# Ensayo 2 todo (pTet)

February 13, 2018

```
In [2]: 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 [3]: tt=np.fromfile('t', sep=',')

        #arrays replicas glucosa
        cfp15211=np.fromfile('p1521gCFP1', sep=',')
        rfp15211=np.fromfile('p1521gRFP1', sep=',')
        yfp15211=np.fromfile('p1521gYFP1', sep=',')
        od15211=np.fromfile('p1521gOD1', sep=',')
        cfp15212=np.fromfile('p1521gCFP2', sep=',')
        rfp15212=np.fromfile('p1521gRFP2', sep=',')
        yfp15212=np.fromfile('p1521gYFP2', sep=',')
        od15212=np.fromfile('p1521gOD2', sep=',')
        cfp15213=np.fromfile('p1521gCFP3', sep=',')
        rfp15213=np.fromfile('p1521gRFP3', sep=',')
        yfp15213=np.fromfile('p1521gYFP3', sep=',')
        od15213=np.fromfile('p1521gOD3', sep=',')
        '''
        print(cfp15211.shape)
        print(rfp15211.shape)
        print(yfp15211.shape)
        print(od15211.shape)
        print(cfp15212.shape)
        print(rfp15212.shape)
        print(yfp15212.shape)
        print(od15212.shape)
        print(cfp15213.shape)
        print(rfp15213.shape)
        print(yfp15213.shape)
```

```
print(od15213.shape)'''


cfp15231=np.fromfile('p1523gCFP1', sep=',')
rfp15231=np.fromfile('p1523gRFP1', sep=',')
yfp15231=np.fromfile('p1523gYFP1', sep=',')
od15231=np.fromfile('p1523gOD1', sep=',')
cfp15232=np.fromfile('p1523gCFP2', sep=',')
rfp15232=np.fromfile('p1523gRFP2', sep=',')
yfp15232=np.fromfile('p1523gYFP2', sep=',')
od15232=np.fromfile('p1523gOD2', sep=',')
cfp15233=np.fromfile('p1523gCFP3', sep=',')
rfp15233=np.fromfile('p1523gRFP3', sep=',')
yfp15233=np.fromfile('p1523gYFP3', sep=',')
od15233=np.fromfile('p1523gOD3', sep=',')

'''
print(cfp15231.shape)
print(rfp15231.shape)
print(yfp15231.shape)
print(od15231.shape)
print(cfp15232.shape)
print(rfp15232.shape)
print(yfp15232.shape)
print(od15232.shape)
print(cfp15233.shape)
print(rfp15233.shape)
print(yfp15233.shape)
print(od15233.shape)'''

cfp15261=np.fromfile('p1526gCFP1', sep=',')
rfp15261=np.fromfile('p1526gRFP1', sep=',')
yfp15261=np.fromfile('p1526gYFP1', sep=',')
od15261=np.fromfile('p1526gOD1', sep=',')
cfp15262=np.fromfile('p1526gCFP2', sep=',')
rfp15262=np.fromfile('p1526gRFP2', sep=',')
yfp15262=np.fromfile('p1526gYFP2', sep=',')
od15262=np.fromfile('p1526gOD2', sep=',')
cfp15263=np.fromfile('p1526gCFP3', sep=',')
rfp15263=np.fromfile('p1526gRFP3', sep=',')
yfp15263=np.fromfile('p1526gYFP3', sep=',')
od15263=np.fromfile('p1526gOD3', sep=',')
'''
print(cfp15261.shape)
print(rfp15261.shape)
print(yfp15261.shape)
print(od15261.shape)
print(cfp15262.shape)
```

```
print(rfp15262.shape)
print(yfp15262.shape)
print(od15262.shape)
print(cfp15263.shape)
print(rfp15263.shape)
print(yfp15263.shape)
print(od15263.shape)'''

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)'''

#Controles
#Promedios controles glucosa
cfpcg1=np.fromfile('pcgCFP1', sep=',')
rfpcg1=np.fromfile('pcgRFP1', sep=',')
yfpcg1=np.fromfile('pcgYFP1', sep=',')
odcg1=np.fromfile('pcgOD1', sep=',')
cfpcg2=np.fromfile('pcgCFP2', sep=',')
rfpcg2=np.fromfile('pcgRFP2', sep=',')
yfpcg2=np.fromfile('pcgYFP2', sep=',')
odcg2=np.fromfile ('pcgOD2',sep=',')
cfpcg3=np.fromfile('pcgCFP3', sep=',')
rfpcg3=np.fromfile('pcgRFP3', sep=',')
yfpcg3=np.fromfile('pcgYFP3', sep=',')
```

```
odcg3=np.fromfile('pcgOD3', sep=',')
'''
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)'''
```

Out[3]: '\nprint(cfpcg1.shape)\nprint(rfpcg1.shape)\nprint(yfpcg1.shape)\nprint(odcg1.shape)\npr

In [4]:
```
#Promedios glicerol
#arrays replicas glicerol
cfp1521g1=np.fromfile('p1521glCFP1', sep=',')
rfp1521g1=np.fromfile('p1521glRFP1', sep=',')
yfp1521g1=np.fromfile('p1521glYFP1', sep=',')
od1521g1=np.fromfile('p1521glOD1', sep=',')
cfp1521g2=np.fromfile('p1521glCFP2', sep=',')
rfp1521g2=np.fromfile('p1521glRFP2', sep=',')
yfp1521g2=np.fromfile('p1521glYFP2', sep=',')
od1521g2=np.fromfile('p1521glOD2', sep=',')
cfp1521g3=np.fromfile('p1521glCFP3', sep=',')
rfp1521g3=np.fromfile('p1521glRFP3', sep=',')
yfp1521g3=np.fromfile('p1521glYFP3', sep=',')
od1521g3=np.fromfile('p1521glOD3', sep=',')
'''
print(cfp1521g1.shape)
print(rfp1521g1.shape)
print(yfp1521g1.shape)
print(od1521g1.shape)
print(cfp1521g2.shape)
print(rfp1521g2.shape)
print(yfp1521g2.shape)
print(od1521g2.shape)
print(cfp1521g3.shape)
print(rfp1521g3.shape)
print(yfp1521g3.shape)
print(od1521g3.shape)'''


cfp1523g1=np.fromfile('p1523glCFP1', sep=',')
rfp1523g1=np.fromfile('p1523glRFP1', sep=',')
```

```python
yfp1523g1=np.fromfile('p1523glYFP1', sep=',')
od1523g1=np.fromfile('p1523glOD1', sep=',')
cfp1523g2=np.fromfile('p1523glCFP2', sep=',')
rfp1523g2=np.fromfile('p1523glRFP2', sep=',')
yfp1523g2=np.fromfile('p1523glYFP2', sep=',')
od1523g2=np.fromfile('p1523glOD2', sep=',')
cfp1523g3=np.fromfile('p1523glCFP3', sep=',')
rfp1523g3=np.fromfile('p1523glRFP3', sep=',')
yfp1523g3=np.fromfile('p1523glYFP3', sep=',')
od1523g3=np.fromfile('p1523glOD3', sep=',')

'''
print(cfp1523g1.shape)
print(rfp1523g1.shape)
print(yfp1523g1.shape)
print(od1523g1.shape)
print(cfp1523g2.shape)
print(rfp1523g2.shape)
print(yfp1523g2.shape)
print(od1523g2.shape)
print(cfp1523g3.shape)
print(rfp1523g3.shape)
print(yfp1523g3.shape)
print(od1523g3.shape)'''


cfp1526g1=np.fromfile('p1526glCFP1', sep=',')
rfp1526g1=np.fromfile('p1526glRFP1', sep=',')
yfp1526g1=np.fromfile('p1526glYFP1', sep=',')
od1526g1=np.fromfile('p1526glOD1', sep=',')
cfp1526g2=np.fromfile('p1526glCFP2', sep=',')
rfp1526g2=np.fromfile('p1526glRFP2', sep=',')
yfp1526g2=np.fromfile('p1526glYFP2', sep=',')
od1526g2=np.fromfile('p1526glOD2', sep=',')
cfp1526g3=np.fromfile('p1526glCFP3', sep=',')
rfp1526g3=np.fromfile('p1526glRFP3', sep=',')
yfp1526g3=np.fromfile('p1526glYFP3', sep=',')
od1526g3=np.fromfile('p1526glOD3', sep=',')
'''
print(cfp1526g1.shape)
print(rfp1526g1.shape)
print(yfp1526g1.shape)
print(od1526g1.shape)
print(cfp1526g2.shape)
print(rfp1526g2.shape)
print(yfp1526g2.shape)
print(od1526g2.shape)
print(cfp1526g3.shape)
```

```
print(rfp1526g3.shape)
print(yfp1526g3.shape)
print(od1526g3.shape)'''

cfp1527g1=np.fromfile('p1527glCFP1', sep=',')
rfp1527g1=np.fromfile('p1527glRFP1', sep=',')
yfp1527g1=np.fromfile('p1527glYFP1', sep=',')
od1527g1=np.fromfile('p1527glOD1', sep=',')
cfp1527g2=np.fromfile('p1527glCFP2', sep=',')
rfp1527g2=np.fromfile('p1527glRFP2', sep=',')
yfp1527g2=np.fromfile('p1527glYFP2', sep=',')
od1527g2=np.fromfile('p1527glOD2', sep=',')
cfp1527g3=np.fromfile('p1527glCFP3', sep=',')
rfp1527g3=np.fromfile('p1527glRFP3', sep=',')
yfp1527g3=np.fromfile('p1527glYFP3', sep=',')
od1527g3=np.fromfile('p1527glOD3', 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)'''


#Promedios controles glicerol
cfpcgl1=np.fromfile('pcglCFP1', sep=',')
rfpcgl1=np.fromfile('pcglRFP1', sep=',')
yfpcgl1=np.fromfile('pcglYFP1', sep=',')
odcgl1=np.fromfile('pcglOD1', sep=',')
cfpcgl2=np.fromfile('pcglCFP2', sep=',')
rfpcgl2=np.fromfile('pcglRFP2', sep=',')
yfpcgl2=np.fromfile('pcglYFP2', sep=',')
odcgl2=np.fromfile('pcglOD2', sep=',')
cfpcgl3=np.fromfile('pcglCFP3', sep=',')
rfpcgl3=np.fromfile('pcglRFP3', sep=',')
yfpcgl3=np.fromfile('pcglYFP3', sep=',')
odcgl3=np.fromfile('pcglOD3', sep=',')
'''
print(cfpcgl1.shape)
print(rfpcgl1.shape)
```

```
print(yfpcgl1.shape)
print(odcgl1.shape)
print(cfpcgl1.shape)
print(rfpcgl1.shape)
print(yfpcgl1.shape)
print(odcgl1.shape)
print(cfpcgl1.shape)
print(rfpcgl1.shape)
print(yfpcgl1.shape)
print(odcgl1.shape)'''
```

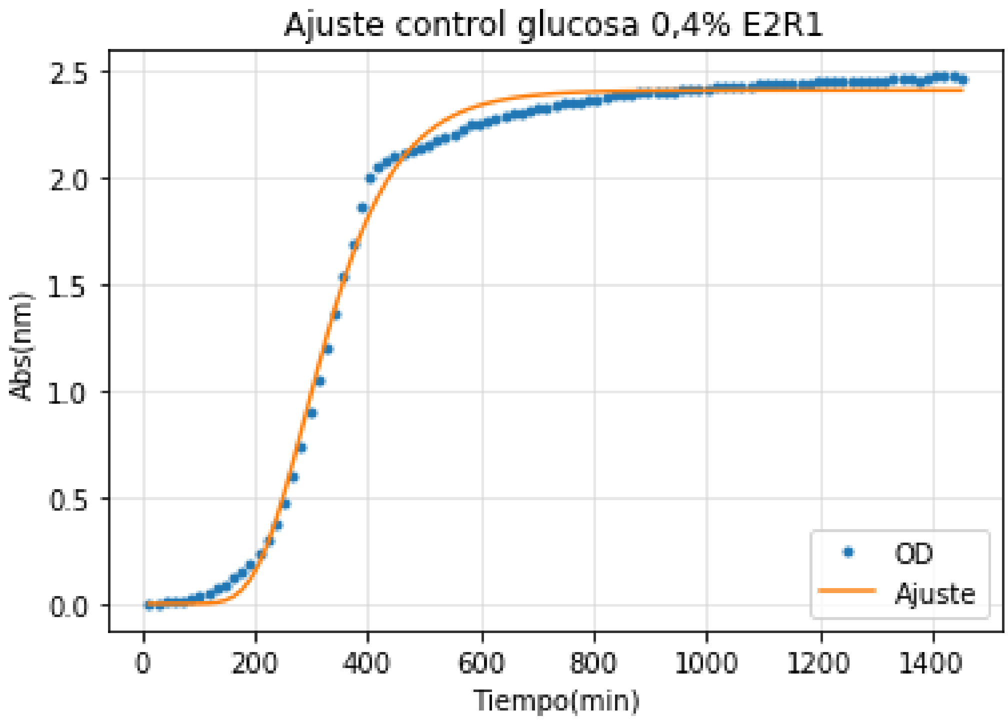
Out[4]: '\nprint(cfpcgl1.shape)\nprint(rfpcgl1.shape)\nprint(yfpcgl1.shape)\nprint(odcgl1.shape)

In [5]:
```
#Funciones para ajuste Gompertz
def F_sigma(t, A, um,l):
    return ((A*np.exp(-np.exp((((um*np.exp(1))/A)*(l-t))+1))))

def Function_fit(xdata,ydata,init,end,func=F_sigma,ParamBounds=([0,0,0],[3,1,300]), titl
        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 [6]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 1
y1 = np.log(odcgl1)-np.log(np.min(odcgl1))
print('Min OD = %e'%((np.min(odcgl1))))
evaly, params=Function_fit(tt,y1,0,-1,title = 'Ajuste control glucosa 0,4% E2R1')
A1 = params[0]
```

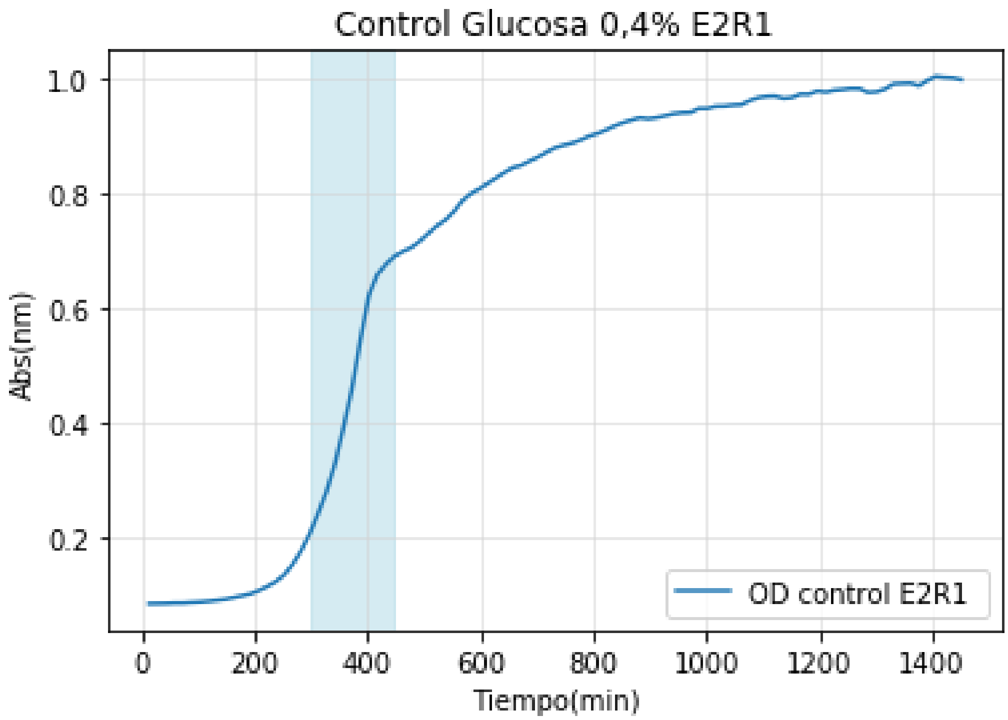
```
um1=params[1]
l1=params[2]
print('A=%e'%(A1))
print('um=%e'%(um1))
print('l=%e'%(l1))

#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,odcg1,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],odcg1[19:30],label='OD control E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

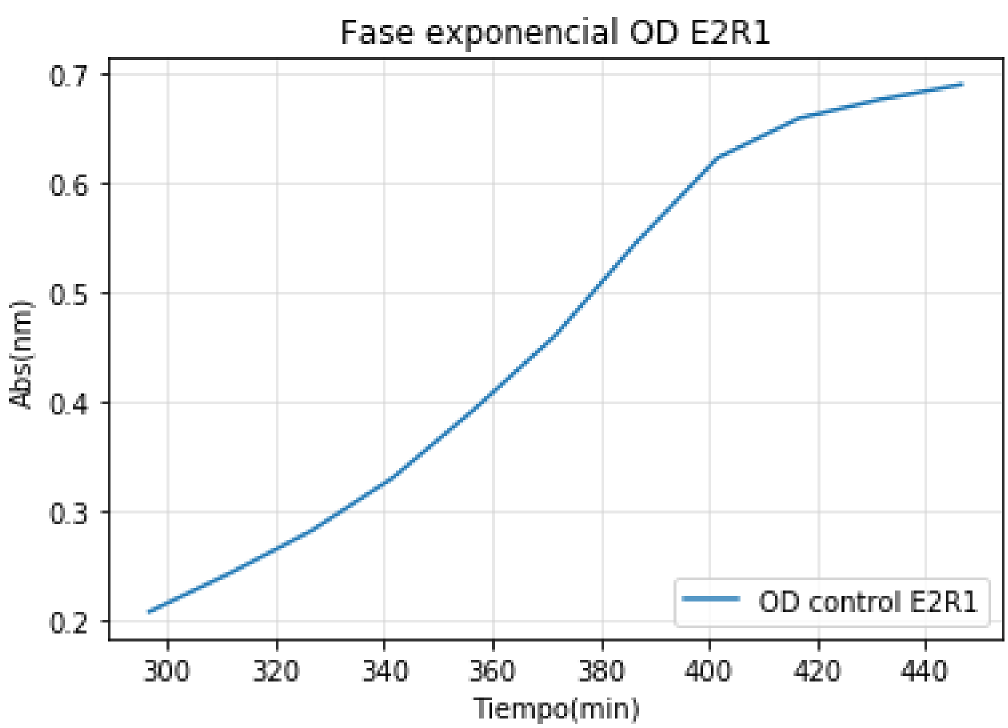
```
Min OD = 8.475000e-02
[  2.40766598e+00   1.00663667e-02   2.01163416e+02]
```




```
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
```

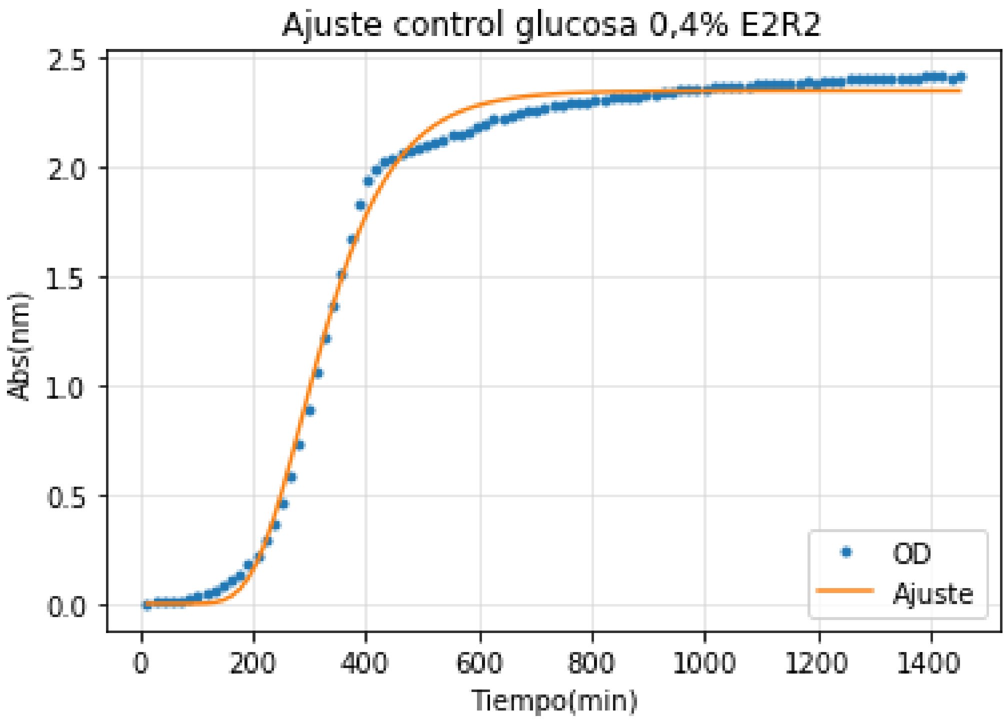
```
Out[6]: <matplotlib.legend.Legend at 0x267e9a887f0>
```


Control Glucosa 0,4% E2R1
Abs(nm)
Tiempo(min)
OD control E2R1
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E2R1
Abs(nm)
Tiempo(min)
OD control E2R1
0.2
0.3
0.4
0.5
0.6
0.7
300
320
340
360
380
400
420
440

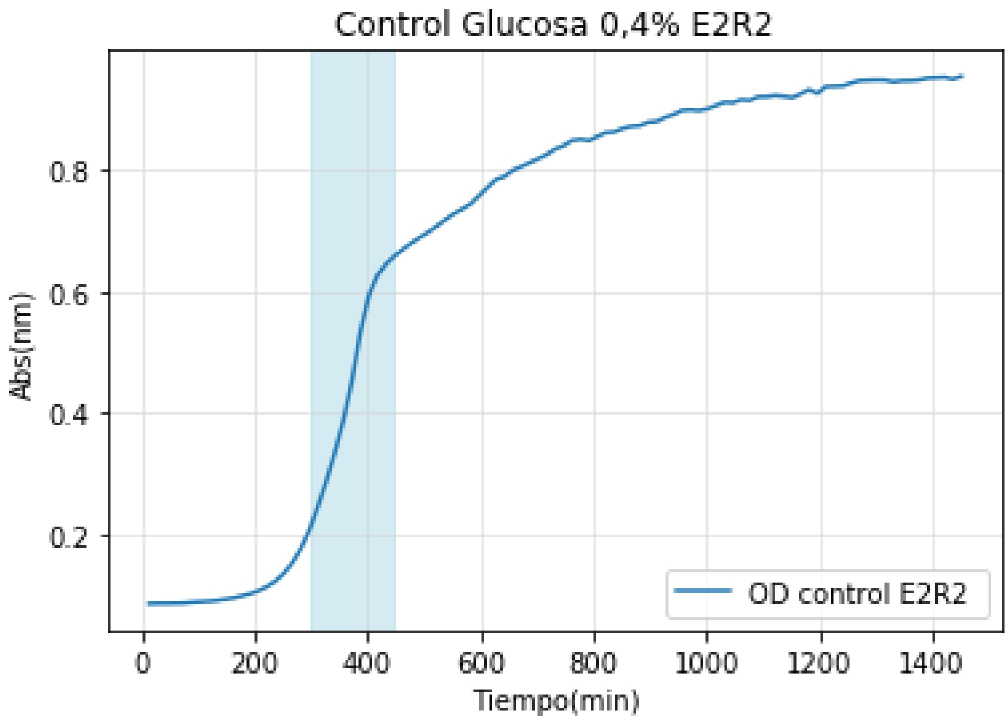
In [7]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y2= np.log(odcg2)-np.log(np.min(odcg2))
print('Min OD = %e'%((np.min(odcg2))))
evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E2R2')
A2 = params[0]
um2=params[1]
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,odcg2,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],odcg2[19:30],label='OD control E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

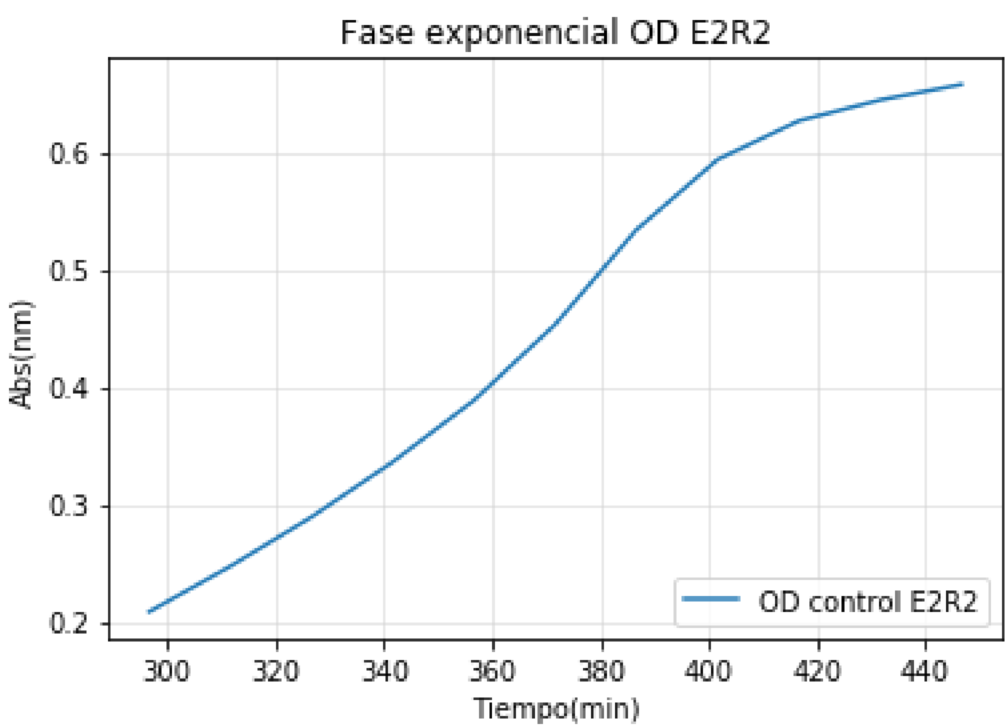
```
Min OD = 8.550000e-02
[  2.34701846e+00   9.83004768e-03   1.99784486e+02]
```




```
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
```

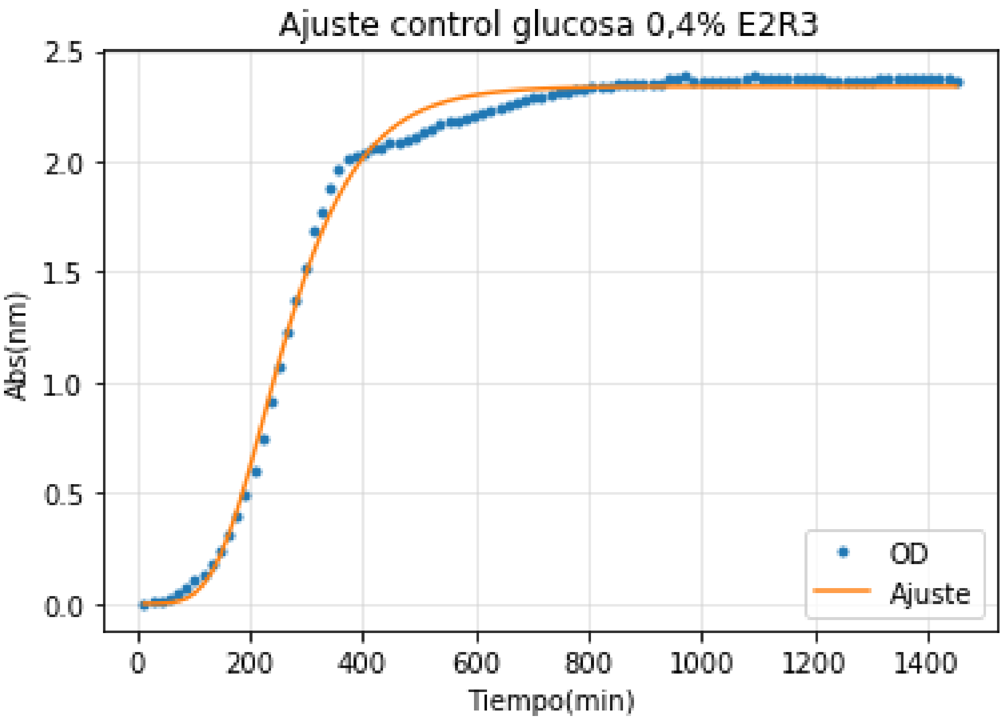
```
Out[7]: <matplotlib.legend.Legend at 0x267eac4b748>
```


Control Glucosa 0,4% E2R2
Abs(nm)
Tiempo(min)
OD control E2R2
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E2R2
Abs(nm)
Tiempo(min)
OD control E2R2
0.2
0.3
0.4
0.5
0.6
300
320
340
360
380
400
420
440

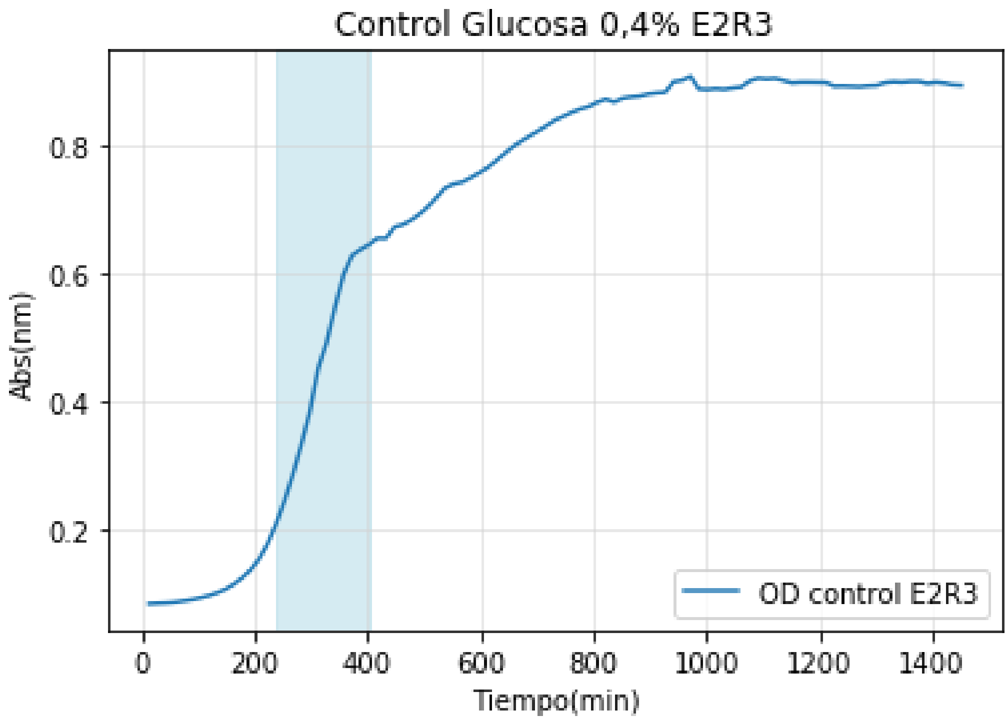
```
In [8]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control glucosa rep 3
        y3= np.log(odcg3)-np.log(np.min(odcg3))
        print('Min OD = %e'%((np.min(odcg3))))
        evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E2R3')
        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,odcg3,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],odcg3[15:27],label='OD control E2R3')
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')
```

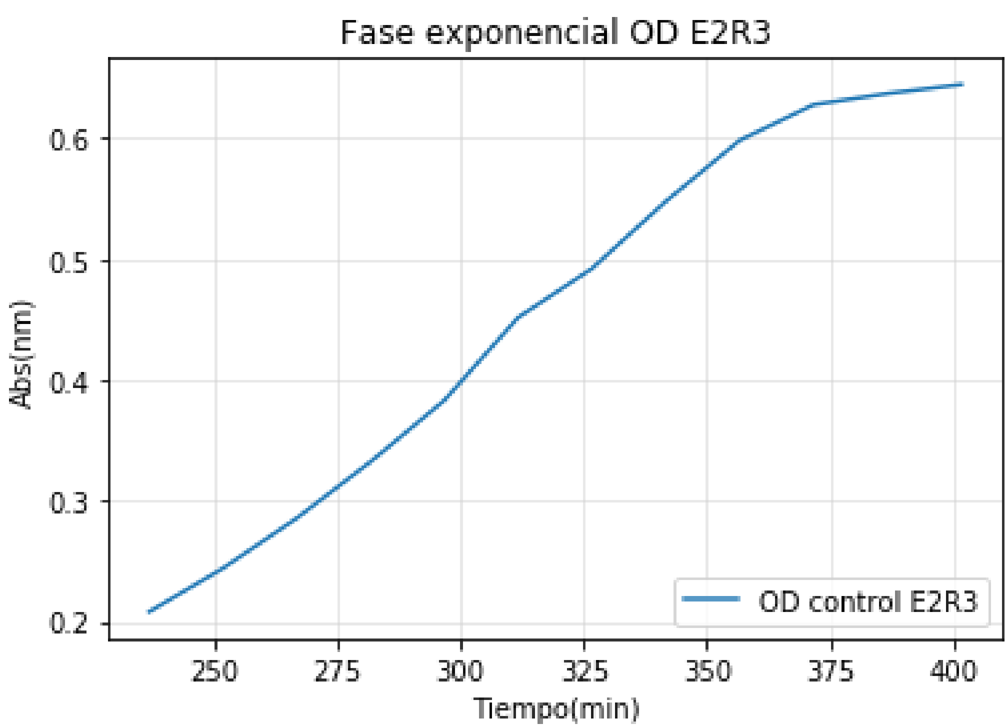
```
Min OD = 8.375000e-02
[  2.33929203e+00   9.40512328e-03   1.34262384e+02]
```



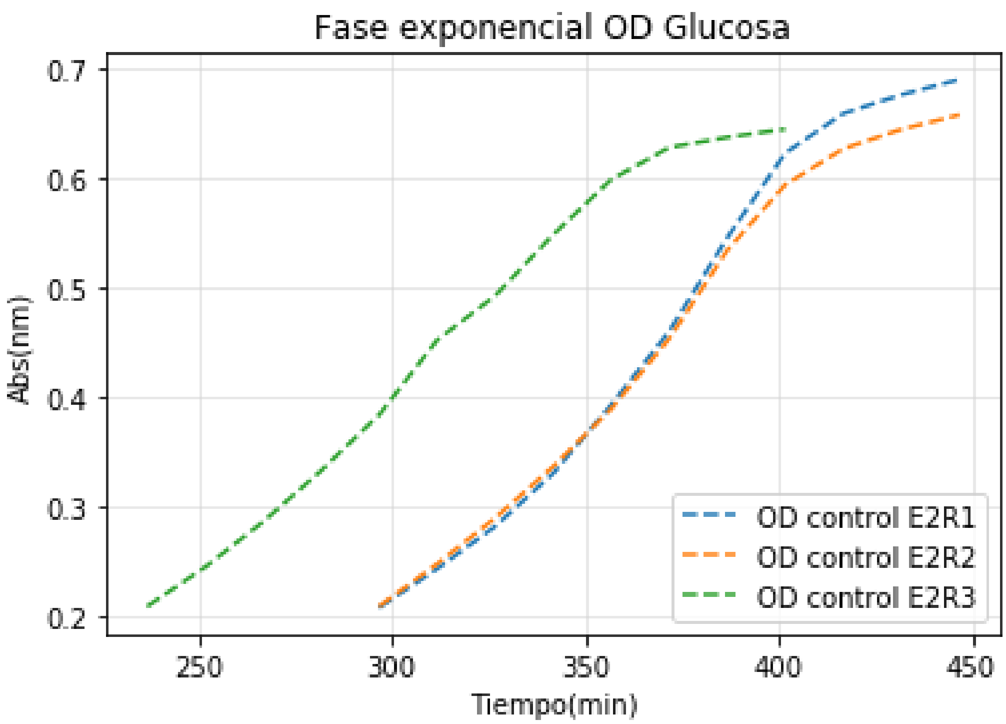

```
A=2.339292e+00
um=9.405123e-03
l=1.342624e+02
Tm=2.257633e+02
doubpe=7.369889e+01
ext=1.473978e+02
Tfinal=3.731611e+02
```

```
Out[8]: <matplotlib.legend.Legend at 0x267eae5cf98>
```


Control Glucosa 0,4% E2R3
Abs(nm)
Tiempo(min)
OD control E2R3
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E2R3
Abs(nm)
Tiempo(min)
OD control E2R3
0.2
0.3
0.4
0.5
0.6
250
275
300
325
350
375
400

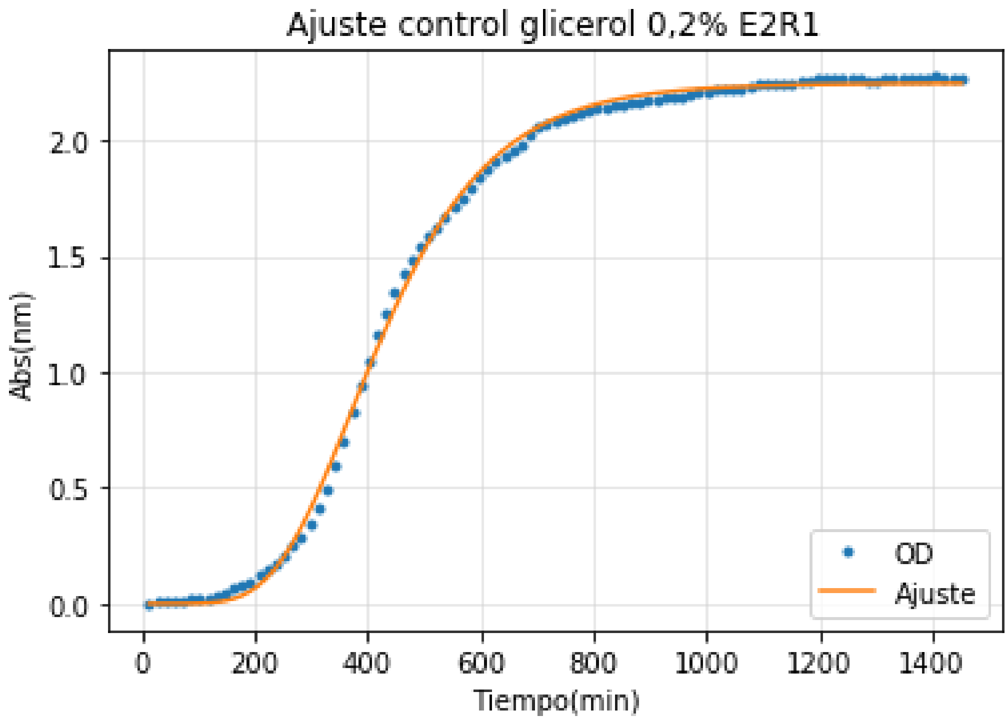
```
In [9]: #Fase exponencial OD/tiempo
        plt.figure()
        plt.title('Fase exponencial OD Glucosa')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt[19:30],odcg1[19:30],'--',label='OD control E2R1')
        plt.plot(tt[19:30],odcg2[19:30],'--',label='OD control E2R2')
        plt.plot(tt[15:27],odcg3[15:27],'--',label='OD control E2R3')
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')

Out[9]: <matplotlib.legend.Legend at 0x267eaf76f60>
```




```
In [10]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 1
         y4= np.log(odcgl1)-np.log(np.min(odcgl1))
         print('Min OD = %e'%((np.min(odcgl1))))
         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))
```

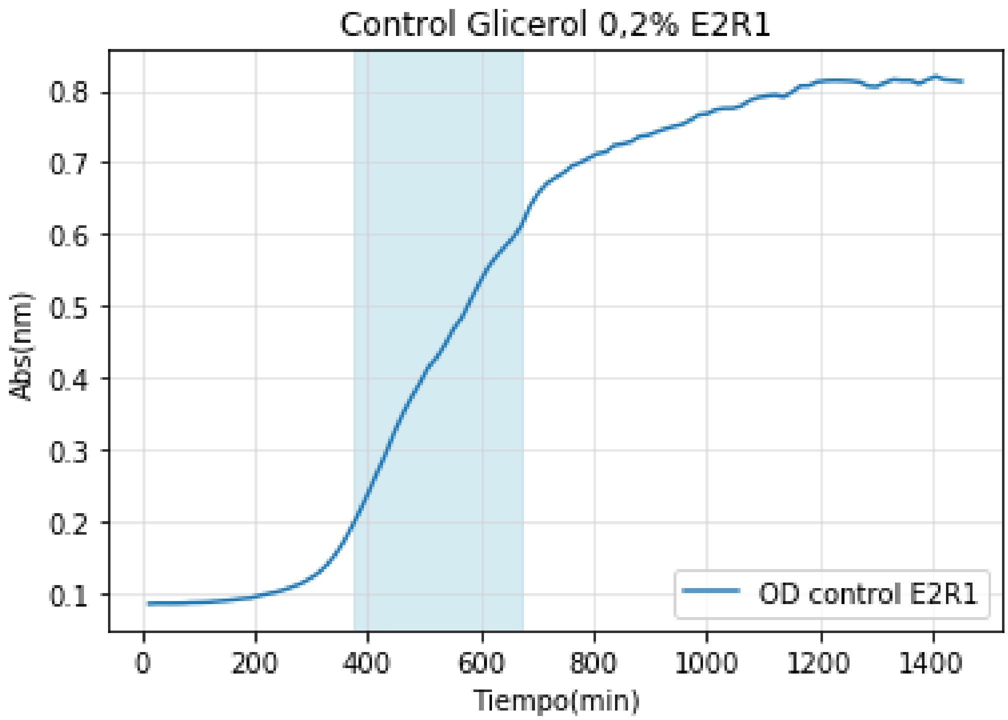
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+l4)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

#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,odcgl1,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],odcgl1[24:45],label='OD control E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

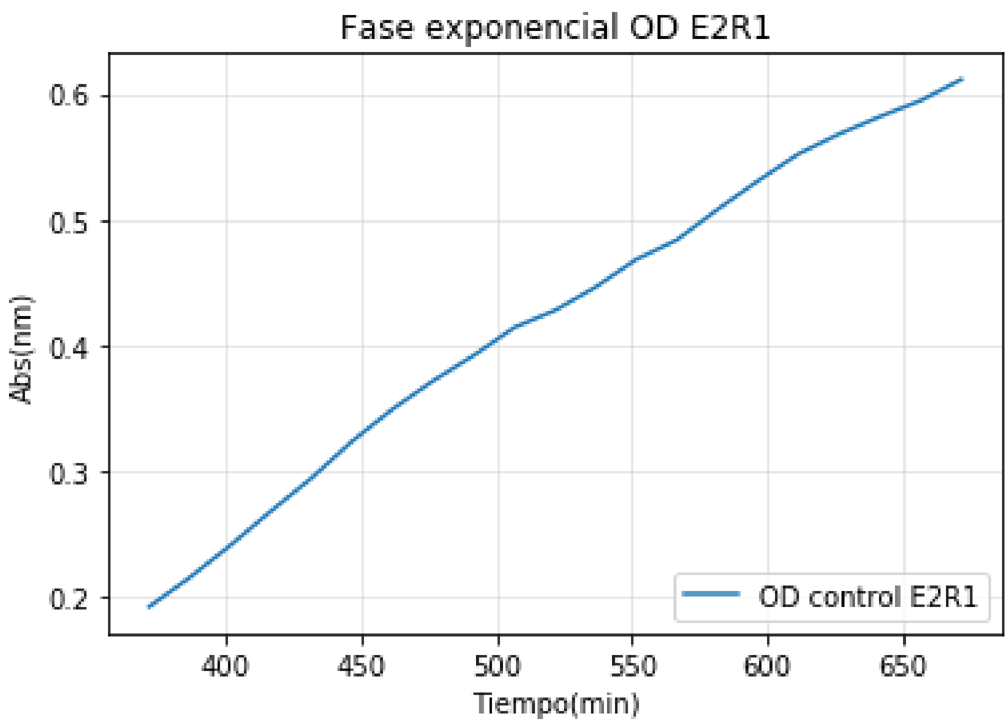
```
Min OD = 8.425000e-02
[  2.24595733e+00   6.08336131e-03   2.34248586e+02]
```




```
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
```

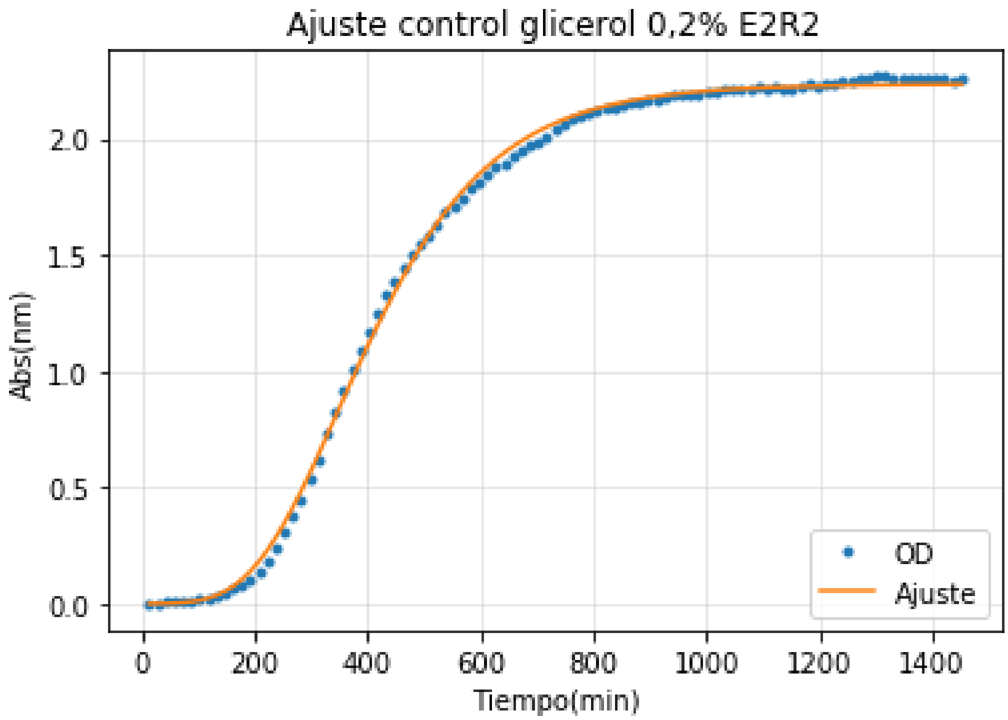
```
Out[10]: <matplotlib.legend.Legend at 0x267eaecc0f0>
```


Control Glicerol 0,2% E2R1
Abs(nm)
Tiempo(min)
OD control E2R1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E2R1
Abs(nm)
Tiempo(min)
OD control E2R1
0.2
0.3
0.4
0.5
0.6
400
450
500
550
600
650

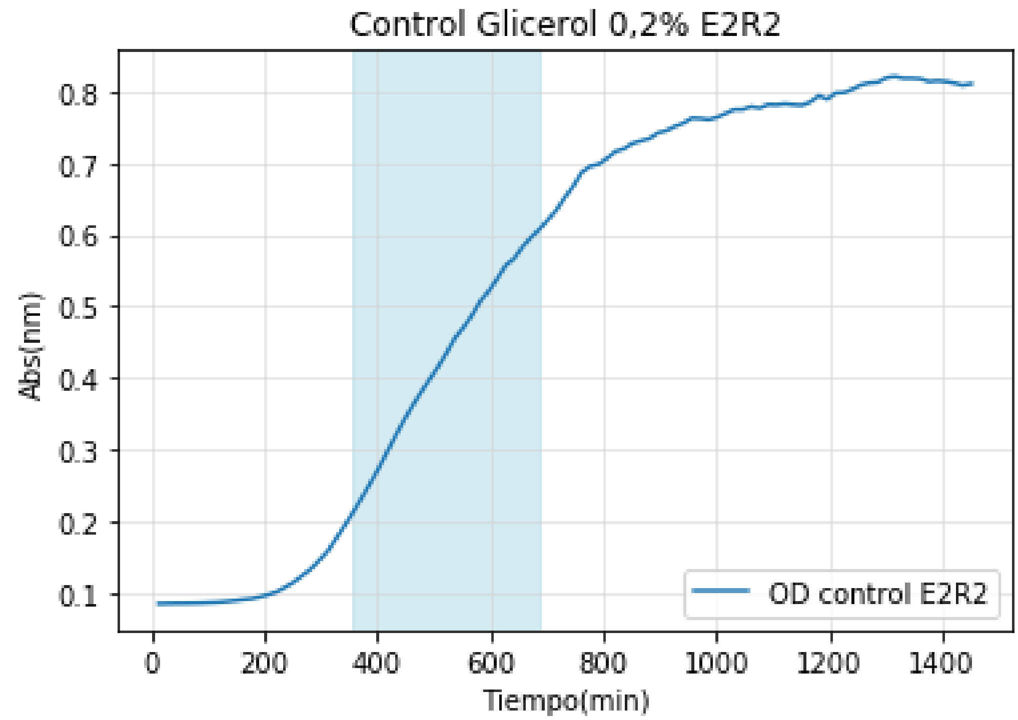
```
In [11]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerol rep 2
         y5= np.log(odcgl2)-np.log(np.min(odcgl2))
         print('Min OD = %e'%((np.min(odcgl2))))
         evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 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,odcgl2,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],odcgl2[23:46],label='OD control E2R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

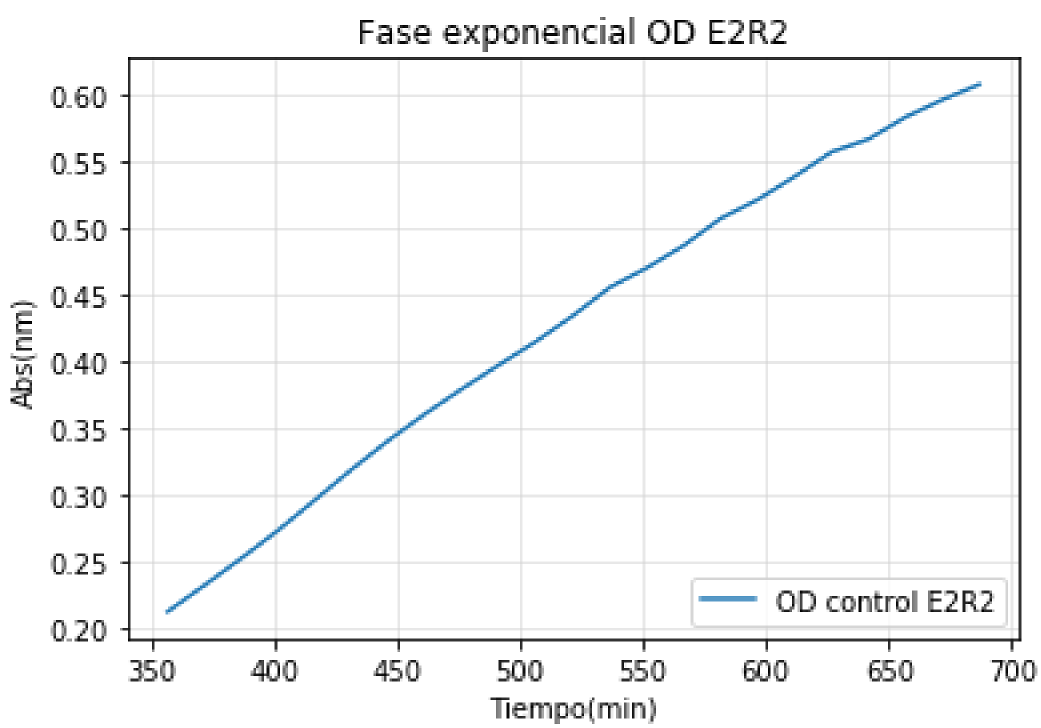
```
Min OD = 8.500000e-02
[  2.23521178e+00   5.43425313e-03   1.94280385e+02]
```




```
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
```

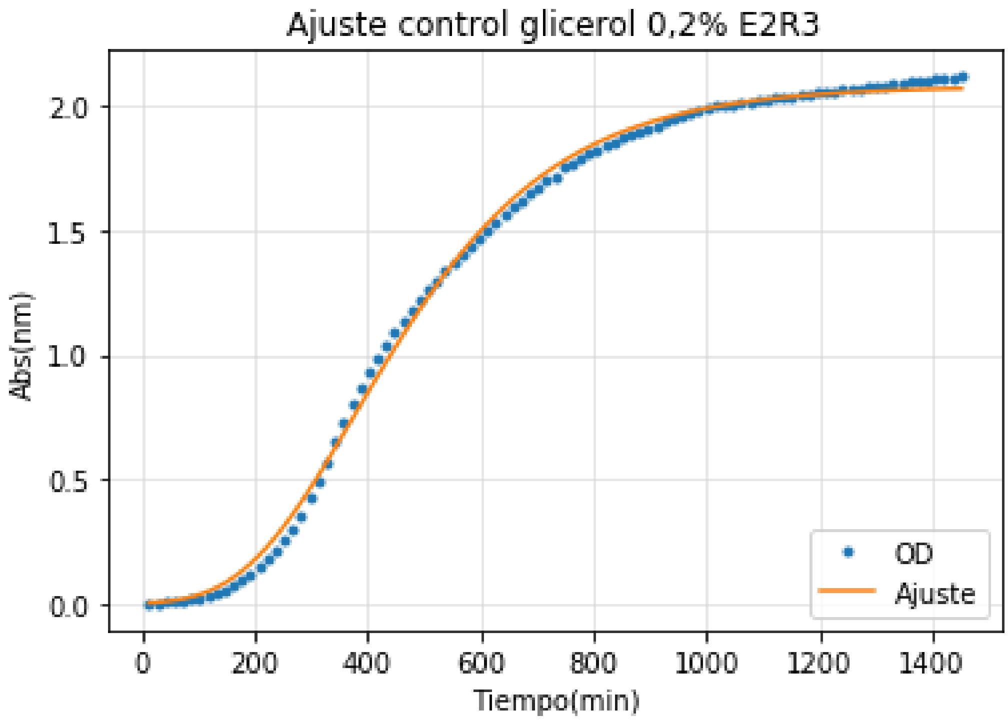
```
Out[11]: <matplotlib.legend.Legend at 0x267e9a637f0>
```


Control Glicerol 0,2% E2R2
Abs(nm)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E2R2


Fase exponencial OD E2R2
Abs(nm)
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
350
400
450
500
550
600
650
700
Tiempo(min)
OD control E2R2

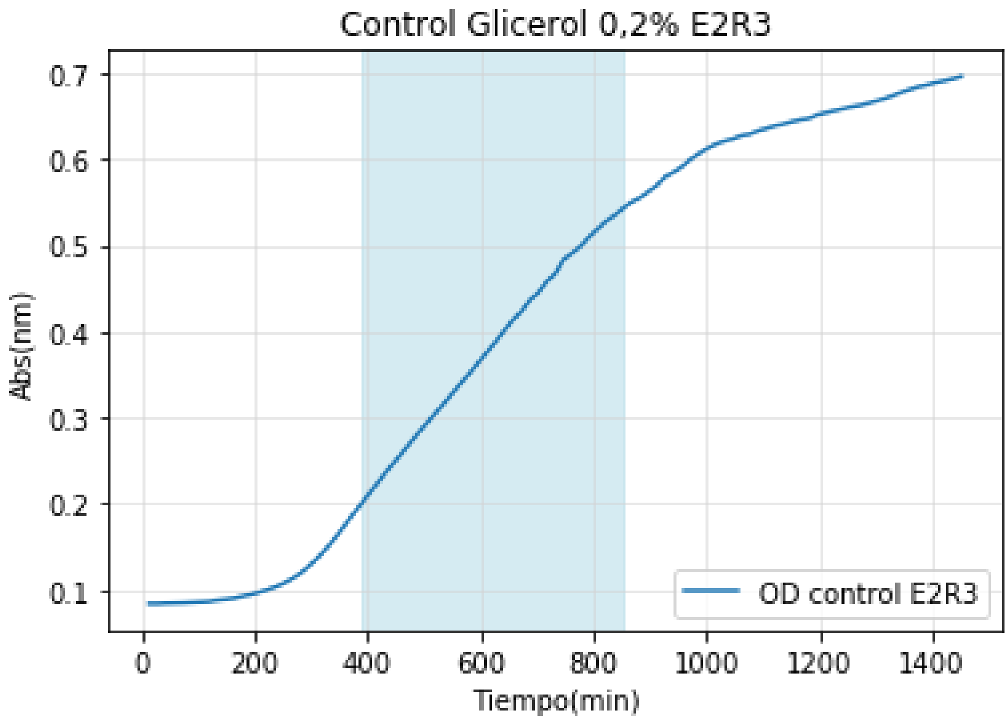
```
In [12]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerol rep 3
         y6= np.log(odcgl3)-np.log(np.min(odcgl3))
         print('Min OD = %e'%((np.min(odcgl3))))
         evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 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,odcgl3,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],odcgl3[25:57],label='OD control E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

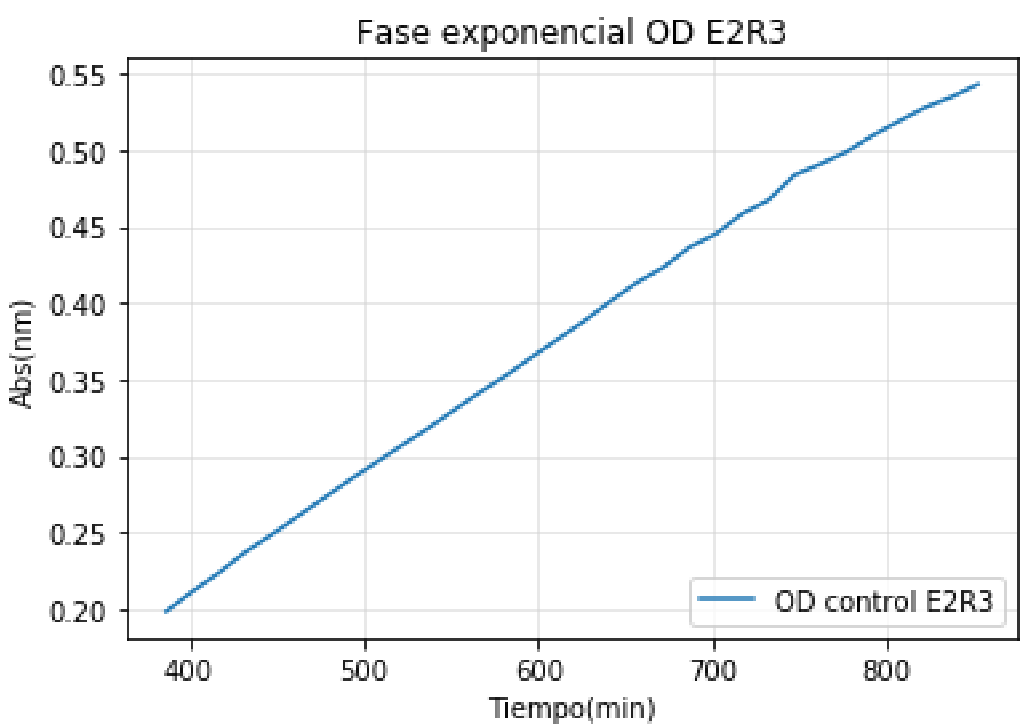
```
Min OD = 8.375000e-02
[  2.07943812e+00   3.85150055e-03   1.79490460e+02]
```




```
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
```

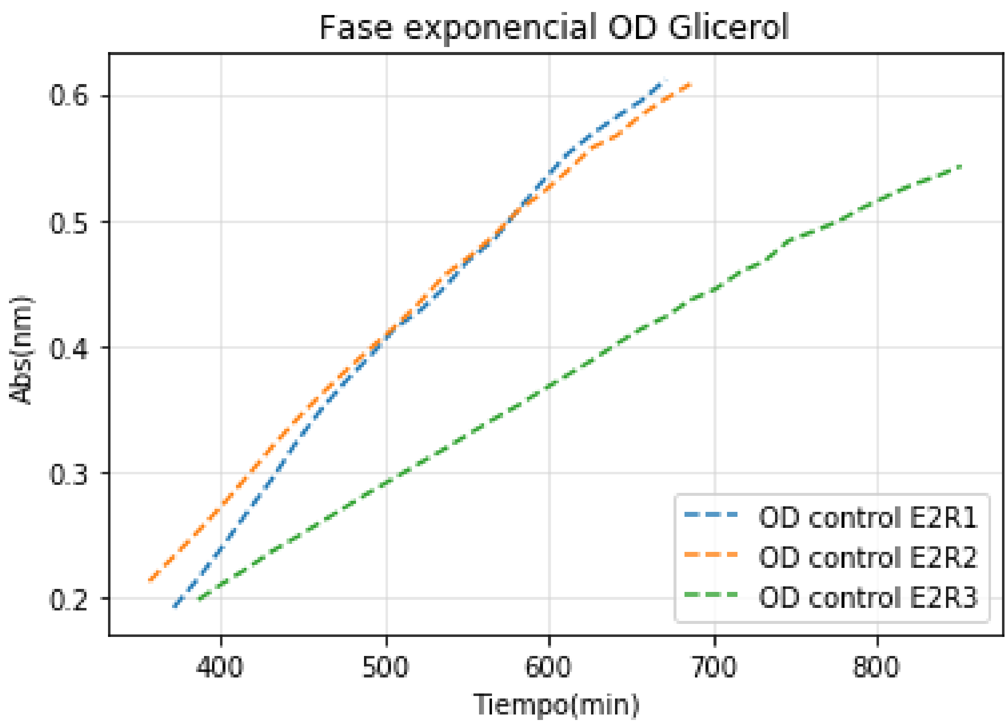
```
Out[12]: <matplotlib.legend.Legend at 0x267e9bbce48>
```


Control Glicerol 0,2% E2R3
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Abs(nm)
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E2R3


Fase exponencial OD E2R3
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
400
500
600
700
800
Tiempo(min)
OD control E2R3

```
In [13]: #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],odcgl1[24:45],'--',label='OD control E2R1')
         plt.plot(tt[23:46],odcgl2[23:46],'--',label='OD control E2R2')
         plt.plot(tt[25:57],odcgl3[25:57],'--',label='OD control E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

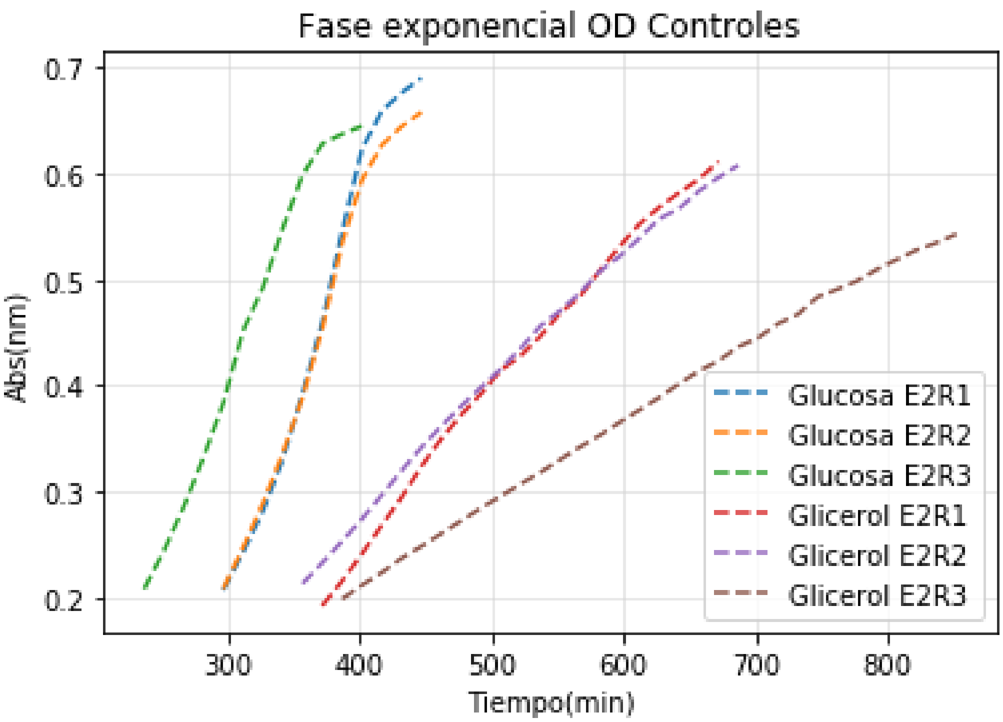
```
Out[13]: <matplotlib.legend.Legend at 0x267eb181e48>
```




```
In [14]: #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],odcg1[19:30],'--',label='Glucosa E2R1')
         plt.plot(tt[19:30],odcg2[19:30],'--',label='Glucosa E2R2')
         plt.plot(tt[15:27],odcg3[15:27],'--',label='Glucosa E2R3')
         plt.plot(tt[24:45],odcgl1[24:45],'--',label='Glicerol E2R1')
         plt.plot(tt[23:46],odcgl2[23:46],'--',label='Glicerol E2R2')
         plt.plot(tt[25:57],odcgl3[25:57],'--',label='Glicerol E2R3')
```

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

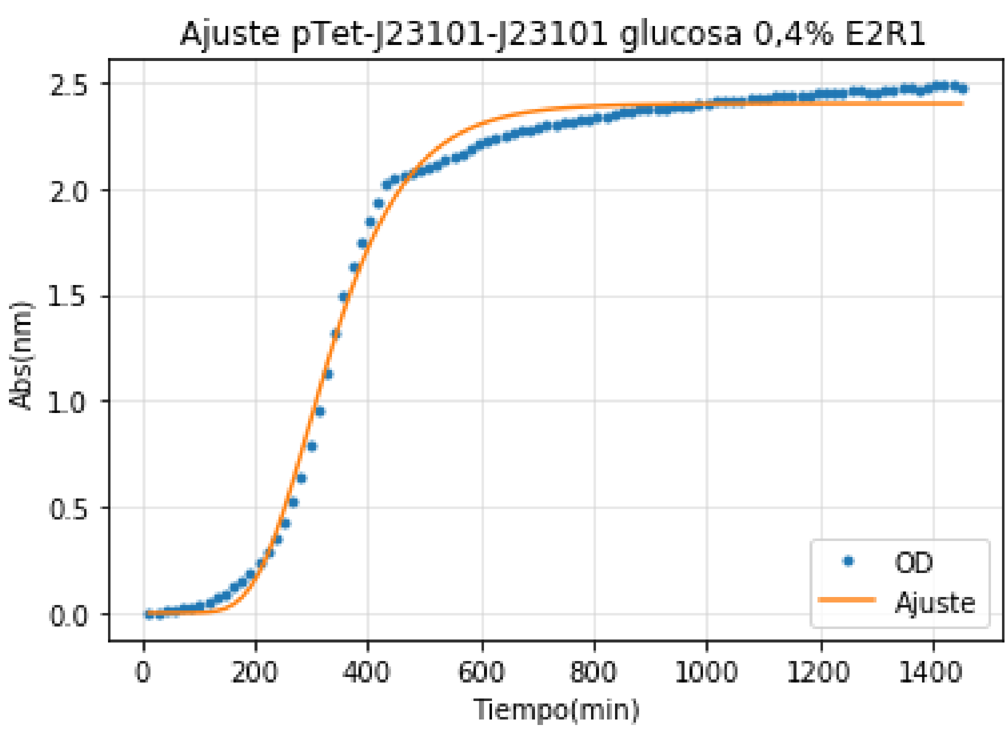
Out[14]: <matplotlib.legend.Legend at 0x267eb23bcf8>




In [15]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-J23101-std glucosa rep 1
y7= np.log(od15211)-np.log(np.min(od15211))
print('Min OD = %e'%((np.min(od15211))))
evaly, params=Function_fit(tt,y7,0,-1,title = 'Ajuste pTet-J23101-J23101 glucosa 0,4% E
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)
```

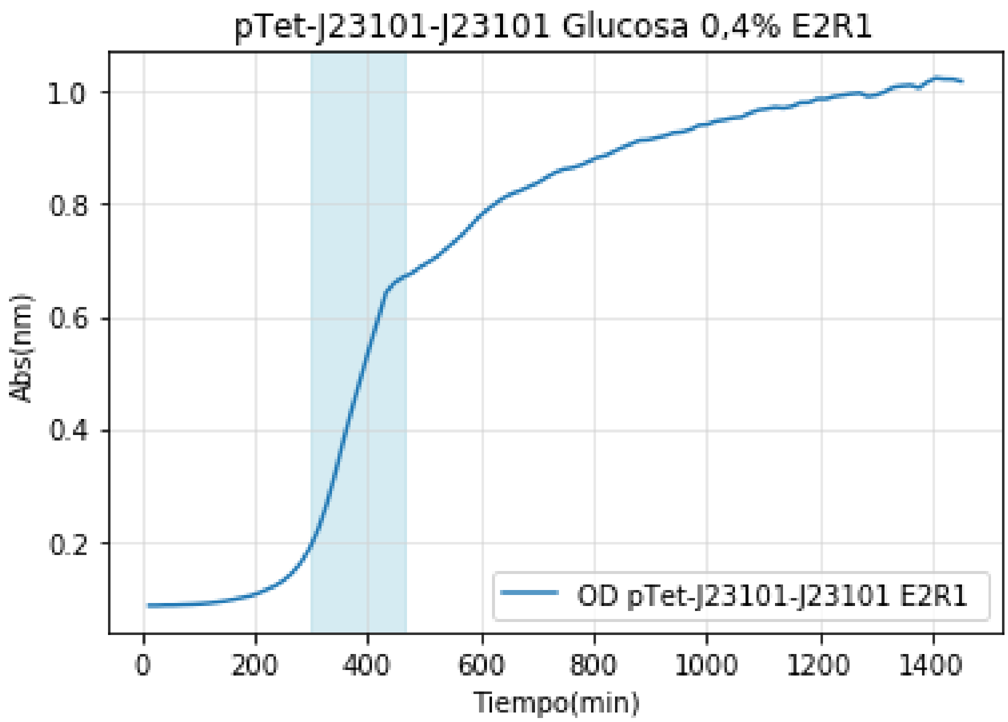
```
ttot7=tm7+extdp7
print('Tfinal=%e'%ttot7)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[19]
y2=tt[30]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15211,label='OD pTet-J23101-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[19:31],od15211[19:31],label='OD pTet-J23101-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

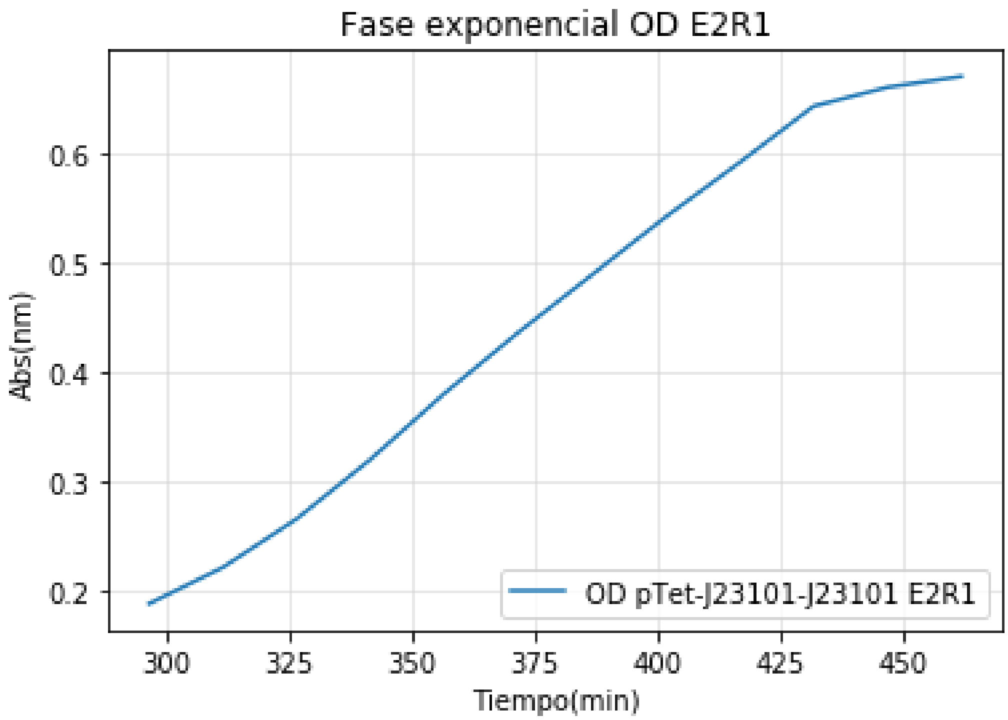
```
Min OD = 8.530000e-02
[  2.39988520e+00   9.27231338e-03   2.00377623e+02]
```



```
A=2.399885e+00
um=9.272313e-03
l=2.003776e+02
Tm=2.955932e+02
doubpe=7.475450e+01
ext=1.495090e+02
Tfinal=4.451022e+02
```

```
Out[15]: <matplotlib.legend.Legend at 0x267e9657668>
```

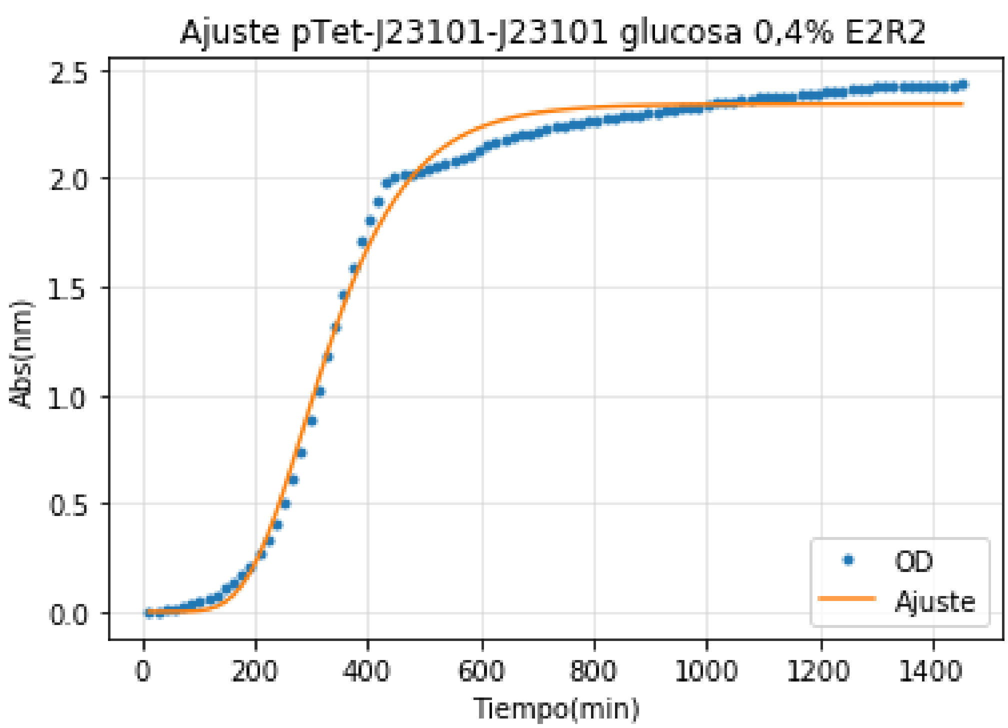





In [16]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 2
y8= np.log(od15212)-np.log(np.min(od15212))
print('Min OD = %e'%((np.min(od15212))))
evaly, params=Function_fit(tt,y8,0,-1,title = 'Ajuste pTet-J23101-J23101 glucosa 0,4% E
A8= params[0]
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[19]
```

```
y2=tt[31]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15212,label='OD pTet-J23101-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[19:32],od15212[19:32],label='OD pTet-J23101-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

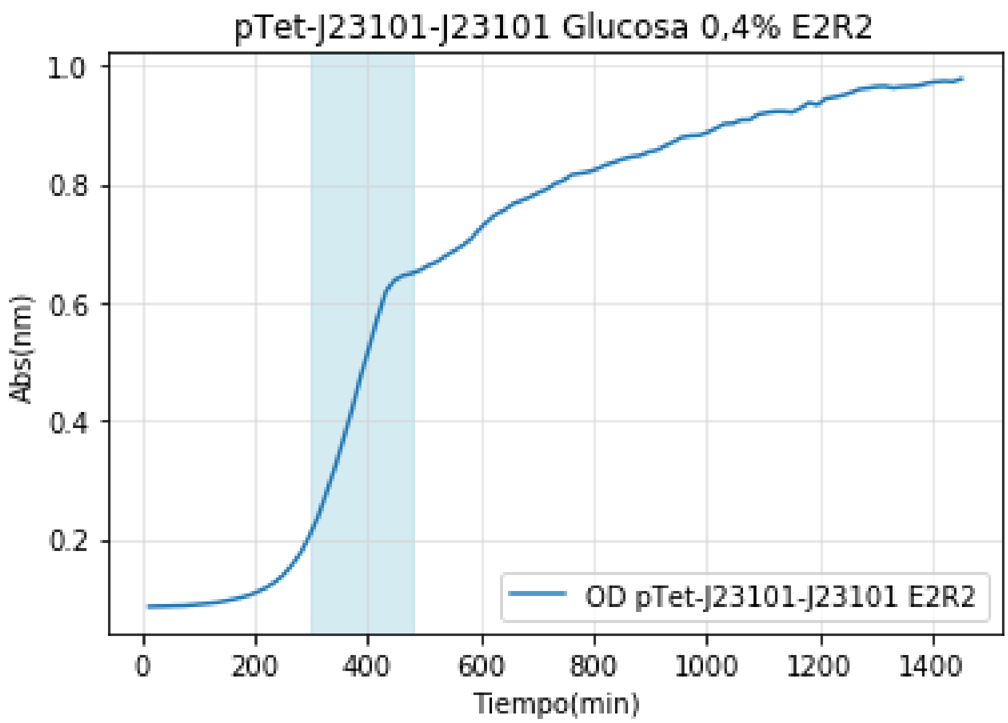
```
Min OD = 8.580000e-02
[  2.34140500e+00   8.43333849e-03   1.84992926e+02]
```

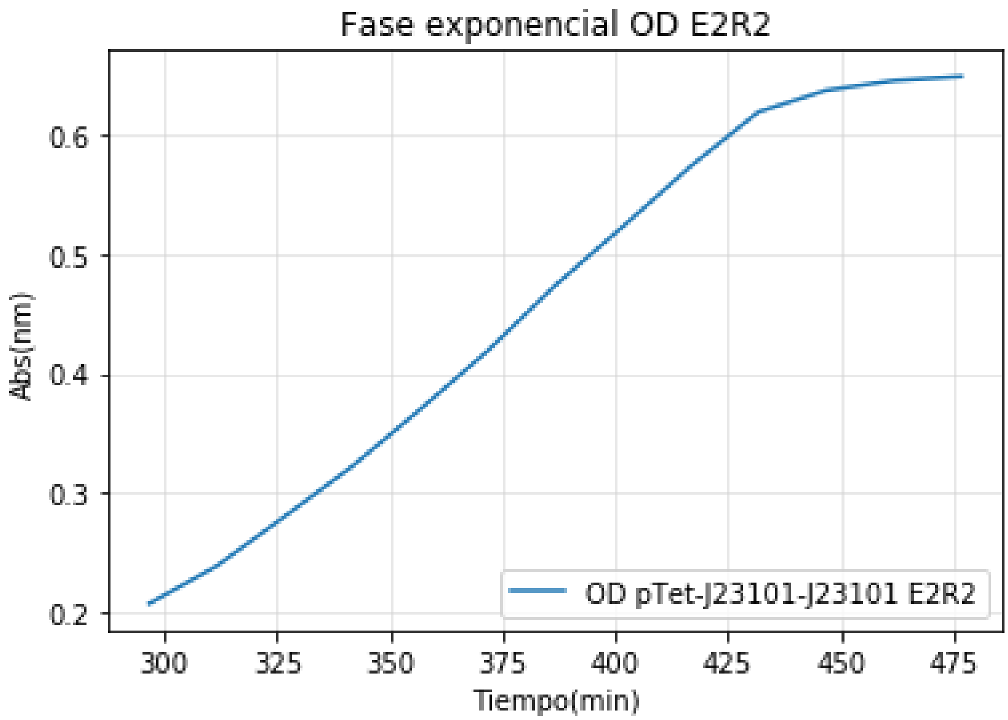



```
A=2.341405e+00
um=8.433338e-03
```

```
l=1.849929e+02
Tm=2.871298e+02
doubpe=8.219132e+01
ext=1.643826e+02
Tfinal=4.515124e+02
```

```
Out[16]: <matplotlib.legend.Legend at 0x267e9bc4080>
```

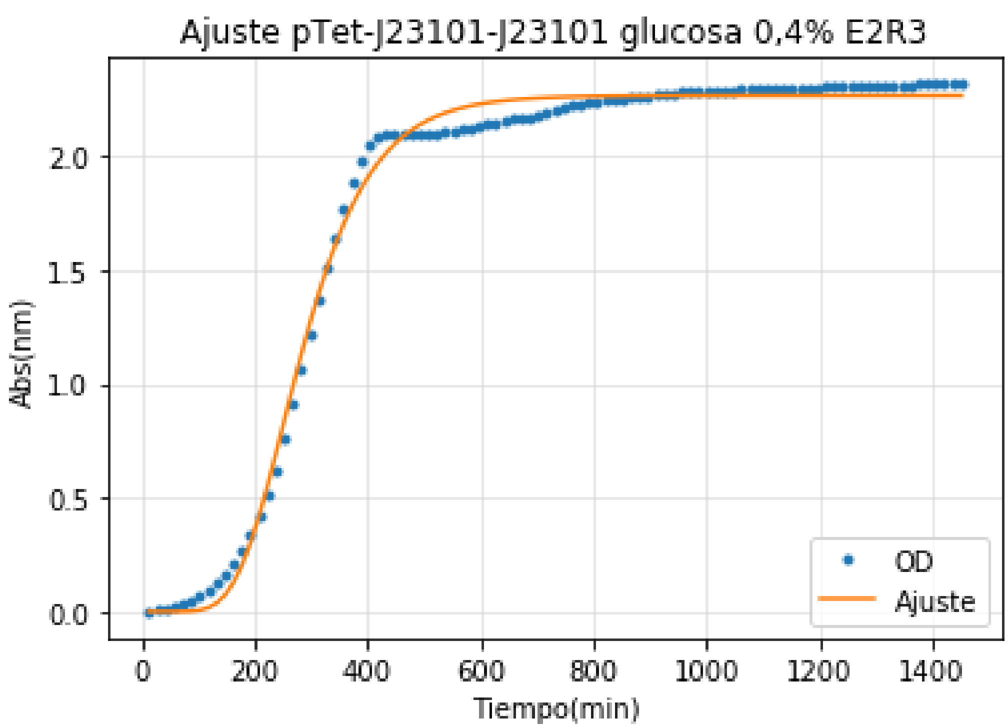





In [17]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 3
y9= np.log(od15213)-np.log(np.min(od15213))
print('Min OD = %e'%((np.min(od15213))))
evaly, params=Function_fit(tt,y9,0,-1,title = 'Ajuste pTet-J23101-J23101 glucosa 0,4% E
A9= params[0]
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[16]
```

```python
y2=tt[27]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15213,label='OD pTet-J23101-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[16:28],od15213[16:28],label='OD pTet-J23101-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

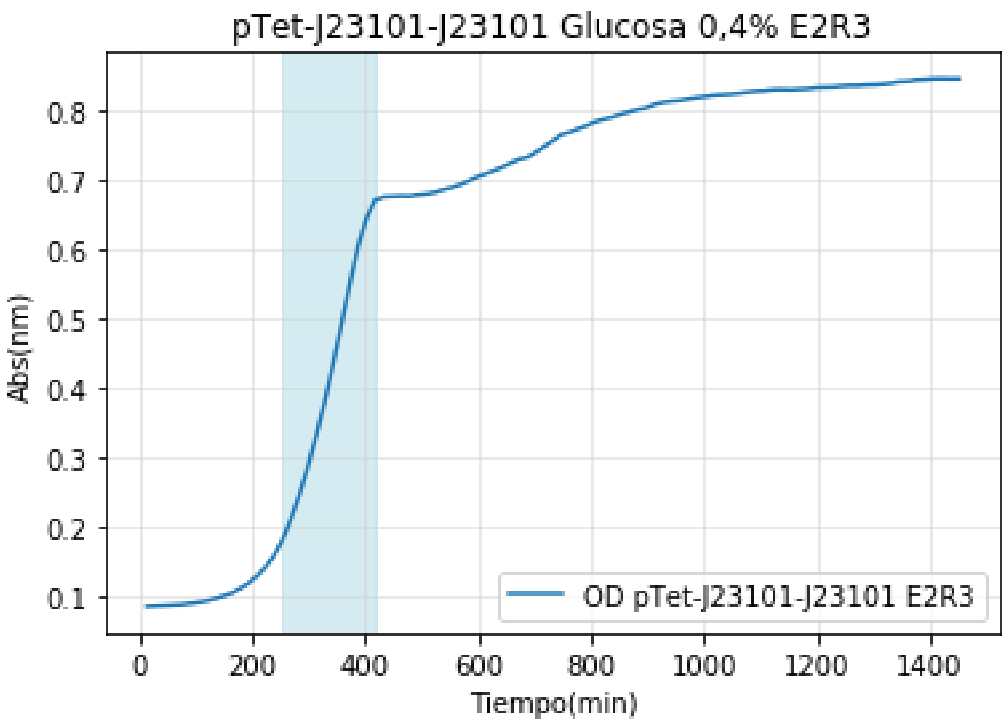
```
Min OD = 8.360000e-02
[  2.26457592e+00   9.83425423e-03   1.66237345e+02]
```

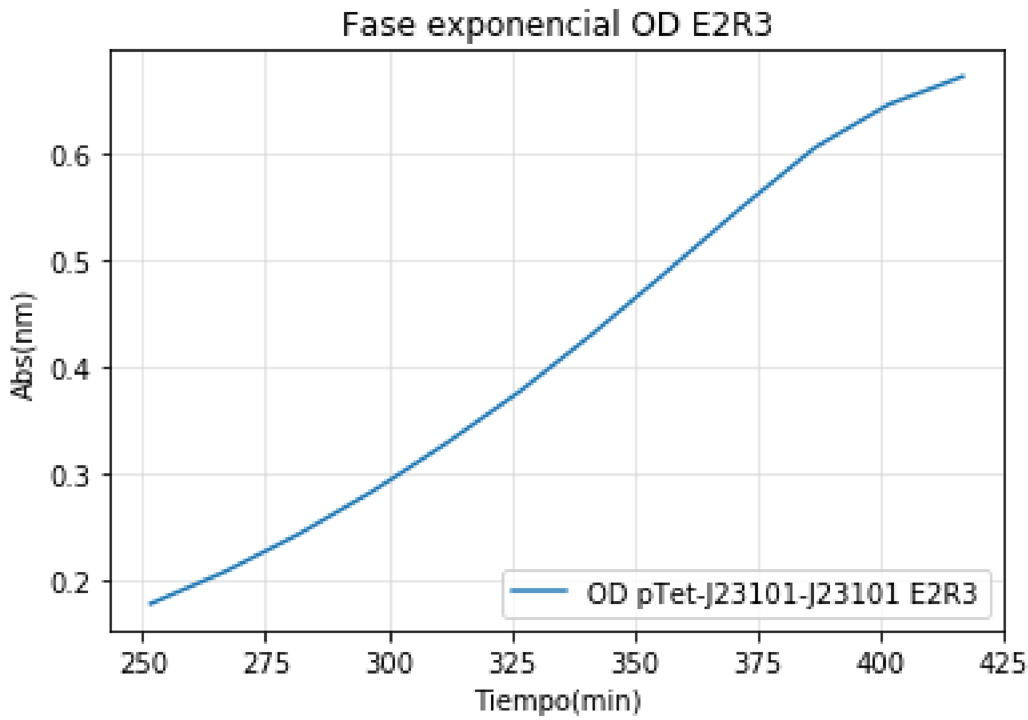



```
A=2.264576e+00
um=9.834254e-03
```

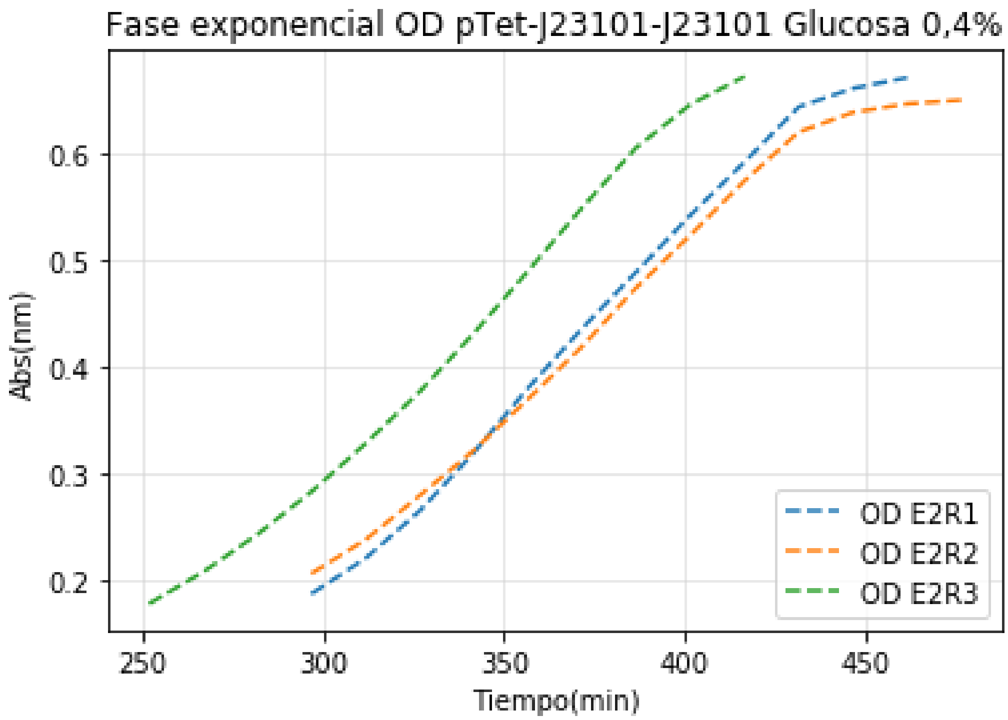
```
l=1.662373e+02
Tm=2.509505e+02
doubpe=7.048294e+01
ext=1.409659e+02
Tfinal=3.919164e+02
```

Out[17]: <matplotlib.legend.Legend at 0x267eb4690f0>





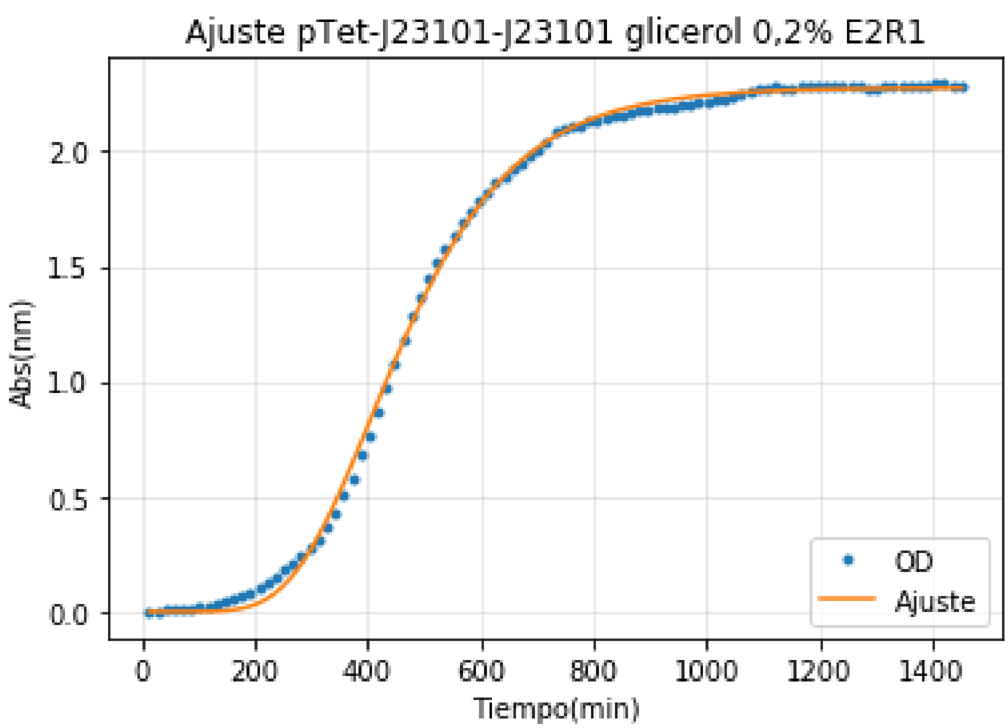

```
In [18]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:31],od15211[19:31],'--',label='OD E2R1')
         plt.plot(tt[19:32],od15212[19:32],'--',label='OD E2R2')
         plt.plot(tt[16:28],od15213[16:28],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[18]: <matplotlib.legend.Legend at 0x267eb58a710>
```




In [19]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 1
y10= np.log(od1521g1)-np.log(np.min(od1521g1))
print('Min OD = %e'%((np.min(od1521g1))))
evaly, params=Function_fit(tt,y10,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
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[27]
```

```python
y2=tt[47]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g1,label='OD pTet-J23101-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[27:48],od1521g1[27:48],label='OD pTet-J23101-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

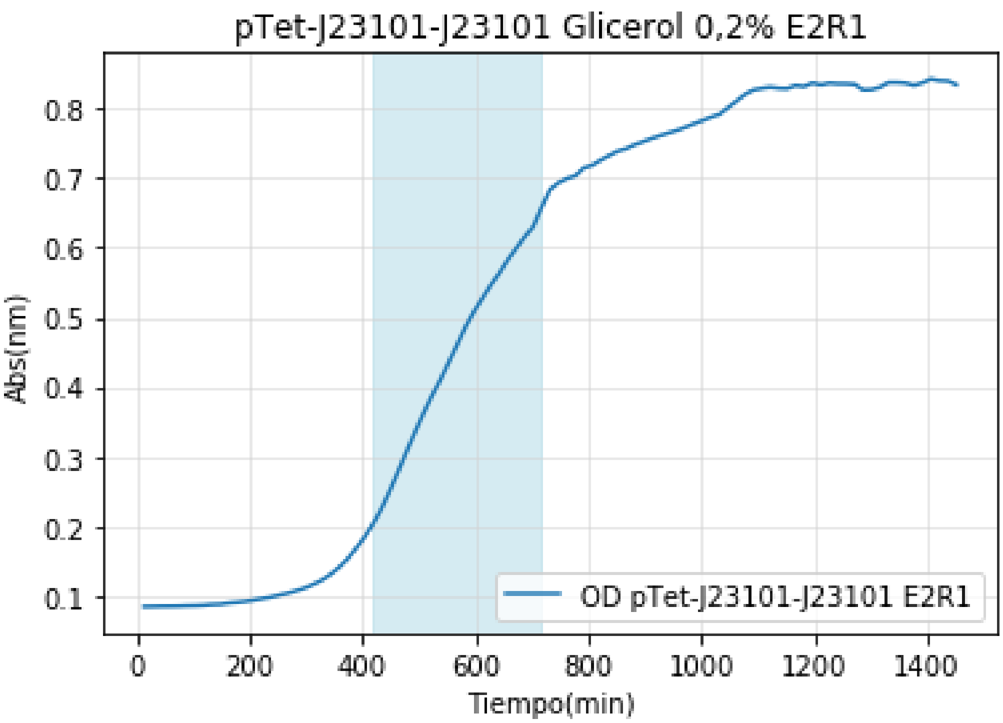
```
Min OD = 8.510000e-02
[  2.27552043e+00   5.96588621e-03   2.64362418e+02]
```

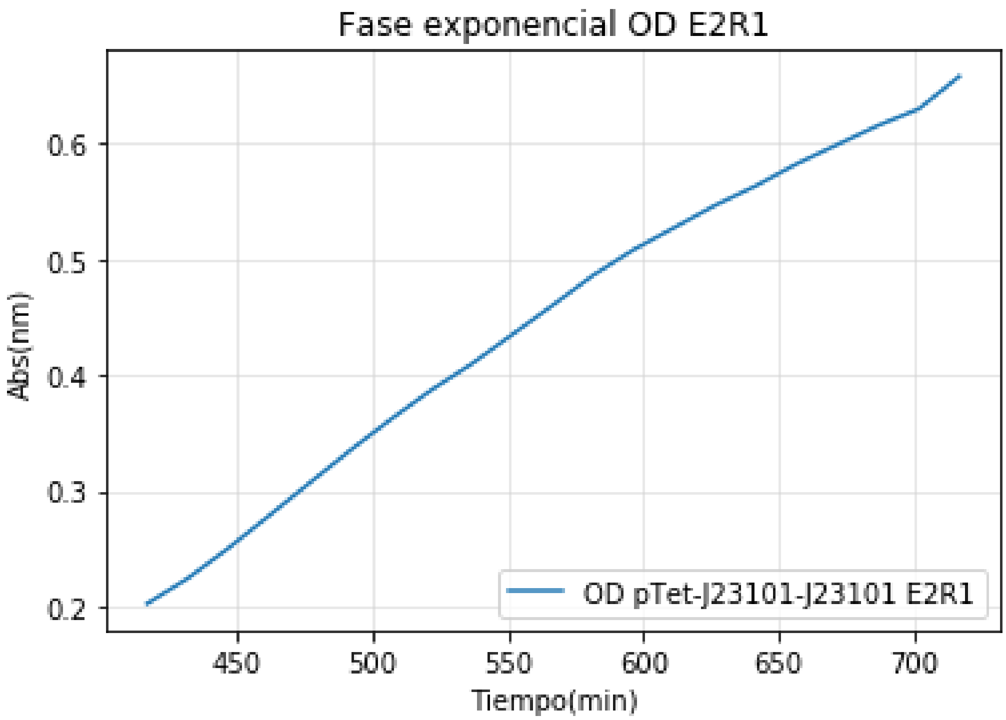



```
A=2.275520e+00
um=5.965886e-03
```

```
l=2.643624e+02
Tm=4.046797e+02
doubpe=1.161851e+02
ext=2.904628e+02
Tfinal=6.951425e+02
```

```
Out[19]: <matplotlib.legend.Legend at 0x267eb436358>
```

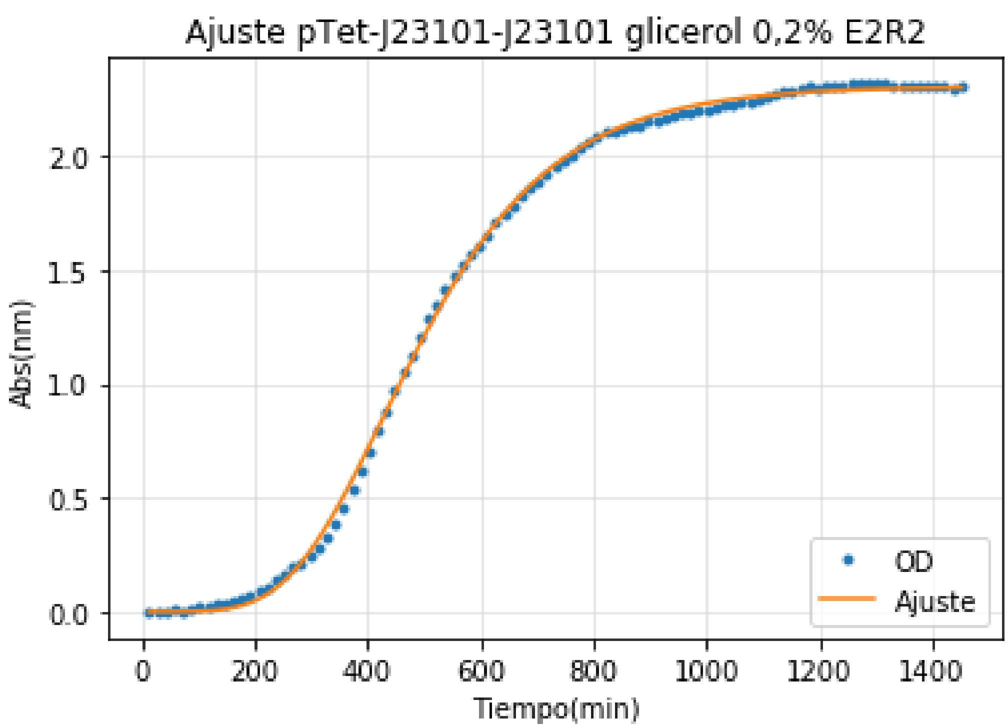





In [20]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 2
y11= np.log(od1521g2)-np.log(np.min(od1521g2))
print('Min OD = %e'%((np.min(od1521g2))))
evaly, params=Function_fit(tt,y11,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
A11= 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[28]
```

```
y2=tt[52]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g2,label='OD pTet-J23101-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[28:53],od1521g2[28:53],label='OD pTet-J23101-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

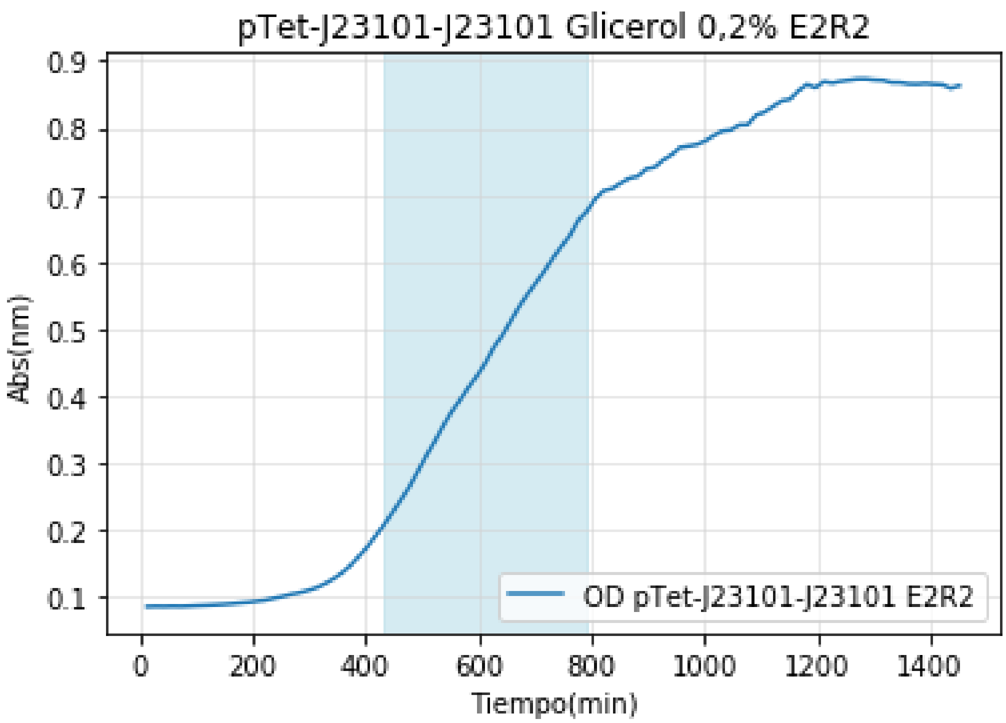
```
Min OD = 8.600000e-02
[  2.30705217e+00   5.07308656e-03   2.58634310e+02]
```

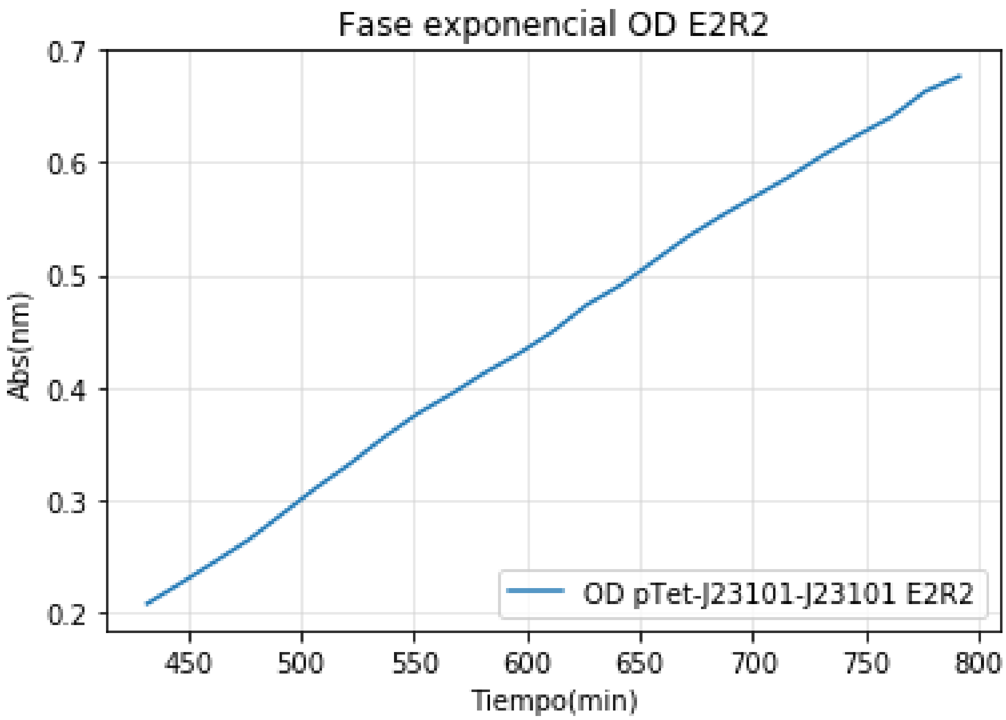



```
A=2.307052e+00
um=5.073087e-03
```

```
l=2.586343e+02
Tm=4.259323e+02
doubpe=1.366322e+02
ext=3.415806e+02
Tfinal=7.675129e+02
```

```
Out[20]: <matplotlib.legend.Legend at 0x267eb14b320>
```

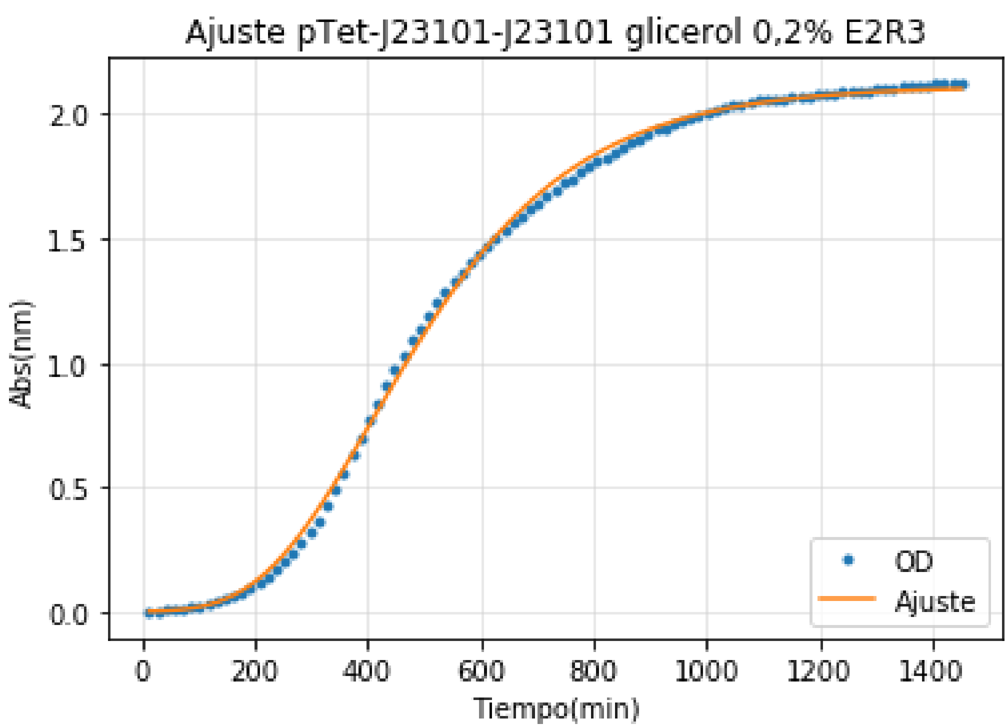





In [21]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 3
y12= np.log(od1521g3)-np.log(np.min(od1521g3))
print('Min OD = %e'%((np.min(od1521g3))))
evaly, params=Function_fit(tt,y12,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
A12= 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*t212
print('ext=%e'%extdp12)
ttot12=tm12+extdp12
print('Tfinal=%e'%ttot12)

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

```python
y2=tt[52]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g3,label='OD pTet-J23101-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[27:53],od1521g3[27:53],label='OD pTet-J23101-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

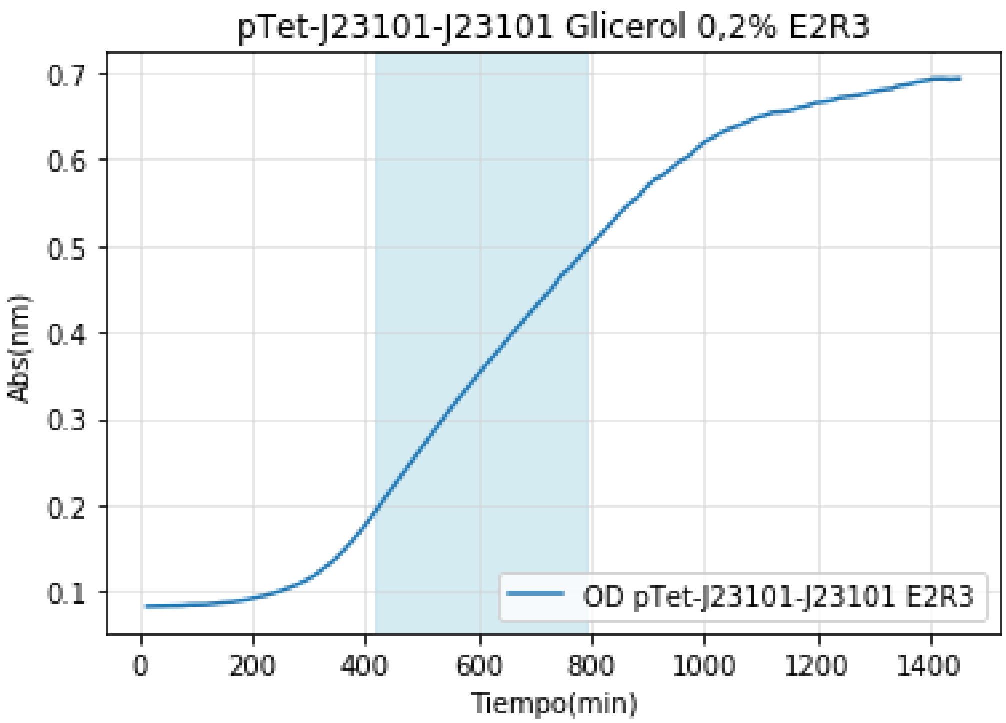
```
Min OD = 8.320000e-02
[  2.11194516e+00   3.89343431e-03   2.09853636e+02]
```

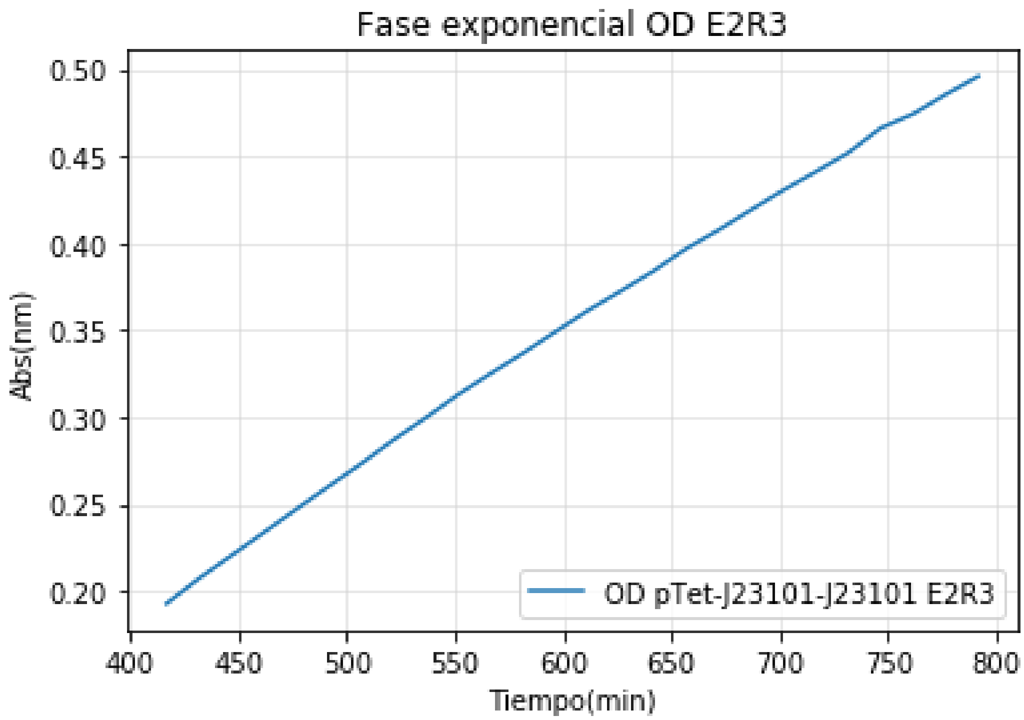



```
A=2.111945e+00
um=3.893434e-03
```

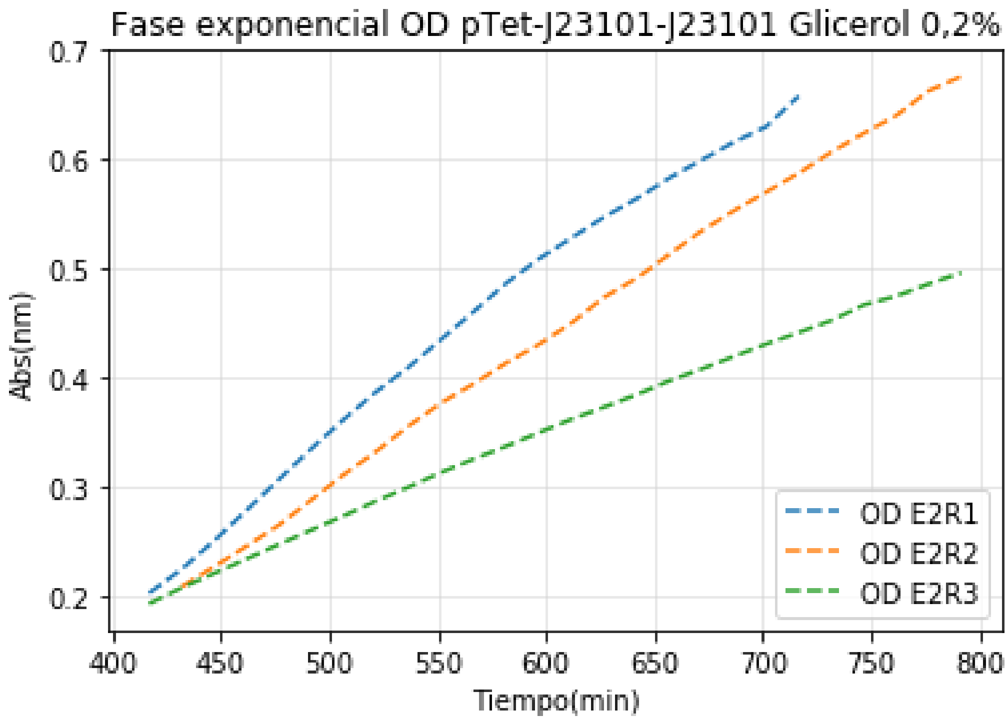
```
l=2.098536e+02
Tm=4.094053e+02
doubpe=1.780298e+02
ext=3.560595e+02
Tfinal=7.654648e+02
```

Out[21]: <matplotlib.legend.Legend at 0x267eb6395f8>

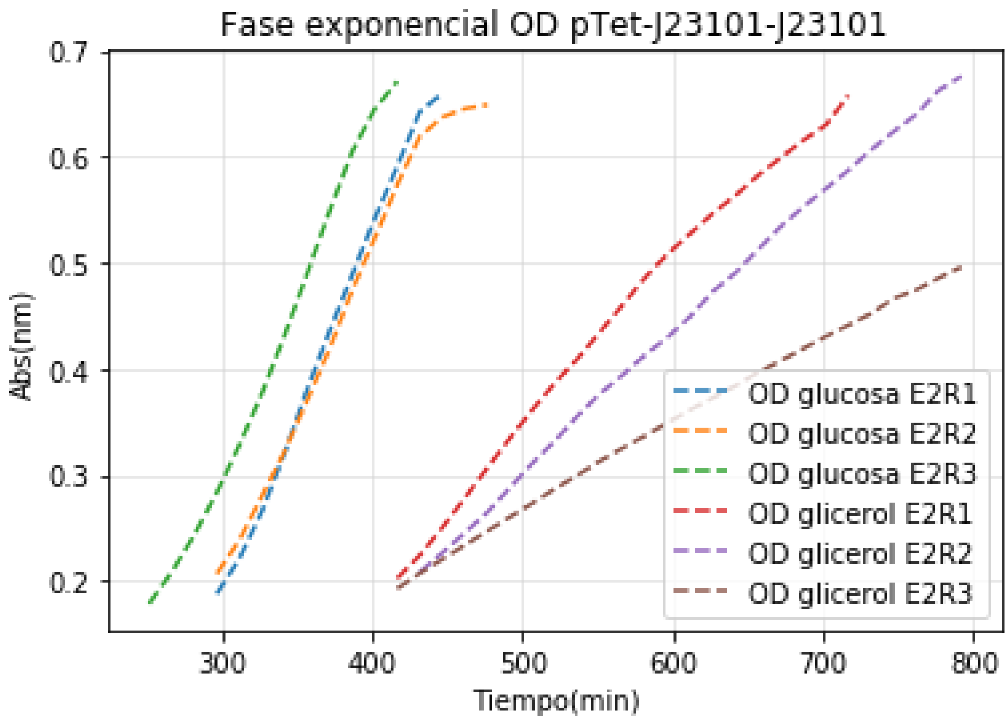





```
In [22]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[27:48],od1521g1[27:48],'--',label='OD E2R1')
         plt.plot(tt[28:53],od1521g2[28:53],'--',label='OD E2R2')
         plt.plot(tt[27:53],od1521g3[27:53],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[22]: <matplotlib.legend.Legend at 0x267eb789ef0>
```



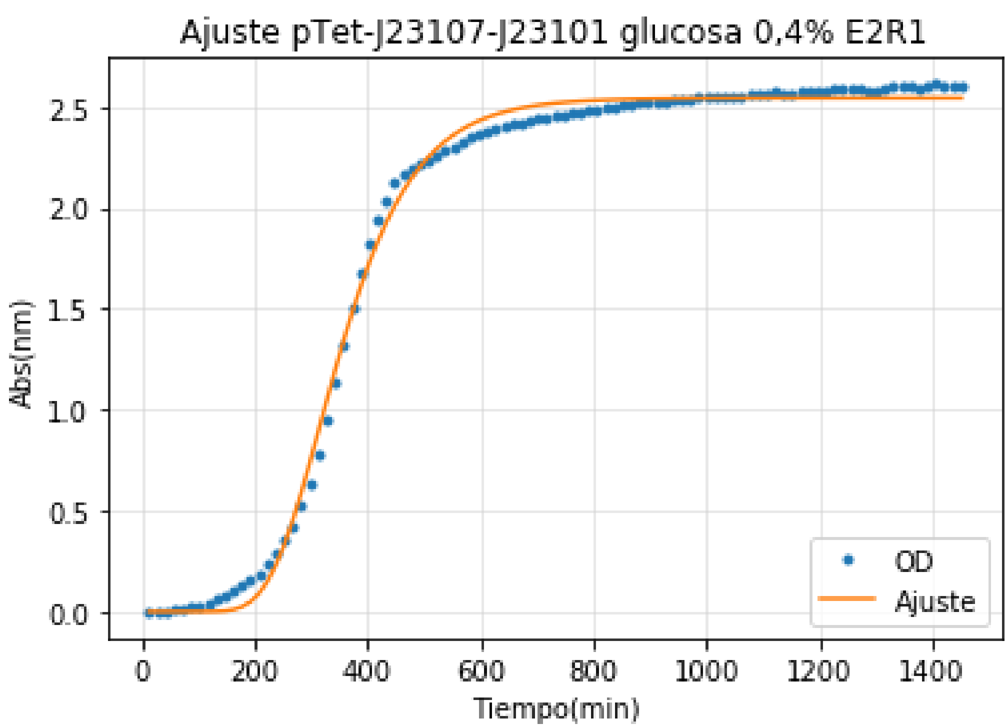

```
In [23]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:30],od15211[19:30],'--',label='OD glucosa E2R1')
         plt.plot(tt[19:32],od15212[19:32],'--',label='OD glucosa E2R2')
         plt.plot(tt[16:28],od15213[16:28],'--',label='OD glucosa E2R3')
         plt.plot(tt[27:48],od1521g1[27:48],'--',label='OD glicerol E2R1')
         plt.plot(tt[28:53],od1521g2[28:53],'--',label='OD glicerol E2R2')
         plt.plot(tt[27:53],od1521g3[27:53],'--',label='OD glicerol E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[23]: <matplotlib.legend.Legend at 0x267eb85acc0>
```




In [24]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-107-std glucosa rep 1
y13= np.log(od15231)-np.log(np.min(od15231))
print('Min OD = %e'%((np.min(od15231))))
evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste pTet-J23107-J23101 glucosa 0,4%
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[21]
```

```
y2=tt[31]
plt.figure()
plt.title('pTet-J23107-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15231,label='OD pTet-J23107-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 E23R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[21:32],od15231[21:32],label='OD pTet-J23107-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

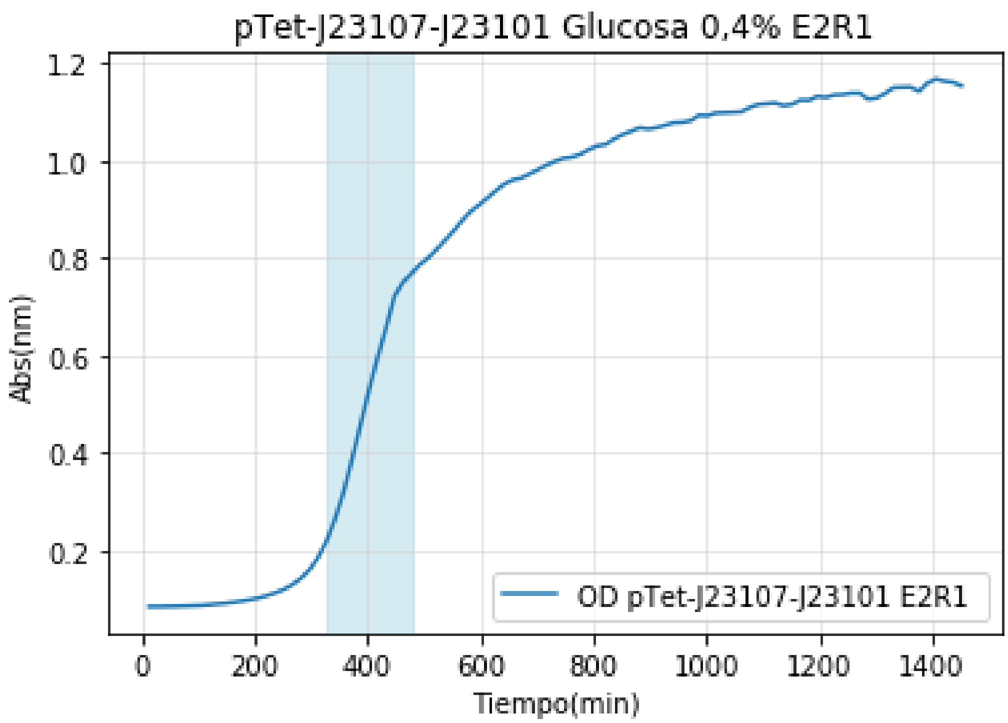
```
Min OD = 8.570000e-02
[  2.54342379e+00   1.03117507e-02   2.25391606e+02]
```

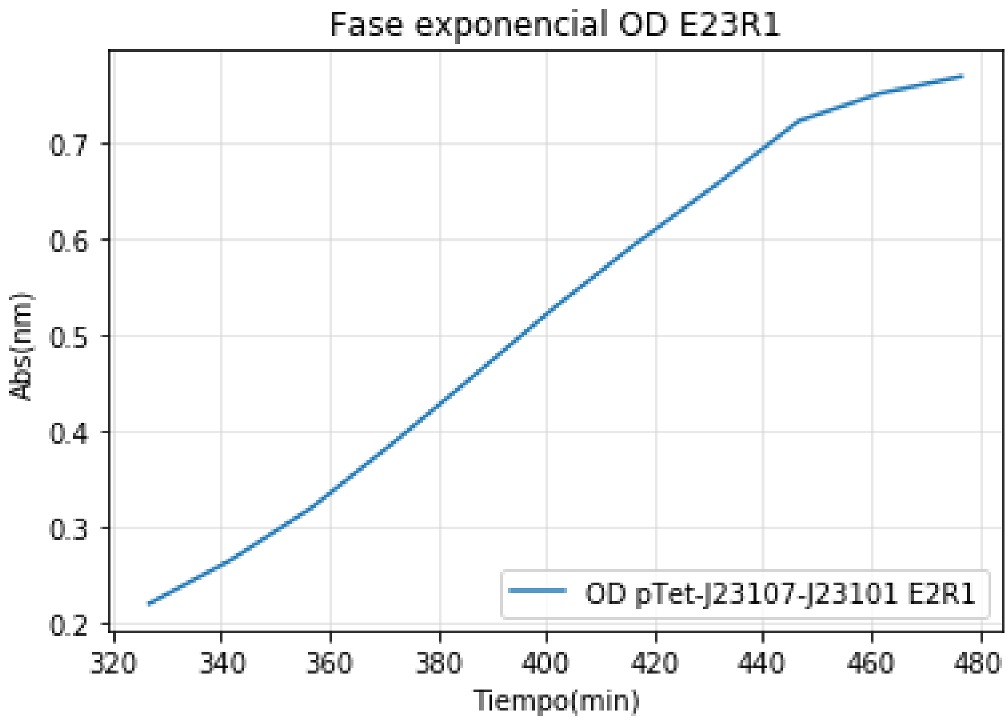



```
A=2.543424e+00
um=1.031175e-02
```

```
l=2.253916e+02
Tm=3.161302e+02
doubpe=6.721916e+01
ext=1.344383e+02
Tfinal=4.505685e+02
```

Out[24]: <matplotlib.legend.Legend at 0x267eb076048>

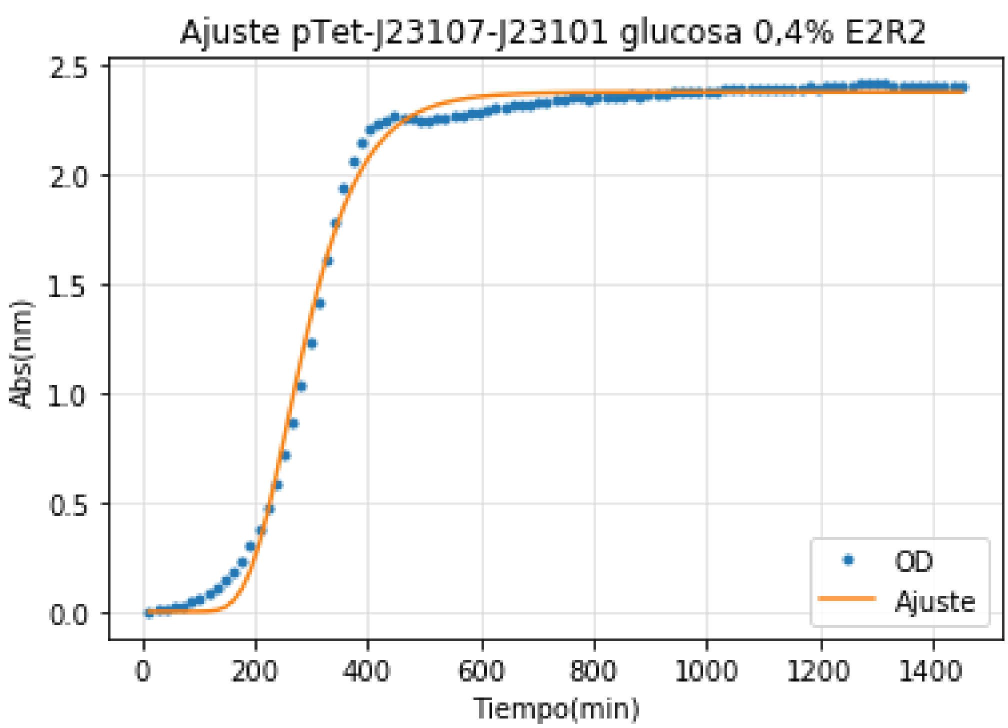





In [25]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-107-std glucosa rep 2
y14= np.log(od15232)-np.log(np.min(od15232))
print('Min OD = %e'%((np.min(od15232))))
evaly, params=Function_fit(tt,y14,0,-1,title = 'Ajuste pTet-J23107-J23101 glucosa 0,4%
A14= params[0]
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[17]
```

```
y2=tt[26]
plt.figure()
plt.title('pTet-J23107-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15232,label='OD pTet-J23107-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[17:27],od15232[17:27],label='OD pTet-J23107-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

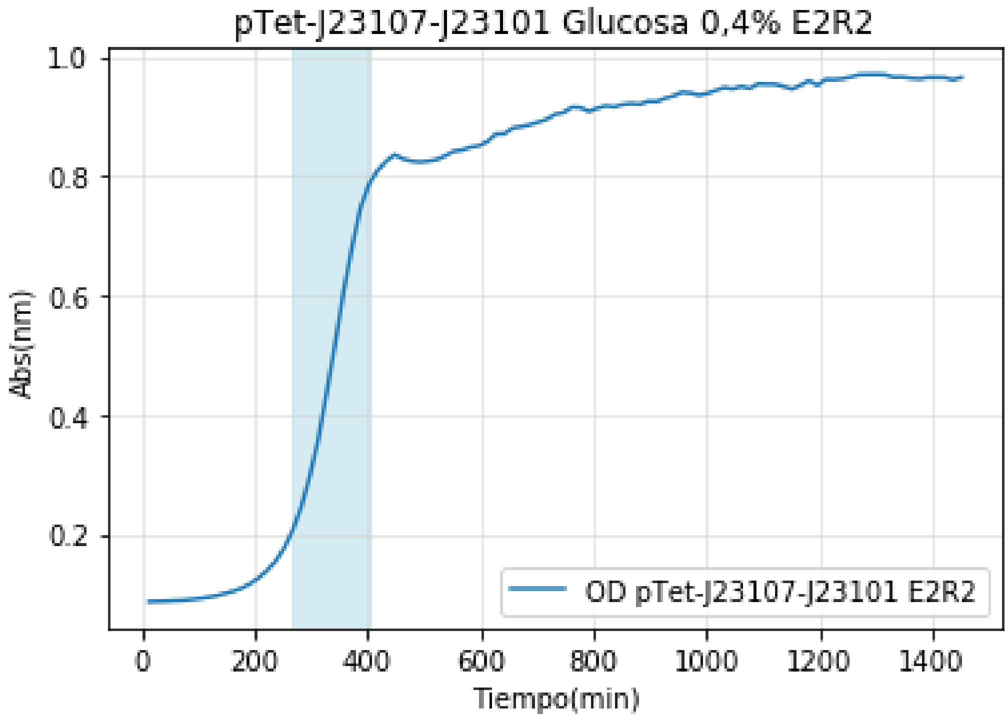
```
Min OD = 8.650000e-02
[  2.38049312e+00   1.22442727e-02   1.86353375e+02]
```

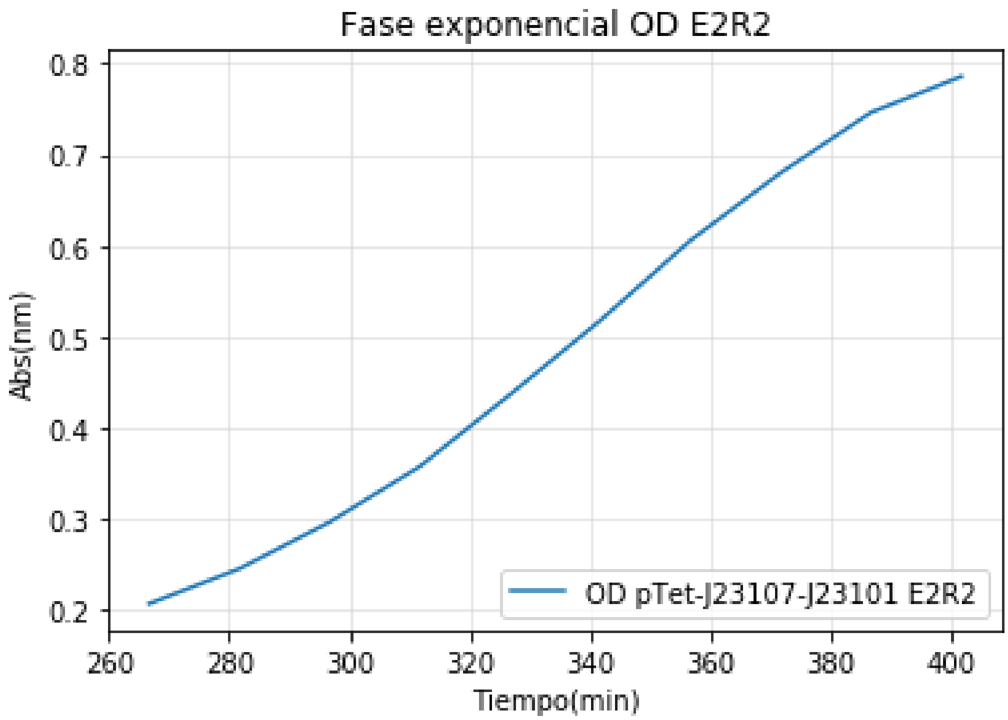



```
A=2.380493e+00
um=1.224427e-02
```

```
l=1.863534e+02
Tm=2.578753e+02
doubpe=5.660991e+01
ext=1.132198e+02
Tfinal=3.710952e+02
```

```
Out[25]: <matplotlib.legend.Legend at 0x267eb678a20>
```

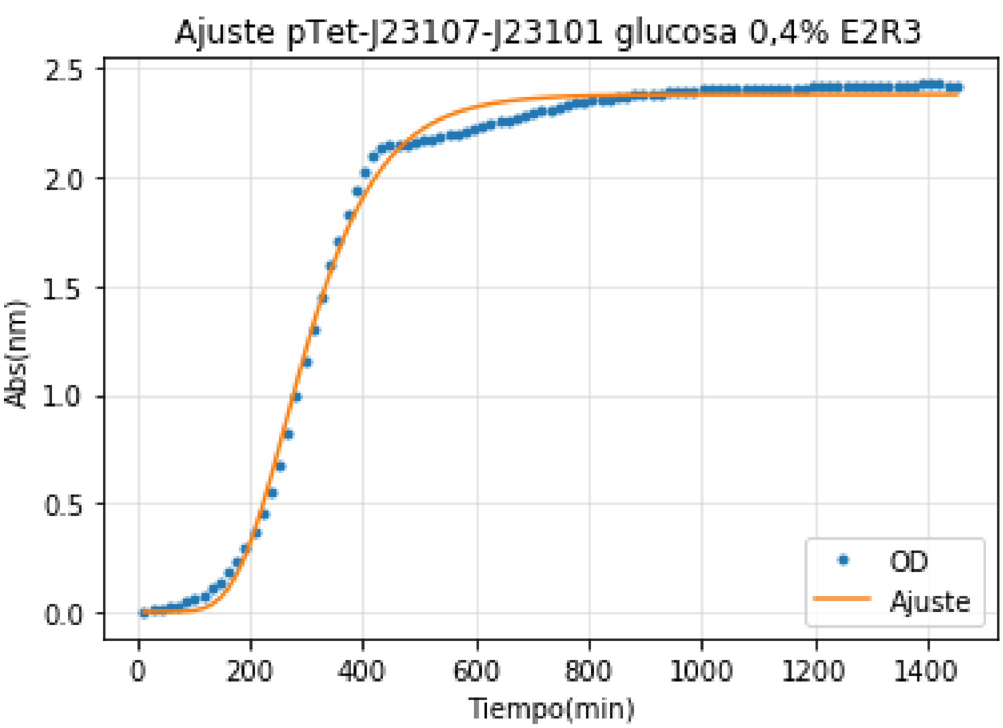





In [26]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-107-std glucosa rep 3
y15= np.log(od15233)-np.log(np.min(od15233))
print('Min OD = %e'%((np.min(od15233))))
evaly, params=Function_fit(tt,y15,0,-1,title = 'Ajuste pTet-J23107-J23101 glucosa 0,4%
A15= params[0]
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('pTet-J23107-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15233,label='OD pTet-J23107-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[17:29],od15233[17:29],label='OD pTet-J23107-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

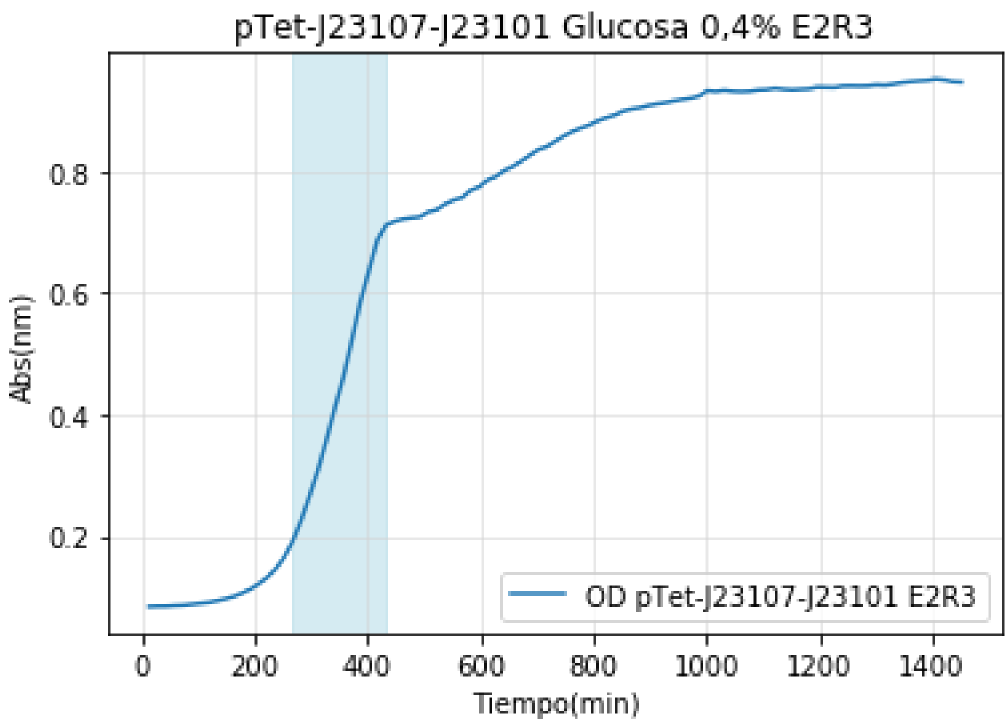
```
Min OD = 8.400000e-02
[  2.38113195e+00   9.65546943e-03   1.72796102e+02]
```

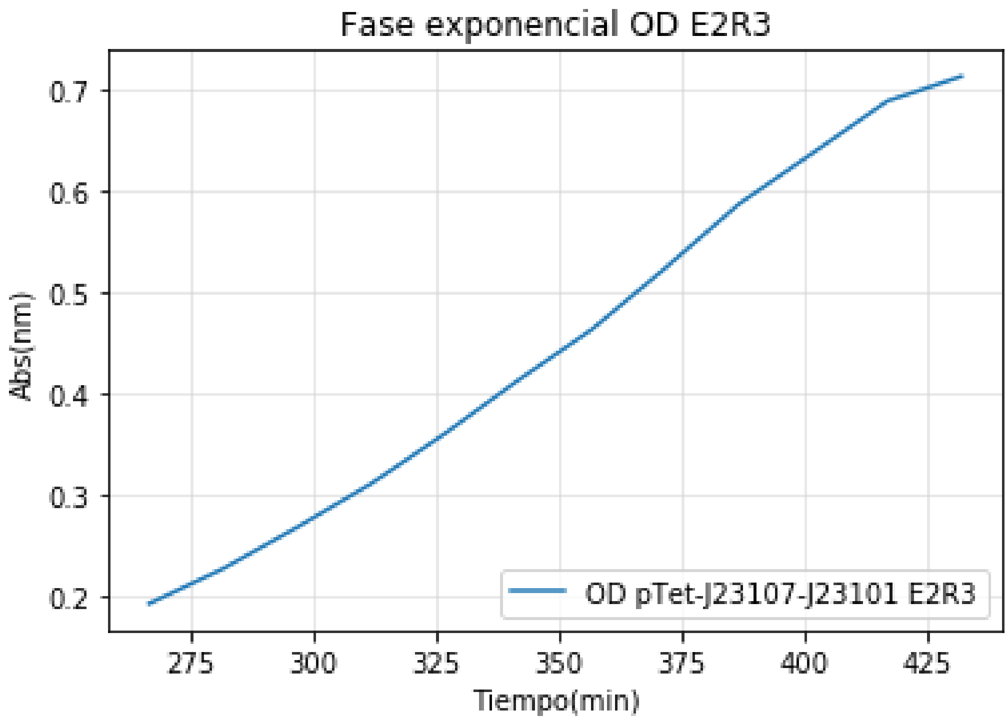



```
A=2.381132e+00
um=9.655469e-03
```

```
l=1.727961e+02
Tm=2.635187e+02
doubpe=7.178804e+01
ext=1.435761e+02
Tfinal=4.070948e+02
```

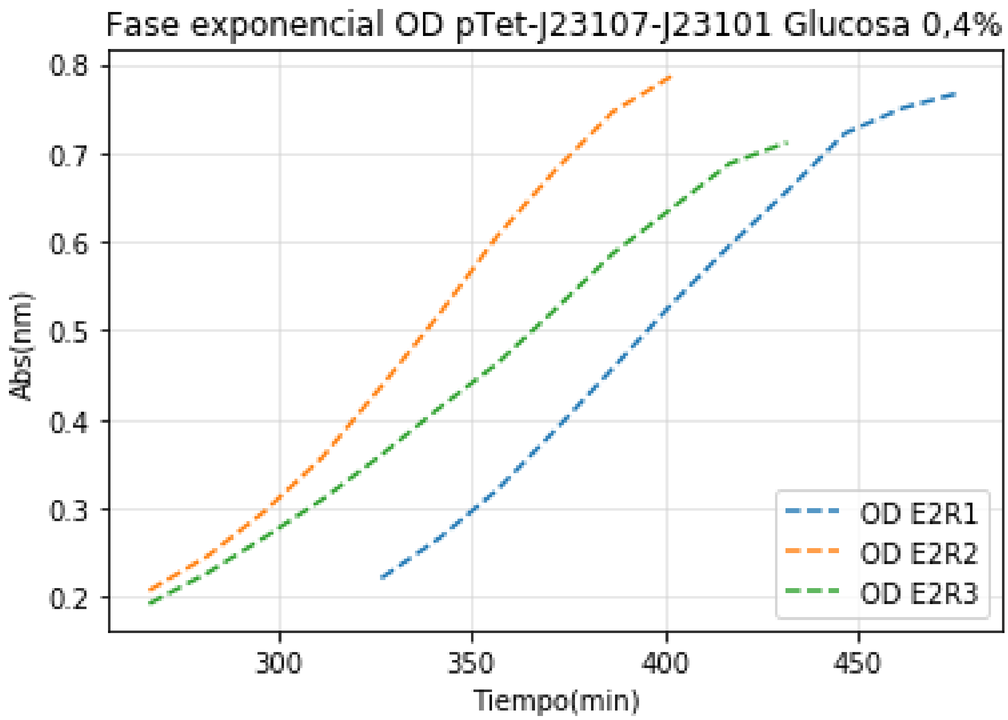
```
Out[26]: <matplotlib.legend.Legend at 0x267eca63e48>
```






In [27]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pTet-J23107-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[21:32],od15231[21:32],'--',label='OD E2R1')
plt.plot(tt[17:27],od15232[17:27],'--',label='OD E2R2')
plt.plot(tt[17:29],od15233[17:29],'--',label='OD E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

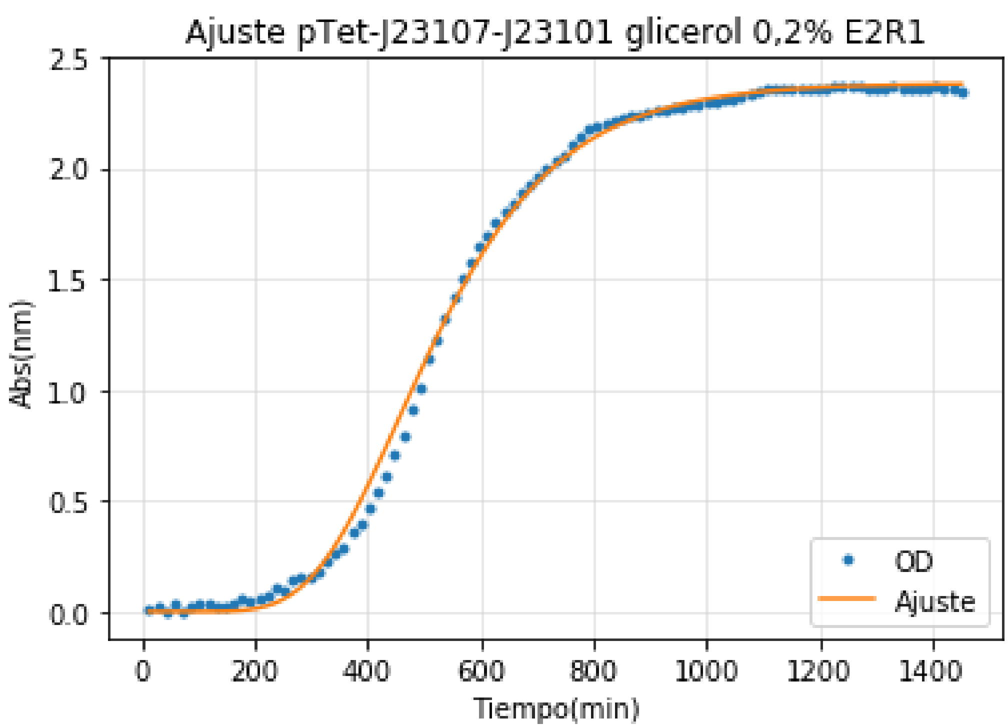
Out[27]:
```
<matplotlib.legend.Legend at 0x267ecb7d7b8>
```




```
In [28]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #ptet-107-std glicerol rep 1
         y16= np.log(od1523g1)-np.log(np.min(od1523g1))
         print('Min OD = %e'%((np.min(od1523g1))))
         evaly, params=Function_fit(tt,y16,0,-1,title = 'Ajuste pTet-J23107-J23101 glicerol 0,2%
         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[30]
```

```python
y2=tt[52]
plt.figure()
plt.title('pTet-J23107-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1523g1,label='OD pTet-J23107-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[30:53],od1523g1[30:53],label='OD pTet-J23107-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

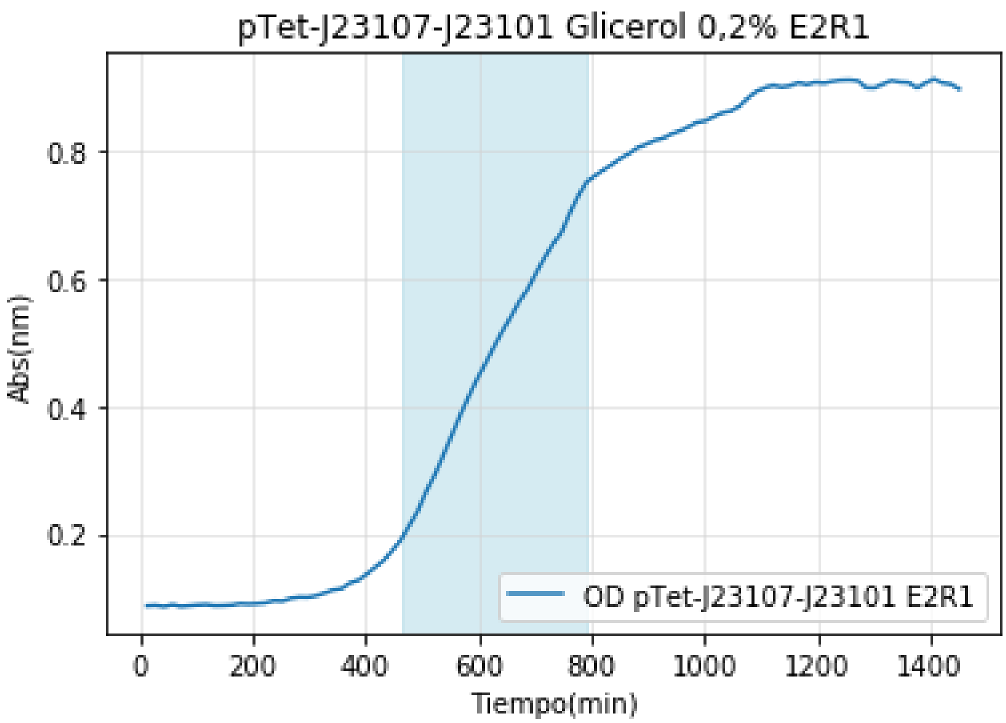
```
Min OD = 8.580000e-02
[  2.37993854e+00   5.62626737e-03   3.00000000e+02]
```

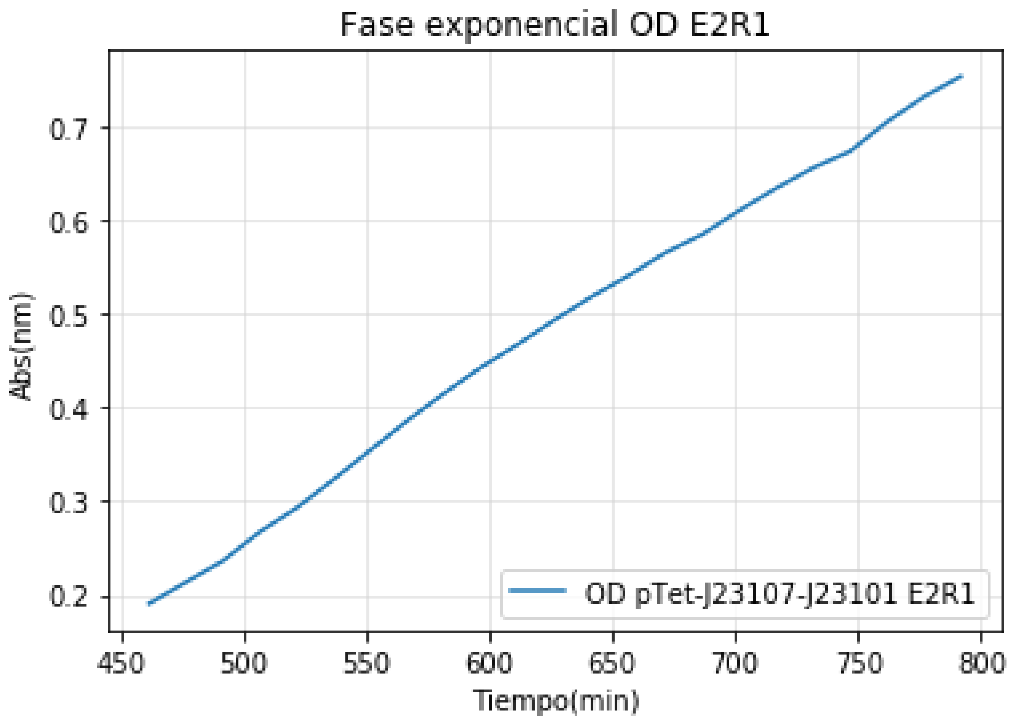



```
A=2.379939e+00
um=5.626267e-03
```

```
l=3.000000e+02
Tm=4.556148e+02
doubpe=1.231984e+02
ext=3.079960e+02
Tfinal=7.636108e+02
```

Out[28]: `<matplotlib.legend.Legend at 0x267ecac6b38>`

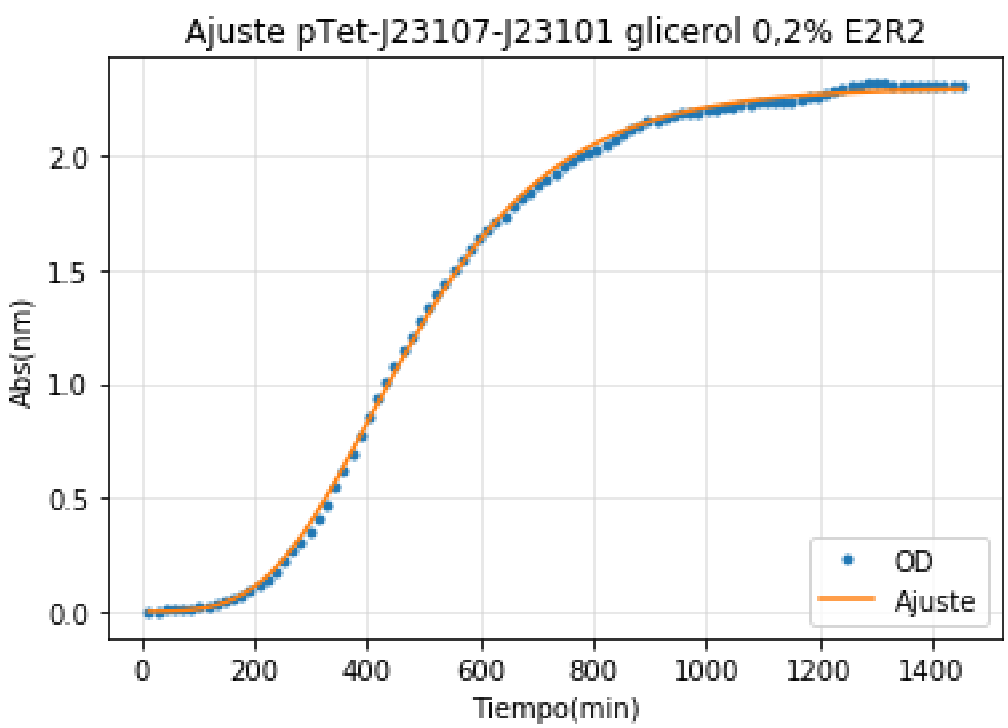





In [29]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-107-std glicerol rep 2
y17= np.log(od1523g2)-np.log(np.min(od1523g2))
print('Min OD = %e'%((np.min(od1523g2))))
evaly, params=Function_fit(tt,y17,0,-1,title = 'Ajuste pTet-J23107-J23101 glicerol 0,2%
A17= 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[27]
```

```
y2=tt[53]
plt.figure()
plt.title('pTet-J23107-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1523g2,label='OD pTet-J23107-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[27:54],od1523g2[27:54],label='OD pTet-J23107-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

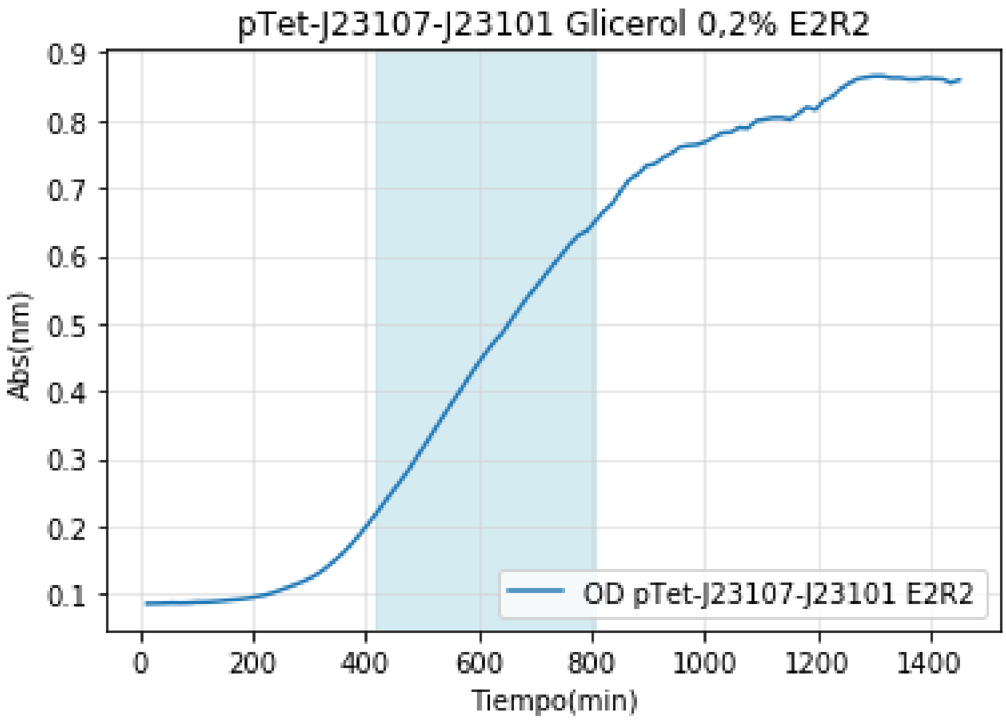
```
Min OD = 8.550000e-02
[  2.29805766e+00   4.61718516e-03   2.20040295e+02]
```

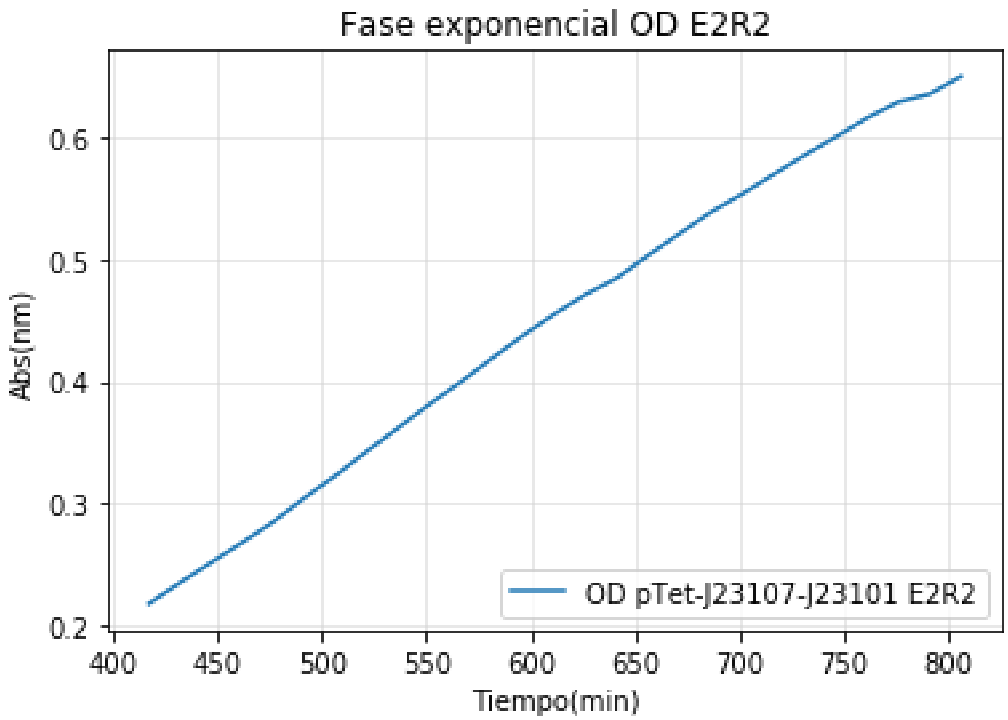



```
A=2.298058e+00
um=4.617185e-03
```

```
l=2.200403e+02
Tm=4.031406e+02
doubpe=1.501233e+02
ext=3.753083e+02
Tfinal=7.784489e+02
```

Out[29]: `<matplotlib.legend.Legend at 0x267eb65dbe0>`

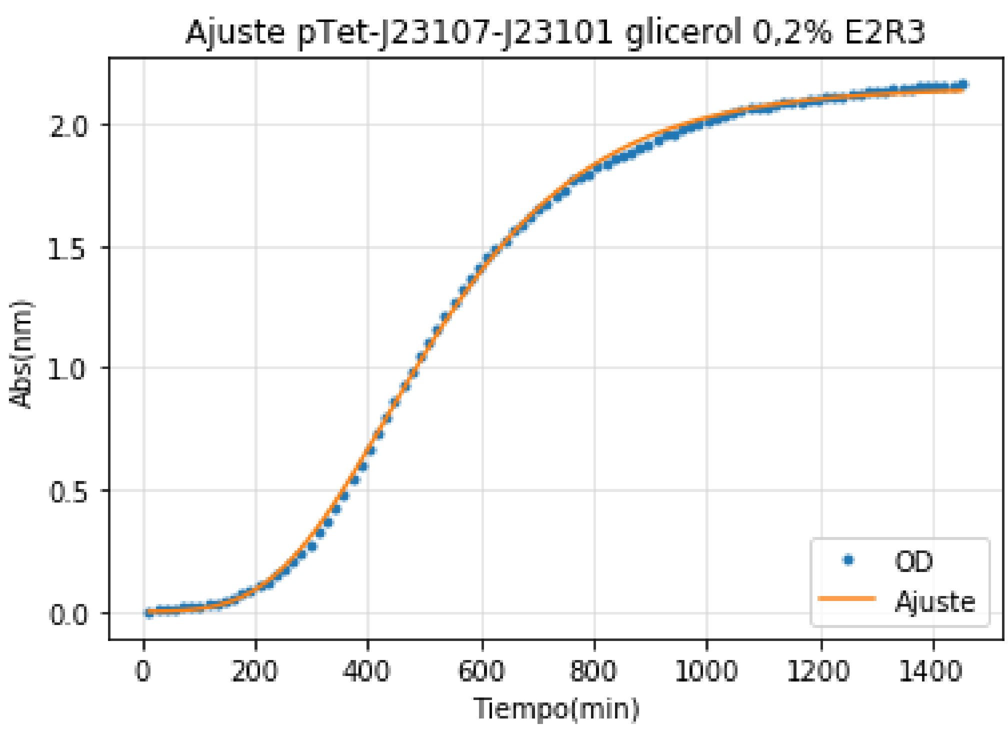





In [30]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-107-std glicerol rep 3
y18= np.log(od1523g3)-np.log(np.min(od1523g3))
print('Min OD = %e'%((np.min(od1523g3))))
evaly, params=Function_fit(tt,y18,0,-1,title = 'Ajuste pTet-J23107-J23101 glicerol 0,2%
A18= 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[28]
```

```
y2=tt[59]
plt.figure()
plt.title('pTet-J23107-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1523g3,label='OD pTet-J23107-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 E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:60],od1523g3[28:60],label='OD pTet-J23107-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

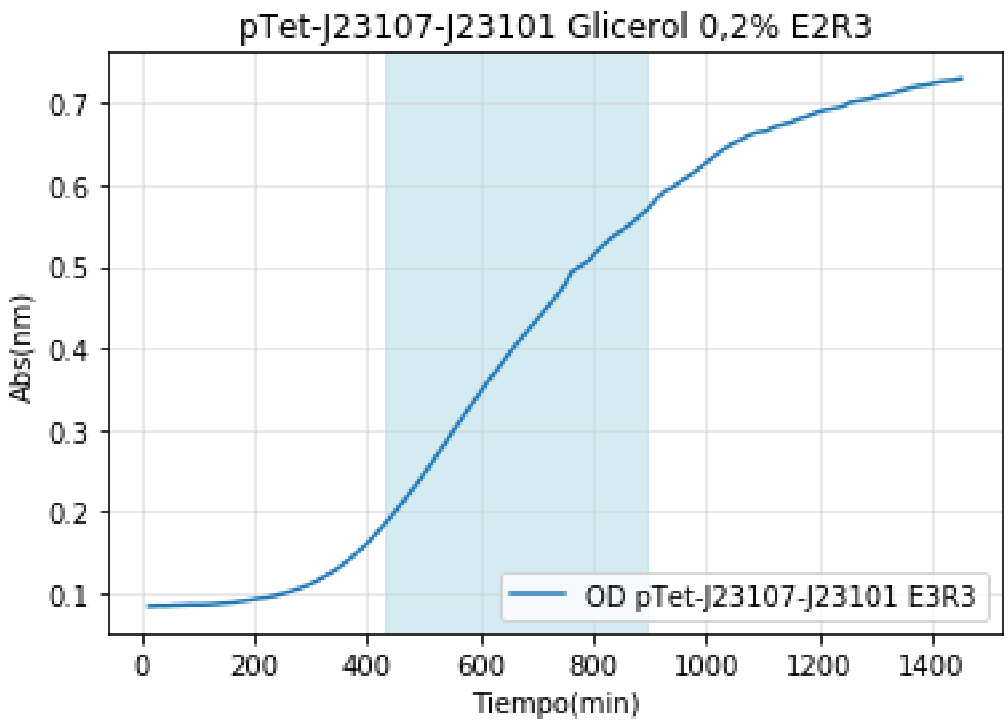
```
Min OD = 8.400000e-02
[  2.14788549e+00   3.93566781e-03   2.30548233e+02]
```

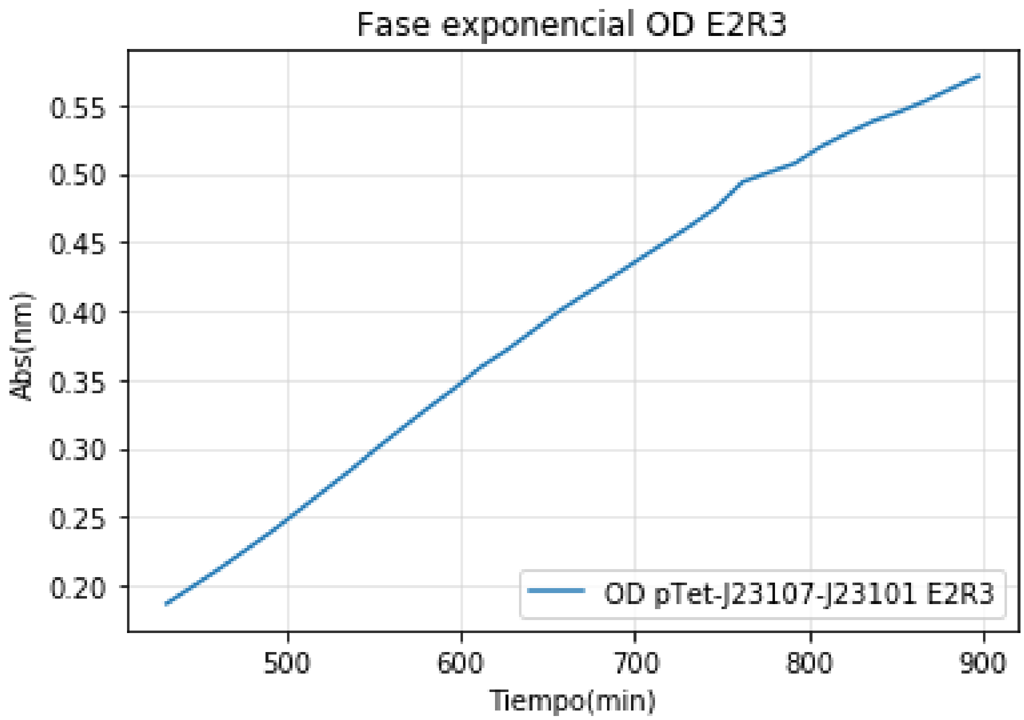



```
A=2.147885e+00
um=3.935668e-03
```

```
l=2.305482e+02
Tm=4.313180e+02
doubpe=1.761193e+02
ext=4.402983e+02
Tfinal=8.716163e+02
```

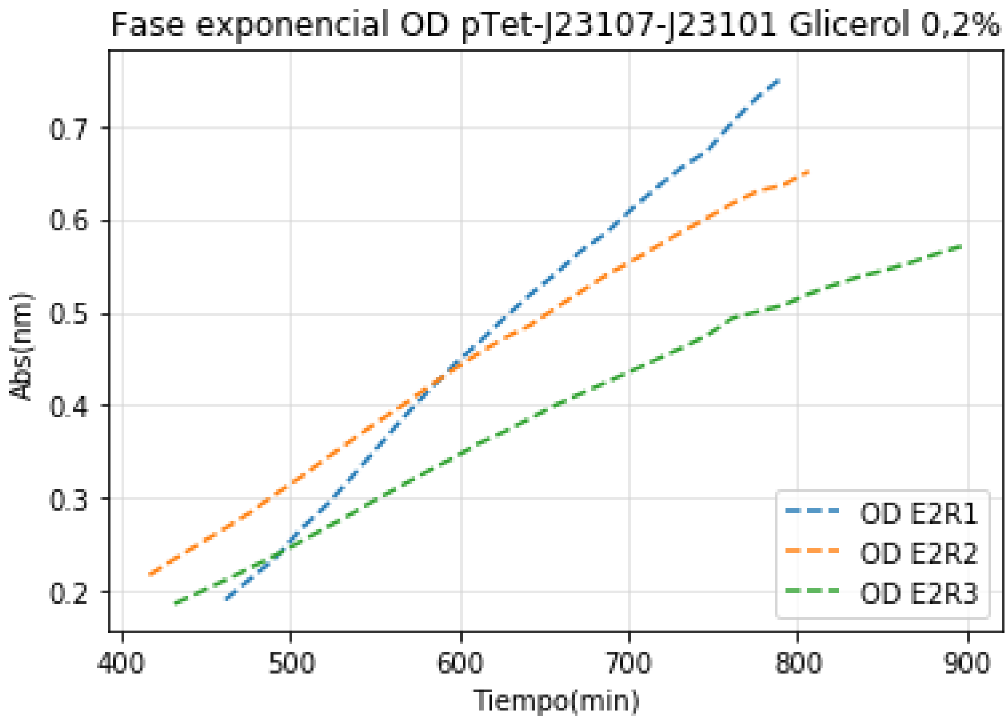
```
Out[30]: <matplotlib.legend.Legend at 0x267ecc6bfd0>
```



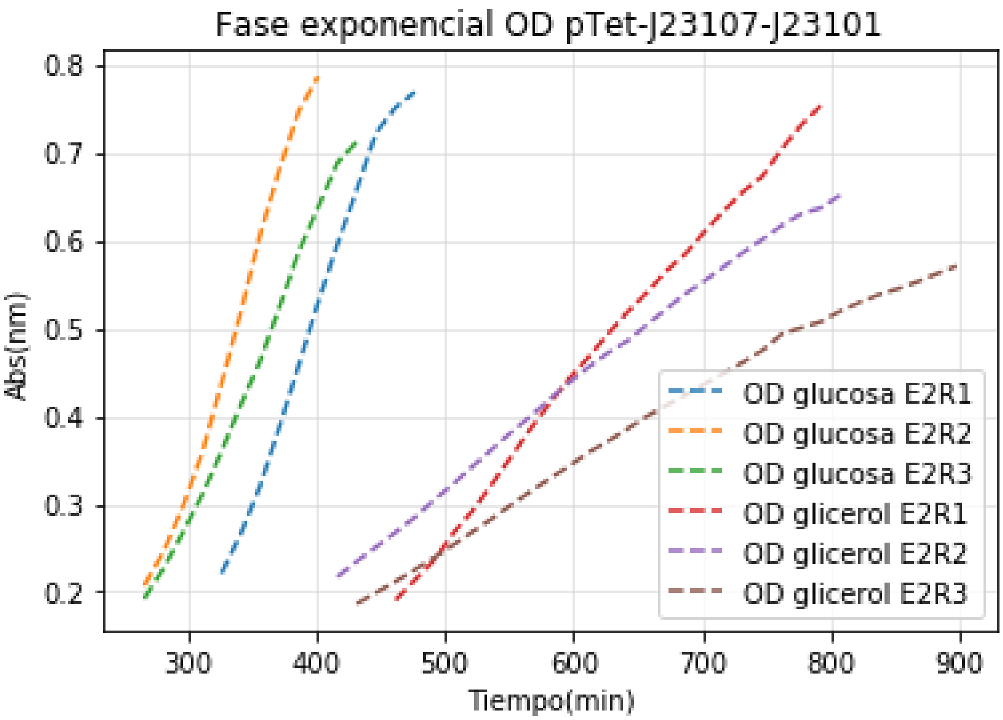



In [31]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pTet-J23107-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[30:53],od1523g1[30:53],'--',label='OD E2R1')
plt.plot(tt[27:54],od1523g2[27:54],'--',label='OD E2R2')
plt.plot(tt[28:60],od1523g3[28:60],'--',label='OD E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[31]:
```
<matplotlib.legend.Legend at 0x267ecd79a58>
```



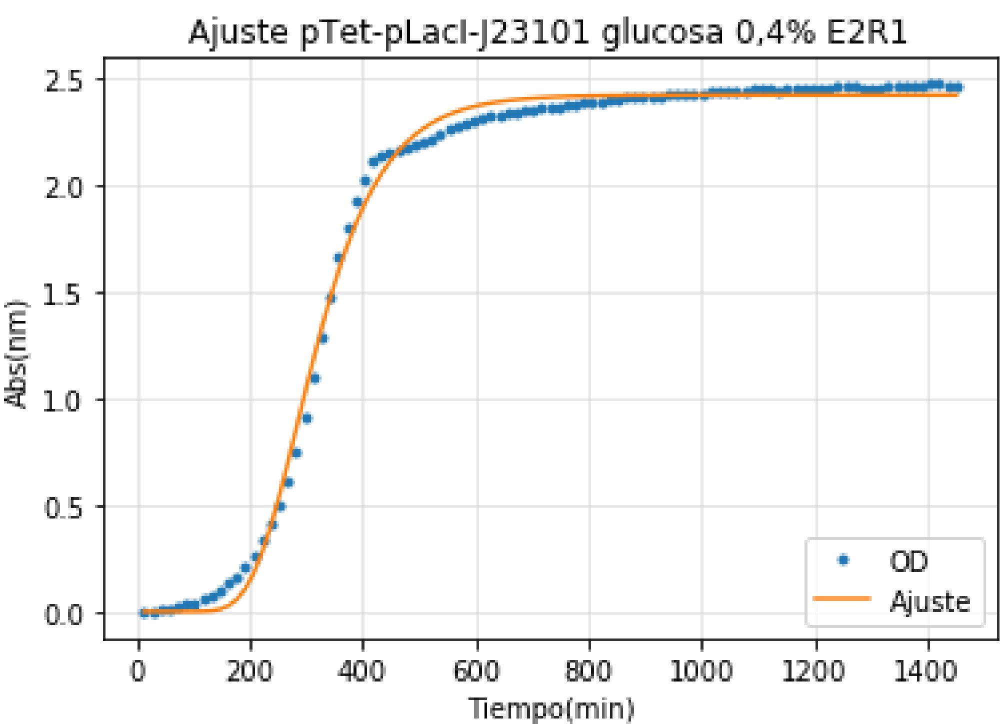

```
In [32]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23107-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[21:32],od15231[21:32],'--',label='OD glucosa E2R1')
         plt.plot(tt[17:27],od15232[17:27],'--',label='OD glucosa E2R2')
         plt.plot(tt[17:29],od15233[17:29],'--',label='OD glucosa E2R3')
         plt.plot(tt[30:53],od1523g1[30:53],'--',label='OD glicerol E2R1')
         plt.plot(tt[27:54],od1523g2[27:54],'--',label='OD glicerol E2R2')
         plt.plot(tt[28:60],od1523g3[28:60],'--',label='OD glicerol E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[32]: <matplotlib.legend.Legend at 0x267ece3afd0>
```




```
In [33]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #ptet-plac-std glucosa rep 1
         y19= np.log(od15261)-np.log(np.min(od15261))
         print('Min OD = %e'%((np.min(od15261))))
         evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste pTet-pLacI-J23101 glucosa 0,4% E
         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]
```

```python
y2=tt[28]
plt.figure()
plt.title('pTet-pLacI-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15261,label='OD pTet-pLacI-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],od15261[19:29],label='OD pTet-pLacI-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

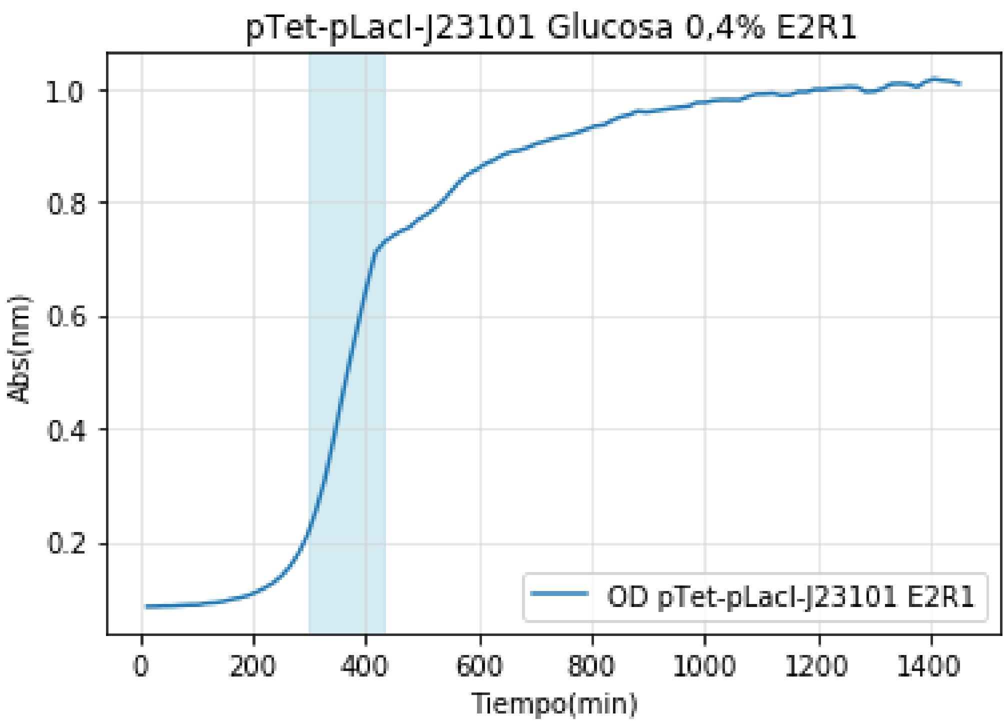
```
Min OD = 8.530000e-02
[  2.42681432e+00   1.08626281e-02   2.02426630e+02]
```

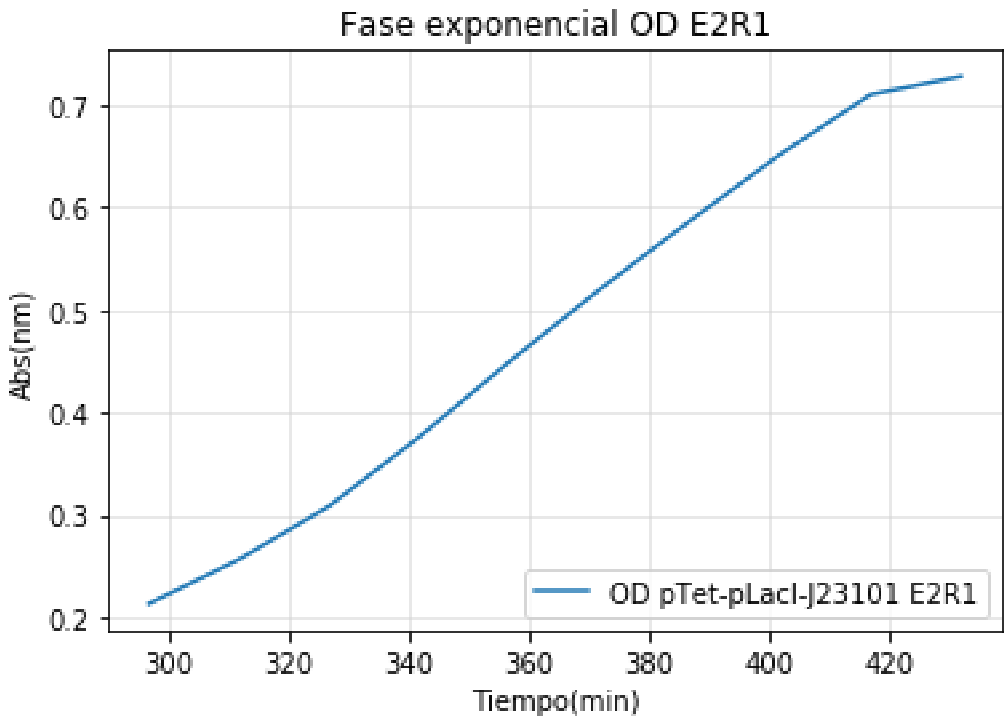



```
A=2.426814e+00
um=1.086263e-02
```

```
l=2.024266e+02
Tm=2.846144e+02
doubpe=6.381027e+01
ext=1.276205e+02
Tfinal=4.122349e+02
```

Out[33]: <matplotlib.legend.Legend at 0x267eb375ac8>

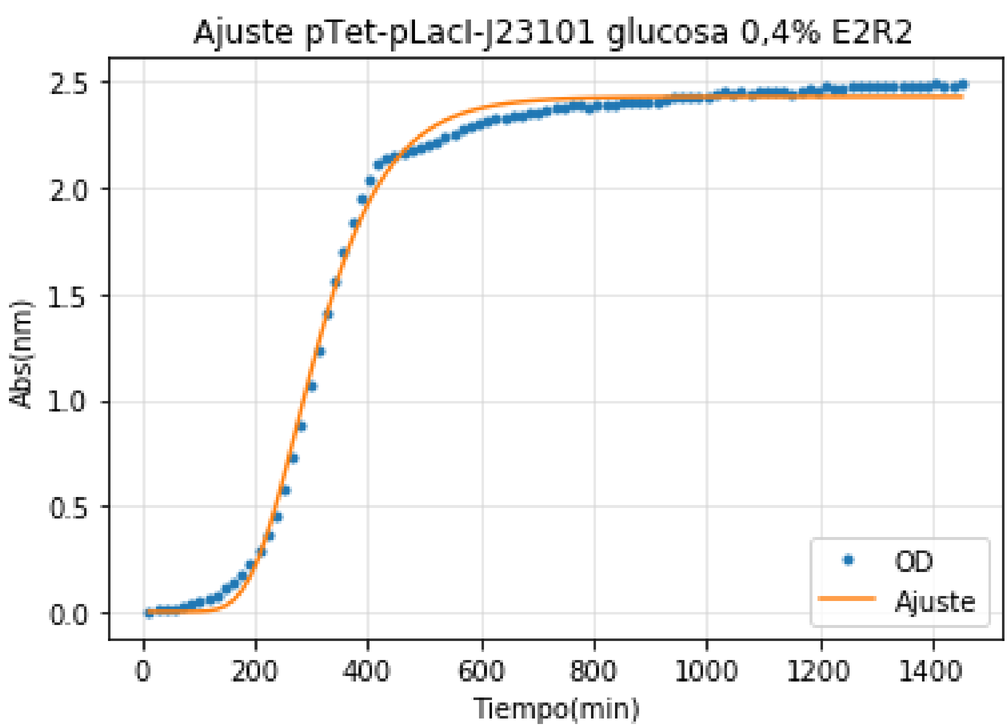





```
In [34]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #ptet-placI-std glucosa rep 2
         y20= np.log(od15262)-np.log(np.min(od15262))
         print('Min OD = %e'%((np.min(od15262))))
         evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste pTet-pLacI-J23101 glucosa 0,4% E
         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[18]
```

```
y2=tt[28]
plt.figure()
plt.title('pTet-pLacI-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15262,label='OD pTet-pLacI-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[18:29],od15262[18:29],label='OD pTet-pLacI-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

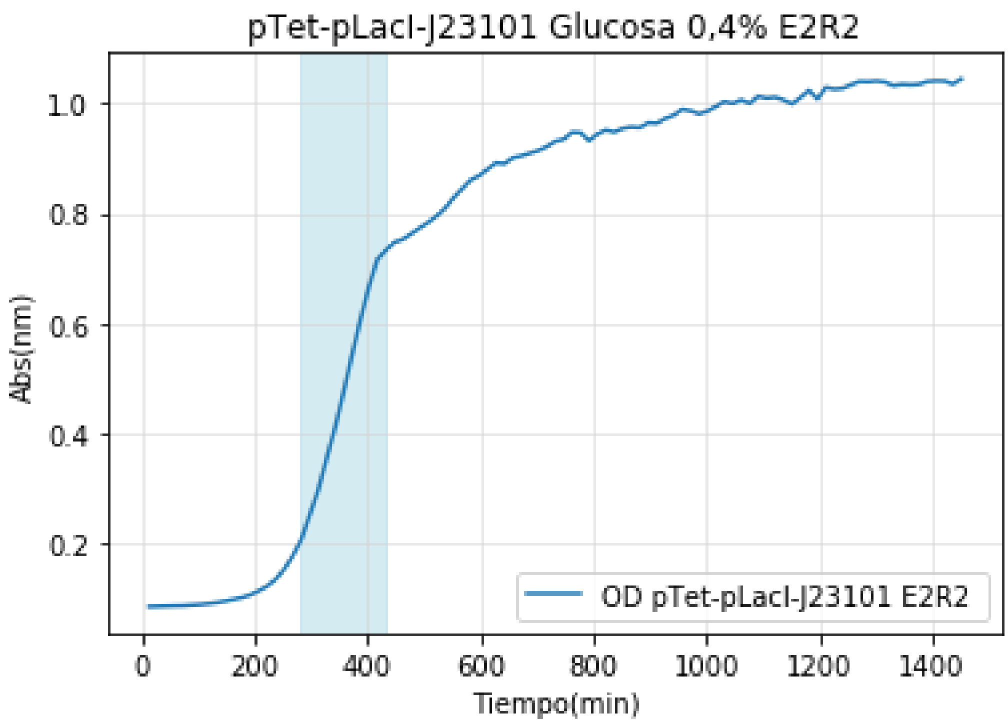
```
Min OD = 8.650000e-02
[  2.43066594e+00   1.04655379e-02   1.90115277e+02]
```

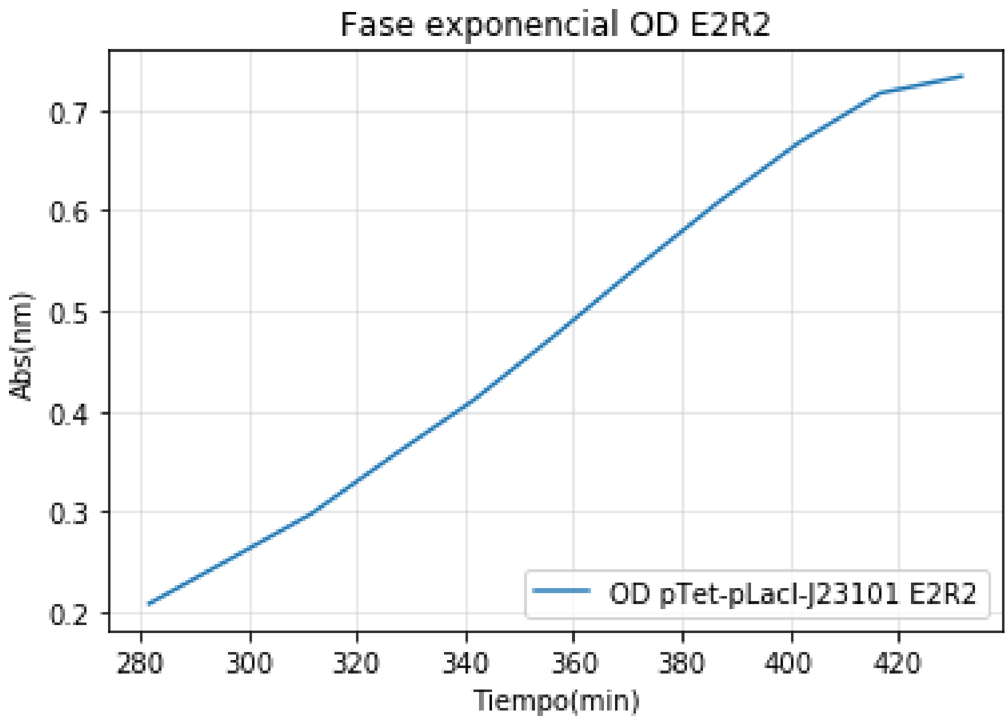



```
A=2.430666e+00
um=1.046554e-02
```

```
l=1.901153e+02
Tm=2.755569e+02
doubpe=6.623140e+01
ext=1.324628e+02
Tfinal=4.080196e+02
```

```
Out[34]: <matplotlib.legend.Legend at 0x267ecc57710>
```

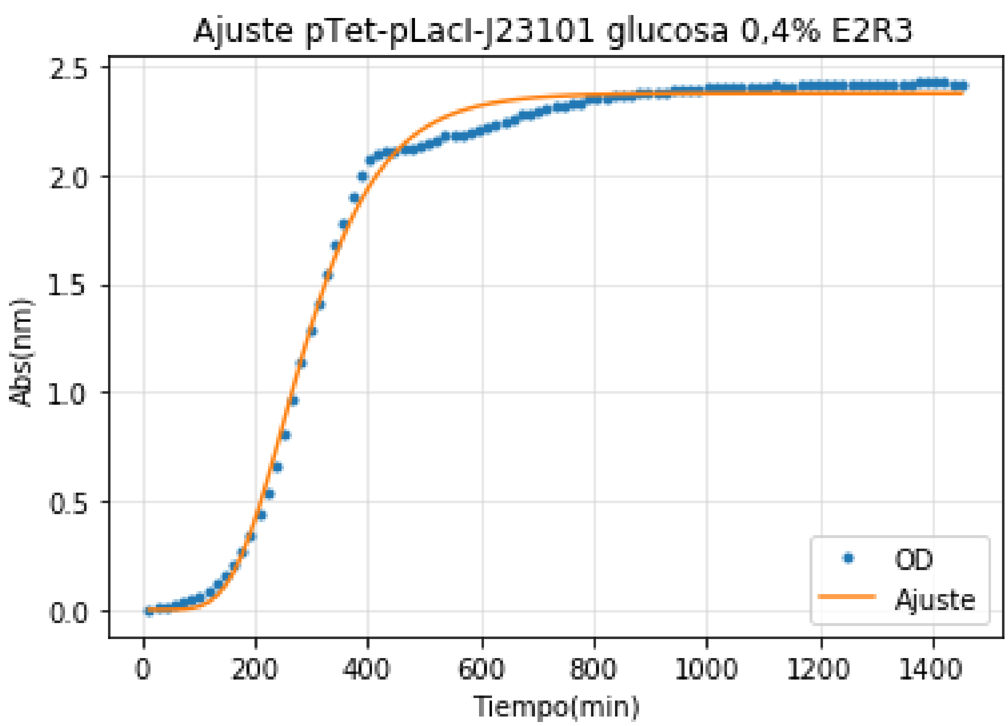





In [35]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-plac-std glucosa rep 3
y21= np.log(od15263)-np.log(np.min(od15263))
print('Min OD = %e'%((np.min(od15263))))
evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste pTet-pLacI-J23101 glucosa 0,4% E
A21= params[0]
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]
```

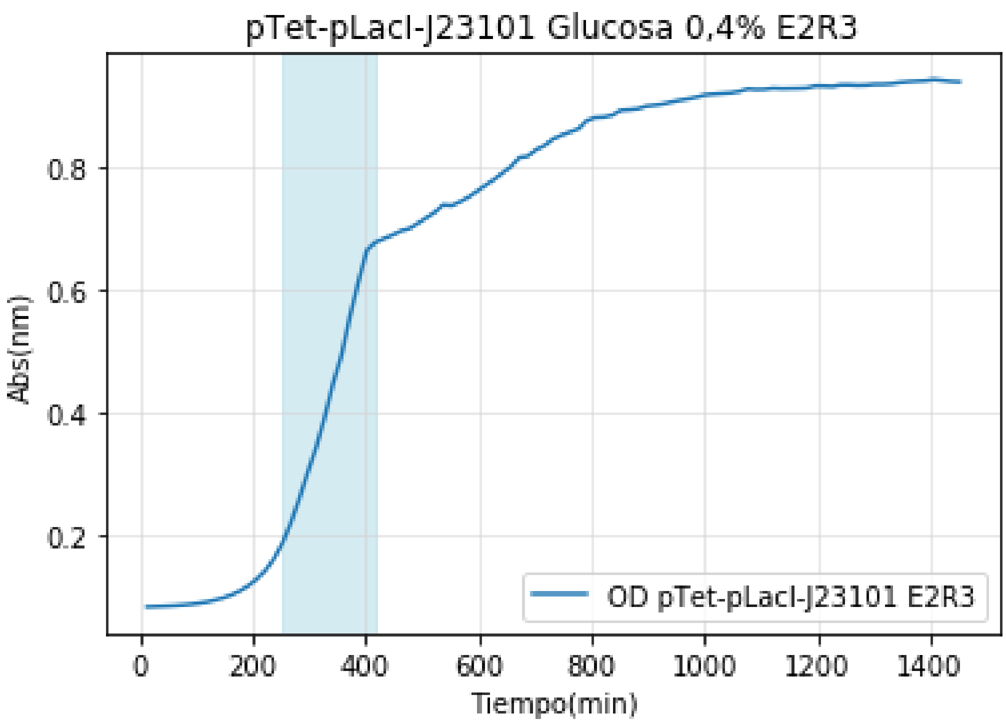
```
y2=tt[27]
plt.figure()
plt.title('pTet-pLacI-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15263,label='OD pTet-pLacI-J23101 E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')


#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:28],od15263[16:28],label='OD pTet-pLacI-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

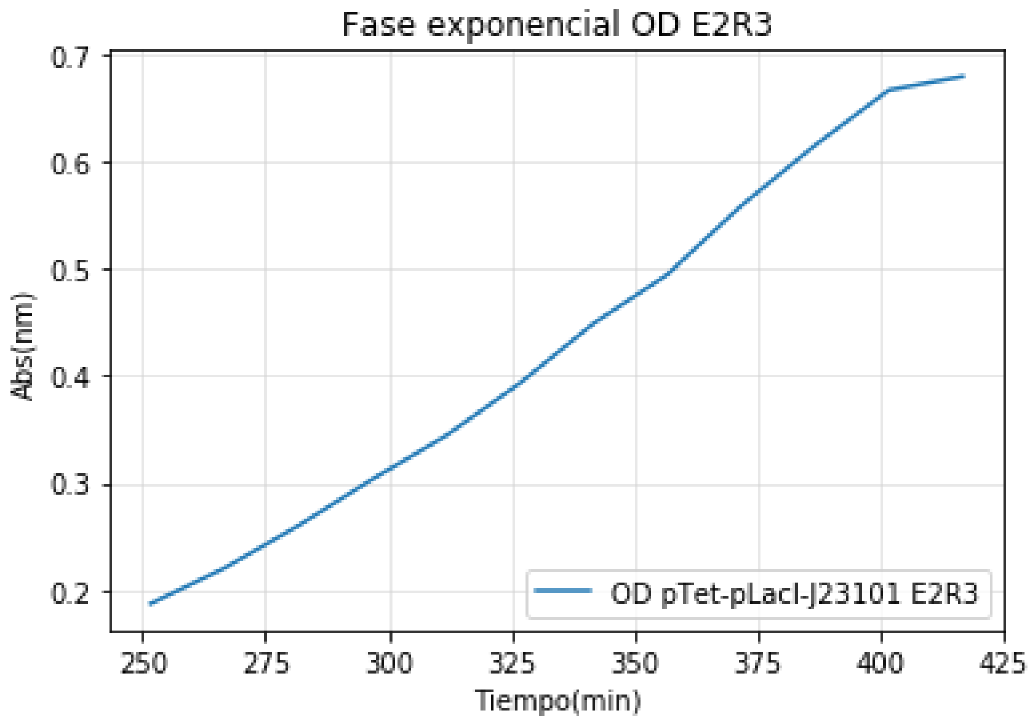
```
Min OD = 8.370000e-02
[  2.37459827e+00   9.36201784e-03   1.58277013e+02]
```



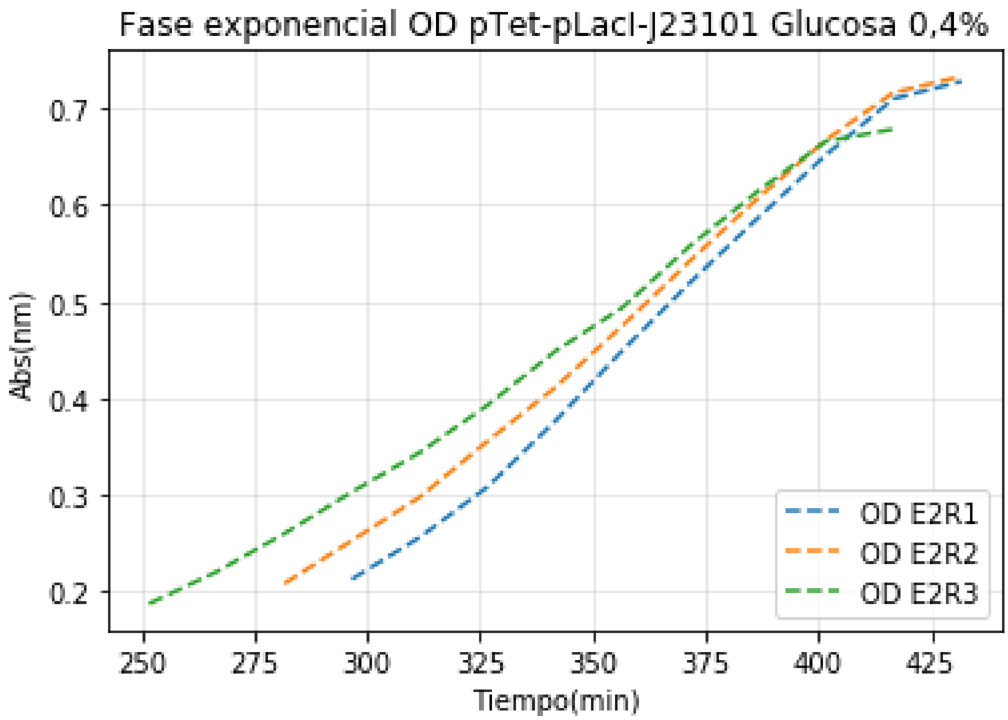
```
A=2.374598e+00
um=9.362018e-03
l=1.582770e+02
Tm=2.515866e+02
doubpe=7.403822e+01
ext=1.480764e+02
Tfinal=3.996630e+02
```

```
Out[35]: <matplotlib.legend.Legend at 0x267ed081518>
```





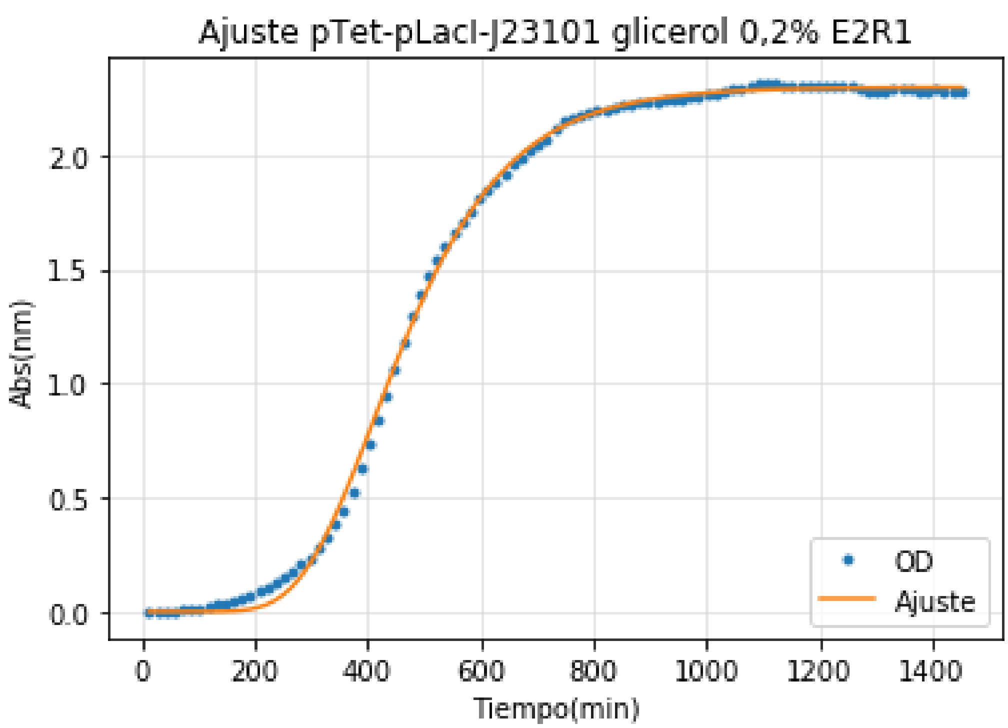

```
In [36]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-pLacI-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:29],od15261[19:29],'--',label='OD E2R1')
         plt.plot(tt[18:29],od15262[18:29],'--',label='OD E2R2')
         plt.plot(tt[16:28],od15263[16:28],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[36]: <matplotlib.legend.Legend at 0x267ed196eb8>
```




In [37]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-plac-std glicerol rep 1
y22= np.log(od1526g1)-np.log(np.min(od1526g1))
print('Min OD = %e'%((np.min(od1526g1))))
evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste pTet-pLacI-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[27]
```

```
y2=tt[46]
plt.figure()
plt.title('pTet-pLacI-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1526g1,label='OD pTet-pLacI-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[27:47],od1526g1[27:47],label='OD pTet-pLacI-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

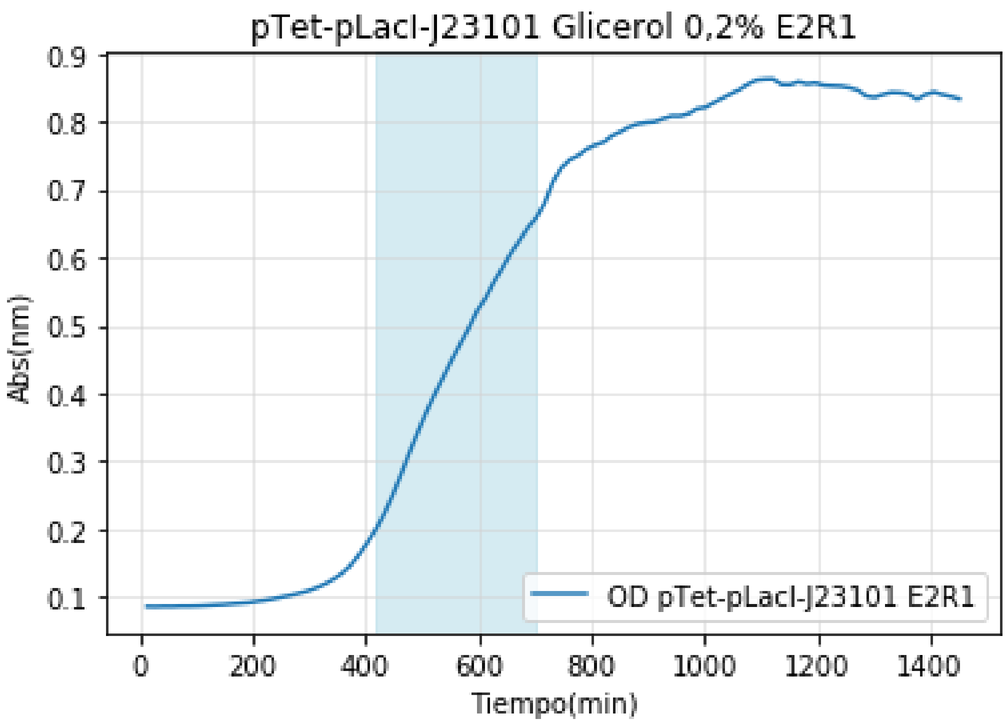
```
Min OD = 8.570000e-02
[  2.29532726e+00   6.46446174e-03   2.80386578e+02]
```

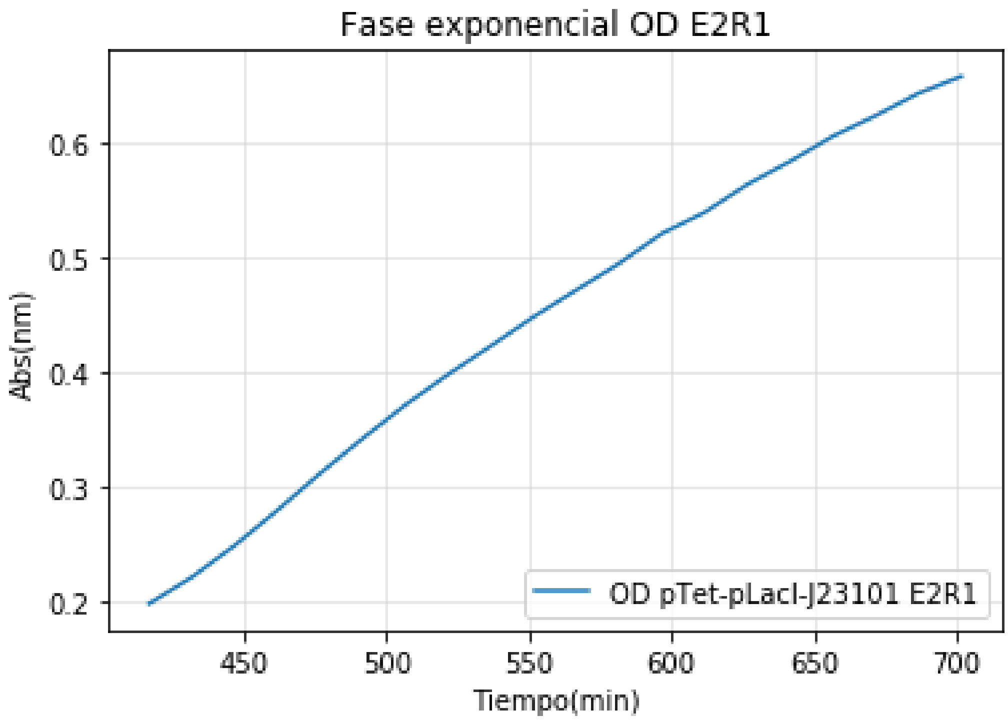



```
A=2.295327e+00
um=6.464462e-03
```

```
l=2.803866e+02
Tm=4.110090e+02
doubpe=1.072243e+02
ext=2.680607e+02
Tfinal=6.790697e+02
```

Out[37]: `<matplotlib.legend.Legend at 0x267ecf91c50>`

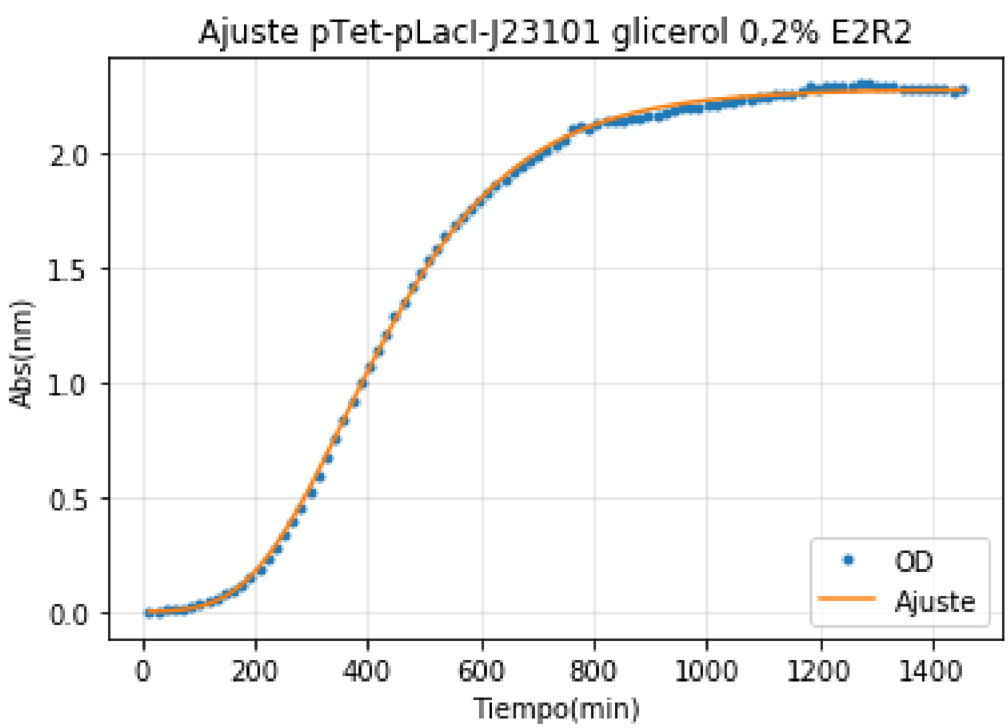





In [38]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-plac-std glicerol rep 2
y23= np.log(od1526g2)-np.log(np.min(od1526g2))
print('Min OD = %e'%((np.min(od1526g2))))
evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste pTet-pLacI-J23101 glicerol 0,2%
A23= 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[24]
```

```
y2=tt[48]
plt.figure()
plt.title('pTet-pLacI-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1526g2,label='OD pTet-pLacI-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[24:49],od1526g2[24:49],label='OD pTet-pLacI-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

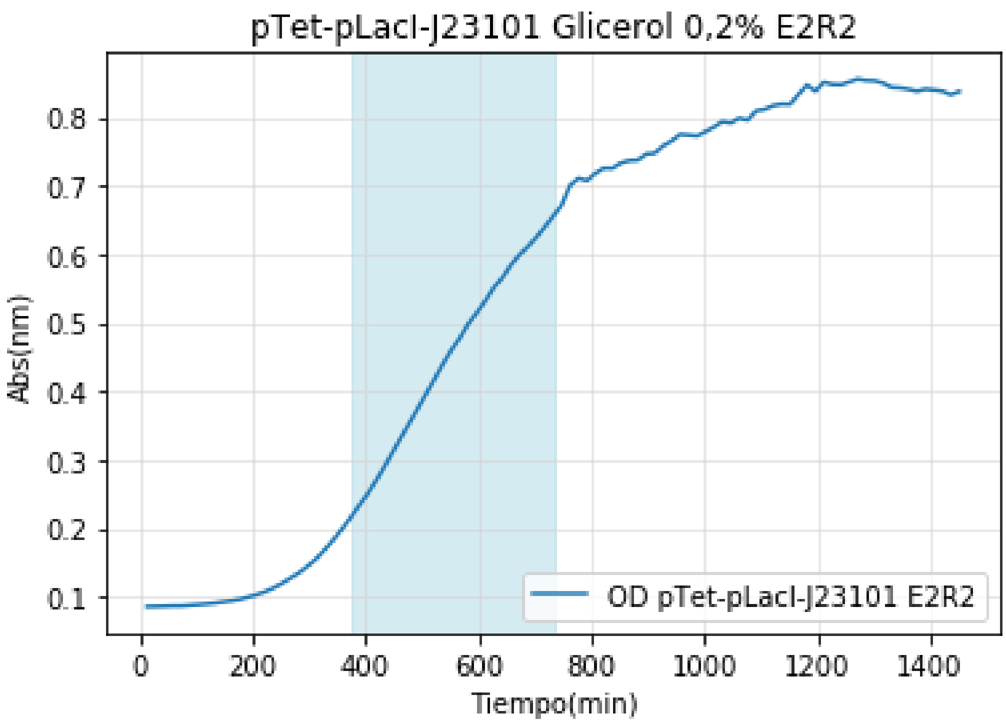
```
Min OD = 8.610000e-02
[  2.27382560e+00   4.99992320e-03   1.89857406e+02]
```

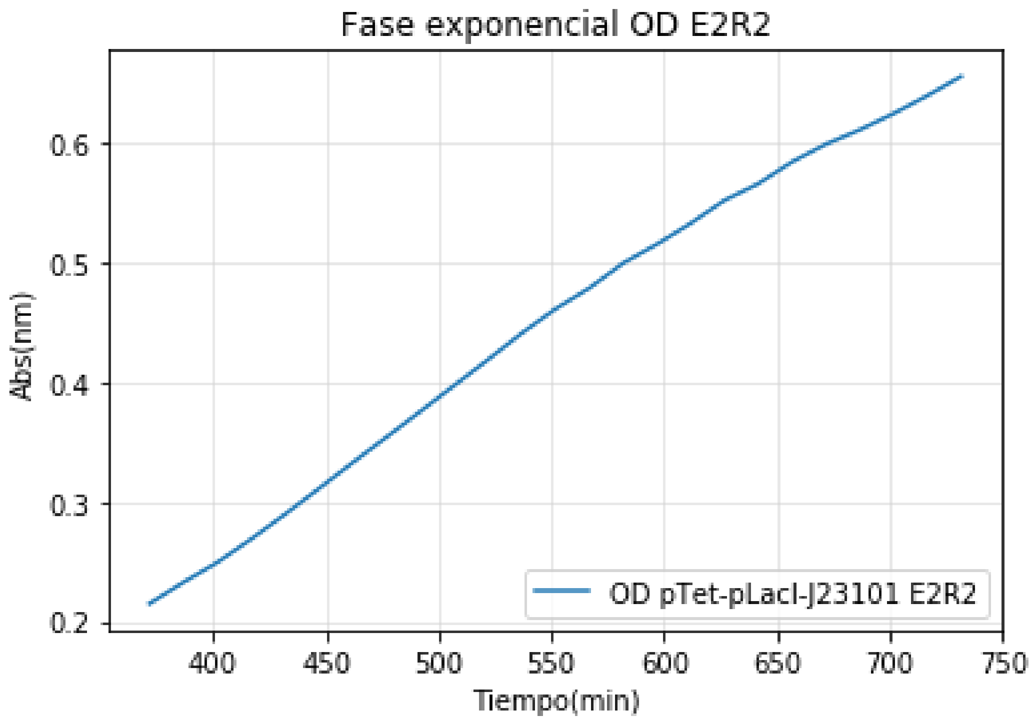



```
A=2.273826e+00
um=4.999923e-03
```

```
l=1.898574e+02
Tm=3.571587e+02
doubpe=1.386316e+02
ext=3.465789e+02
Tfinal=7.037376e+02
```

```
Out[38]: <matplotlib.legend.Legend at 0x267ec9e37f0>
```

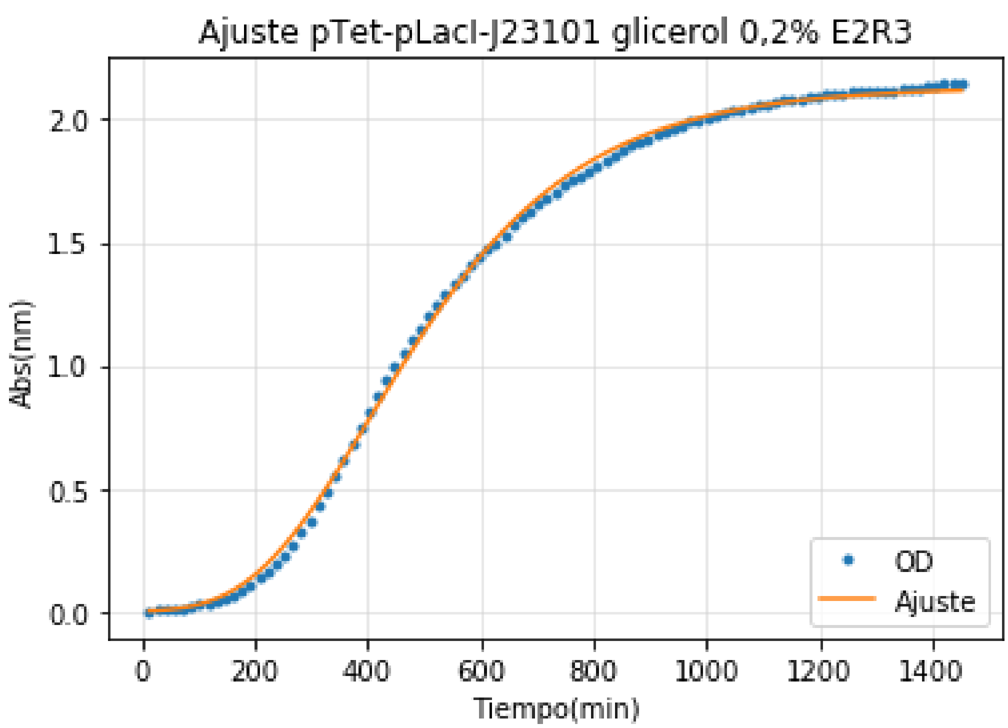





```
In [39]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #pTet-plac-std glicerol rep 3
         y24= np.log(od1526g3)-np.log(np.min(od1526g3))
         print('Min OD = %e'%((np.min(od1526g3))))
         evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste pTet-pLacI-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=2.5*t224
         print('ext=%e'%extdp24)
         ttot24=tm24+extdp24
         print('Tfinal=%e'%ttot24)

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

```
y2=tt[58]
plt.figure()
plt.title('pTet-pLacI-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1526g3,label='OD pTet-pLacI-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[27:59],od1526g3[27:59],label='OD pTet-pLacI-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

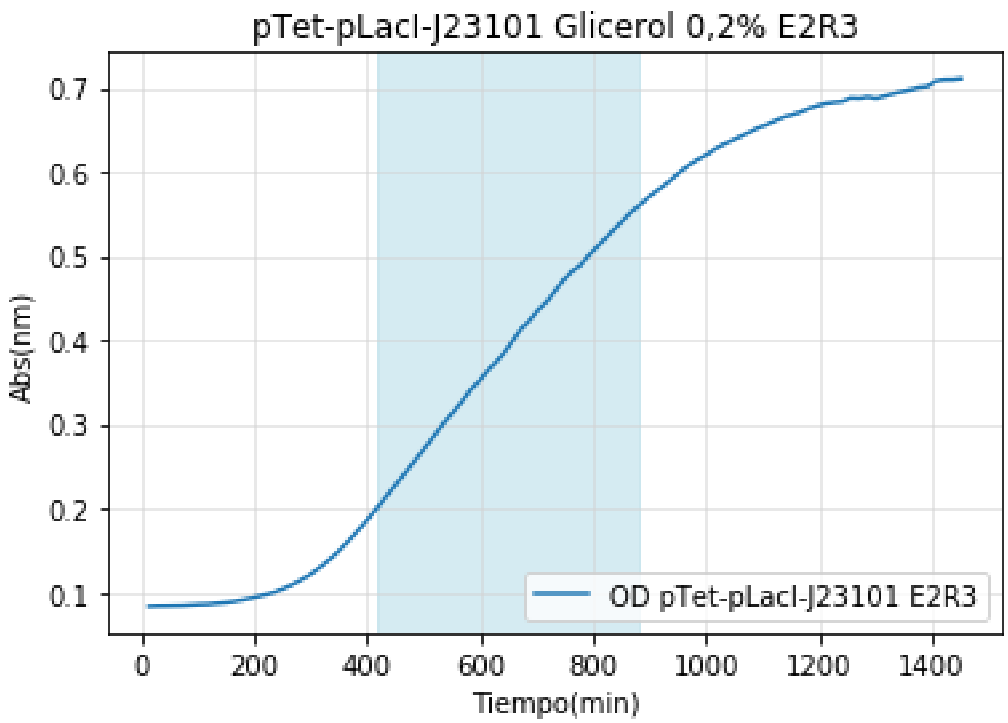
```
Min OD = 8.340000e-02
[  2.13015451e+00   3.76288308e-03   1.94366098e+02]
```

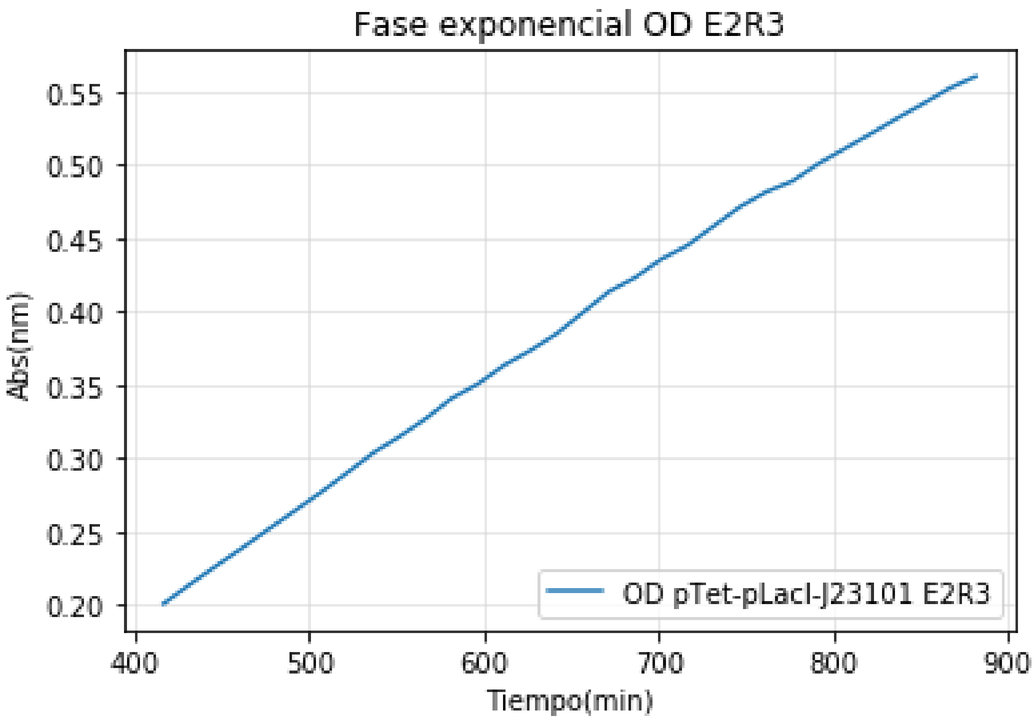



```
A=2.130155e+00
um=3.762883e-03
```

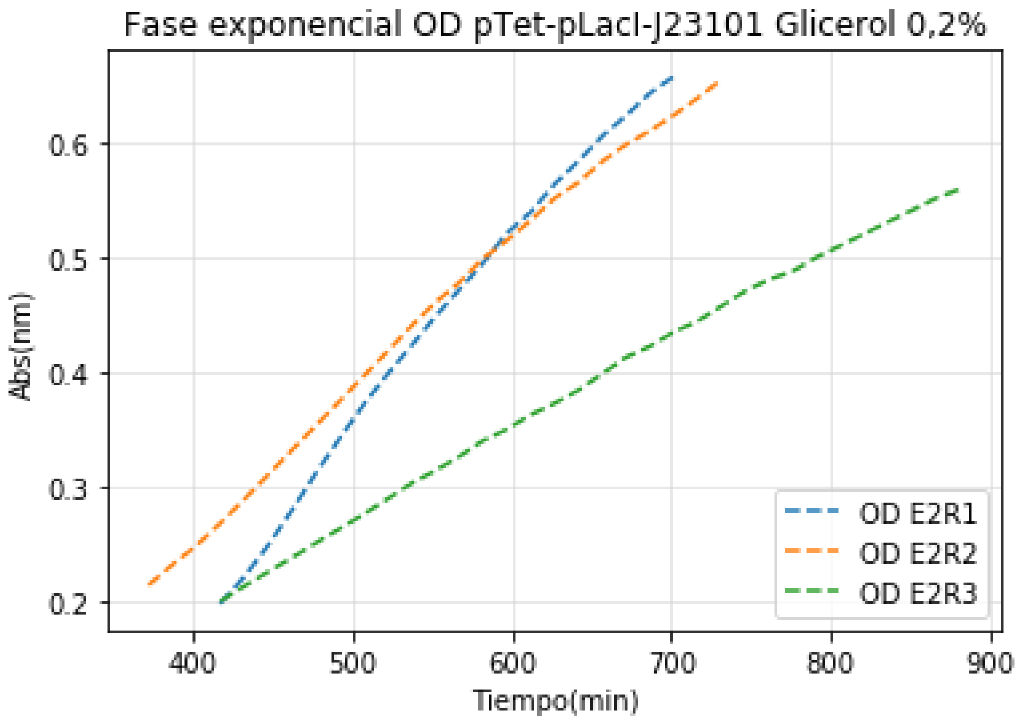
```
l=1.943661e+02
Tm=4.026213e+02
doubpe=1.842064e+02
ext=4.605160e+02
Tfinal=8.631373e+02
```

Out[39]: <matplotlib.legend.Legend at 0x267ed23d4a8>

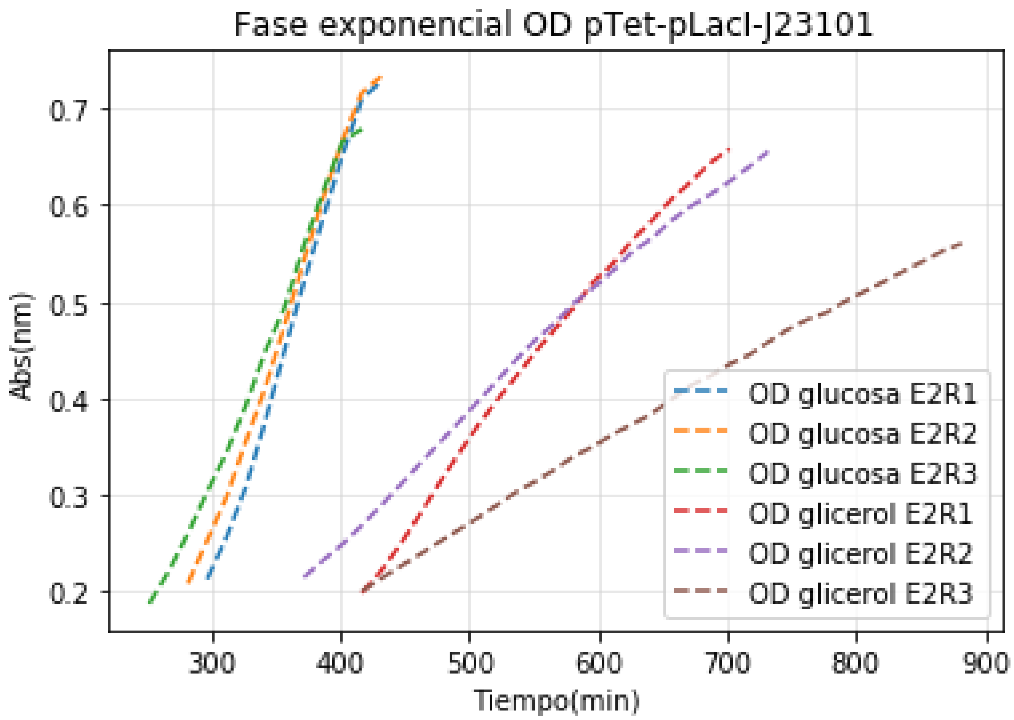





```
In [40]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-pLacI-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[27:47],od1526g1[27:47],'--',label='OD E2R1')
         plt.plot(tt[24:49],od1526g2[24:49],'--',label='OD E2R2')
         plt.plot(tt[27:59],od1526g3[27:59],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[40]: <matplotlib.legend.Legend at 0x267ed3a9c88>
```



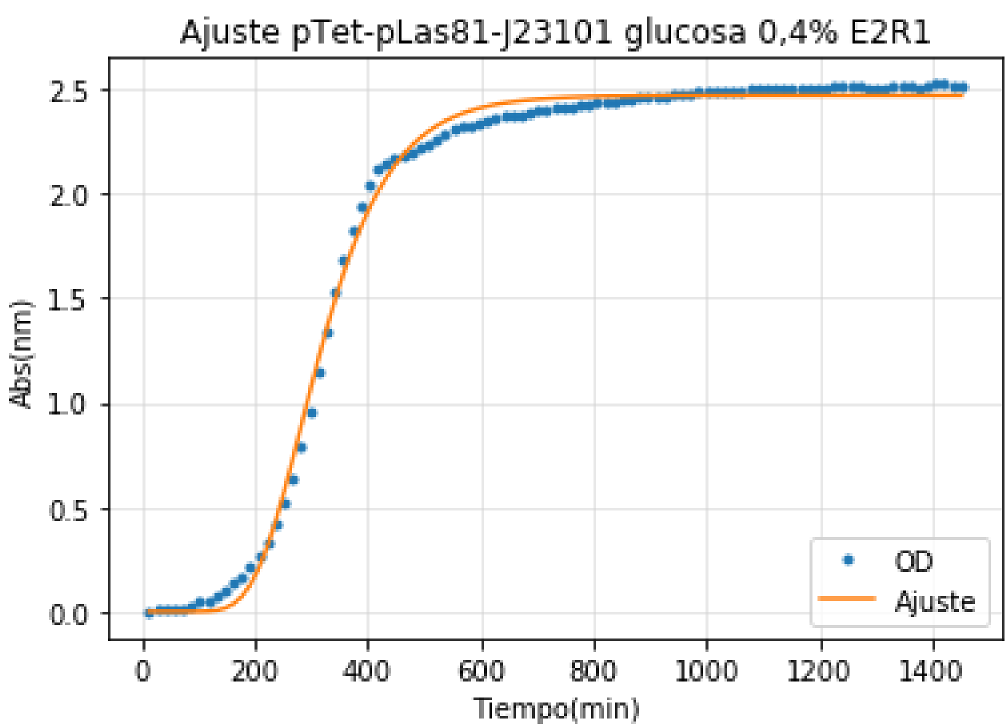

```
In [41]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-pLacI-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:29],od15261[19:29],'--',label='OD glucosa E2R1')
         plt.plot(tt[18:29],od15262[18:29],'--',label='OD glucosa E2R2')
         plt.plot(tt[16:28],od15263[16:28],'--',label='OD glucosa E2R3')
         plt.plot(tt[27:47],od1526g1[27:47],'--',label='OD glicerol E2R1')
         plt.plot(tt[24:49],od1526g2[24:49],'--',label='OD glicerol E2R2')
         plt.plot(tt[27:59],od1526g3[27:59],'--',label='OD glicerol E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[41]: <matplotlib.legend.Legend at 0x267ed46ce80>
```




In [42]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-plas-std glucosa rep 1
y25= np.log(od15271)-np.log(np.min(od15271))
print('Min OD = %e'%((np.min(od15271))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste pTet-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[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[18:29],od15271[18:29],label='OD pTet-pLas81-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

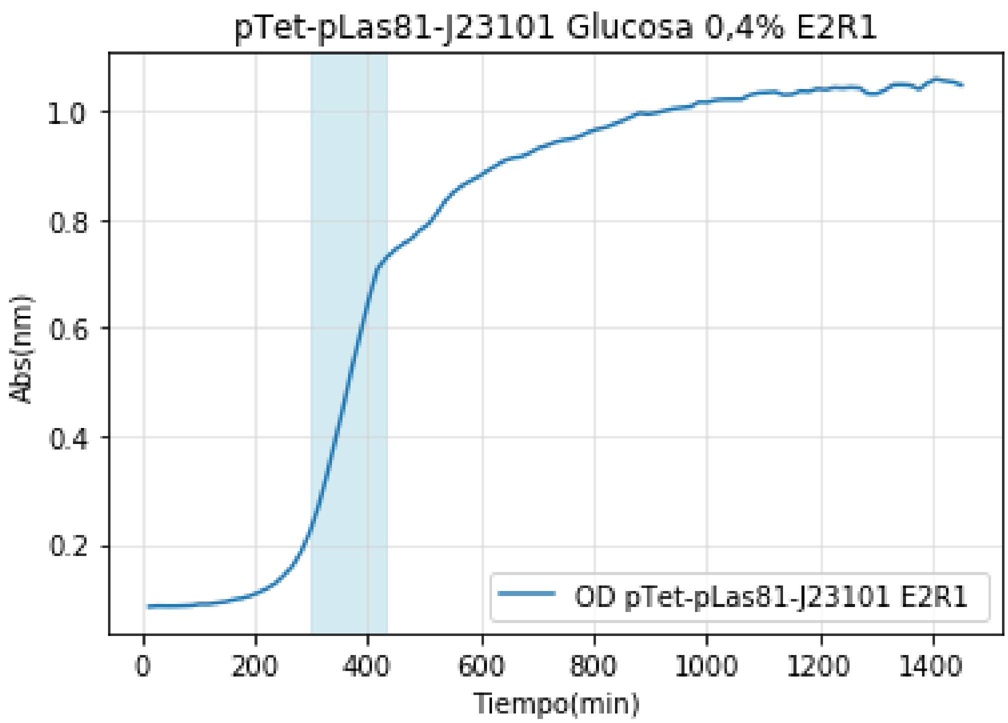
```
Min OD = 8.510000e-02
[  2.46838538e+00   1.06746902e-02   1.98321599e+02]
```

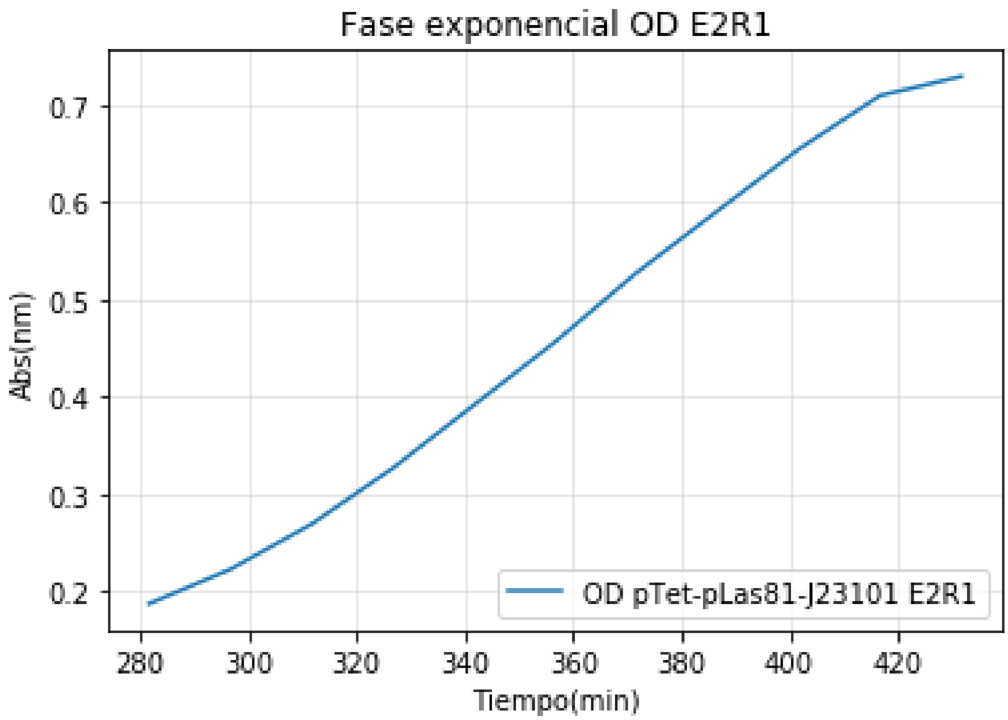



```
A=2.468385e+00
um=1.067469e-02
```

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

```
Out[42]: <matplotlib.legend.Legend at 0x267ed61a6d8>
```

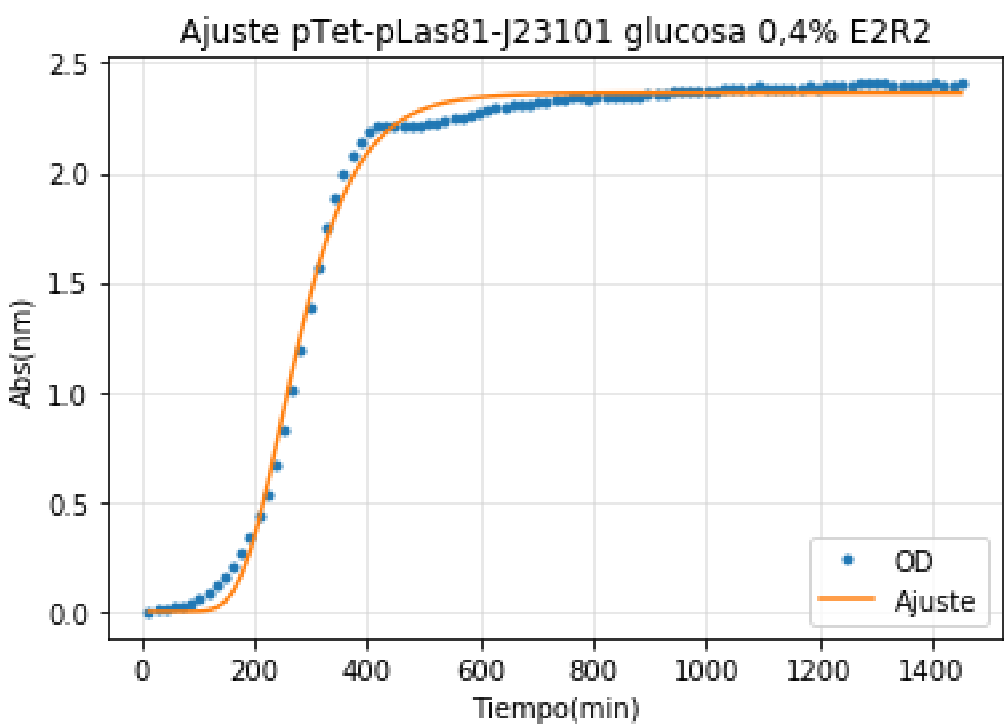





In [43]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-pLas-std glucosa rep 2
y26= np.log(od15272)-np.log(np.min(od15272))
print('Min OD = %e'%((np.min(od15272))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste pTet-pLas81-J23101 glucosa 0,4%
A26= params[0]
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[16]
```

```python
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')
```

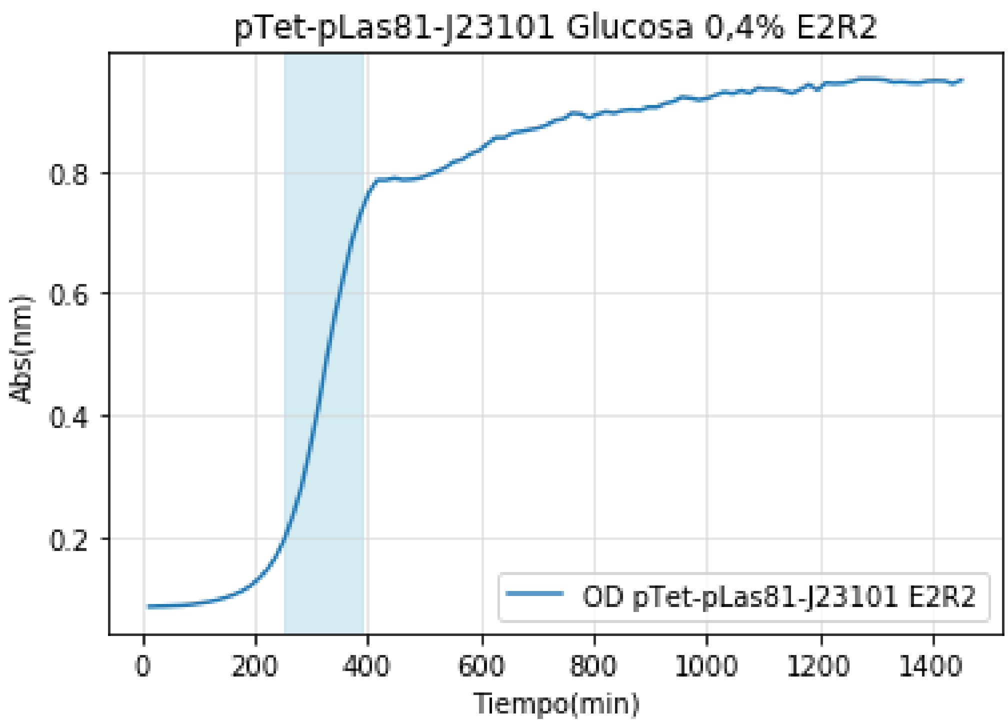
```
Min OD = 8.580000e-02
[  2.36424970e+00   1.19691673e-02   1.74474154e+02]
```

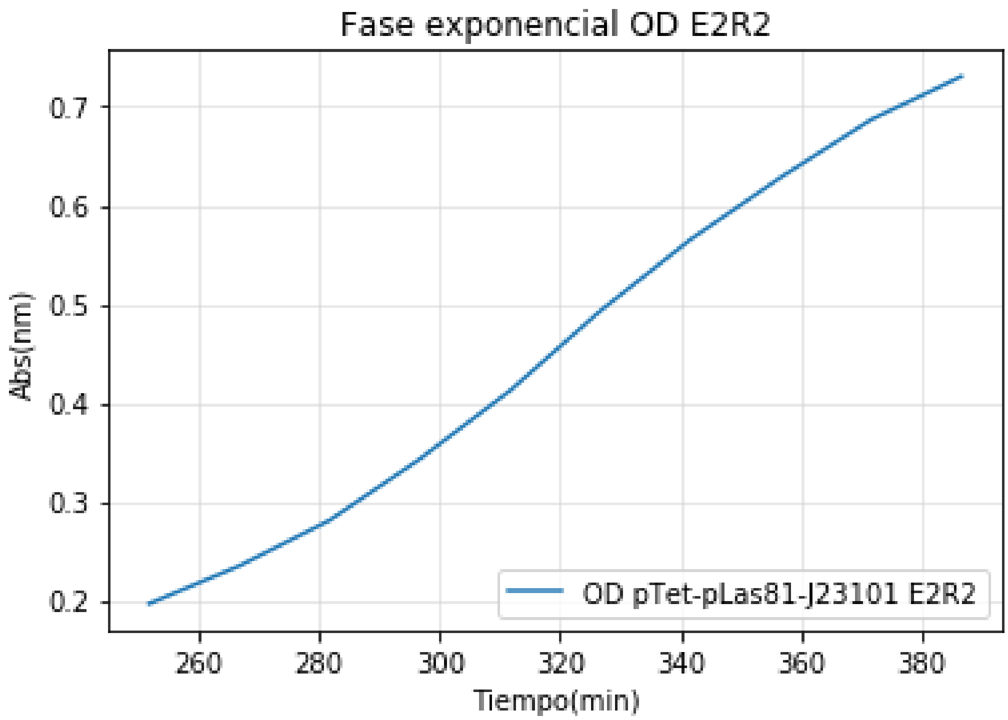



```
A=2.364250e+00
um=1.196917e-02
```

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

```
Out[43]: <matplotlib.legend.Legend at 0x267ecbb8a20>
```

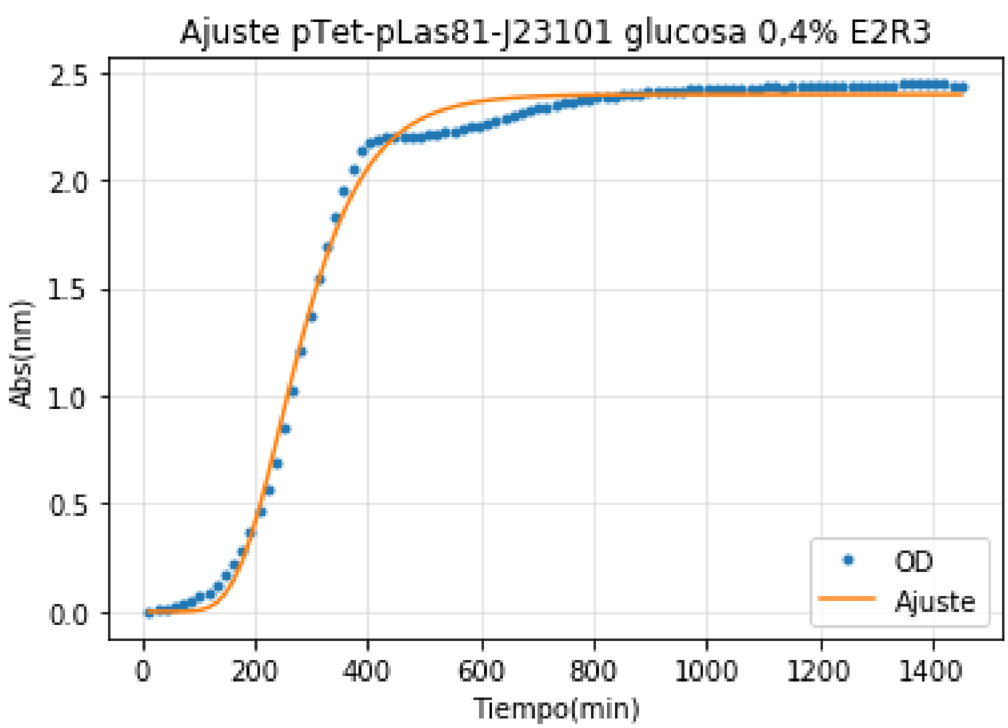





In [44]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-pLas-std glucosa rep 3
y27= np.log(od15273)-np.log(np.min(od15273))
print('Min OD = %e'%((np.min(od15273))))
evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste pTet-pLas81-J23101 glucosa 0,4%
A27= params[0]
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[16]
```

```python
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')
```

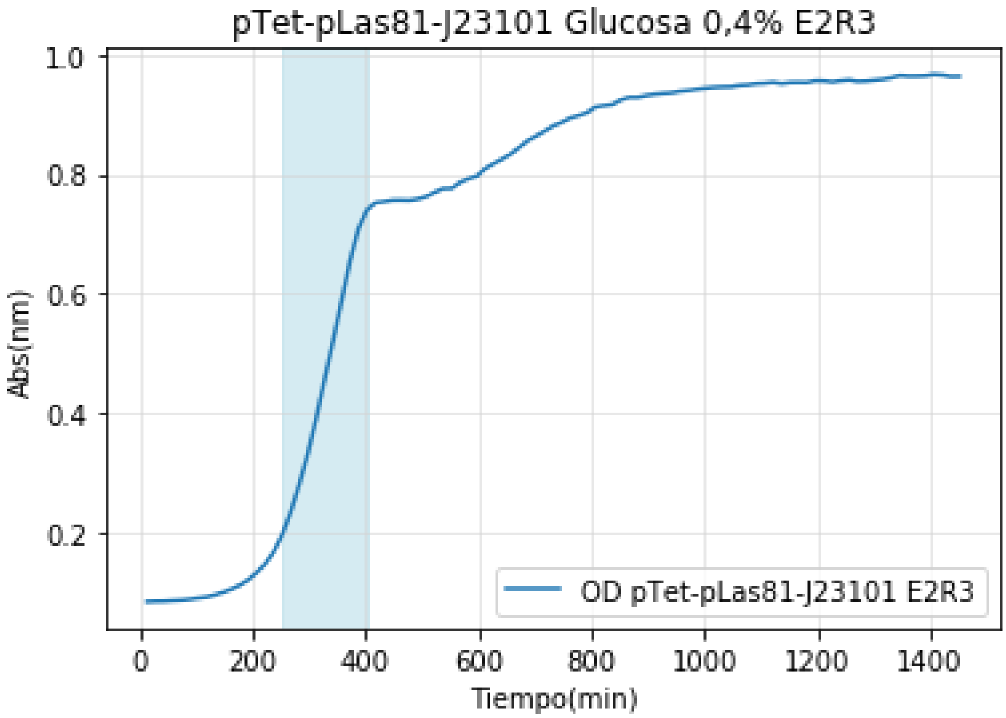
```
Min OD = 8.390000e-02
[  2.39690020e+00   1.07573329e-02   1.64117750e+02]
```

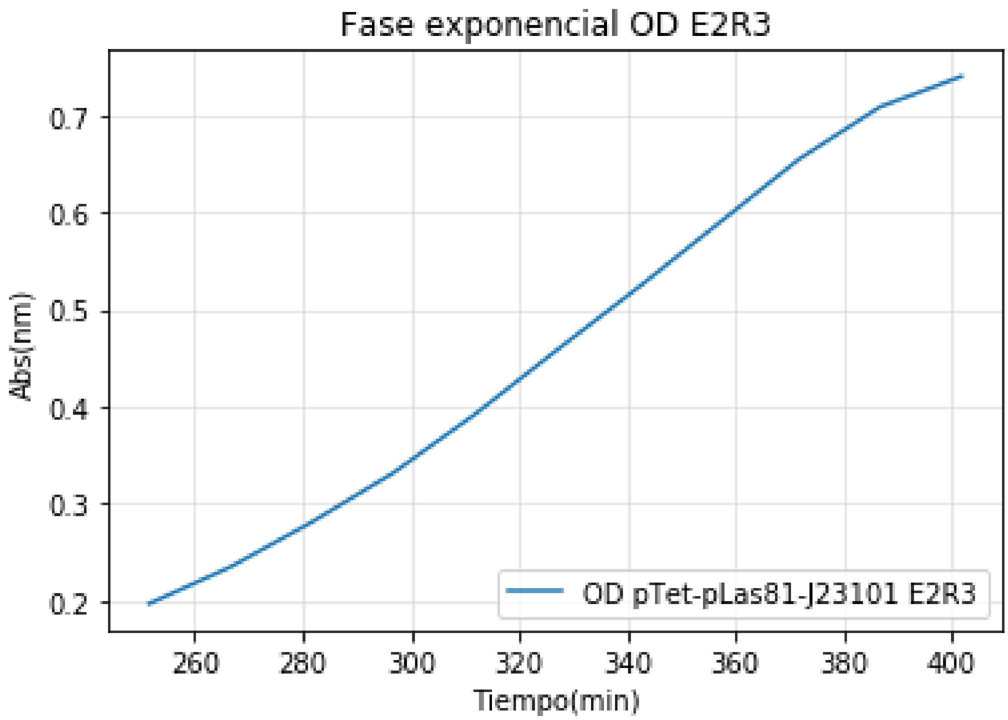



```
A=2.396900e+00
um=1.075733e-02
```

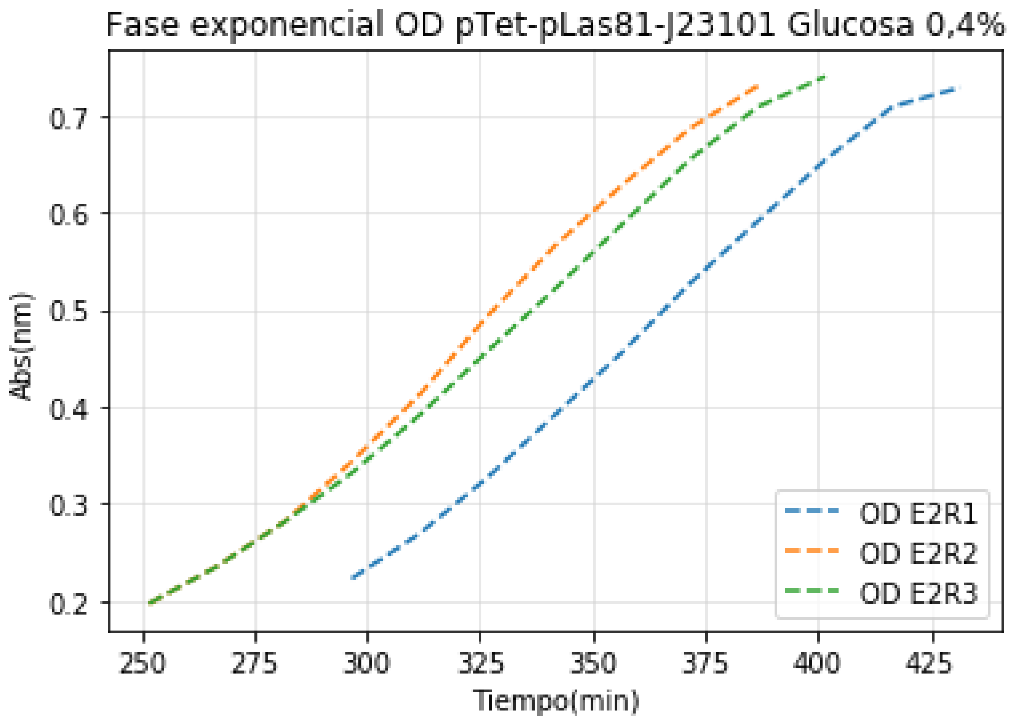
```
l=1.641177e+02
Tm=2.460870e+02
doubpe=6.443485e+01
ext=1.288697e+02
Tfinal=3.749567e+02
```

Out[44]: `<matplotlib.legend.Legend at 0x267ed24cf60>`





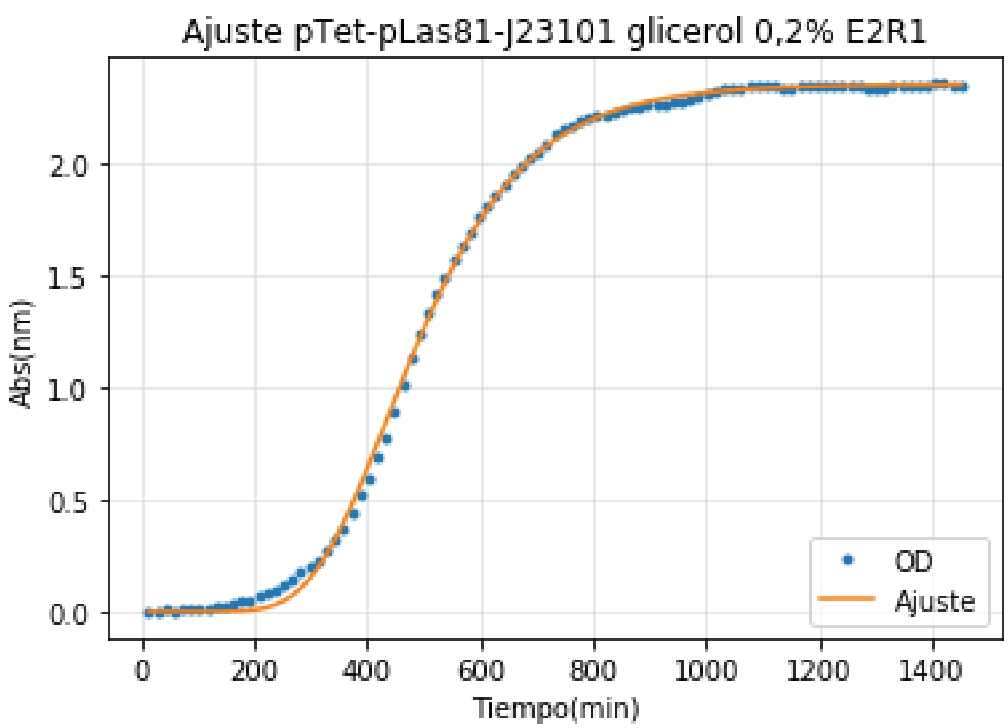

```
In [45]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-pLas81-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:29],od15271[19:29],'--',label='OD E2R1')
         plt.plot(tt[16:26],od15272[16:26],'--',label='OD 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[45]: <matplotlib.legend.Legend at 0x267ee6db710>
```




In [46]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-pLas-std glicerol rep 1
y28= np.log(od1527g1)-np.log(np.min(od1527g1))
print('Min OD = %e'%((np.min(od1527g1))))
evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%
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[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')
```

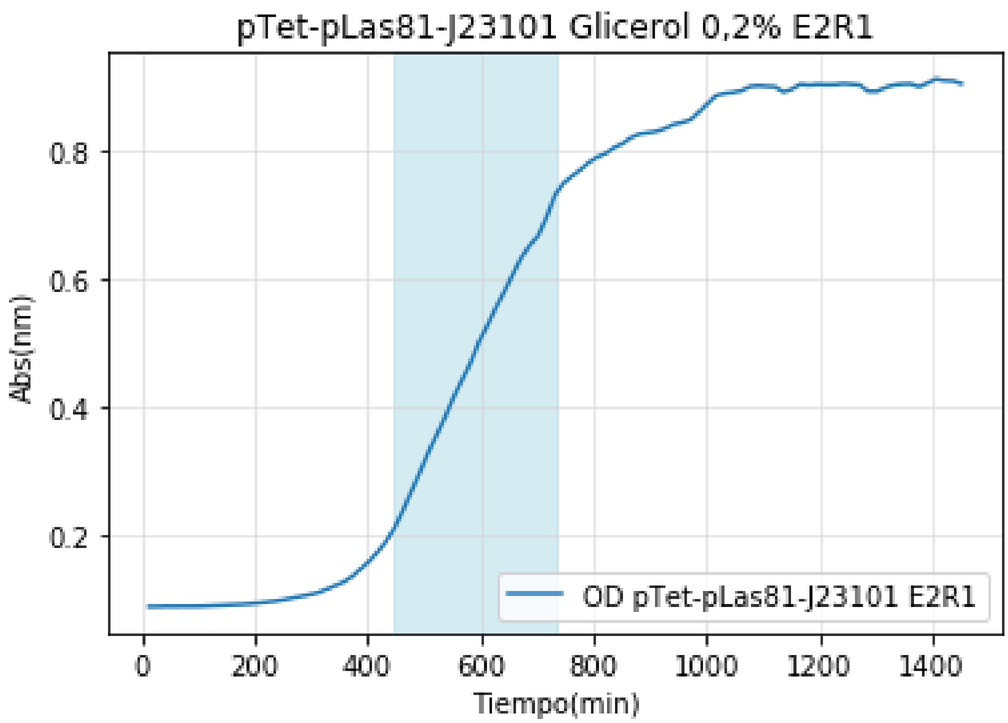
```
Min OD = 8.620000e-02
[  2.35645905e+00   6.42374075e-03   3.00000000e+02]
```

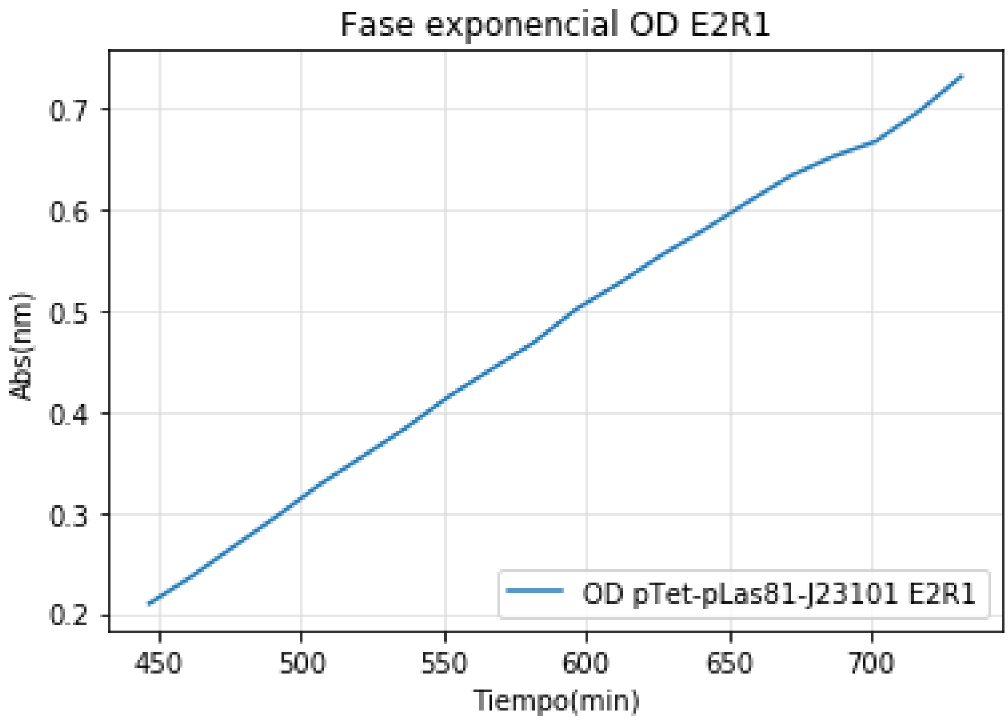



```
A=2.356459e+00
um=6.423741e-03
```

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

```
Out[46]: <matplotlib.legend.Legend at 0x267ee8719b0>
```

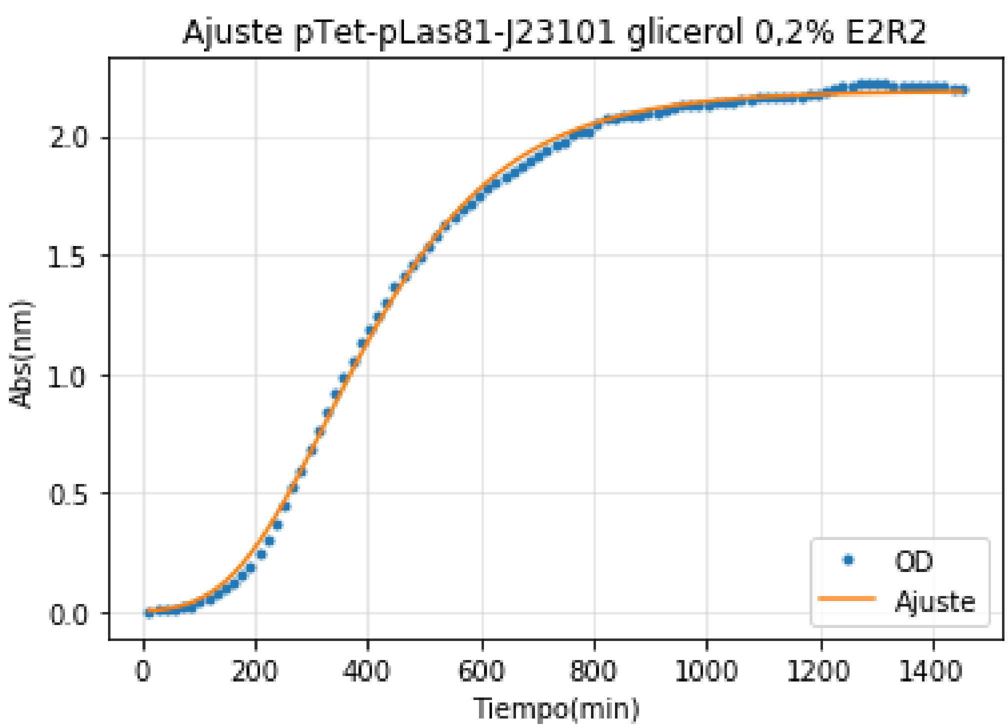





```
In [47]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #ptet-pLas-std glicerol rep 2
         y29= np.log(od1527g2)-np.log(np.min(od1527g2))
         print('Min OD = %e'%((np.min(od1527g2))))
         evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%
         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[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')
```

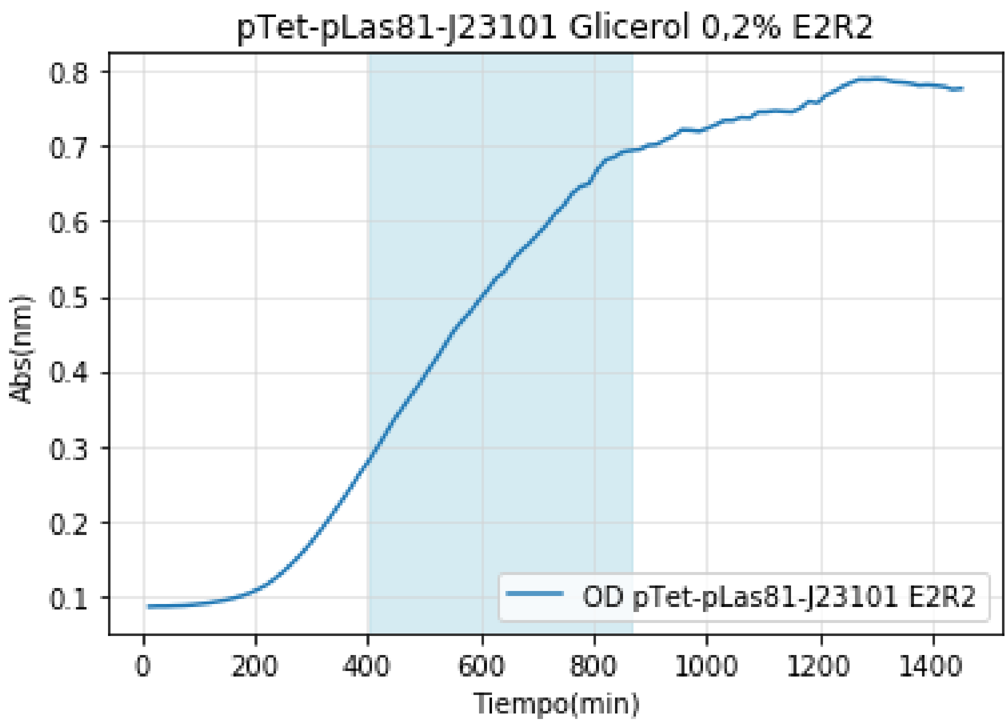
```
Min OD = 8.590000e-02
[  2.18898193e+00   4.67358353e-03   1.54692670e+02]
```

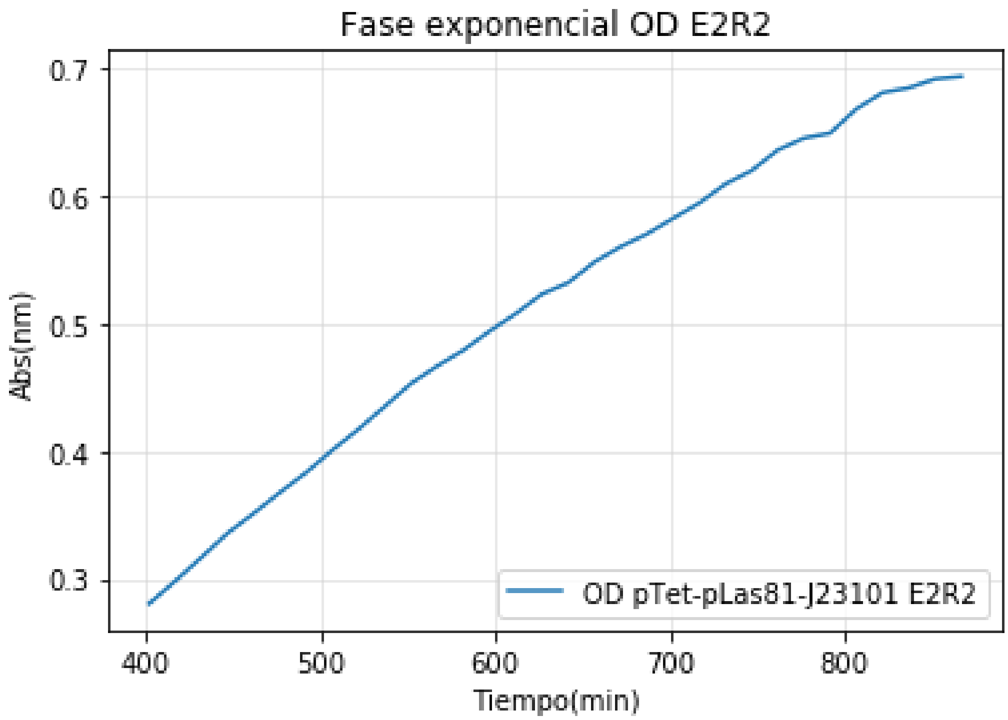



```
A=2.188982e+00
um=4.673584e-03
```

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

```
Out[47]: <matplotlib.legend.Legend at 0x267eea668d0>
```

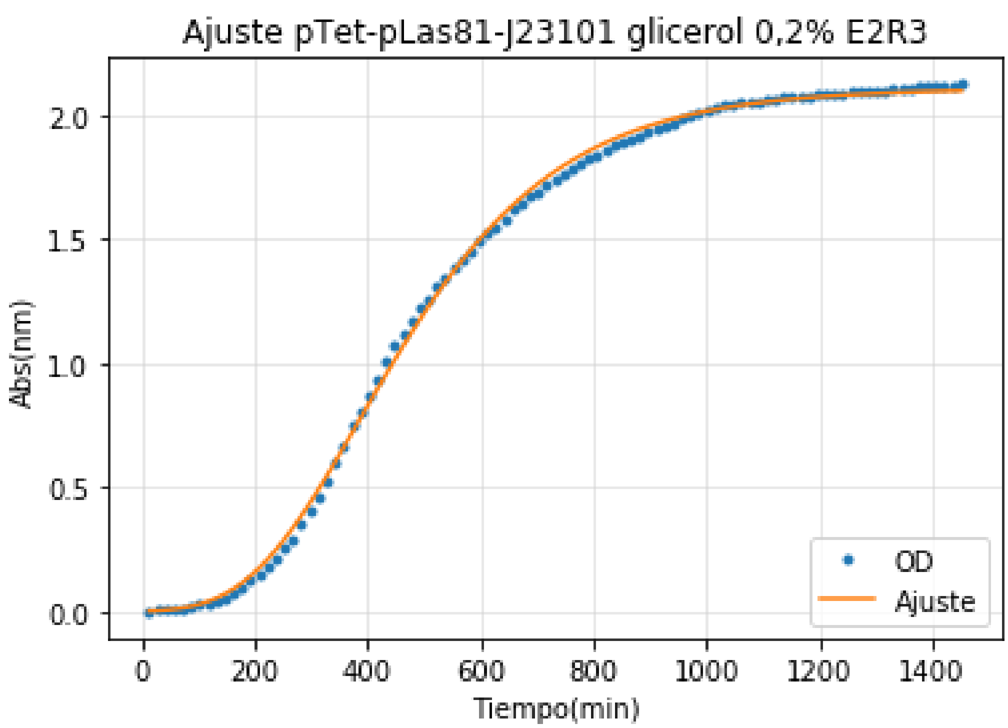





In [48]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-pLas-std glicerol rep 3
y30= np.log(od1527g3)-np.log(np.min(od1527g3))
print('Min OD = %e'%((np.min(od1527g3))))
evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste pTet-pLas81-J23101 glicerol 0,2%
A30= 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=2.5*t230
print('ext=%e'%extdp30)
ttot30=tm30+extdp30
print('Tfinal=%e'%ttot30)

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

```python
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')
```

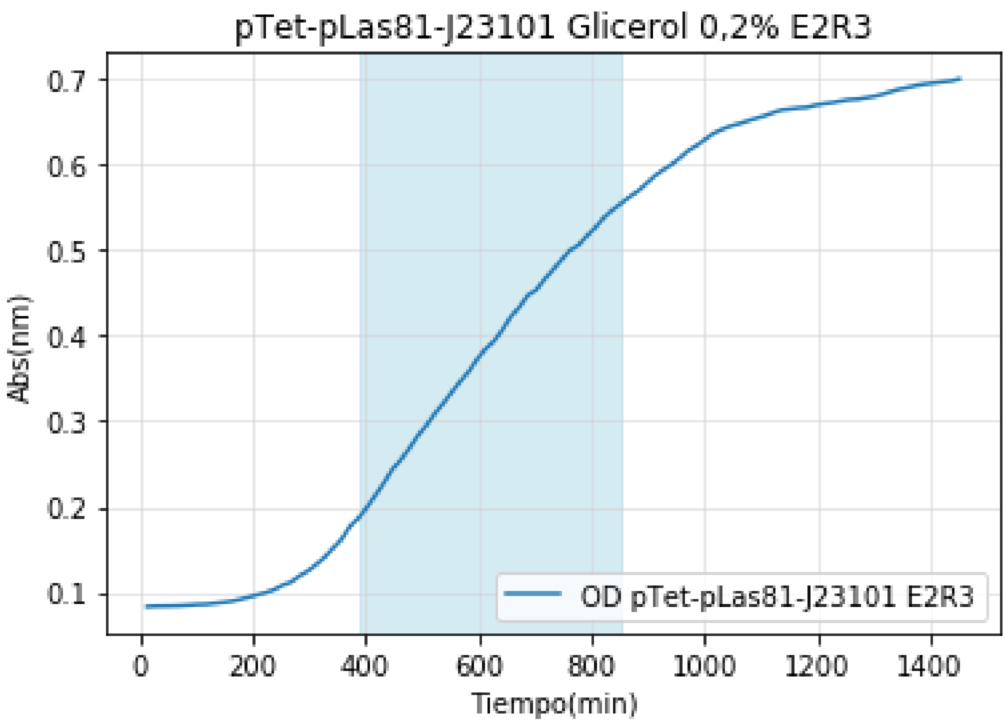
```
Min OD = 8.350000e-02
[  2.10762223e+00   3.94236767e-03   1.88946108e+02]
```

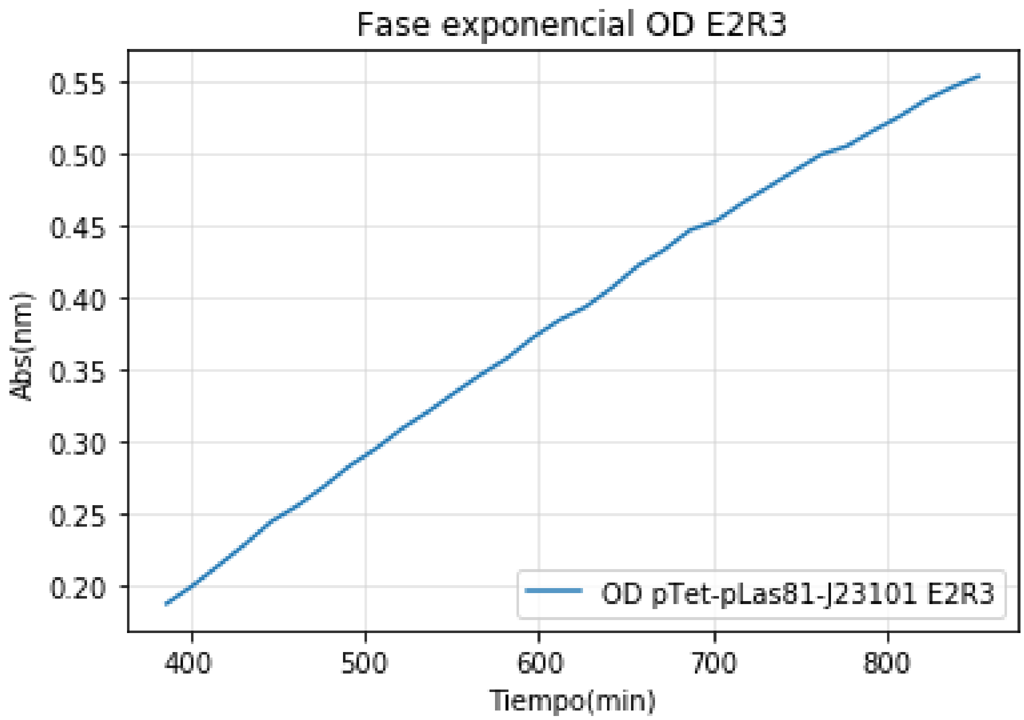



```
A=2.107622e+00
um=3.942368e-03
```

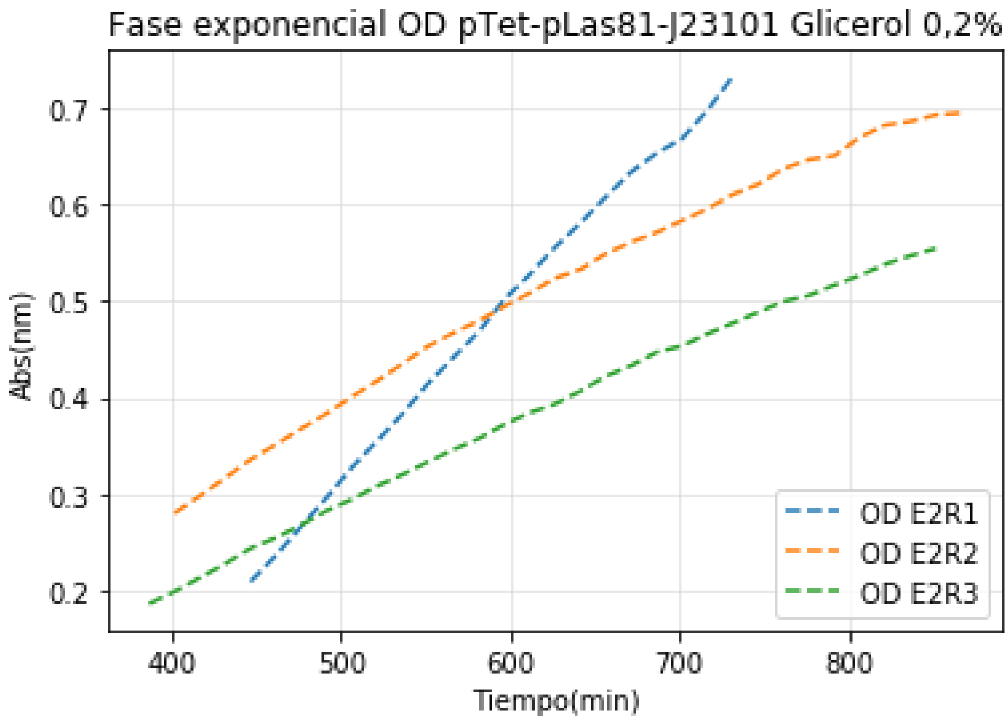
```
l=1.889461e+02
Tm=3.856175e+02
doubpe=1.758200e+02
ext=4.395501e+02
Tfinal=8.251675e+02
```

```
Out[48]: <matplotlib.legend.Legend at 0x267ee7adf28>
```



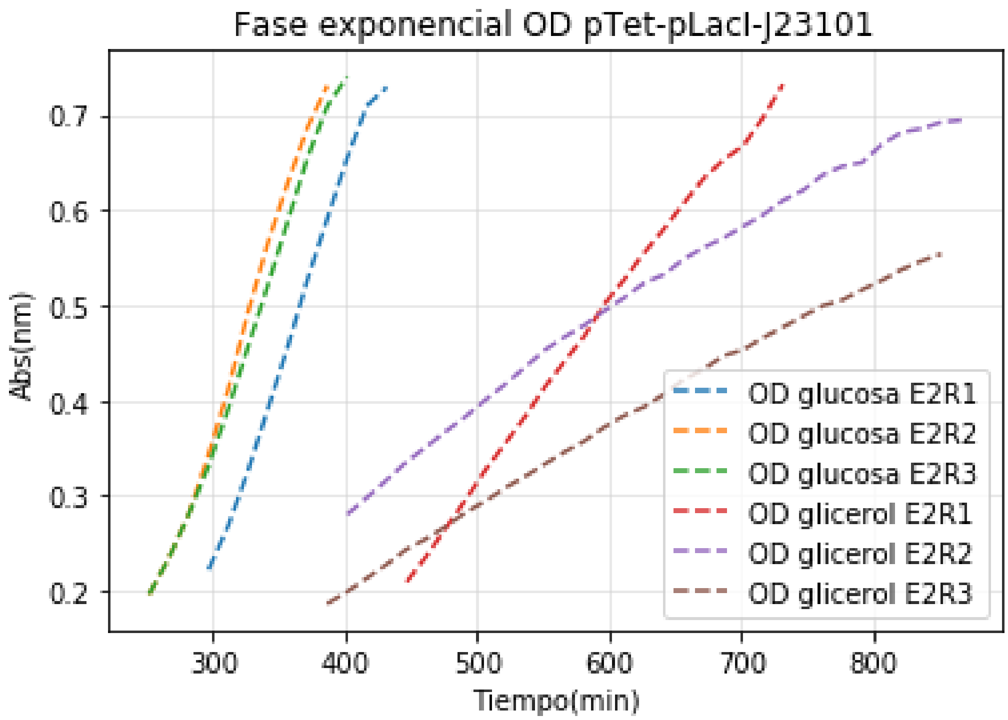



```
In [49]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-pLas81-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[29:49],od1527g1[29:49],'--',label='OD E2R1')
         plt.plot(tt[26:58],od1527g2[26:58],'--',label='OD 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[49]: <matplotlib.legend.Legend at 0x267ecc09ba8>
```




In [50]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pTet-pLacI-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[50]:
```
<matplotlib.legend.Legend at 0x267eccdcf98>
```




In [51]:
```
#Selección de datos en arrays, según lo determinado
#controles glucosa
o1=odcg1[19:30]
c1=cfpcg1[19:30]
r1=rfpcg1[19:30]
y1=yfpcg1[19:30]

o2=odcg2[19:30]
c2=cfpcg2[19:30]
r2=rfpcg2[19:30]
y2=yfpcg2[19:30]

o3=odcg3[15:27]
c3=cfpcg3[15:27]
r3=rfpcg3[15:27]
y3=yfpcg3[15:27]

#controles glicerol
o4=odcgl1[24:45]
c4=cfpcgl1[24:45]
r4=rfpcgl1[24:45]
y4=yfpcgl1[24:45]

o5=odcgl2[23:46]
```

```
c5=cfpcgl2[23:46]
r5=rfpcgl2[23:46]
y5=yfpcgl2[23:46]

o6=odcgl3[25:57]
c6=cfpcgl3[25:57]
r6=rfpcgl3[25:57]
y6=yfpcgl3[25:57]

#ptet-std-std glucosa
o7=od15211[19:30]
c7=cfp15211[19:30]
r7=rfp15211[19:30]
y7=yfp15211[19:30]

o8=od15212[19:32]
c8=cfp15212[19:32]
r8=rfp15212[19:32]
y8=yfp15212[19:32]

o9=od15213[16:28]
c9=cfp15213[16:28]
r9=rfp15213[16:28]
y9=yfp15213[16:28]

#ptet-std-std glicerol
o10=od1521g1[27:48]
c10=cfp1521g1[27:48]
r10=rfp1521g1[27:48]
y10=yfp1521g1[27:48]

o11=od1521g2[28:53]
c11=cfp1521g2[28:53]
r11=rfp1521g2[28:53]
y11=yfp1521g2[28:53]

o12=od1521g3[27:53]
c12=cfp1521g3[27:53]
r12=rfp1521g3[27:53]
y12=yfp1521g3[27:53]

#ptet-107-std glucosa
o13=od15231[21:32]
c13=cfp15231[21:32]
r13=rfp15231[21:32]
y13=yfp15231[21:32]

o14=od15232[17:27]
```

```
c14=cfp15232[17:27]
r14=rfp15232[17:27]
y14=yfp15232[17:27]

o15=od15233[17:29]
c15=cfp15233[17:29]
r15=rfp15233[17:29]
y15=yfp15233[17:29]

#ptet-107-std glicerol
o16=od1523g1[30:49]
c16=cfp1523g1[30:49]
r16=rfp1523g1[30:49]
y16=yfp1523g1[30:49]

o17=od1523g2[27:54]
c17=cfp1523g2[27:54]
r17=rfp1523g2[27:54]
y17=yfp1523g2[27:54]

o18=od1523g3[28:60]
c18=cfp1523g3[28:60]
r18=rfp1523g3[28:60]
y18=yfp1523g3[28:60]

#ptet-pLac-std glucosa
o19=od15261[19:29]
c19=cfp15261[19:29]
r19=rfp15261[19:29]
y19=yfp15261[19:29]

o20=od15262[18:29]
c20=cfp15262[18:29]
r20=rfp15262[18:29]
y20=yfp15262[18:29]

o21=od15263[16:28]
c21=cfp15263[16:28]
r21=rfp15263[16:28]
y21=yfp15263[16:28]

#ptet-pLac-std glicerol
o22=od1526g1[27:47]
c22=cfp1526g1[27:47]
r22=rfp1526g1[27:47]
y22=yfp1526g1[27:47]

o23=od1526g2[24:49]
```

```
c23=cfp1526g2[24:49]
r23=rfp1526g2[24:49]
y23=yfp1526g2[24:49]

o24=od1526g3[27:59]
c24=cfp1526g3[27:59]
r24=rfp1526g3[27:59]
y24=yfp1526g3[27:59]

#ptet-pLas-std glucosa
o25=od15271[19:29]
c25=cfp15271[19:29]
r25=rfp15271[19:29]
y25=yfp15271[19:29]

o26=od15272[16:26]
c26=cfp15272[16:26]
r26=rfp15272[16:26]
y26=yfp15272[16:26]

o27=od15273[16:26]
c27=cfp15273[16:26]
r27=rfp15273[16:26]
y27=yfp15273[16:26]

#ptet-pLas-std glicerol
o28=od1527g1[29:49]
c28=cfp1527g1[29:49]
r28=rfp1527g1[29:49]
y28=yfp1527g1[29:49]

o29=od1527g2[26:58]
c29=cfp1527g2[26:58]
r29=rfp1527g2[26:58]
y29=yfp1527g2[26:58]

o30=od1527g3[25:57]
c30=cfp1527g3[25:57]
r30=rfp1527g3[25:57]
y30=yfp1527g3[25:57]
```

```
In [52]: #regresion lineal de replicas
#Controles glucosa
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
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

#ptet-std-std glucosa
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
```

```python
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

#ptet-std-std glicerol
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

#ptet-107-std glucosa
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

#ptet-107-std glicerol
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)
slopey17=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)
slopey18=slope

#ptet-pLacI-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)
slopey19=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)
slopey20=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)
```

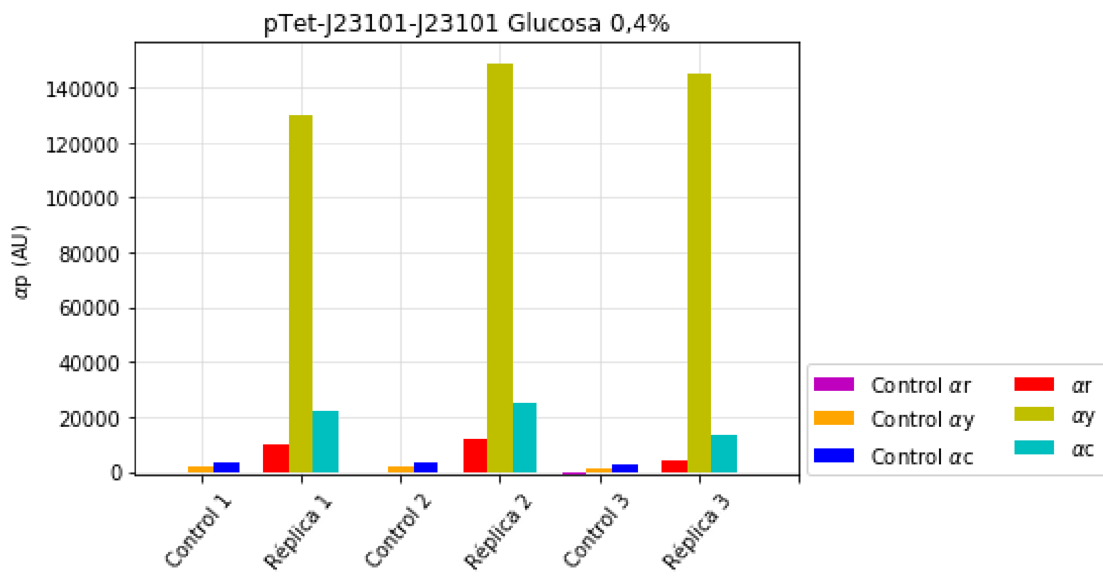
```
sloper21=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o21,y21)
slopey21=slope

#ptet-LacI-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)
slopey22=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)
slopey23=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o24,c24)
slopec24=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o24,r24)
sloper24=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o24,y24)
slopey24=slope

#ptet-pLas81-std glucosa
slope, intercept, r_value, p_value,std_err=stats.linregress(o25,c25)
slopec25=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o25,r25)
sloper25=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o25,y25)
slopey25=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o26,c26)
slopec26=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o26,r26)
sloper26=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o26,y26)
slopey26=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o27,c27)
slopec27=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o27,r27)
sloper27=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o27,y27)
slopey27=slope
```

```
#ptet-pLas81-std glicerol
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,c28)
slopec28=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,r28)
sloper28=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,y28)
slopey28=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o29,c29)
slopec29=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o29,r29)
sloper29=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o29,y29)
slopey29=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)
slopey30=slope
```

In [53]:
```
pendientesc=[slopec1,slopec2,slopec3,slopec4,slopec5,slopec6,slopec7,slopec8,slopec9,sl
pendientesr=[sloper1,sloper2,sloper3,sloper4,sloper5,sloper6,sloper7,sloper8,sloper9,sl
pendientesy=[slopey1,slopey2,slopey3,slopey4,slopey5,slopey6,slopey7,slopey8,slopey9,sl

#Grafico pendientes ptet-std-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\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,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],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,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],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.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)
```

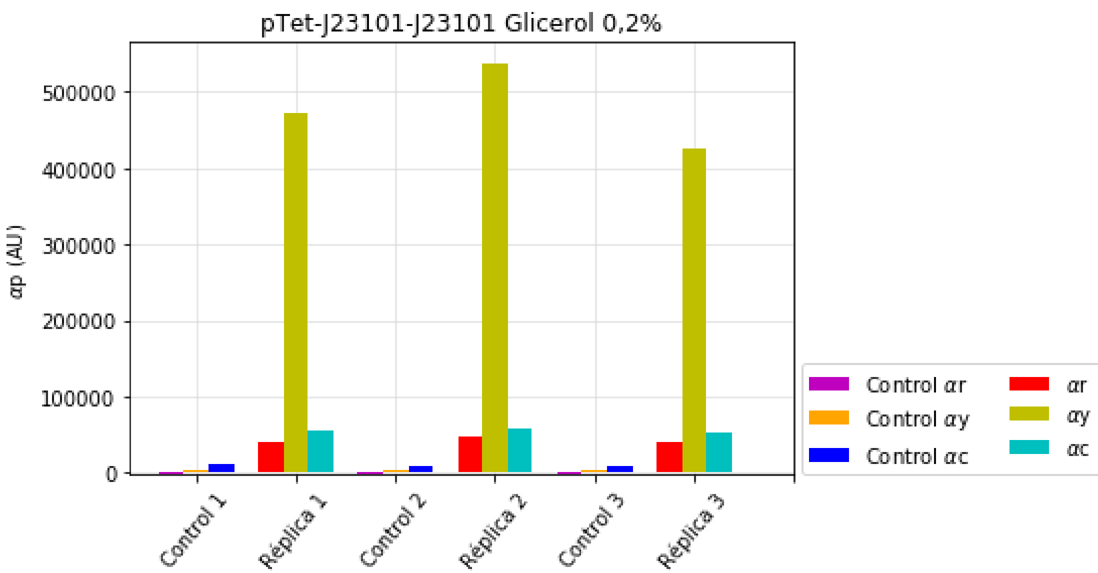
Out[53]: <matplotlib.legend.Legend at 0x267ed5157b8>




```
In [54]: #Grafico pendientes ptet-std-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[9],color='r',width=0.25,label=r'$\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,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[4],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,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[5],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)
```

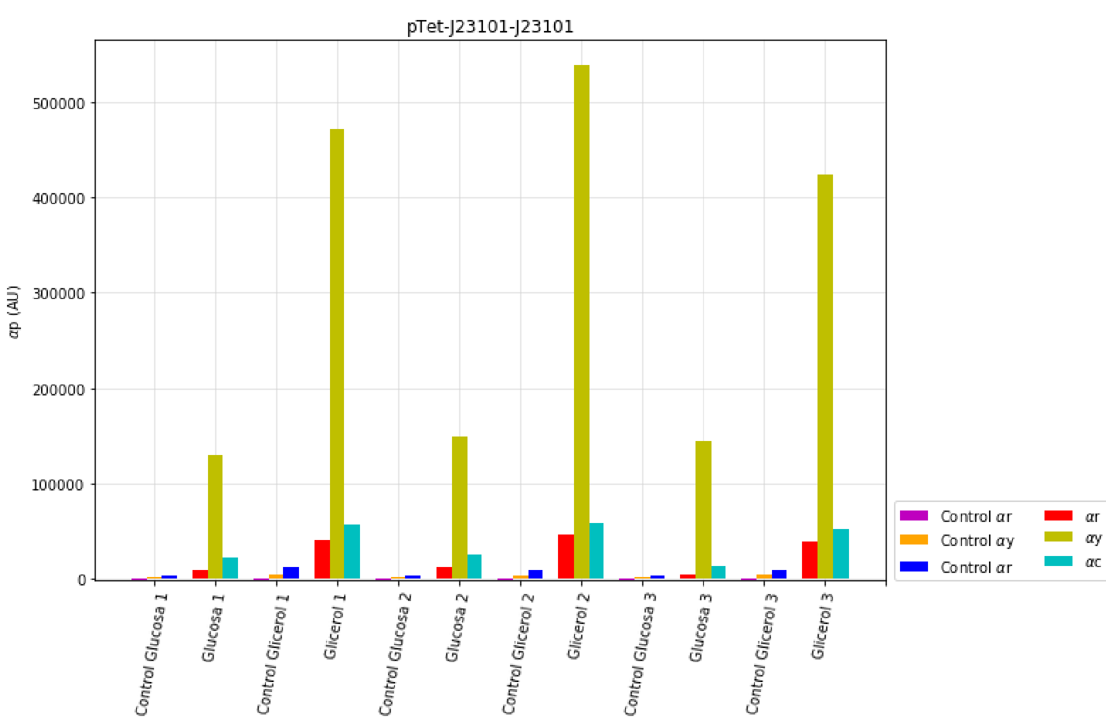
Out[54]: <matplotlib.legend.Legend at 0x267eecf5ac8>




```
In [55]: #Grafico pendientes ptet-std-std
X = np.arange(13)
plt.figure(figsize=(10,7))
plt.title('pTet-J23101-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\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,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[9],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[9],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[9],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
```

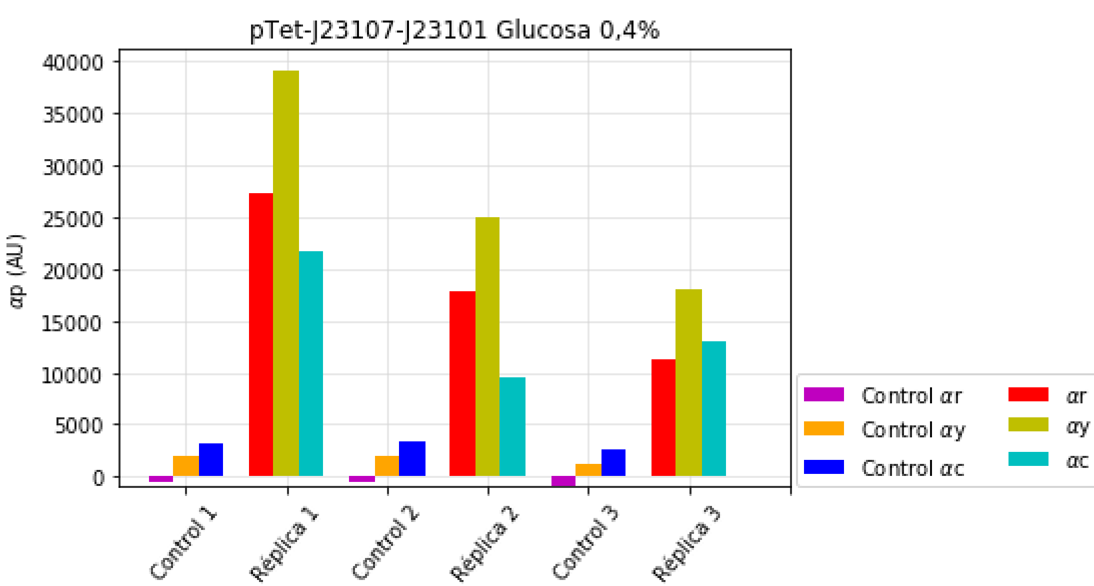
```
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[8],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,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[5],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",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[55]: <matplotlib.legend.Legend at 0x267eeea2940>



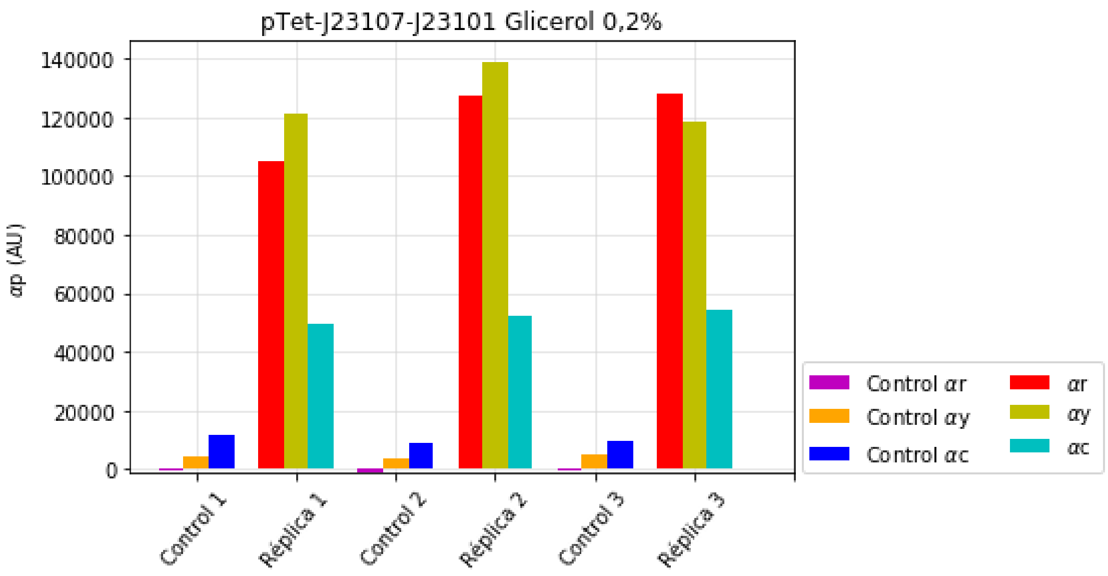
In [56]:
```python
#Grafico pendientes ptet-107-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pTet-J23107-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\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$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[12],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],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,pendientesc[13],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],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.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[56]: <matplotlib.legend.Legend at 0x267ee784e80>



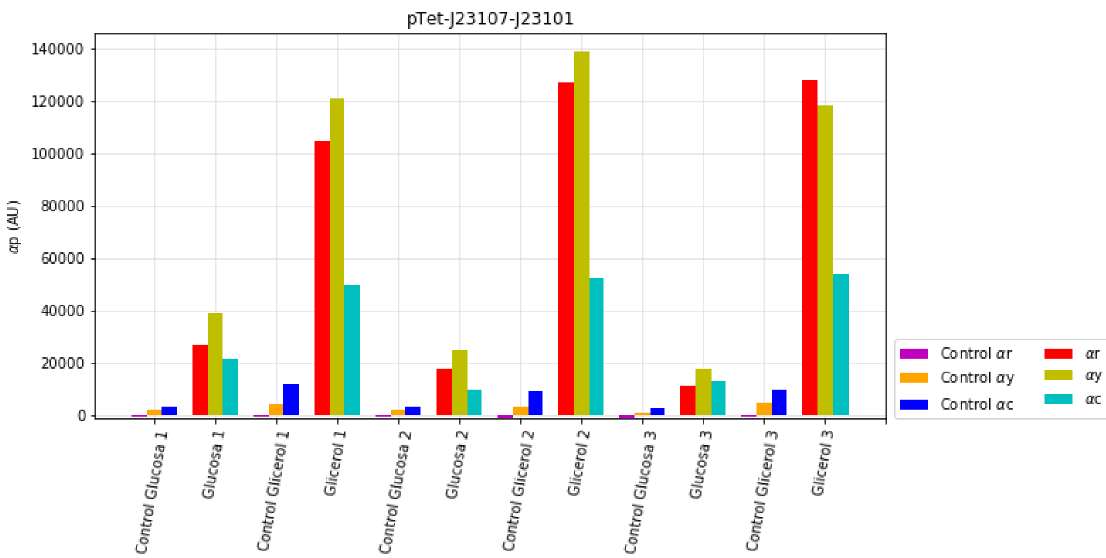
In [57]:
```
#Grafico pendientes ptet-107-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pTet-J23107-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[15],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[15],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[15],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[4],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,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[5],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,pendientesc[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[57]: <matplotlib.legend.Legend at 0x267ef281470>

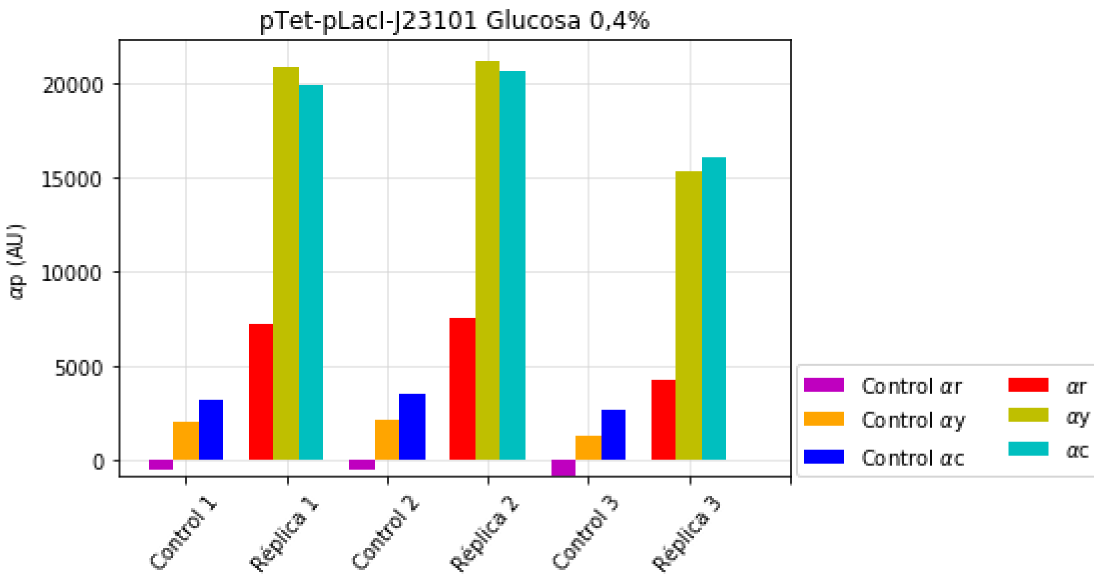


In [58]:
```
#Grafico pendientes ptet-std-std
X = np.arange(13)
plt.figure(figsize=(10,5))
plt.title('pTet-J23107-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\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$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[12],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[15],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[14],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,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[5],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",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

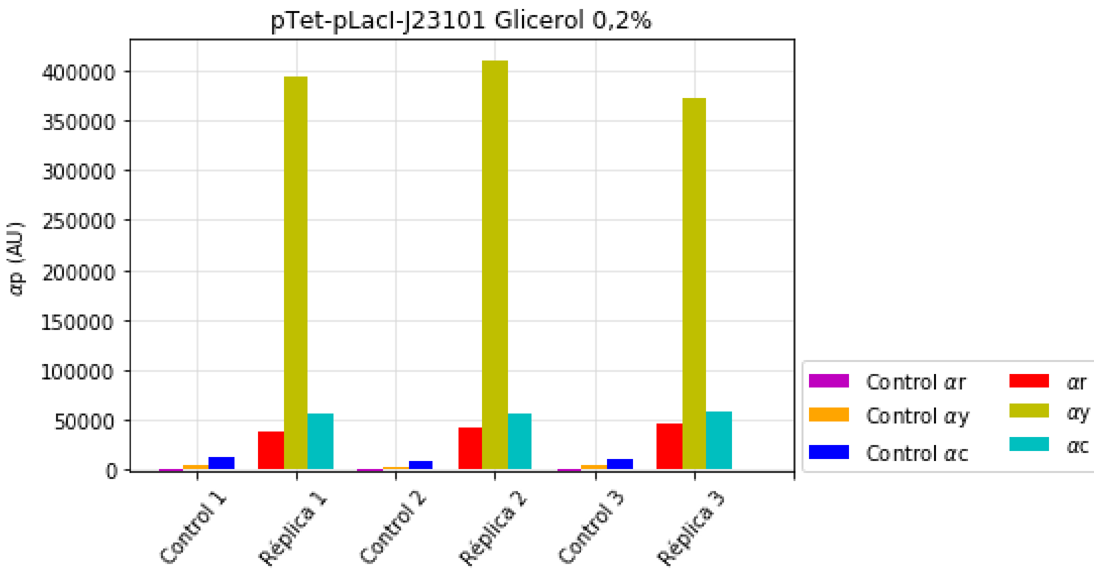
Out[58]:
```
<matplotlib.legend.Legend at 0x267ef34a080>
```




```
In [59]: #Grafico pendientes ptet-plac-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('pTet-pLacI-J23101 Glucosa 0,4%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[18],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[18],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[18],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[20],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[20],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[20],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[59]: <matplotlib.legend.Legend at 0x267ef6bf780>
```



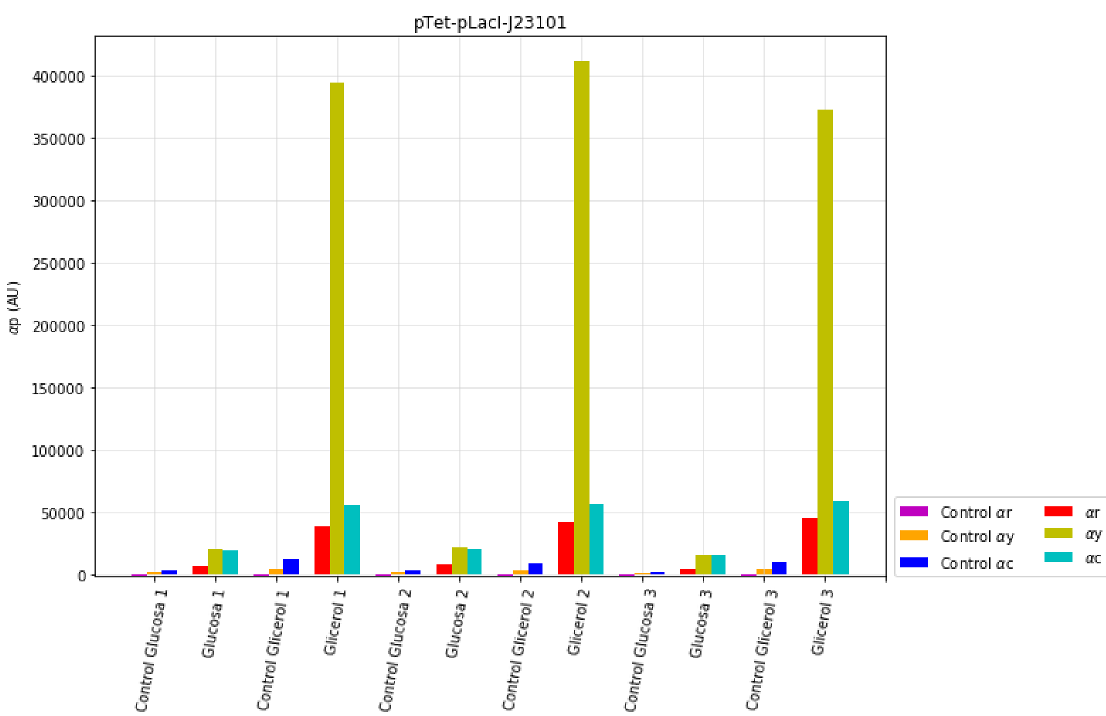

```
In [60]: #Grafico pendientes ptet-plac-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pTet-pLacI-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[21],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[21],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[21],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[22],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[23],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[23],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[23],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[60]: <matplotlib.legend.Legend at 0x267ef7e8f60>
```




```
In [61]: #Grafico pendientes ptet-lac-std
         X = np.arange(13)
         plt.figure(figsize=(10,7))
         plt.title('pTet-pLacI-J23101')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[18],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[18],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[18],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[21],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[21],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[21],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[7]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
         plt.bar(X[7]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
```

```
plt.bar(X[7]+0.25,pendientesc[22],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[20],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[20],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[20],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,pendientesy[5],color='orange',width=0.25)
plt.bar(X[10]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[23],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[23],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[23],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

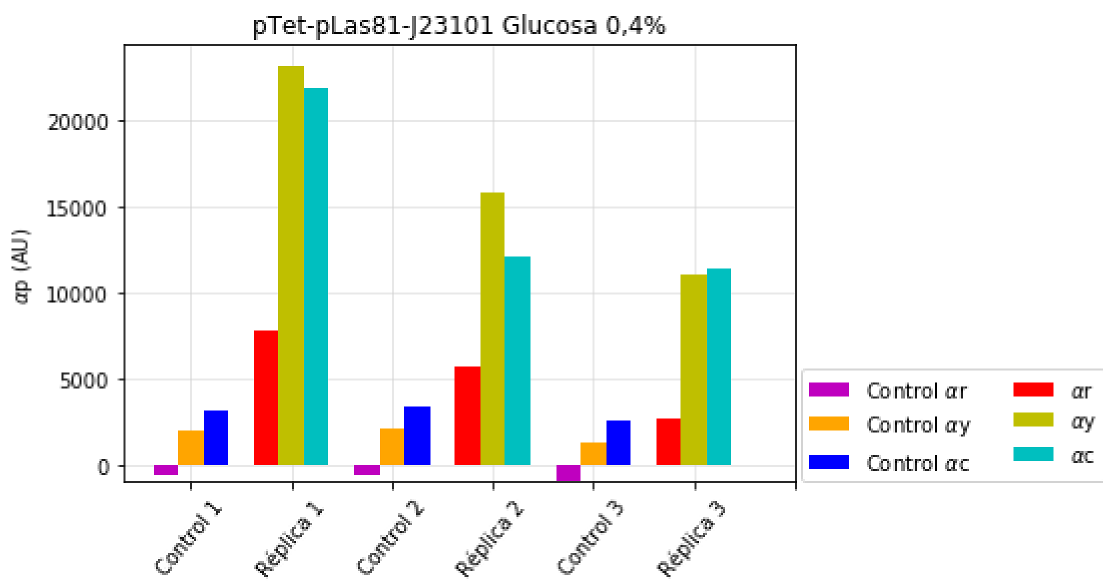
```
Out[61]: <matplotlib.legend.Legend at 0x267ef499630>
```




```
In [62]: #Grafico pendientes ptet-pLas-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,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[24],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[24],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[26],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)
```

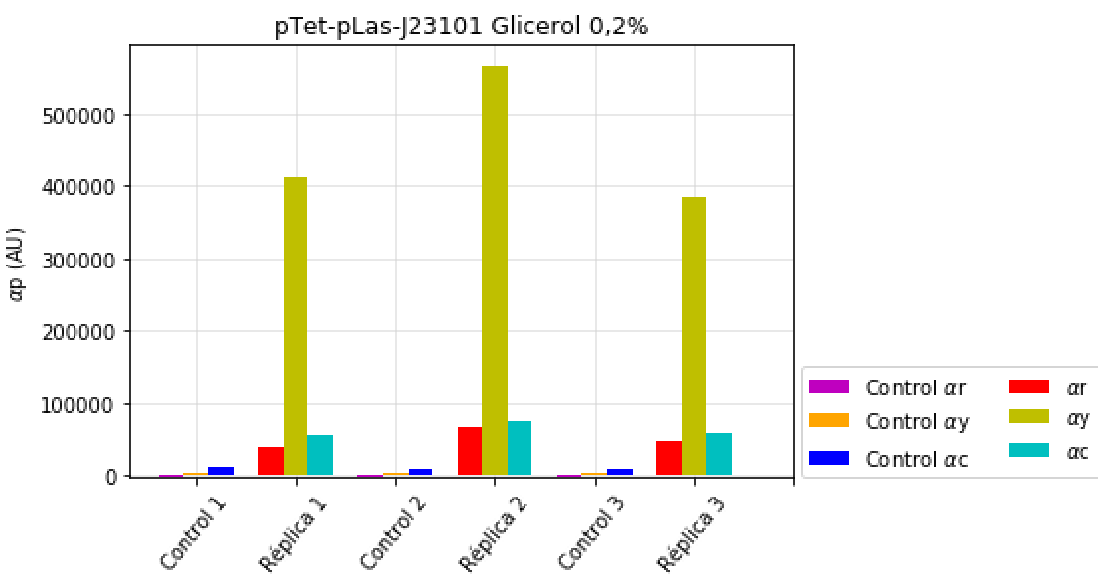
```
Out[62]: <matplotlib.legend.Legend at 0x267eee757b8>
```




```
In [63]: #Grafico pendientes ptet-plas-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pTet-pLas-J23101 Glicerol 0,2%')
```

```
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[27],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[27],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[27],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[29],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)
```

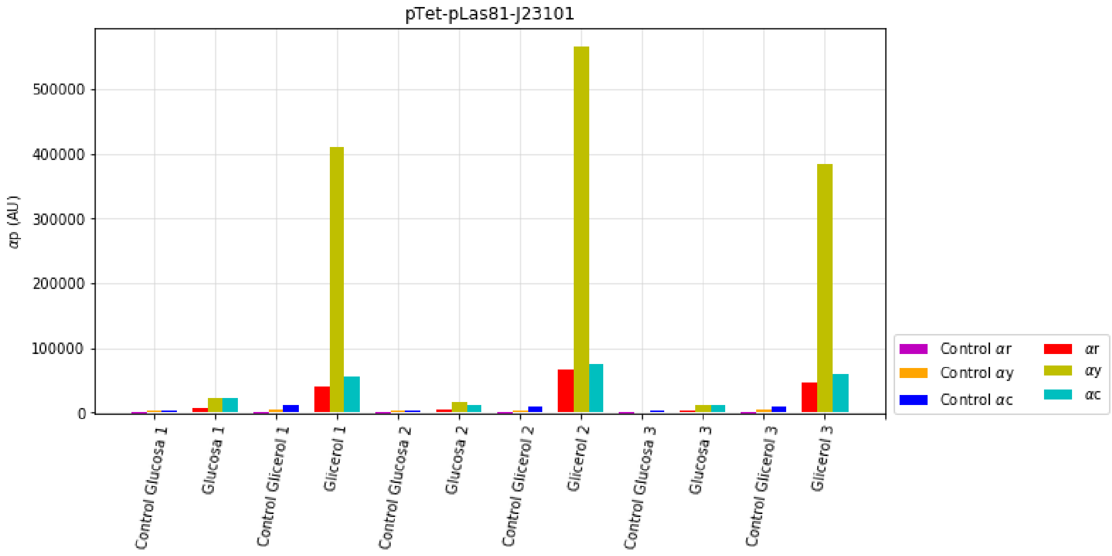
Out[63]: <matplotlib.legend.Legend at 0x267efc7fdd8>




```
In [64]: #Grafico pendientes ptet-plas-std
X = np.arange(13)
plt.figure(figsize=(10,5))
```

```
plt.title('pTet-pLas81-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[24],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[24],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[27],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[27],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[27],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[26],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,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[29],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

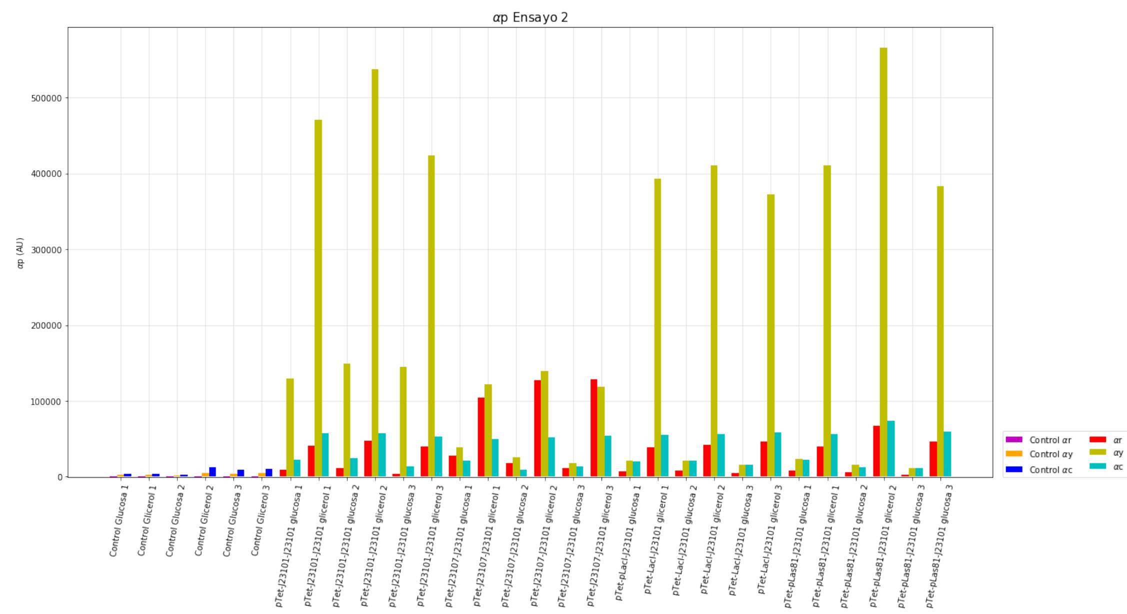
```
Out[64]: <matplotlib.legend.Legend at 0x267f0df9a20>
```




```
In [141]: #Grafico pendientes todo
          X = np.arange(30)
          plt.figure(figsize=(20,10))
          plt.title(r'$\alpha$p Ensayo 2', fontsize=15.0)
          plt.ylabel(r'$\alpha$p (AU)')
          plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
          plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\al
          plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c
          plt.bar(X[1]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
          plt.bar(X[1]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
          plt.bar(X[1]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
          plt.bar(X[2]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
          plt.bar(X[2]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
          plt.bar(X[2]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
          plt.bar(X[3]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
          plt.bar(X[3]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
          plt.bar(X[3]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
          plt.bar(X[4]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
          plt.bar(X[4]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
          plt.bar(X[4]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
          plt.bar(X[5]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
          plt.bar(X[5]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
          plt.bar(X[5]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
          plt.bar(X[6]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
          plt.bar(X[6]+0.00,pendientesy[6],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
          plt.bar(X[6]+0.25,pendientesc[6],color='c',width=0.25,label=r'$\alpha$c',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,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[7],color='c',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,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[8],color='c',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.bar(X[12]-0.25,pendientesr[12],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,pendientesy[12],color='y',width=0.25,zorder=3)
plt.bar(X[12]+0.25,pendientesc[12],color='c',width=0.25,zorder=3)
plt.bar(X[13]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
plt.bar(X[13]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
plt.bar(X[13]+0.25,pendientesc[15],color='c',width=0.25,zorder=3)
plt.bar(X[14]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[14]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
plt.bar(X[15]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[15]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[16]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[16]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
plt.bar(X[17]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
plt.bar(X[17]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
plt.bar(X[17]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
plt.bar(X[18]-0.25,pendientesr[18],color='r',width=0.25,zorder=3)
plt.bar(X[18]+0.00,pendientesy[18],color='y',width=0.25,zorder=3)
plt.bar(X[18]+0.25,pendientesc[18],color='c',width=0.25,zorder=3)
plt.bar(X[19]-0.25,pendientesr[21],color='r',width=0.25,zorder=3)
plt.bar(X[19]+0.00,pendientesy[21],color='y',width=0.25,zorder=3)
plt.bar(X[19]+0.25,pendientesc[21],color='c',width=0.25,zorder=3)
plt.bar(X[20]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
plt.bar(X[20]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
plt.bar(X[20]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
plt.bar(X[21]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
plt.bar(X[21]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
plt.bar(X[21]+0.25,pendientesc[22],color='c',width=0.25,zorder=3)
plt.bar(X[22]-0.25,pendientesr[20],color='r',width=0.25,zorder=3)
plt.bar(X[22]+0.00,pendientesy[20],color='y',width=0.25,zorder=3)
plt.bar(X[22]+0.25,pendientesc[20],color='c',width=0.25,zorder=3)
plt.bar(X[23]-0.25,pendientesr[23],color='r',width=0.25,zorder=3)
plt.bar(X[23]+0.00,pendientesy[23],color='y',width=0.25,zorder=3)
plt.bar(X[23]+0.25,pendientesc[23],color='c',width=0.25,zorder=3)
```

```
plt.bar(X[24]-0.25,pendientesr[24],color='r',width=0.25,zorder=3)
plt.bar(X[24]+0.00,pendientesy[24],color='y',width=0.25,zorder=3)
plt.bar(X[24]+0.25,pendientesc[24],color='c',width=0.25,zorder=3)
plt.bar(X[25]-0.25,pendientesr[27],color='r',width=0.25,zorder=3)
plt.bar(X[25]+0.00,pendientesy[27],color='y',width=0.25,zorder=3)
plt.bar(X[25]+0.25,pendientesc[27],color='c',width=0.25,zorder=3)
plt.bar(X[26]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[26]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[26]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[27]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[27]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[27]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[28]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[28]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[28]+0.25,pendientesc[26],color='c',width=0.25,zorder=3)
plt.bar(X[29]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[29]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[29]+0.25,pendientesc[29],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control G
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-p ensayo 2.png', dpi=300, facecolor='w', edgecolor='w',bbox
```

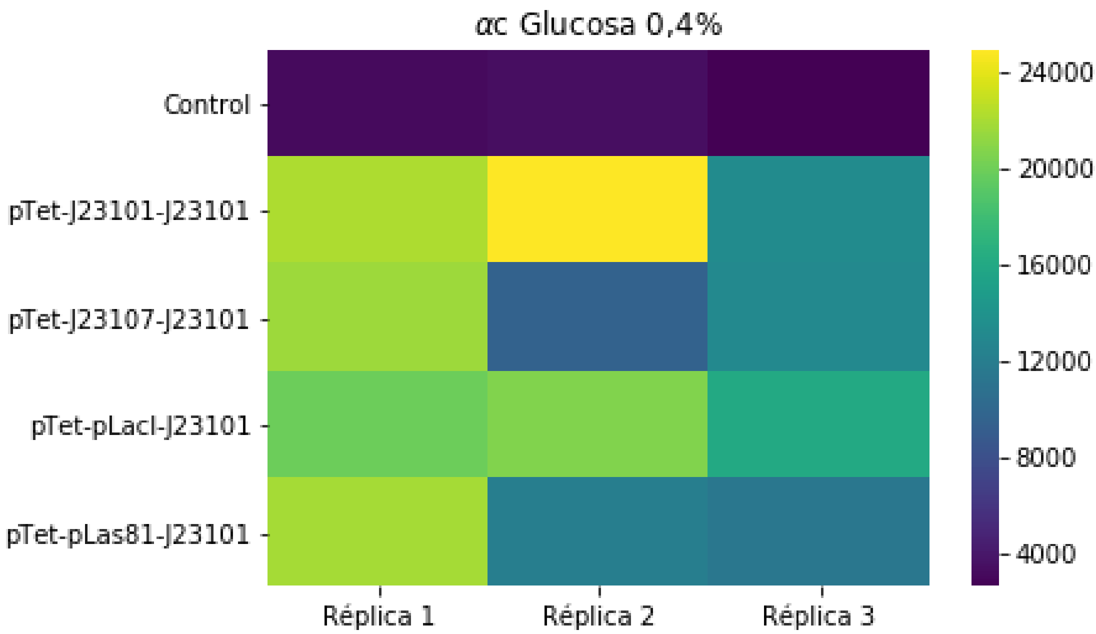


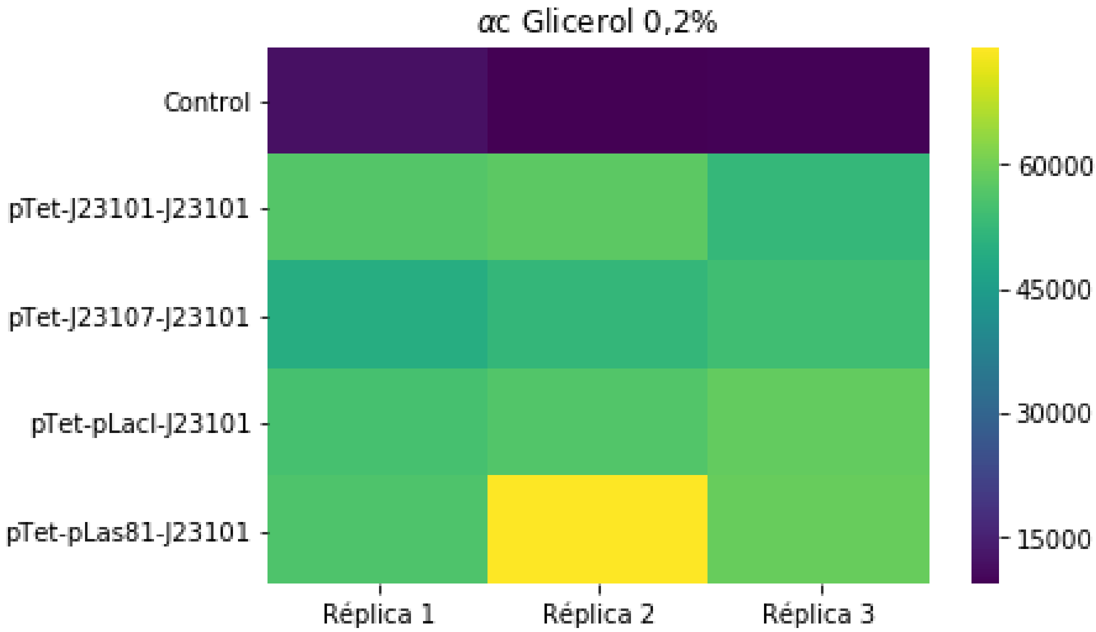

In [66]:
```
cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15],
cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec1
rglu=[[sloper1,sloper2,sloper3],[sloper7,sloper8,sloper9],[sloper13,sloper14,sloper15],
rgli=[[sloper4,sloper5,sloper6],[sloper10,sloper11,sloper12],[sloper16,sloper17,sloper1
```

```
yglu=[[slopey1,slopey2,slopey3],[slopey7,slopey8,slopey9],[slopey13,slopey14,slopey15],
ygli=[[slopey4,slopey5,slopey6],[slopey10,slopey11,slopey12],[slopey16,slopey17,slopey1
```

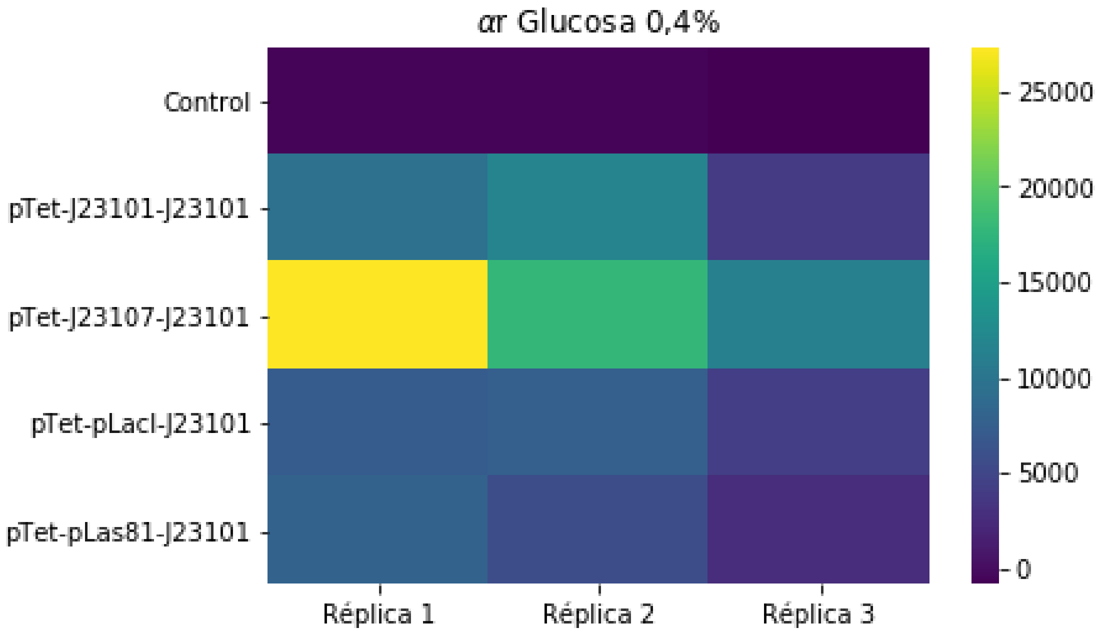
In [67]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

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

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

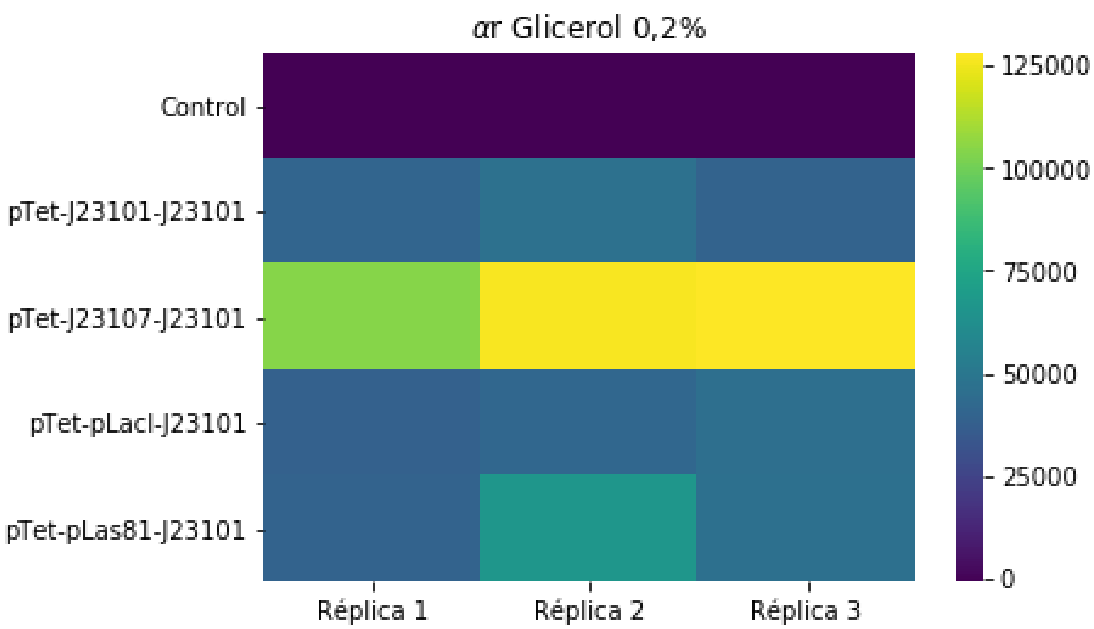
Out[67]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x267faca8f28>
```






```
In [68]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p


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

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

Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x267eeff0e10>
```

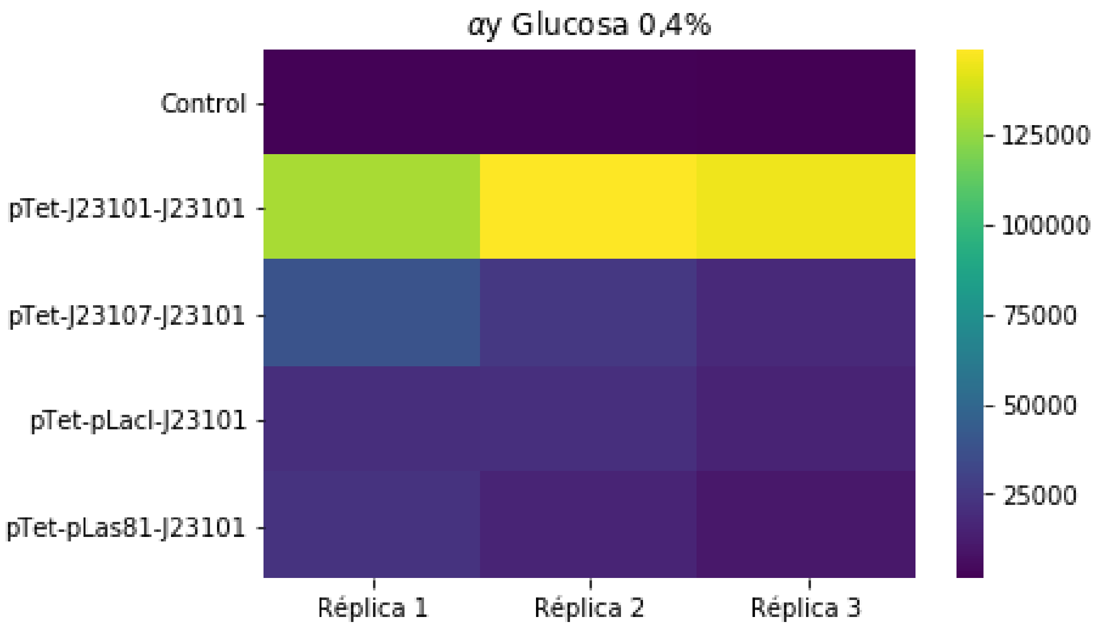


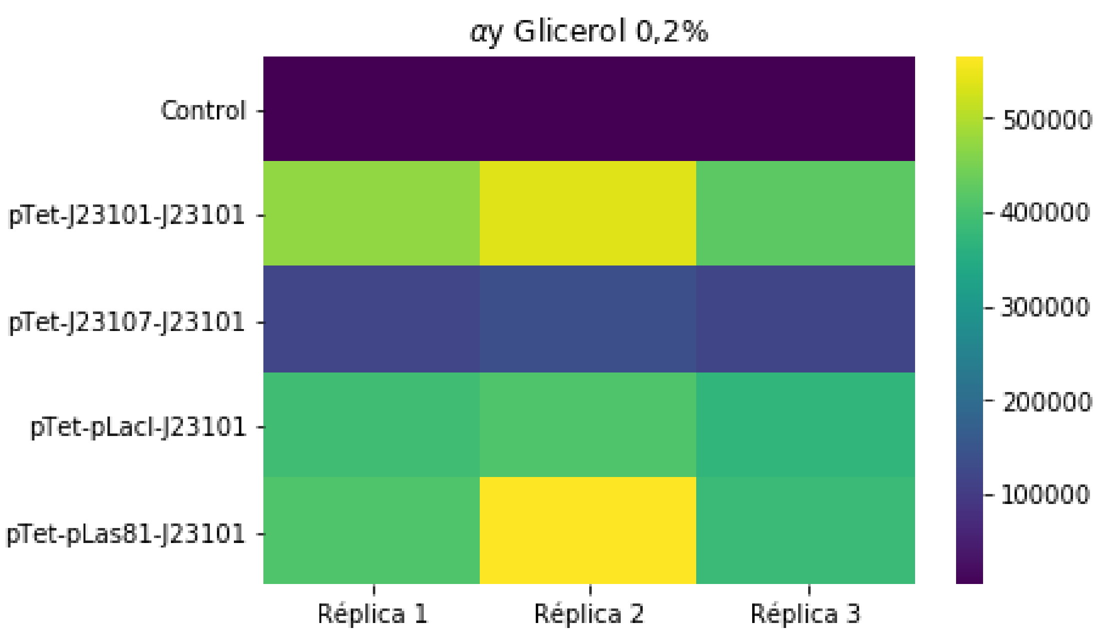




```
In [69]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

         plt.figure()
```

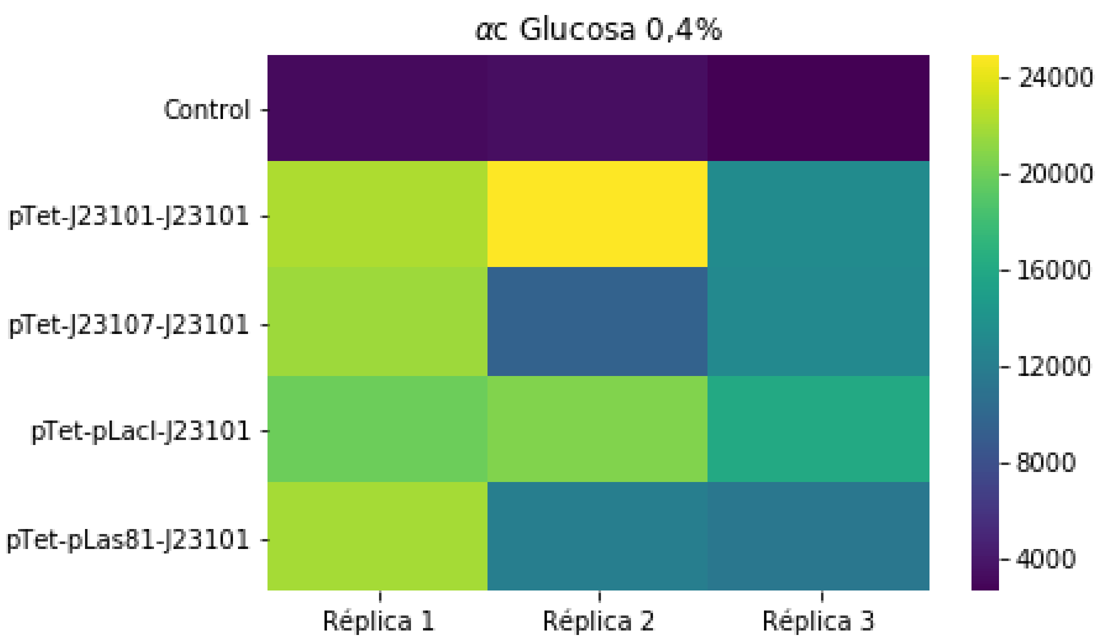
```
    plt.title(r'$\alpha$y Glucosa 0,4%')
    sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

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

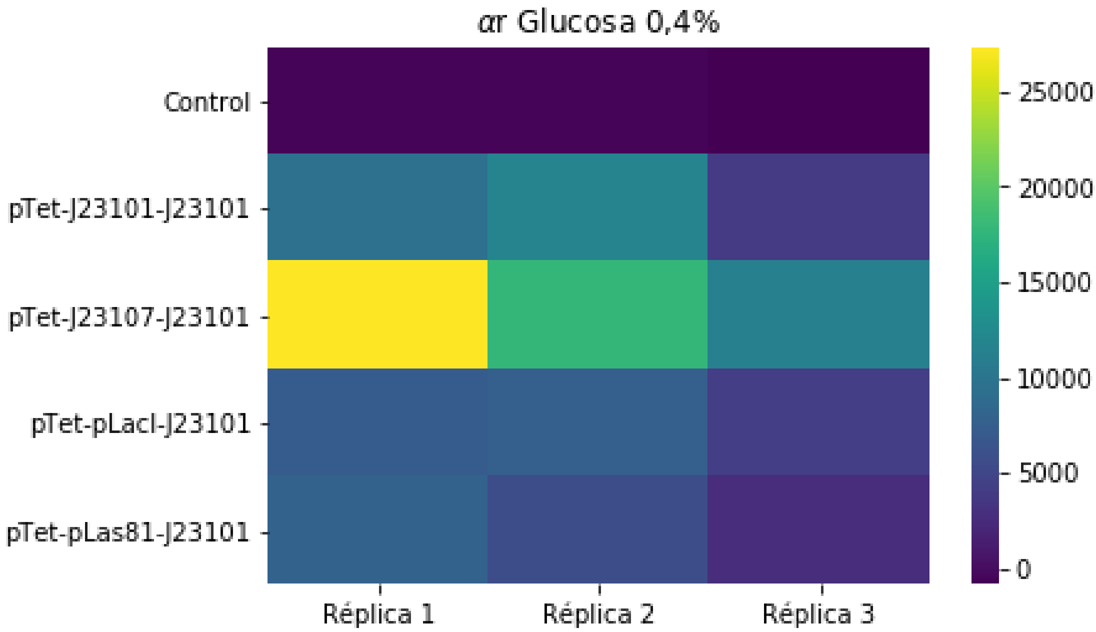
```
Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x267f179f748>
```

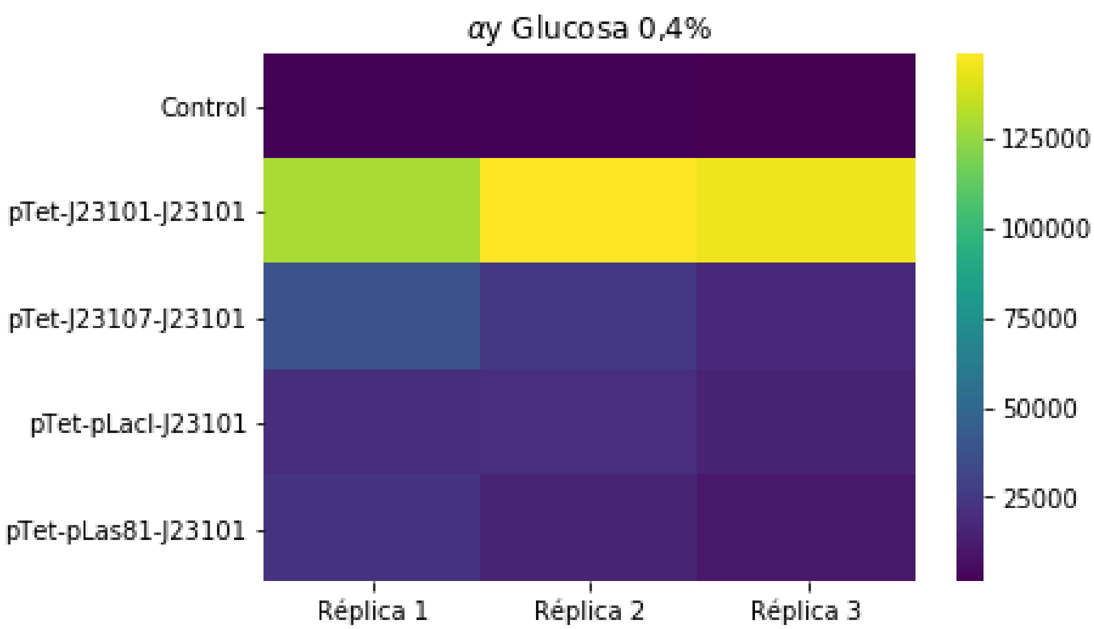


```
In [70]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

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

Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x267f1799518>
```



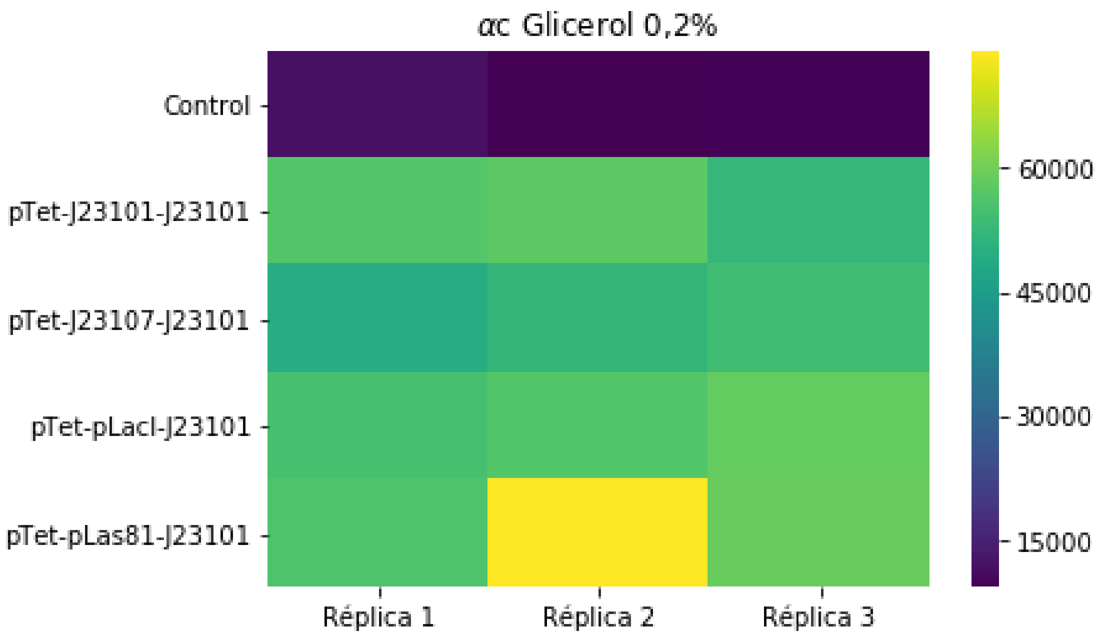


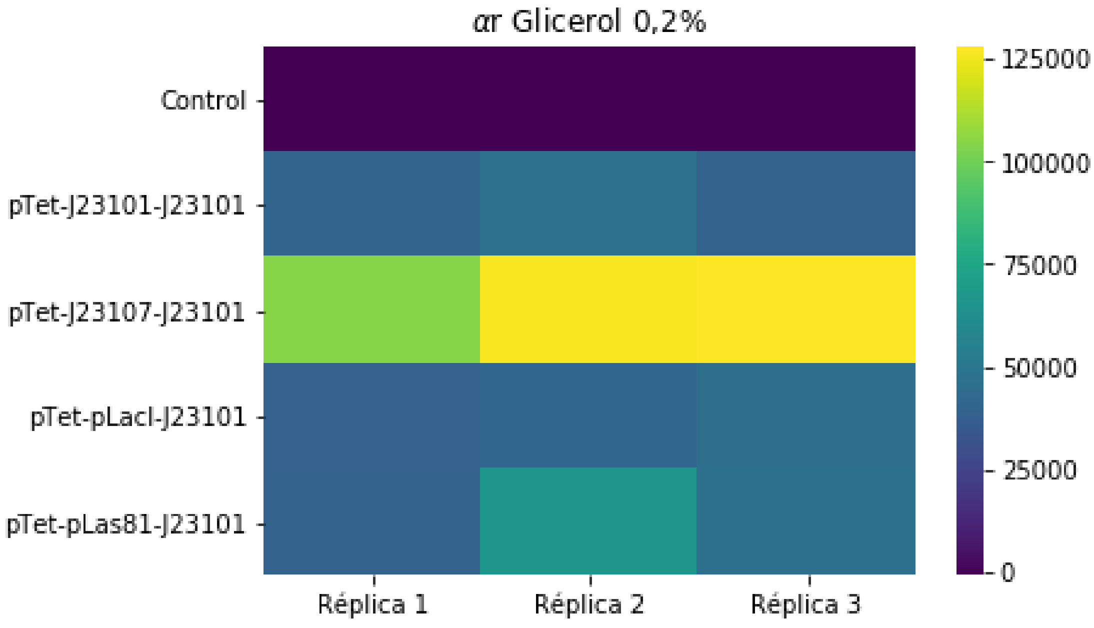



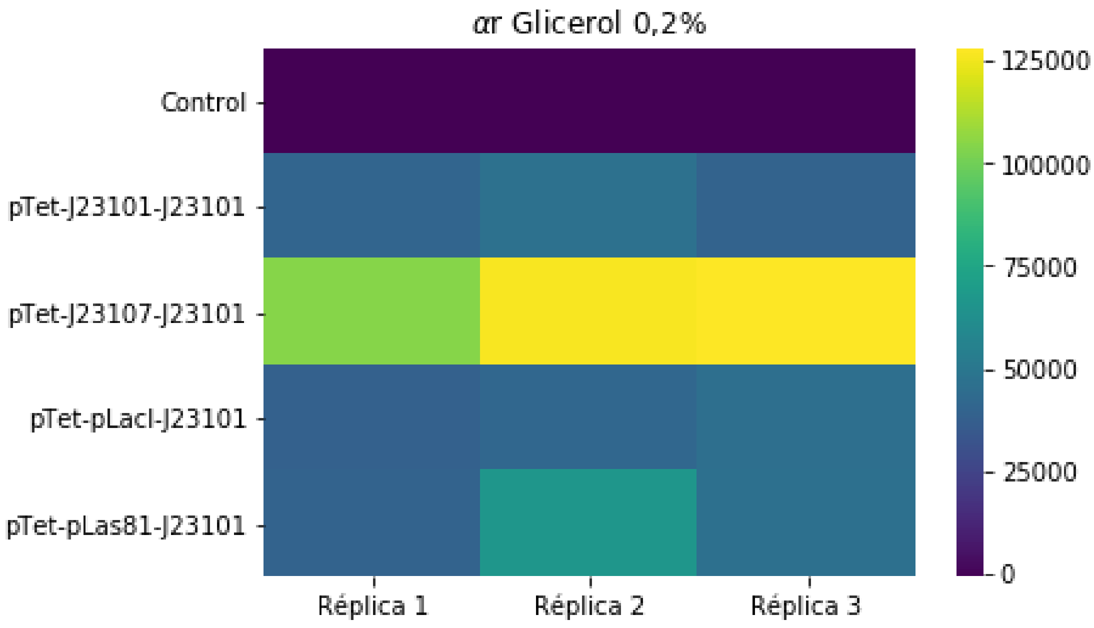

In [71]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

plt.figure()
```

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

```
Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x267f1ba0dd8>
```



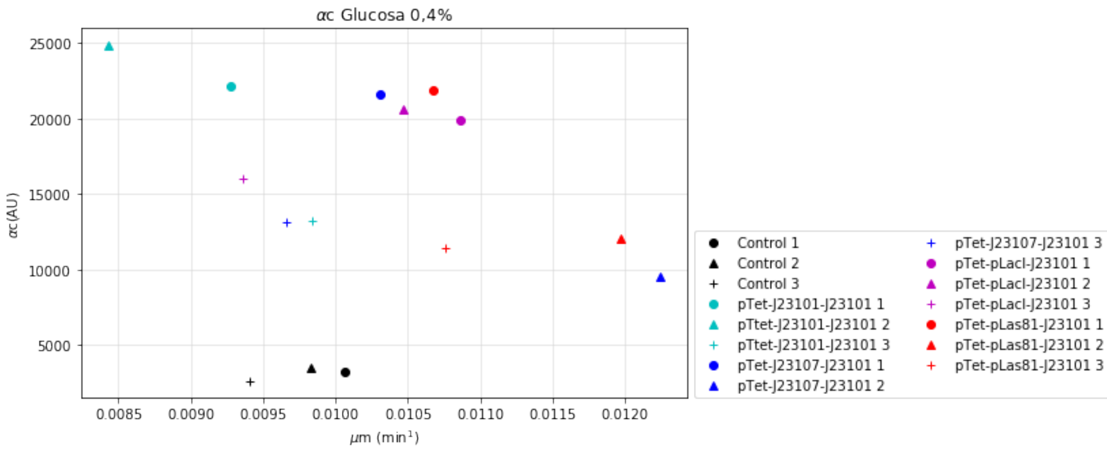


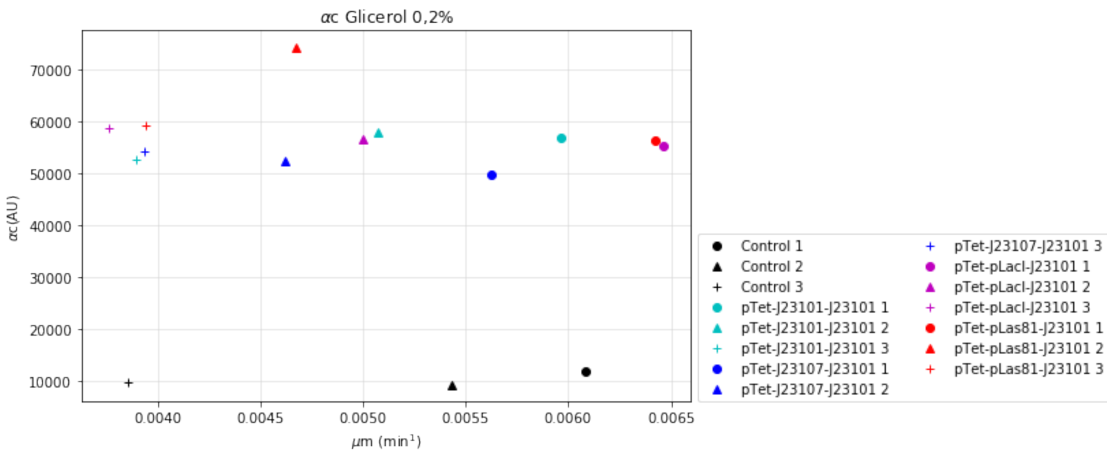




```
In [72]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Glucosa 0,4%')
         plt.xlabel(r'$\mu$m (min$^1$)')
```

```
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um1,slopec1,'ko',label='Control 1')
plt.plot(um2,slopec2,'k^',label='Control 2')
plt.plot(um3,slopec3,'k+',label='Control 3')
plt.plot(um7,slopec7,'co',label='pTet-J23101-J23101 1')
plt.plot(um8,slopec8,'c^',label='pTtet-J23101-J23101 2')
plt.plot(um9,slopec9,'c+',label='pTtet-J23101-J23101 3')
plt.plot(um13,slopec13,'bo',label='pTet-J23107-J23101 1')
plt.plot(um14,slopec14,'b^',label='pTet-J23107-J23101 2')
plt.plot(um15,slopec15,'b+',label='pTet-J23107-J23101 3')
plt.plot(um19,slopec19,'mo',label='pTet-pLacI-J23101 1')
plt.plot(um20,slopec20,'m^',label='pTet-pLacI-J23101 2')
plt.plot(um21,slopec21,'m+',label='pTet-pLacI-J23101 3')
plt.plot(um25,slopec25,'ro',label='pTet-pLas81-J23101 1')
plt.plot(um26,slopec26,'r^',label='pTet-pLas81-J23101 2')
plt.plot(um27,slopec27,'r+',label='pTet-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'$\alpha$c Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um4,slopec4,'ko',label='Control 1')
plt.plot(um5,slopec5,'k^',label='Control 2')
plt.plot(um6,slopec6,'k+',label='Control 3')
plt.plot(um10,slopec10,'co',label='pTet-J23101-J23101 1')
plt.plot(um11,slopec11,'c^',label='pTet-J23101-J23101 2')
plt.plot(um12,slopec12,'c+',label='pTet-J23101-J23101 3')
plt.plot(um16,slopec16,'bo',label='pTet-J23107-J23101 1')
plt.plot(um17,slopec17,'b^',label='pTet-J23107-J23101 2')
plt.plot(um18,slopec18,'b+',label='pTet-J23107-J23101 3')
plt.plot(um22,slopec22,'mo',label='pTet-pLacI-J23101 1')
plt.plot(um23,slopec23,'m^',label='pTet-pLacI-J23101 2')
plt.plot(um24,slopec24,'m+',label='pTet-pLacI-J23101 3')
plt.plot(um28,slopec28,'ro',label='pTet-pLas81-J23101 1')
plt.plot(um29,slopec29,'r^',label='pTet-pLas81-J23101 2')
plt.plot(um30,slopec30,'r+',label='pTet-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[72]: <matplotlib.legend.Legend at 0x267f1cf6048>
```

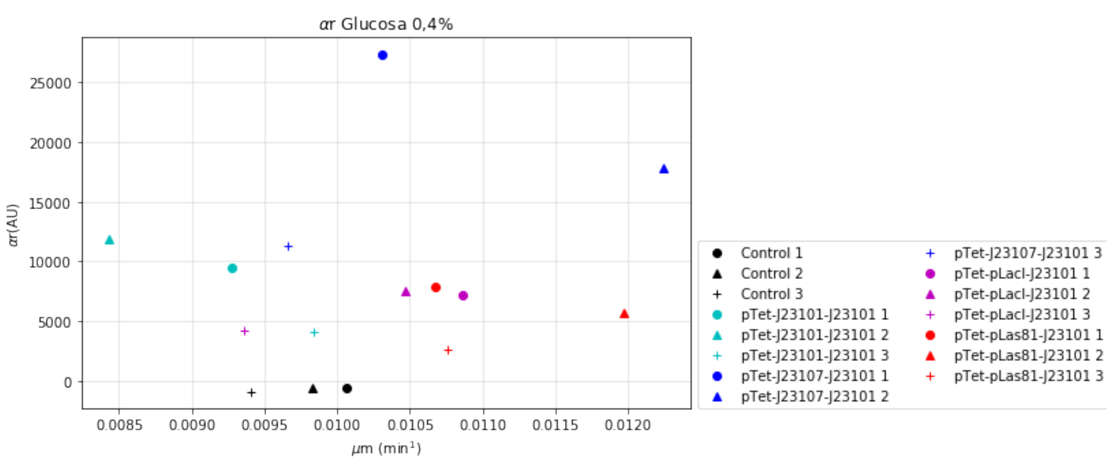


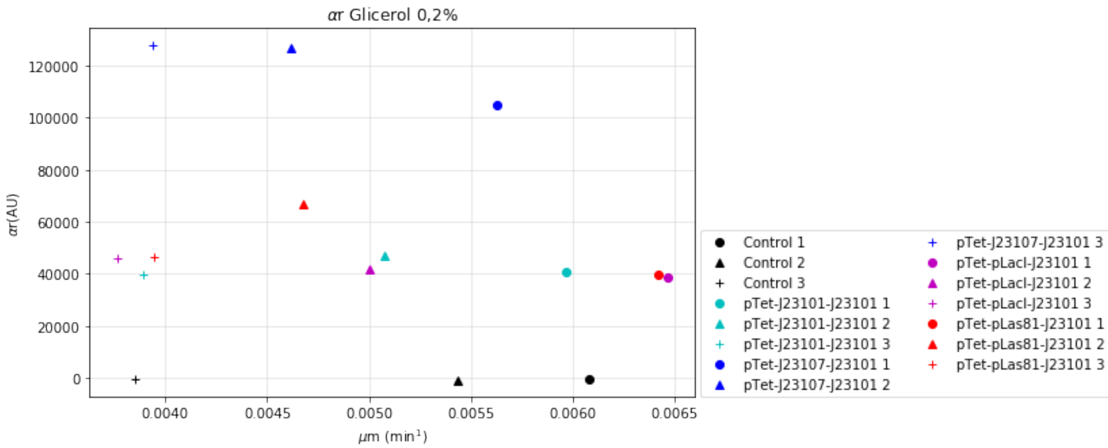




In [73]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'ko',label='Control 1')
plt.plot(um2,sloper2,'k^',label='Control 2')
plt.plot(um3,sloper3,'k+',label='Control 3')
plt.plot(um7,sloper7,'co',label='pTet-J23101-J23101 1')
plt.plot(um8,sloper8,'c^',label='pTet-J23101-J23101 2')
plt.plot(um9,sloper9,'c+',label='pTet-J23101-J23101 3')
plt.plot(um13,sloper13,'bo',label='pTet-J23107-J23101 1')
plt.plot(um14,sloper14,'b^',label='pTet-J23107-J23101 2')
plt.plot(um15,sloper15,'b+',label='pTet-J23107-J23101 3')
plt.plot(um19,sloper19,'mo',label='pTet-pLacI-J23101 1')
```

```python
plt.plot(um20,sloper20,'m^',label='pTet-pLacI-J23101 2')
plt.plot(um21,sloper21,'m+',label='pTet-pLacI-J23101 3')
plt.plot(um25,sloper25,'ro',label='pTet-pLas81-J23101 1')
plt.plot(um26,sloper26,'r^',label='pTet-pLas81-J23101 2')
plt.plot(um27,sloper27,'r+',label='pTet-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'$\alpha$r Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um4,sloper4,'ko',label='Control 1')
plt.plot(um5,sloper5,'k^',label='Control 2')
plt.plot(um6,sloper6,'k+',label='Control 3')
plt.plot(um10,sloper10,'co',label='pTet-J23101-J23101 1')
plt.plot(um11,sloper11,'c^',label='pTet-J23101-J23101 2')
plt.plot(um12,sloper12,'c+',label='pTet-J23101-J23101 3')
plt.plot(um16,sloper16,'bo',label='pTet-J23107-J23101 1')
plt.plot(um17,sloper17,'b^',label='pTet-J23107-J23101 2')
plt.plot(um18,sloper18,'b+',label='pTet-J23107-J23101 3')
plt.plot(um22,sloper22,'mo',label='pTet-pLacI-J23101 1')
plt.plot(um23,sloper23,'m^',label='pTet-pLacI-J23101 2')
plt.plot(um24,sloper24,'m+',label='pTet-pLacI-J23101 3')
plt.plot(um28,sloper28,'ro',label='pTet-pLas81-J23101 1')
plt.plot(um29,sloper29,'r^',label='pTet-pLas81-J23101 2')
plt.plot(um30,sloper30,'r+',label='pTet-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[73]: <matplotlib.legend.Legend at 0x267efb64710>
```

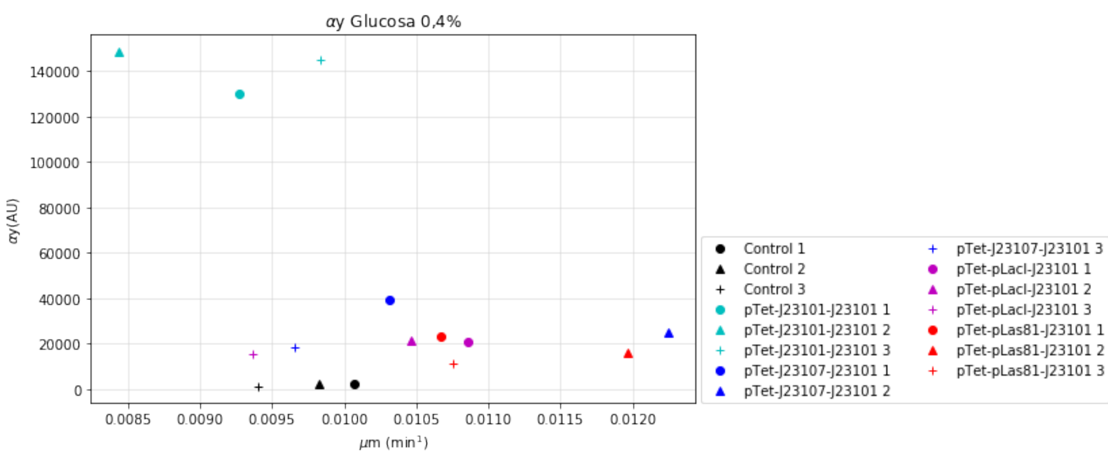


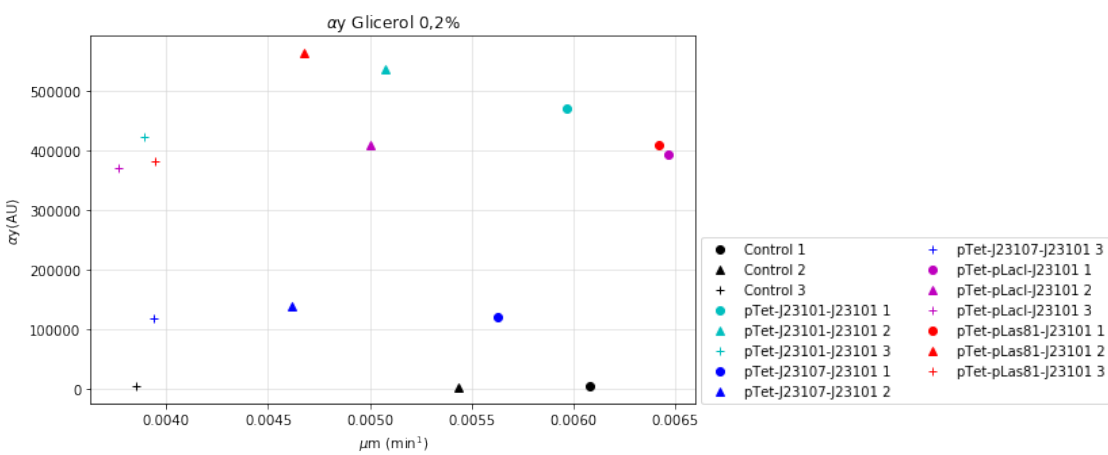



```
In [74]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$y Glucosa 0,4%')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$y(AU)')
         plt.plot(um1,slopey1,'ko',label='Control 1')
         plt.plot(um2,slopey2,'k^',label='Control 2')
         plt.plot(um3,slopey3,'k+',label='Control 3')
         plt.plot(um7,slopey7,'co',label='pTet-J23101-J23101 1')
         plt.plot(um8,slopey8,'c^',label='pTet-J23101-J23101 2')
         plt.plot(um9,slopey9,'c+',label='pTet-J23101-J23101 3')
         plt.plot(um13,slopey13,'bo',label='pTet-J23107-J23101 1')
         plt.plot(um14,slopey14,'b^',label='pTet-J23107-J23101 2')
         plt.plot(um15,slopey15,'b+',label='pTet-J23107-J23101 3')
         plt.plot(um19,slopey19,'mo',label='pTet-pLacI-J23101 1')
         plt.plot(um20,slopey20,'m^',label='pTet-pLacI-J23101 2')
         plt.plot(um21,slopey21,'m+',label='pTet-pLacI-J23101 3')
         plt.plot(um25,slopey25,'ro',label='pTet-pLas81-J23101 1')
         plt.plot(um26,slopey26,'r^',label='pTet-pLas81-J23101 2')
         plt.plot(um27,slopey27,'r+',label='pTet-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'$\alpha$y Glicerol 0,2%')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$y(AU)')
         plt.plot(um4,slopey4,'ko',label='Control 1')
         plt.plot(um5,slopey5,'k^',label='Control 2')
         plt.plot(um6,slopey6,'k+',label='Control 3')
```

```
plt.plot(um10,slopey10,'co',label='pTet-J23101-J23101 1')
plt.plot(um11,slopey11,'c^',label='pTet-J23101-J23101 2')
plt.plot(um12,slopey12,'c+',label='pTet-J23101-J23101 3')
plt.plot(um16,slopey16,'bo',label='pTet-J23107-J23101 1')
plt.plot(um17,slopey17,'b^',label='pTet-J23107-J23101 2')
plt.plot(um18,slopey18,'b+',label='pTet-J23107-J23101 3')
plt.plot(um22,slopey22,'mo',label='pTet-pLacI-J23101 1')
plt.plot(um23,slopey23,'m^',label='pTet-pLacI-J23101 2')
plt.plot(um24,slopey24,'m+',label='pTet-pLacI-J23101 3')
plt.plot(um28,slopey28,'ro',label='pTet-pLas81-J23101 1')
plt.plot(um29,slopey29,'r^',label='pTet-pLas81-J23101 2')
plt.plot(um30,slopey30,'r+',label='pTet-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[74]: <matplotlib.legend.Legend at 0x267f1245f28>
```

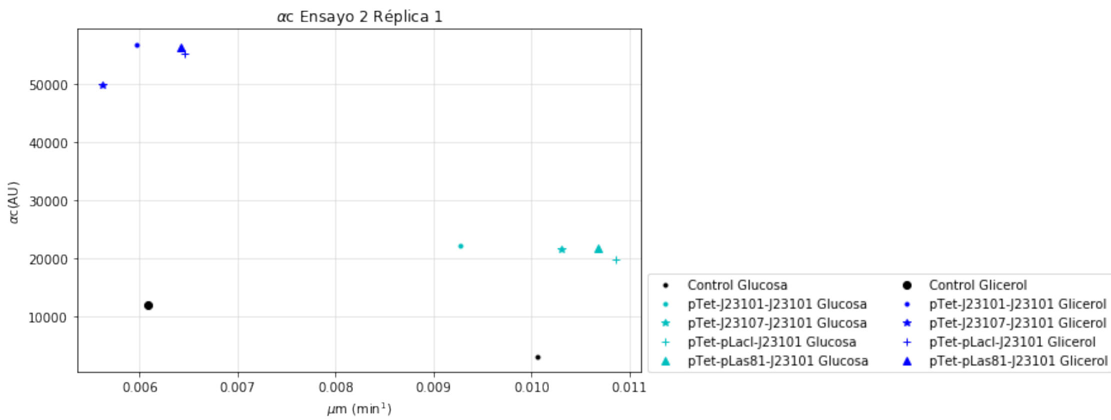


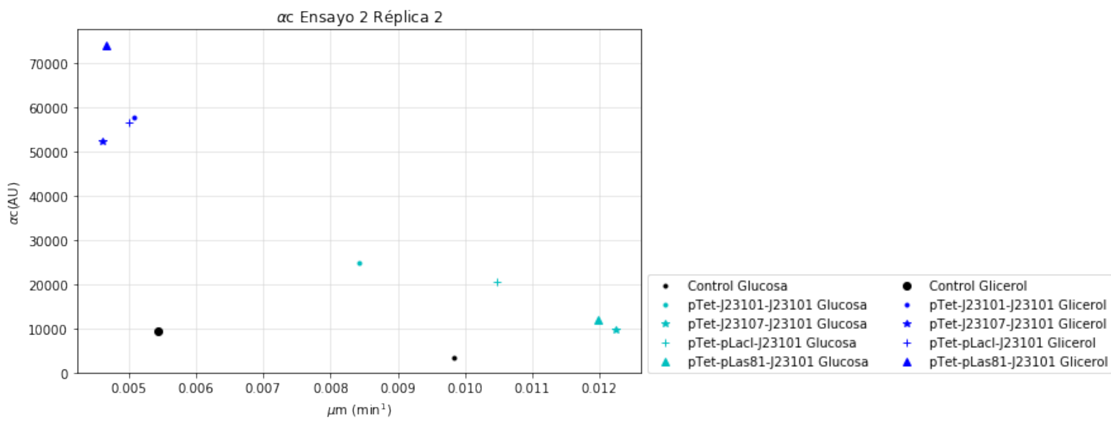



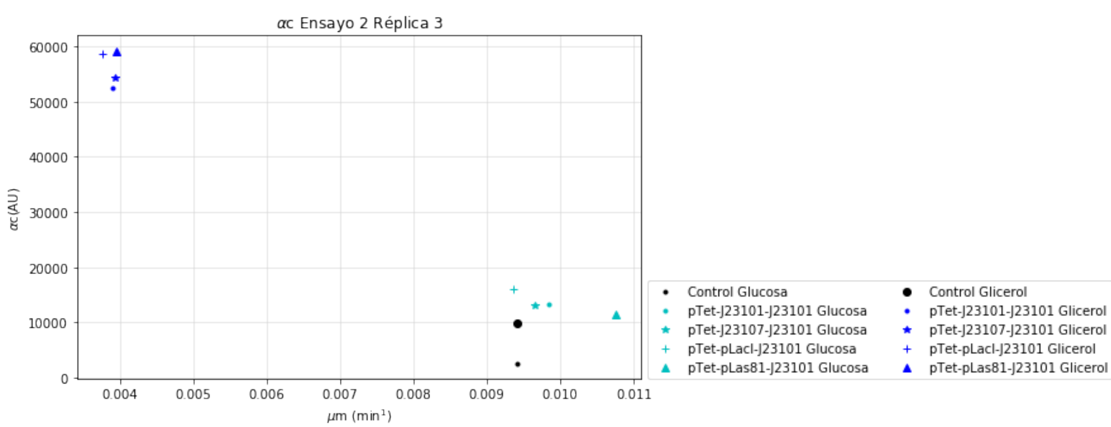
```
In [75]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Ensayo 2 Réplica 1')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$c(AU)')
         plt.plot(um1,slopec1,'k.',label='Control Glucosa')
         plt.plot(um7,slopec7,'c.',label='pTet-J23101-J23101 Glucosa')
         plt.plot(um13,slopec13,'c*',label='pTet-J23107-J23101 Glucosa')
         plt.plot(um19,slopec19,'c+',label='pTet-pLacI-J23101 Glucosa')
         plt.plot(um25,slopec25,'c^',label='pTet-pLas81-J23101 Glucosa')
         plt.plot(um4,slopec4,'ko',label='Control Glicerol')
         plt.plot(um10,slopec10,'b.',label='pTet-J23101-J23101 Glicerol')
         plt.plot(um16,slopec16,'b*',label='pTet-J23107-J23101 Glicerol')
         plt.plot(um22,slopec22,'b+',label='pTet-pLacI-J23101 Glicerol')
         plt.plot(um28,slopec28,'b^',label='pTet-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Ensayo 2 Réplica 2')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$c(AU)')
         plt.plot(um2,slopec2,'k.',label='Control Glucosa')
         plt.plot(um8,slopec8,'c.',label='pTet-J23101-J23101 Glucosa')
         plt.plot(um14,slopec14,'c*',label='pTet-J23107-J23101 Glucosa')
         plt.plot(um20,slopec20,'c+',label='pTet-pLacI-J23101 Glucosa')
         plt.plot(um26,slopec26,'c^',label='pTet-pLas81-J23101 Glucosa')
         plt.plot(um5,slopec5,'ko',label='Control Glicerol')
         plt.plot(um11,slopec11,'b.',label='pTet-J23101-J23101 Glicerol')
         plt.plot(um17,slopec17,'b*',label='pTet-J23107-J23101 Glicerol')
         plt.plot(um23,slopec23,'b+',label='pTet-pLacI-J23101 Glicerol')
         plt.plot(um29,slopec29,'b^',label='pTet-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Ensayo 2 Réplica 3')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$c(AU)')
         plt.plot(um3,slopec3,'k.',label='Control Glucosa')
         plt.plot(um9,slopec9,'c.',label='pTet-J23101-J23101 Glucosa')
         plt.plot(um15,slopec15,'c*',label='pTet-J23107-J23101 Glucosa')
         plt.plot(um21,slopec21,'c+',label='pTet-pLacI-J23101 Glucosa')
         plt.plot(um27,slopec27,'c^',label='pTet-pLas81-J23101 Glucosa')
         plt.plot(um3,slopec6,'ko',label='Control Glicerol')
         plt.plot(um12,slopec12,'b.',label='pTet-J23101-J23101 Glicerol')
         plt.plot(um18,slopec18,'b*',label='pTet-J23107-J23101 Glicerol')
         plt.plot(um24,slopec24,'b+',label='pTet-pLacI-J23101 Glicerol')
```

```python
plt.plot(um30,slopec30,'b^',label='pTet-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[75]: <matplotlib.legend.Legend at 0x267f1139da0>



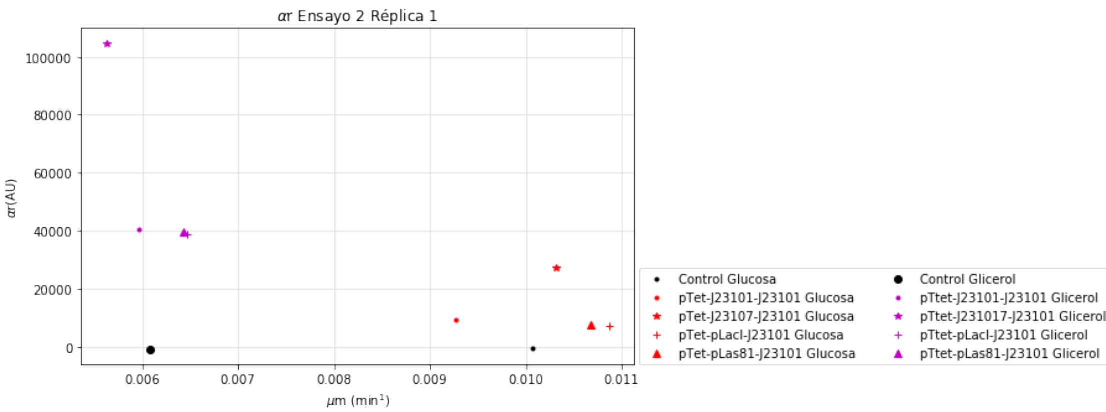



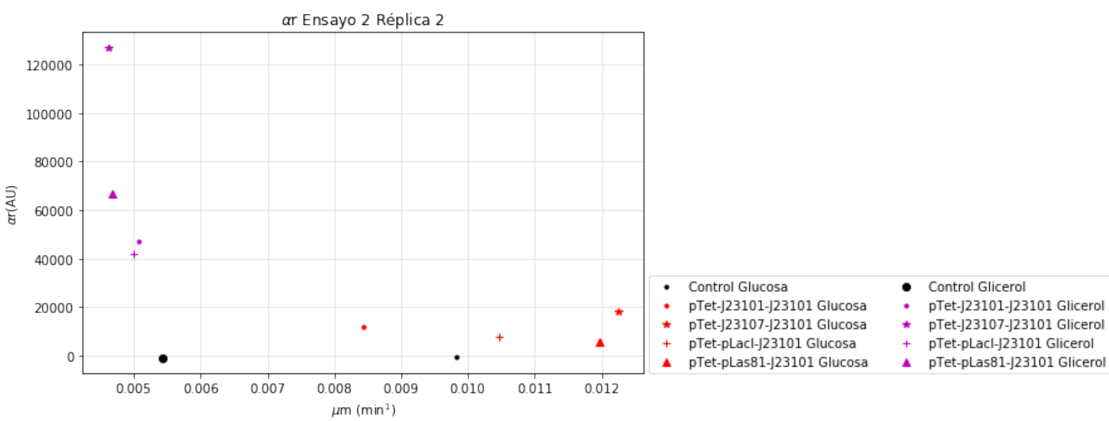



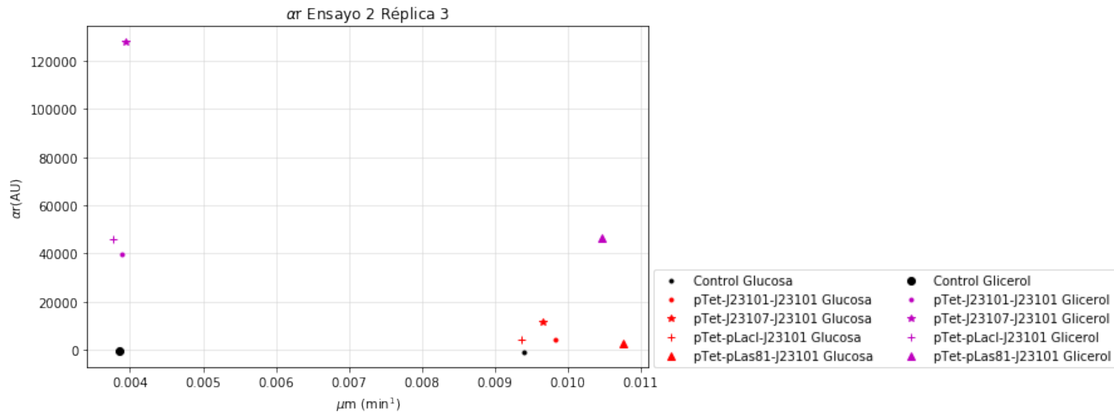
```
In [76]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 2 Réplica 1')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um1,sloper1,'k.',label='Control Glucosa ')
         plt.plot(um7,sloper7,'r.',label='pTet-J23101-J23101 Glucosa ')
         plt.plot(um13,sloper13,'r*',label='pTet-J23107-J23101 Glucosa ')
         plt.plot(um19,sloper19,'r+',label='pTet-pLacI-J23101 Glucosa ')
         plt.plot(um25,sloper25,'r^',label='pTet-pLas81-J23101 Glucosa ')
         plt.plot(um4,sloper4,'ko',label='Control Glicerol ')
         plt.plot(um10,sloper10,'m.',label='pTtet-J23101-J23101 Glicerol ')
         plt.plot(um16,sloper16,'m*',label='pTtet-J231017-J23101 Glicerol')
         plt.plot(um22,sloper22,'m+',label='pTtet-pLacI-J23101 Glicerol ')
         plt.plot(um28,sloper28,'m^',label='pTtet-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 2 Réplica 2')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um2,sloper2,'k.',label='Control Glucosa')
         plt.plot(um8,sloper8,'r.',label='pTet-J23101-J23101 Glucosa')
         plt.plot(um14,sloper14,'r*',label='pTet-J23107-J23101 Glucosa')
         plt.plot(um20,sloper20,'r+',label='pTet-pLacI-J23101 Glucosa')
         plt.plot(um26,sloper26,'r^',label='pTet-pLas81-J23101 Glucosa')
         plt.plot(um5,sloper5,'ko',label='Control Glicerol')
         plt.plot(um11,sloper11,'m.',label='pTet-J23101-J23101 Glicerol')
         plt.plot(um17,sloper17,'m*',label='pTet-J23107-J23101 Glicerol')
         plt.plot(um23,sloper23,'m+',label='pTet-pLacI-J23101 Glicerol')
         plt.plot(um29,sloper29,'m^',label='pTet-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 2 Réplica 3')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um3,sloper3,'k.',label='Control Glucosa')
         plt.plot(um9,sloper9,'r.',label='pTet-J23101-J23101 Glucosa')
         plt.plot(um15,sloper15,'r*',label='pTet-J23107-J23101 Glucosa')
         plt.plot(um21,sloper21,'r+',label='pTet-pLacI-J23101 Glucosa')
         plt.plot(um27,sloper27,'r^',label='pTet-pLas81-J23101 Glucosa')
         plt.plot(um6,sloper6,'ko',label='Control Glicerol')
```

```
plt.plot(um12,sloper12,'m.',label='pTet-J23101-J23101 Glicerol')
plt.plot(um18,sloper18,'m*',label='pTet-J23107-J23101 Glicerol')
plt.plot(um24,sloper24,'m+',label='pTet-pLacI-J23101 Glicerol')
plt.plot(um20,sloper30,'m^',label='pTet-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[76]: <matplotlib.legend.Legend at 0x267f391fd30>
```



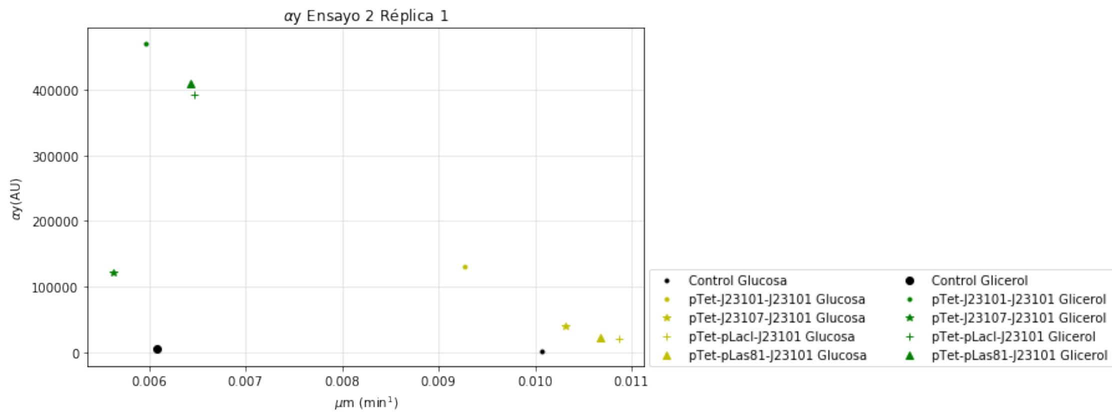



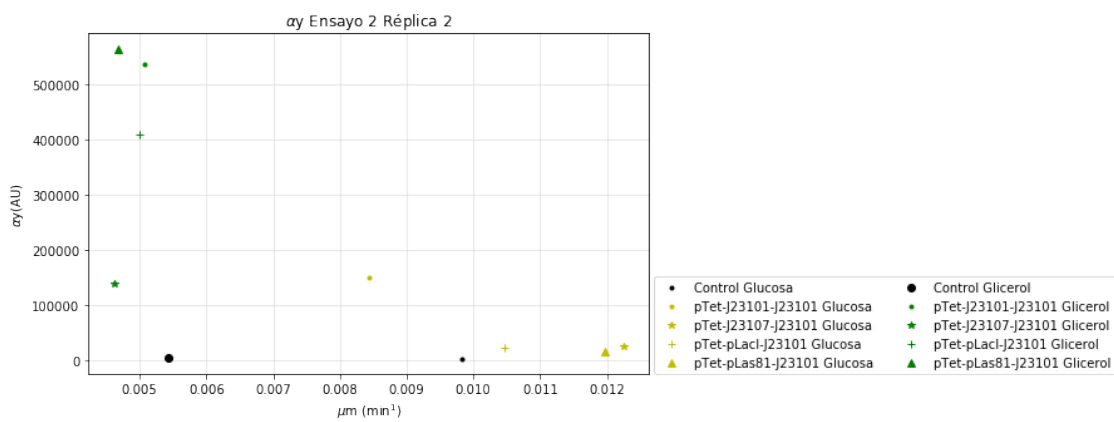


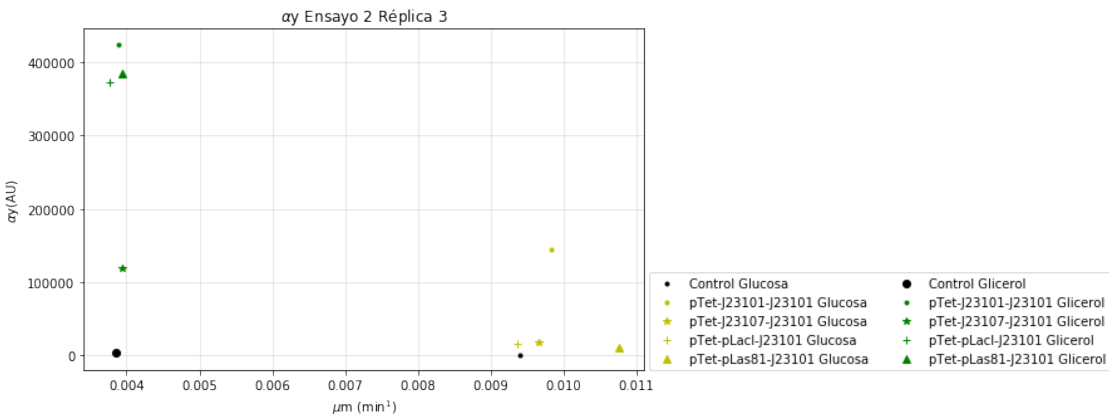

In [77]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 2 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'k.',label='Control Glucosa ')
plt.plot(um7,slopey7,'y.',label='pTet-J23101-J23101 Glucosa ')
plt.plot(um13,slopey13,'y*',label='pTet-J23107-J23101 Glucosa ')
plt.plot(um19,slopey19,'y+',label='pTet-pLacI-J23101 Glucosa ')
plt.plot(um25,slopey25,'y^',label='pTet-pLas81-J23101 Glucosa ')
plt.plot(um4,slopey4,'ko',label='Control Glicerol ')
plt.plot(um10,slopey10,'g.',label='pTet-J23101-J23101 Glicerol ')
plt.plot(um16,slopey16,'g*',label='pTet-J23107-J23101 Glicerol')
plt.plot(um22,slopey22,'g+',label='pTet-pLacI-J23101 Glicerol ')
plt.plot(um28,slopey28,'g^',label='pTet-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 2 Réplica 2')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um2,slopey2,'k.',label='Control Glucosa')
plt.plot(um8,slopey8,'y.',label='pTet-J23101-J23101 Glucosa')
plt.plot(um14,slopey14,'y*',label='pTet-J23107-J23101 Glucosa')
plt.plot(um20,slopey20,'y+',label='pTet-pLacI-J23101 Glucosa')
plt.plot(um26,slopey26,'y^',label='pTet-pLas81-J23101 Glucosa')
plt.plot(um5,slopey5,'ko',label='Control Glicerol')
plt.plot(um11,slopey11,'g.',label='pTet-J23101-J23101 Glicerol')
plt.plot(um17,slopey17,'g*',label='pTet-J23107-J23101 Glicerol')
plt.plot(um23,slopey23,'g+',label='pTet-pLacI-J23101 Glicerol')
plt.plot(um29,slopey29,'g^',label='pTet-pLas81-J23101 Glicerol')
```

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

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 2 Réplica 3')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um3,slopey3,'k.',label='Control Glucosa')
plt.plot(um9,slopey9,'y.',label='pTet-J23101-J23101 Glucosa')
plt.plot(um15,slopey15,'y*',label='pTet-J23107-J23101 Glucosa')
plt.plot(um21,slopey21,'y+',label='pTet-pLacI-J23101 Glucosa')
plt.plot(um27,slopey27,'y^',label='pTet-pLas81-J23101 Glucosa')
plt.plot(um6,slopey6,'ko',label='Control Glicerol')
plt.plot(um12,slopey12,'g.',label='pTet-J23101-J23101 Glicerol')
plt.plot(um18,slopey18,'g*',label='pTet-J23107-J23101 Glicerol')
plt.plot(um24,slopey24,'g+',label='pTet-pLacI-J23101 Glicerol')
plt.plot(um30,slopey30,'g^',label='pTet-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[77]: <matplotlib.legend.Legend at 0x267eef7dc50>



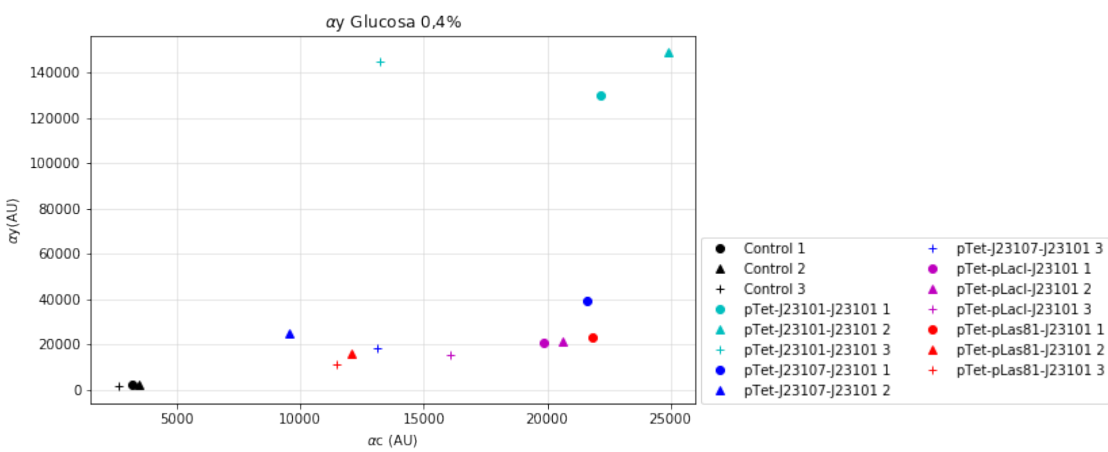



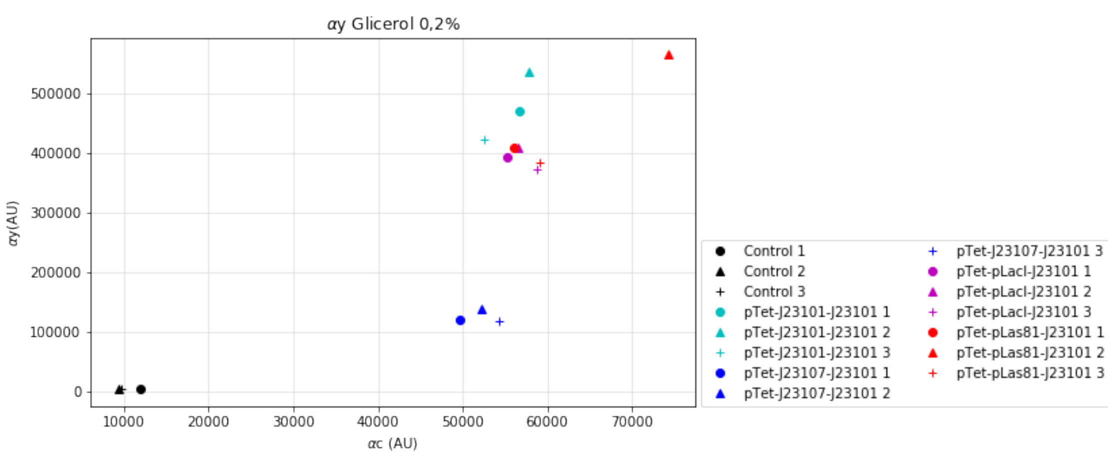



In [78]:
```
#grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopey1,'ko',label='Control 1')
plt.plot(slopec2,slopey2,'k^',label='Control 2')
plt.plot(slopec3,slopey3,'k+',label='Control 3')
plt.plot(slopec7,slopey7,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec8,slopey8,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec9,slopey9,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec13,slopey13,'bo',label='pTet-J23107-J23101 1')
plt.plot(slopec14,slopey14,'b^',label='pTet-J23107-J23101 2')
plt.plot(slopec15,slopey15,'b+',label='pTet-J23107-J23101 3')
plt.plot(slopec19,slopey19,'mo',label='pTet-pLacI-J23101 1')
plt.plot(slopec20,slopey20,'m^',label='pTet-pLacI-J23101 2')
plt.plot(slopec21,slopey21,'m+',label='pTet-pLacI-J23101 3')
plt.plot(slopec25,slopey25,'ro',label='pTet-pLas81-J23101 1')
plt.plot(slopec26,slopey26,'r^',label='pTet-pLas81-J23101 2')
plt.plot(slopec27,slopey27,'r+',label='pTet-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(slopec4,slopey4,'ko',label='Control 1')
plt.plot(slopec5,slopey5,'k^',label='Control 2')
```

```
plt.plot(slopec6,slopey6,'k+',label='Control 3')
plt.plot(slopec10,slopey10,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec11,slopey11,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec12,slopey12,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec16,slopey16,'bo',label='pTet-J23107-J23101 1')
plt.plot(slopec17,slopey17,'b^',label='pTet-J23107-J23101 2')
plt.plot(slopec18,slopey18,'b+',label='pTet-J23107-J23101 3')
plt.plot(slopec22,slopey22,'mo',label='pTet-pLacI-J23101 1')
plt.plot(slopec23,slopey23,'m^',label='pTet-pLacI-J23101 2')
plt.plot(slopec24,slopey24,'m+',label='pTet-pLacI-J23101 3')
plt.plot(slopec28,slopey28,'ro',label='pTet-pLas81-J23101 1')
plt.plot(slopec29,slopey29,'r^',label='pTet-pLas81-J23101 2')
plt.plot(slopec30,slopey30,'r+',label='pTet-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

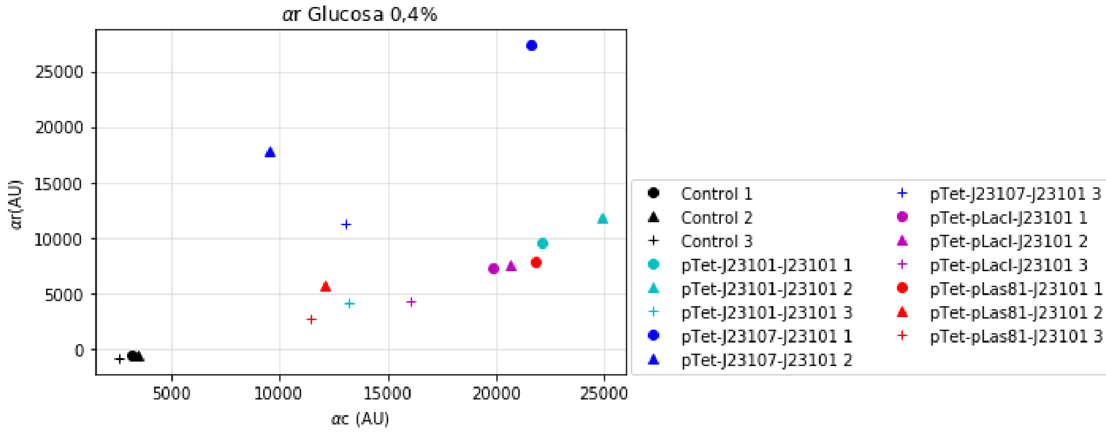
```
Out[78]: <matplotlib.legend.Legend at 0x267f2006c50>
```

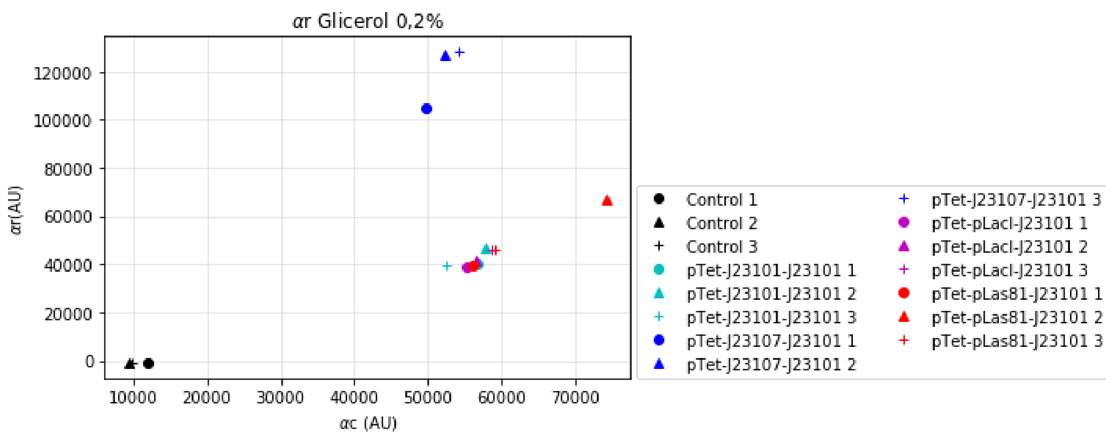


In [79]:
```
#grafico de ac versus Um
plt.figure()
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(slopec1,sloper1,'ko',label='Control 1')
plt.plot(slopec2,sloper2,'k^',label='Control 2')
plt.plot(slopec3,sloper3,'k+',label='Control 3')
plt.plot(slopec7,sloper7,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec8,sloper8,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec9,sloper9,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec13,sloper13,'bo',label='pTet-J23107-J23101 1')
plt.plot(slopec14,sloper14,'b^',label='pTet-J23107-J23101 2')
plt.plot(slopec15,sloper15,'b+',label='pTet-J23107-J23101 3')
plt.plot(slopec19,sloper19,'mo',label='pTet-pLacI-J23101 1')
plt.plot(slopec20,sloper20,'m^',label='pTet-pLacI-J23101 2')
plt.plot(slopec21,sloper21,'m+',label='pTet-pLacI-J23101 3')
plt.plot(slopec25,sloper25,'ro',label='pTet-pLas81-J23101 1')
plt.plot(slopec26,sloper26,'r^',label='pTet-pLas81-J23101 2')
plt.plot(slopec27,sloper27,'r+',label='pTet-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()
plt.title(r'$\alpha$r Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(slopec4,sloper4,'ko',label='Control 1')
plt.plot(slopec5,sloper5,'k^',label='Control 2')
plt.plot(slopec6,sloper6,'k+',label='Control 3')
plt.plot(slopec10,sloper10,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec11,sloper11,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec12,sloper12,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec16,sloper16,'bo',label='pTet-J23107-J23101 1')
plt.plot(slopec17,sloper17,'b^',label='pTet-J23107-J23101 2')
plt.plot(slopec18,sloper18,'b+',label='pTet-J23107-J23101 3')
plt.plot(slopec22,sloper22,'mo',label='pTet-pLacI-J23101 1')
plt.plot(slopec23,sloper23,'m^',label='pTet-pLacI-J23101 2')
plt.plot(slopec24,sloper24,'m+',label='pTet-pLacI-J23101 3')
plt.plot(slopec28,sloper28,'ro',label='pTet-pLas81-J23101 1')
plt.plot(slopec29,sloper29,'r^',label='pTet-pLas81-J23101 2')
plt.plot(slopec30,sloper30,'r+',label='pTet-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[79]: <matplotlib.legend.Legend at 0x267f34fddd8>
```

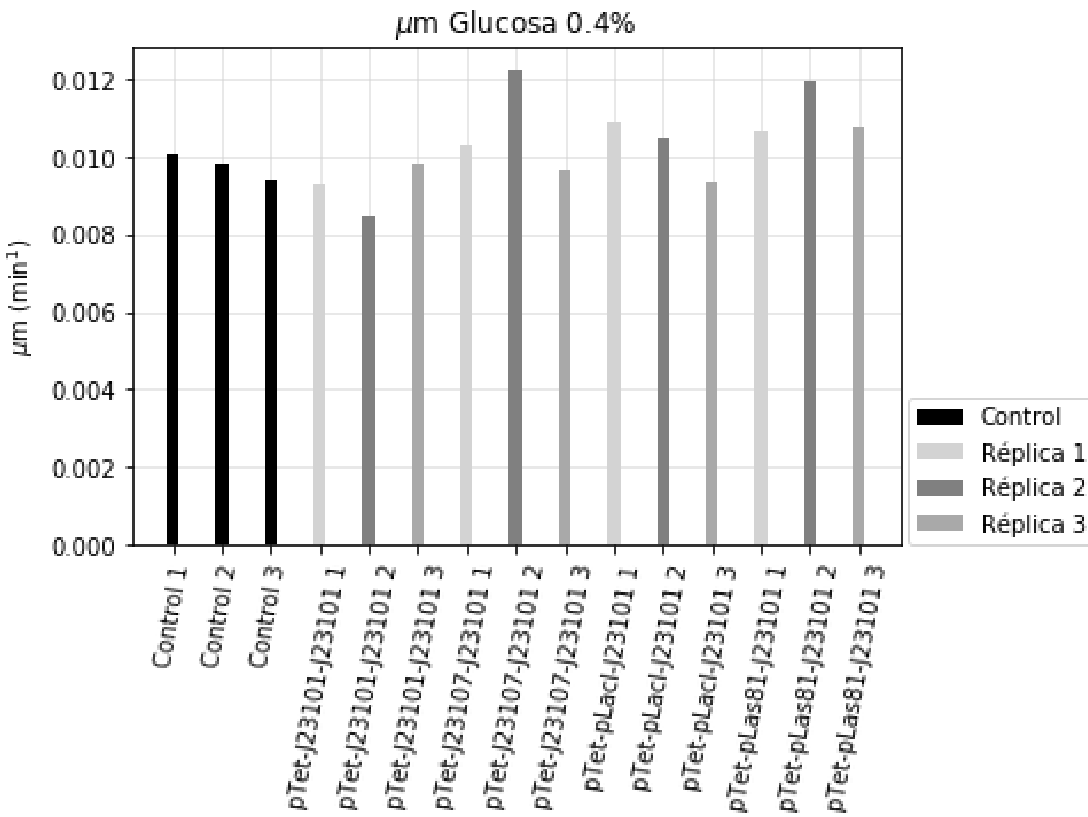



```
In [80]: #Grafico de barras um de FPs
         uglu=[um1,um2,um3,um7,um8,um9,um13,um14,um15,um19,um20,um21,um25,um26,um27]
         ugli=[um4,um5,um6,um10,um11,um12,um16,um17,um18,um22,um23,um24,um28,um29,um30]

         X = np.arange(15)
         plt.figure()
         plt.title(r'$\mu$m Glucosa 0.4%')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.bar(X[0]+0.00,uglu[0],color='k',width=0.25,label='Control',zorder=3)
         plt.bar(X[1]+0.00,uglu[1],color='k',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,uglu[2],color='k',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,uglu[3],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
         plt.bar(X[4]+0.00,uglu[4],color='grey',width=0.25,label='Réplica 2',zorder=3)
         plt.bar(X[5]+0.00,uglu[5],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
```

```
plt.bar(X[6]+0.00,uglu[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,uglu[7],color='grey',width=0.25,zorder=3)
plt.bar(X[8]+0.00,uglu[8],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,uglu[9],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,uglu[10],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,uglu[11],color='darkgrey',width=0.25,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.xticks(X,['Control 1','Control 2','Control 3','pTet-J23101-J23101 1','pTet-J23101-J
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

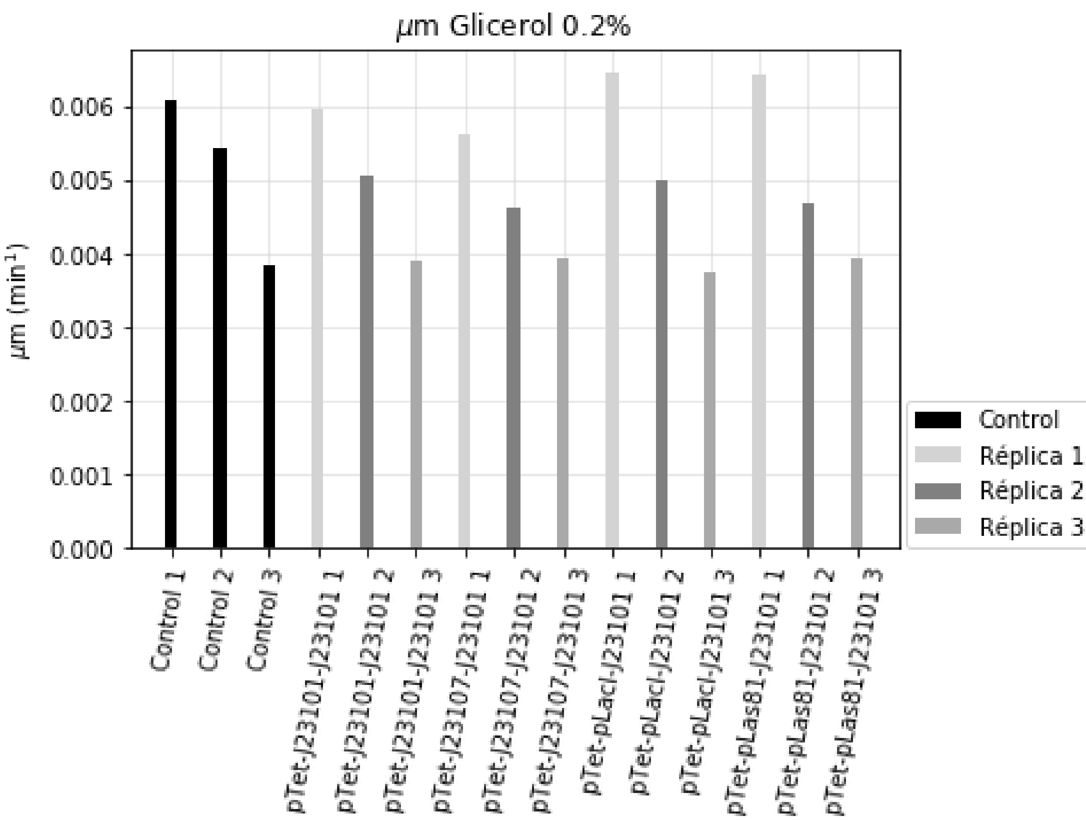
Out[80]: <matplotlib.legend.Legend at 0x267f40c35f8>




In [81]:
```
X = np.arange(15)
plt.figure()
plt.title(r'$\mu$m Glicerol 0.2%')
plt.ylabel(r'$\mu$m (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='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='grey',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,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.xticks(X,['Control 1','Control 2','Control 3','pTet-J23101-J23101 1','pTet-J23101-J
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

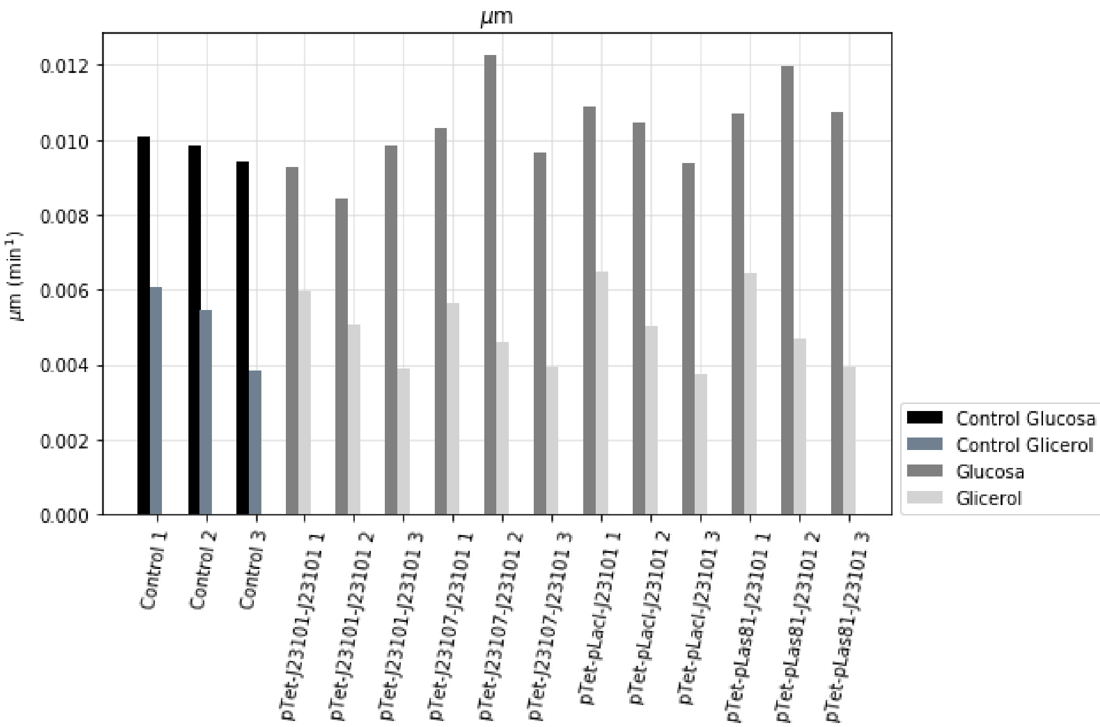
Out[81]: <matplotlib.legend.Legend at 0x267f42b1550>




```
In [82]: X = np.arange(15)
         plt.figure(figsize=(8,5))
```

```
plt.title(r'$\mu$m')
plt.ylabel(r'$\mu$m (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=
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]-0.25,uglu[3],color='grey',width=0.25,label='Glucosa',zorder=3)
plt.bar(X[3]+0.00,ugli[3],color='lightgrey',width=0.25,label='Glicerol',zorder=3)
plt.bar(X[4]-0.25,uglu[4],color='grey',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[5]-0.25,uglu[5],color='grey',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[6]-0.25,uglu[6],color='grey',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]-0.25,uglu[7],color='grey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[8]-0.25,uglu[8],color='grey',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[9]-0.25,uglu[9],color='grey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,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.xticks(X,['Control 1','Control 2','Control 3','pTet-J23101-J23101 1','pTet-J23101-J
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

```
Out[82]: <matplotlib.legend.Legend at 0x267f445a780>
```




In [83]:
```
#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
```

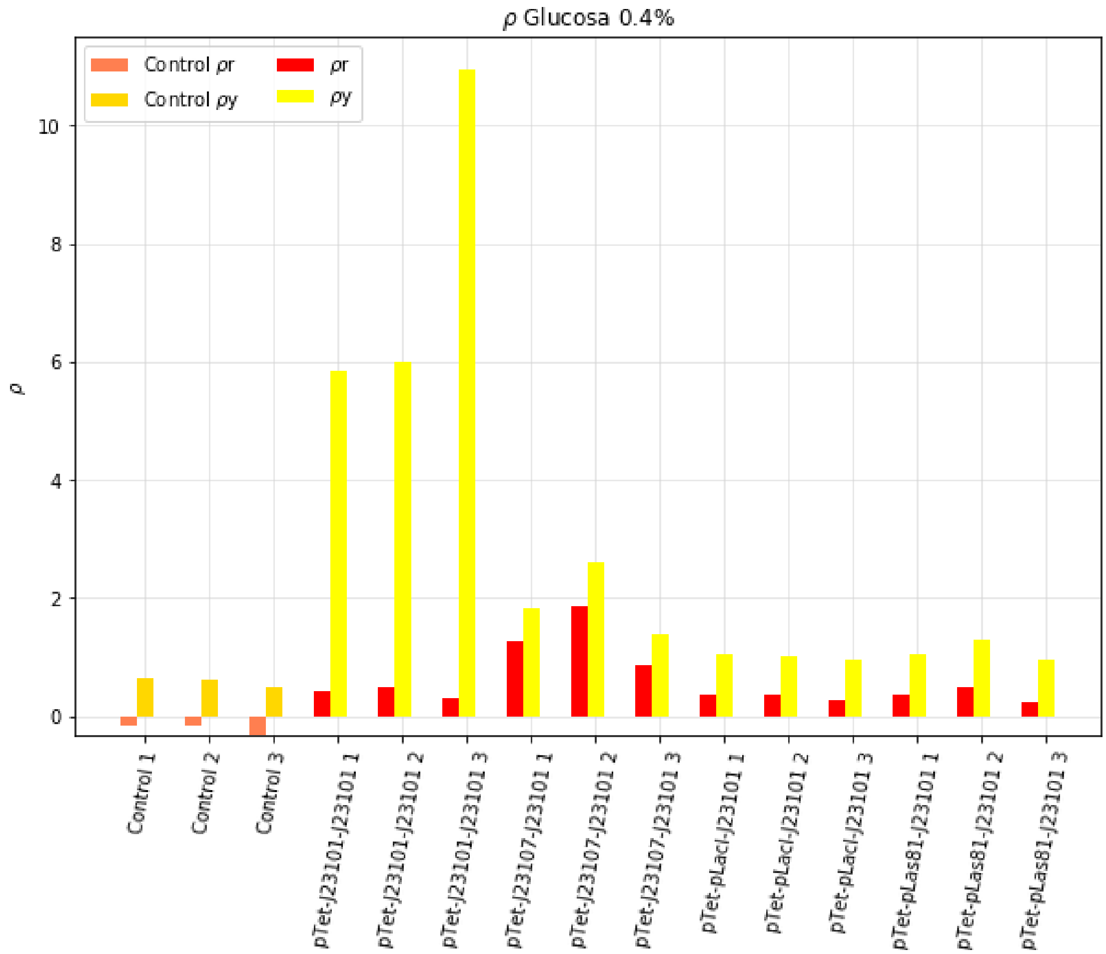
```
pr24=sloper24/slopec24
pr25=sloper25/slopec25
pr26=sloper26/slopec26
pr27=sloper27/slopec27
pr28=sloper28/slopec28
pr29=sloper29/slopec29
pr30=sloper30/slopec30


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

In [84]:
```
#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


ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py1
ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py2
```

```
ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,
```

```
In [85]: #grafico ro de yfp y de rfp
         ro_rfp=[pr1,pr2,pr3,pr4,pr5,pr6,pr7,pr8,pr9,pr10,pr11,pr12,pr13,pr14,pr15,pr16,pr17,pr1
         ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py1


         X = np.arange(15)
         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',zo
         plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y',zor
         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='r',width=0.25,label=r'$\rho$r',zorder=3)
         plt.bar(X[3]+0.00,ro_yfp[6],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
         plt.bar(X[4]-0.25,ro_rfp[7],color='r',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,ro_yfp[7],color='yellow',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,ro_rfp[8],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,ro_yfp[8],color='yellow',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,ro_rfp[12],color='r',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,ro_yfp[12],color='yellow',width=0.25,zorder=3)
         plt.bar(X[7]-0.25,ro_rfp[13],color='r',width=0.25,zorder=3)
         plt.bar(X[7]+0.00,ro_yfp[13],color='yellow',width=0.25,zorder=3)
         plt.bar(X[8]-0.25,ro_rfp[14],color='r',width=0.25,zorder=3)
         plt.bar(X[8]+0.00,ro_yfp[14],color='yellow',width=0.25,zorder=3)
         plt.bar(X[9]-0.25,ro_rfp[18],color='r',width=0.25,zorder=3)
         plt.bar(X[9]+0.00,ro_yfp[18],color='yellow',width=0.25,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.xticks(X,['Control 1','Control 2','Control 3','pTet-J23101-J23101 1','pTet-J23101-J
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper left',ncol=2)
```

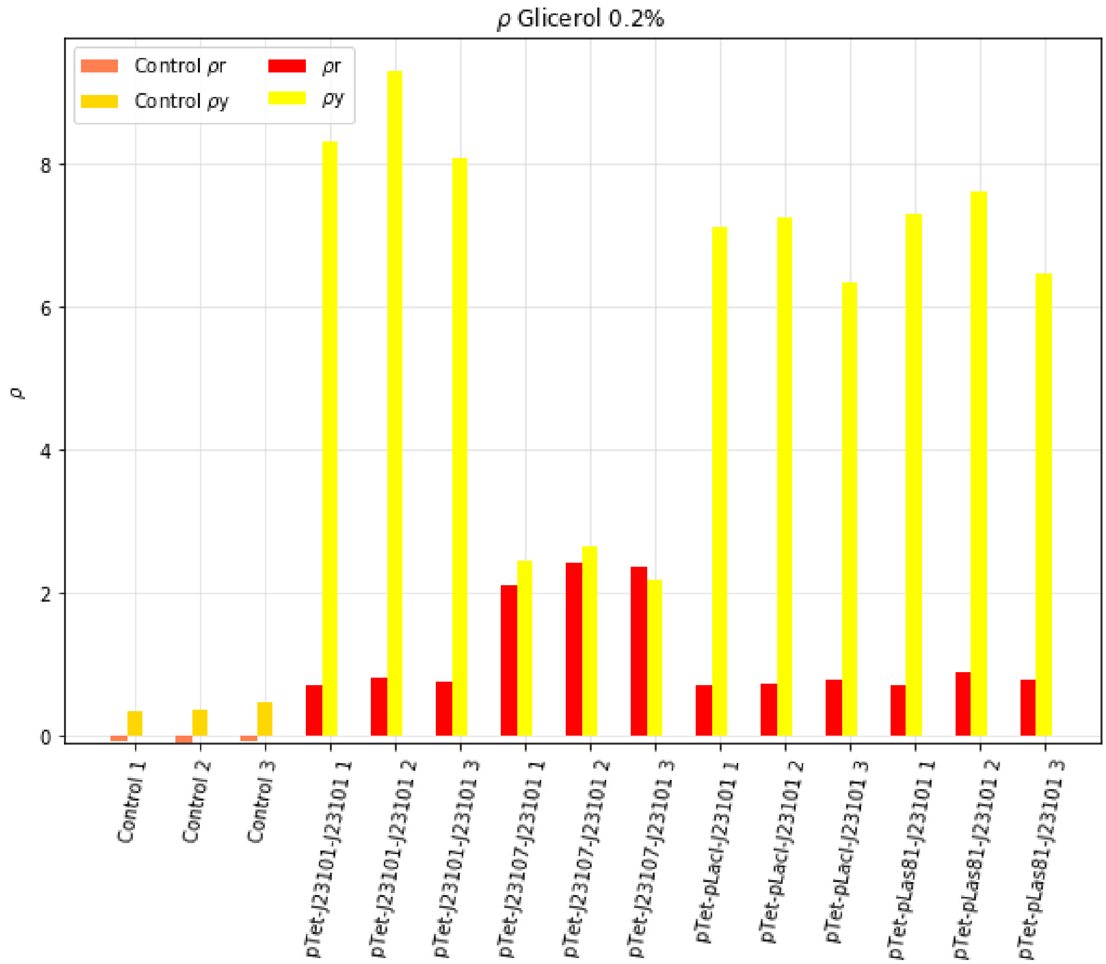
```
Out[85]: <matplotlib.legend.Legend at 0x267f1116c88>
```




```
In [86]: X = np.arange(15)
         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',zo
         plt.bar(X[0]+0.00,ro_yfp[3],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y',zor
         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='r',width=0.25,label=r'$\rho$r',zorder=3)
         plt.bar(X[3]+0.00,ro_yfp[9],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
         plt.bar(X[4]-0.25,ro_rfp[10],color='r',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,ro_yfp[10],color='yellow',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,ro_rfp[11],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,ro_yfp[11],color='yellow',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,ro_rfp[15],color='r',width=0.25,zorder=3)
```

```
plt.bar(X[6]+0.00,ro_yfp[15],color='yellow',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[16],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[16],color='yellow',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[17],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[17],color='yellow',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[21],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ro_yfp[21],color='yellow',width=0.25,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.xticks(X,['Control 1','Control 2','Control 3','pTet-J23101-J23101 1','pTet-J23101-J
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

```
Out[86]: <matplotlib.legend.Legend at 0x267f1dd0d68>
```

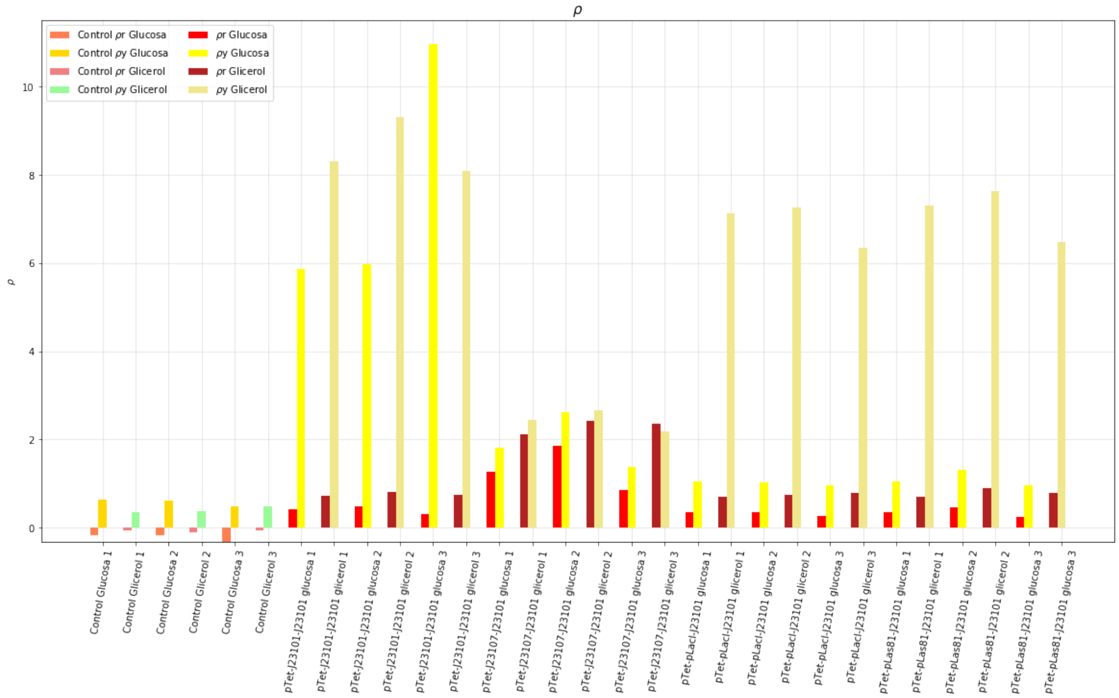



```
In [87]: X = np.arange(30)
         plt.figure(figsize=(20,10))
         plt.title(r'$\rho$',fontsize=15.0)
         plt.ylabel(r'$\rho$')
         plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r Glu
         plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y Gluc
         plt.bar(X[1]-0.25,ro_rfp[3],color='lightcoral',width=0.25,label= 'Control'+' '+r'$\rho$
         plt.bar(X[1]+0.00,ro_yfp[3],color='palegreen',width=0.25,label= 'Control'+' '+r'$\rho$y
         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='r',width=0.25,label=r'$\rho$r Glucosa',zorder=3)
```

```
plt.bar(X[6]+0.00,ro_yfp[6],color='yellow',width=0.25,label=r'$\rho$y Glucosa',zorder=3
plt.bar(X[7]-0.25,ro_rfp[9],color='firebrick',width=0.25,label=r'$\rho$r Glicerol',zord
plt.bar(X[7]+0.00,ro_yfp[9],color='khaki',width=0.25,label=r'$\rho$y Glicerol',zorder=3
plt.bar(X[8]-0.25,ro_rfp[7],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[7],color='yellow',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[10],color='firebrick',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ro_yfp[10],color='khaki',width=0.25,zorder=3)
plt.bar(X[10]-0.25,ro_rfp[8],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[8],color='yellow',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[11],color='firebrick',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[11],color='khaki',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[12],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[12],color='yellow',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[15],color='firebrick',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[15],color='khaki',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[13],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[13],color='yellow',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[16],color='firebrick',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[16],color='khaki',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[14],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[14],color='yellow',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[17],color='firebrick',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[17],color='khaki',width=0.25,zorder=3)
plt.bar(X[18]-0.25,ro_rfp[18],color='r',width=0.25,zorder=3)
plt.bar(X[18]+0.00,ro_yfp[18],color='yellow',width=0.25,zorder=3)
plt.bar(X[19]-0.25,ro_rfp[21],color='firebrick',width=0.25,zorder=3)
plt.bar(X[19]+0.00,ro_yfp[21],color='khaki',width=0.25,zorder=3)
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.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control Gl
```

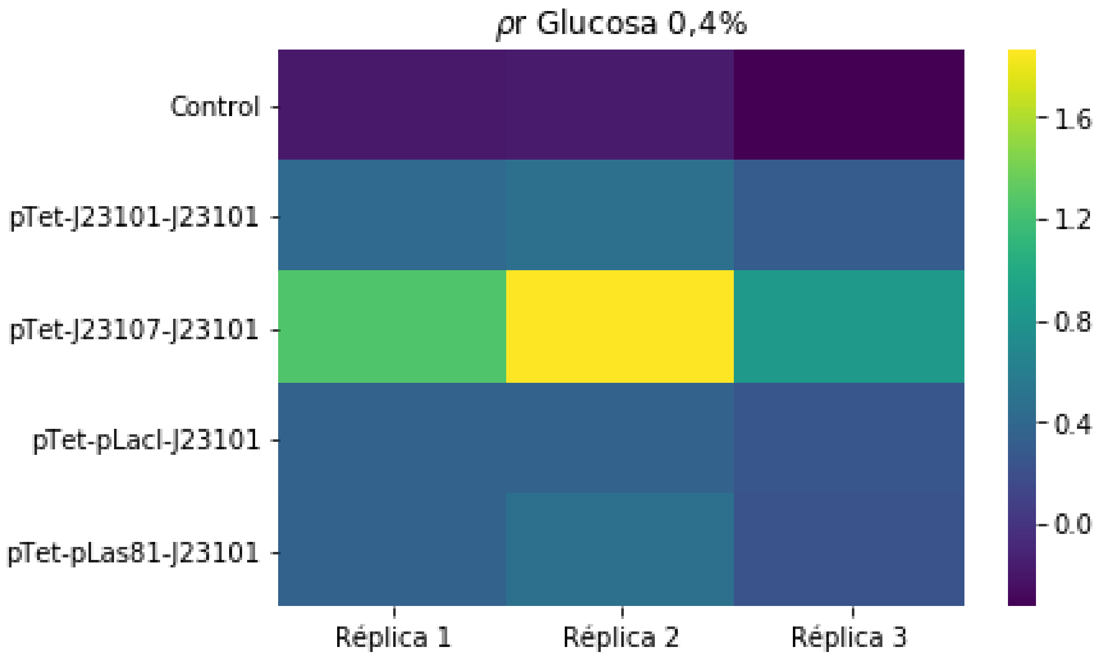
```
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

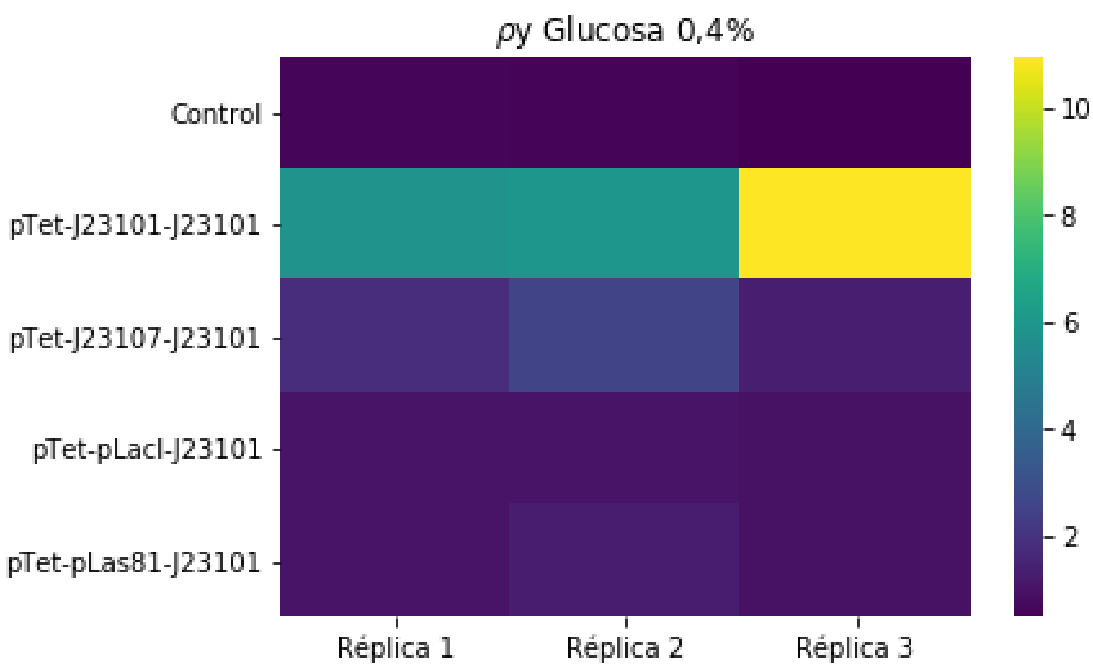
Out[87]: <matplotlib.legend.Legend at 0x267f1798c18>



```
In [88]: ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr2

         ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py2

         xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

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

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

Out[88]: <matplotlib.axes._subplots.AxesSubplot at 0x267f1139b38>

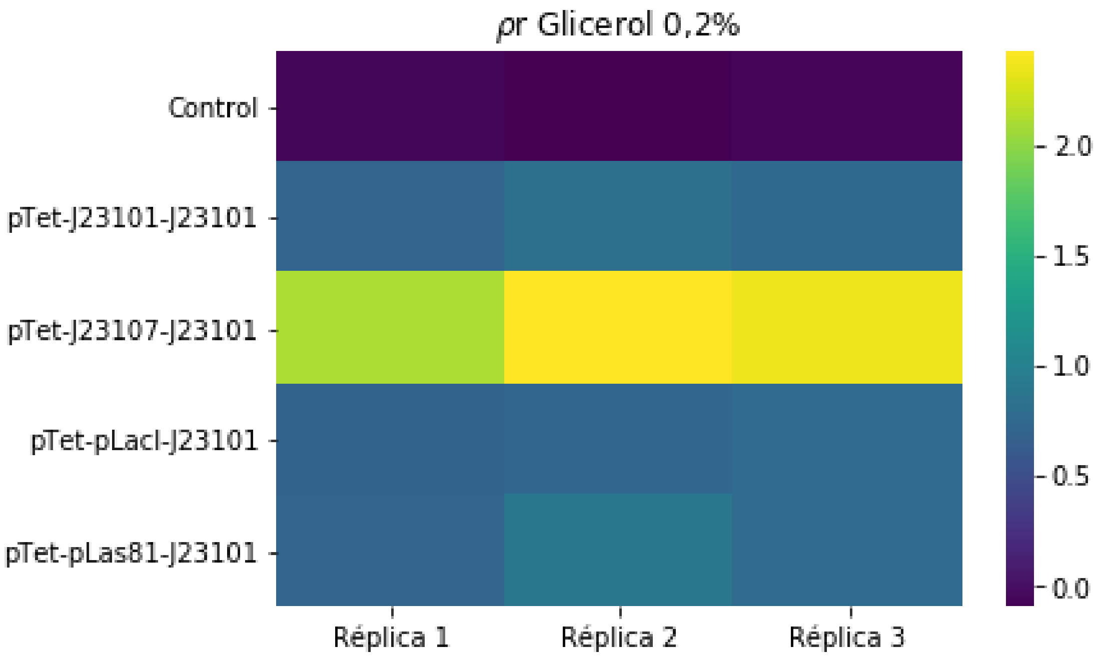


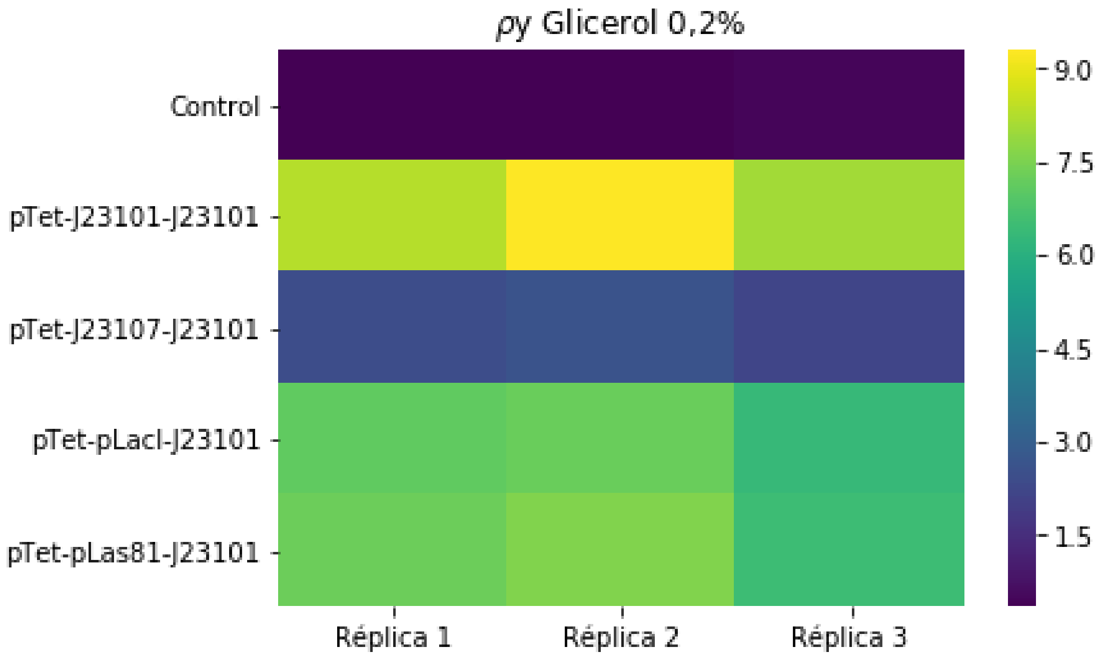




In [89]:
```
ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,
```

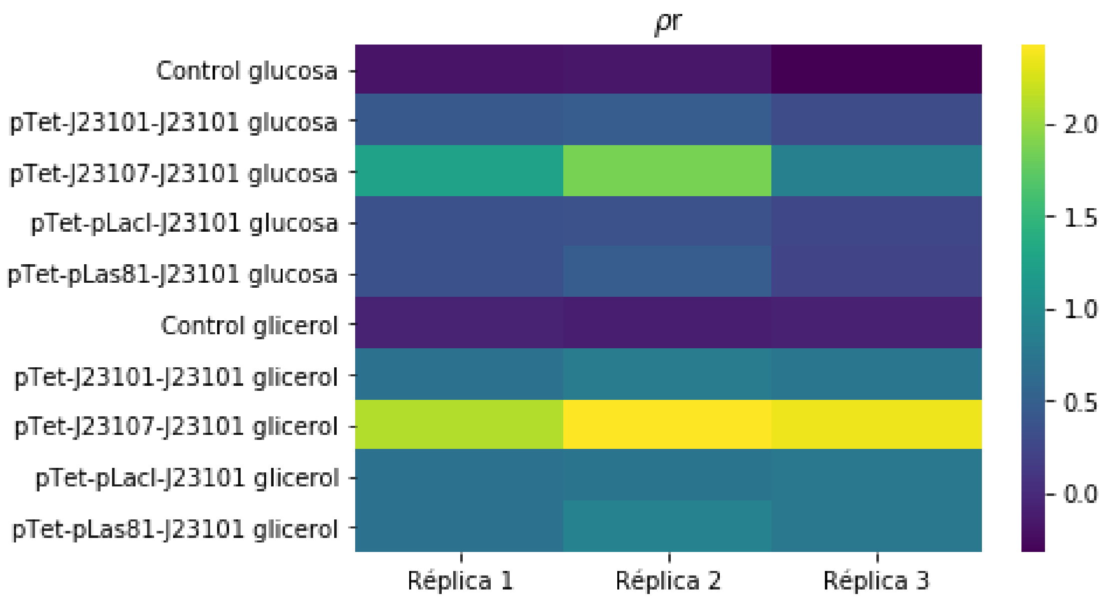
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','pTet-J23101-J23101','pTet-J23107-J23101','pTet-pLacI-J23101','pTet-p

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

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

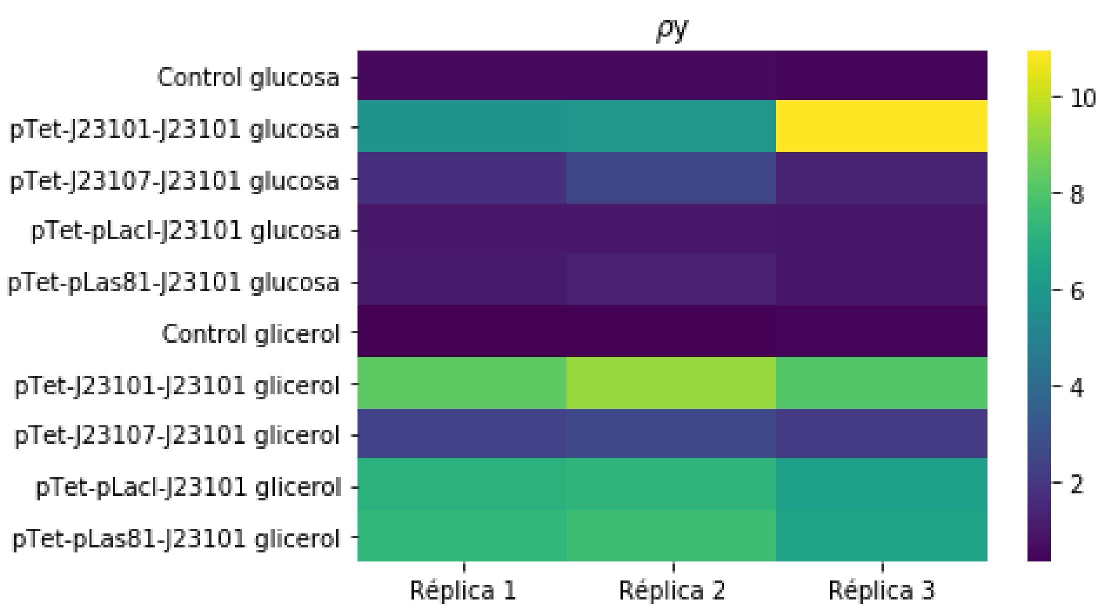
Out[89]: <matplotlib.axes._subplots.AxesSubplot at 0x267f155c6a0>






```
In [90]: ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
         ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr2
         bothrfp=[[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,py2
         ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,
         bothyfp=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py27]

         xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control glucosa','pTet-J23101-J23101 glucosa','pTet-J23107-J23101 glucosa','pT

         plt.figure()
         plt.title(r'$\rho$r')
         sns.heatmap(bothrfp,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

         plt.figure()
         plt.title(r'$\rho$y')
         sns.heatmap(bothyfp,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

Out[90]: <matplotlib.axes._subplots.AxesSubplot at 0x267f34698d0>
```

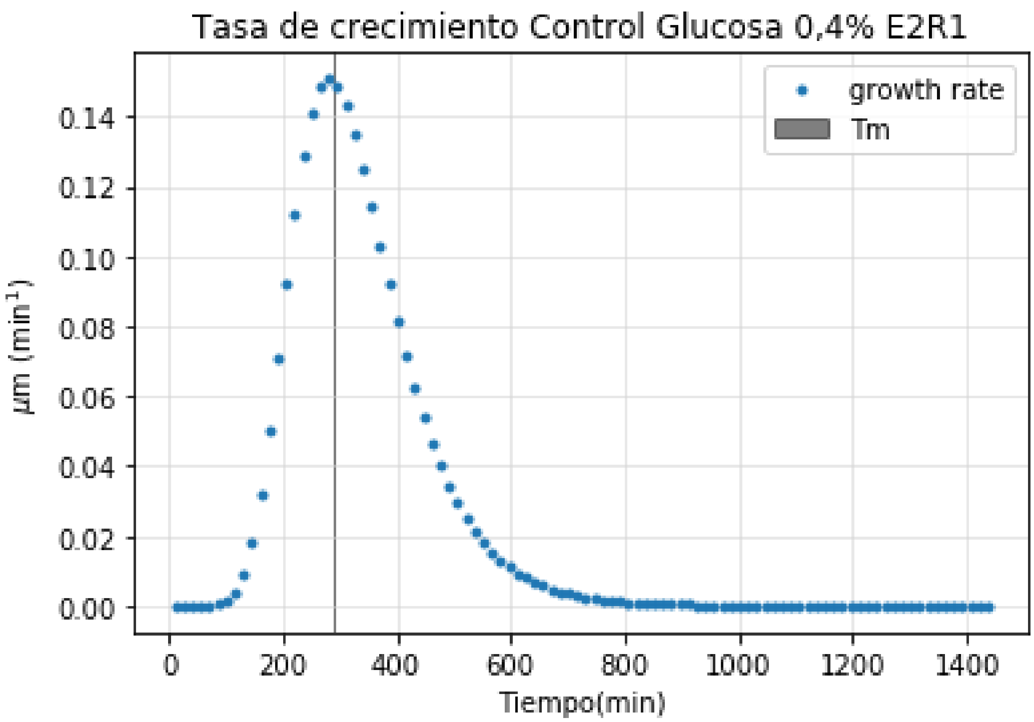



```
In [91]: #tasa de crecimiento
         ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(l1-tt))+1))))
         #Con diff
         dy1=(np.diff(ye1))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R1')
```

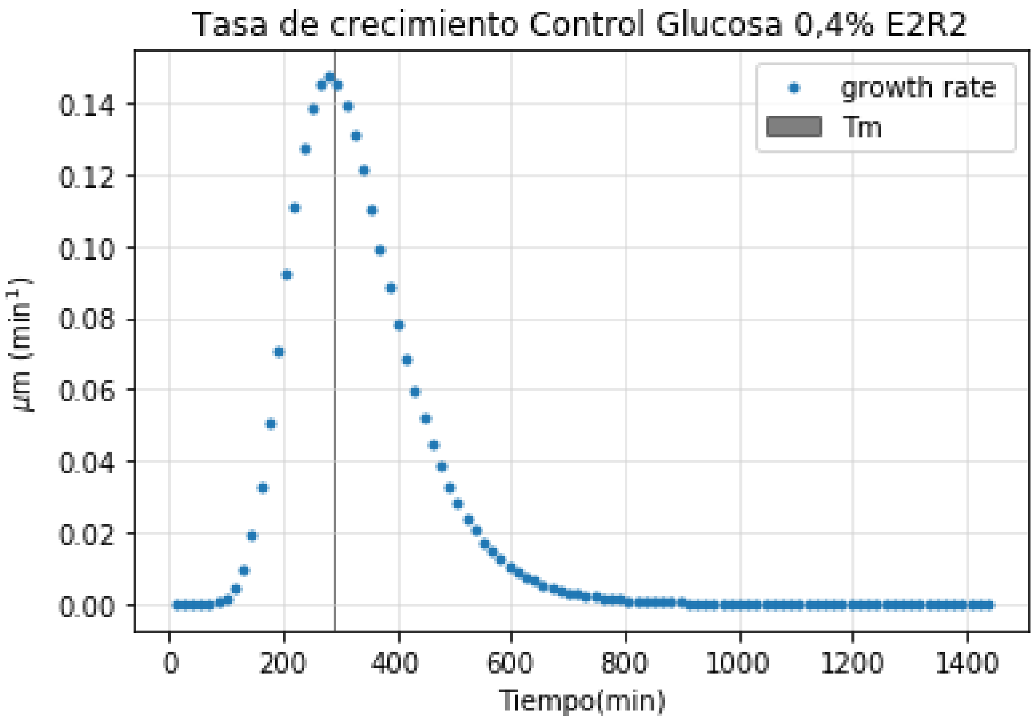
```
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy1,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[91]: <matplotlib.legend.Legend at 0x267f5087c18>

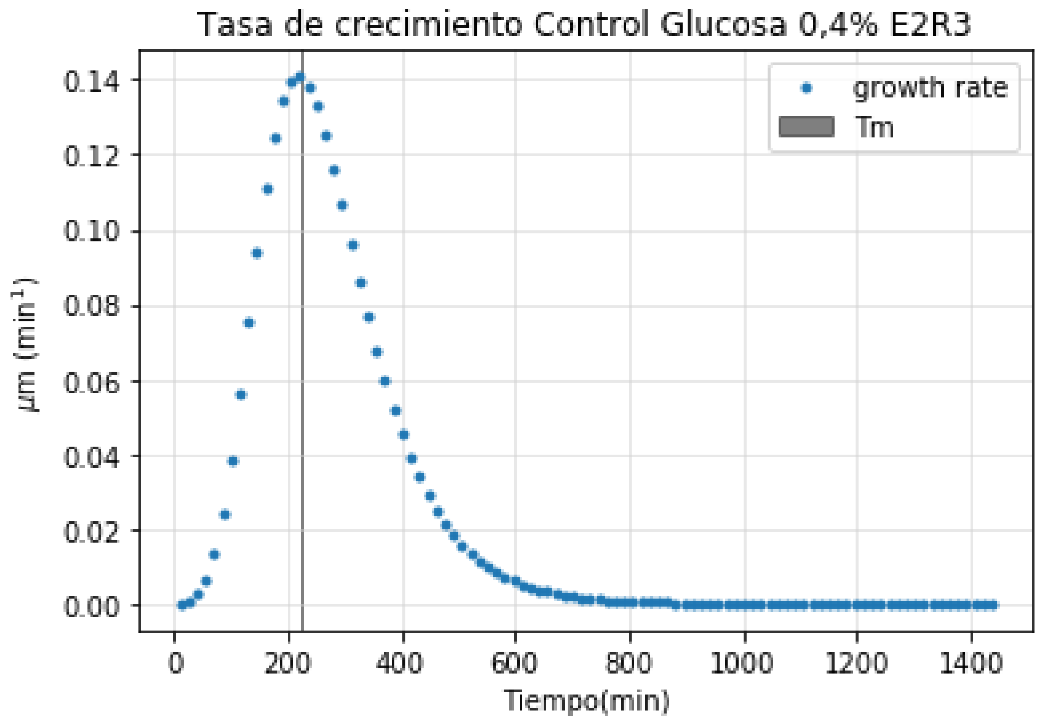



In [92]:
```
#tasa de crecimiento

ye2=((A2*np.exp(-np.exp((((um2*np.exp(1))/A2)*(l2-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'$\mu$m (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')
```

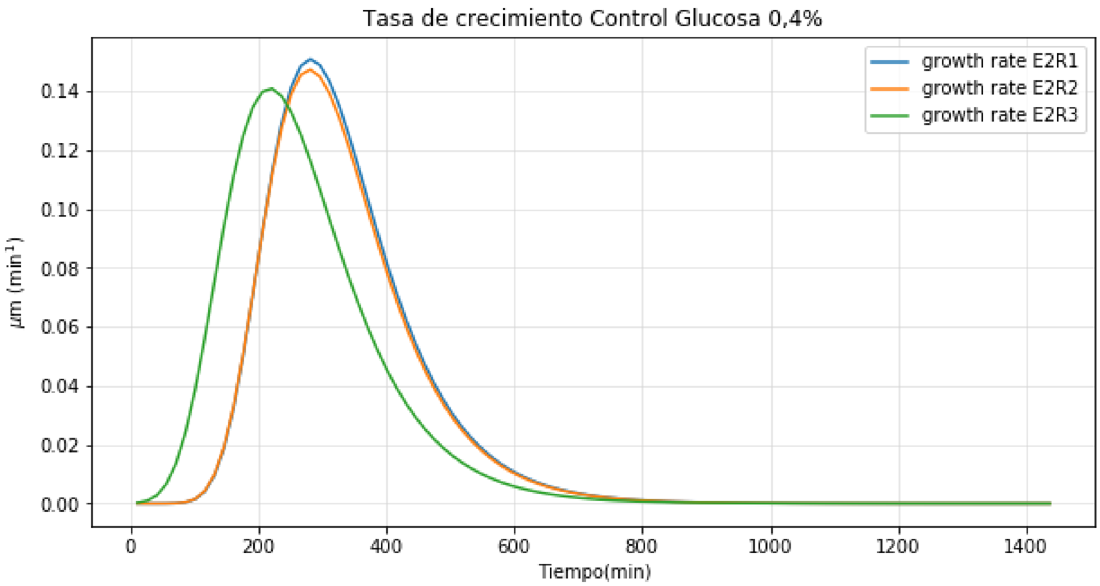
Out[92]: <matplotlib.legend.Legend at 0x267f5167d68>



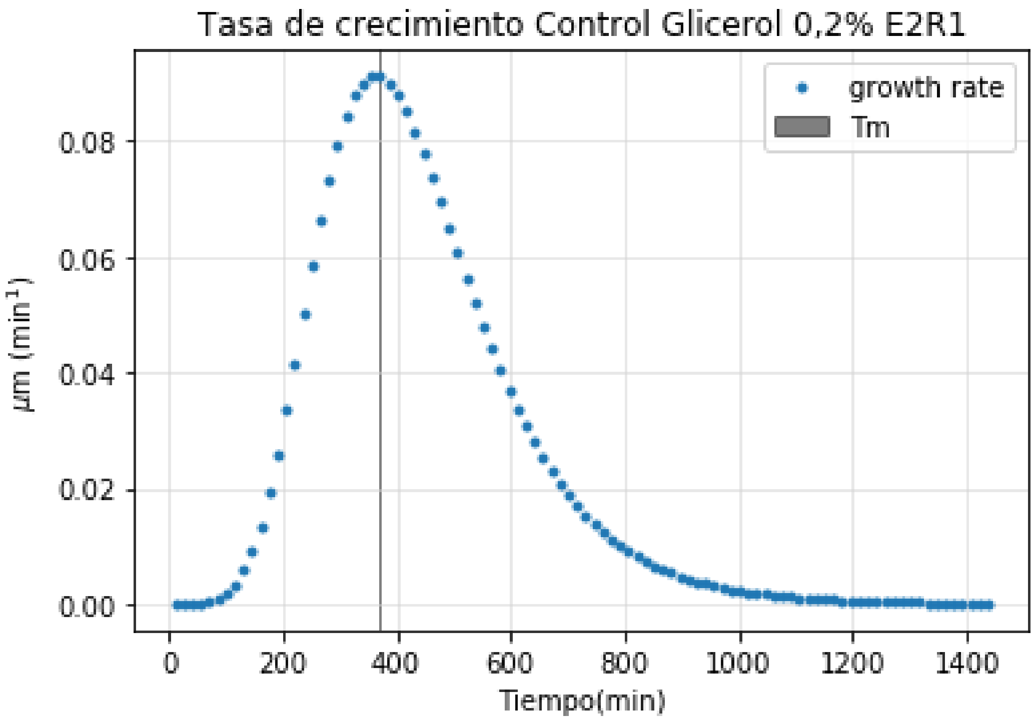

```
In [93]: #tasa de crecimiento
         ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(l3-tt))+1))))
         #Con diff
         dy3=(np.diff(ye3))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[93]: <matplotlib.legend.Legend at 0x267f5237d30>
```



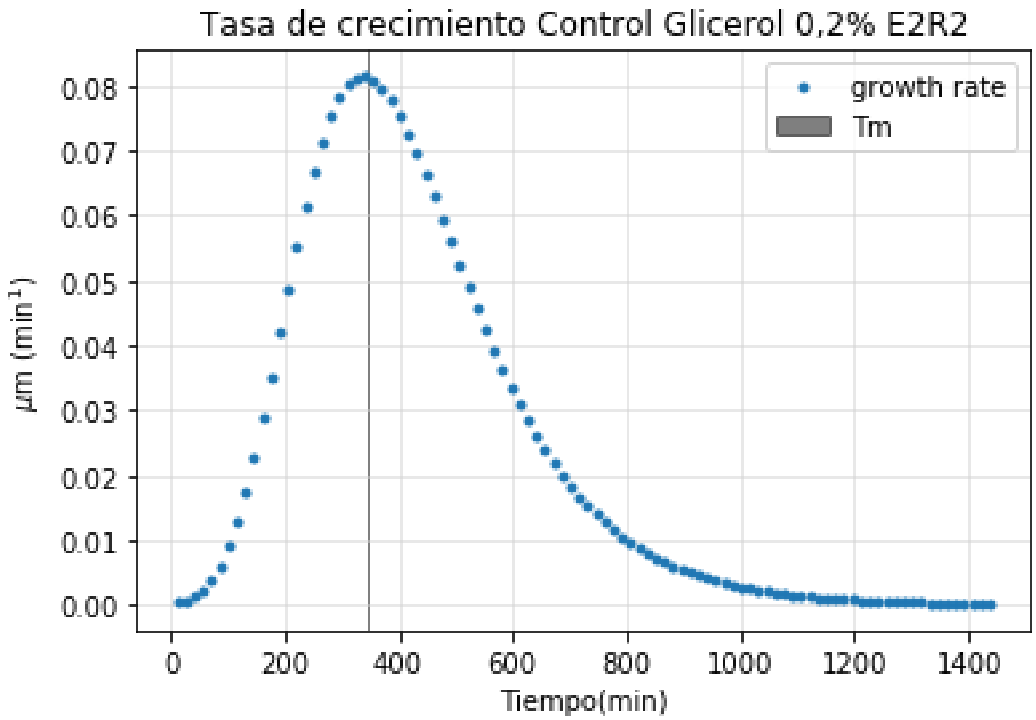

```
In [94]: #Tasas control réplicas glucosa
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy1,label='growth rate E2R1')
         plt.plot(tt[:-1],dy2,label='growth 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[94]: <matplotlib.legend.Legend at 0x267f39fa208>
```




```
In [95]: #tasa de crecimiento
         ye4=((A4*np.exp(-np.exp((((um4*np.exp(1))/A4)*(l4-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'$\mu$m (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[95]: <matplotlib.legend.Legend at 0x267f1964b38>
```

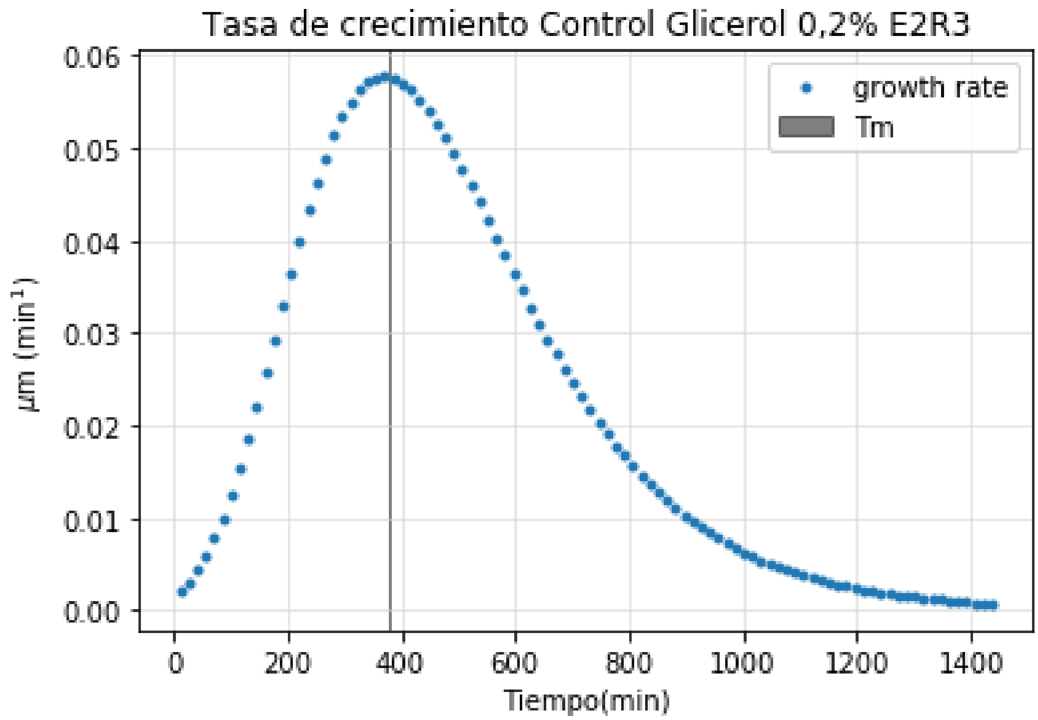



```
In [96]: #tasa de crecimiento
         ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(l5-tt))+1))))
         #Con diff
         dy5=(np.diff(ye5))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[96]: <matplotlib.legend.Legend at 0x267f3965390>
```

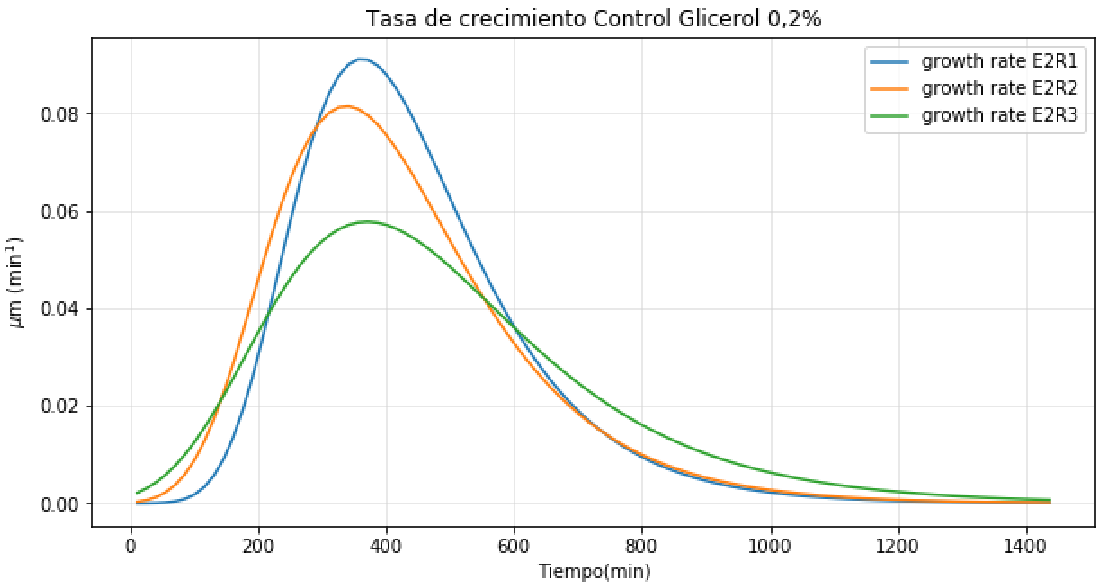



In [97]:
```
#tasa de crecimiento
ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(l6-tt))+1))))
#Con diff
dy6=(np.diff(ye6))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm6,tm6, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy6,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

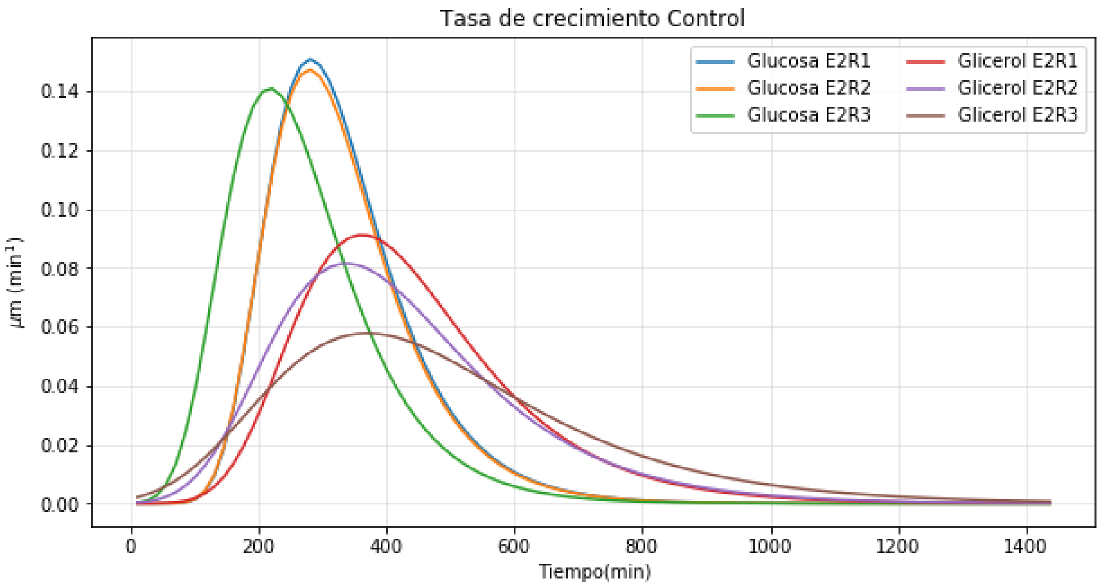
Out[97]: <matplotlib.legend.Legend at 0x267f1fd1b38>



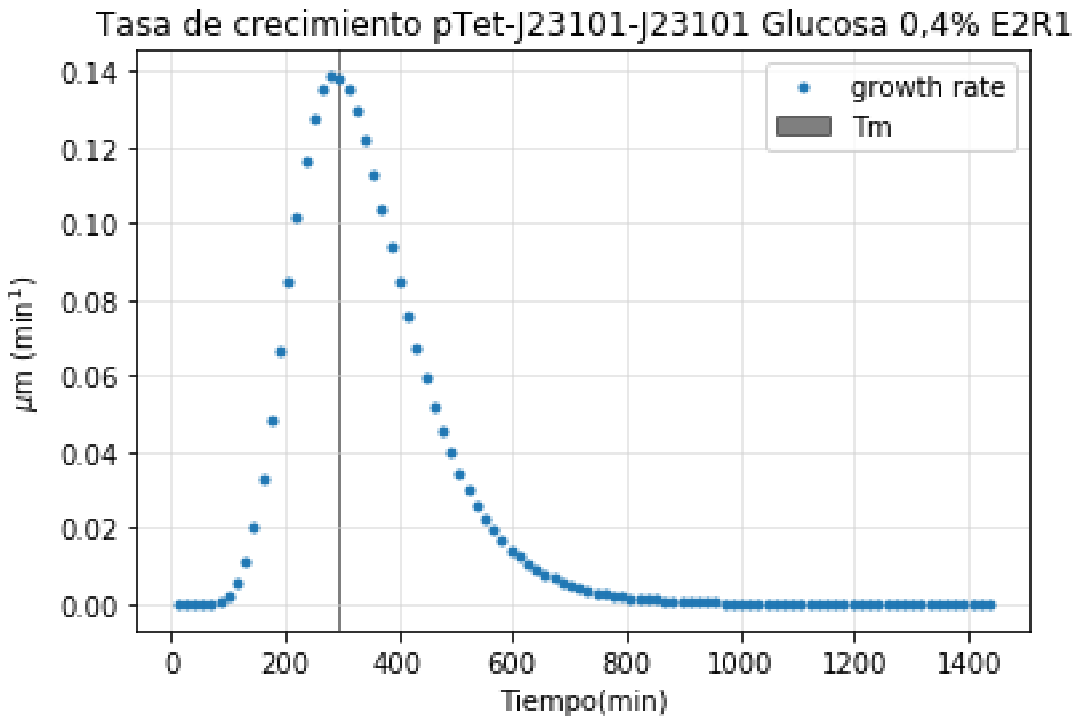

```
In [98]: #Tasas control réplicas glicerol
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy4,label='growth rate E2R1')
         plt.plot(tt[:-1],dy5,label='growth 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[98]: <matplotlib.legend.Legend at 0x267f409d550>
```



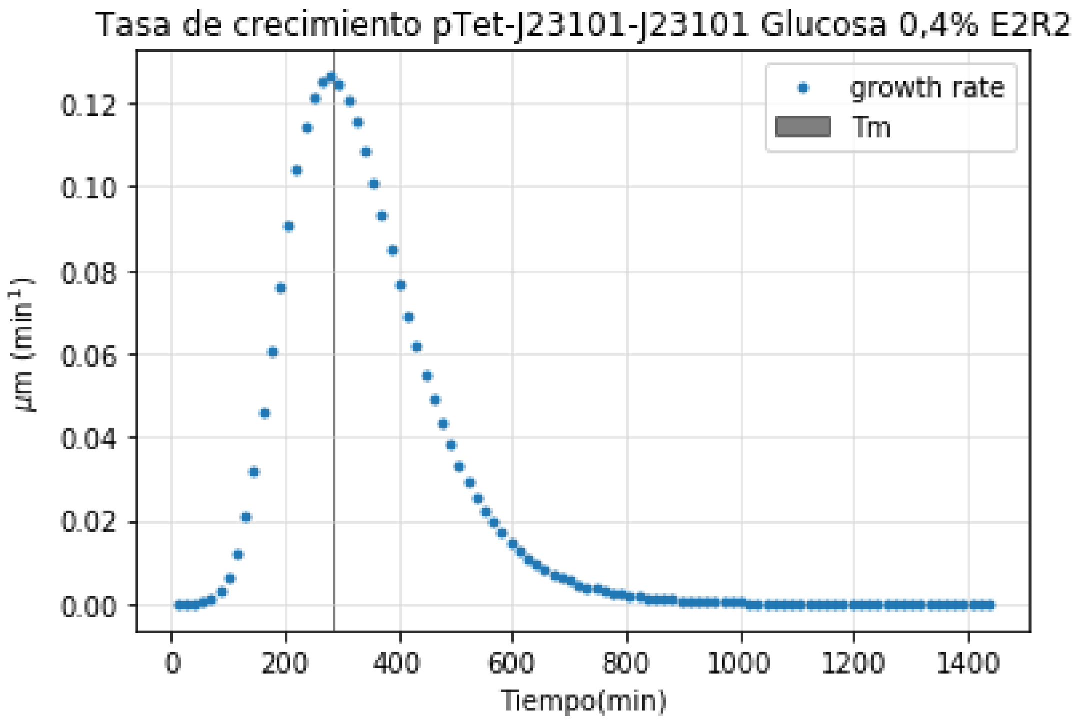

```
In [99]: #Tasas control réplicas controles
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy1,label='Glucosa E2R1')
         plt.plot(tt[:-1],dy2,label='Glucosa 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[99]: <matplotlib.legend.Legend at 0x267f3342978>
```




```
In [100]: #tasa de crecimiento
          ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(l7-tt))+1))))
          #Con diff
          dy7=(np.diff(ye7))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[100]: <matplotlib.legend.Legend at 0x267f3c8aa20>
```

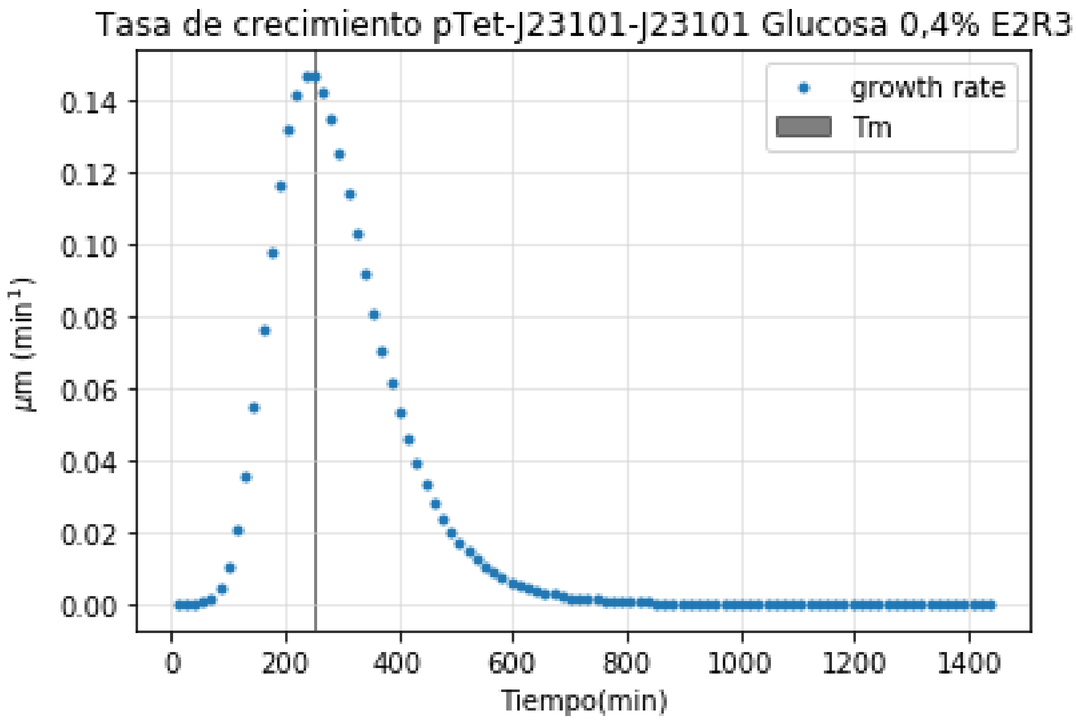



```
In [101]: #tasa de crecimiento
          ye8=((A8*np.exp(-np.exp((((um8*np.exp(1))/A8)*(l8-tt))+1))))
          #Con diff
          dy8=(np.diff(ye8))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[101]: <matplotlib.legend.Legend at 0x267f3d57cc0>
```

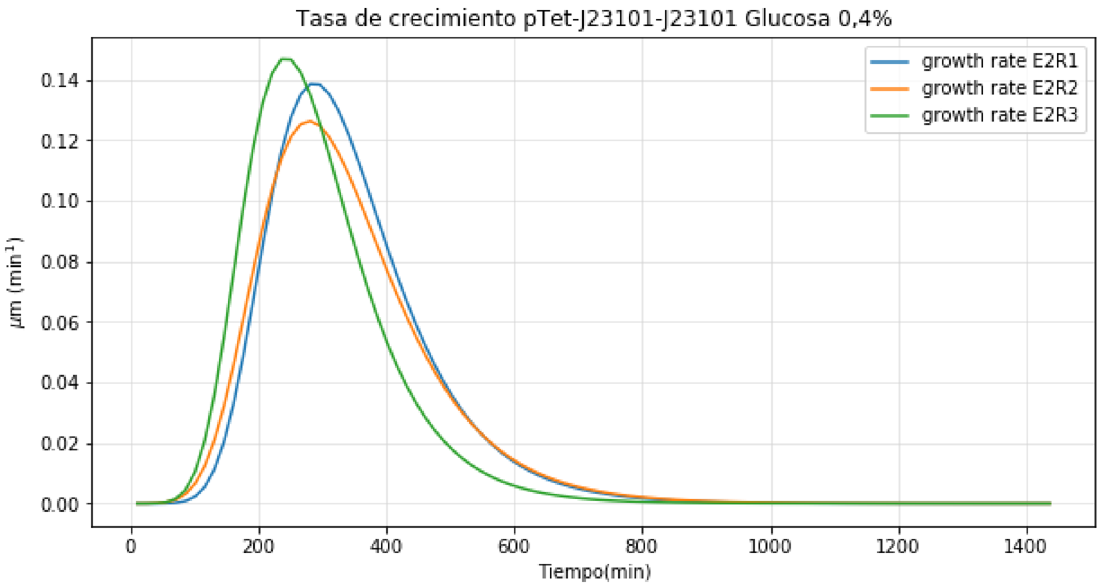



In [102]:
```
#tasa de crecimiento
ye9=((A9*np.exp(-np.exp((((um9*np.exp(1))/A9)*(l9-tt))+1))))
#Con diff
dy9=(np.diff(ye9))
plt.figure()
plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm9,tm9, color='k', alpha=0.5, 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')
```

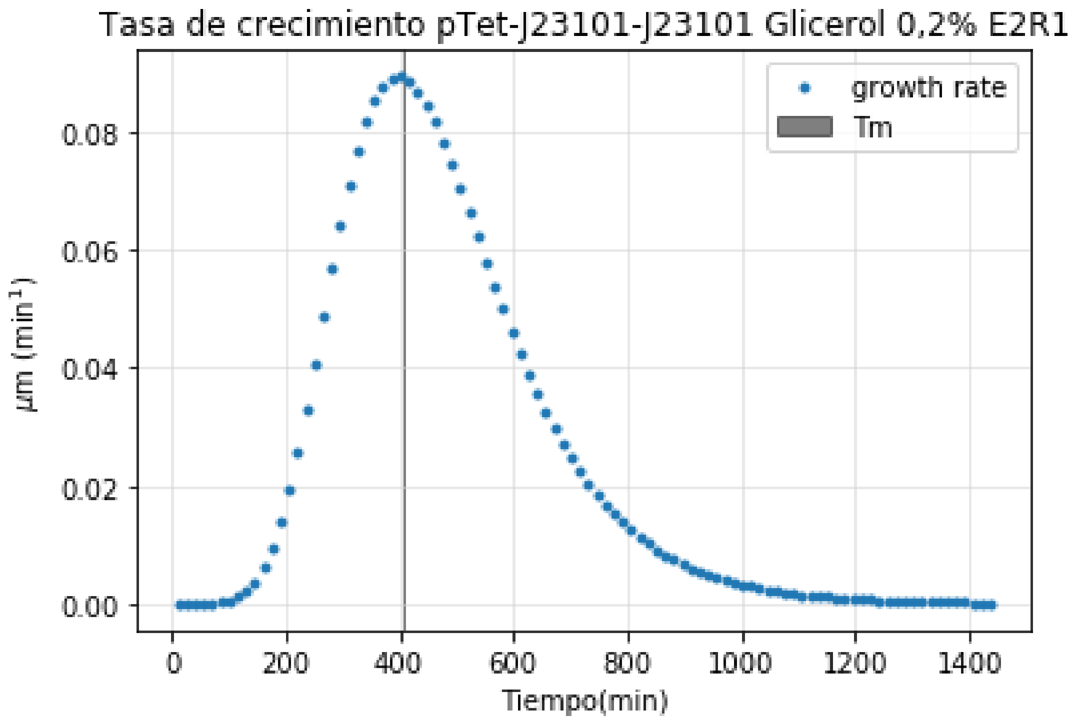
Out[102]: <matplotlib.legend.Legend at 0x267f46e3860>



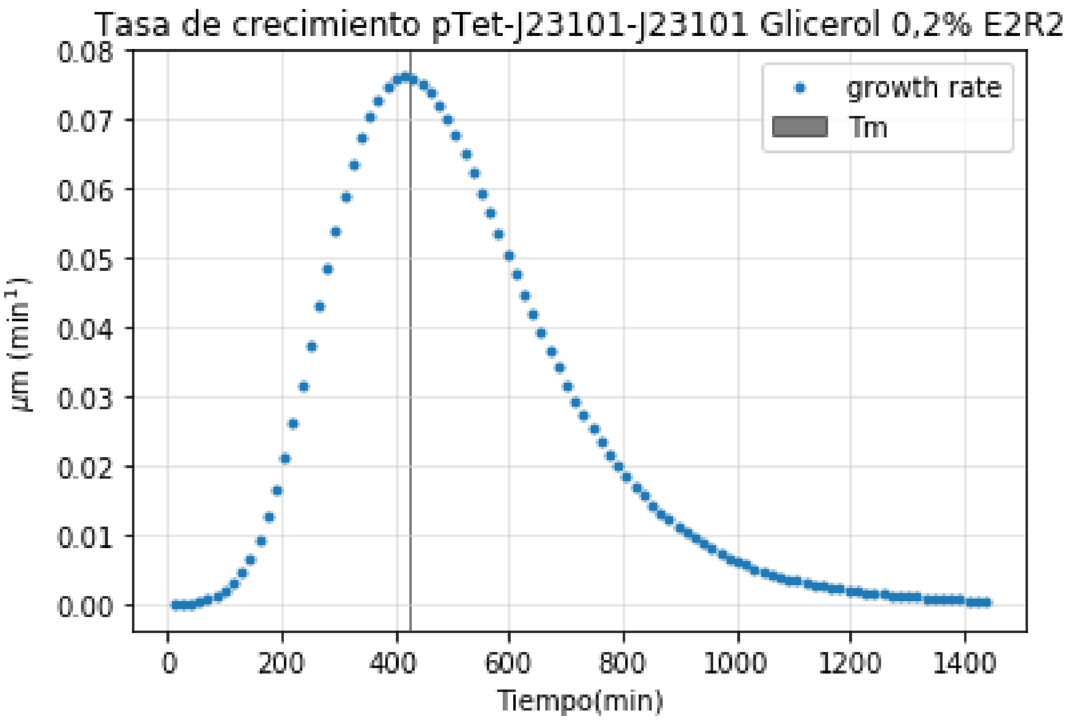

```
In [103]: #Tasas pLux-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='growth rate E2R1')
          plt.plot(tt[:-1],dy8,label='growth rate E2R2')
          plt.plot(tt[:-1],dy9,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[103]: <matplotlib.legend.Legend at 0x267f49397f0>
```



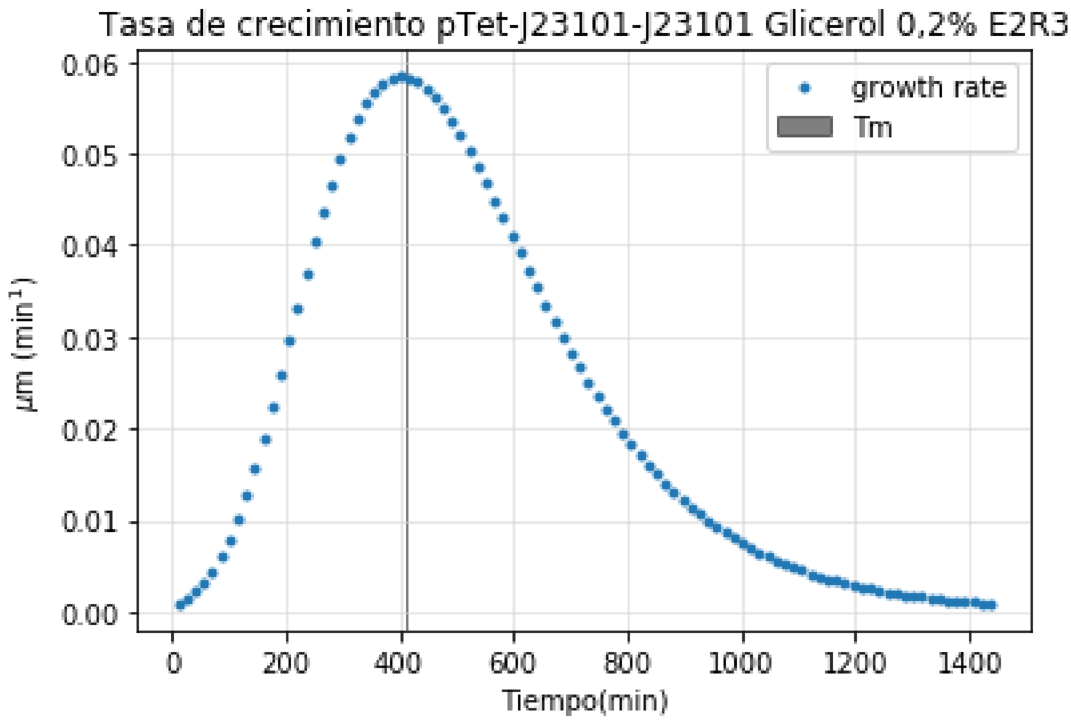

```
In [104]: #tasa de crecimiento
          ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(l10-tt))+1))))
          #Con diff
          dy10=(np.diff(ye10))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[104]: <matplotlib.legend.Legend at 0x267f4a1ed30>
```



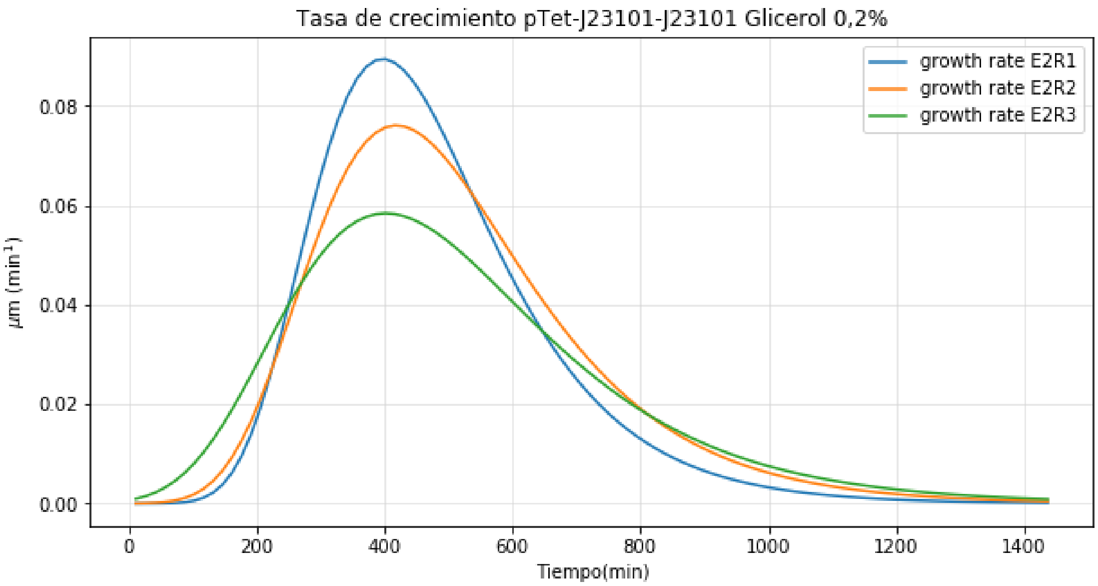

```
In [105]: #tasa de crecimiento
          ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(l11-tt))+1))))
          #Con diff
          dy11=(np.diff(ye11))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[105]: <matplotlib.legend.Legend at 0x267f49afa20>
```



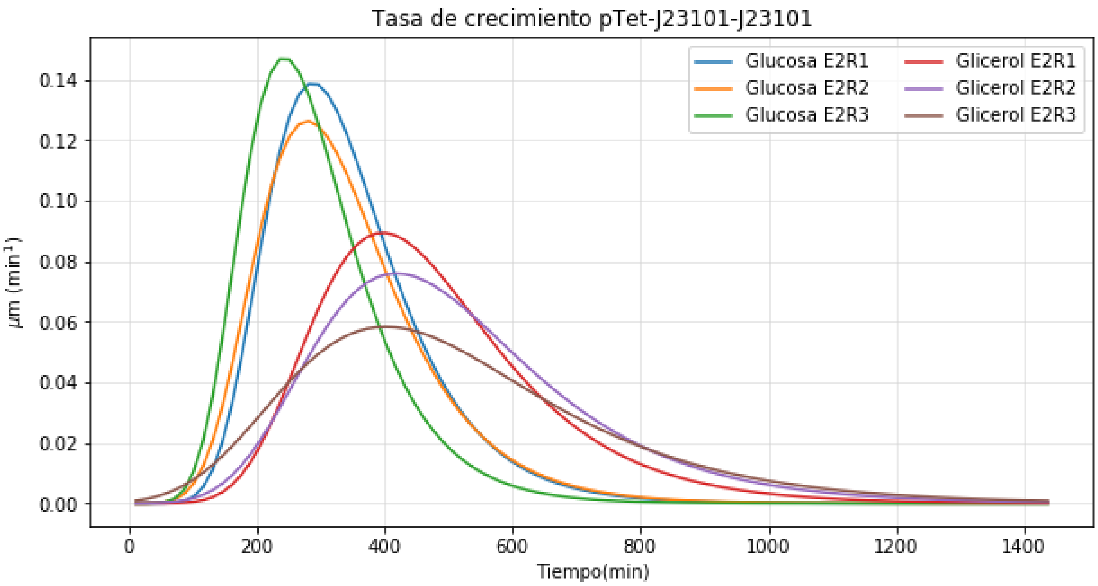

```
In [106]: #tasa de crecimiento
          ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(l12-tt))+1))))
          #Con diff
          dy12=(np.diff(ye12))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[106]: <matplotlib.legend.Legend at 0x267f4bc1dd8>
```



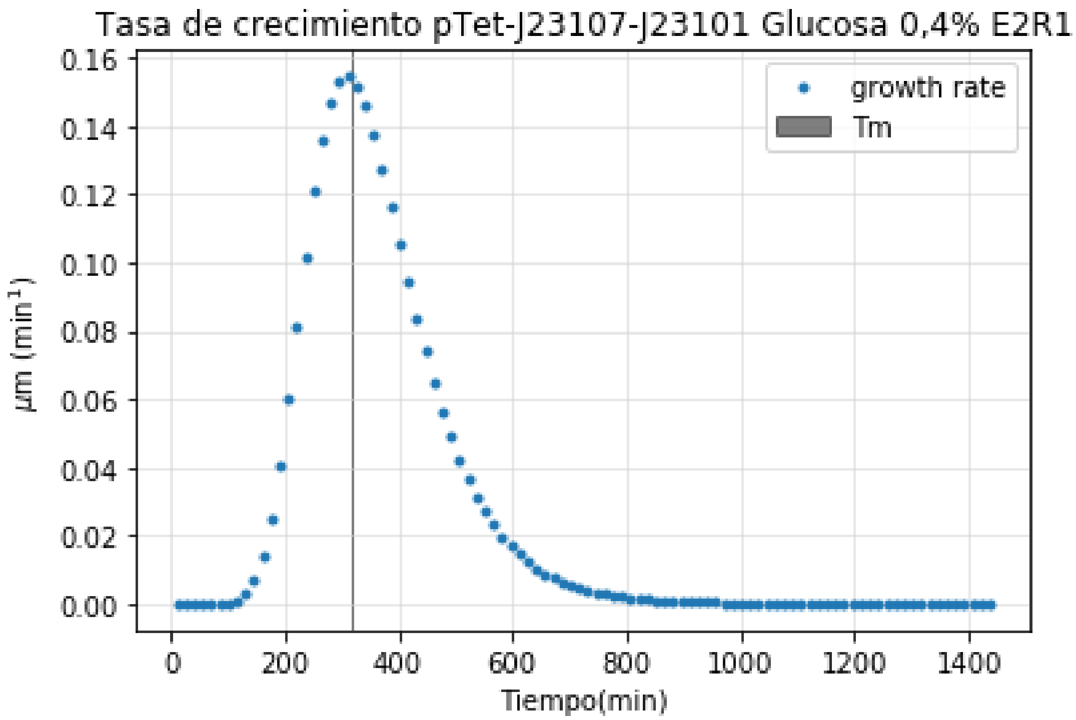

```
In [107]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy10,label='growth rate E2R1')
          plt.plot(tt[:-1],dy11,label='growth rate E2R2')
          plt.plot(tt[:-1],dy12,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[107]: <matplotlib.legend.Legend at 0x267f4c94b70>
```




```
In [108]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='Glucosa E2R1')
          plt.plot(tt[:-1],dy8,label='Glucosa E2R2')
          plt.plot(tt[:-1],dy9,label='Glucosa E2R3')
          plt.plot(tt[:-1],dy10,label='Glicerol E2R1')
          plt.plot(tt[:-1],dy11,label='Glicerol E2R2')
          plt.plot(tt[:-1],dy12,label='Glicerol E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[108]: <matplotlib.legend.Legend at 0x267f5353780>
```

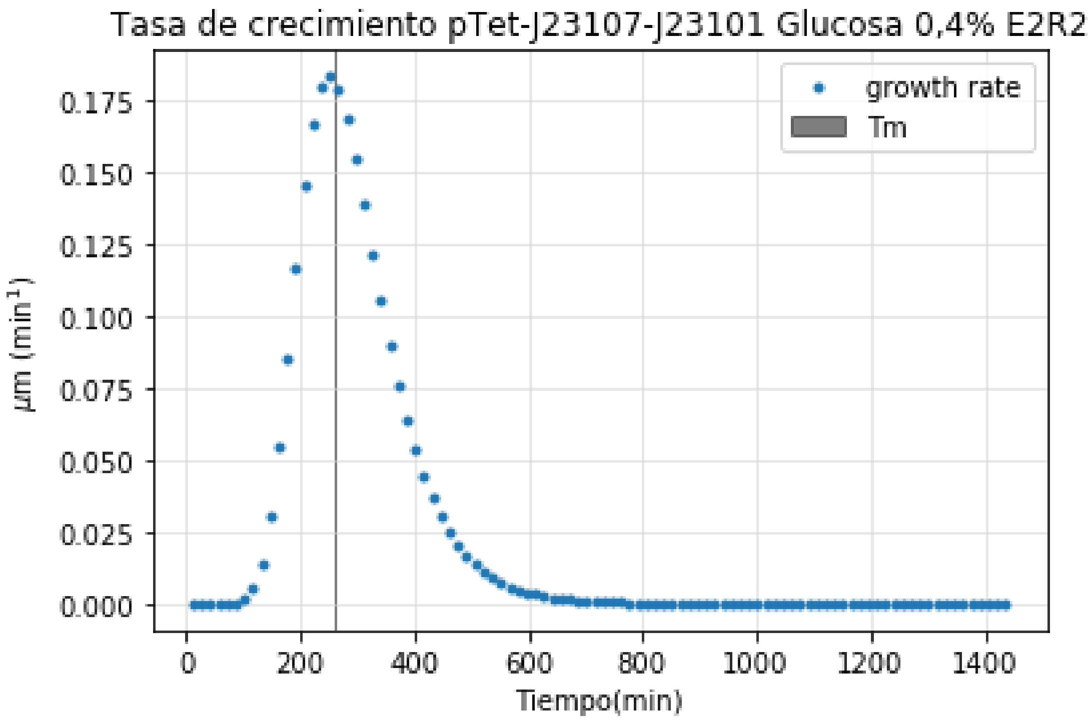



```
In [109]: #tasa de crecimiento
          ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(l13-tt))+1))))
          #Con diff
          dy13=(np.diff(ye13))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glucosa 0,4% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[109]: <matplotlib.legend.Legend at 0x267f542d940>
```

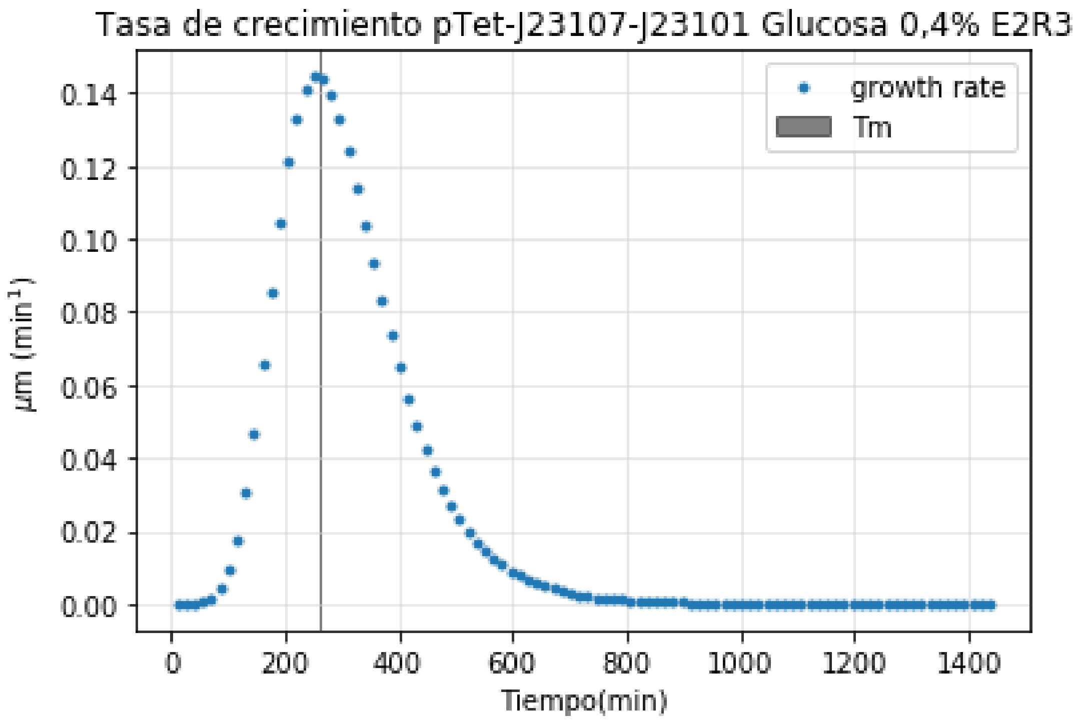



In [110]:
```
#tasa de crecimiento
ye14=((A14*np.exp(-np.exp((((um14*np.exp(1))/A14)*(l14-tt))+1))))
#Con diff
dy14=(np.diff(ye14))
plt.figure()
plt.title('Tasa de crecimiento pTet-J23107-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm14,tm14, color='k', 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')
```

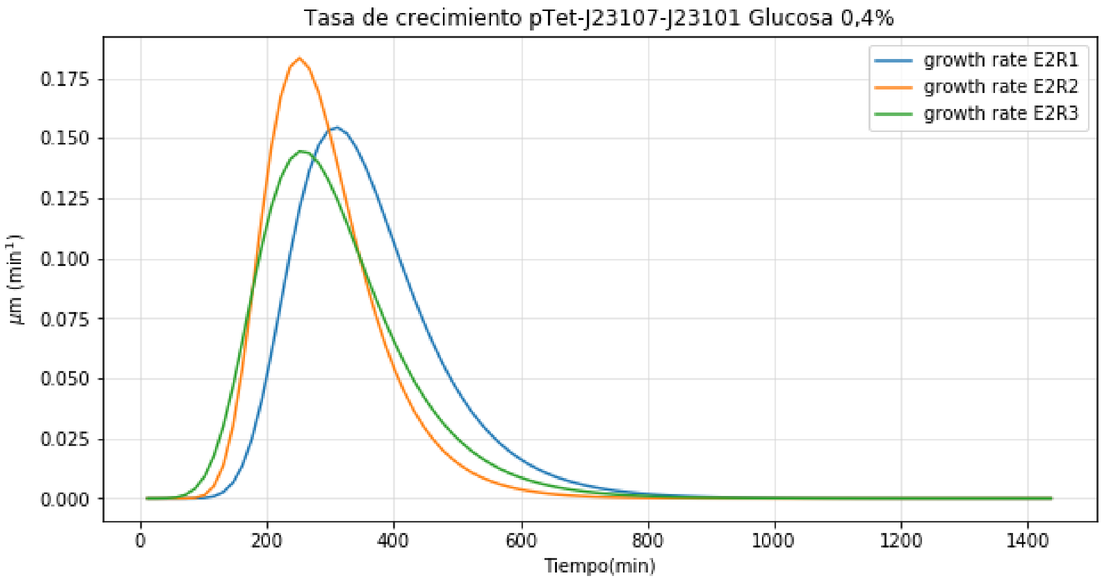
Out[110]:
```
<matplotlib.legend.Legend at 0x267f5505e80>
```



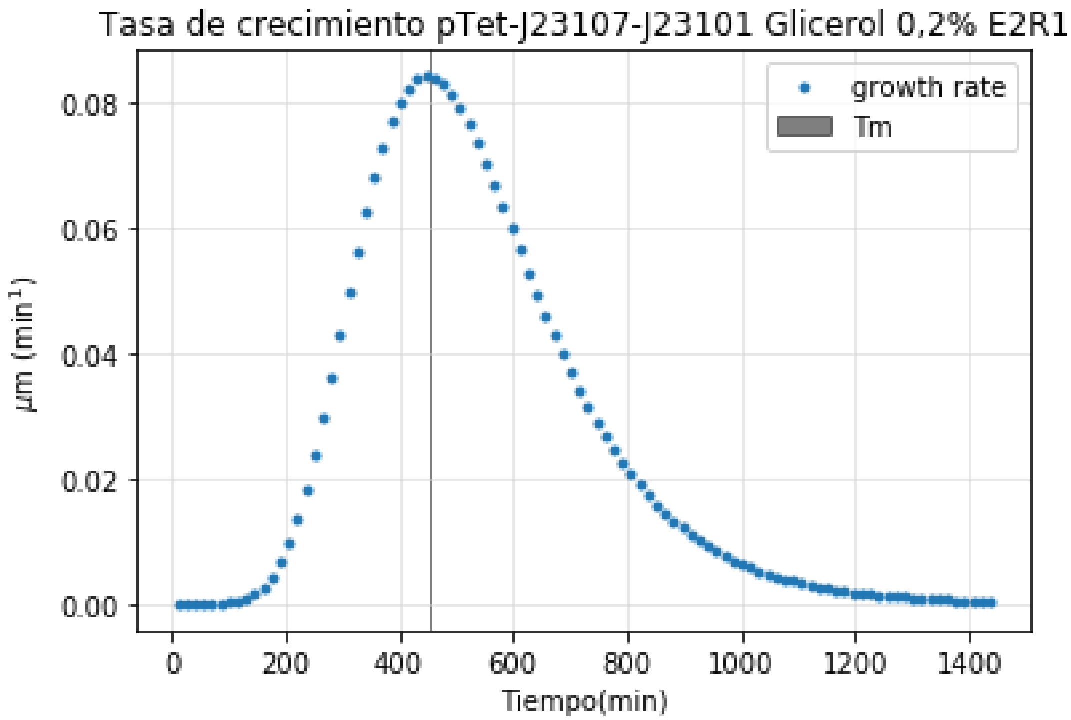

```
In [111]: #tasa de crecimiento
          ye15=((A15*np.exp(-np.exp((((um15*np.exp(1))/A15)*(l15-tt))+1))))
          #Con diff
          dy15=(np.diff(ye15))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glucosa 0,4% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[111]: <matplotlib.legend.Legend at 0x267f55e6d68>
```



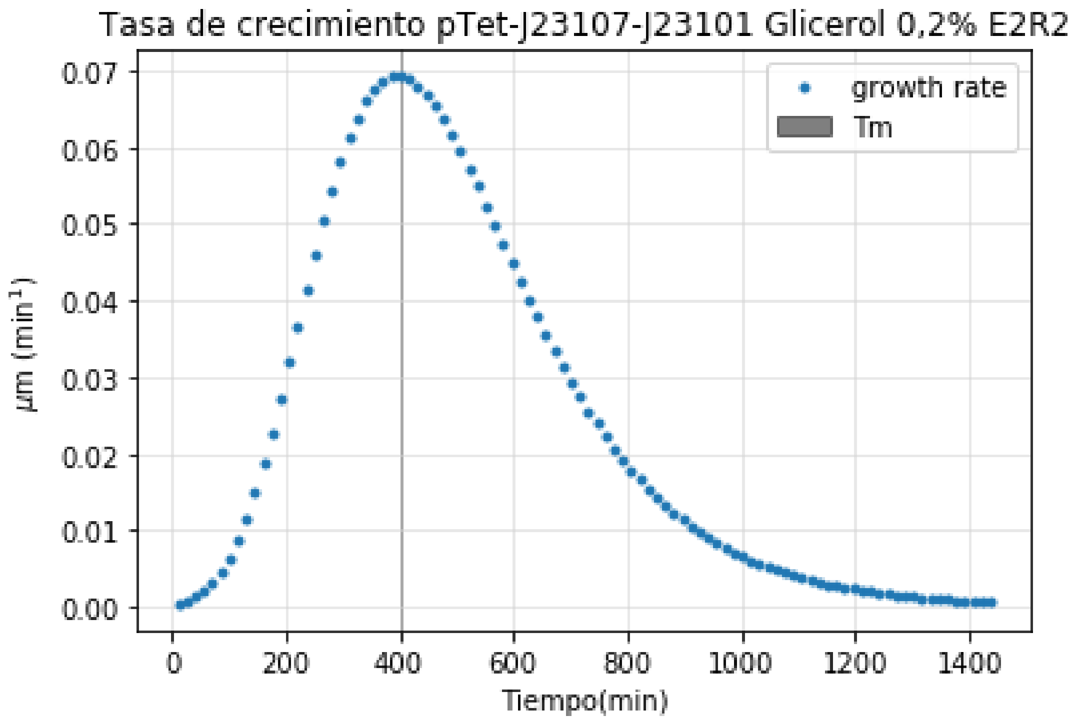

```
In [112]: #Tasas plux76-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='growth rate E2R1')
          plt.plot(tt[:-1],dy14,label='growth rate E2R2')
          plt.plot(tt[:-1],dy15,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[112]: <matplotlib.legend.Legend at 0x267f56b6cc0>
```



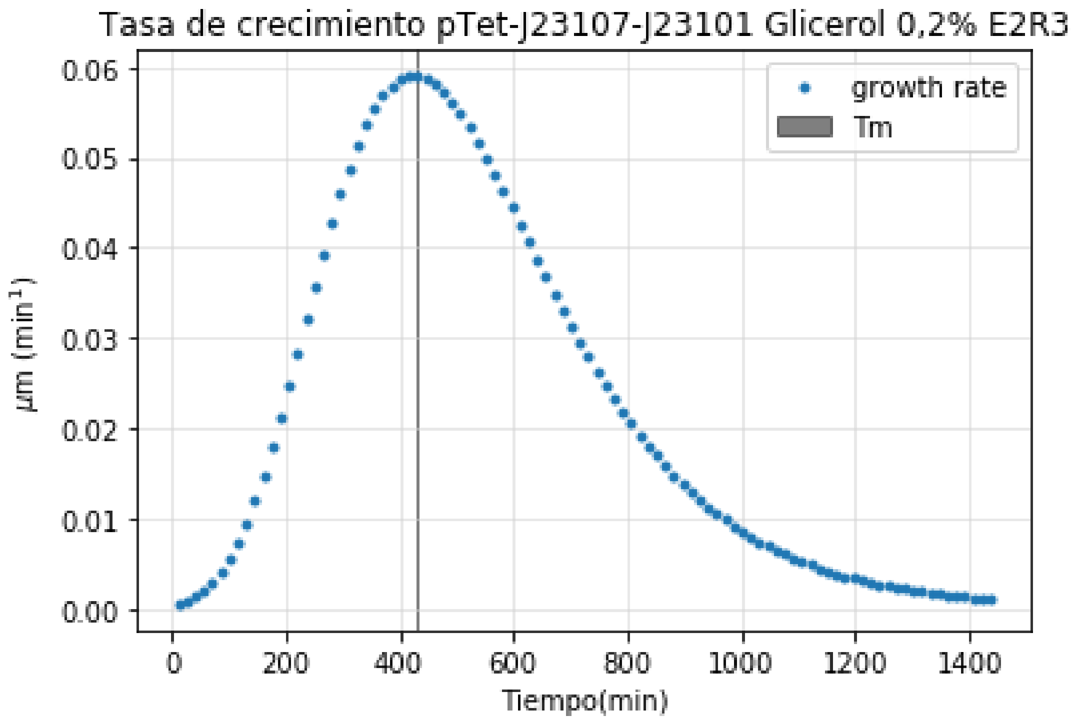

```
In [113]: #tasa de crecimiento
          ye16=((A16*np.exp(-np.exp((((um16*np.exp(1))/A16)*(l16-tt))+1))))
          #Con diff
          dy16=(np.diff(ye16))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glicerol 0,2% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[113]: <matplotlib.legend.Legend at 0x267f77b9ac8>
```




```
In [114]: #tasa de crecimiento
          ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(l17-tt))+1))))
          #Con diff
          dy17=(np.diff(ye17))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[114]: <matplotlib.legend.Legend at 0x267f788aeb8>
```

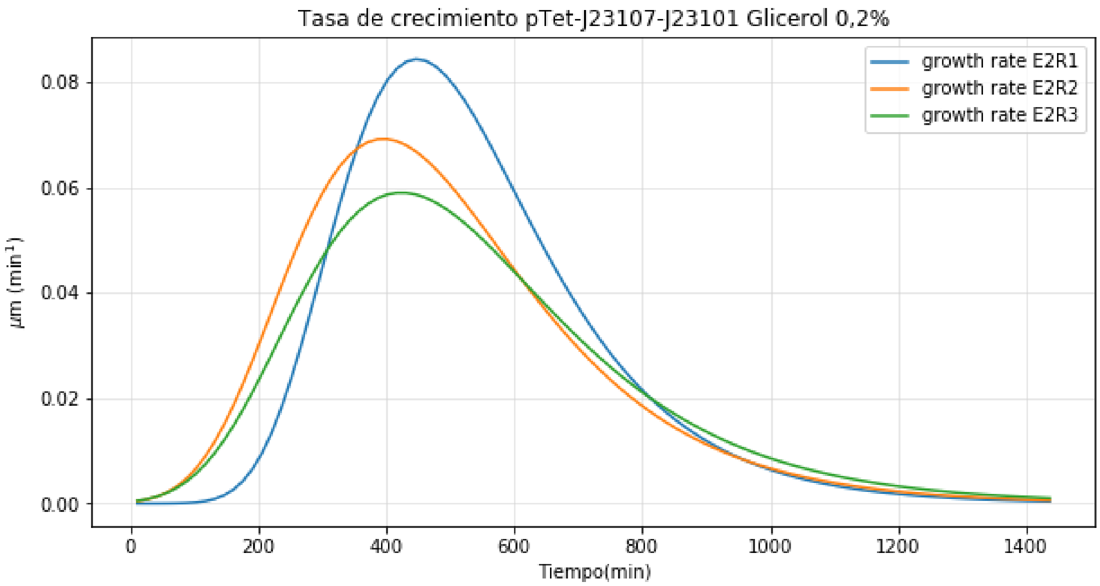



```
In [115]: #tasa de crecimiento
          ye18=((A18*np.exp(-np.exp((((um18*np.exp(1))/A18)*(l18-tt))+1))))
          #Con diff
          dy18=(np.diff(ye18))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23107-J23101 Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[115]: <matplotlib.legend.Legend at 0x267f4cee048>
```



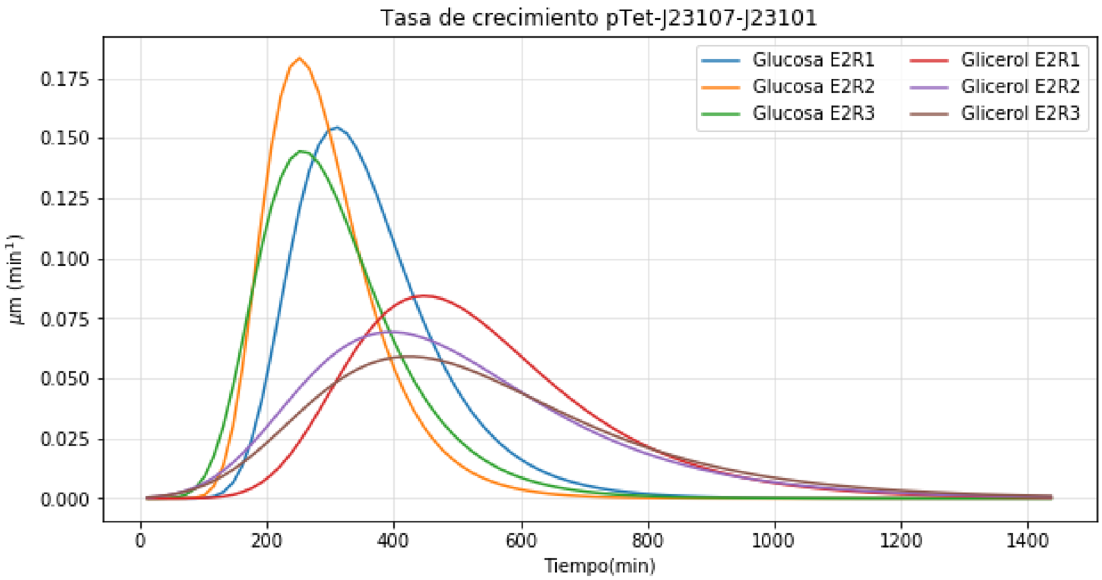

In [116]:
```
#Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-J23107-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy16,label='growth rate E2R1')
plt.plot(tt[:-1],dy17,label='growth rate E2R2')
plt.plot(tt[:-1],dy18,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[116]:
```
<matplotlib.legend.Legend at 0x267f44743c8>
```

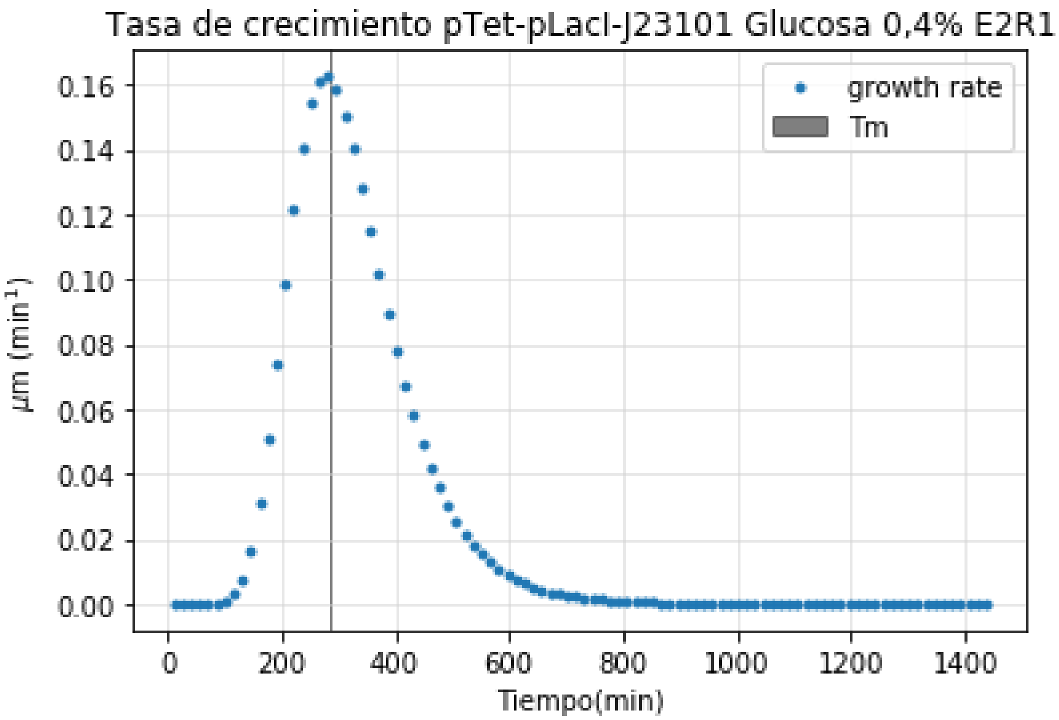



In [117]:
```
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-J23107-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy13,label='Glucosa E2R1')
plt.plot(tt[:-1],dy14,label='Glucosa E2R2')
plt.plot(tt[:-1],dy15,label='Glucosa E2R3')
plt.plot(tt[:-1],dy16,label='Glicerol E2R1')
plt.plot(tt[:-1],dy17,label='Glicerol E2R2')
plt.plot(tt[:-1],dy18,label='Glicerol E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

Out[117]:
```
<matplotlib.legend.Legend at 0x267f792ee10>
```

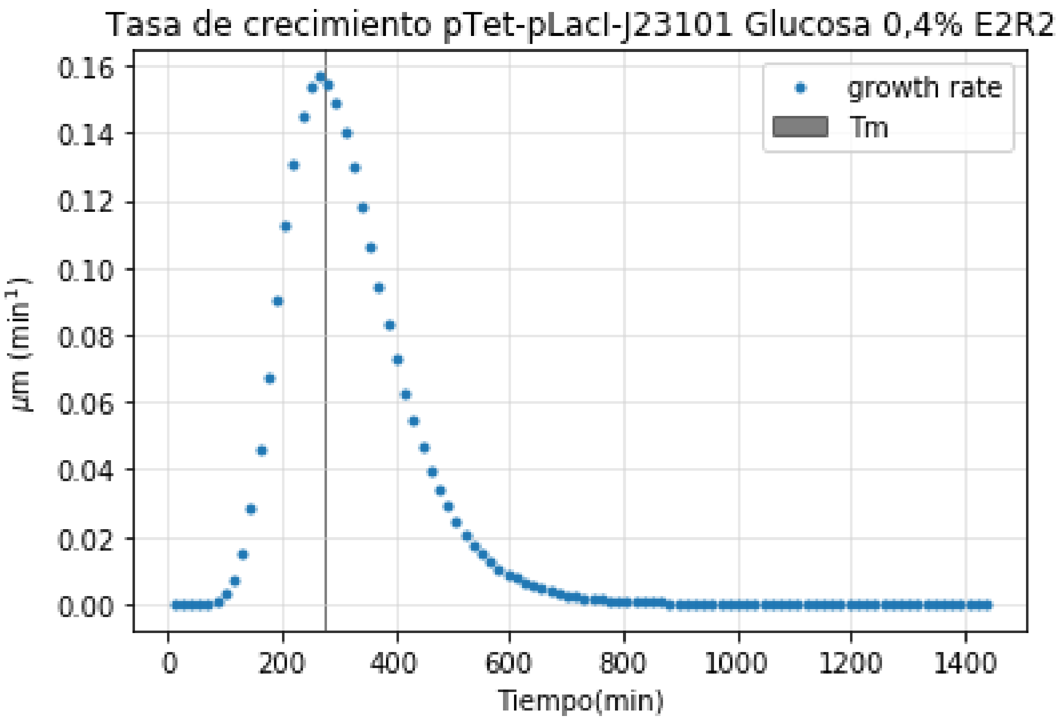



```
In [118]: #tasa de crecimiento
          ye19=((A19*np.exp(-np.exp((((um19*np.exp(1))/A19)*(l19-tt))+1))))
          #Con diff
          dy19=(np.diff(ye19))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glucosa 0,4% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[118]: <matplotlib.legend.Legend at 0x267f79d4f60>
```

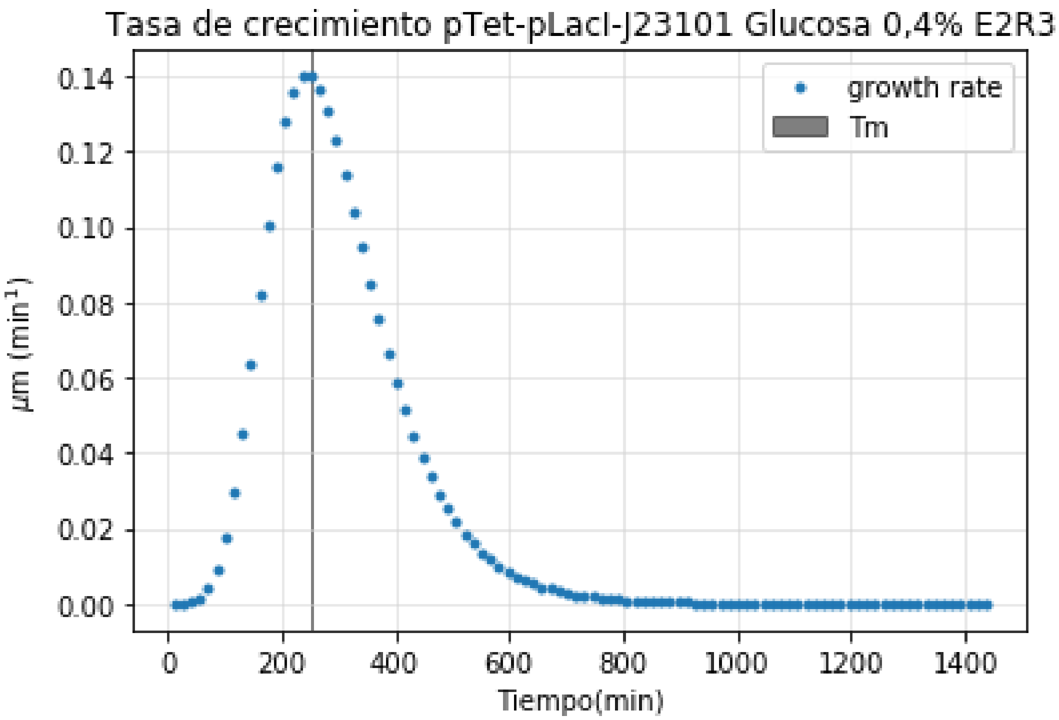



In [119]:
```
#tasa de crecimiento
ye20=((A20*np.exp(-np.exp((((um20*np.exp(1))/A20)*(l20-tt))+1))))
#Con diff
dy20=(np.diff(ye20))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm20,tm20, color='k', 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')
```

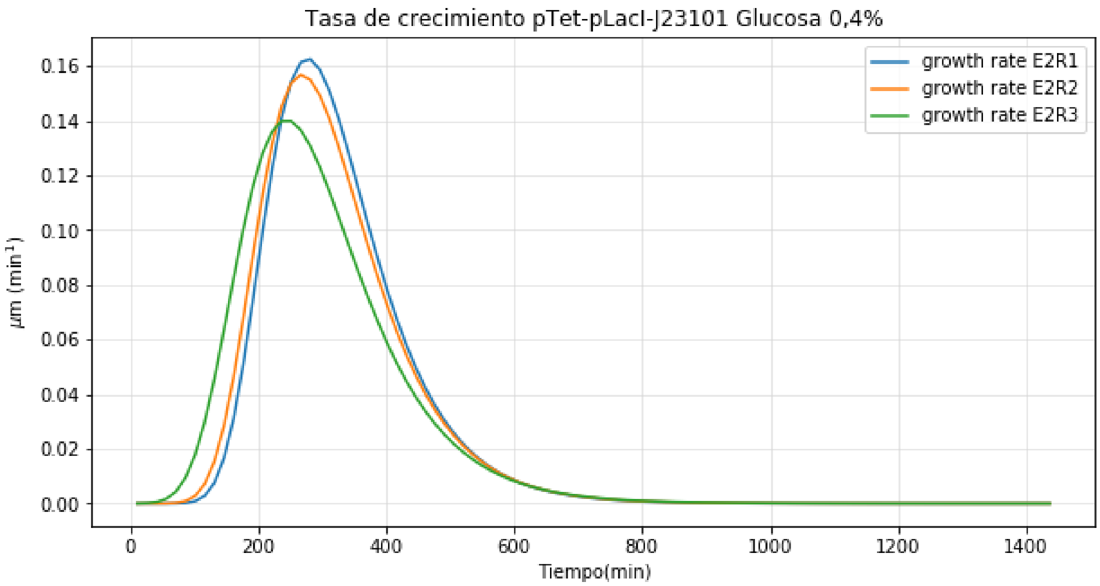
Out[119]:
```
<matplotlib.legend.Legend at 0x267f7aa5b00>
```



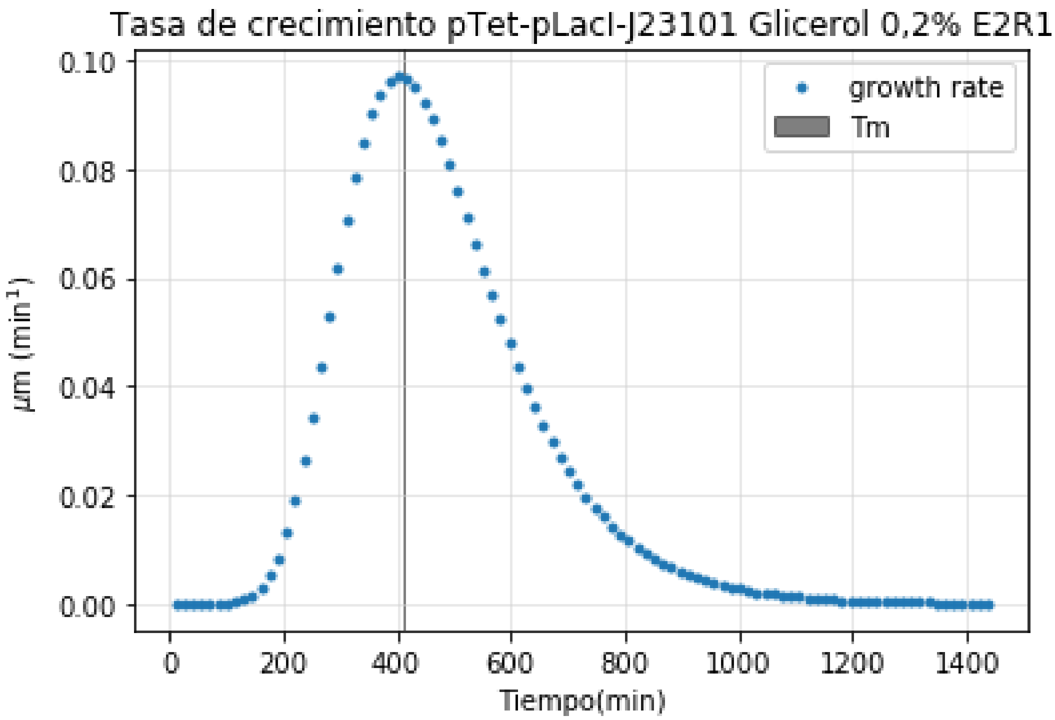

```
In [120]: #tasa de crecimiento
          ye21=((A21*np.exp(-np.exp((((um21*np.exp(1))/A21)*(l21-tt))+1))))
          #Con diff
          dy21=(np.diff(ye21))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glucosa 0,4% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[120]: <matplotlib.legend.Legend at 0x267f7b3beb8>
```



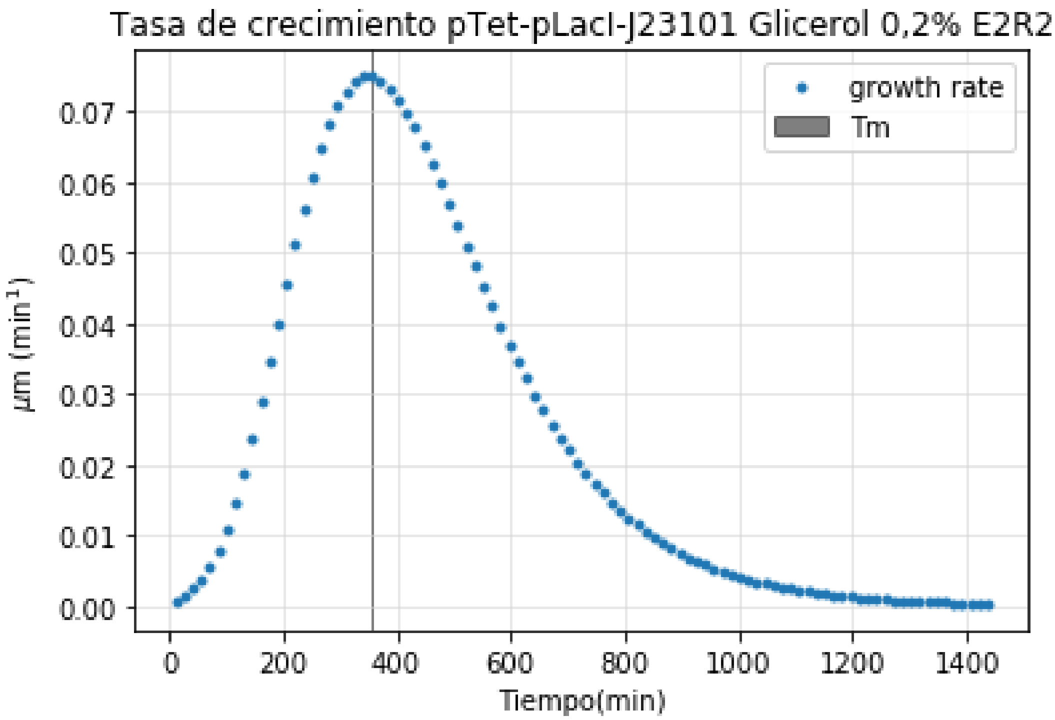

```
In [121]: #Tasas plux76-ptet-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='growth rate E2R1')
          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[121]: <matplotlib.legend.Legend at 0x267f7c389e8>
```



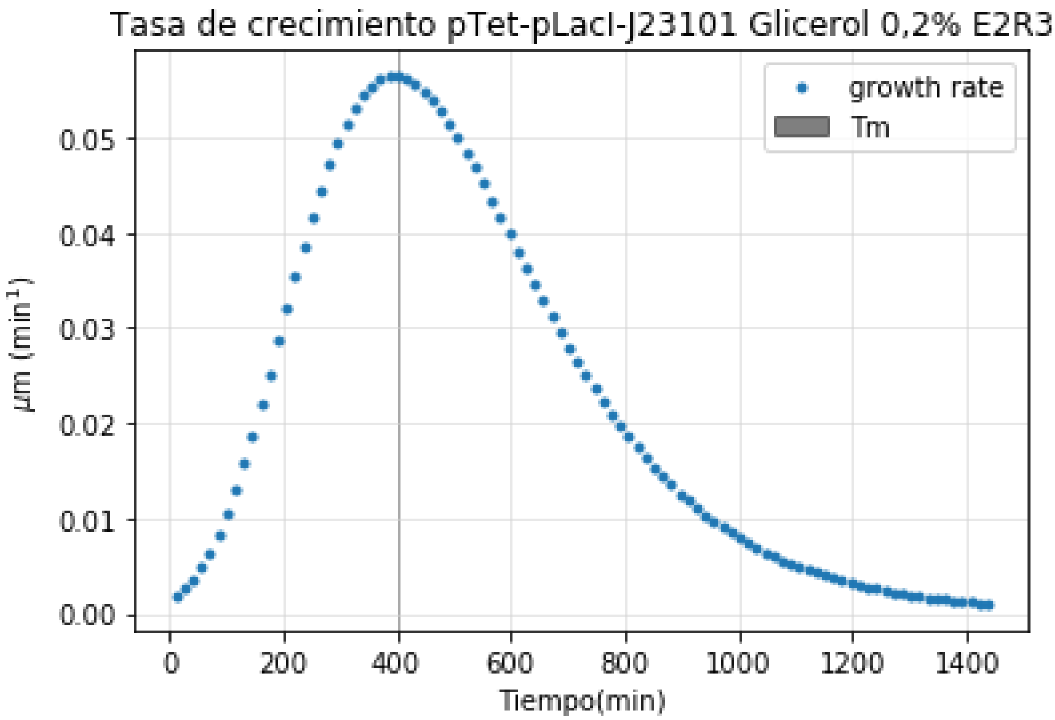

```
In [122]: #tasa de crecimiento
          ye22=((A22*np.exp(-np.exp((((um22*np.exp(1))/A22)*(l22-tt))+1))))
          #Con diff
          dy22=(np.diff(ye22))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glicerol 0,2% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[122]: <matplotlib.legend.Legend at 0x267f7d027b8>
```



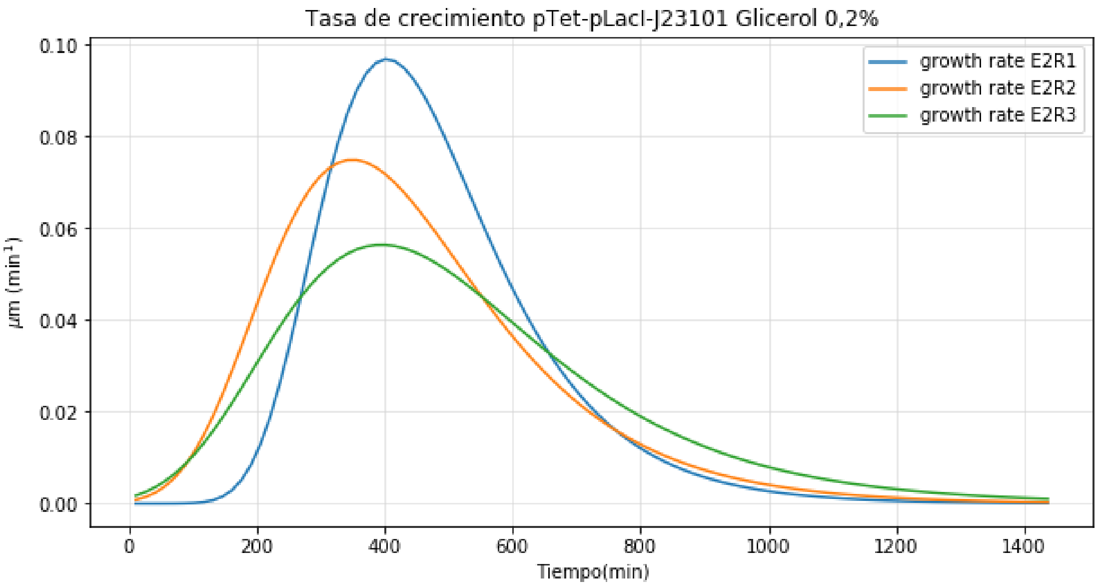

```
In [123]: #tasa de crecimiento
          ye23=((A23*np.exp(-np.exp((((um23*np.exp(1))/A23)*(l23-tt))+1))))
          #Con diff
          dy23=(np.diff(ye23))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[123]: <matplotlib.legend.Legend at 0x267f7dd57f0>
```



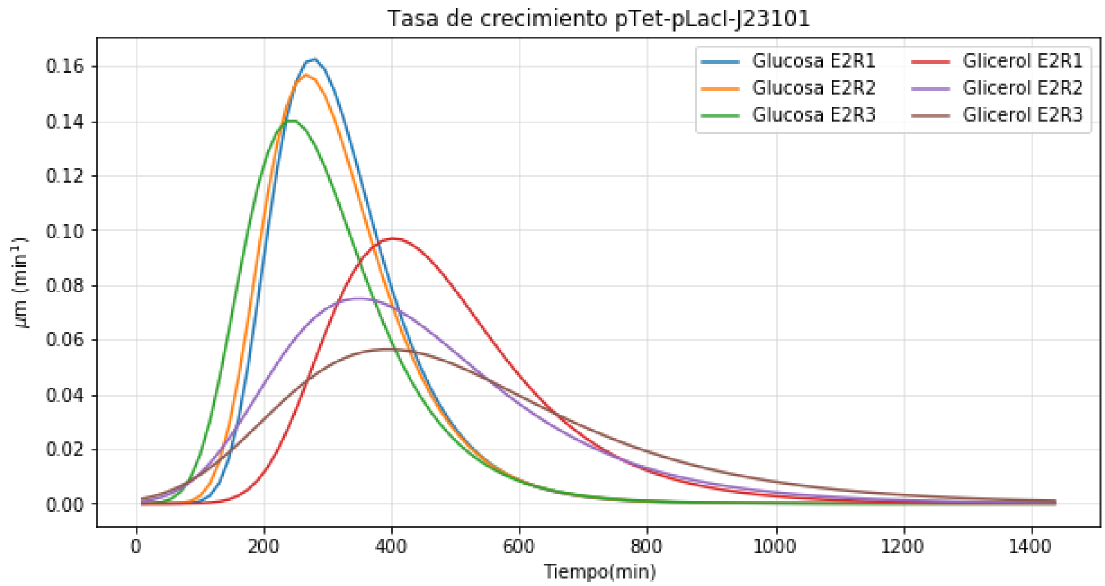

```
In [124]: #tasa de crecimiento
          ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(l24-tt))+1))))
          #Con diff
          dy24=(np.diff(ye24))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[124]: <matplotlib.legend.Legend at 0x267f7ea5fd0>
```



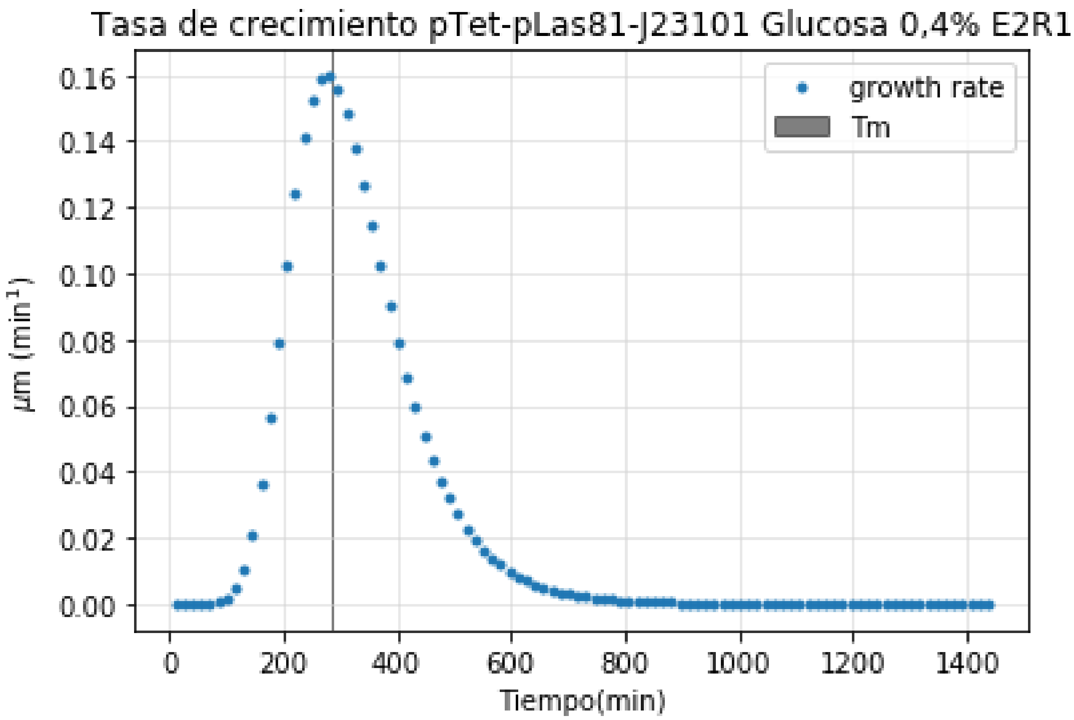

```
In [125]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLacI-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy22,label='growth rate E2R1')
          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[125]: <matplotlib.legend.Legend at 0x267f7f76d68>
```



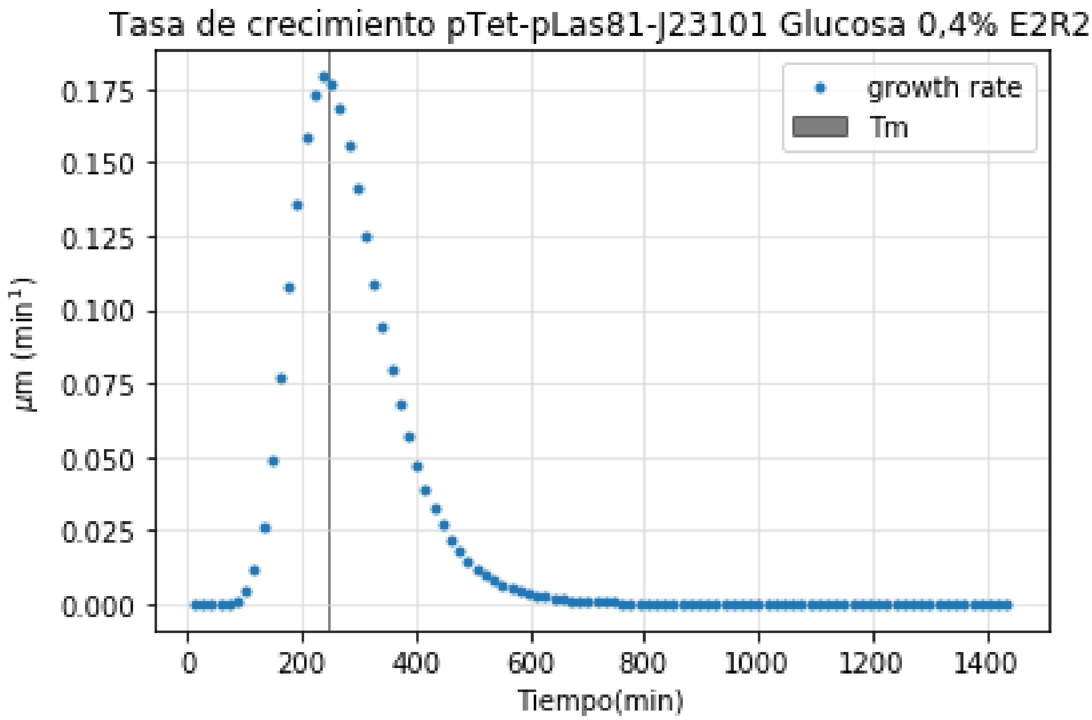

```
In [126]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLacI-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='Glucosa E2R1')
          plt.plot(tt[:-1],dy20,label='Glucosa 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[126]: <matplotlib.legend.Legend at 0x267f805ed68>
```



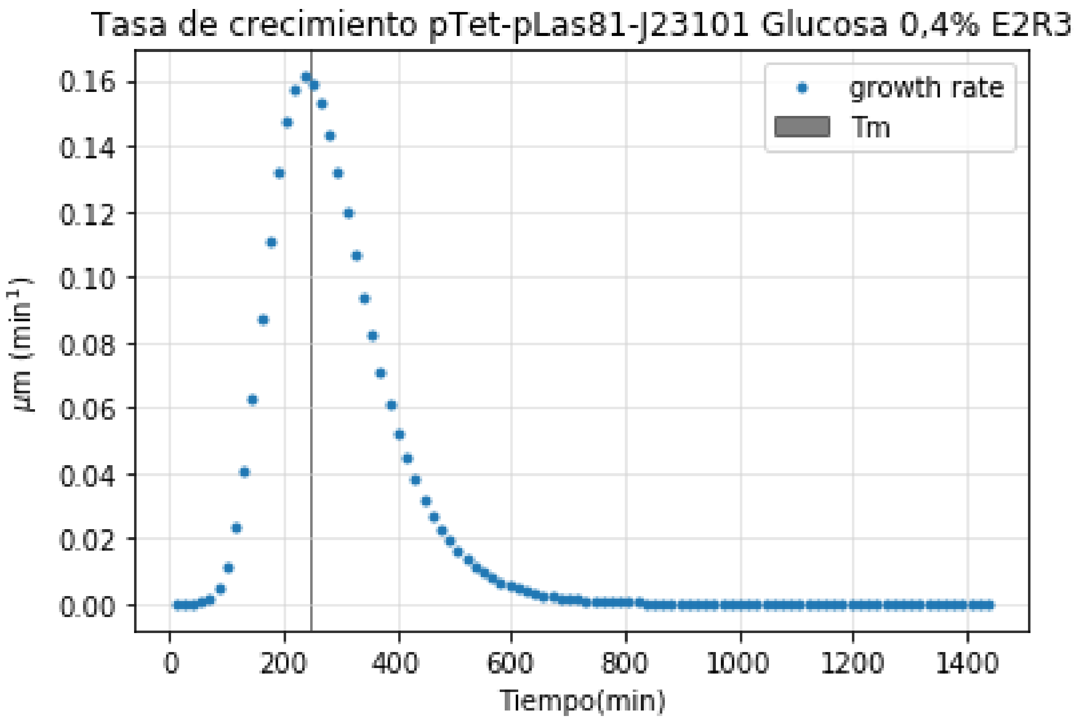

```
In [127]: #tasa de crecimiento
          ye25=((A25*np.exp(-np.exp((((um25*np.exp(1))/A25)*(l25-tt))+1))))
          #Con diff
          dy25=(np.diff(ye25))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[127]: <matplotlib.legend.Legend at 0x267f8140dd8>
```



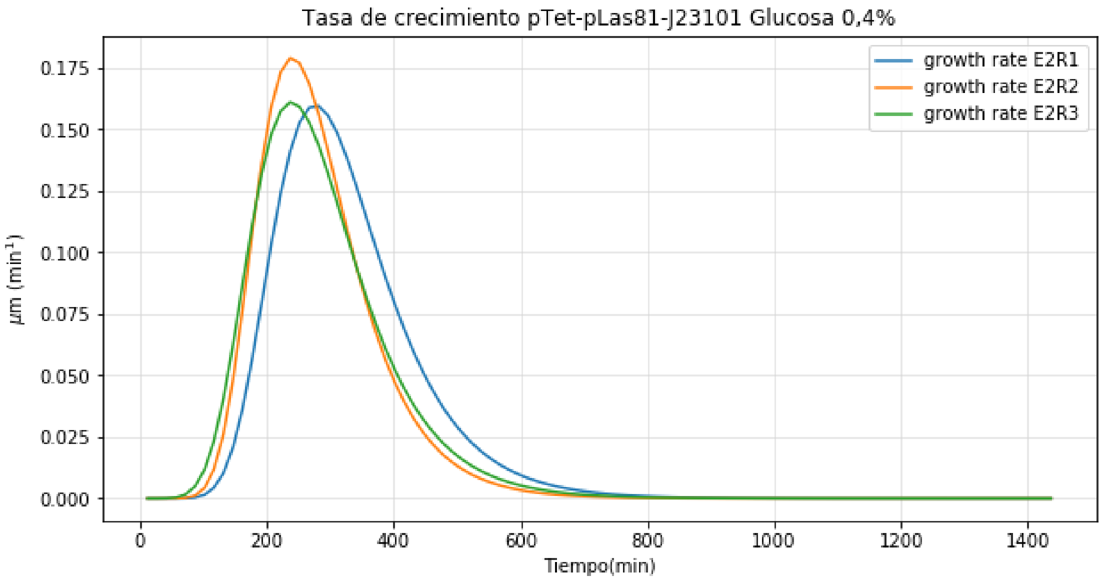

```
In [128]: #tasa de crecimiento
          ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(l26-tt))+1))))
          #Con diff
          dy26=(np.diff(ye26))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[128]: <matplotlib.legend.Legend at 0x267f8218978>
```




```
In [129]: #tasa de crecimiento
          ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(l27-tt))+1))))
          #Con diff
          dy27=(np.diff(ye27))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[129]: <matplotlib.legend.Legend at 0x267f92c0940>
```

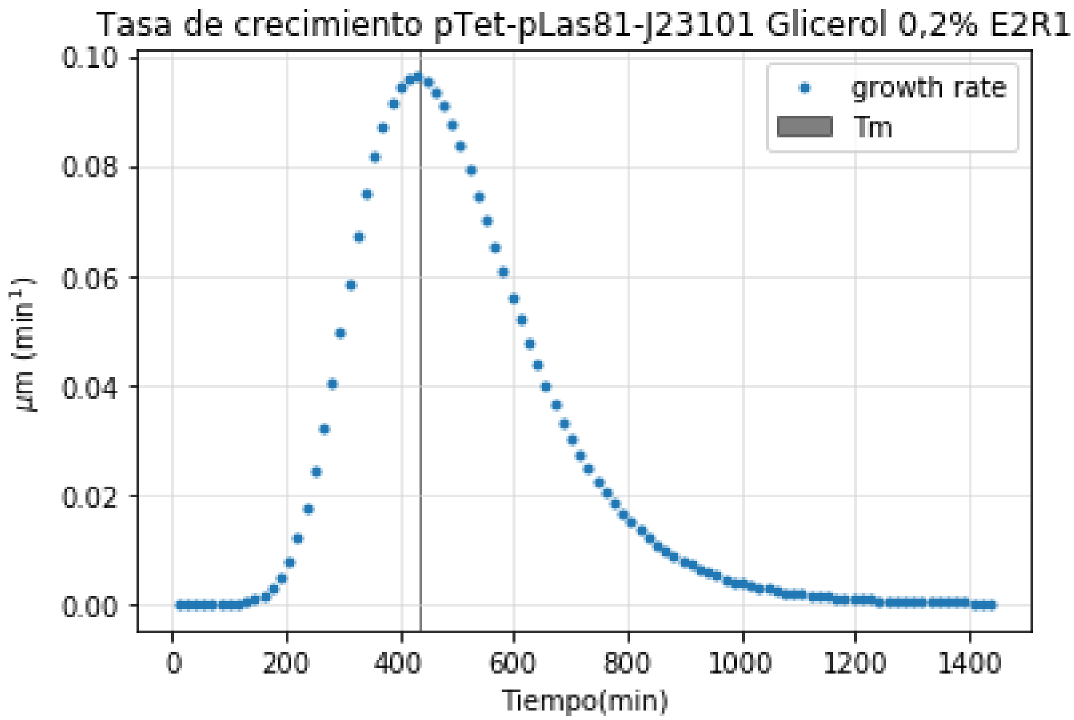



```
In [130]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='growth rate E2R1')
          plt.plot(tt[:-1],dy26,label='growth rate E2R2')
          plt.plot(tt[:-1],dy27,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[130]: <matplotlib.legend.Legend at 0x267f937eb38>
```

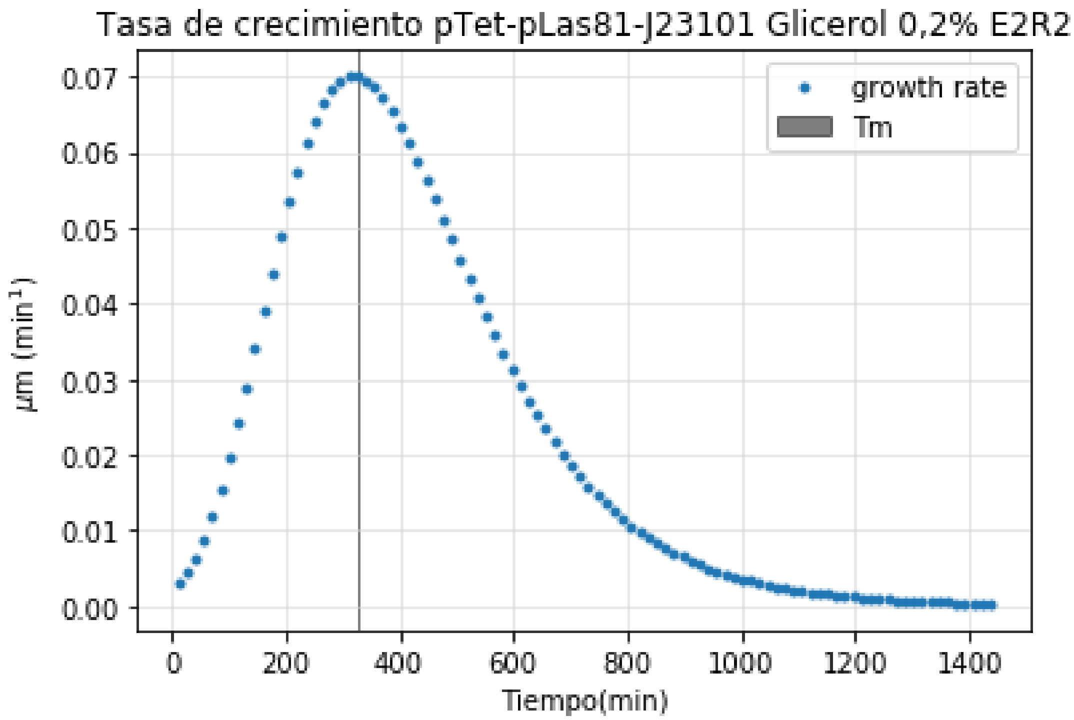



In [131]:
```
#tasa de crecimiento
ye28=((A28*np.exp(-np.exp((((um28*np.exp(1))/A28)*(l28-tt))+1))))
#Con diff
dy28=(np.diff(ye28))
plt.figure()
plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm28,tm28, color='k', 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')
```

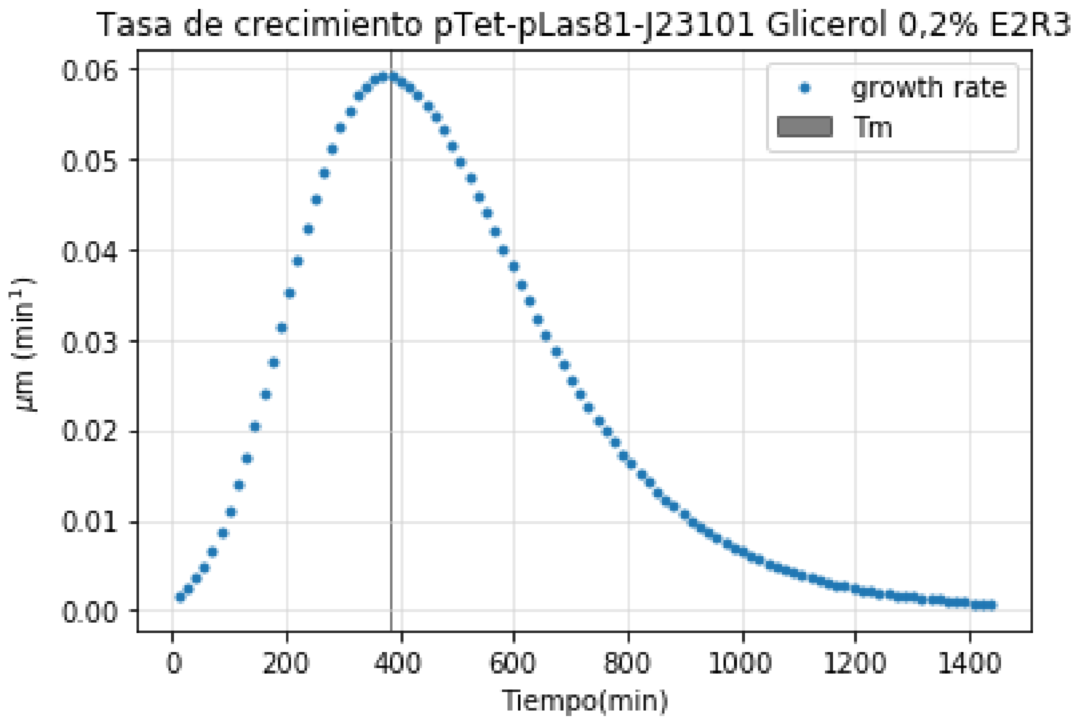
Out[131]:
```
<matplotlib.legend.Legend at 0x267f945bef0>
```



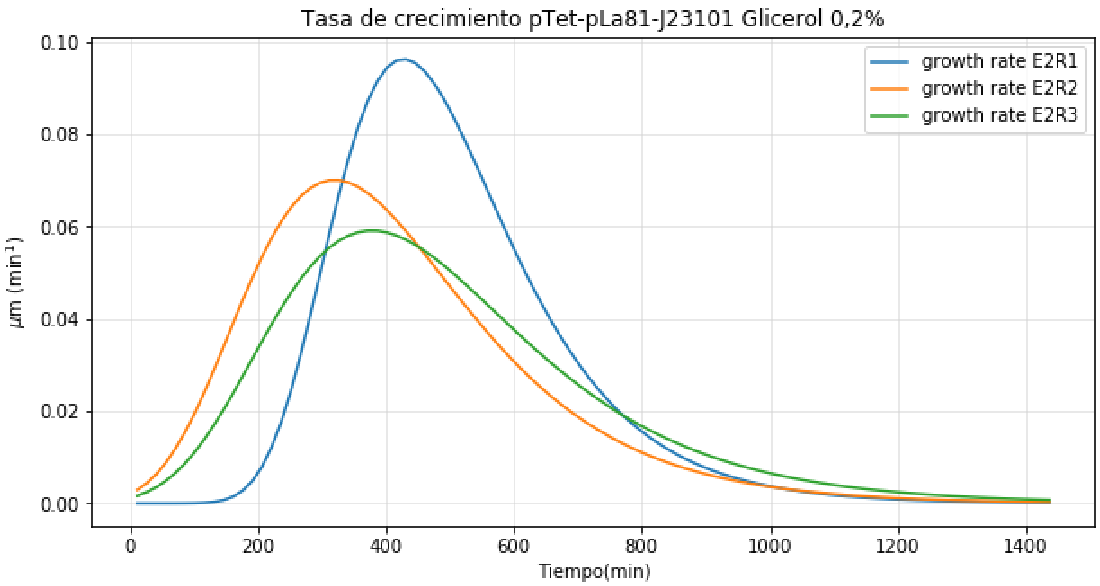

```
In [132]: #tasa de crecimiento
          ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(l29-tt))+1))))
          #Con diff
          dy29=(np.diff(ye29))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[132]: <matplotlib.legend.Legend at 0x267f9533f60>
```



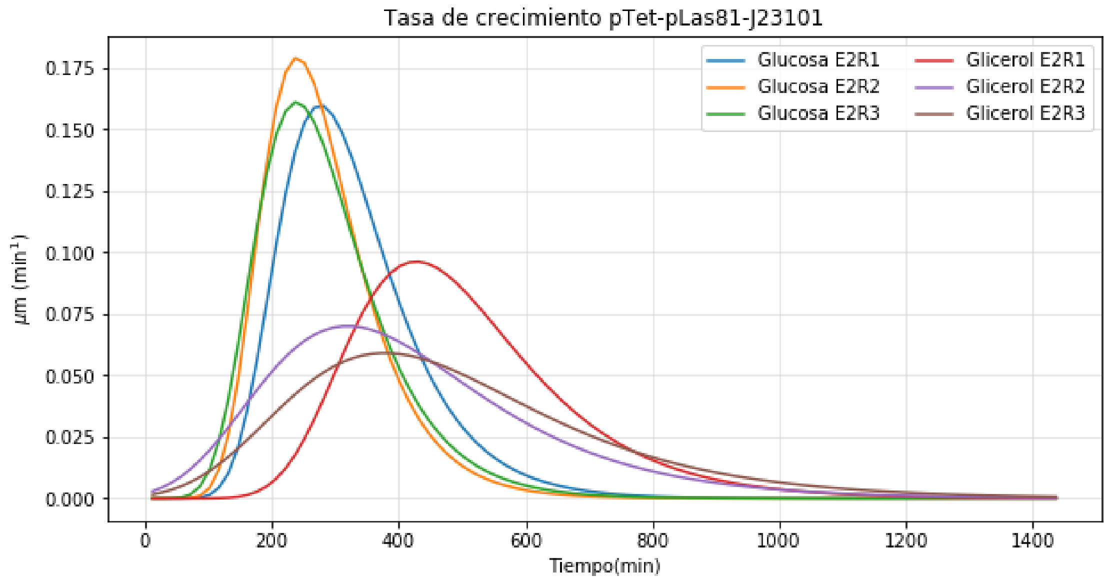

```
In [133]: #tasa de crecimiento
          ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(l30-tt))+1))))
          #Con diff
          dy30=(np.diff(ye30))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-pLas81-J23101 Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[133]: <matplotlib.legend.Legend at 0x267f9612d30>
```



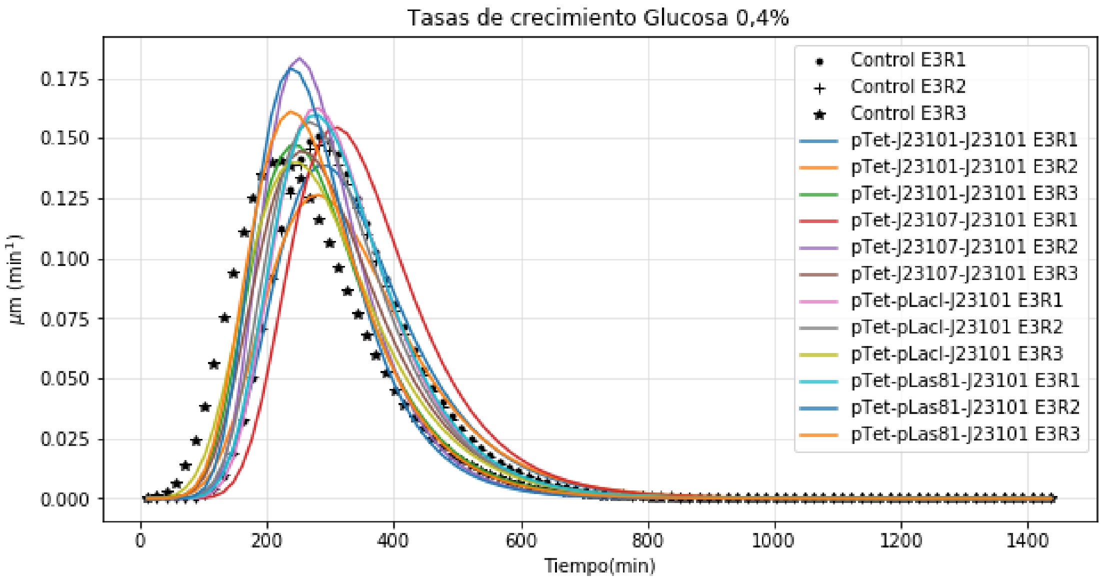

```
In [134]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLa81-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy28,label='growth rate E2R1')
          plt.plot(tt[:-1],dy29,label='growth rate E2R2')
          plt.plot(tt[:-1],dy30,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[134]: <matplotlib.legend.Legend at 0x267f96e4ac8>
```




```
In [135]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-pLas81-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='Glucosa E2R1')
          plt.plot(tt[:-1],dy26,label='Glucosa E2R2')
          plt.plot(tt[:-1],dy27,label='Glucosa E2R3')
          plt.plot(tt[:-1],dy28,label='Glicerol E2R1')
          plt.plot(tt[:-1],dy29,label='Glicerol E2R2')
          plt.plot(tt[:-1],dy30,label='Glicerol E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[135]: <matplotlib.legend.Legend at 0x267f97d3ac8>
```

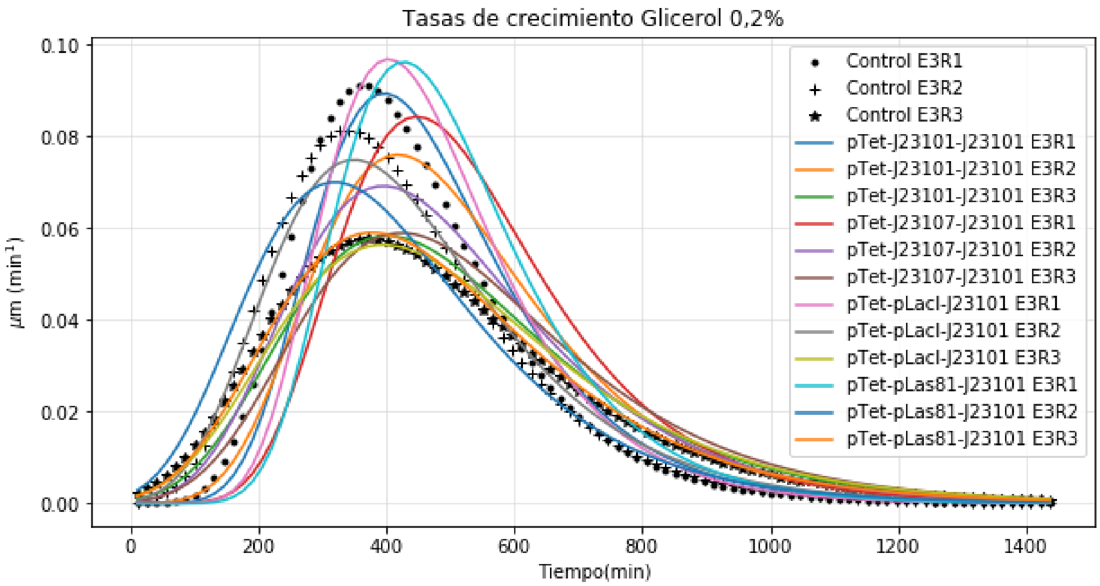



```
In [136]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,'k.',label='Control E3R1')
          plt.plot(tt[:-1],dy2,'k+',label='Control E3R2')
          plt.plot(tt[:-1],dy3,'k*',label='Control E3R3')
          plt.plot(tt[:-1],dy7,label='pTet-J23101-J23101 E3R1')
          plt.plot(tt[:-1],dy8,label='pTet-J23101-J23101 E3R2')
          plt.plot(tt[:-1],dy9,label='pTet-J23101-J23101 E3R3')
          plt.plot(tt[:-1],dy13,label='pTet-J23107-J23101 E3R1')
          plt.plot(tt[:-1],dy14,label='pTet-J23107-J23101 E3R2')
          plt.plot(tt[:-1],dy15,label='pTet-J23107-J23101 E3R3')
          plt.plot(tt[:-1],dy19,label='pTet-pLacI-J23101 E3R1')
          plt.plot(tt[:-1],dy20,label='pTet-pLacI-J23101 E3R2')
          plt.plot(tt[:-1],dy21,label='pTet-pLacI-J23101 E3R3')
          plt.plot(tt[:-1],dy25,label='pTet-pLas81-J23101 E3R1')
          plt.plot(tt[:-1],dy26,label='pTet-pLas81-J23101 E3R2')
          plt.plot(tt[:-1],dy27,label='pTet-pLas81-J23101 E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[136]: <matplotlib.legend.Legend at 0x267f98e57f0>
```




In [137]:
```
#Tasas réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy4,'k.',label='Control E3R1')
plt.plot(tt[:-1],dy5,'k+',label='Control E3R2')
plt.plot(tt[:-1],dy6,'k*',label='Control E3R3')
plt.plot(tt[:-1],dy10,label='pTet-J23101-J23101 E3R1')
plt.plot(tt[:-1],dy11,label='pTet-J23101-J23101 E3R2')
plt.plot(tt[:-1],dy12,label='pTet-J23101-J23101 E3R3')
plt.plot(tt[:-1],dy16,label='pTet-J23107-J23101 E3R1')
plt.plot(tt[:-1],dy17,label='pTet-J23107-J23101 E3R2')
plt.plot(tt[:-1],dy18,label='pTet-J23107-J23101 E3R3')
plt.plot(tt[:-1],dy22,label='pTet-pLacI-J23101 E3R1')
plt.plot(tt[:-1],dy23,label='pTet-pLacI-J23101 E3R2')
plt.plot(tt[:-1],dy24,label='pTet-pLacI-J23101 E3R3')
plt.plot(tt[:-1],dy28,label='pTet-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy29,label='pTet-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy30,label='pTet-pLas81-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

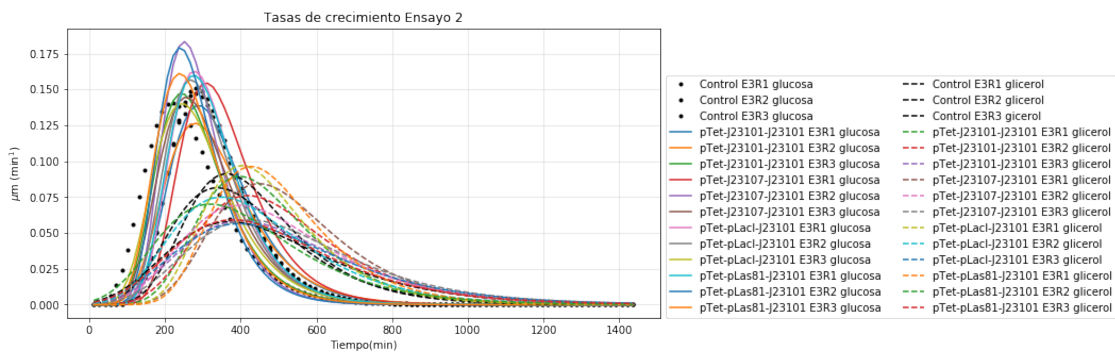
Out[137]: <matplotlib.legend.Legend at 0x267f9a54cf8>




```
In [138]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Ensayo 2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,'k.',label='Control E3R1 glucosa')
          plt.plot(tt[:-1],dy2,'k.',label='Control E3R2 glucosa')
          plt.plot(tt[:-1],dy3,'k.',label='Control E3R3 glucosa')
          plt.plot(tt[:-1],dy7,label='pTet-J23101-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy8,label='pTet-J23101-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy9,label='pTet-J23101-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy13,label='pTet-J23107-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy14,label='pTet-J23107-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy15,label='pTet-J23107-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy19,label='pTet-pLacI-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy20,label='pTet-pLacI-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy21,label='pTet-pLacI-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy25,label='pTet-pLas81-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy26,label='pTet-pLas81-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy27,label='pTet-pLas81-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy4,'k--',label='Control E3R1 glicerol')
          plt.plot(tt[:-1],dy5,'k--',label='Control E3R2 glicerol')
          plt.plot(tt[:-1],dy6,'k--',label='Control E3R3 gicerol')
          plt.plot(tt[:-1],dy10,'--',label='pTet-J23101-J23101 E3R1 glicerol')
          plt.plot(tt[:-1],dy11,'--',label='pTet-J23101-J23101 E3R2 glicerol')
          plt.plot(tt[:-1],dy12,'--',label='pTet-J23101-J23101 E3R3 glicerol')
          plt.plot(tt[:-1],dy16,'--',label='pTet-J23107-J23101 E3R1 glicerol')
          plt.plot(tt[:-1],dy17,'--',label='pTet-J23107-J23101 E3R2 glicerol')
```

```python
plt.plot(tt[:-1],dy18,'--',label='pTet-J23107-J23101 E3R3 glicerol')
plt.plot(tt[:-1],dy22,'--',label='pTet-pLacI-J23101 E3R1 glicerol')
plt.plot(tt[:-1],dy23,'--',label='pTet-pLacI-J23101 E3R2 glicerol')
plt.plot(tt[:-1],dy24,'--',label='pTet-pLacI-J23101 E3R3 glicerol')
plt.plot(tt[:-1],dy28,'--',label='pTet-pLas81-J23101 E3R1 glicerol')
plt.plot(tt[:-1],dy29,'--',label='pTet-pLas81-J23101 E3R2 glicerol')
plt.plot(tt[:-1],dy30,'--',label='pTet-pLas81-J23101 E3R3 glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[138]: <matplotlib.legend.Legend at 0x267f82180f0>




In [ ]: