
plt.figure()
plt.title('J23106-pLas81-J23101 Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1227g1,label='OD J23106-pLas81-J23101 E4R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[21:47],od1227g1[21:47],label='OD J23106-pLas81-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.410000e-02

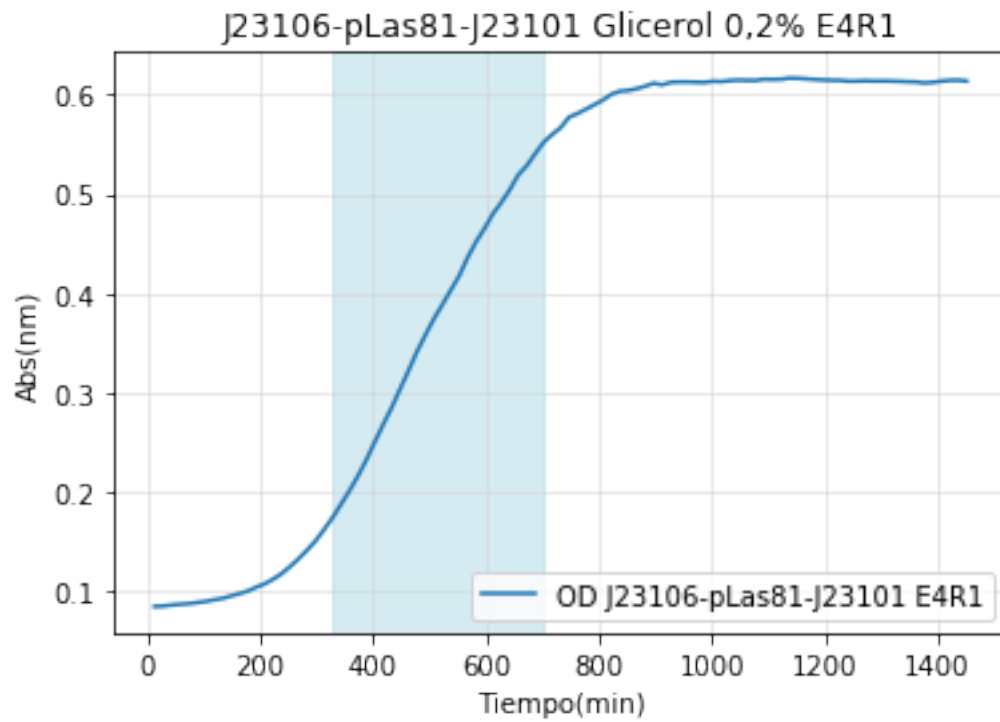
[ 2.00880290e+00 4.95098248e-03 1.76571690e+02]

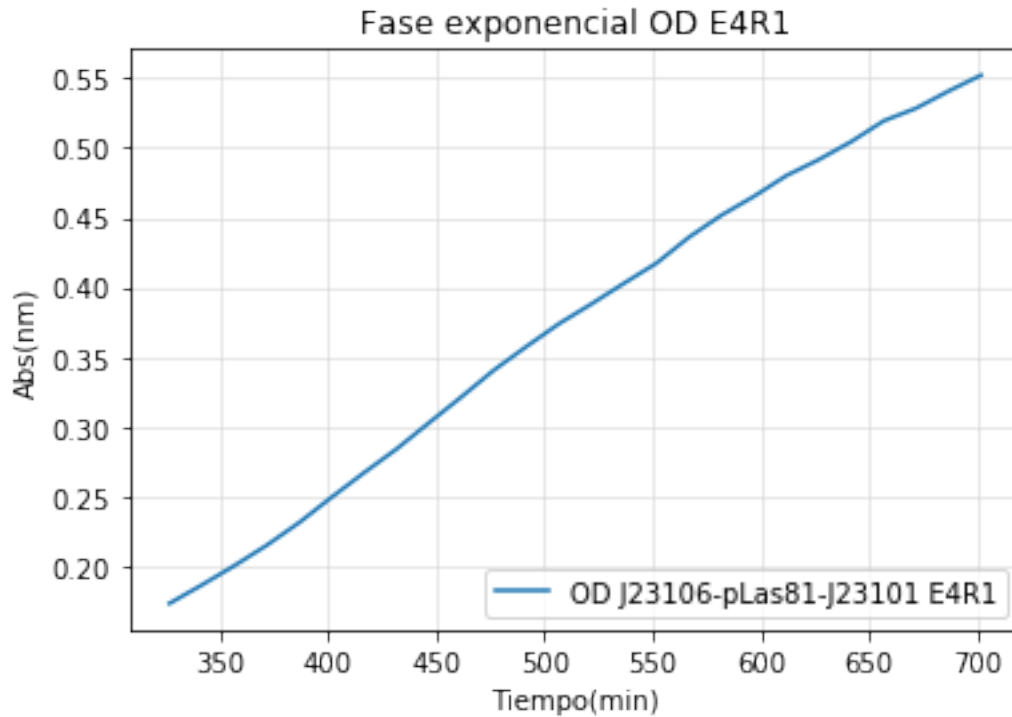


A=2.008803e+00  
um=4.950982e-03

```
l=1.765717e+02  
Tm=3.258344e+02  
doubpe=1.400019e+02  
ext=3.500049e+02  
Tfinal=6.758393e+02
```

Out[114]: <matplotlib.legend.Legend at 0x1788cb928d0>





```
In [115]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 2
y35= np.log(od1227g2)-np.log(np.min(od1227g2))
print('Min OD = %e'%((np.min(od1227g2))))
evaly, params=Function_fit(tt,y35,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0
A35= params[0]
um35=params[1]
l35=params[2]
print('A=%e'%(A35))
print('um=%e'%(um35))
print('l=%e'%(l35))

#Cálculo datos para determinar extensión de la fase exponencial
tm35=((A35/(np.exp(1)*um35))+l35)
print('Tm=%e'%(tm35))
t235=((np.log(2))/um35)
print('doubpe=%e'%(t235))
extdp35=2.5*t235
print('ext=%e'%extdp35)
ttot35=tm35+extdp35
print('Tfinal=%e'%ttot35)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[22]
```

```

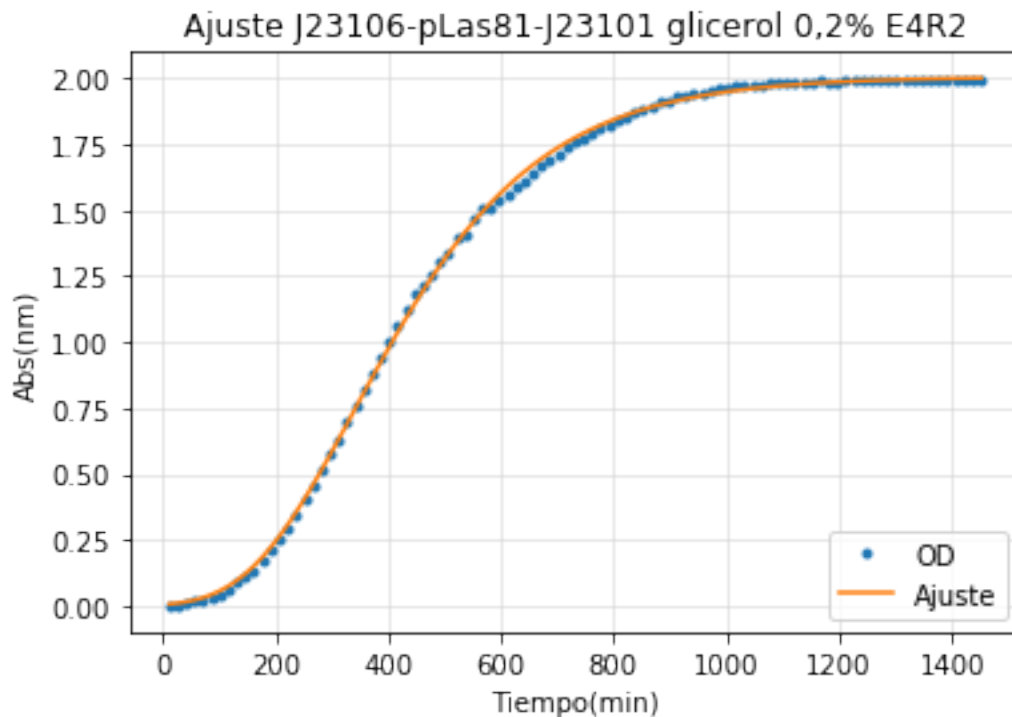
y2=tt[53]
plt.figure()
plt.title('J23106-pLas81-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1227g2,label='OD J23106-pLas81-J23101 E4R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[22:54],od1227g2[22:54],label='OD J23106-pLas81-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.530000e-02

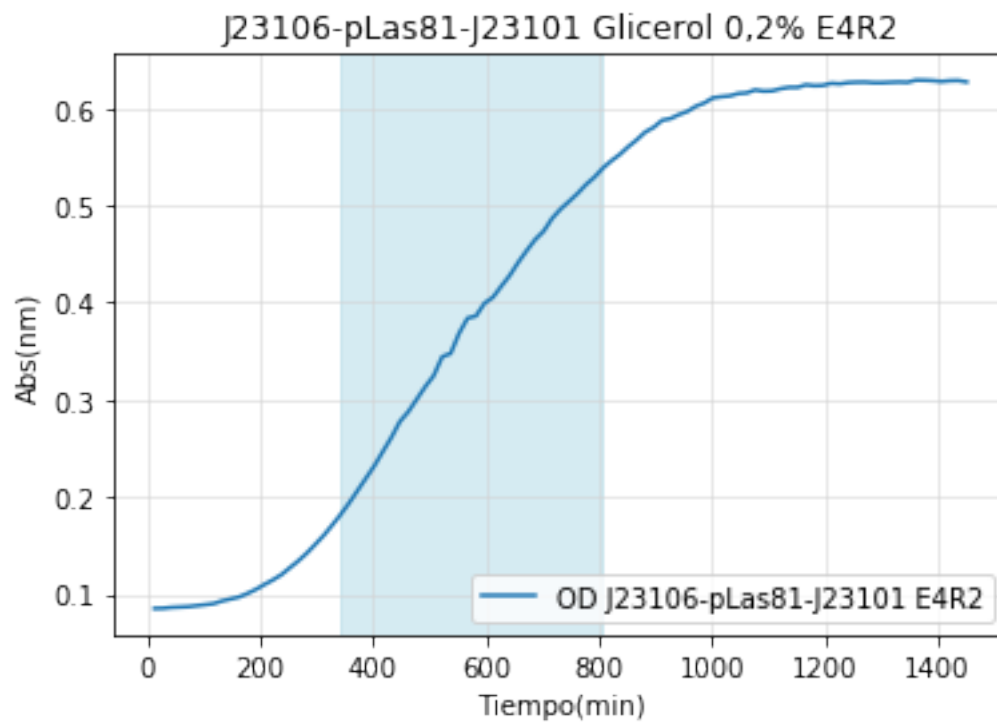
[ 2.00719284e+00 3.95512057e-03 1.52315901e+02]



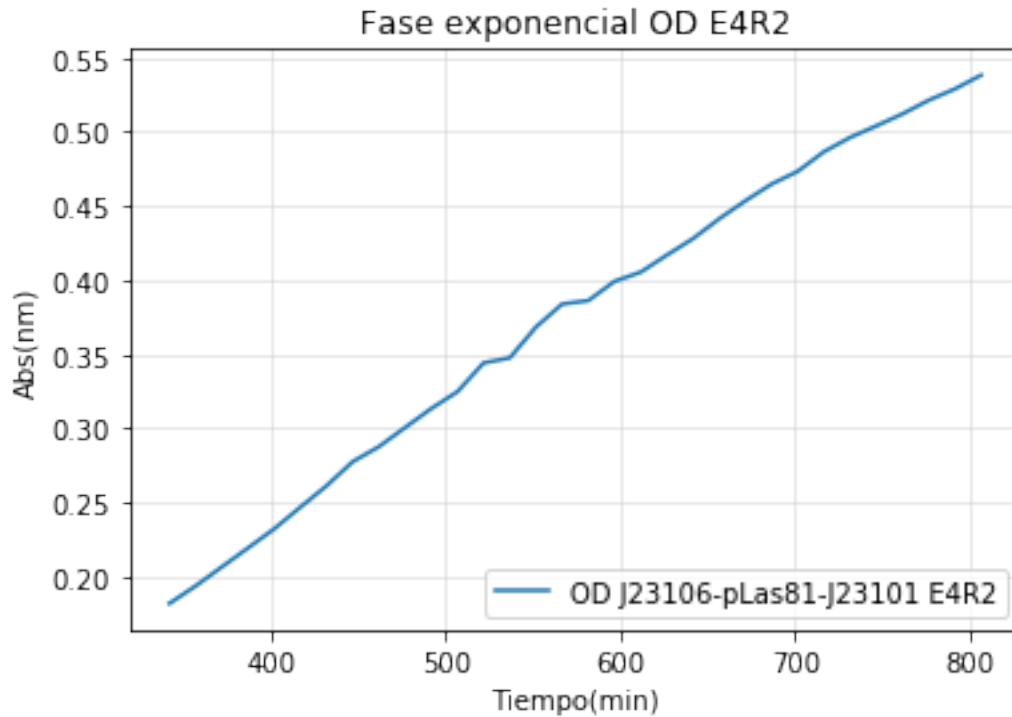
A=2.007193e+00  
um=3.955121e-03

```
l=1.523159e+02  
Tm=3.390118e+02  
doubpe=1.752531e+02  
ext=4.381328e+02  
Tfinal=7.771446e+02
```

Out[115]: <matplotlib.legend.Legend at 0x17898ad38d0>







```
In [116]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 3
y36= np.log(od1227g3)-np.log(np.min(od1227g3))
print('Min OD = %e'%((np.min(od1227g3))))
evaly, params=Function_fit(tt,y36,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0
A36= params[0]
um36=params[1]
l36=params[2]
print('A=%e'%(A36))
print('um=%e'%(um36))
print('l=%e'%(l36))

#Cálculo datos para determinar extensión de la fase exponencial
tm36=((A36/(np.exp(1)*um36))+l36)
print('Tm=%e'%(tm36))
t236=((np.log(2))/um36)
print('doubpe=%e'%(t236))
extdp36=2*t236
print('ext=%e'%extdp36)
ttot36=tm36+extdp36
print('Tfinal=%e'%ttot36)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[23]
```

```

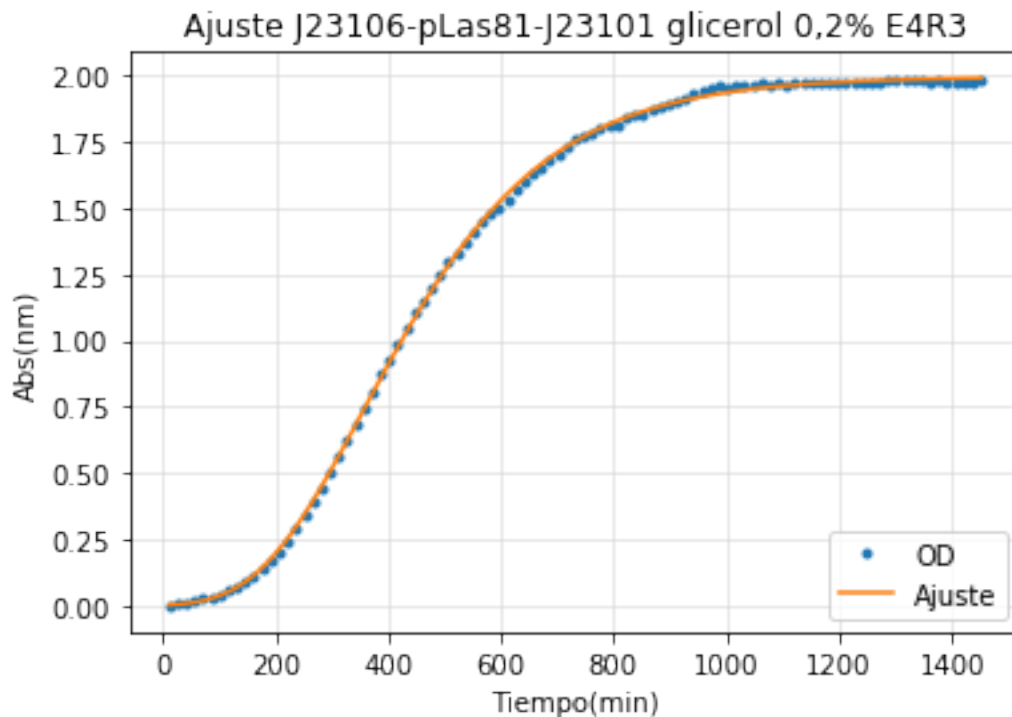
y2=tt[54]
plt.figure()
plt.title('J23106-pLas81-J23101 Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1227g3,label='OD J23106-pLas81-J23101 E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[23:55],od1227g3[23:55],label='OD J23106-pLas81-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

```

Min OD = 8.360000e-02

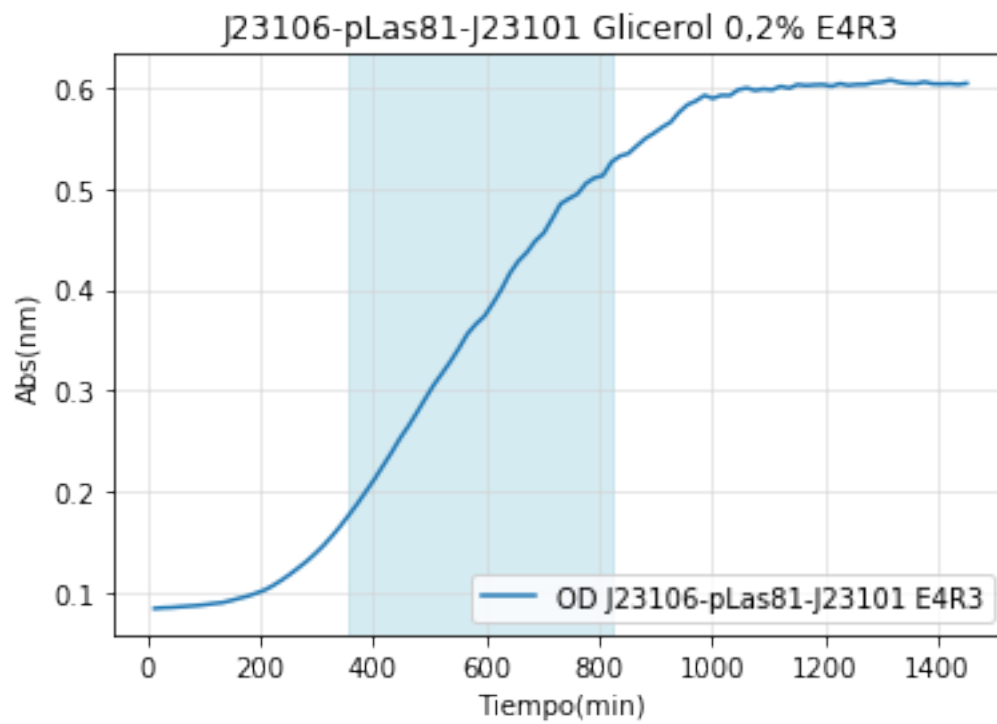
[ 1.99609670e+00 3.97479297e-03 1.70376430e+02]

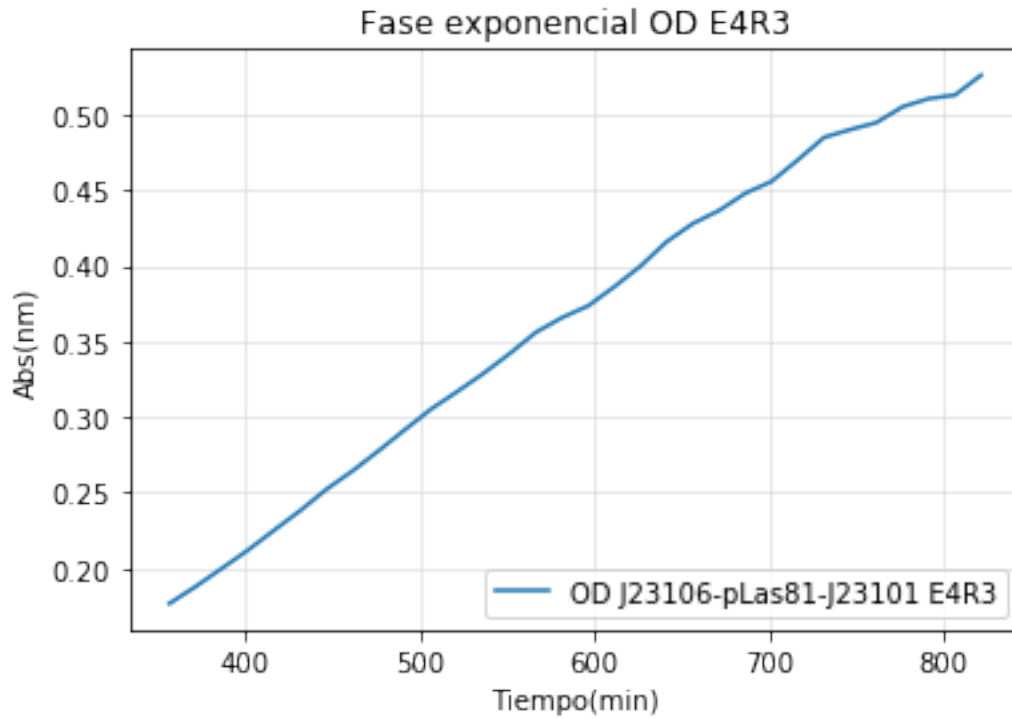


A=1.996097e+00  
um=3.974793e-03

l=1.703764e+02  
Tm=3.551214e+02  
doubpe=1.743857e+02  
ext=3.487715e+02  
Tfinal=7.038928e+02

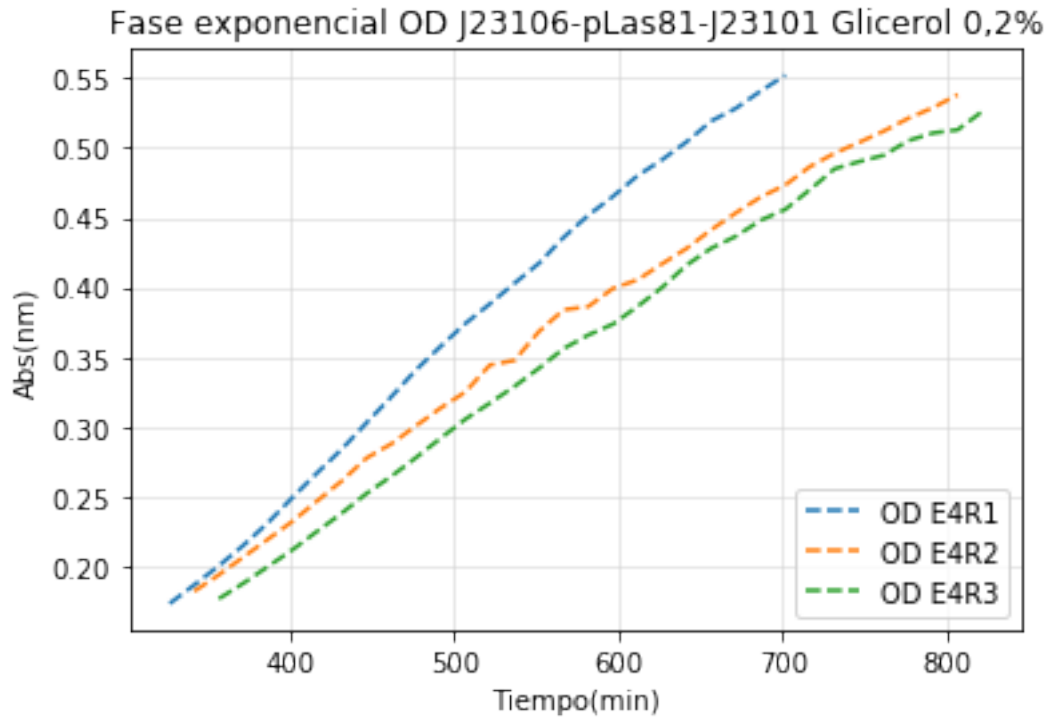
Out[116]: <matplotlib.legend.Legend at 0x17898c91dd8>





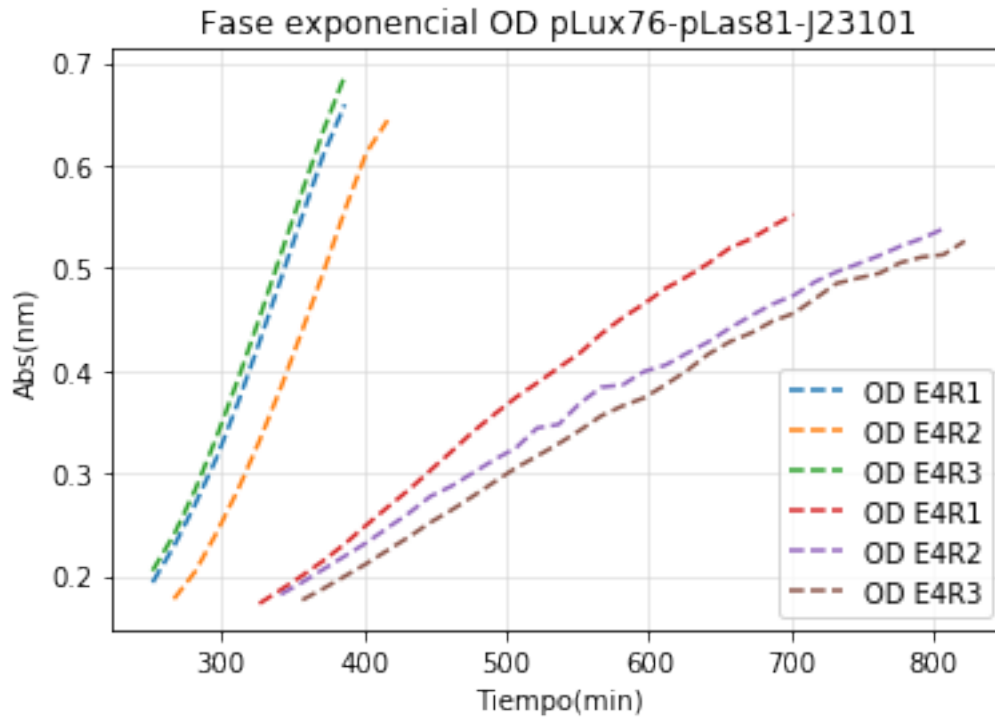
```
In [117]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23106-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[21:47],od1227g1[21:47], '--',label='OD E4R1')
plt.plot(tt[22:54],od1227g2[22:54], '--',label='OD E4R2')
plt.plot(tt[23:55],od1227g3[23:55], '--',label='OD E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[117]: <matplotlib.legend.Legend at 0x17898db1cc0>
```



```
In [119]: #Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12271[16:26], '--',label='OD E4R1')
plt.plot(tt[17:28],od12272[17:28], '--',label='OD E4R2')
plt.plot(tt[16:26],od12273[16:26], '--',label='OD E4R3')
plt.plot(tt[21:47],od1227g1[21:47], '--',label='OD E4R1')
plt.plot(tt[22:54],od1227g2[22:54], '--',label='OD E4R2')
plt.plot(tt[23:55],od1227g3[23:55], '--',label='OD E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Out[119]: <matplotlib.legend.Legend at 0x17898f4bb00>
```



```
In [121]: #Selección de datos en arrays, según lo determinado
          #controles glucosa 15
          o1=odcg151[19:30]
          c1=cfpcg151[19:30]
          r1=rfpcg151[19:30]
          y1=yfpcg151[19:30]

          o2=odcg152[19:30]
          c2=cfpcg152[19:30]
          r2=rfpcg152[19:30]
          y2=yfpcg152[19:30]

          o3=odcg153[15:27]
          c3=cfpcg153[15:27]
          r3=rfpcg153[15:27]
          y3=yfpcg153[15:27]

          #controles glicerol 15
          o4=odcg1151[24:45]
          c4=cfpcg1151[24:45]
          r4=rfpcg1151[24:45]
          y4=yfpcg1151[24:45]

          o5=odcg1152[23:46]
```

```
c5=cfpcgl152[23:46]
r5=rfpcgl152[23:46]
y5=yfpcgl152[23:46]
```

```
o6=odcgl153[25:57]
c6=cfpcgl153[25:57]
r6=rfpcgl153[25:57]
y6=yfpcgl153[25:57]
```

#### *#controles glucosa 18*

```
o7=odcgl181[16:26]
c7=cfpcgl181[16:26]
r7=rfpcgl181[16:26]
y7=yfpcgl181[16:26]
```

```
o8=odcgl182[17:30]
c8=cfpcgl182[17:30]
r8=rfpcgl182[17:30]
y8=yfpcgl182[17:30]
```

```
o9=odcgl183[17:29]
c9=cfpcgl183[17:29]
r9=rfpcgl183[17:29]
y9=yfpcgl183[17:29]
```

#### *#controles glicerol 18*

```
o10=odcgl181[25:48]
c10=cfpcgl181[25:48]
r10=rfpcgl181[25:48]
y10=yfpcgl181[25:48]
```

```
o11=odcgl182[25:53]
c11=cfpcgl182[25:53]
r11=rfpcgl182[25:53]
y11=yfpcgl182[25:53]
```

```
o12=odcgl183[24:51]
c12=cfpcgl183[24:51]
r12=rfpcgl183[24:51]
y12=yfpcgl183[24:51]
```

#### *#controles glucosa 12*

```
o13=odcgl121[17:28]
c13=cfpcgl121[17:28]
r13=rfpcgl121[17:28]
y13=yfpcgl121[17:28]
```

```
o14=odcgl122[14:25]
```

```
c14=cfpcg122[14:25]
r14=rfpcg122[14:25]
y14=yfpcg122[14:25]
```

```
o15=odcg123[17:29]
c15=cfpcg123[17:29]
r15=rfpcg123[17:29]
y15=yfpcg123[17:29]
```

*#controles glicerol 12*

```
o16=odcgl121[24:52]
c16=cfpcgl121[24:52]
r16=rfpcgl121[24:52]
y16=yfpcgl121[24:52]
```

```
o17=odcgl122[25:53]
c17=cfpcgl122[25:53]
r17=rfpcgl122[25:53]
y17=yfpcgl122[25:53]
```

```
o18=odcgl123[25:57]
c18=cfpcgl123[25:57]
r18=rfpcgl123[25:57]
y18=yfpcgl123[25:57]
```

*#ptet-plas-std glucosa*

```
o19=od15271[19:29]
c19=cfp15271[19:29]
r19=rfp15271[19:29]
y19=yfp15271[19:29]
```

```
o20=od15272[16:26]
c20=cfp15272[16:26]
r20=rfp15272[16:26]
y20=yfp15272[16:26]
```

```
o21=od15273[16:27]
c21=cfp15273[16:27]
r21=rfp15273[16:27]
y21=yfp15273[16:27]
```

*#ptet-plas-std glicerol*

```
o22=od1527g1[29:49]
c22=cfp1527g1[29:49]
r22=rfp1527g1[29:49]
y22=yfp1527g1[29:49]
```

```
o23=od1527g2[26:58]
```



c23=cfp1527g2[26:58]  
r23=rfp1527g2[26:58]  
y23=yfp1527g2[26:58]

o24=od1527g3[25:57]  
c24=cfp1527g3[25:57]  
r24=rfp1527g3[25:57]  
y24=yfp1527g3[25:57]

*#pLux-plas-std glucosa*

o25=od18271[17:27]  
c25=cfp18271[17:27]  
r25=rfp18271[17:27]  
y25=yfp18271[17:27]

o26=od18272[17:29]  
c26=cfp18272[17:29]  
r26=rfp18272[17:29]  
y26=yfp18272[17:29]

o27=od18273[17:28]  
c27=cfp18273[17:28]  
r27=rfp18273[17:28]  
y27=yfp18273[17:28]

*#plux-plas-std glycerol*

o28=od1827g1[27:52]  
c28=cfp1827g1[27:52]  
r28=rfp1827g1[27:52]  
y28=yfp1827g1[27:52]

o29=od1827g2[26:58]  
c29=cfp1827g2[26:58]  
r29=rfp1827g2[26:58]  
y29=yfp1827g2[26:58]

o30=od1827g3[25:53]  
c30=cfp1827g3[25:53]  
r30=rfp1827g3[25:53]  
y30=yfp1827g3[25:53]

*#106-plas-std glucosa*

o31=od12271[16:26]  
c31=cfp12271[16:26]  
r31=rfp12271[16:26]  
y31=yfp12271[16:26]

o32=od12272[17:28]

```

c32=cfp12272[17:28]
r32=rfp12272[17:28]
y32=yfp12272[17:28]

o33=od12273[16:26]
c33=cfp12273[16:26]
r33=rfp12273[16:26]
y33=yfp12273[16:26]

#106-plas-std glicerol
o34=od1227g1[21:47]
c34=cfp1227g1[21:47]
r34=rfp1227g1[21:47]
y34=yfp1227g1[21:47]

o35=od1227g2[22:54]
c35=cfp1227g2[22:54]
r35=rfp1227g2[22:54]
y35=yfp1227g2[22:54]

o36=od1227g3[23:55]
c36=cfp1227g3[23:55]
r36=rfp1227g3[23:55]
y36=yfp1227g3[23:55]

```

```

In [122]: #regresion lineal de replicas
          #Controles glucosa 15
          slope, intercept, r_value, p_value, std_err=stats.linregress(o1,c1)
          slopec1=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o1,r1)
          sloper1=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o1,y1)
          slopey1=slope

          slope, intercept, r_value, p_value, std_err=stats.linregress(o2,c2)
          slopec2=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o2,r2)
          sloper2=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o2,y2)
          slopey2=slope

          slope, intercept, r_value, p_value, std_err=stats.linregress(o3,c3)
          slopec3=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o3,r3)
          sloper3=slope
          slope, intercept, r_value, p_value, std_err=stats.linregress(o3,y3)
          slopey3=slope

```

*#Controles glicerol 15*

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o4,c4)
slopec4=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o4,r4)
sloper4=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o4,y4)
slopey4=slope
```

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o5,c5)
slopec5=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o5,r5)
sloper5=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o5,y5)
slopey5=slope
```

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o6,c6)
slopec6=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o6,r6)
sloper6=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o6,y6)
slopey6=slope
```

*#controles glucosa 18*

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o7,c7)
slopec7=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o7,r7)
sloper7=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o7,y7)
slopey7=slope
```

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o8,c8)
slopec8=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o8,r8)
sloper8=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o8,y8)
slopey8=slope
```

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o9,c9)
slopec9=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o9,r9)
sloper9=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o9,y9)
slopey9=slope
```

*#controles glicerol 18*

```
slope, intercept, r_value, p_value, std_err=stats.linregress(o10,c10)
slopec10=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o10,r10)
```

```

sloper10=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o10,y10)
slopey10=slope

slope, intercept, r_value, p_value, std_err=stats.linregress(o11,c11)
slopec11=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o11,r11)
sloper11=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o11,y11)
slopey11=slope

slope, intercept, r_value, p_value, std_err=stats.linregress(o12,c12)
slopec12=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o12,r12)
sloper12=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o12,y12)
slopey12=slope

#controles glucosa 12
slope, intercept, r_value, p_value, std_err=stats.linregress(o13,c13)
slopec13=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o13,r13)
sloper13=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o13,y13)
slopey13=slope

slope, intercept, r_value, p_value, std_err=stats.linregress(o14,c14)
slopec14=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o14,r14)
sloper14=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o14,y14)
slopey14=slope

slope, intercept, r_value, p_value, std_err=stats.linregress(o15,c15)
slopec15=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o15,r15)
sloper15=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o15,y15)
slopey15=slope

#controles glicerol 12
slope, intercept, r_value, p_value, std_err=stats.linregress(o16,c16)
slopec16=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o16,r16)
sloper16=slope
slope, intercept, r_value, p_value, std_err=stats.linregress(o16,y16)
slopey16=slope

```

```

slope, intercept, r_value, p_value, std_err = stats.linregress(o17, c17)
slopec17 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o17, r17)
sloper17 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o17, y17)
slopepy17 = slope

slope, intercept, r_value, p_value, std_err = stats.linregress(o18, c18)
slopec18 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o18, r18)
sloper18 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o18, y18)
slopepy18 = slope

#ptet-plac-std glucosa
slope, intercept, r_value, p_value, std_err = stats.linregress(o19, c19)
slopec19 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o19, r19)
sloper19 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o19, y19)
slopepy19 = slope

slope, intercept, r_value, p_value, std_err = stats.linregress(o20, c20)
slopec20 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o20, r20)
sloper20 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o20, y20)
slopepy20 = slope

slope, intercept, r_value, p_value, std_err = stats.linregress(o21, c21)
slopec21 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o21, r21)
sloper21 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o21, y21)
slopepy21 = slope

#ptet-plac-std glicerol
slope, intercept, r_value, p_value, std_err = stats.linregress(o22, c22)
slopec22 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o22, r22)
sloper22 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o22, y22)
slopepy22 = slope

slope, intercept, r_value, p_value, std_err = stats.linregress(o23, c23)
slopec23 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o23, r23)
sloper23 = slope

```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o23, y23)
slope_y23 = slope
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o24, c24)
slope_c24 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o24, r24)
slope_r24 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o24, y24)
slope_y24 = slope
```

```
#plux-plac-std glucosa
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o25, c25)
slope_c25 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o25, r25)
slope_r25 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o25, y25)
slope_y25 = slope
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o26, c26)
slope_c26 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o26, r26)
slope_r26 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o26, y26)
slope_y26 = slope
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o27, c27)
slope_c27 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o27, r27)
slope_r27 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o27, y27)
slope_y27 = slope
```

```
#plux-plac-std glycerol
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o28, c28)
slope_c28 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o28, r28)
slope_r28 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o28, y28)
slope_y28 = slope
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o29, c29)
slope_c29 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o29, r29)
slope_r29 = slope
slope, intercept, r_value, p_value, std_err = stats.linregress(o29, y29)
slope_y29 = slope
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(o30, c30)
```

```

slopec30=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o30,r30)
sloper30=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o30,y30)
slopepy30=slope

```

#### *#106-plac-std glucosa*

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o31,c31)
slopec31=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o31,r31)
sloper31=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o31,y31)
slopepy31=slope

```

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o32,c32)
slopec32=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o32,r32)
sloper32=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o32,y32)
slopepy32=slope

```

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o33,c33)
slopec33=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o33,r33)
sloper33=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o33,y33)
slopepy33=slope

```

#### *#106-plac-std glicerol*

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o34,c34)
slopec34=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o34,r34)
sloper34=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o34,y34)
slopepy34=slope

```

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o35,c35)
slopec35=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o35,r35)
sloper35=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o35,y35)
slopepy35=slope

```

```

slope, intercept, r_value, p_value,std_err=stats.linregress(o36,c36)
slopec36=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o36,r36)
sloper36=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o36,y36)

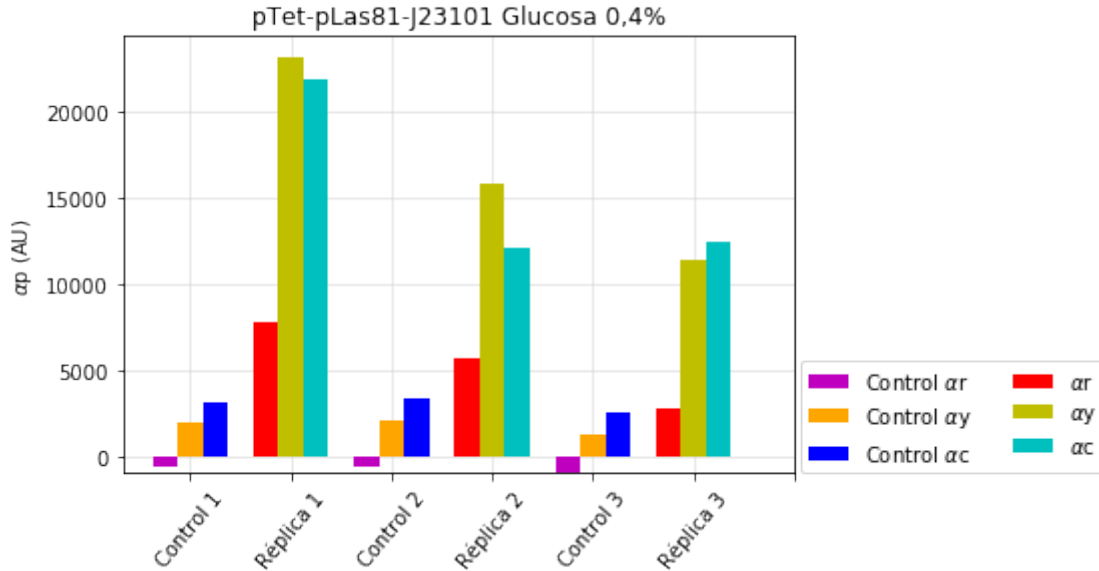
```

```
slopey36=slope
```

```
In [123]: pendientescc=[slopec1,slopec2,slopec3,slopec4,slopec5,slopec6,slopec7,slopec8,slopec9,
pendientescc=[slopec19,slopec20,slopec21,slopec22,slopec23,slopec24,slopec25,slopec26,s
pendientescr=[sloper1,sloper2,sloper3,sloper4,sloper5,sloper6,sloper7,sloper8,sloper9,
pendientesr=[sloper19,sloper20,sloper21,sloper22,sloper23,sloper24,sloper25,sloper26,s
pendientescy=[slopey1,slopey2,slopey3,slopey4,slopey5,slopey6,slopey7,slopey8,slopey9,
pendientesy=[slopey19,slopey20,slopey21,slopey22,slopey23,slopey24,slopey25,slopey26,s
#Grafico pendientes ptet-std-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pTet-pLas81-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+ ' '+ r'$\alpha$r
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+ ' '+ r'$\alpha$y
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+ ' '+ r'$\alpha$c
plt.bar(X[1]-0.25,pendientesr[0],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[0],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientescc[0],color='c',width=0.25,label= r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[1],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[1],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[1],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[2],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[2],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[2],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[123]: <matplotlib.legend.Legend at 0x17898c30438>
```

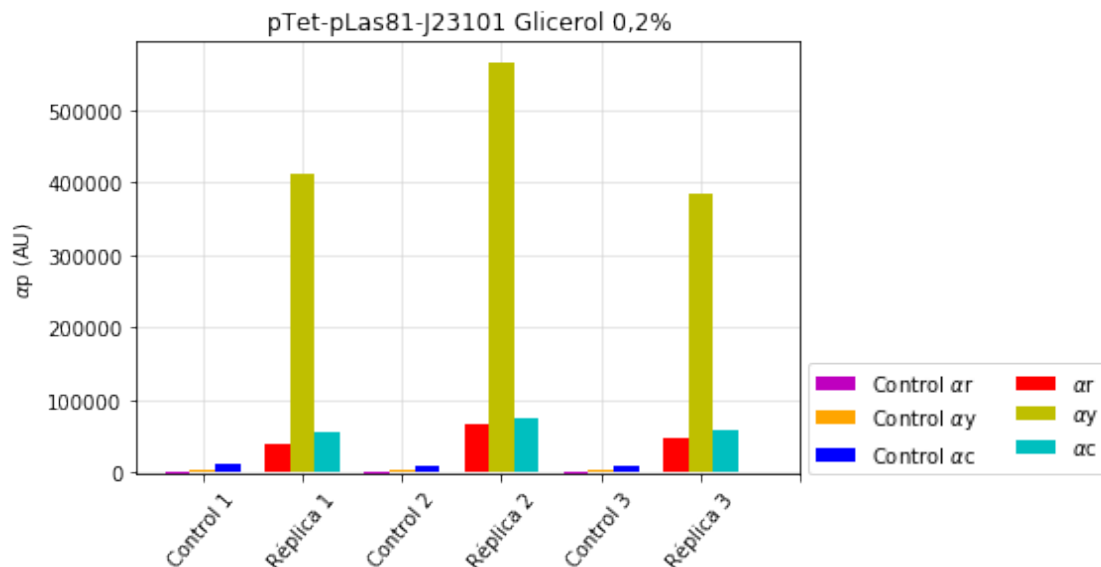




```
In [124]: #Grafico pendientes ptet-std-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pTet-pLas81-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha_p$ (AU)')
plt.bar(X[0]-0.25,pendientescr[3],color='m',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[0]+0.00,pendientescy[3],color='orange',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[0]+0.25,pendientescc[3],color='b',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[1]-0.25,pendientesr[3],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[3],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientescc[3],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[4],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[4],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[4],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[4],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[5],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[5],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[5],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[5],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])

plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[124]: <matplotlib.legend.Legend at 0x178919b6d30>



In [125]: *#Grafico pendientes ptet-std-std*

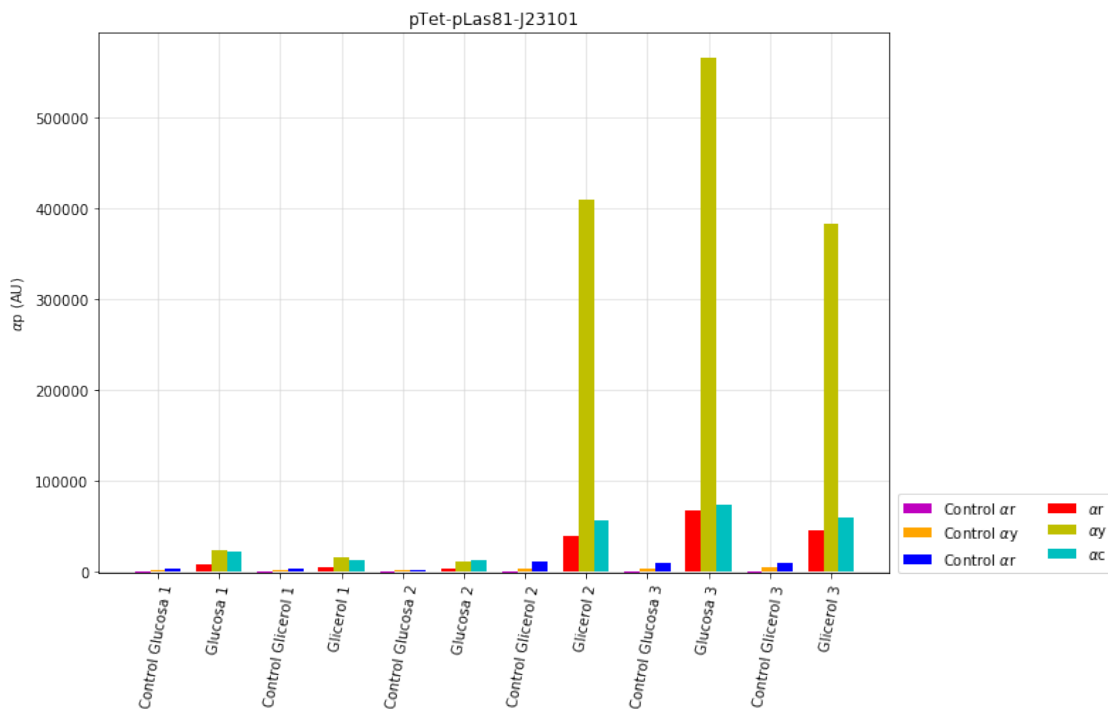
```
X = np.arange(12)
plt.figure(figsize=(10,7))
plt.title('pTet-pLas81-J23101')
plt.ylabel(r'$\alpha$ p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+ ' '+ r'$\alpha$ r')
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+ ' '+ r'$\alpha$ y')
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+ ' '+ r'$\alpha$ c')
plt.bar(X[1]-0.25,pendientesr[0],color='r',width=0.25,label=r'$\alpha$ r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[0],color='y',width=0.25,label= r'$\alpha$ y',zorder=3)
plt.bar(X[1]+0.25,pendientescc[0],color='c',width=0.25,label=r'$\alpha$ c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[1],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[1],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[1],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[2],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[2],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[2],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[3],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientescy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientescc[3],color='b',width=0.25,zorder=3)
```

```

plt.bar(X[7]-0.25,pendientesr[3],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[3],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesec[3],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientescy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientescc[4],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[4],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[4],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesec[4],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientescy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientescc[5],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[5],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[5],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesec[5],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Cont
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[125]: <matplotlib.legend.Legend at 0x178937e9128>



```

In [126]: #Grafico pendientes plux-std-std Glucosa
X = np.arange(7)
plt.figure()

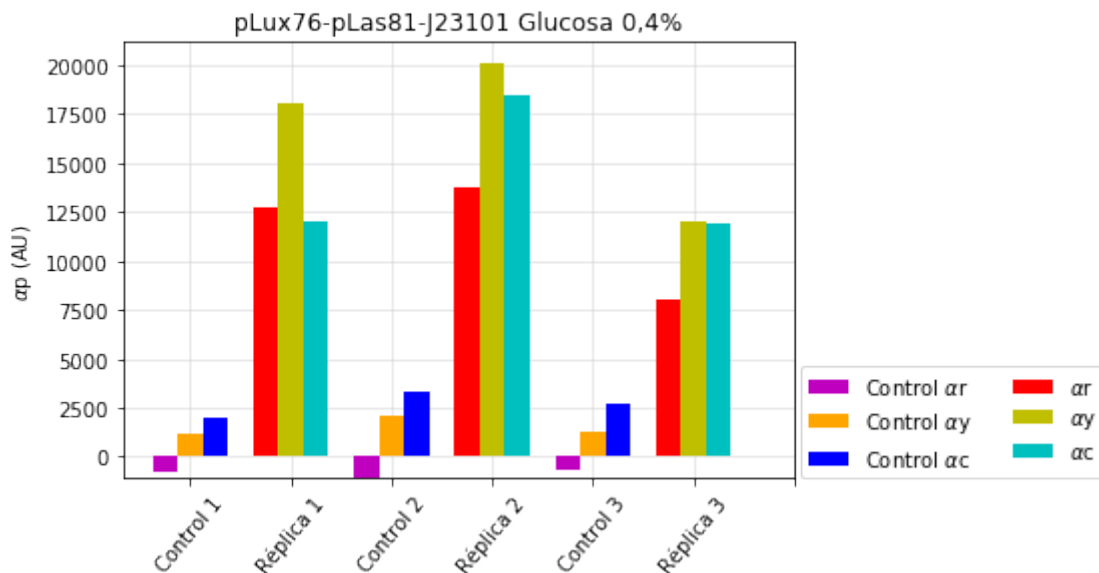
```

```

plt.title('pLux76-pLas81-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha_p$ (AU)')
plt.bar(X[0]-0.25,pendientescr[6],color='m',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.00,pendientescy[6],color='orange',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.25,pendientescc[6],color='b',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[6],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientescc[6],color='c',width=0.25,label= r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[7],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[7],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[7],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[7],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[8],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[8],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[8],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[8],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[126]: <matplotlib.legend.Legend at 0x1788cbbc358>



In [127]: #Grafico pendientes plux-std-std Glicerol  
X = np.arange(7)

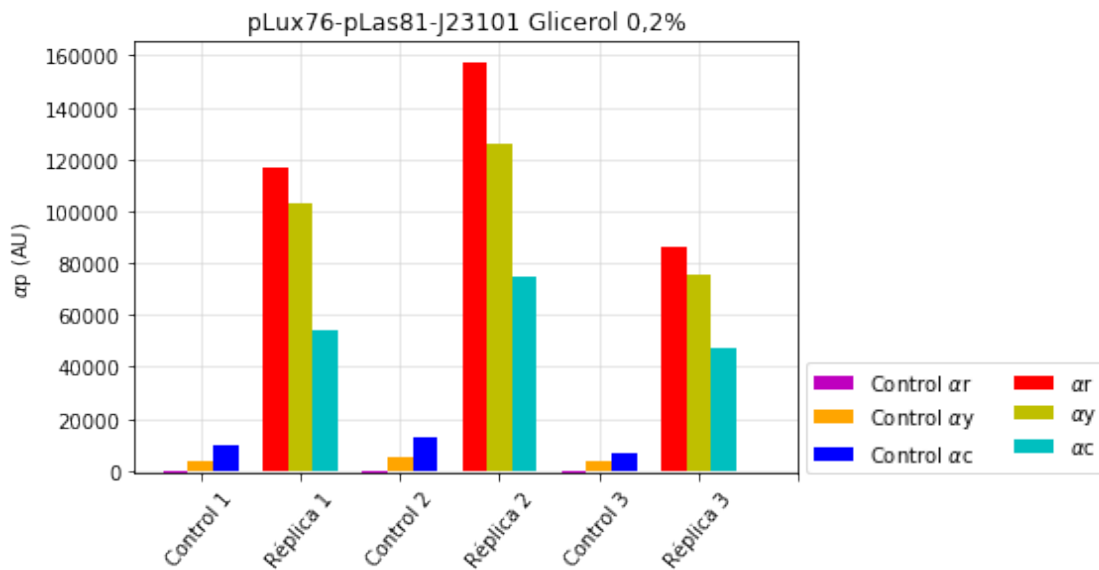
```

plt.figure()
plt.title('pLux76-pLas81-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha$ (AU)')
plt.bar(X[0]-0.25,pendientescr[9],color='m',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.00,pendientescy[9],color='orange',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.25,pendientescc[9],color='b',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[1]-0.25,pendientesr[9],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[9],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[9],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[10],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[10],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[10],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[11],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[11],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[11],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[11],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])

plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[127]: <matplotlib.legend.Legend at 0x1789767bb00>



```

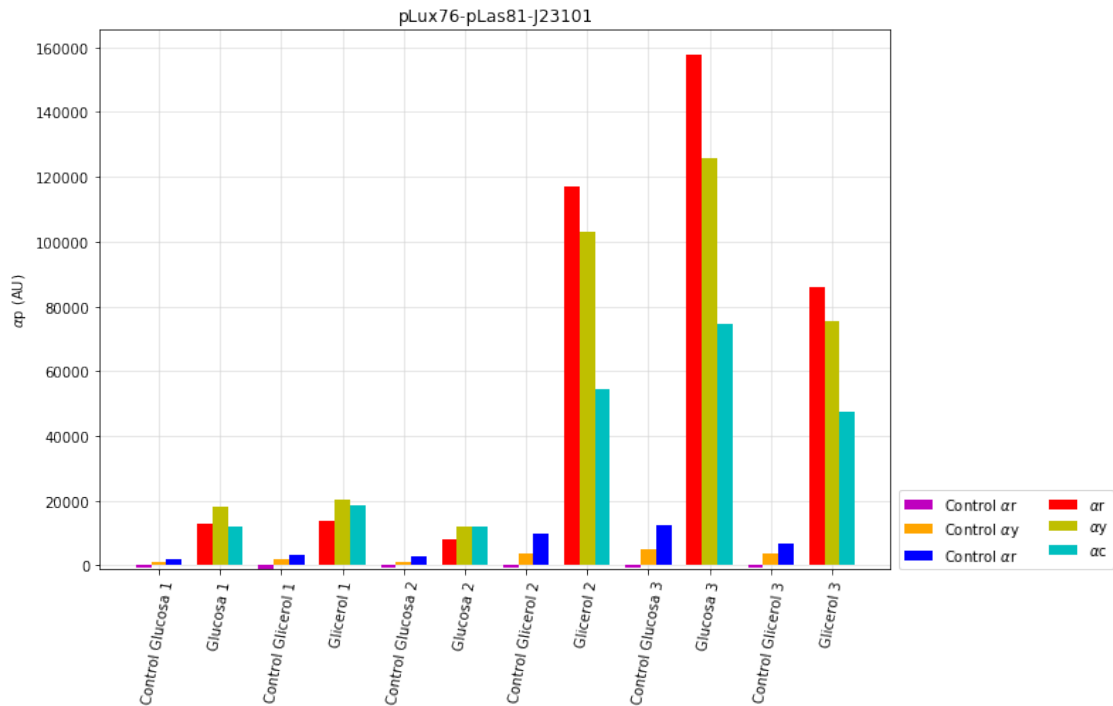
In [128]: #Grafico pendientes plux-std-std
X = np.arange(12)
plt.figure(figsize=(10,7))
plt.title('pLux76-pLas81-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[6],color='m',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[0]+0.00,pendientescy[6],color='orange',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[0]+0.25,pendientescc[6],color='b',width=0.25,label='Control'+ ' '+ r'$\alpha$')
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[6],color='y',width=0.25,label= r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[6],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[7],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[7],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[7],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[8],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[8],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[8],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[8],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[9],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientescy[9],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientescc[9],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[9],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[9],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[9],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientescr[10],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientescy[10],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientescc[10],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientescr[10],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientescy[11],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientescc[11],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[11],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1","Control Glicerol 1","Glicerol 1","Cont
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

```

Out[128]: <matplotlib.legend.Legend at 0x1789944bcf8>

```

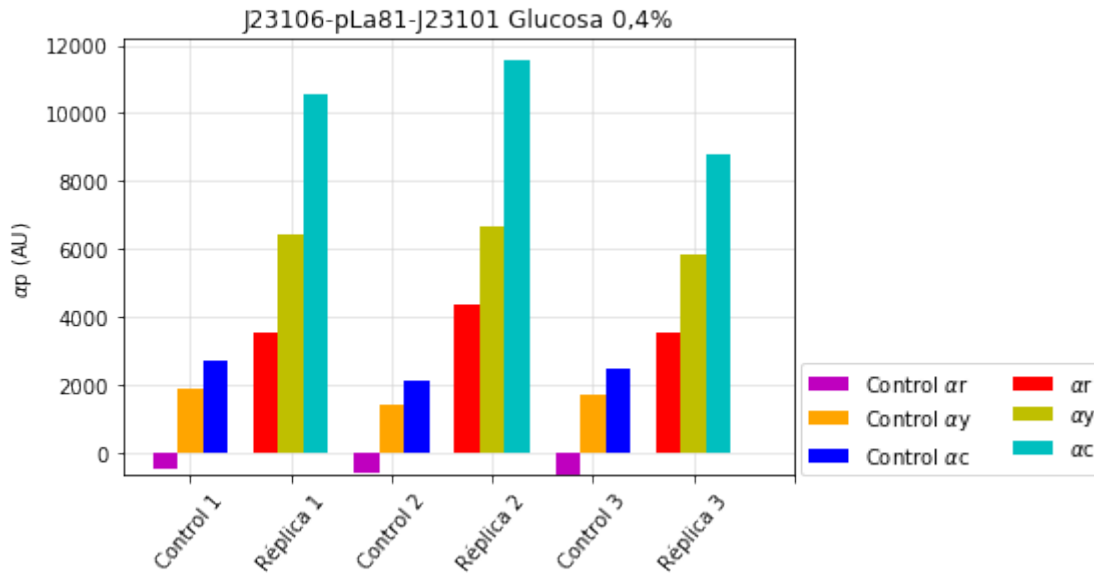


In [129]: *#Grafico pendientes 106-std-std Glucosa*

```
X = np.arange(7)
plt.figure()
plt.title('J23106-pLa81-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha_p$ (AU)')
plt.bar(X[0]-0.25,pendientescr[12],color='m',width=0.25,label='Control'+ ' ' + r'$\alpha_p$')
plt.bar(X[0]+0.00,pendientescy[12],color='orange',width=0.25,label='Control'+ ' ' + r'$\alpha_y$')
plt.bar(X[0]+0.25,pendientescc[12],color='b',width=0.25,label='Control'+ ' ' + r'$\alpha_c$')
plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\alpha_r$')
plt.bar(X[1]+0.00,pendientesy[12],color='y',width=0.25,label=r'$\alpha_y$')
plt.bar(X[1]+0.25,pendientesr[12],color='c',width=0.25,label= r'$\alpha_c$',zorder=3)
plt.bar(X[2]-0.25,pendientescr[13],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[13],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[13],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesr[13],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[14],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[14],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[14],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesr[14],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])
```

```
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[129]: <matplotlib.legend.Legend at 0x178998b1c50>



In [130]: #Grafico pendientes 106-std-std Glicerol

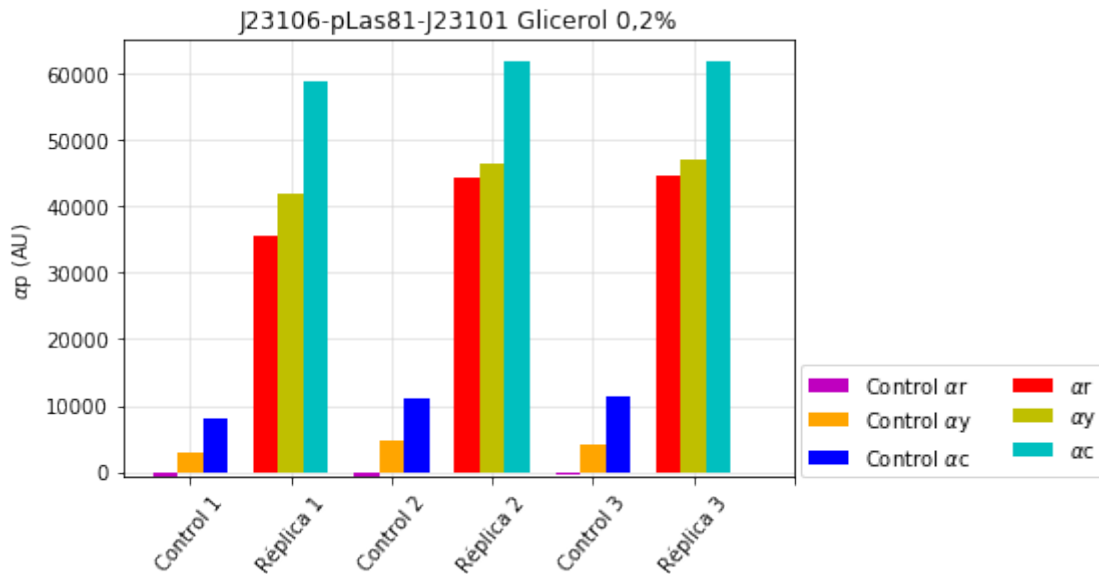
```
X = np.arange(7)
plt.figure()
plt.title('J23106-pLas81-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha_p$ (AU)')
plt.bar(X[0]-0.25,pendientescr[15],color='m',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.00,pendientescy[15],color='orange',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[0]+0.25,pendientescc[15],color='b',width=0.25,label='Control'+ ' ' + r'$\alpha$')
plt.bar(X[1]-0.25,pendientesr[15],color='r',width=0.25,label=r'$\alpha_r$',zorder=3)
plt.bar(X[1]+0.00,pendientesy[15],color='y',width=0.25,label=r'$\alpha_y$',zorder=3)
plt.bar(X[1]+0.25,pendientescc[15],color='c',width=0.25,label=r'$\alpha_c$',zorder=3)
plt.bar(X[2]-0.25,pendientescr[16],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[16],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[16],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[16],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[17],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[17],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[17],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[17],color='c',width=0.25,zorder=3)
```



```
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"])

plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[130]: <matplotlib.legend.Legend at 0x178999f54e0>



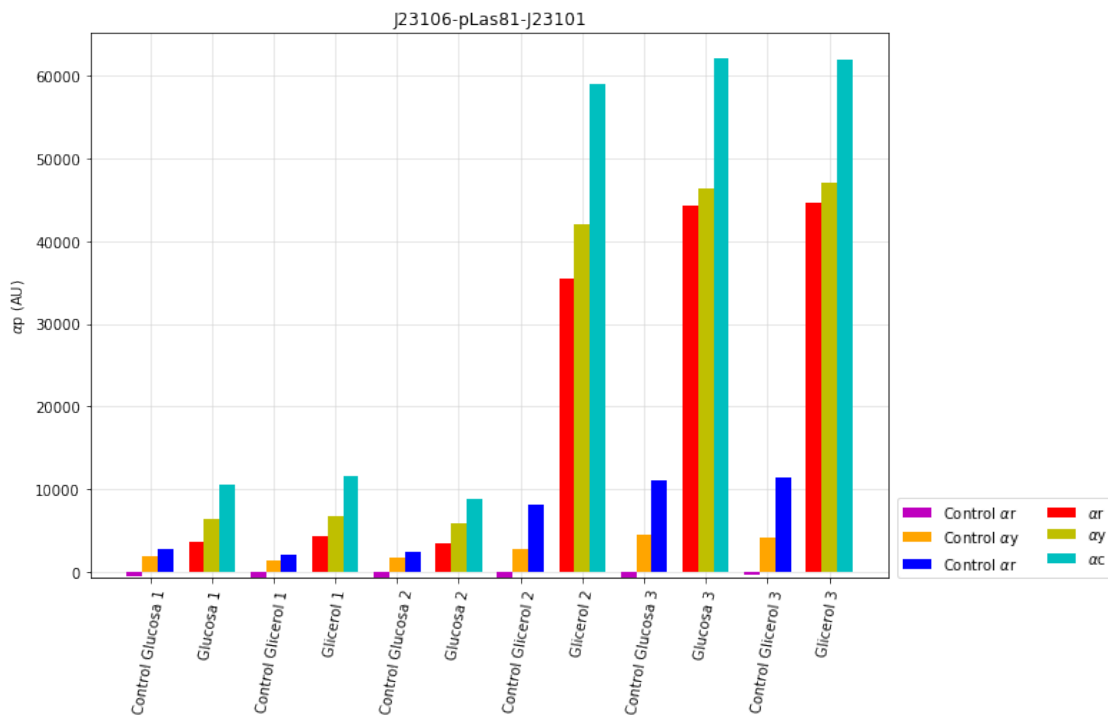
```
In [131]: #Grafico pendientes plux-std-std
X = np.arange(12)
plt.figure(figsize=(10,7))
plt.title('J23106-pLas81-J23101')
plt.ylabel(r'$\alpha_p$ (AU)')
plt.bar(X[0]-0.25,pendientescr[12],color='m',width=0.25,label='Control'+ ' ' + r'$\alpha_r$')
plt.bar(X[0]+0.00,pendientescy[12],color='orange',width=0.25,label='Control'+ ' ' + r'$\alpha_y$')
plt.bar(X[0]+0.25,pendientescc[12],color='b',width=0.25,label='Control'+ ' ' + r'$\alpha_c$')
plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\alpha_r$',zorder=3)
plt.bar(X[1]+0.00,pendientesy[12],color='y',width=0.25,label= r'$\alpha_y$',zorder=3)
plt.bar(X[1]+0.25,pendientesr[12],color='c',width=0.25,label=r'$\alpha_c$',zorder=3)
plt.bar(X[2]-0.25,pendientescr[13],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[13],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[13],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesr[13],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[14],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[14],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[14],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
```

```

plt.bar(X[5]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[15],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientescy[15],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientescc[15],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[15],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientescr[16],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientescy[16],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientescc[16],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientescr[17],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientescy[17],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientescc[17],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Cont
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[131]: <matplotlib.legend.Legend at 0x1789ab80e10>



In [132]: *#Grafico pendientes todo*

```
X = np.arange(36)
plt.figure(figsize=(20,10))
plt.title(r'$\alpha$ p', fontsize=15.0)
plt.ylabel(r'$\alpha$ p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+ ' '+ r'$\alpha$ p')
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+ ' '+ r'$\alpha$ p')
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+ ' '+ r'$\alpha$ p')
plt.bar(X[1]-0.25,pendientescr[3],color='m',width=0.25,zorder=3)
plt.bar(X[1]+0.00,pendientescy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[1]+0.25,pendientescc[3],color='b',width=0.25,zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientescr[4],color='m',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientescy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[4],color='b',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientescr[5],color='m',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientescy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[5],color='b',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[6],color='m',width=0.25,label=r'$\alpha$ r',zorder=3)
plt.bar(X[6]+0.00,pendientescy[6],color='orange',width=0.25,label=r'$\alpha$ y',zorder=3)
plt.bar(X[6]+0.25,pendientescc[6],color='blue',width=0.25,label=r'$\alpha$ c',zorder=3)
plt.bar(X[7]-0.25,pendientescr[9],color='m',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientescy[9],color='orange',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientescc[9],color='b',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientescr[7],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientescy[7],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientescc[7],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientescr[10],color='m',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientescy[10],color='orange',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientescc[10],color='b',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientescr[8],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientescy[8],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientescc[8],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientescr[11],color='m',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientescy[11],color='orange',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientescc[11],color='b',width=0.25,zorder=3)
plt.bar(X[12]-0.25,pendientescr[12],color='m',width=0.25,zorder=3)
plt.bar(X[12]+0.00,pendientescy[12],color='orange',width=0.25,zorder=3)
plt.bar(X[12]+0.25,pendientescc[12],color='b',width=0.25,zorder=3)
plt.bar(X[13]-0.25,pendientescr[15],color='m',width=0.25,zorder=3)
plt.bar(X[13]+0.00,pendientescy[15],color='orange',width=0.25,zorder=3)
plt.bar(X[13]+0.25,pendientescc[15],color='b',width=0.25,zorder=3)
plt.bar(X[14]-0.25,pendientescr[13],color='m',width=0.25,zorder=3)
```

```

plt.bar(X[14]+0.00,pendientescy[13],color='orange',width=0.25,zorder=3)
plt.bar(X[14]+0.25,pendientescc[13],color='b',width=0.25,zorder=3)
plt.bar(X[15]-0.25,pendientescr[16],color='m',width=0.25,zorder=3)
plt.bar(X[15]+0.00,pendientescy[16],color='orange',width=0.25,zorder=3)
plt.bar(X[15]+0.25,pendientescc[16],color='b',width=0.25,zorder=3)
plt.bar(X[16]-0.25,pendientescr[14],color='m',width=0.25,zorder=3)
plt.bar(X[16]+0.00,pendientescy[14],color='orange',width=0.25,zorder=3)
plt.bar(X[16]+0.25,pendientescc[14],color='b',width=0.25,zorder=3)
plt.bar(X[17]-0.25,pendientescr[17],color='m',width=0.25,zorder=3)
plt.bar(X[17]+0.00,pendientescy[17],color='orange',width=0.25,zorder=3)
plt.bar(X[17]+0.25,pendientescc[17],color='b',width=0.25,zorder=3)
plt.bar(X[18]-0.25,pendientesr[0],color='r',width=0.25,zorder=3)
plt.bar(X[18]+0.00,pendientesy[0],color='y',width=0.25,zorder=3)
plt.bar(X[18]+0.25,pendientesr[0],color='c',width=0.25,zorder=3)
plt.bar(X[19]-0.25,pendientesr[3],color='r',width=0.25,zorder=3)
plt.bar(X[19]+0.00,pendientesy[3],color='y',width=0.25,zorder=3)
plt.bar(X[19]+0.25,pendientesr[3],color='c',width=0.25,zorder=3)
plt.bar(X[20]-0.25,pendientesr[1],color='r',width=0.25,zorder=3)
plt.bar(X[20]+0.00,pendientesy[1],color='y',width=0.25,zorder=3)
plt.bar(X[20]+0.25,pendientesr[1],color='c',width=0.25,zorder=3)
plt.bar(X[21]-0.25,pendientesr[4],color='r',width=0.25,zorder=3)
plt.bar(X[21]+0.00,pendientesy[4],color='y',width=0.25,zorder=3)
plt.bar(X[21]+0.25,pendientesr[4],color='c',width=0.25,zorder=3)
plt.bar(X[22]-0.25,pendientesr[2],color='r',width=0.25,zorder=3)
plt.bar(X[22]+0.00,pendientesy[2],color='y',width=0.25,zorder=3)
plt.bar(X[22]+0.25,pendientesr[2],color='c',width=0.25,zorder=3)
plt.bar(X[23]-0.25,pendientesr[5],color='r',width=0.25,zorder=3)
plt.bar(X[23]+0.00,pendientesy[5],color='y',width=0.25,zorder=3)
plt.bar(X[23]+0.25,pendientesr[5],color='c',width=0.25,zorder=3)
plt.bar(X[24]-0.25,pendientesr[6],color='r',width=0.25,zorder=3)
plt.bar(X[24]+0.00,pendientesy[6],color='y',width=0.25,zorder=3)
plt.bar(X[24]+0.25,pendientesr[6],color='c',width=0.25,zorder=3)
plt.bar(X[25]-0.25,pendientesr[9],color='r',width=0.25,zorder=3)
plt.bar(X[25]+0.00,pendientesy[9],color='y',width=0.25,zorder=3)
plt.bar(X[25]+0.25,pendientesr[9],color='c',width=0.25,zorder=3)
plt.bar(X[26]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[26]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[26]+0.25,pendientesr[7],color='c',width=0.25,zorder=3)
plt.bar(X[27]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[27]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[27]+0.25,pendientesr[10],color='c',width=0.25,zorder=3)
plt.bar(X[28]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[28]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[28]+0.25,pendientesr[8],color='c',width=0.25,zorder=3)
plt.bar(X[29]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
plt.bar(X[29]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[29]+0.25,pendientesr[11],color='c',width=0.25,zorder=3)
plt.bar(X[30]-0.25,pendientesr[12],color='r',width=0.25,zorder=3)

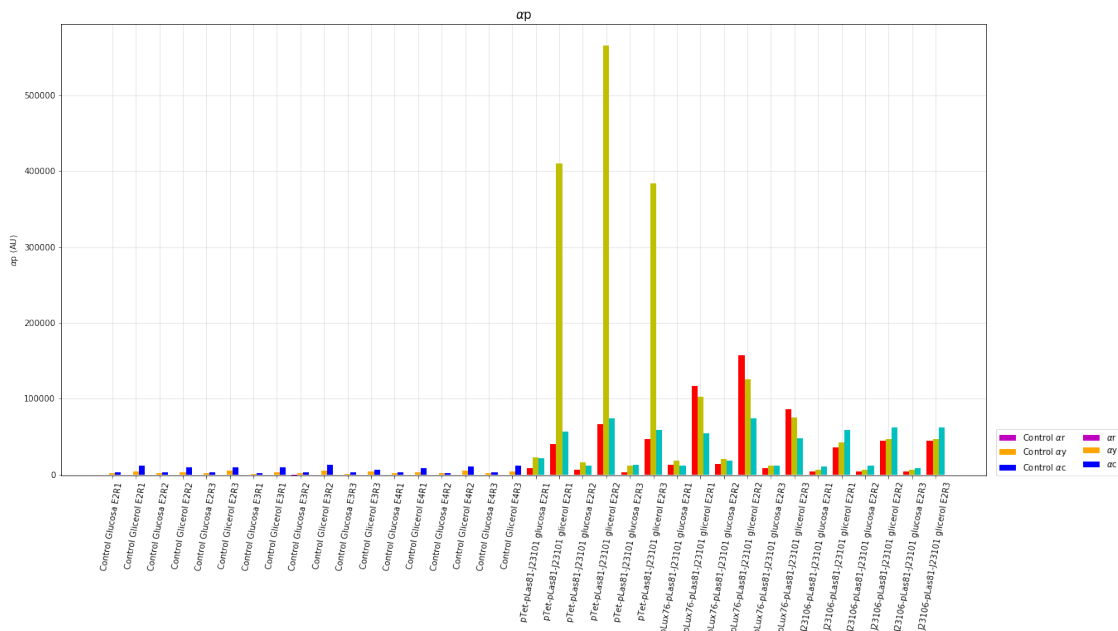
```

```

plt.bar(X[30]+0.00,pendientesy[12],color='y',width=0.25,zorder=3)
plt.bar(X[30]+0.25,pendientesr[12],color='c',width=0.25,zorder=3)
plt.bar(X[31]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
plt.bar(X[31]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
plt.bar(X[31]+0.25,pendientesr[15],color='c',width=0.25,zorder=3)
plt.bar(X[32]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[32]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[32]+0.25,pendientesr[13],color='c',width=0.25,zorder=3)
plt.bar(X[33]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[33]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[33]+0.25,pendientesr[16],color='c',width=0.25,zorder=3)
plt.bar(X[34]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[34]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[34]+0.25,pendientesr[14],color='c',width=0.25,zorder=3)
plt.bar(X[35]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
plt.bar(X[35]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
plt.bar(X[35]+0.25,pendientesr[17],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa E2R1','Control Glicerol E2R1','Control Glucosa E2R2','Control Glicerol E2R2','Control Glucosa E2R3','Control Glicerol E2R3'],
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[132]: <matplotlib.legend.Legend at 0x178976c0908>



```

In [133]: cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15]
cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec18]
rglu=[[sloper1,sloper2,sloper3],[sloper7,sloper8,sloper9],[sloper13,sloper14,sloper15]

```

```

rgli=[[sloper4,sloper5,sloper6],[sloper10,sloper11,sloper12],[sloper16,sloper17,sloper
yglu=[[slopey1,slopey2,slopey3],[slopey7,slopey8,slopey9],[slopey13,slopey14,slopey15]
ygli=[[slopey4,slopey5,slopey6],[slopey10,slopey11,slopey12],[slopey16,slopey17,slopey

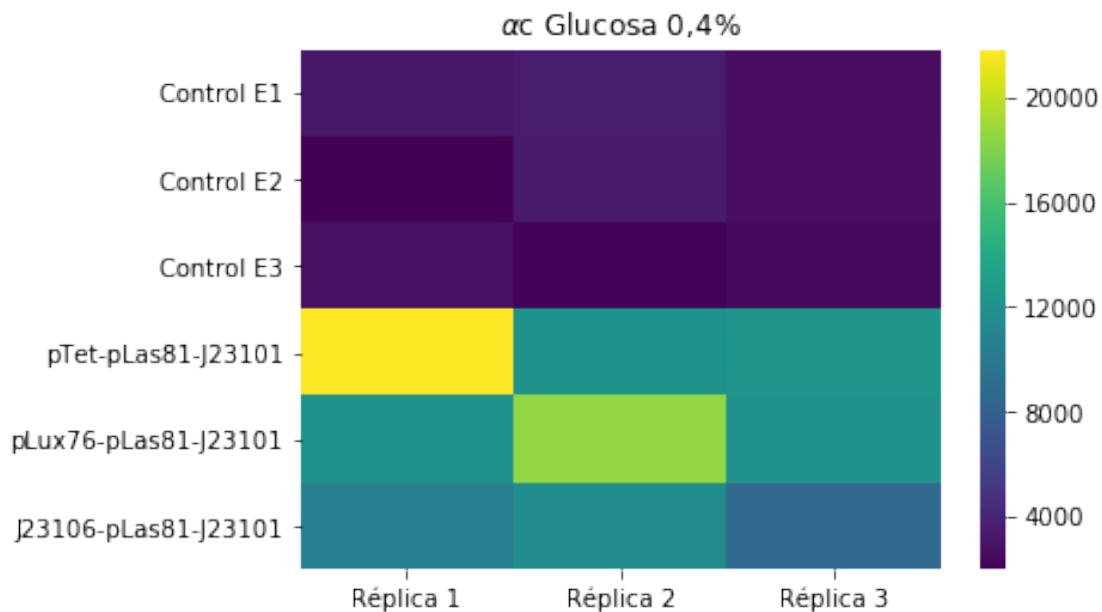
In [134]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
          ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23

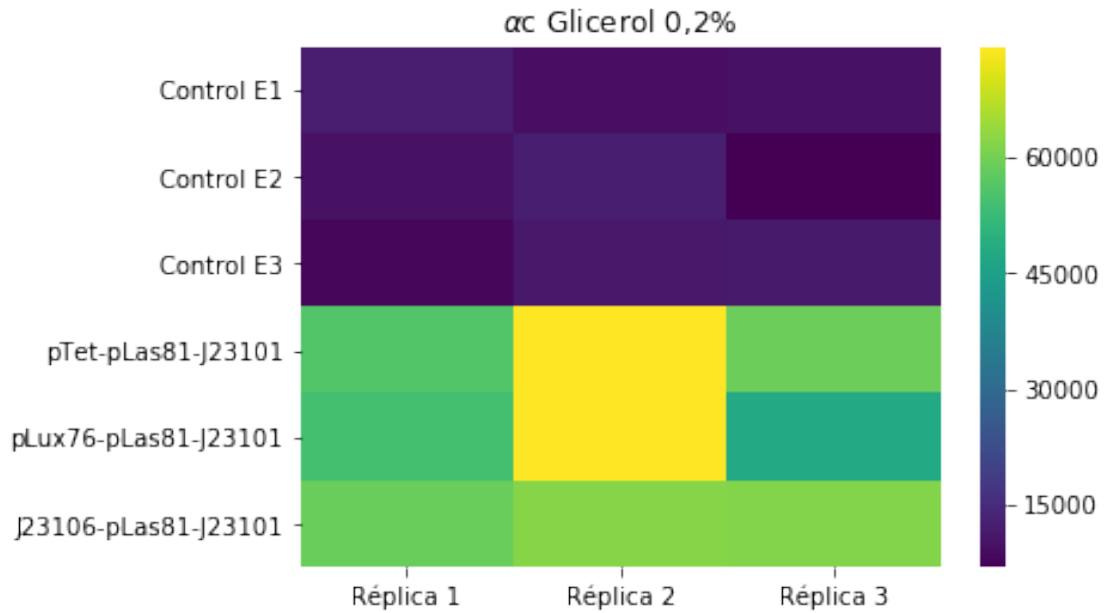
plt.figure()
plt.title(r'$\alpha$ Glucosa 0,4%')
sns.heatmap(cglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

plt.figure()
plt.title(r'$\alpha$ Glicerol 0,2%')
sns.heatmap(cgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

Out[134]: <matplotlib.axes._subplots.AxesSubplot at 0x178976d0978>

```



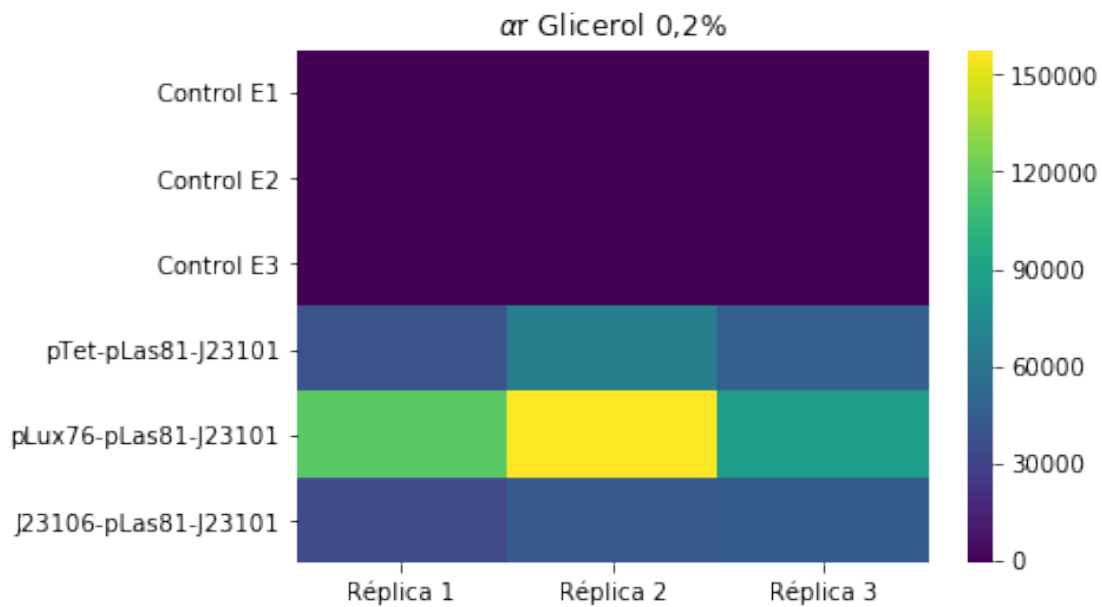
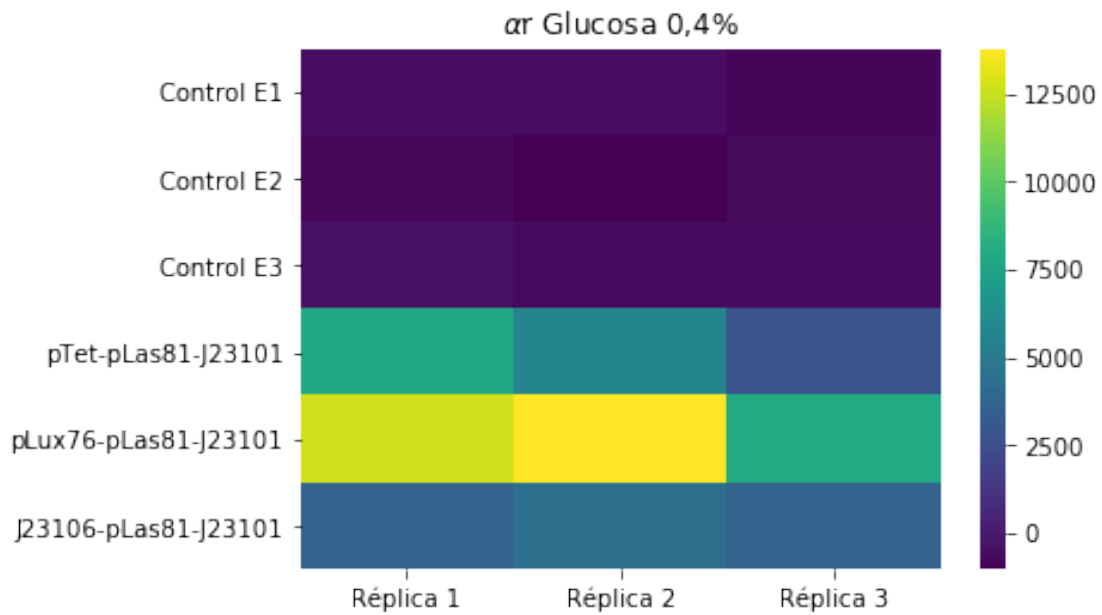


```
In [135]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
          ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101','J23106-pLas81-J23101']

          plt.figure()
          plt.title(r'$\alpha$ Glucosa 0,4%')
          sns.heatmap(rglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

          plt.figure()
          plt.title(r'$\alpha$ Glicerol 0,2%')
          sns.heatmap(rgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

Out[135]: <matplotlib.axes._subplots.AxesSubplot at 0x17899561c50>
```



```
In [136]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
          ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101','J23106-pLas81-J23101']

          plt.figure()
          plt.title(r'$\alpha$ Glucosa 0,4%')
```



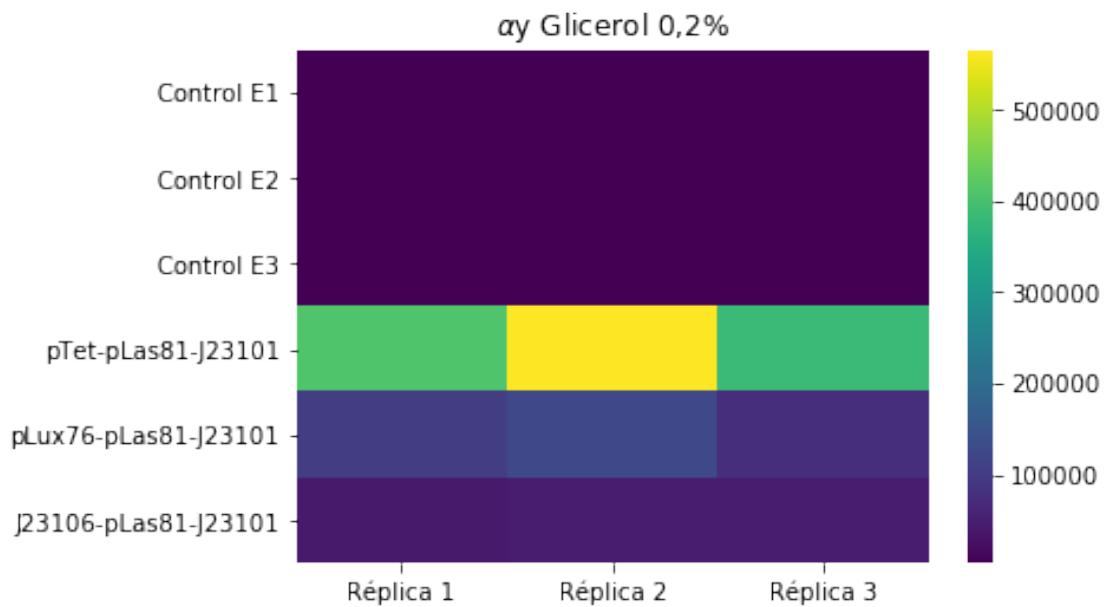
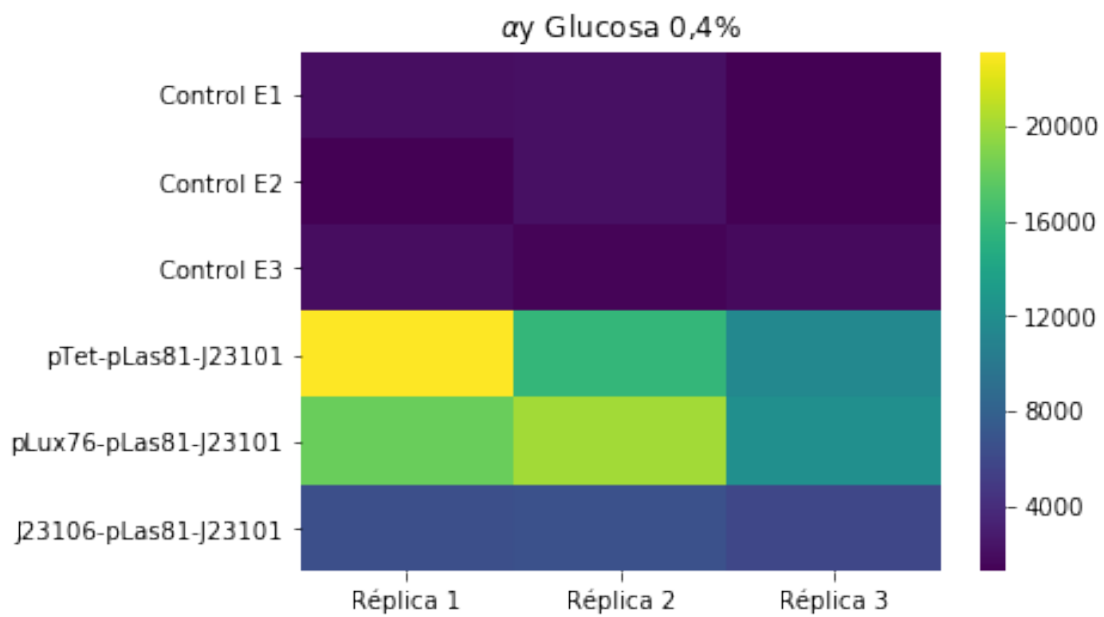
```

sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

plt.figure()
plt.title(r'$\alpha$y Glicerol 0,2%')
sns.heatmap(ygli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

```

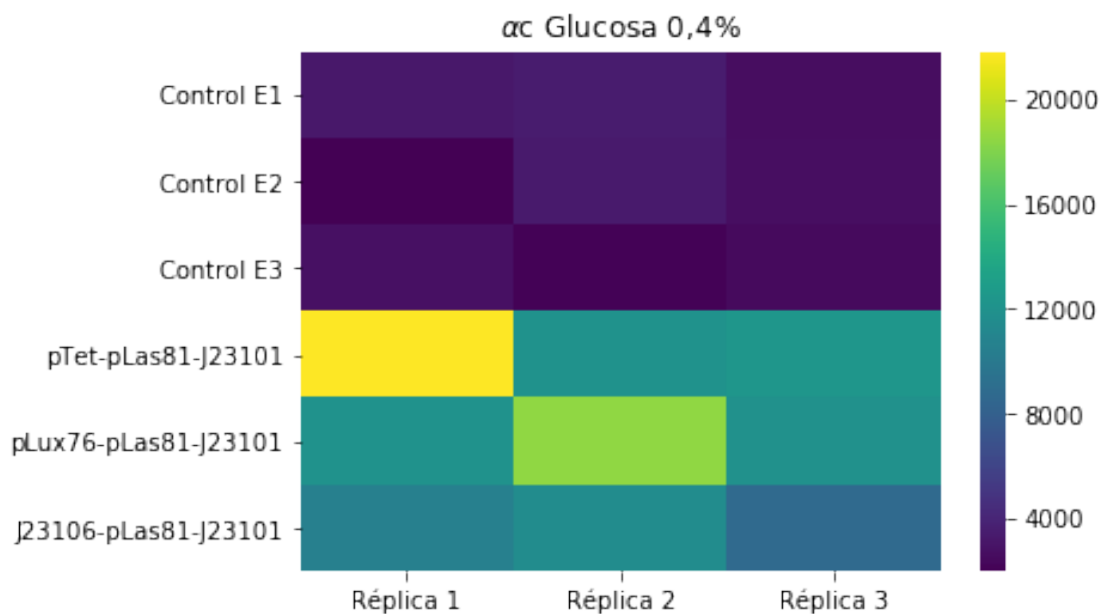
Out[136]: <matplotlib.axes.\_subplots.AxesSubplot at 0x178994fb0b8>

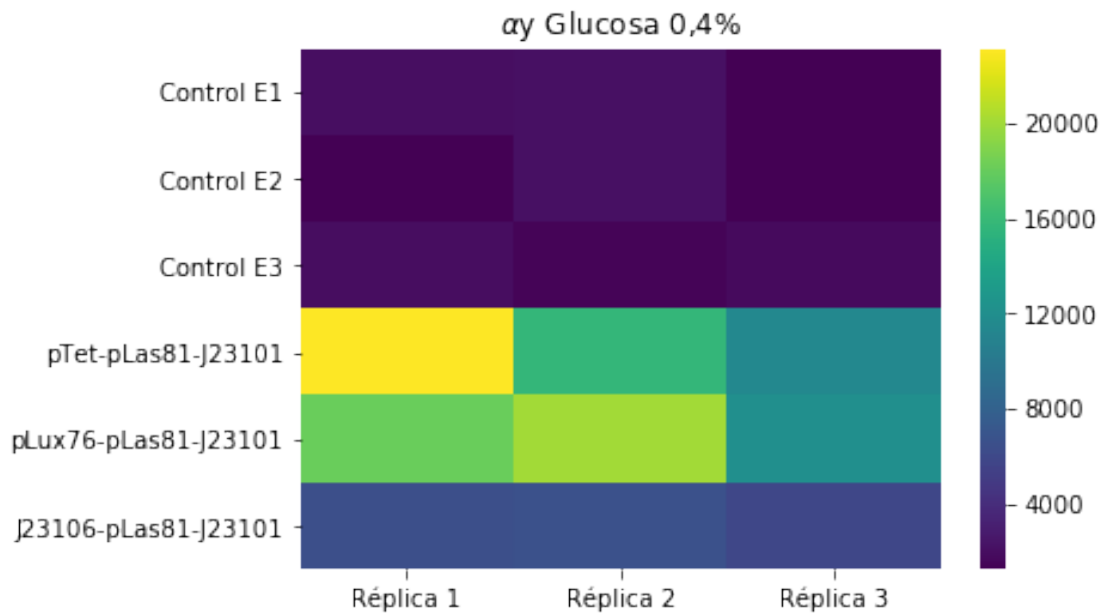
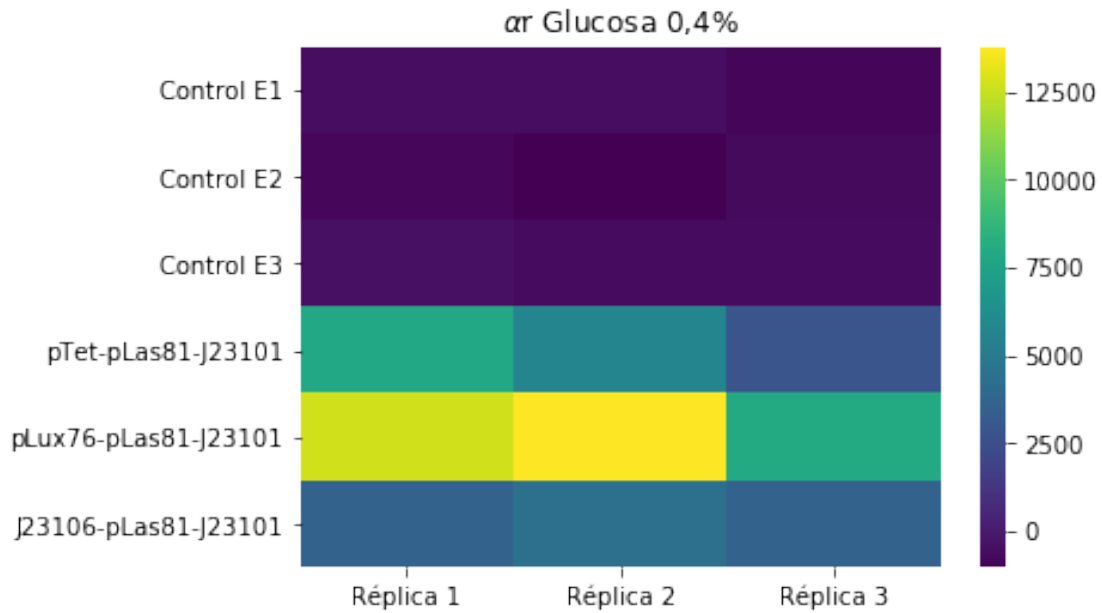


```
In [137]: ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101','J23106-pLas81-J23101']
          xlabel=['Réplica 1','Réplica 2','Réplica 3']
```

```
plt.figure()
plt.title(r'$\alpha$ Glucosa 0,4%')
sns.heatmap(cglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.figure()
plt.title(r'$\alpha$ Glucosa 0,4%')
sns.heatmap(rglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.figure()
plt.title(r'$\alpha$ Glucosa 0,4%')
sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
```

```
Out[137]: <matplotlib.axes._subplots.AxesSubplot at 0x1789bb90f28>
```





```
In [138]: ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101']
          xlabel=['Réplica 1','Réplica 2','Réplica 3']

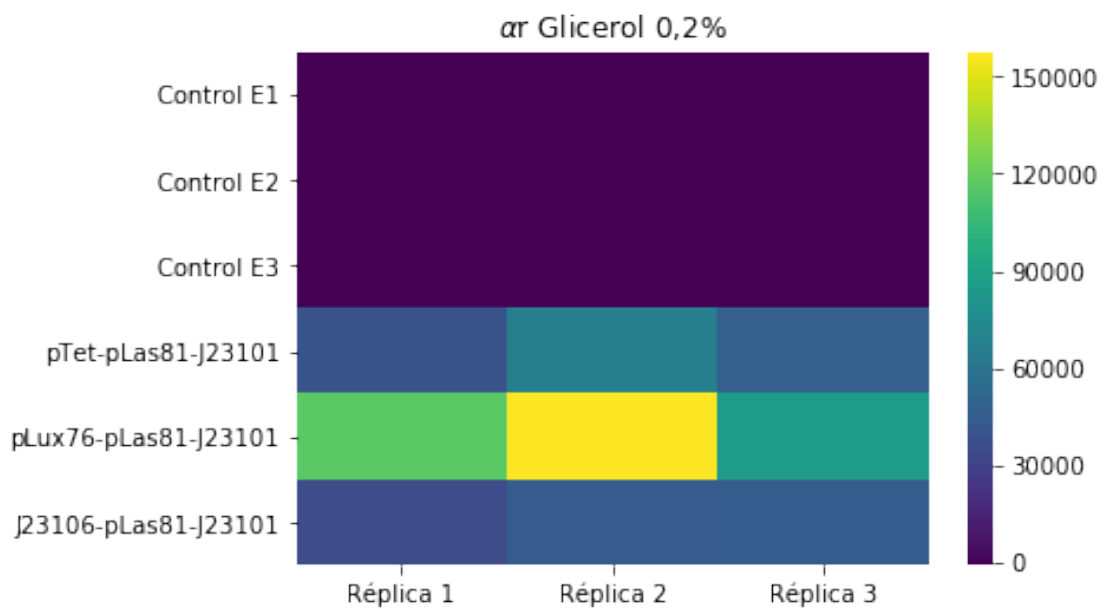
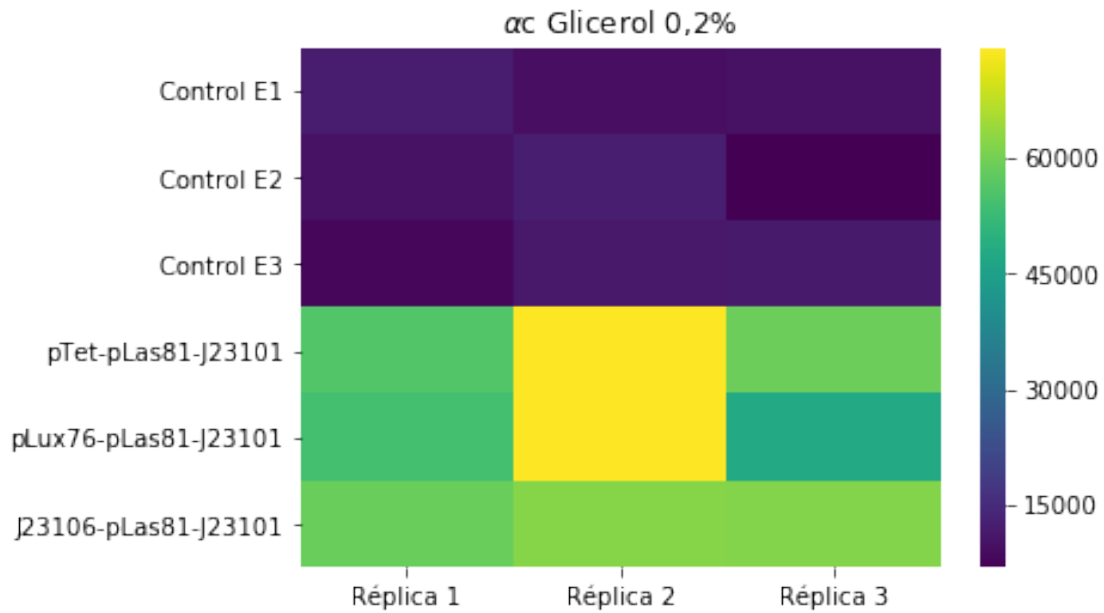
          plt.figure()
          plt.title(r'$\alpha$ Glicerol 0,2%')
```

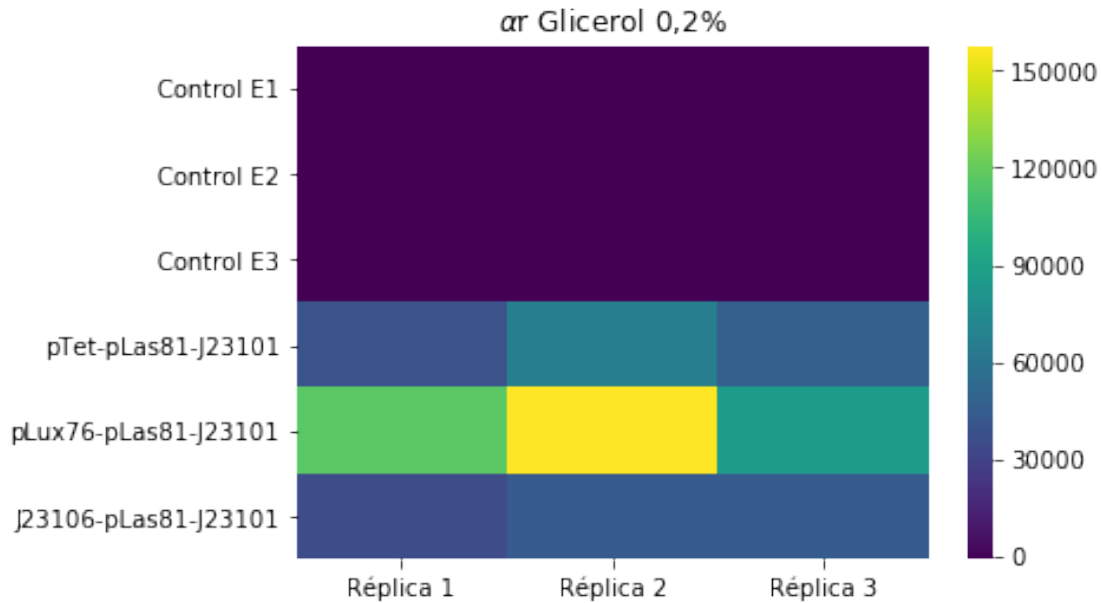
```

sns.heatmap(cgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.figure()
plt.title(r'$\alpha$ Glicerol 0,2%')
sns.heatmap(rgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.figure()
plt.title(r'$\alpha$ Glicerol 0,2%')
sns.heatmap(rgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

```

Out[138]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1789bc5d4a8>



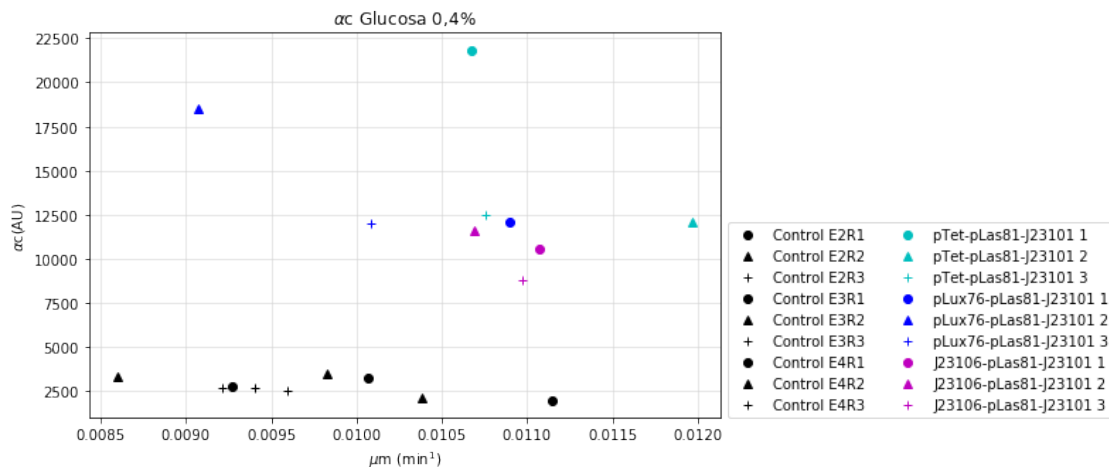


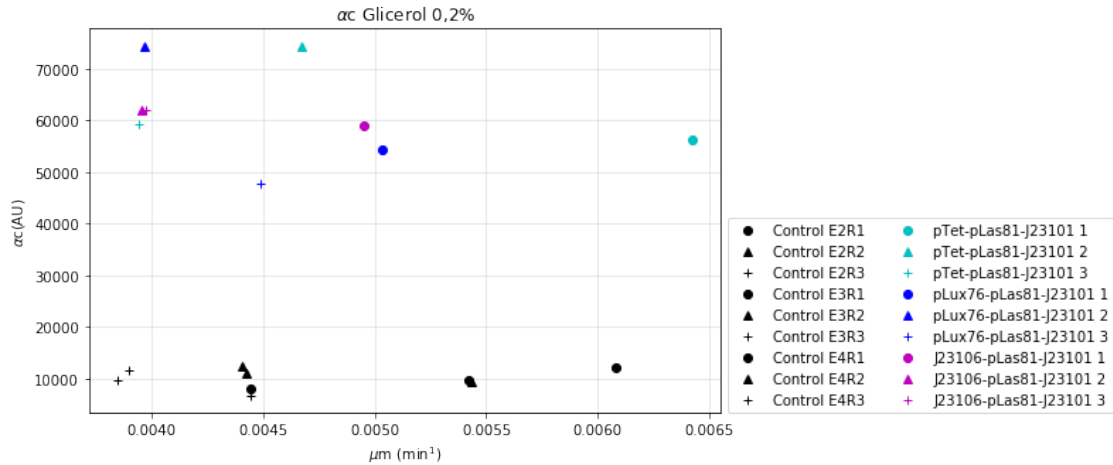
```
In [139]: #grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glucosa 0,4%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$(AU)')
plt.plot(um1,slopec1,'ko',label='Control E2R1')
plt.plot(um2,slopec2,'k^',label='Control E2R2')
plt.plot(um3,slopec3,'k+',label='Control E2R3')
plt.plot(um7,slopec7,'ko',label='Control E3R1')
plt.plot(um8,slopec8,'k^',label='Control E3R2')
plt.plot(um9,slopec9,'k+',label='Control E3R3')
plt.plot(um13,slopec13,'ko',label='Control E4R1')
plt.plot(um14,slopec14,'k^',label='Control E4R2')
plt.plot(um15,slopec15,'k+',label='Control E4R3')
plt.plot(um19,slopec19,'co',label='pTet-pLas81-J23101 1')
plt.plot(um20,slopec20,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um21,slopec21,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um25,slopec25,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um26,slopec26,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um27,slopec27,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um31,slopec31,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um32,slopec32,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um33,slopec33,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
```

```
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glicerol 0,2%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$ (AU)')
plt.plot(um4,slopec4,'ko',label='Control E2R1')
plt.plot(um5,slopec5,'k^',label='Control E2R2')
plt.plot(um6,slopec6,'k+',label='Control E2R3')
plt.plot(um10,slopec10,'ko',label='Control E3R1')
plt.plot(um11,slopec11,'k^',label='Control E3R2')
plt.plot(um12,slopec12,'k+',label='Control E3R3')
plt.plot(um16,slopec16,'ko',label='Control E4R1')
plt.plot(um17,slopec17,'k^',label='Control E4R2')
plt.plot(um18,slopec18,'k+',label='Control E4R3')
plt.plot(um22,slopec22,'co',label='pTet-pLas81-J23101 1')
plt.plot(um23,slopec23,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um24,slopec24,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um28,slopec28,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um29,slopec29,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,slopec30,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um34,slopec34,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um35,slopec35,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um36,slopec36,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[139]: <matplotlib.legend.Legend at 0x1789c1e7f60>





```
In [140]: #grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glucosa 0,4%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$ (AU)')
plt.plot(um1,sloper1,'ko',label='Control E2R1')
plt.plot(um2,sloper2,'k^',label='Control E2R2')
plt.plot(um3,sloper3,'k+',label='Control E2R3')
plt.plot(um7,sloper7,'ko',label='Control E3R1')
plt.plot(um8,sloper8,'k^',label='Control E3R2')
plt.plot(um9,sloper9,'k+',label='Control E3R3')
plt.plot(um13,sloper13,'ko',label='Control E4R1')
plt.plot(um14,sloper14,'k^',label='Control E4R2')
plt.plot(um15,sloper15,'k+',label='Control E4R3')
plt.plot(um19,sloper19,'co',label='pTet-pLas81-J23101 1')
plt.plot(um20,sloper20,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um21,sloper21,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um25,sloper25,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um26,sloper26,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um27,sloper27,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um31,sloper31,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um32,sloper32,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um33,sloper33,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

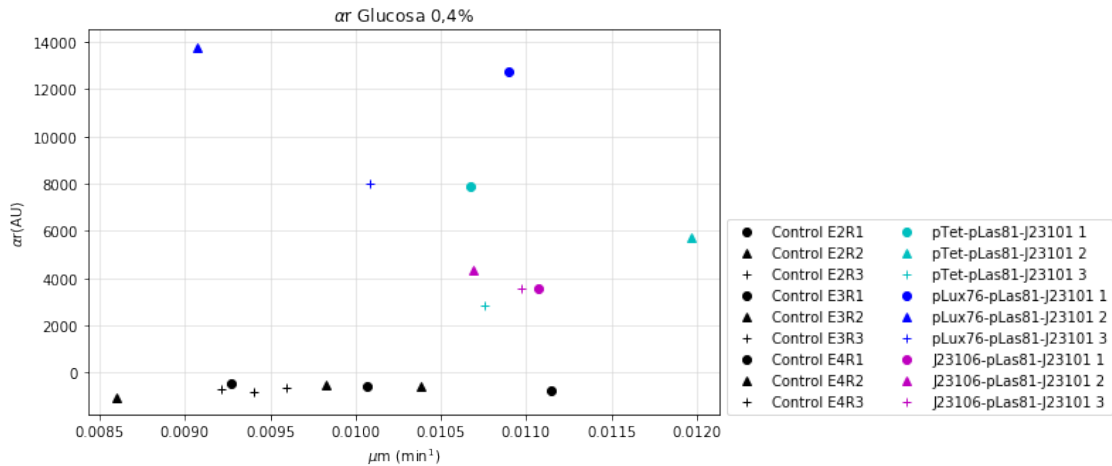
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glicerol 0,2%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$ (AU)')
```

```

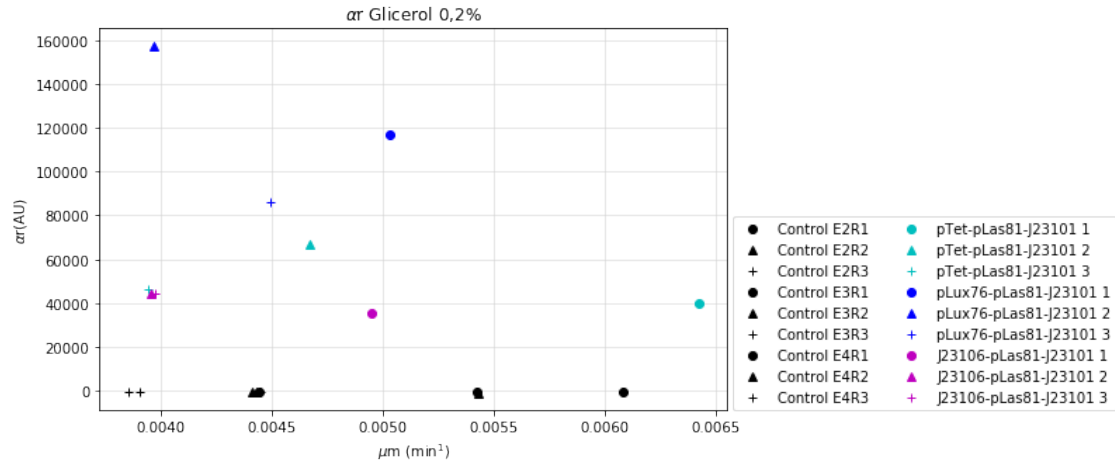
plt.plot(um4,sloper4,'ko',label='Control E2R1')
plt.plot(um5,sloper5,'k^',label='Control E2R2')
plt.plot(um6,sloper6,'k+',label='Control E2R3')
plt.plot(um10,sloper10,'ko',label='Control E3R1')
plt.plot(um11,sloper11,'k^',label='Control E3R2')
plt.plot(um12,sloper12,'k+',label='Control E3R3')
plt.plot(um16,sloper16,'ko',label='Control E4R1')
plt.plot(um17,sloper17,'k^',label='Control E4R2')
plt.plot(um18,sloper18,'k+',label='Control E4R3')
plt.plot(um22,sloper22,'co',label='pTet-pLas81-J23101 1')
plt.plot(um23,sloper23,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um24,sloper24,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um28,sloper28,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um29,sloper29,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,sloper30,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um34,sloper34,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um35,sloper35,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um36,sloper36,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[140]: <matplotlib.legend.Legend at 0x1789c734b38>







```
In [141]: #grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glucosa 0,4%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$ (AU)')
plt.plot(um1,slopy1,'ko',label='Control E2R1')
plt.plot(um2,slopy2,'k^',label='Control E2R2')
plt.plot(um3,slopy3,'k+',label='Control E2R3')
plt.plot(um7,slopy7,'ko',label='Control E3R1')
plt.plot(um8,slopy8,'k^',label='Control E3R2')
plt.plot(um9,slopy9,'k+',label='Control E3R3')
plt.plot(um13,slopy13,'ko',label='Control E4R1')
plt.plot(um14,slopy14,'k^',label='Control E4R2')
plt.plot(um15,slopy15,'k+',label='Control E4R3')
plt.plot(um19,slopy19,'co',label='pTet-pLas81-J23101 1')
plt.plot(um20,slopy20,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um21,slopy21,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um25,slopy25,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um26,slopy26,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um27,slopy27,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um31,slopy31,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um32,slopy32,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um33,slopy33,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

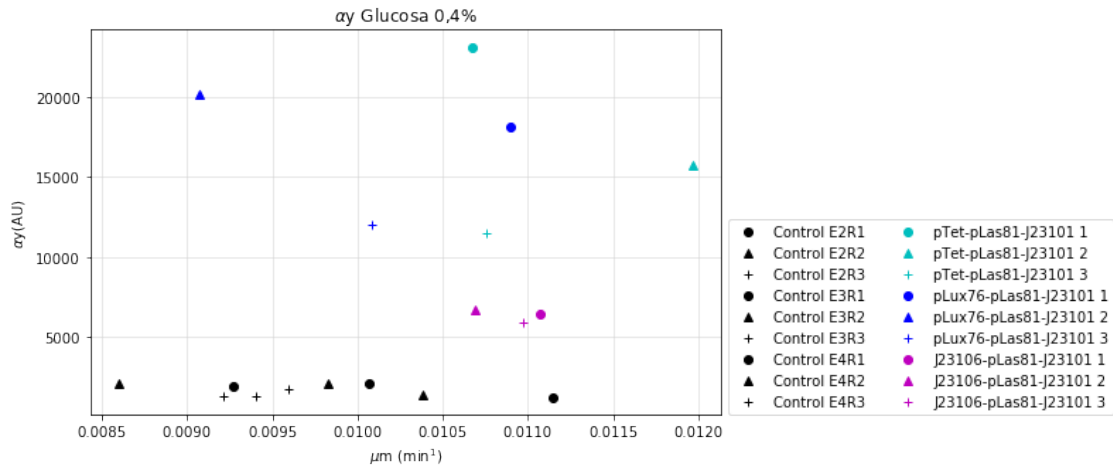
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glicerol 0,2%')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$ (AU)')
```

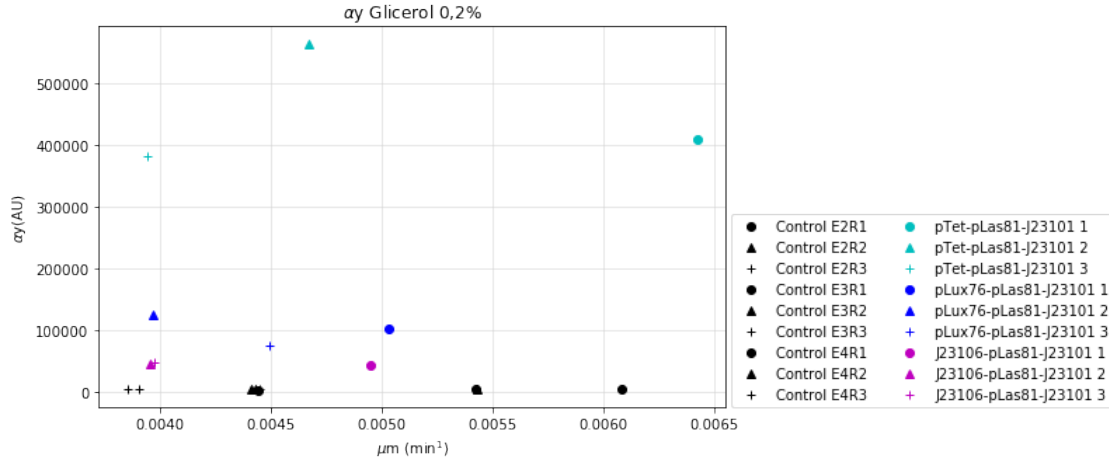
```

plt.plot(um4,slopy4,'ko',label='Control E2R1')
plt.plot(um5,slopy5,'k^',label='Control E2R2')
plt.plot(um6,slopy6,'k+',label='Control E2R3')
plt.plot(um10,slopy10,'ko',label='Control E3R1')
plt.plot(um11,slopy11,'k^',label='Control E3R2')
plt.plot(um12,slopy12,'k+',label='Control E3R3')
plt.plot(um16,slopy16,'ko',label='Control E4R1')
plt.plot(um17,slopy17,'k^',label='Control E4R2')
plt.plot(um18,slopy18,'k+',label='Control E4R3')
plt.plot(um22,slopy22,'co',label='pTet-pLas81-J23101 1')
plt.plot(um23,slopy23,'c^',label='pTet-pLas81-J23101 2')
plt.plot(um24,slopy24,'c+',label='pTet-pLas81-J23101 3')
plt.plot(um28,slopy28,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(um29,slopy29,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,slopy30,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(um34,slopy34,'mo',label='J23106-pLas81-J23101 1')
plt.plot(um35,slopy35,'m^',label='J23106-pLas81-J23101 2')
plt.plot(um36,slopy36,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[141]: <matplotlib.legend.Legend at 0x1789bbd2128>





In [142]: *#grafico de ac versus Um*

```
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Réplica 1')
plt.xlabel(r'$\mu$m (min$^{-1}$)')
plt.ylabel(r'$\alpha$c (AU)')
plt.plot(um1,slopec1,'k.',label='Control Glucosa E2R1')
plt.plot(um7,slopec7,'k*',label='Control Glucosa E3R1')
plt.plot(um13,slopec13,'k+',label='Control Glucosa E4R1')
plt.plot(um19,slopec19,'c.',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um25,slopec25,'c*',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um31,slopec31,'c+',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.plot(um4,slopec4,'ko',label='Control Glucosa E2R1')
plt.plot(um10,slopec10,'kp',label='Control Glucosa E3R1')
plt.plot(um16,slopec16,'k^',label='Control Glucosa E4R1')
plt.plot(um22,slopec22,'bo',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um28,slopec28,'bp',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um34,slopec34,'b^',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

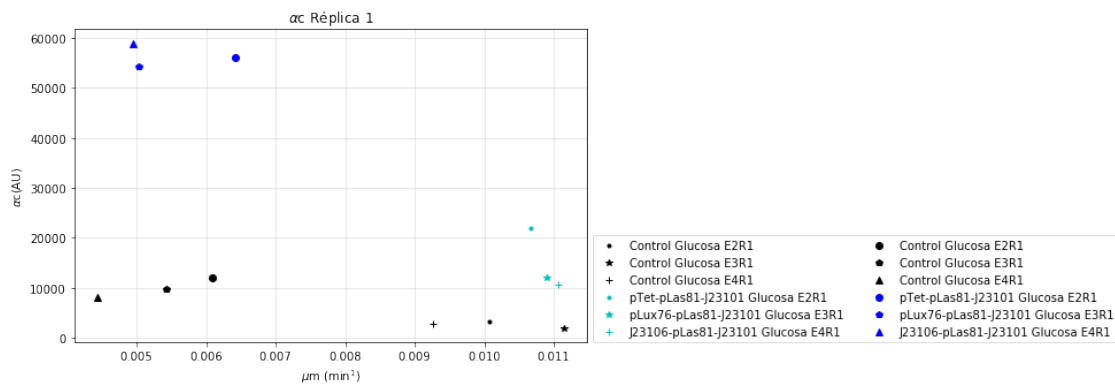
*#grafico de ac versus Um*

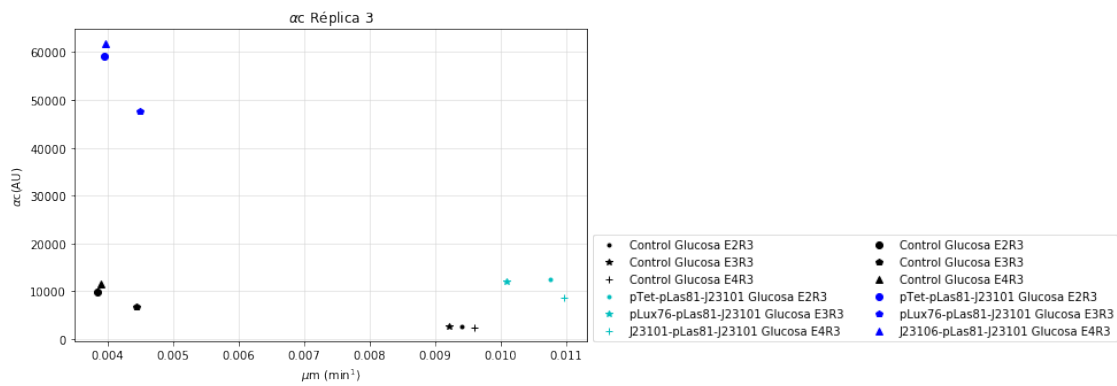
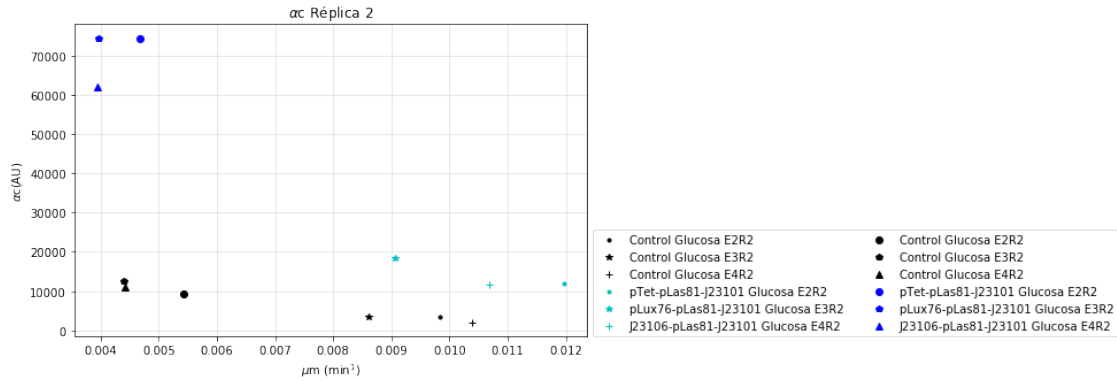
```
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Réplica 2')
plt.xlabel(r'$\mu$m (min$^{-1}$)')
plt.ylabel(r'$\alpha$c (AU)')
plt.plot(um2,slopec2,'k.',label='Control Glucosa E2R2')
plt.plot(um8,slopec8,'k*',label='Control Glucosa E3R2')
plt.plot(um14,slopec14,'k+',label='Control Glucosa E4R2')
plt.plot(um20,slopec20,'c.',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um26,slopec26,'c*',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um32,slopec32,'c+',label='J23106-pLas81-J23101 Glucosa E4R2')
```

```
plt.plot(um5,slopec5,'ko',label='Control Glucosa E2R2')
plt.plot(um11,slopec11,'kp',label='Control Glucosa E3R2')
plt.plot(um17,slopec17,'k^',label='Control Glucosa E4R2')
plt.plot(um23,slopec23,'bo',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um29,slopec29,'bp',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um35,slopec35,'b^',label='J23106-pLas81-J23101 Glucosa E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Réplica 3')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha$(AU)')
plt.plot(um3,slopec3,'k.',label='Control Glucosa E2R3')
plt.plot(um9,slopec9,'k*',label='Control Glucosa E3R3')
plt.plot(um15,slopec15,'k+',label='Control Glucosa E4R3')
plt.plot(um21,slopec21,'c.',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um27,slopec27,'c*',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um33,slopec33,'c+',label='J23101-pLas81-J23101 Glucosa E4R3')
plt.plot(um6,slopec6,'ko',label='Control Glucosa E2R3')
plt.plot(um12,slopec12,'kp',label='Control Glucosa E3R3')
plt.plot(um18,slopec18,'k^',label='Control Glucosa E4R3')
plt.plot(um24,slopec24,'bo',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um30,slopec30,'bp',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um36,slopec36,'b^',label='J23106-pLas81-J23101 Glucosa E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[142]: <matplotlib.legend.Legend at 0x17899535b38>





In [143]: *#grafico de ar versus Um*

```
plt.figure(figsize=(8,5))
plt.title(r'$\alpha_r$ Réplica 1')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
plt.ylabel(r'$\alpha_r$ (AU)')
plt.plot(um1,sloper1,'k.',label='Control Glucosa E2R1')
plt.plot(um7,sloper7,'k*',label='Control Glucosa E3R1')
plt.plot(um13,sloper13,'k+',label='Control Glucosa E4R1')
plt.plot(um19,sloper19,'r.',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um25,sloper25,'r*',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um31,sloper31,'r+',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.plot(um4,sloper4,'ko',label='Control Glucosa E2R1')
plt.plot(um10,sloper10,'kp',label='Control Glucosa E3R1')
plt.plot(um16,sloper16,'k^',label='Control Glucosa E4R1')
plt.plot(um22,sloper22,'mo',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um28,sloper28,'mp',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um34,sloper34,'m^',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```

#grafico de ar versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Réplica 2')
plt.xlabel(r'$\mu$ (min$^1$)')
plt.ylabel(r'$\alpha$(AU)')
plt.plot(um2,sloper2,'k.',label='Control Glucosa E2R2')
plt.plot(um8,sloper8,'k*',label='Control Glucosa E3R2')
plt.plot(um14,sloper14,'k+',label='Control Glucosa E4R2')
plt.plot(um20,sloper20,'r.',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um26,sloper26,'r*',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um32,sloper32,'r+',label='J23106-pLas81-J23101 Glucosa E4R2')
plt.plot(um5,sloper5,'ko',label='Control Glucosa E2R2')
plt.plot(um11,sloper11,'kp',label='Control Glucosa E3R2')
plt.plot(um17,sloper17,'k^',label='Control Glucosa E4R2')
plt.plot(um23,sloper23,'mo',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um29,sloper29,'mp',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um35,sloper35,'m^',label='J23106-pLas81-J23101 Glucosa E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

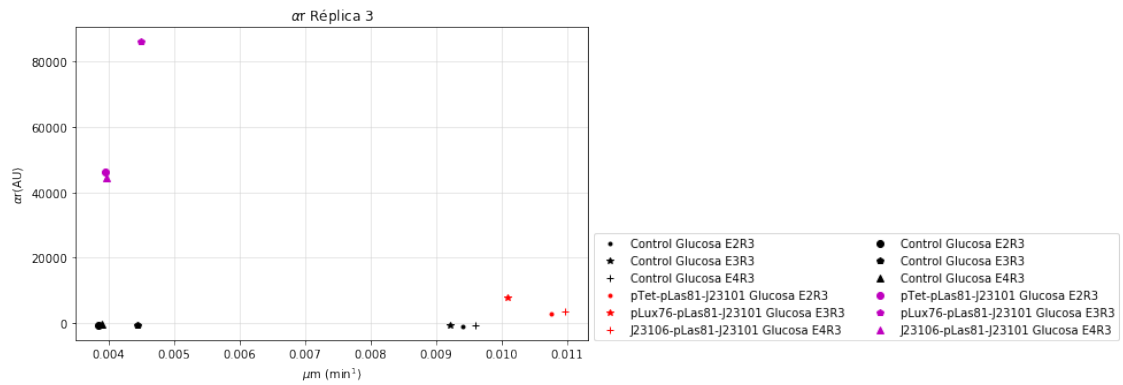
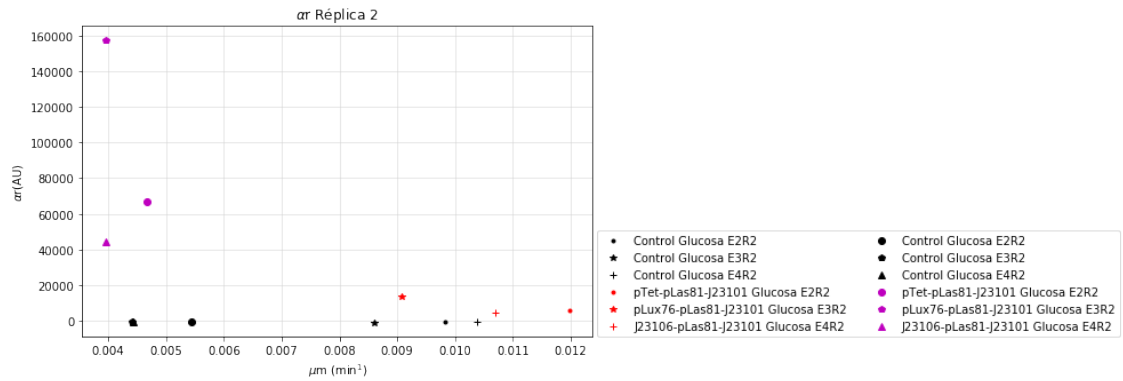
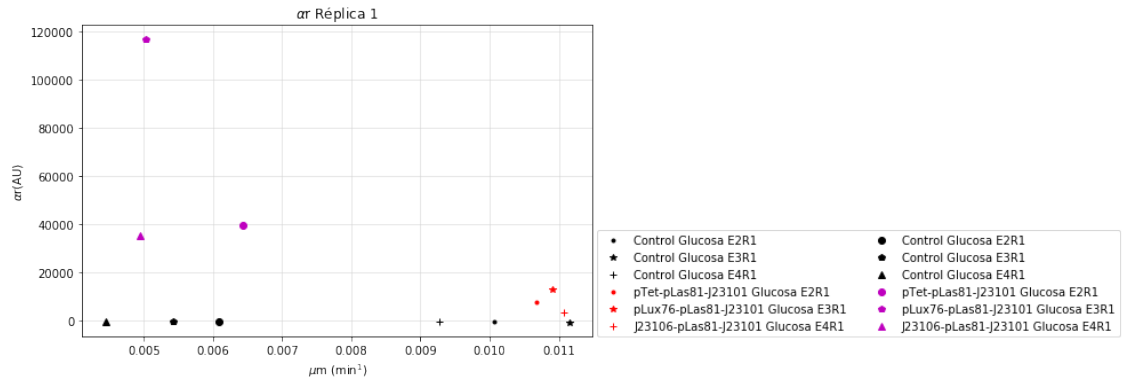
```

```

#grafico de ar versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Réplica 3')
plt.xlabel(r'$\mu$ (min$^1$)')
plt.ylabel(r'$\alpha$(AU)')
plt.plot(um3,sloper3,'k.',label='Control Glucosa E2R3')
plt.plot(um9,sloper9,'k*',label='Control Glucosa E3R3')
plt.plot(um15,sloper15,'k+',label='Control Glucosa E4R3')
plt.plot(um21,sloper21,'r.',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um27,sloper27,'r*',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um33,sloper33,'r+',label='J23106-pLas81-J23101 Glucosa E4R3')
plt.plot(um6,sloper6,'ko',label='Control Glucosa E2R3')
plt.plot(um12,sloper12,'kp',label='Control Glucosa E3R3')
plt.plot(um18,sloper18,'k^',label='Control Glucosa E4R3')
plt.plot(um24,sloper24,'mo',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um30,sloper30,'mp',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um36,sloper36,'m^',label='J23106-pLas81-J23101 Glucosa E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[143]: <matplotlib.legend.Legend at 0x1789b47ba58>



```
In [144]: #grafico de ay versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Réplica 1')
plt.xlabel(r'$\mu$ (min$^{-1}$)')
```

```

plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopy1,'k.',label='Control Glucosa E2R1')
plt.plot(um7,slopy7,'k*',label='Control Glucosa E3R1')
plt.plot(um13,slopy13,'k+',label='Control Glucosa E4R1')
plt.plot(um19,slopy19,'y.',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um25,slopy25,'y*',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um31,slopy31,'y+',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.plot(um4,slopy4,'ko',label='Control Glucosa E2R1')
plt.plot(um10,slopy10,'kp',label='Control Glucosa E3R1')
plt.plot(um16,slopy16,'k^',label='Control Glucosa E4R1')
plt.plot(um22,slopy22,'go',label='pTet-pLas81-J23101 Glucosa E2R1')
plt.plot(um28,slopy28,'gp',label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(um34,slopy34,'g^',label='J23106-pLas81-J23101 Glucosa E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

*#grafico de ay versus Um*

```

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Réplica 2')
plt.xlabel(r'$\mu$M (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um2,slopy2,'k.',label='Control Glucosa E2R2')
plt.plot(um8,slopy8,'k*',label='Control Glucosa E3R2')
plt.plot(um14,slopy14,'k+',label='Control Glucosa E4R2')
plt.plot(um20,slopy20,'y.',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um26,slopy26,'y*',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um32,slopy32,'y+',label='J23106-pLas81-J23101 Glucosa E4R2')
plt.plot(um5,slopy5,'ko',label='Control Glucosa E2R2')
plt.plot(um11,slopy11,'kp',label='Control Glucosa E3R2')
plt.plot(um17,slopy17,'k^',label='Control Glucosa E4R2')
plt.plot(um23,slopy23,'go',label='pTet-pLas81-J23101 Glucosa E2R2')
plt.plot(um29,slopy29,'gp',label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(um35,slopy35,'g^',label='J23106-pLas81-J23101 Glucosa E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

*#grafico de ay versus Um*

```

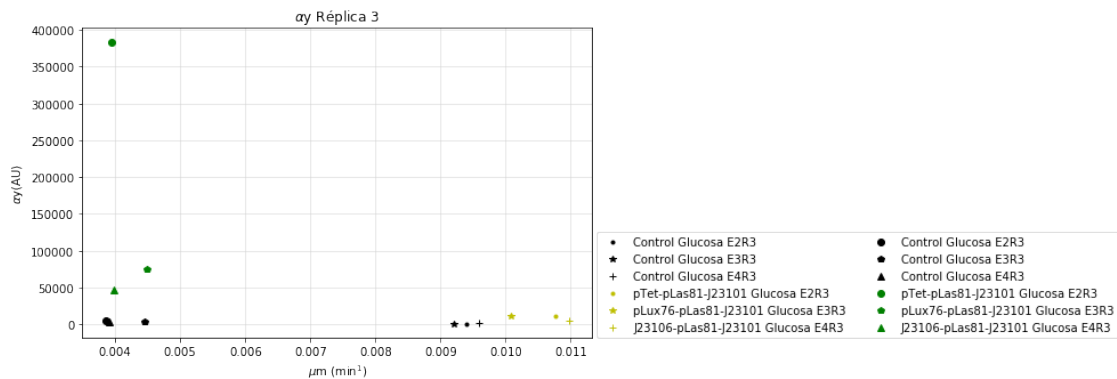
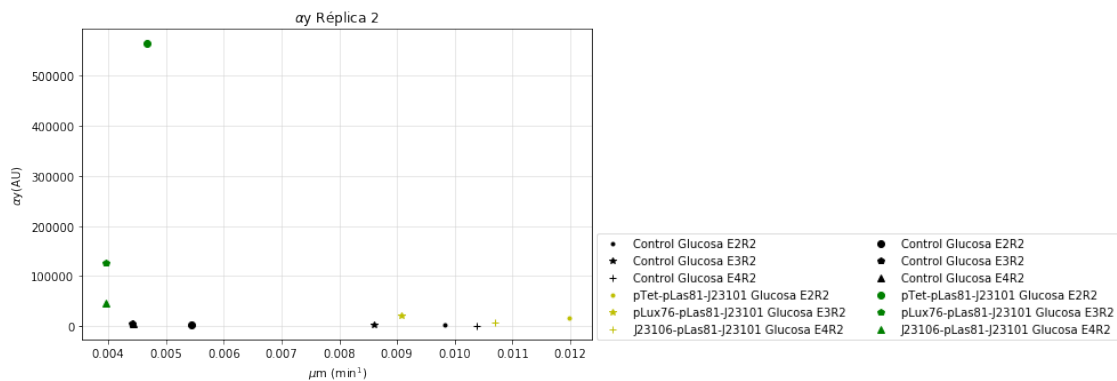
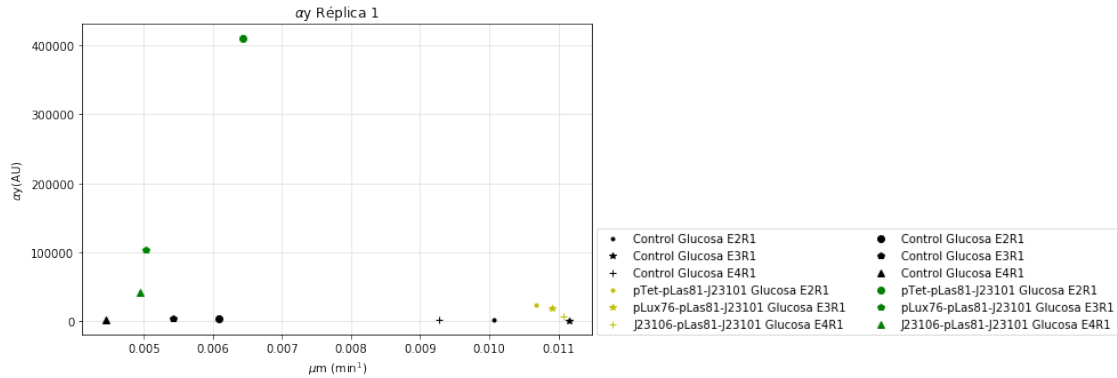
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Réplica 3')
plt.xlabel(r'$\mu$M (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um3,slopy3,'k.',label='Control Glucosa E2R3')
plt.plot(um9,slopy9,'k*',label='Control Glucosa E3R3')
plt.plot(um15,slopy15,'k+',label='Control Glucosa E4R3')
plt.plot(um21,slopy21,'y.',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um27,slopy27,'y*',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um33,slopy33,'y+',label='J23106-pLas81-J23101 Glucosa E4R3')
plt.plot(um6,slopy6,'ko',label='Control Glucosa E2R3')

```



```
plt.plot(um12,slopy12,'kp',label='Control Glucosa E3R3')
plt.plot(um18,slopy18,'k^',label='Control Glucosa E4R3')
plt.plot(um24,slopy24,'go',label='pTet-pLas81-J23101 Glucosa E2R3')
plt.plot(um30,slopy30,'gp',label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(um36,slopy36,'g^',label='J23106-pLas81-J23101 Glucosa E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[144]: <matplotlib.legend.Legend at 0x1789d338f28>



```

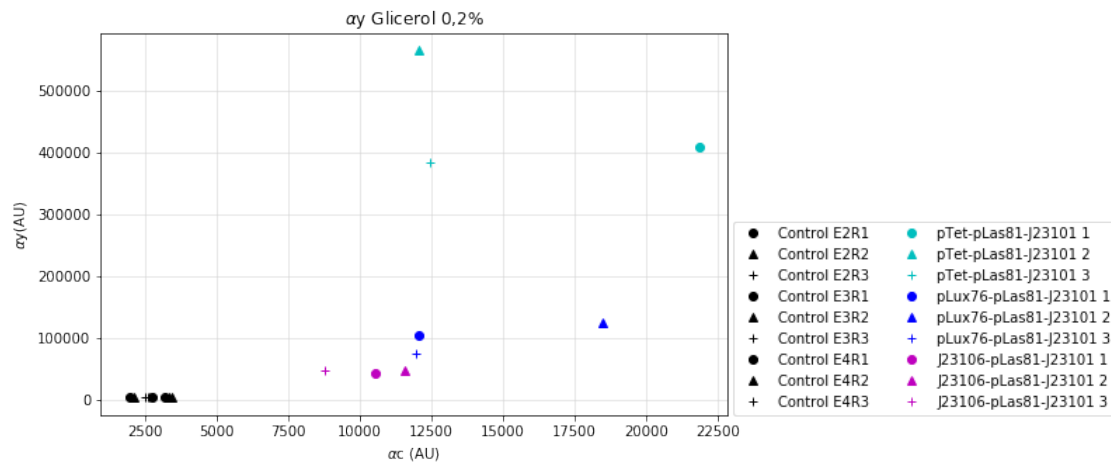
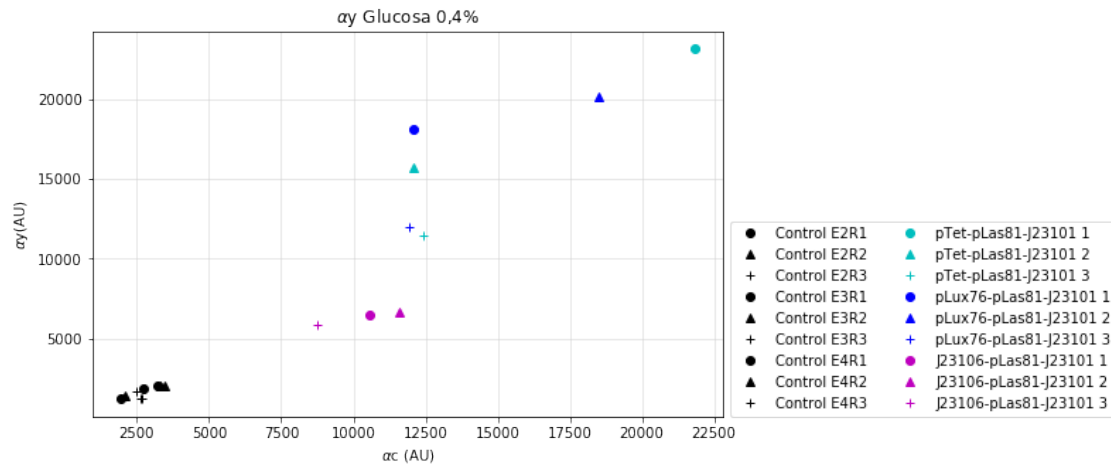
In [145]: #grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopy1,'ko',label='Control E2R1')
plt.plot(slopec2,slopy2,'k^',label='Control E2R2')
plt.plot(slopec3,slopy3,'k+',label='Control E2R3')
plt.plot(slopec7,slopy7,'ko',label='Control E3R1')
plt.plot(slopec8,slopy8,'k^',label='Control E3R2')
plt.plot(slopec9,slopy9,'k+',label='Control E3R3')
plt.plot(slopec13,slopy13,'ko',label='Control E4R1')
plt.plot(slopec14,slopy14,'k^',label='Control E4R2')
plt.plot(slopec15,slopy15,'k+',label='Control E4R3')
plt.plot(slopec19,slopy19,'co',label='pTet-pLas81-J23101 1')
plt.plot(slopec20,slopy20,'c^',label='pTet-pLas81-J23101 2')
plt.plot(slopec21,slopy21,'c+',label='pTet-pLas81-J23101 3')
plt.plot(slopec25,slopy25,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,slopy26,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec27,slopy27,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(slopec31,slopy31,'mo',label='J23106-pLas81-J23101 1')
plt.plot(slopec32,slopy32,'m^',label='J23106-pLas81-J23101 2')
plt.plot(slopec33,slopy33,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ay vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopy4,'ko',label='Control E2R1')
plt.plot(slopec2,slopy5,'k^',label='Control E2R2')
plt.plot(slopec3,slopy6,'k+',label='Control E2R3')
plt.plot(slopec7,slopy10,'ko',label='Control E3R1')
plt.plot(slopec8,slopy11,'k^',label='Control E3R2')
plt.plot(slopec9,slopy12,'k+',label='Control E3R3')
plt.plot(slopec13,slopy16,'ko',label='Control E4R1')
plt.plot(slopec14,slopy17,'k^',label='Control E4R2')
plt.plot(slopec15,slopy18,'k+',label='Control E4R3')
plt.plot(slopec19,slopy22,'co',label='pTet-pLas81-J23101 1')
plt.plot(slopec20,slopy23,'c^',label='pTet-pLas81-J23101 2')
plt.plot(slopec21,slopy24,'c+',label='pTet-pLas81-J23101 3')
plt.plot(slopec25,slopy28,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,slopy29,'b^',label='pLux76-pLas81-J23101 2')

```

```
plt.plot(slopec27,slopy30,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(slopec31,slopy34,'mo',label='J23106-pLas81-J23101 1')
plt.plot(slopec32,slopy35,'m^',label='J23106-pLas81-J23101 2')
plt.plot(slopec33,slopy36,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[145]: <matplotlib.legend.Legend at 0x1789e9c73c8>



```
In [146]: #grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
```

```

plt.plot(slopec1,sloper1,'ko',label='Control E2R1')
plt.plot(slopec2,sloper2,'k^',label='Control E2R2')
plt.plot(slopec3,sloper3,'k+',label='Control E2R3')
plt.plot(slopec7,sloper7,'ko',label='Control E3R1')
plt.plot(slopec8,sloper8,'k^',label='Control E3R2')
plt.plot(slopec9,sloper9,'k+',label='Control E3R3')
plt.plot(slopec13,sloper13,'ko',label='Control E4R1')
plt.plot(slopec14,sloper14,'k^',label='Control E4R2')
plt.plot(slopec15,sloper15,'k+',label='Control E4R3')
plt.plot(slopec19,sloper19,'co',label='pTet-pLas81-J23101 1')
plt.plot(slopec20,sloper20,'c^',label='pTet-pLas81-J23101 2')
plt.plot(slopec21,sloper21,'c+',label='pTet-pLas81-J23101 3')
plt.plot(slopec25,sloper25,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,sloper26,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec27,sloper27,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(slopec31,sloper31,'mo',label='J23106-pLas81-J23101 1')
plt.plot(slopec32,sloper32,'m^',label='J23106-pLas81-J23101 2')
plt.plot(slopec33,sloper33,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

```

#grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$ Glicerol 0,2%')
plt.xlabel(r'$\alpha$ c (AU)')
plt.ylabel(r'$\alpha$ r(AU)')
plt.plot(slopec1,sloper4,'ko',label='Control E2R1')
plt.plot(slopec2,sloper5,'k^',label='Control E2R2')
plt.plot(slopec3,sloper6,'k+',label='Control E2R3')
plt.plot(slopec7,sloper10,'ko',label='Control E3R1')
plt.plot(slopec8,sloper11,'k^',label='Control E3R2')
plt.plot(slopec9,sloper12,'k+',label='Control E3R3')
plt.plot(slopec13,sloper16,'ko',label='Control E4R1')
plt.plot(slopec14,sloper17,'k^',label='Control E4R2')
plt.plot(slopec15,sloper18,'k+',label='Control E4R3')
plt.plot(slopec19,sloper22,'co',label='pTet-pLas81-J23101 1')
plt.plot(slopec20,sloper23,'c^',label='pTet-pLas81-J23101 2')
plt.plot(slopec21,sloper24,'c+',label='pTet-pLas81-J23101 3')
plt.plot(slopec25,sloper28,'bo',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,sloper29,'b^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec27,sloper30,'b+',label='pLux76-pLas81-J23101 3')
plt.plot(slopec31,sloper34,'mo',label='J23106-pLas81-J23101 1')
plt.plot(slopec32,sloper35,'m^',label='J23106-pLas81-J23101 2')
plt.plot(slopec33,sloper36,'m+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[146]: <matplotlib.legend.Legend at 0x1789ef07dd8>

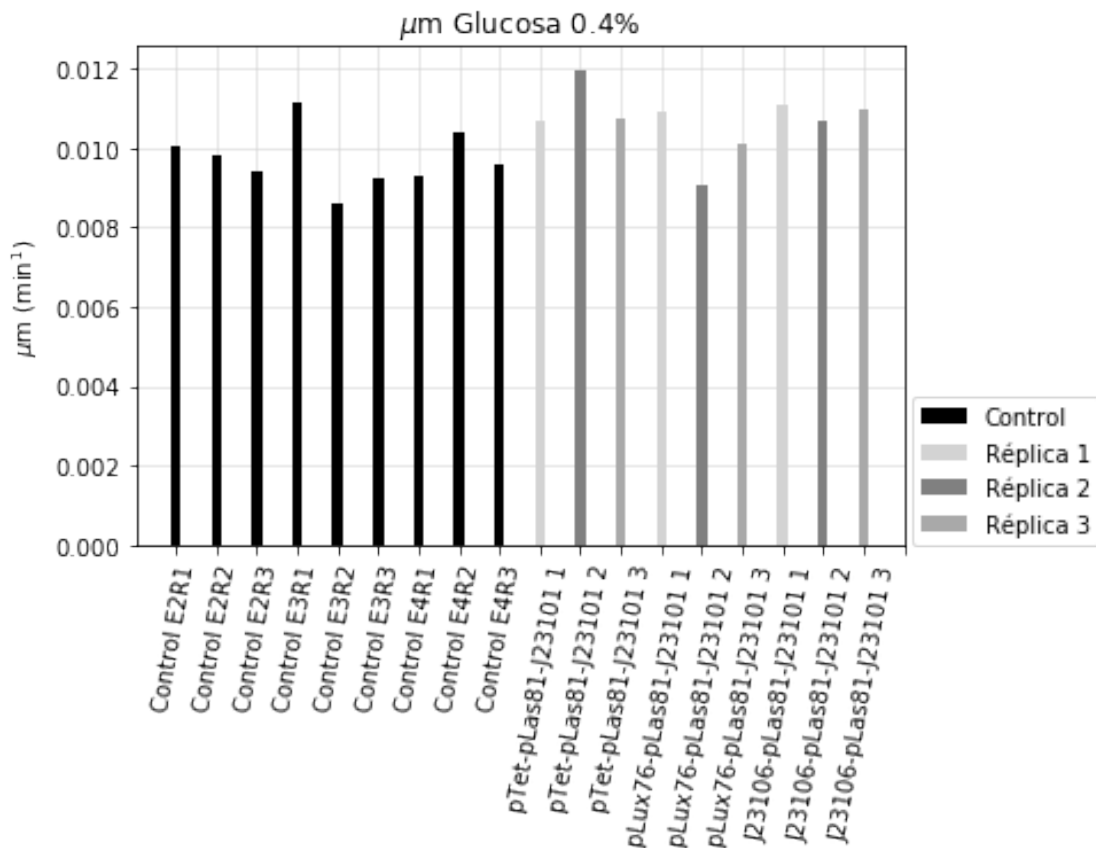


```

plt.bar(X[6]+0.00,uglu[6],color='k',width=0.25,zorder=3)
plt.bar(X[7]+0.00,uglu[7],color='k',width=0.25,zorder=3)
plt.bar(X[8]+0.00,uglu[8],color='k',width=0.25,zorder=3)
plt.bar(X[9]+0.00,uglu[9],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[10]+0.00,uglu[10],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[11]+0.00,uglu[11],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[12]+0.00,uglu[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,uglu[13],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,uglu[14],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[15]+0.00,uglu[15],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[16]+0.00,uglu[16],color='grey',width=0.25,zorder=3)
plt.bar(X[17]+0.00,uglu[17],color='darkgrey',width=0.25,zorder=3)
plt.xticks(X,['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2',
'Control E3R3','Control E4R1','Control E4R2','Control E4R3',
'pTet-plas81-j23101 1','pTet-plas81-j23101 2','pTet-plas81-j23101 3',
'pLux76-plas81-j23101 1','pLux76-plas81-j23101 2','pLux76-plas81-j23101 3',
'J23106-plas81-j23101 1','J23106-plas81-j23101 2','J23106-plas81-j23101 3'])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))

```

Out[152]: <matplotlib.legend.Legend at 0x1789acad358>



```

In [151]: X = np.arange(19)
plt.figure()

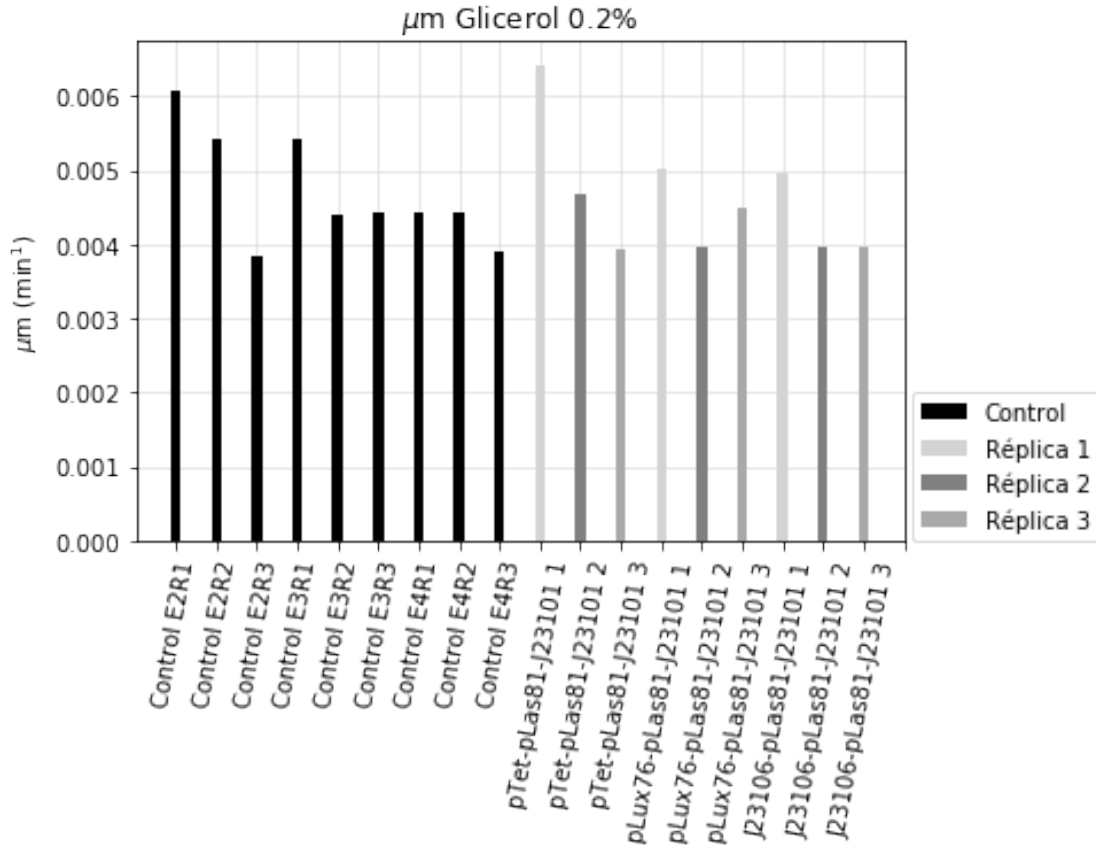
```

```

plt.title(r'$\mu$ Glicerol 0.2%')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.bar(X[0]+0.00,ugli[0],color='k',width=0.25,label='Control',zorder=3)
plt.bar(X[1]+0.00,ugli[1],color='k',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ugli[2],color='k',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ugli[3],color='k',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='k',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='k',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='k',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='k',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='k',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ugli[15],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ugli[16],color='grey',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ugli[17],color='darkgrey',width=0.25,zorder=3)
plt.xticks(X,['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2'])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))

```

Out[151]: <matplotlib.legend.Legend at 0x1789bb7bf28>



```
In [150]: X = np.arange(18)
plt.figure(figsize=(8,5))
plt.title(r'$\mu$')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.bar(X[0]-0.25,uglu[0],color='k',width=0.25,label='Control Glucosa',zorder=3)
plt.bar(X[0]+0.00,ugli[0],color='slategrey',width=0.25,label='Control Glicerol',zorder=3)
plt.bar(X[1]-0.20,uglu[1],color='k',width=0.25,zorder=3)
plt.bar(X[1]+0.00,ugli[1],color='slategrey',width=0.25,zorder=3)
plt.bar(X[2]-0.25,uglu[2],color='k',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ugli[2],color='slategrey',width=0.25,zorder=3)
plt.bar(X[3]-0.20,uglu[3],color='k',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ugli[3],color='slategrey',width=0.25,zorder=3)
plt.bar(X[4]-0.20,uglu[4],color='k',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='slategrey',width=0.25,zorder=3)
plt.bar(X[5]-0.20,uglu[5],color='k',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='slategrey',width=0.25,zorder=3)
plt.bar(X[6]-0.20,uglu[6],color='k',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='slategrey',width=0.25,zorder=3)
plt.bar(X[7]-0.20,uglu[7],color='k',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='slategrey',width=0.25,zorder=3)
```

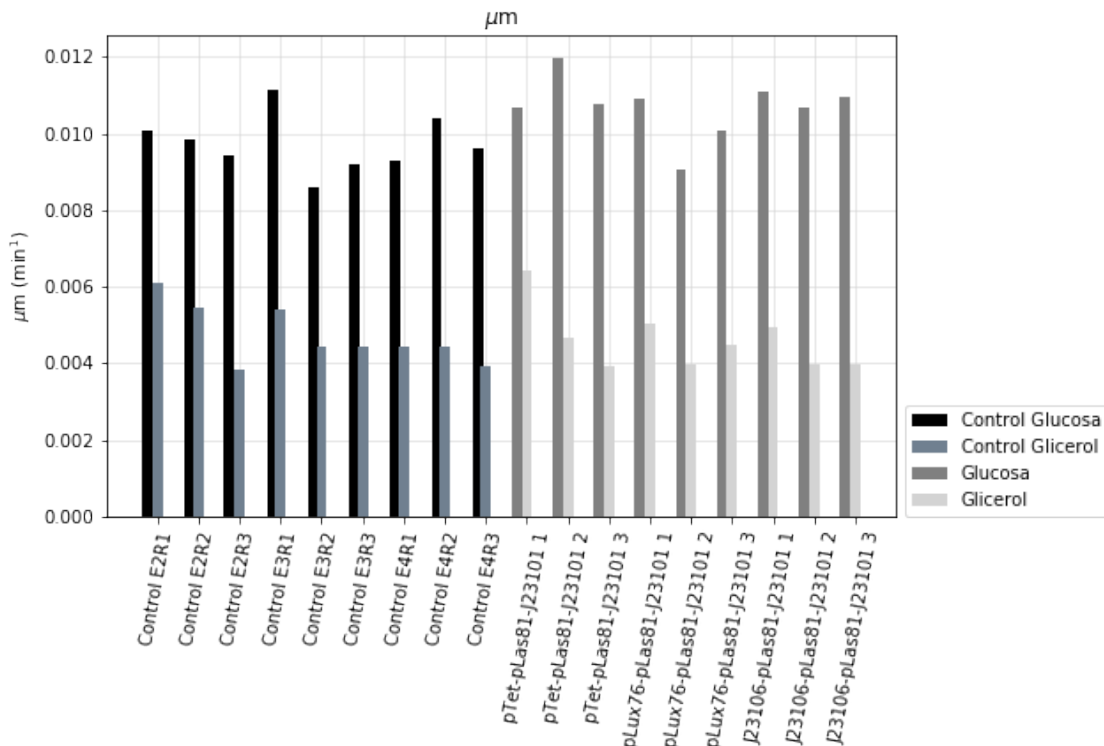


```

plt.bar(X[8]-0.20,uglu[8],color='k',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='slategrey',width=0.25,zorder=3)
plt.bar(X[9]-0.25,uglu[9],color='grey',width=0.25,label='Glucosa',zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,label='Glicerol',zorder=3)
plt.bar(X[10]-0.25,uglu[10],color='grey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[11]-0.25,uglu[11],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[12]-0.25,uglu[12],color='grey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]-0.25,uglu[13],color='grey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[14]-0.25,uglu[14],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[15]-0.25,uglu[15],color='grey',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ugli[15],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[16]-0.25,uglu[16],color='grey',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ugli[16],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[17]-0.25,uglu[17],color='grey',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ugli[17],color='lightgrey',width=0.25,zorder=3)
plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2','Control E3R3','Control E4R1','Control E4R2','Control E4R3',
               'pTet-pLas81-j23101 1','pTet-pLas81-j23101 2','pTet-pLas81-j23101 3',
               'pLux76-pLas81-j23101 1','pLux76-pLas81-j23101 2','pLux76-pLas81-j23101 3',
               'p3106-pLas81-j23101 1','p3106-pLas81-j23101 2','p3106-pLas81-j23101 3'])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))

```

Out[150]: <matplotlib.legend.Legend at 0x17896ee7198>



In [153]: *#Ro RFP*

```
pr1=sloper1/slopec1
pr2=sloper2/slopec2
pr3=sloper3/slopec3
pr4=sloper4/slopec4
pr5=sloper5/slopec5
pr6=sloper6/slopec6
pr7=sloper7/slopec7
pr8=sloper8/slopec8
pr9=sloper9/slopec9
pr10=sloper10/slopec10
pr11=sloper11/slopec11
pr12=sloper12/slopec12
pr13=sloper13/slopec13
pr14=sloper14/slopec14
pr15=sloper15/slopec15
pr16=sloper16/slopec16
pr17=sloper17/slopec17
pr18=sloper18/slopec18
pr19=sloper19/slopec19
pr20=sloper20/slopec20
pr21=sloper21/slopec21
pr22=sloper22/slopec22
pr23=sloper23/slopec23
pr24=sloper24/slopec24
pr25=sloper25/slopec25
pr26=sloper26/slopec26
pr27=sloper27/slopec27
pr28=sloper28/slopec28
pr29=sloper29/slopec29
pr30=sloper30/slopec30
pr31=sloper31/slopec31
pr32=sloper32/slopec32
pr33=sloper33/slopec33
pr34=sloper34/slopec34
pr35=sloper35/slopec35
pr36=sloper36/slopec36
```

```
ro_rfp=[pr1,pr2,pr3,pr4,pr5,pr6,pr7,pr8,pr9,pr10,pr11,pr12,pr13,pr14,pr15,pr16,pr17,pr18,pr19,pr20,pr21,pr22,pr23,pr24,pr25,pr26,pr27,pr28,pr29,pr30,pr31,pr32,pr33,pr34,pr35,pr36]
ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr27],[pr31,pr32,pr33]]
ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,pr30],[pr34,pr35,pr36]]
```

In [154]: *#Ro YFP*

```
py1=slopey1/slopec1
```

```

py2=slopey2/slopec2
py3=slopey3/slopec3
py4=slopey4/slopec4
py5=slopey5/slopec5
py6=slopey6/slopec6
py7=slopey7/slopec7
py8=slopey8/slopec8
py9=slopey9/slopec9
py10=slopey10/slopec10
py11=slopey11/slopec11
py12=slopey12/slopec12
py13=slopey13/slopec13
py14=slopey14/slopec14
py15=slopey15/slopec15
py16=slopey16/slopec16
py17=slopey17/slopec17
py18=slopey18/slopec18
py19=slopey19/slopec19
py20=slopey20/slopec20
py21=slopey21/slopec21
py22=slopey22/slopec22
py23=slopey23/slopec23
py24=slopey24/slopec24
py25=slopey25/slopec25
py26=slopey26/slopec26
py27=slopey27/slopec27
py28=slopey28/slopec28
py29=slopey29/slopec29
py30=slopey30/slopec30
py31=slopey31/slopec31
py32=slopey32/slopec32
py33=slopey33/slopec33
py34=slopey34/slopec34
py35=slopey35/slopec35
py36=slopey36/slopec36

```

```

ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py18,py19,py20,py21,py22,py23,py24,py25,py26,py27,py28,py29]
ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py27],[py31,py32,py33],[py34,py35,py36]]
ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29]]

```

```

In [155]: X = np.arange(18)
plt.figure(figsize=(10,7))
plt.title(r'$\rho$ Glucosa 0.4%')
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+ ' '+r'$\rho$r',zorder=1)
plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+ ' '+r'$\rho$y',zorder=2)
plt.bar(X[1]-0.25,ro_rfp[1],color='coral',width=0.25,zorder=3)

```

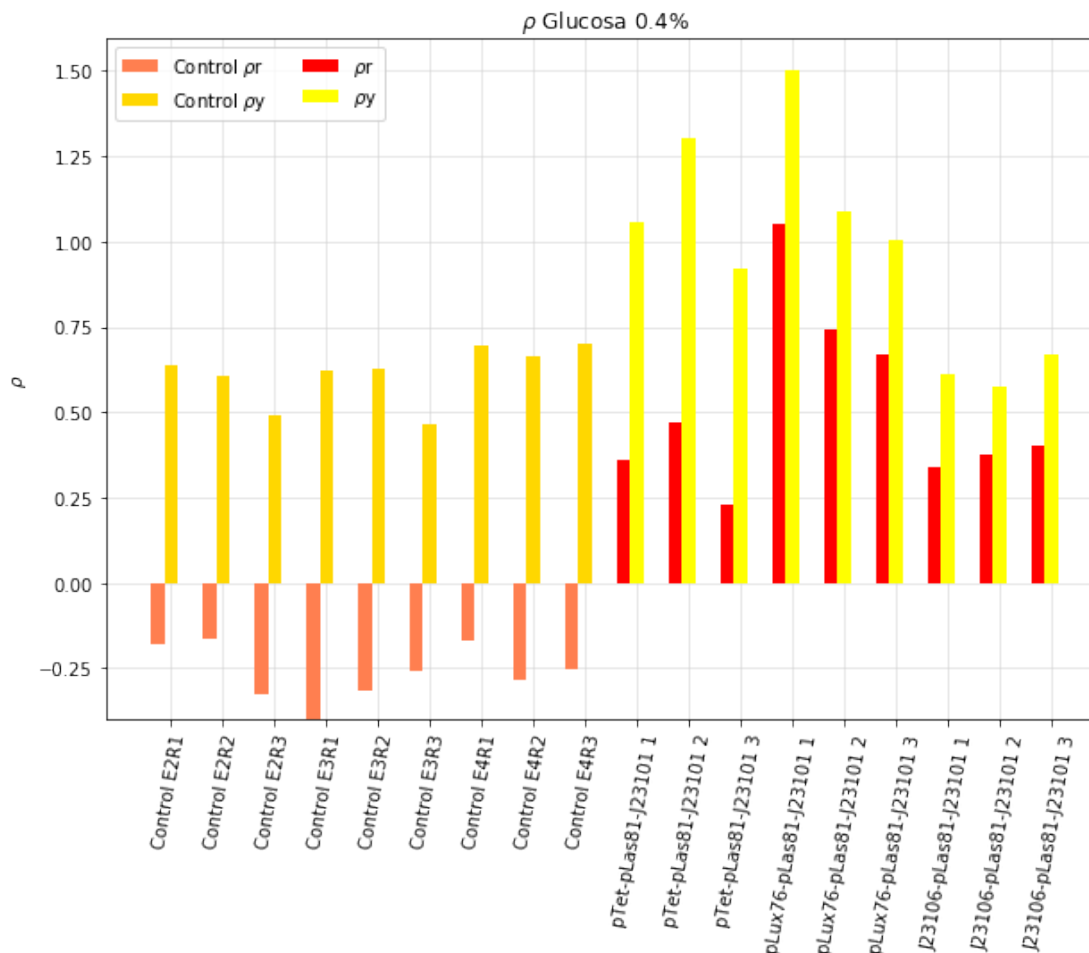
```

plt.bar(X[1]+0.00,ro_yfp[1],color='gold',width=0.25,zorder=3)
plt.bar(X[2]-0.25,ro_rfp[2],color='coral',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ro_yfp[2],color='gold',width=0.25,zorder=3)
plt.bar(X[3]-0.25,ro_rfp[6],color='coral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[6],color='gold',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[7],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[7],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[8],color='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[8],color='gold',width=0.25,zorder=3)
plt.bar(X[6]-0.25,ro_rfp[12],color='coral',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ro_yfp[12],color='gold',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[13],color='coral',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[13],color='gold',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[14],color='coral',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[14],color='gold',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[18],color='r',width=0.25,label=r'$\rho$r',zorder=3)
plt.bar(X[9]+0.00,ro_yfp[18],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
plt.bar(X[10]-0.25,ro_rfp[19],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[19],color='yellow',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[20],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[20],color='yellow',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[24],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[24],color='yellow',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[25],color='r',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[25],color='yellow',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[26],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[26],color='yellow',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[30],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[30],color='yellow',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[31],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[31],color='yellow',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[32],color='r',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[32],color='yellow',width=0.25,zorder=3)

plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2','Control E3R3'])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)

```

Out[155]: <matplotlib.legend.Legend at 0x1789c1209e8>



```
In [156]: X = np.arange(18)
plt.figure(figsize=(10,7))
plt.title(r'$\rho$ Glicerol 0.2%')
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[3],color='coral',width=0.25,label= 'Control'+ ' '+r'$\rho$r',zorder=3)
plt.bar(X[0]+0.00,ro_yfp[3],color='gold',width=0.25,label= 'Control'+ ' '+r'$\rho$y',zorder=3)
plt.bar(X[1]-0.25,ro_rfp[4],color='coral',width=0.25,zorder=3)
plt.bar(X[1]+0.00,ro_yfp[4],color='gold',width=0.25,zorder=3)
plt.bar(X[2]-0.25,ro_rfp[5],color='coral',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ro_yfp[5],color='gold',width=0.25,zorder=3)
plt.bar(X[3]-0.25,ro_rfp[9],color='coral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[9],color='gold',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[10],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[10],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[11],color='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[11],color='gold',width=0.25,zorder=3)
plt.bar(X[6]-0.25,ro_rfp[15],color='coral',width=0.25,zorder=3)
```

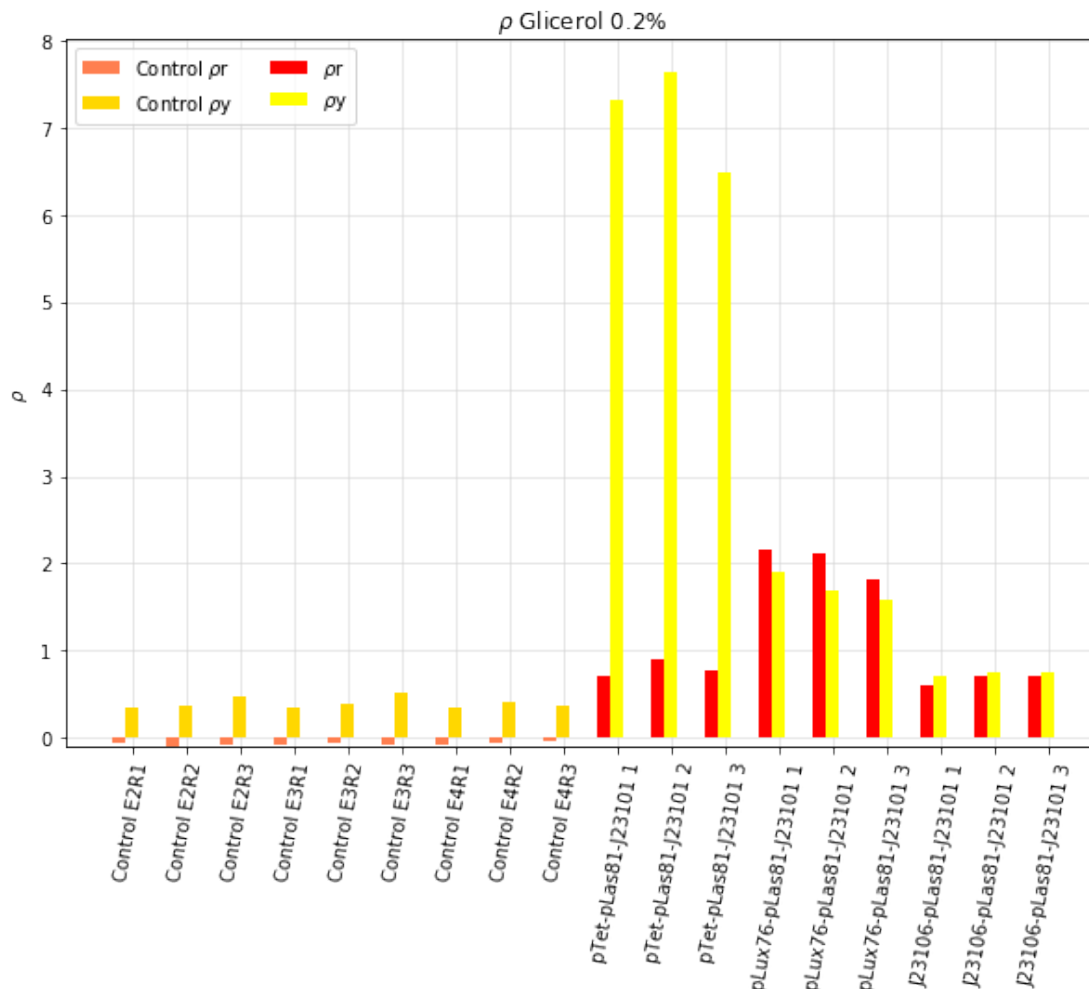
```

plt.bar(X[6]+0.00,ro_yfp[15],color='gold',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[16],color='coral',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[16],color='gold',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[17],color='coral',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[17],color='gold',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[21],color='r',width=0.25,label=r'$\rho$r',zorder=3)
plt.bar(X[9]+0.00,ro_yfp[21],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
plt.bar(X[10]-0.25,ro_rfp[22],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[22],color='yellow',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[23],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[23],color='yellow',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[27],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[27],color='yellow',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[28],color='r',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[28],color='yellow',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[29],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[29],color='yellow',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[33],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[33],color='yellow',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[34],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[34],color='yellow',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[35],color='r',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[35],color='yellow',width=0.25,zorder=3)

plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2','Control E3R3'])
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)

```

Out[156]: <matplotlib.legend.Legend at 0x1789cdf6588>



```
In [157]: X = np.arange(36)
plt.figure(figsize=(20,10))
plt.title(r'$\rho$', fontsize=15.0)
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+ ' '+r'$\rho$r Glu')
plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+ ' '+r'$\rho$y Glu')
plt.bar(X[1]-0.25,ro_rfp[3],color='lightcoral',width=0.25,label= 'Control'+ ' '+r'$\rho$r Glu')
plt.bar(X[1]+0.00,ro_yfp[3],color='palegreen',width=0.25,label= 'Control'+ ' '+r'$\rho$y Glu')
plt.bar(X[2]-0.25,ro_rfp[1],color='coral',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ro_yfp[1],color='gold',width=0.25,zorder=3)
plt.bar(X[3]-0.25,ro_rfp[4],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[4],color='palegreen',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[2],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[2],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[5],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[5],color='palegreen',width=0.25,zorder=3)
```

```

plt.bar(X[6]-0.25,ro_rfp[6],color='coral',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ro_yfp[6],color='gold',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[9],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[9],color='palegreen',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[7],color='coral',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[7],color='gold',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[10],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ro_yfp[10],color='palegreen',width=0.25,zorder=3)
plt.bar(X[10]-0.25,ro_rfp[8],color='coral',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[8],color='gold',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[11],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[11],color='palegreen',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[12],color='coral',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[12],color='gold',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[15],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[15],color='palegreen',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[13],color='coral',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[13],color='gold',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[16],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[16],color='palegreen',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[14],color='coral',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[14],color='gold',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[17],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[17],color='palegreen',width=0.25,zorder=3)

```

```

plt.bar(X[18]-0.25,ro_rfp[18],color='r',width=0.25,label=r'$\rho_r$ Glucosa',zorder=3)
plt.bar(X[18]+0.00,ro_yfp[18],color='yellow',width=0.25,label=r'$\rho_y$ Glucosa',zorde
plt.bar(X[19]-0.25,ro_rfp[21],color='firebrick',width=0.25,label=r'$\rho_r$ Glicerol',z
plt.bar(X[19]+0.00,ro_yfp[21],color='khaki',width=0.25,label=r'$\rho_y$ Glicerol',zorde
plt.bar(X[20]-0.25,ro_rfp[19],color='r',width=0.25,zorder=3)
plt.bar(X[20]+0.00,ro_yfp[19],color='yellow',width=0.25,zorder=3)
plt.bar(X[21]-0.25,ro_rfp[22],color='firebrick',width=0.25,zorder=3)
plt.bar(X[21]+0.00,ro_yfp[22],color='khaki',width=0.25,zorder=3)
plt.bar(X[22]-0.25,ro_rfp[20],color='r',width=0.25,zorder=3)
plt.bar(X[22]+0.00,ro_yfp[20],color='yellow',width=0.25,zorder=3)
plt.bar(X[23]-0.25,ro_rfp[23],color='firebrick',width=0.25,zorder=3)
plt.bar(X[23]+0.00,ro_yfp[23],color='khaki',width=0.25,zorder=3)
plt.bar(X[24]-0.25,ro_rfp[24],color='r',width=0.25,zorder=3)
plt.bar(X[24]+0.00,ro_yfp[24],color='yellow',width=0.25,zorder=3)
plt.bar(X[25]-0.25,ro_rfp[27],color='firebrick',width=0.25,zorder=3)
plt.bar(X[25]+0.00,ro_yfp[27],color='khaki',width=0.25,zorder=3)
plt.bar(X[26]-0.25,ro_rfp[25],color='r',width=0.25,zorder=3)
plt.bar(X[26]+0.00,ro_yfp[25],color='yellow',width=0.25,zorder=3)
plt.bar(X[27]-0.25,ro_rfp[28],color='firebrick',width=0.25,zorder=3)
plt.bar(X[27]+0.00,ro_yfp[28],color='khaki',width=0.25,zorder=3)
plt.bar(X[28]-0.25,ro_rfp[26],color='r',width=0.25,zorder=3)
plt.bar(X[28]+0.00,ro_yfp[26],color='yellow',width=0.25,zorder=3)

```

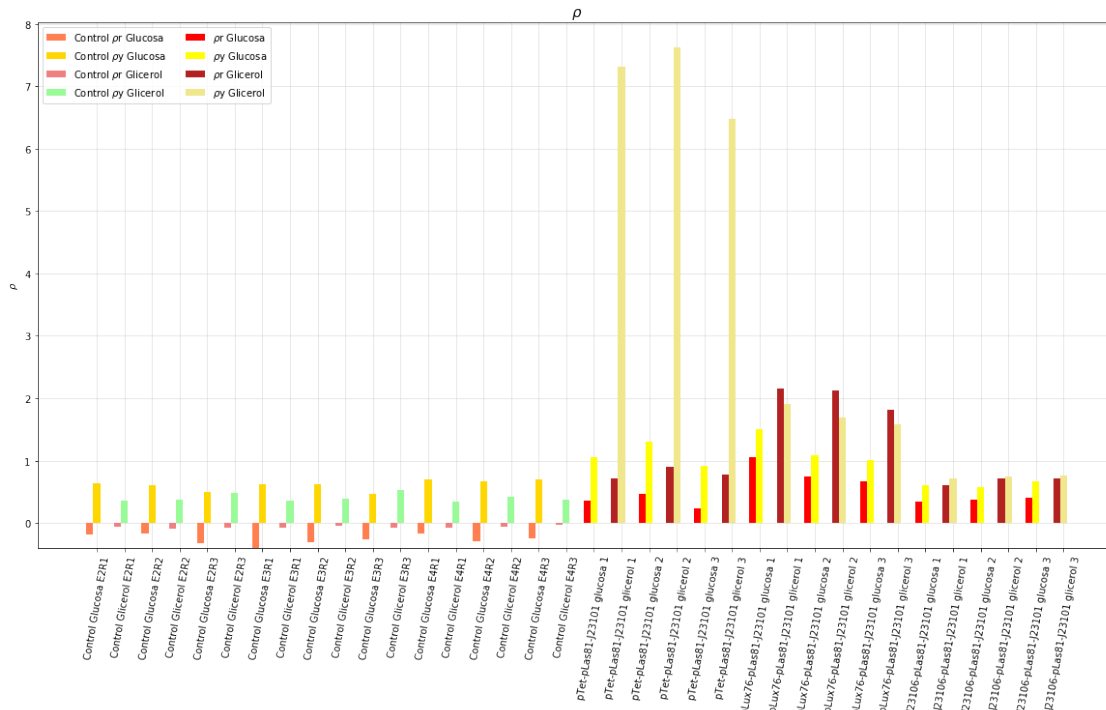


```

plt.bar(X[29]-0.25,ro_rfp[29],color='firebrick',width=0.25,zorder=3)
plt.bar(X[29]+0.00,ro_yfp[29],color='khaki',width=0.25,zorder=3)
plt.bar(X[30]-0.25,ro_rfp[30],color='r',width=0.25,zorder=3)
plt.bar(X[30]+0.00,ro_yfp[30],color='yellow',width=0.25,zorder=3)
plt.bar(X[31]-0.25,ro_rfp[33],color='firebrick',width=0.25,zorder=3)
plt.bar(X[31]+0.00,ro_yfp[33],color='khaki',width=0.25,zorder=3)
plt.bar(X[32]-0.25,ro_rfp[31],color='r',width=0.25,zorder=3)
plt.bar(X[32]+0.00,ro_yfp[31],color='yellow',width=0.25,zorder=3)
plt.bar(X[33]-0.25,ro_rfp[34],color='firebrick',width=0.25,zorder=3)
plt.bar(X[33]+0.00,ro_yfp[34],color='khaki',width=0.25,zorder=3)
plt.bar(X[34]-0.25,ro_rfp[32],color='r',width=0.25,zorder=3)
plt.bar(X[34]+0.00,ro_yfp[32],color='yellow',width=0.25,zorder=3)
plt.bar(X[35]-0.25,ro_rfp[35],color='firebrick',width=0.25,zorder=3)
plt.bar(X[35]+0.00,ro_yfp[35],color='khaki',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa E2R1','Control Glicerol E2R1','Control Glucosa E2R2',
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)

```

Out[157]: <matplotlib.legend.Legend at 0x1789ee239e8>



```

In [158]: ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr27],
ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py27],

```

```

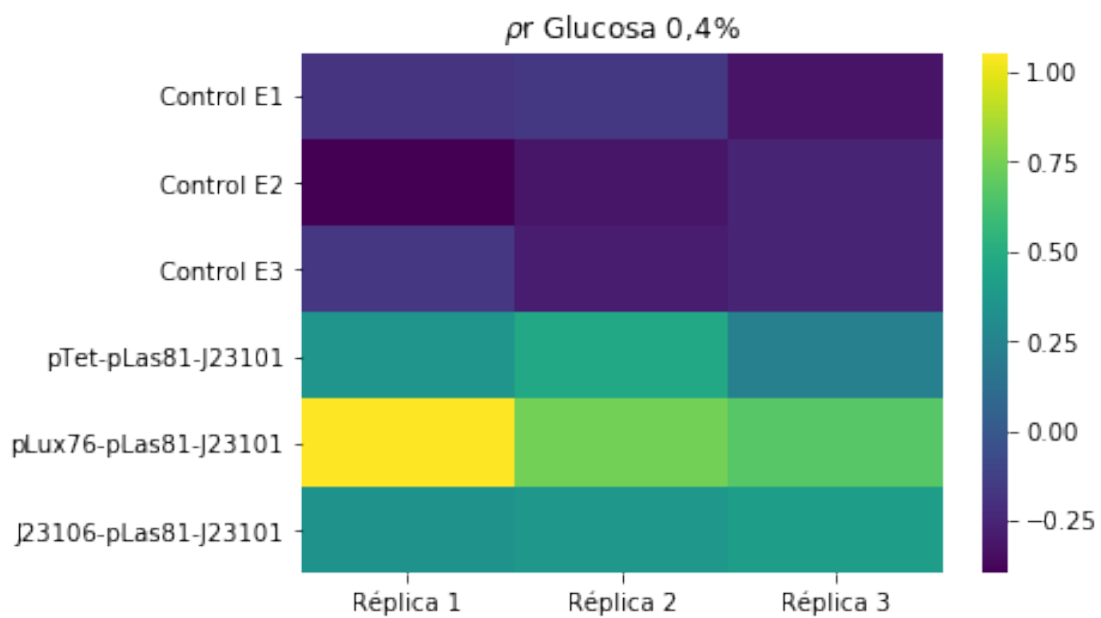
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101']

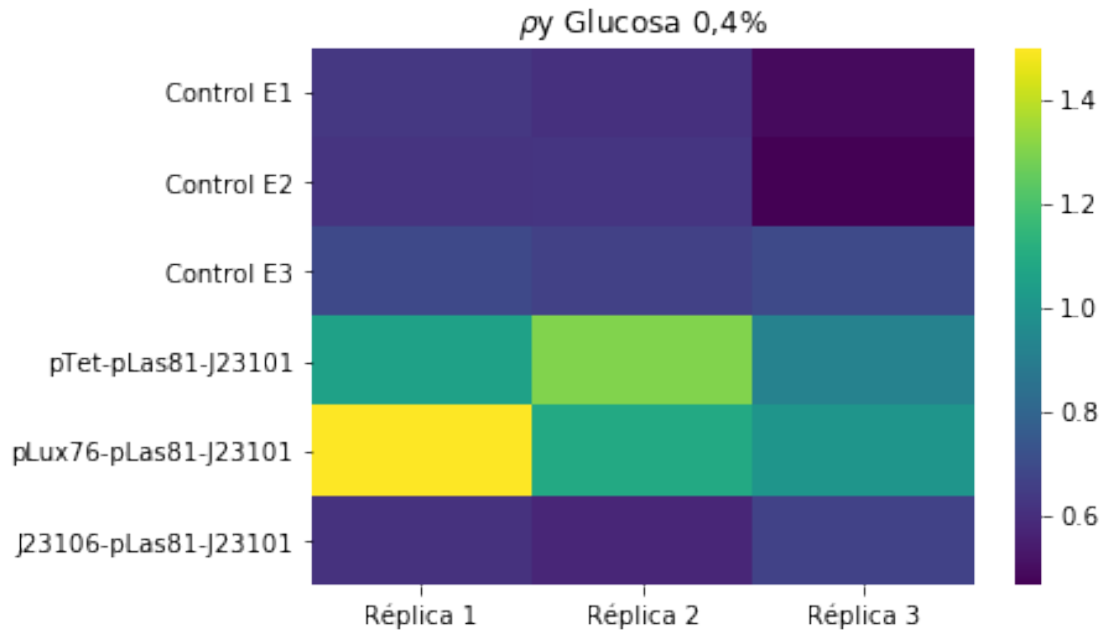
plt.figure()
plt.title(r'$\rho$ Glucosa 0,4%')
sns.heatmap(ro_rfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

plt.figure()
plt.title(r'$\rho$ Glucosa 0,4%')
sns.heatmap(ro_yfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

```

Out[158]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1789b164080>





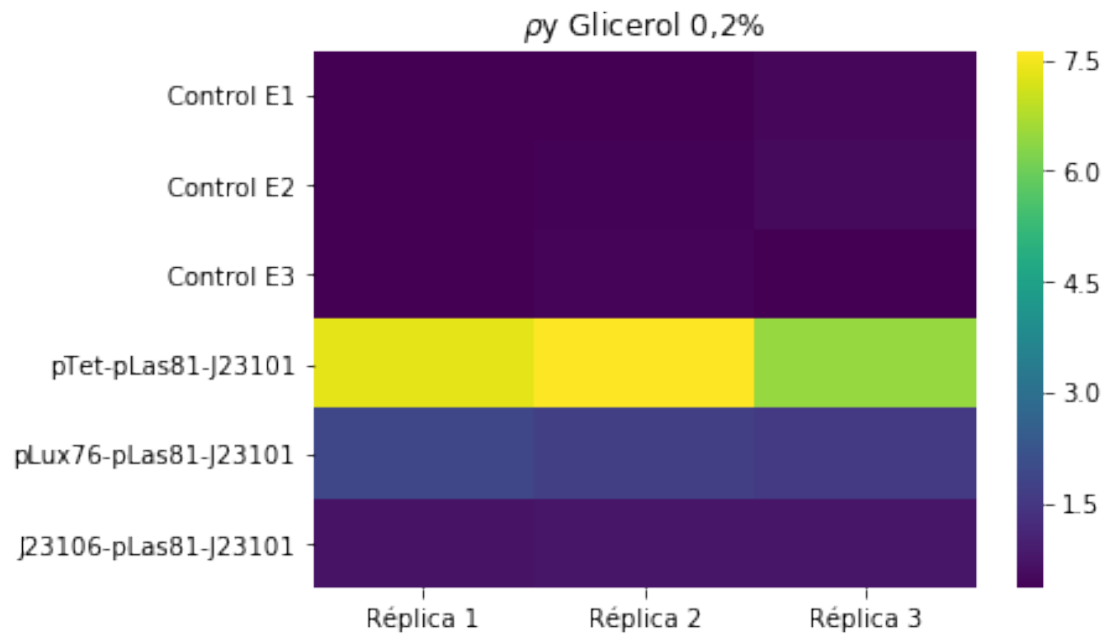
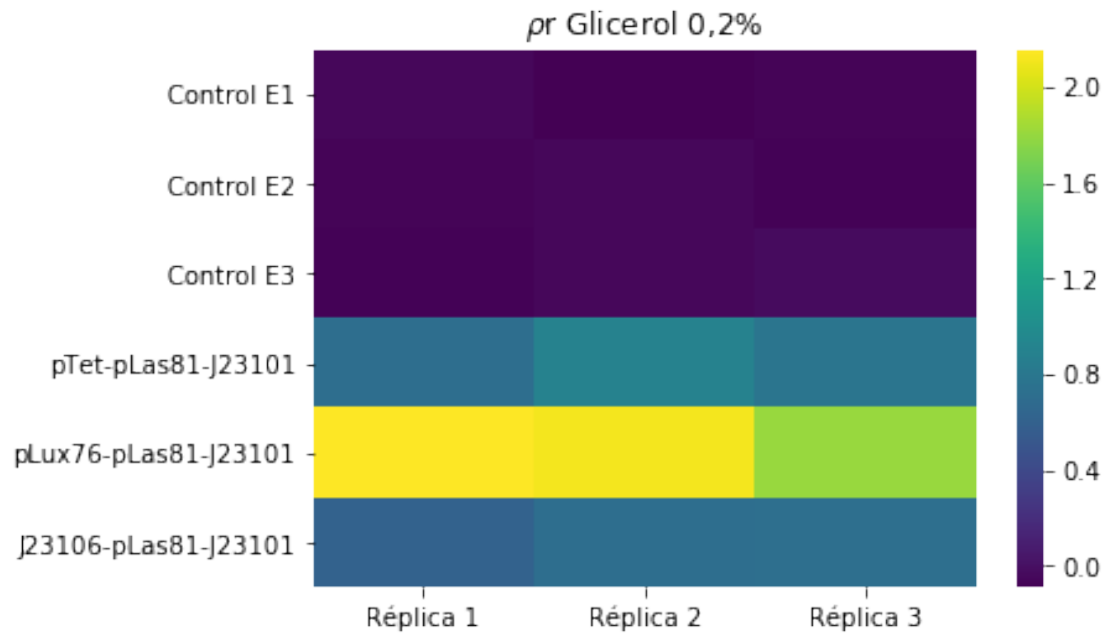
```
In [161]: ro_rfpqli=[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29]
          ro_yfpqli=[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29]

          xlabel=['Réplica 1','Réplica 2','Réplica 3']
          ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101']

          plt.figure()
          plt.title(r'$\rho_r$ Glicerol 0,2%')
          sns.heatmap(ro_rfpqli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

          plt.figure()
          plt.title(r'$\rho_y$ Glicerol 0,2%')
          sns.heatmap(ro_yfpqli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
```

```
Out[161]: <matplotlib.axes._subplots.AxesSubplot at 0x1789ec157f0>
```



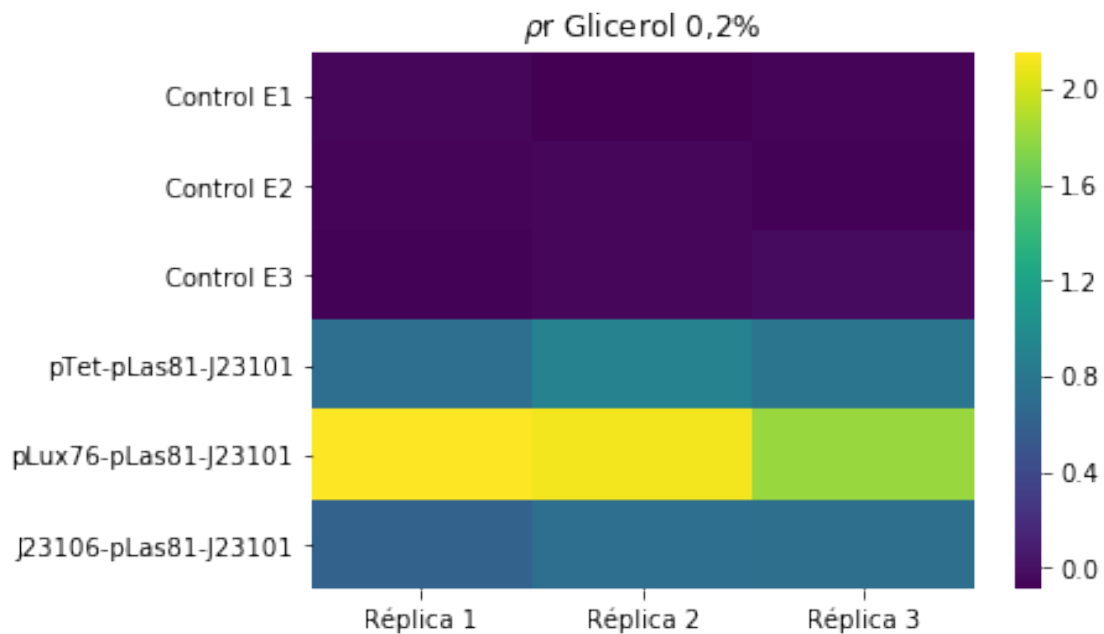
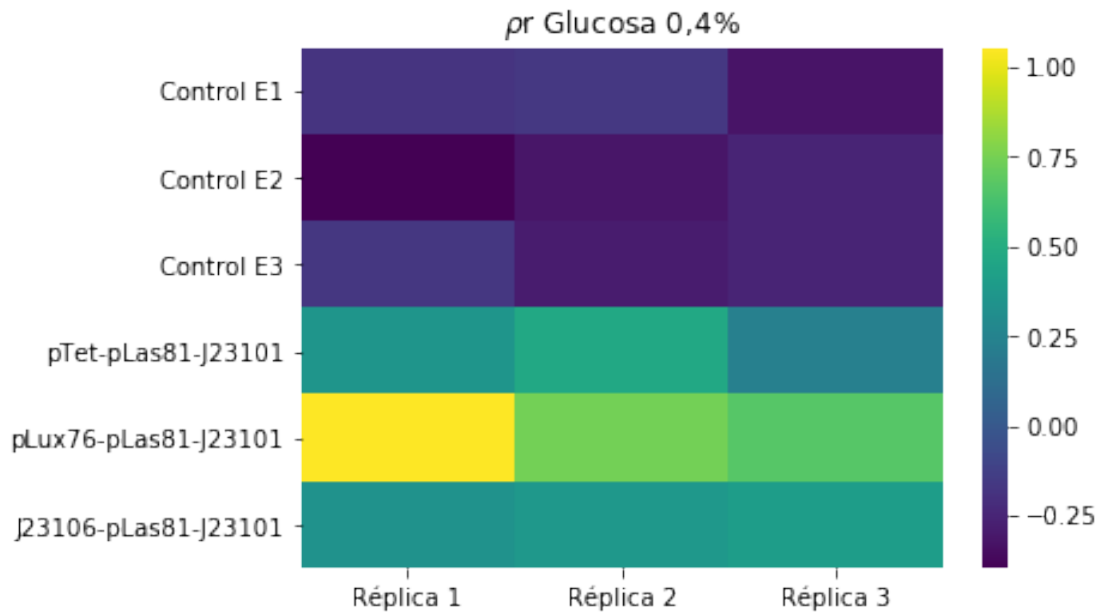
```
In [162]: ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101']
          xlabel=['Réplica 1','Réplica 2','Réplica 3']
```

```
plt.figure()
```

```
plt.title(r'$\rho$ Glucosa 0,4%')
sns.heatmap(ro_rfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

plt.figure()
plt.title(r'$\rho$ Glicerol 0,2%')
sns.heatmap(ro_rfpgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
```

Out[162]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1789b155a20>

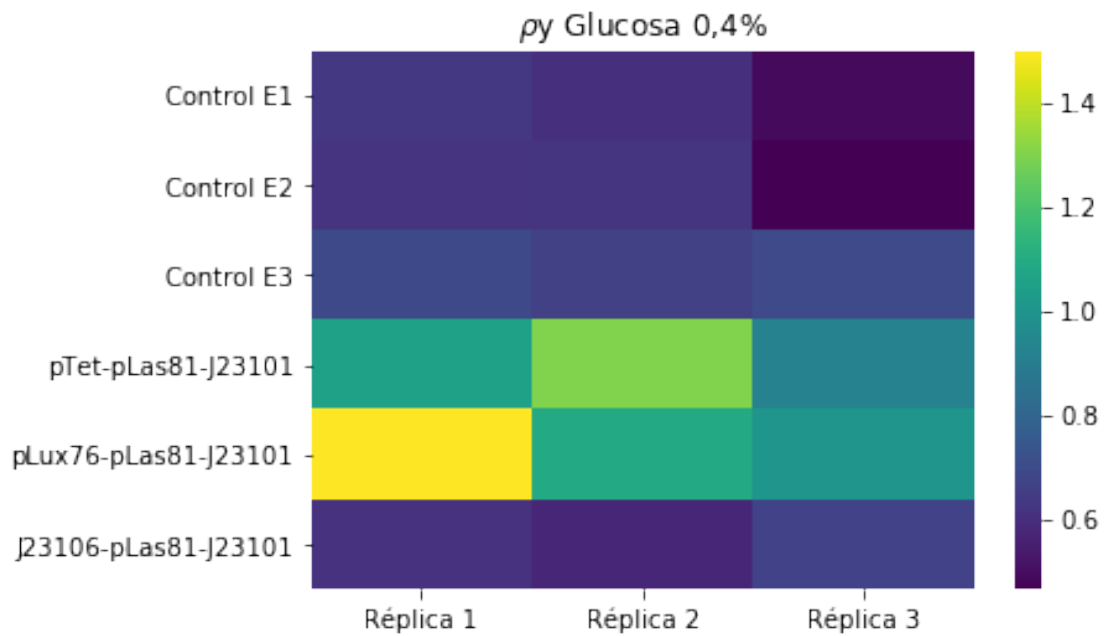


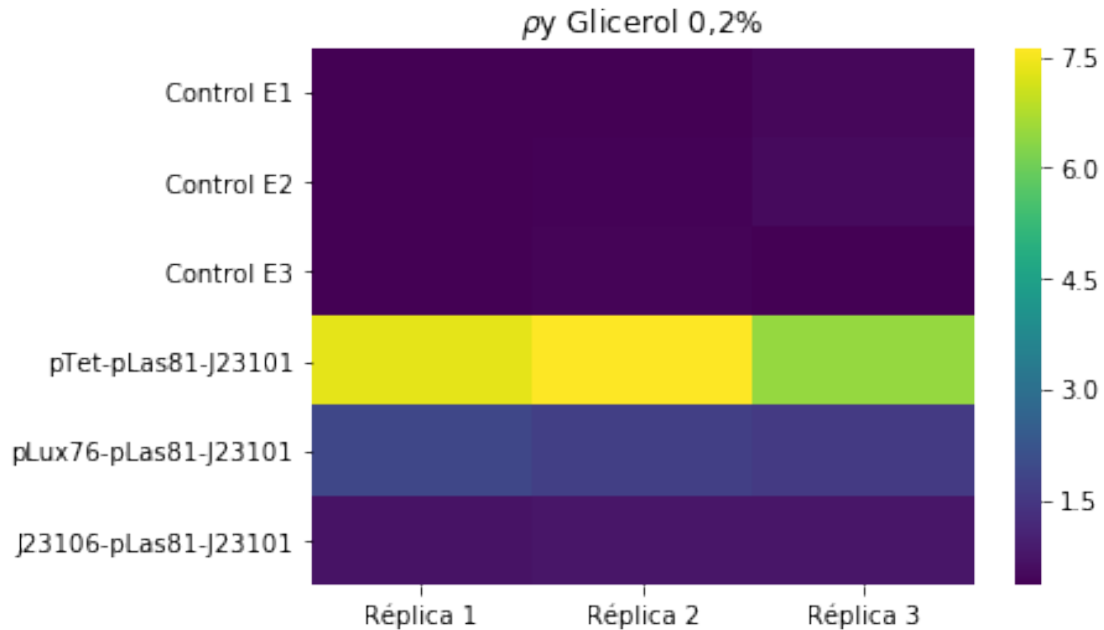
```
In [163]: ylabel=['Control E1','Control E2','Control E3','pTet-pLas81-J23101','pLux76-pLas81-J23101']
          xlabel=['Réplica 1','Réplica 2','Réplica 3']
```

```
plt.figure()
plt.title(r'$\rho$ Glucosa 0,4%')
sns.heatmap(ro_yfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
```

```
plt.figure()
plt.title(r'$\rho$ Glicerol 0,2%')
sns.heatmap(ro_yfpqli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
```

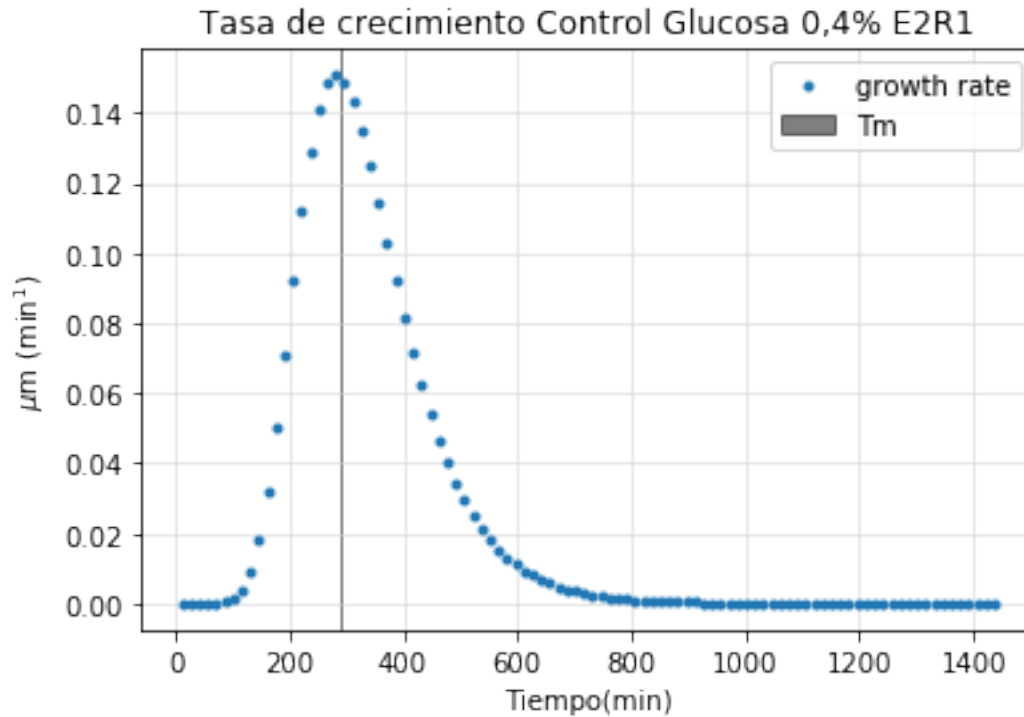
```
Out[163]: <matplotlib.axes._subplots.AxesSubplot at 0x178968526a0>
```





```
In [164]: #tasa de crecimiento
ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(11-tt))+1))))
#Con diff
dy1=(np.diff(ye1))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy1,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[164]: <matplotlib.legend.Legend at 0x1789f162c18>
```

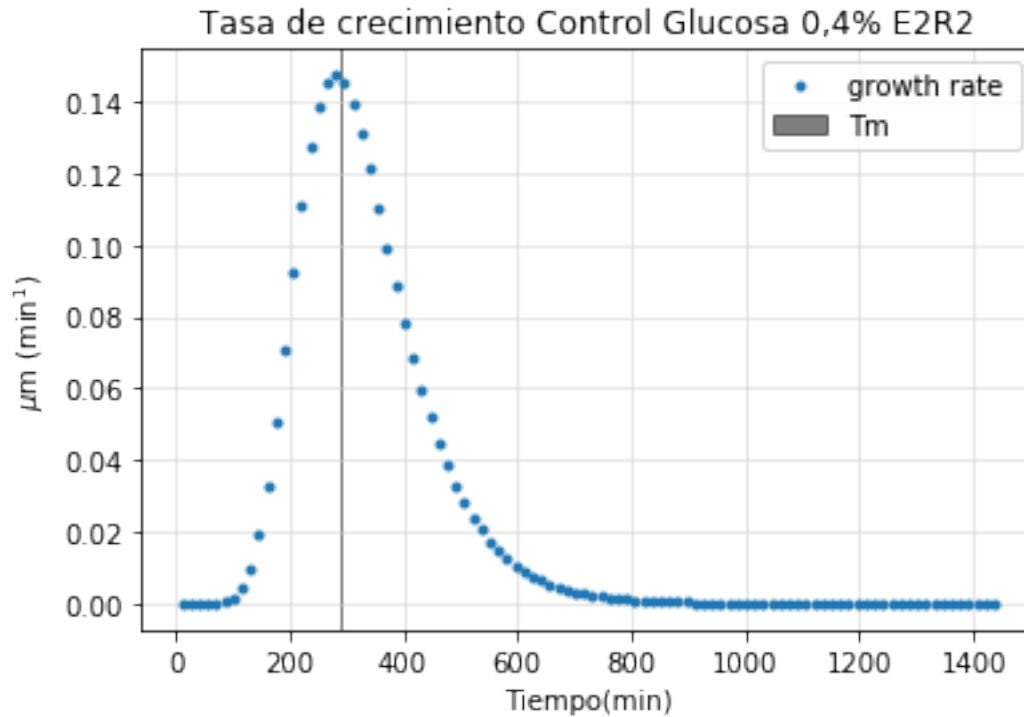


In [165]: *#tasa de crecimiento*

```
ye2=((A2*np.exp(-np.exp((((um2*np.exp(1))/A2)*(12-tt))+1))))
#Con diff
dy2=(np.diff(ye2))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm2,tm2, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy2,'.',label='growth rate ')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

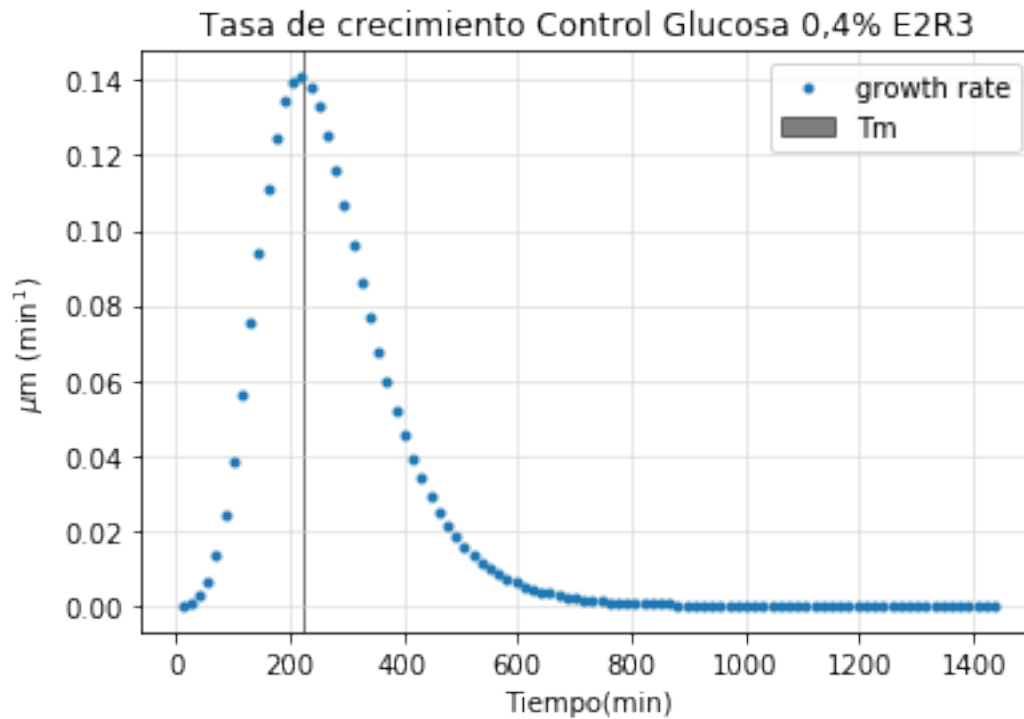
Out[165]: <matplotlib.legend.Legend at 0x1789f4a3cf8>





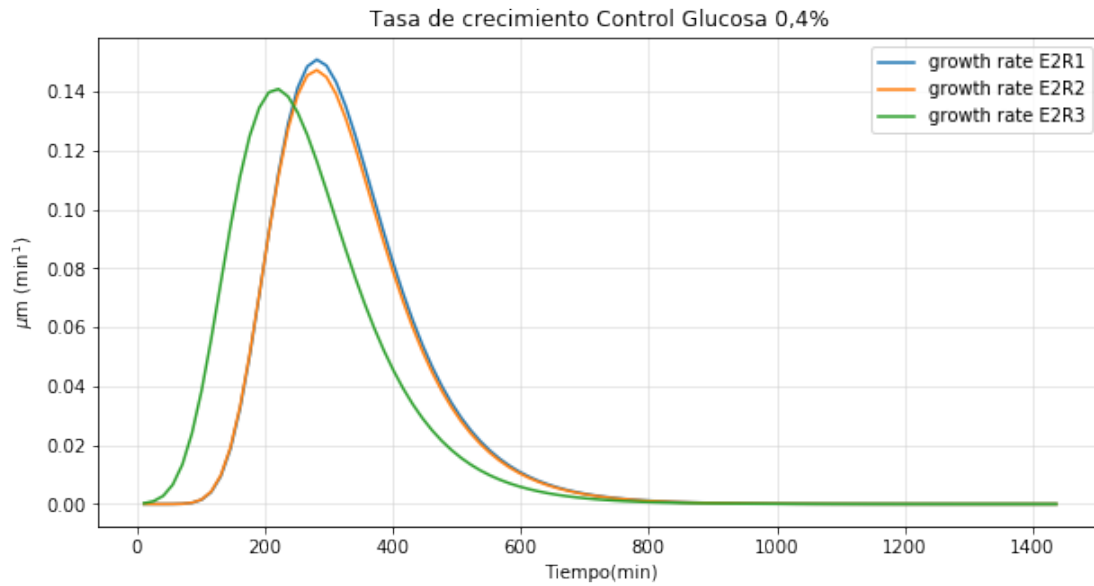
```
In [166]: #tasa de crecimiento
ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(13-tt))+1))))
#Con diff
dy3=(np.diff(ye3))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm3,tm3, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy3,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[166]: <matplotlib.legend.Legend at 0x1789f578a90>
```



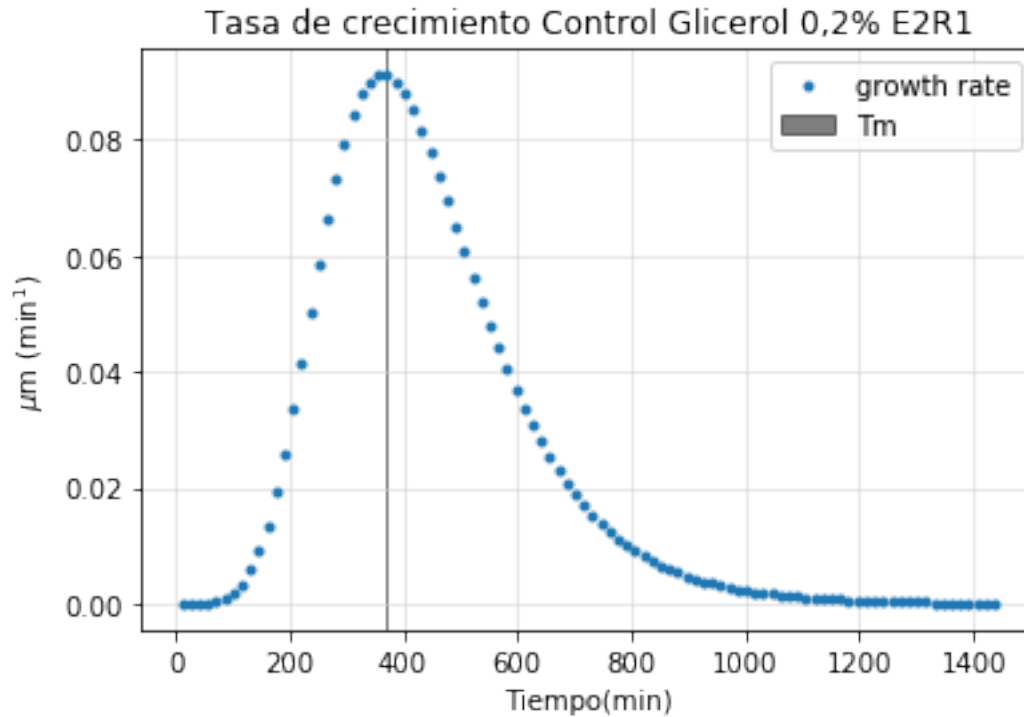
```
In [167]: #Tasas control réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy1,label='growth rate E2R1')
plt.plot(tt[:-1],dy2,label='growth rate E2R2')
plt.plot(tt[:-1],dy3,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[167]: <matplotlib.legend.Legend at 0x1789f649b38>
```



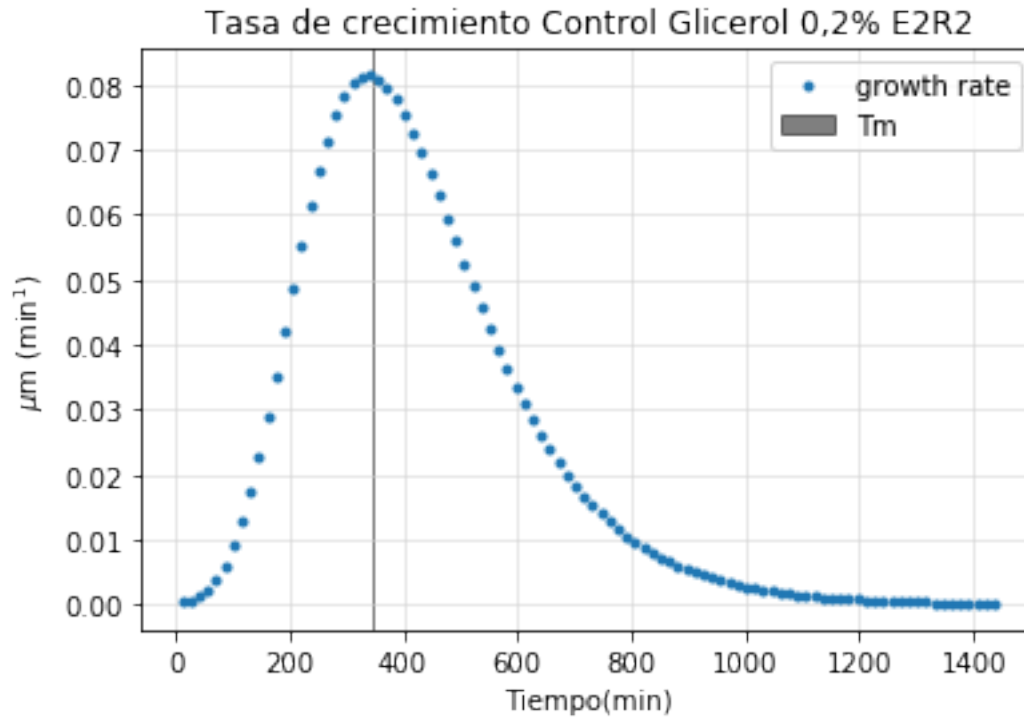
```
In [168]: #tasa de crecimiento
ye4=((A4*np.exp(-np.exp((((um4*np.exp(1))/A4)*(14-tt))+1))))
#Con diff
dy4=(np.diff(ye4))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm4,tm4, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy4,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[168]: <matplotlib.legend.Legend at 0x1789f72df28>
```



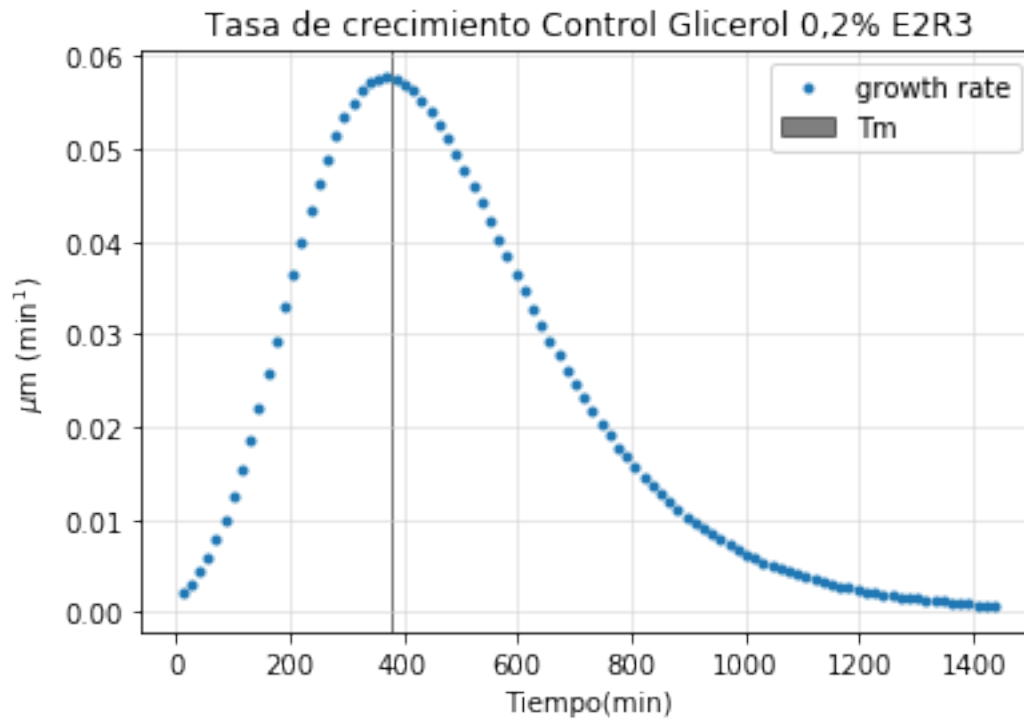
```
In [169]: #tasa de crecimiento
ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(15-tt))+1))))
#Con diff
dy5=(np.diff(ye5))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm5,tm5, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy5,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[169]: <matplotlib.legend.Legend at 0x1789f7fbcf8>
```



```
In [170]: #tasa de crecimiento
ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(16-tt))+1))))
#Con diff
dy6=(np.diff(ye6))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm6,tm6, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy6,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[170]: <matplotlib.legend.Legend at 0x1789f8b8ac8>
```



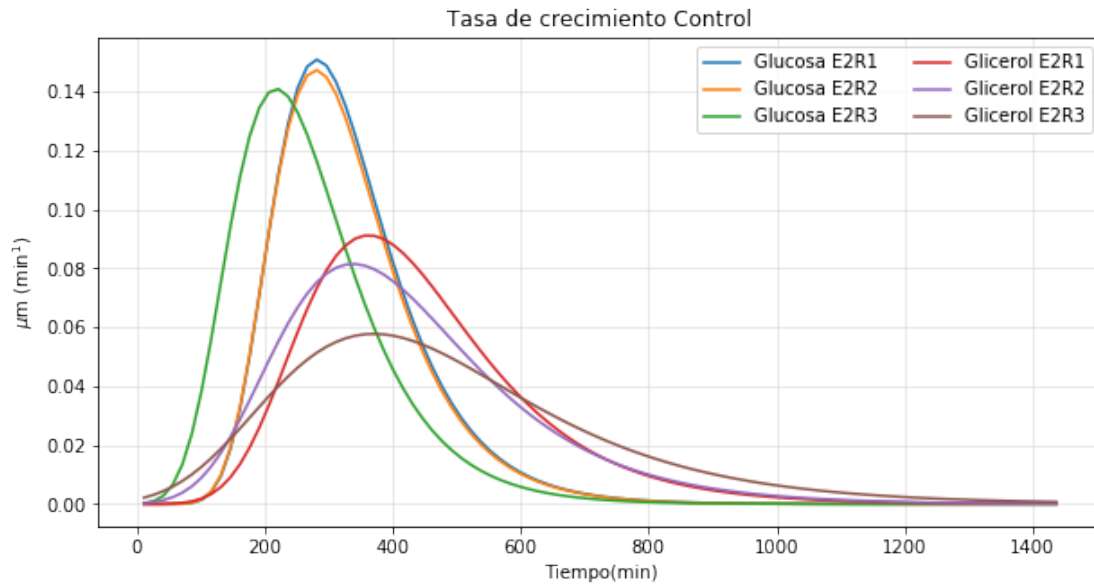
```
In [171]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^{-1}$)')
plt.plot(tt[:-1],dy4,label='growth rate E2R1')
plt.plot(tt[:-1],dy5,label='growth rate E2R2')
plt.plot(tt[:-1],dy6,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[171]: <matplotlib.legend.Legend at 0x1789f987c50>
```



```
In [172]: #Tasas control réplicas controles
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^{-1}$)')
plt.plot(tt[:-1],dy1,label='Glucosa E2R1')
plt.plot(tt[:-1],dy2,label='Glucosa E2R2')
plt.plot(tt[:-1],dy3,label='Glucosa E2R3')
plt.plot(tt[:-1],dy4,label='Glicerol E2R1')
plt.plot(tt[:-1],dy5,label='Glicerol E2R2')
plt.plot(tt[:-1],dy6,label='Glicerol E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

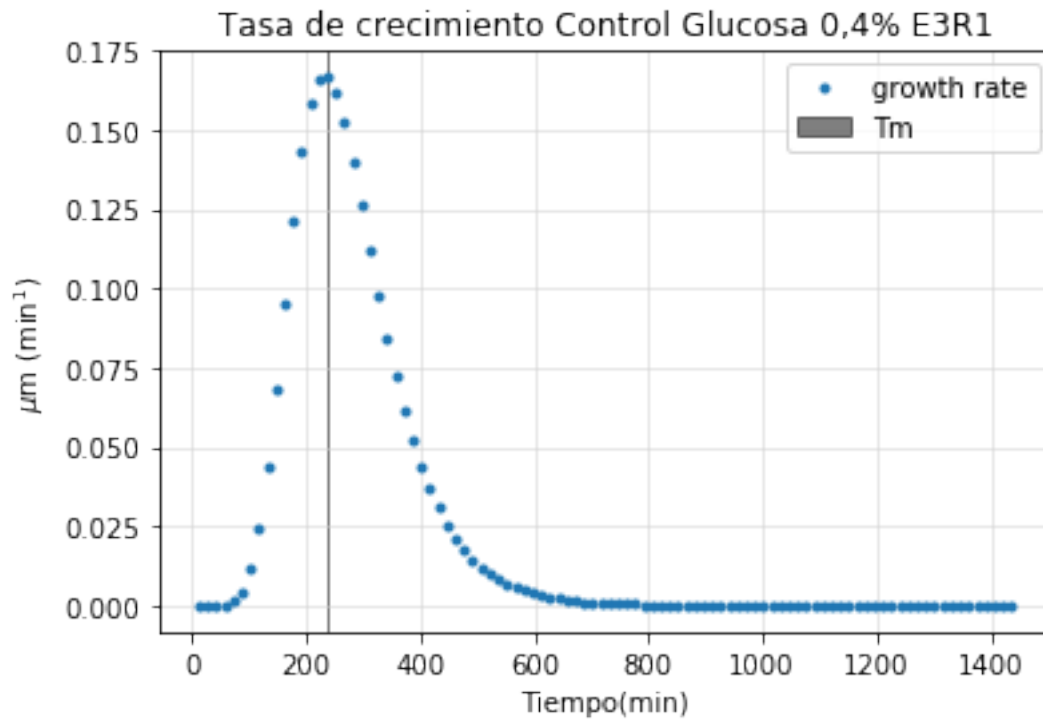
```
Out[172]: <matplotlib.legend.Legend at 0x1789fa76860>
```



```
In [174]: #tasa de crecimiento
ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(17-tt))+1))))
#Con diff
dy7=(np.diff(ye7))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm7,tm7, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy7,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[174]: <matplotlib.legend.Legend at 0x178a0f8bb70>

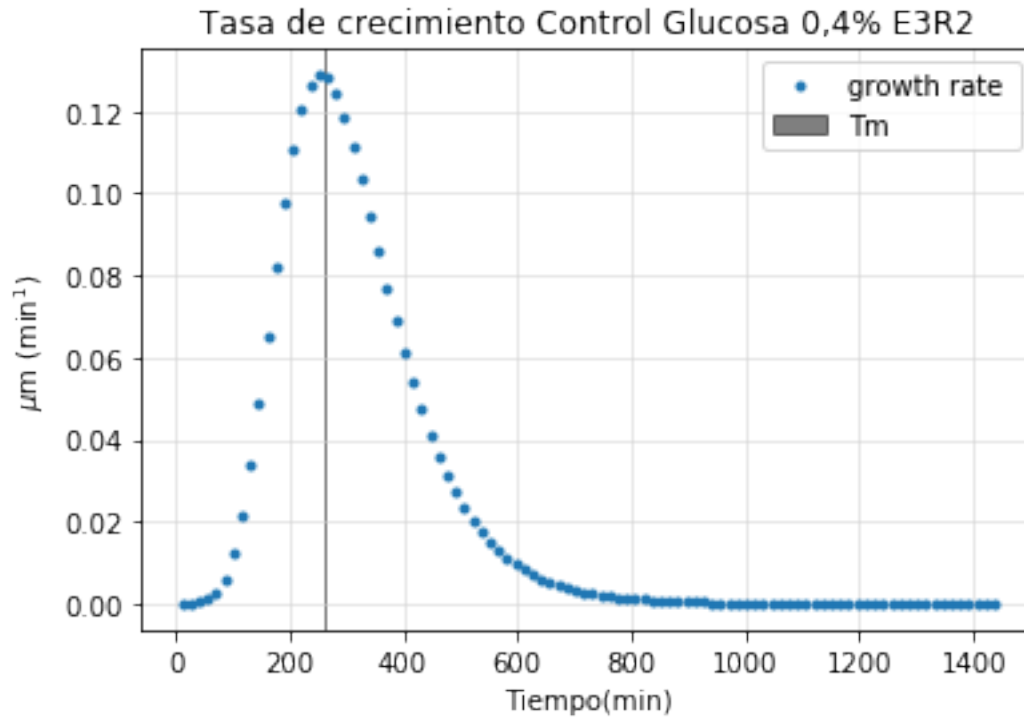




In [175]: *#tasa de crecimiento*

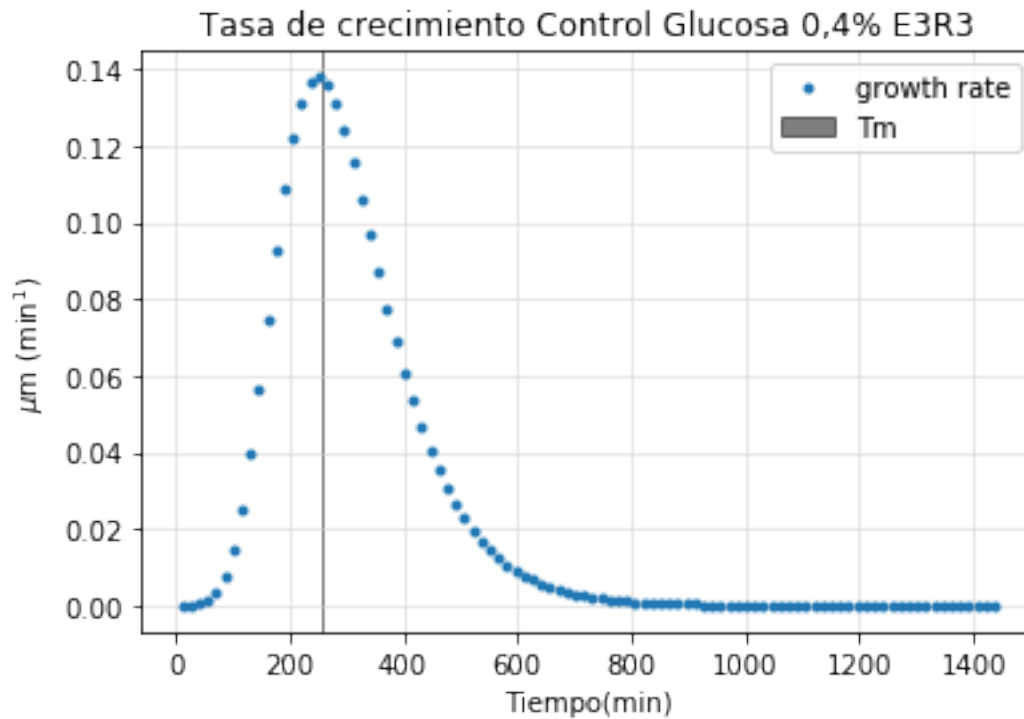
```
ye8=((A8*np.exp(-np.exp((((um8*np.exp(1))/A8)*(18-tt))+1))))
#Con diff
dy8=(np.diff(ye8))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm8,tm8, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy8,'.',label='growth rate ')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[175]: <matplotlib.legend.Legend at 0x178a105ed68>



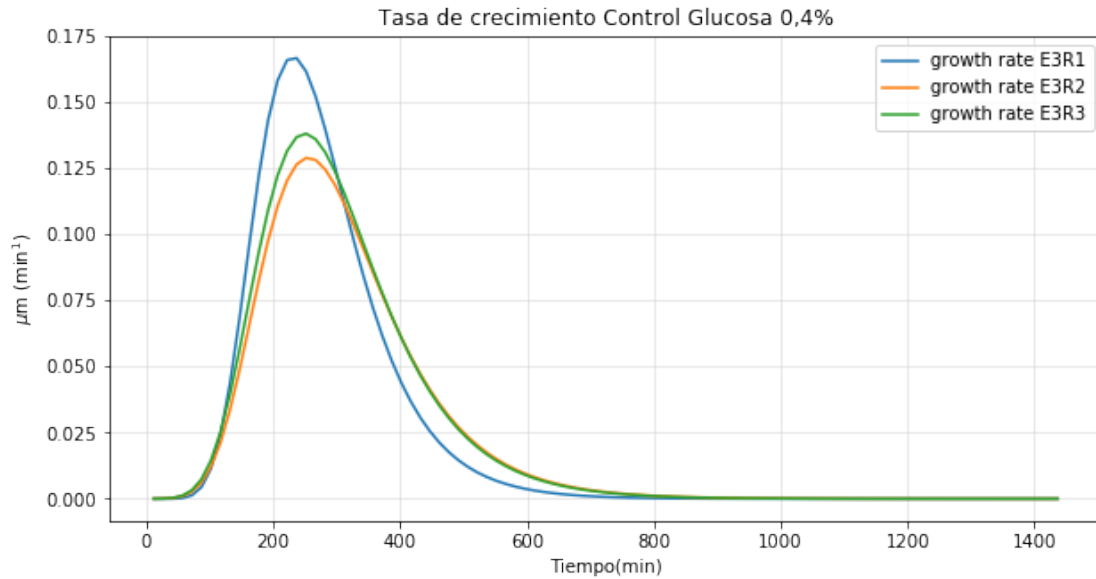
```
In [176]: #tasa de crecimiento
ye9=((A9*np.exp(-np.exp((((um9*np.exp(1))/A9)*(19-tt))+1))))
#Con diff
dy9=(np.diff(ye9))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm9,tm9, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy9,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[176]: <matplotlib.legend.Legend at 0x178a112fbe0>
```



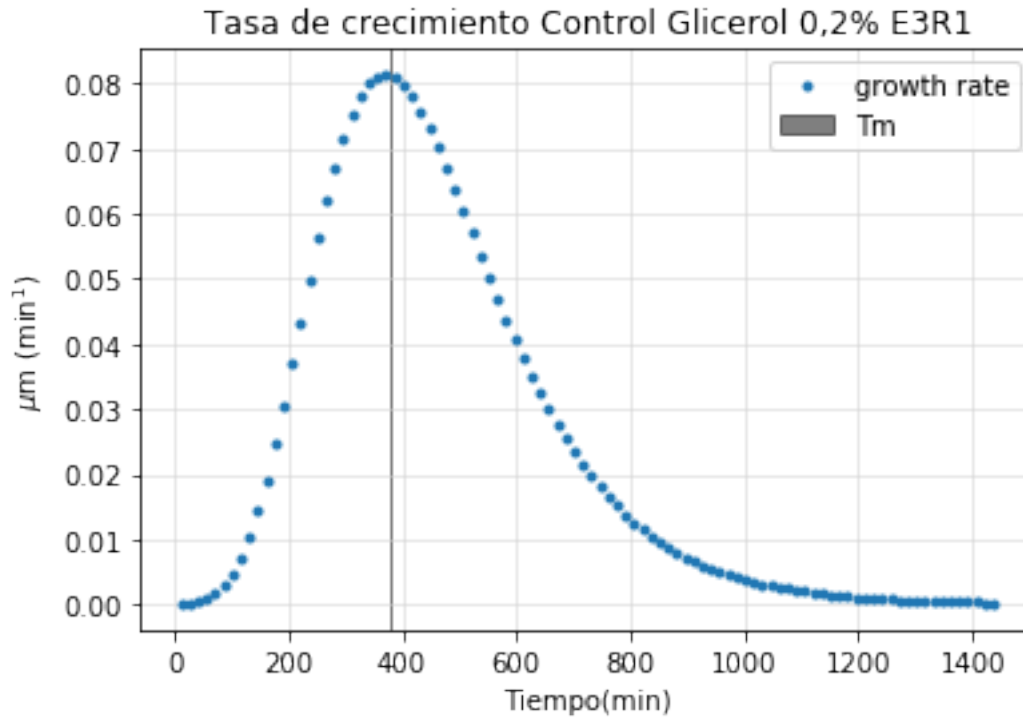
```
In [177]: #Tasas control réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^{-1}$)')
plt.plot(tt[:-1],dy7,label='growth rate E3R1')
plt.plot(tt[:-1],dy8,label='growth rate E3R2')
plt.plot(tt[:-1],dy9,label='growth rate E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[177]: <matplotlib.legend.Legend at 0x178a120ac88>
```



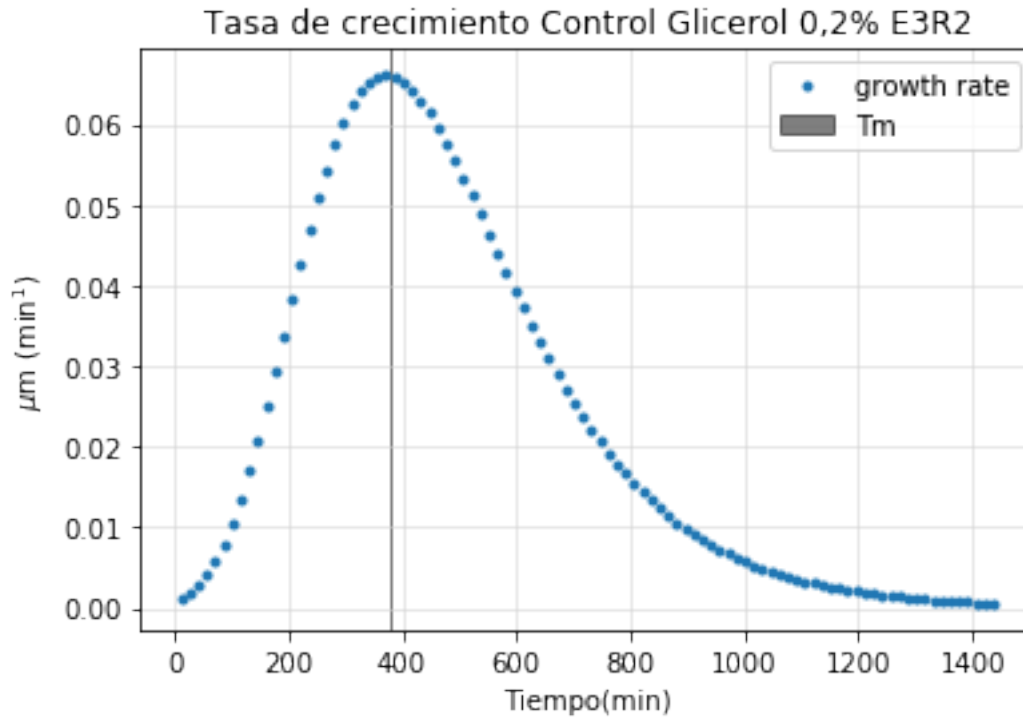
```
In [178]: #tasa de crecimiento
ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(110-tt))+1))))
#Con diff
dy10=(np.diff(ye10))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm10,tm10, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy10,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[178]: <matplotlib.legend.Legend at 0x178a12ecb70>
```



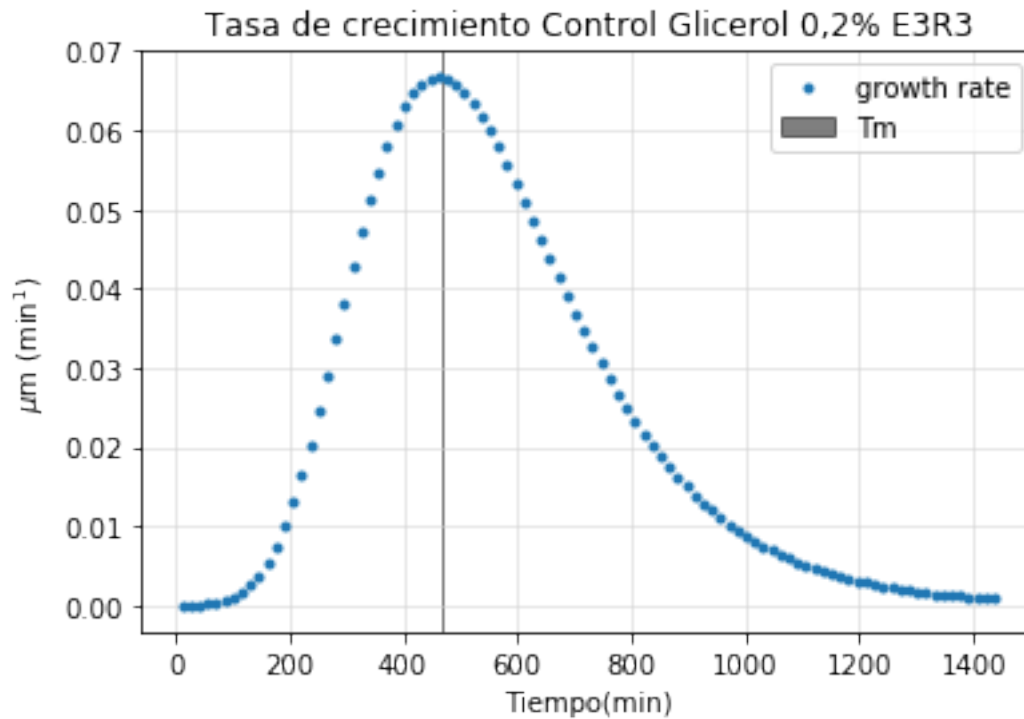
```
In [179]: #tasa de crecimiento
ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(111-tt))+1))))
#Con diff
dy11=(np.diff(ye11))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm11,tm11, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy11,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[179]: <matplotlib.legend.Legend at 0x178a2379fd0>
```



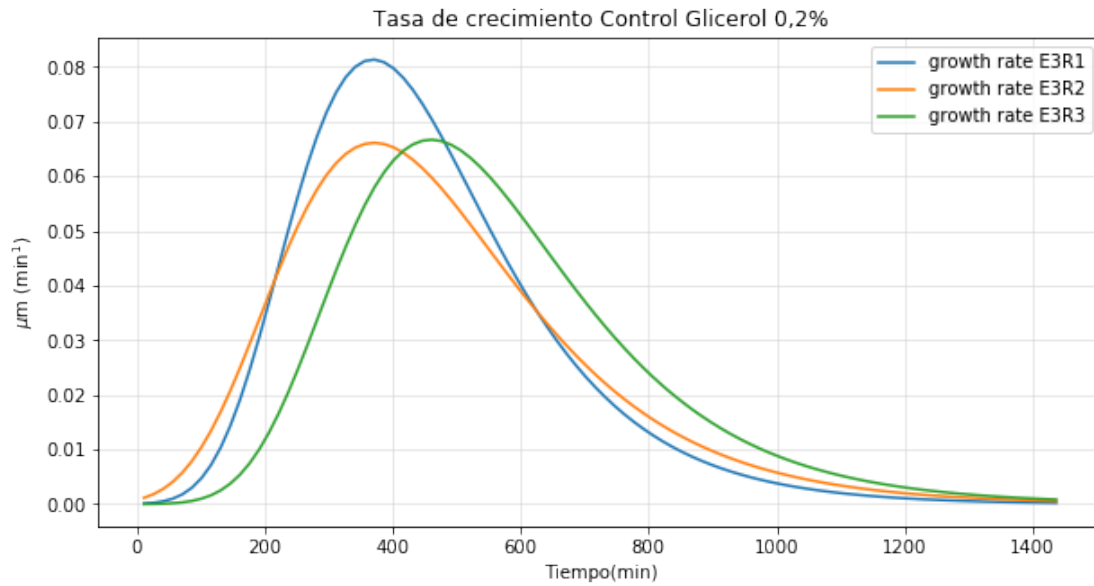
```
In [180]: #tasa de crecimiento
ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(112-tt))+1))))
#Con diff
dy12=(np.diff(ye12))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm12,tm12, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy12,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[180]: <matplotlib.legend.Legend at 0x178a24519b0>
```



```
In [181]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy10,label='growth rate E3R1')
plt.plot(tt[:-1],dy11,label='growth rate E3R2')
plt.plot(tt[:-1],dy12,label='growth rate E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

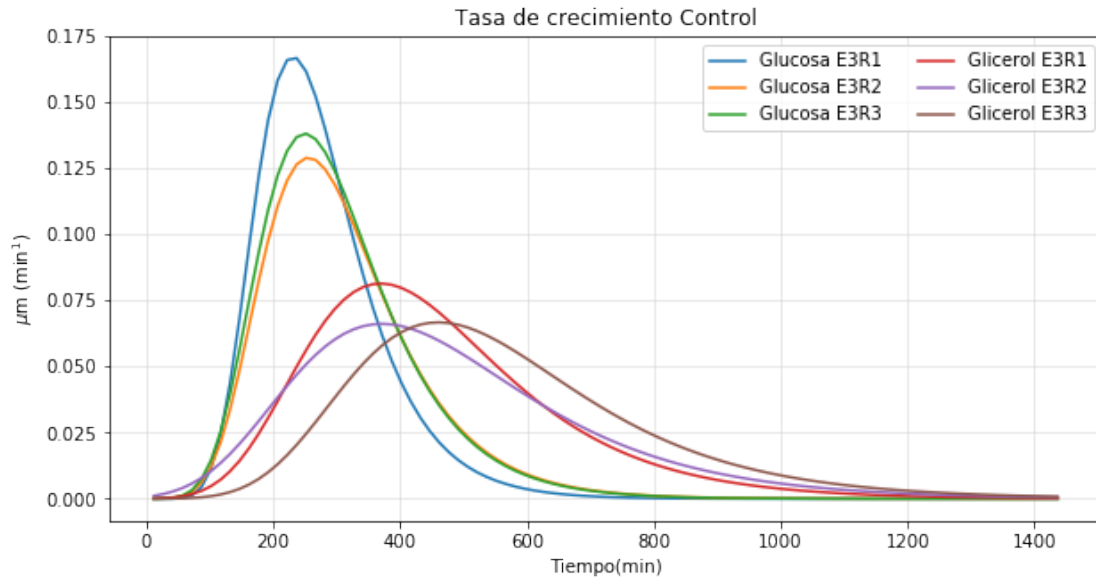
```
Out[181]: <matplotlib.legend.Legend at 0x178a251dda0>
```



```
In [182]: #Tasas control réplicas controles
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy7,label='Glucosa E3R1')
plt.plot(tt[:-1],dy8,label='Glucosa E3R2')
plt.plot(tt[:-1],dy9,label='Glucosa E3R3')
plt.plot(tt[:-1],dy10,label='Glicerol E3R1')
plt.plot(tt[:-1],dy11,label='Glicerol E3R2')
plt.plot(tt[:-1],dy12,label='Glicerol E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

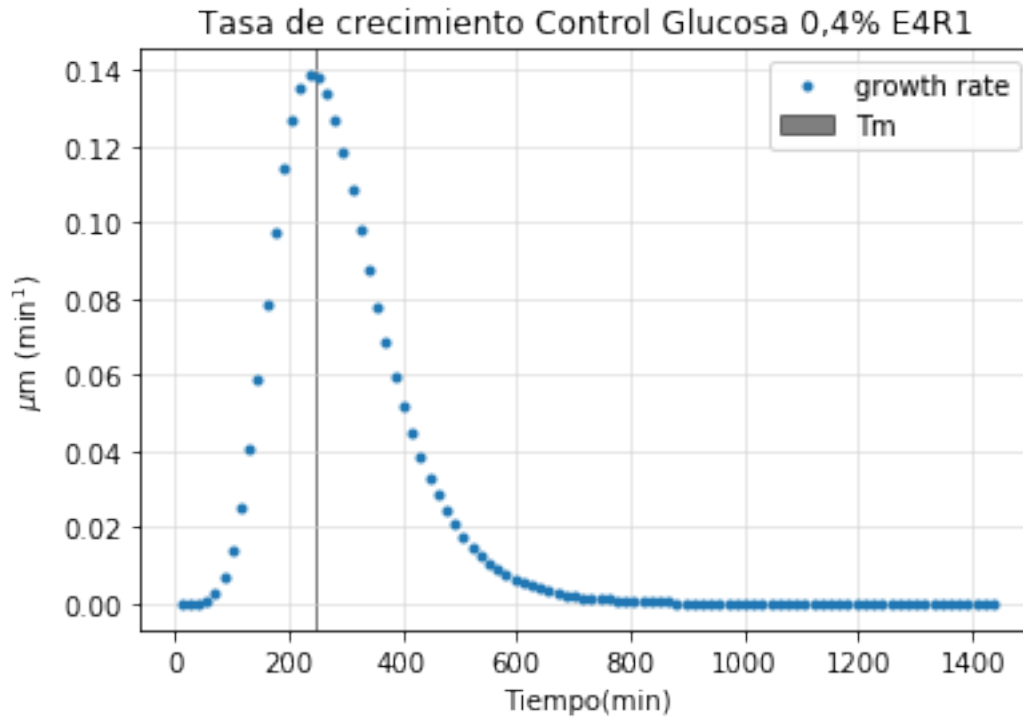
```
Out[182]: <matplotlib.legend.Legend at 0x178a25fffd0>
```





```
In [183]: #tasa de crecimiento
ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(113-tt))+1))))
#Con diff
dy13=(np.diff(ye13))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm13,tm13, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy13,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

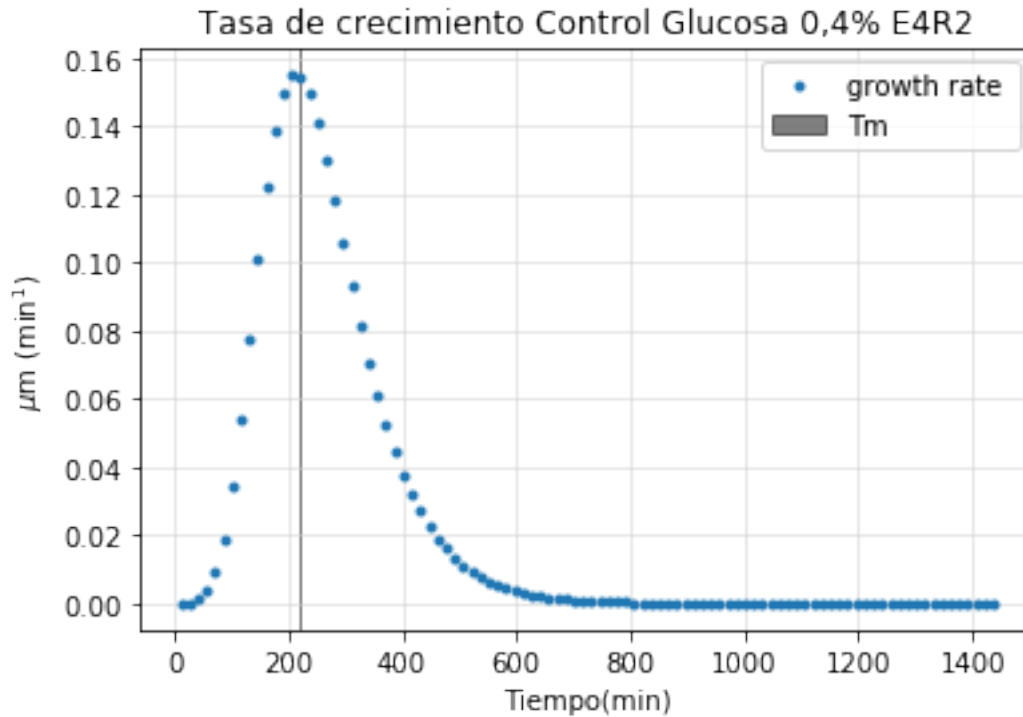
```
Out[183]: <matplotlib.legend.Legend at 0x178a26d9d68>
```



In [184]: *#tasa de crecimiento*

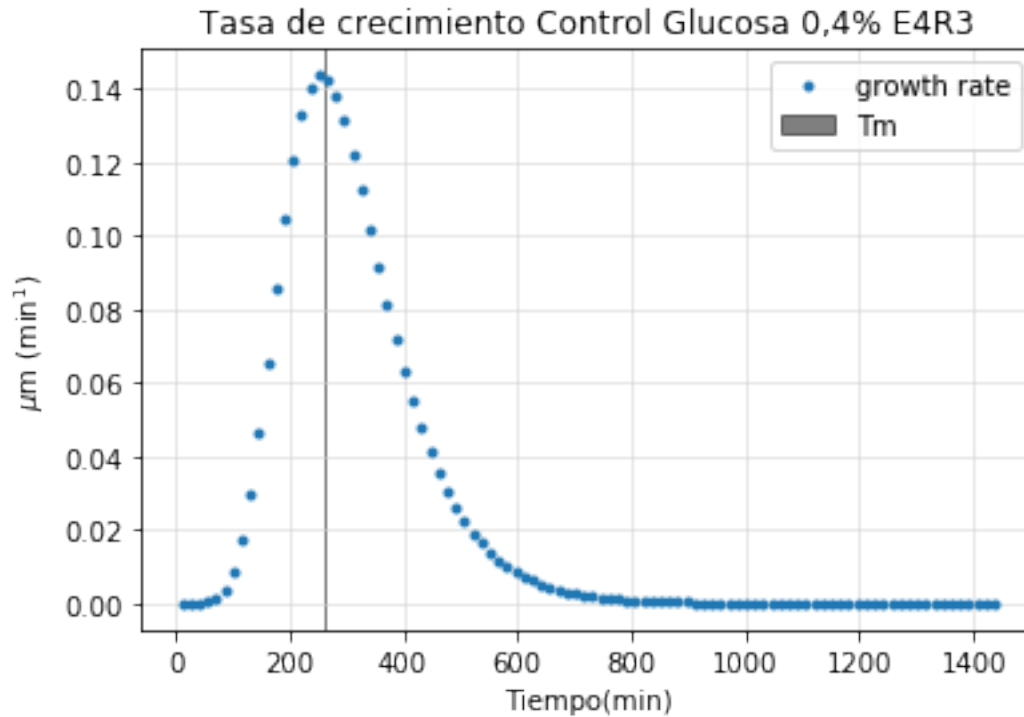
```
ye14=((A14*np.exp(-np.exp((((um14*np.exp(1))/A14)*(114-tt))+1))))
#Con diff
dy14=(np.diff(ye14))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm14,tm14, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy14,'.',label='growth rate ')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[184]: <matplotlib.legend.Legend at 0x178a27ae940>



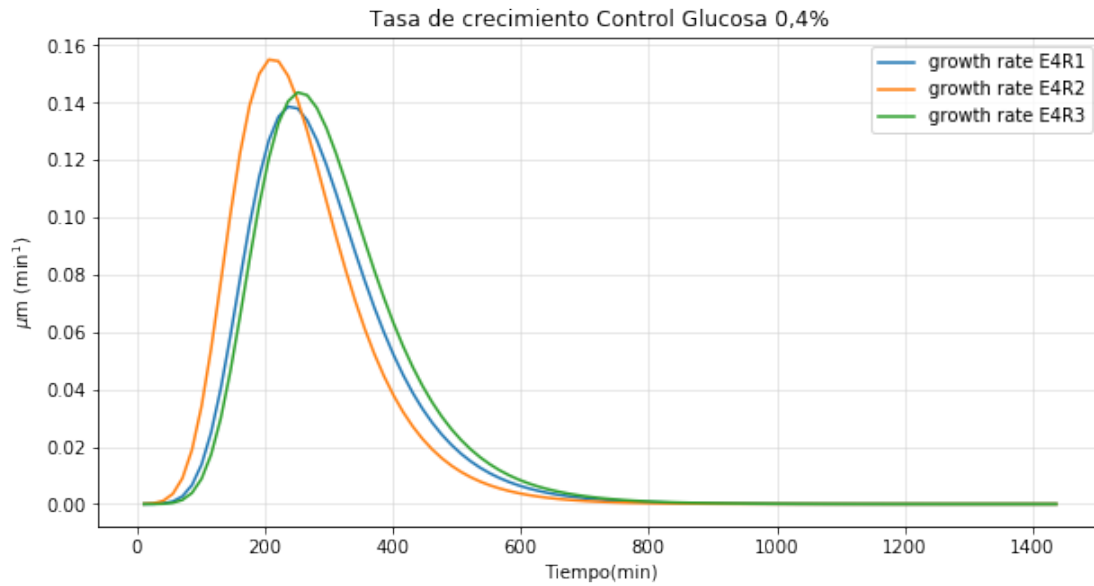
```
In [185]: #tasa de crecimiento
ye15=((A15*np.exp(-np.exp((((um15*np.exp(1))/A15)*(115-tt))+1))))
#Con diff
dy15=(np.diff(ye15))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm15,tm15, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy15,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[185]: <matplotlib.legend.Legend at 0x178a286eda0>
```



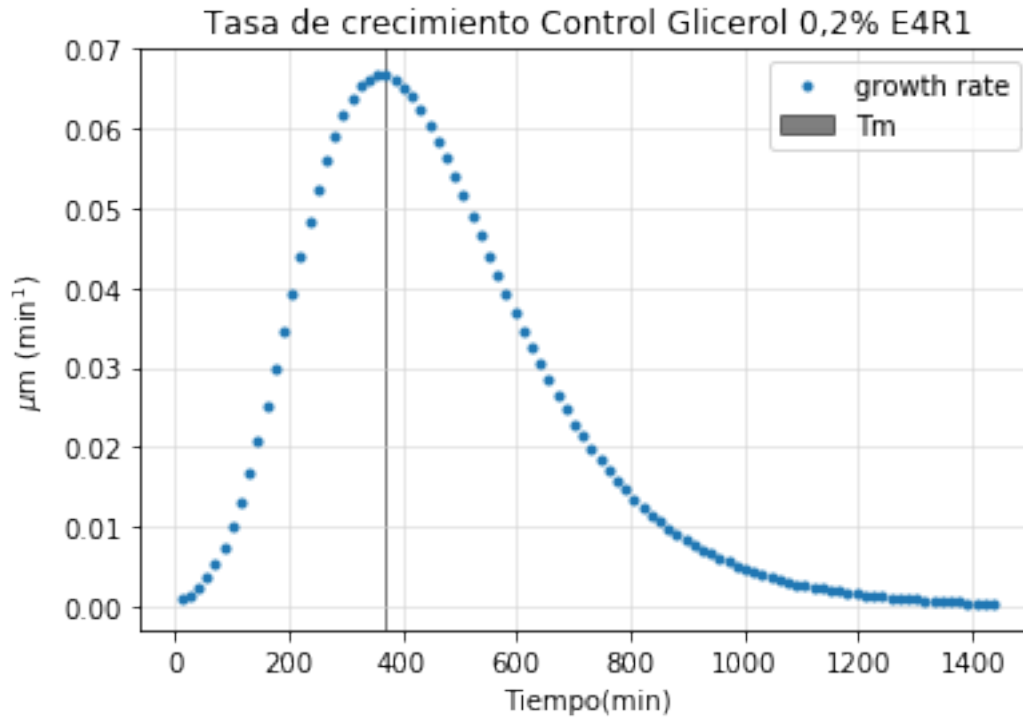
```
In [186]: #Tasas control réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy13,label='growth rate E4R1')
plt.plot(tt[:-1],dy14,label='growth rate E4R2')
plt.plot(tt[:-1],dy15,label='growth rate E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[186]: <matplotlib.legend.Legend at 0x178a2947828>
```



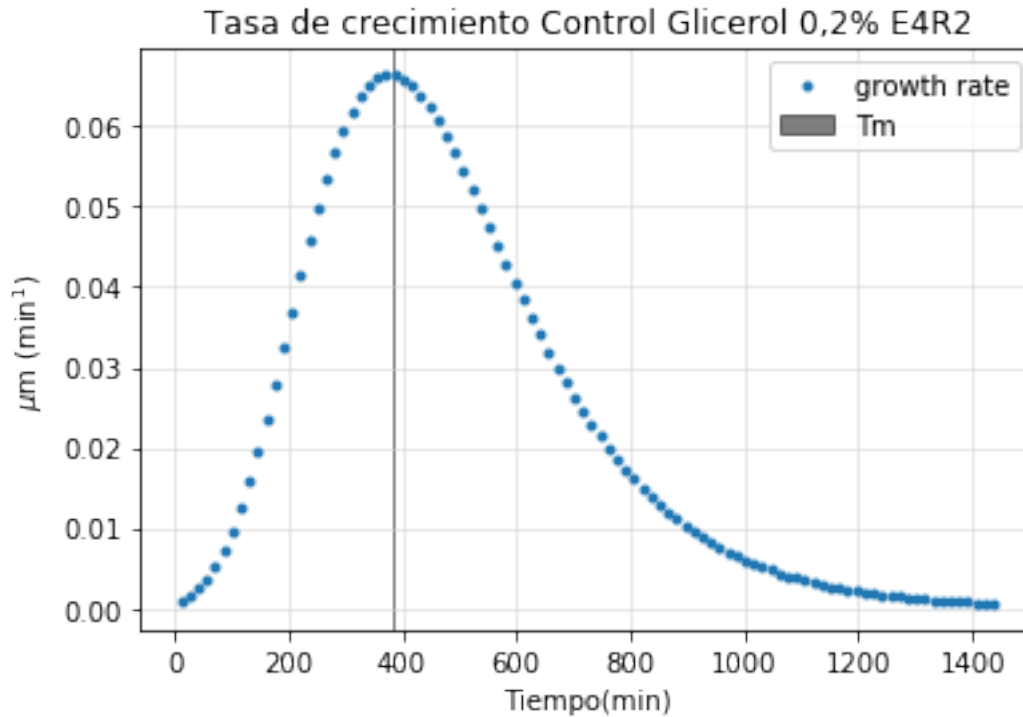
```
In [187]: #tasa de crecimiento
ye16=((A16*np.exp(-np.exp((((um16*np.exp(1))/A16)*(116-tt))+1))))
#Con diff
dy16=(np.diff(ye16))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm16,tm16, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy16,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[187]: <matplotlib.legend.Legend at 0x178a2a0def0>
```



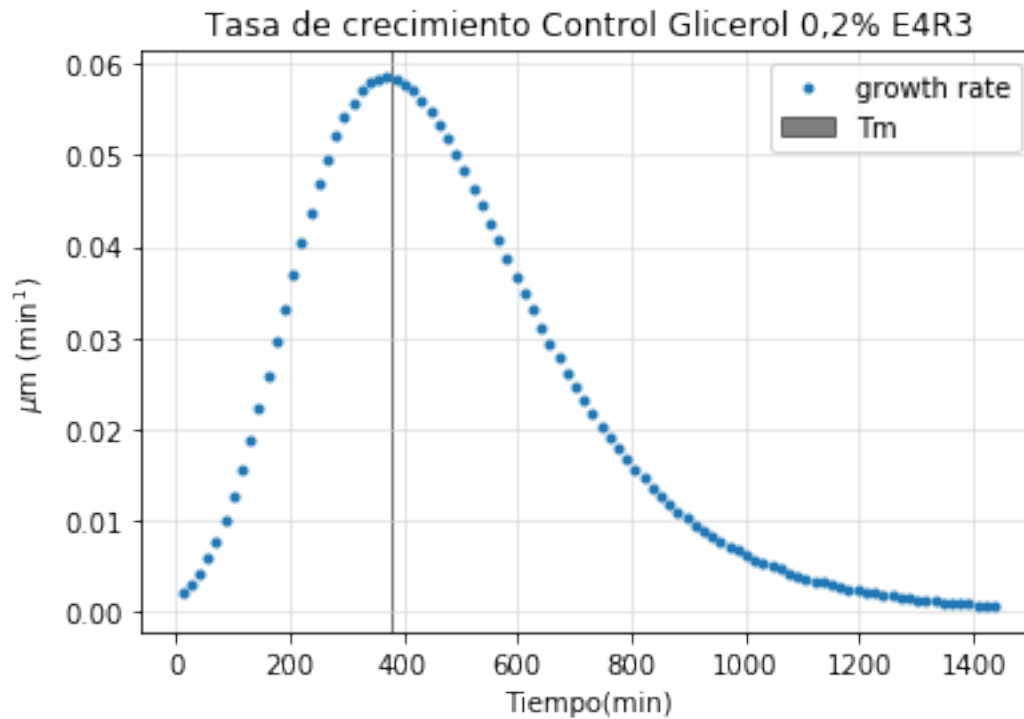
```
In [188]: #tasa de crecimiento
ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(117-tt))+1))))
#Con diff
dy17=(np.diff(ye17))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm17,tm17, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy17,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[188]: <matplotlib.legend.Legend at 0x178a2aeac18>
```



```
In [189]: #tasa de crecimiento
ye18=((A18*np.exp(-np.exp((((um18*np.exp(1))/A18)*(118-tt))+1))))
#Con diff
dy18=(np.diff(ye18))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm18,tm18, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy18,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

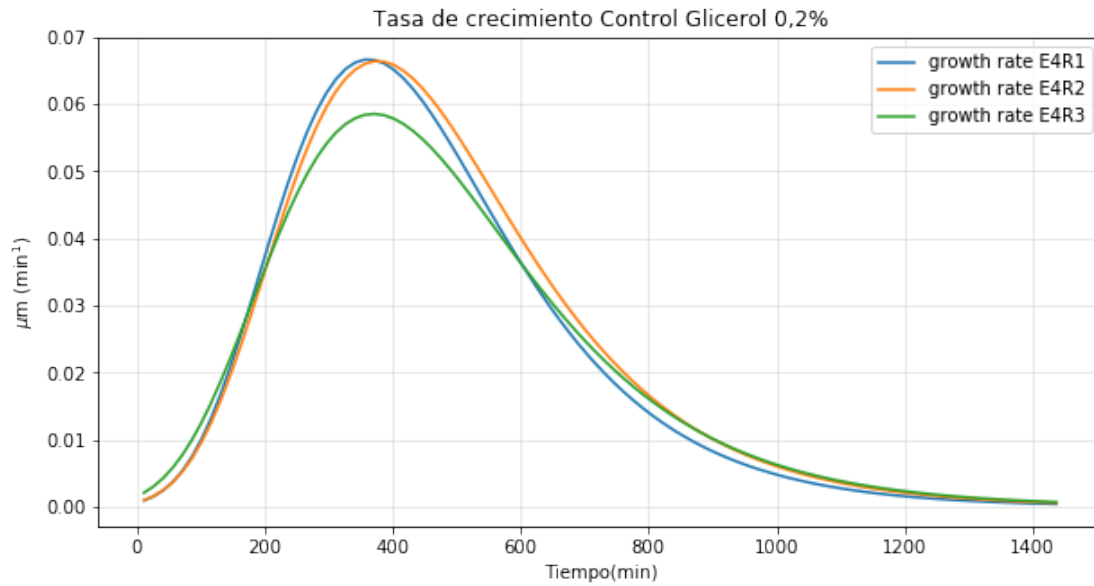
```
Out[189]: <matplotlib.legend.Legend at 0x178a241e470>
```



```
In [190]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy16,label='growth rate E4R1')
plt.plot(tt[:-1],dy17,label='growth rate E4R2')
plt.plot(tt[:-1],dy18,label='growth rate E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

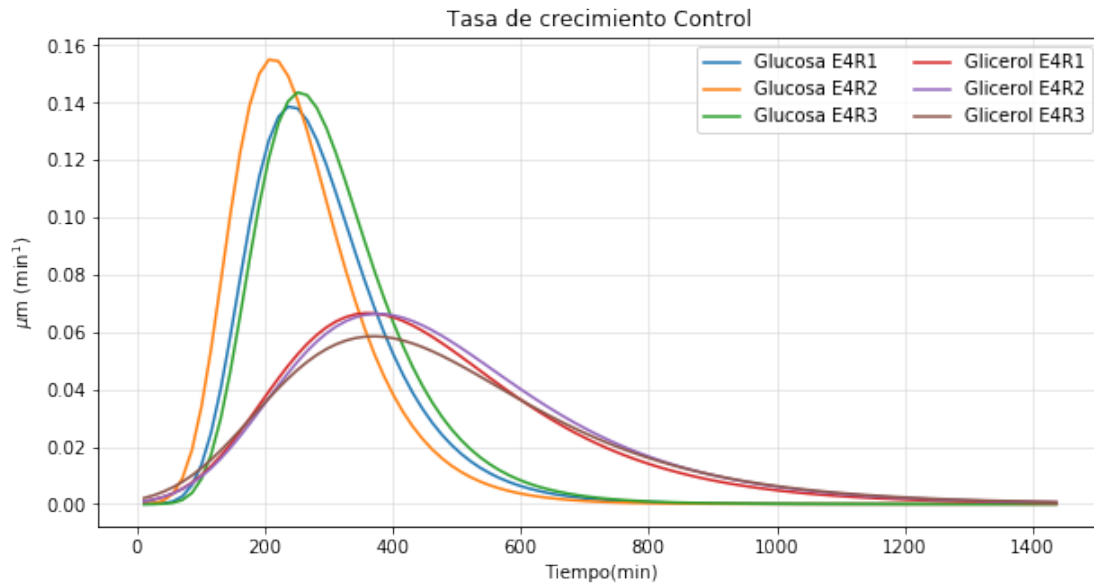
```
Out[190]: <matplotlib.legend.Legend at 0x1789f9ca6d8>
```





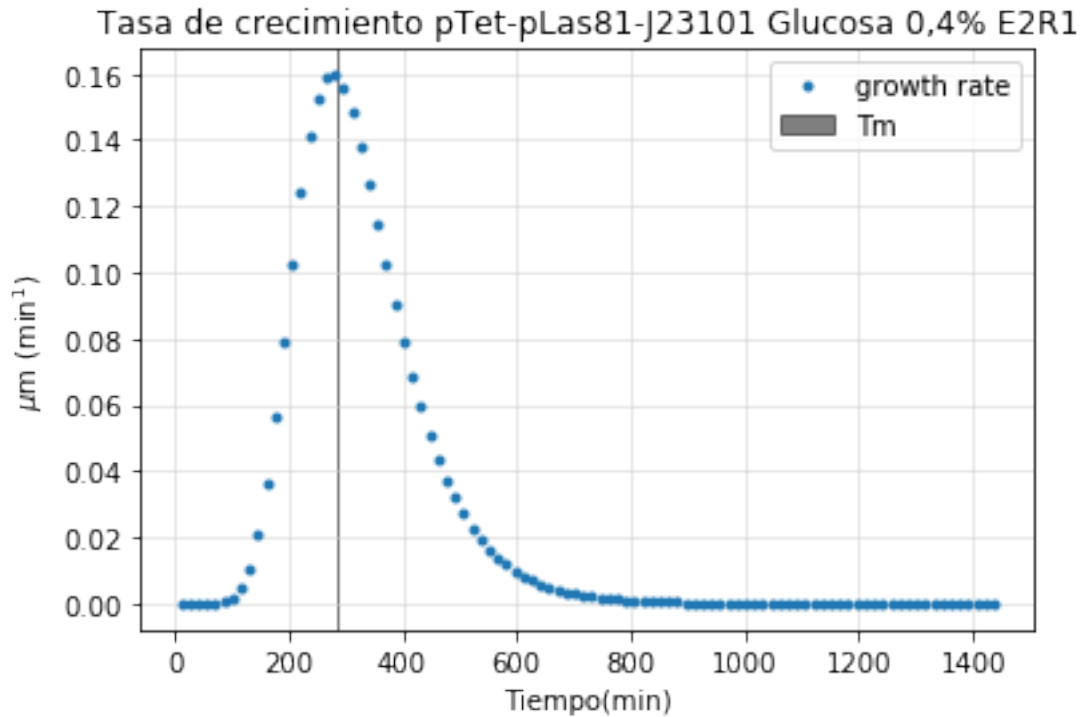
```
In [191]: #Tasas control réplicas controles
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento Control')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy13,label='Glucosa E4R1')
plt.plot(tt[:-1],dy14,label='Glucosa E4R2')
plt.plot(tt[:-1],dy15,label='Glucosa E4R3')
plt.plot(tt[:-1],dy16,label='Glicerol E4R1')
plt.plot(tt[:-1],dy17,label='Glicerol E4R2')
plt.plot(tt[:-1],dy18,label='Glicerol E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

```
Out[191]: <matplotlib.legend.Legend at 0x1789f392278>
```



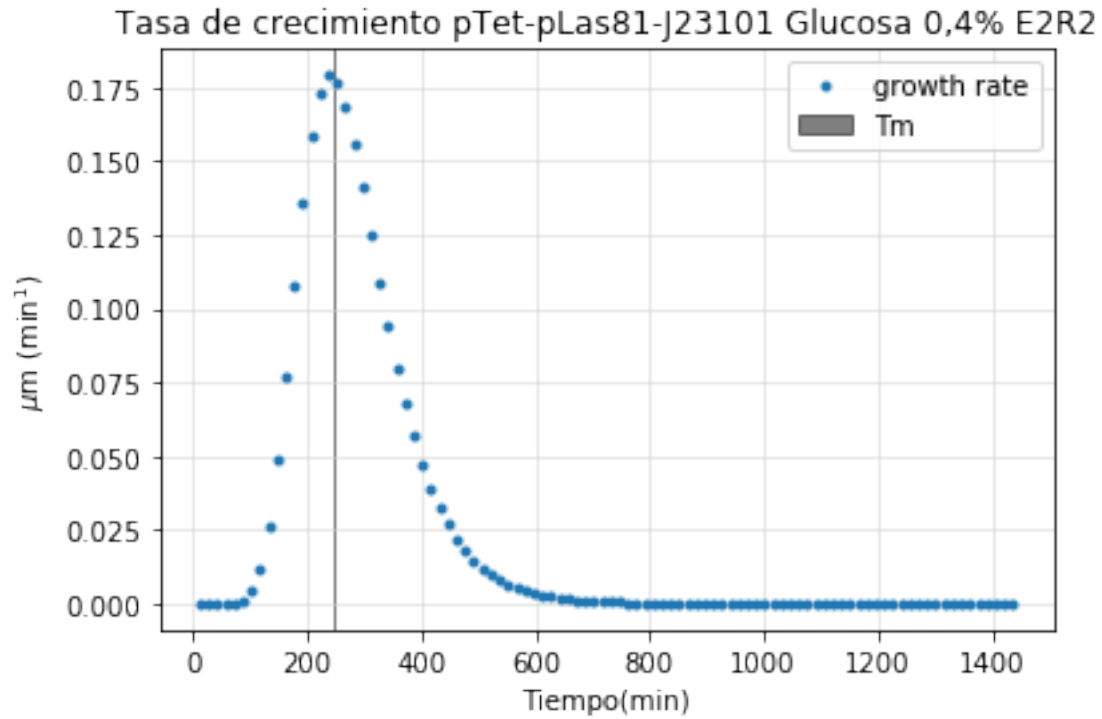
```
In [222]: #tasa de crecimiento
ye19=((A19*np.exp(-np.exp((((um19*np.exp(1))/A19)*(119-tt))+1))))
#Con diff
dy19=(np.diff(ye19))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm19,tm19, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy19,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[222]: <matplotlib.legend.Legend at 0x178a5a851d0>
```



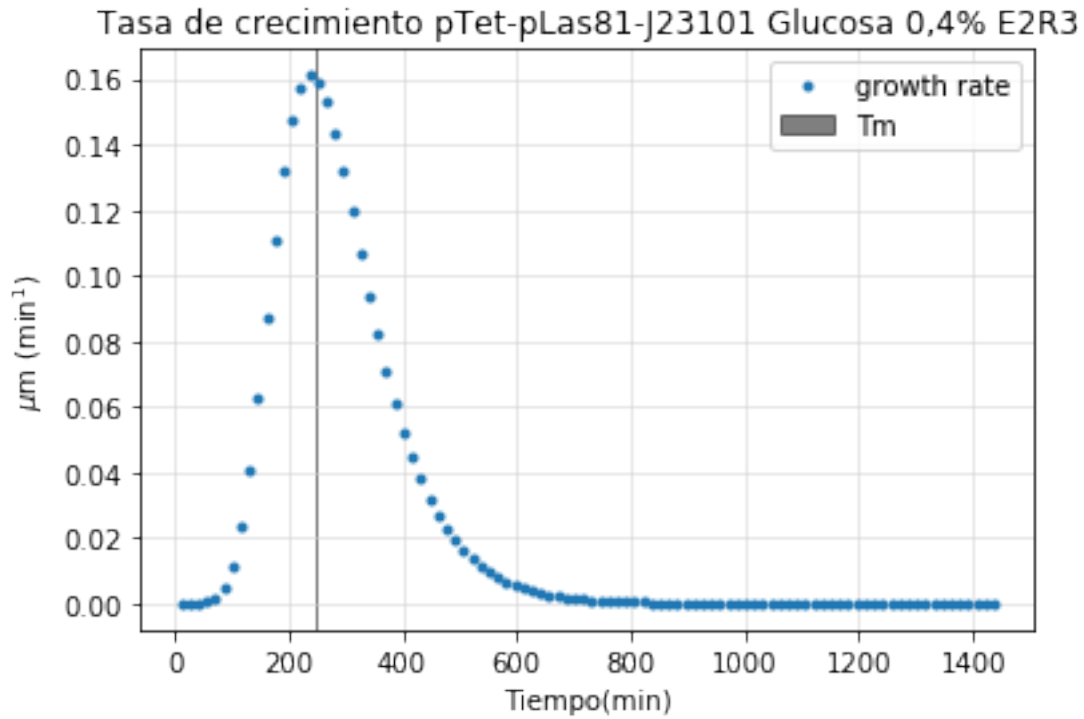
```
In [223]: #tasa de crecimiento
ye20=((A20*np.exp(-np.exp(((um20*np.exp(1))/A20)*(120-tt))+1))))
#Con diff
dy20=(np.diff(ye20))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm20,tm20, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy20,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[223]: <matplotlib.legend.Legend at 0x178a4226550>
```



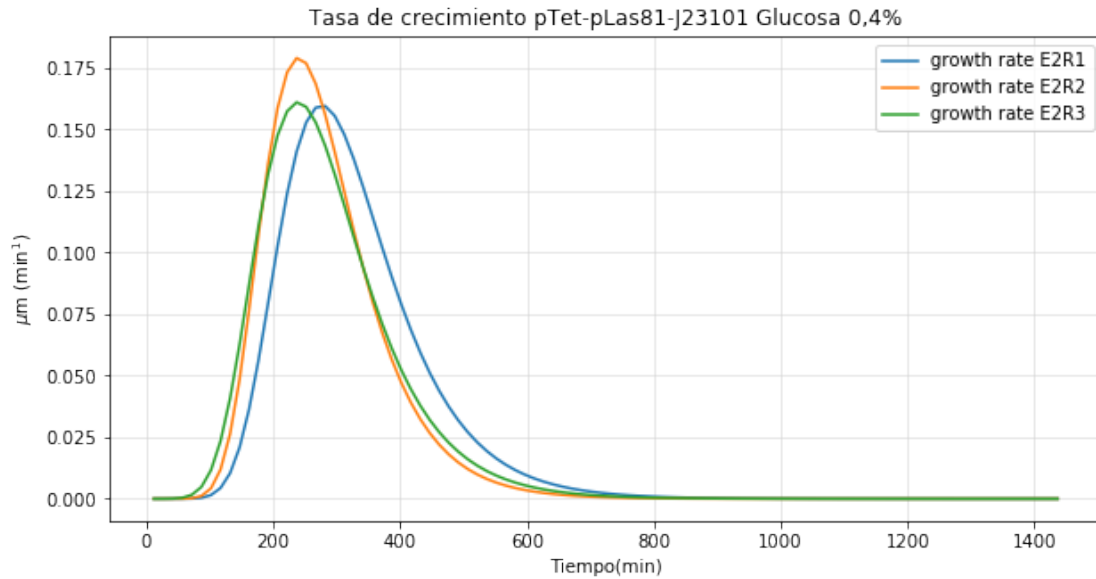
```
In [224]: #tasa de crecimiento
ye21=((A21*np.exp(-np.exp(((um21*np.exp(1))/A21)*(121-tt))+1))))
#Con diff
dy21=(np.diff(ye21))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm21,tm21, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy21,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[224]: <matplotlib.legend.Legend at 0x1789b19b668>
```



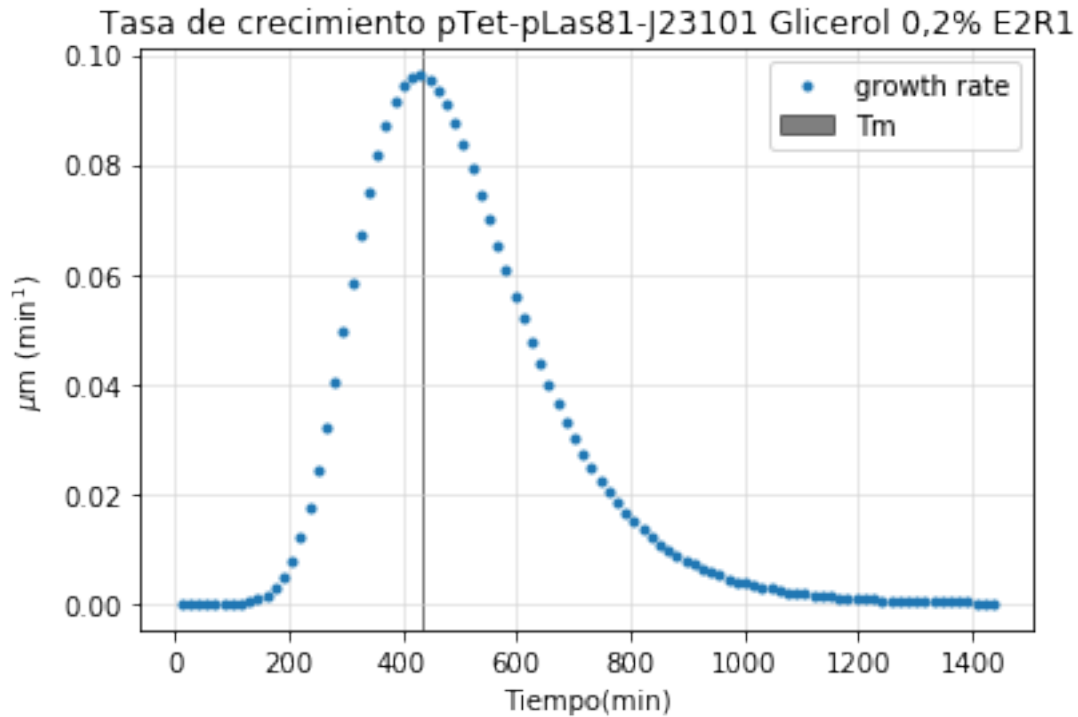
```
In [225]: #Tasas pLux-J23101-J23101 réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy19,label='growth rate E2R1')
plt.plot(tt[:-1],dy20,label='growth rate E2R2')
plt.plot(tt[:-1],dy21,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[225]: <matplotlib.legend.Legend at 0x178a759bc88>
```



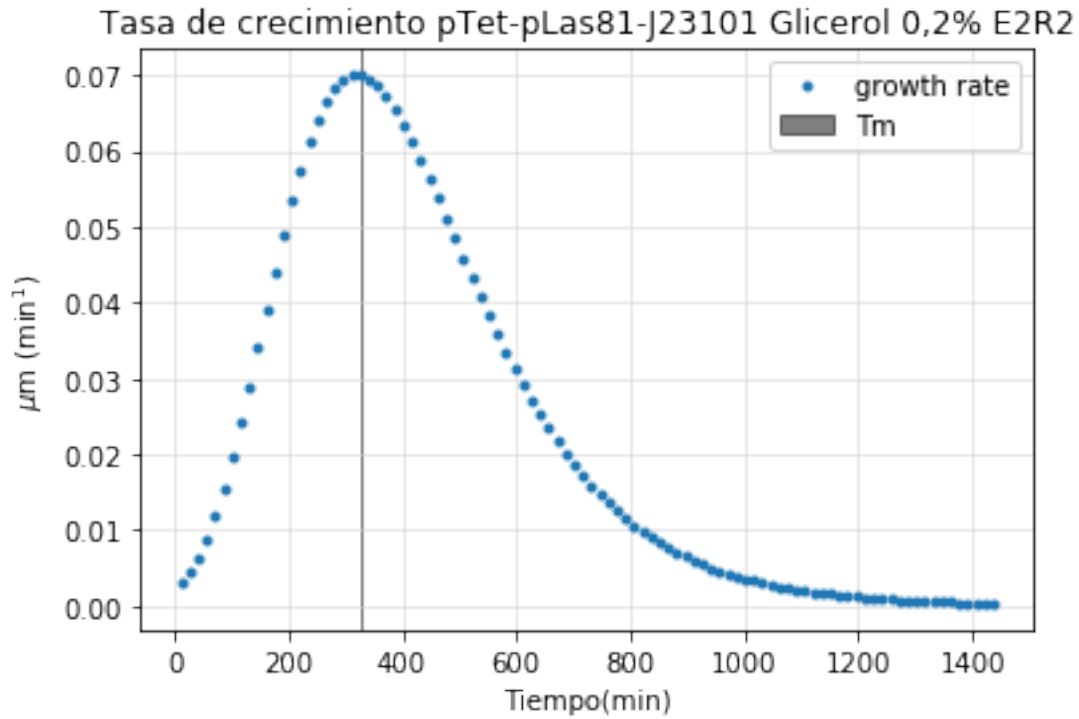
```
In [226]: #tasa de crecimiento
ye22=((A22*np.exp(-np.exp((((um22*np.exp(1))/A22)*(122-tt))+1))))
#Con diff
dy22=(np.diff(ye22))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm22,tm22, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy22,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[226]: <matplotlib.legend.Legend at 0x178a762ee10>
```



```
In [227]: #tasa de crecimiento
ye23=((A23*np.exp(-np.exp((((um23*np.exp(1))/A23)*(123-tt))+1))))
#Con diff
dy23=(np.diff(ye23))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm23,tm23, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy23,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

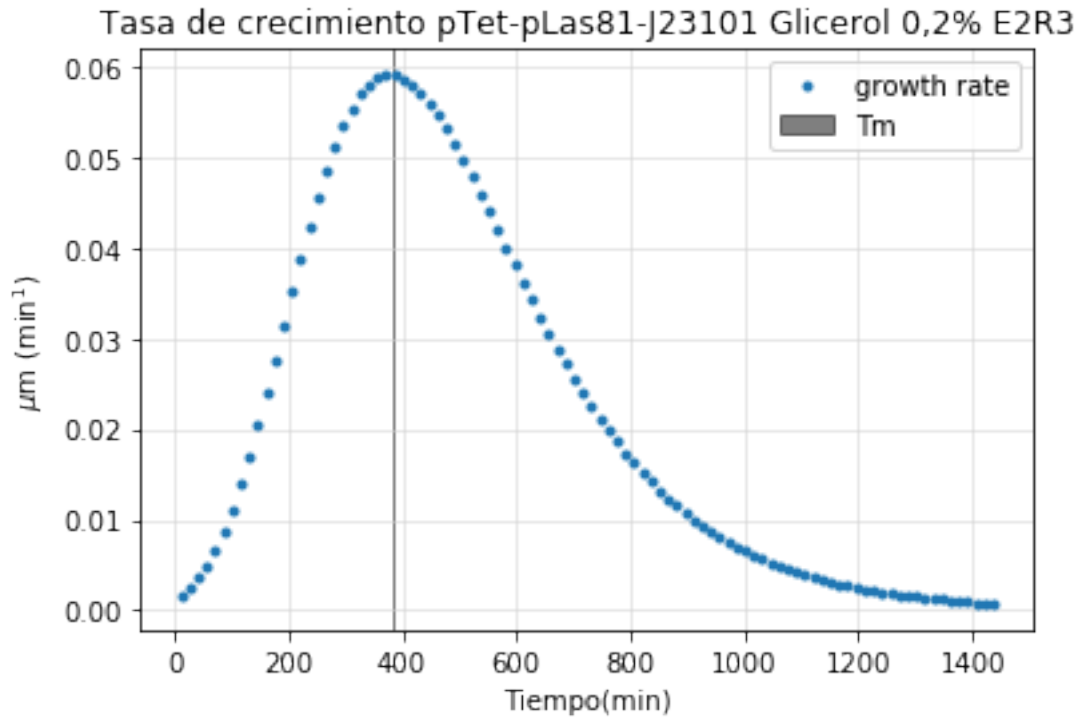
```
Out[227]: <matplotlib.legend.Legend at 0x178a76dcbe0>
```



```
In [228]: #tasa de crecimiento
ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(124-tt))+1))))
#Con diff
dy24=(np.diff(ye24))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm24,tm24, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy24,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

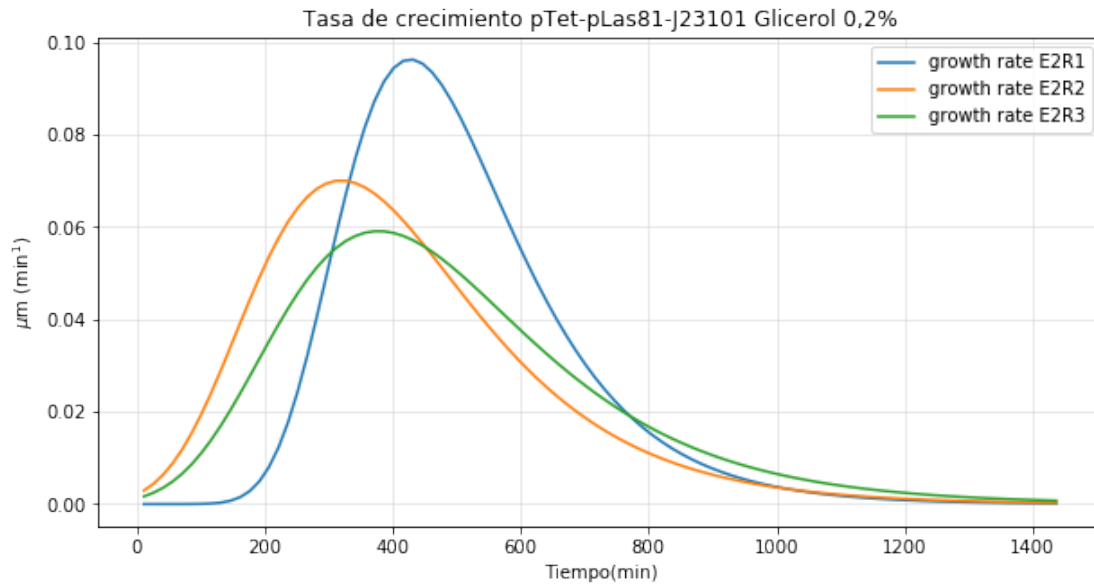
```
Out[228]: <matplotlib.legend.Legend at 0x178a77a0be0>
```





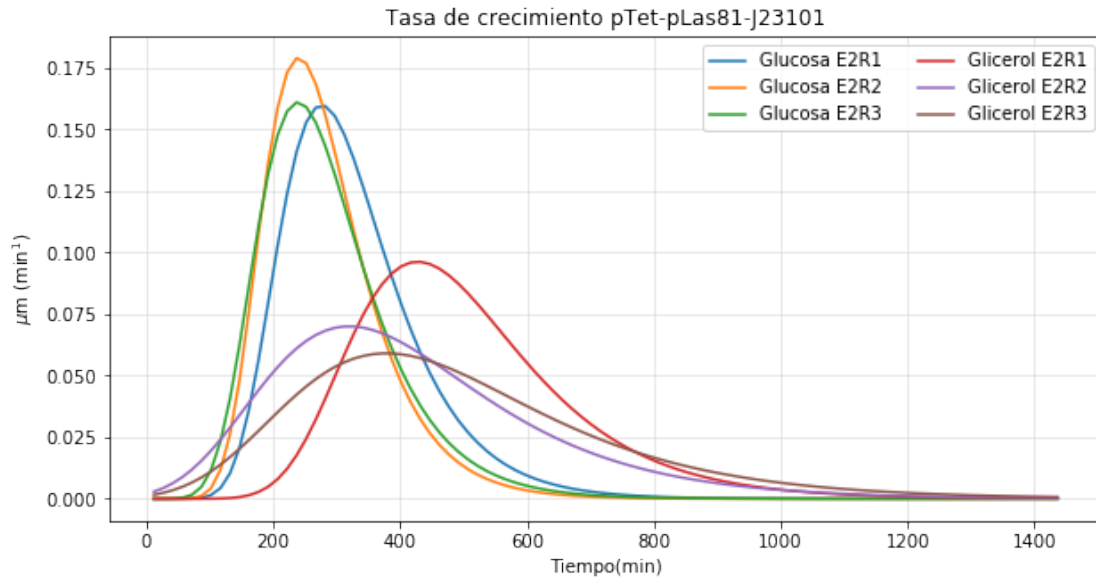
```
In [229]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy22,label='growth rate E2R1')
plt.plot(tt[:-1],dy23,label='growth rate E2R2')
plt.plot(tt[:-1],dy24,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[229]: <matplotlib.legend.Legend at 0x178a786b9b0>
```



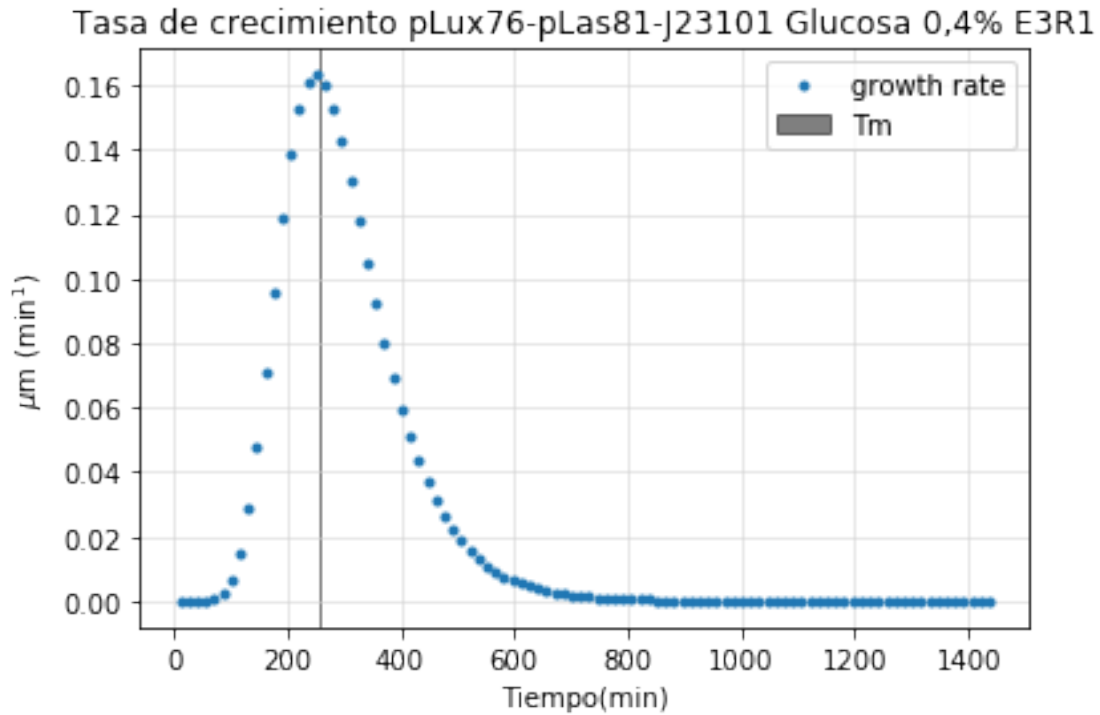
```
In [230]: #Tasas
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy19,label='Glucosa E2R1')
plt.plot(tt[:-1],dy20,label='Glucosa E2R2')
plt.plot(tt[:-1],dy21,label='Glucosa E2R3')
plt.plot(tt[:-1],dy22,label='Glicerol E2R1')
plt.plot(tt[:-1],dy23,label='Glicerol E2R2')
plt.plot(tt[:-1],dy24,label='Glicerol E2R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

```
Out[230]: <matplotlib.legend.Legend at 0x178a7951eb8>
```



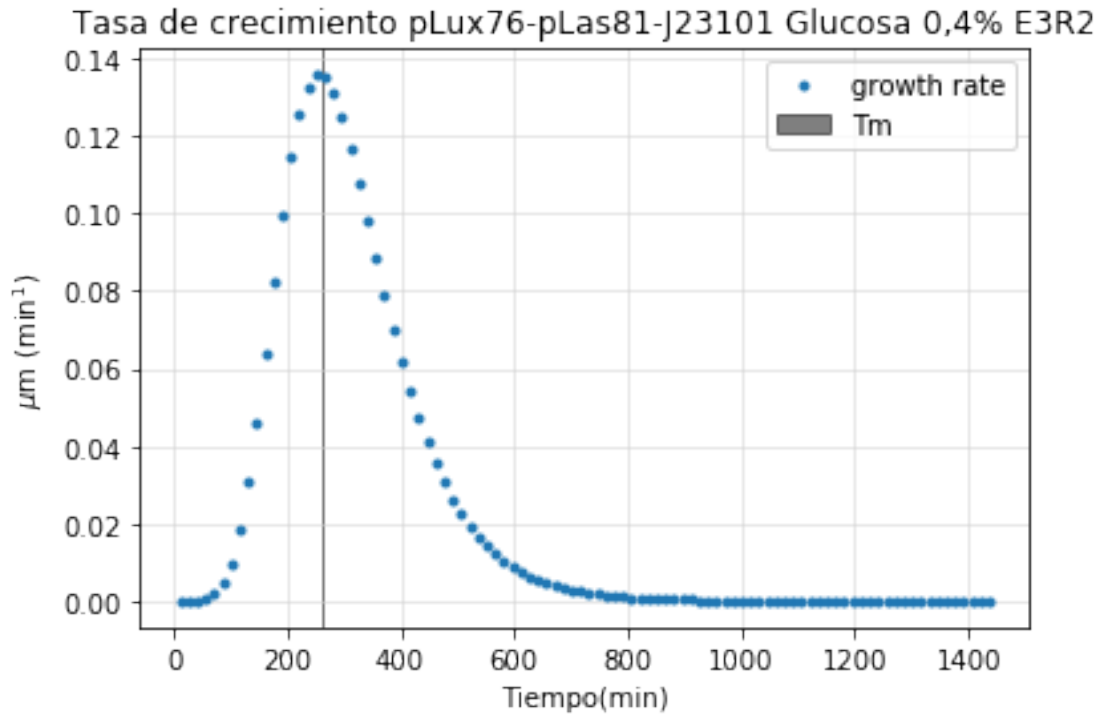
```
In [231]: #tasa de crecimiento
ye25=((A25*np.exp(-np.exp((((um25*np.exp(1))/A25)*(125-tt))+1))))
#Con diff
dy25=(np.diff(ye25))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm25,tm25, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy25,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[231]: <matplotlib.legend.Legend at 0x178a7a2b9b0>
```



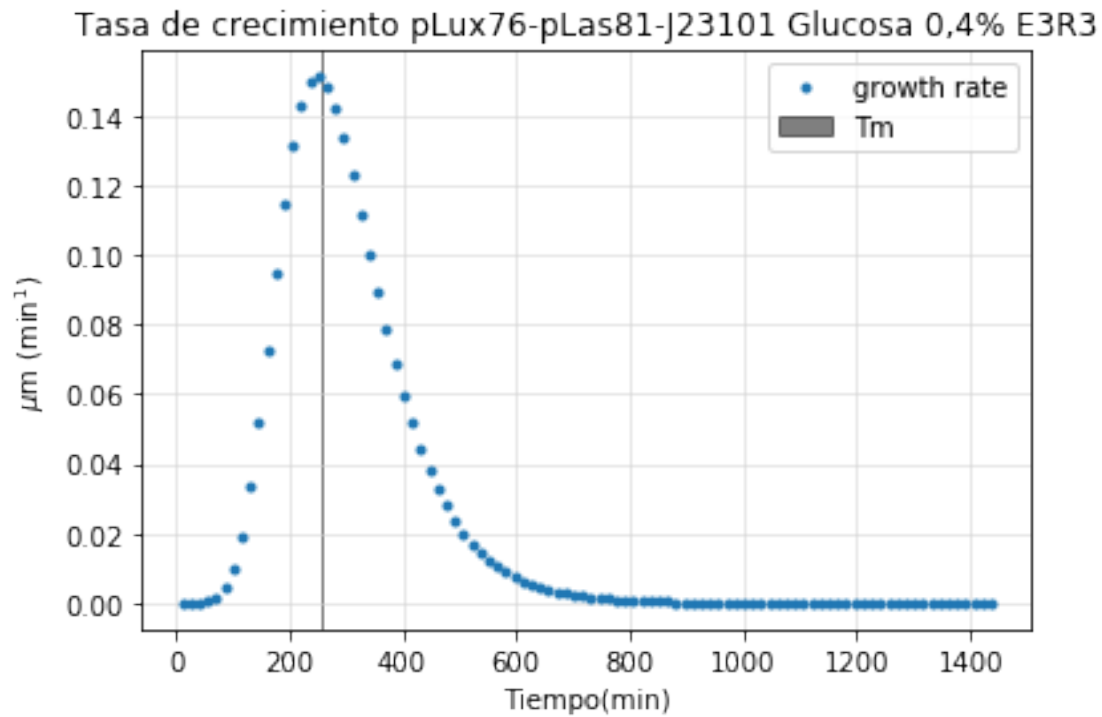
```
In [232]: #tasa de crecimiento
ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(126-tt))+1))))
#Con diff
dy26=(np.diff(ye26))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm26,tm26, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy26,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[232]: <matplotlib.legend.Legend at 0x178a7b05be0>
```



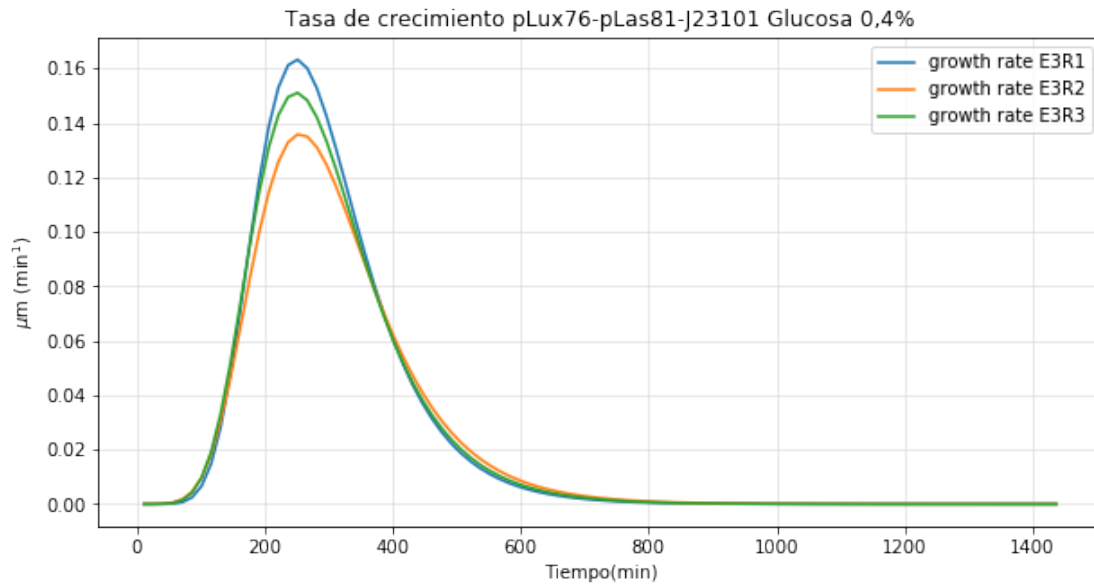
```
In [233]: #tasa de crecimiento
ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(127-tt))+1))))
#Con diff
dy27=(np.diff(ye27))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm27,tm27, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy27,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[233]: <matplotlib.legend.Legend at 0x178a7bd7c88>
```



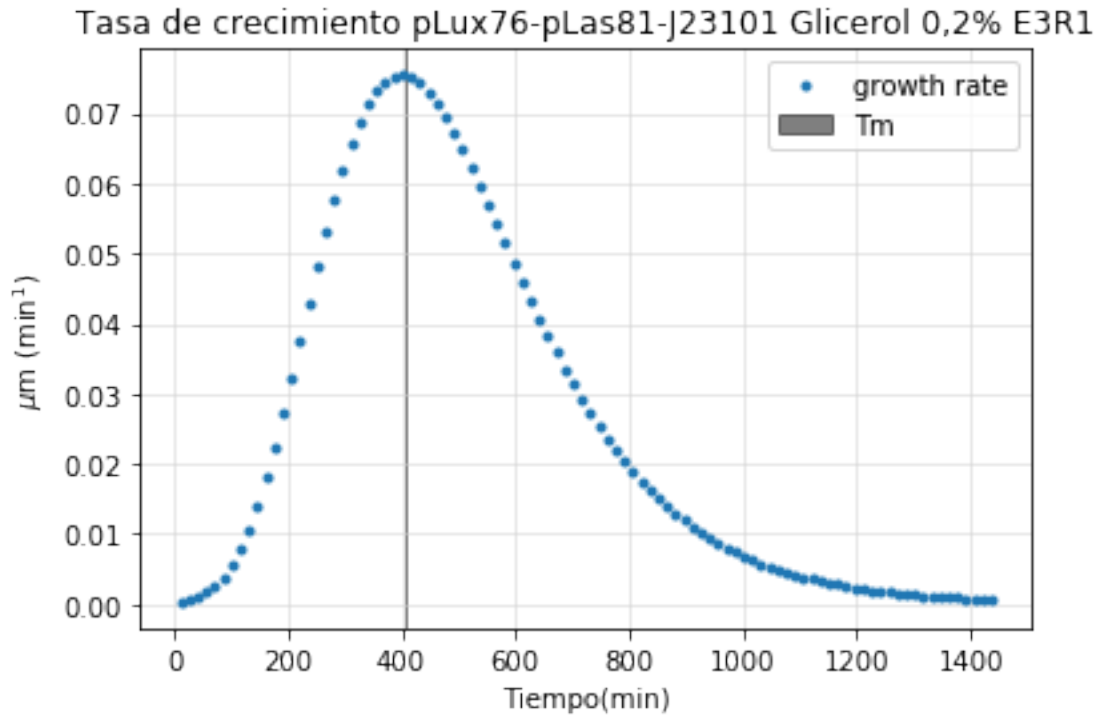
```
In [234]: #Tasas pLux-J23101-J23101 réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy25,label='growth rate E3R1')
plt.plot(tt[:-1],dy26,label='growth rate E3R2')
plt.plot(tt[:-1],dy27,label='growth rate E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[234]: <matplotlib.legend.Legend at 0x178a7ca2cc0>
```



```
In [235]: #tasa de crecimiento
ye28=((A28*np.exp(-np.exp((((um28*np.exp(1))/A28)*(128-tt))+1))))
#Con diff
dy28=(np.diff(ye28))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm28,tm28, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy28,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

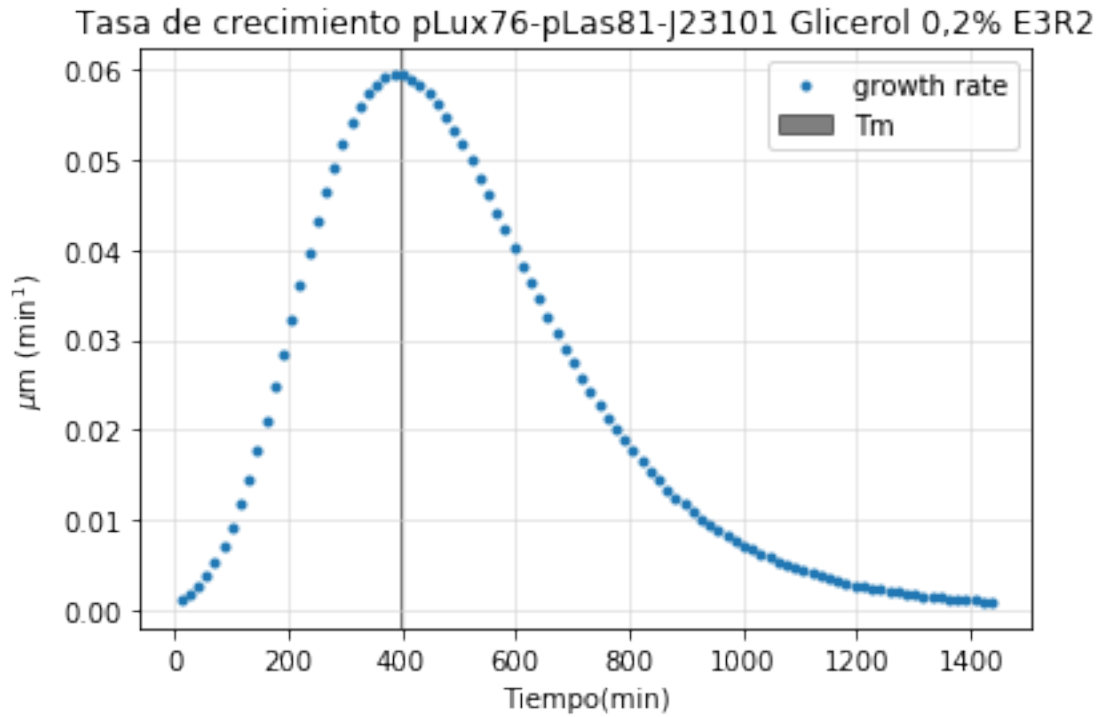
```
Out[235]: <matplotlib.legend.Legend at 0x178a7d6e978>
```



```
In [236]: #tasa de crecimiento
ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(129-tt))+1))))
#Con diff
dy29=(np.diff(ye29))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm29,tm29, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy29,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

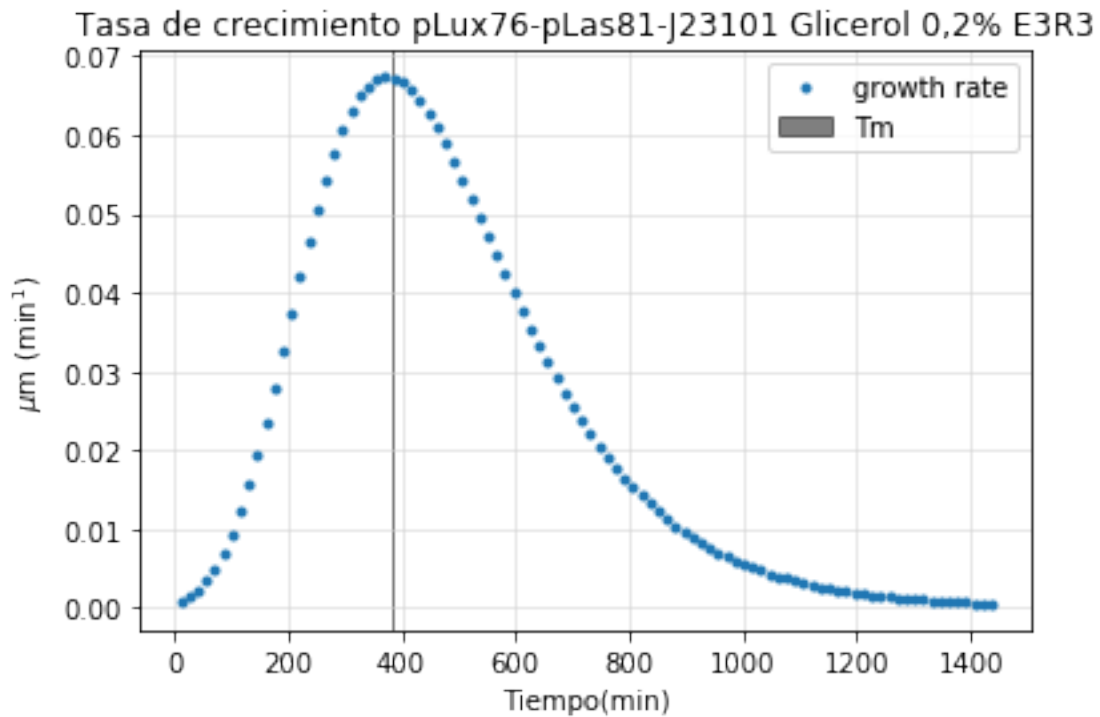
```
Out[236]: <matplotlib.legend.Legend at 0x178a7e46be0>
```





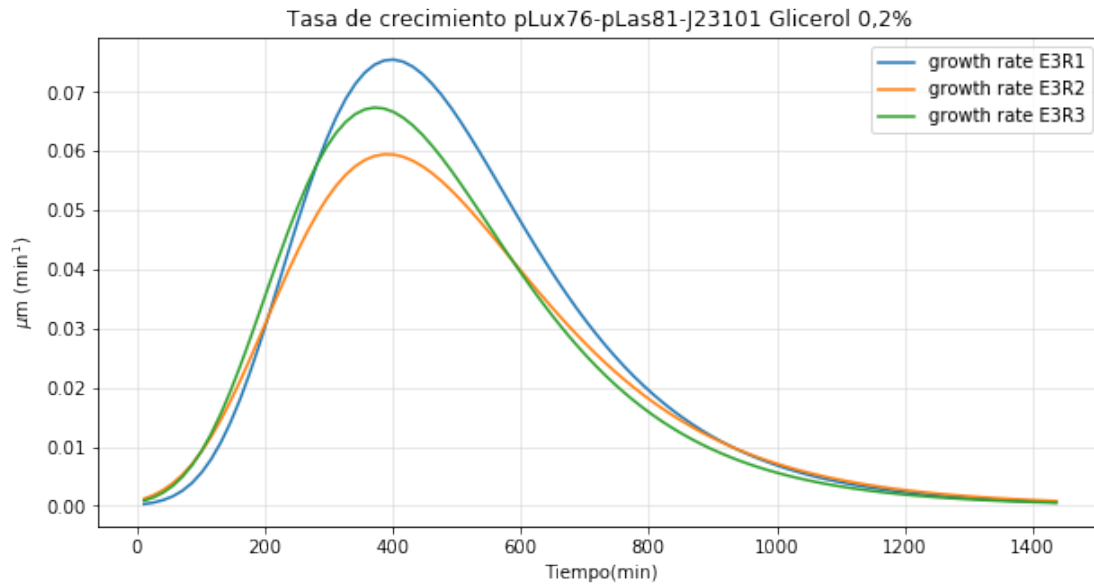
```
In [237]: #tasa de crecimiento
ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(130-tt))+1))))
#Con diff
dy30=(np.diff(ye30))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm30,tm30, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy30,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[237]: <matplotlib.legend.Legend at 0x178a7f159e8>
```



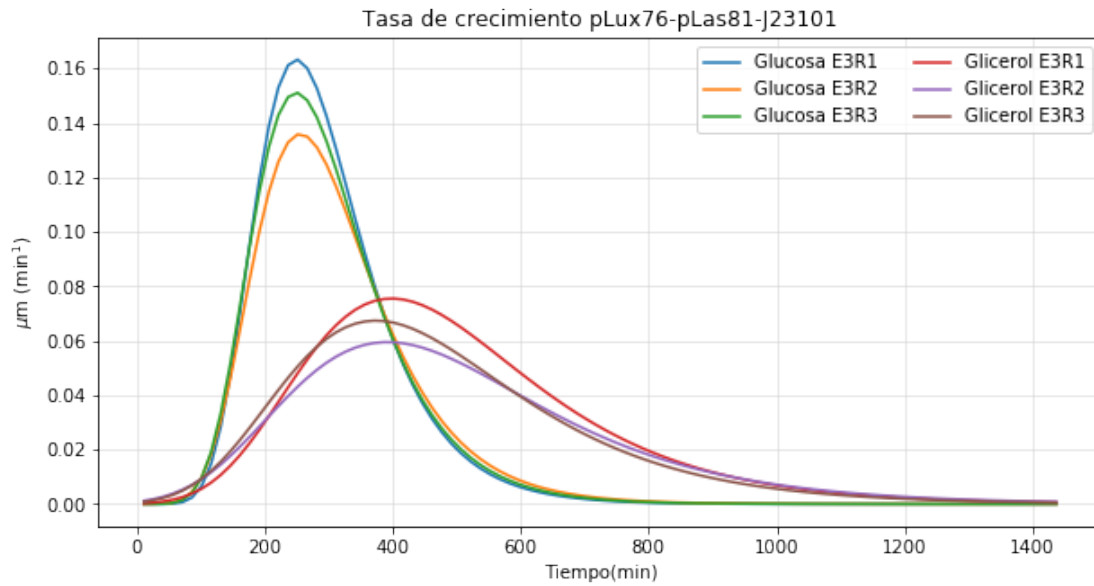
```
In [238]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy28,label='growth rate E3R1')
plt.plot(tt[:-1],dy29,label='growth rate E3R2')
plt.plot(tt[:-1],dy30,label='growth rate E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[238]: <matplotlib.legend.Legend at 0x178a7feca58>
```



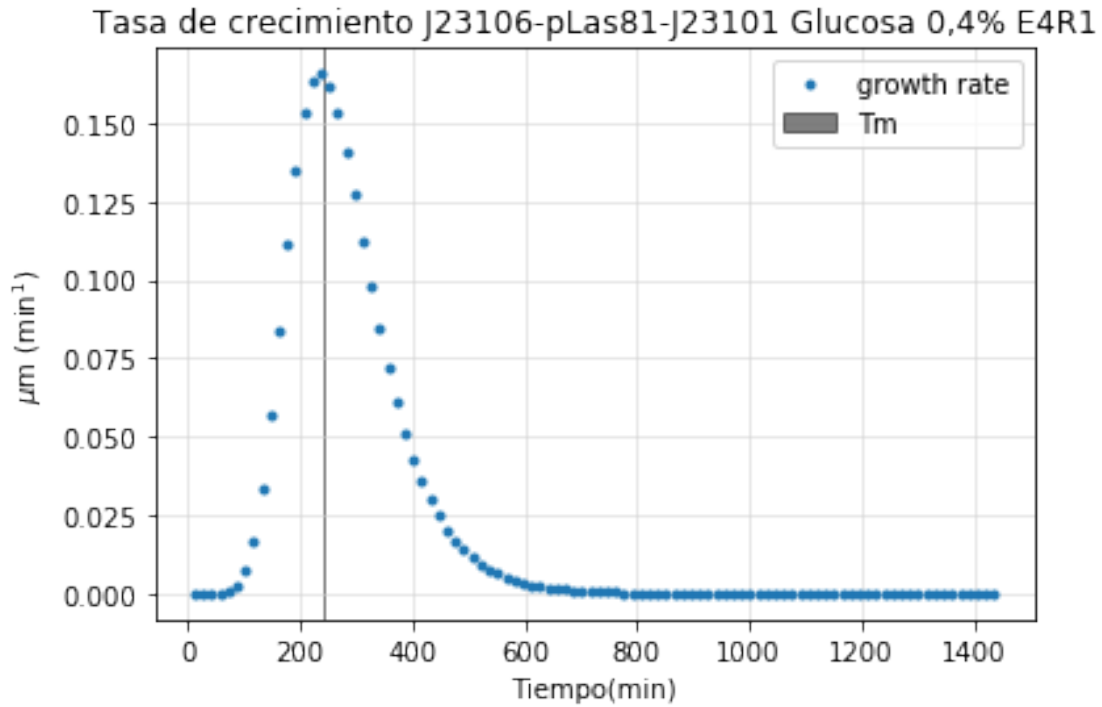
```
In [239]: #Tasas control réplicas
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^{-1}$)')
plt.plot(tt[:-1],dy25,label='Glucosa E3R1')
plt.plot(tt[:-1],dy26,label='Glucosa E3R2')
plt.plot(tt[:-1],dy27,label='Glucosa E3R3')
plt.plot(tt[:-1],dy28,label='Glicerol E3R1')
plt.plot(tt[:-1],dy29,label='Glicerol E3R2')
plt.plot(tt[:-1],dy30,label='Glicerol E3R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

```
Out[239]: <matplotlib.legend.Legend at 0x178a80e1710>
```



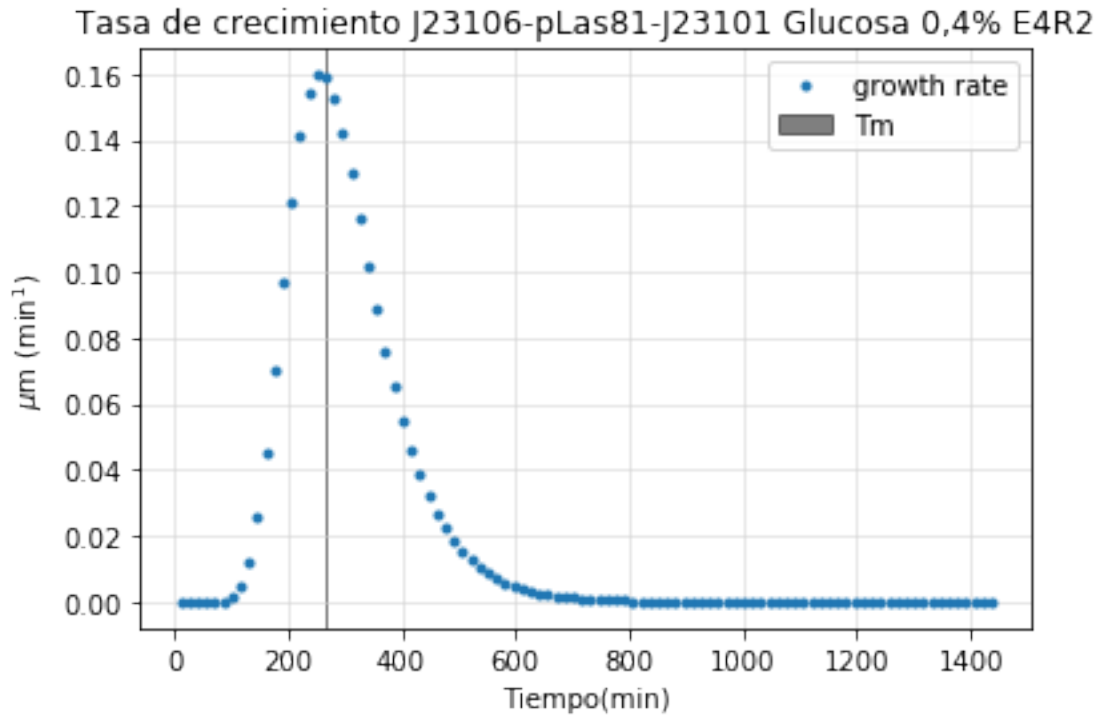
```
In [240]: #tasa de crecimiento
ye31=((A31*np.exp(-np.exp((((um31*np.exp(1))/A31)*(131-tt))+1))))
#Con diff
dy31=(np.diff(ye31))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm31,tm31, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy31,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[240]: <matplotlib.legend.Legend at 0x178a81b7a90>



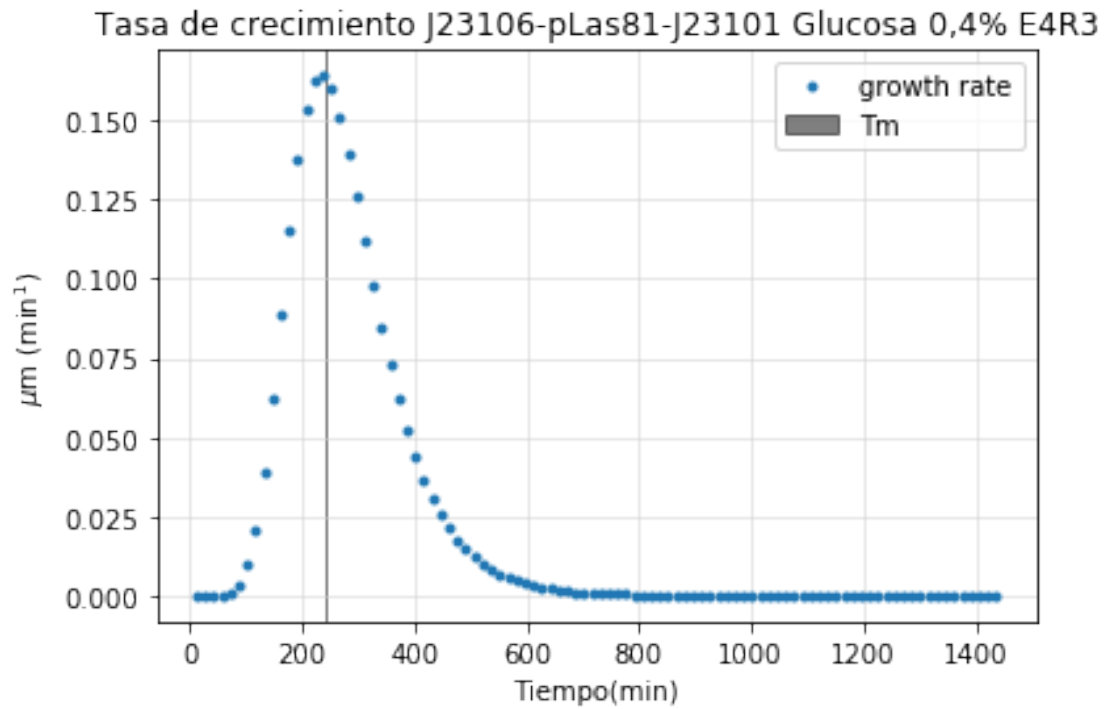
```
In [241]: #tasa de crecimiento
ye32=((A32*np.exp(-np.exp((((um32*np.exp(1))/A32)*(132-tt))+1))))
#Con diff
dy32=(np.diff(ye32))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm32,tm32, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy32,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[241]: <matplotlib.legend.Legend at 0x178a8293b38>
```



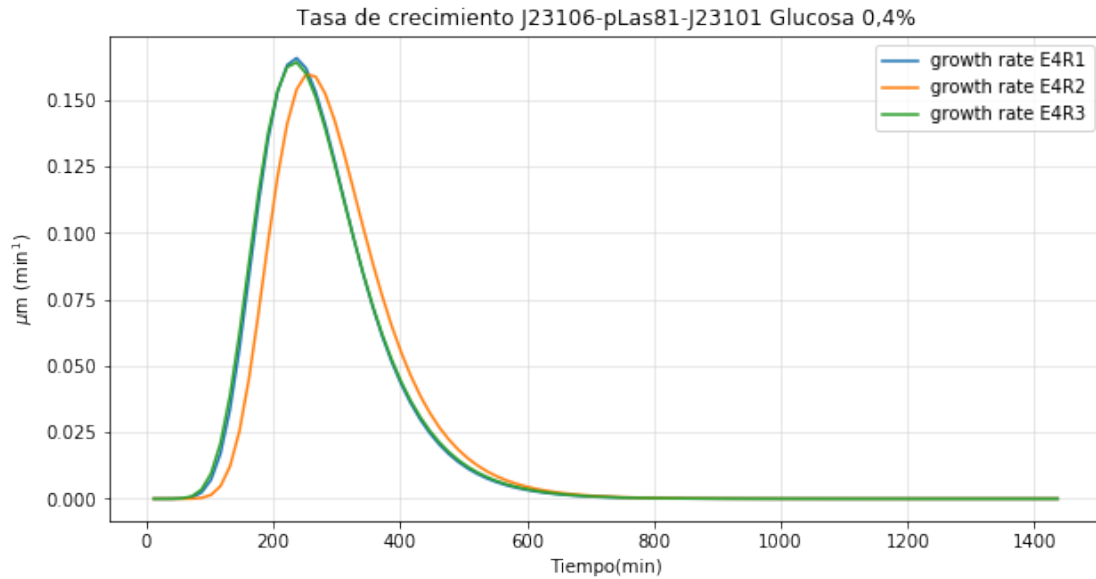
```
In [242]: #tasa de crecimiento
ye33=((A33*np.exp(-np.exp((((um33*np.exp(1))/A33)*(133-tt))+1))))
#Con diff
dy33=(np.diff(ye33))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm33,tm33, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy33,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[242]: <matplotlib.legend.Legend at 0x178a9323828>
```



```
In [243]: #Tasas J23101-J23101-J23101 réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy31,label='growth rate E4R1')
plt.plot(tt[:-1],dy32,label='growth rate E4R2')
plt.plot(tt[:-1],dy33,label='growth rate E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

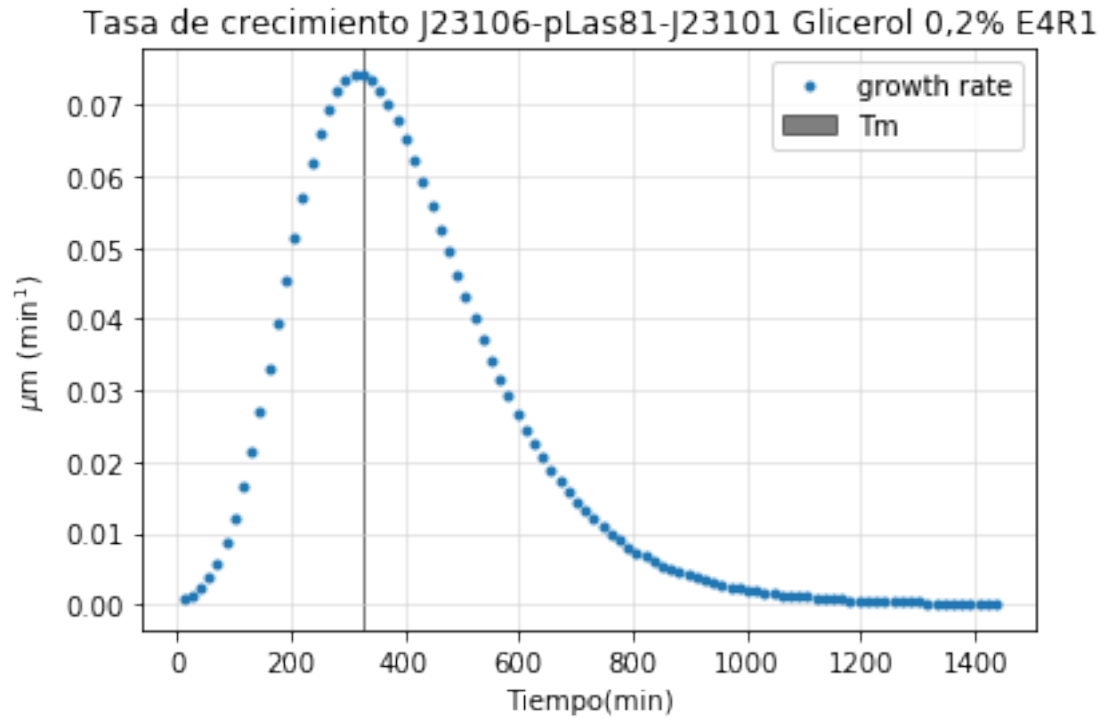
```
Out[243]: <matplotlib.legend.Legend at 0x178a93f69b0>
```



```
In [244]: #tasa de crecimiento
ye34=((A34*np.exp(-np.exp((((um34*np.exp(1))/A34)*(134-tt))+1))))
#Con diff
dy34=(np.diff(ye34))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm34,tm34, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy34,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

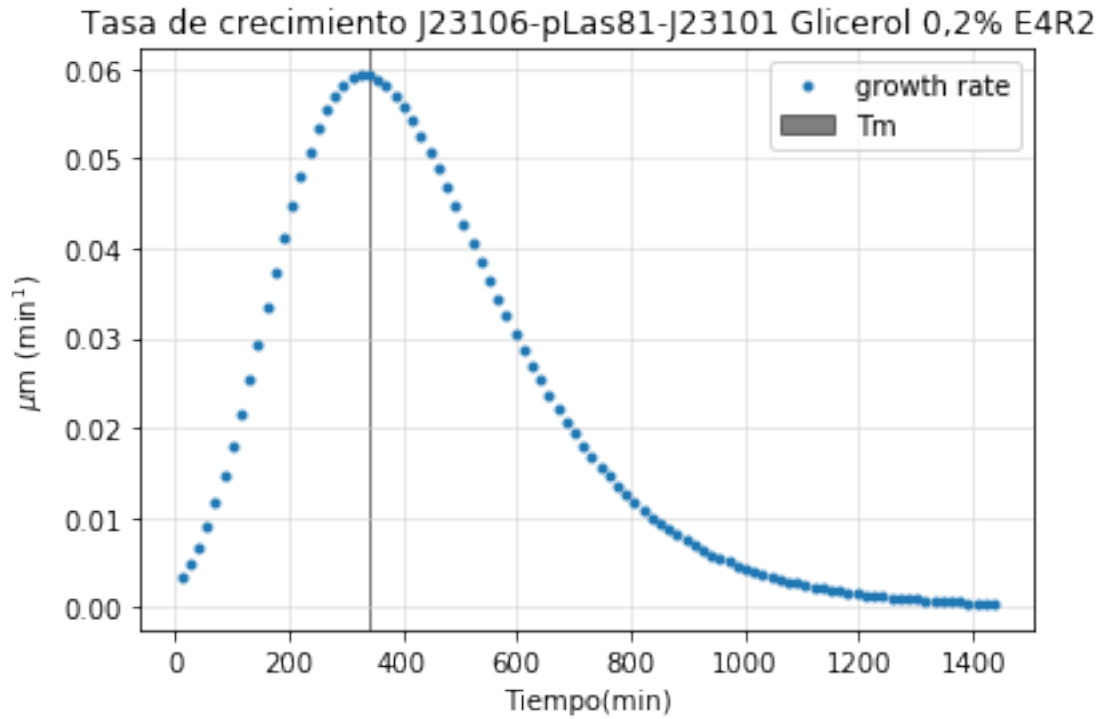
```
Out[244]: <matplotlib.legend.Legend at 0x178a94cbe80>
```





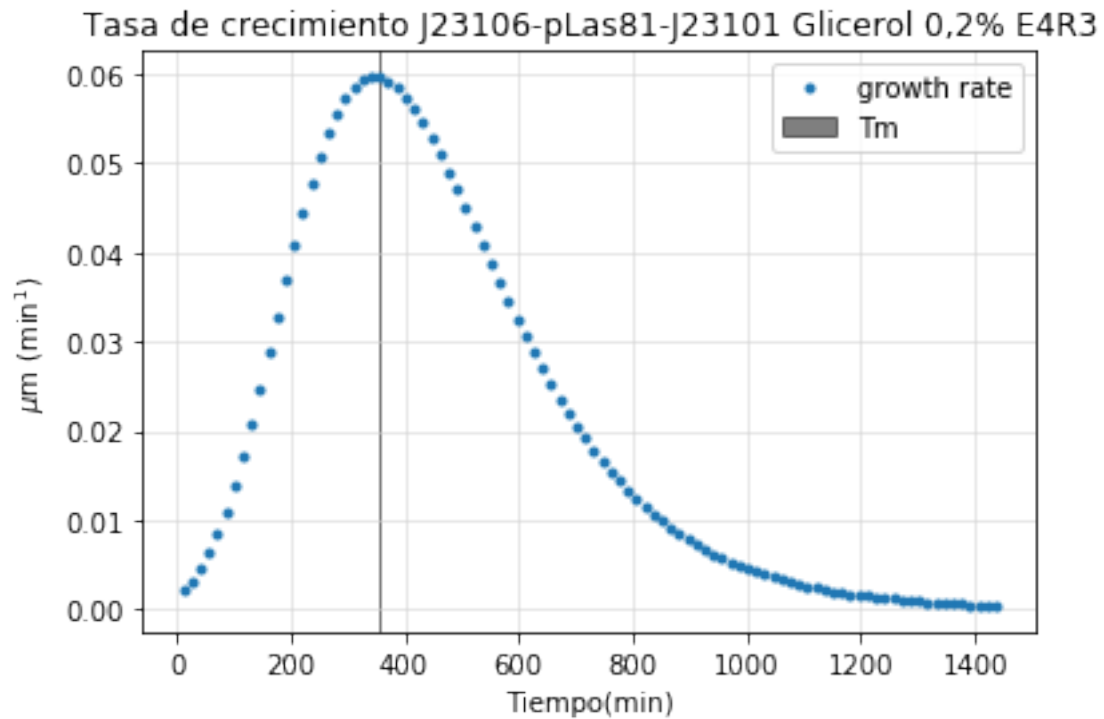
```
In [245]: #tasa de crecimiento
ye35=((A35*np.exp(-np.exp((((um35*np.exp(1))/A35)*(135-tt))+1))))
#Con diff
dy35=(np.diff(ye35))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm35,tm35, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy35,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[245]: <matplotlib.legend.Legend at 0x178a95a9b38>
```



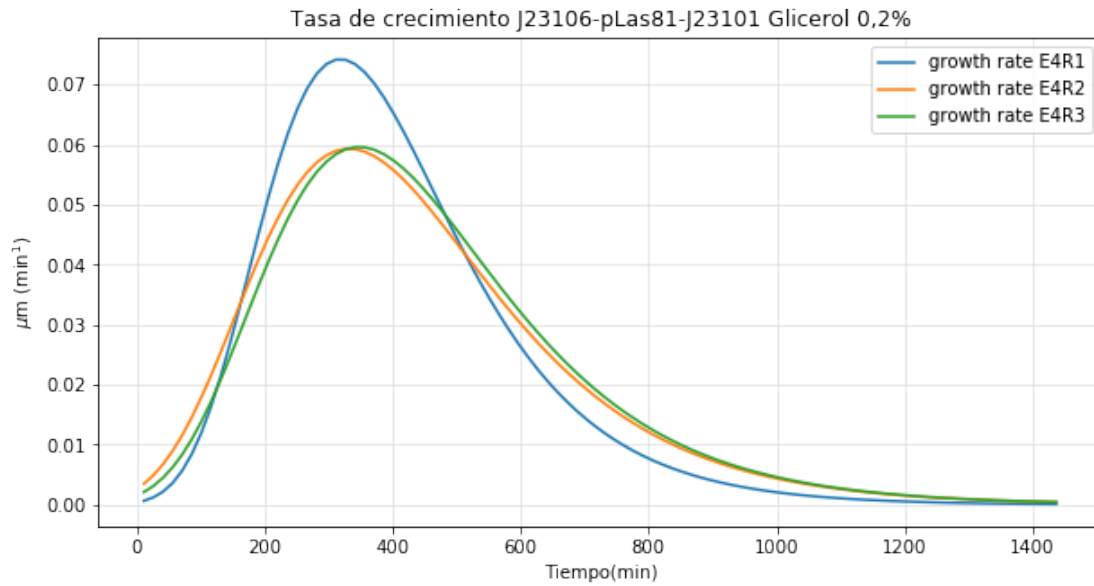
```
In [246]: #tasa de crecimiento
ye36=((A36*np.exp(-np.exp((((um36*np.exp(1))/A36)*(136-tt))+1))))
#Con diff
dy36=(np.diff(ye36))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.axvspan(tm36,tm36, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy36,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[246]: <matplotlib.legend.Legend at 0x178a9677940>
```



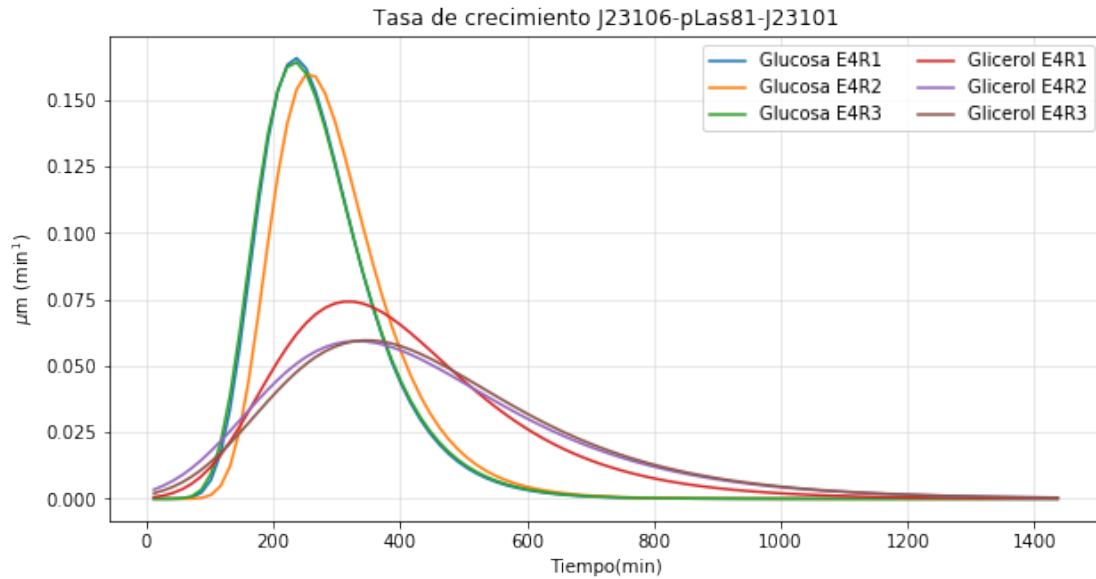
```
In [247]: #Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy34,label='growth rate E4R1')
plt.plot(tt[:-1],dy35,label='growth rate E4R2')
plt.plot(tt[:-1],dy36,label='growth rate E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[247]: <matplotlib.legend.Legend at 0x178a9743ac8>
```



```
In [248]: #Tasas control réplicas
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento J23106-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy31,label='Glucosa E4R1')
plt.plot(tt[:-1],dy32,label='Glucosa E4R2')
plt.plot(tt[:-1],dy33,label='Glucosa E4R3')
plt.plot(tt[:-1],dy34,label='Glicerol E4R1')
plt.plot(tt[:-1],dy35,label='Glicerol E4R2')
plt.plot(tt[:-1],dy36,label='Glicerol E4R3')
plt.grid(color='lightgray', linestyle='--', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

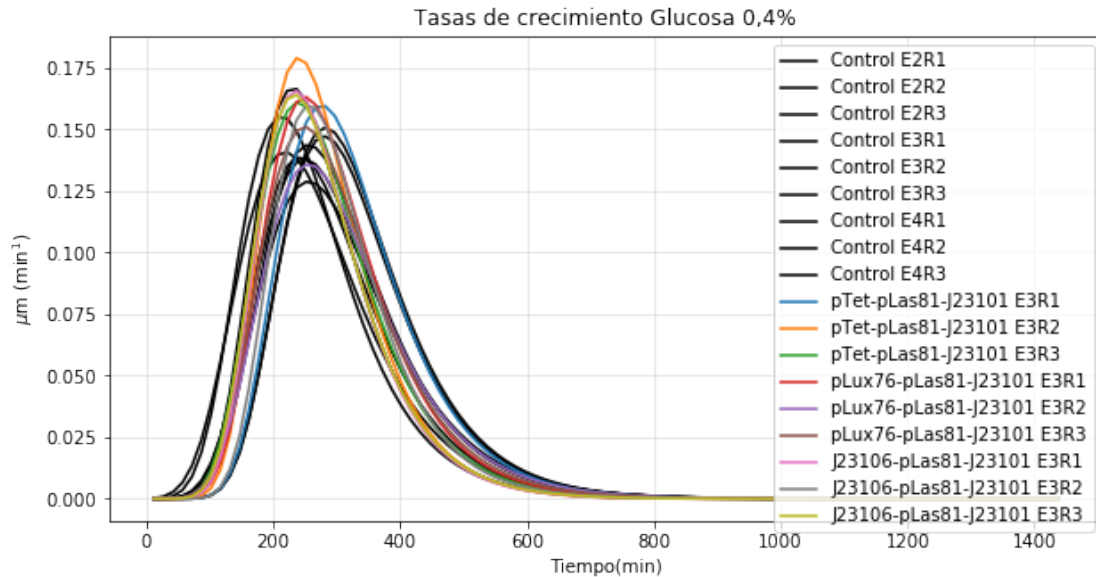
Out[248]: <matplotlib.legend.Legend at 0x178a9841748>



```
In [249]: #Tasas réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy1,'k',label='Control E2R1')
plt.plot(tt[:-1],dy2,'k',label='Control E2R2')
plt.plot(tt[:-1],dy3,'k',label='Control E2R3')
plt.plot(tt[:-1],dy7,'k',label='Control E3R1')
plt.plot(tt[:-1],dy8,'k',label='Control E3R2')
plt.plot(tt[:-1],dy9,'k',label='Control E3R3')
plt.plot(tt[:-1],dy13,'k',label='Control E4R1')
plt.plot(tt[:-1],dy14,'k',label='Control E4R2')
plt.plot(tt[:-1],dy15,'k',label='Control E4R3')
plt.plot(tt[:-1],dy19,label='pTet-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy20,label='pTet-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy21,label='pTet-pLas81-J23101 E3R3')
plt.plot(tt[:-1],dy25,label='pLux76-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy26,label='pLux76-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy27,label='pLux76-pLas81-J23101 E3R3')
plt.plot(tt[:-1],dy31,label='J23106-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy32,label='J23106-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy33,label='J23106-pLas81-J23101 E3R3')

plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

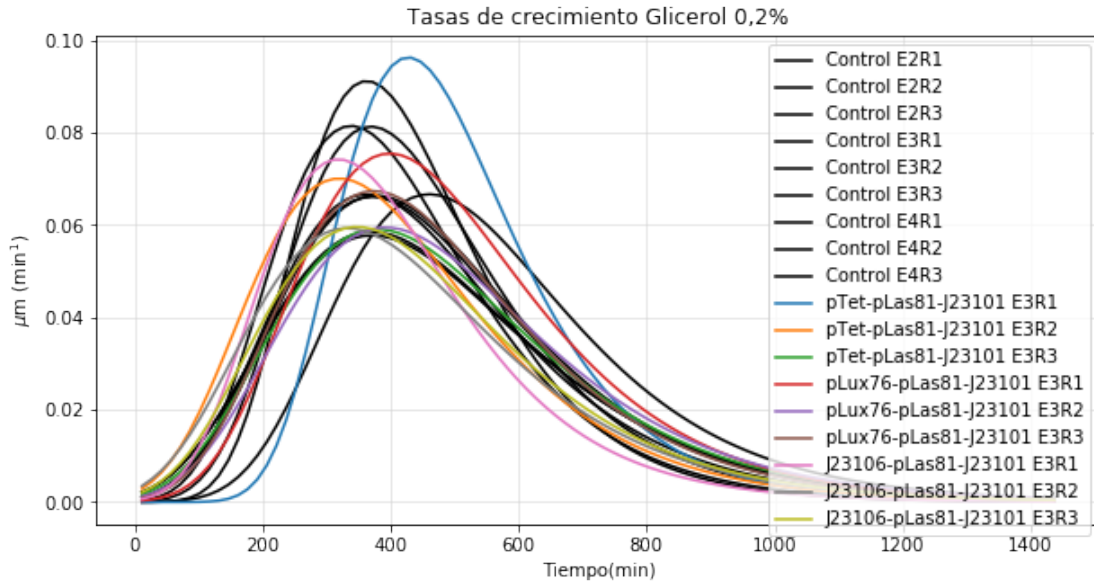
```
Out[249]: <matplotlib.legend.Legend at 0x178a994fe48>
```



```
In [250]: #Tasas réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy4,'k',label='Control E2R1')
plt.plot(tt[:-1],dy5,'k',label='Control E2R2')
plt.plot(tt[:-1],dy6,'k',label='Control E2R3')
plt.plot(tt[:-1],dy10,'k',label='Control E3R1')
plt.plot(tt[:-1],dy11,'k',label='Control E3R2')
plt.plot(tt[:-1],dy12,'k',label='Control E3R3')
plt.plot(tt[:-1],dy16,'k',label='Control E4R1')
plt.plot(tt[:-1],dy17,'k',label='Control E4R2')
plt.plot(tt[:-1],dy18,'k',label='Control E4R3')
plt.plot(tt[:-1],dy22,label='pTet-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy23,label='pTet-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy24,label='pTet-pLas81-J23101 E3R3')
plt.plot(tt[:-1],dy28,label='pLux76-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy29,label='pLux76-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy30,label='pLux76-pLas81-J23101 E3R3')
plt.plot(tt[:-1],dy34,label='J23106-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy35,label='J23106-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy36,label='J23106-pLas81-J23101 E3R3')

plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

```
Out[250]: <matplotlib.legend.Legend at 0x178a9ab8d30>
```



```
In [251]: #Tasas rélicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$ (min$^{-1}$)')
plt.plot(tt[:-1],dy1,'k',label='Control Glucosa E2R1')
plt.plot(tt[:-1],dy2,'k',label='Control Glucosa E2R2')
plt.plot(tt[:-1],dy3,'k',label='Control Glucosa E2R3')
plt.plot(tt[:-1],dy7,'k',label='Control Glucosa E3R1')
plt.plot(tt[:-1],dy8,'k',label='Control Glucosa E3R2')
plt.plot(tt[:-1],dy9,'k',label='Control Glucosa E3R3')
plt.plot(tt[:-1],dy13,'k',label='Control Glucosa E4R1')
plt.plot(tt[:-1],dy14,'k',label='Control Glucosa E4R2')
plt.plot(tt[:-1],dy15,'k',label='Control Glucosa E4R3')
plt.plot(tt[:-1],dy19,label='pTet-pLas81-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy20,label='pTet-pLas81-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy21,label='pTet-pLas81-J23101 Glucosa E3R3')
plt.plot(tt[:-1],dy25,label='pLux76-pLas81-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy26,label='pLux76-pLas81-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy27,label='pLux76-pLas81-J23101 Glucosa E3R3')
plt.plot(tt[:-1],dy31,label='J23106-pLas81-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy32,label='J23106-pLas81-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy33,label='J23106-pLas81-J23101 Glucosa E3R3')

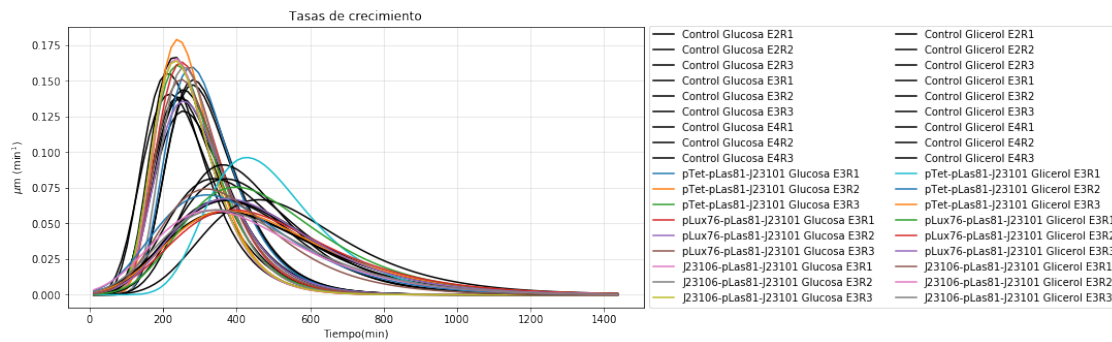
plt.plot(tt[:-1],dy4,'k',label='Control Glicerol E2R1')
plt.plot(tt[:-1],dy5,'k',label='Control Glicerol E2R2')
plt.plot(tt[:-1],dy6,'k',label='Control Glicerol E2R3')
```

```

plt.plot(tt[:-1],dy10,'k',label='Control Glicerol E3R1')
plt.plot(tt[:-1],dy11,'k',label='Control Glicerol E3R2')
plt.plot(tt[:-1],dy12,'k',label='Control Glicerol E3R3')
plt.plot(tt[:-1],dy16,'k',label='Control Glicerol E4R1')
plt.plot(tt[:-1],dy17,'k',label='Control Glicerol E4R2')
plt.plot(tt[:-1],dy18,'k',label='Control Glicerol E4R3')
plt.plot(tt[:-1],dy22,label='pTet-pLas81-J23101 Glicerol E3R1')
plt.plot(tt[:-1],dy23,label='pTet-pLas81-J23101 Glicerol E3R2')
plt.plot(tt[:-1],dy24,label='pTet-pLas81-J23101 Glicerol E3R3')
plt.plot(tt[:-1],dy28,label='pLux76-pLas81-J23101 Glicerol E3R1')
plt.plot(tt[:-1],dy29,label='pLux76-pLas81-J23101 Glicerol E3R2')
plt.plot(tt[:-1],dy30,label='pLux76-pLas81-J23101 Glicerol E3R3')
plt.plot(tt[:-1],dy34,label='J23106-pLas81-J23101 Glicerol E3R1')
plt.plot(tt[:-1],dy35,label='J23106-pLas81-J23101 Glicerol E3R2')
plt.plot(tt[:-1],dy36,label='J23106-pLas81-J23101 Glicerol E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

```

Out[251]: <matplotlib.legend.Legend at 0x178aac2fc50>



In [ ]: