Ensayo 3 todo(pLux76)

February 13, 2018

In [1]:
```
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 [1]:
```
tt=np.fromfile('t', sep=',')

#arrays replicas glucosa
cfp18211=np.fromfile('p1821gCFP1', sep=',')
rfp18211=np.fromfile('p1821gRFP1', sep=',')
yfp18211=np.fromfile('p1821gYFP1', sep=',')
od18211=np.fromfile('p1821gOD1', sep=',')
cfp18212=np.fromfile('p1821gCFP2', sep=',')
rfp18212=np.fromfile('p1821gRFP2', sep=',')
yfp18212=np.fromfile('p1821gYFP2', sep=',')
od18212=np.fromfile('p1821gOD2', sep=',')
cfp18213=np.fromfile('p1821gCFP3', sep=',')
rfp18213=np.fromfile('p1821gRFP3', sep=',')
yfp18213=np.fromfile('p1821gYFP3', sep=',')
od18213=np.fromfile('p1821gOD3', sep=',')
'''
print(cfp18211.shape)
print(rfp18211.shape)
print(yfp18211.shape)
print(od18211.shape)
print(cfp18212.shape)
print(rfp18212.shape)
print(yfp18212.shape)
print(od18212.shape)
print(cfp18213.shape)
print(rfp18213.shape)
print(yfp18213.shape)
```

```
print(od18213.shape)'''

cfp18231=np.fromfile('p1823gCFP1', sep=',')
rfp18231=np.fromfile('p1823gRFP1', sep=',')
yfp18231=np.fromfile('p1823gYFP1', sep=',')
od18231=np.fromfile('p1823gOD1', sep=',')
cfp18232=np.fromfile('p1823gCFP2', sep=',')
rfp18232=np.fromfile('p1823gRFP2', sep=',')
yfp18232=np.fromfile('p1823gYFP2', sep=',')
od18232=np.fromfile('p1823gOD2', sep=',')
cfp18233=np.fromfile('p1823gCFP3', sep=',')
rfp18233=np.fromfile('p1823gRFP3', sep=',')
yfp18233=np.fromfile('p1823gYFP3', sep=',')
od18233=np.fromfile('p1823gOD3', sep=',')

'''
print(cfp18231.shape)
print(rfp18231.shape)
print(yfp18231.shape)
print(od18231.shape)
print(cfp18232.shape)
print(rfp18232.shape)
print(yfp18232.shape)
print(od18232.shape)
print(cfp18233.shape)
print(rfp18233.shape)
print(yfp18233.shape)
print(od18233.shape)'''

cfp18261=np.fromfile('p1826gCFP1', sep=',')
rfp18261=np.fromfile('p1826gRFP1', sep=',')
yfp18261=np.fromfile('p1826gYFP1', sep=',')
od18261=np.fromfile('p1826gOD1', sep=',')
cfp18262=np.fromfile('p1826gCFP2', sep=',')
rfp18262=np.fromfile('p1826gRFP2', sep=',')
yfp18262=np.fromfile('p1826gYFP2', sep=',')
od18262=np.fromfile('p1826gOD2', sep=',')
cfp18263=np.fromfile('p1826gCFP3', sep=',')
rfp18263=np.fromfile('p1826gRFP3', sep=',')
yfp18263=np.fromfile('p1826gYFP3', sep=',')
od18263=np.fromfile('p1826gOD3', sep=',')

'''
print(cfp18261.shape)
print(rfp18261.shape)
print(yfp18261.shape)
print(od18261.shape)
print(cfp18262.shape)
```

```
print(rfp18262.shape)
print(yfp18262.shape)
print(od18262.shape)
print(cfp18263.shape)
print(rfp18263.shape)
print(yfp18263.shape)
print(od18263.shape)'''

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

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

#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)'''
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-1-6ad46f4fd900> in <module>()
----> 1 tt=np.fromfile('t', sep=',')
      2
      3 #arrays replicas glucosa
      4 cfp18211=np.fromfile('p1821gCFP1', sep=',')
      5 rfp18211=np.fromfile('p1821gRFP1', sep=',')


NameError: name 'np' is not defined
```

In [3]:
```
#Promedios glicerol
#arrays replicas glicerol
cfp1821g1=np.fromfile('p1821glCFP1', sep=',')
rfp1821g1=np.fromfile('p1821glRFP1', sep=',')
yfp1821g1=np.fromfile('p1821glYFP1', sep=',')
od1821g1=np.fromfile('p1821glOD1', sep=',')
cfp1821g2=np.fromfile('p1821glCFP2', sep=',')
rfp1821g2=np.fromfile('p1821glRFP2', sep=',')
yfp1821g2=np.fromfile('p1821glYFP2', sep=',')
od1821g2=np.fromfile('p1821glOD2', sep=',')
cfp1821g3=np.fromfile('p1821glCFP3', sep=',')
rfp1821g3=np.fromfile('p1821glRFP3', sep=',')
yfp1821g3=np.fromfile('p1821glYFP3', sep=',')
od1821g3=np.fromfile('p1821glOD3', sep=',')
'''
print(cfp1821g1.shape)
print(rfp1821g1.shape)
```

```
print(yfp1821g1.shape)
print(od1821g1.shape)
print(cfp1821g2.shape)
print(rfp1821g2.shape)
print(yfp1821g2.shape)
print(od1821g2.shape)
print(cfp1821g3.shape)
print(rfp1821g3.shape)
print(yfp1821g3.shape)
print(od1821g3.shape)'''


cfp1823g1=np.fromfile('p1823glCFP1', sep=',')
rfp1823g1=np.fromfile('p1823glRFP1', sep=',')
yfp1823g1=np.fromfile('p1823glYFP1', sep=',')
od1823g1=np.fromfile('p1823glOD1', sep=',')
cfp1823g2=np.fromfile('p1823glCFP2', sep=',')
rfp1823g2=np.fromfile('p1823glRFP2', sep=',')
yfp1823g2=np.fromfile('p1823glYFP2', sep=',')
od1823g2=np.fromfile('p1823glOD2', sep=',')
cfp1823g3=np.fromfile('p1823glCFP3', sep=',')
rfp1823g3=np.fromfile('p1823glRFP3', sep=',')
yfp1823g3=np.fromfile('p1823glYFP3', sep=',')
od1823g3=np.fromfile('p1823glOD3', sep=',')
'''
print(cfp1823g1.shape)
print(rfp1823g1.shape)
print(yfp1823g1.shape)
print(od1823g1.shape)
print(cfp1823g2.shape)
print(rfp1823g2.shape)
print(yfp1823g2.shape)
print(od1823g2.shape)
print(cfp1823g3.shape)
print(rfp1823g3.shape)
print(yfp1823g3.shape)
print(od1823g3.shape)'''


cfp1826g1=np.fromfile('p1826glCFP1', sep=',')
rfp1826g1=np.fromfile('p1826glRFP1', sep=',')
yfp1826g1=np.fromfile('p1826glYFP1', sep=',')
od1826g1=np.fromfile('p1826glOD1', sep=',')
cfp1826g2=np.fromfile('p1826glCFP2', sep=',')
rfp1826g2=np.fromfile('p1826glRFP2', sep=',')
yfp1826g2=np.fromfile('p1826glYFP2', sep=',')
od1826g2=np.fromfile('p1826glOD2', sep=',')
cfp1826g3=np.fromfile('p1826glCFP3', sep=',')
```

```python
rfp1826g3=np.fromfile('p1826glRFP3', sep=',')
yfp1826g3=np.fromfile('p1826glYFP3', sep=',')
od1826g3=np.fromfile('p1826glOD3', sep=',')

'''
print(cfp1826g1.shape)
print(rfp1826g1.shape)
print(yfp1826g1.shape)
print(od1826g1.shape)
print(cfp1826g2.shape)
print(rfp1826g2.shape)
print(yfp1826g2.shape)
print(od1826g2.shape)
print(cfp1826g3.shape)
print(rfp1826g3.shape)
print(yfp1826g3.shape)
print(od1826g3.shape)'''

cfp1827g1=np.fromfile('p1827glCFP1', sep=',')
rfp1827g1=np.fromfile('p1827glRFP1', sep=',')
yfp1827g1=np.fromfile('p1827glYFP1', sep=',')
od1827g1=np.fromfile('p1827glOD1', sep=',')
cfp1827g2=np.fromfile('p1827glCFP2', sep=',')
rfp1827g2=np.fromfile('p1827glRFP2', sep=',')
yfp1827g2=np.fromfile('p1827glYFP2', sep=',')
od1827g2=np.fromfile('p1827glOD2', sep=',')
cfp1827g3=np.fromfile('p1827glCFP3', sep=',')
rfp1827g3=np.fromfile('p1827glRFP3', sep=',')
yfp1827g3=np.fromfile('p1827glYFP3', sep=',')
od1827g3=np.fromfile('p1827glOD3', sep=',')

'''
print(cfp1827g1.shape)
print(rfp1827g1.shape)
print(yfp1827g1.shape)
print(od1827g1.shape)
print(cfp1827g2.shape)
print(rfp1827g2.shape)
print(yfp1827g2.shape)
print(od1827g2.shape)
print(cfp1827g3.shape)
print(rfp1827g3.shape)
print(yfp1827g3.shape)
print(od1827g3.shape)'''


#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[3]: '\nprint(cfpcgl1.shape)\nprint(rfpcgl1.shape)\nprint(yfpcgl1.shape)\nprint(odcgl1.shape)

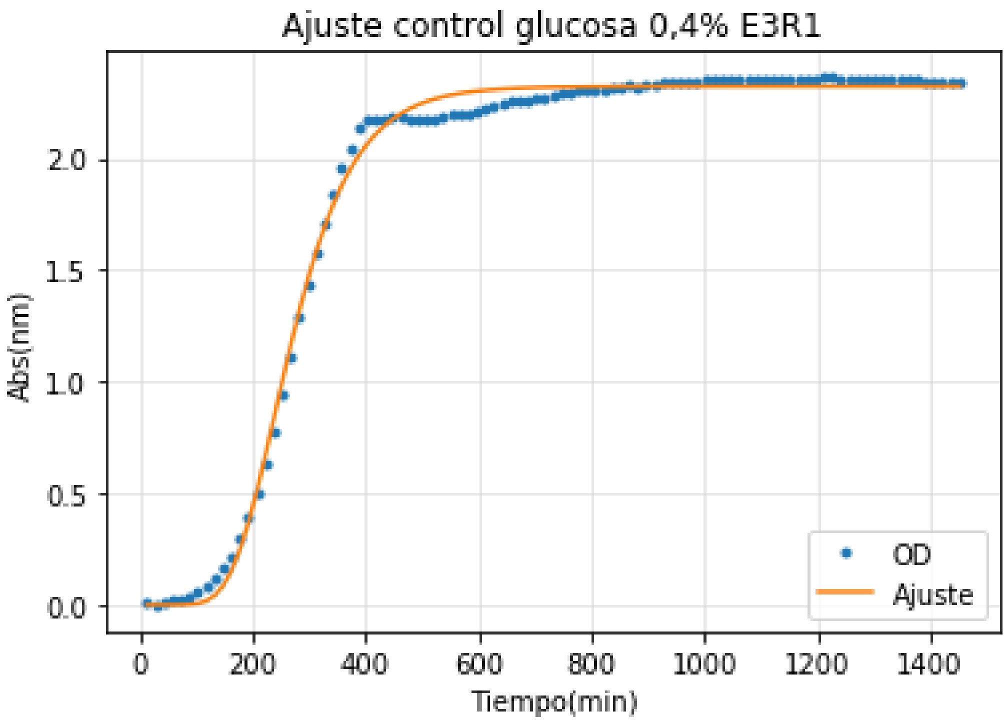
In [4]:
```
#Funciones para ajuste Gompertz
def F_sigma(t, A, um,l):
    return ((A*np.exp(-np.exp((((um*np.exp(1))/A)*(l-t))+1))))

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 [5]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control glucosa rep 1
        y1 = np.log(odcg1)-np.log(np.min(odcg1))
        print('Min OD = %e'%((np.min(odcg1))))
        evaly, params=Function_fit(tt,y1,0,-1,title = 'Ajuste control glucosa 0,4% E3R1')
        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[16]
        y2=tt[25]
        plt.figure()
        plt.title('Control Glucosa 0,4% E3R1')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt,odcg1,label='OD control E3R1 ')
        plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')

        #Fase exponencial OD/tiempo
        plt.figure()
        plt.title('Fase exponencial OD E3R1')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt[16:26],odcg1[16:26],label='OD control E3R1')
```

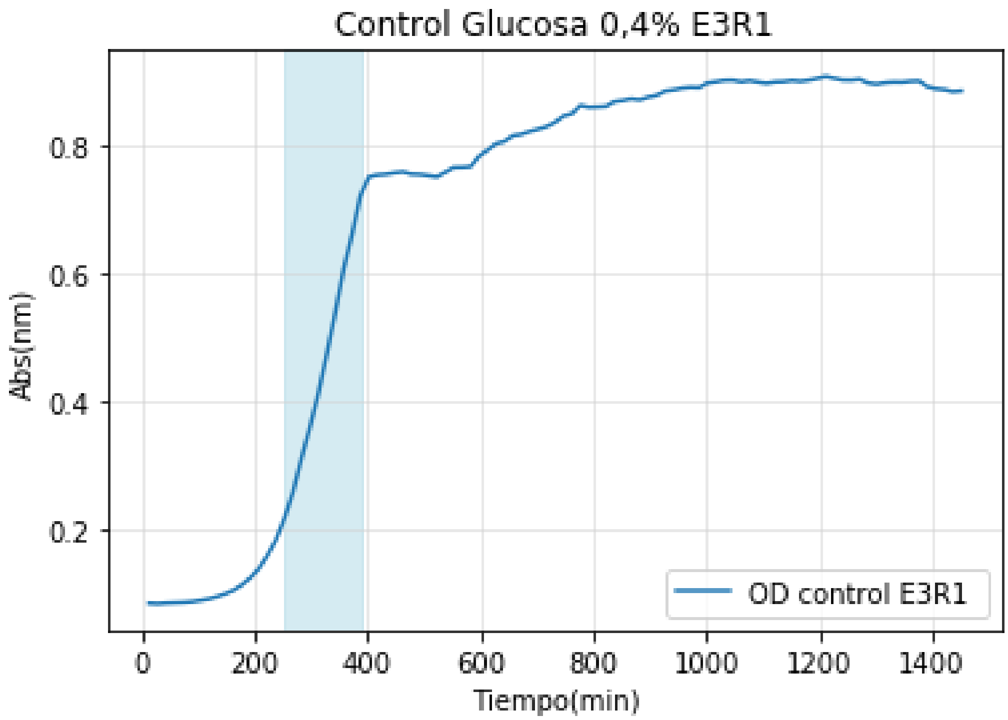
```
    plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
    plt.legend(loc='lower right')
```

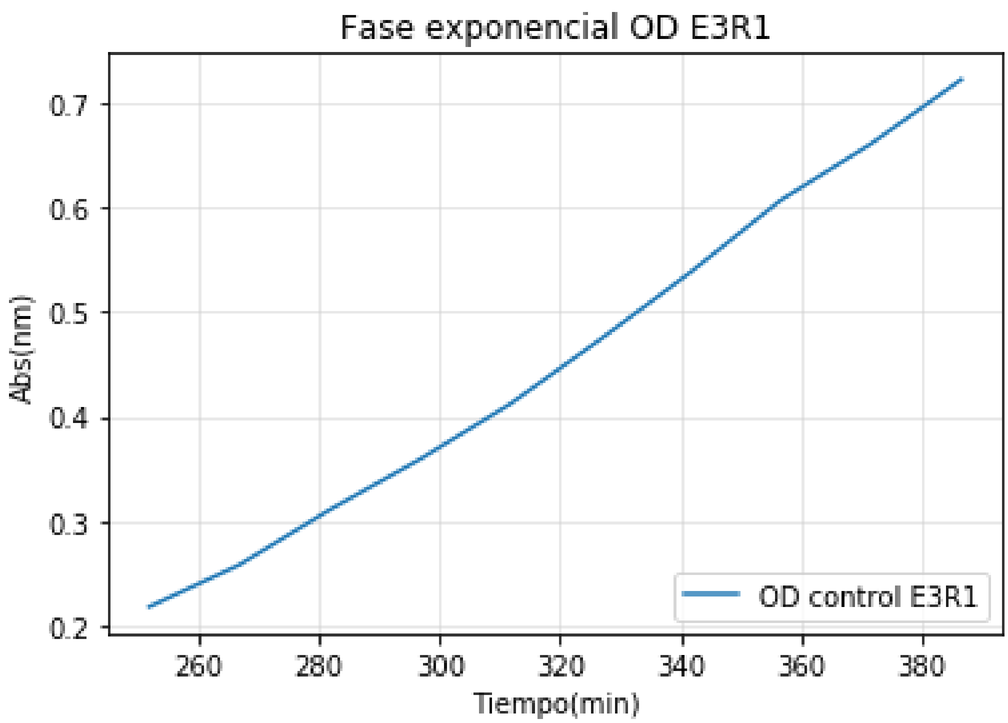
```
Min OD = 8.550000e-02
[  2.32234986e+00   1.11487857e-02   1.61538943e+02]
```




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

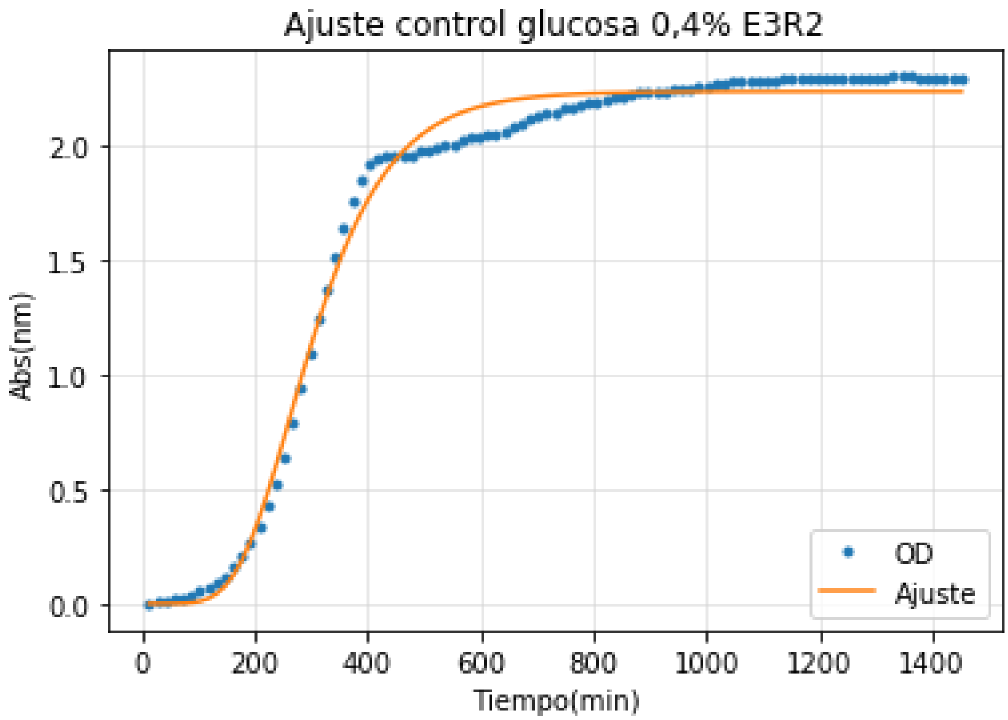
```
Out[5]: <matplotlib.legend.Legend at 0x1d719a168d0>
```


Control Glucosa 0,4% E3R1
Abs(nm)
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E3R1


Fase exponencial OD E3R1
Abs(nm)
0.7
0.6
0.5
0.4
0.3
0.2
260
280
300
320
340
360
380
Tiempo(min)
OD control E3R1

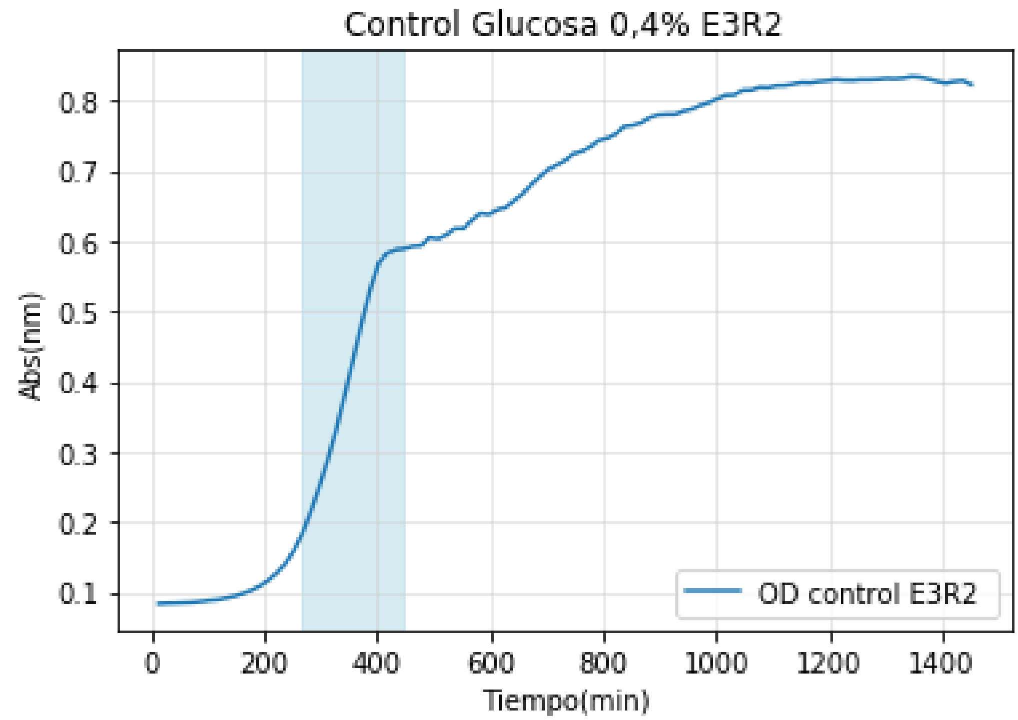
```
In [6]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control glucosa rep 2
        y2= np.log(odcg2)-np.log(np.min(odcg2))
        print('Min OD = %e'%((np.min(odcg2))))
        evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E3R2')
        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[17]
        y2=tt[29]
        plt.figure()
        plt.title('Control Glucosa 0,4% E3R2')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt,odcg2,label='OD control E3R2 ')
        plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')

        #Fase exponencial OD/tiempo
        plt.figure()
        plt.title('Fase exponencial OD E3R2')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt[17:30],odcg2[17:30],label='OD control E3R2')
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')
```

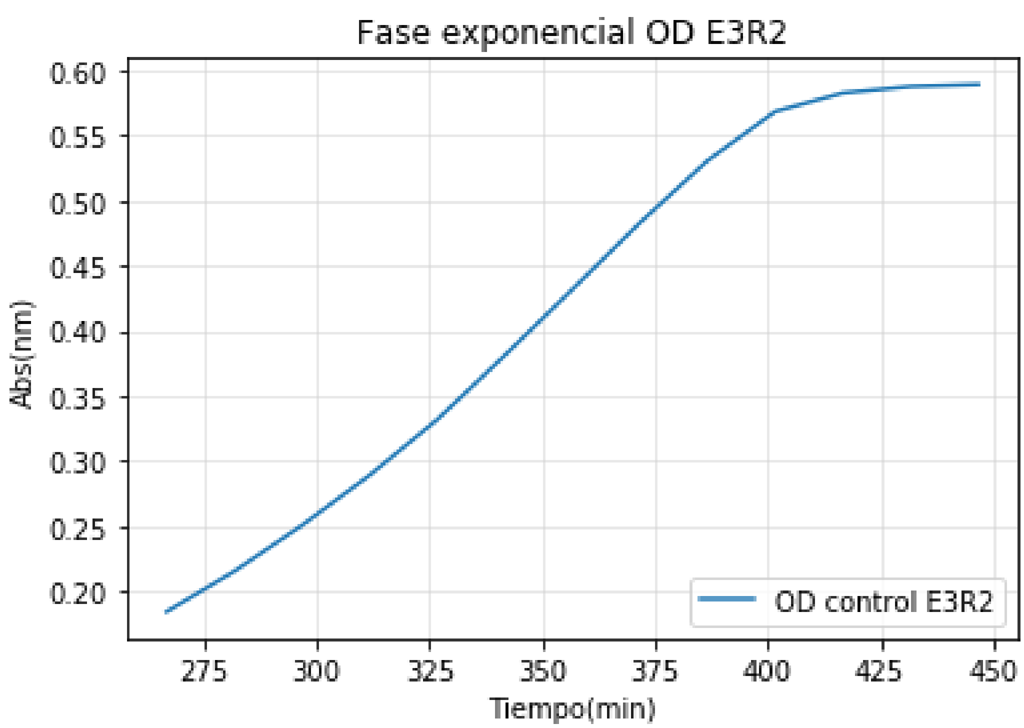
```
Min OD = 8.375000e-02
[  2.23365370e+00   8.60171922e-03   1.67065878e+02]
```




```
A=2.233654e+00
um=8.601719e-03
l=1.670659e+02
Tm=2.625951e+02
doubpe=8.058240e+01
ext=1.611648e+02
Tfinal=4.237599e+02
```

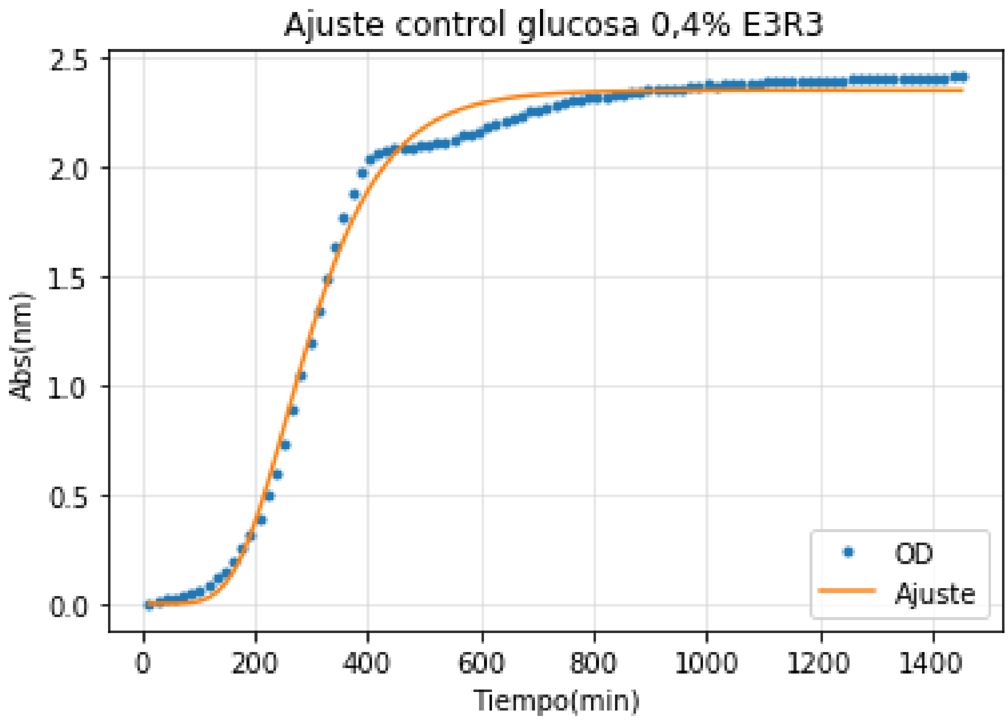
```
Out[6]: <matplotlib.legend.Legend at 0x1d719baaf60>
```


Control Glucosa 0,4% E3R2
Abs(nm)
Tiempo(min)
OD control E3R2
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 E3R2
Abs(nm)
Tiempo(min)
OD control E3R2
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
275
300
325
350
375
400
425
450

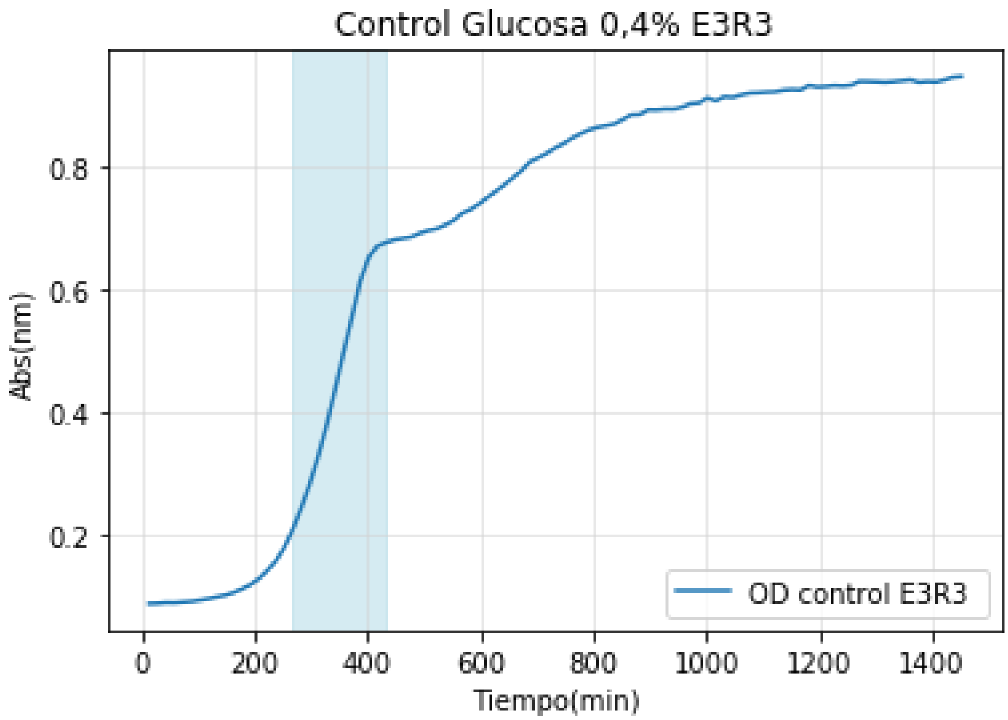
In [7]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y3= np.log(odcg3)-np.log(np.min(odcg3))
print('Min OD = %e'%((np.min(odcg3))))
evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E3R3')
A3= params[0]
um3=params[1]
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[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg3,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],odcg3[17:29],label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

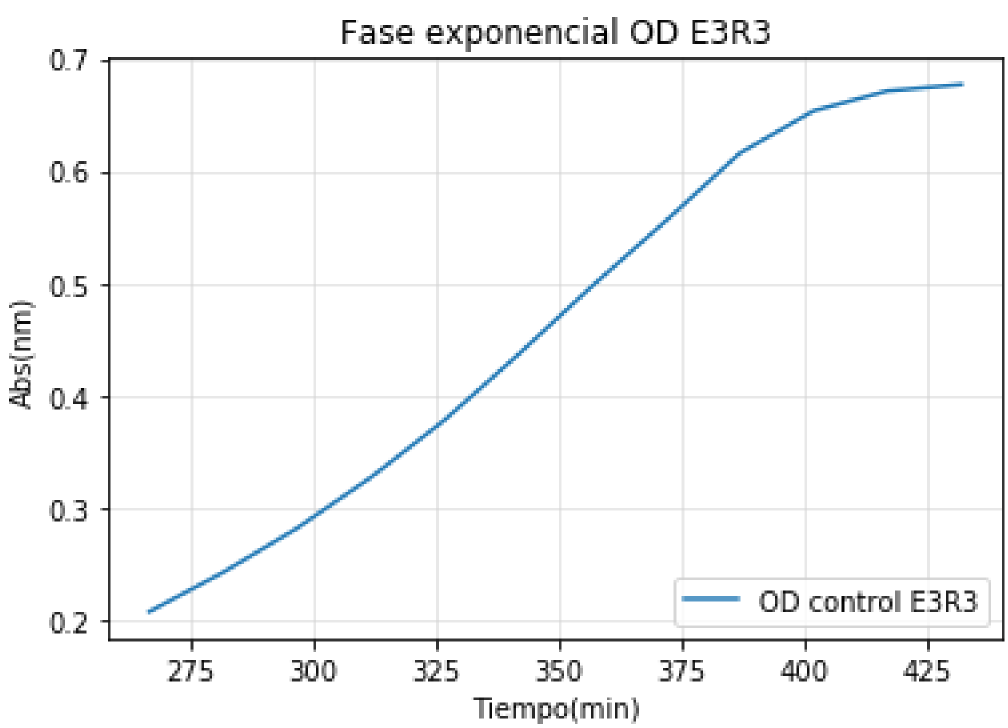
```
Min OD = 8.525000e-02
[  2.34758891e+00   9.20947423e-03   1.63131184e+02]
```




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

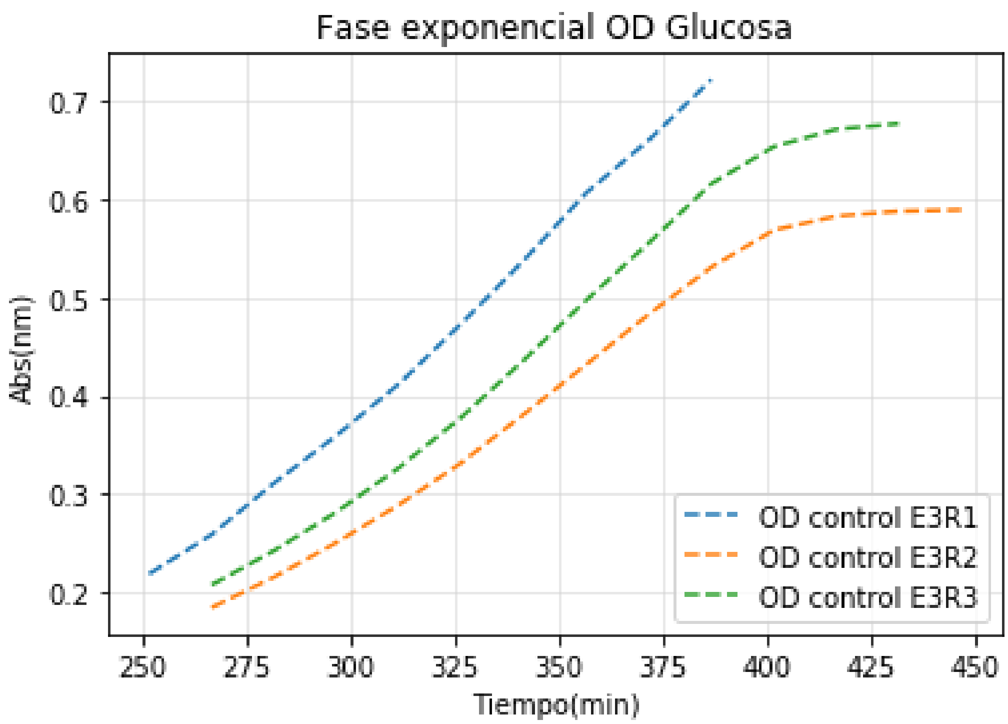
```
Out[7]: <matplotlib.legend.Legend at 0x1d71adab588>
```


Control Glucosa 0,4% E3R3
Abs(nm)
Tiempo(min)
OD control E3R3


Fase exponencial OD E3R3
Abs(nm)
Tiempo(min)
OD control E3R3

In [8]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glucosa')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],odcg1[16:26],'--',label='OD control E3R1')
plt.plot(tt[17:30],odcg2[17:30],'--',label='OD control E3R2')
plt.plot(tt[17:29],odcg3[17:29],'--',label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

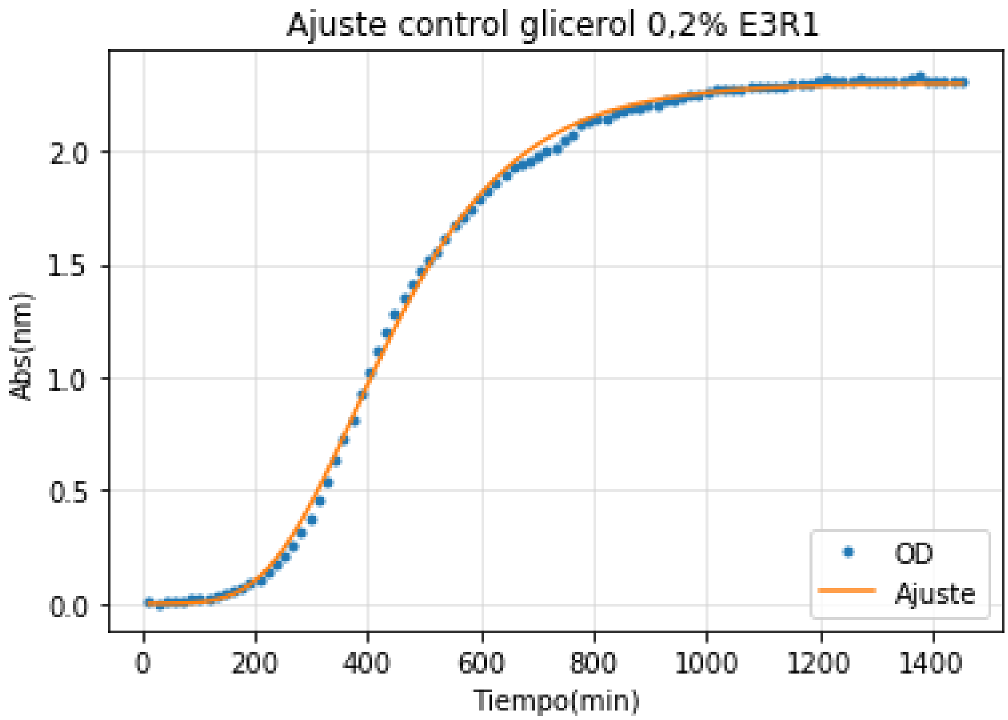
Out[8]: <matplotlib.legend.Legend at 0x1d71aec2b70>



In [9]:
```
#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% E3R1')
A4= params[0]
um4=params[1]
l4=params[2]
print('A=%e'%(A4))
print('um=%e'%(um4))
print('l=%e'%(l4))
```

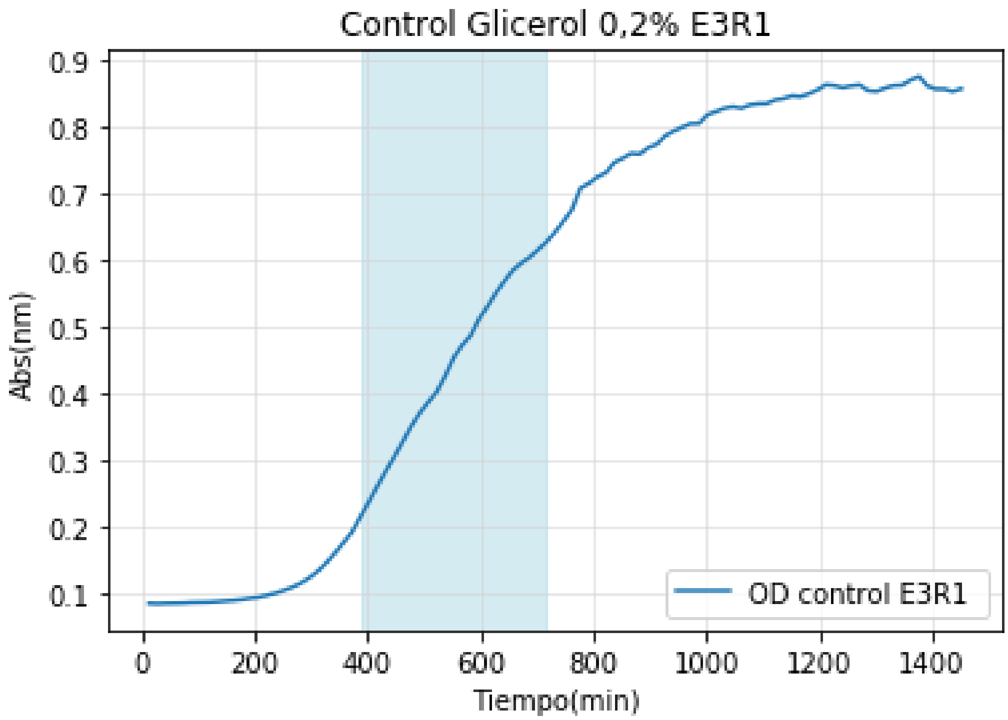
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+l4)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[25]
y2=tt[47]
plt.figure()
plt.title('Control Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl1,label='OD control E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:48],odcgl1[25:48],label='OD control E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

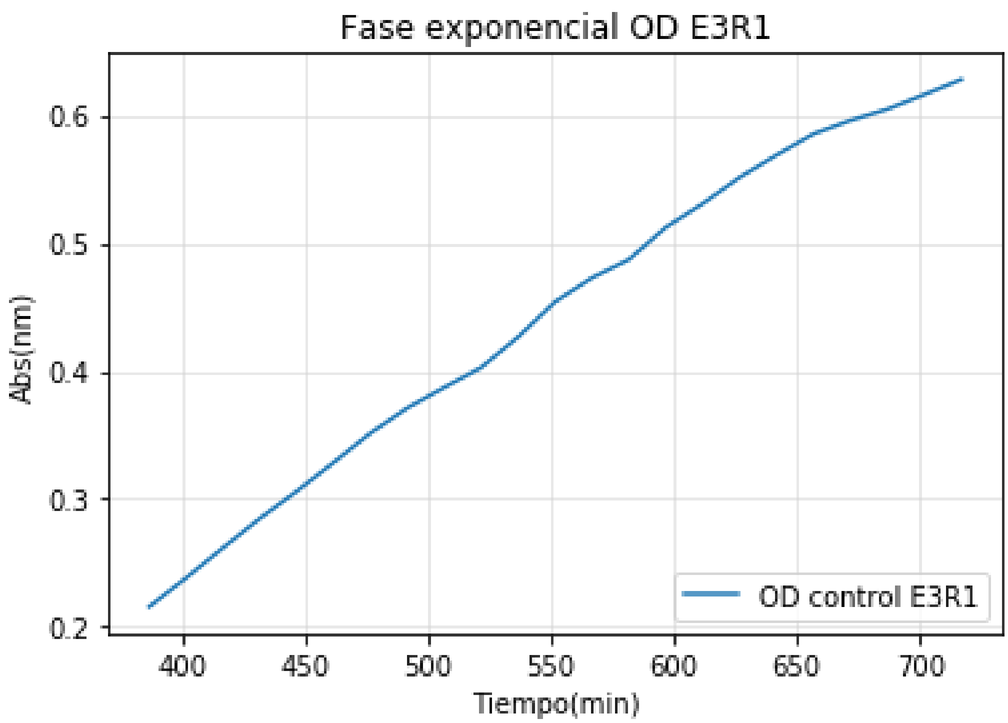
```
Min OD = 8.525000e-02
[  2.29993135e+00   5.42425453e-03   2.20607778e+02]
```




```
A=2.299931e+00
um=5.424255e-03
l=2.206078e+02
Tm=3.765919e+02
doubpe=1.277866e+02
ext=3.194666e+02
Tfinal=6.960584e+02
```

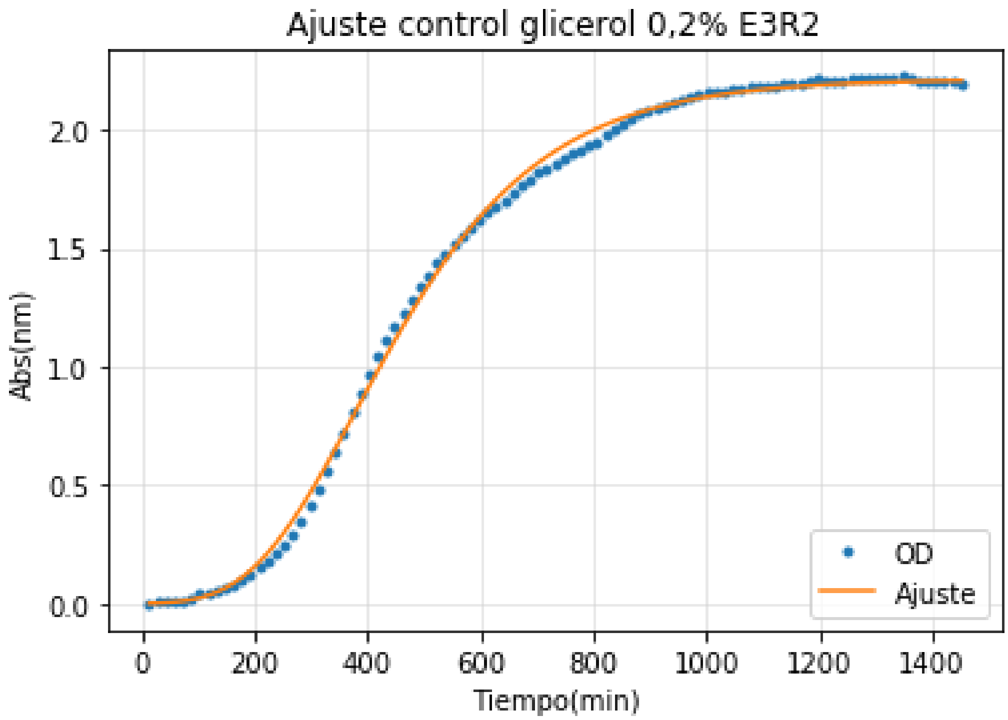
```
Out[9]: <matplotlib.legend.Legend at 0x1d71b05b198>
```


Control Glicerol 0,2% E3R1
Abs(nm)
Tiempo(min)
OD control E3R1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E3R1
Abs(nm)
Tiempo(min)
OD control E3R1
0.6
0.5
0.4
0.3
0.2
400
450
500
550
600
650
700

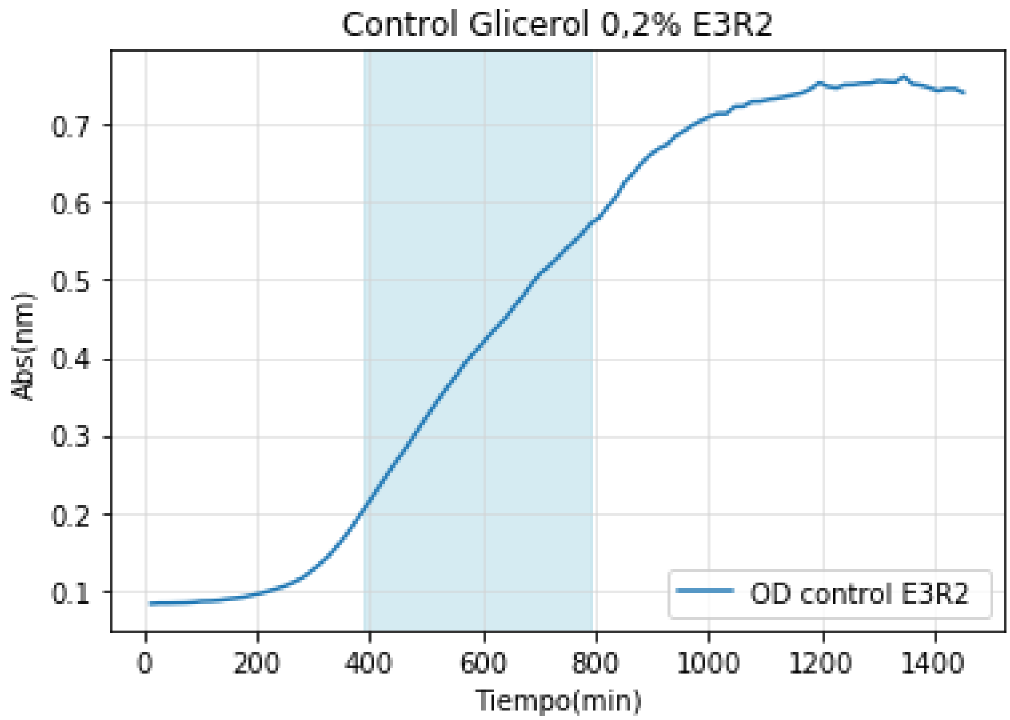
```
In [10]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 2
         y5= np.log(odcgl2)-np.log(np.min(odcgl2))
         print('Min OD = %e'%((np.min(odcgl2))))
         evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 0,2% E3R2')
         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[25]
         y2=tt[52]
         plt.figure()
         plt.title('Control Glicerol 0,2% E3R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odcgl2,label='OD control E3R2 ')
         plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

         #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD E3R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[25:53],odcgl2[25:53],label='OD control E3R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

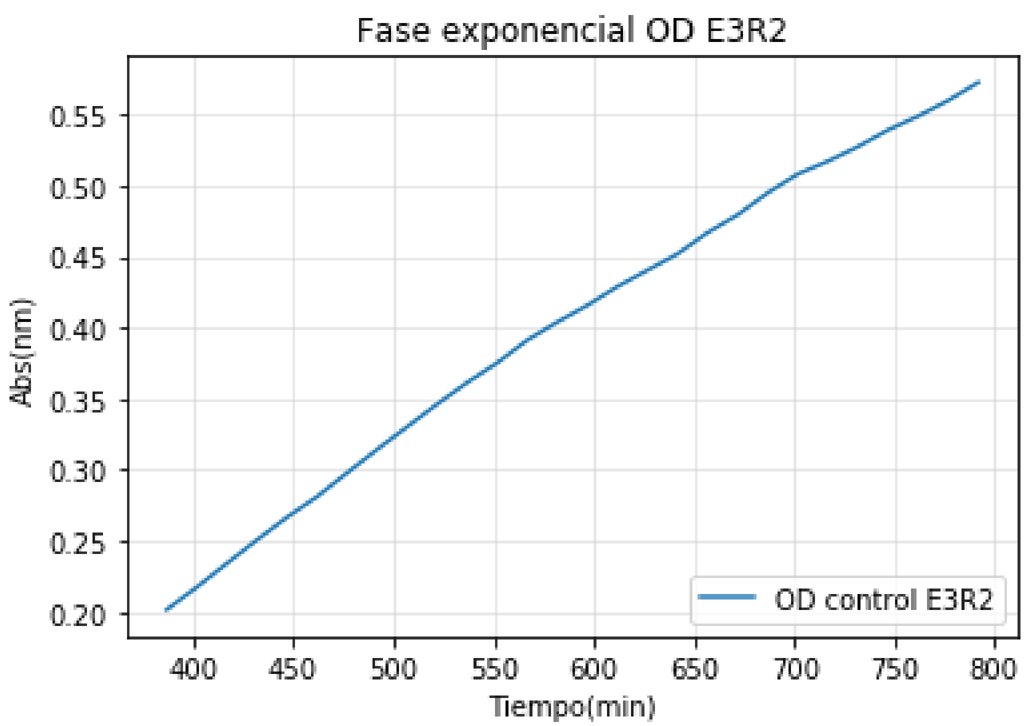
```
Min OD = 8.250000e-02
[  2.21283483e+00   4.40822880e-03   1.94027274e+02]
```




```
A=2.212835e+00
um=4.408229e-03
l=1.940273e+02
Tm=3.786947e+02
doubpe=1.572394e+02
ext=3.930985e+02
Tfinal=7.717932e+02
```

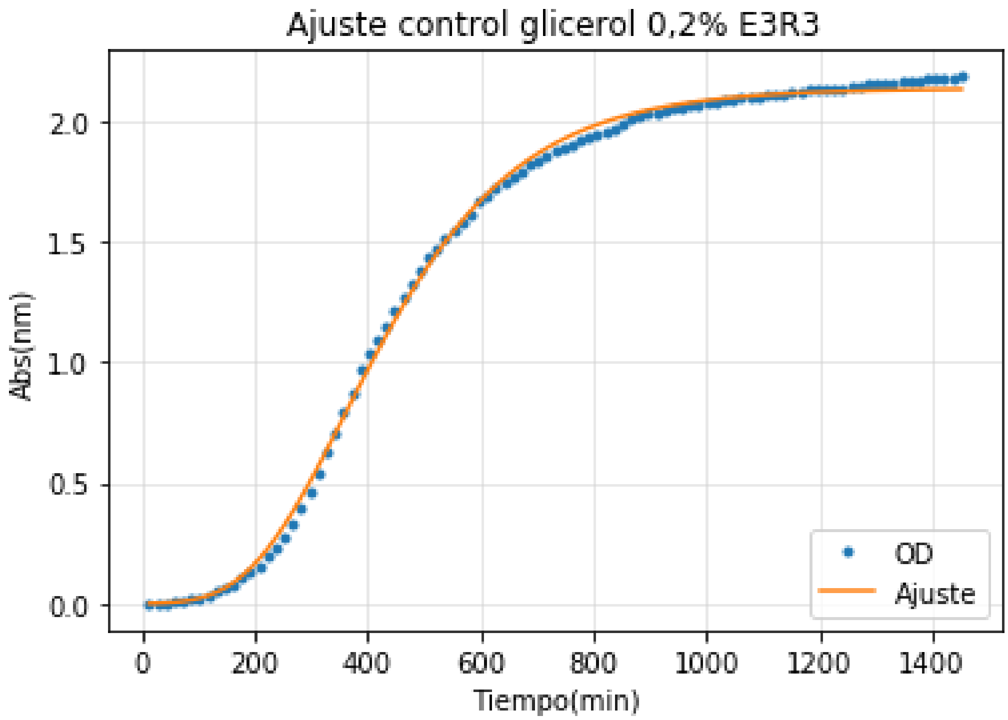
```
Out[10]: <matplotlib.legend.Legend at 0x1d719c2ada0>
```


Control Glicerol 0,2% E3R2
Abs(nm)
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 E3R2


Fase exponencial OD E3R2
Abs(nm)
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
400
450
500
550
600
650
700
750
800
Tiempo(min)
OD control E3R2

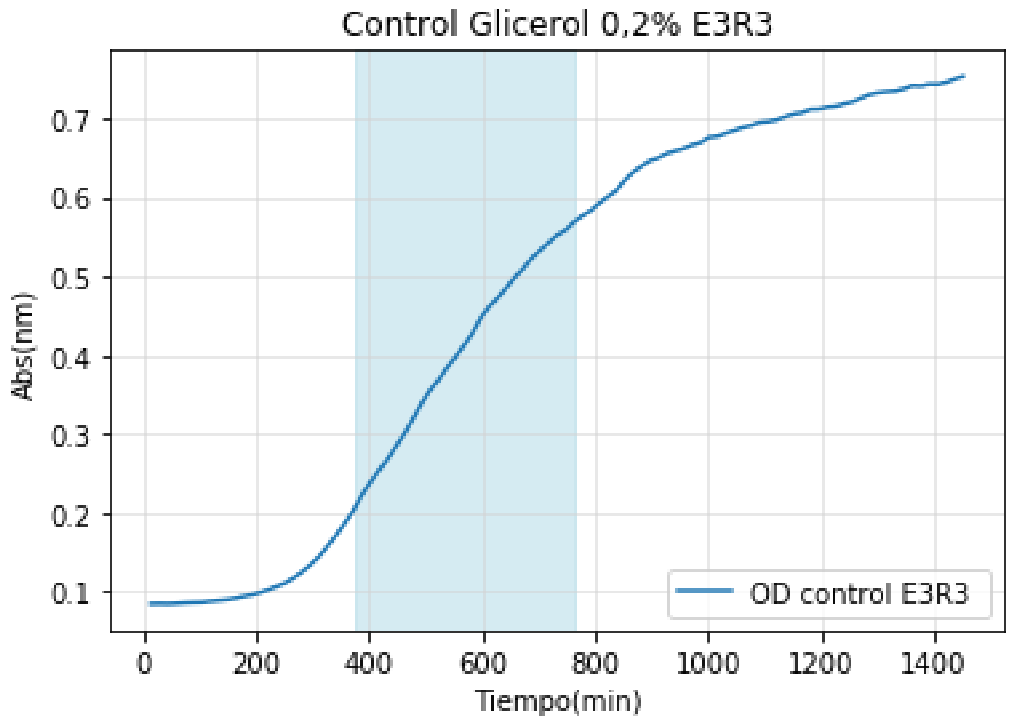
In [11]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 3
y6= np.log(odcgl3)-np.log(np.min(odcgl3))
print('Min OD = %e'%((np.min(odcgl3))))
evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 0,2% E3R3')
A6= params[0]
um6=params[1]
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[24]
y2=tt[50]
plt.figure()
plt.title('Control Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl3,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:51],odcgl3[24:51],label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

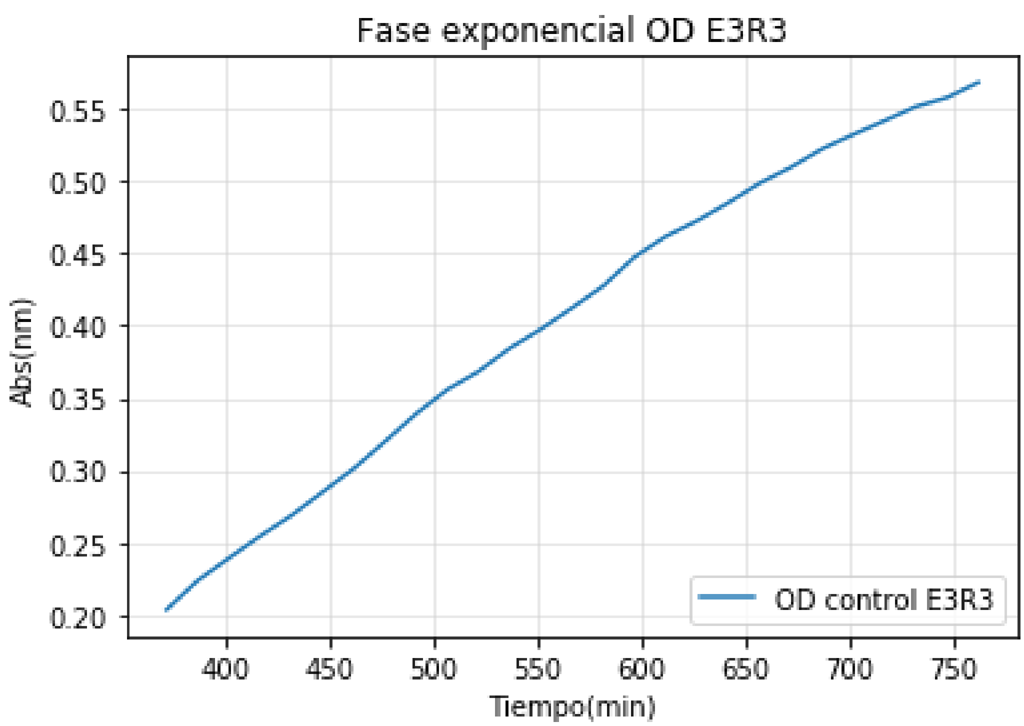
```
Min OD = 8.500000e-02
[  2.13297340e+00   4.59461284e-03   1.88519216e+02]
```




```
A=2.132973e+00
um=4.594613e-03
l=1.885192e+02
Tm=3.593012e+02
doubpe=1.508608e+02
ext=3.771521e+02
Tfinal=7.364533e+02
```

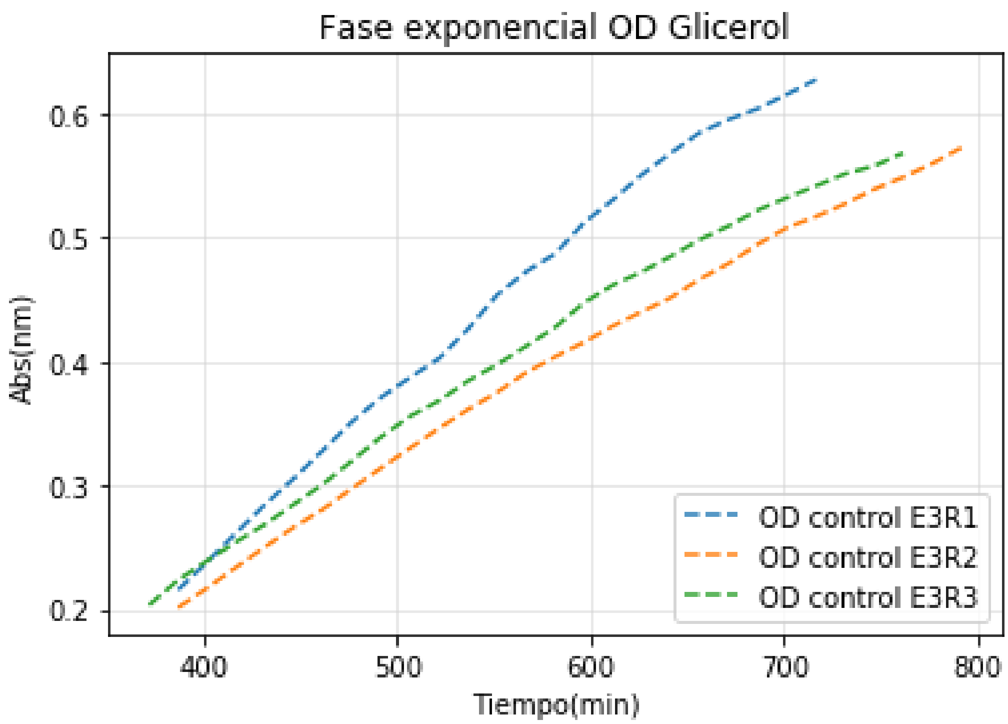
Out[11]: <matplotlib.legend.Legend at 0x1d719b0edd8>


Control Glicerol 0,2% E3R3
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 E3R3


Fase exponencial OD E3R3
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
400
450
500
550
600
650
700
750
Tiempo(min)
OD control E3R3

In [12]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glicerol')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:48],odcgl1[25:48],'--',label='OD control E3R1')
plt.plot(tt[25:53],odcgl2[25:53],'--',label='OD control E3R2')
plt.plot(tt[24:51],odcgl3[24:51],'--',label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

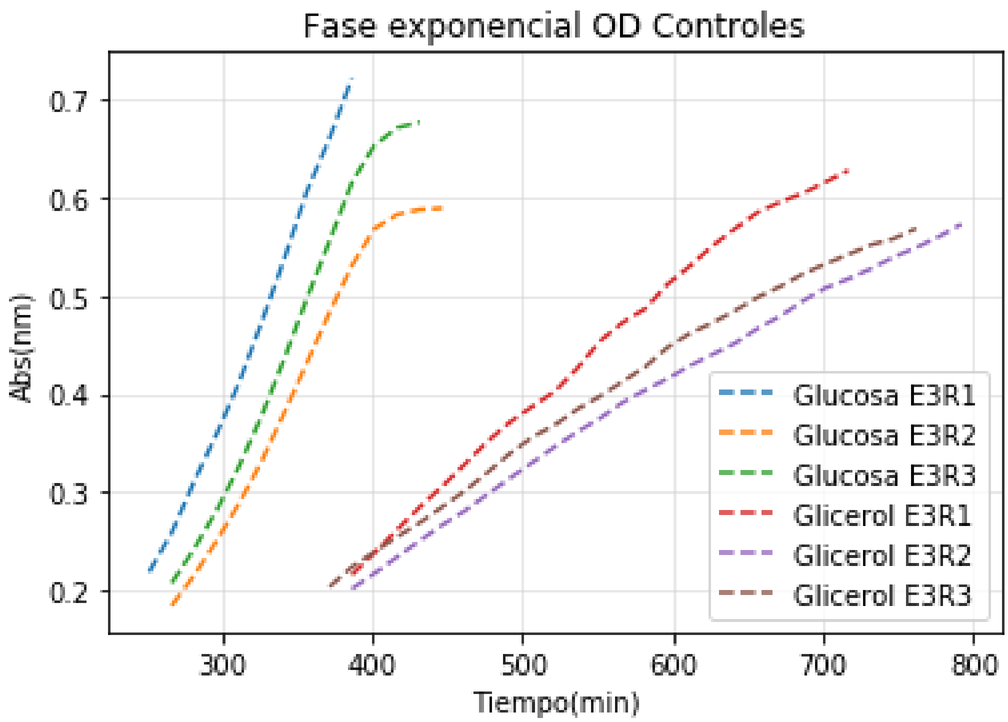
Out[12]: <matplotlib.legend.Legend at 0x1d71b196c88>



In [13]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Controles')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],odcg1[16:26],'--',label='Glucosa E3R1')
plt.plot(tt[17:30],odcg2[17:30],'--',label='Glucosa E3R2')
plt.plot(tt[17:29],odcg3[17:29],'--',label='Glucosa E3R3')
plt.plot(tt[25:48],odcgl1[25:48],'--',label='Glicerol E3R1')
plt.plot(tt[25:53],odcgl2[25:53],'--',label='Glicerol E3R2')
plt.plot(tt[24:51],odcgl3[24:51],'--',label='Glicerol E3R3')
```

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

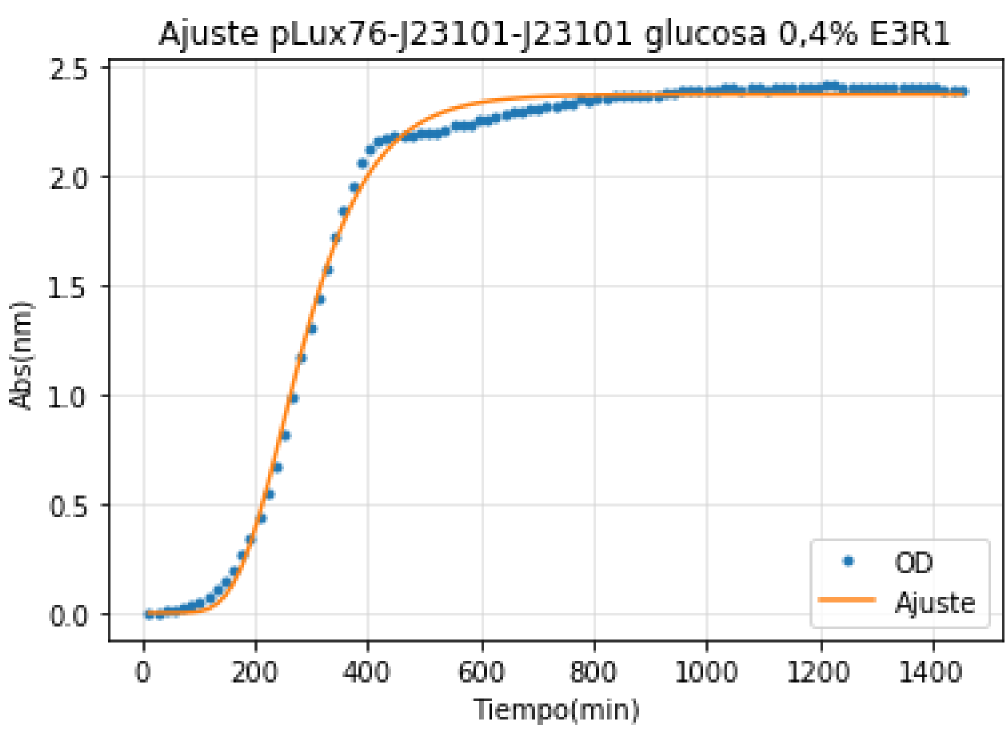
Out[13]: <matplotlib.legend.Legend at 0x1d71b255ef0>



In [14]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 1
y7= np.log(od18211)-np.log(np.min(od18211))
print('Min OD = %e'%((np.min(od18211))))
evaly, params=Function_fit(tt,y7,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4%
A7 = params[0]
um7=params[1]
l7=params[2]
print('A=%e'%(A7))
print('um=%e'%(um7))
print('l=%e'%(l7))

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

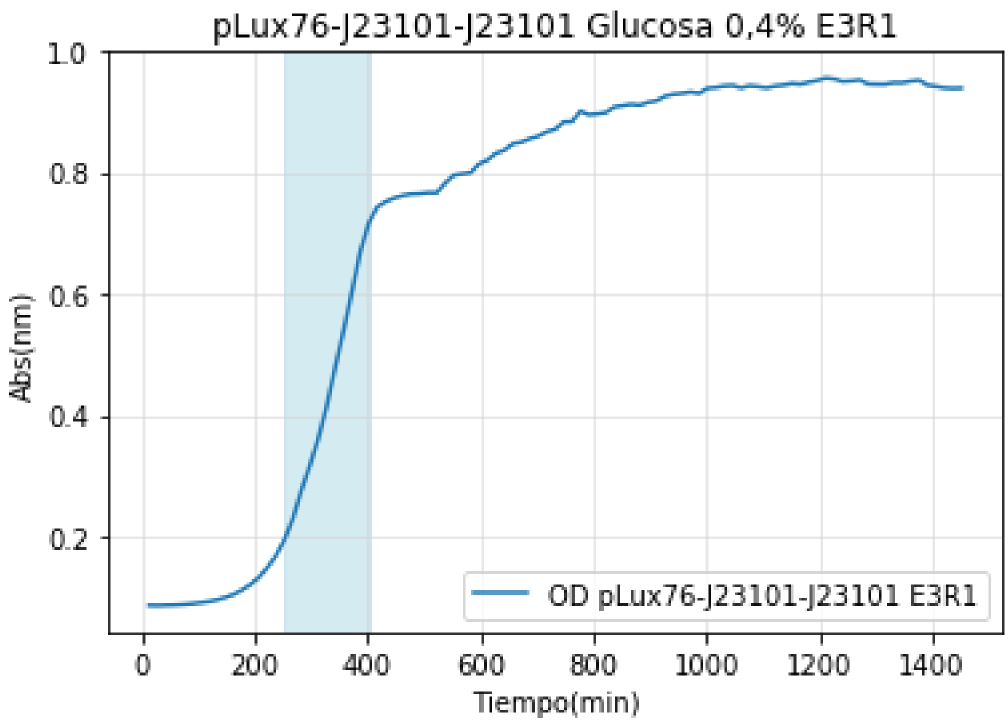
```
ttot7=tm7+extdp7
print('Tfinal=%e'%ttot7)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[16]
y2=tt[26]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18211,label='OD pLux76-J23101-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:27],od18211[16:27],label='OD pLux76-J23101-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

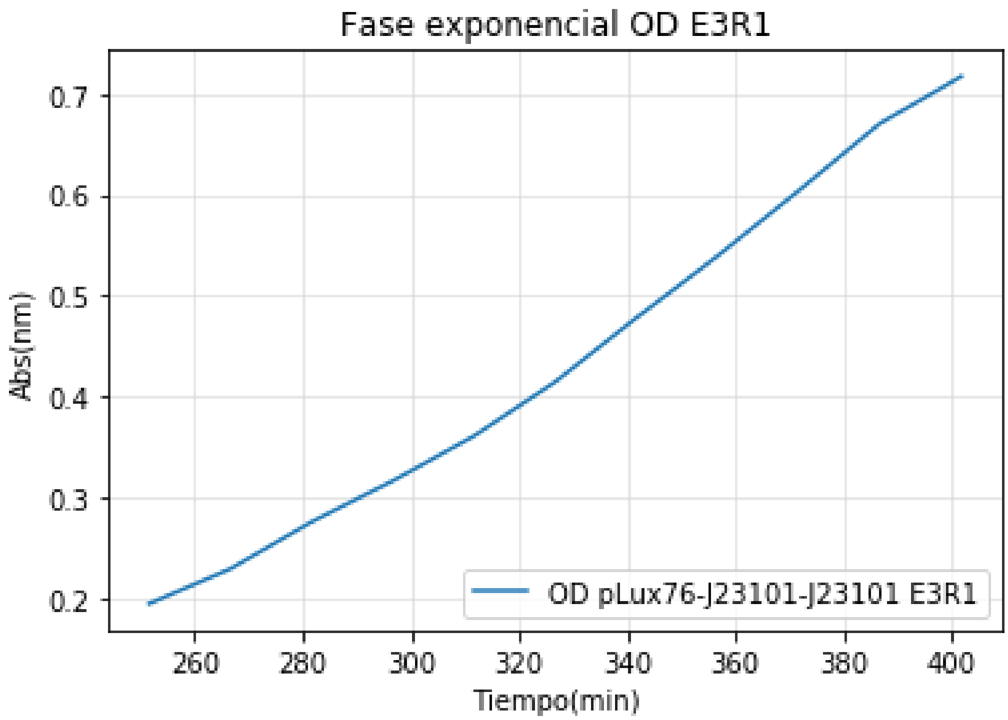
```
Min OD = 8.580000e-02
[  2.37037352e+00   1.03321767e-02   1.65952870e+02]
```



```
A=2.370374e+00
um=1.033218e-02
l=1.659529e+02
Tm=2.503505e+02
doubpe=6.708627e+01
ext=1.341725e+02
Tfinal=3.845231e+02
```

```
Out[14]: <matplotlib.legend.Legend at 0x1d71b0ca278>
```

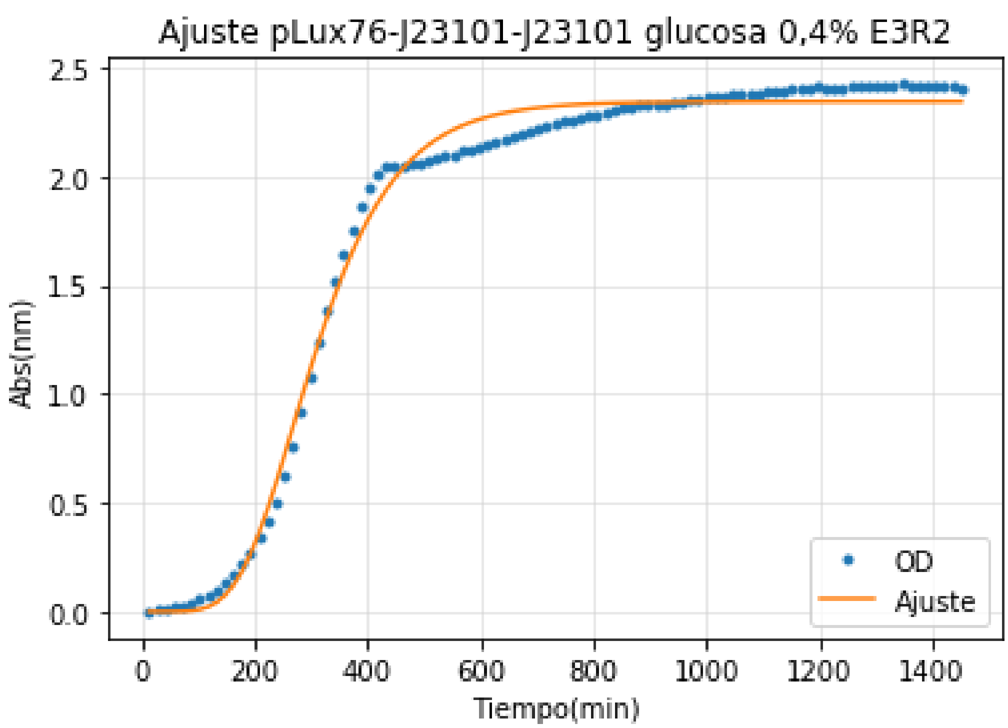




In [15]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 2
y8= np.log(od18212)-np.log(np.min(od18212))
print('Min OD = %e'%((np.min(od18212))))
evaly, params=Function_fit(tt,y8,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4%
A8= params[0]
um8=params[1]
l8=params[2]
print('A=%e'%(A8))
print('um=%e'%(um8))
print('l=%e'%(l8))

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

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

```python
y2=tt[29]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18212,label='OD pLux76-J23101-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:30],od18212[17:30],label='OD pLux76-J23101-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

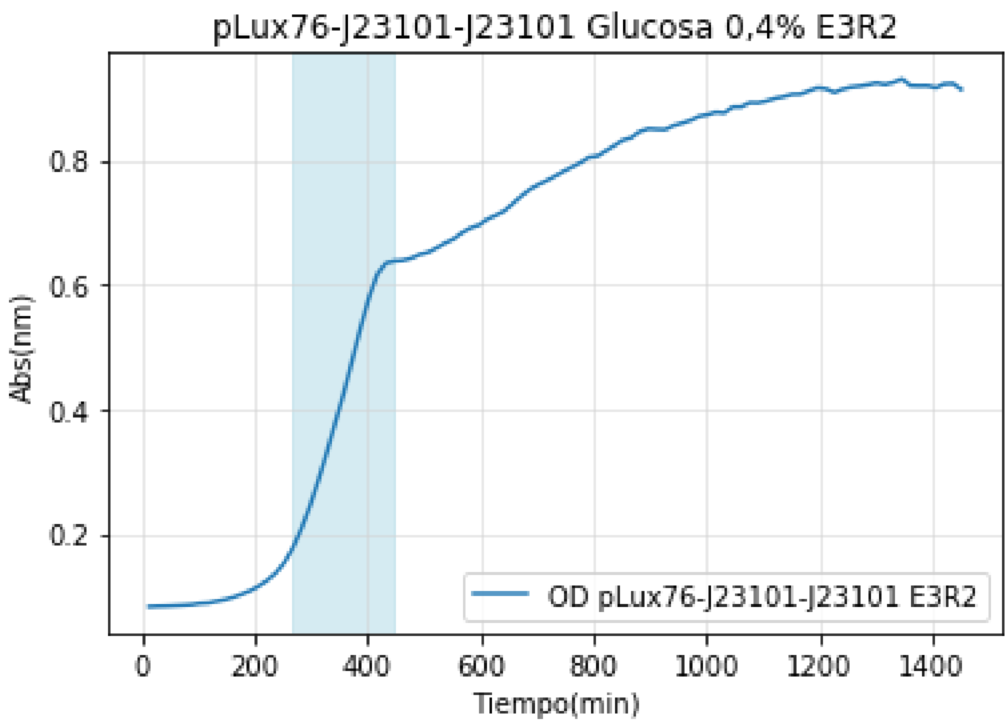
```
Min OD = 8.250000e-02
[  2.34308642e+00   8.73368317e-03   1.69674212e+02]
```

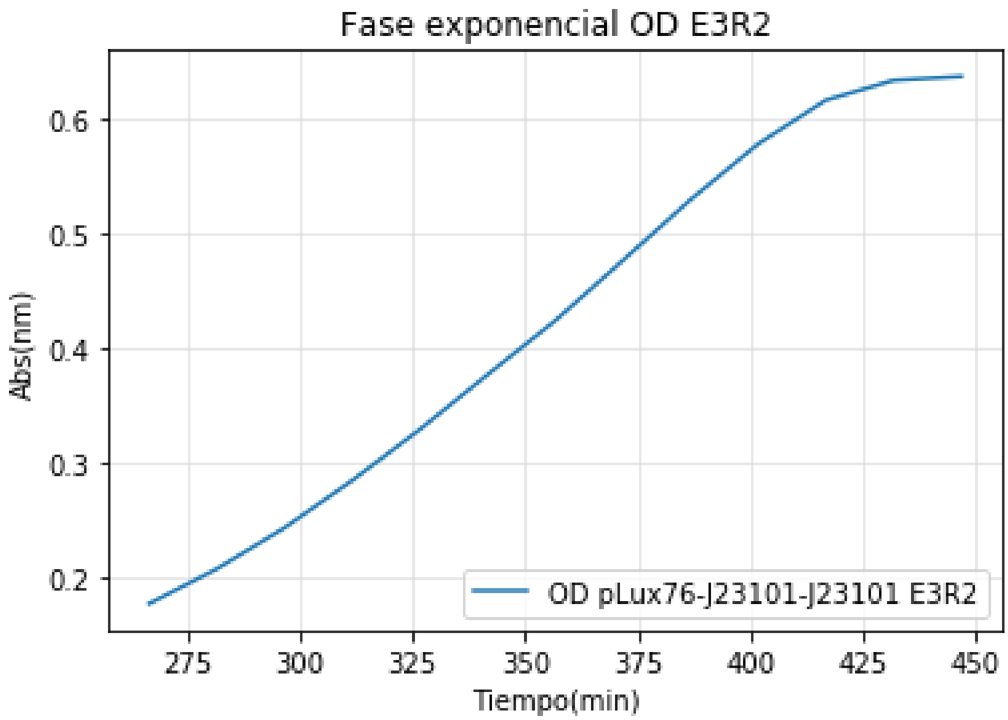



```
A=2.343086e+00
um=8.733683e-03
```

```
l=1.696742e+02
Tm=2.683695e+02
doubpe=7.936482e+01
ext=1.587296e+02
Tfinal=4.270991e+02
```

```
Out[15]: <matplotlib.legend.Legend at 0x1d71acbc9b0>
```

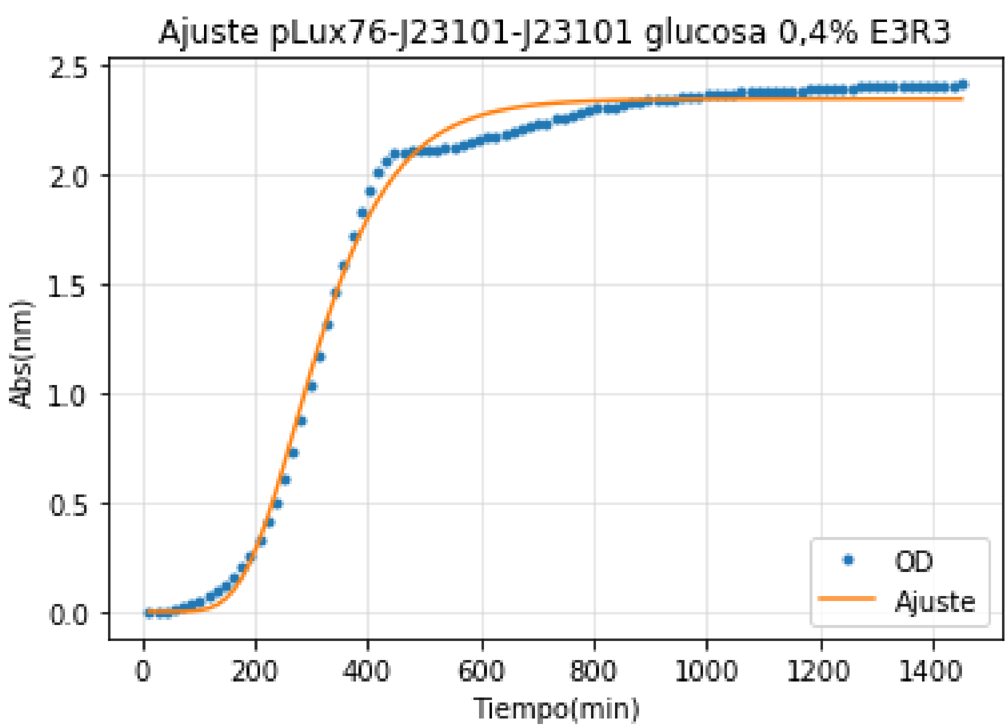





```
In [16]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-std-std glucosa rep 3
         y9= np.log(od18213)-np.log(np.min(od18213))
         print('Min OD = %e'%((np.min(od18213))))
         evaly, params=Function_fit(tt,y9,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4%
         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[18]
```

```
y2=tt[29]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18213,label='OD pLux76-J23101-J23101 E3R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:30],od18213[18:30],label='OD pLux76-J23101-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

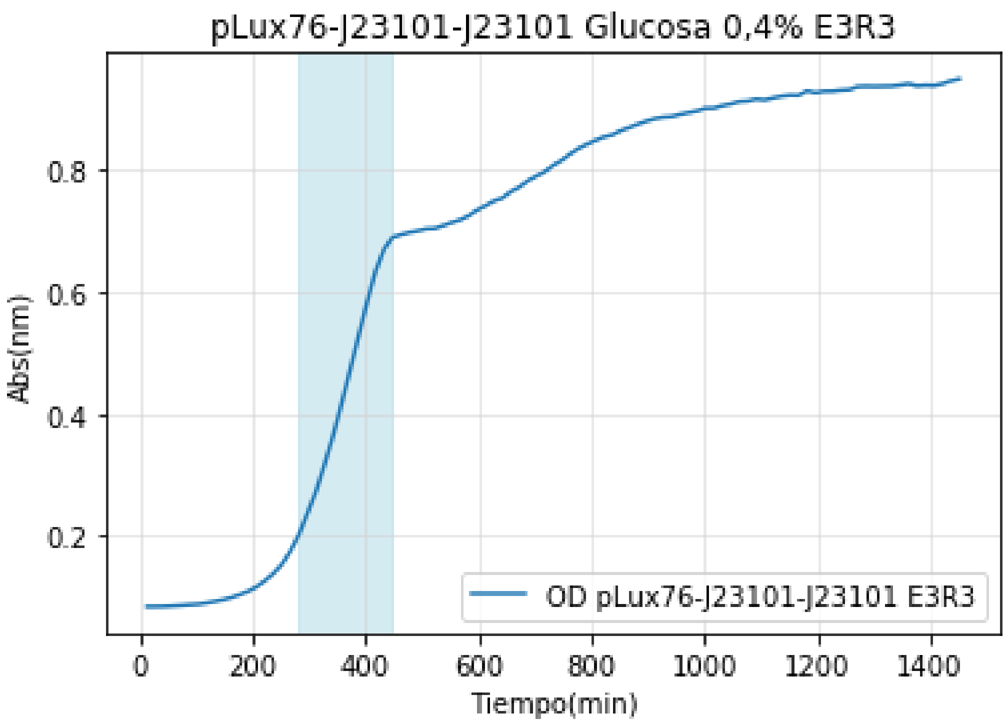
```
Min OD = 8.460000e-02
[  2.34879777e+00   9.01713157e-03   1.76447498e+02]
```

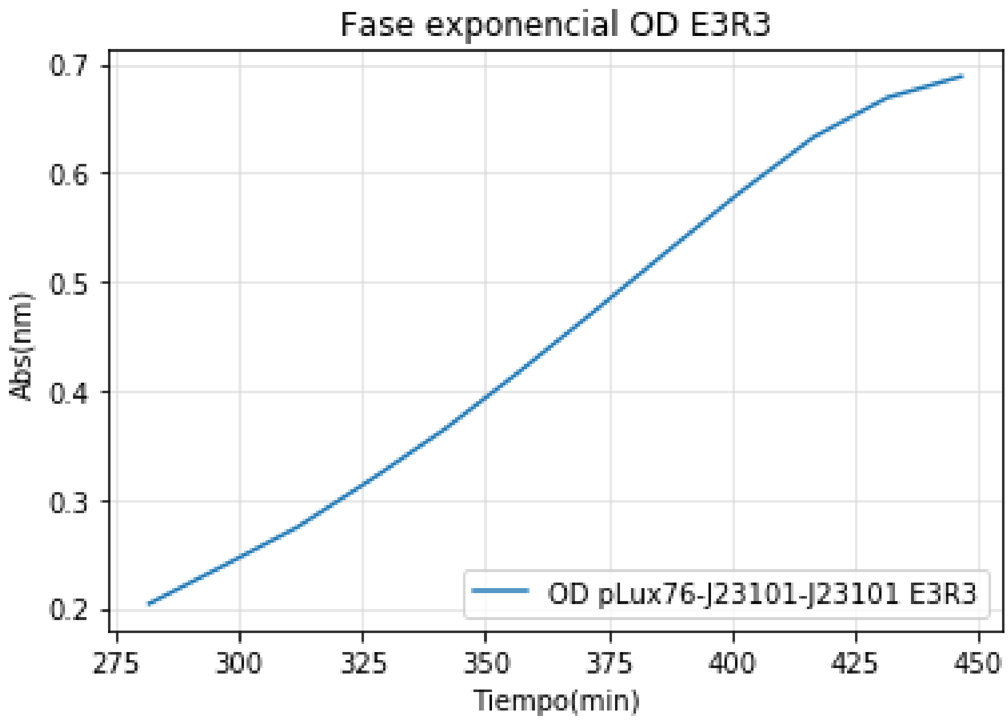



```
A=2.348798e+00
um=9.017132e-03
```

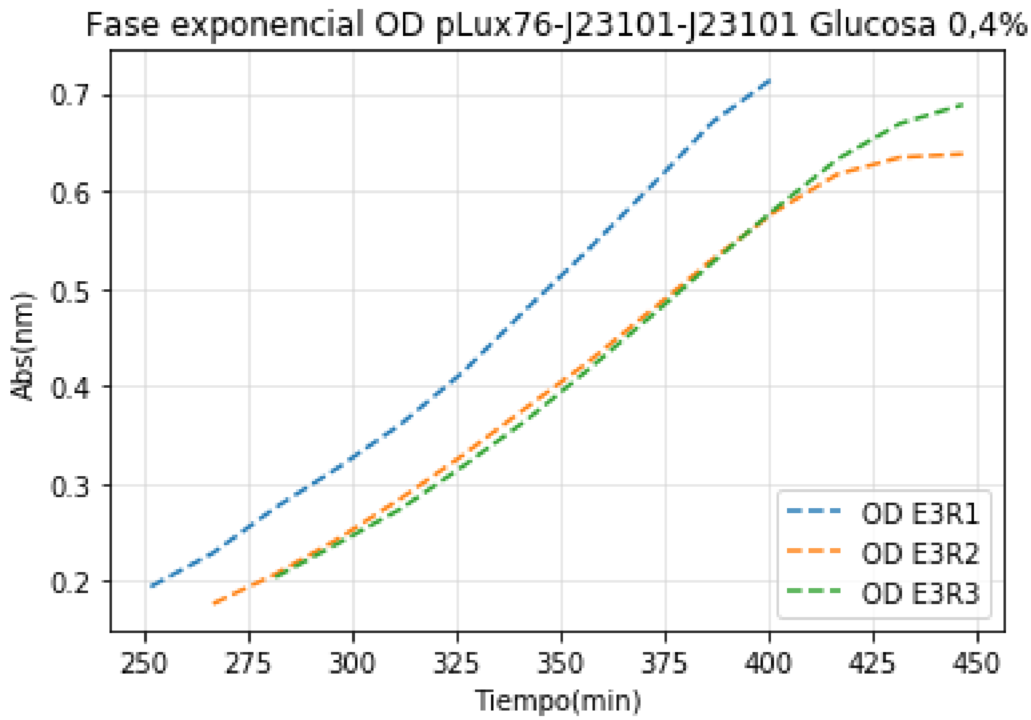
```
l=1.764475e+02
Tm=2.722734e+02
doubpe=7.687003e+01
ext=1.537401e+02
Tfinal=4.260134e+02
```

Out[16]: <matplotlib.legend.Legend at 0x1d71b3cbe80>





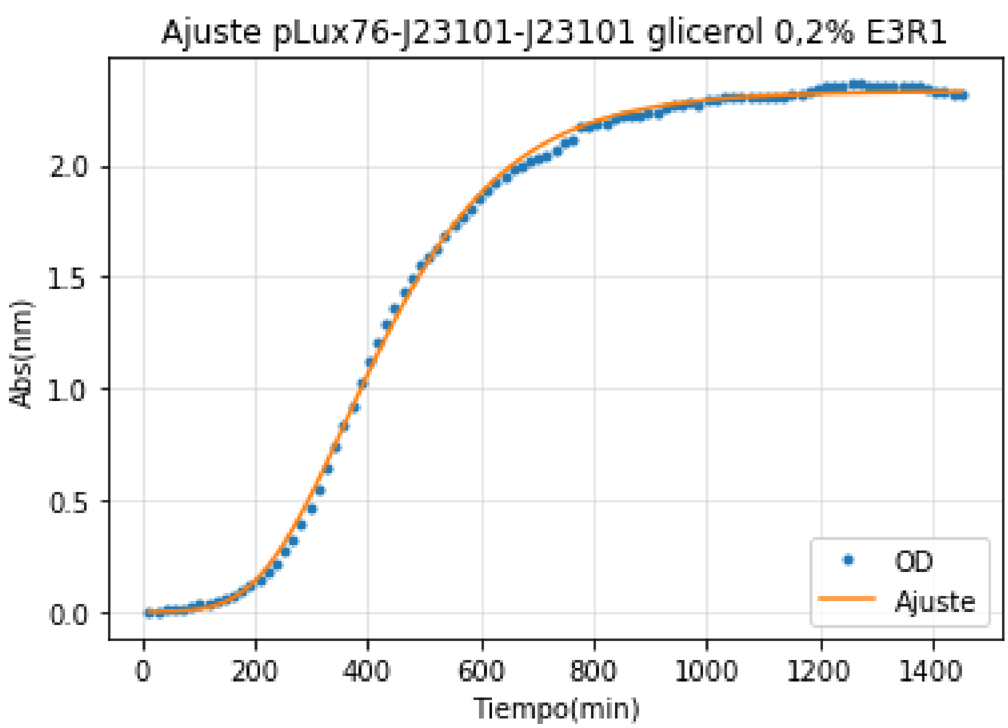

```
In [17]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],od18211[16:27],'--',label='OD E3R1')
         plt.plot(tt[17:30],od18212[17:30],'--',label='OD E3R2')
         plt.plot(tt[18:30],od18213[18:30],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[17]: <matplotlib.legend.Legend at 0x1d71b4ebbe0>
```



In [18]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glicerol rep 1
y10= np.log(od1821g1)-np.log(np.min(od1821g1))
print('Min OD = %e'%((np.min(od1821g1))))
evaly, params=Function_fit(tt,y10,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
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[24]
```

```python
y2=tt[46]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1821g1,label='OD pLux76-J23101-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:47],od1821g1[24:47],label='OD pLux76-J23101-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

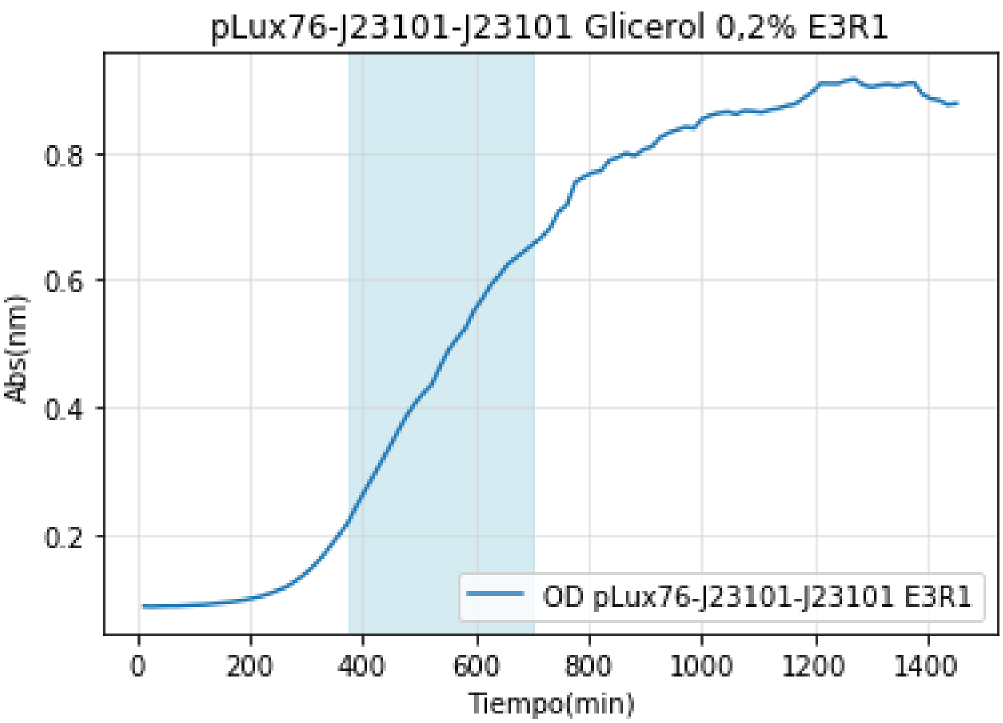
```
Min OD = 8.620000e-02
[  2.33021092e+00   5.48552233e-03   2.05388922e+02]
```

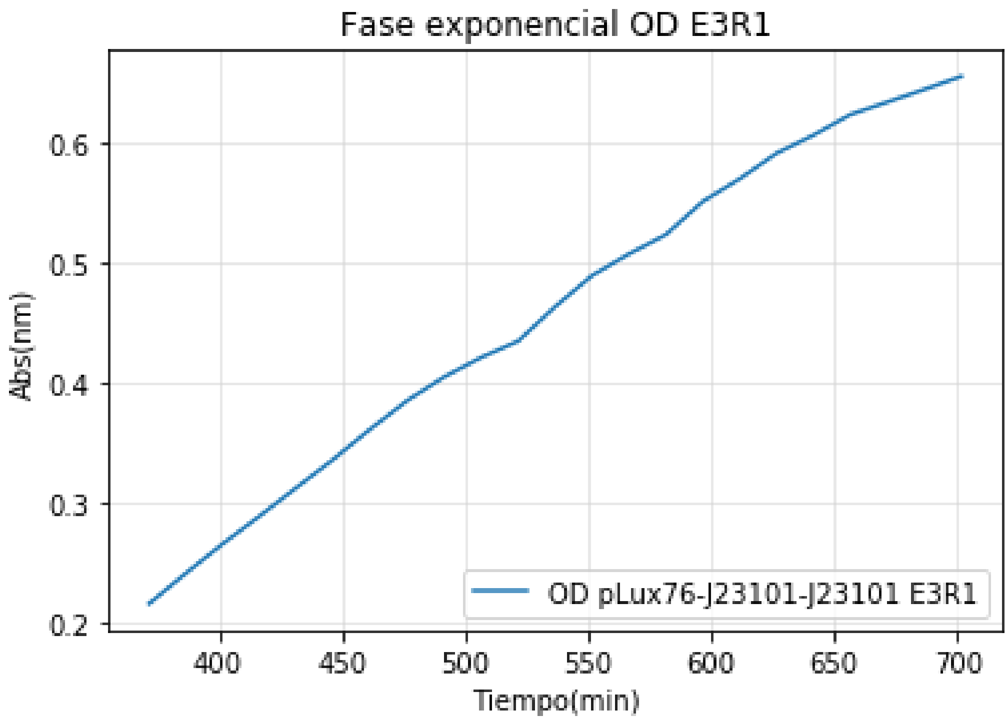



```
A=2.330211e+00
um=5.485522e-03
```

```
l=2.053889e+02
Tm=3.616615e+02
doubpe=1.263594e+02
ext=3.158984e+02
Tfinal=6.775599e+02
```

Out[18]: <matplotlib.legend.Legend at 0x1d71b68bdd8>

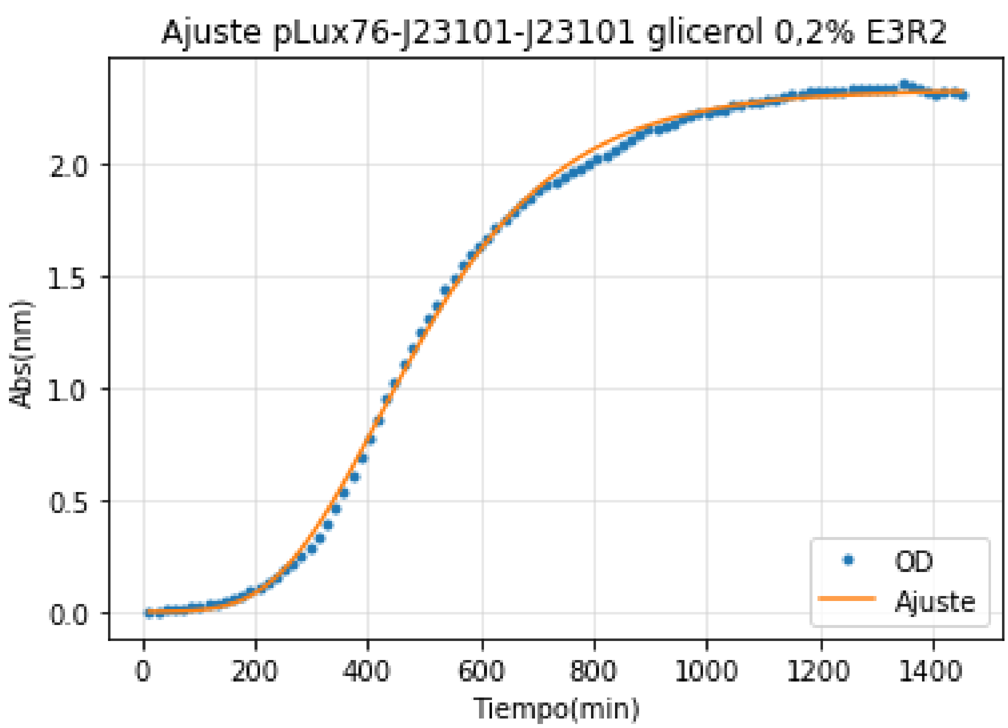





```
In [19]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-std-std glicerol rep 2
         y11= np.log(od1821g2)-np.log(np.min(od1821g2))
         print('Min OD = %e'%((np.min(od1821g2))))
         evaly, params=Function_fit(tt,y11,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
         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[29]
```

```python
y2=tt[53]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1821g2,label='OD pLux76-J23101-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[29:54],od1821g2[29:54],label='OD pLux76-J23101-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

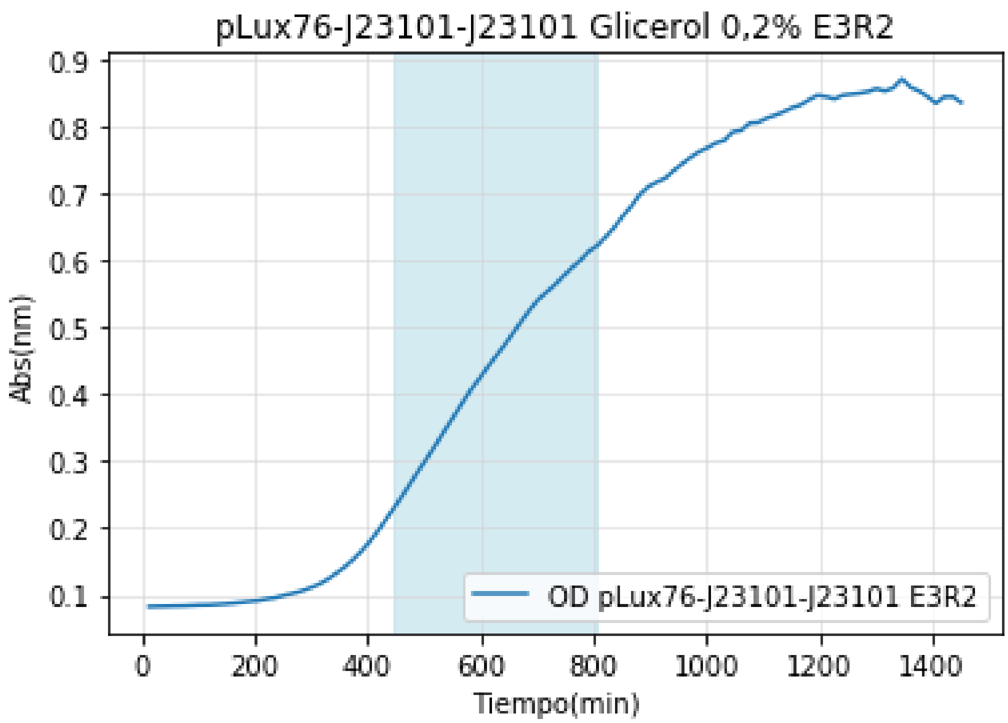
```
Min OD = 8.260000e-02
[  2.33071292e+00   4.73754815e-03   2.36596064e+02]
```

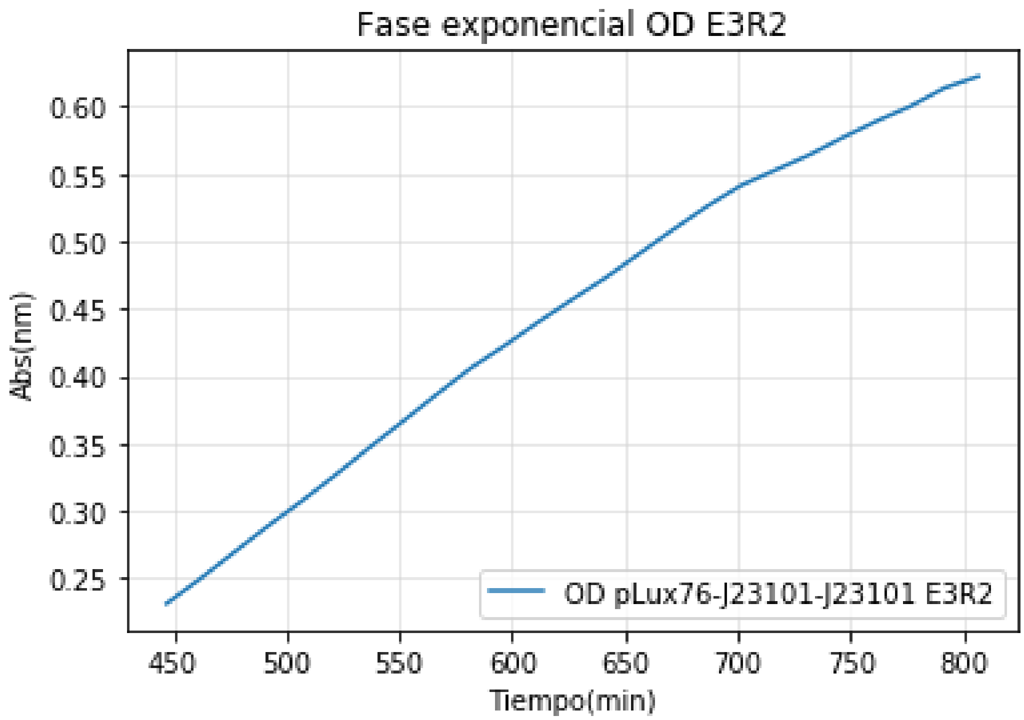



```
A=2.330713e+00
um=4.737548e-03
```

```
l=2.365961e+02
Tm=4.175803e+02
doubpe=1.463093e+02
ext=3.657732e+02
Tfinal=7.833534e+02
```

```
Out[19]: <matplotlib.legend.Legend at 0x1d71ae3ef28>
```

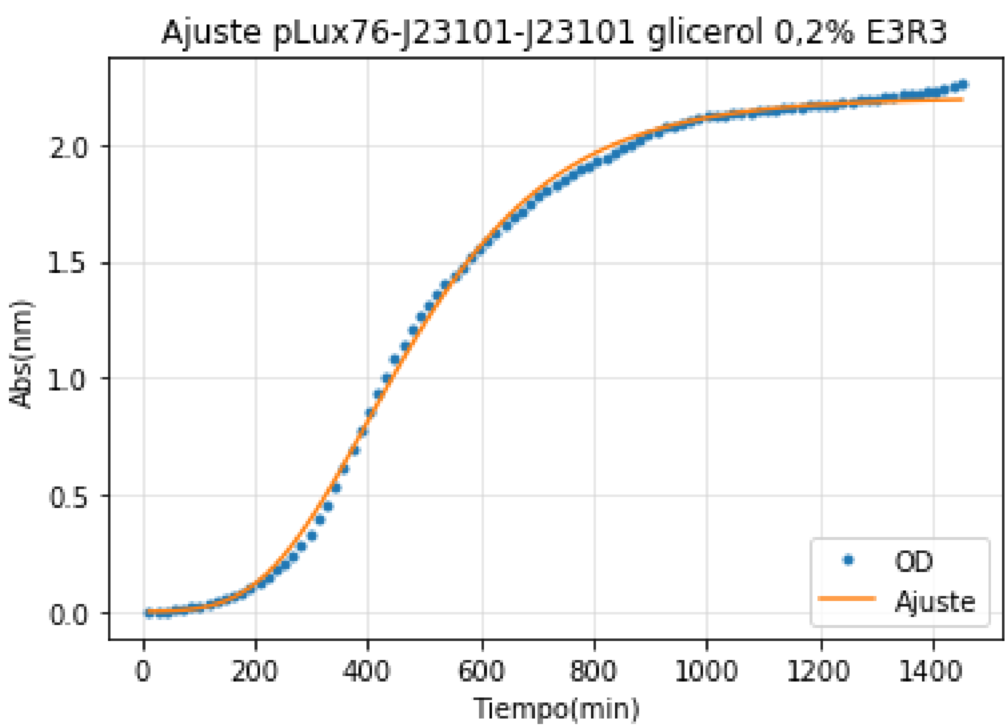




In [20]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glicerol rep 3
y12= np.log(od1821g3)-np.log(np.min(od1821g3))
print('Min OD = %e'%((np.min(od1821g3))))
evaly, params=Function_fit(tt,y12,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
A12= 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[26]
```

```
y2=tt[49]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1821g3,label='OD pLux76-J23101-J23101 E3R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:50],od1821g3[26:50],label='OD pLux76-J23101-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

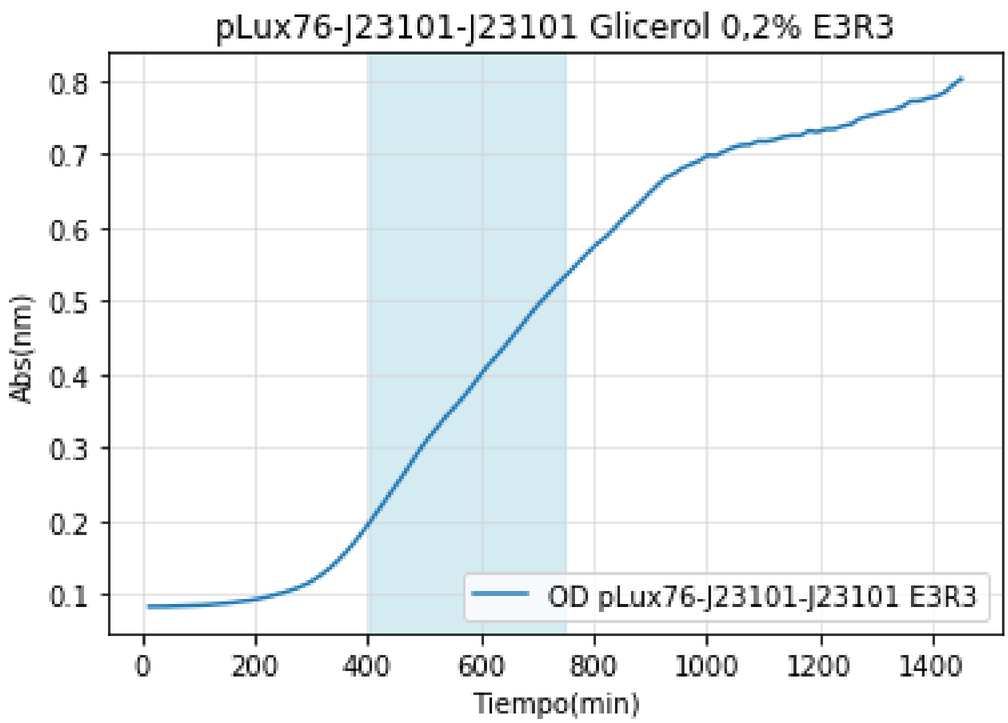
```
Min OD = 8.390000e-02
[  2.19954398e+00   4.34680854e-03   2.12218285e+02]
```

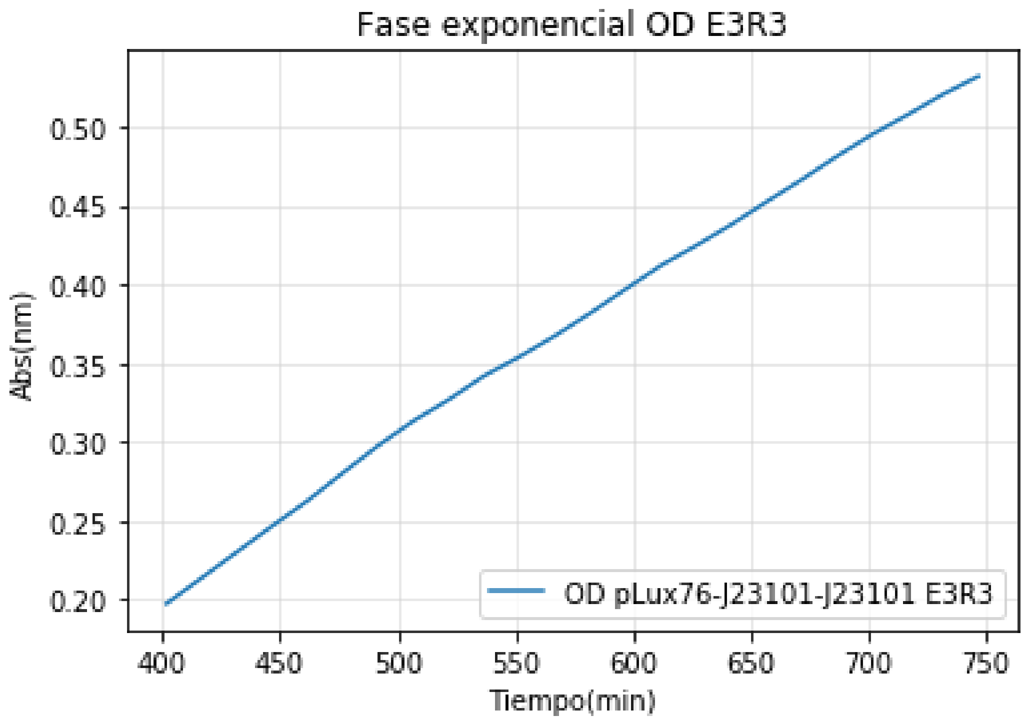



```
A=2.199544e+00
um=4.346809e-03
```

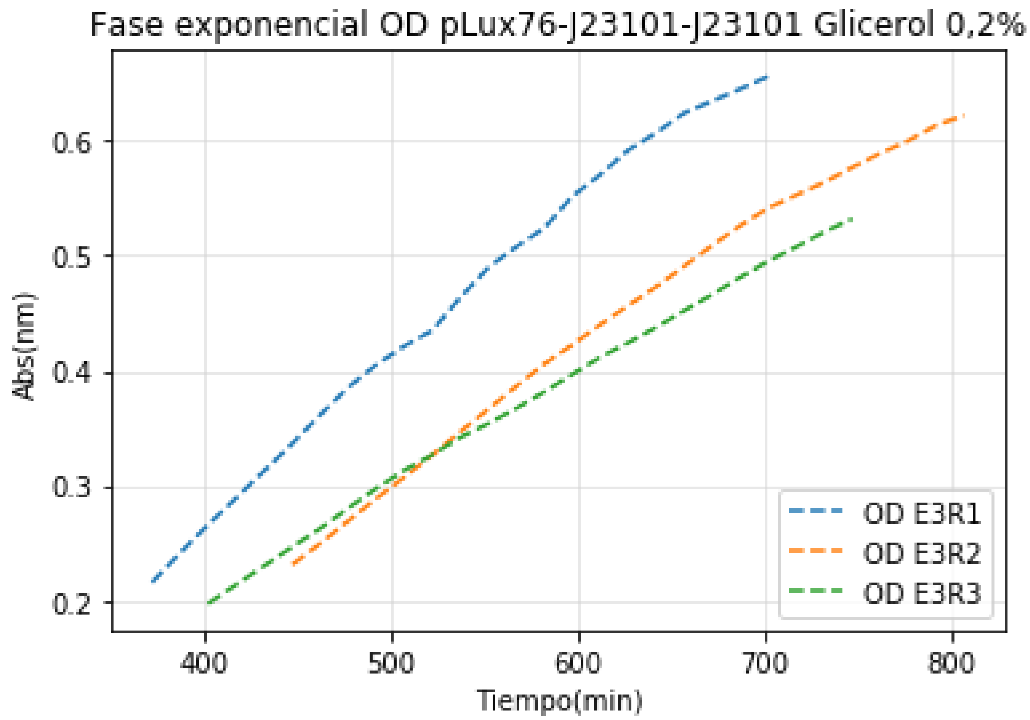
```
l=2.122183e+02
Tm=3.983703e+02
doubpe=1.594612e+02
ext=3.189223e+02
Tfinal=7.172926e+02
```

Out[20]: <matplotlib.legend.Legend at 0x1d71af3f320>

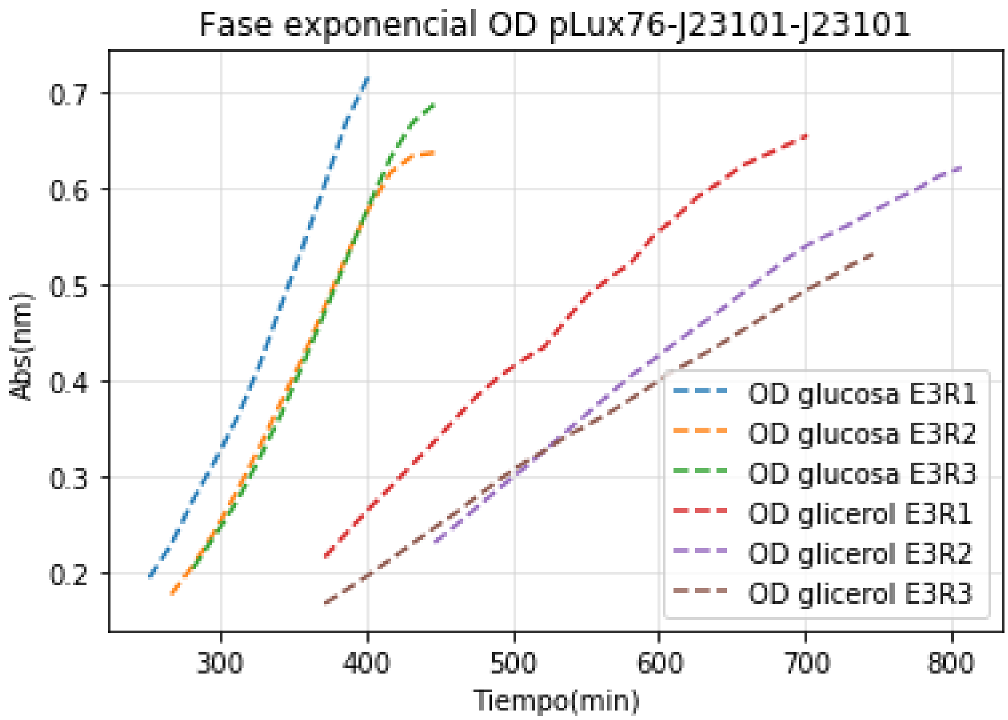





```
In [21]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[24:47],od1821g1[24:47],'--',label='OD E3R1')
         plt.plot(tt[29:54],od1821g2[29:54],'--',label='OD E3R2')
         plt.plot(tt[26:50],od1821g3[26:50],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[21]: <matplotlib.legend.Legend at 0x1d71b7babe0>
```



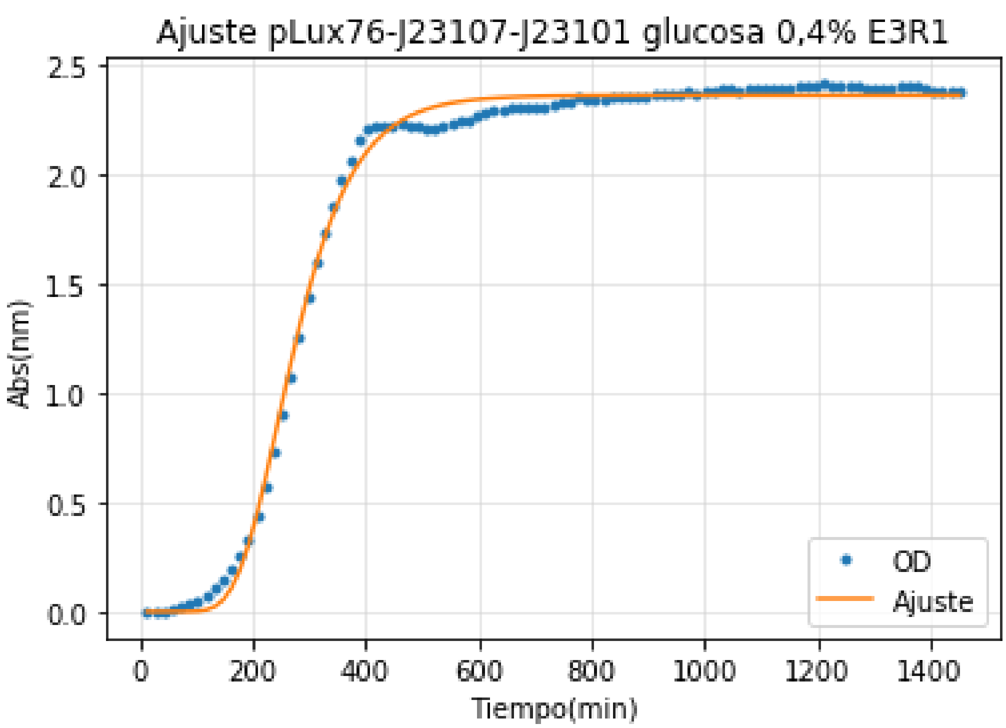

```
In [22]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],od18211[16:27],'--',label='OD glucosa E3R1')
         plt.plot(tt[17:30],od18212[17:30],'--',label='OD glucosa E3R2')
         plt.plot(tt[18:30],od18213[18:30],'--',label='OD glucosa E3R3')
         plt.plot(tt[24:47],od1821g1[24:47],'--',label='OD glicerol E3R1')
         plt.plot(tt[29:54],od1821g2[29:54],'--',label='OD glicerol E3R2')
         plt.plot(tt[24:50],od1821g3[24:50],'--',label='OD glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[22]: <matplotlib.legend.Legend at 0x1d71b87e6a0>
```



In [23]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-107-std glucosa rep 1
y13= np.log(od18231)-np.log(np.min(od18231))
print('Min OD = %e'%((np.min(od18231))))
evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste pLux76-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[16]
```

```python
y2=tt[25]
plt.figure()
plt.title('pLux76-J23107-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18231,label='OD pLux76-J23107-J23101 E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od18231[16:26],label='OD pLux76-J23107-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

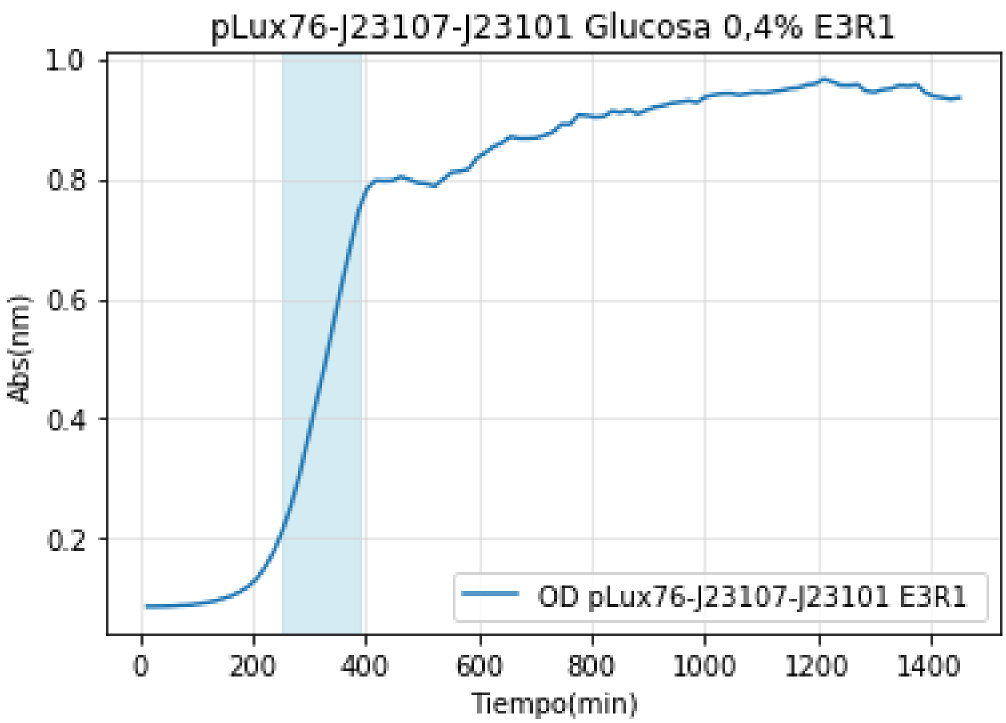
```
Min OD = 8.640000e-02
[  2.36341157e+00   1.18901032e-02   1.70723410e+02]
```

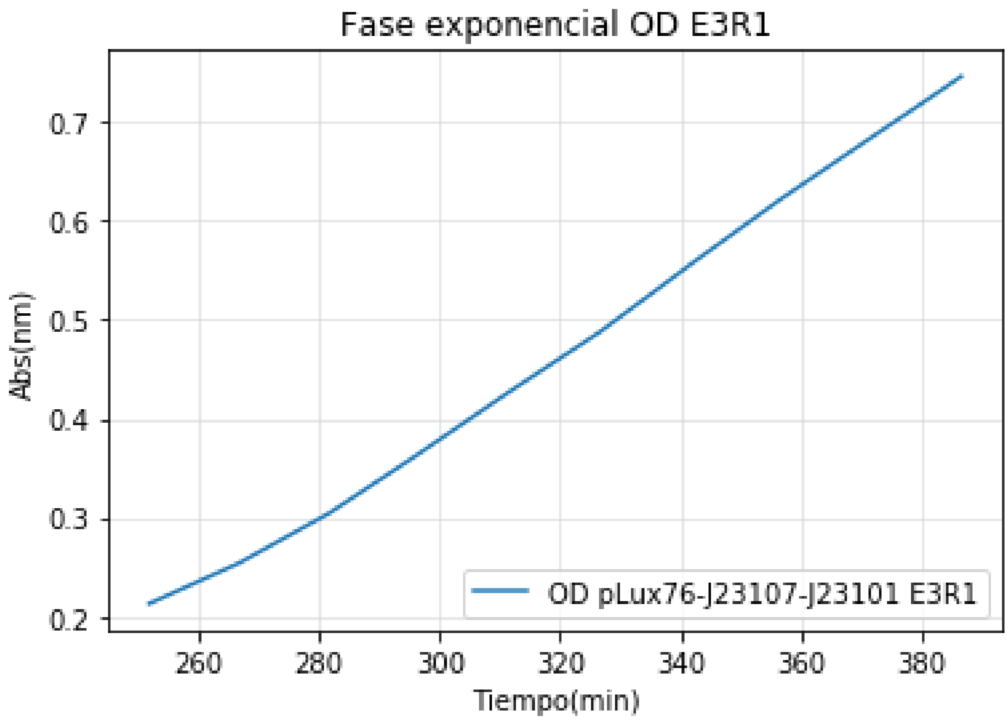



```
A=2.363412e+00
um=1.189010e-02
```

```
l=1.707234e+02
Tm=2.438473e+02
doubpe=5.829614e+01
ext=1.165923e+02
Tfinal=3.604396e+02
```

```
Out[23]: <matplotlib.legend.Legend at 0x1d71b3383c8>
```

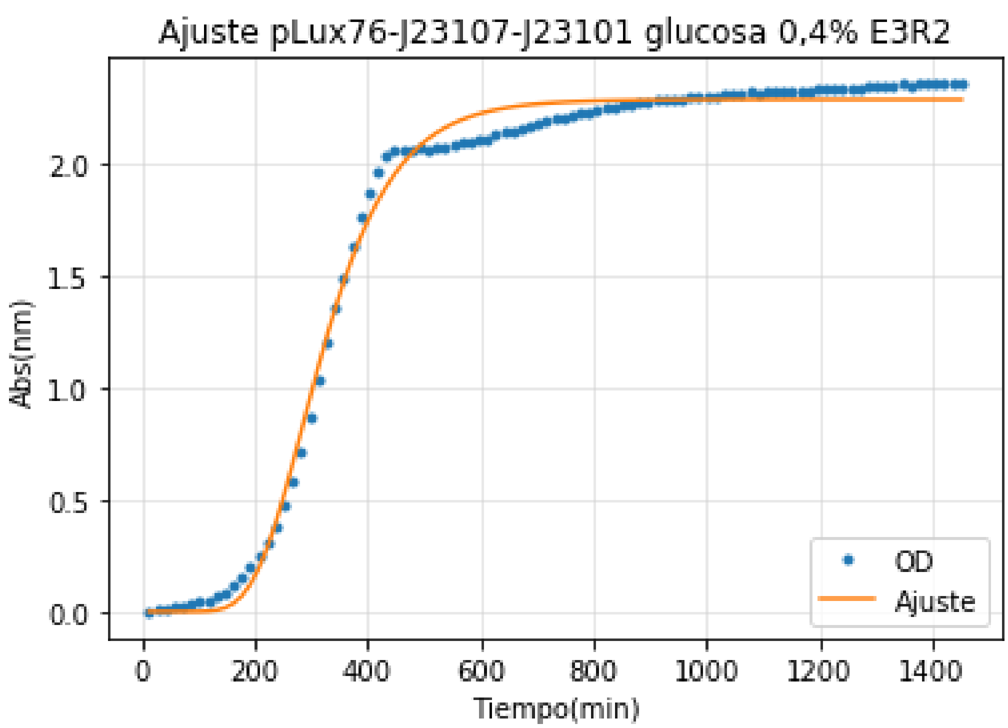




In [24]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-107-std glucosa rep 2
y14= np.log(od18232)-np.log(np.min(od18232))
print('Min OD = %e'%((np.min(od18232))))
evaly, params=Function_fit(tt,y14,0,-1,title = 'Ajuste pLux76-J23107-J23101 glucosa 0,4
A14= params[0]
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[19]
```

```python
y2=tt[29]
plt.figure()
plt.title('pLux76-J23107-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18232,label='OD pLux76-J23107-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],od18232[19:30],label='OD pLux76-J23107-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

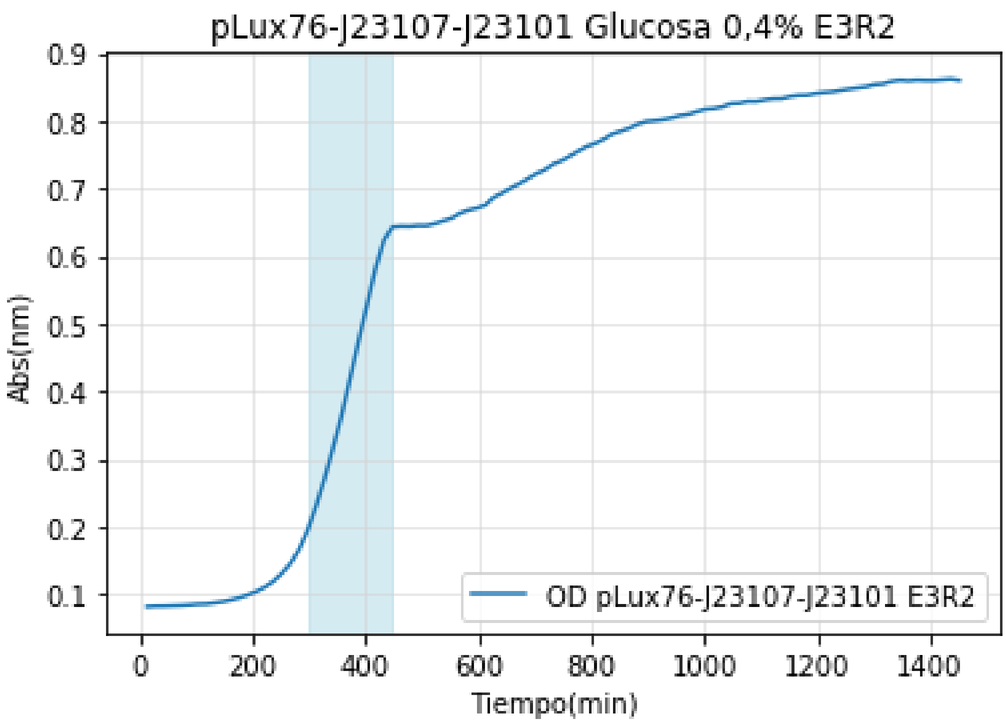
```
Min OD = 8.140000e-02
[ 2.29077880e+00  9.63771897e-03  1.97867927e+02]
```

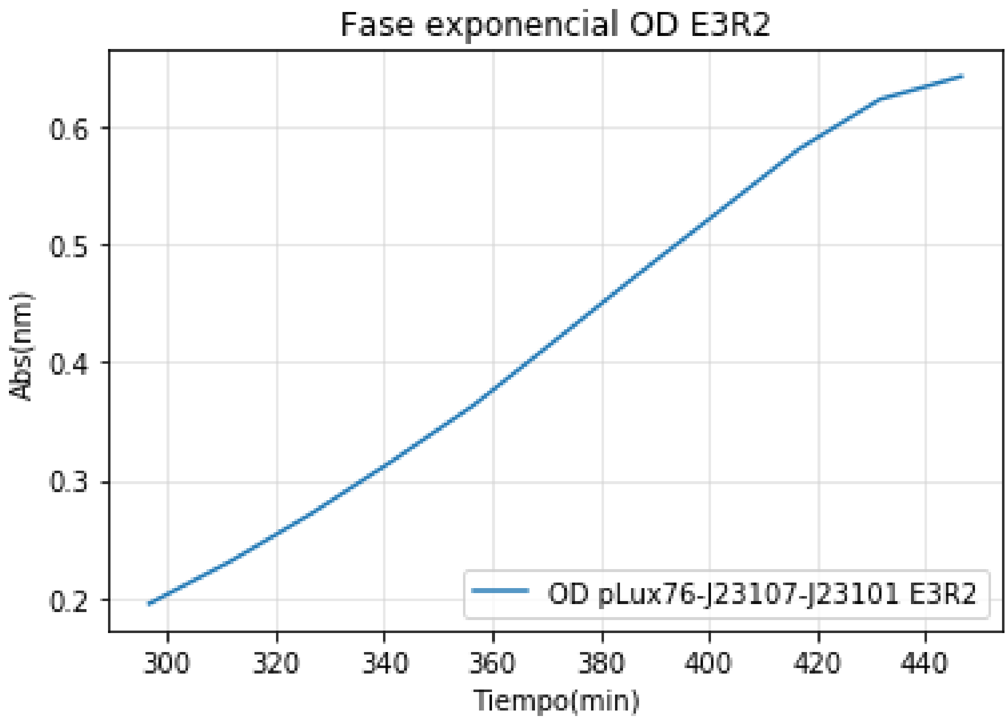



```
A=2.290779e+00
um=9.637719e-03
```

```
l=1.978679e+02
Tm=2.853088e+02
doubpe=7.192025e+01
ext=1.438405e+02
Tfinal=4.291493e+02
```

Out[24]: <matplotlib.legend.Legend at 0x1d71af71630>

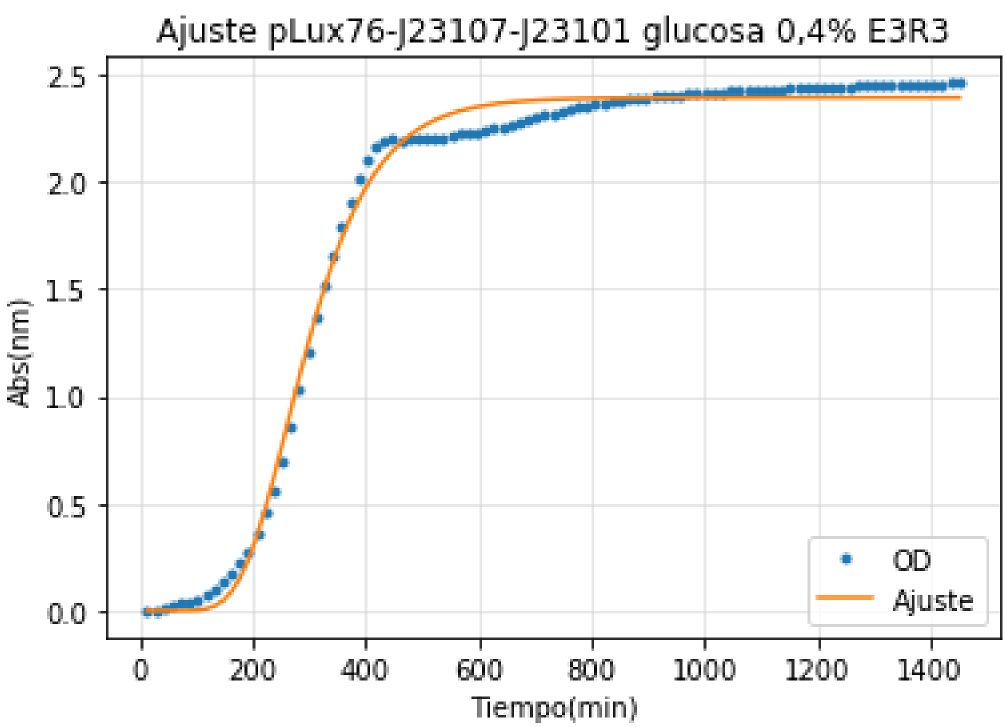




In [25]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-107-std glucosa rep 3
y15= np.log(od18233)-np.log(np.min(od18233))
print('Min OD = %e'%((np.min(od18233))))
evaly, params=Function_fit(tt,y15,0,-1,title = 'Ajuste pLux76-J23107-J23101 glucosa 0,4
A15= params[0]
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[27]
plt.figure()
plt.title('pLux76-J23107-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18233,label='OD pLux76-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 E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],od18233[17:28],label='OD pLux76-J23107-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

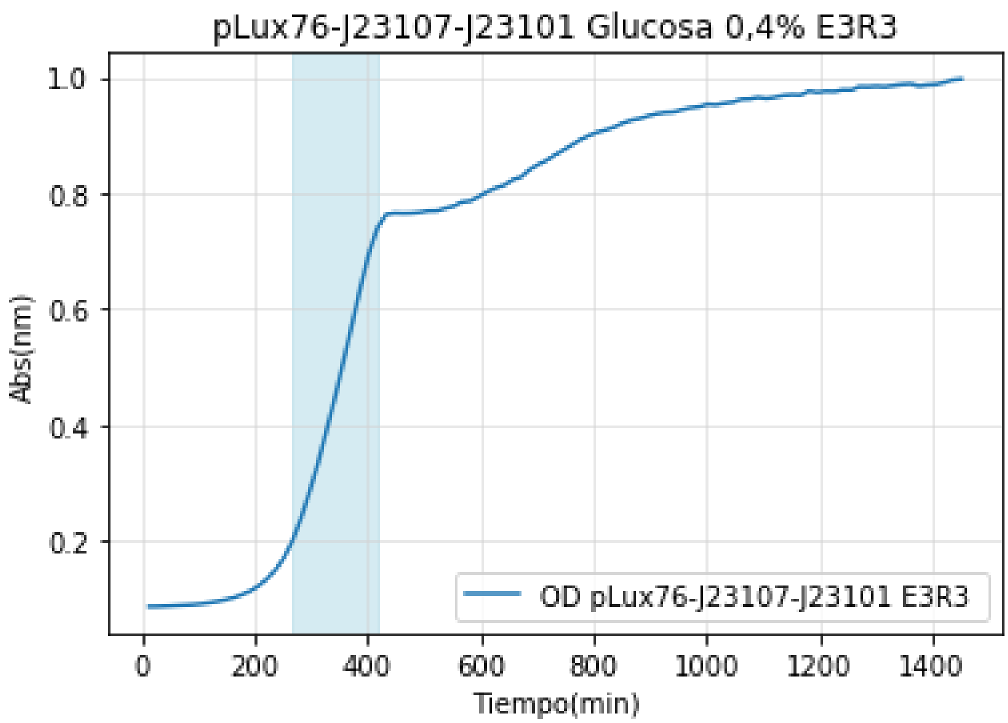
```
Min OD = 8.510000e-02
[  2.39424750e+00   1.04879647e-02   1.77610590e+02]
```

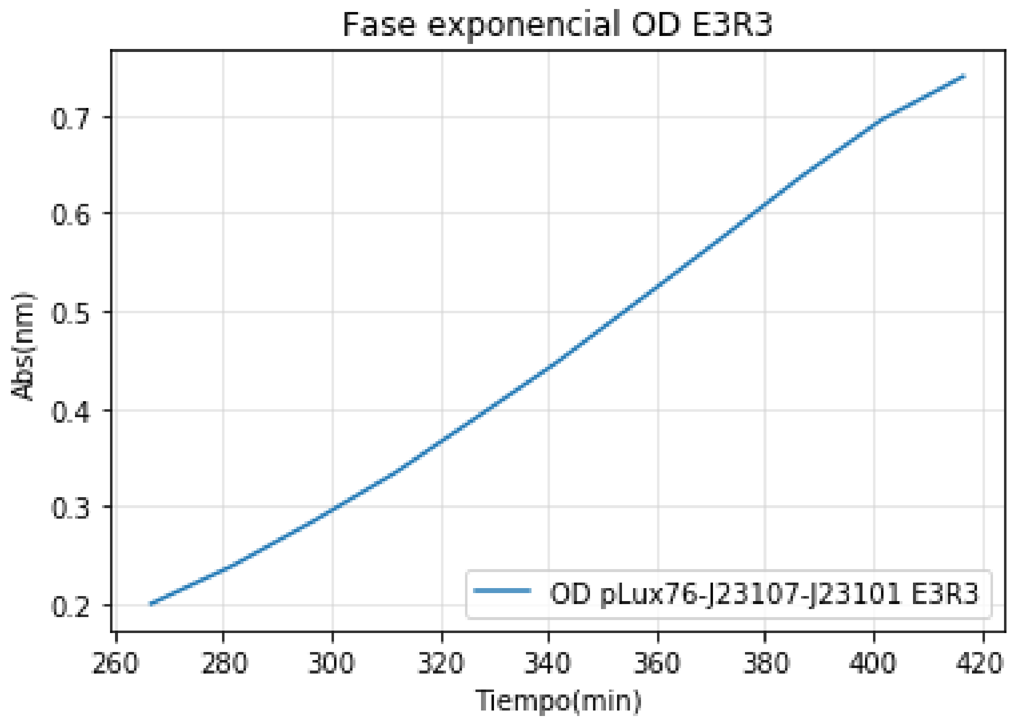



```
A=2.394248e+00
um=1.048796e-02
```

```
l=1.776106e+02
Tm=2.615920e+02
doubpe=6.608977e+01
ext=1.321795e+02
Tfinal=3.937716e+02
```

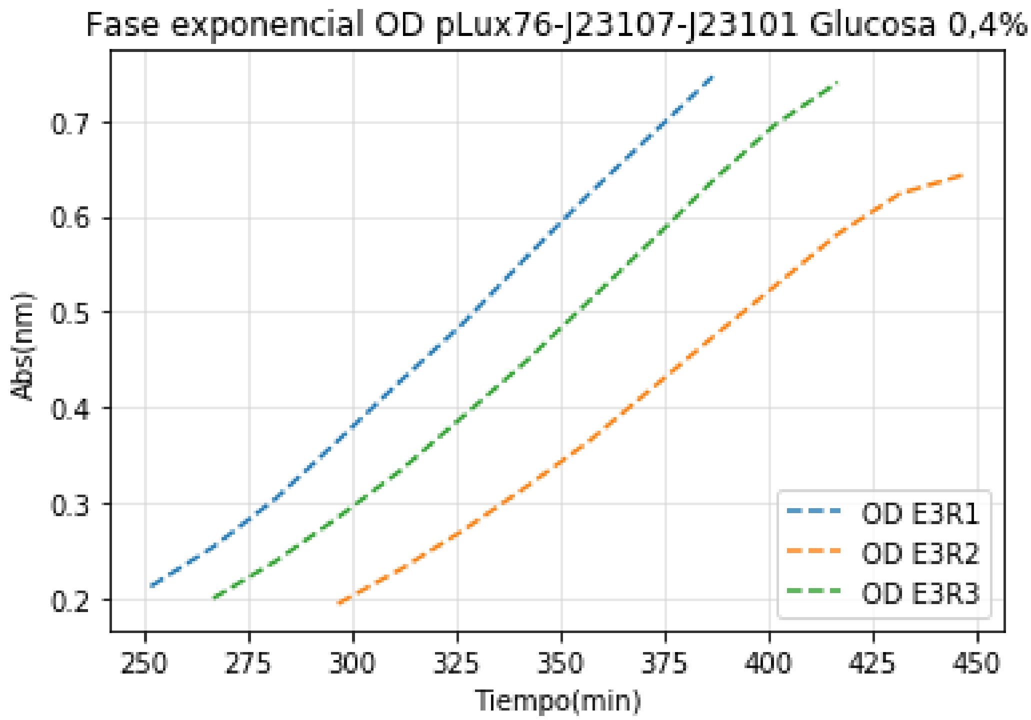
Out[25]: <matplotlib.legend.Legend at 0x1d71c9e08d0>






```
In [26]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23107-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],od18231[16:26],'--',label='OD E3R1')
         plt.plot(tt[19:30],od18232[19:30],'--',label='OD E3R2')
         plt.plot(tt[17:28],od18233[17:28],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

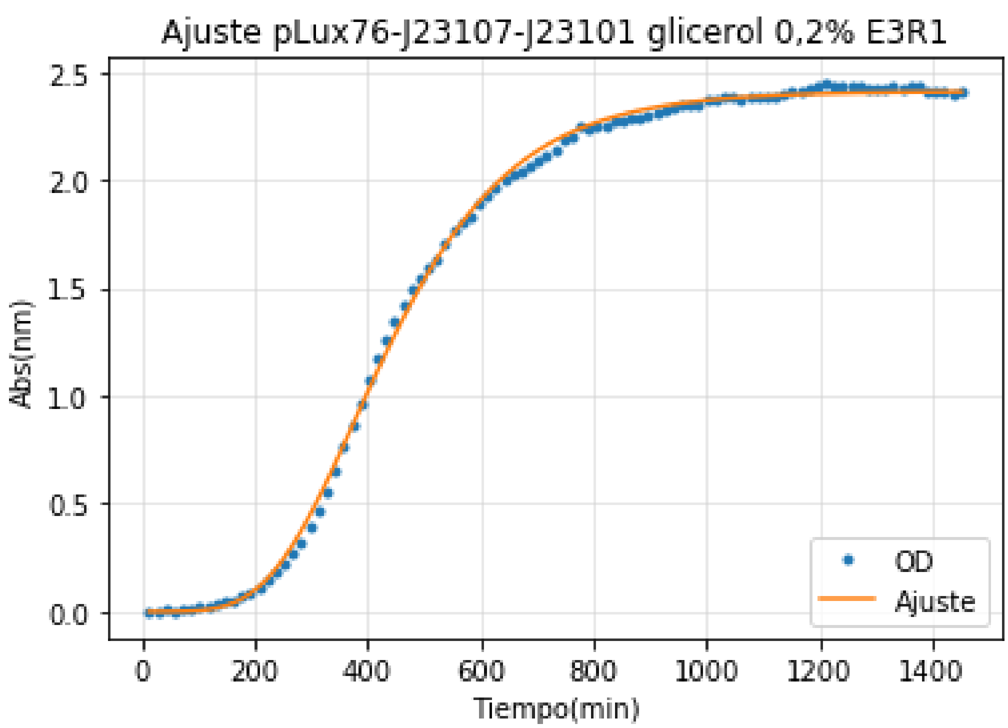
```
Out[26]: <matplotlib.legend.Legend at 0x1d71cb01ba8>
```



In [27]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-107-std glicerol rep 1
y16= np.log(od1823g1)-np.log(np.min(od1823g1))
print('Min OD = %e'%((np.min(od1823g1))))
evaly, params=Function_fit(tt,y16,0,-1,title = 'Ajuste pLux76-J23107-J23101 glicerol 0,
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[25]
```

```python
y2=tt[46]
plt.figure()
plt.title('pLux76-J23107-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1823g1,label='OD pLux76-J23107-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:47],od1823g1[25:47],label='OD pLux76-J23107-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

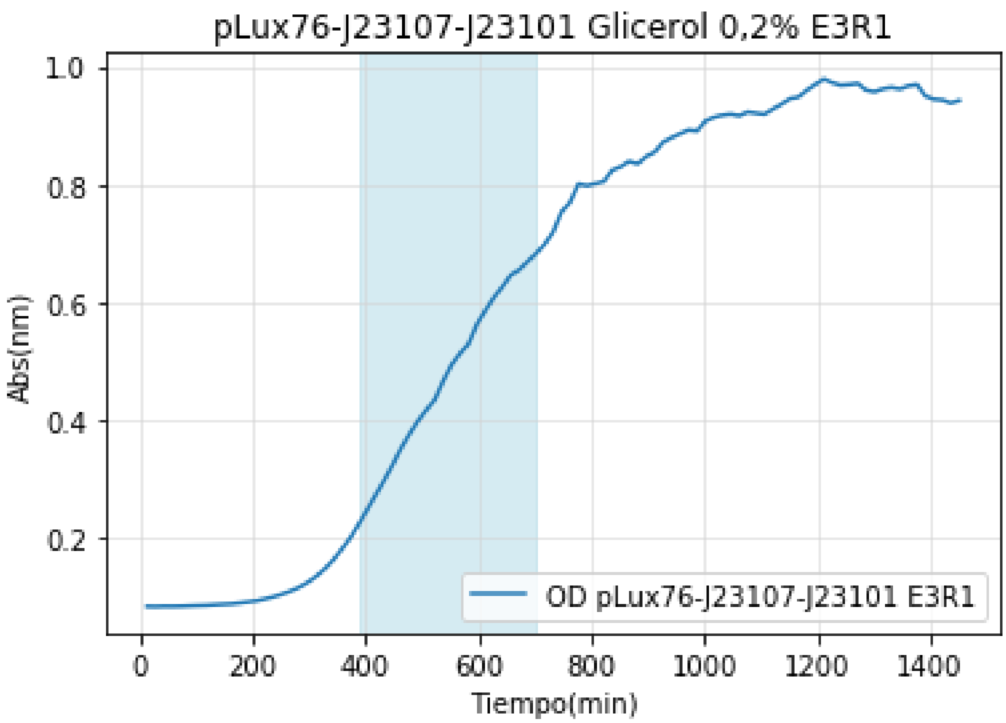
```
Min OD = 8.500000e-02
[  2.40986015e+00   5.77973374e-03   2.23325401e+02]
```

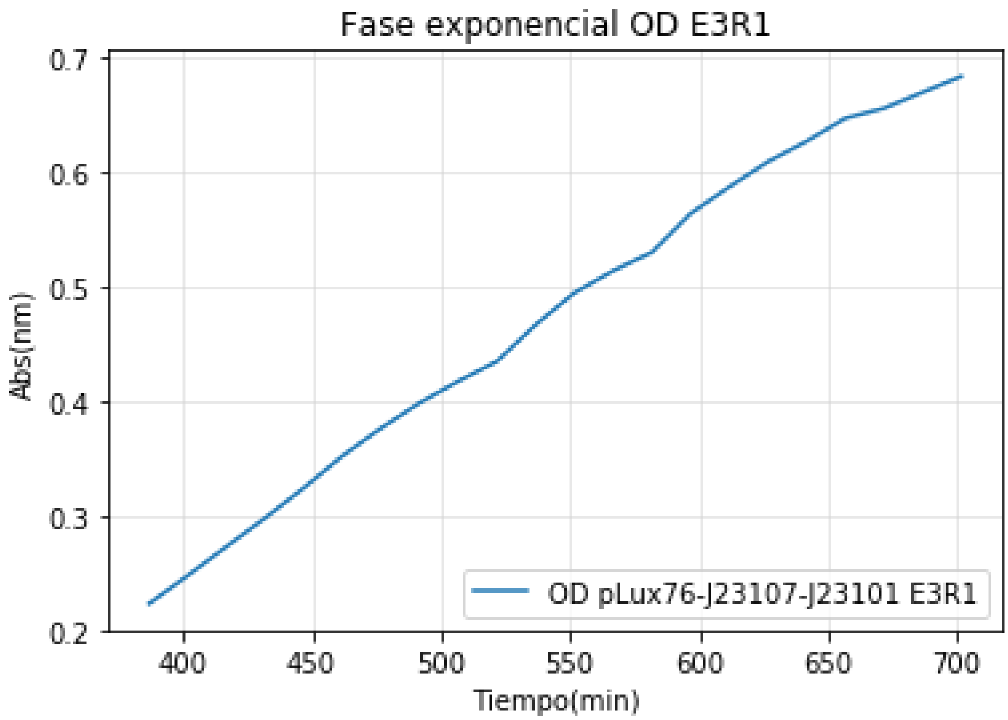



```
A=2.409860e+00
um=5.779734e-03
```

```
l=2.233254e+02
Tm=3.767127e+02
doubpe=1.199272e+02
ext=2.998180e+02
Tfinal=6.765307e+02
```

```
Out[27]: <matplotlib.legend.Legend at 0x1d71cca29b0>
```

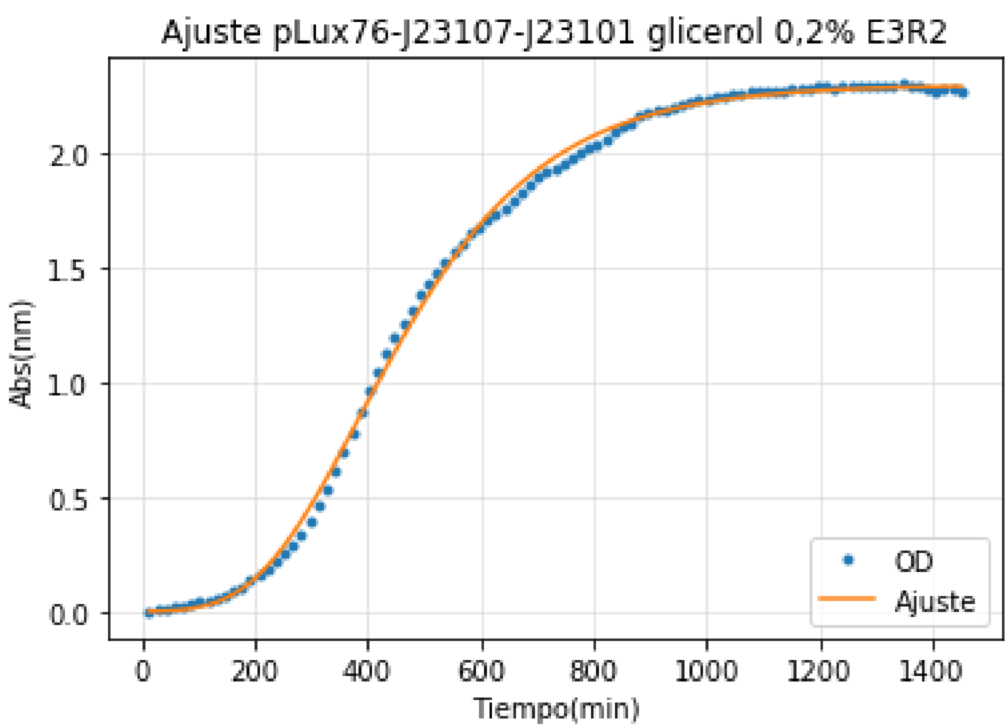




In [28]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-107-std glicerol rep 2
y17= np.log(od1823g2)-np.log(np.min(od1823g2))
print('Min OD = %e'%((np.min(od1823g2))))
evaly, params=Function_fit(tt,y17,0,-1,title = 'Ajuste pLux76-J23107-J23101 glicerol 0,
A17= params[0]
um17=params[1]
l17=params[2]
print('A=%e'%(A17))
print('um=%e'%(um17))
print('l=%e'%(l17))

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

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

```
y2=tt[51]
plt.figure()
plt.title('pLux76-J23107-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1823g2,label='OD pLux76-J23107-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:52],od1823g2[25:52],label='OD pLux76-J23107-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

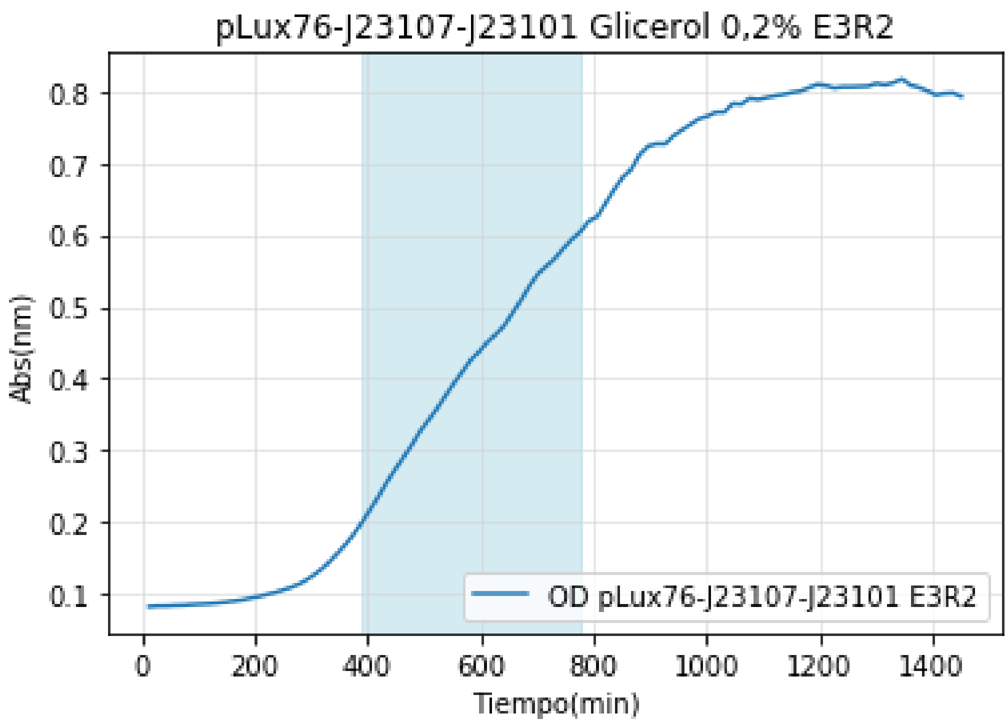
```
Min OD = 8.230000e-02
[  2.29194044e+00   4.63599113e-03   2.02332658e+02]
```

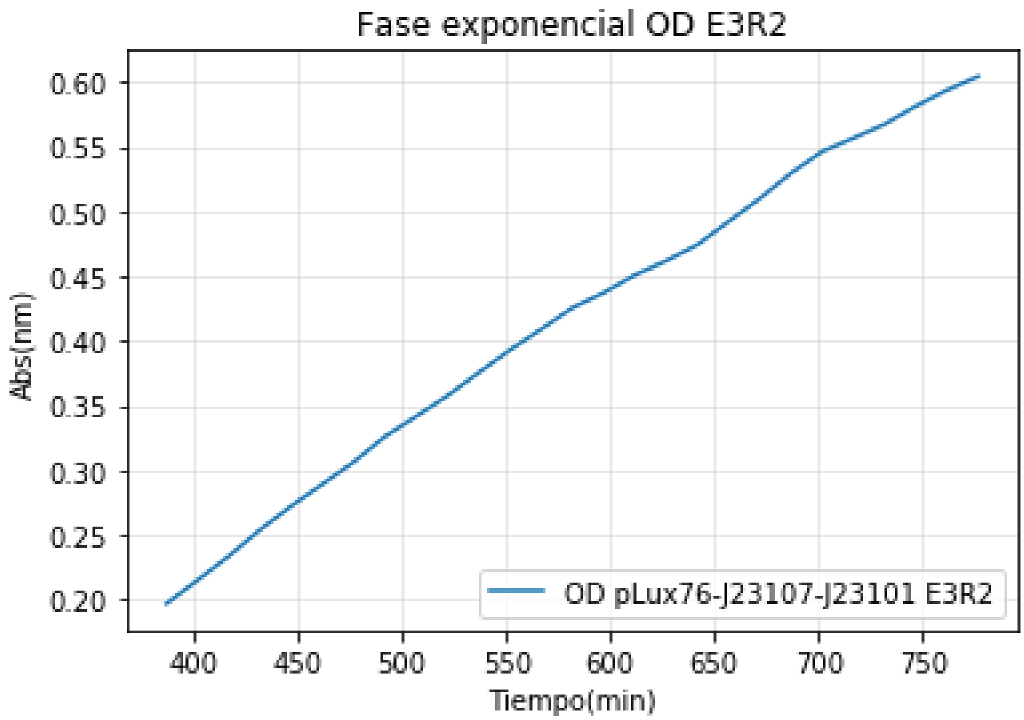



```
A=2.291940e+00
um=4.635991e-03
```

```
l=2.023327e+02
Tm=3.842048e+02
doubpe=1.495143e+02
ext=3.737859e+02
Tfinal=7.579907e+02
```

```
Out[28]: <matplotlib.legend.Legend at 0x1d71b362438>
```

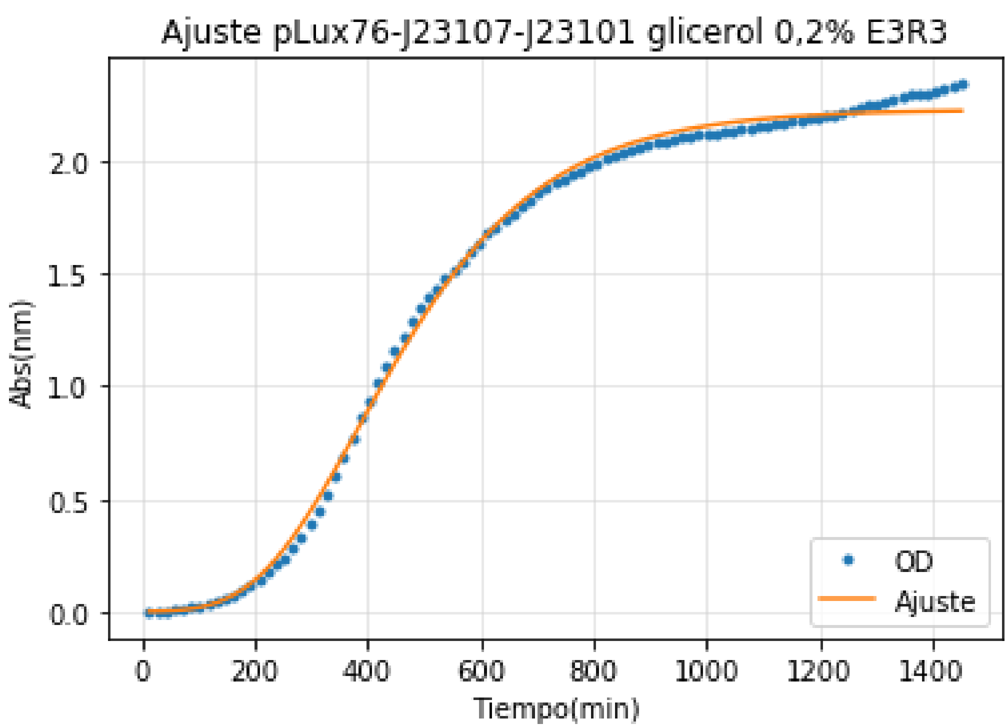




In [29]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-107-std glicerol rep 3
y18= np.log(od1823g3)-np.log(np.min(od1823g3))
print('Min OD = %e'%((np.min(od1823g3))))
evaly, params=Function_fit(tt,y18,0,-1,title = 'Ajuste pLux76-J23107-J23101 glicerol 0,
A18= params[0]
um18=params[1]
l18=params[2]
print('A=%e'%(A18))
print('um=%e'%(um18))
print('l=%e'%(l18))

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

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

```
y2=tt[52]
plt.figure()
plt.title('pLux76-J23107-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1823g3,label='OD pLux76-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 E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],od1823g3[25:53],label='OD pLux76-J23107-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

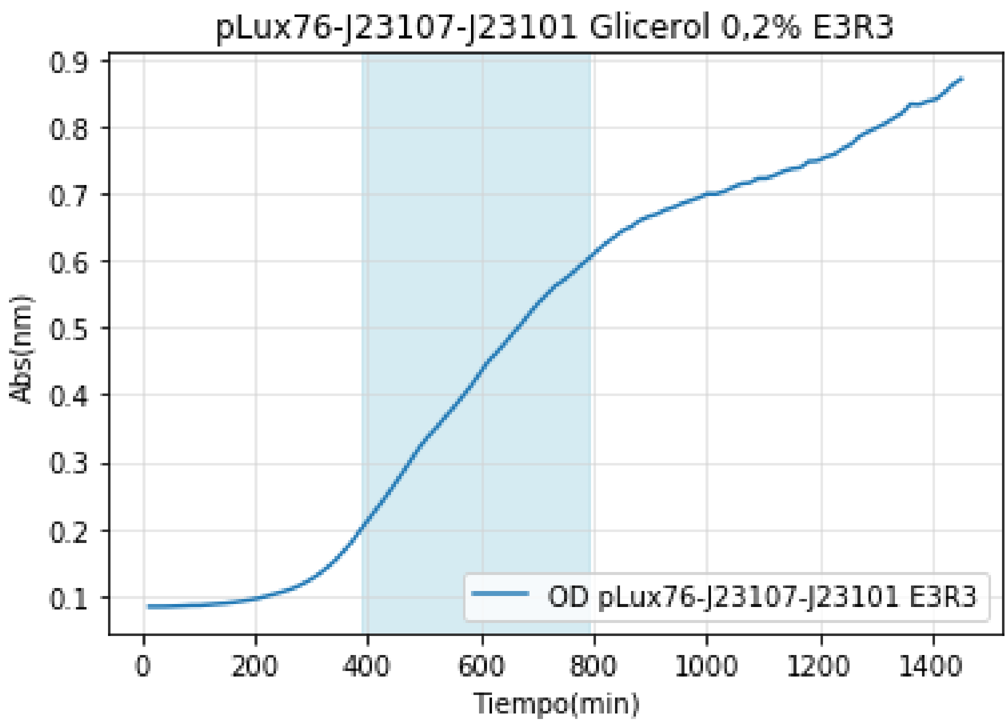
```
Min OD = 8.390000e-02
[  2.22773053e+00   4.51419548e-03   2.02345221e+02]
```

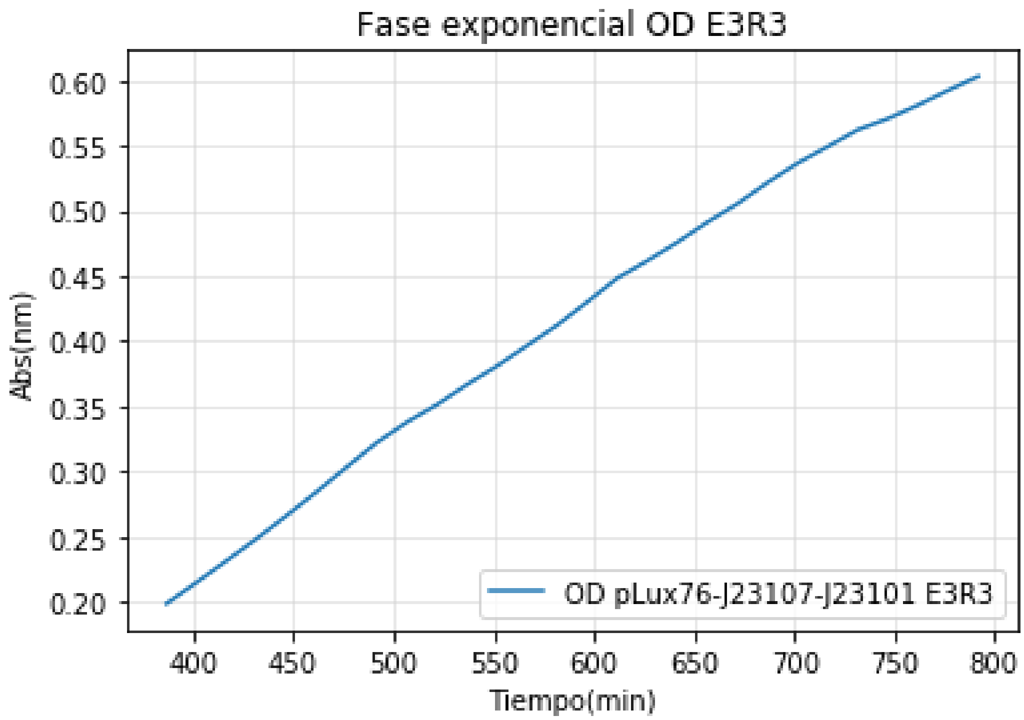



```
A=2.227731e+00
um=4.514195e-03
```

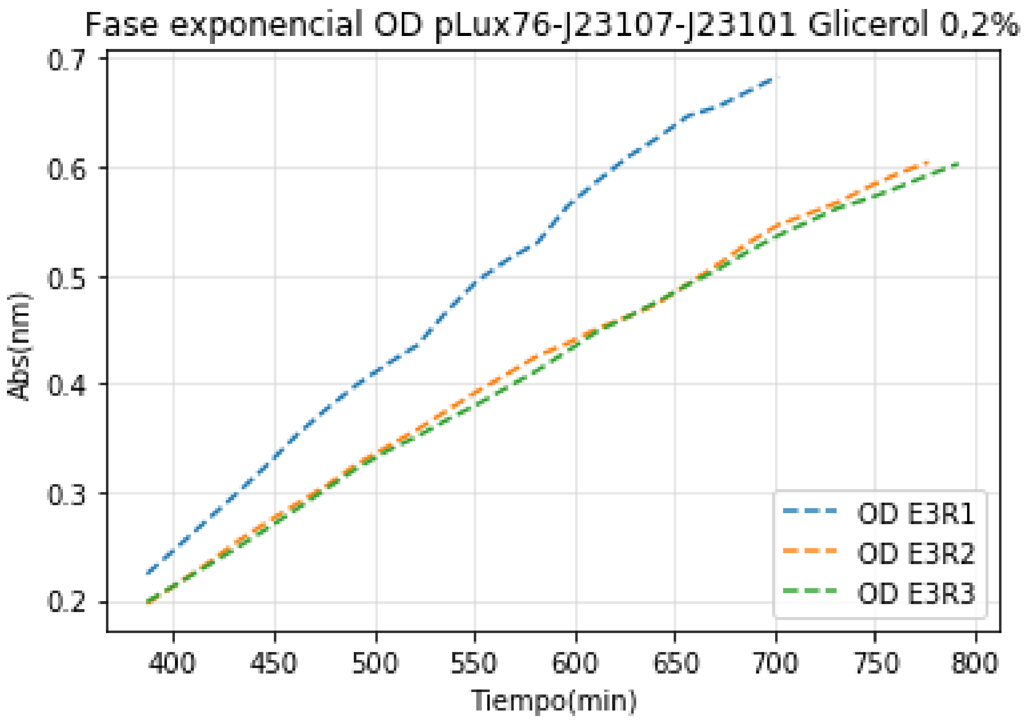
```
l=2.023452e+02
Tm=3.838917e+02
doubpe=1.535483e+02
ext=3.838708e+02
Tfinal=7.677625e+02
```

Out[29]: <matplotlib.legend.Legend at 0x1d71b2f1c88>

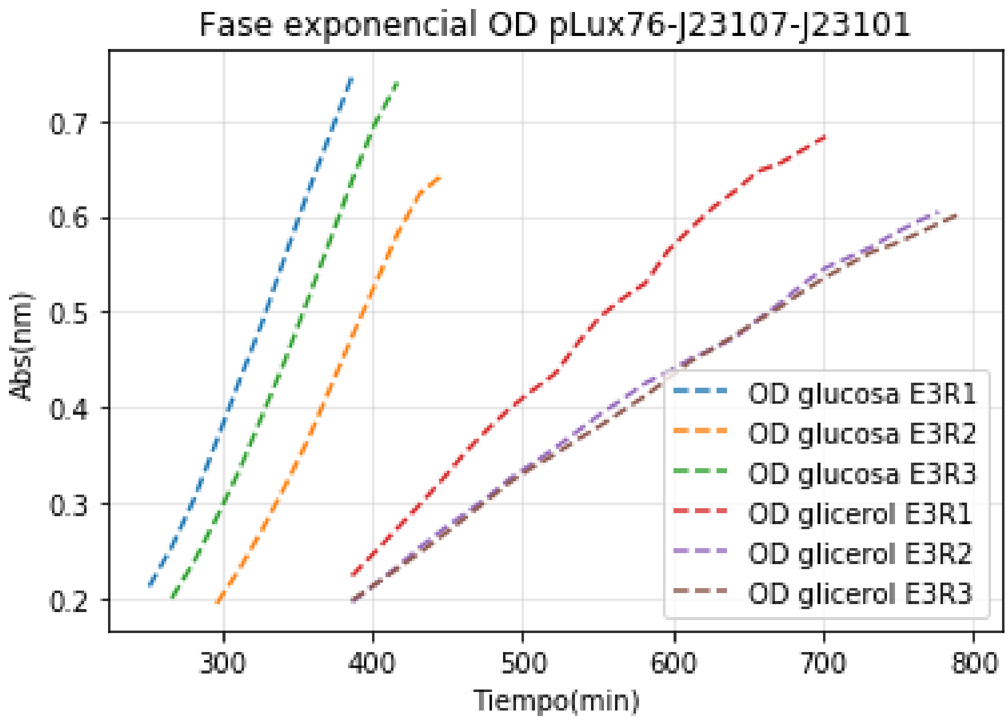





```
In [30]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23107-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[25:47],od1823g1[25:47],'--',label='OD E3R1')
         plt.plot(tt[25:52],od1823g2[25:52],'--',label='OD E3R2')
         plt.plot(tt[25:53],od1823g3[25:53],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[30]: <matplotlib.legend.Legend at 0x1d71cdd0a58>
```



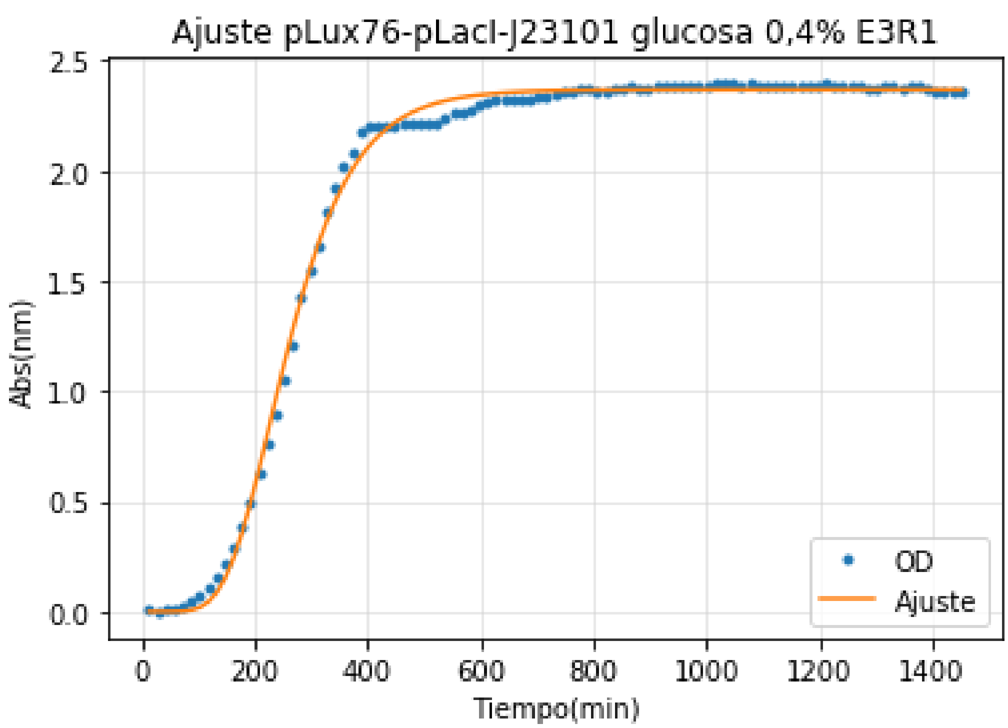

```
In [31]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23107-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],od18231[16:26],'--',label='OD glucosa E3R1')
         plt.plot(tt[19:30],od18232[19:30],'--',label='OD glucosa E3R2')
         plt.plot(tt[17:28],od18233[17:28],'--',label='OD glucosa E3R3')
         plt.plot(tt[25:47],od1823g1[25:47],'--',label='OD glicerol E3R1')
         plt.plot(tt[25:52],od1823g2[25:52],'--',label='OD glicerol E3R2')
         plt.plot(tt[25:53],od1823g3[25:53],'--',label='OD glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[31]: <matplotlib.legend.Legend at 0x1d71cea4710>
```



In [32]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-plac-std glucosa rep 1
y19= np.log(od18261)-np.log(np.min(od18261))
print('Min OD = %e'%((np.min(od18261))))
evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glucosa 0,4%
A19= params[0]
um19=params[1]
l19=params[2]
print('A=%e'%(A19))
print('um=%e'%(um19))
print('l=%e'%(l19))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[15]
```

```python
y2=tt[24]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18261,label='OD pLux76-pLacI-J23101 E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[15:25],od18261[15:25],label='OD pLux76-pLacI-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

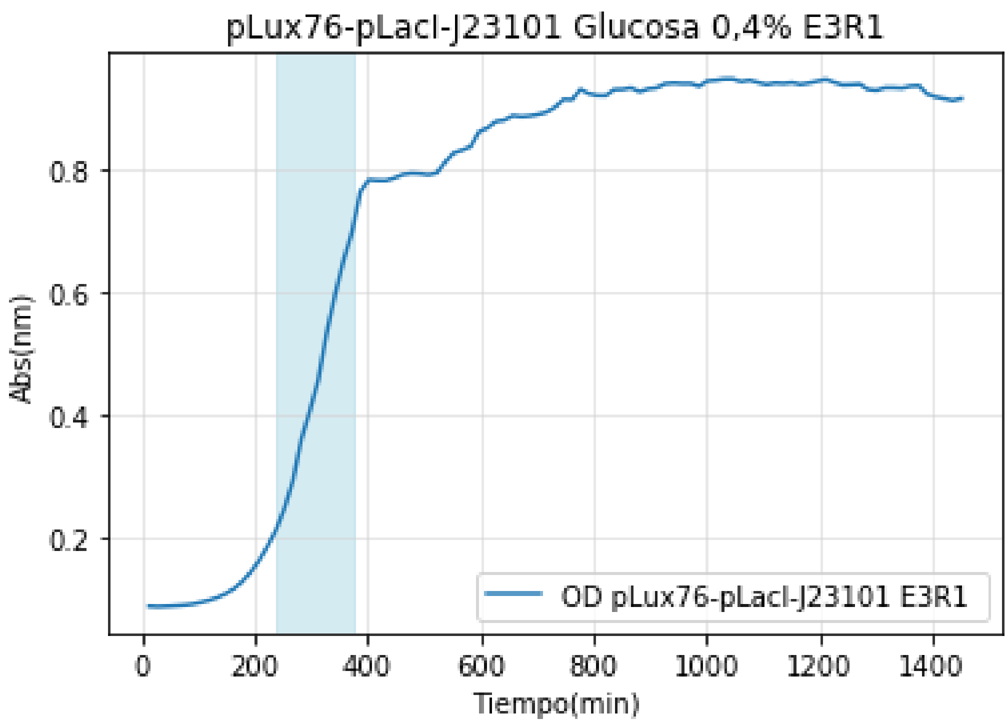
```
Min OD = 8.640000e-02
[  2.36831728e+00   1.08607178e-02   1.47377017e+02]
```

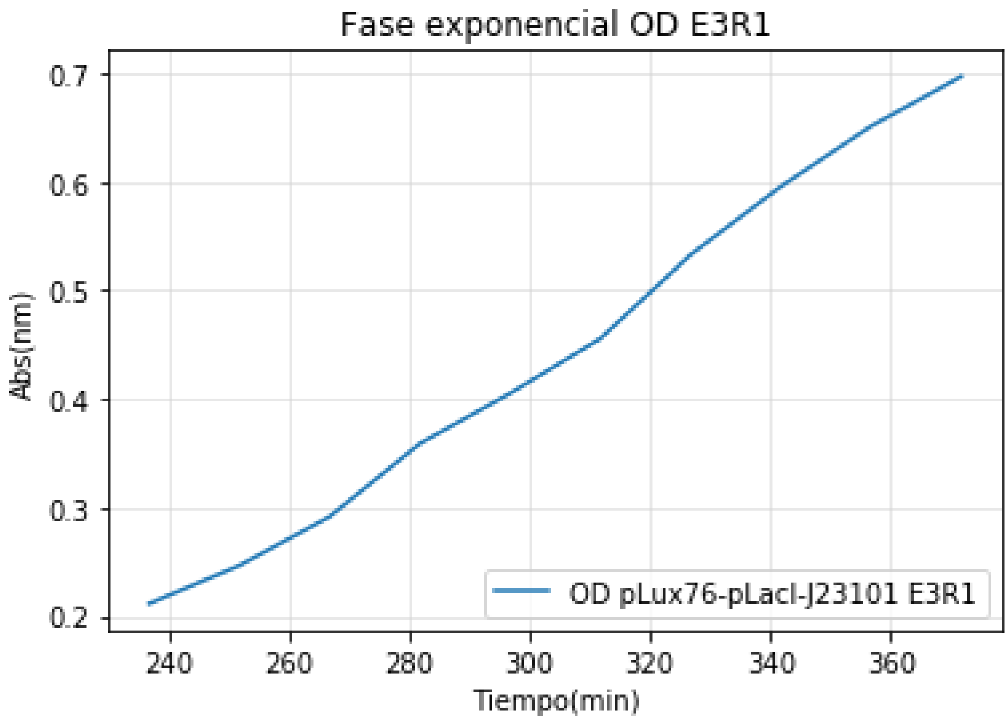



```
A=2.368317e+00
um=1.086072e-02
```

```
l=1.473770e+02
Tm=2.275978e+02
doubpe=6.382149e+01
ext=1.276430e+02
Tfinal=3.552408e+02
```

Out[32]: <matplotlib.legend.Legend at 0x1d71ccf1e10>

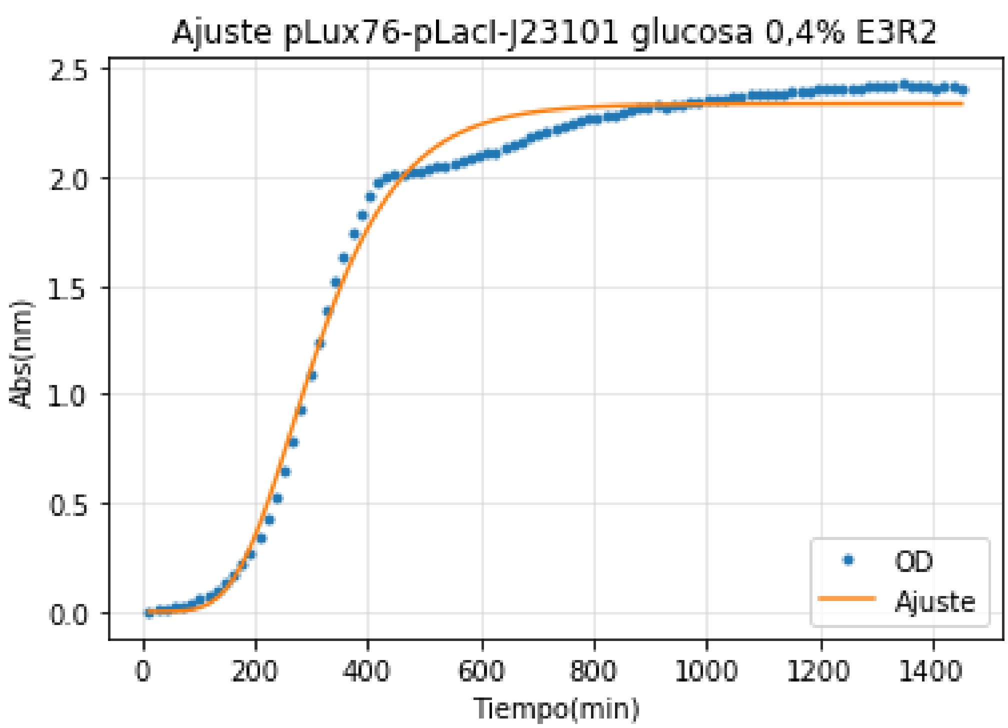




In [33]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-placI-std glucosa rep 2
y20= np.log(od18262)-np.log(np.min(od18262))
print('Min OD = %e'%((np.min(od18262))))
evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glucosa 0,4%
A20= params[0]
um20=params[1]
l20=params[2]
print('A=%e'%(A20))
print('um=%e'%(um20))
print('l=%e'%(l20))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[18]
```

```
y2=tt[30]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18262,label='OD pLux76-pLacI-J23101 E3R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:31],od18262[18:31],label='OD pLux76-pLacI-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

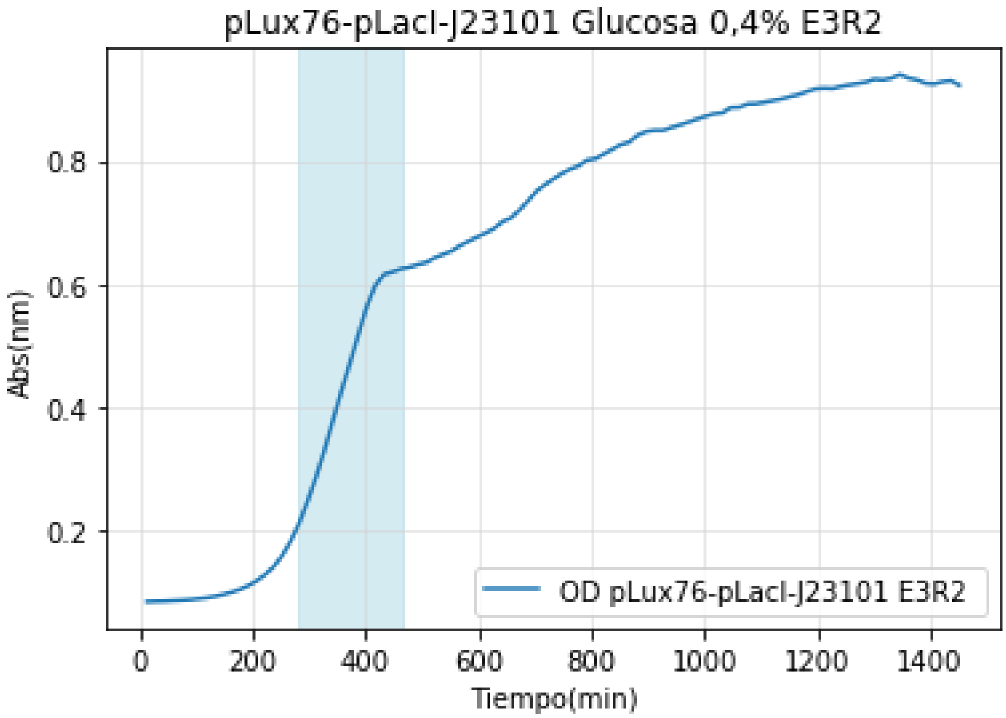
```
Min OD = 8.300000e-02
[  2.33750027e+00   8.21668869e-03   1.62353250e+02]
```

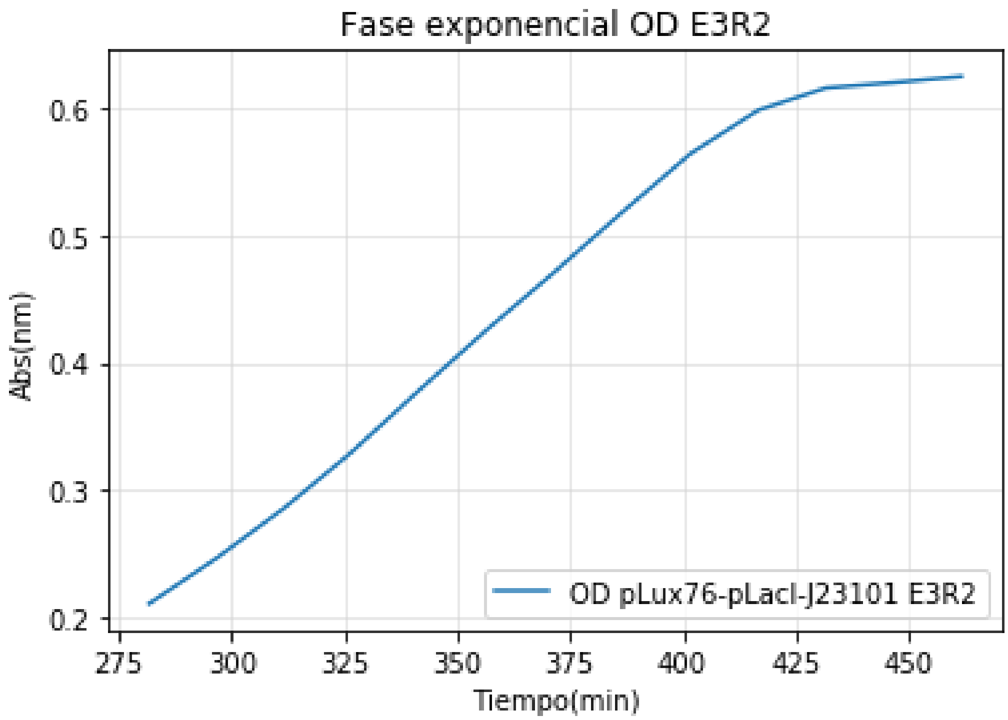



```
A=2.337500e+00
um=8.216689e-03
```

```
l=1.623532e+02
Tm=2.670083e+02
doubpe=8.435846e+01
ext=1.687169e+02
Tfinal=4.357253e+02
```

Out[33]: `<matplotlib.legend.Legend at 0x1d71c982f98>`

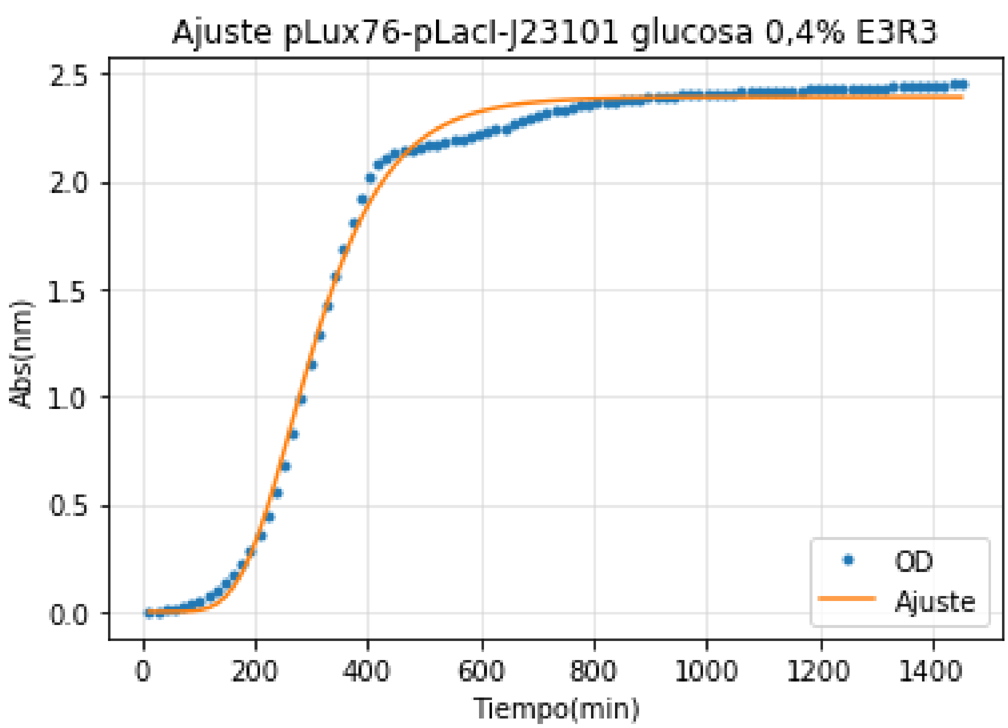




In [34]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-plac-std glucosa rep 3
y21= np.log(od18263)-np.log(np.min(od18263))
print('Min OD = %e'%((np.min(od18263))))
evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glucosa 0,4%
A21= params[0]
um21=params[1]
l21=params[2]
print('A=%e'%(A21))
print('um=%e'%(um21))
print('l=%e'%(l21))

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

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

```python
y2=tt[28]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od18263,label='OD pLux76-pLacI-J23101 E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],od18263[17:29],label='OD pLux76-pLacI-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

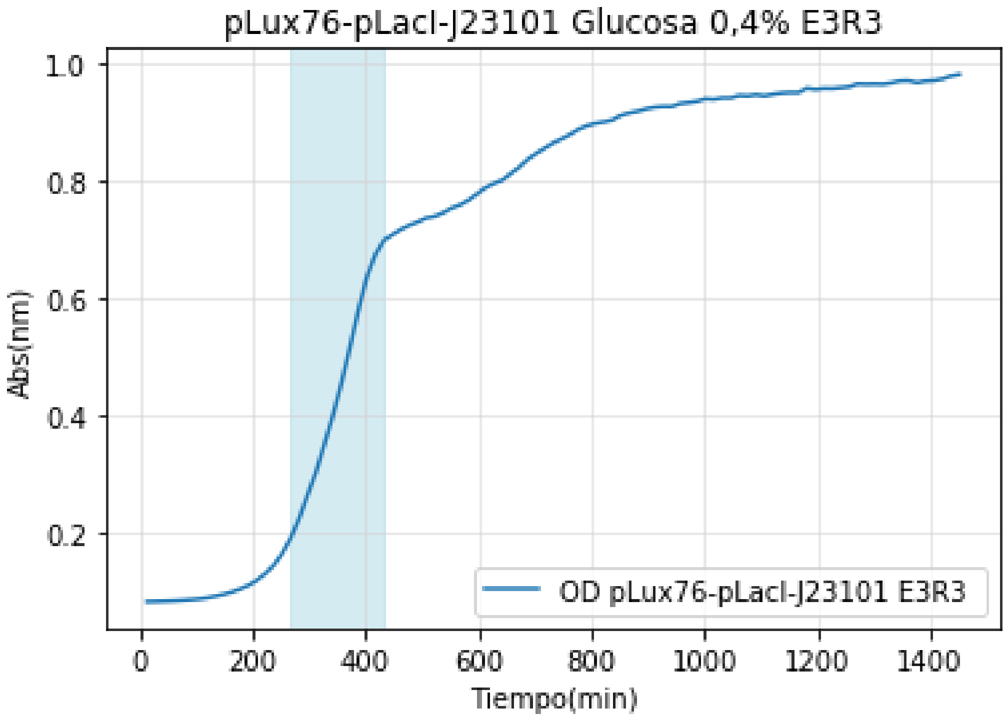
```
Min OD = 8.450000e-02
[  2.39074012e+00   9.41670538e-03   1.71774749e+02]
```

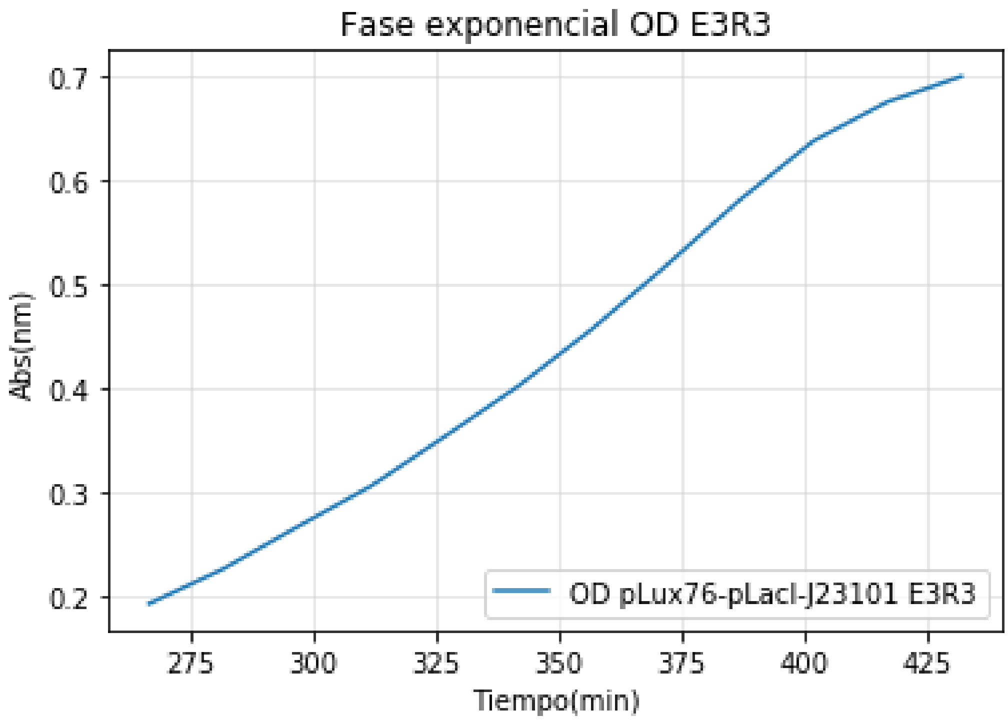



```
A=2.390740e+00
um=9.416705e-03
```

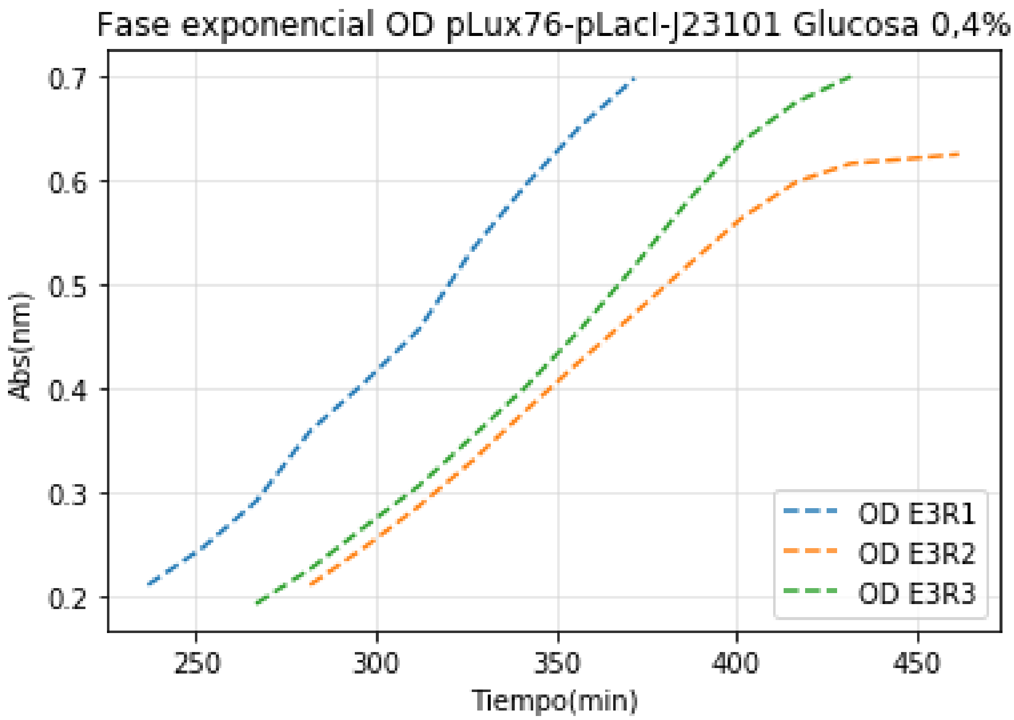
```
l=1.717747e+02
Tm=2.651730e+02
doubpe=7.360825e+01
ext=1.472165e+02
Tfinal=4.123895e+02
```

```
Out[34]: <matplotlib.legend.Legend at 0x1d71d0186a0>
```





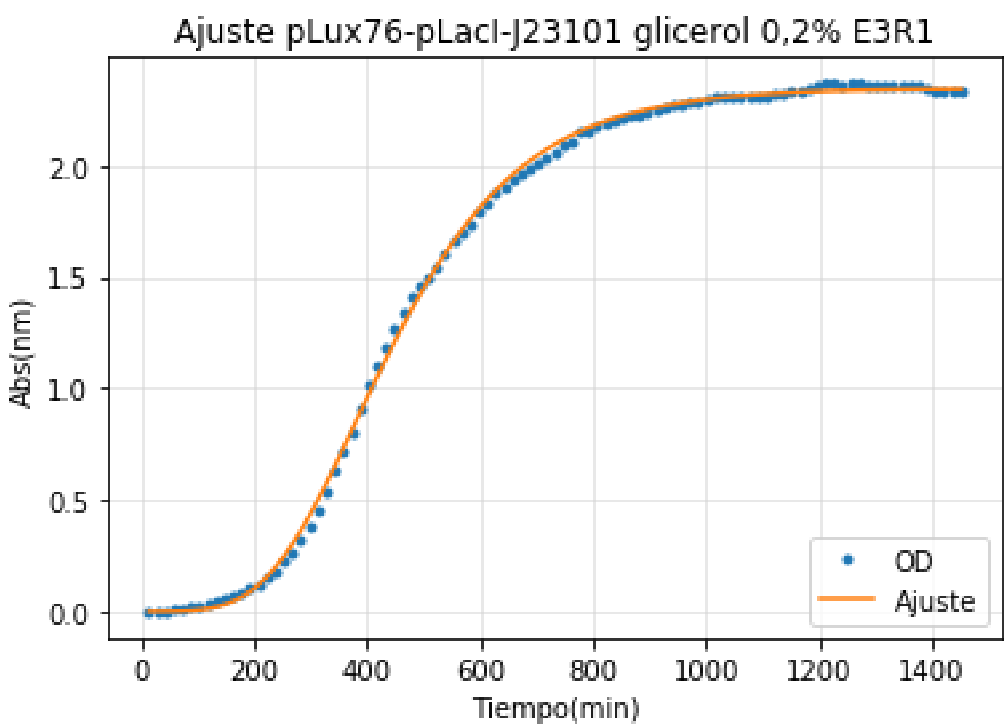

```
In [35]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLacI-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[15:25],od18261[15:25],'--',label='OD E3R1')
         plt.plot(tt[18:31],od18262[18:31],'--',label='OD E3R2')
         plt.plot(tt[17:29],od18263[17:29],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[35]: <matplotlib.legend.Legend at 0x1d71d13a828>
```



In [36]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-plac-std glicerol rep 1
y22= np.log(od1826g1)-np.log(np.min(od1826g1))
print('Min OD = %e'%((np.min(od1826g1))))
evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste pLux76-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[25]
```

```
y2=tt[48]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1826g1,label='OD pLux76-pLacI-J23101 E3R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:49],od1826g1[25:49],label='OD pLux76-pLacI-J23101 E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

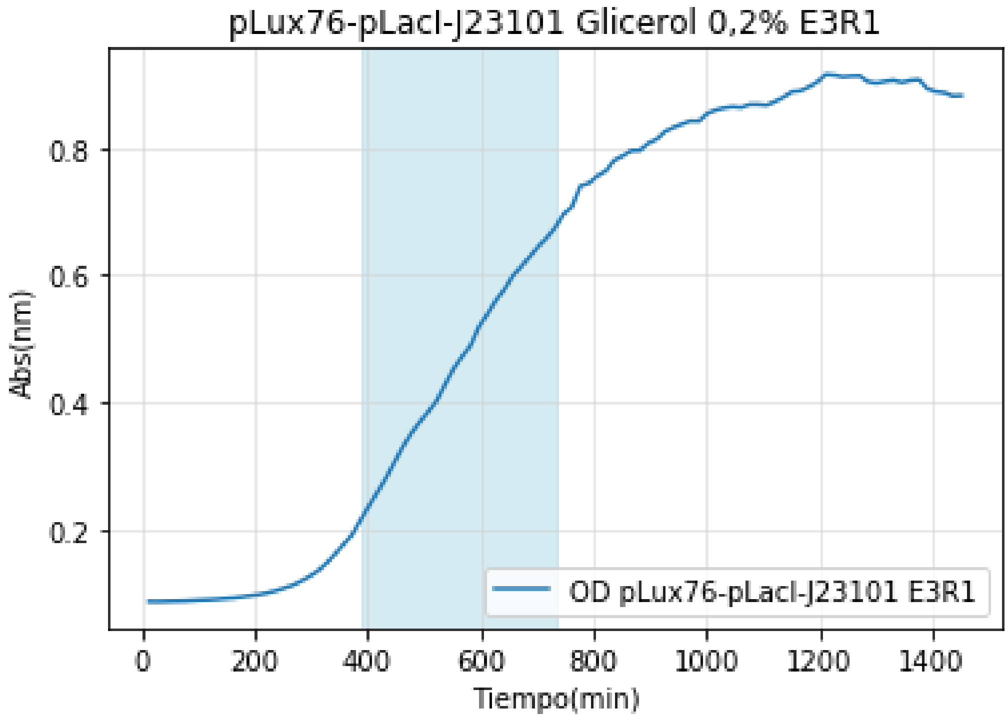
```
Min OD = 8.600000e-02
[ 2.34483347e+00  5.37585239e-03  2.20927356e+02]
```

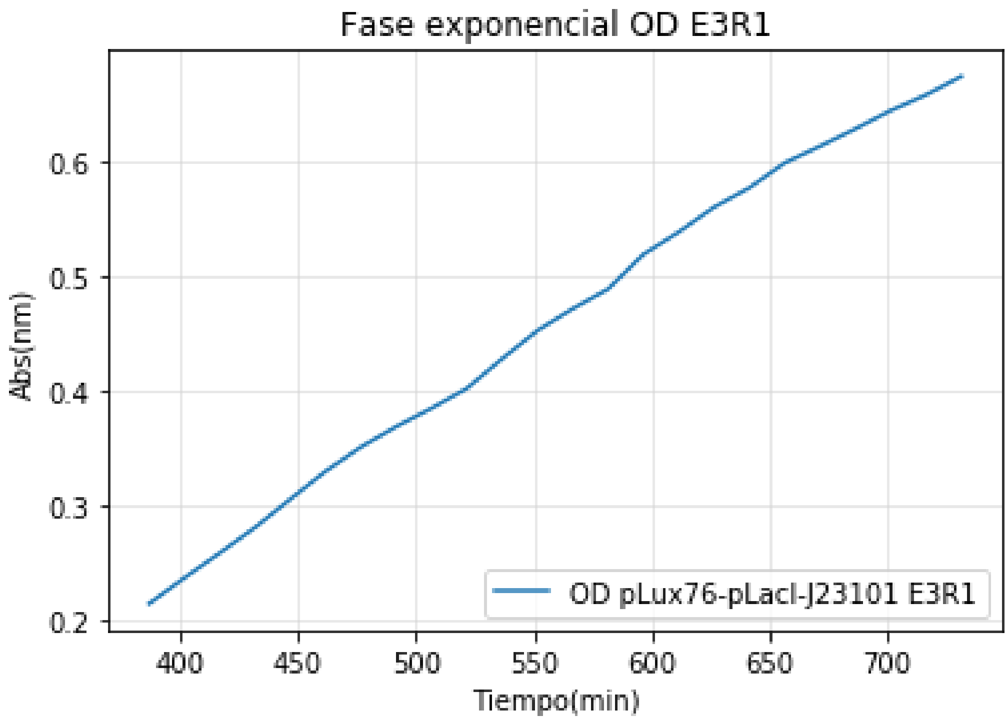



```
A=2.344833e+00
um=5.375852e-03
```

```
l=2.209274e+02
Tm=3.813886e+02
doubpe=1.289372e+02
ext=3.223429e+02
Tfinal=7.037315e+02
```

Out[36]: `<matplotlib.legend.Legend at 0x1d71d2c3128>`

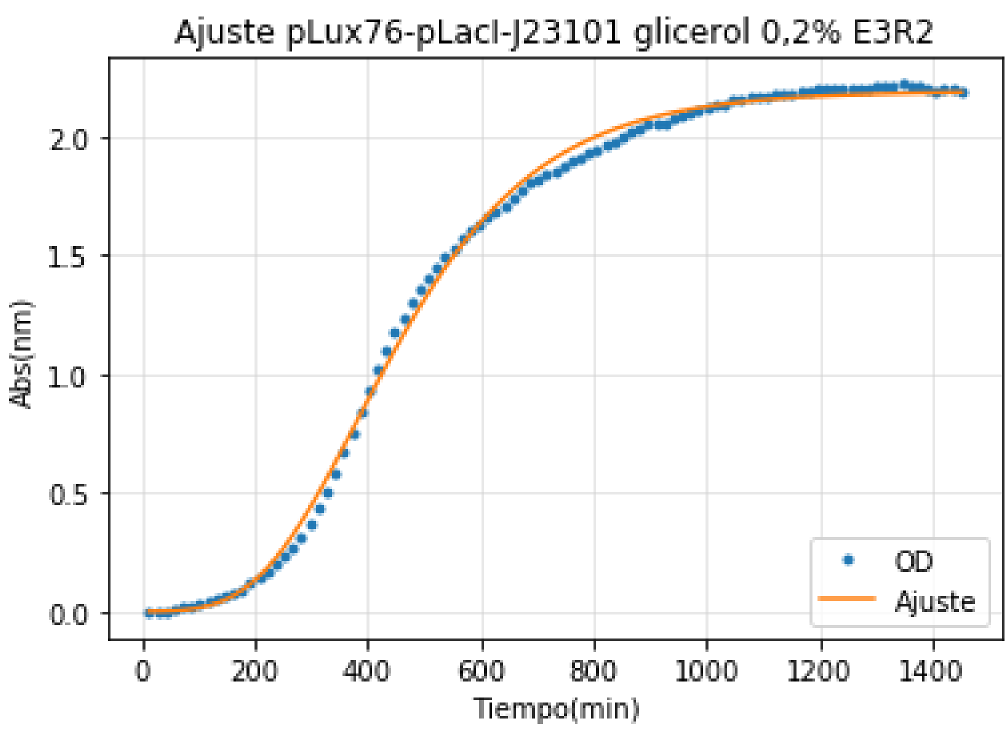




In [37]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-plac-std glicerol rep 2
y23= np.log(od1826g2)-np.log(np.min(od1826g2))
print('Min OD = %e'%((np.min(od1826g2))))
evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glicerol 0,2
A23= 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[25]
```

```
y2=tt[52]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1826g2,label='OD pLux76-pLacI-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],od1826g2[25:53],label='OD pLux76-pLacI-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

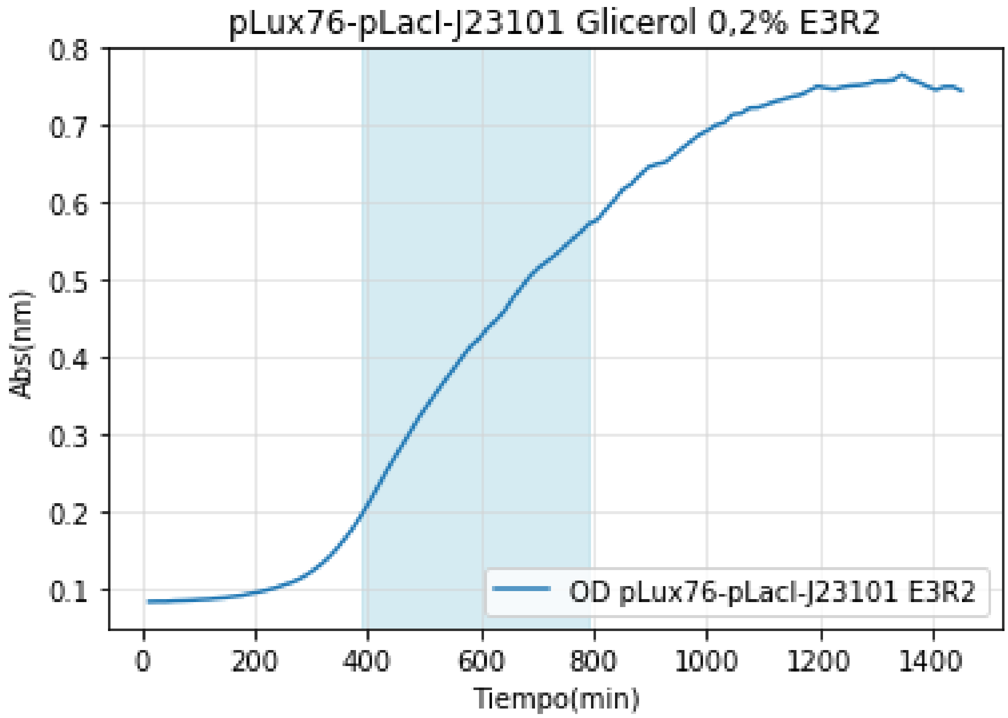
```
Min OD = 8.320000e-02
[  2.18863930e+00   4.56019770e-03   2.04683236e+02]
```

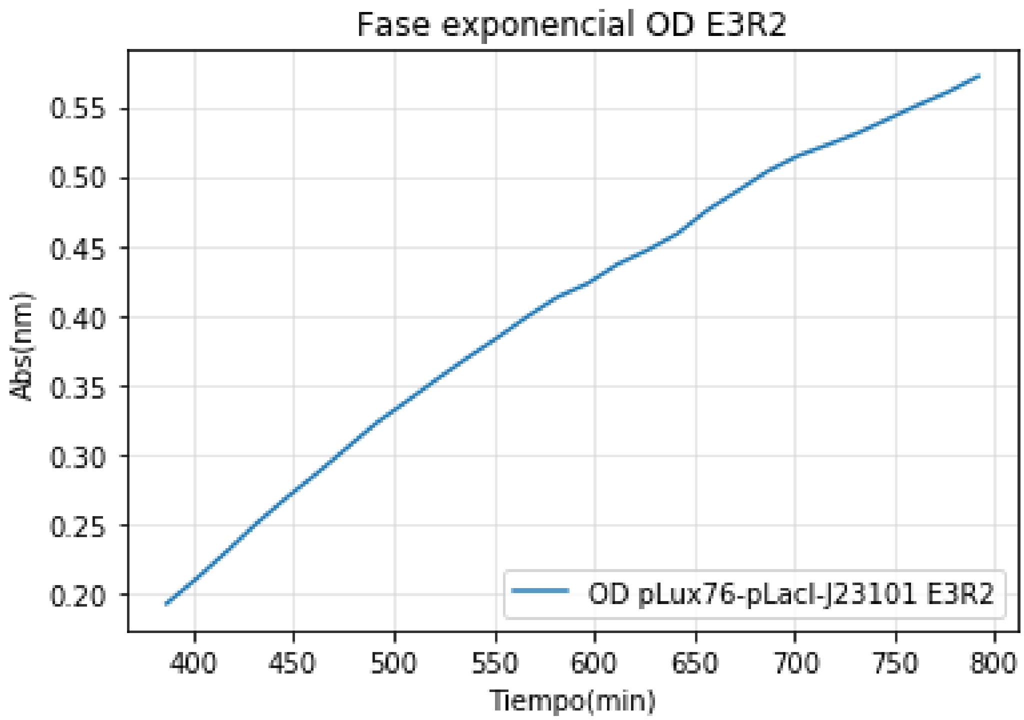



```
A=2.188639e+00
um=4.560198e-03
```

```
l=2.046832e+02
Tm=3.812447e+02
doubpe=1.519994e+02
ext=3.799984e+02
Tfinal=7.612432e+02
```

Out[37]: <matplotlib.legend.Legend at 0x1d71b5f4518>

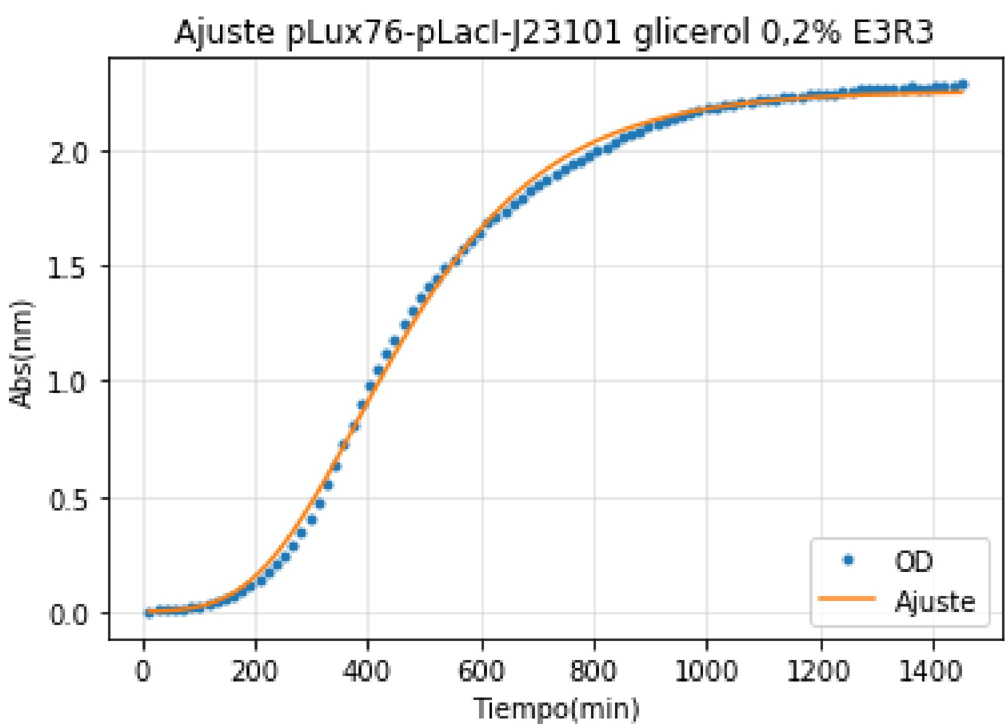




In [38]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-plac-std glicerol rep 3
y24= np.log(od1826g3)-np.log(np.min(od1826g3))
print('Min OD = %e'%((np.min(od1826g3))))
evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glicerol 0,2
A24= 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[25]
```

```
y2=tt[52]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1826g3,label='OD pLux76-pLacI-J23101 E3R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],od1826g3[25:53],label='OD pLux76-pLacI-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

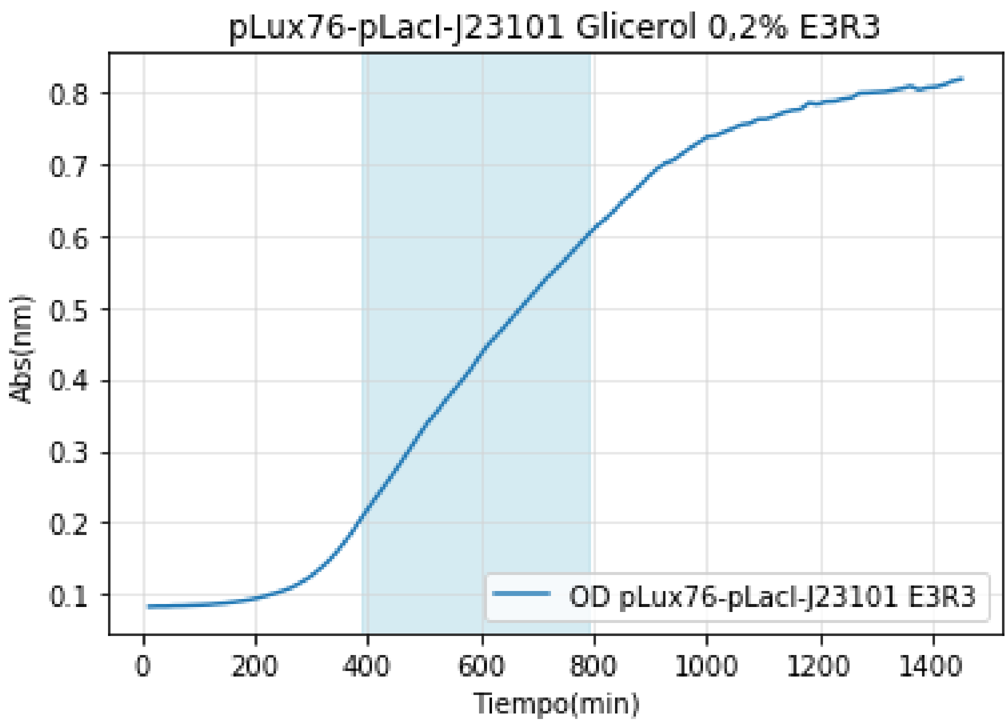
```
Min OD = 8.310000e-02
[  2.25715444e+00   4.50329664e-03   1.97734078e+02]
```

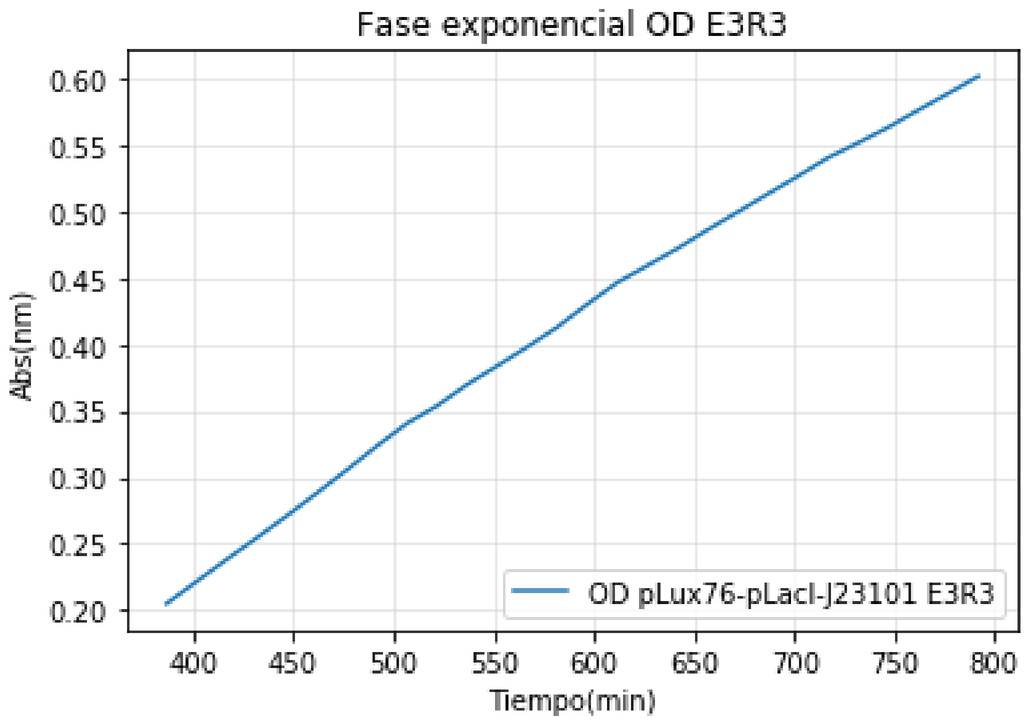



```
A=2.257154e+00
um=4.503297e-03
```

```
l=1.977341e+02
Tm=3.821236e+02
doubpe=1.539199e+02
ext=3.847999e+02
Tfinal=7.669235e+02
```

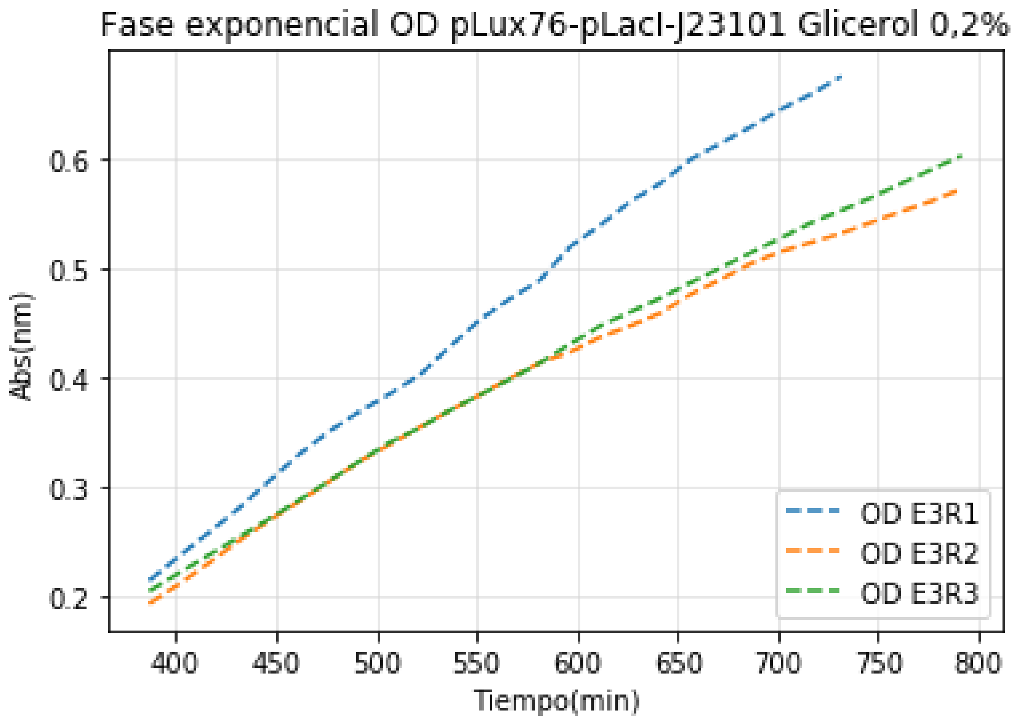
```
Out[38]: <matplotlib.legend.Legend at 0x1d71d1d51d0>
```



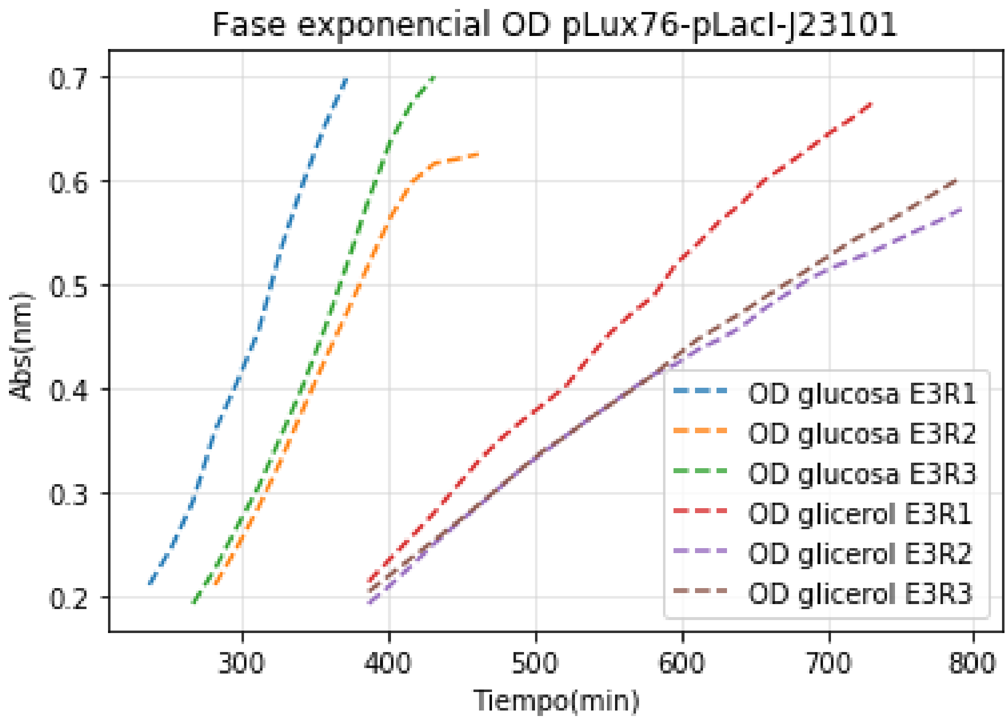


In [39]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLacI-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:49],od1826g1[25:49],'--',label='OD E3R1')
plt.plot(tt[25:53],od1826g2[25:53],'--',label='OD E3R2')
plt.plot(tt[25:53],od1826g3[25:53],'--',label='OD E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[39]:
```
<matplotlib.legend.Legend at 0x1d71d3fe978>
```



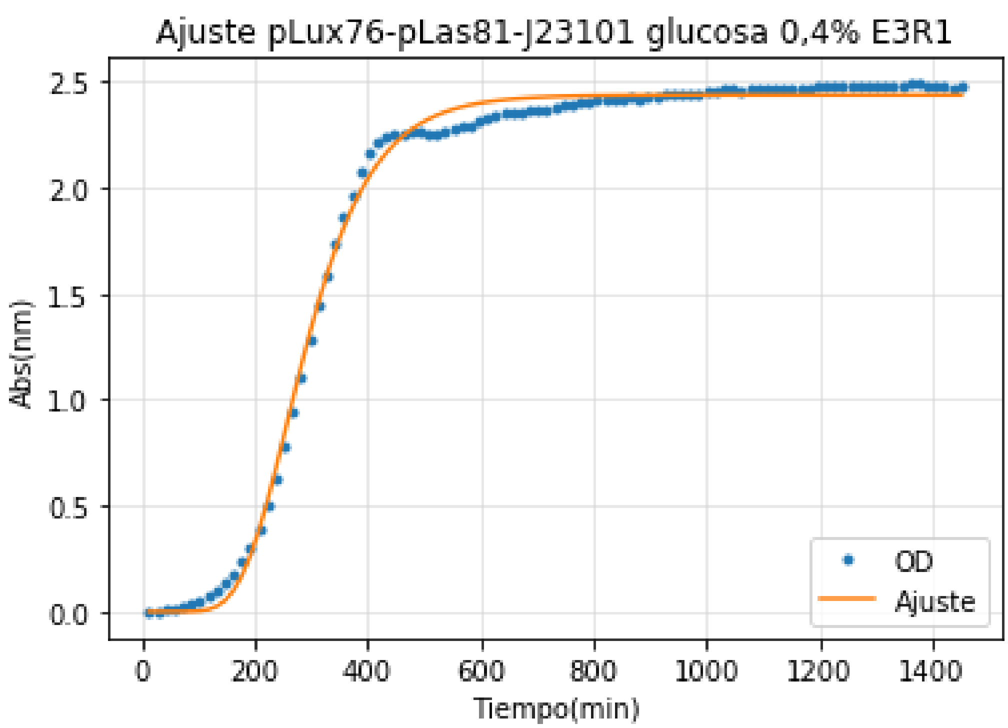

```
In [40]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLacI-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[15:25],od18261[15:25],'--',label='OD glucosa E3R1')
         plt.plot(tt[18:31],od18262[18:31],'--',label='OD glucosa E3R2')
         plt.plot(tt[17:29],od18263[17:29],'--',label='OD glucosa E3R3')
         plt.plot(tt[25:49],od1826g1[25:49],'--',label='OD glicerol E3R1')
         plt.plot(tt[25:53],od1826g2[25:53],'--',label='OD glicerol E3R2')
         plt.plot(tt[25:53],od1826g3[25:53],'--',label='OD glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[40]: <matplotlib.legend.Legend at 0x1d71d4cbd30>
```



In [41]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-plas-std glucosa rep 1
y25= np.log(od18271)-np.log(np.min(od18271))
print('Min OD = %e'%((np.min(od18271))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,4
A25= params[0]
um25=params[1]
l25=params[2]
print('A=%e'%(A25))
print('um=%e'%(um25))
print('l=%e'%(l25))

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

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

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

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

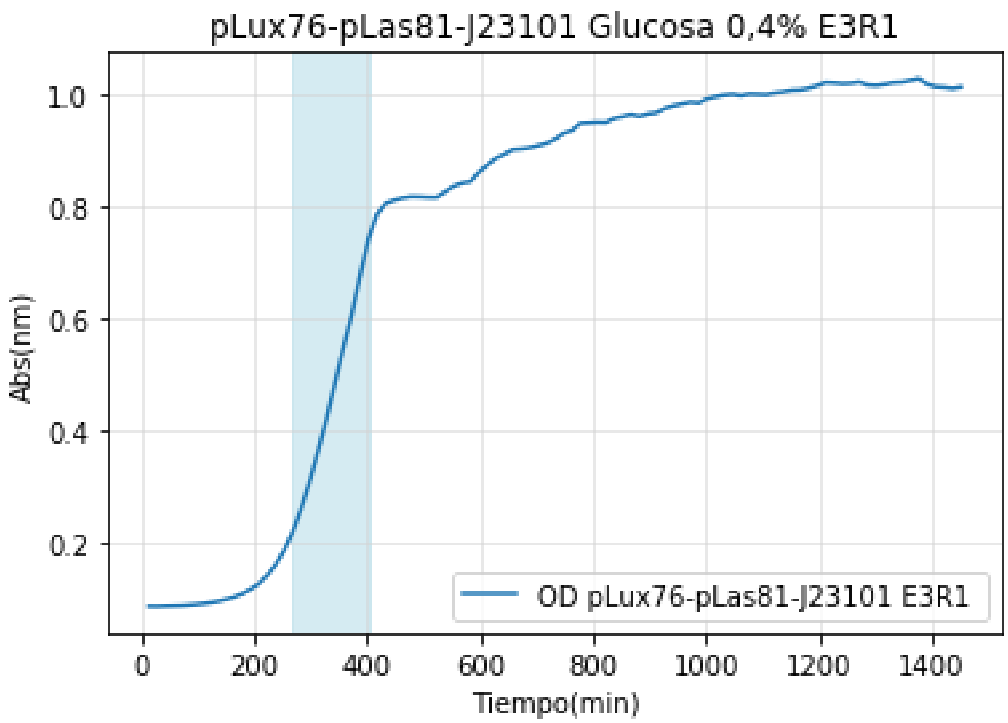
```
Min OD = 8.540000e-02
[  2.43767957e+00   1.09002530e-02   1.74738266e+02]
```

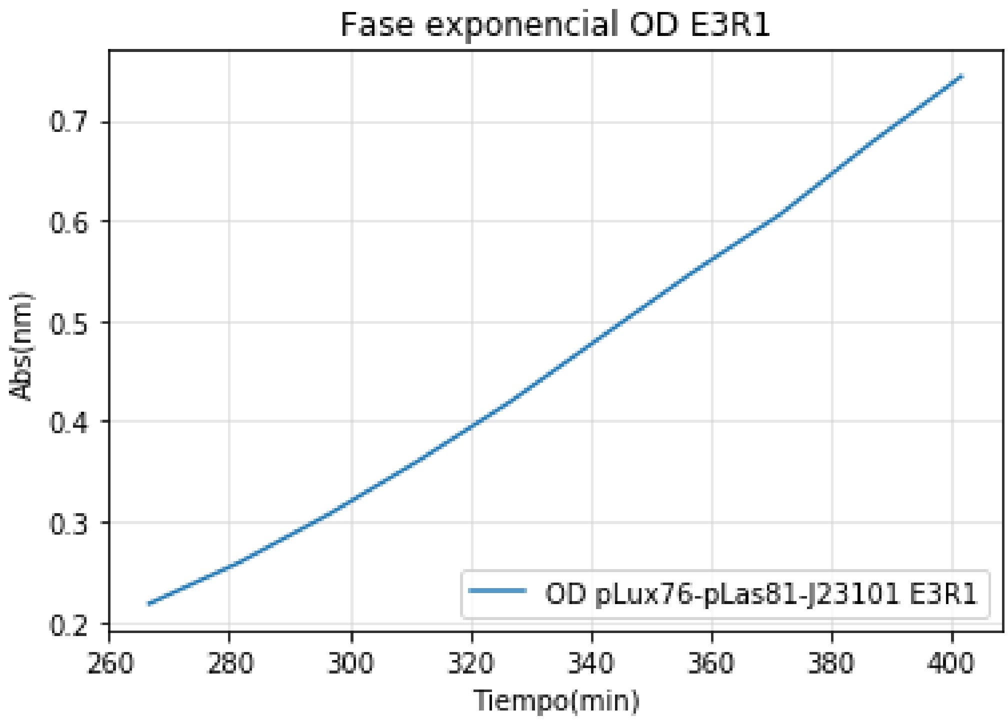



```
A=2.437680e+00
um=1.090025e-02
```

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

```
Out[41]: <matplotlib.legend.Legend at 0x1d71d658080>
```

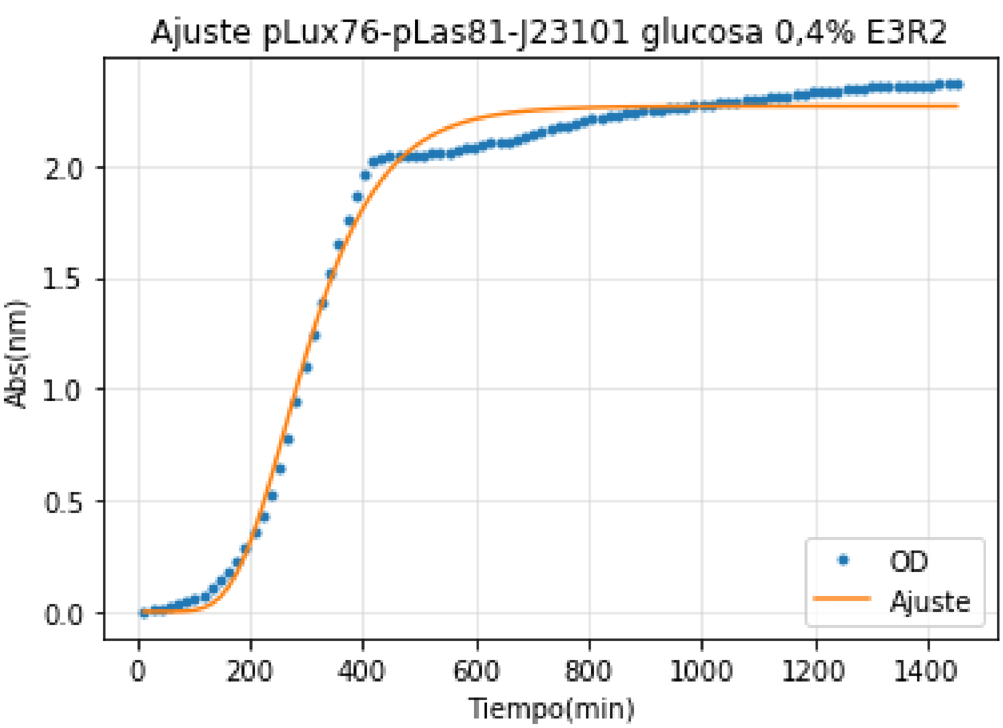




In [42]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-pLas-std glucosa rep 2
y26= np.log(od18272)-np.log(np.min(od18272))
print('Min OD = %e'%((np.min(od18272))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,4
A26= params[0]
um26=params[1]
l26=params[2]
print('A=%e'%(A26))
print('um=%e'%(um26))
print('l=%e'%(l26))

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

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

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

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

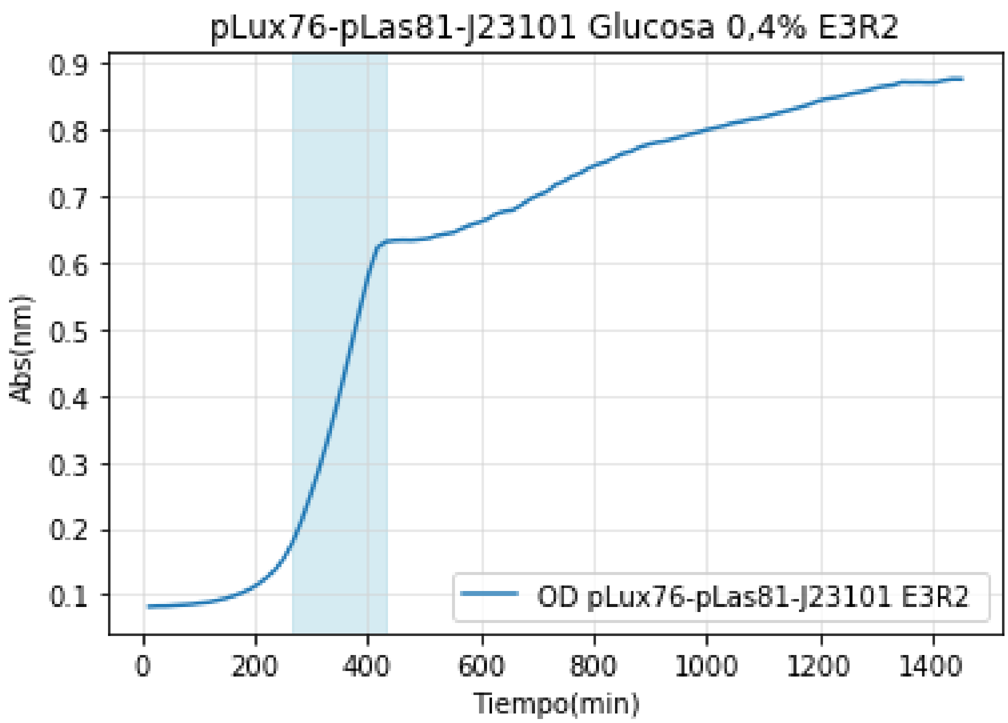
```
Min OD = 8.220000e-02
[  2.26837994e+00   9.06999393e-03   1.71041461e+02]
```

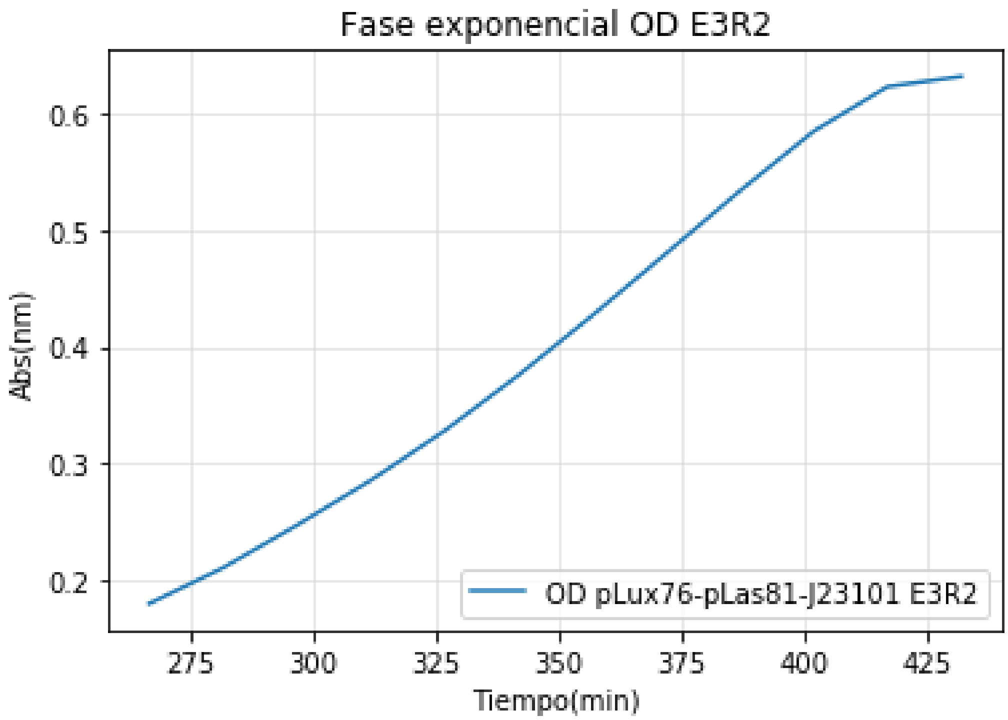



```
A=2.268380e+00
um=9.069994e-03
```

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

```
Out[42]: <matplotlib.legend.Legend at 0x1d71d323400>
```

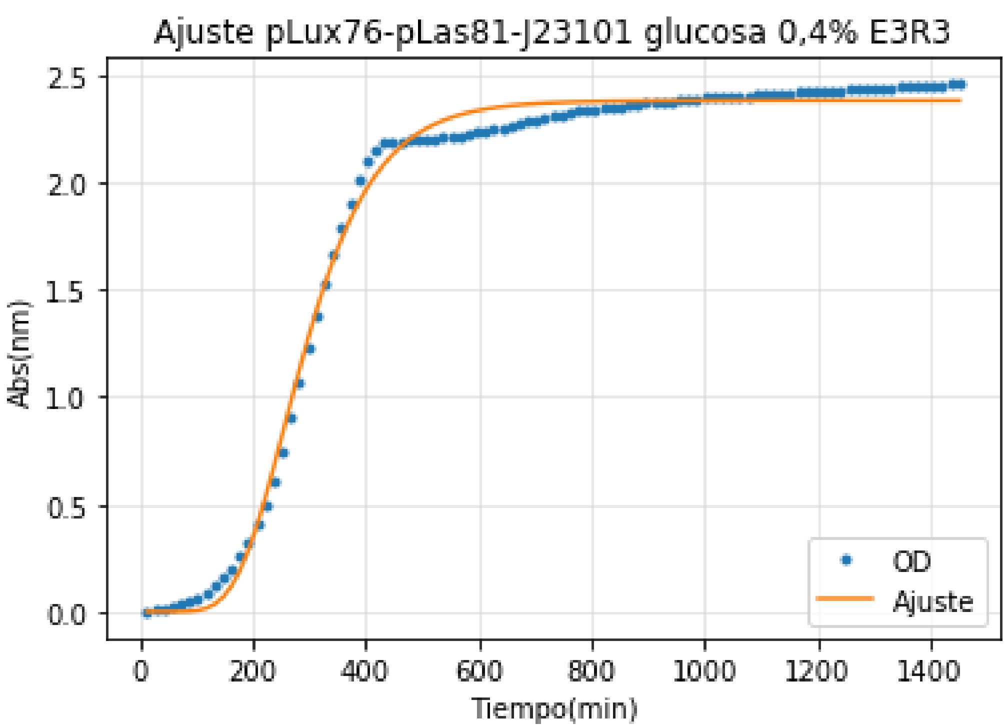





```
In [43]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-pLas-std glucosa rep 3
         y27= np.log(od18273)-np.log(np.min(od18273))
         print('Min OD = %e'%((np.min(od18273))))
         evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glucosa 0,4
         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[17]
```

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

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

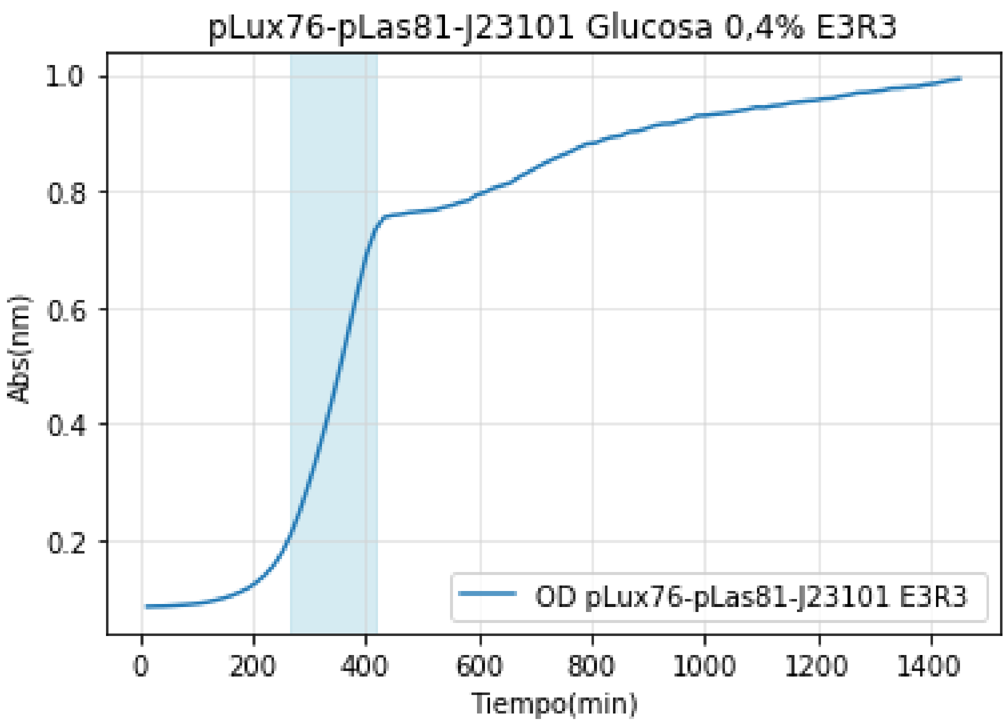
```
Min OD = 8.500000e-02
[  2.38059703e+00   1.00857409e-02   1.69775385e+02]
```

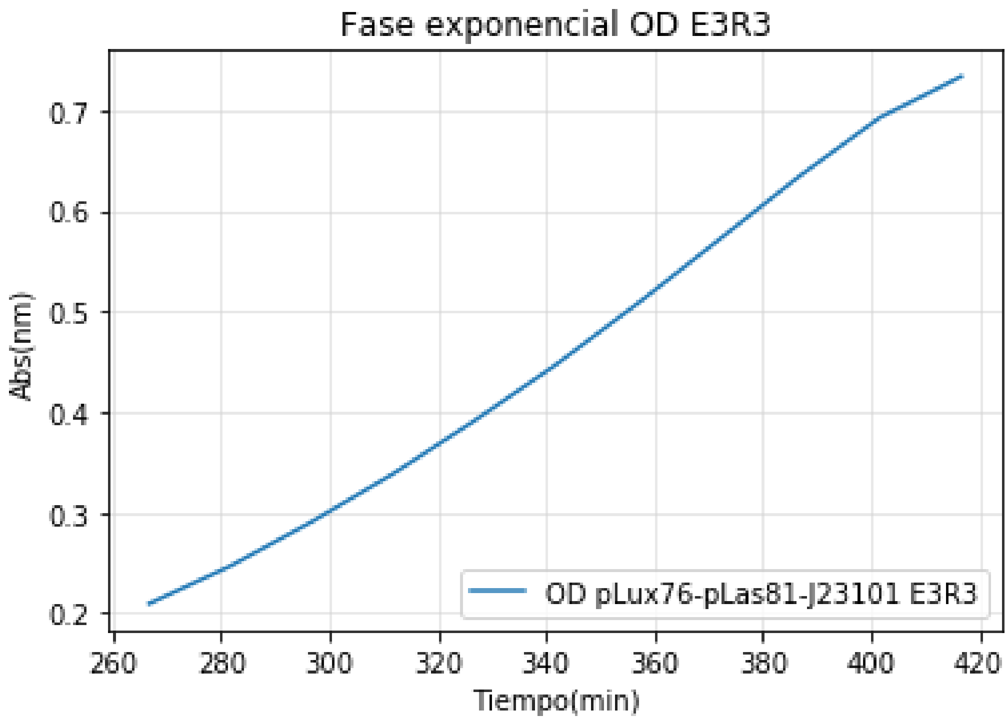



```
A=2.380597e+00
um=1.008574e-02
```

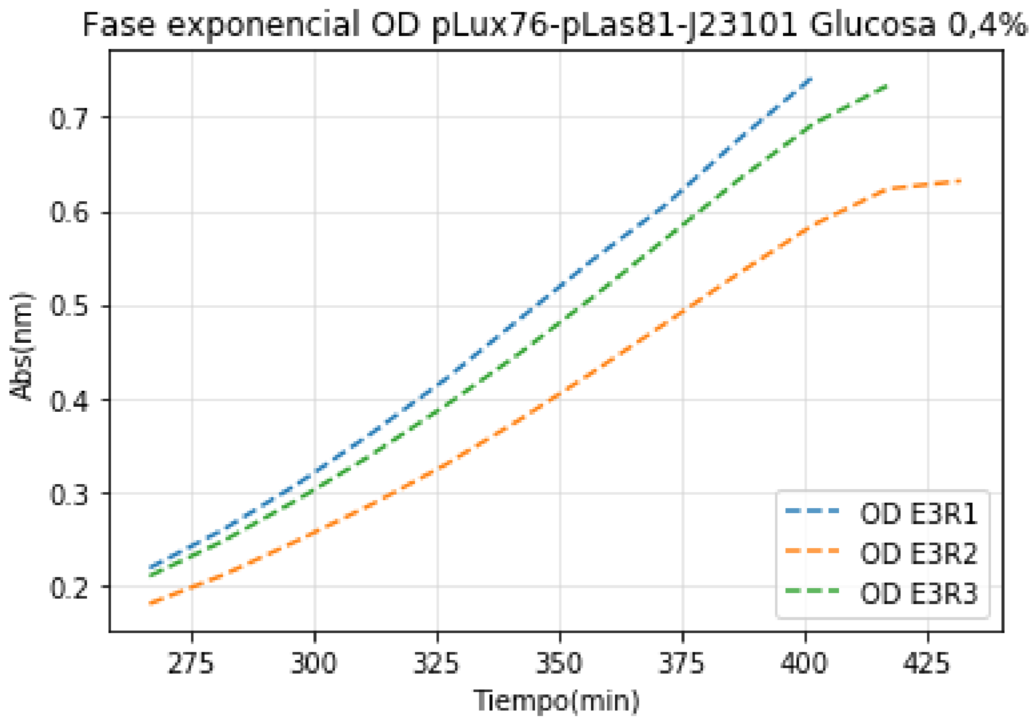
```
l=1.697754e+02
Tm=2.566081e+02
doubpe=6.872546e+01
ext=1.374509e+02
Tfinal=3.940591e+02
```

```
Out[43]: <matplotlib.legend.Legend at 0x1d71d24cf28>
```





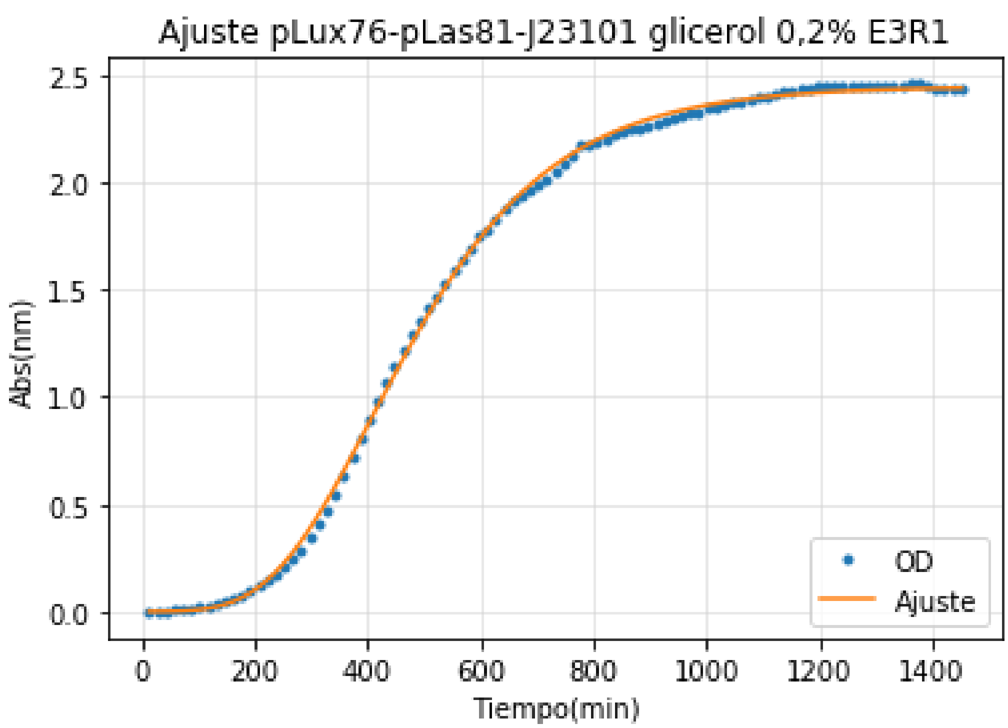

```
In [44]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLas81-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:27],od18271[17:27],'--',label='OD E3R1')
         plt.plot(tt[17:29],od18272[17:29],'--',label='OD E3R2')
         plt.plot(tt[17:28],od18273[17:28],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[44]: <matplotlib.legend.Legend at 0x1d71d5abf60>
```



In [45]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pLas-std glicerol rep 1
y28= np.log(od1827g1)-np.log(np.min(od1827g1))
print('Min OD = %e'%((np.min(od1827g1))))
evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0,
A28= params[0]
um28=params[1]
l28=params[2]
print('A=%e'%(A28))
print('um=%e'%(um28))
print('l=%e'%(l28))

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

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

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

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

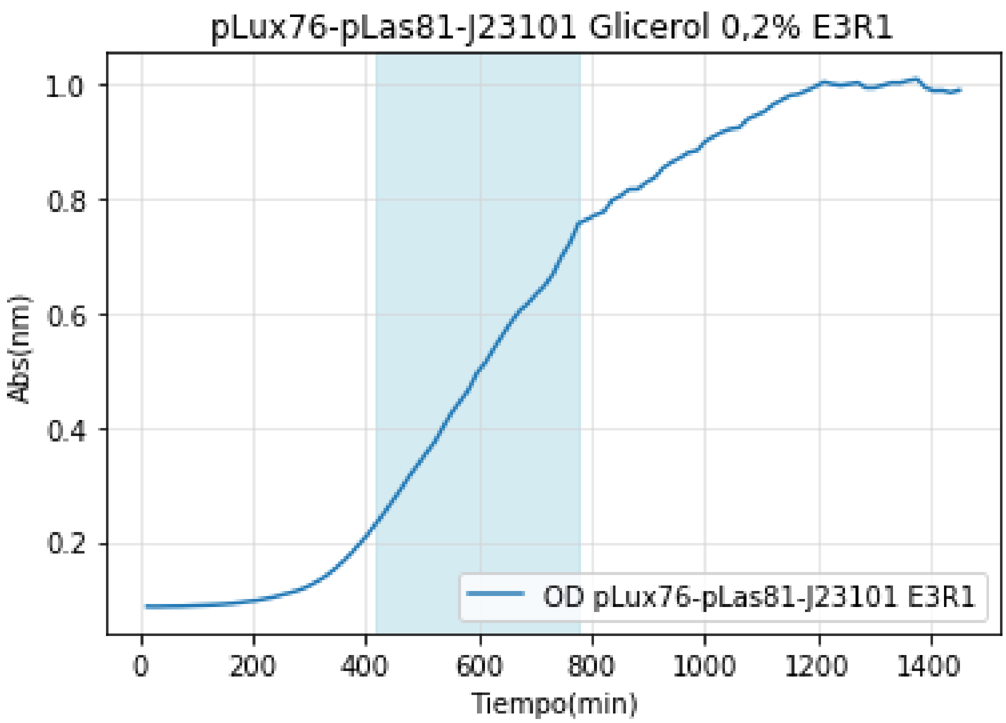
```
Min OD = 8.650000e-02
[  2.44638066e+00   5.03429015e-03   2.27140894e+02]
```

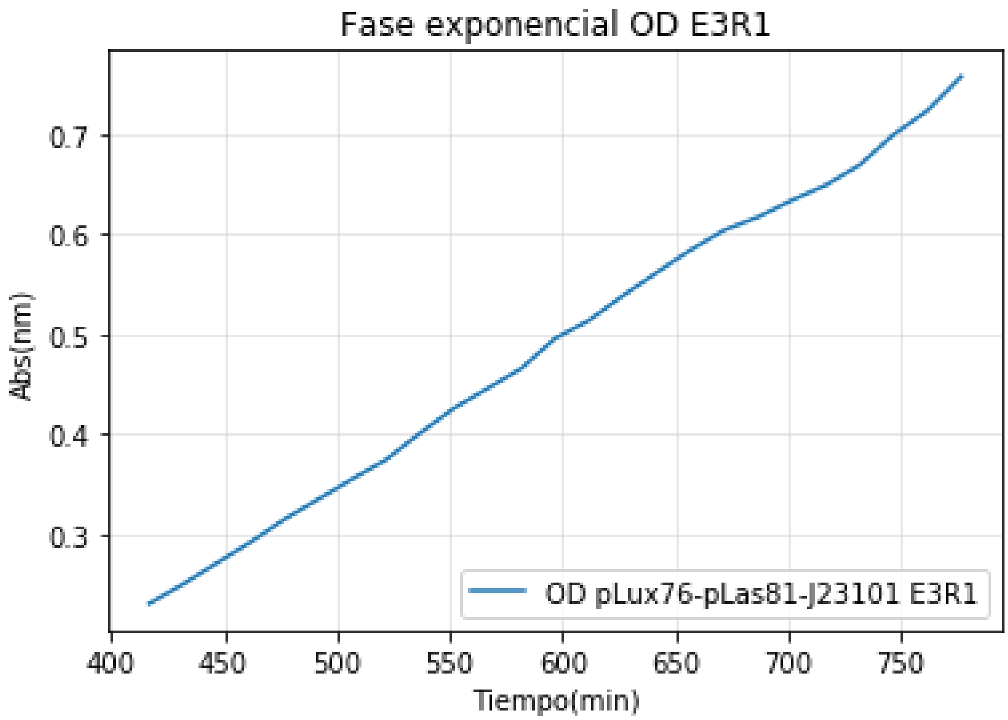



```
A=2.446381e+00
um=5.034290e-03
```

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

```
Out[45]: <matplotlib.legend.Legend at 0x1d71e83f240>
```

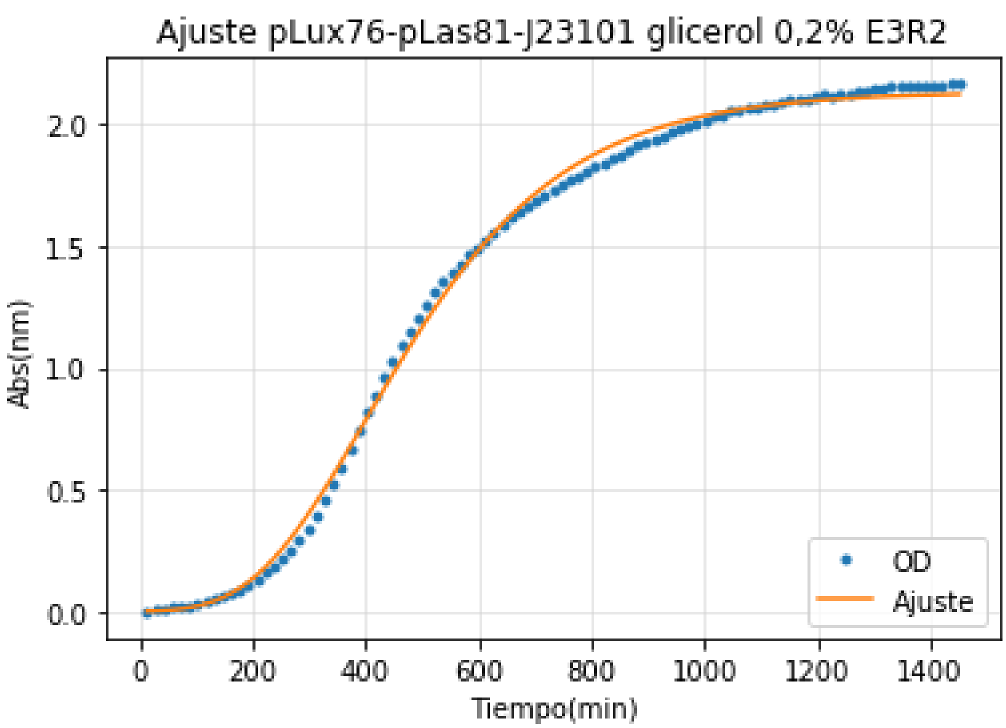




In [46]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pLas-std glicerol rep 2
y29= np.log(od1827g2)-np.log(np.min(od1827g2))
print('Min OD = %e'%((np.min(od1827g2))))
evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0,
A29= params[0]
um29=params[1]
l29=params[2]
print('A=%e'%(A29))
print('um=%e'%(um29))
print('l=%e'%(l29))

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

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

```
y2=tt[57]
plt.figure()
plt.title('pLux76-pLas81-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1827g2,label='OD pLux76-pLas81-J23101 E3R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:58],od1827g2[26:58],label='OD pLux76-pLas81-J23101 E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

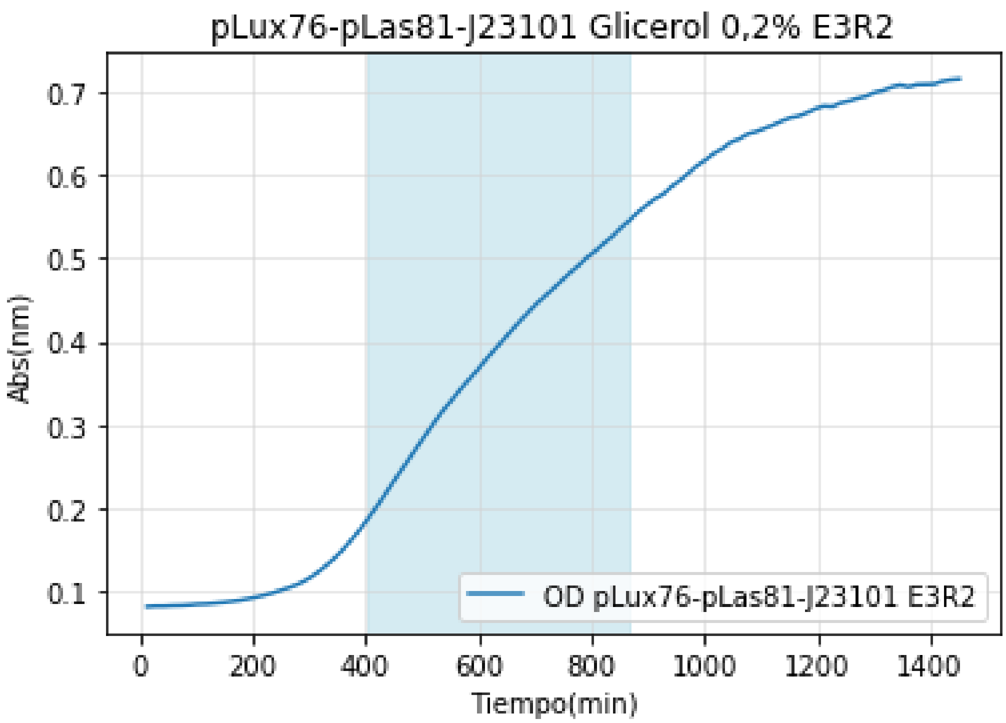
```
Min OD = 8.210000e-02
[  2.13299146e+00   3.96657723e-03   2.01445904e+02]
```

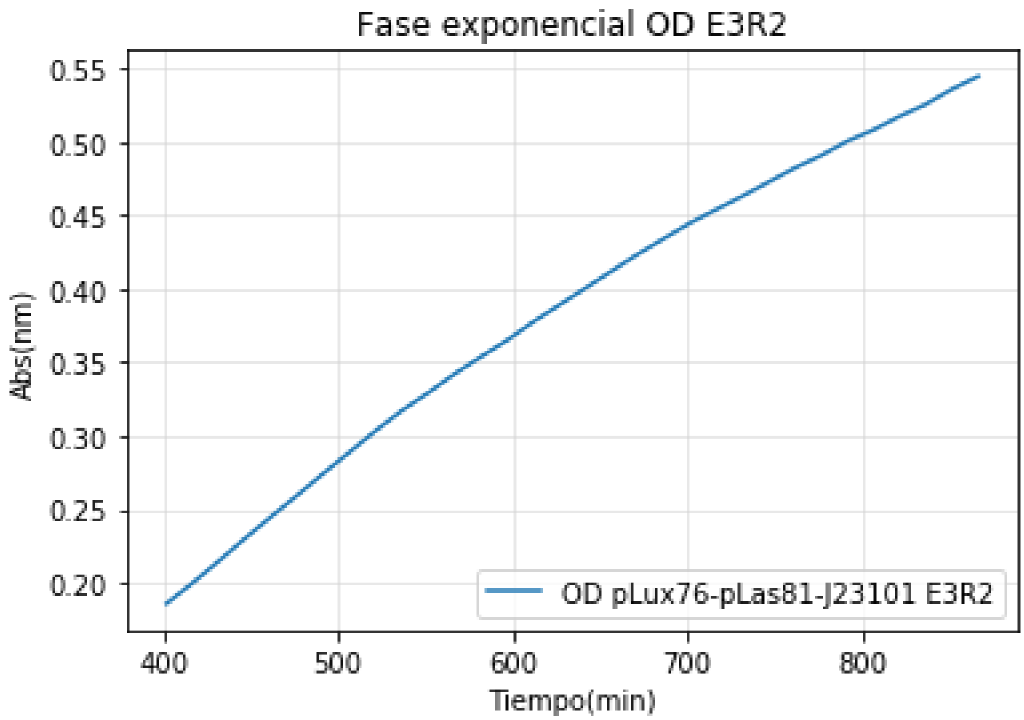



```
A=2.132991e+00
um=3.966577e-03
```

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

```
Out[46]: <matplotlib.legend.Legend at 0x1d71ea3ff98>
```

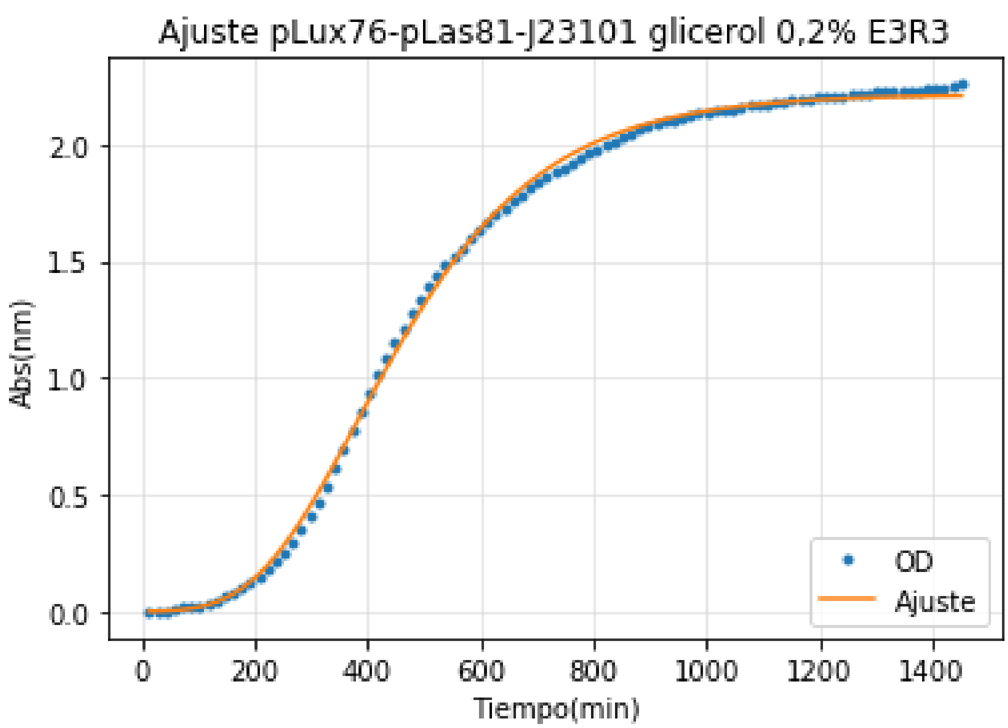





```
In [47]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #pLux-pLas-std glicerol rep 3
         y30= np.log(od1827g3)-np.log(np.min(od1827g3))
         print('Min OD = %e'%((np.min(od1827g3))))
         evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste pLux76-pLas81-J23101 glicerol 0,
         A30= params[0]
         um30=params[1]
         l30=params[2]
         print('A=%e'%(A30))
         print('um=%e'%(um30))
         print('l=%e'%(l30))

         #Cálculo datos para determinar extensión de la fase exponencial
         tm30=((A30/(np.exp(1)*um30))+l30)
         print('Tm=%e'%(tm30))
         t230=((np.log(2))/um30)
         print('doubpe=%e'%(t230))
         extdp30=2.5*t230
         print('ext=%e'%extdp30)
         ttot30=tm30+extdp30
         print('Tfinal=%e'%ttot30)

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

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

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

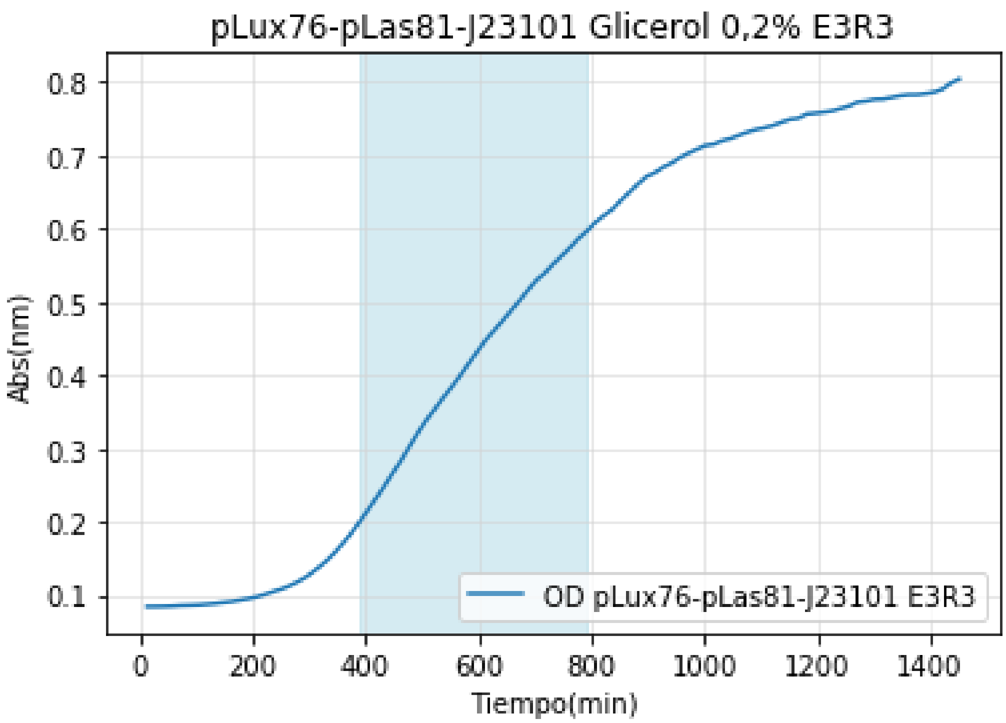
```
Min OD = 8.420000e-02
[  2.21392685e+00   4.49243847e-03   1.99905763e+02]
```

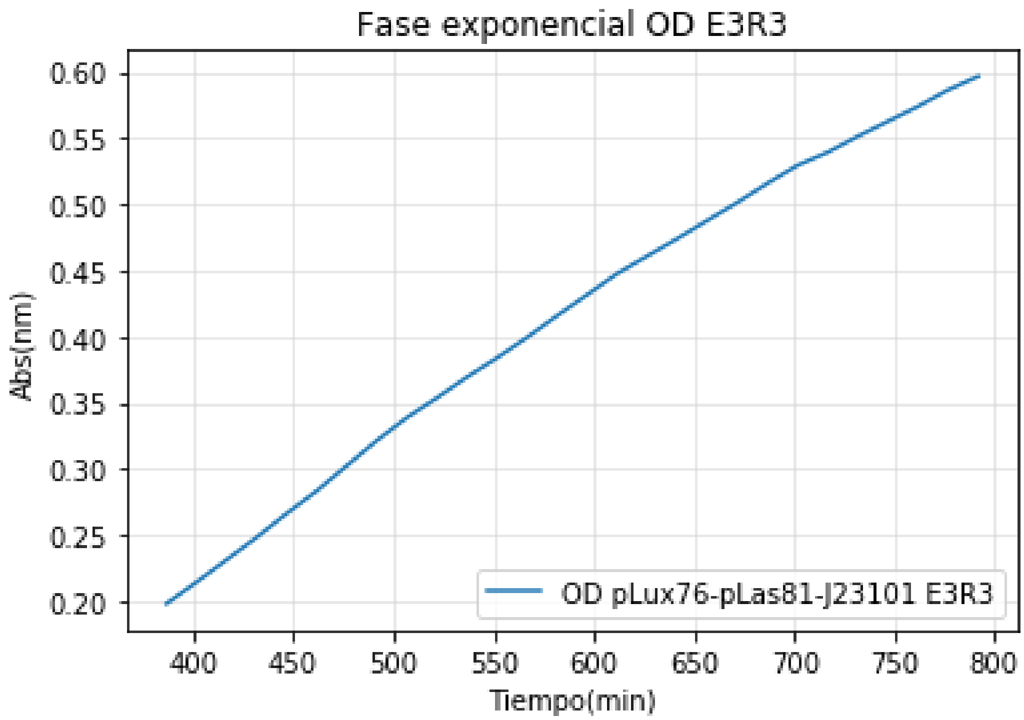



```
A=2.213927e+00
um=4.492438e-03
```

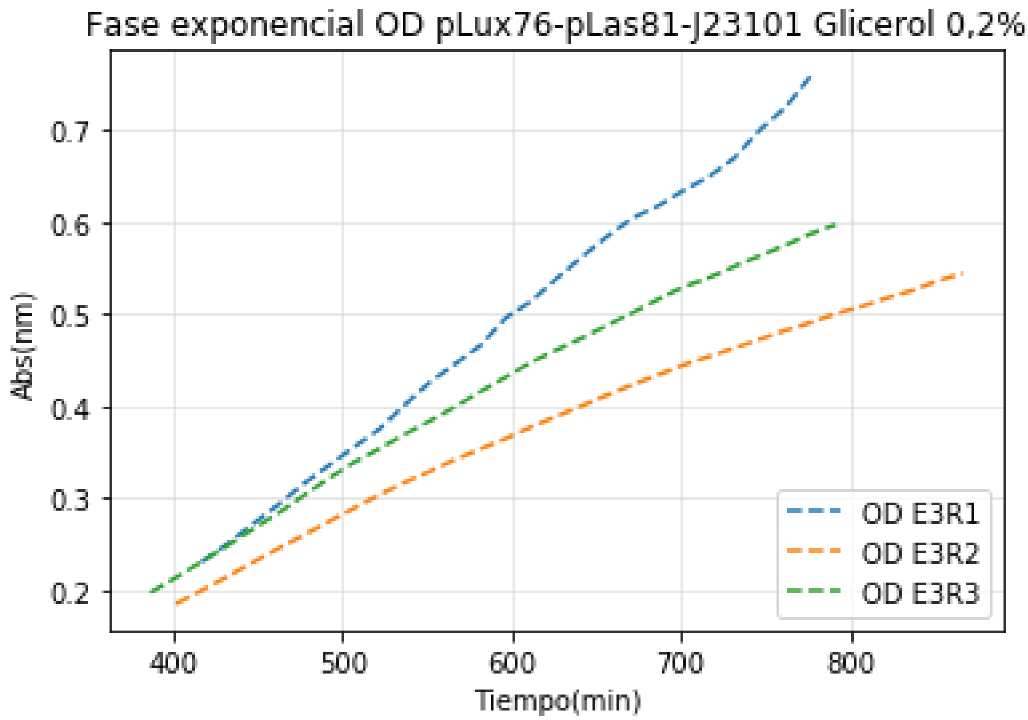
```
l=1.999058e+02
Tm=3.812011e+02
doubpe=1.542920e+02
ext=3.857299e+02
Tfinal=7.669310e+02
```

Out[47]: <matplotlib.legend.Legend at 0x1d71d560630>



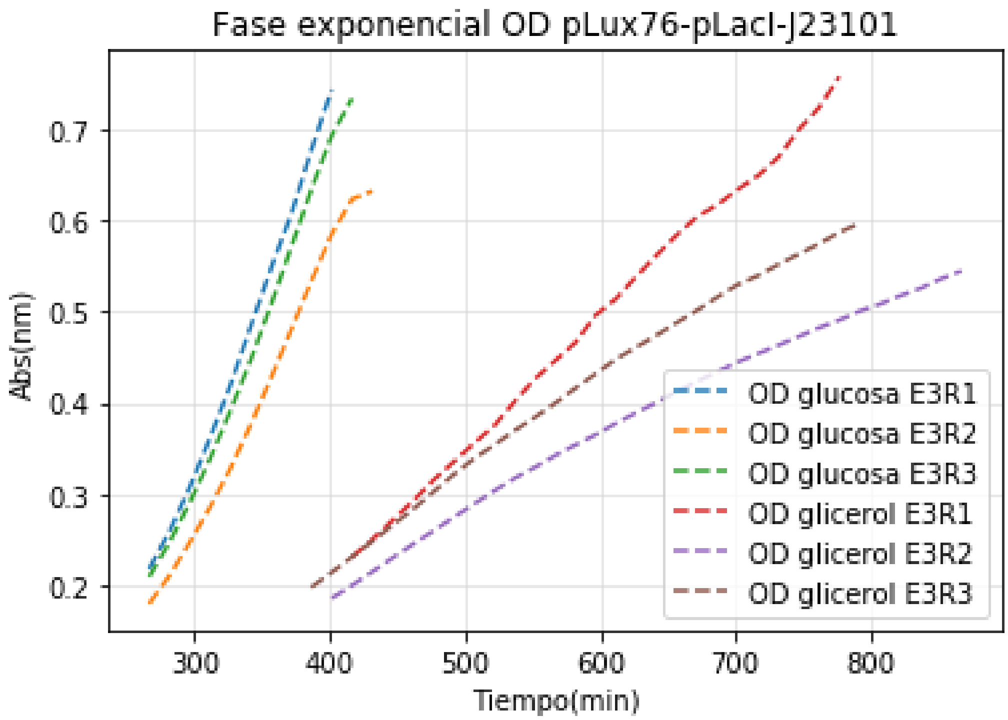



```
In [48]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLas81-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[27:52],od1827g1[27:52],'--',label='OD E3R1')
         plt.plot(tt[26:58],od1827g2[26:58],'--',label='OD E3R2')
         plt.plot(tt[25:53],od1827g3[25:53],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[48]: <matplotlib.legend.Legend at 0x1d71b62ce80>
```



In [49]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD pLux76-pLas81-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:27],od18271[17:27],'--',label='OD glucosa E3R1')
plt.plot(tt[17:29],od18272[17:29],'--',label='OD glucosa E3R2')
plt.plot(tt[17:28],od18273[17:28],'--',label='OD glucosa E3R3')
plt.plot(tt[27:52],od1827g1[27:52],'--',label='OD glicerol E3R1')
plt.plot(tt[26:58],od1827g2[26:58],'--',label='OD glicerol E3R2')
plt.plot(tt[25:53],od1827g3[25:53],'--',label='OD glicerol E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[49]: <matplotlib.legend.Legend at 0x1d71d6887f0>



In [50]:
```
#Selección de datos en arrays, según lo determinado
#controles glucosa
o1=odcg1[16:26]
c1=cfpcg1[16:26]
r1=rfpcg1[16:26]
y1=yfpcg1[16:26]

o2=odcg2[17:30]
c2=cfpcg2[17:30]
r2=rfpcg2[17:30]
y2=yfpcg2[17:30]

o3=odcg3[17:29]
c3=cfpcg3[17:29]
r3=rfpcg3[17:29]
y3=yfpcg3[17:29]

#controles glicerol
o4=odcgl1[25:48]
c4=cfpcgl1[25:48]
r4=rfpcgl1[25:48]
y4=yfpcgl1[25:48]

o5=odcgl2[25:53]
```

```
c5=cfpcgl2[25:53]
r5=rfpcgl2[25:53]
y5=yfpcgl2[25:53]

o6=odcgl3[24:51]
c6=cfpcgl3[24:51]
r6=rfpcgl3[24:51]
y6=yfpcgl3[24:51]

#plux-std-std glucosa
o7=od18211[16:27]
c7=cfp18211[16:27]
r7=rfp18211[16:27]
y7=yfp18211[16:27]

o8=od18212[17:30]
c8=cfp18212[17:30]
r8=rfp18212[17:30]
y8=yfp18212[17:30]

o9=od18213[18:30]
c9=cfp18213[18:30]
r9=rfp18213[18:30]
y9=yfp18213[18:30]

#plux-std-std glicerol
o10=od1821g1[24:47]
c10=cfp1821g1[24:47]
r10=rfp1821g1[24:47]
y10=yfp1821g1[24:47]

o11=od1821g2[32:60]
c11=cfp1821g2[32:60]
r11=rfp1821g2[32:60]
y11=yfp1821g2[32:60]

o12=od1821g3[26:50]
c12=cfp1821g3[26:50]
r12=rfp1821g3[26:50]
y12=yfp1821g3[26:50]

#plux-107-std glucosa
o13=od18231[16:26]
c13=cfp18231[16:26]
r13=rfp18231[16:26]
y13=yfp18231[16:26]

o14=od18232[19:30]
```

```
c14=cfp18232[19:30]
r14=rfp18232[19:30]
y14=yfp18232[19:30]

o15=od18233[17:28]
c15=cfp18233[17:28]
r15=rfp18233[17:28]
y15=yfp18233[17:28]

#plux-107-std glicerol
o16=od1823g1[25:47]
c16=cfp1823g1[25:47]
r16=rfp1823g1[25:47]
y16=yfp1823g1[25:47]

o17=od1823g2[25:52]
c17=cfp1823g2[25:52]
r17=rfp1823g2[25:52]
y17=yfp1823g2[25:52]

o18=od1823g3[25:53]
c18=cfp1823g3[25:53]
r18=rfp1823g3[25:53]
y18=yfp1823g3[25:53]

#plux-pLac-std glucosa
o19=od18261[15:25]
c19=cfp18261[15:25]
r19=rfp18261[15:25]
y19=yfp18261[15:25]

o20=od18262[18:31]
c20=cfp18262[18:31]
r20=rfp18262[18:31]
y20=yfp18262[18:31]

o21=od18263[17:29]
c21=cfp18263[17:29]
r21=rfp18263[17:29]
y21=yfp18263[17:29]

#plux-pLac-std glicerol
o22=od1826g1[25:49]
c22=cfp1826g1[25:49]
r22=rfp1826g1[25:49]
y22=yfp1826g1[25:49]

o23=od1826g2[25:53]
```

```
c23=cfp1826g2[25:53]
r23=rfp1826g2[25:53]
y23=yfp1826g2[25:53]

o24=od1826g3[25:53]
c24=cfp1826g3[25:53]
r24=rfp1826g3[25:53]
y24=yfp1826g3[25:53]

#plux-pLas-std glucosa
o25=od18271[17:27]
c25=cfp18271[17:27]
r25=rfp18271[17:27]
y25=yfp18271[17:27]

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

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

#plux-pLas-std glicerol
o28=od1827g1[27:52]
c28=cfp1827g1[27:52]
r28=rfp1827g1[27:52]
y28=yfp1827g1[27:52]

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

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

```
In [51]: #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

#pLux76-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
```

```
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

#pLux76-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

#plux-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

#plux-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

#plux-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)
```

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

#plux-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

#plux-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
```

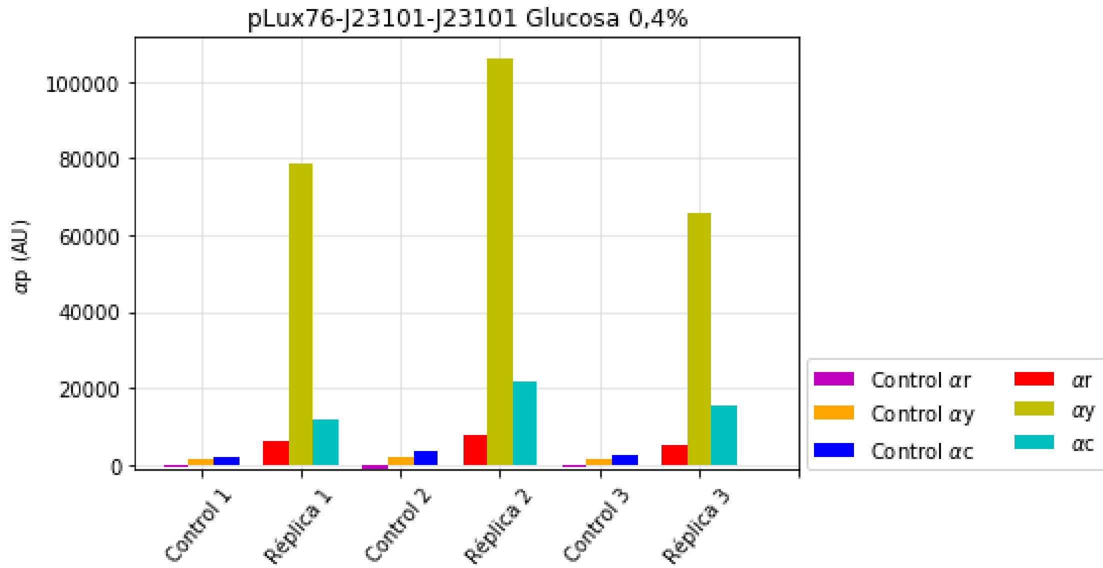
```
#plux-pLas81-std glicerol
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,c28)
slopec28=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,r28)
sloper28=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o28,y28)
slopey28=slope

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 [52]:

```
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 plux-std-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pLux76-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)
```

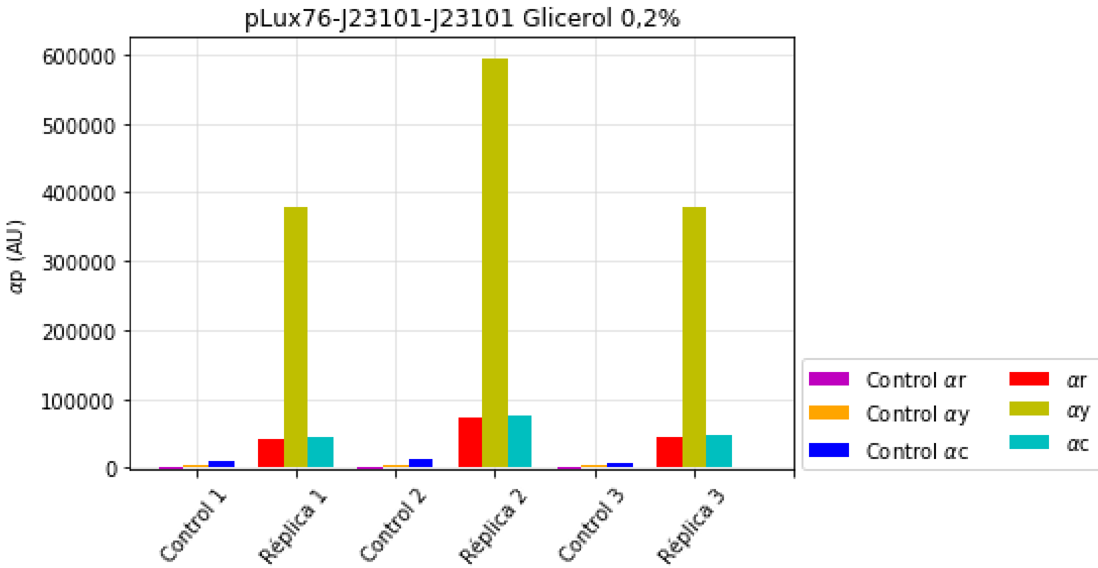
Out[52]: <matplotlib.legend.Legend at 0x1d71eacc4a8>




```
In [53]: #Grafico pendientes plux-std-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pLux76-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)
```

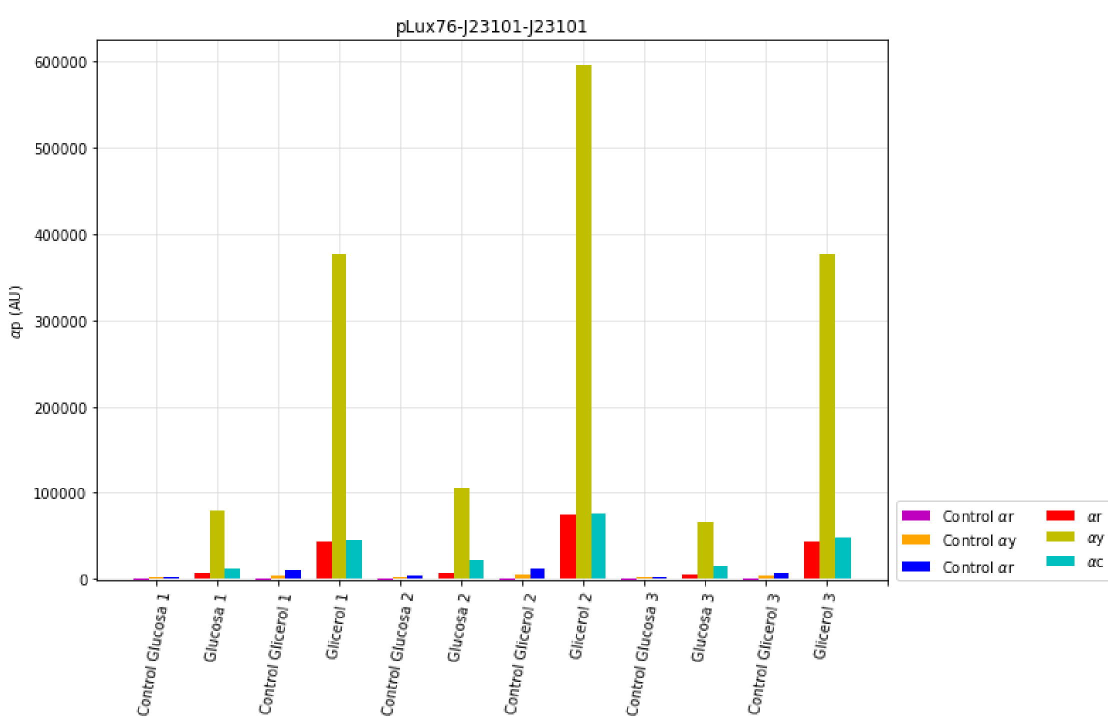
Out[53]: <matplotlib.legend.Legend at 0x1d71ed70518>



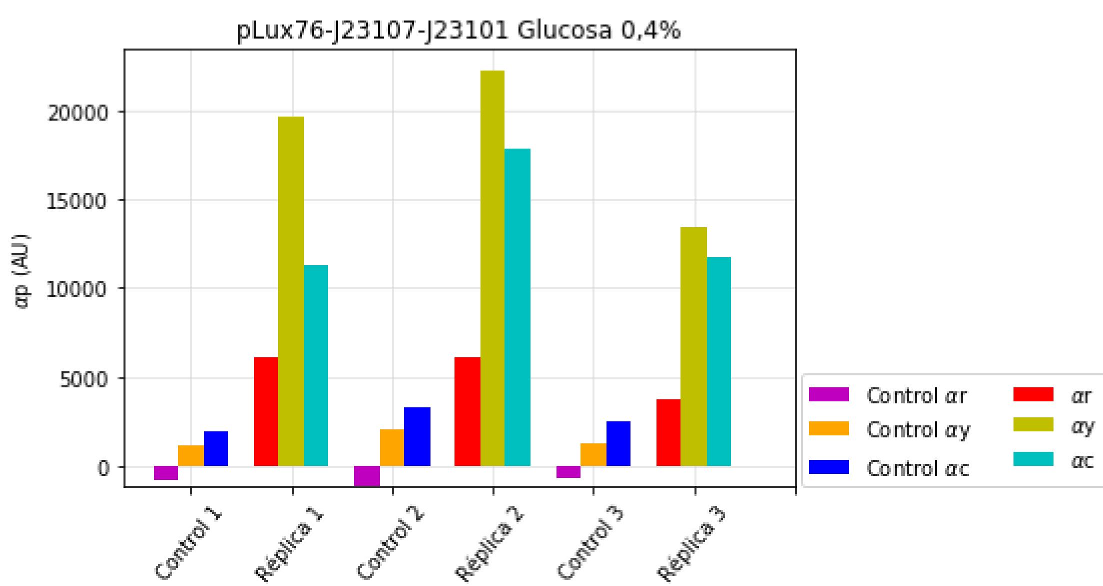
In [54]:
```
#Grafico pendientes plux-std-std
X = np.arange(13)
plt.figure(figsize=(10,7))
plt.title('pLux76-J23101-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ 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)
```

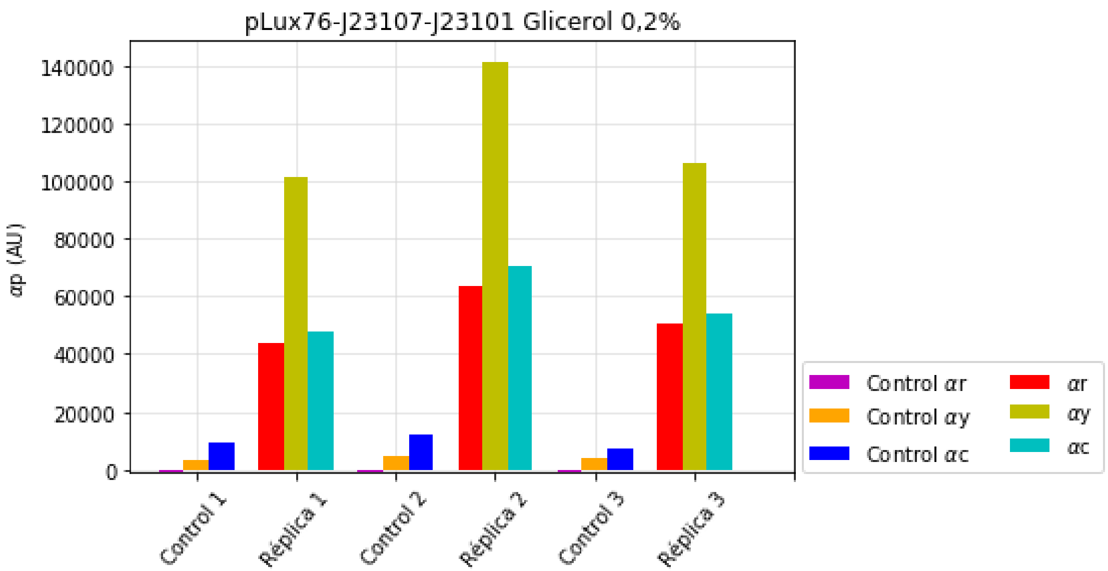
Out[54]: <matplotlib.legend.Legend at 0x1d71ef37cf8>



```
In [55]: #Grafico pendientes plux-107-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-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[55]: <matplotlib.legend.Legend at 0x1d71f3974a8>
```

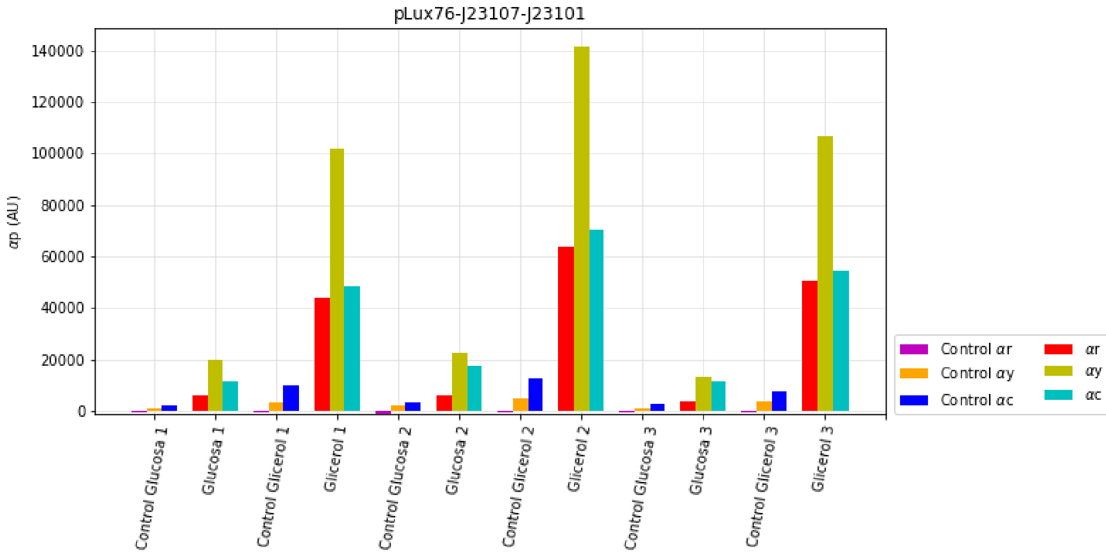


```
In [56]: #Grafico pendientes plux-107-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-J23107-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         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[56]: <matplotlib.legend.Legend at 0x1d71e755748>
```

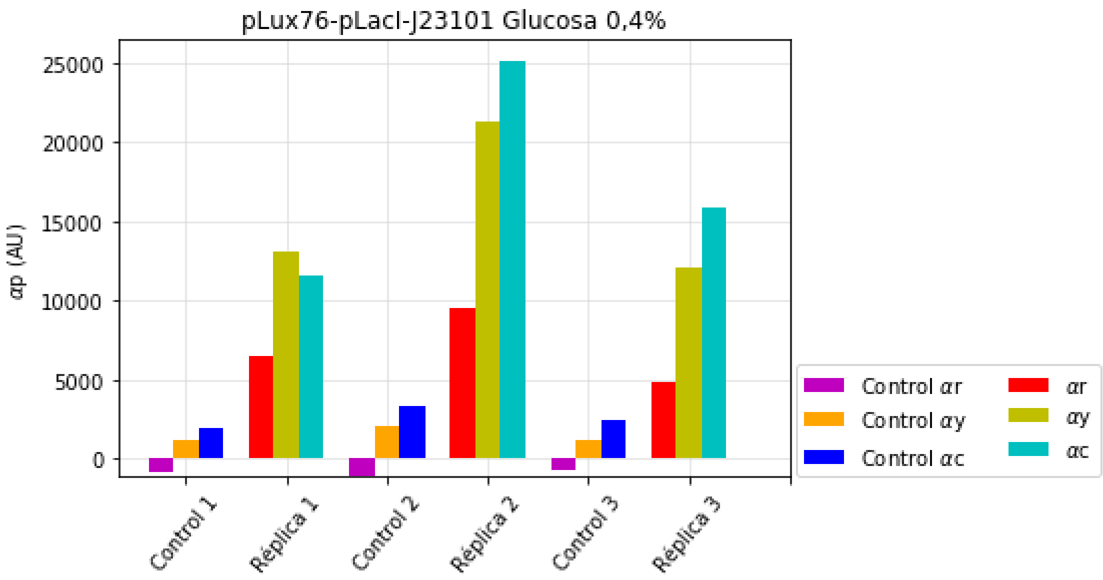


In [57]:
```python
#Grafico pendientes plux-std-std
X = np.arange(13)
plt.figure(figsize=(10,5))
plt.title('pLux76-J23107-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ 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[57]: <matplotlib.legend.Legend at 0x1d71f02d898>



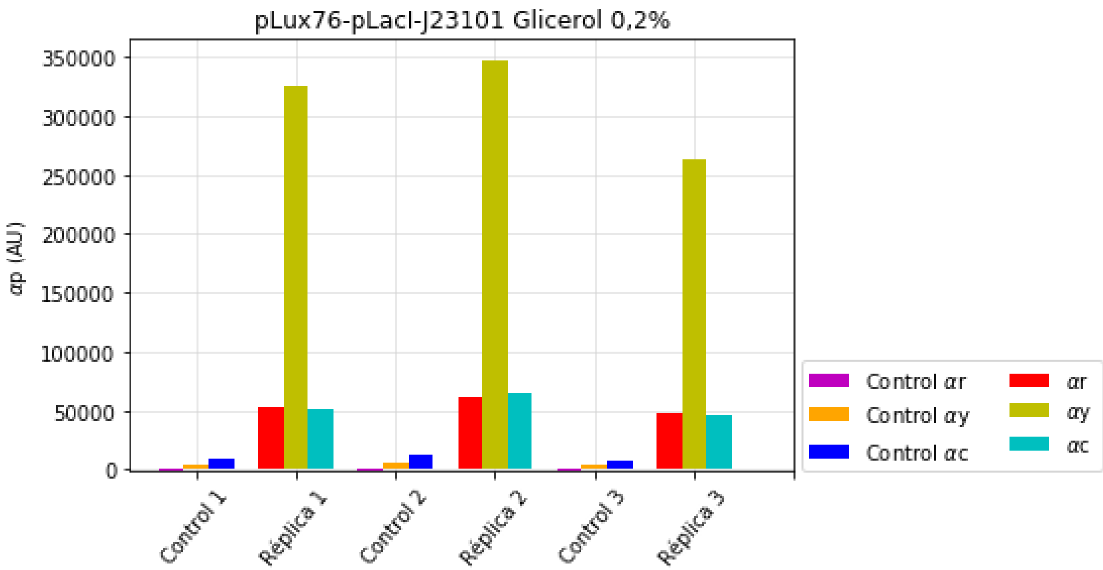

```
In [58]: #Grafico pendientes plux-plac-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-pLacI-J23101 Glucosa 0,4%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         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[58]: <matplotlib.legend.Legend at 0x1d71f5ac080>
```



In [59]:

```python
#Grafico pendientes plux-plac-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pLux76-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)
```

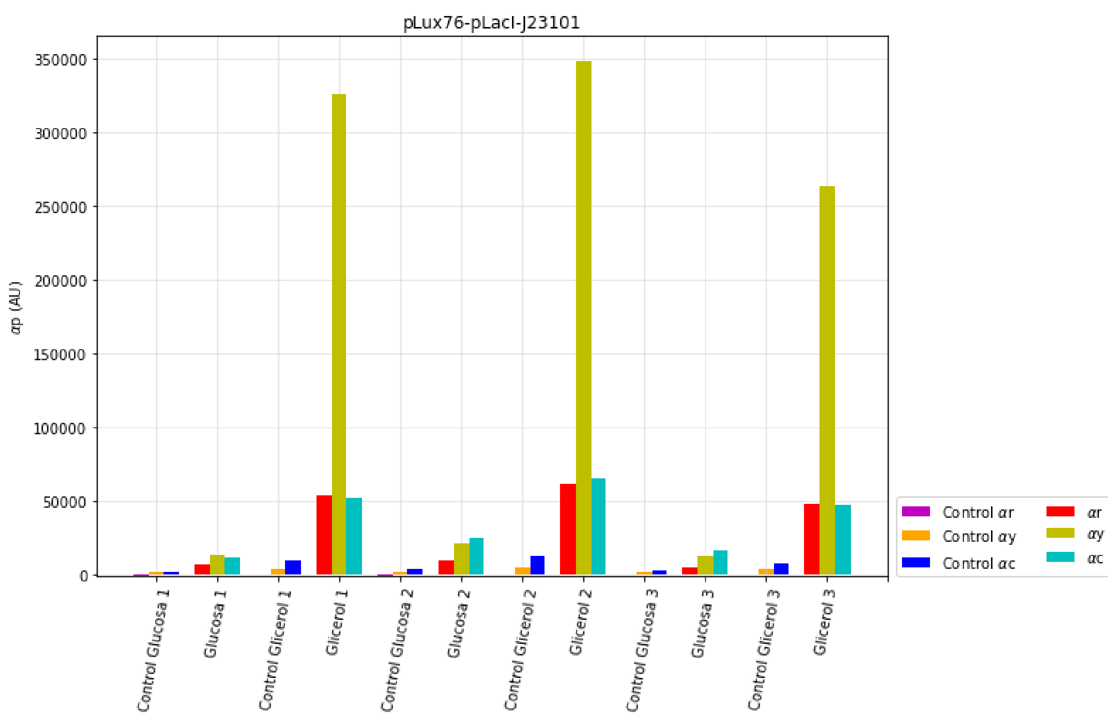
Out[59]: <matplotlib.legend.Legend at 0x1d71f6dfda0>




```
In [60]: #Grafico pendientes plux-lac-std
         X = np.arange(13)
         plt.figure(figsize=(10,7))
         plt.title('pLux76-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)
```

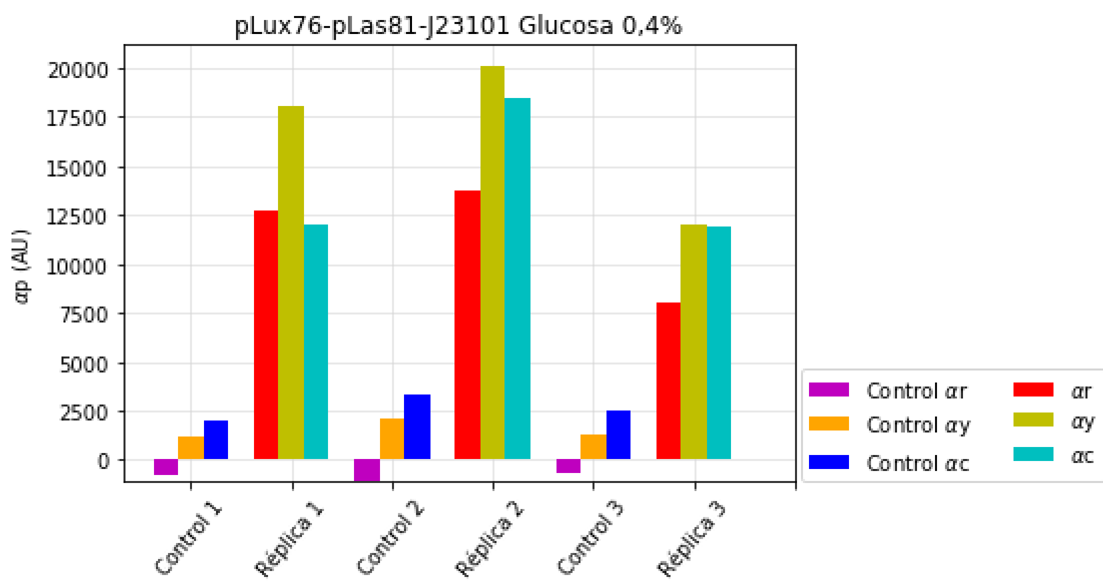
Out[60]: <matplotlib.legend.Legend at 0x1d71f895160>



In [61]:
```
#Grafico pendientes plux-pLas-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pLux76-pLas81-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
```

```
plt.bar(X[0]-0.25,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)
```

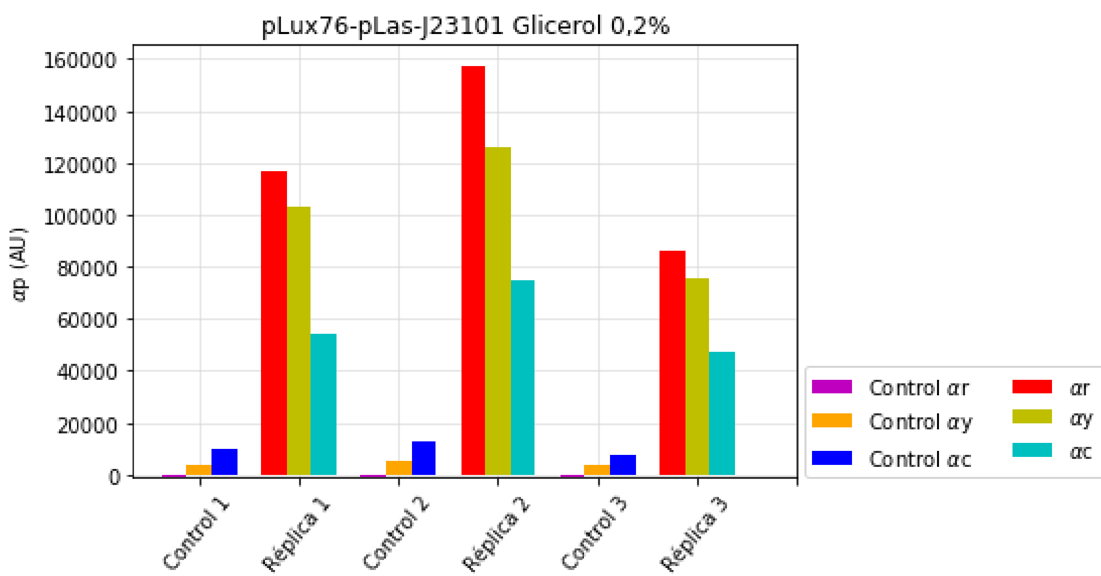
Out[61]: <matplotlib.legend.Legend at 0x1d71fd17048>



In [62]:
```
#Grafico pendientes plux-plas-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pLux76-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)
```

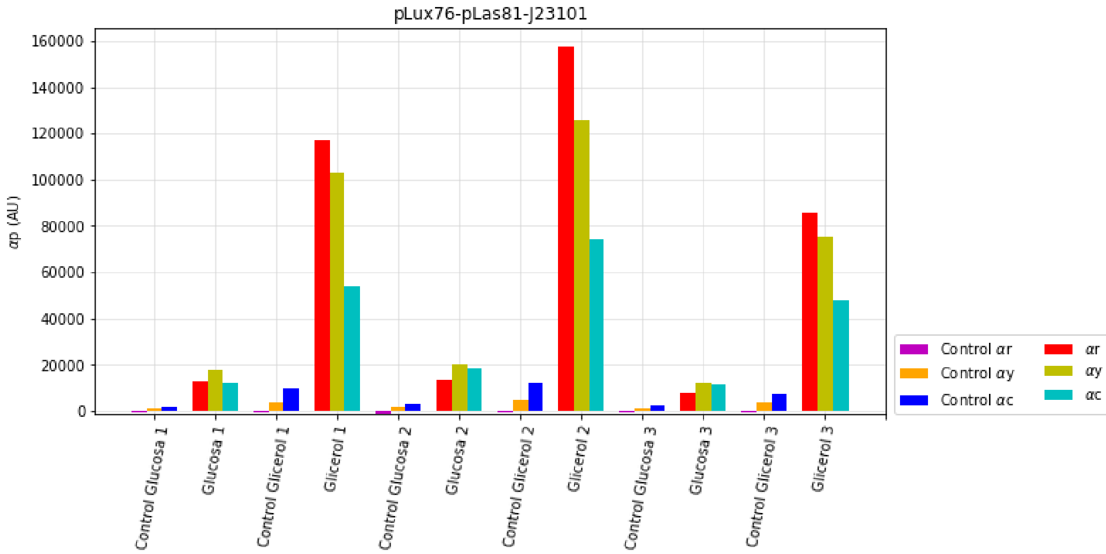
Out[62]: <matplotlib.legend.Legend at 0x1d71eace4e0>




```
In [63]: #Grafico pendientes plux-plas-std
         X = np.arange(13)
         plt.figure(figsize=(10,5))
```

```
plt.title('pLux76-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[63]: <matplotlib.legend.Legend at 0x1d720d324a8>
```

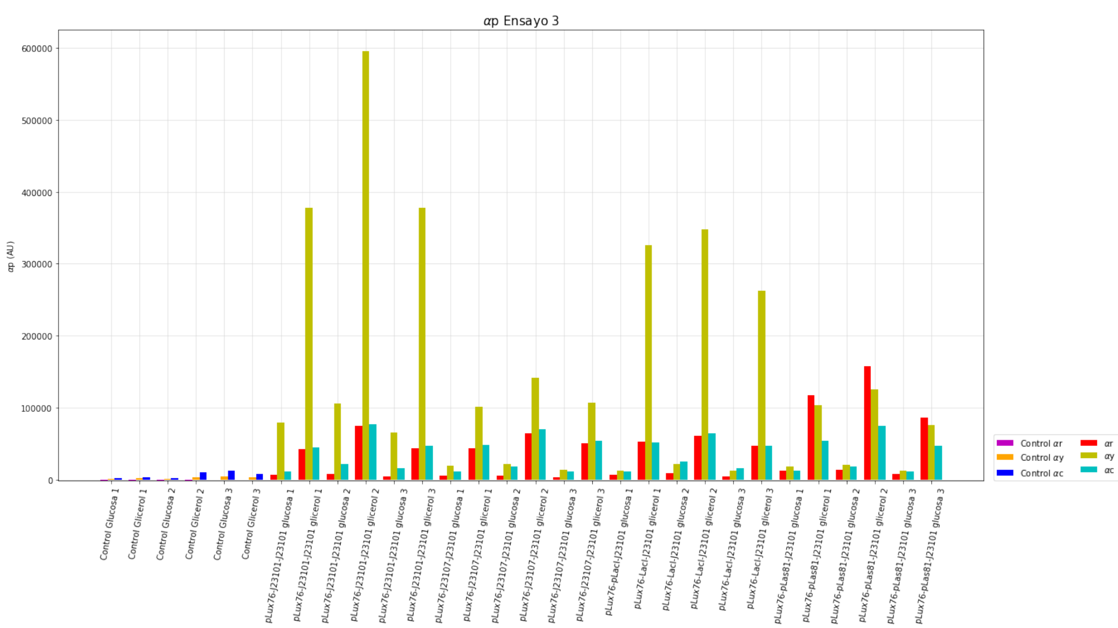



```
In [141]: #Grafico pendientes todo
          X = np.arange(30)
          plt.figure(figsize=(20,10))
          plt.title(r'$\alpha$p Ensayo 3',fontsize=15.0)
          plt.ylabel(r'$\alpha$p (AU)')
          plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
          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)
```

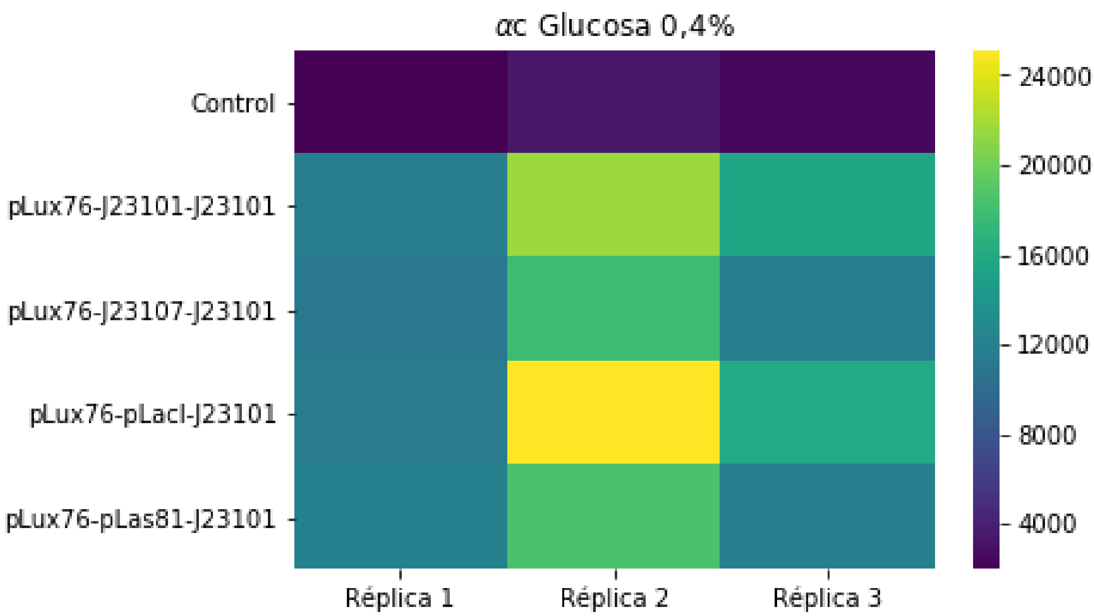
Out[141]: <matplotlib.legend.Legend at 0x1d72c6e60b8>

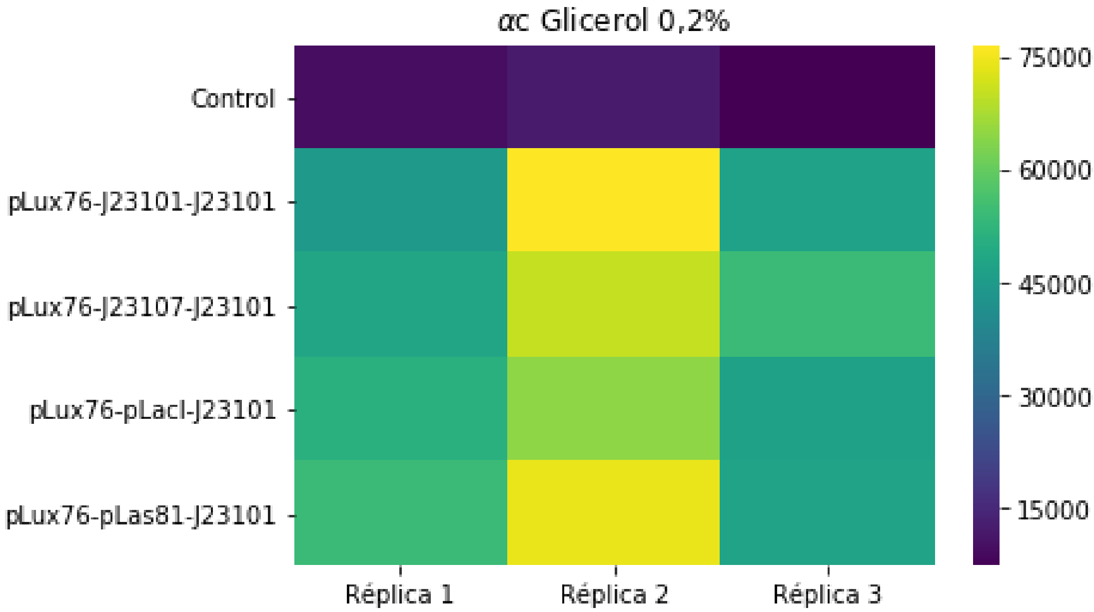



```
In [65]: cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15],
         cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec1
         rglu=[[sloper1,sloper2,sloper3],[sloper7,sloper8,sloper9],[sloper13,sloper14,sloper15],
```

```
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
```

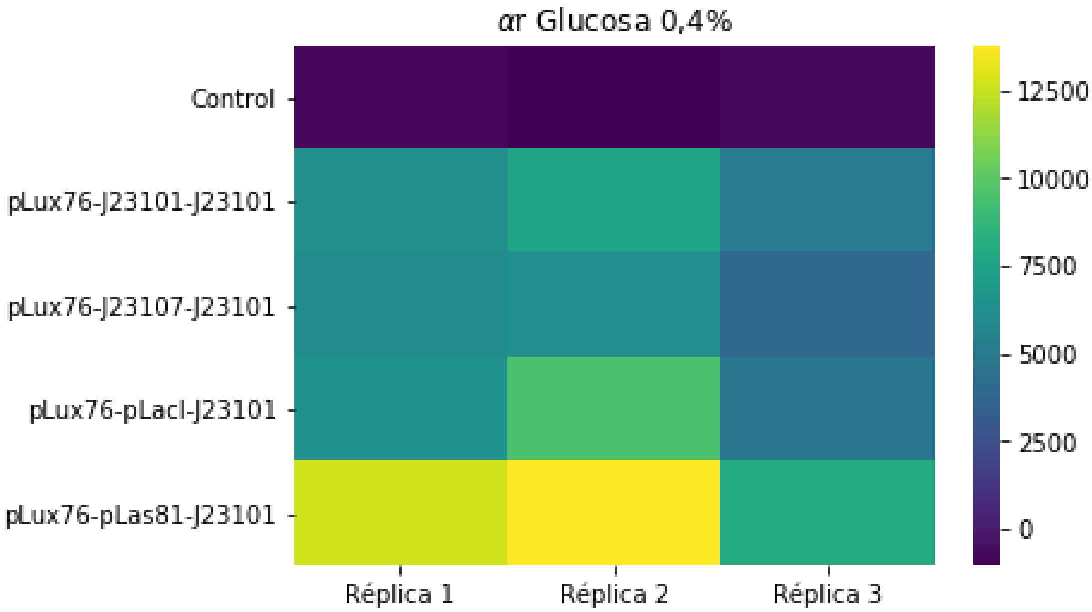
```
In [66]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

         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[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1d721aa0da0>
```

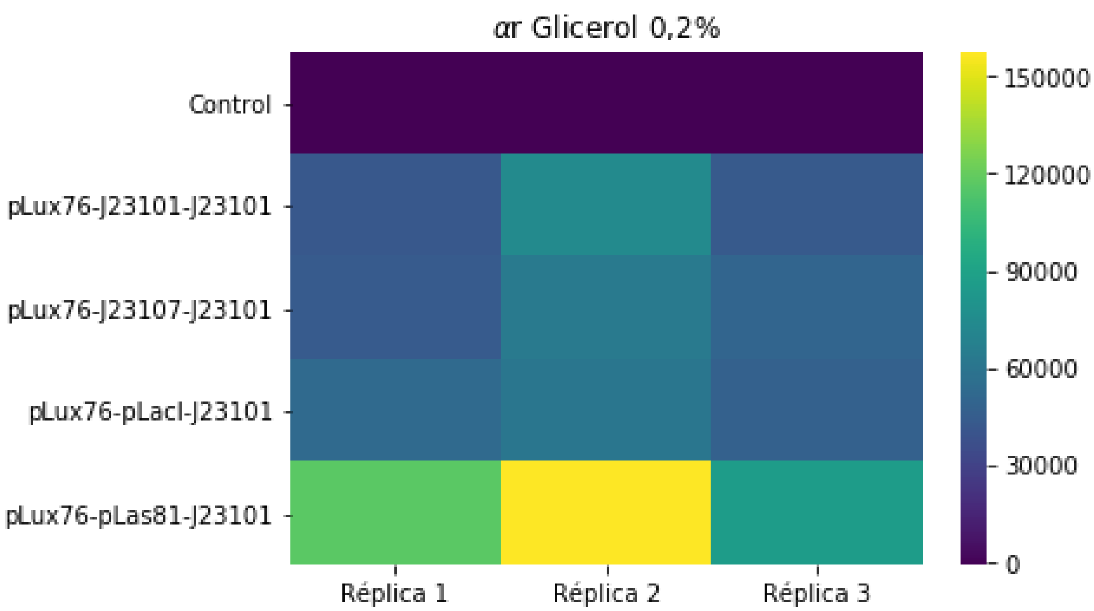




In [67]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','


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[67]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d721bf08d0>
```

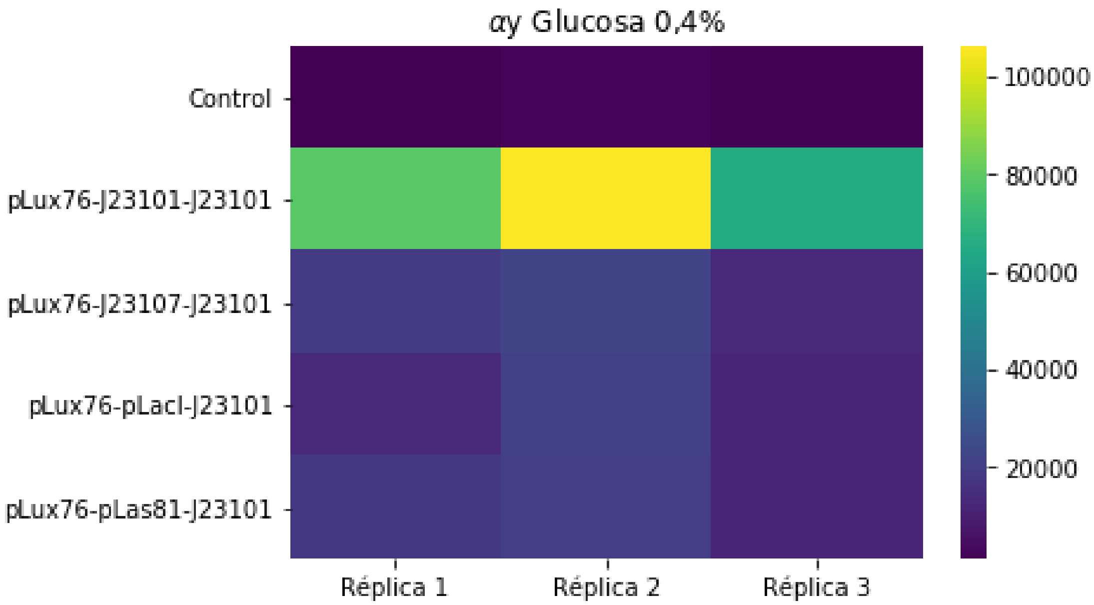


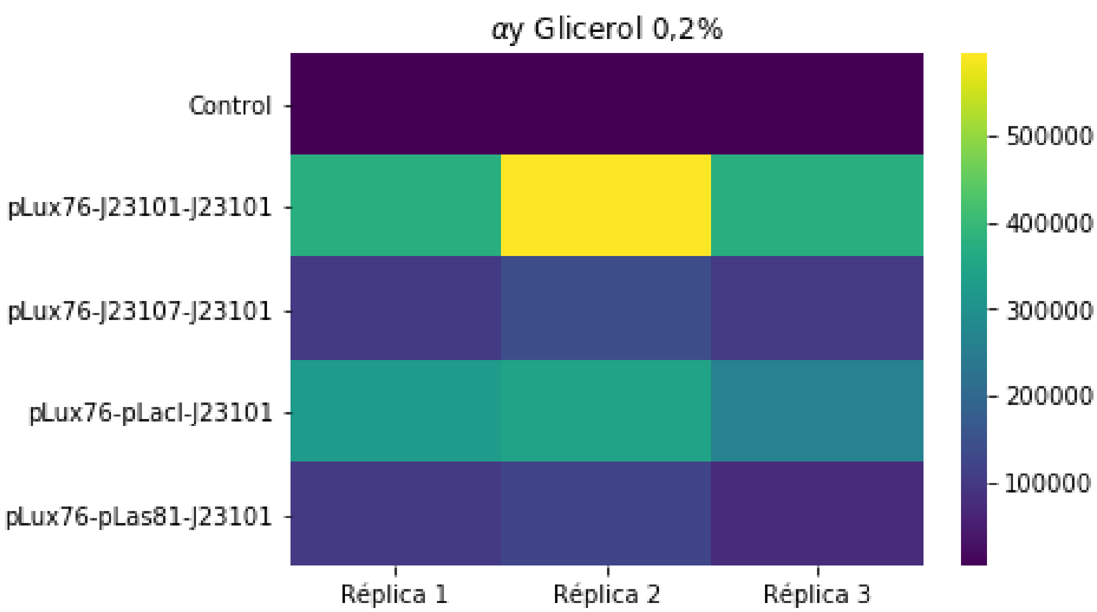


In [68]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

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

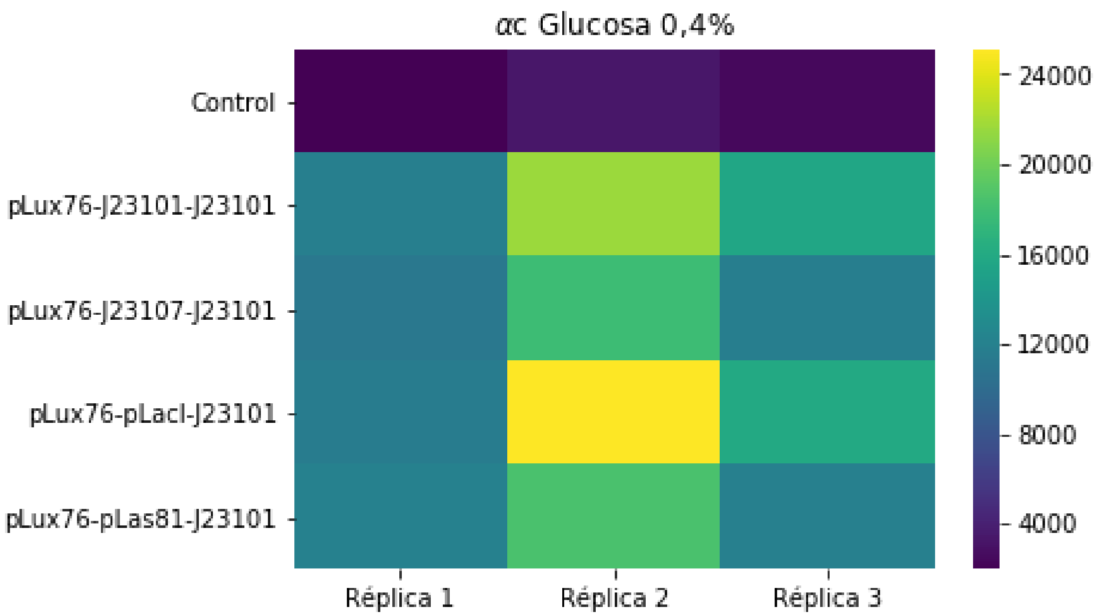
```
sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

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

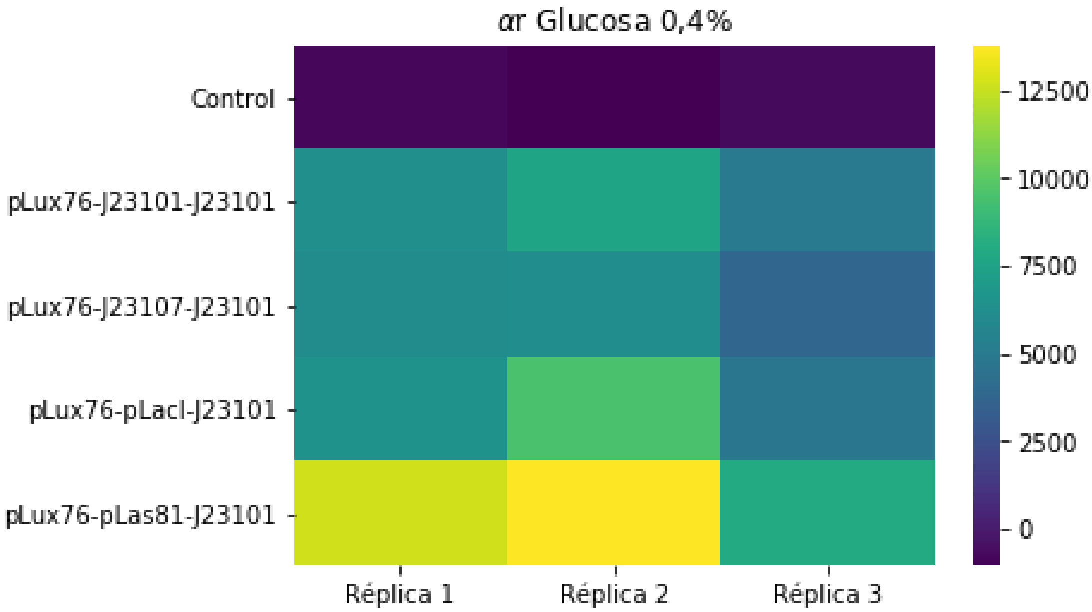
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x1d720d555f8>

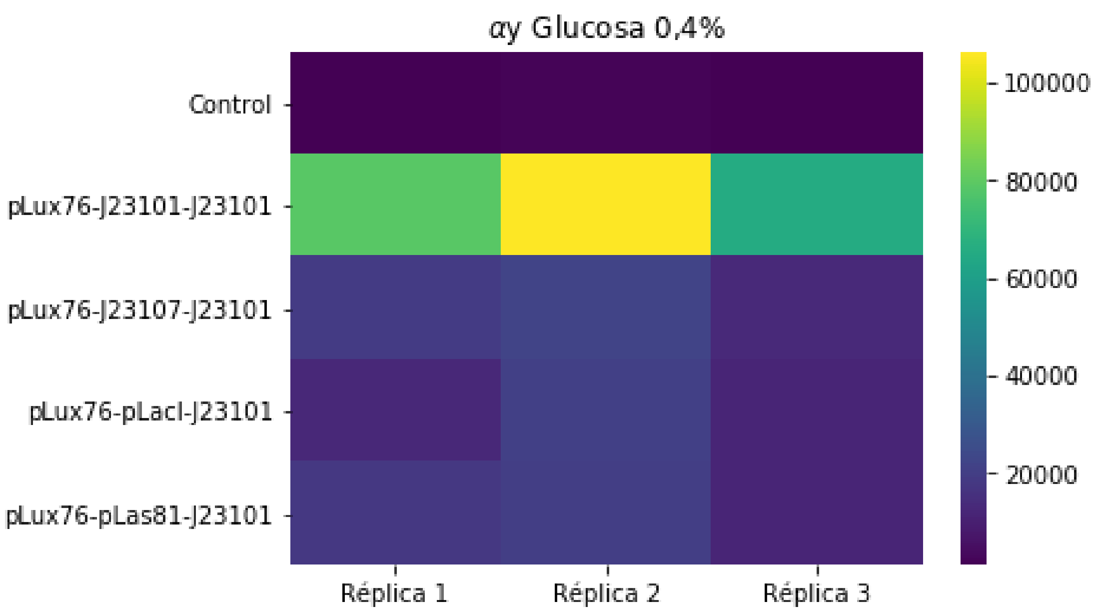




```
In [69]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

         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[69]: <matplotlib.axes._subplots.AxesSubplot at 0x1d721ae8710>
```



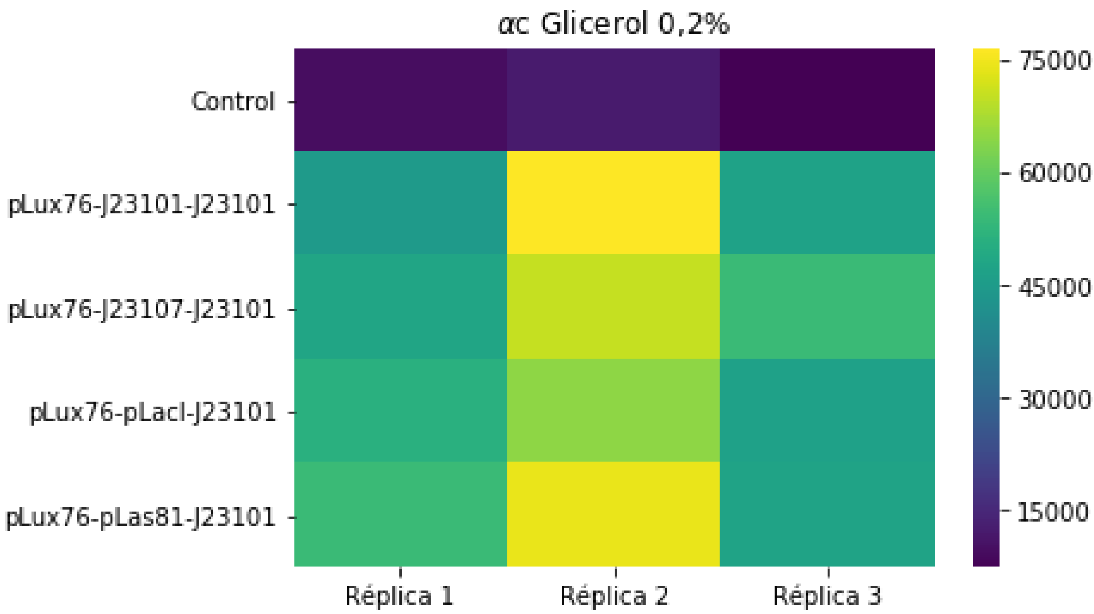


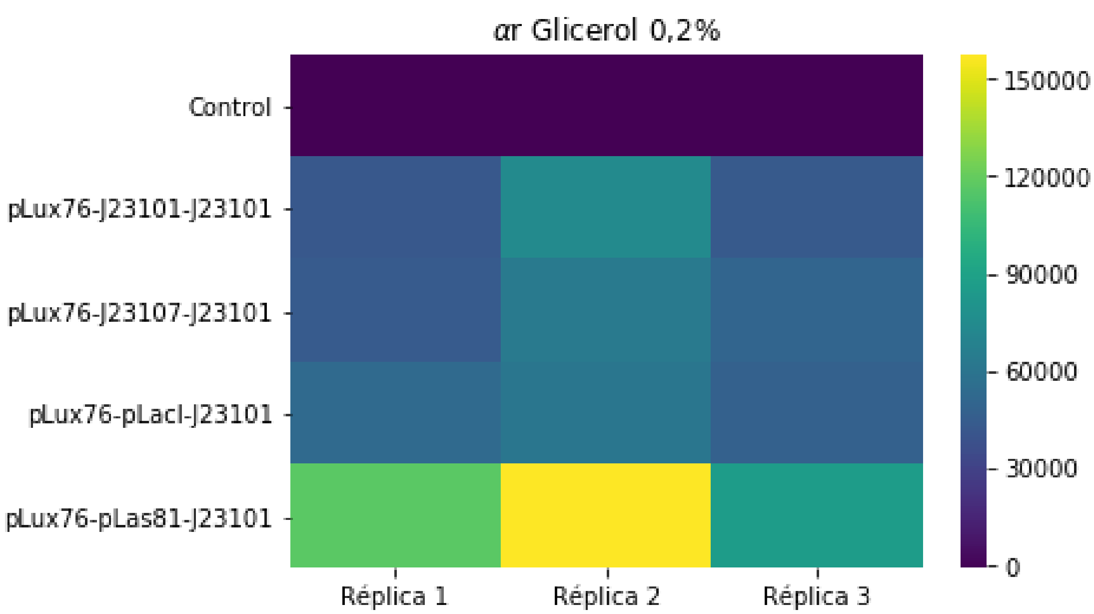


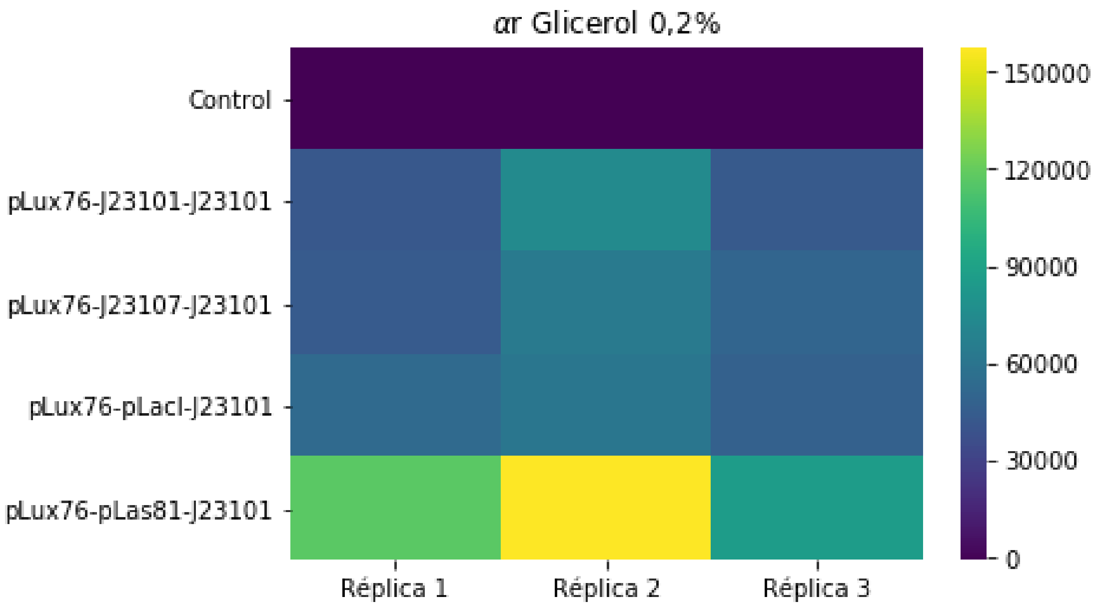

```
In [70]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

         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[70]: <matplotlib.axes._subplots.AxesSubplot at 0x1d7213e8f28>
```



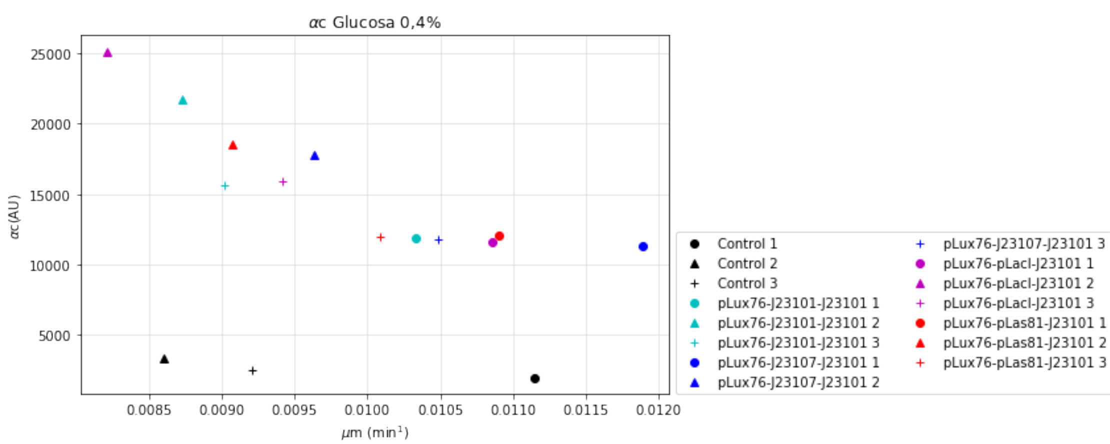


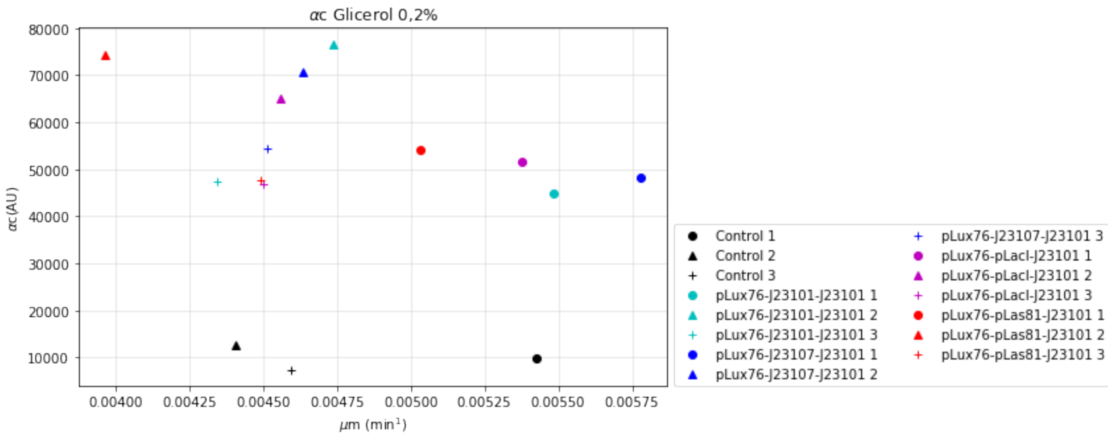



```
In [71]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Glucosa 0,4%')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$c(AU)')
         plt.plot(um1,slopec1,'ko',label='Control 1')
         plt.plot(um2,slopec2,'k^',label='Control 2')
         plt.plot(um3,slopec3,'k+',label='Control 3')
         plt.plot(um7,slopec7,'co',label='pLux76-J23101-J23101 1')
         plt.plot(um8,slopec8,'c^',label='pLux76-J23101-J23101 2')
         plt.plot(um9,slopec9,'c+',label='pLux76-J23101-J23101 3')
         plt.plot(um13,slopec13,'bo',label='pLux76-J23107-J23101 1')
         plt.plot(um14,slopec14,'b^',label='pLux76-J23107-J23101 2')
         plt.plot(um15,slopec15,'b+',label='pLux76-J23107-J23101 3')
         plt.plot(um19,slopec19,'mo',label='pLux76-pLacI-J23101 1')
         plt.plot(um20,slopec20,'m^',label='pLux76-pLacI-J23101 2')
         plt.plot(um21,slopec21,'m+',label='pLux76-pLacI-J23101 3')
         plt.plot(um25,slopec25,'ro',label='pLux76-pLas81-J23101 1')
         plt.plot(um26,slopec26,'r^',label='pLux76-pLas81-J23101 2')
         plt.plot(um27,slopec27,'r+',label='pLux76-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='pLux76-J23101-J23101 1')
plt.plot(um11,slopec11,'c^',label='pLux76-J23101-J23101 2')
plt.plot(um12,slopec12,'c+',label='pLux76-J23101-J23101 3')
plt.plot(um16,slopec16,'bo',label='pLux76-J23107-J23101 1')
plt.plot(um17,slopec17,'b^',label='pLux76-J23107-J23101 2')
plt.plot(um18,slopec18,'b+',label='pLux76-J23107-J23101 3')
plt.plot(um22,slopec22,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(um23,slopec23,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(um24,slopec24,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(um28,slopec28,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(um29,slopec29,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,slopec30,'r+',label='pLux76-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[71]: <matplotlib.legend.Legend at 0x1d722113be0>
```

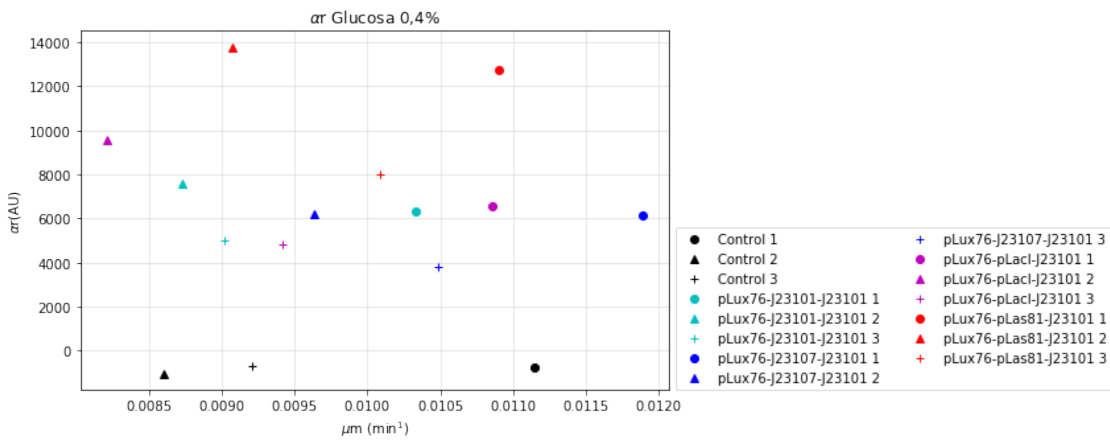


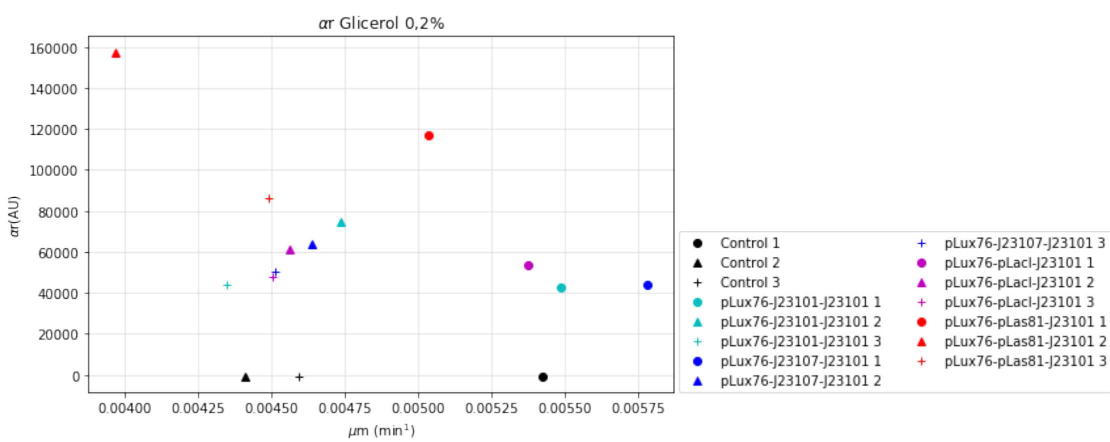


In [72]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'ko',label='Control 1')
plt.plot(um2,sloper2,'k^',label='Control 2')
plt.plot(um3,sloper3,'k+',label='Control 3')
plt.plot(um7,sloper7,'co',label='pLux76-J23101-J23101 1')
plt.plot(um8,sloper8,'c^',label='pLux76-J23101-J23101 2')
plt.plot(um9,sloper9,'c+',label='pLux76-J23101-J23101 3')
plt.plot(um13,sloper13,'bo',label='pLux76-J23107-J23101 1')
plt.plot(um14,sloper14,'b^',label='pLux76-J23107-J23101 2')
plt.plot(um15,sloper15,'b+',label='pLux76-J23107-J23101 3')
plt.plot(um19,sloper19,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(um20,sloper20,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(um21,sloper21,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(um25,sloper25,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(um26,sloper26,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(um27,sloper27,'r+',label='pLux76-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='pLux76-J23101-J23101 1')
```

```python
plt.plot(um11,sloper11,'c^',label='pLux76-J23101-J23101 2')
plt.plot(um12,sloper12,'c+',label='pLux76-J23101-J23101 3')
plt.plot(um16,sloper16,'bo',label='pLux76-J23107-J23101 1')
plt.plot(um17,sloper17,'b^',label='pLux76-J23107-J23101 2')
plt.plot(um18,sloper18,'b+',label='pLux76-J23107-J23101 3')
plt.plot(um22,sloper22,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(um23,sloper23,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(um24,sloper24,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(um28,sloper28,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(um29,sloper29,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,sloper30,'r+',label='pLux76-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

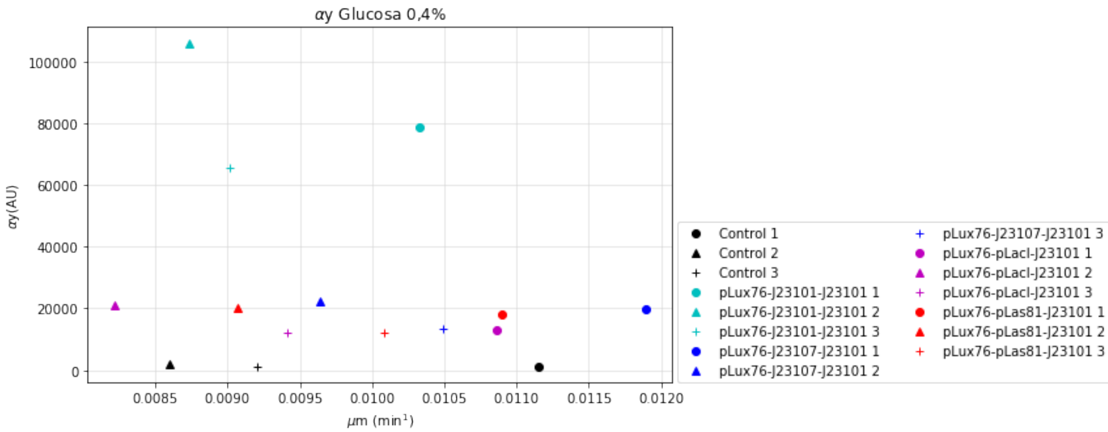
```
Out[72]: <matplotlib.legend.Legend at 0x1d72263fcc0>
```

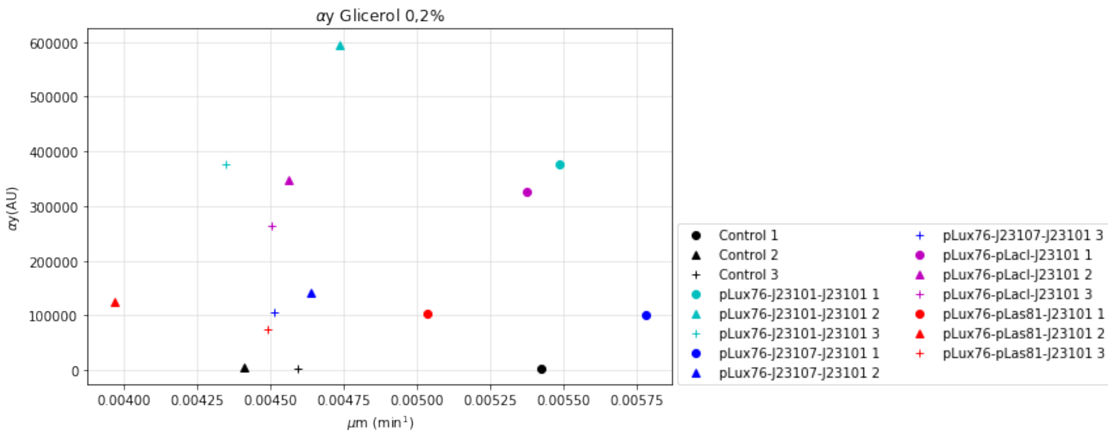




In [73]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'ko',label='Control 1')
plt.plot(um2,slopey2,'k^',label='Control 2')
plt.plot(um3,slopey3,'k+',label='Control 3')
plt.plot(um7,slopey7,'co',label='pLux76-J23101-J23101 1')
plt.plot(um8,slopey8,'c^',label='pLux76-J23101-J23101 2')
plt.plot(um9,slopey9,'c+',label='pLux76-J23101-J23101 3')
plt.plot(um13,slopey13,'bo',label='pLux76-J23107-J23101 1')
plt.plot(um14,slopey14,'b^',label='pLux76-J23107-J23101 2')
plt.plot(um15,slopey15,'b+',label='pLux76-J23107-J23101 3')
plt.plot(um19,slopey19,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(um20,slopey20,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(um21,slopey21,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(um25,slopey25,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(um26,slopey26,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(um27,slopey27,'r+',label='pLux76-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='pLux76-J23101-J23101 1')
plt.plot(um11,slopey11,'c^',label='pLux76-J23101-J23101 2')
plt.plot(um12,slopey12,'c+',label='pLux76-J23101-J23101 3')
plt.plot(um16,slopey16,'bo',label='pLux76-J23107-J23101 1')
plt.plot(um17,slopey17,'b^',label='pLux76-J23107-J23101 2')
plt.plot(um18,slopey18,'b+',label='pLux76-J23107-J23101 3')
plt.plot(um22,slopey22,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(um23,slopey23,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(um24,slopey24,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(um28,slopey28,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(um29,slopey29,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(um30,slopey30,'r+',label='pLux76-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 0x1d71e98f2b0>
```

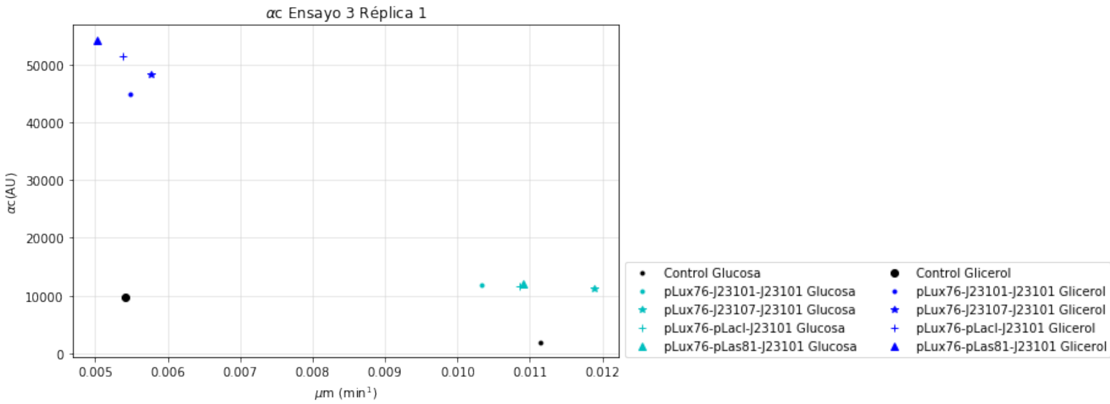


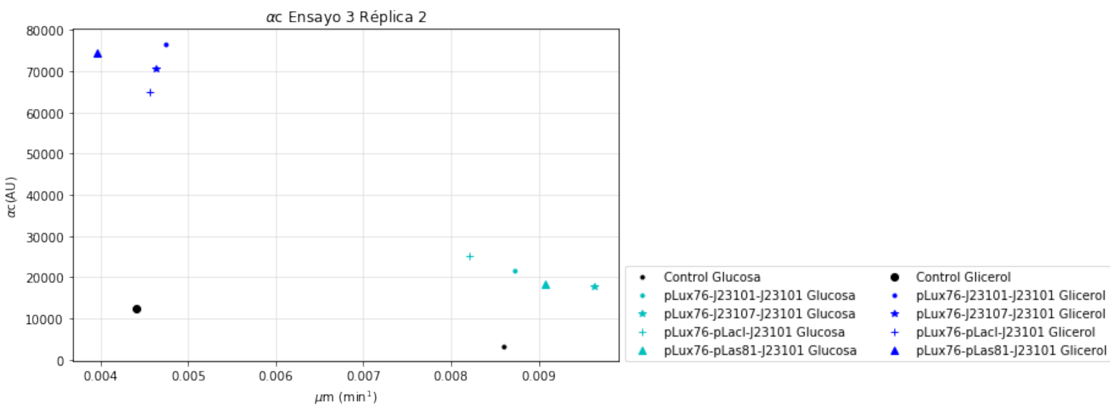


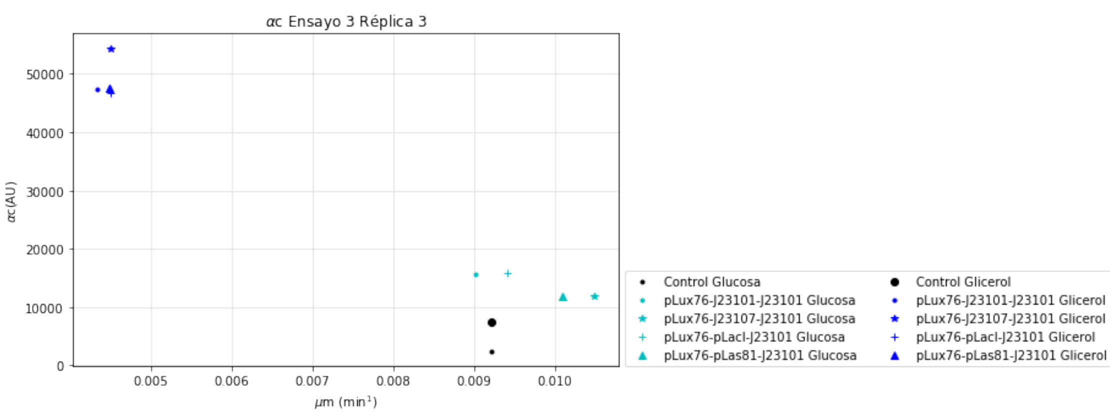
In [74]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 3 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um1,slopec1,'k.',label='Control Glucosa')
plt.plot(um7,slopec7,'c.',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um13,slopec13,'c*',label='pLux76-J23107-J23101 Glucosa')
plt.plot(um19,slopec19,'c+',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um25,slopec25,'c^',label='pLux76-pLas81-J23101 Glucosa')
plt.plot(um4,slopec4,'ko',label='Control Glicerol')
plt.plot(um10,slopec10,'b.',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um16,slopec16,'b*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um22,slopec22,'b+',label='pLux76-pLacI-J23101 Glicerol')
plt.plot(um28,slopec28,'b^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
    plt.plot(um14,slopec14,'c*',label='pLux76-J23107-J23101 Glucosa')
    plt.plot(um20,slopec20,'c+',label='pLux76-pLacI-J23101 Glucosa')
    plt.plot(um26,slopec26,'c^',label='pLux76-pLas81-J23101 Glucosa')
    plt.plot(um5,slopec5,'ko',label='Control Glicerol')
    plt.plot(um11,slopec11,'b.',label='pLux76-J23101-J23101 Glicerol')
    plt.plot(um17,slopec17,'b*',label='pLux76-J23107-J23101 Glicerol')
    plt.plot(um23,slopec23,'b+',label='pLux76-pLacI-J23101 Glicerol')
    plt.plot(um29,slopec29,'b^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
    plt.plot(um15,slopec15,'c*',label='pLux76-J23107-J23101 Glucosa')
    plt.plot(um21,slopec21,'c+',label='pLux76-pLacI-J23101 Glucosa')
    plt.plot(um27,slopec27,'c^',label='pLux76-pLas81-J23101 Glucosa')
    plt.plot(um3,slopec6,'ko',label='Control Glicerol')
    plt.plot(um12,slopec12,'b.',label='pLux76-J23101-J23101 Glicerol')
    plt.plot(um18,slopec18,'b*',label='pLux76-J23107-J23101 Glicerol')
    plt.plot(um24,slopec24,'b+',label='pLux76-pLacI-J23101 Glicerol')
    plt.plot(um30,slopec30,'b^',label='pLux76-pLas81-J23101 Glicerol')
    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 0x1d721294978>
```





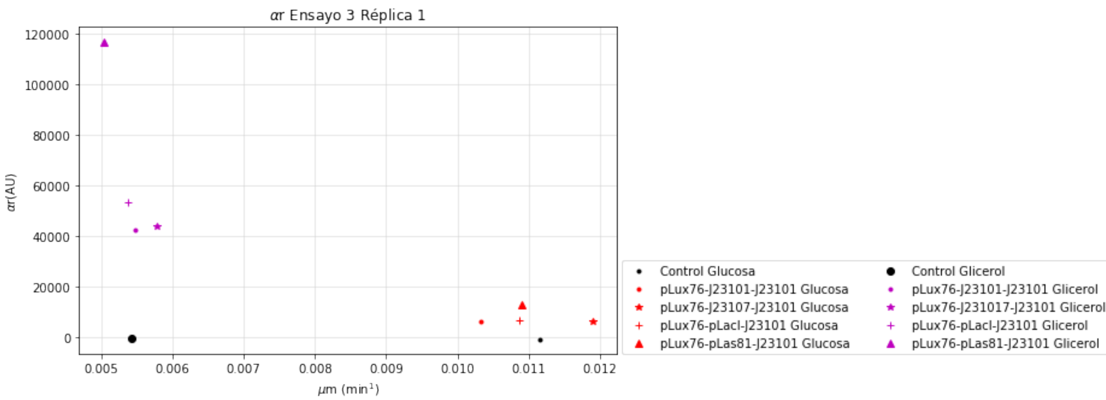


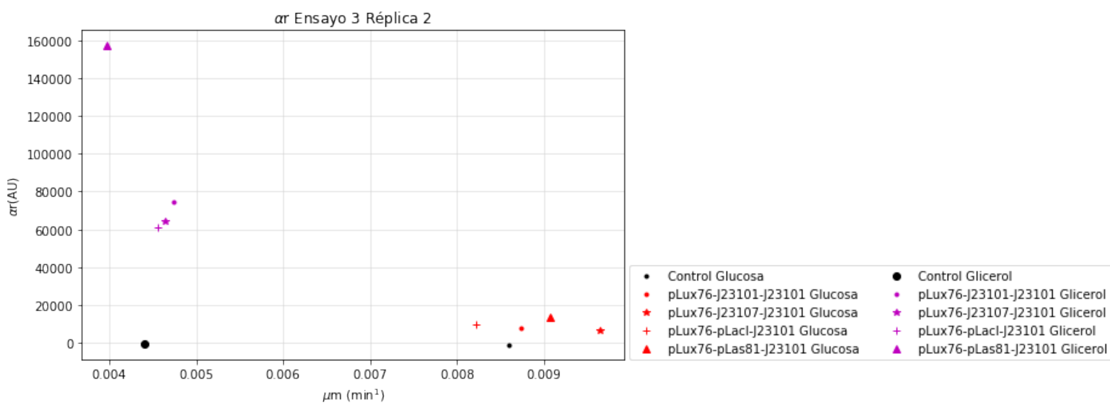
In [75]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
```

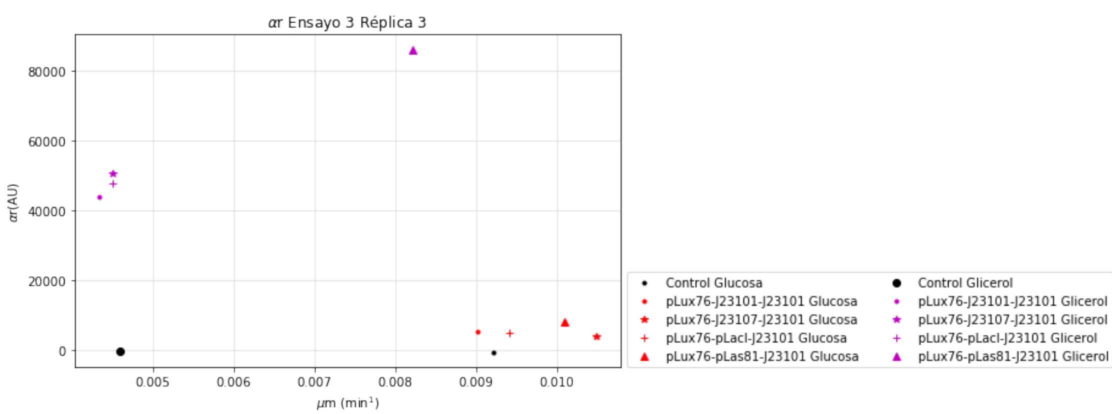
```python
plt.title(r'$\alpha$r Ensayo 3 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'k.',label='Control Glucosa ')
plt.plot(um7,sloper7,'r.',label='pLux76-J23101-J23101 Glucosa ')
plt.plot(um13,sloper13,'r*',label='pLux76-J23107-J23101 Glucosa ')
plt.plot(um19,sloper19,'r+',label='pLux76-pLacI-J23101 Glucosa ')
plt.plot(um25,sloper25,'r^',label='pLux76-pLas81-J23101 Glucosa ')
plt.plot(um4,sloper4,'ko',label='Control Glicerol ')
plt.plot(um10,sloper10,'m.',label='pLux76-J23101-J23101 Glicerol ')
plt.plot(um16,sloper16,'m*',label='pLux76-J231017-J23101 Glicerol')
plt.plot(um22,sloper22,'m+',label='pLux76-pLacI-J23101 Glicerol ')
plt.plot(um28,sloper28,'m^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
plt.plot(um14,sloper14,'r*',label='pLux76-J23107-J23101 Glucosa')
plt.plot(um20,sloper20,'r+',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um26,sloper26,'r^',label='pLux76-pLas81-J23101 Glucosa')
plt.plot(um5,sloper5,'ko',label='Control Glicerol')
plt.plot(um11,sloper11,'m.',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um17,sloper17,'m*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um23,sloper23,'m+',label='pLux76-pLacI-J23101 Glicerol')
plt.plot(um29,sloper29,'m^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
plt.plot(um15,sloper15,'r*',label='pLux76-J23107-J23101 Glucosa')
plt.plot(um21,sloper21,'r+',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um27,sloper27,'r^',label='pLux76-pLas81-J23101 Glucosa')
plt.plot(um6,sloper6,'ko',label='Control Glicerol')
plt.plot(um12,sloper12,'m.',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um18,sloper18,'m*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um24,sloper24,'m+',label='pLux76-pLacI-J23101 Glicerol')
plt.plot(um20,sloper30,'m^',label='pLux76-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 0x1d723fbea58>



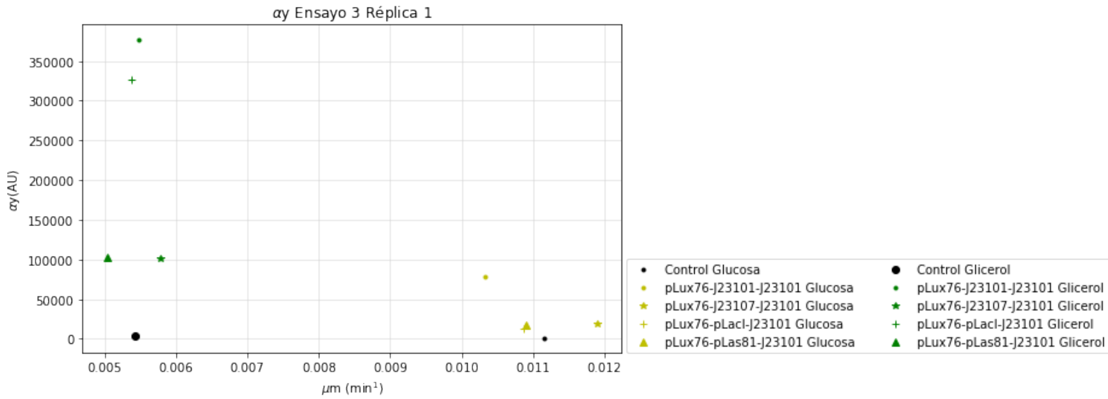


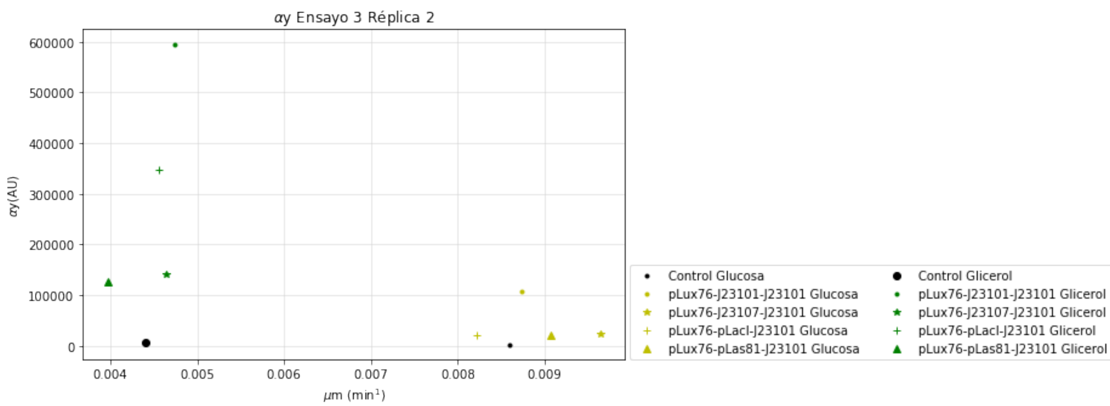


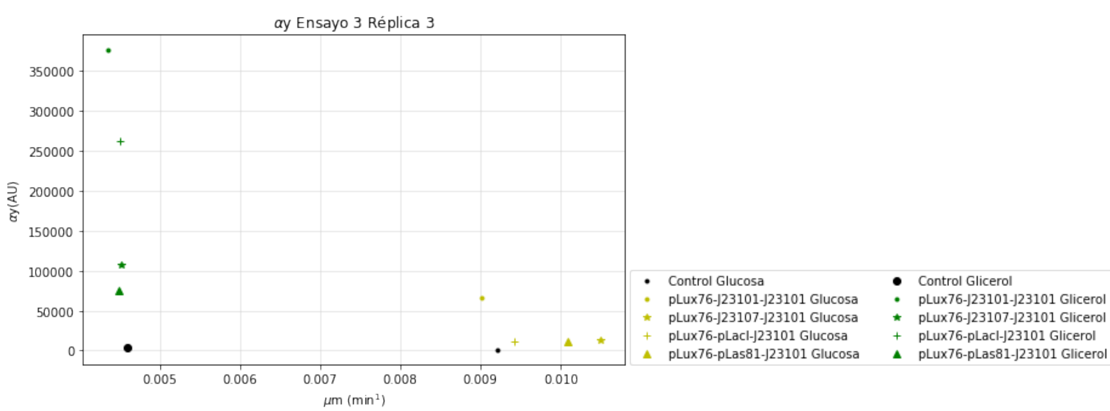
In [76]:
```python
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 3 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'k.',label='Control Glucosa ')
plt.plot(um7,slopey7,'y.',label='pLux76-J23101-J23101 Glucosa ')
plt.plot(um13,slopey13,'y*',label='pLux76-J23107-J23101 Glucosa ')
plt.plot(um19,slopey19,'y+',label='pLux76-pLacI-J23101 Glucosa ')
plt.plot(um25,slopey25,'y^',label='pLux76-pLas81-J23101 Glucosa ')
plt.plot(um4,slopey4,'ko',label='Control Glicerol ')
plt.plot(um10,slopey10,'g.',label='pLux76-J23101-J23101 Glicerol ')
plt.plot(um16,slopey16,'g*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um22,slopey22,'g+',label='pLux76-pLacI-J23101 Glicerol ')
plt.plot(um28,slopey28,'g^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
plt.plot(um14,slopey14,'y*',label='pLux76-J23107-J23101 Glucosa')
plt.plot(um20,slopey20,'y+',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um26,slopey26,'y^',label='pLux76-pLas81-J23101 Glucosa')
plt.plot(um5,slopey5,'ko',label='Control Glicerol')
plt.plot(um11,slopey11,'g.',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um17,slopey17,'g*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um23,slopey23,'g+',label='pLux76-pLacI-J23101 Glicerol')
plt.plot(um29,slopey29,'g^',label='pLux76-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 3 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='pLux76-J23101-J23101 Glucosa')
plt.plot(um15,slopey15,'y*',label='pLux76-J23107-J23101 Glucosa')
plt.plot(um21,slopey21,'y+',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um27,slopey27,'y^',label='pLux76-pLas81-J23101 Glucosa')
plt.plot(um6,slopey6,'ko',label='Control Glicerol')
plt.plot(um12,slopey12,'g.',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um18,slopey18,'g*',label='pLux76-J23107-J23101 Glicerol')
plt.plot(um24,slopey24,'g+',label='pLux76-pLacI-J23101 Glicerol')
```

```python
plt.plot(um30,slopey30,'g^',label='pLux76-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 0x1d7246f18d0>

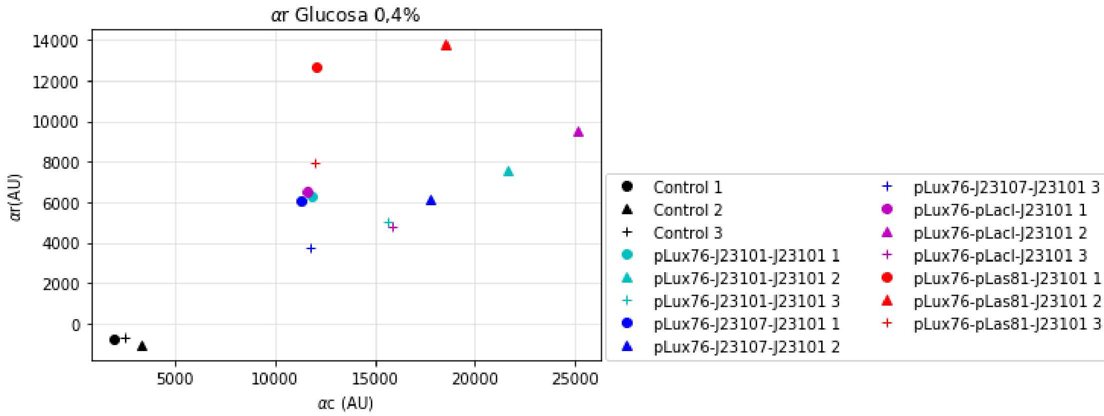


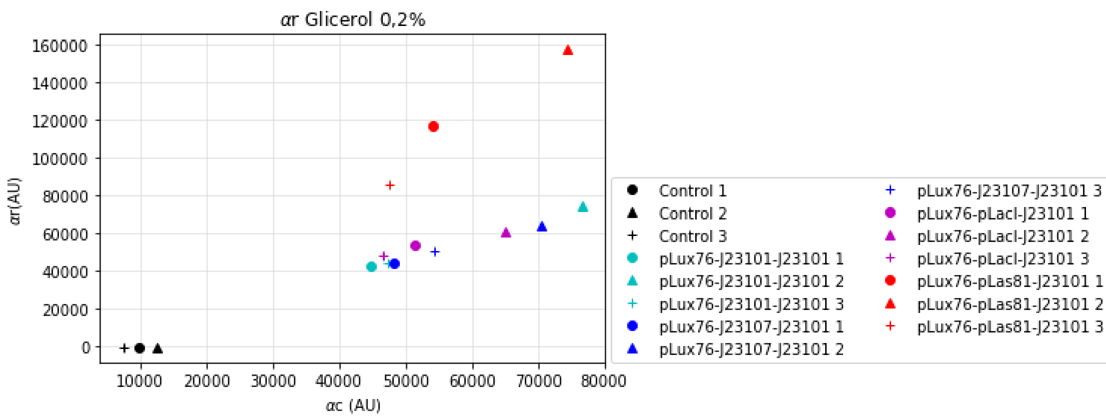




In [77]:
```
#grafico de ac versus Um
plt.figure()
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(slopec1,sloper1,'ko',label='Control 1')
plt.plot(slopec2,sloper2,'k^',label='Control 2')
plt.plot(slopec3,sloper3,'k+',label='Control 3')
plt.plot(slopec7,sloper7,'co',label='pLux76-J23101-J23101 1')
plt.plot(slopec8,sloper8,'c^',label='pLux76-J23101-J23101 2')
plt.plot(slopec9,sloper9,'c+',label='pLux76-J23101-J23101 3')
plt.plot(slopec13,sloper13,'bo',label='pLux76-J23107-J23101 1')
plt.plot(slopec14,sloper14,'b^',label='pLux76-J23107-J23101 2')
plt.plot(slopec15,sloper15,'b+',label='pLux76-J23107-J23101 3')
plt.plot(slopec19,sloper19,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(slopec20,sloper20,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec21,sloper21,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(slopec25,sloper25,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,sloper26,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec27,sloper27,'r+',label='pLux76-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='pLux76-J23101-J23101 1')
plt.plot(slopec11,sloper11,'c^',label='pLux76-J23101-J23101 2')
plt.plot(slopec12,sloper12,'c+',label='pLux76-J23101-J23101 3')
plt.plot(slopec16,sloper16,'bo',label='pLux76-J23107-J23101 1')
plt.plot(slopec17,sloper17,'b^',label='pLux76-J23107-J23101 2')
plt.plot(slopec18,sloper18,'b+',label='pLux76-J23107-J23101 3')
plt.plot(slopec22,sloper22,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(slopec23,sloper23,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec24,sloper24,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(slopec28,sloper28,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(slopec29,sloper29,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec30,sloper30,'r+',label='pLux76-pLas81-J23101 3')
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 0x1d721378828>



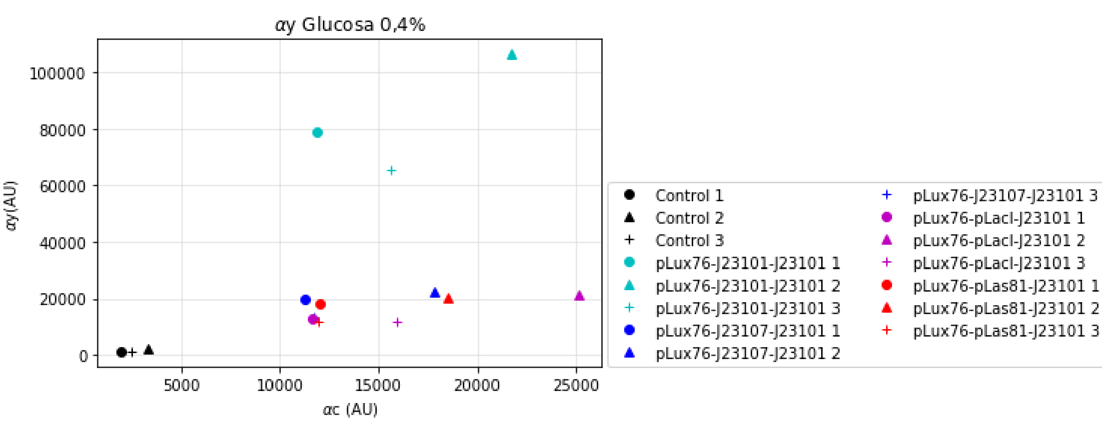


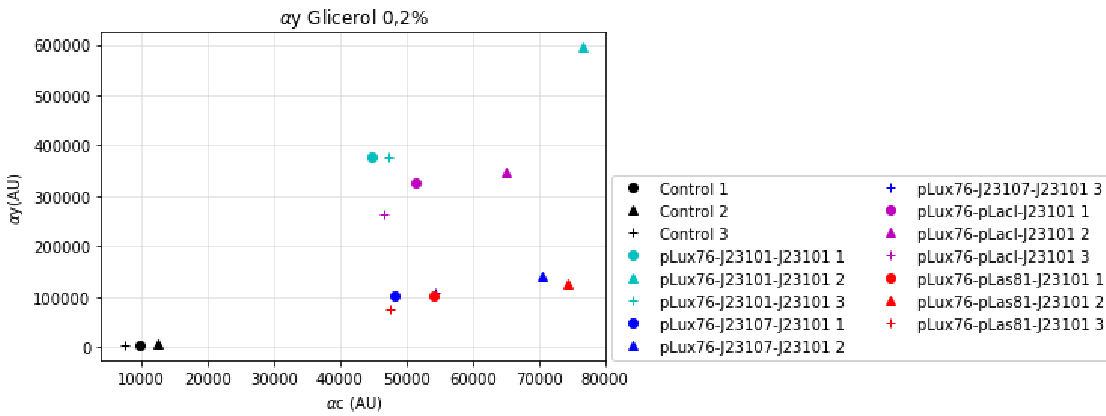
In [78]:

```python
#grafico de ar vs ac
plt.figure()
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='pLux76-J23101-J23101 1')
plt.plot(slopec8,slopey8,'c^',label='pLux76-J23101-J23101 2')
plt.plot(slopec9,slopey9,'c+',label='pLux76-J23101-J23101 3')
plt.plot(slopec13,slopey13,'bo',label='pLux76-J23107-J23101 1')
plt.plot(slopec14,slopey14,'b^',label='pLux76-J23107-J23101 2')
plt.plot(slopec15,slopey15,'b+',label='pLux76-J23107-J23101 3')
plt.plot(slopec19,slopey19,'mo',label='pLux76-pLacI-J23101 1')
```

```
plt.plot(slopec20,slopey20,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec21,slopey21,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(slopec25,slopey25,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(slopec26,slopey26,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec27,slopey27,'r+',label='pLux76-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()
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='pLux76-J23101-J23101 1')
plt.plot(slopec11,slopey11,'c^',label='pLux76-J23101-J23101 2')
plt.plot(slopec12,slopey12,'c+',label='pLux76-J23101-J23101 3')
plt.plot(slopec16,slopey16,'bo',label='pLux76-J23107-J23101 1')
plt.plot(slopec17,slopey17,'b^',label='pLux76-J23107-J23101 2')
plt.plot(slopec18,slopey18,'b+',label='pLux76-J23107-J23101 3')
plt.plot(slopec22,slopey22,'mo',label='pLux76-pLacI-J23101 1')
plt.plot(slopec23,slopey23,'m^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec24,slopey24,'m+',label='pLux76-pLacI-J23101 3')
plt.plot(slopec28,slopey28,'ro',label='pLux76-pLas81-J23101 1')
plt.plot(slopec29,slopey29,'r^',label='pLux76-pLas81-J23101 2')
plt.plot(slopec30,slopey30,'r+',label='pLux76-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

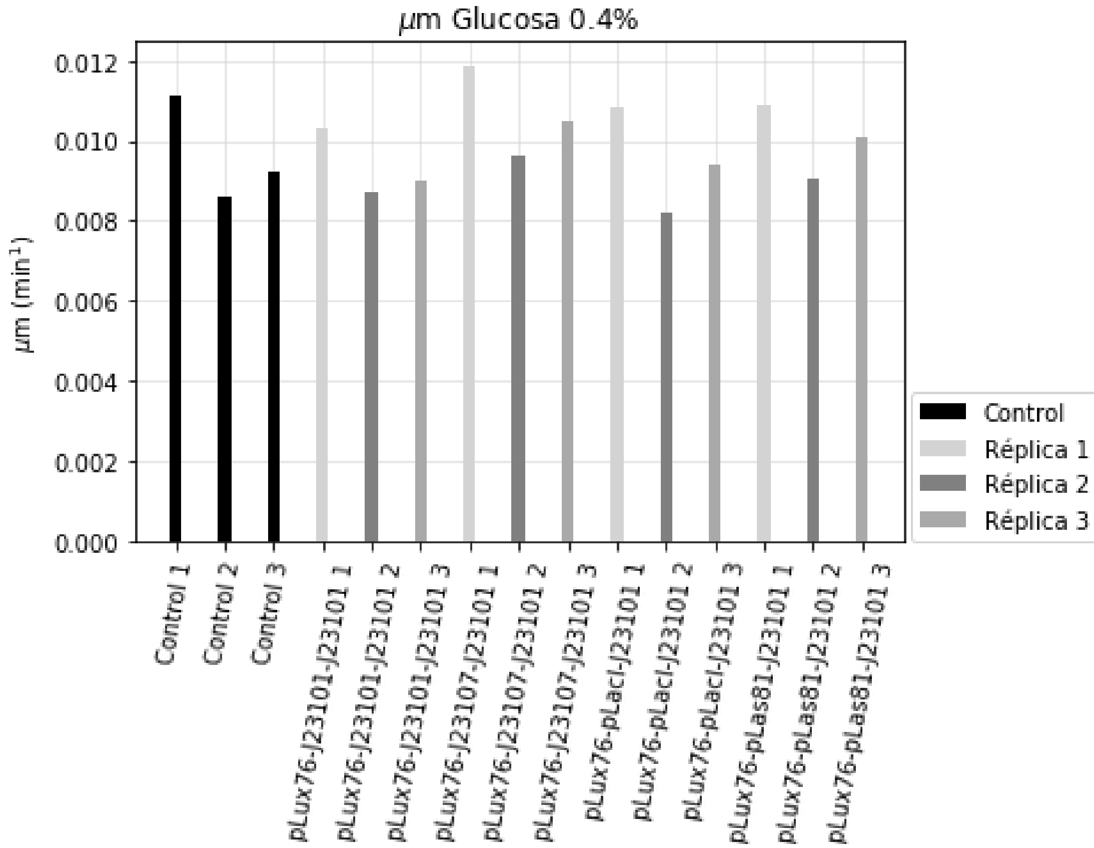
```
Out[78]: <matplotlib.legend.Legend at 0x1d72258ccc0>
```





In [79]:
```
#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','pLux76-J23101-J23101 1','pLux76-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

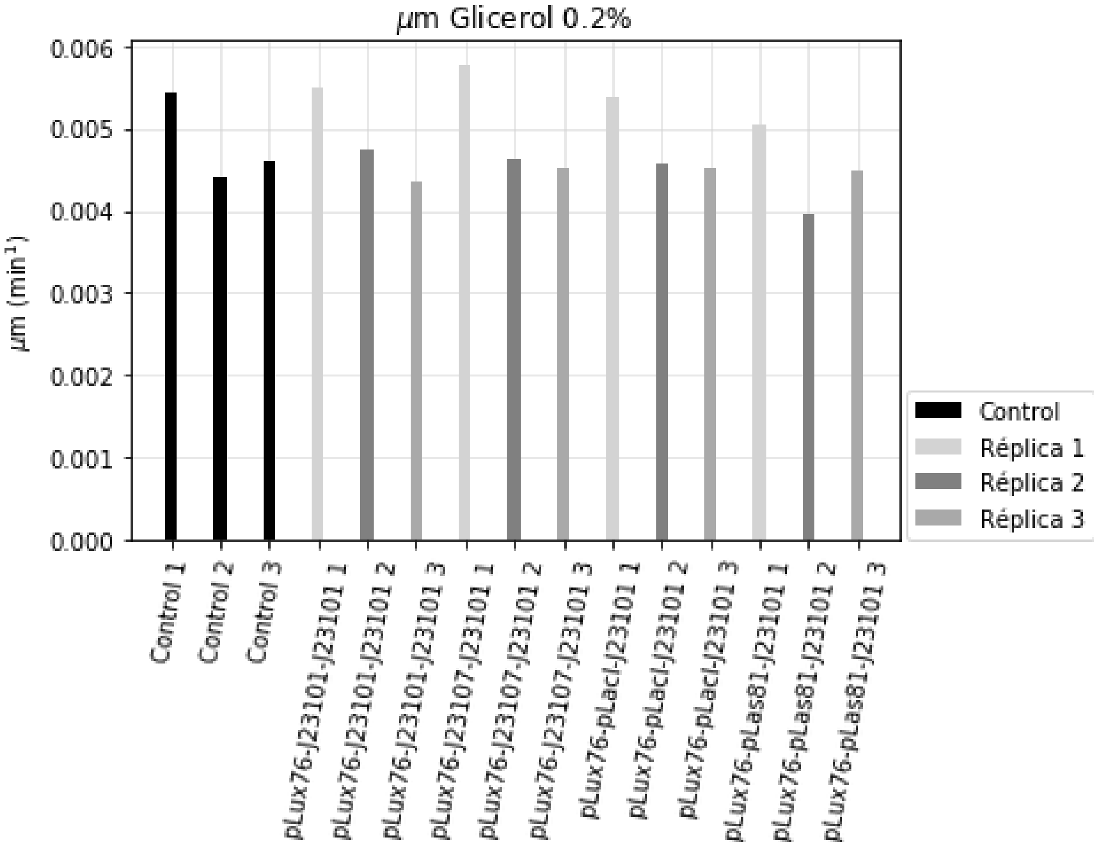
Out[79]: <matplotlib.legend.Legend at 0x1d721308828>



In [80]:
```
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','pLux76-J23101-J23101 1','pLux76-J231
```

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

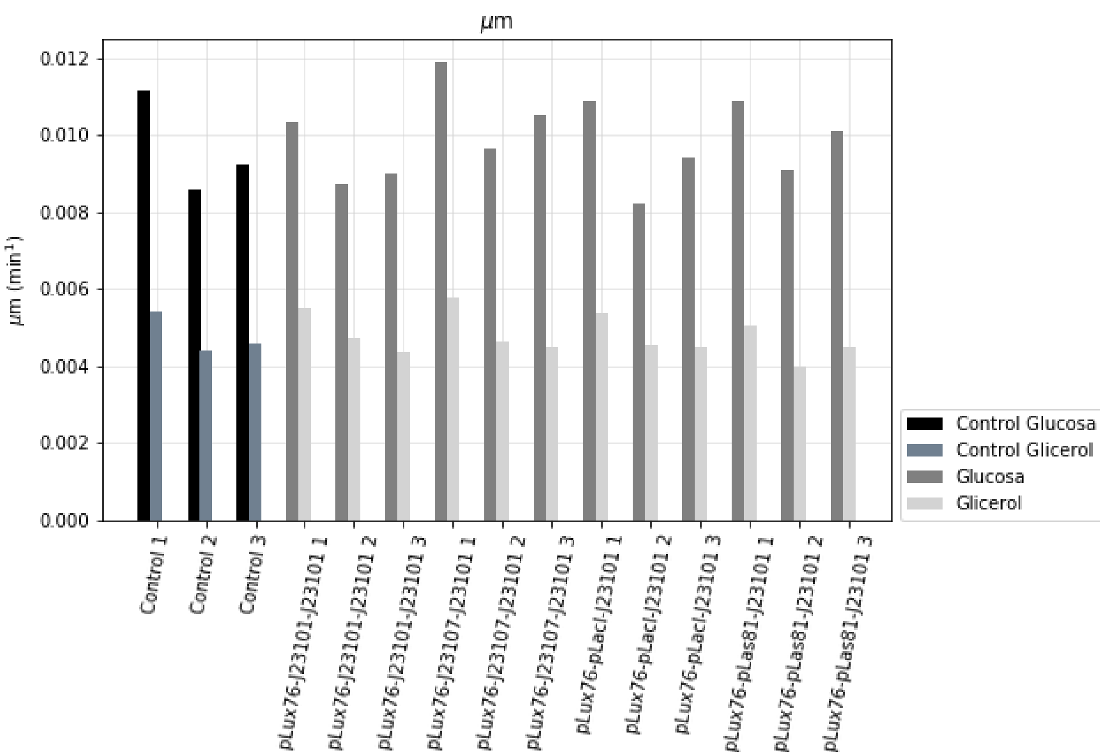
Out[80]: <matplotlib.legend.Legend at 0x1d721e294a8>



In [140]:
```
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','pLux76-J23101-J23101 1','pLux76-J23
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

```
Out[140]: <matplotlib.legend.Legend at 0x1d72c2d5a90>
```



In [82]:
```
#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,pr27
ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
```

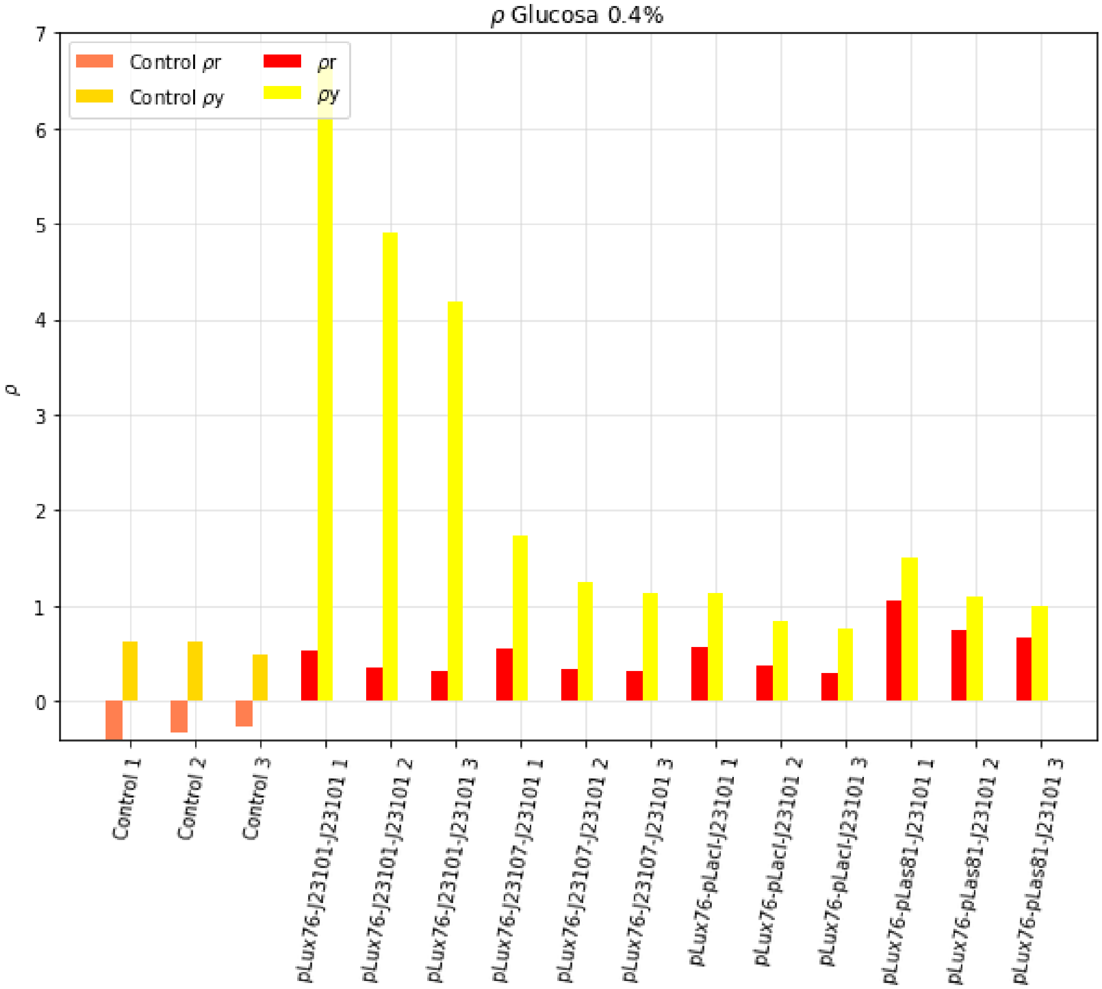
In [83]:
```
#Ro YFP
py1=slopey1/slopec1
py2=slopey2/slopec2
py3=slopey3/slopec3
py4=slopey4/slopec4
py5=slopey5/slopec5
py6=slopey6/slopec6
py7=slopey7/slopec7
py8=slopey8/slopec8
py9=slopey9/slopec9
py10=slopey10/slopec10
```

```
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 [84]: #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','pLux76-J23101-J23101 1','pLux76-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

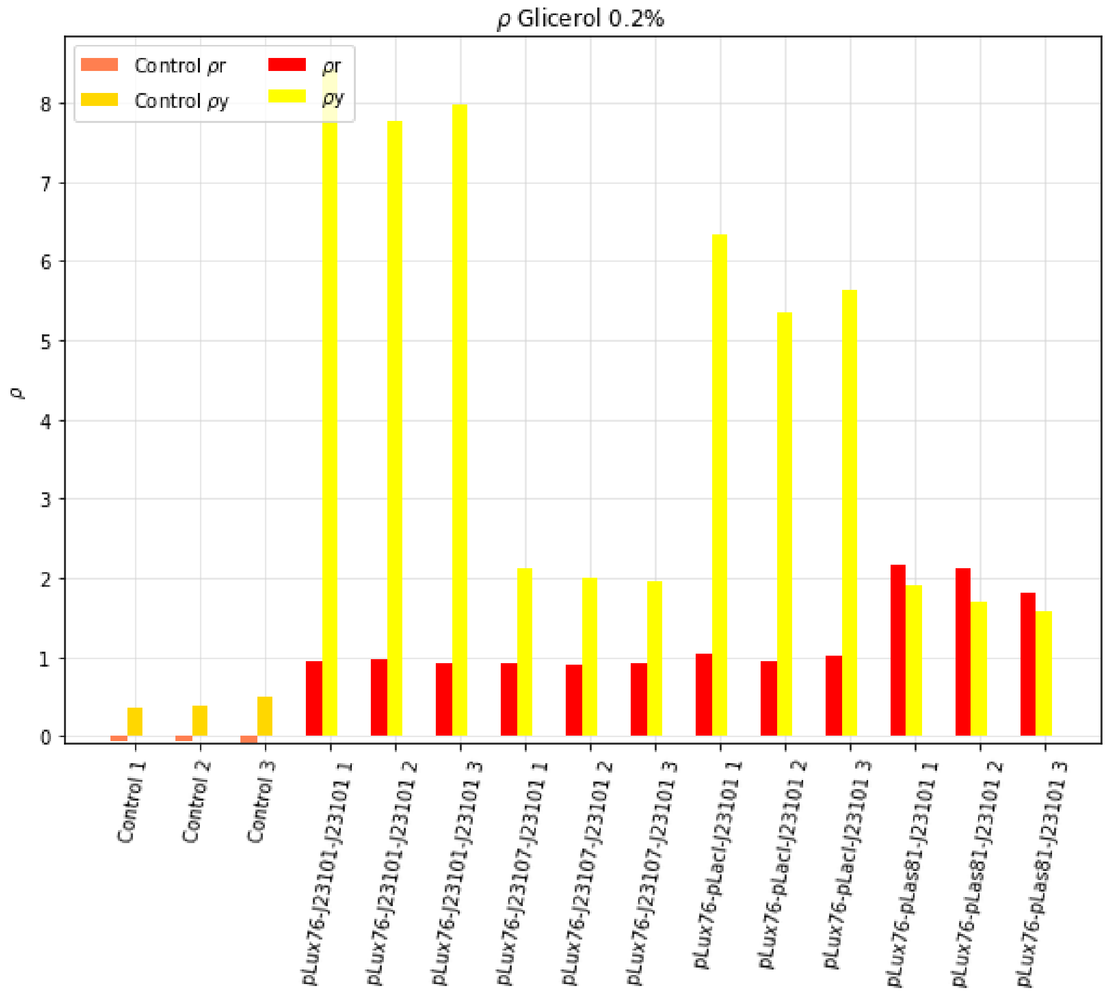
```
Out[84]: <matplotlib.legend.Legend at 0x1d722b06be0>
```




```
In [85]: 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','pLux76-J23101-J23101 1','pLux76-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

```
Out[85]: <matplotlib.legend.Legend at 0x1d723c90860>
```

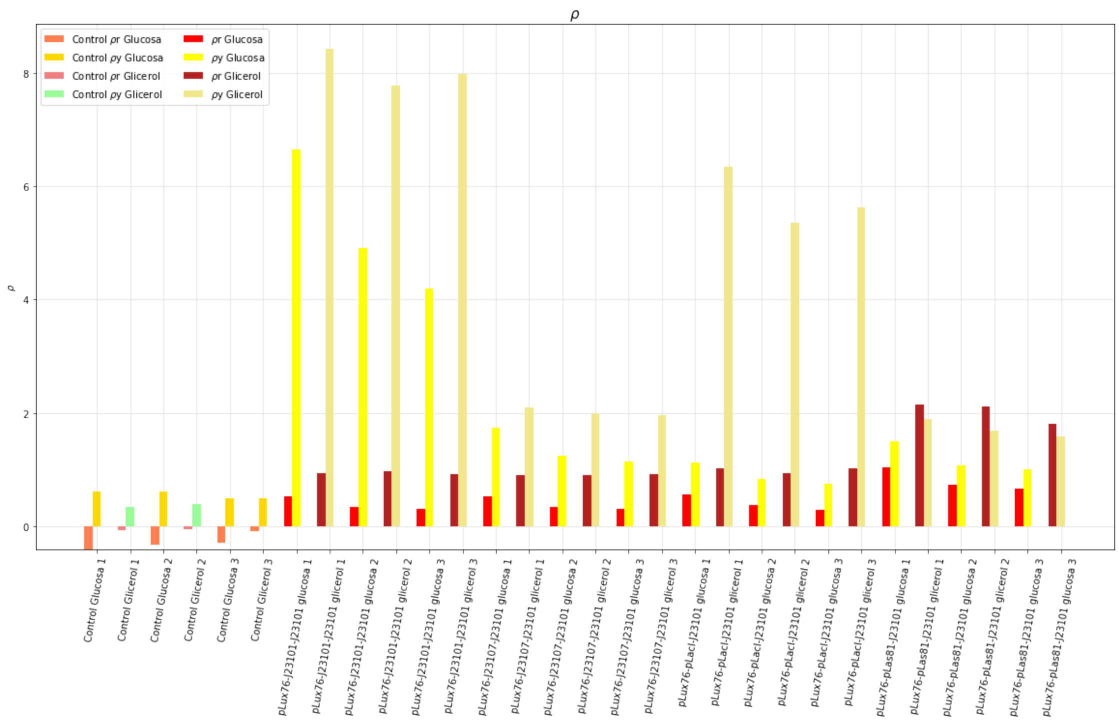


In [138]:
```
X = np.arange(30)
plt.figure(figsize=(20,10))
plt.title(r'$\rho$',fontsize=15.0)
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r Gl
plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y Glu
plt.bar(X[1]-0.25,ro_rfp[3],color='lightcoral',width=0.25,label= 'Control'+' '+r'$\rho
plt.bar(X[1]+0.00,ro_yfp[3],color='palegreen',width=0.25,label= 'Control'+' '+r'$\rho$
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='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[5],color='gold',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=
plt.bar(X[7]-0.25,ro_rfp[9],color='firebrick',width=0.25,label=r'$\rho$r Glicerol',zor
plt.bar(X[7]+0.00,ro_yfp[9],color='khaki',width=0.25,label=r'$\rho$y Glicerol',zorder=
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)
```

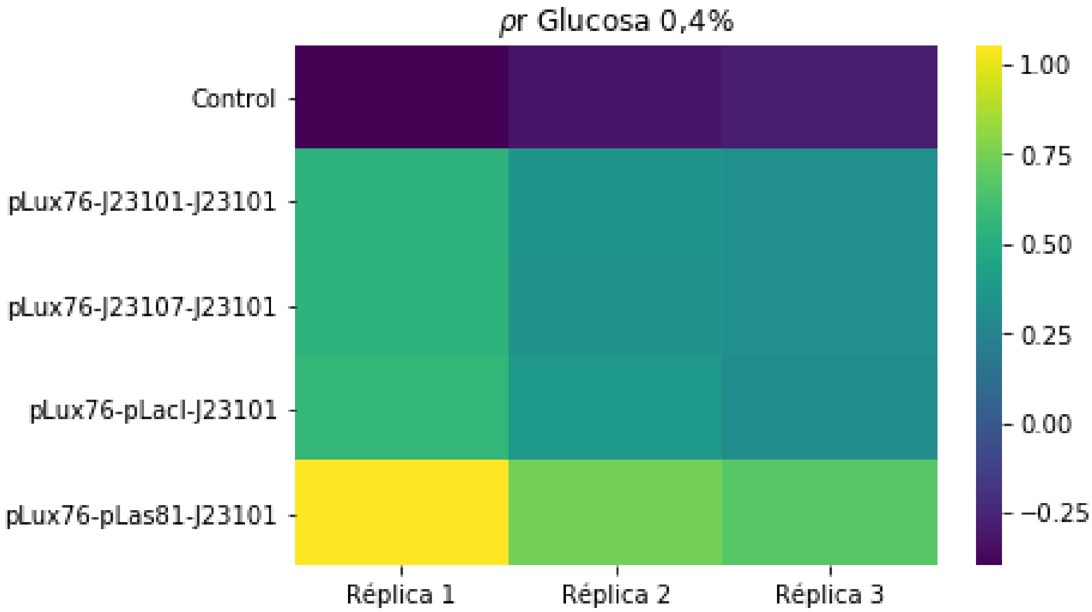
```
plt.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control G
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

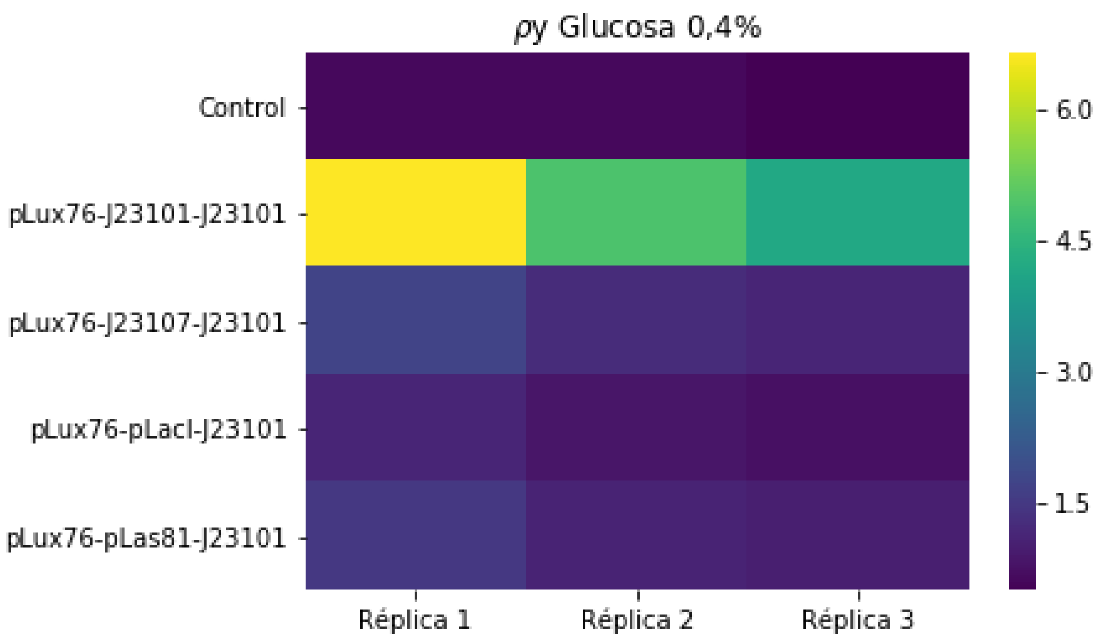
Out[138]: <matplotlib.legend.Legend at 0x1d72b8cb908>



In [87]:
```
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','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

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[87]: <matplotlib.axes._subplots.AxesSubplot at 0x1d722077898>

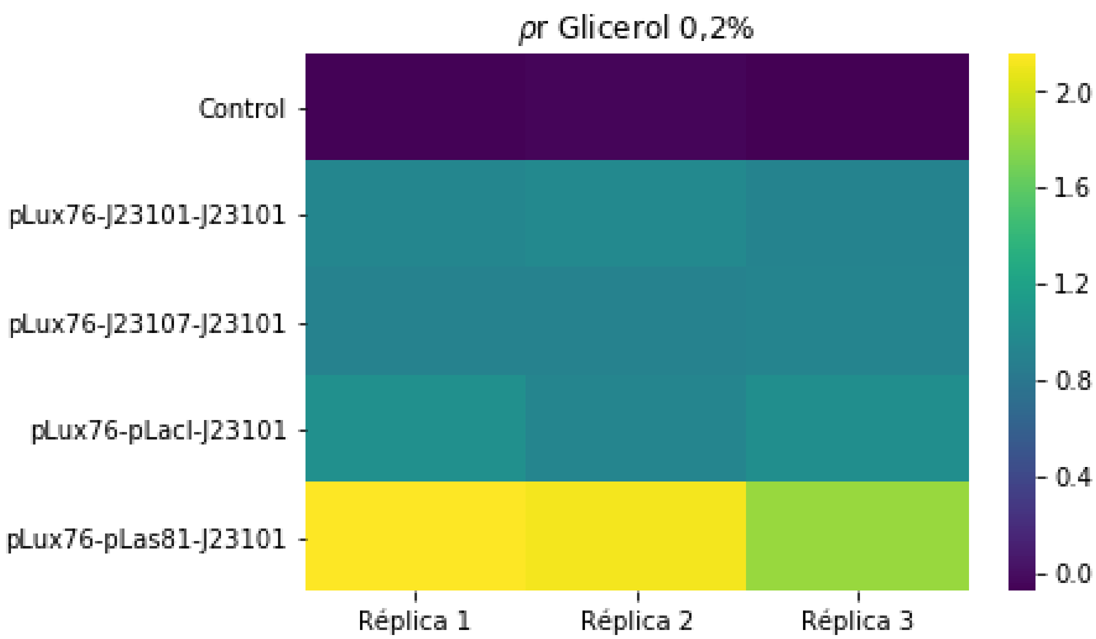


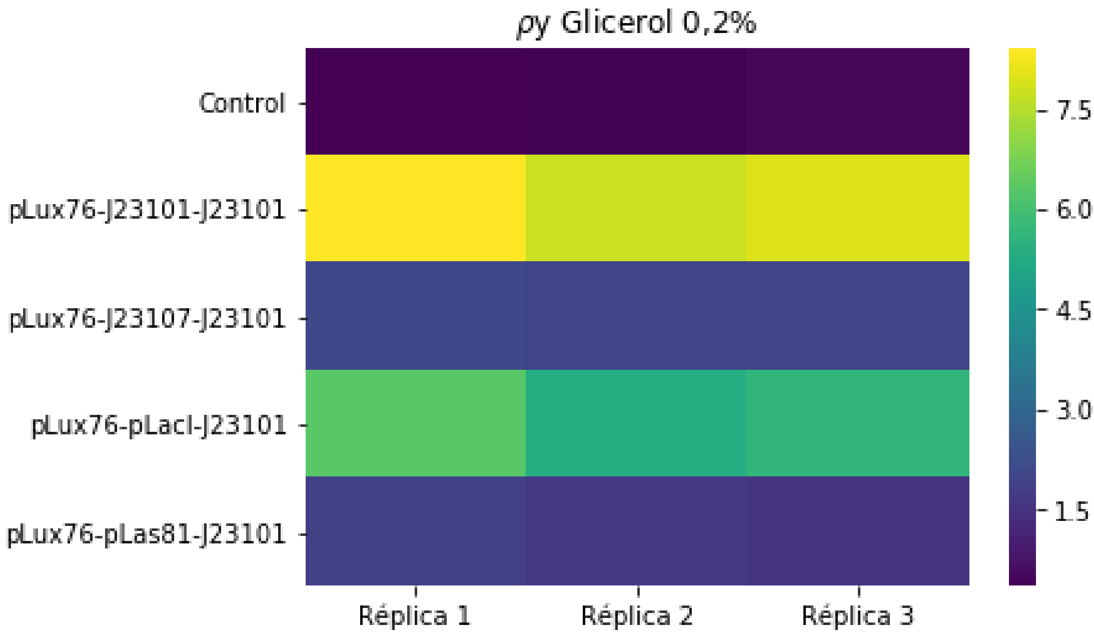


In [88]:
```
ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,

xlabel=['Réplica 1','Réplica 2','Réplica 3']
```

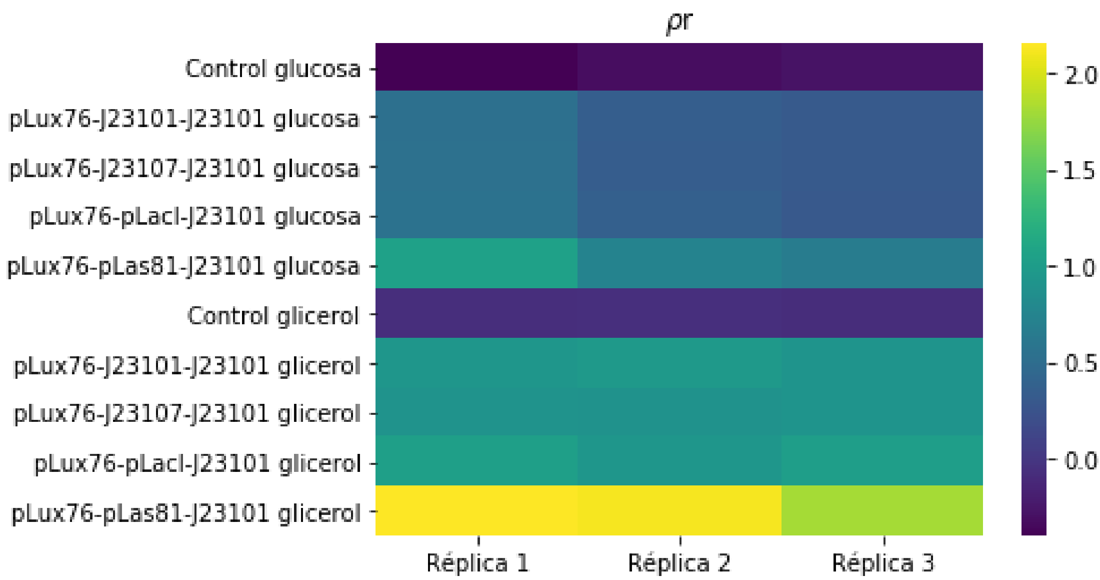
```
ylabel=['Control','pLux76-J23101-J23101','pLux76-J23107-J23101','pLux76-pLacI-J23101','

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)
```

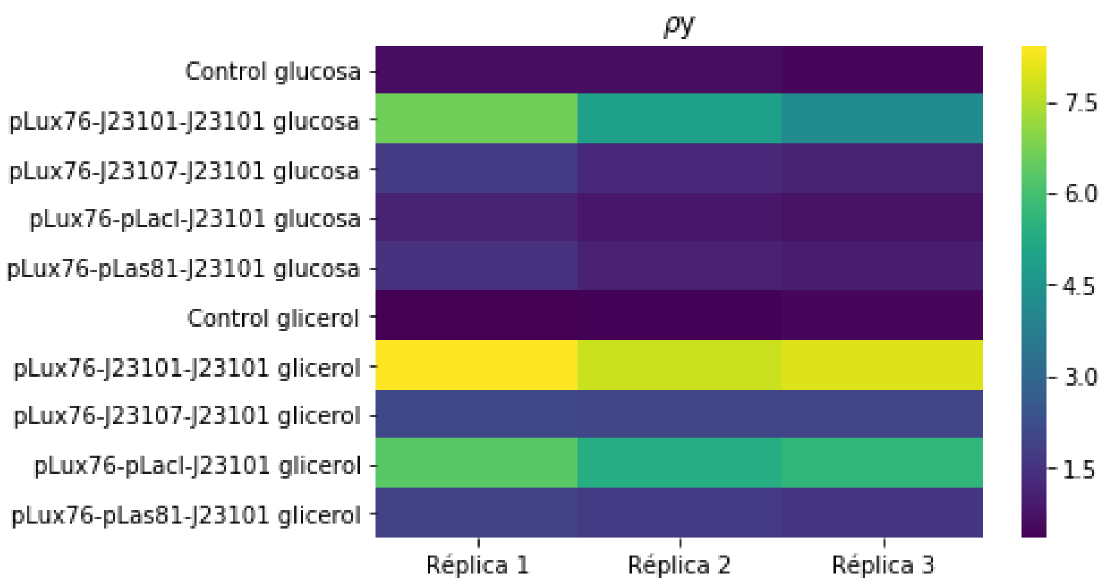
Out[88]: <matplotlib.axes._subplots.AxesSubplot at 0x1d724630d68>






```
In [89]: 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','pLux76-J23101-J23101 glucosa','pLux76-J23107-J23101 glucosa'

         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[89]: <matplotlib.axes._subplots.AxesSubplot at 0x1d7245949b0>
```

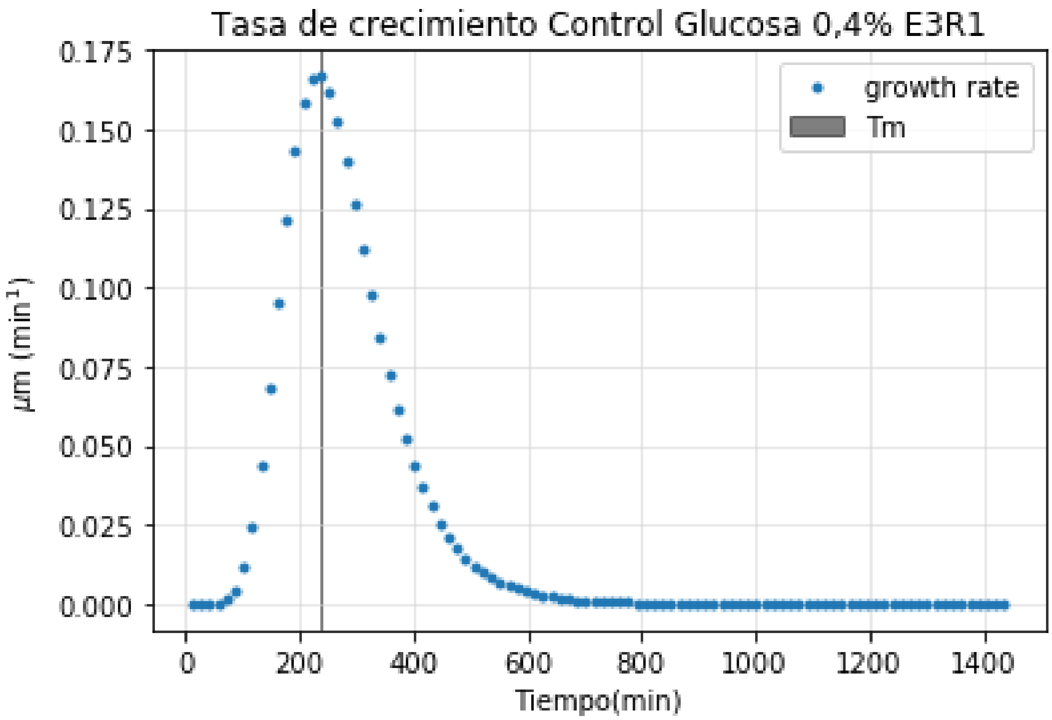





```
In [90]: #tasa de crecimiento
         ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(l1-tt))+1))))
         #Con diff
         dy1=(np.diff(ye1))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R1')
         plt.xlabel('Tiempo(min)')
```

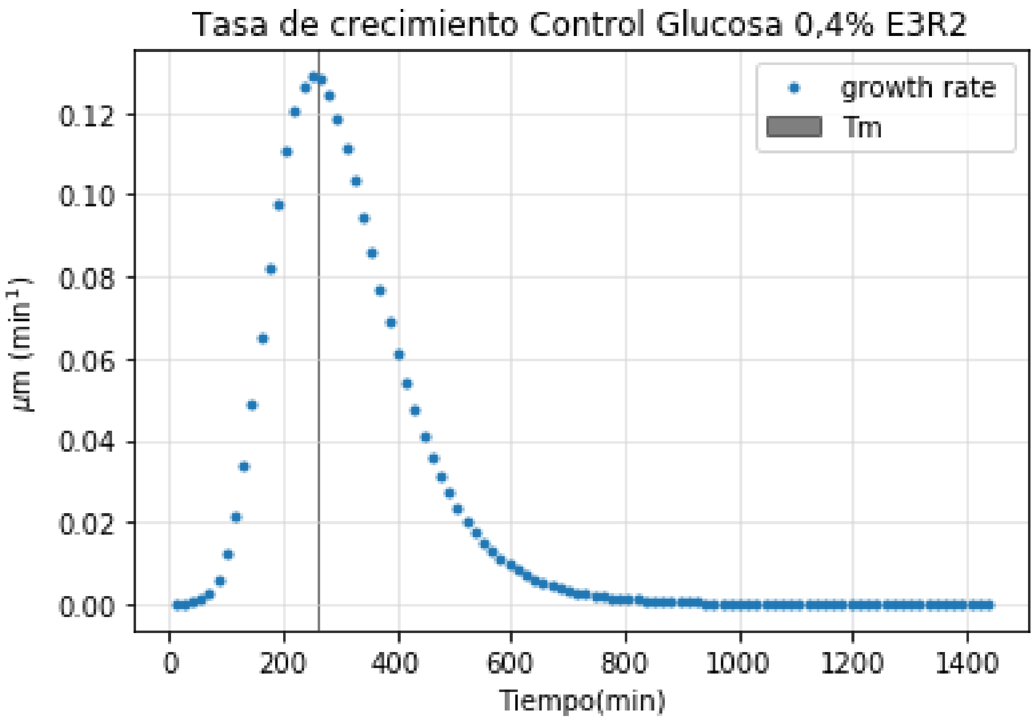
```
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy1,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[90]: <matplotlib.legend.Legend at 0x1d724375d30>

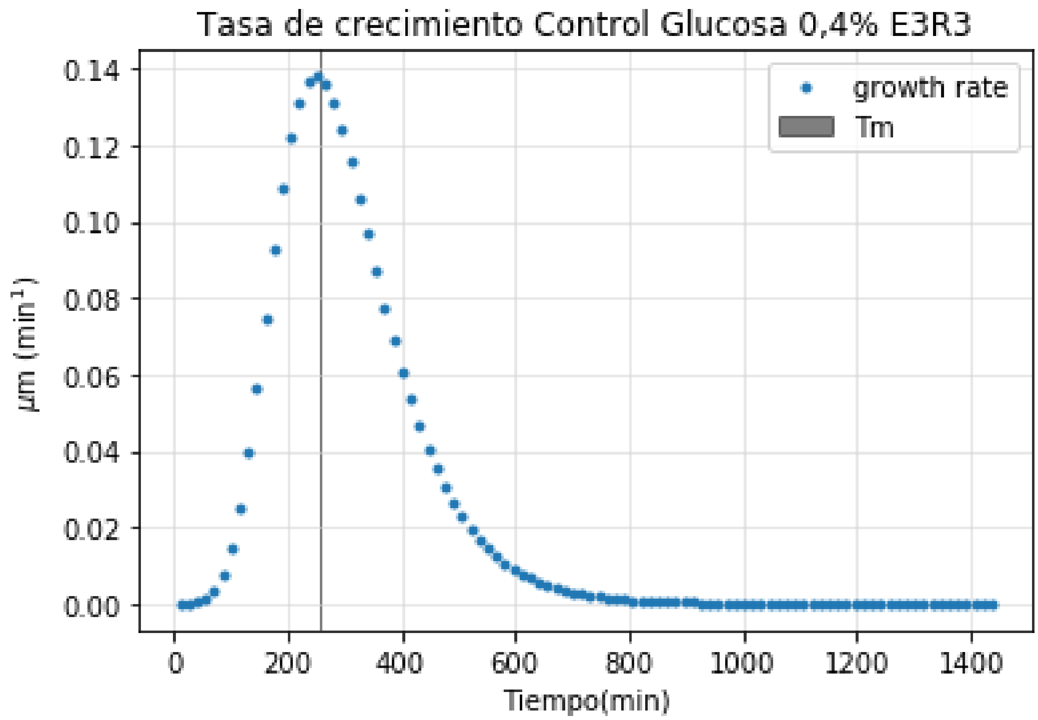


In [91]:
```
#tasa de crecimiento

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% E3R2')
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')
```

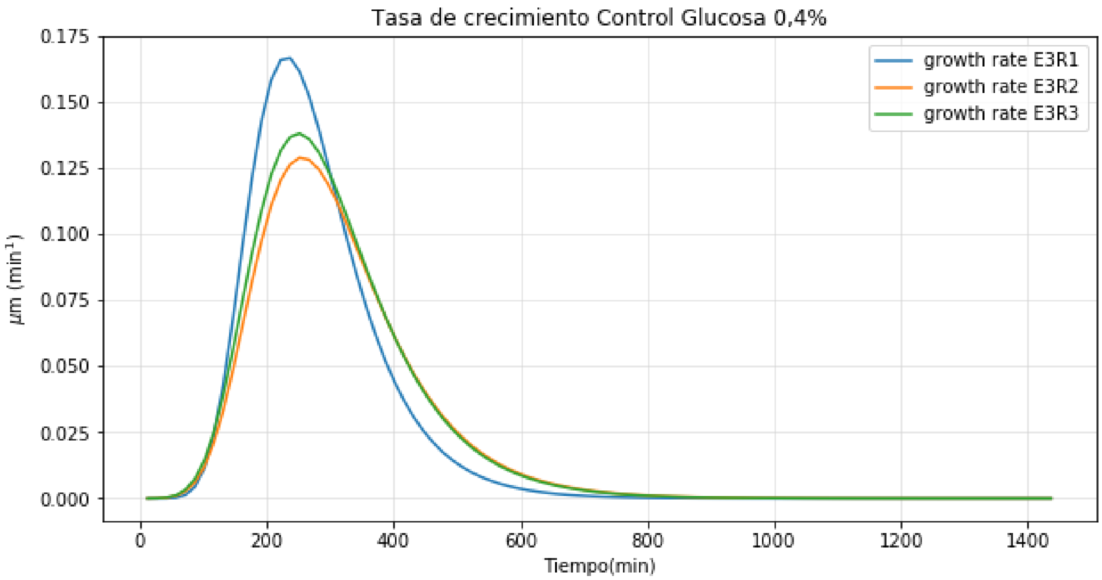
Out[91]: <matplotlib.legend.Legend at 0x1d72488bb70>



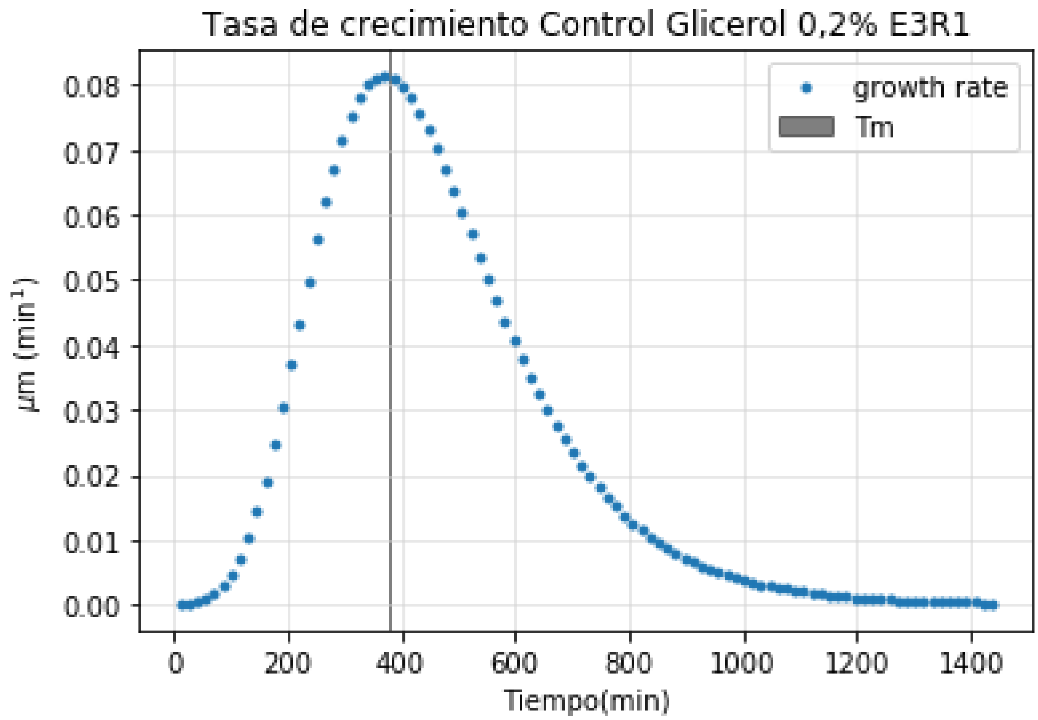

```
In [92]: #tasa de crecimiento
         ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(l3-tt))+1))))
         #Con diff
         dy3=(np.diff(ye3))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[92]: <matplotlib.legend.Legend at 0x1d7249f2978>
```



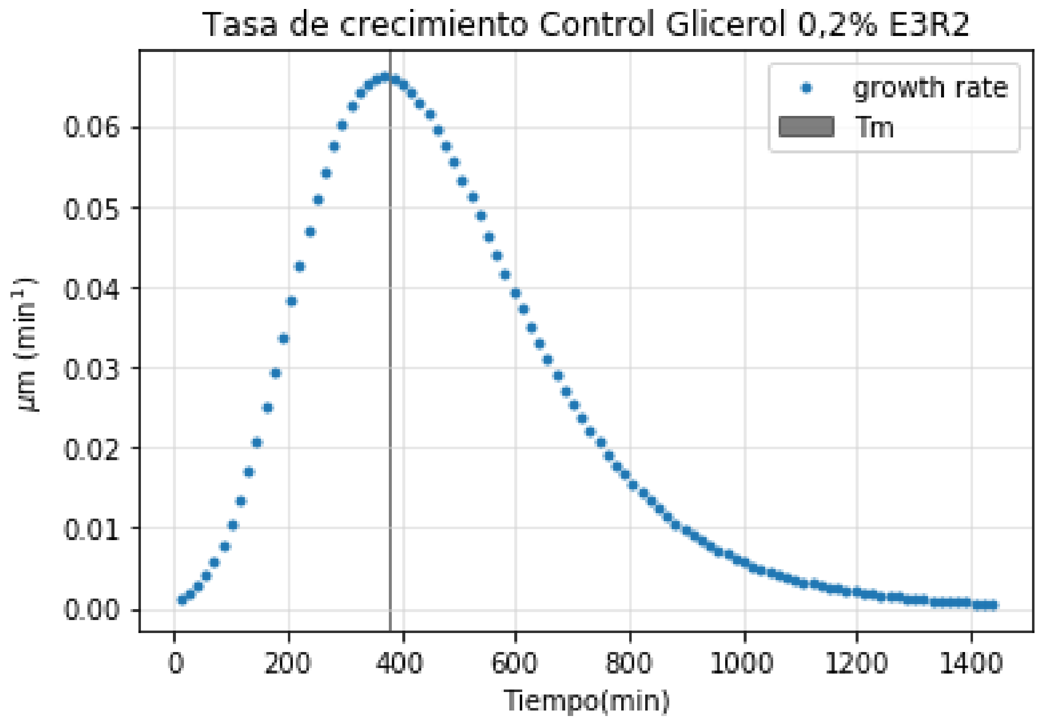

```
In [93]: #Tasas control réplicas glucosa
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy1,label='growth rate E3R1')
         plt.plot(tt[:-1],dy2,label='growth rate E3R2')
         plt.plot(tt[:-1],dy3,label='growth rate E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[93]: <matplotlib.legend.Legend at 0x1d724ac1b70>
```



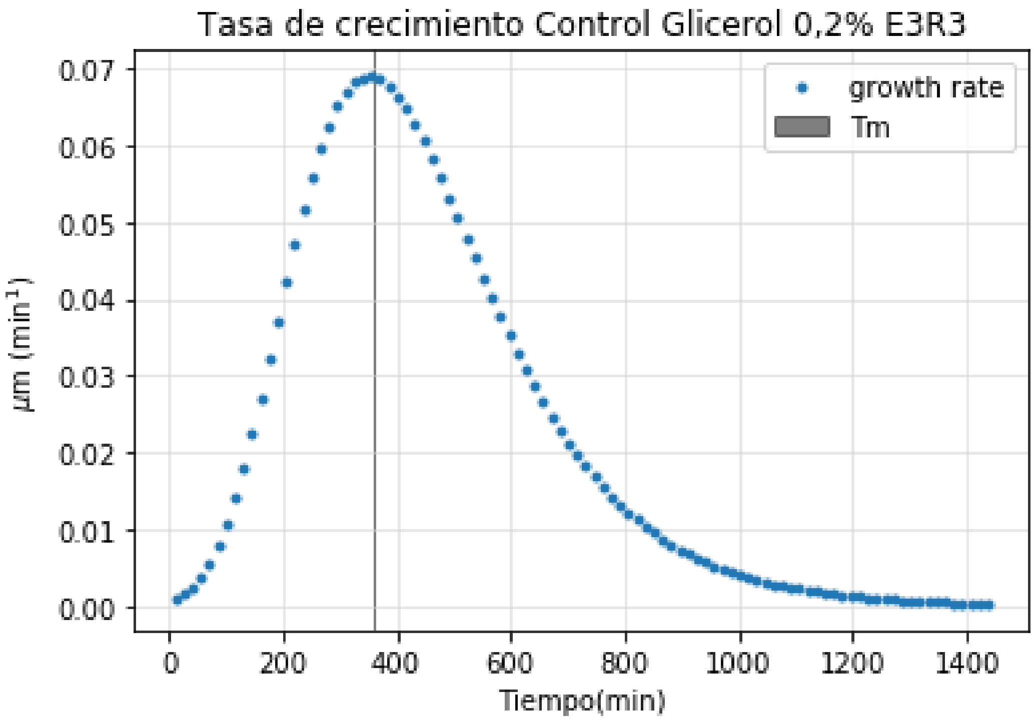

```
In [94]: #tasa de crecimiento
         ye4=((A4*np.exp(-np.exp((((um4*np.exp(1))/A4)*(l4-tt))+1))))
         #Con diff
         dy4=(np.diff(ye4))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R1')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[94]: <matplotlib.legend.Legend at 0x1d724d41a58>
```



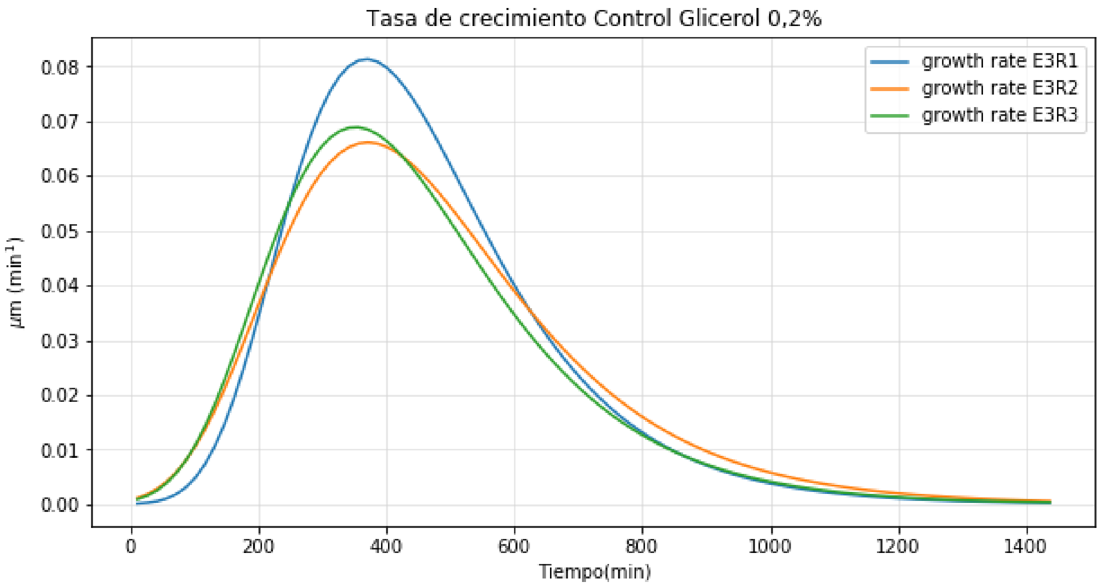

```
In [95]: #tasa de crecimiento
         ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(l5-tt))+1))))
         #Con diff
         dy5=(np.diff(ye5))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[95]: <matplotlib.legend.Legend at 0x1d724f7beb8>
```



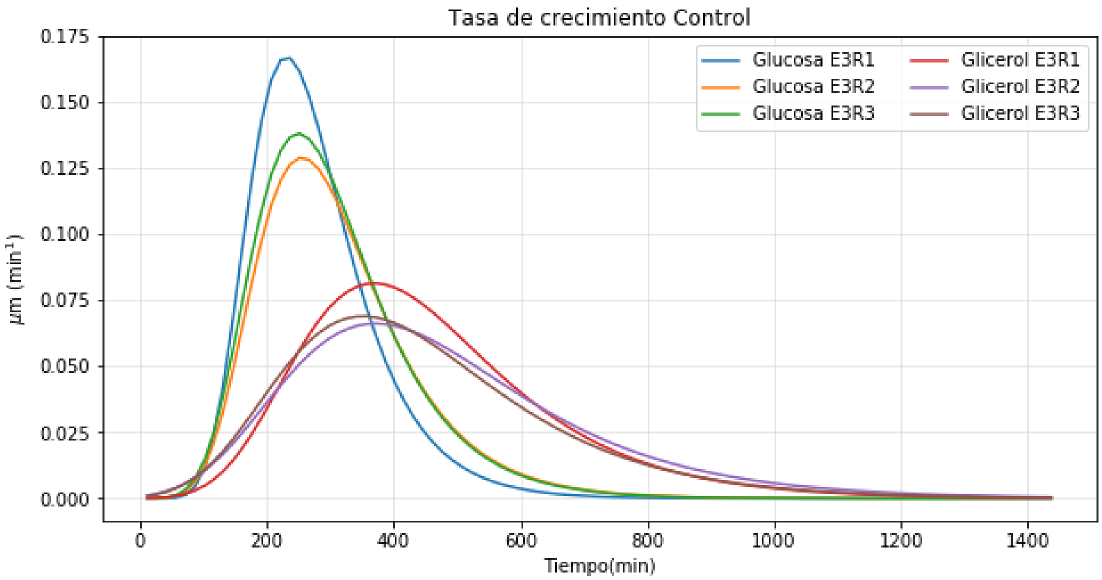

```
In [96]: #tasa de crecimiento
         ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(l6-tt))+1))))
         #Con diff
         dy6=(np.diff(ye6))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[96]: <matplotlib.legend.Legend at 0x1d725054860>
```



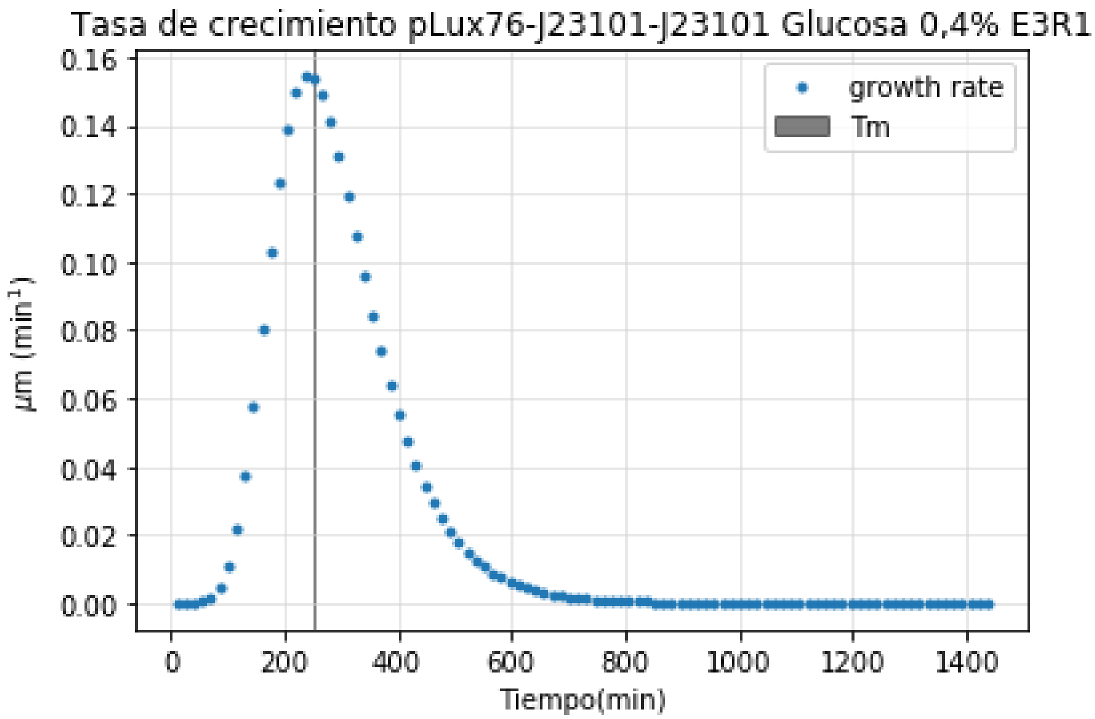

```
In [97]: #Tasas control réplicas glicerol
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy4,label='growth rate E3R1')
         plt.plot(tt[:-1],dy5,label='growth rate E3R2')
         plt.plot(tt[:-1],dy6,label='growth rate E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[97]: <matplotlib.legend.Legend at 0x1d7269d57f0>
```



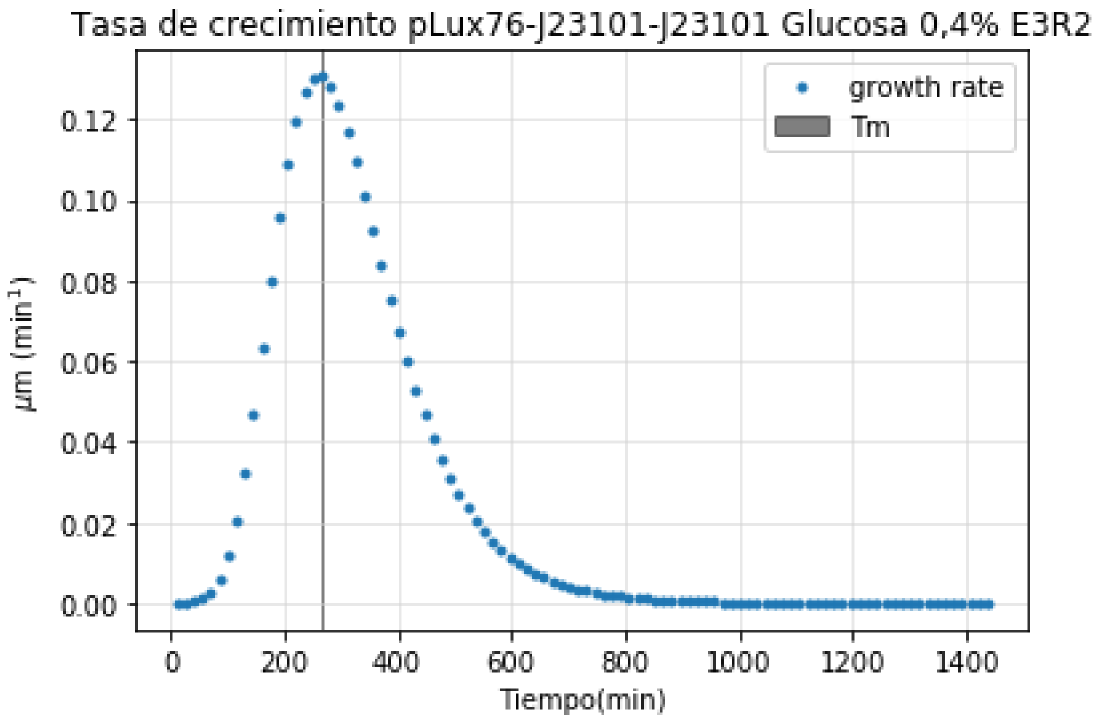

```
In [98]: #Tasas control réplicas controles
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy1,label='Glucosa E3R1')
         plt.plot(tt[:-1],dy2,label='Glucosa E3R2')
         plt.plot(tt[:-1],dy3,label='Glucosa E3R3')
         plt.plot(tt[:-1],dy4,label='Glicerol E3R1')
         plt.plot(tt[:-1],dy5,label='Glicerol E3R2')
         plt.plot(tt[:-1],dy6,label='Glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right',ncol=2)

Out[98]: <matplotlib.legend.Legend at 0x1d726aa9d68>
```



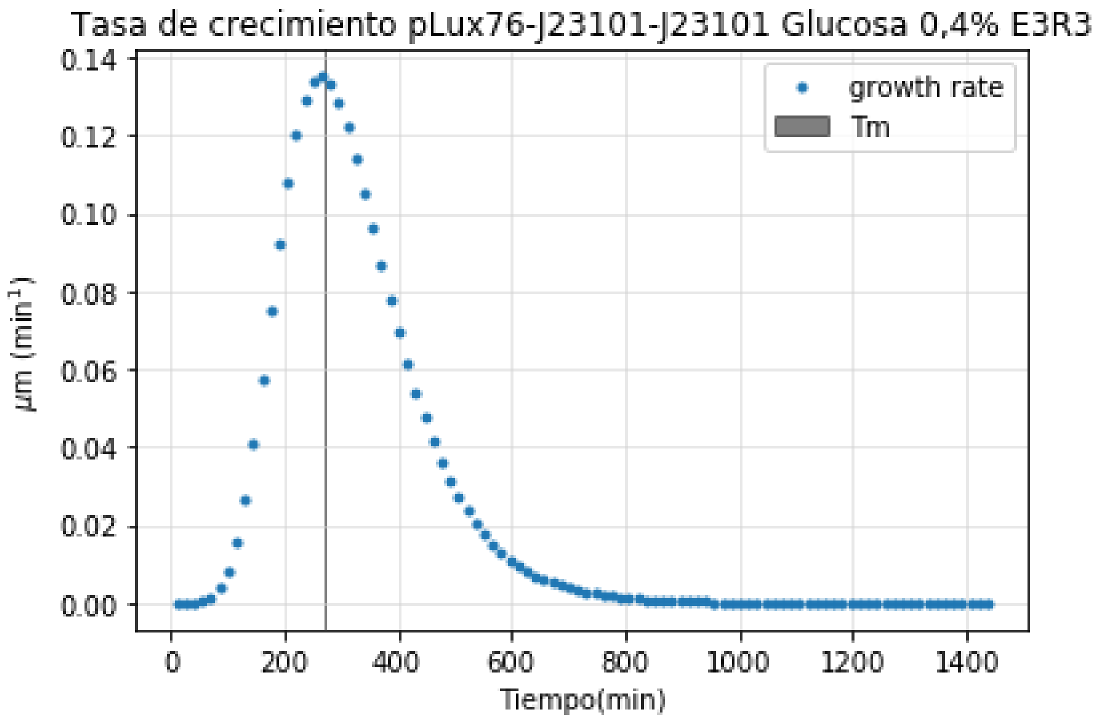

```
In [99]: #tasa de crecimiento
         ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(l7-tt))+1))))
         #Con diff
         dy7=(np.diff(ye7))
         plt.figure()
         plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E3R1')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[99]: <matplotlib.legend.Legend at 0x1d726b7eef0>
```



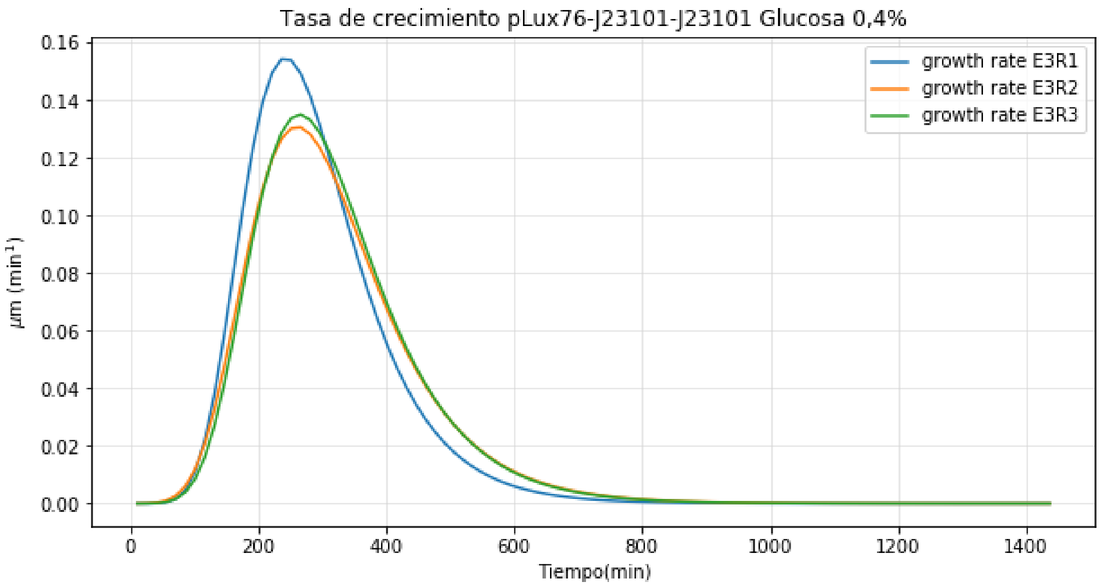

```
In [100]: #tasa de crecimiento
          ye8=((A8*np.exp(-np.exp((((um8*np.exp(1))/A8)*(l8-tt))+1))))
          #Con diff
          dy8=(np.diff(ye8))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E3R2')
          plt.xlabel('Tiempo(min)')
          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[100]: <matplotlib.legend.Legend at 0x1d7242f5320>
```




```
In [101]: #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 pLux76-J23101-J23101 Glucosa 0,4% E3R3')
          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[101]: <matplotlib.legend.Legend at 0x1d72452eda0>
```

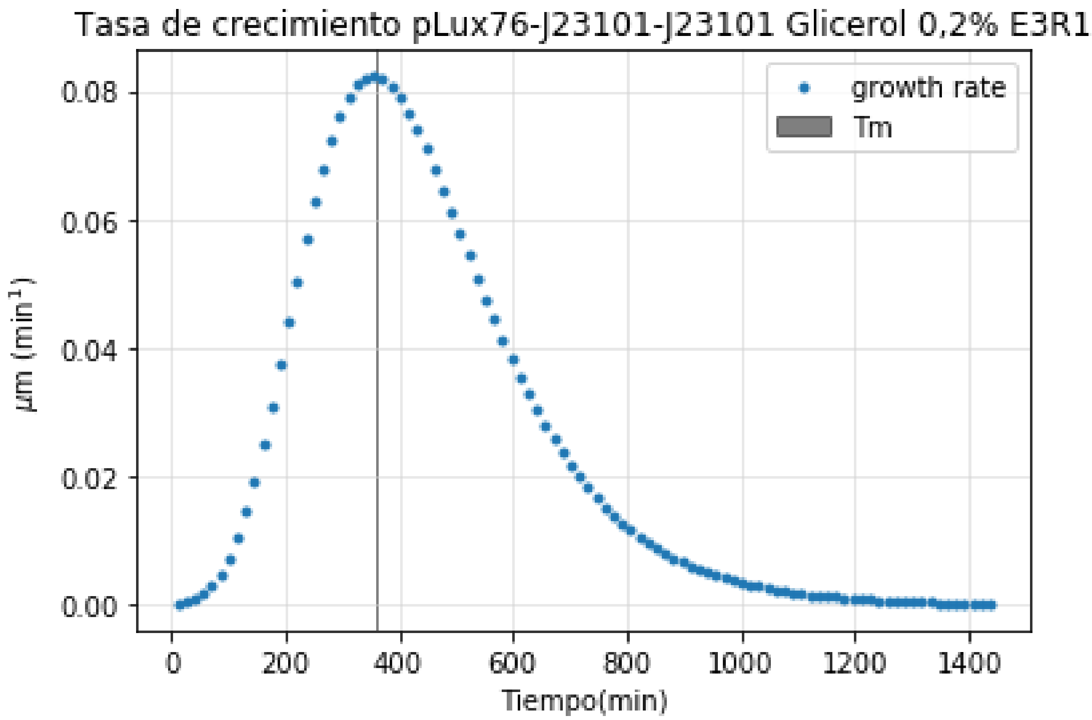



```
In [102]: #Tasas pLux-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='growth rate E3R1')
          plt.plot(tt[:-1],dy8,label='growth rate E3R2')
          plt.plot(tt[:-1],dy9,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[102]: <matplotlib.legend.Legend at 0x1d7214ee5f8>
```



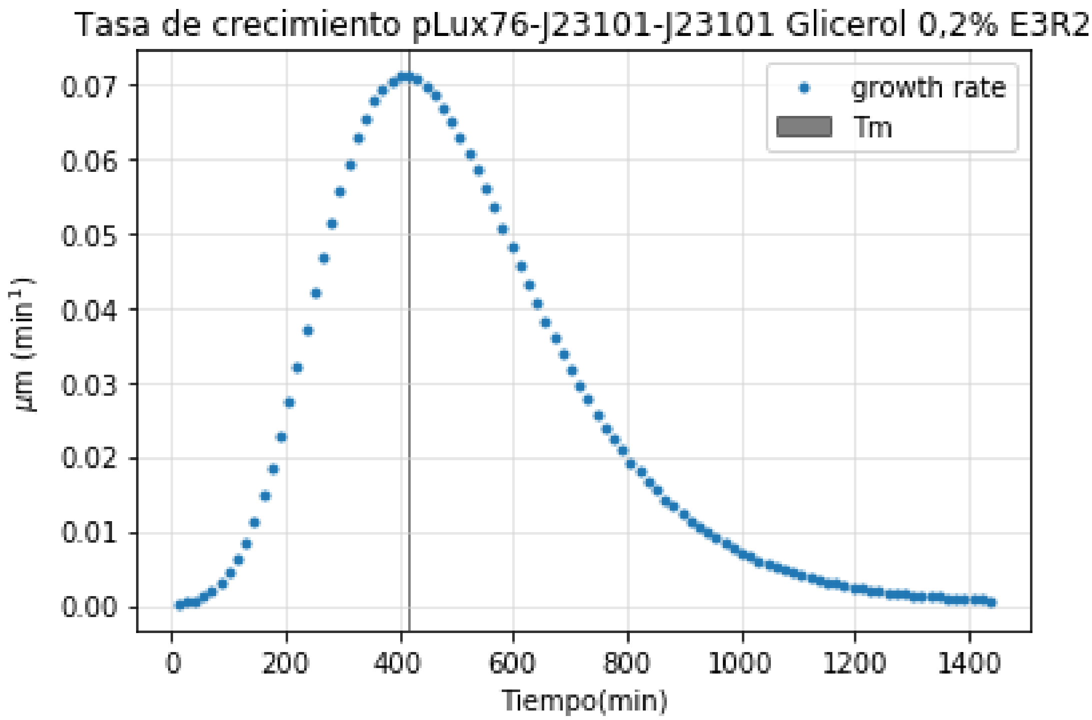
In [103]:
```
#tasa de crecimiento
ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(l10-tt))+1))))
#Con diff
dy10=(np.diff(ye10))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm10,tm10, color='k', 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[103]:
```
<matplotlib.legend.Legend at 0x1d7215f3cc0>
```

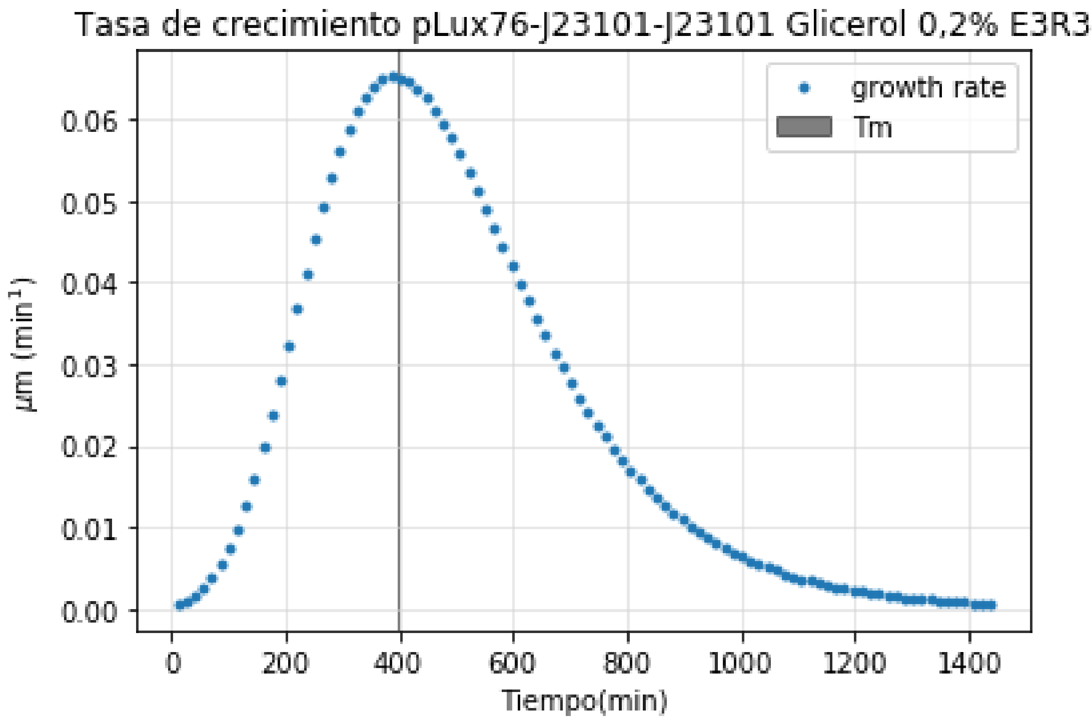


In [104]:
```
#tasa de crecimiento
ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(l11-tt))+1))))
#Con diff
dy11=(np.diff(ye11))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm11,tm11, color='k', 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')
```

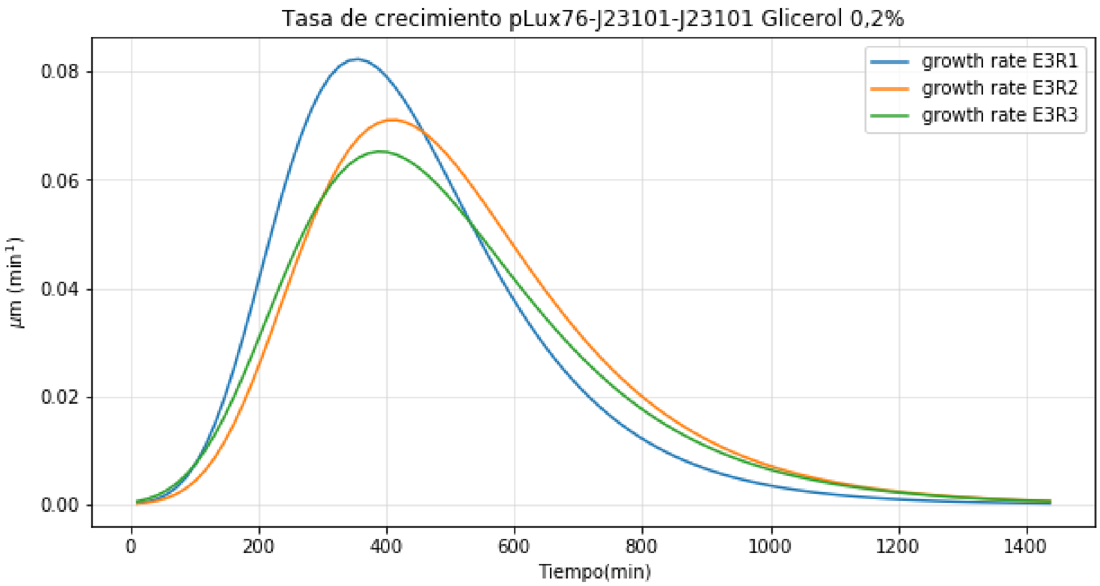
Out[104]: <matplotlib.legend.Legend at 0x1d723f633c8>



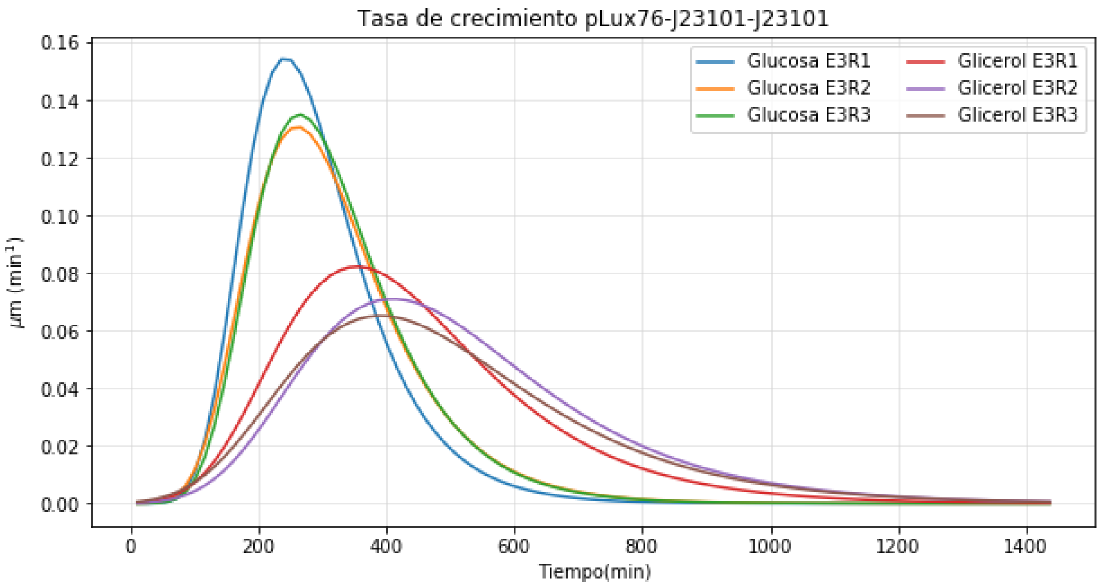

```
In [105]: #tasa de crecimiento
          ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(l12-tt))+1))))
          #Con diff
          dy12=(np.diff(ye12))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R3')
          plt.xlabel('Tiempo(min)')
          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[105]: <matplotlib.legend.Legend at 0x1d72456bb70>
```




```
In [106]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy10,label='growth rate E3R1')
          plt.plot(tt[:-1],dy11,label='growth rate E3R2')
          plt.plot(tt[:-1],dy12,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[106]: <matplotlib.legend.Legend at 0x1d725274fd0>
```

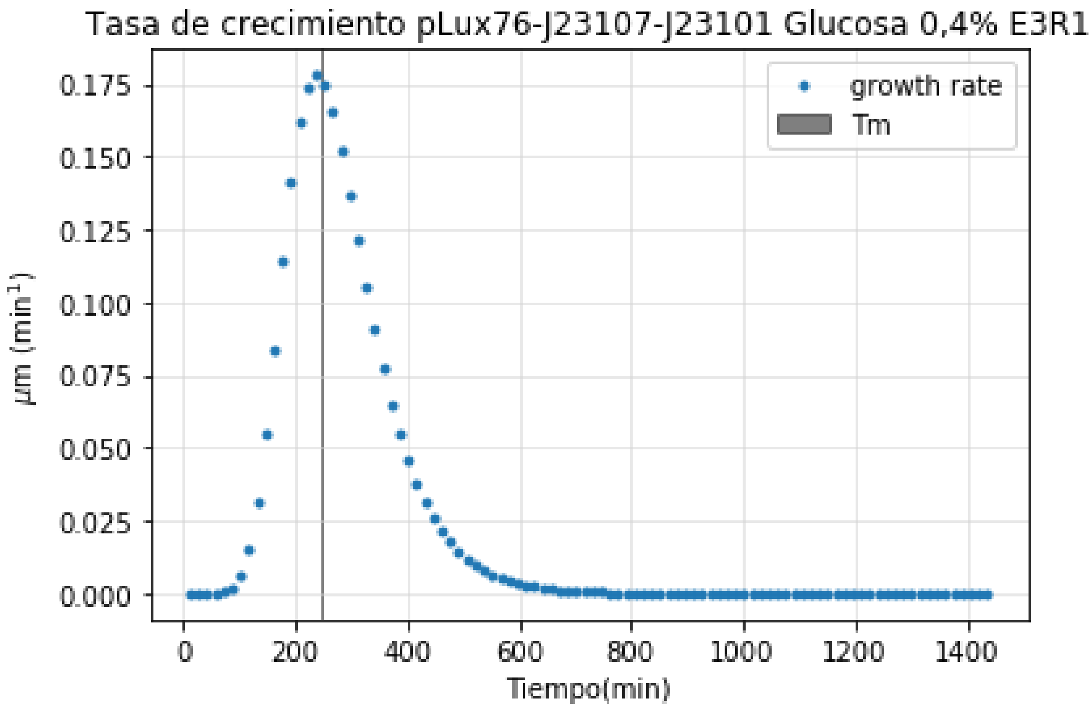



```
In [107]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy8,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy9,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy10,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy11,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy12,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[107]: <matplotlib.legend.Legend at 0x1d72535cd68>
```



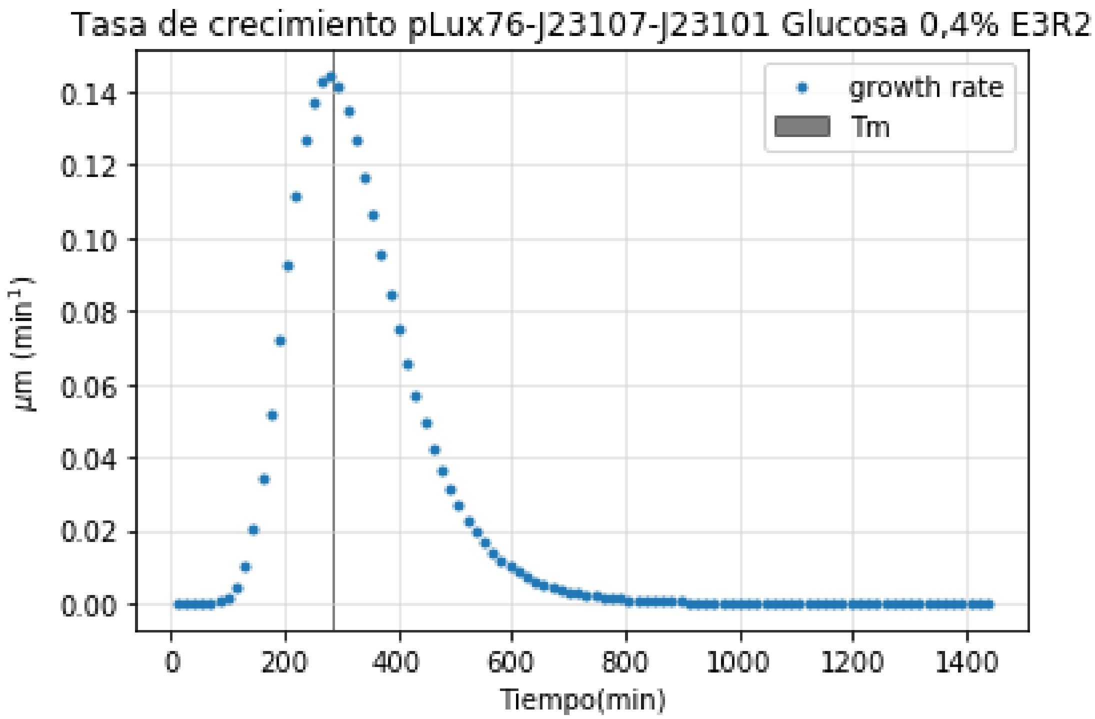
In [108]:
```
#tasa de crecimiento
ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(l13-tt))+1))))
#Con diff
dy13=(np.diff(ye13))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm13,tm13, color='k', 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[108]:
```
<matplotlib.legend.Legend at 0x1d72544ab00>
```



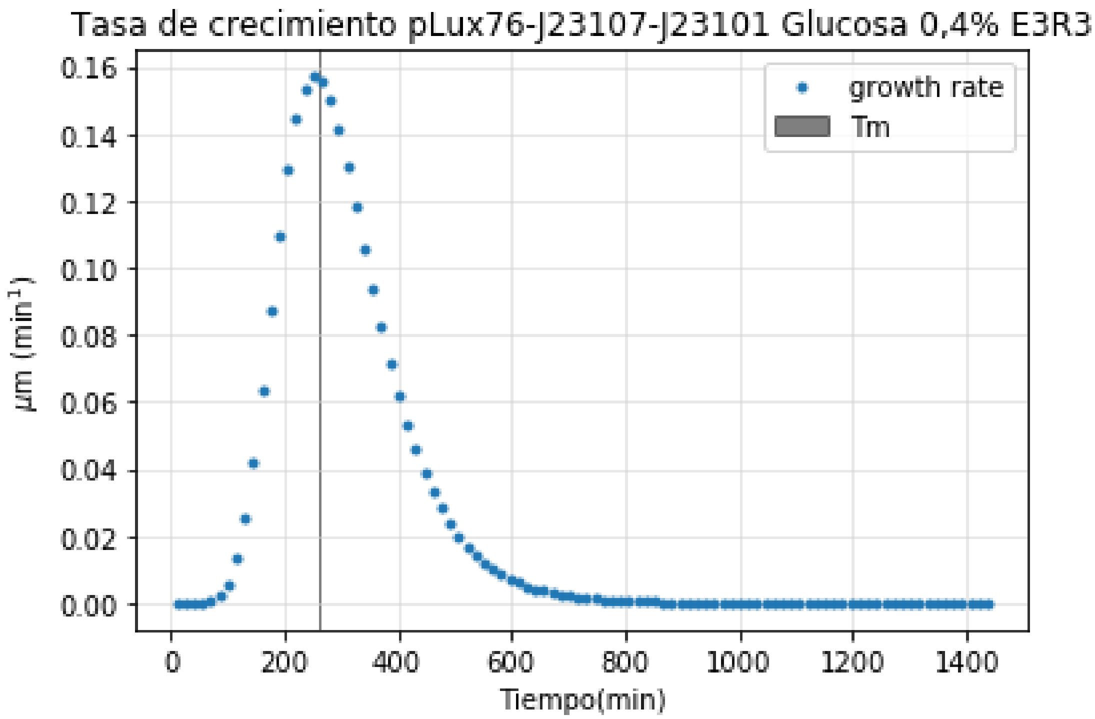
In [109]:
```
#tasa de crecimiento
ye14=((A14*np.exp(-np.exp((((um14*np.exp(1))/A14)*(l14-tt))+1))))
#Con diff
dy14=(np.diff(ye14))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm14,tm14, color='k', 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[109]: <matplotlib.legend.Legend at 0x1d72551ddd8>



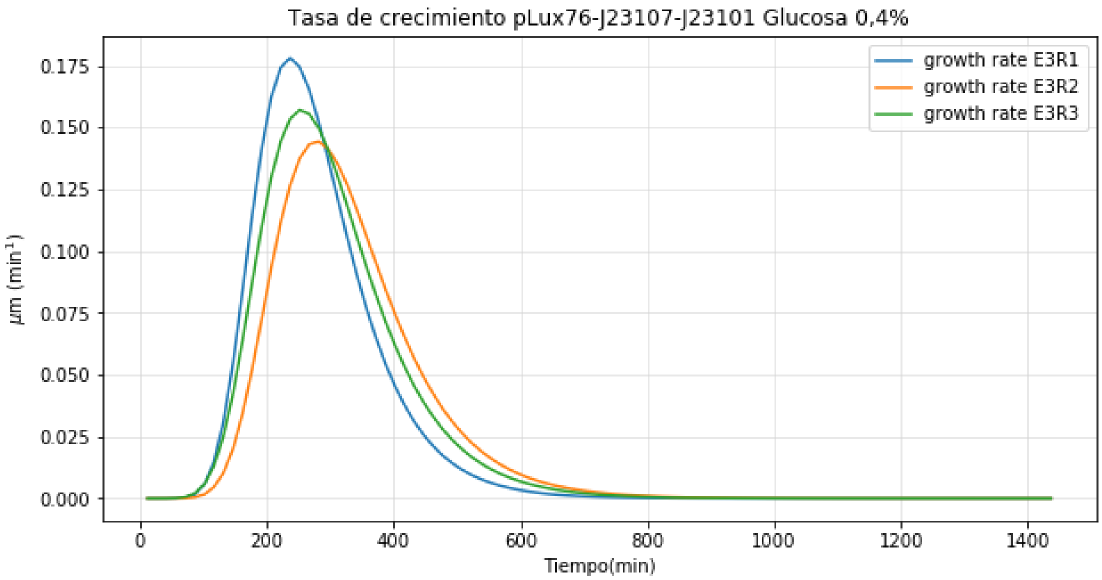
In [110]:
```
#tasa de crecimiento
ye15=((A15*np.exp(-np.exp((((um15*np.exp(1))/A15)*(l15-tt))+1))))
#Con diff
dy15=(np.diff(ye15))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm15,tm15, color='k', 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[110]: <matplotlib.legend.Legend at 0x1d7255f3b00>

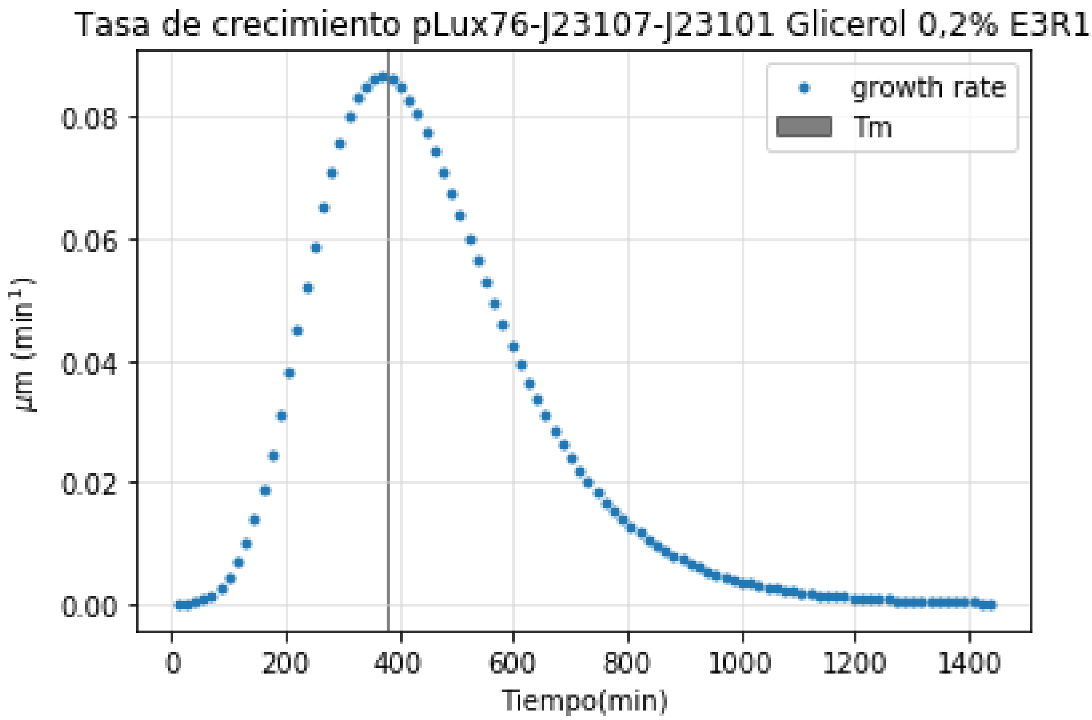


In [111]:
```
#Tasas plux76-J23101-J23101 réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy13,label='growth rate E3R1')
plt.plot(tt[:-1],dy14,label='growth rate E3R2')
plt.plot(tt[:-1],dy15,label='growth rate E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

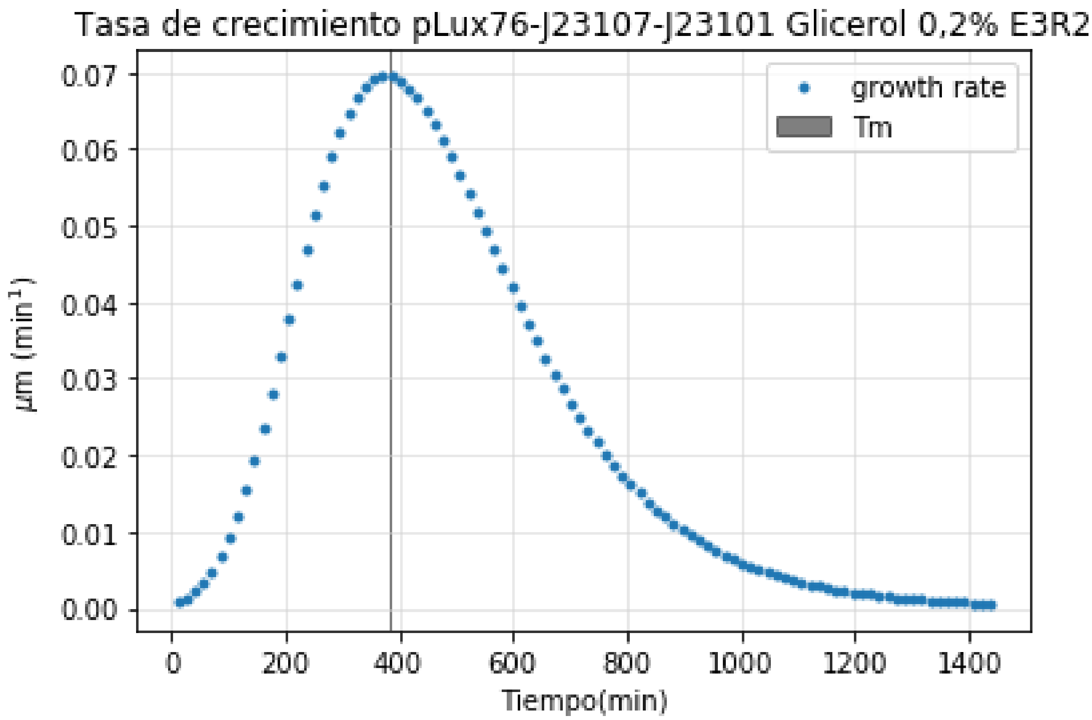
Out[111]: <matplotlib.legend.Legend at 0x1d727cb3dd8>




```
In [112]: #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 pLux76-J23107-J23101 Glicerol 0,2% E3R1')
          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[112]: <matplotlib.legend.Legend at 0x1d727d9d828>
```

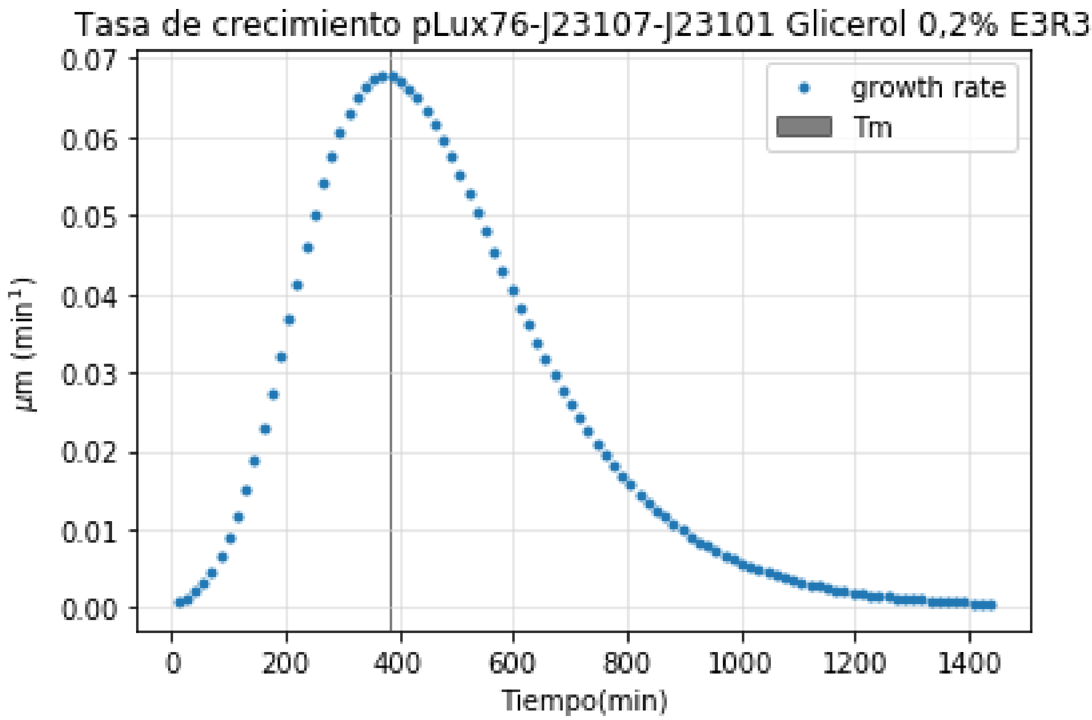



```
In [113]: #tasa de crecimiento
          ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(l17-tt))+1))))
          #Con diff
          dy17=(np.diff(ye17))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glicerol 0,2% E3R2')
          plt.xlabel('Tiempo(min)')
          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[113]: <matplotlib.legend.Legend at 0x1d727e65d68>
```

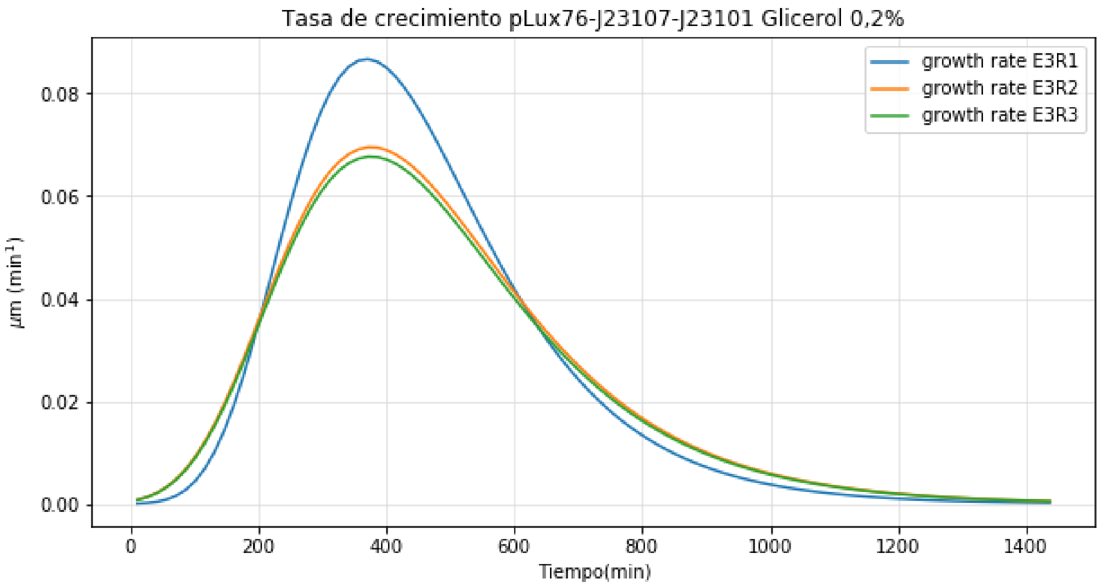


In [114]:
```
#tasa de crecimiento
ye18=((A18*np.exp(-np.exp((((um18*np.exp(1))/A18)*(l18-tt))+1))))
#Con diff
dy18=(np.diff(ye18))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^-1$)')
plt.axvspan(tm18,tm18, color='k', 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')
```

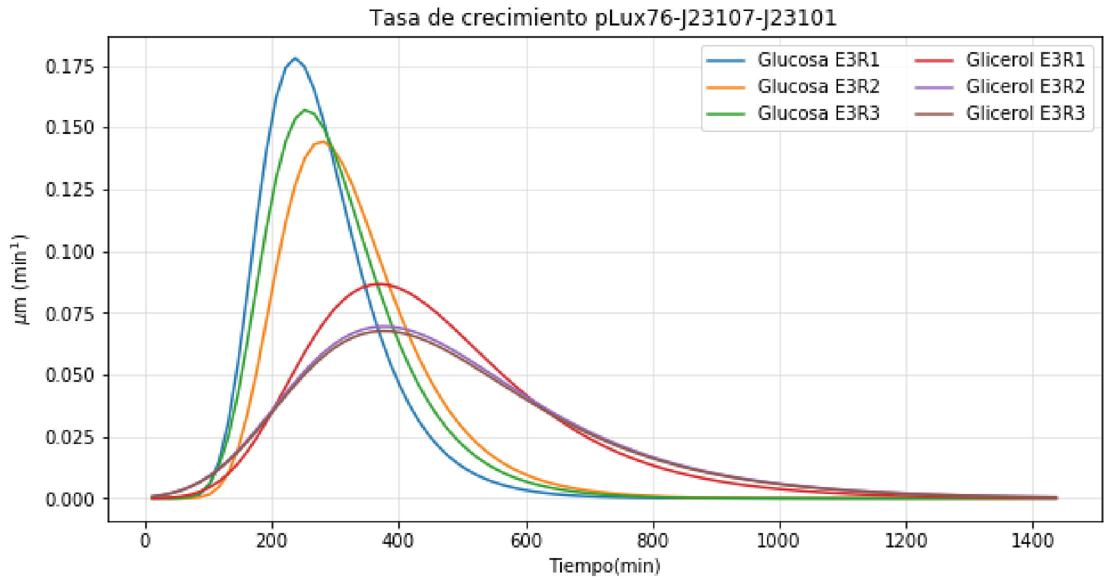
Out[114]: <matplotlib.legend.Legend at 0x1d727f41cf8>



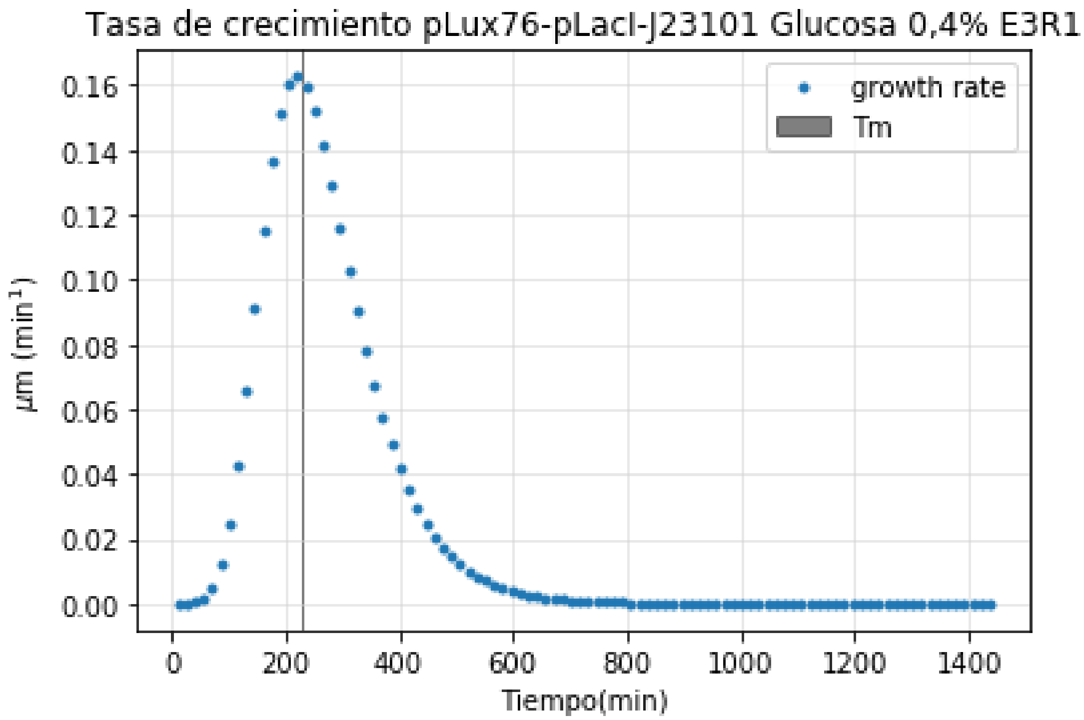

```
In [115]: plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23107-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy16,label='growth rate E3R1')
          plt.plot(tt[:-1],dy17,label='growth rate E3R2')
          plt.plot(tt[:-1],dy18,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[115]: <matplotlib.legend.Legend at 0x1d728017c50>
```



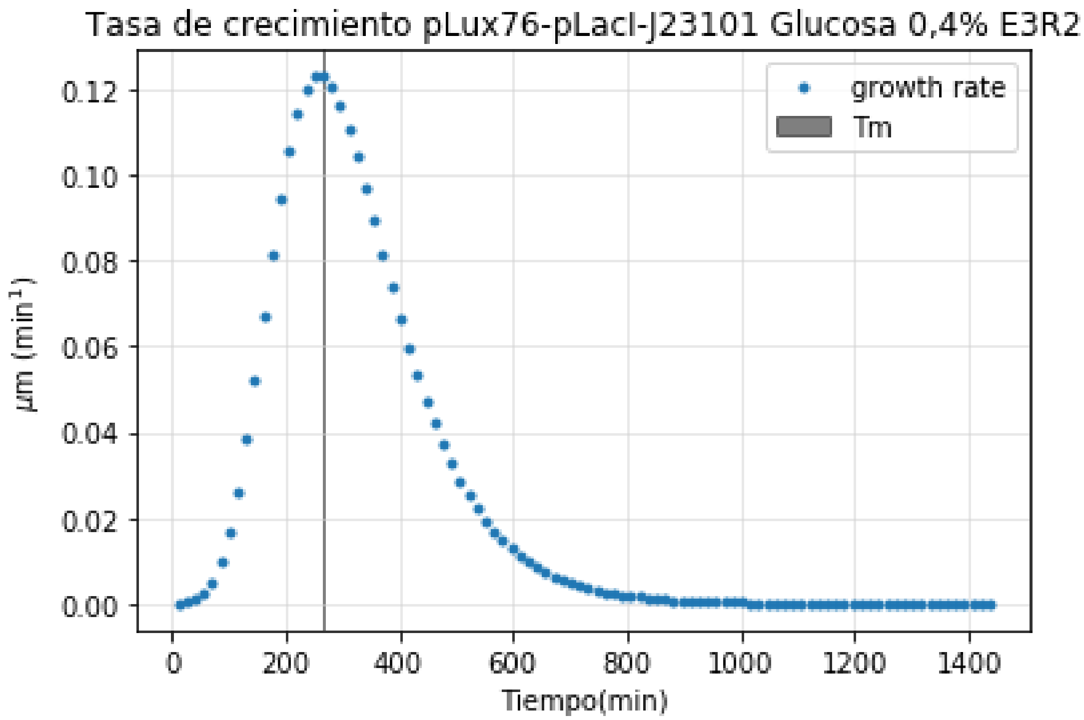

```
In [116]: plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23107-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy14,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy15,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy16,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy17,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy18,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[116]: <matplotlib.legend.Legend at 0x1d728101908>
```




```
In [117]: #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 pLux76-pLacI-J23101 Glucosa 0,4% E3R1')
          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[117]: <matplotlib.legend.Legend at 0x1d7281db978>
```

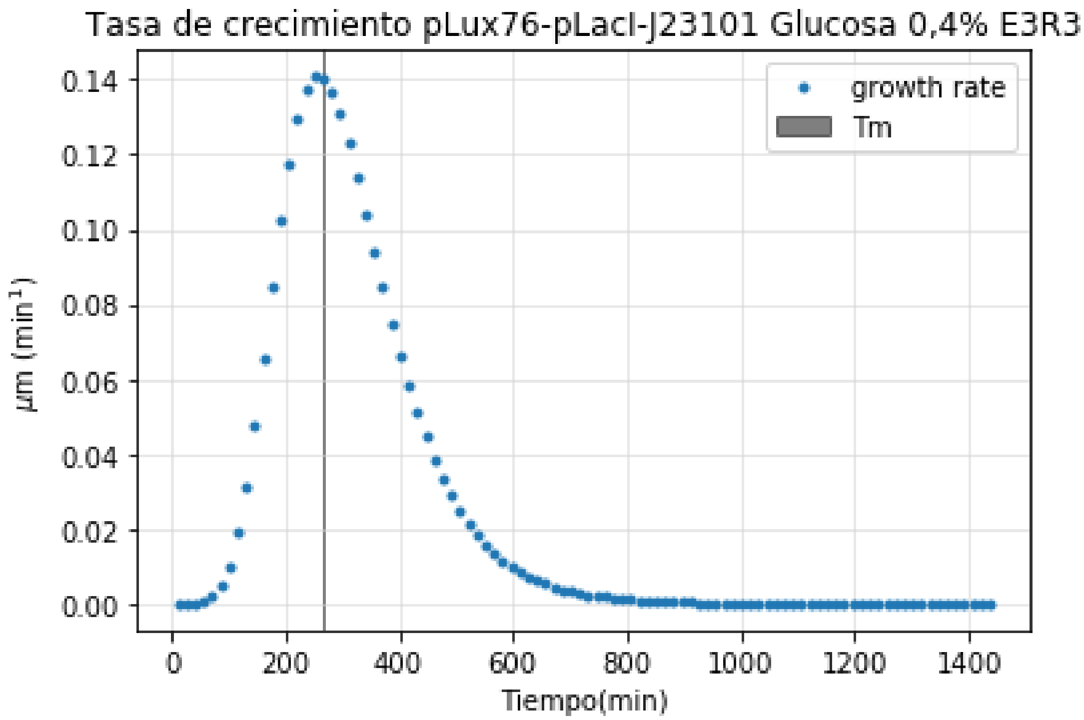



```
In [118]: #tasa de crecimiento
          ye20=((A20*np.exp(-np.exp((((um20*np.exp(1))/A20)*(l20-tt))+1))))
          #Con diff
          dy20=(np.diff(ye20))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4% E3R2')
          plt.xlabel('Tiempo(min)')
          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[118]: <matplotlib.legend.Legend at 0x1d7282b4f28>
```

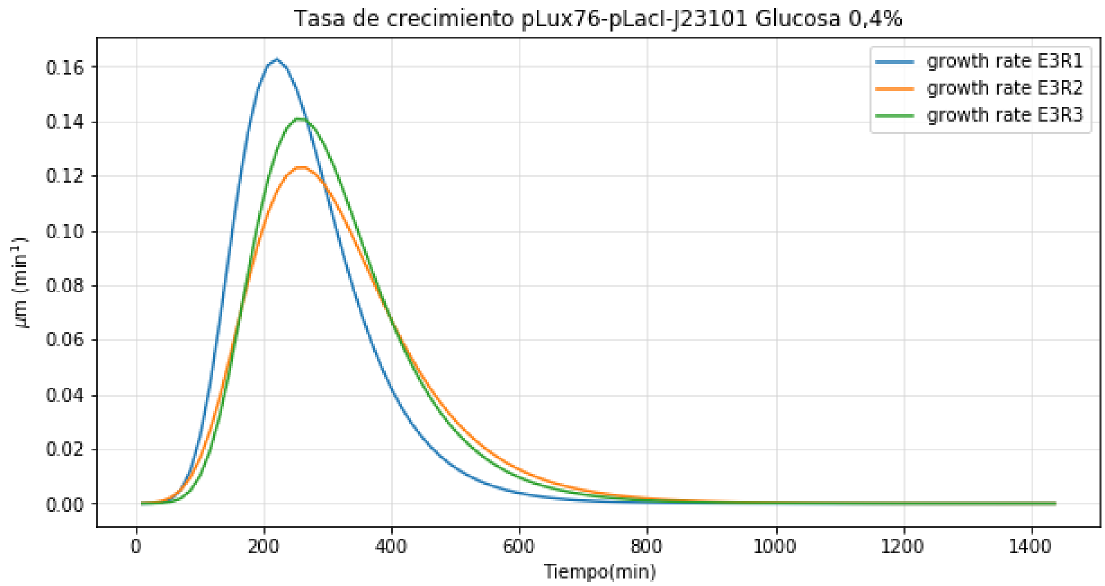


In [119]:
```
#tasa de crecimiento
ye21=((A21*np.exp(-np.exp((((um21*np.exp(1))/A21)*(l21-tt))+1))))
#Con diff
dy21=(np.diff(ye21))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm21,tm21, color='k', 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')
```

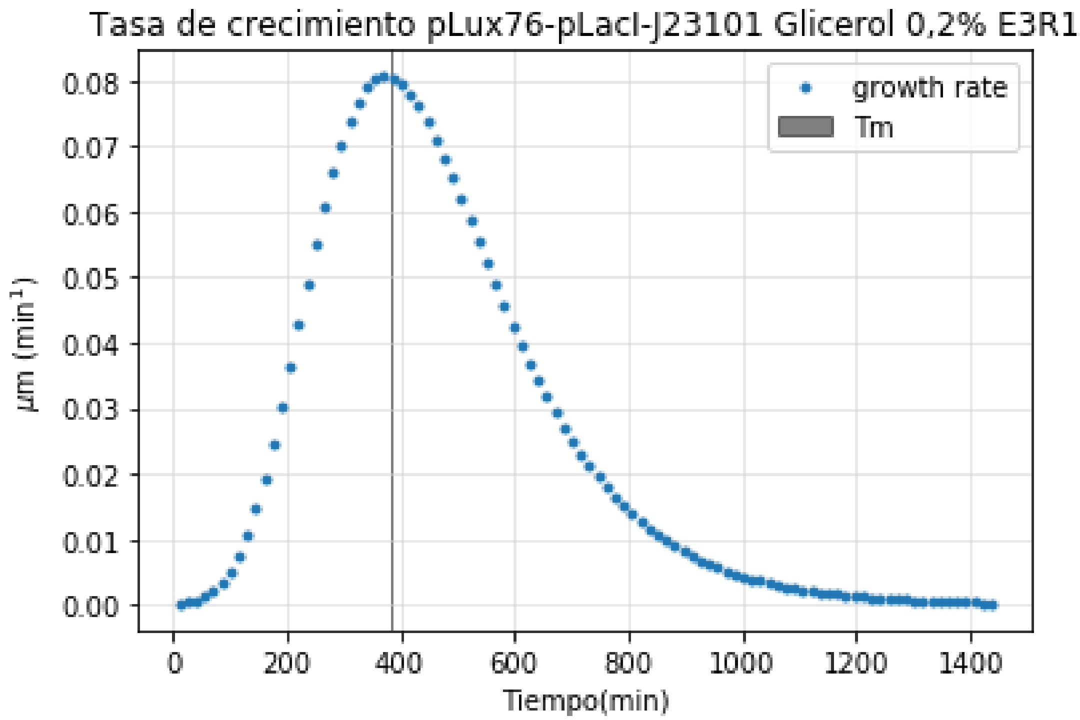
Out[119]:
```
<matplotlib.legend.Legend at 0x1d728386dd8>
```



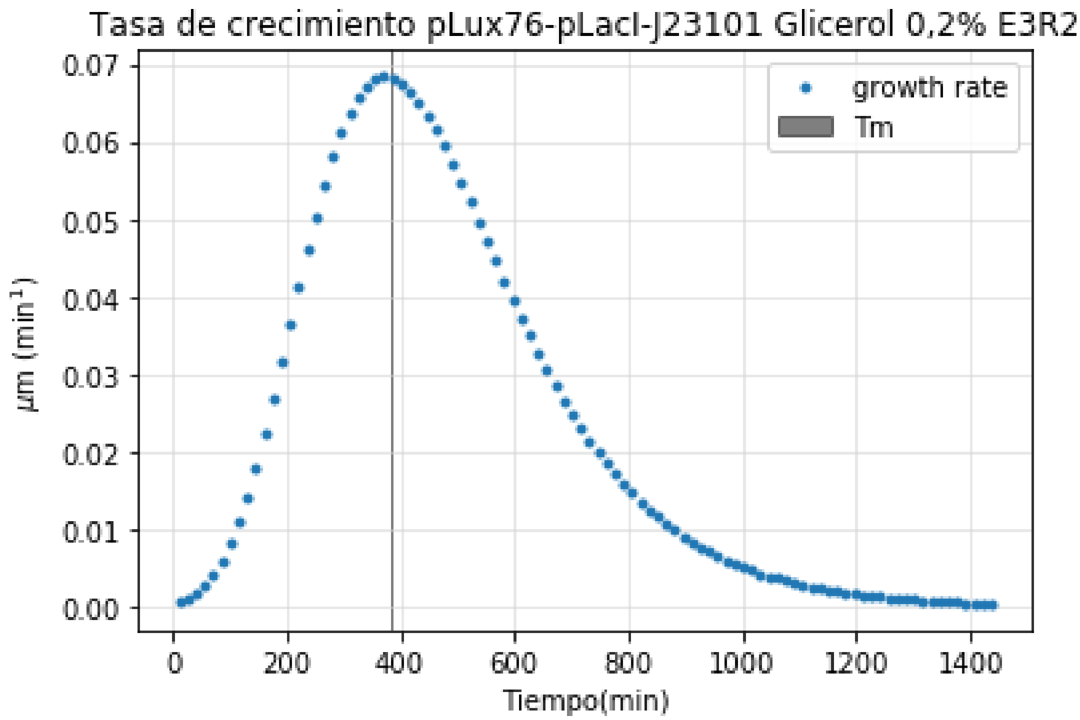

```
In [120]: #Tasas plux76-ptet-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='growth rate E3R1')
          plt.plot(tt[:-1],dy20,label='growth rate E3R2')
          plt.plot(tt[:-1],dy21,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[120]: <matplotlib.legend.Legend at 0x1d728463dd8>
```




```
In [121]: #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 pLux76-pLacI-J23101 Glicerol 0,2% E3R1')
          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[121]: <matplotlib.legend.Legend at 0x1d7294f9b00>
```

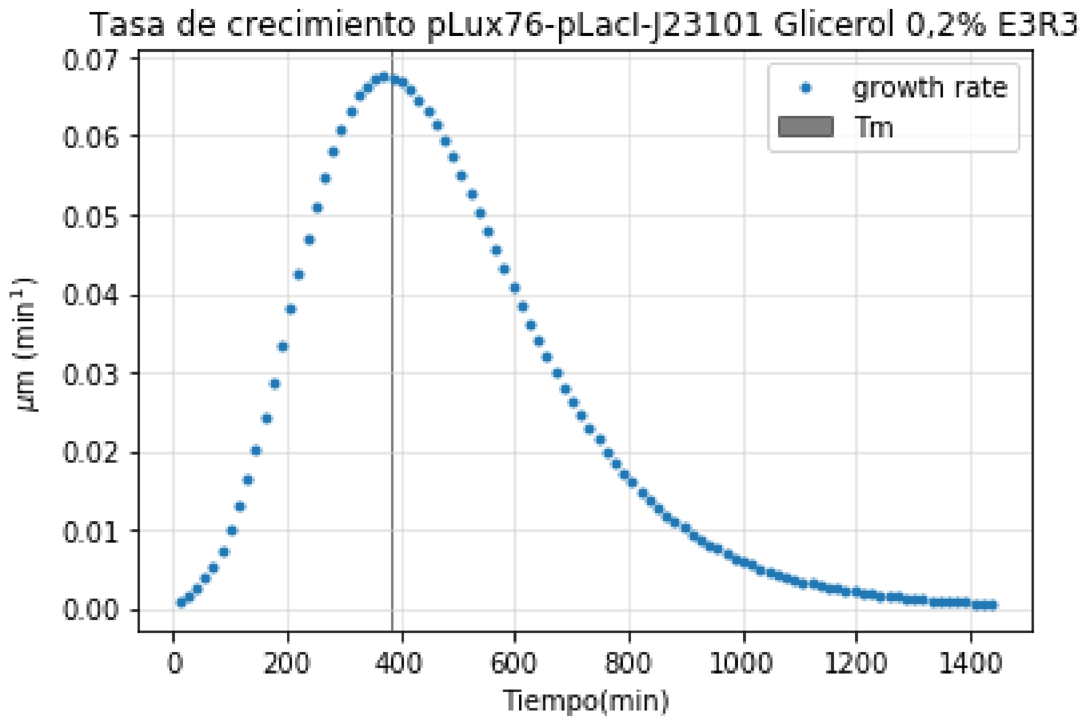



```
In [122]: #tasa de crecimiento
          ye23=((A23*np.exp(-np.exp((((um23*np.exp(1))/A23)*(l23-tt))+1))))
          #Con diff
          dy23=(np.diff(ye23))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2% E3R2')
          plt.xlabel('Tiempo(min)')
          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[122]: <matplotlib.legend.Legend at 0x1d7295d9da0>
```

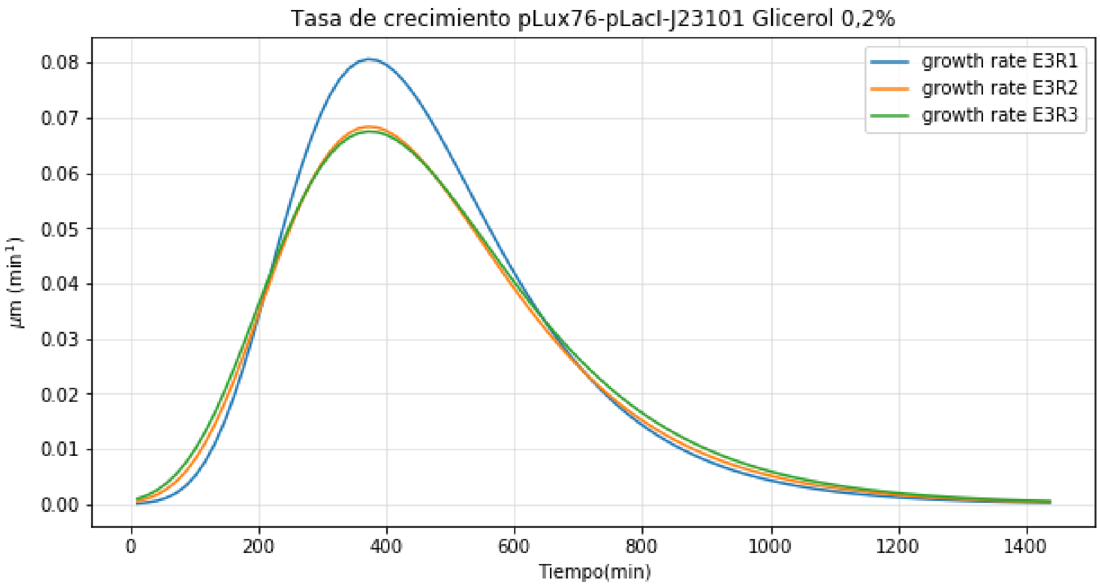


In [123]:
```
#tasa de crecimiento
ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(l24-tt))+1))))
#Con diff
dy24=(np.diff(ye24))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm24,tm24, color='k', 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')
```

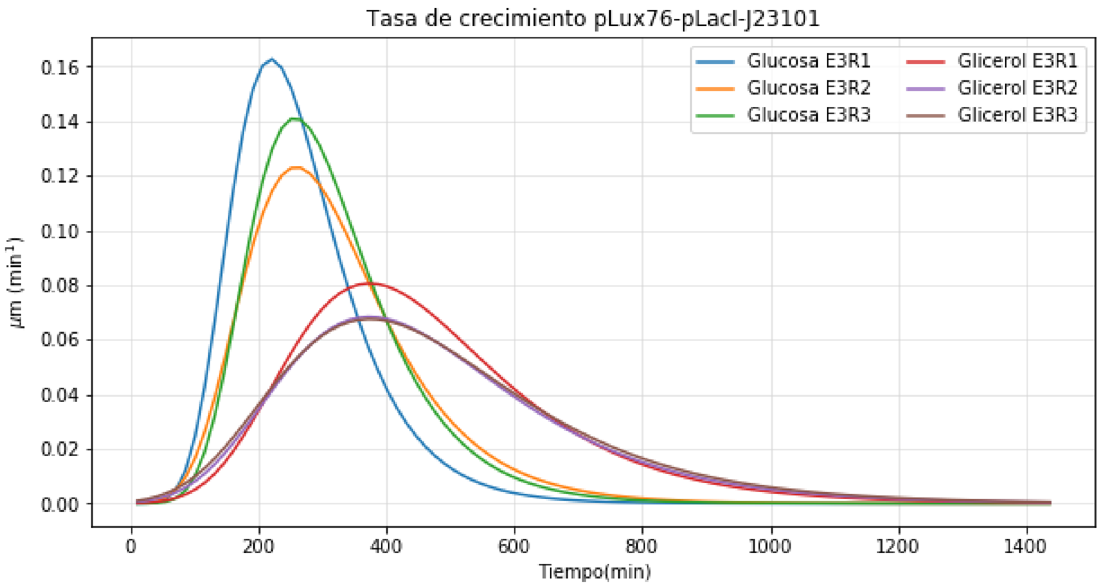
Out[123]:
```
<matplotlib.legend.Legend at 0x1d727f51a58>
```




```
In [124]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy22,label='growth rate E3R1')
          plt.plot(tt[:-1],dy23,label='growth rate E3R2')
          plt.plot(tt[:-1],dy24,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[124]: <matplotlib.legend.Legend at 0x1d726c02ba8>
```

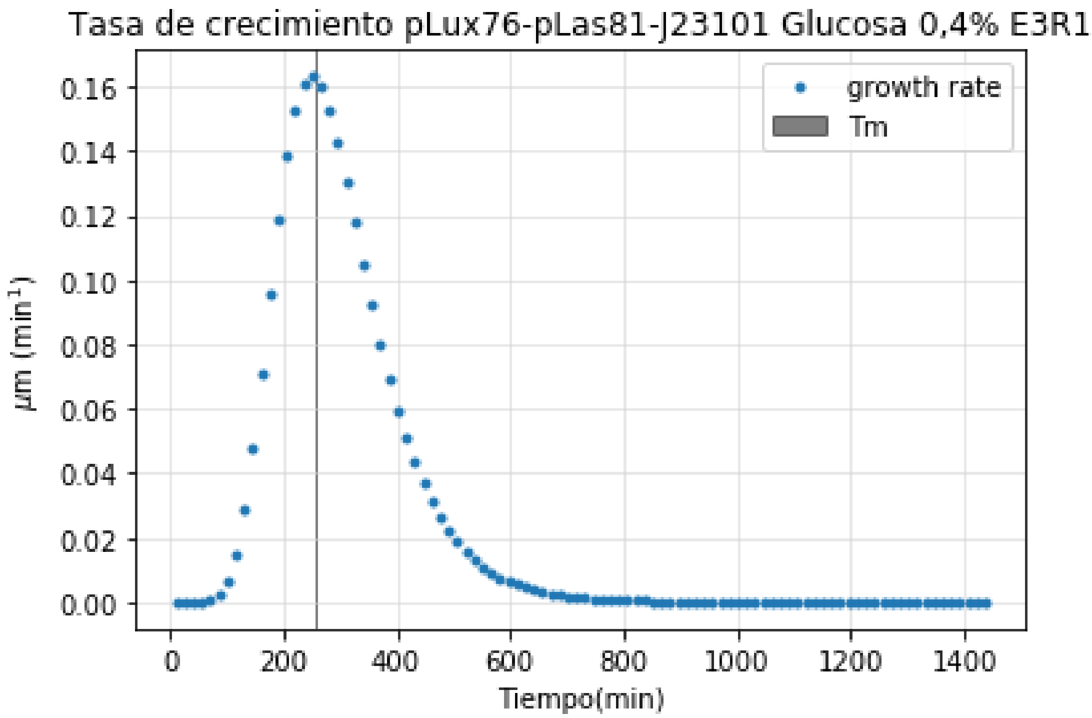



```
In [125]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy20,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy21,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy22,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy23,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy24,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[125]: <matplotlib.legend.Legend at 0x1d729674898>
```



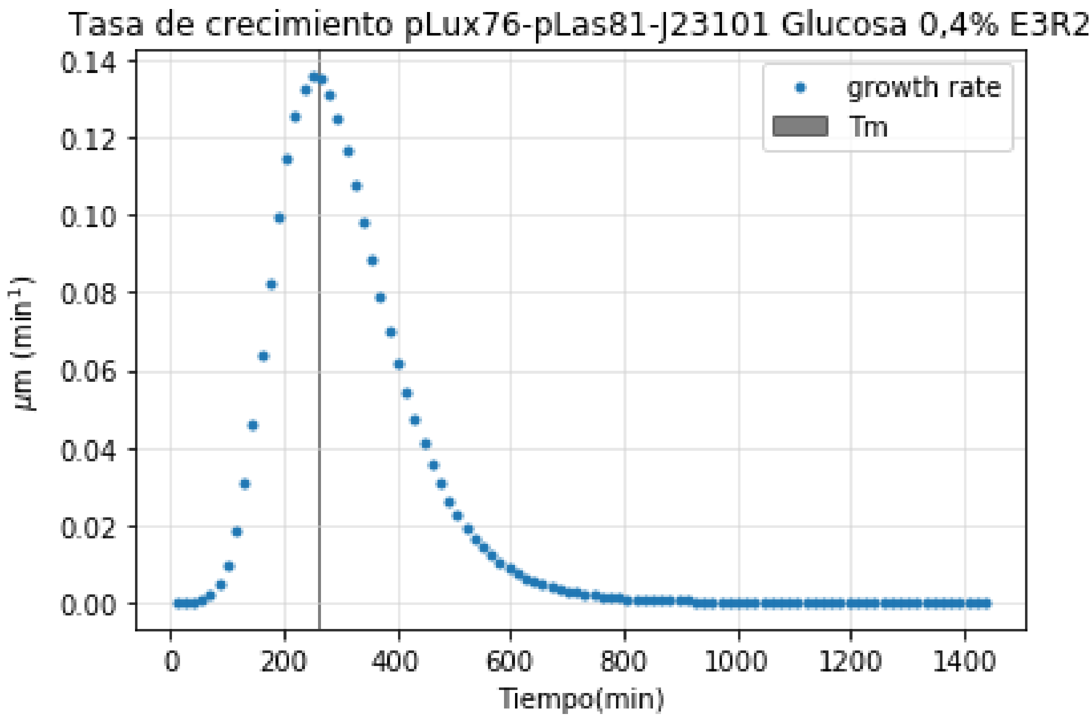
In [126]:
```
#tasa de crecimiento
ye25=((A25*np.exp(-np.exp((((um25*np.exp(1))/A25)*(l25-tt))+1))))
#Con diff
dy25=(np.diff(ye25))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm25,tm25, color='k', 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[126]:
```
<matplotlib.legend.Legend at 0x1d72970eb38>
```



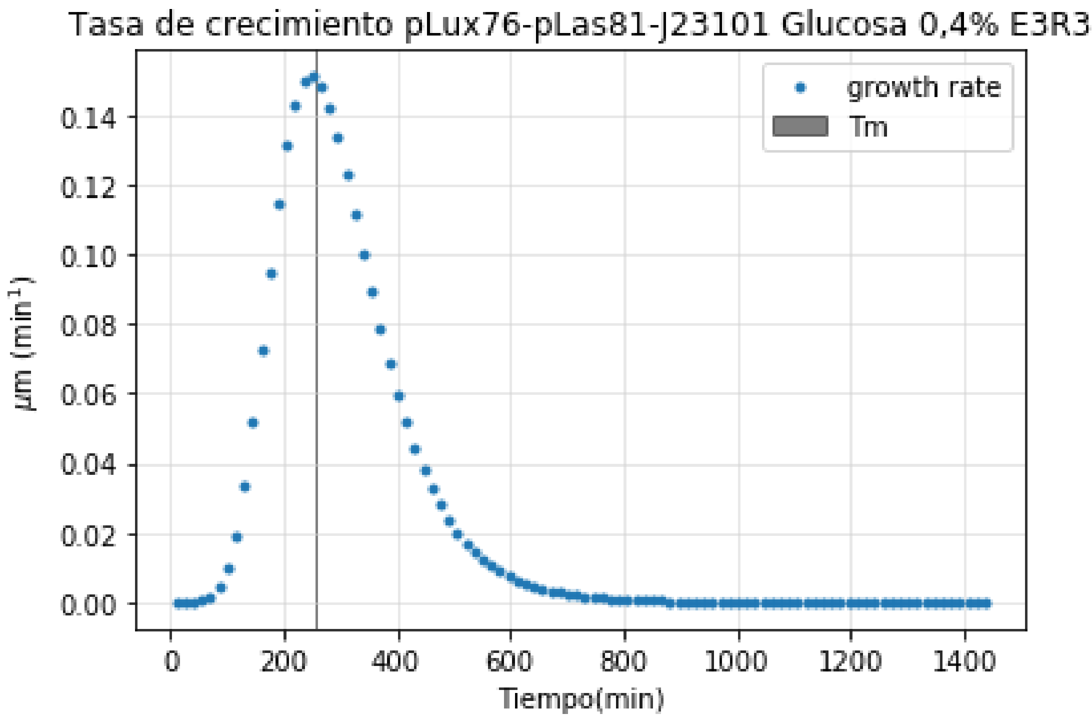
In [127]:
```
#tasa de crecimiento
ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(l26-tt))+1))))
#Con diff
dy26=(np.diff(ye26))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm26,tm26, color='k', 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[127]: <matplotlib.legend.Legend at 0x1d7186dff28>

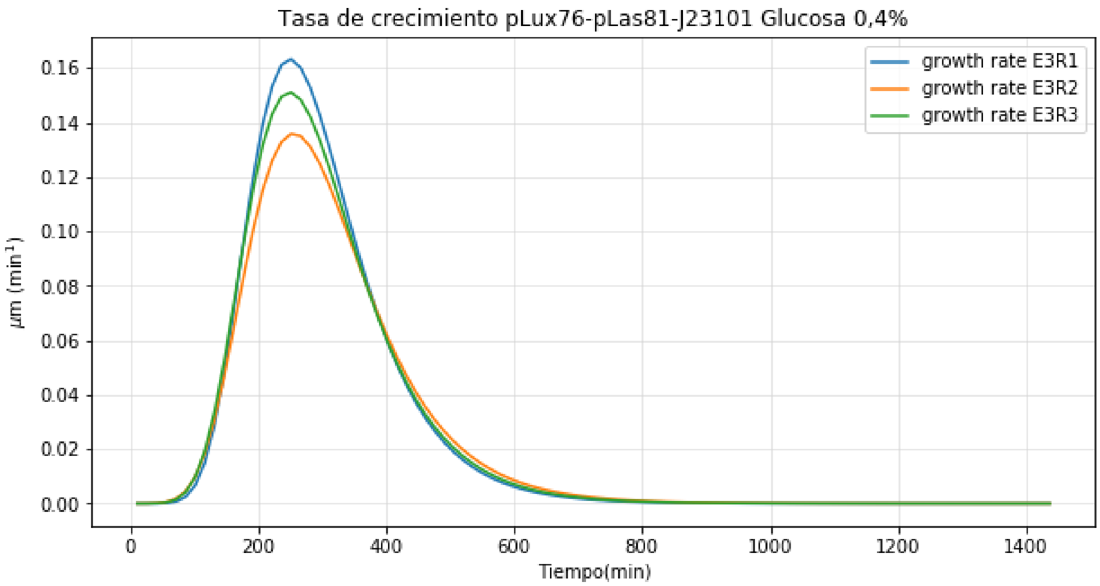


In [128]:
```
#tasa de crecimiento
ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(l27-tt))+1))))
#Con diff
dy27=(np.diff(ye27))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm27,tm27, color='k', 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')
```

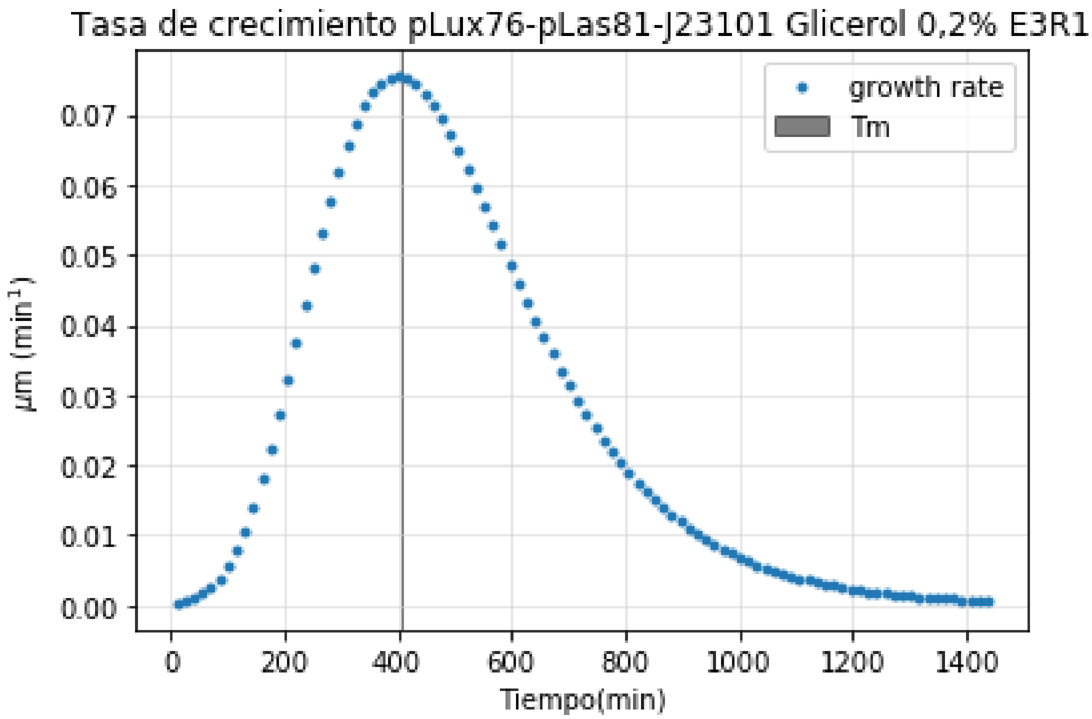
Out[128]:
```
<matplotlib.legend.Legend at 0x1d7187acef0>
```



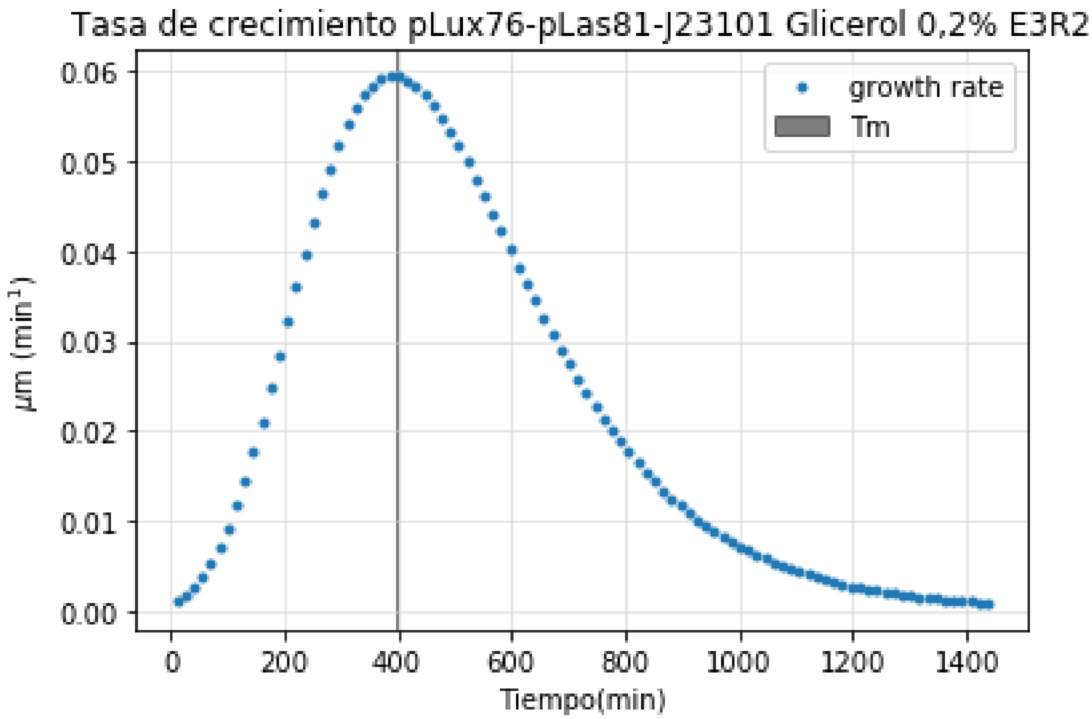

```
In [129]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='growth rate E3R1')
          plt.plot(tt[:-1],dy26,label='growth rate E3R2')
          plt.plot(tt[:-1],dy27,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[129]: <matplotlib.legend.Legend at 0x1d718879f60>
```




```
In [130]: #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 pLux76-pLas81-J23101 Glicerol 0,2% E3R1')
          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[130]: <matplotlib.legend.Legend at 0x1d729cbdc88>
```

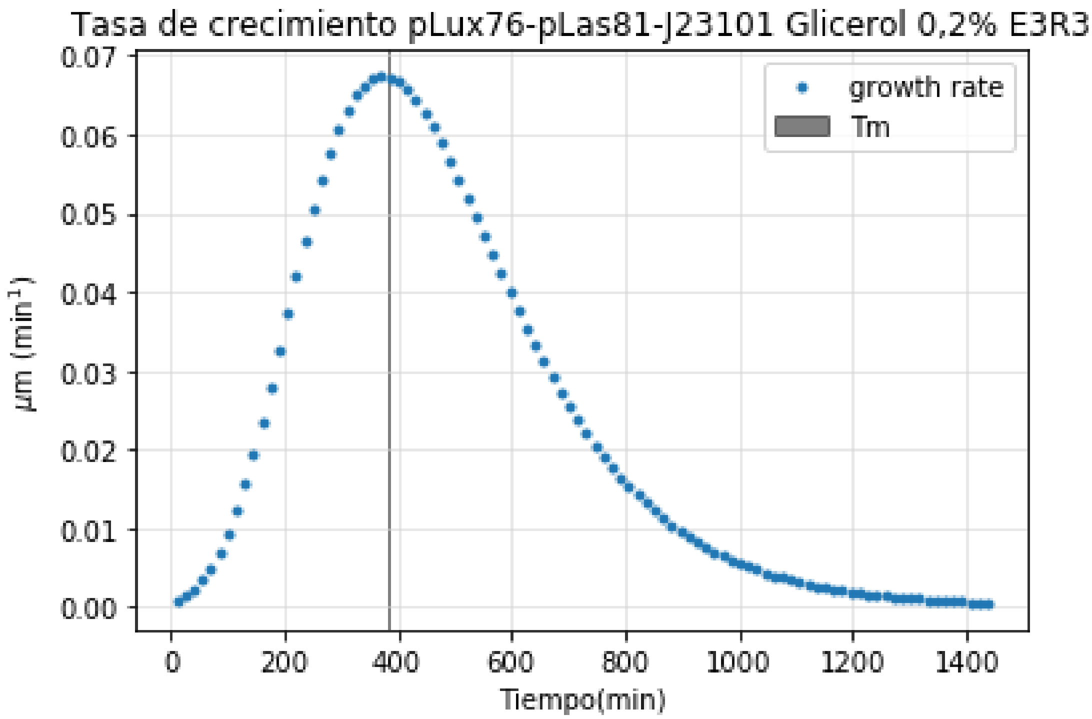



```
In [131]: #tasa de crecimiento
          ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(l29-tt))+1))))
          #Con diff
          dy29=(np.diff(ye29))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2% E3R2')
          plt.xlabel('Tiempo(min)')
          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[131]: <matplotlib.legend.Legend at 0x1d729d97f98>
```

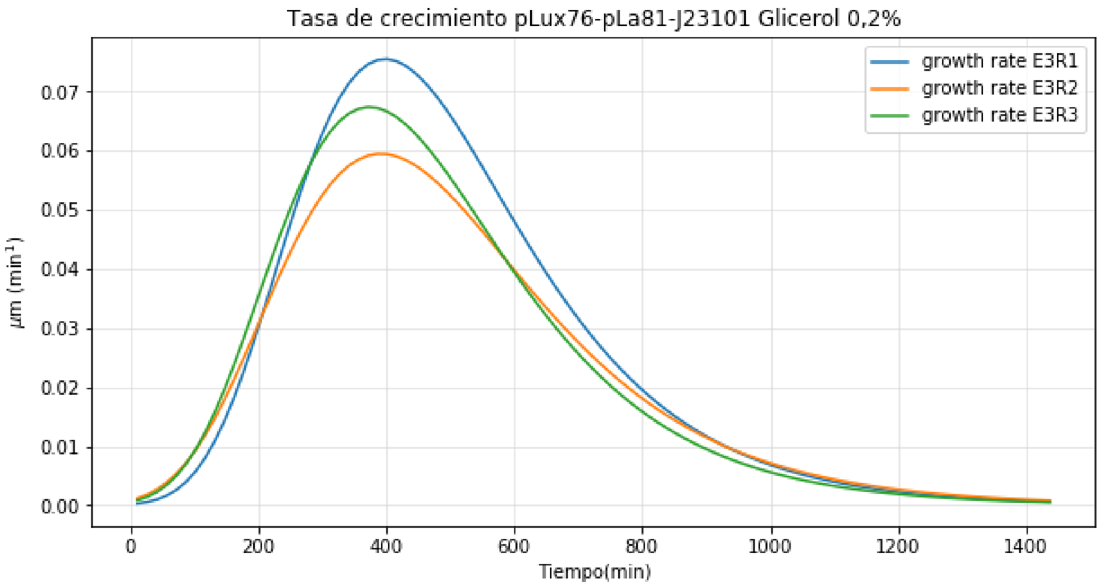


In [132]:
```
#tasa de crecimiento
ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(l30-tt))+1))))
#Con diff
dy30=(np.diff(ye30))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLas81-J23101 Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm30,tm30, color='k', 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')
```

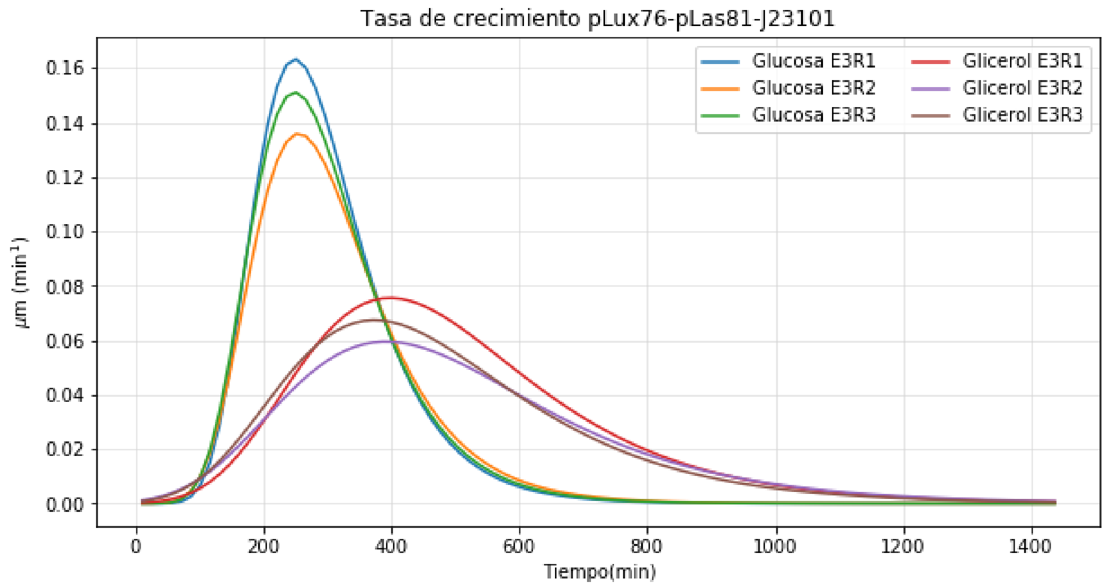
Out[132]: <matplotlib.legend.Legend at 0x1d729e63860>



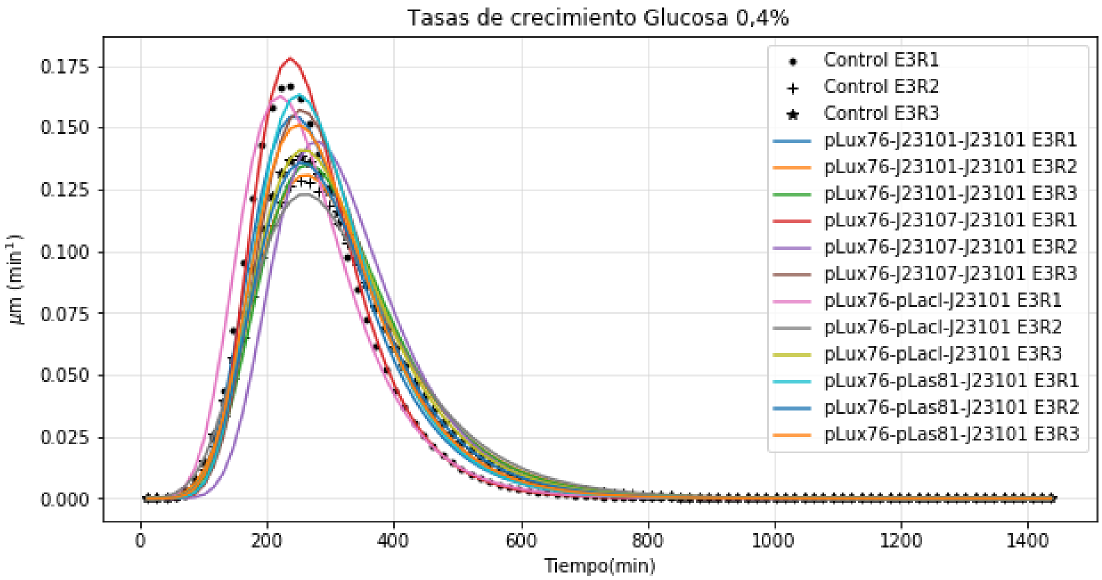

```
In [133]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLa81-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy28,label='growth rate E3R1')
          plt.plot(tt[:-1],dy29,label='growth rate E3R2')
          plt.plot(tt[:-1],dy30,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[133]: <matplotlib.legend.Legend at 0x1d729f3aef0>
```




```
In [134]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLas81-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy26,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy27,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy28,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy29,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy30,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[134]: <matplotlib.legend.Legend at 0x1d72a034c18>
```

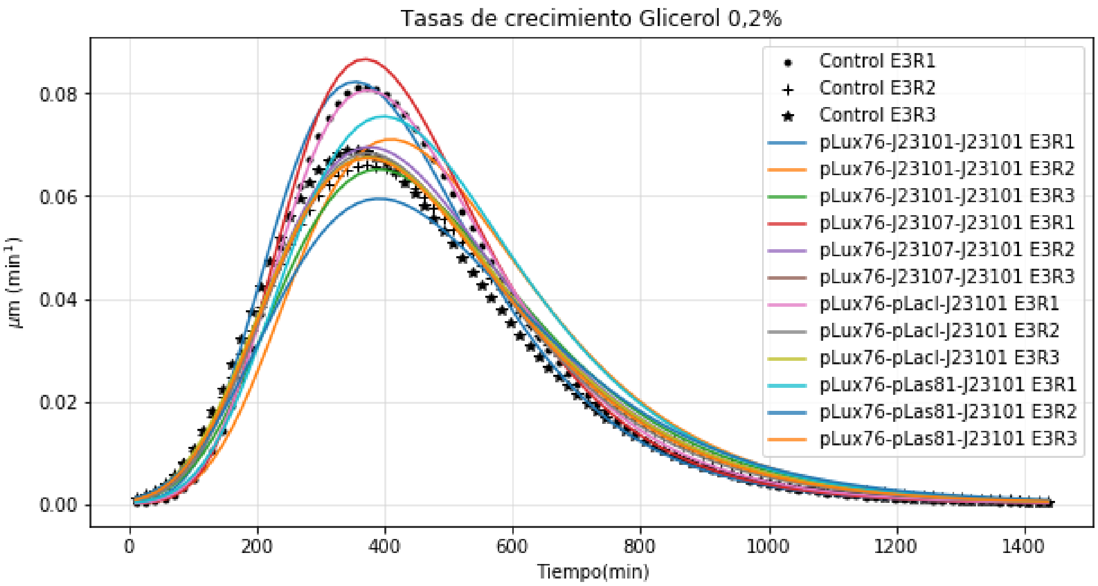



```
In [135]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,'k.',label='Control E3R1')
          plt.plot(tt[:-1],dy2,'k+',label='Control E3R2')
          plt.plot(tt[:-1],dy3,'k*',label='Control E3R3')
          plt.plot(tt[:-1],dy7,label='pLux76-J23101-J23101 E3R1')
          plt.plot(tt[:-1],dy8,label='pLux76-J23101-J23101 E3R2')
          plt.plot(tt[:-1],dy9,label='pLux76-J23101-J23101 E3R3')
          plt.plot(tt[:-1],dy13,label='pLux76-J23107-J23101 E3R1')
          plt.plot(tt[:-1],dy14,label='pLux76-J23107-J23101 E3R2')
          plt.plot(tt[:-1],dy15,label='pLux76-J23107-J23101 E3R3')
          plt.plot(tt[:-1],dy19,label='pLux76-pLacI-J23101 E3R1')
          plt.plot(tt[:-1],dy20,label='pLux76-pLacI-J23101 E3R2')
          plt.plot(tt[:-1],dy21,label='pLux76-pLacI-J23101 E3R3')
          plt.plot(tt[:-1],dy25,label='pLux76-pLas81-J23101 E3R1')
          plt.plot(tt[:-1],dy26,label='pLux76-pLas81-J23101 E3R2')
          plt.plot(tt[:-1],dy27,label='pLux76-pLas81-J23101 E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[135]: <matplotlib.legend.Legend at 0x1d72a144ac8>
```



In [136]:
```
#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='pLux76-J23101-J23101 E3R1')
plt.plot(tt[:-1],dy11,label='pLux76-J23101-J23101 E3R2')
plt.plot(tt[:-1],dy12,label='pLux76-J23101-J23101 E3R3')
plt.plot(tt[:-1],dy16,label='pLux76-J23107-J23101 E3R1')
plt.plot(tt[:-1],dy17,label='pLux76-J23107-J23101 E3R2')
plt.plot(tt[:-1],dy18,label='pLux76-J23107-J23101 E3R3')
plt.plot(tt[:-1],dy22,label='pLux76-pLacI-J23101 E3R1')
plt.plot(tt[:-1],dy23,label='pLux76-pLacI-J23101 E3R2')
plt.plot(tt[:-1],dy24,label='pLux76-pLacI-J23101 E3R3')
plt.plot(tt[:-1],dy28,label='pLux76-pLas81-J23101 E3R1')
plt.plot(tt[:-1],dy29,label='pLux76-pLas81-J23101 E3R2')
plt.plot(tt[:-1],dy30,label='pLux76-pLas81-J23101 E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[136]: <matplotlib.legend.Legend at 0x1d72a2b6978>




```
In [137]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Ensayo 3')
          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='pLux76-J23101-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy8,label='pLux76-J23101-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy9,label='pLux76-J23101-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy13,label='pLux76-J23107-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy14,label='pLux76-J23107-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy15,label='pLux76-J23107-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy19,label='pLux76-pLacI-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy20,label='pLux76-pLacI-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy21,label='pLux76-pLacI-J23101 E3R3 glucosa')
          plt.plot(tt[:-1],dy25,label='pLux76-pLas81-J23101 E3R1 glucosa')
          plt.plot(tt[:-1],dy26,label='pLux76-pLas81-J23101 E3R2 glucosa')
          plt.plot(tt[:-1],dy27,label='pLux76-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='pLux76-J23101-J23101 E3R1 glicerol')
          plt.plot(tt[:-1],dy11,'--',label='pLux76-J23101-J23101 E3R2 glicerol')
          plt.plot(tt[:-1],dy12,'--',label='pLux76-J23101-J23101 E3R3 glicerol')
          plt.plot(tt[:-1],dy16,'--',label='pLux76-J23107-J23101 E3R1 glicerol')
          plt.plot(tt[:-1],dy17,'--',label='pLux76-J23107-J23101 E3R2 glicerol')
```

```
plt.plot(tt[:-1],dy18,'--',label='pLux76-J23107-J23101 E3R3 glicerol')
plt.plot(tt[:-1],dy22,'--',label='pLux76-pLacI-J23101 E3R1 glicerol')
plt.plot(tt[:-1],dy23,'--',label='pLux76-pLacI-J23101 E3R2 glicerol')
plt.plot(tt[:-1],dy24,'--',label='pLux76-pLacI-J23101 E3R3 glicerol')
plt.plot(tt[:-1],dy28,'--',label='pLux76-pLas81-J23101 E3R1 glicerol')
plt.plot(tt[:-1],dy29,'--',label='pLux76-pLas81-J23101 E3R2 glicerol')
plt.plot(tt[:-1],dy30,'--',label='pLux76-pLas81-J23101 E3R3 glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[137]: <matplotlib.legend.Legend at 0x1d72a3c46d8>
```