Ensayo 1 todo

February 13, 2018

```
In [6]: 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 [7]: tt=np.fromfile('t', sep=',')

        #arrays replicas glucosa
        cfpsss1=np.fromfile('psssgCFP1', sep=',')
        rfpsss1=np.fromfile('psssgRFP1', sep=',')
        yfpsss1=np.fromfile('psssgYFP1', sep=',')
        odsss1=np.fromfile('psssgOD1', sep=',')
        cfpsss2=np.fromfile('psssgCFP2', sep=',')
        rfpsss2=np.fromfile('psssgRFP2', sep=',')
        yfpsss2=np.fromfile('psssgYFP2', sep=',')
        odsss2=np.fromfile('psssgOD2', sep=',')
        cfpsss3=np.fromfile('psssgCFP3', sep=',')
        rfpsss3=np.fromfile('psssgRFP3', sep=',')
        yfpsss3=np.fromfile('psssgYFP3', sep=',')
        odsss3=np.fromfile('psssgOD3', sep=',')
        '''
        print(cfpsss1.shape)
        print(rfpsss1.shape)
        print(yfpsss1.shape)
        print(odsss1.shape)
        print(cfpsss2.shape)
        print(rfpsss2.shape)
        print(yfpsss2.shape)
        print(odsss2.shape)
        print(cfpsss3.shape)
        print(rfpsss3.shape)
        print(yfpsss3.shape)
```

```
print(odsss3.shape)'''


cfppss1=np.fromfile('ppssgCFP1', sep=',')
rfppss1=np.fromfile('ppssgRFP1', sep=',')
yfppss1=np.fromfile('ppssgYFP1', sep=',')
odpss1=np.fromfile('ppssgOD1', sep=',')
cfppss2=np.fromfile('ppssgCFP2', sep=',')
rfppss2=np.fromfile('ppssgRFP2', sep=',')
yfppss2=np.fromfile('ppssgYFP2', sep=',')
odpss2=np.fromfile('ppssgOD2', sep=',')
cfppss3=np.fromfile('ppssgCFP3', sep=',')
rfppss3=np.fromfile('ppssgRFP3', sep=',')
yfppss3=np.fromfile('ppssgYFP3', sep=',')
odpss3=np.fromfile('ppssgOD3', sep=',')
'''
print(cfppss1.shape)
print(rfppss1.shape)
print(yfppss1.shape)
print(odpss1.shape)
print(cfppss2.shape)
print(rfppss2.shape)
print(yfppss2.shape)
print(odpss2.shape)
print(cfppss3.shape)
print(rfppss3.shape)
print(yfppss3.shape)
print(odpss3.shape)'''

cfppts1=np.fromfile('pptsgCFP1', sep=',')
rfppts1=np.fromfile('pptsgRFP1', sep=',')
yfppts1=np.fromfile('pptsgYFP1', sep=',')
odpts1=np.fromfile('pptsgOD1', sep=',')
cfppts2=np.fromfile('pptsgCFP2', sep=',')
rfppts2=np.fromfile('pptsgRFP2', sep=',')
yfppts2=np.fromfile('pptsgYFP2', sep=',')
odpts2=np.fromfile('pptsgOD2', sep=',')
cfppts3=np.fromfile('pptsgCFP3', sep=',')
rfppts3=np.fromfile('pptsgRFP3', sep=',')
yfppts3=np.fromfile('pptsgYFP3', sep=',')
odpts3=np.fromfile('pptsgOD3', sep=',')
'''
print(cfppts1.shape)
print(rfppts1.shape)
print(yfppts1.shape)
print(odpts1.shape)
print(cfppts2.shape)
print(rfppts2.shape)
```

```
print(yfppts2.shape)
print(odpts2.shape)
print(cfppts3.shape)
print(rfppts3.shape)
print(yfppts3.shape)
print(odpts3.shape)'''

cfppls1=np.fromfile('pplsgCFP1', sep=',')
rfppls1=np.fromfile('pplsgRFP1', sep=',')
yfppls1=np.fromfile('pplsgYFP1', sep=',')
odpls1=np.fromfile('pplsgOD1', sep=',')
cfppls2=np.fromfile('pplsgCFP2', sep=',')
rfppls2=np.fromfile('pplsgRFP2', sep=',')
yfppls2=np.fromfile('pplsgYFP2', sep=',')
odpls2=np.fromfile('pplsgOD2', sep=',')
cfppls3=np.fromfile('pplsgCFP3', sep=',')
rfppls3=np.fromfile('pplsgRFP3', sep=',')
yfppls3=np.fromfile('pplsgYFP3', sep=',')
odpls3=np.fromfile('pplsgOD3', sep=',')

'''
print(cfppls1.shape)
print(rfppls1.shape)
print(yfppls1.shape)
print(odpls1.shape)
print(cfppls2.shape)
print(rfppls2.shape)
print(yfppls2.shape)
print(odpls2.shape)
print(cfppls3.shape)
print(rfppls3.shape)
print(yfppls3.shape)
print(odpls3.shape)'''

#Controles
#Promedios controles glucosa
cfpcg1=np.fromfile('pcgCFP1', sep=',')
rfpcg1=np.fromfile('pcgRFP1', sep=',')
yfpcg1=np.fromfile('pcgYFP1', sep=',')
odcg1=np.fromfile('pcgOD1', sep=',')
cfpcg2=np.fromfile('pcgCFP2', sep=',')
rfpcg2=np.fromfile('pcgRFP2', sep=',')
yfpcg2=np.fromfile('pcgYFP2', sep=',')
odcg2=np.fromfile ('pcgOD2',sep=',')
cfpcg3=np.fromfile('pcgCFP3', sep=',')
rfpcg3=np.fromfile('pcgRFP3', sep=',')
yfpcg3=np.fromfile('pcgYFP3', sep=',')
odcg3=np.fromfile('pcgOD3', sep=',')
```

```
'''
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)'''
```

Out[7]: '\nprint(cfpcg1.shape)\nprint(rfpcg1.shape)\nprint(yfpcg1.shape)\nprint(odcg1.shape)\npr

In [8]:
```
#Promedios glicerol
#arrays replicas glucosa
cfpsssg1=np.fromfile('psssglCFP1', sep=',')
rfpsssg1=np.fromfile('psssglRFP1', sep=',')
yfpsssg1=np.fromfile('psssglYFP1', sep=',')
odsssg1=np.fromfile('psssglOD1', sep=',')
cfpsssg2=np.fromfile('psssglCFP2', sep=',')
rfpsssg2=np.fromfile('psssglRFP2', sep=',')
yfpsssg2=np.fromfile('psssglYFP2', sep=',')
odsssg2=np.fromfile('psssglOD2', sep=',')
cfpsssg3=np.fromfile('psssglCFP3', sep=',')
rfpsssg3=np.fromfile('psssglRFP3', sep=',')
yfpsssg3=np.fromfile('psssglYFP3', sep=',')
odsssg3=np.fromfile('psssglOD3', sep=',')
'''
print(cfpsssg1.shape)
print(rfpsssg1.shape)
print(yfpsssg1.shape)
print(odsssg1.shape)
print(cfpsssg2.shape)
print(rfpsssg2.shape)
print(yfpsssg2.shape)
print(odsssg2.shape)
print(cfpsssg3.shape)
print(rfpsssg3.shape)
print(yfpsssg3.shape)
print(odsssg3.shape)'''

cfppssg1=np.fromfile('ppssglCFP1', sep=',')
rfppssg1=np.fromfile('ppssglRFP1', sep=',')
yfppssg1=np.fromfile('ppssglYFP1', sep=',')
odpssg1=np.fromfile('ppssglOD1', sep=',')
```

```
cfppssg2=np.fromfile('ppssglCFP2', sep=',')
rfppssg2=np.fromfile('ppssglRFP2', sep=',')
yfppssg2=np.fromfile('ppssglYFP2', sep=',')
odpssg2=np.fromfile('ppssglOD2', sep=',')
cfppssg3=np.fromfile('ppssglCFP3', sep=',')
rfppssg3=np.fromfile('ppssglRFP3', sep=',')
yfppssg3=np.fromfile('ppssglYFP3', sep=',')
odpssg3=np.fromfile('ppssglOD3', sep=',')
'''
print(cfppssg1.shape)
print(rfppssg1.shape)
print(yfppssg1.shape)
print(odpssg1.shape)
print(cfppssg2.shape)
print(rfppssg2.shape)
print(yfppssg2.shape)
print(odpssg2.shape)
print(cfppssg3.shape)
print(rfppssg3.shape)
print(yfppssg3.shape)
print(odpssg3.shape)'''

cfpptsg1=np.fromfile('pptsglCFP1', sep=',')
rfpptsg1=np.fromfile('pptsglRFP1', sep=',')
yfpptsg1=np.fromfile('pptsglYFP1', sep=',')
odptsg1=np.fromfile('pptsglOD1', sep=',')
cfpptsg2=np.fromfile('pptsglCFP2', sep=',')
rfpptsg2=np.fromfile('pptsglRFP2', sep=',')
yfpptsg2=np.fromfile('pptsglYFP2', sep=',')
odptsg2=np.fromfile('pptsglOD2', sep=',')
cfpptsg3=np.fromfile('pptsglCFP3', sep=',')
rfpptsg3=np.fromfile('pptsglRFP3', sep=',')
yfpptsg3=np.fromfile('pptsglYFP3', sep=',')
odptsg3=np.fromfile('pptsglOD3', sep=',')

'''
print(cfpptsg1.shape)
print(rfpptsg1.shape)
print(yfpptsg1.shape)
print(odptsg1.shape)
print(cfpptsg2.shape)
print(rfpptsg2.shape)
print(yfpptsg2.shape)
print(odptsg2.shape)
print(cfpptsg3.shape)
print(rfpptsg3.shape)
print(yfpptsg3.shape)
print(odptsg3.shape)'''
```

```python
cfpplsg1=np.fromfile('pplsglCFP1', sep=',')
rfpplsg1=np.fromfile('pplsglRFP1', sep=',')
yfpplsg1=np.fromfile('pplsglYFP1', sep=',')
odplsg1=np.fromfile('pplsglOD1', sep=',')
cfpplsg2=np.fromfile('pplsglCFP2', sep=',')
rfpplsg2=np.fromfile('pplsglRFP2', sep=',')
yfpplsg2=np.fromfile('pplsglYFP2', sep=',')
odplsg2=np.fromfile('pplsglOD2', sep=',')
cfpplsg3=np.fromfile('pplsglCFP3', sep=',')
rfpplsg3=np.fromfile('pplsglRFP3', sep=',')
yfpplsg3=np.fromfile('pplsglYFP3', sep=',')
odplsg3=np.fromfile('pplsglOD3', sep=',')
'''
print(cfpplsg1.shape)
print(rfpplsg1.shape)
print(yfpplsg1.shape)
print(odplsg1.shape)
print(cfpplsg2.shape)
print(rfpplsg2.shape)
print(yfpplsg2.shape)
print(odplsg2.shape)
print(cfpplsg3.shape)
print(rfpplsg3.shape)
print(yfpplsg3.shape)
print(odplsg3.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)'''
```

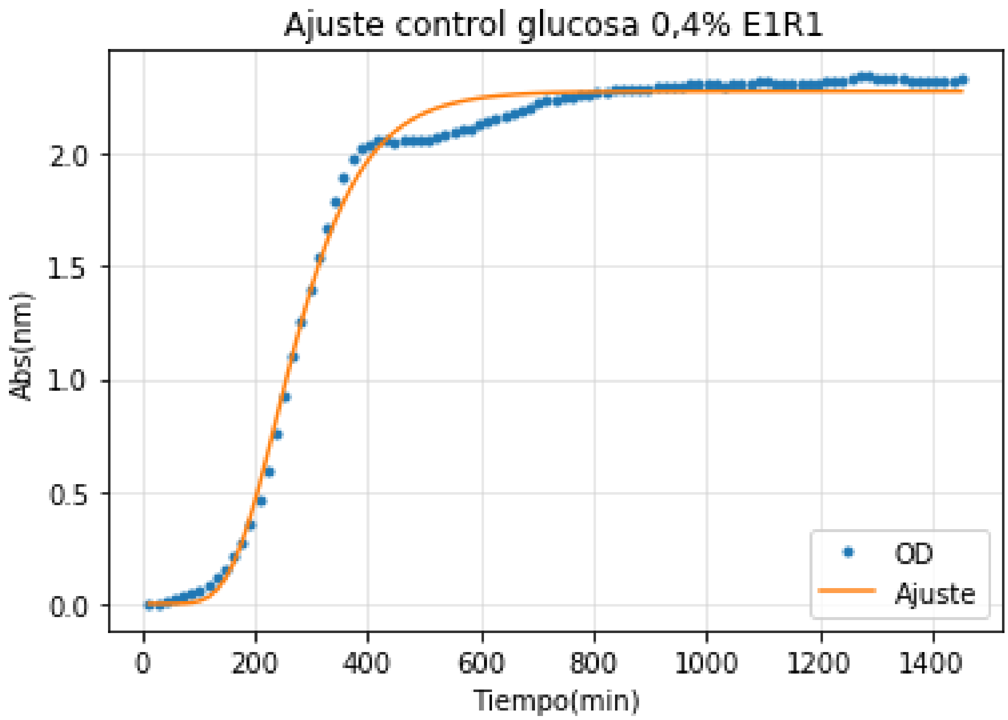
Out[8]: '\nprint(cfpcgl1.shape)\nprint(rfpcgl1.shape)\nprint(yfpcgl1.shape)\nprint(odcgl1.shape)

```
In [9]: #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]),figna
                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')

                if figname != False:
                    plt.savefig(figname, dpi=300, facecolor='w', edgecolor='w',bbox_inches='tigh
                plt.show()

                Y_fit=evalF,z

                return(Y_fit)
```

```
In [12]: #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% E1R1')
         A1 = params[0]
         um1=params[1]
```

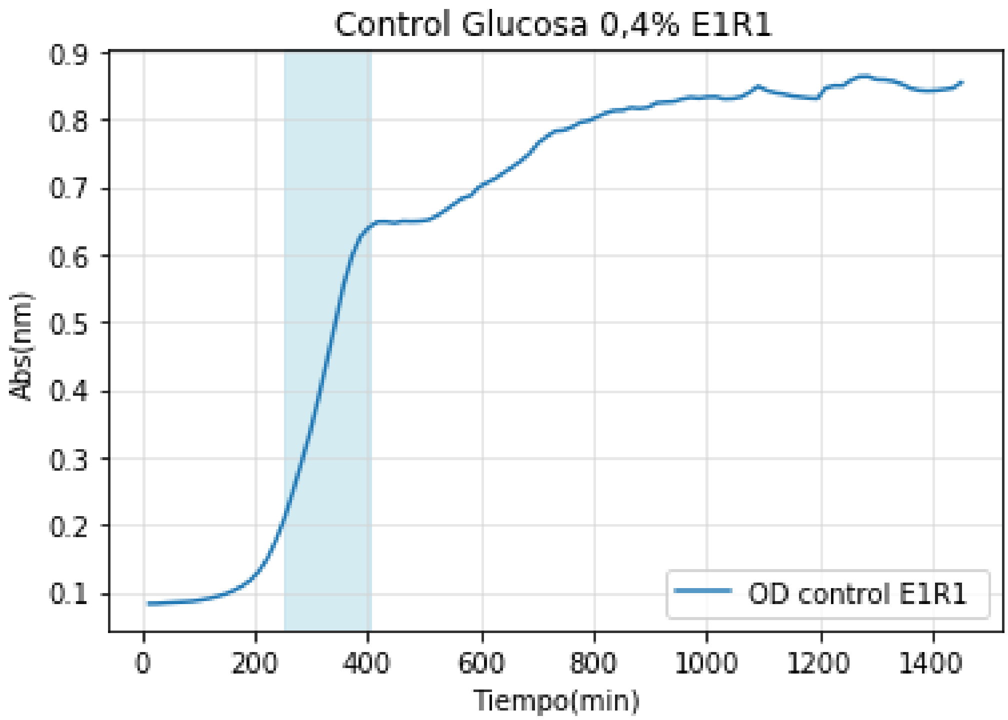
```python
    l1=params[2]
    print('A=%e'%(A1))
    print('um=%e'%(um1))
    print('l=%e'%(l1))

    #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[26]
    plt.figure()
    plt.title('Control Glucosa 0,4% E1R1')
    plt.xlabel('Tiempo(min)')
    plt.ylabel('Abs(nm)')
    plt.plot(tt,odcg1,label='OD control E1R1 ')
    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 E1R1')
    plt.xlabel('Tiempo(min)')
    plt.ylabel('Abs(nm)')
    plt.plot(tt[16:27],odcg1[16:27],label='OD control E1R1')
    plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
    plt.legend(loc='lower right')
```

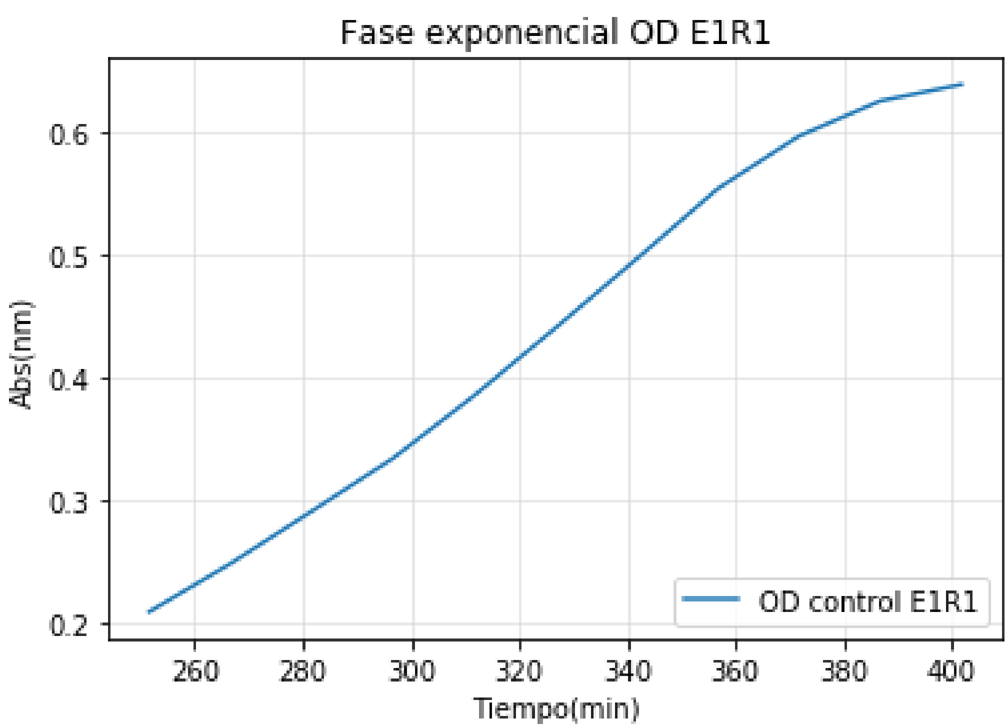
```
Min OD = 8.275000e-02
[  2.27987016e+00   1.00649666e-02   1.55693371e+02]
```



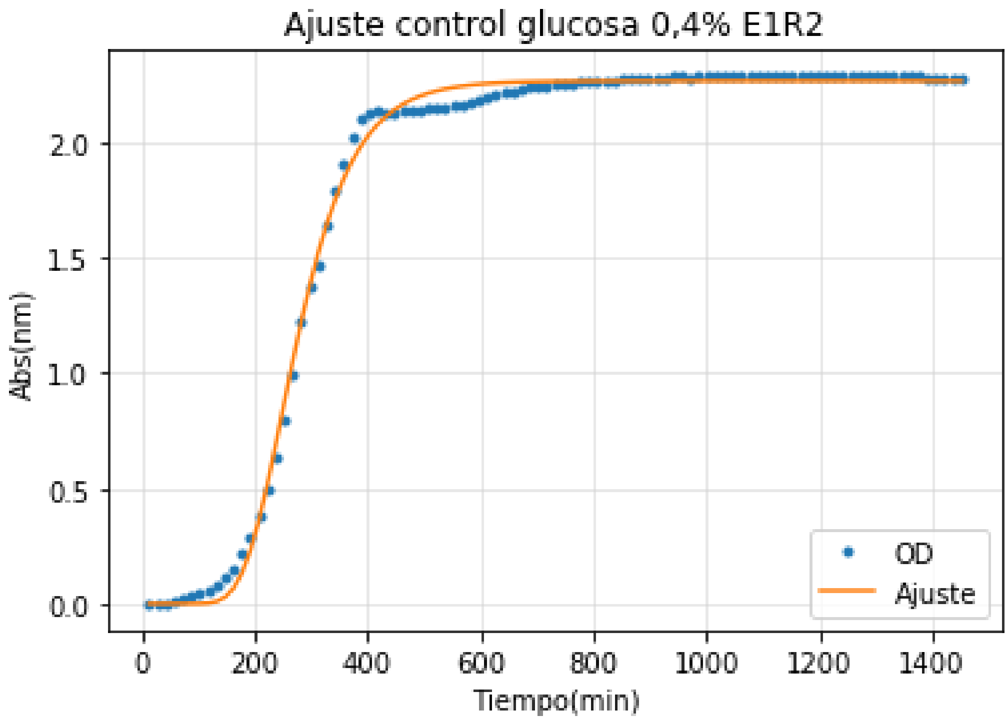

```
A=2.279870e+00
um=1.006497e-02
l=1.556934e+02
Tm=2.390237e+02
doubpe=6.886731e+01
ext=1.377346e+02
Tfinal=3.767584e+02
```

```
Out[12]: <matplotlib.legend.Legend at 0x1aba75232e8>
```


Control Glucosa 0,4% E1R1
0.9
0.8
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 E1R1

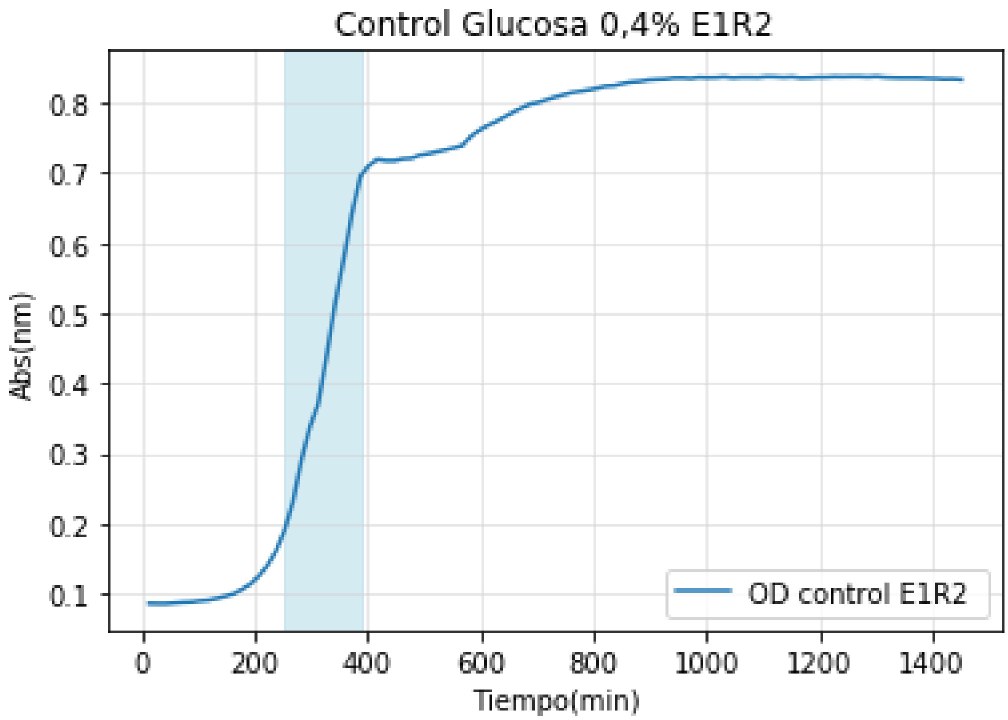

Fase exponencial OD E1R1
0.6
0.5
0.4
0.3
0.2
Abs(nm)
260
280
300
320
340
360
380
400
Tiempo(min)
OD control E1R1

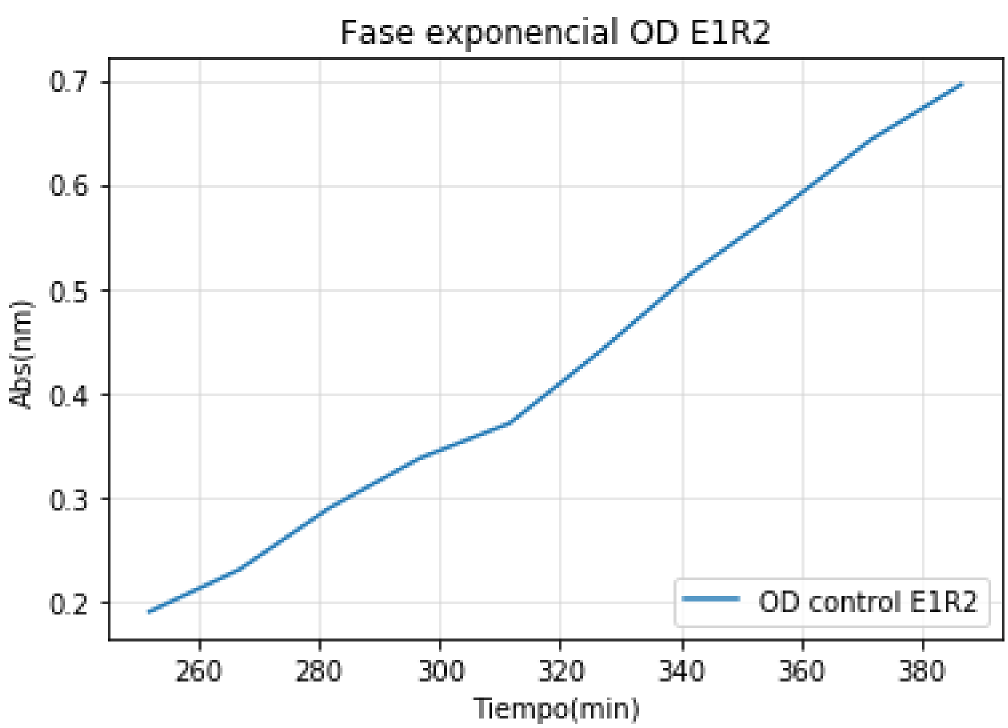
```
In [13]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glucosa rep 2
         y2= np.log(odcg2)-np.log(np.min(odcg2))
         print('Min OD = %e'%((np.min(odcg2))))
         evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E1R2')
         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[16]
         y2=tt[25]
         plt.figure()
         plt.title('Control Glucosa 0,4% E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odcg2,label='OD control E1R2 ')
         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 E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],odcg2[16:26],label='OD control E1R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Min OD = 8.525000e-02
[  2.26624843e+00   1.20912515e-02   1.79347494e+02]
```




```
A=2.266248e+00
um=1.209125e-02
l=1.793475e+02
Tm=2.482987e+02
doubpe=5.732634e+01
ext=1.146527e+02
Tfinal=3.629514e+02
```

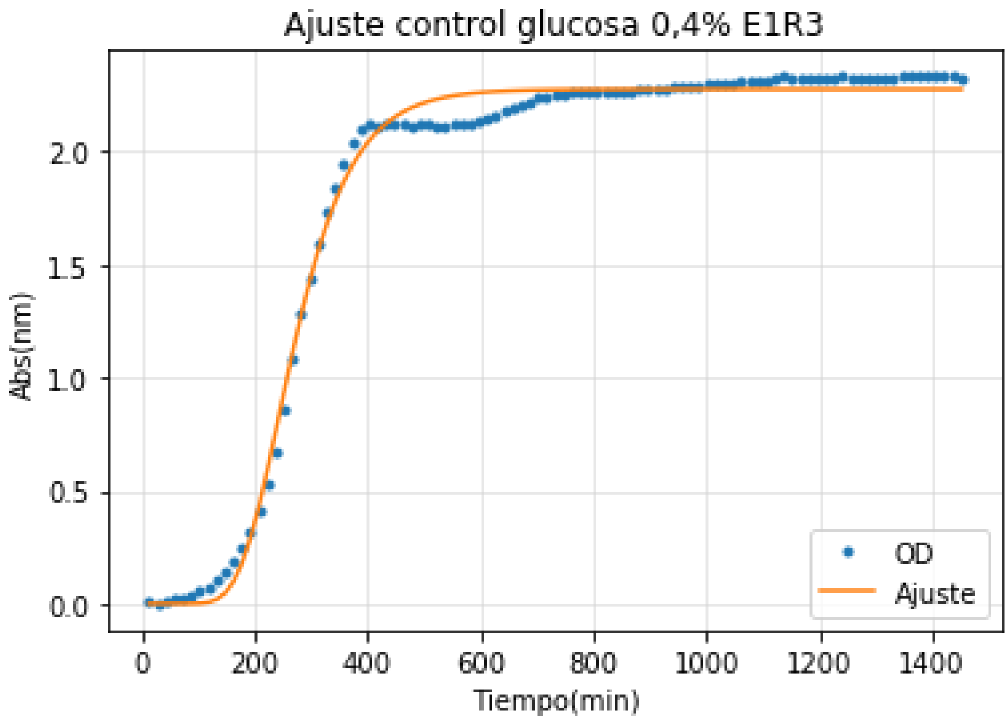
```
Out[13]: <matplotlib.legend.Legend at 0x1aba774e7f0>
```


Control Glucosa 0,4% E1R2
0.8
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 E1R2


Fase exponencial OD E1R2
0.7
0.6
0.5
0.4
0.3
0.2
Abs(nm)
260
280
300
320
340
360
380
Tiempo(min)
OD control E1R2

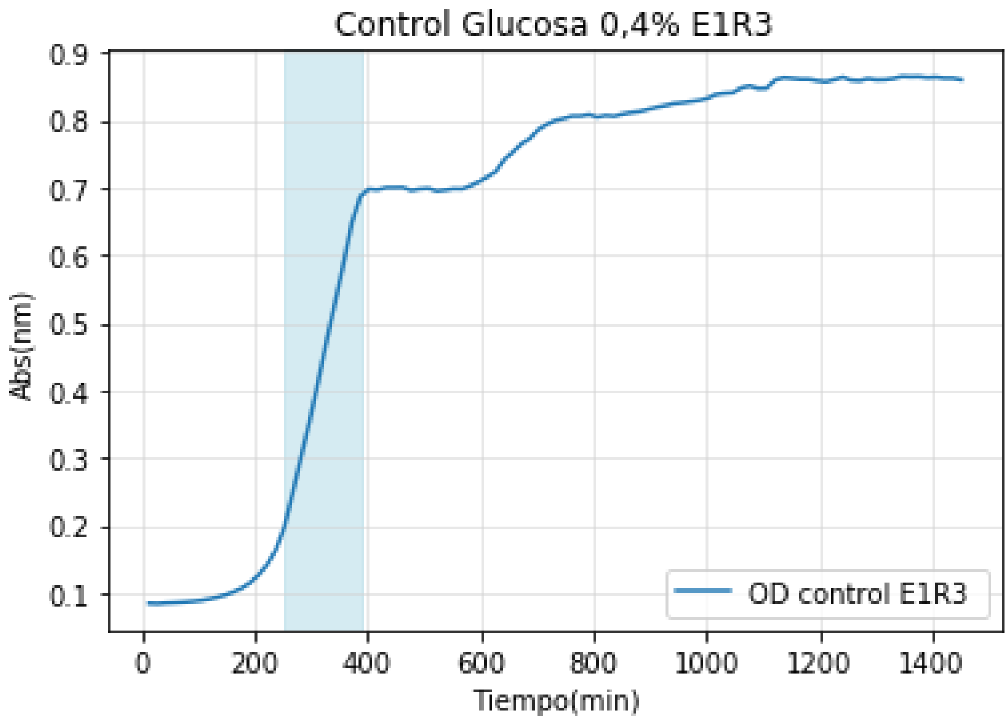
```
In [14]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glucosa rep 3
         y3= np.log(odcg3)-np.log(np.min(odcg3))
         print('Min OD = %e'%((np.min(odcg3))))
         evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E1R3')
         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[16]
         y2=tt[25]
         plt.figure()
         plt.title('Control Glucosa 0,4% E1R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odcg3,label='OD control E1R3 ')
         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 E1R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],odcg3[16:26],label='OD control E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

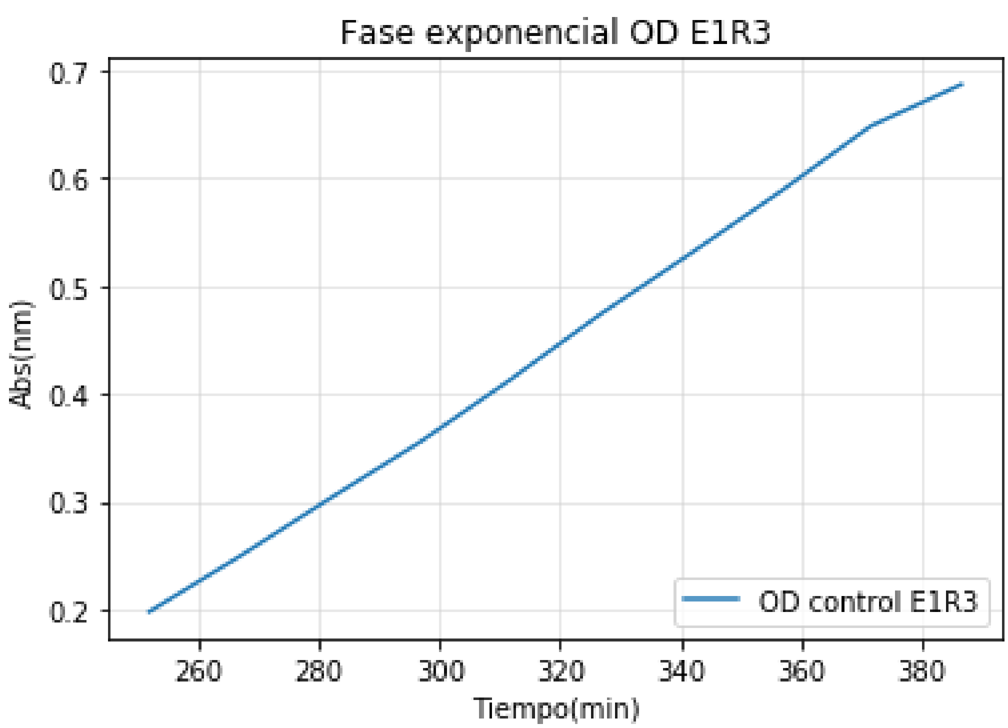
```
Min OD = 8.400000e-02
[  2.27497784e+00   1.18246821e-02   1.71819565e+02]
```




```
A=2.274978e+00
um=1.182468e-02
l=1.718196e+02
Tm=2.425967e+02
doubpe=5.861867e+01
ext=1.172373e+02
Tfinal=3.598341e+02
```

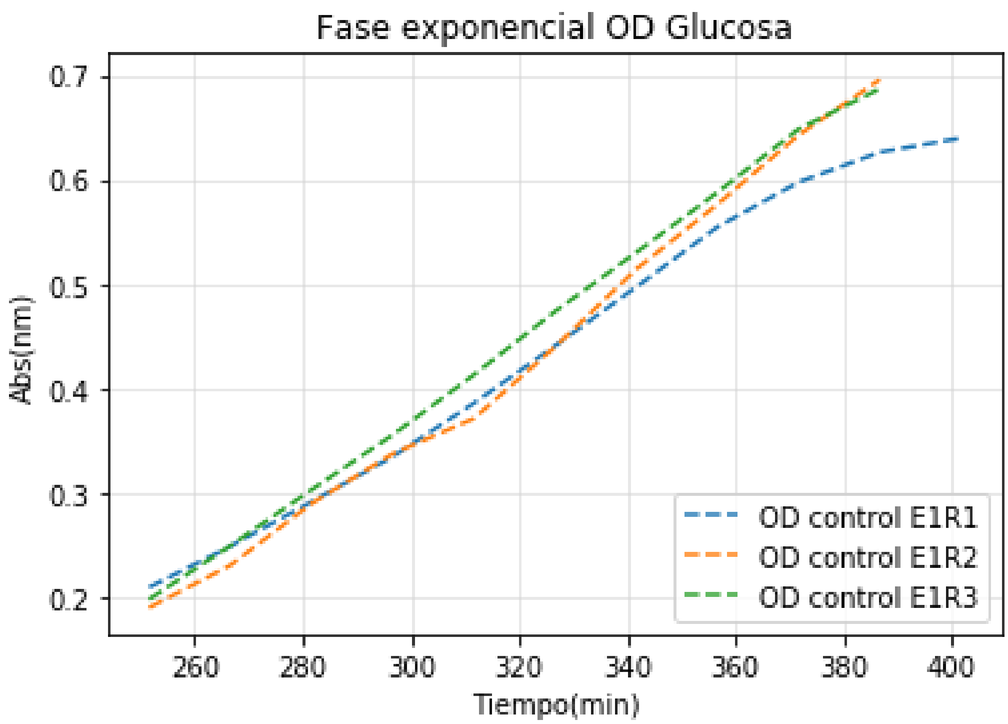
```
Out[14]: <matplotlib.legend.Legend at 0x1aba7952860>
```


Control Glucosa 0,4% E1R3
0.9
0.8
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 E1R3


Fase exponencial OD E1R3
0.7
0.6
0.5
0.4
0.3
0.2
Abs(nm)
260
280
300
320
340
360
380
Tiempo(min)
OD control E1R3

```
In [15]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glucosa')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],odcg1[16:27],'--',label='OD control E1R1')
         plt.plot(tt[16:26],odcg2[16:26],'--',label='OD control E1R2')
         plt.plot(tt[16:26],odcg3[16:26],'--',label='OD control E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

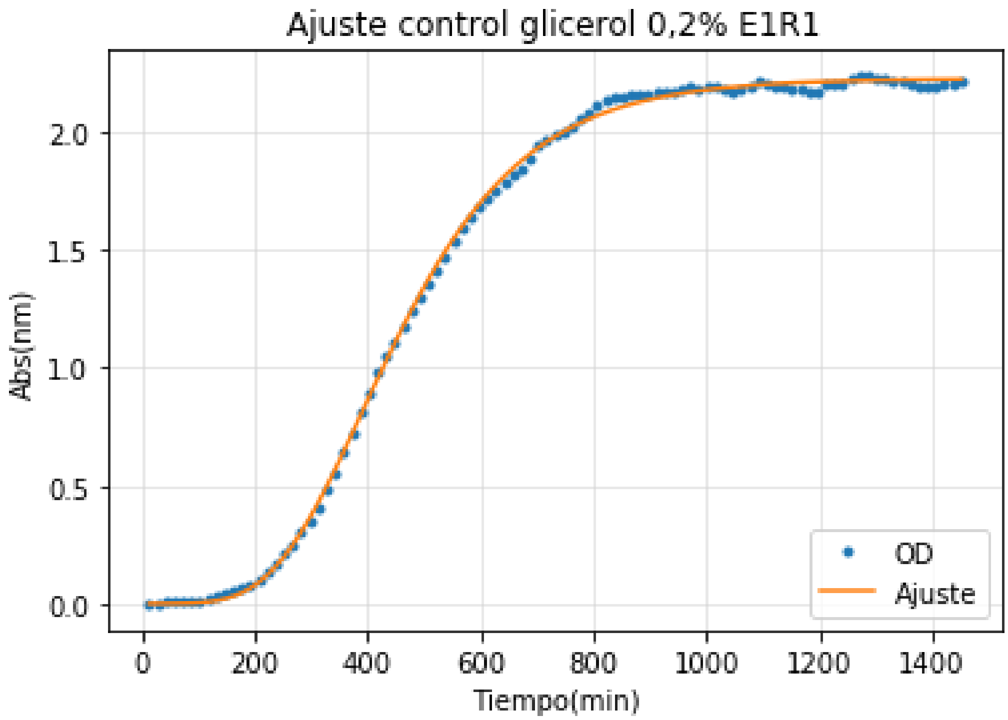
```
Out[15]: <matplotlib.legend.Legend at 0x1aba76eb320>
```




```
In [16]: #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% E1R1')
         A4= params[0]
         um4=params[1]
         l4=params[2]
         print('A=%e'%(A4))
         print('um=%e'%(um4))
         print('l=%e'%(l4))
```

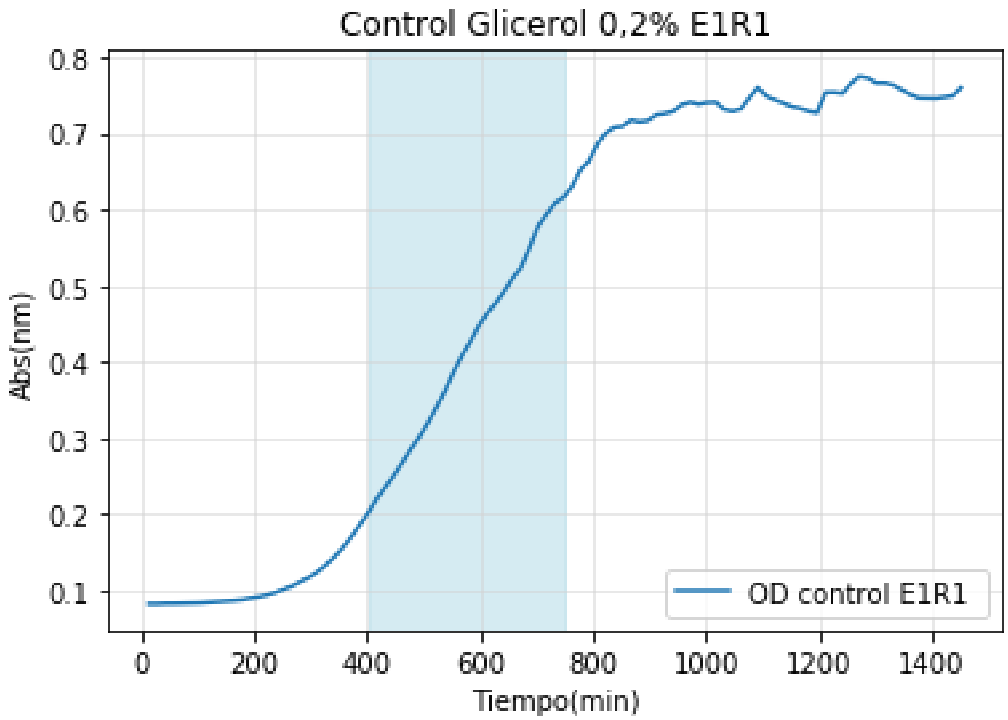
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+l4)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[26]
y2=tt[49]
plt.figure()
plt.title('Control Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl1,label='OD control E1R1 ')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:50],odcgl1[26:50],label='OD control E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

```
Min OD = 8.300000e-02
[  2.22470400e+00   5.15858083e-03   2.32096924e+02]
```

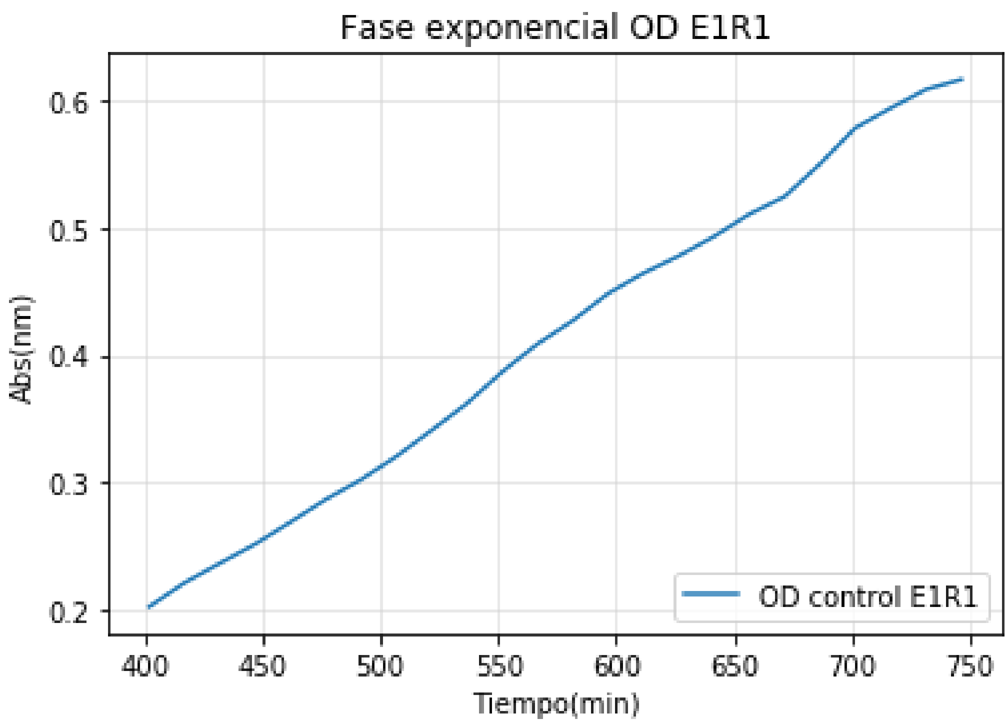



```
A=2.224704e+00
um=5.158581e-03
l=2.320969e+02
Tm=3.907496e+02
doubpe=1.343678e+02
ext=3.359195e+02
Tfinal=7.266692e+02
```

```
Out[16]: <matplotlib.legend.Legend at 0x1aba63feac8>
```



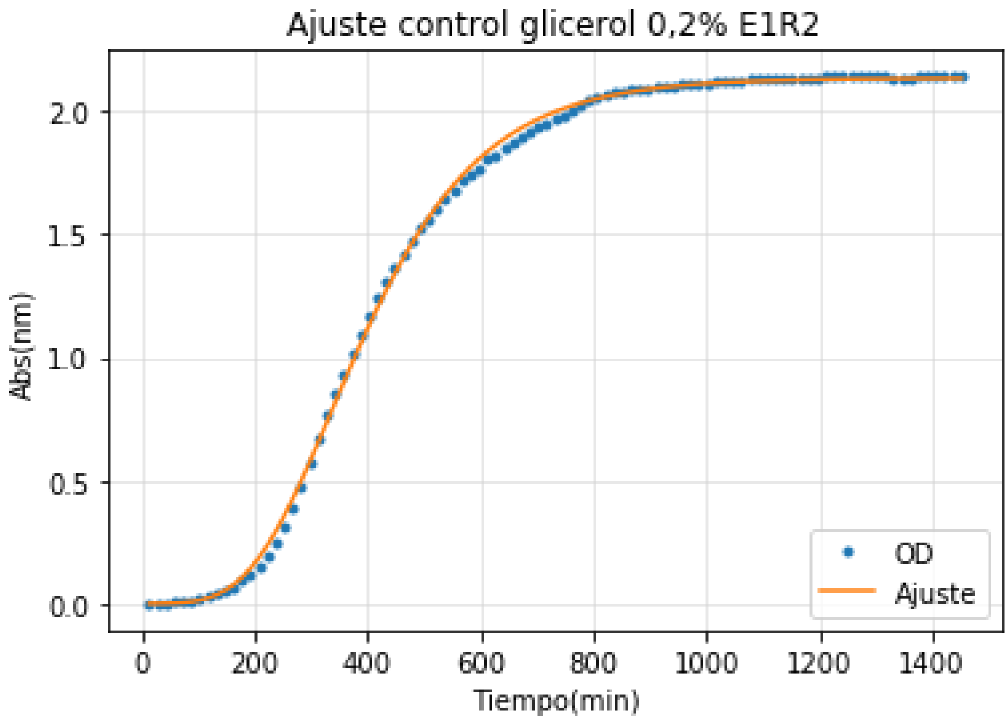
Control Glicerol 0,2% E1R1
Abs(nm)
Tiempo(min)
OD control E1R1



Fase exponencial OD E1R1
Abs(nm)
Tiempo(min)
OD control E1R1

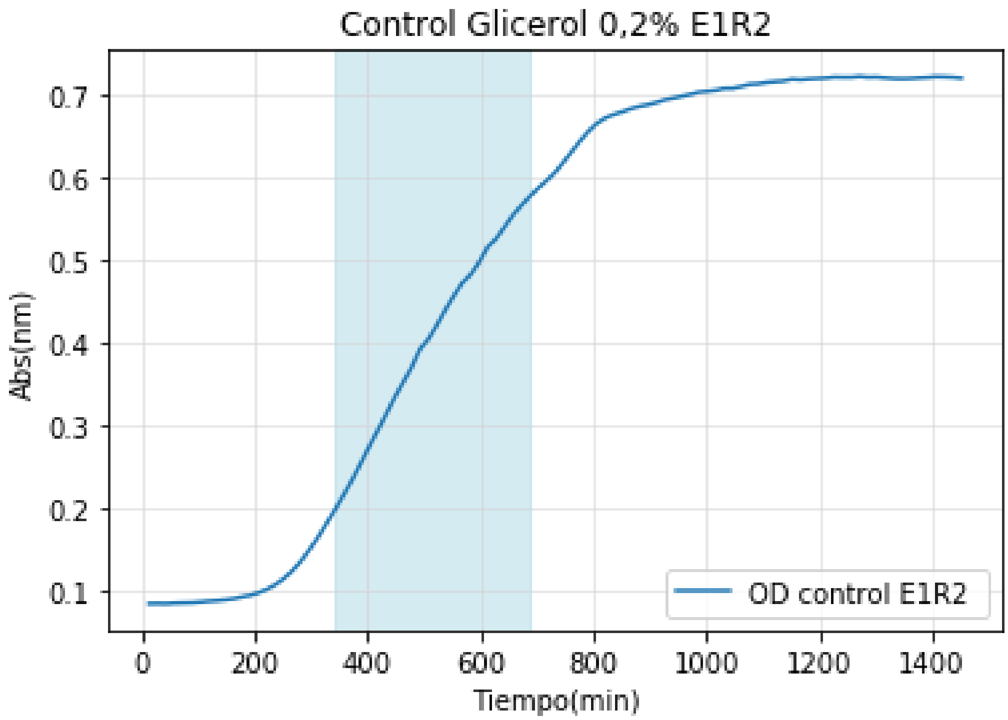
In [17]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 2
y5= np.log(odcgl2)-np.log(np.min(odcgl2))
print('Min OD = %e'%((np.min(odcgl2))))
evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 0,2% E1R2')
A5= params[0]
um5=params[1]
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[22]
y2=tt[45]
plt.figure()
plt.title('Control Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl2,label='OD control E1R2 ')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[22:46],odcgl2[22:46],label='OD control E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

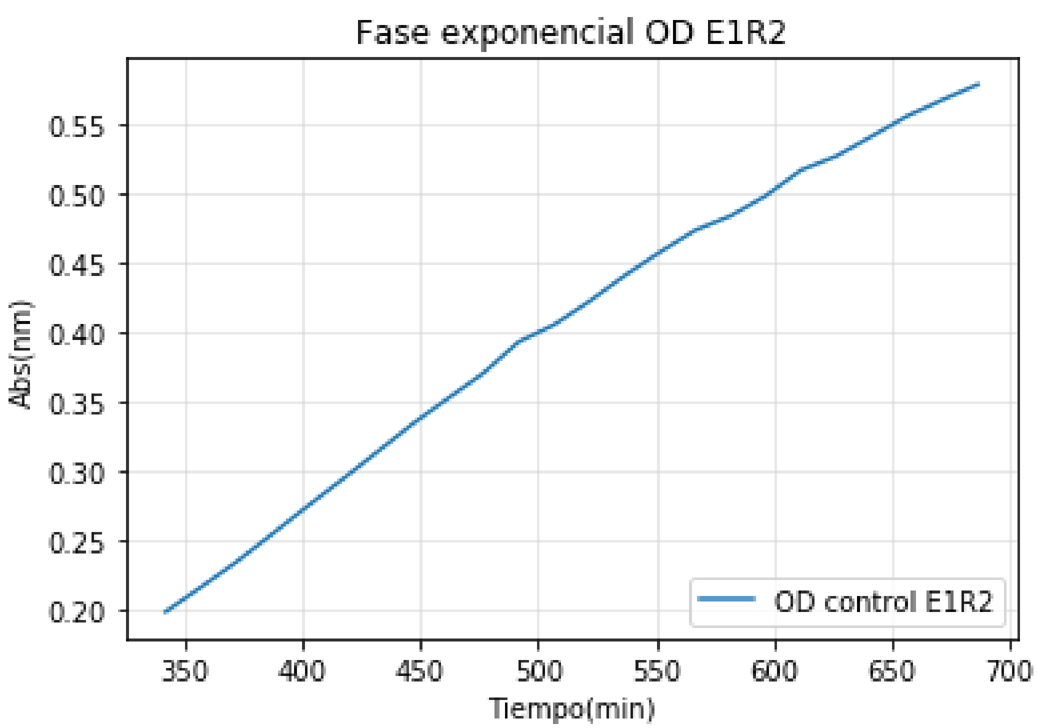
```
Min OD = 8.500000e-02
[  2.13428055e+00   5.39641014e-03   1.90374314e+02]
```




```
A=2.134281e+00
um=5.396410e-03
l=1.903743e+02
Tm=3.358707e+02
doubpe=1.284460e+02
ext=3.211149e+02
Tfinal=6.569856e+02
```

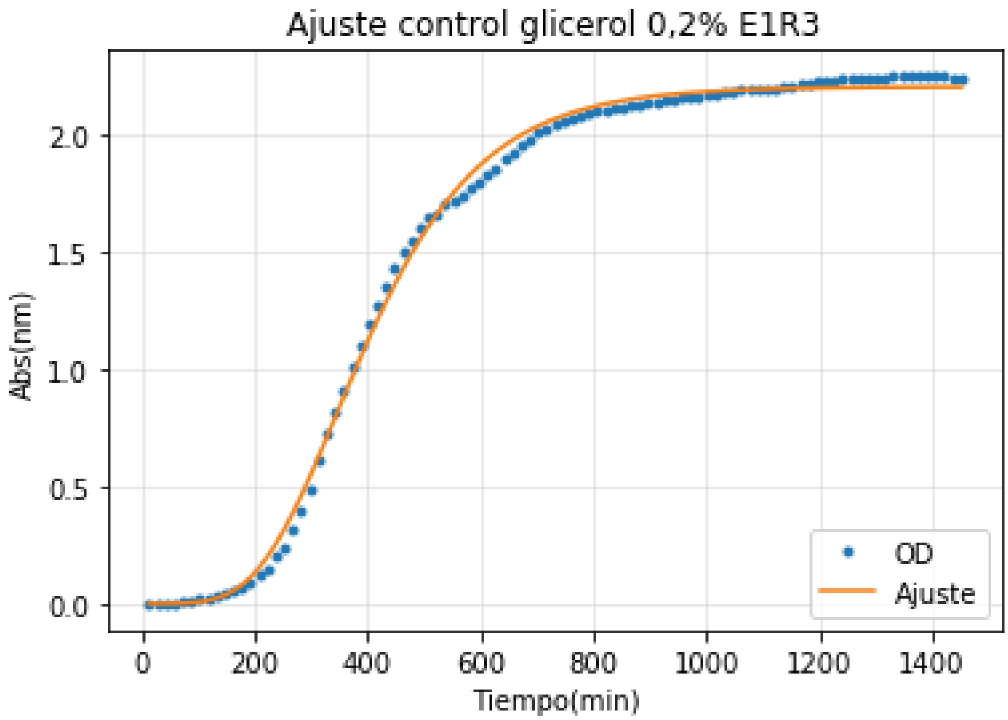
```
Out[17]: <matplotlib.legend.Legend at 0x1aba7501630>
```


Control Glicerol 0,2% E1R2
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 E1R2


Fase exponencial OD E1R2
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
350
400
450
500
550
600
650
700
Tiempo(min)
OD control E1R2

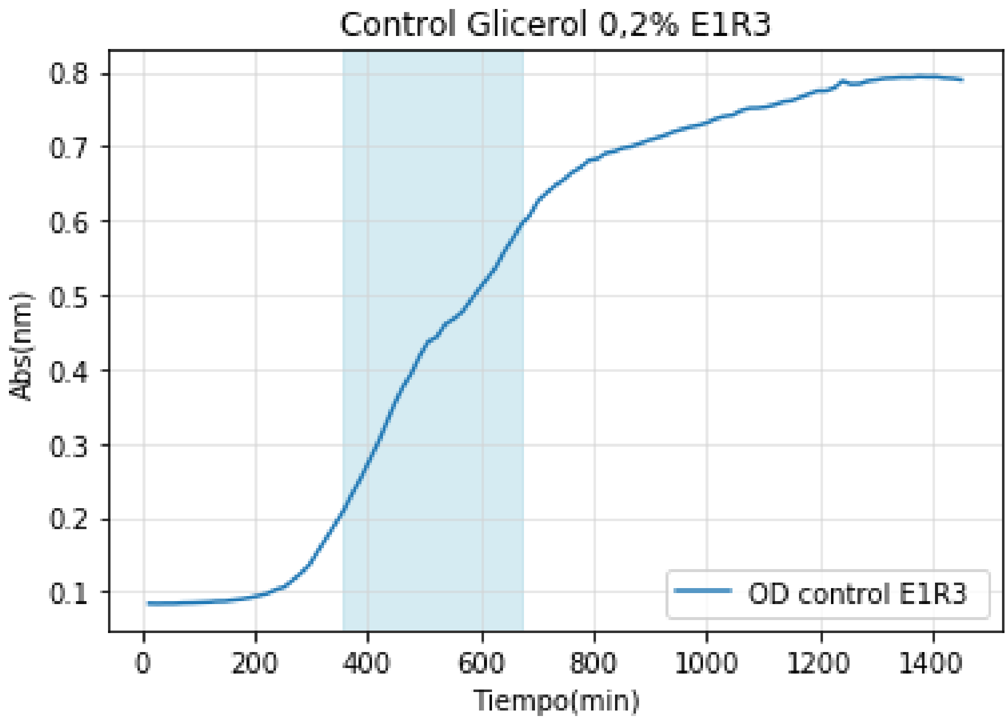
In [18]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 3
y6= np.log(odcgl3)-np.log(np.min(odcgl3))
print('Min OD = %e'%((np.min(odcgl3))))
evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 0,2% E1R3')
A6= params[0]
um6=params[1]
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[23]
y2=tt[44]
plt.figure()
plt.title('Control Glicerol 0,2% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl3,label='OD control E1R3 ')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[23:45],odcgl3[23:45],label='OD control E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

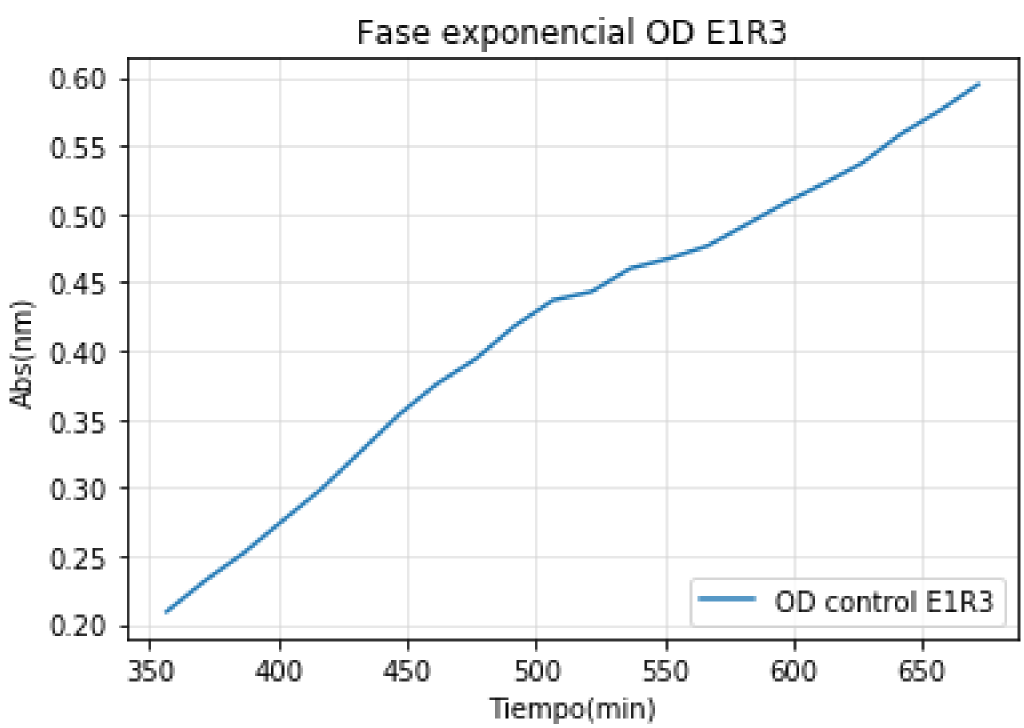
```
Min OD = 8.400000e-02
[  2.20283598e+00   5.76096294e-03   2.03715864e+02]
```



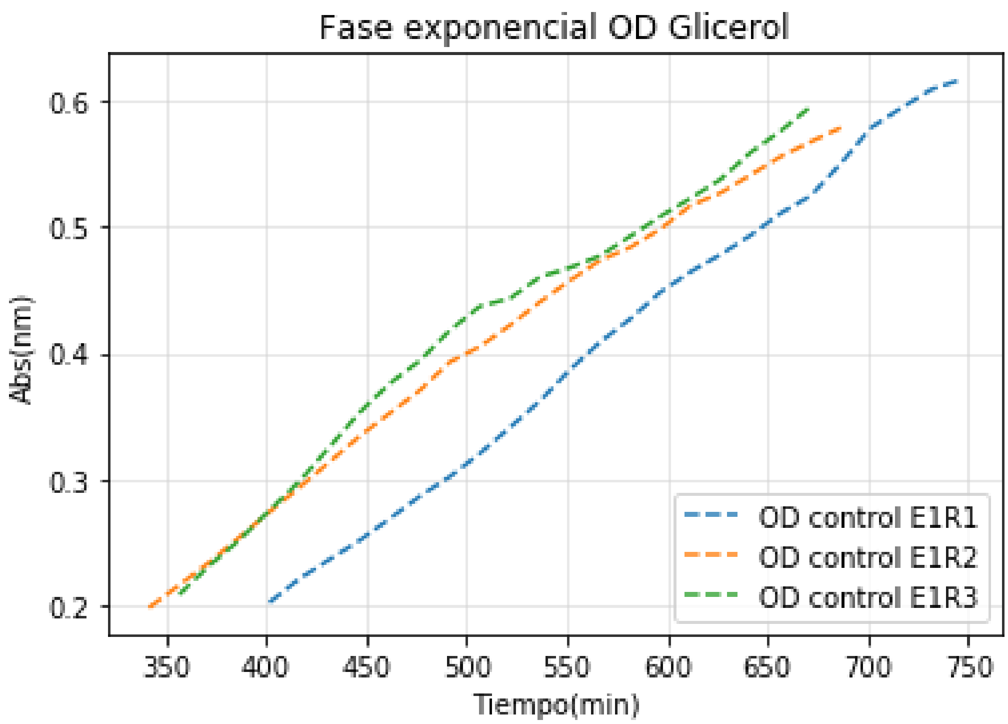

```
A=2.202836e+00
um=5.760963e-03
l=2.037159e+02
Tm=3.443830e+02
doubpe=1.203179e+02
ext=3.007948e+02
Tfinal=6.451778e+02
```

```
Out[18]: <matplotlib.legend.Legend at 0x1aba7b563c8>
```


Control Glicerol 0,2% E1R3
0.8
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 E1R3


Fase exponencial OD E1R3
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
350
400
450
500
550
600
650
Tiempo(min)
OD control E1R3

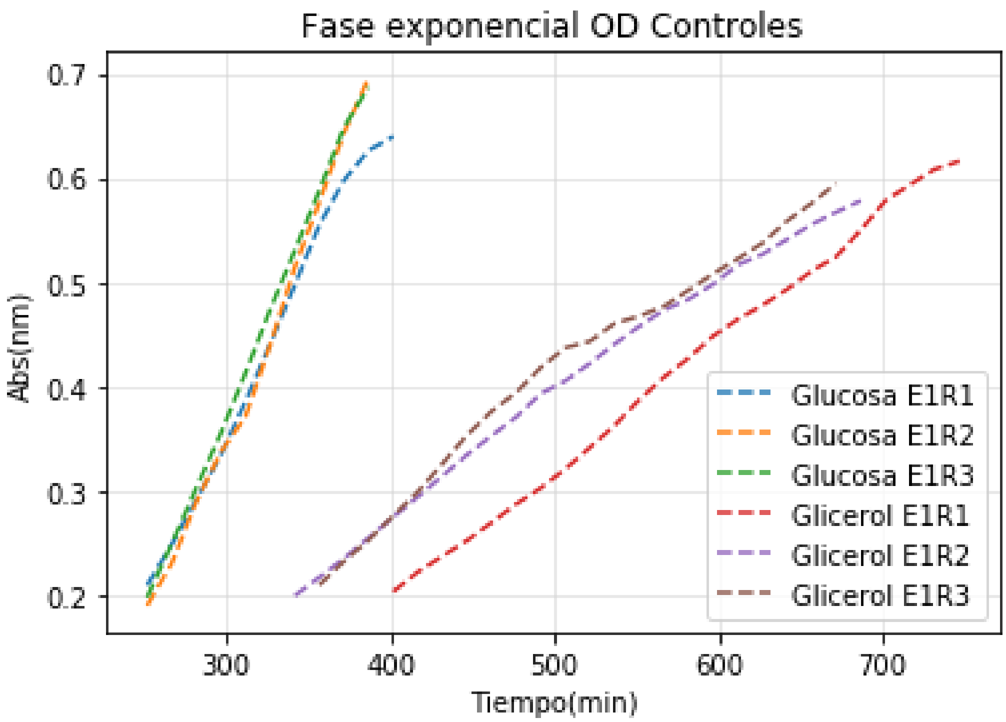
```
In [19]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glicerol')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[26:50],odcgl1[26:50],'--',label='OD control E1R1')
         plt.plot(tt[22:46],odcgl2[22:46],'--',label='OD control E1R2')
         plt.plot(tt[23:45],odcgl3[23:45],'--',label='OD control E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[19]: <matplotlib.legend.Legend at 0x1aba7ac0710>
```




```
In [20]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Controles')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],odcg1[16:27],'--',label='Glucosa E1R1')
         plt.plot(tt[16:26],odcg2[16:26],'--',label='Glucosa E1R2')
         plt.plot(tt[16:26],odcg3[16:26],'--',label='Glucosa E1R3')
         plt.plot(tt[26:50],odcgl1[26:50],'--',label='Glicerol E1R1')
         plt.plot(tt[22:46],odcgl2[22:46],'--',label='Glicerol E1R2')
         plt.plot(tt[23:45],odcgl3[23:45],'--',label='Glicerol E1R3')
```

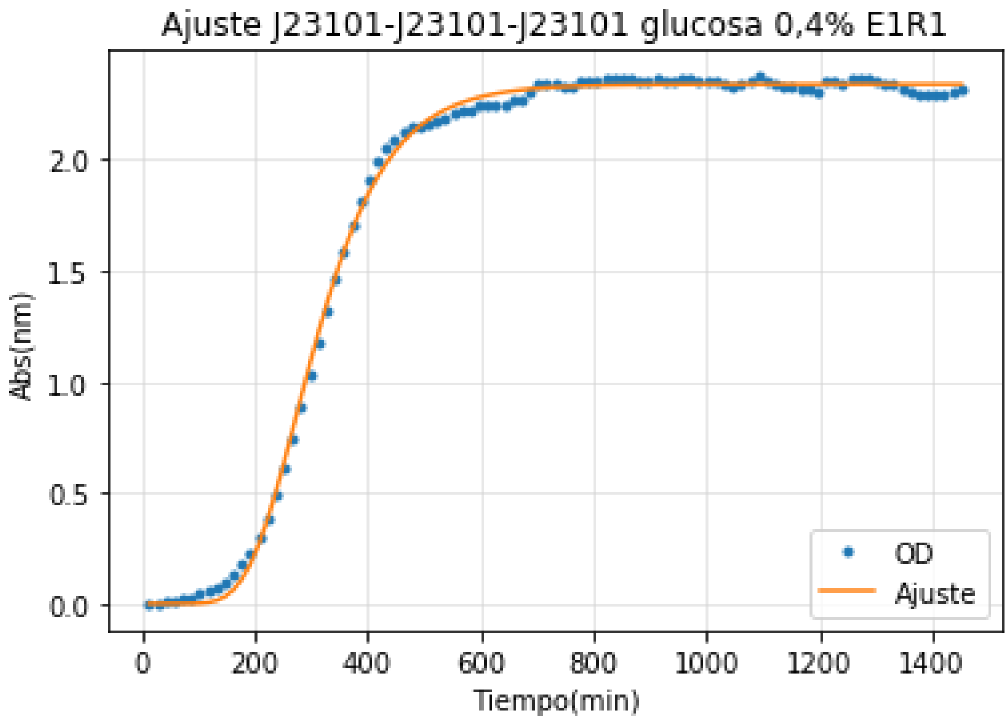
```
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')
```

Out[20]: <matplotlib.legend.Legend at 0x1aba5e8c0b8>

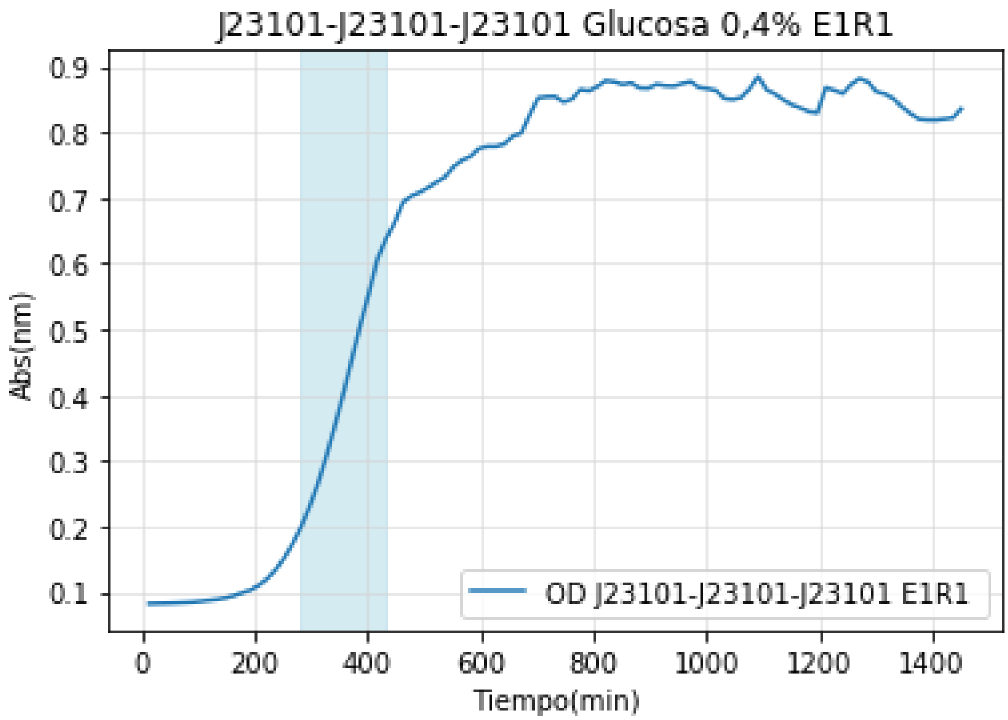


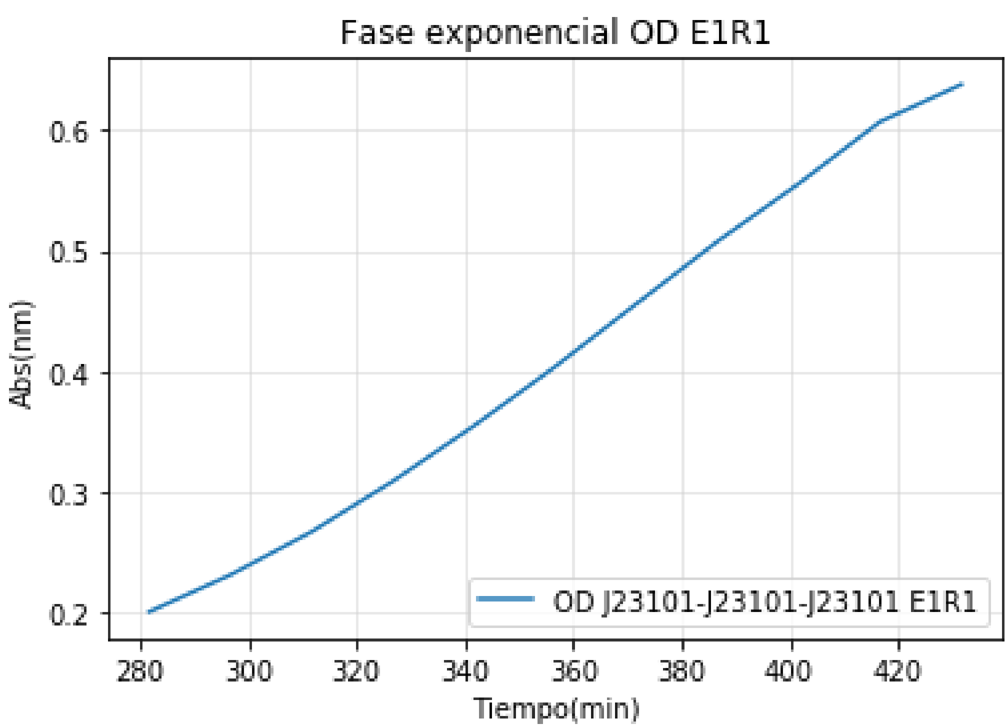

```
In [21]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #std-std-std glucosa rep 1
         y7= np.log(odsss1)-np.log(np.min(odsss1))
         print('Min OD = %e'%((np.min(odsss1))))
         evaly, params=Function_fit(tt,y7,0,-1,figname='Ajuste Stdx3 1.png',title = 'Ajuste J231

         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[18]
    y2=tt[28]
    plt.figure()
    plt.title('J23101-J23101-J23101 Glucosa 0,4% E1R1')
    plt.xlabel('Tiempo(min)')
    plt.ylabel('Abs(nm)')
    plt.plot(tt,odsss1,label='OD J23101-J23101-J23101 E1R1 ')
    plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
    plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
    plt.legend(loc='lower right')
    plt.savefig(filename='delimitacion std-std-std 1.png', dpi=300, facecolor='w', edgecolo

    #Fase exponencial OD/tiempo
    plt.figure()
    plt.title('Fase exponencial OD E1R1')
    plt.xlabel('Tiempo(min)')
    plt.ylabel('Abs(nm)')
    plt.plot(tt[18:29],odsss1[18:29],label='OD J23101-J23101-J23101 E1R1')
    plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
    plt.legend(loc='lower right')
    plt.savefig(filename='fase exp 1.png', dpi=300, facecolor='w', edgecolor='w',bbox_inche

Min OD = 8.240000e-02
[  2.33932298e+00   9.81549169e-03   1.86507491e+02]
```



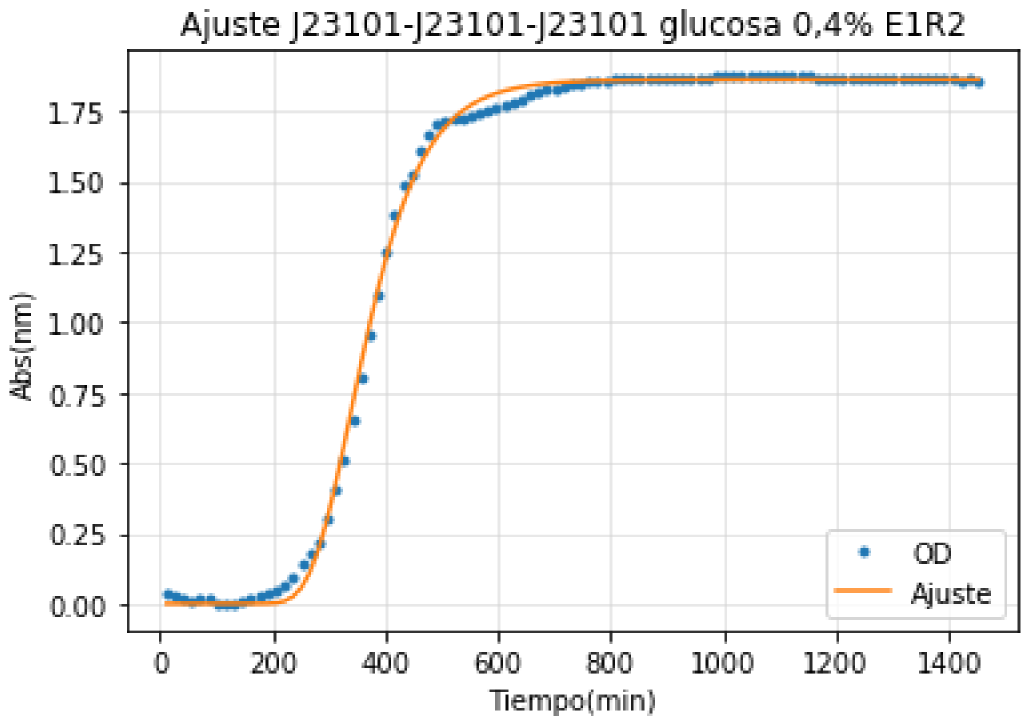

```
A=2.339323e+00
um=9.815492e-03
l=1.865075e+02
Tm=2.741841e+02
doubpe=7.061767e+01
ext=1.412353e+02
Tfinal=4.154194e+02
```


J23101-J23101-J23101 Glucosa 0,4% E1R1
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E1R1
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R1
0.2
0.3
0.4
0.5
0.6
280
300
320
340
360
380
400
420

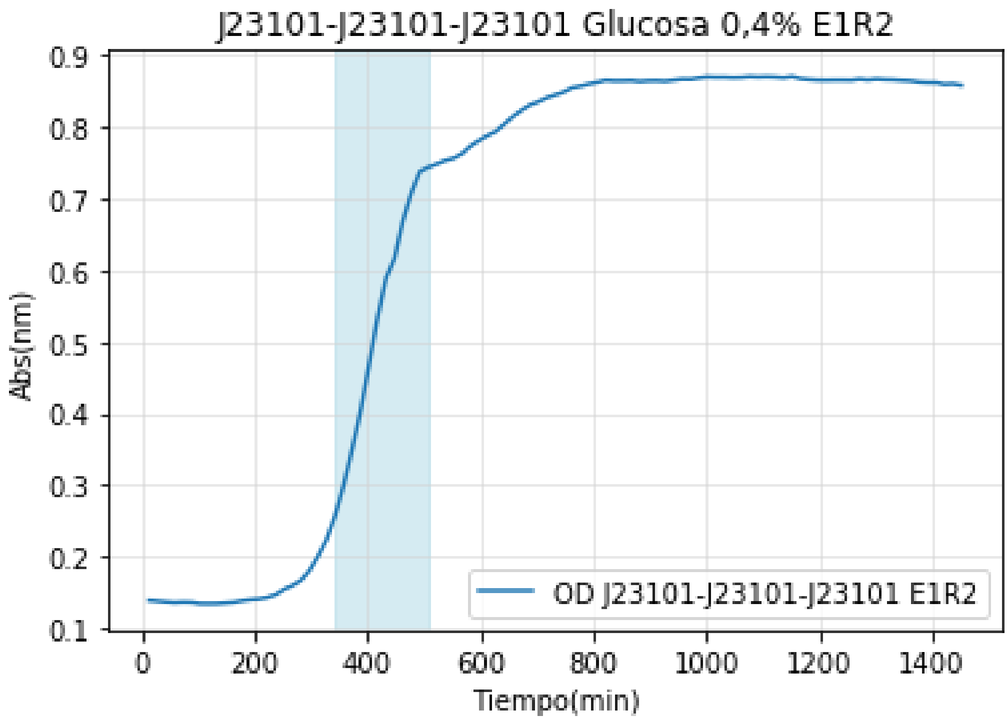
```
In [22]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #std-std-std glucosa rep 2
         y8= np.log(odsss2)-np.log(np.min(odsss2))
         print('Min OD = %e'%((np.min(odsss2))))
         evaly, params=Function_fit(tt,y8,0,-1,title = 'Ajuste J23101-J23101-J23101 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[22]
         y2=tt[33]
         plt.figure()
         plt.title('J23101-J23101-J23101 Glucosa 0,4% E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odsss2,label='OD J23101-J23101-J23101 E1R2')
         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 E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[22:34],odsss2[22:34],label='OD J23101-J23101-J23101 E1R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

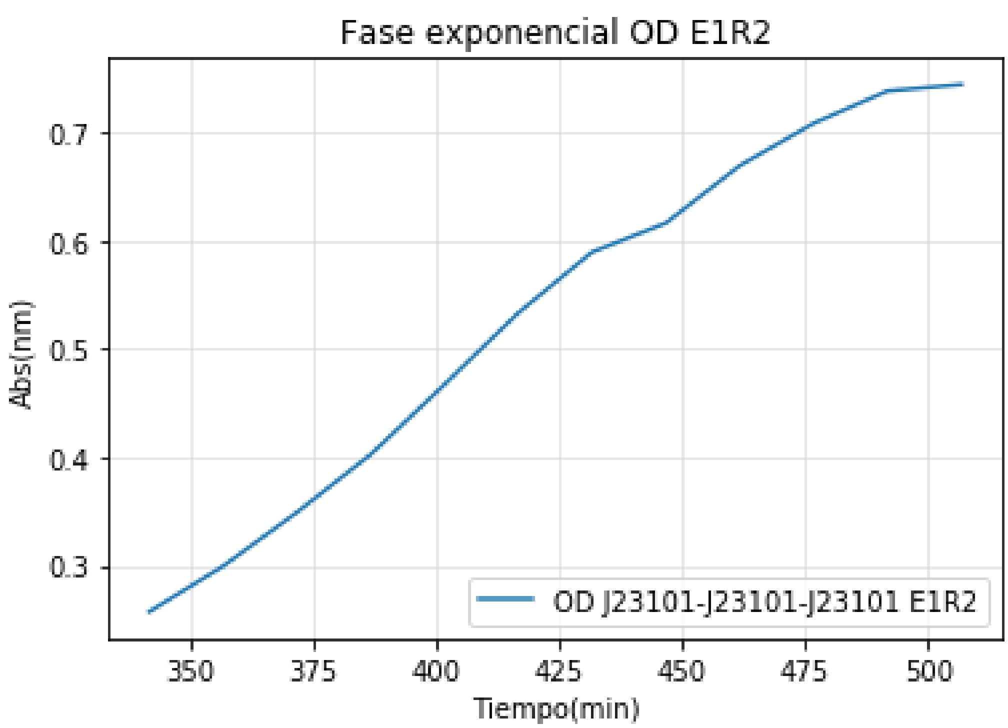
```
Min OD = 1.339000e-01
[  1.86115433e+00   9.68915115e-03   2.68632222e+02]
```




```
A=1.861154e+00
um=9.689151e-03
l=2.686322e+02
Tm=3.392969e+02
doubpe=7.153848e+01
ext=1.430770e+02
Tfinal=4.823738e+02
```

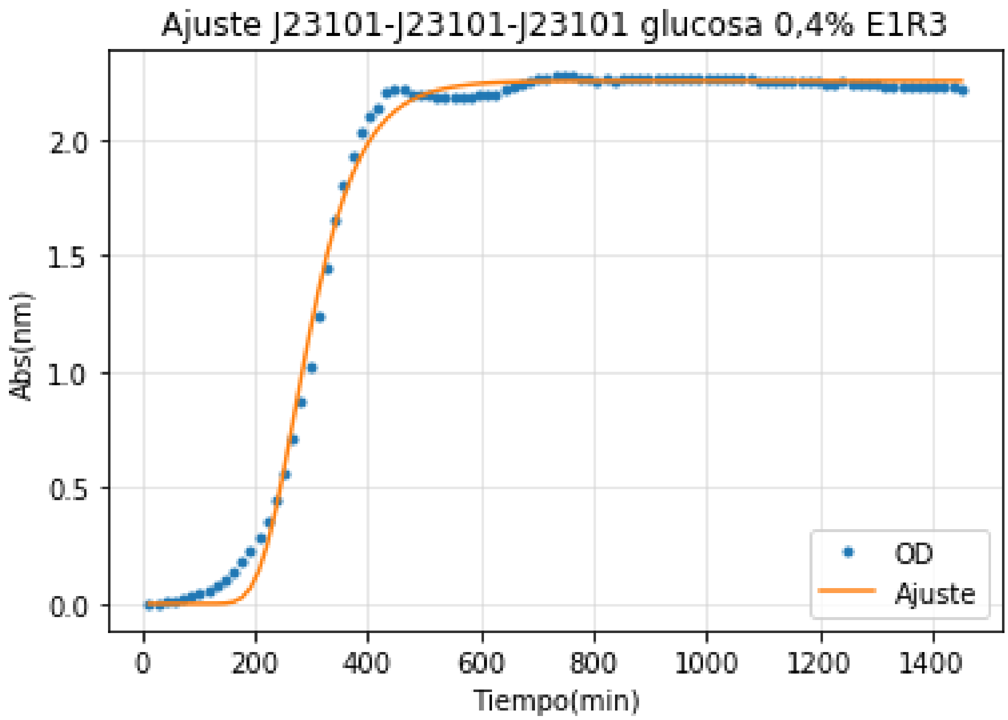
```
Out[22]: <matplotlib.legend.Legend at 0x1aba7bb7e48>
```


J23101-J23101-J23101 Glucosa 0,4% E1R2
0.9
0.8
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 J23101-J23101-J23101 E1R2


Fase exponencial OD E1R2
0.7
0.6
0.5
0.4
0.3
Abs(nm)
350
375
400
425
450
475
500
Tiempo(min)
OD J23101-J23101-J23101 E1R2

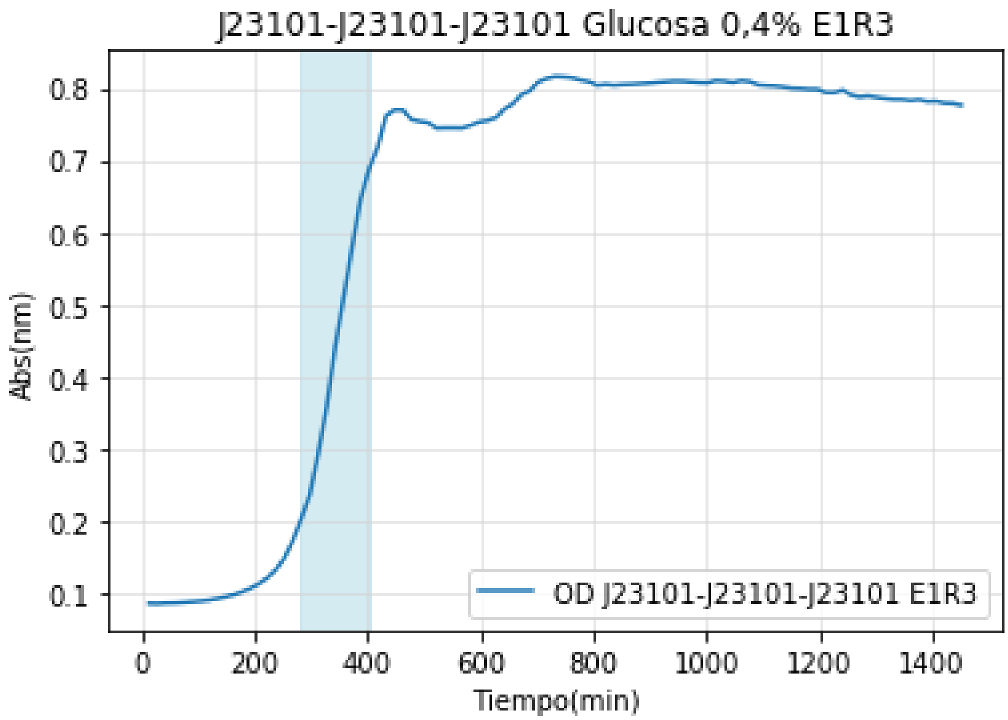
In [23]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#std-std-std glucosa rep 3
y9= np.log(odsss3)-np.log(np.min(odsss3))
print('Min OD = %e'%((np.min(odsss3))))
evaly, params=Function_fit(tt,y9,0,-1,title = 'Ajuste J23101-J23101-J23101 glucosa 0,4%
A9= params[0]
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[26]
plt.figure()
plt.title('J23101-J23101-J23101 Glucosa 0,4% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odsss3,label='OD J23101-J23101-J23101 E1R3')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:27],odsss3[18:27],label='OD J23101-J23101-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

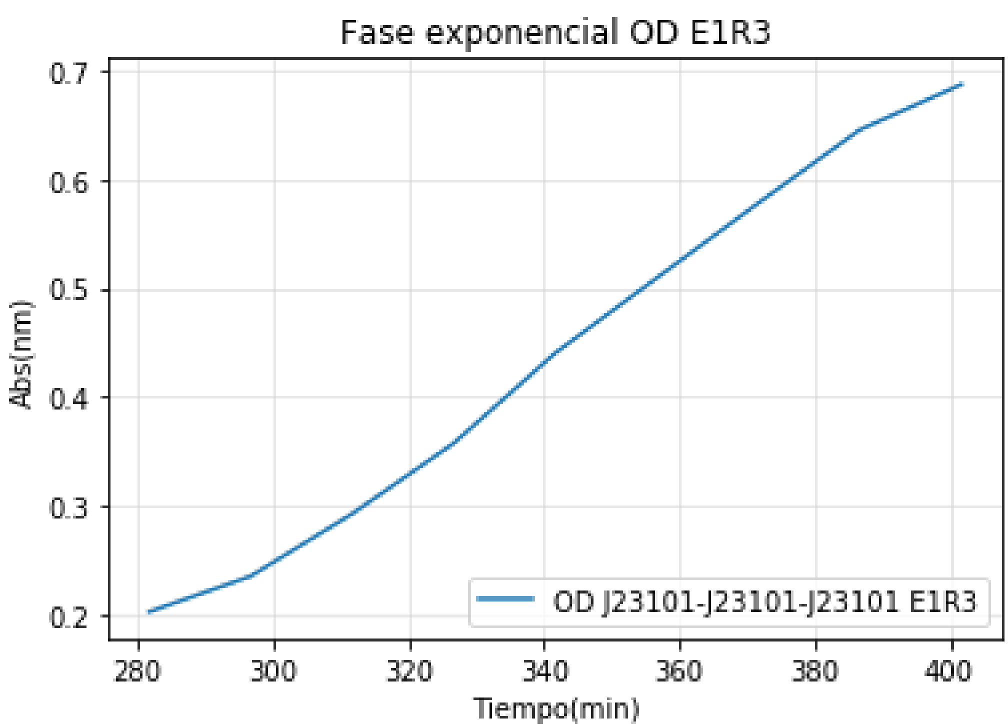
```
Min OD = 8.440000e-02
[  2.25053212e+00   1.31749960e-02   2.07421736e+02]
```




```
A=2.250532e+00
um=1.317500e-02
l=2.074217e+02
Tm=2.702623e+02
doubpe=5.261081e+01
ext=1.052216e+02
Tfinal=3.754839e+02
```

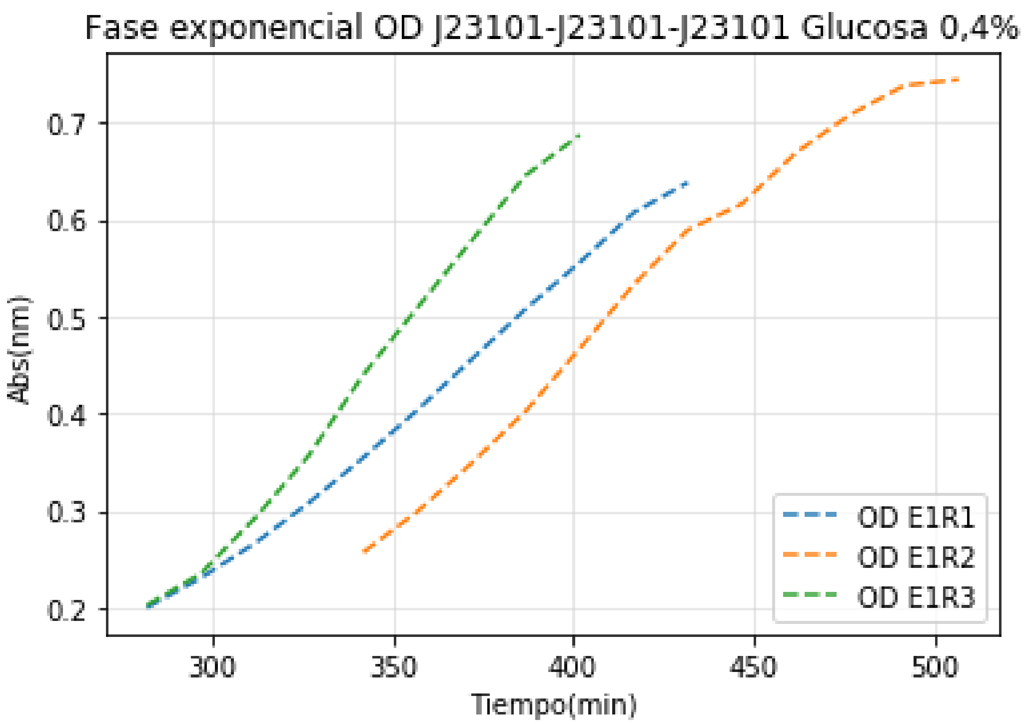
```
Out[23]: <matplotlib.legend.Legend at 0x1aba7e10978>
```


J23101-J23101-J23101 Glucosa 0,4% E1R3
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R3


Fase exponencial OD E1R3
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R3

```
In [24]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23101-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[18:29],odsss1[18:29],'--',label='OD E1R1')
         plt.plot(tt[22:34],odsss2[22:34],'--',label='OD E1R2')
         plt.plot(tt[18:27],odsss3[18:27],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

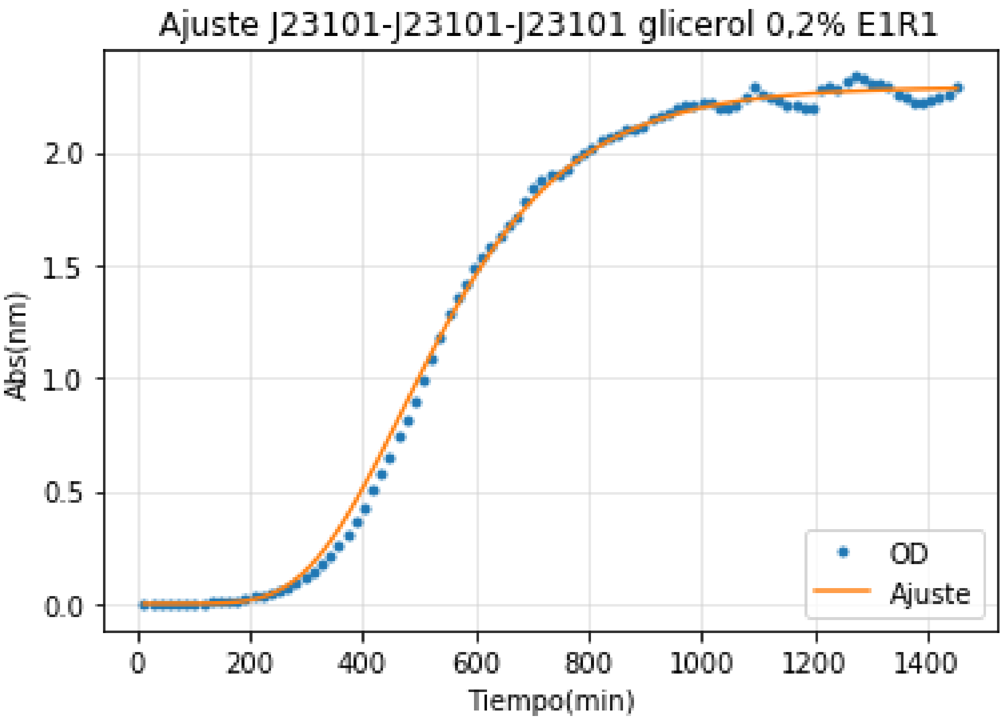
```
Out[24]: <matplotlib.legend.Legend at 0x1aba7912be0>
```




```
In [25]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #std-std-std glicerol rep 1
         y10= np.log(odsssg1)-np.log(np.min(odsssg1))
         print('Min OD = %e'%((np.min(odsssg1))))
         evaly, params=Function_fit(tt,y10,0,-1,title = 'Ajuste J23101-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))
```

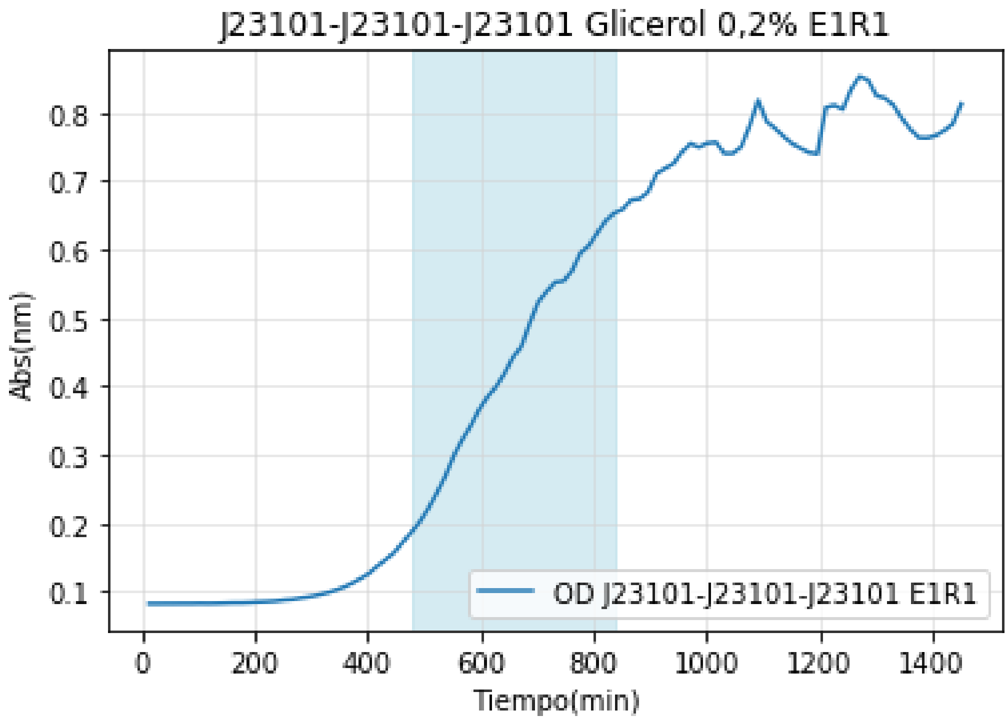
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm10=((A10/(np.exp(1)*um10))+l10)
print('Tm=%e'%(tm10))
t210=((np.log(2))/um10)
print('doubpe=%e'%(t210))
extdp10=2.5*t210
print('ext=%e'%extdp10)
ttot10=tm10+extdp10
print('Tfinal=%e'%ttot10)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[31]
y2=tt[55]
plt.figure()
plt.title('J23101-J23101-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odsssg1,label='OD J23101-J23101-J23101 E1R1')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[31:56],odsssg1[31:56],label='OD J23101-J23101-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

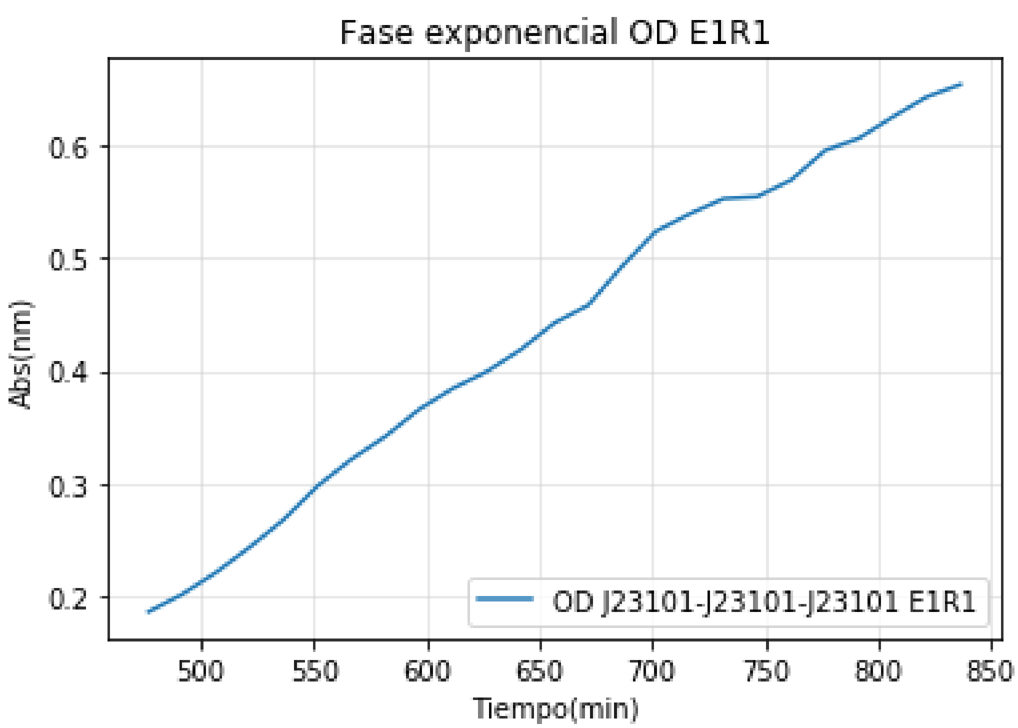
```
Min OD = 8.250000e-02
[  2.28976144e+00   5.04283145e-03   3.00000000e+02]
```




```
A=2.289761e+00
um=5.042831e-03
l=3.000000e+02
Tm=4.670403e+02
doubpe=1.374520e+02
ext=3.436300e+02
Tfinal=8.106703e+02
```

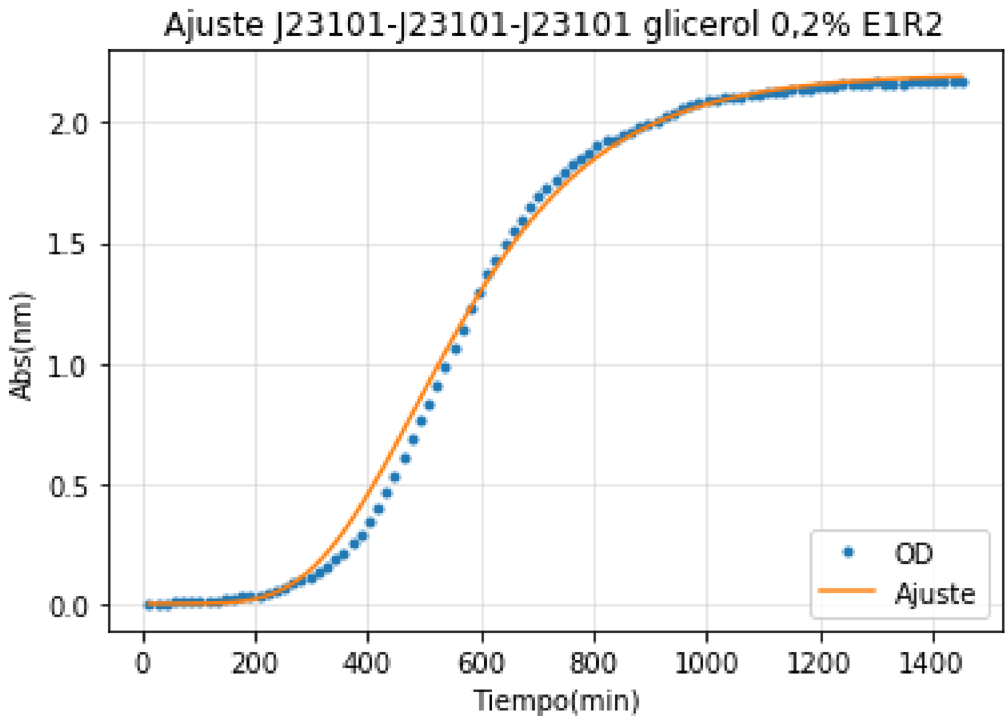
```
Out[25]: <matplotlib.legend.Legend at 0x1aba78a7e10>
```


J23101-J23101-J23101 Glicerol 0,2% E1R1
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R1


Fase exponencial OD E1R1
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R1

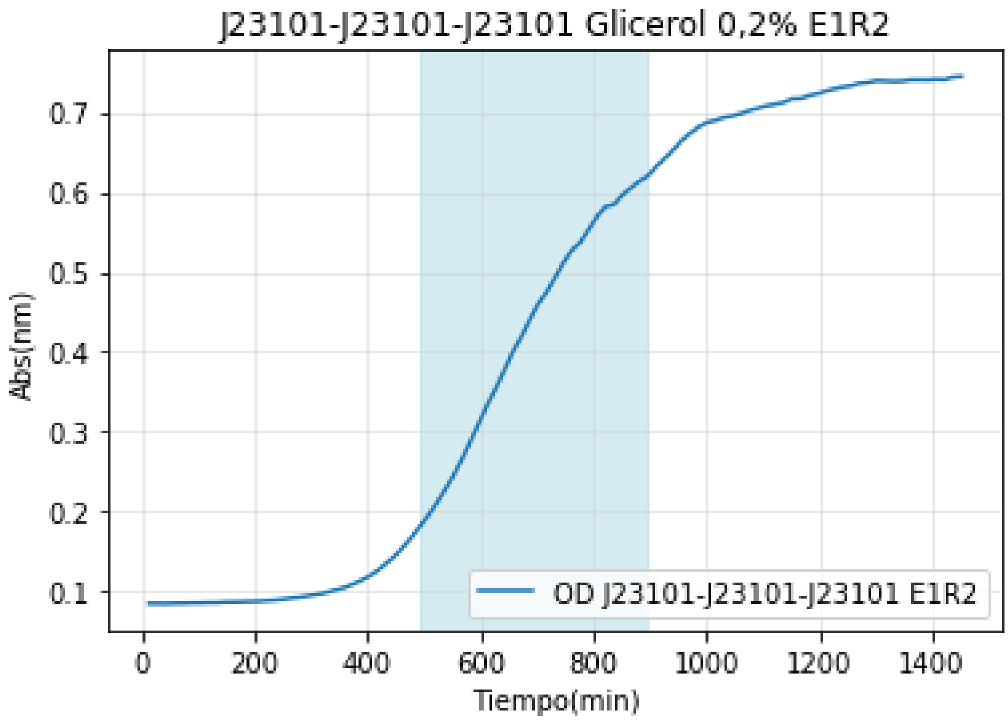
```
In [26]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #std-std-std glicerol rep 2
         y11= np.log(odsssg2)-np.log(np.min(odsssg2))
         print('Min OD = %e'%((np.min(odsssg2))))
         evaly, params=Function_fit(tt,y11,0,-1,title = 'Ajuste J23101-J23101-J23101 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[32]
         y2=tt[59]
         plt.figure()
         plt.title('J23101-J23101-J23101 Glicerol 0,2% E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odsssg2,label='OD J23101-J23101-J23101 E1R2')
         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 E1R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[32:60],odsssg2[32:60],label='OD J23101-J23101-J23101 E1R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

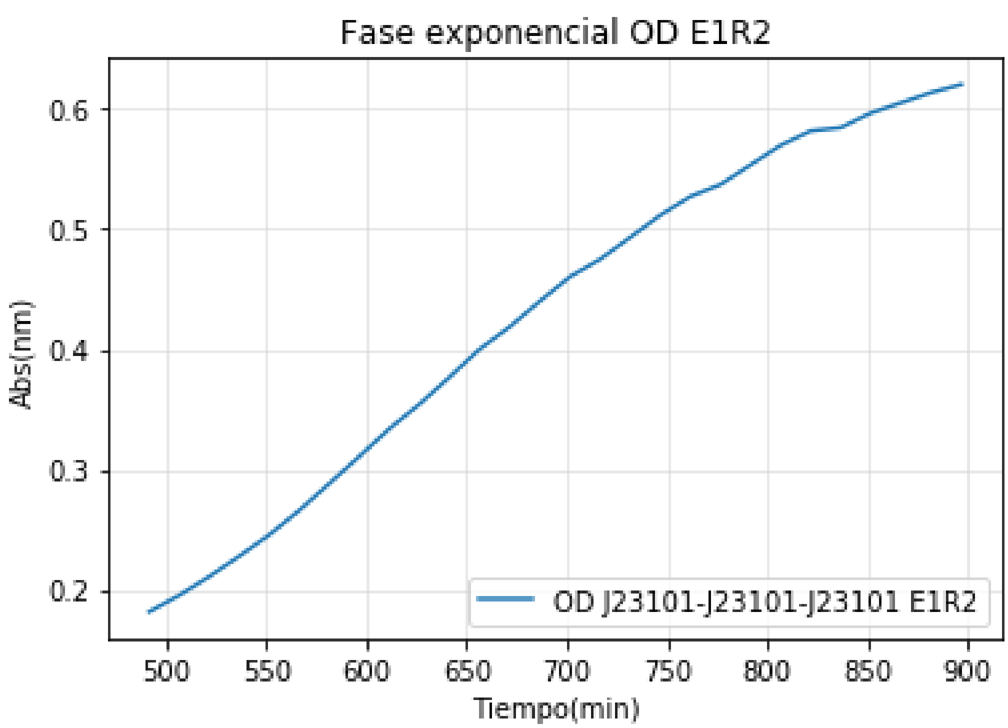
```
Min OD = 8.480000e-02
[  2.20113680e+00   4.42943009e-03   3.00000000e+02]
```




```
A=2.201137e+00
um=4.429430e-03
l=3.000000e+02
Tm=4.828120e+02
doubpe=1.564868e+02
ext=3.912169e+02
Tfinal=8.740289e+02
```

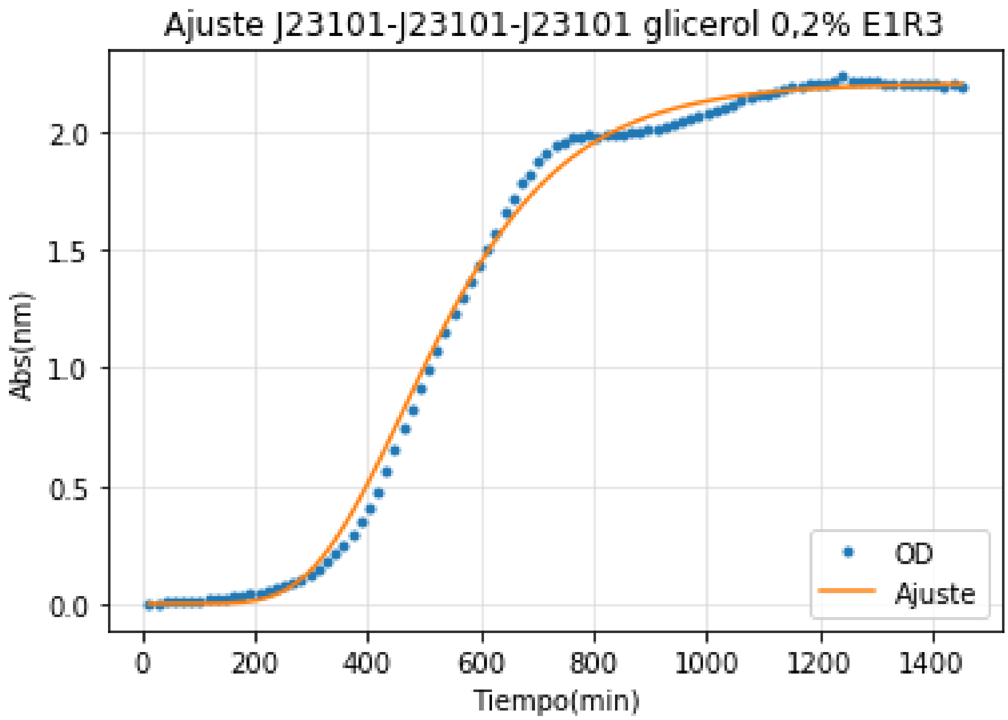
```
Out[26]: <matplotlib.legend.Legend at 0x1aba7ec0668>
```


J23101-J23101-J23101 Glicerol 0,2% E1R2
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E1R2
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R2
0.2
0.3
0.4
0.5
0.6
500
550
600
650
700
750
800
850
900

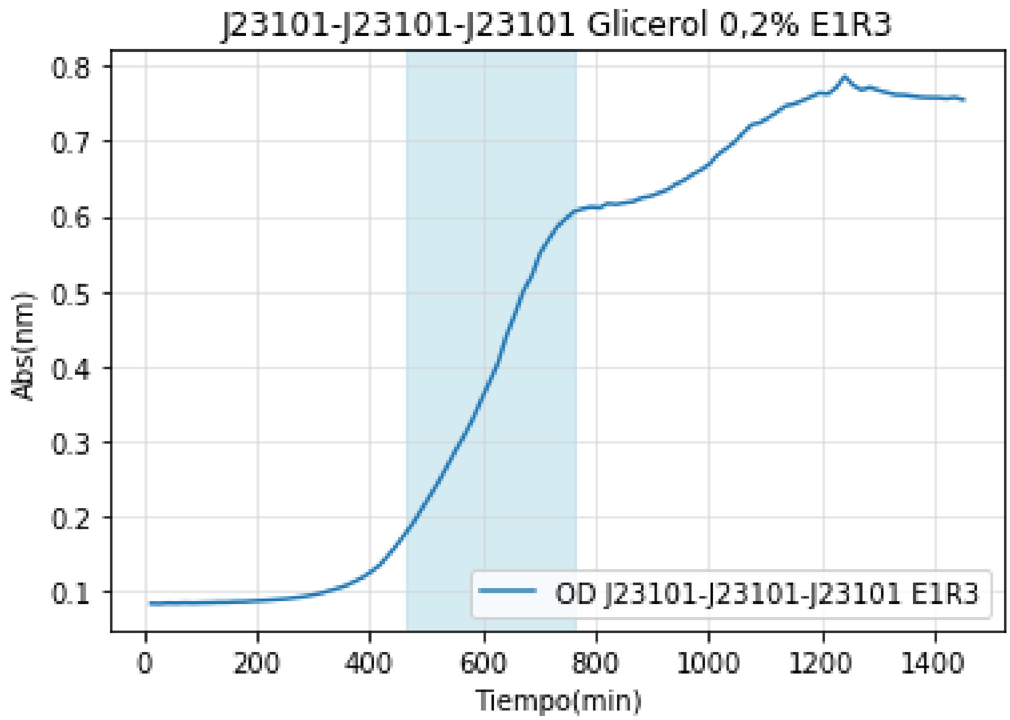
```
In [27]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #std-std-std glicerol rep 3
         y12= np.log(odsssg3)-np.log(np.min(odsssg3))
         print('Min OD = %e'%((np.min(odsssg3))))
         evaly, params=Function_fit(tt,y12,0,-1,title = 'Ajuste J23101-J23101-J23101 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[30]
         y2=tt[50]
         plt.figure()
         plt.title('J23101-J23101-J23101 Glicerol 0,2% E1R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odsssg3,label='OD J23101-J23101-J23101 E1R3')
         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 E1R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[30:51],odsssg3[30:51],label='OD J23101-J23101-J23101 E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

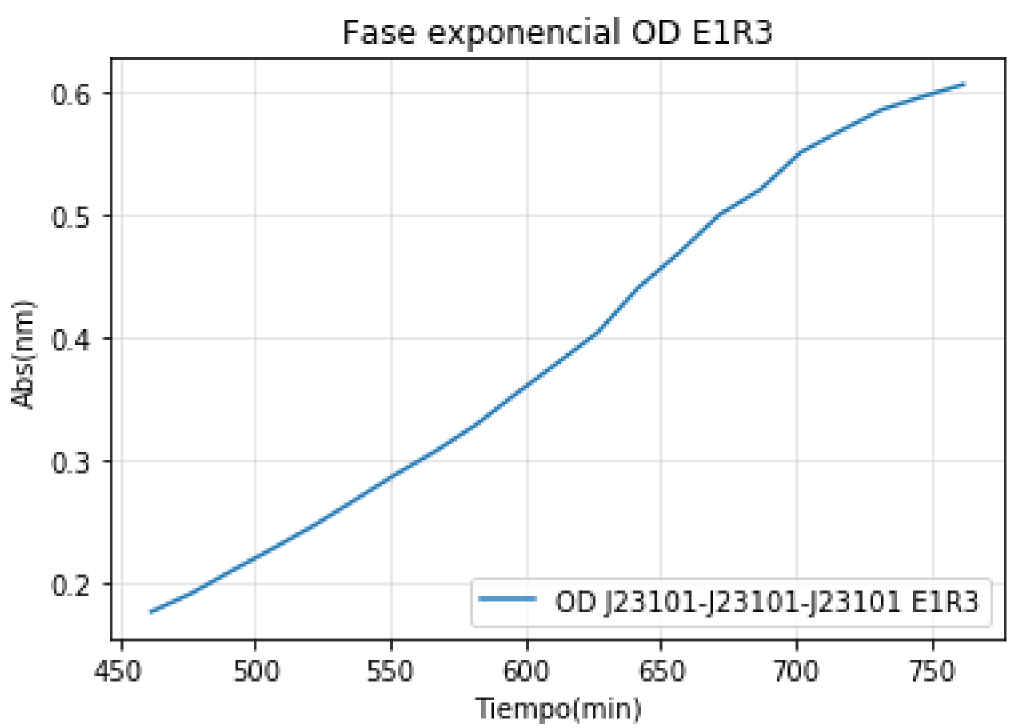
```
Min OD = 8.420000e-02
[  2.20575506e+00   5.04230786e-03   3.00000000e+02]
```




```
A=2.205755e+00
um=5.042308e-03
l=3.000000e+02
Tm=4.609287e+02
doubpe=1.374663e+02
ext=2.749325e+02
Tfinal=7.358612e+02
```

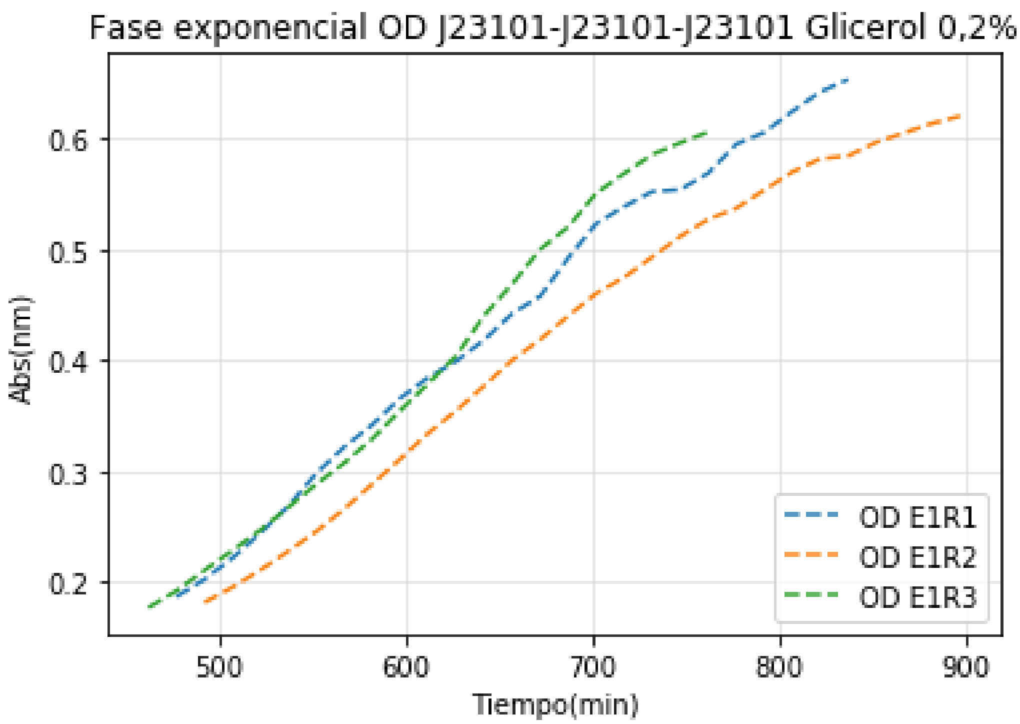
```
Out[27]: <matplotlib.legend.Legend at 0x1aba80c70f0>
```


J23101-J23101-J23101 Glicerol 0,2% E1R3
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R3
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 E1R3
Abs(nm)
Tiempo(min)
OD J23101-J23101-J23101 E1R3
0.6
0.5
0.4
0.3
0.2
450
500
550
600
650
700
750

In [28]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23101-J23101-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[31:56],odsssg1[31:56],'--',label='OD E1R1')
plt.plot(tt[32:60],odsssg2[32:60],'--',label='OD E1R2')
plt.plot(tt[30:51],odsssg3[30:51],'--',label='OD E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

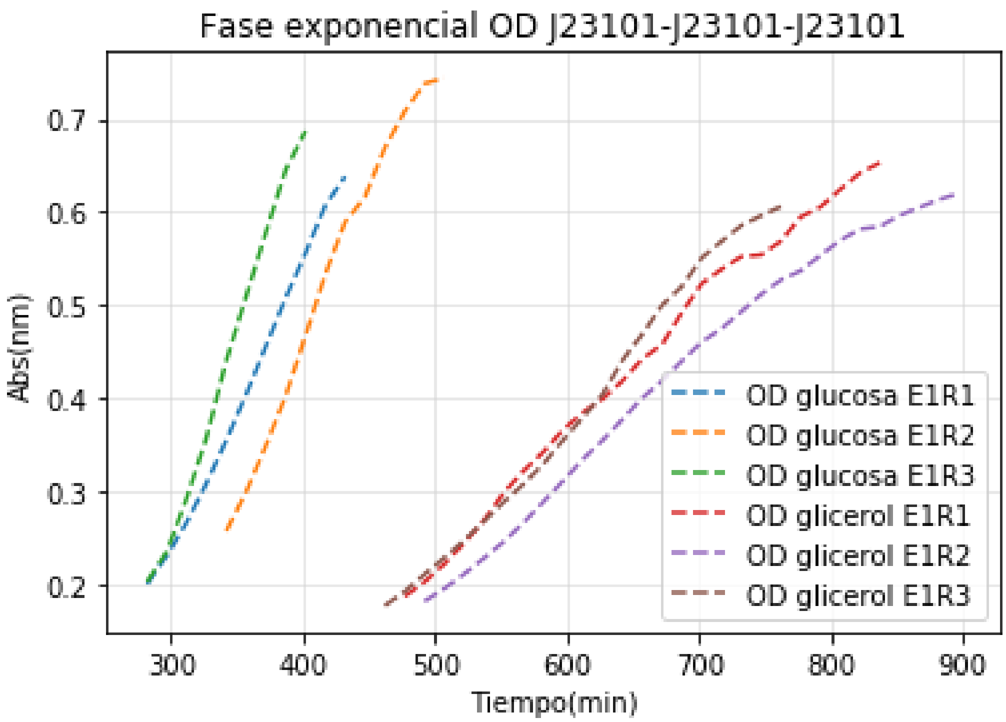
Out[28]: <matplotlib.legend.Legend at 0x1aba803e7f0>



In [29]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23101-J23101-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:29],odsss1[18:29],'--',label='OD glucosa E1R1')
plt.plot(tt[22:34],odsss2[22:34],'--',label='OD glucosa E1R2')
plt.plot(tt[18:27],odsss3[18:27],'--',label='OD glucosa E1R3')
plt.plot(tt[31:56],odsssg1[31:56],'--',label='OD glicerol E1R1')
plt.plot(tt[32:60],odsssg2[32:60],'--',label='OD glicerol E1R2')
plt.plot(tt[30:51],odsssg3[30:51],'--',label='OD glicerol E1R3')
```

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

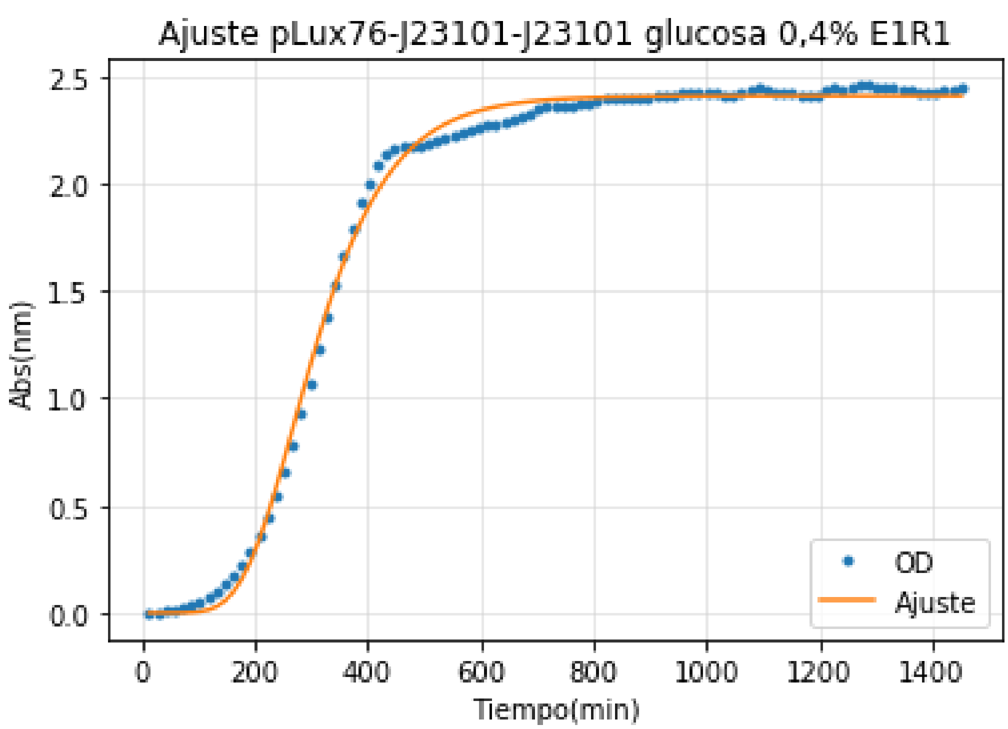
Out[29]: <matplotlib.legend.Legend at 0x1aba75b4828>



In [30]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 1
y13= np.log(odpss1)-np.log(np.min(odpss1))
print('Min OD = %e'%((np.min(odpss1))))
evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste pLux76-J23101-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)
```

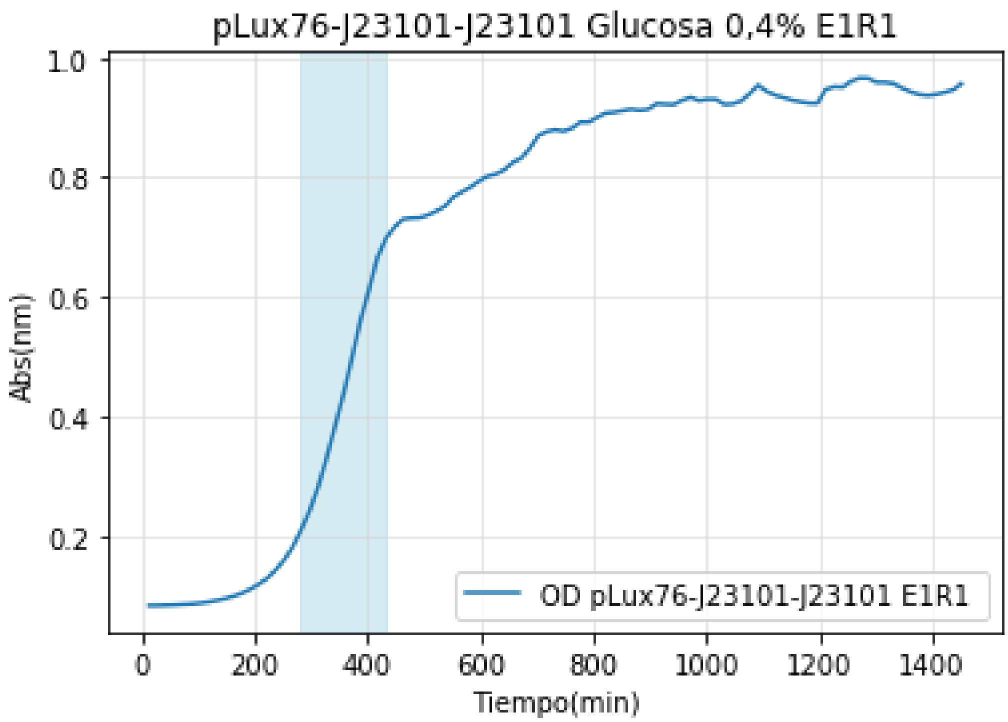
```
ttot13=tm13+extdp13
print('Tfinal=%e'%ttot13)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[18]
y2=tt[28]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpss1,label='OD pLux76-J23101-J23101 E1R1 ')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:29],odpss1[18:29],label='OD pLux76-J23101-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

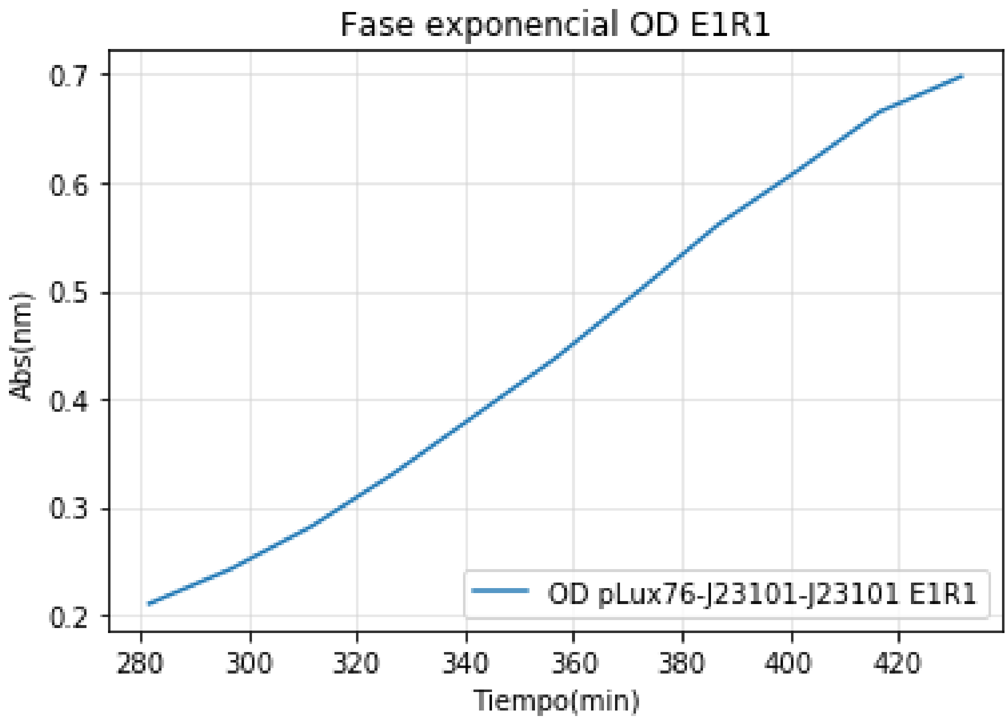
```
Min OD = 8.280000e-02
[  2.40847070e+00   9.55708597e-03   1.76557369e+02]
```



```
A=2.408471e+00
um=9.557086e-03
l=1.765574e+02
Tm=2.692663e+02
doubpe=7.252704e+01
ext=1.450541e+02
Tfinal=4.143203e+02
```

```
Out[30]: <matplotlib.legend.Legend at 0x1aba77108d0>
```

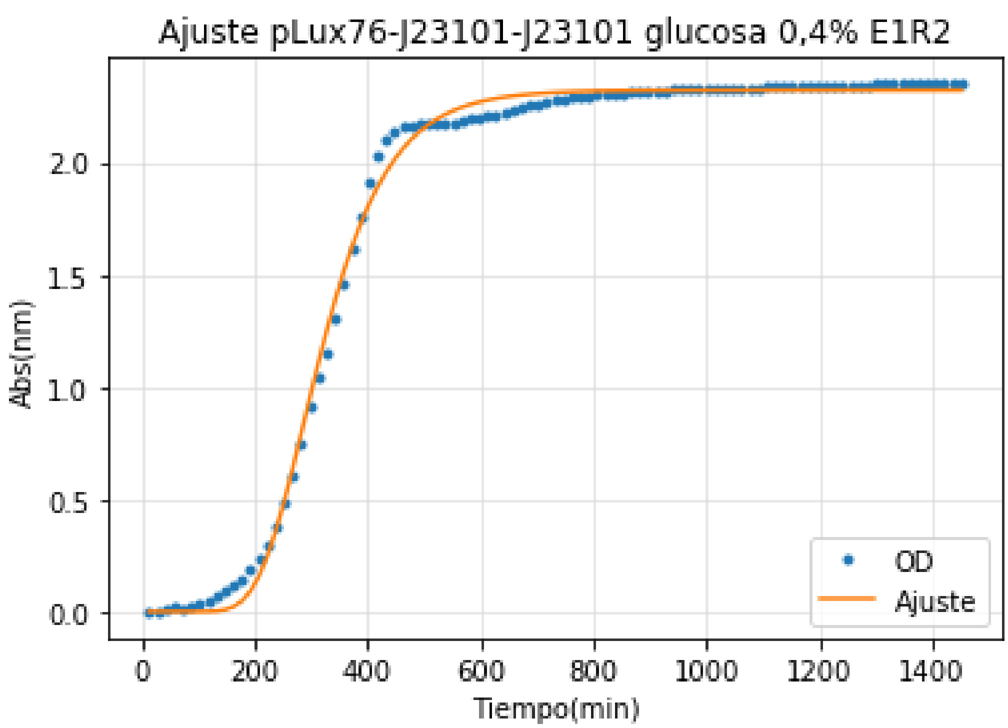




In [31]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 2
y14= np.log(odpss2)-np.log(np.min(odpss2))
print('Min OD = %e'%((np.min(odpss2))))
evaly, params=Function_fit(tt,y14,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4
A14= params[0]
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]
```

```
y2=tt[29]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpss2,label='OD pLux76-J23101-J23101 E1R2 ')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odpss2[19:30],label='OD pLux76-J23101-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

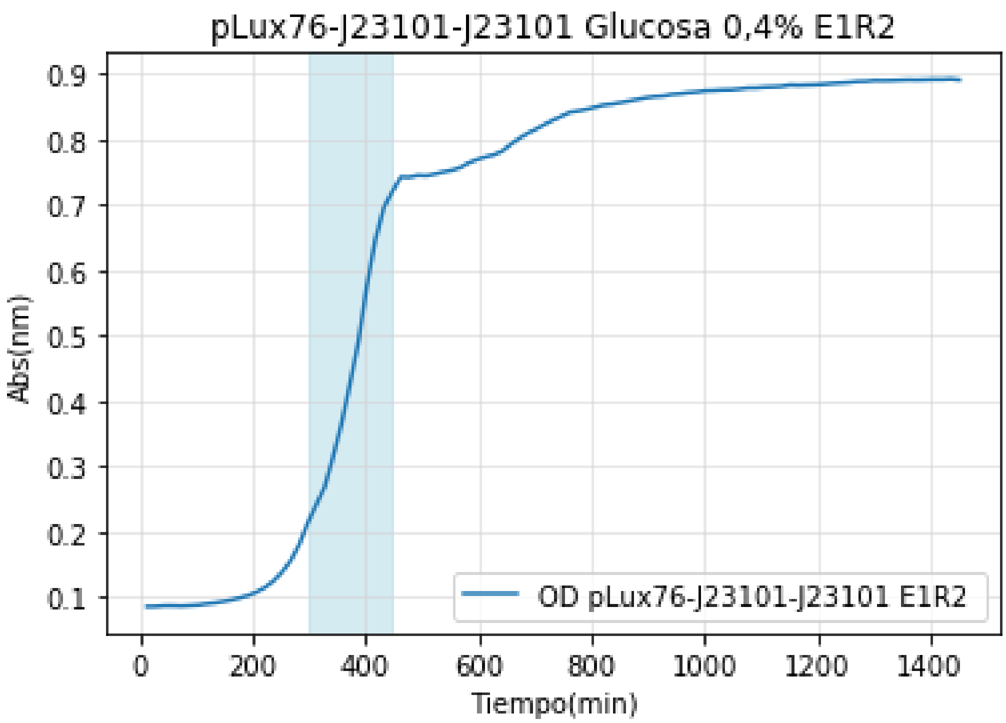
```
Min OD = 8.520000e-02
[ 2.32135340e+00   1.04470064e-02   2.05621430e+02]
```

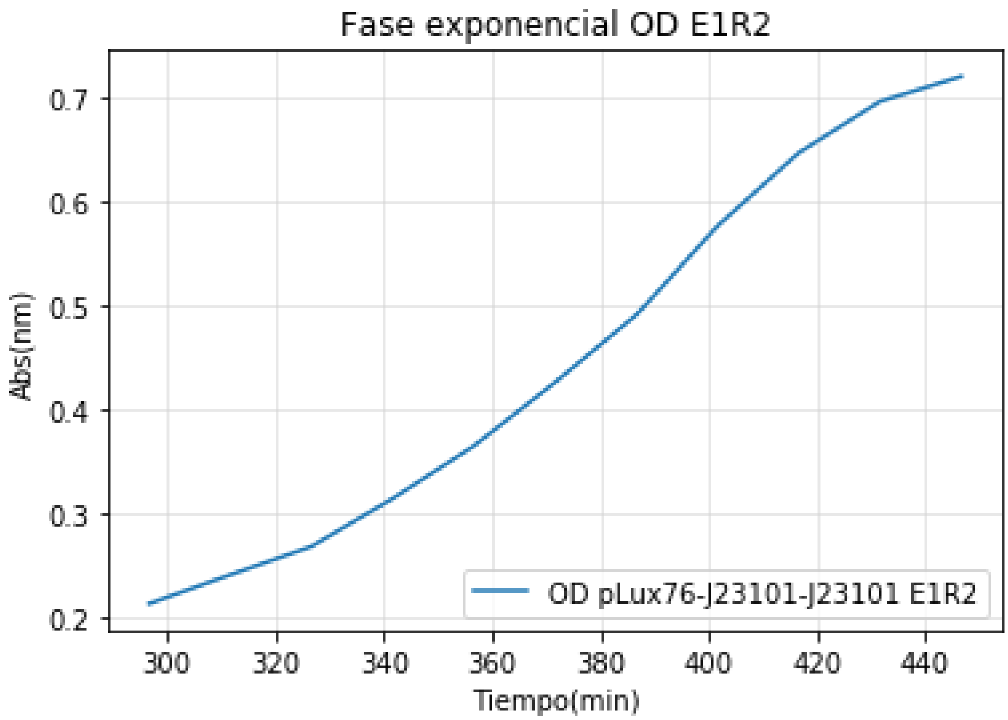



```
A=2.321353e+00
um=1.044701e-02
```

```
l=2.056214e+02
Tm=2.873652e+02
doubpe=6.634888e+01
ext=1.326978e+02
Tfinal=4.200630e+02
```

Out[31]: <matplotlib.legend.Legend at 0x1aba7fc5278>

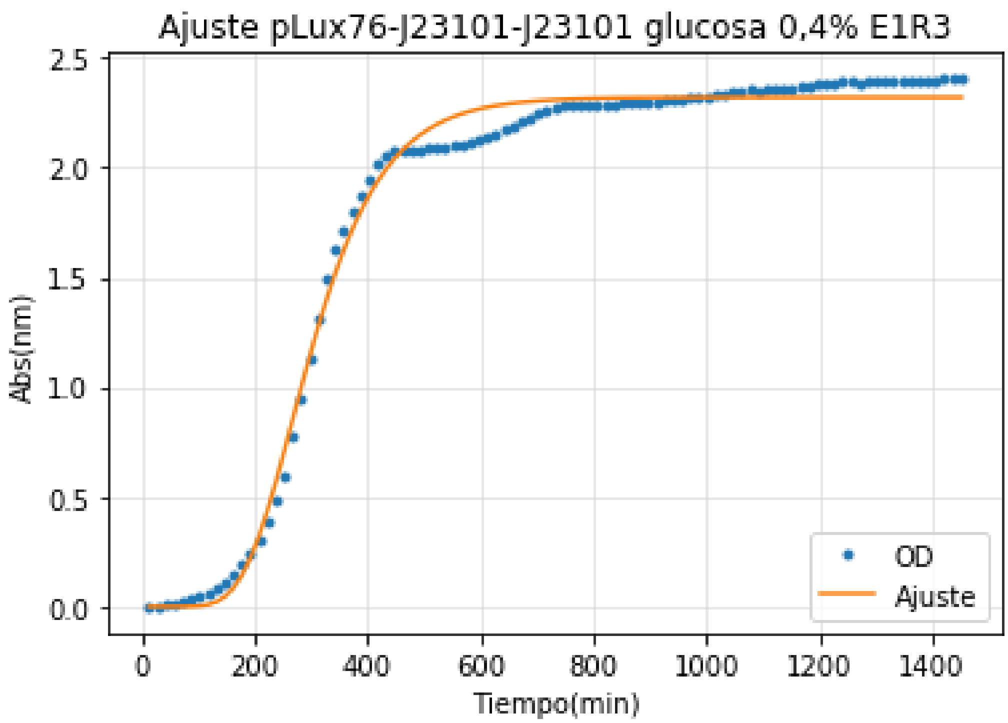




In [32]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 3
y15= np.log(odpss3)-np.log(np.min(odpss3))
print('Min OD = %e'%((np.min(odpss3))))
evaly, params=Function_fit(tt,y15,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4
A15= params[0]
um15=params[1]
l15=params[2]
print('A=%e'%(A15))
print('um=%e'%(um15))
print('l=%e'%(l15))

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

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

```
y2=tt[28]
plt.figure()
plt.title('pLux76-J23101-J23101 Glucosa 0,4% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpss3,label='OD pLux76-J23101-J23101 E1R3 ')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],odpss3[17:29],label='OD pLux76-J23101-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

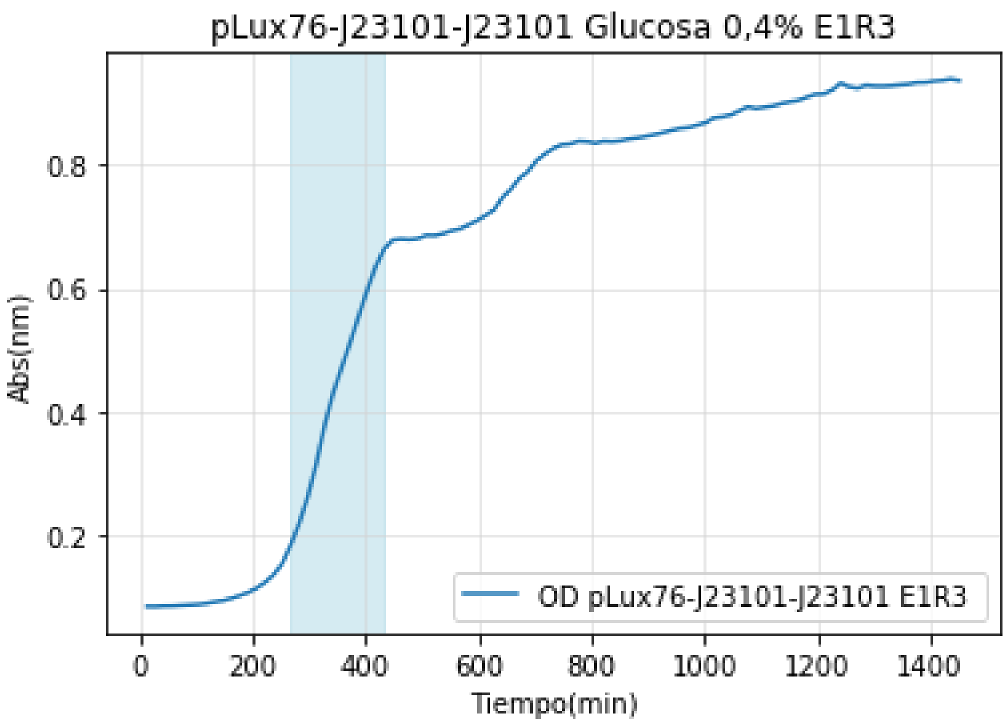
```
Min OD = 8.490000e-02
[  2.32026468e+00   9.71359431e-03   1.78609284e+02]
```

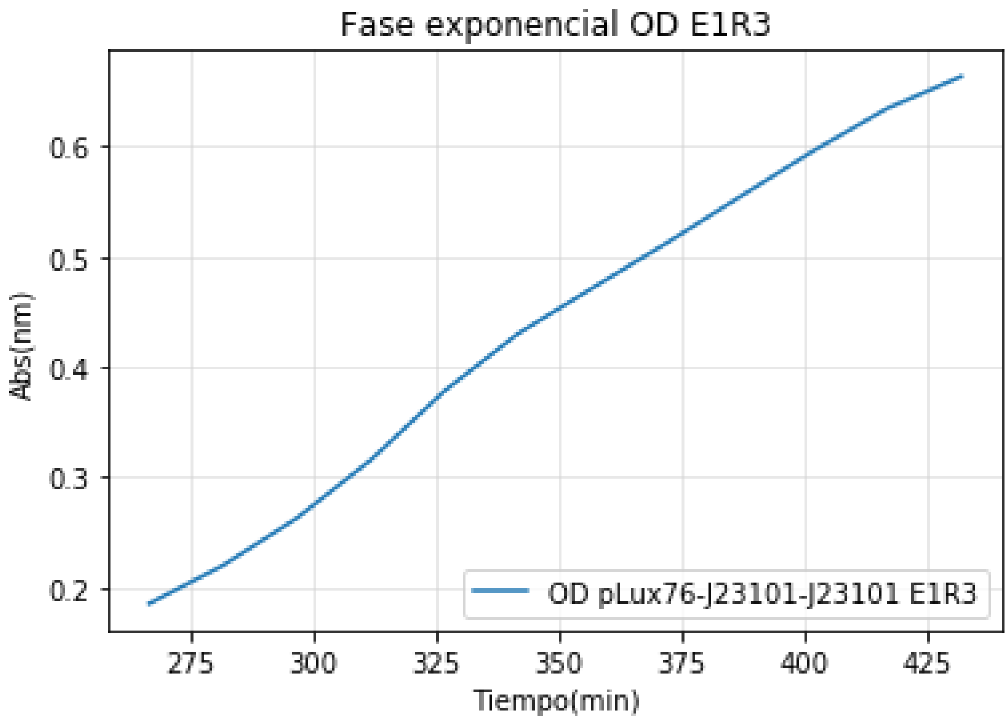



```
A=2.320265e+00
um=9.713594e-03
```

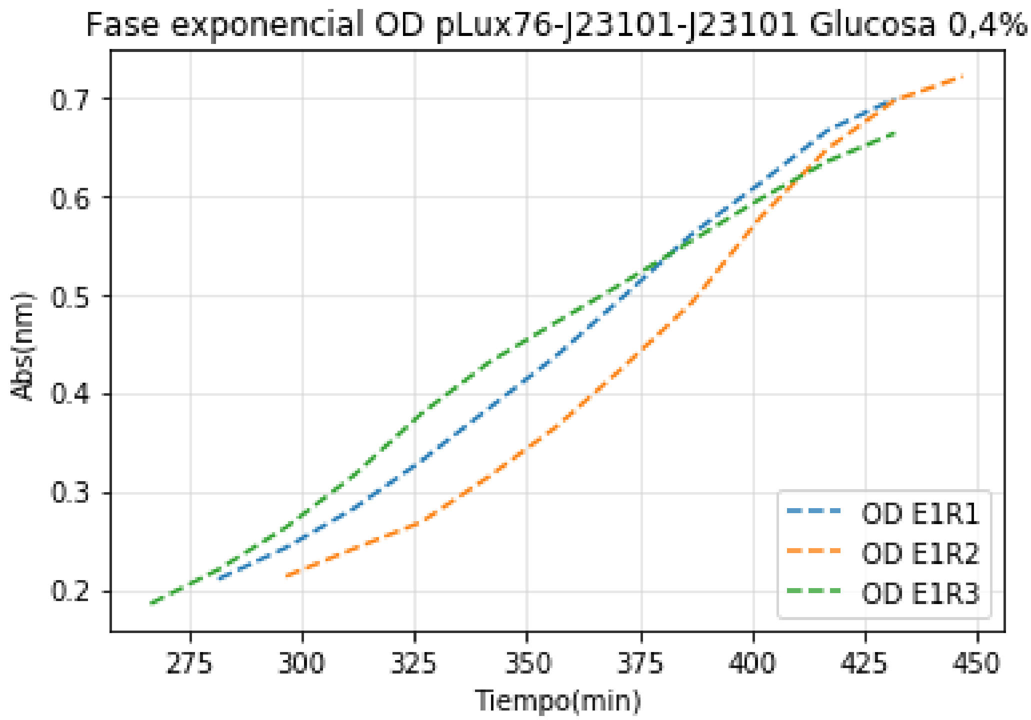
```
l=1.786093e+02
Tm=2.664838e+02
doubpe=7.135847e+01
ext=1.427169e+02
Tfinal=4.092008e+02
```

```
Out[32]: <matplotlib.legend.Legend at 0x1aba9355828>
```





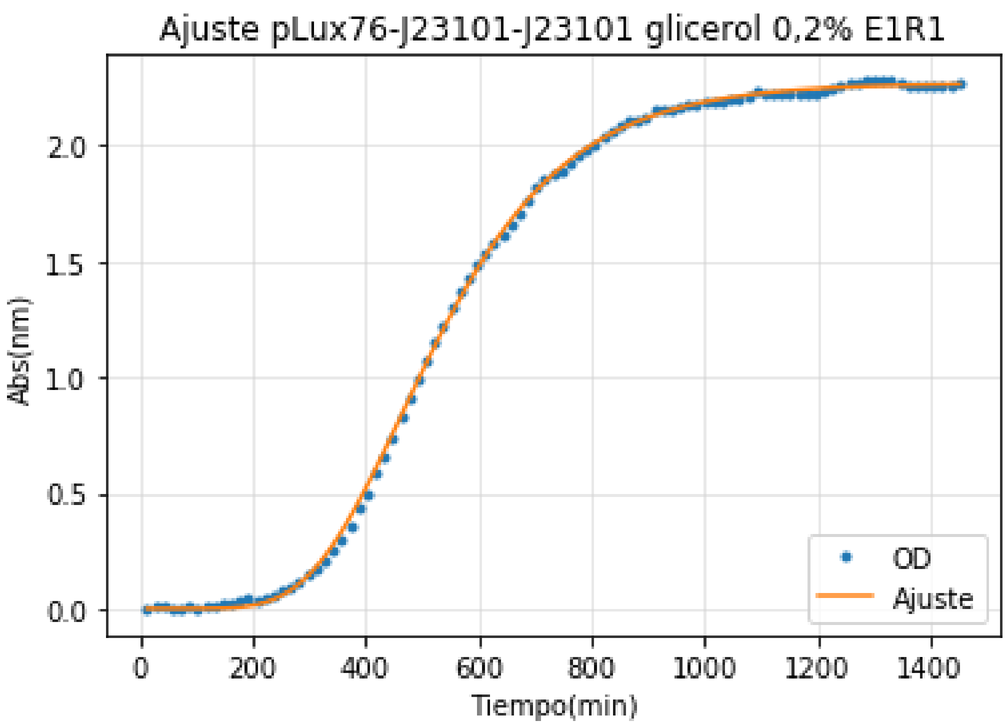

```
In [33]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[18:29],odpss1[18:29],'--',label='OD E1R1')
         plt.plot(tt[19:30],odpss2[19:30],'--',label='OD E1R2')
         plt.plot(tt[17:29],odpss3[17:29],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[33]: <matplotlib.legend.Legend at 0x1aba81d7dd8>
```




```
In [34]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #pLux-std-std glicerol rep 1
         y16= np.log(odpssg1)-np.log(np.min(odpssg1))
         print('Min OD = %e'%((np.min(odpssg1))))
         evaly, params=Function_fit(tt,y16,0,-1,title = 'Ajuste pLux76-J23101-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[31]
```

```
y2=tt[54]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpssg1,label='OD pLux76-J23101-J23101 E1R1')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[31:55],odpssg1[31:55],label='OD pLux76-J23101-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

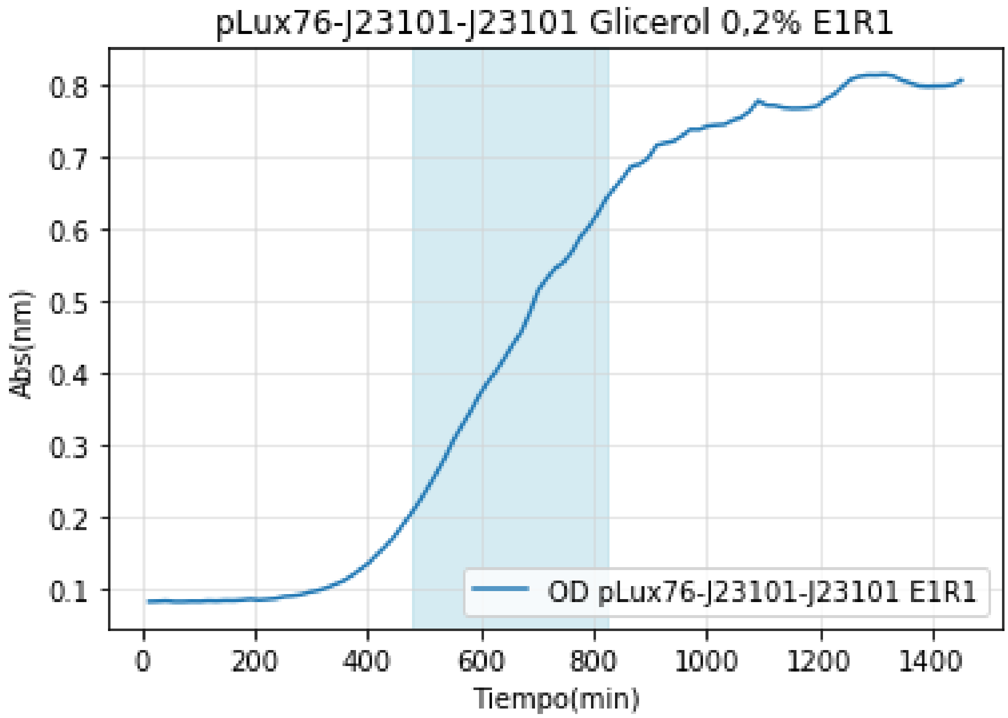
```
Min OD = 8.350000e-02
[  2.26948963e+00   5.14262207e-03   3.00000000e+02]
```

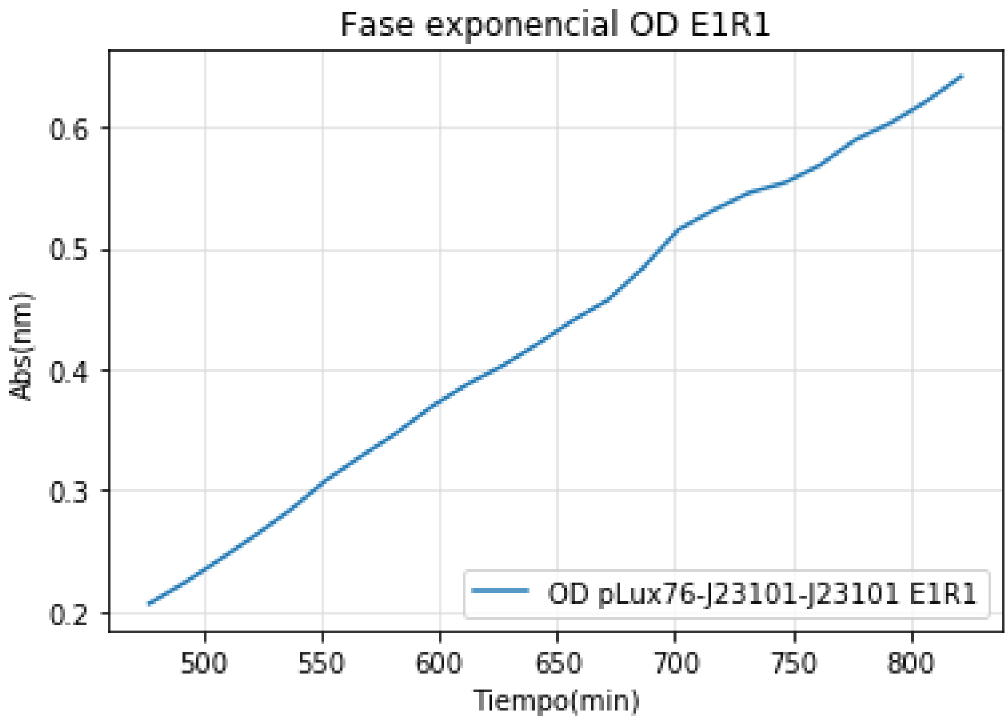



```
A=2.269490e+00
um=5.142622e-03
```

```
l=3.000000e+02
Tm=4.623488e+02
doubpe=1.347848e+02
ext=3.369619e+02
Tfinal=7.993108e+02
```

Out[34]: <matplotlib.legend.Legend at 0x1aba7f46f60>

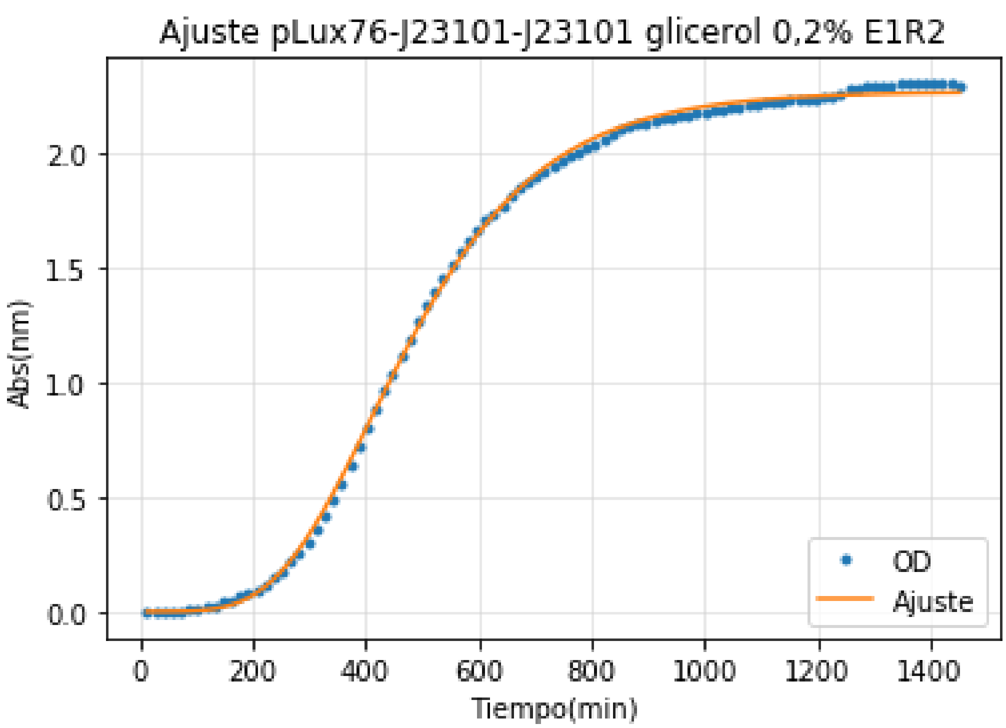




In [35]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-std-std glicerol rep 2
y17= np.log(odpssg2)-np.log(np.min(odpssg2))
print('Min OD = %e'%((np.min(odpssg2))))
evaly, params=Function_fit(tt,y17,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
A17= params[0]
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[27]
```

```python
y2=tt[51]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpssg2,label='OD pLux76-J23101-J23101 E1R2')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:52],odpssg2[27:52],label='OD pLux76-J23101-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

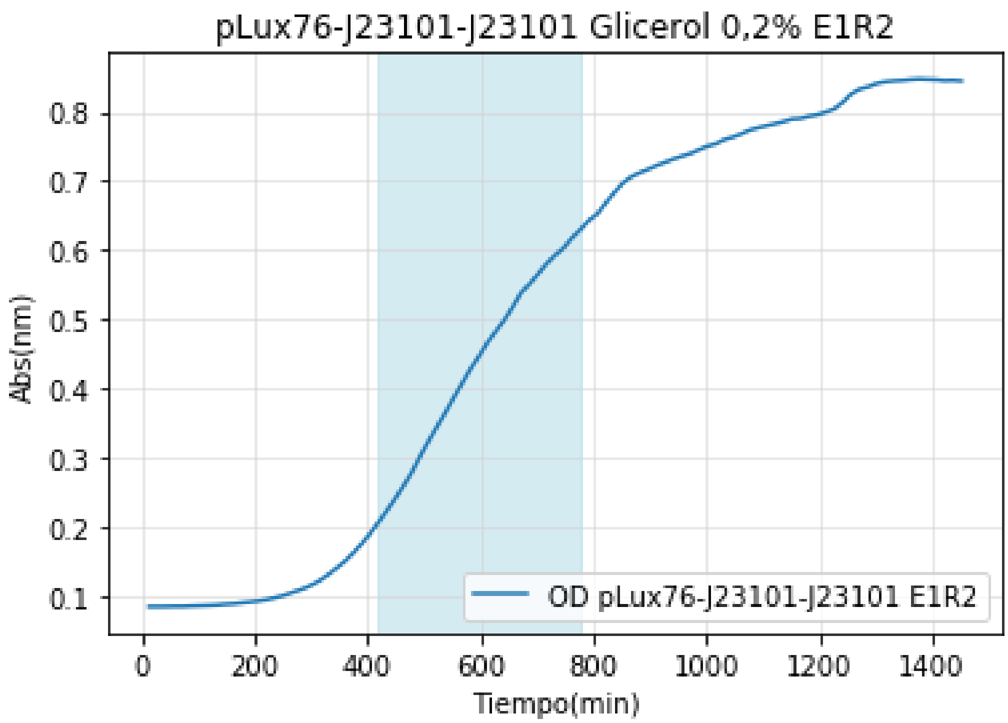
```
Min OD = 8.470000e-02
[  2.27088512e+00   4.98096927e-03   2.39618556e+02]
```

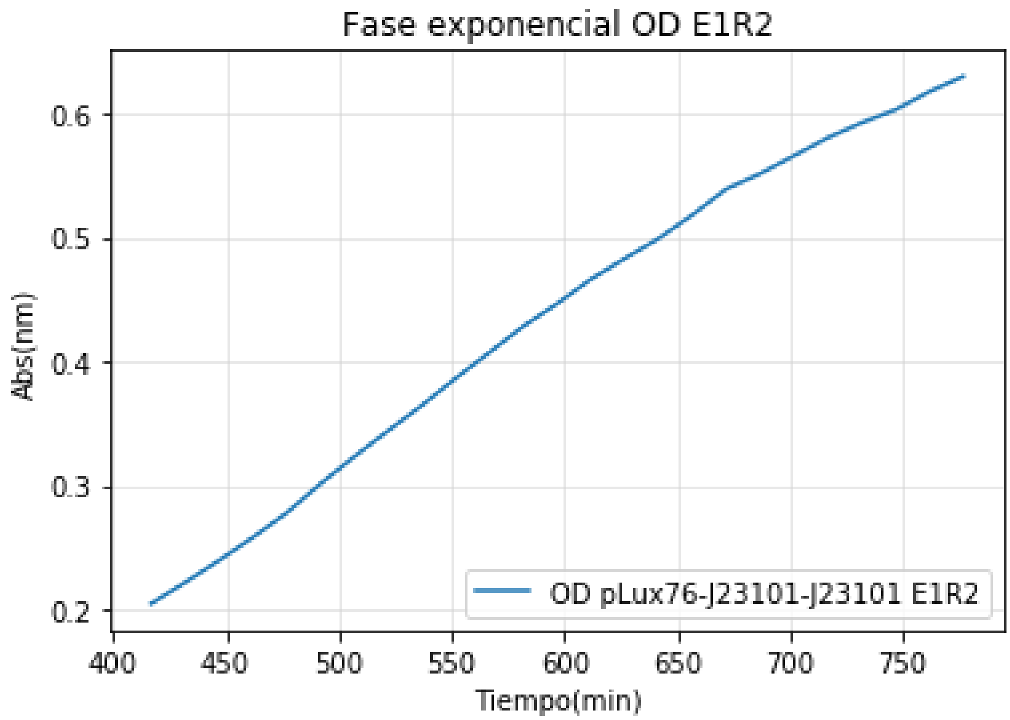



```
A=2.270885e+00
um=4.980969e-03
```

```
l=2.396186e+02
Tm=4.073393e+02
doubpe=1.391591e+02
ext=3.478977e+02
Tfinal=7.552371e+02
```

Out[35]: <matplotlib.legend.Legend at 0x1aba949a208>

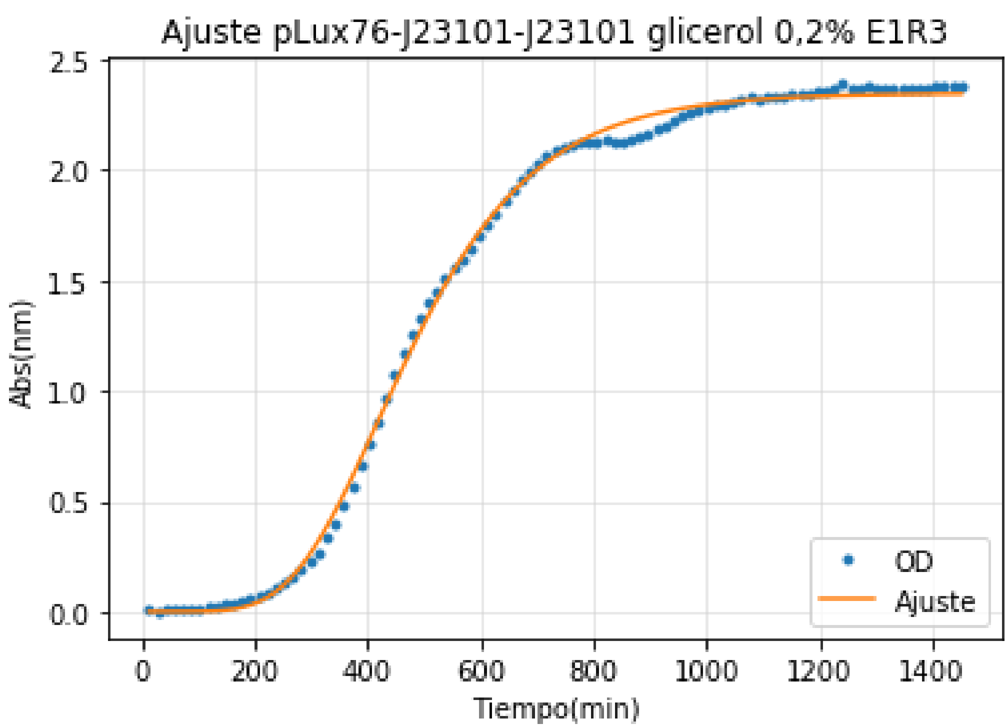




In [36]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-std-std glicerol rep 3
y18= np.log(odpssg3)-np.log(np.min(odpssg3))
print('Min OD = %e'%((np.min(odpssg3))))
evaly, params=Function_fit(tt,y18,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
A18= params[0]
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[28]
```

```
y2=tt[49]
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpssg3,label='OD pLux76-J23101-J23101 E1R3')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:50],odpssg3[28:50],label='OD pLux76-J23101-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

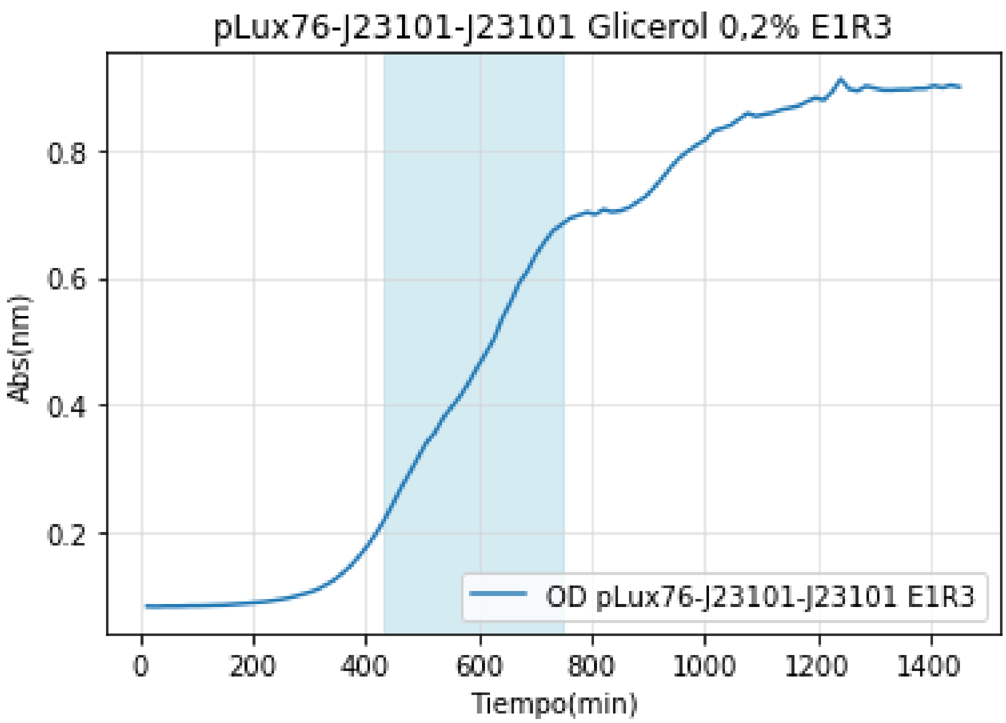
```
Min OD = 8.370000e-02
[  2.34653607e+00   5.61472351e-03   2.64125608e+02]
```

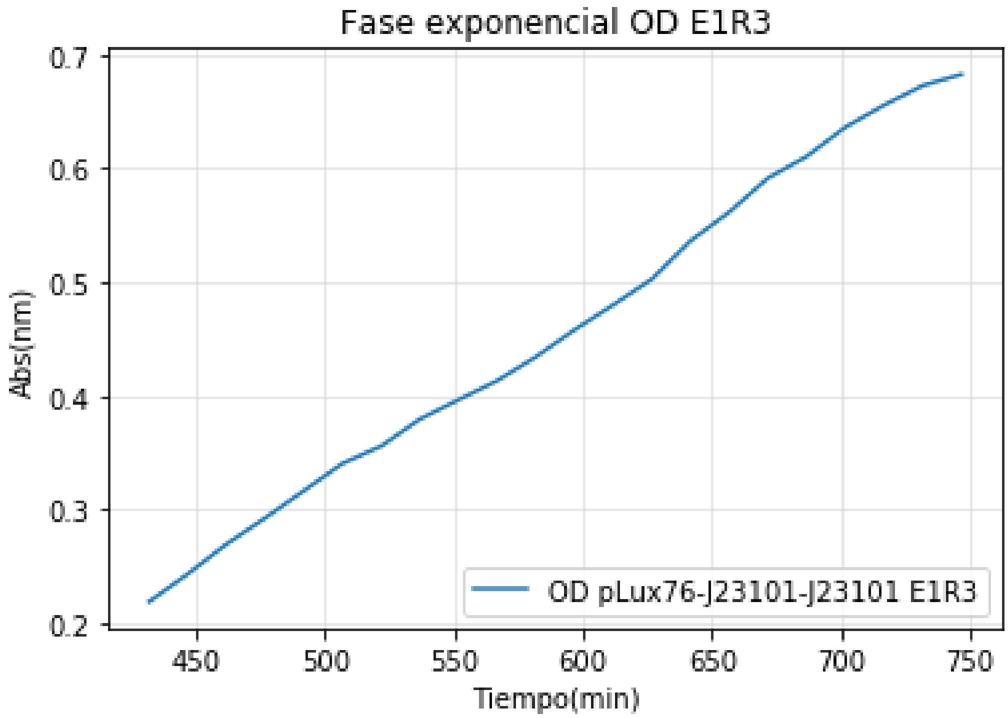



```
A=2.346536e+00
um=5.614724e-03
```

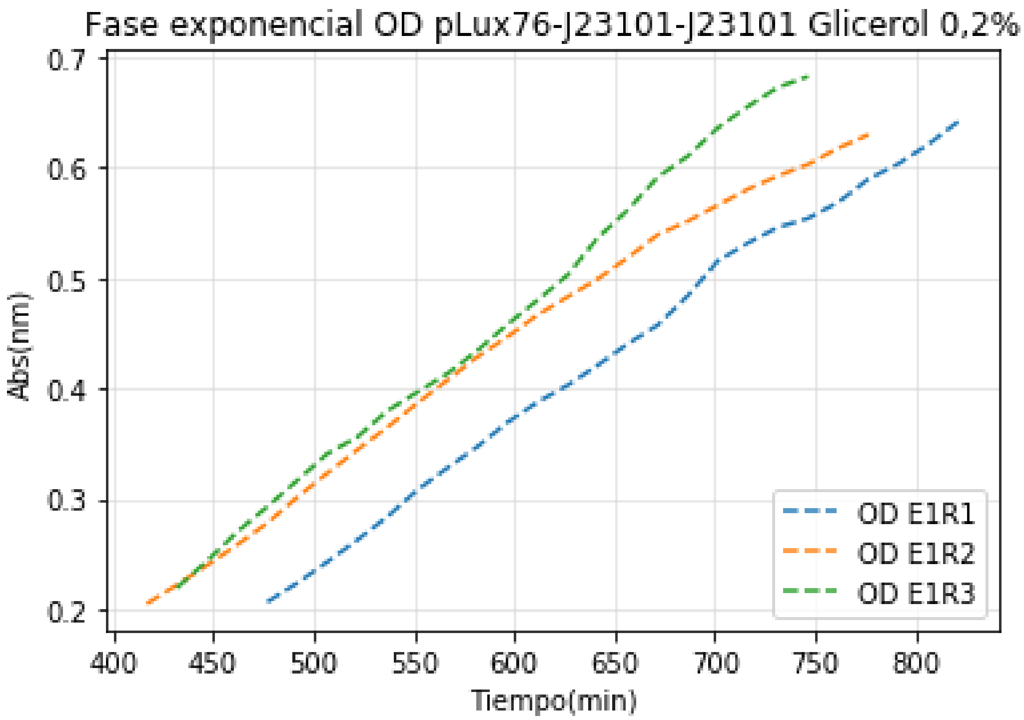
```
l=2.641256e+02
Tm=4.178718e+02
doubpe=1.234517e+02
ext=3.086293e+02
Tfinal=7.265011e+02
```

Out[36]: <matplotlib.legend.Legend at 0x1aba96a3160>

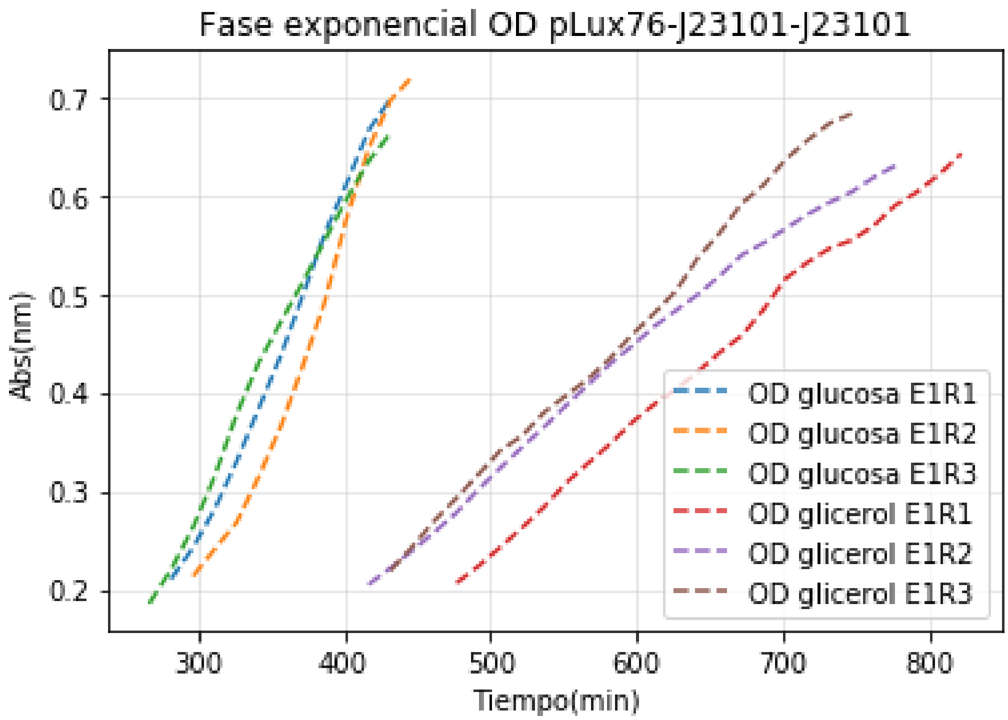





```
In [37]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[31:55],odpssg1[31:55],'--',label='OD E1R1')
         plt.plot(tt[27:52],odpssg2[27:52],'--',label='OD E1R2')
         plt.plot(tt[28:50],odpssg3[28:50],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[37]: <matplotlib.legend.Legend at 0x1aba9594278>
```



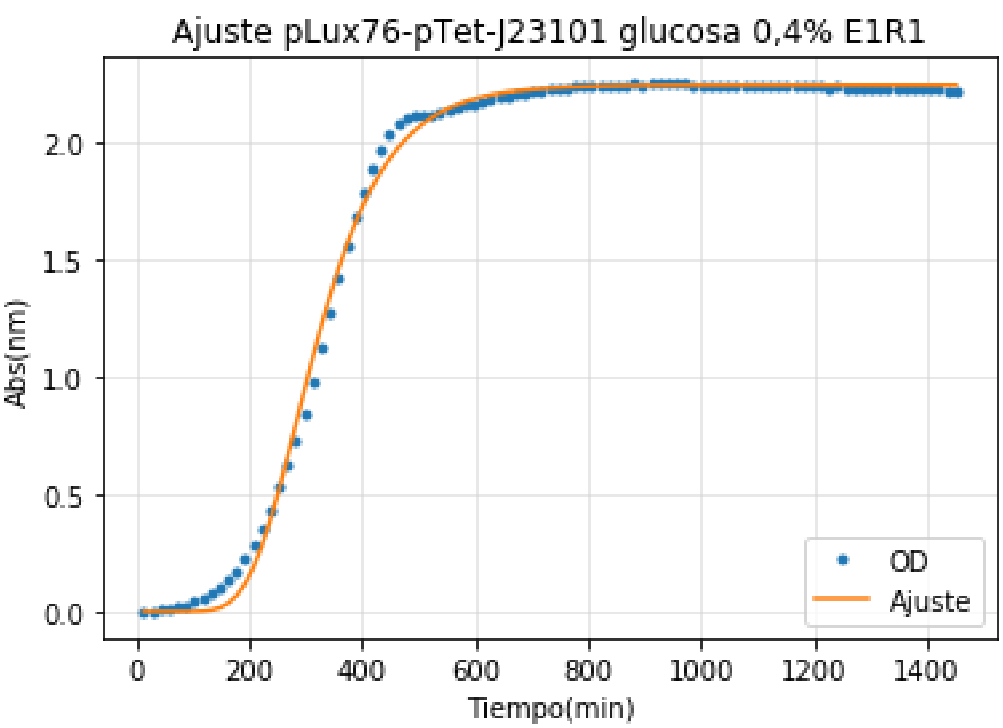

```
In [38]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[18:29],odpss1[18:29],'--',label='OD glucosa E1R1')
         plt.plot(tt[19:30],odpss2[19:30],'--',label='OD glucosa E1R2')
         plt.plot(tt[17:29],odpss3[17:29],'--',label='OD glucosa E1R3')
         plt.plot(tt[31:55],odpssg1[31:55],'--',label='OD glicerol E1R1')
         plt.plot(tt[27:52],odpssg2[27:52],'--',label='OD glicerol E1R2')
         plt.plot(tt[28:50],odpssg3[28:50],'--',label='OD glicerol E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[38]: <matplotlib.legend.Legend at 0x1aba93e4b70>
```



In [39]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-pTet-std glucosa rep 1
y19= np.log(odpts1)-np.log(np.min(odpts1))
print('Min OD = %e'%((np.min(odpts1))))
evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste pLux76-pTet-J23101 glucosa 0,4%
A19= params[0]
um19=params[1]
l19=params[2]
print('A=%e'%(A19))
print('um=%e'%(um19))
print('l=%e'%(l19))

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

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

```
y2=tt[29]
plt.figure()
plt.title('pLux76-pTet-J23101 Glucosa 0,4% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpts1,label='OD pLux76-pTet-J23101 E1R1 ')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odpts1[19:30],label='OD pLux76-pTet-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

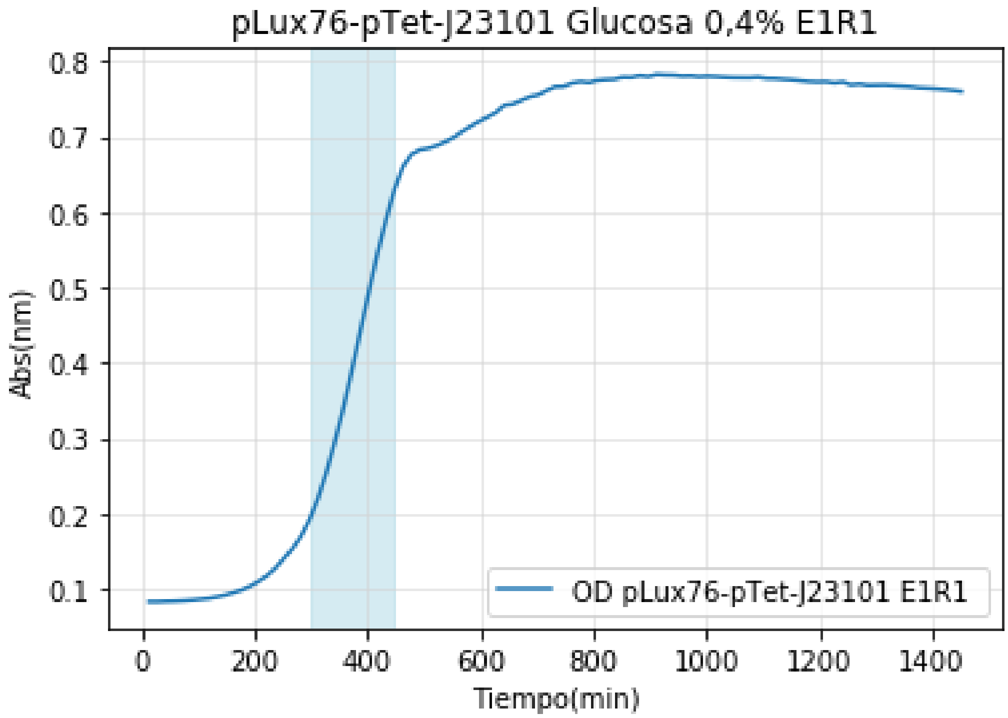
```
Min OD = 8.280000e-02
[  2.24075080e+00   9.58089970e-03   1.98256523e+02]
```

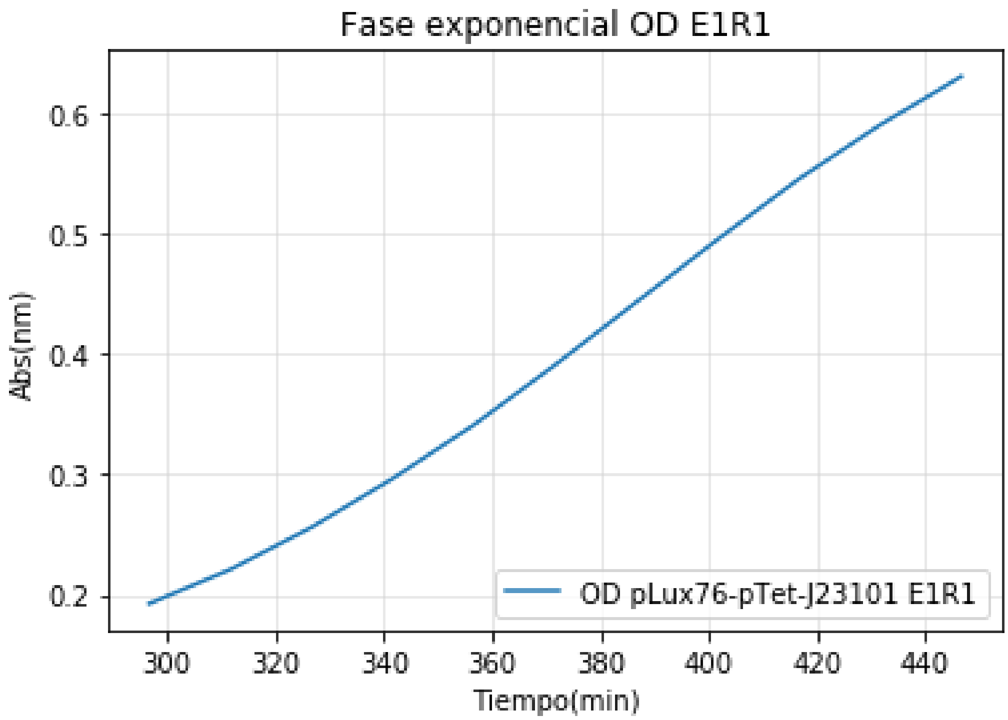



```
A=2.240751e+00
um=9.580900e-03
```

```
l=1.982565e+02
Tm=2.842950e+02
doubpe=7.234677e+01
ext=1.446935e+02
Tfinal=4.289886e+02
```

```
Out[39]: <matplotlib.legend.Legend at 0x1aba9382668>
```

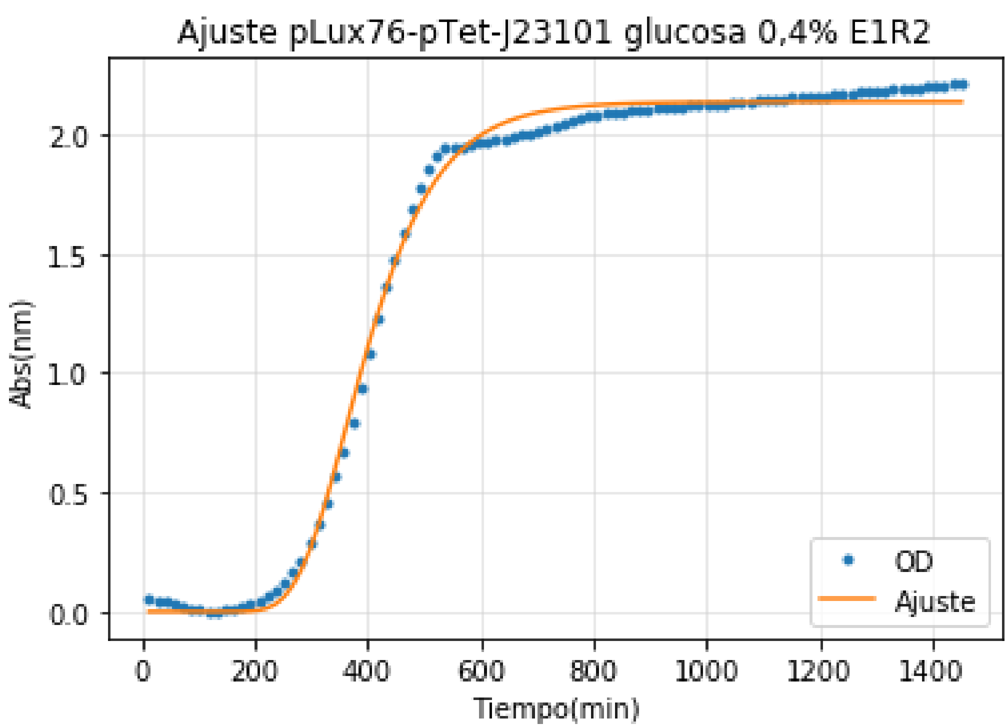





```
In [40]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-pTet-std glucosa rep 2
         y20= np.log(odpts2)-np.log(np.min(odpts2))
         print('Min OD = %e'%((np.min(odpts2))))
         evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste pLux76-pTet-J23101 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[24]
```

```python
y2=tt[35]
plt.figure()
plt.title('pLux76-pTet-J23101 Glucosa 0,4% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpts2,label='OD pLux76-pTet-J23101 E1R2 ')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:36],odpts2[24:36],label='OD pLux76-pTet-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

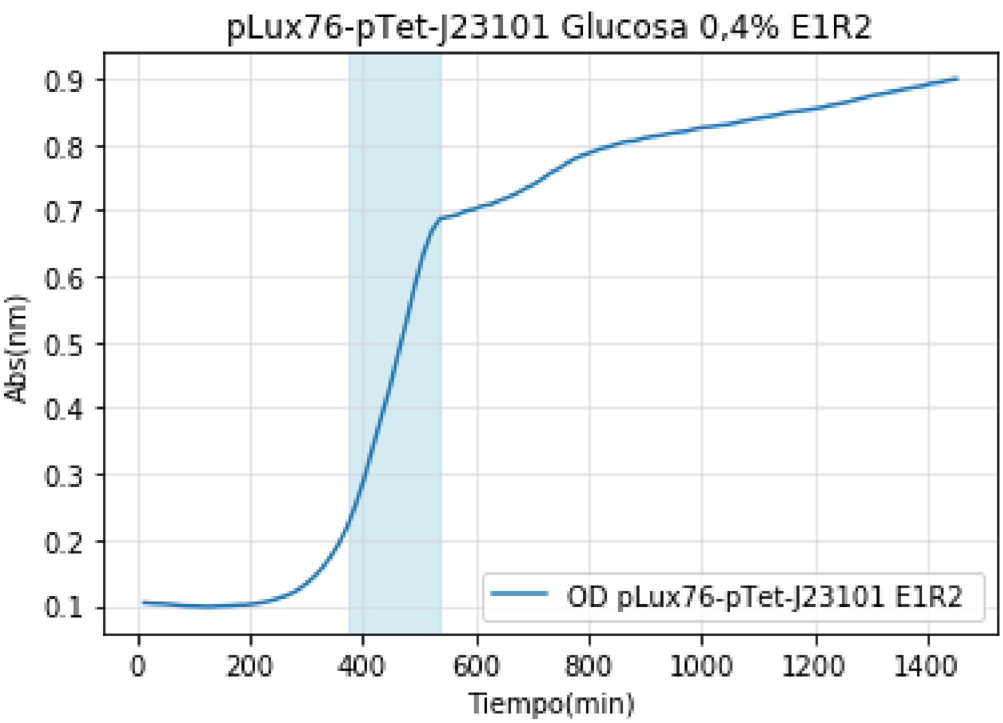
```
Min OD = 9.870000e-02
[  2.13523313e+00   8.92601798e-03   2.74552472e+02]
```

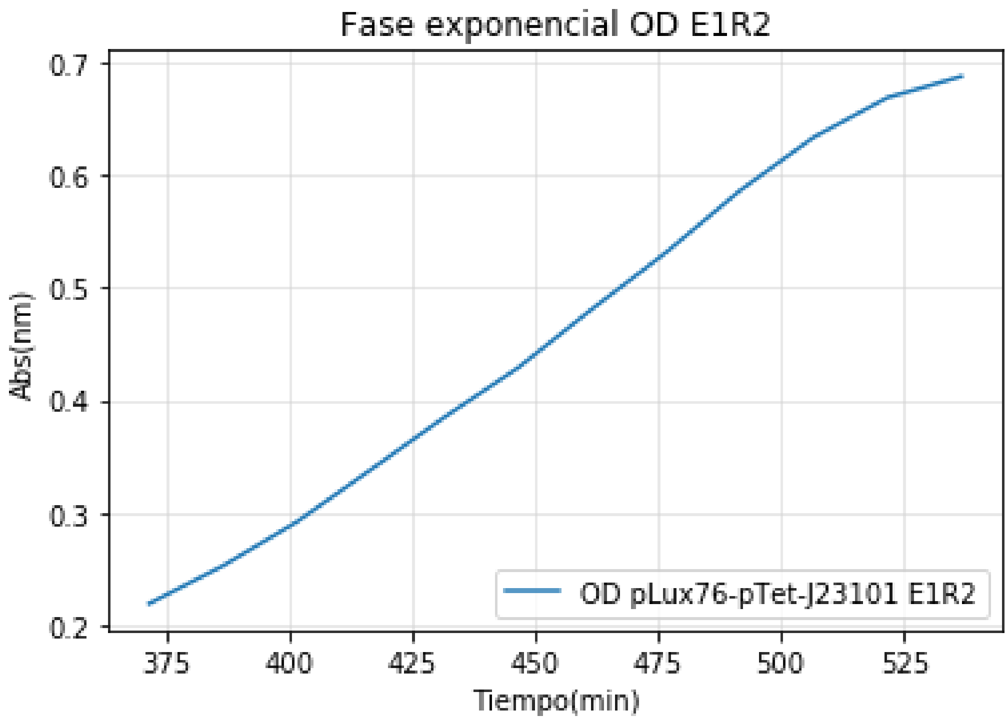



```
A=2.135233e+00
um=8.926018e-03
```

```
l=2.745525e+02
Tm=3.625546e+02
doubpe=7.765469e+01
ext=1.553094e+02
Tfinal=5.178640e+02
```

```
Out[40]: <matplotlib.legend.Legend at 0x1aba9440828>
```

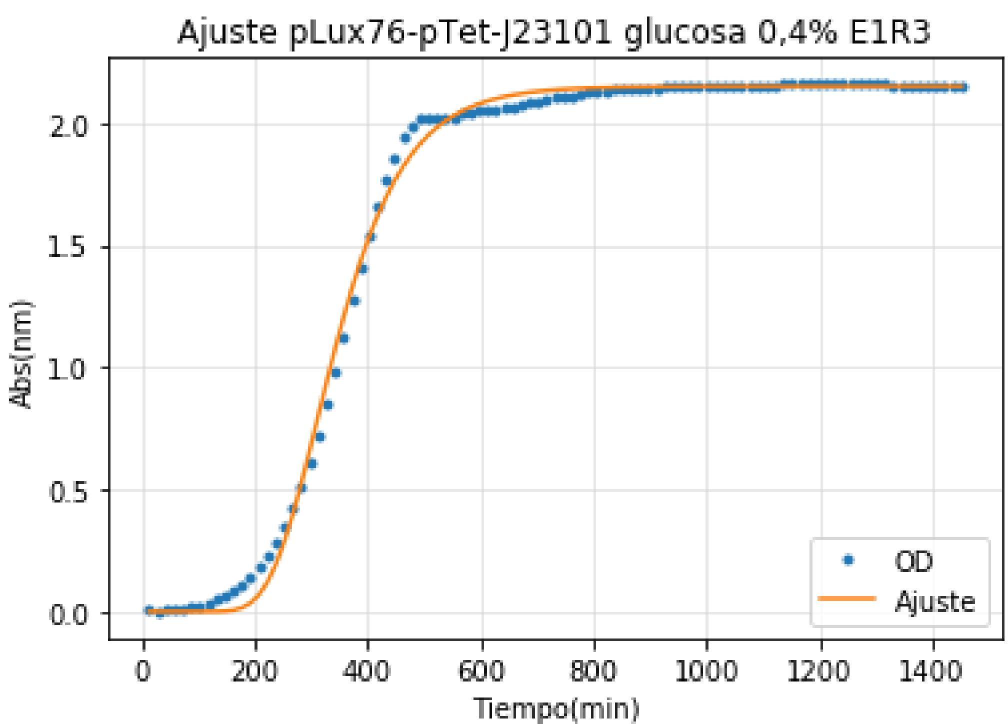




In [41]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-pTet-std glucosa rep 3
y21= np.log(odpts3)-np.log(np.min(odpts3))
print('Min OD = %e'%((np.min(odpts3))))
evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste pLux76-pTet-J23101 glucosa 0,4%
A21= params[0]
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[20]
```

```
y2=tt[31]
plt.figure()
plt.title('pLux76-pTet-J23101 Glucosa 0,4% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpts3,label='OD pLux76-pTet-J23101 E1R3 ')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[20:32],odpts3[20:32],label='OD pLux76-pTet-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

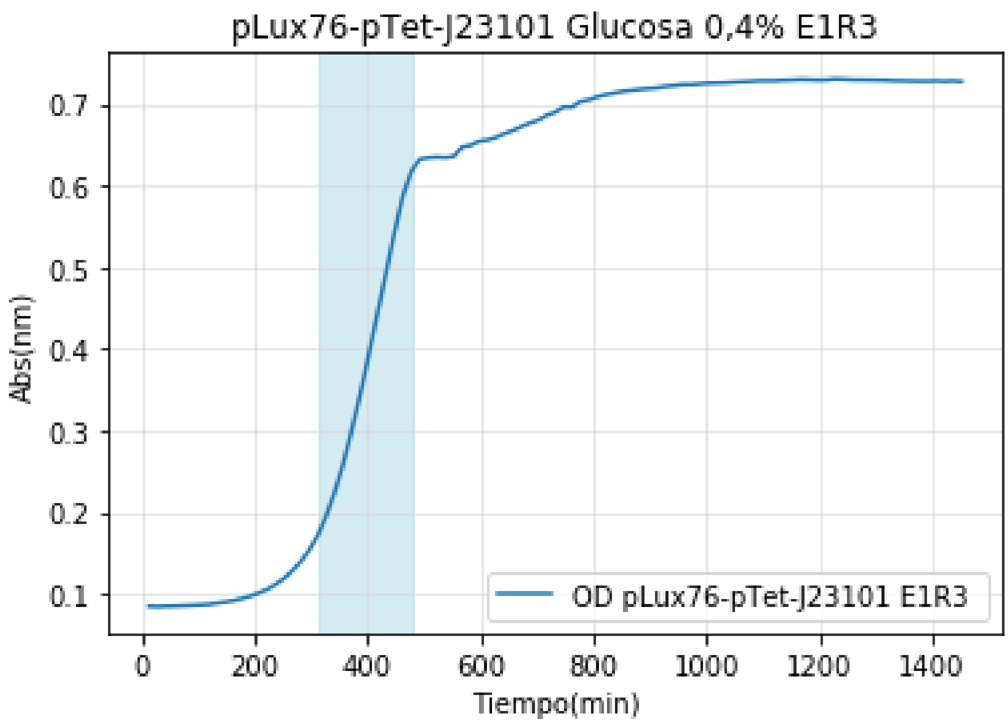
```
Min OD = 8.440000e-02
[  2.14834513e+00   9.38698420e-03   2.26337137e+02]
```

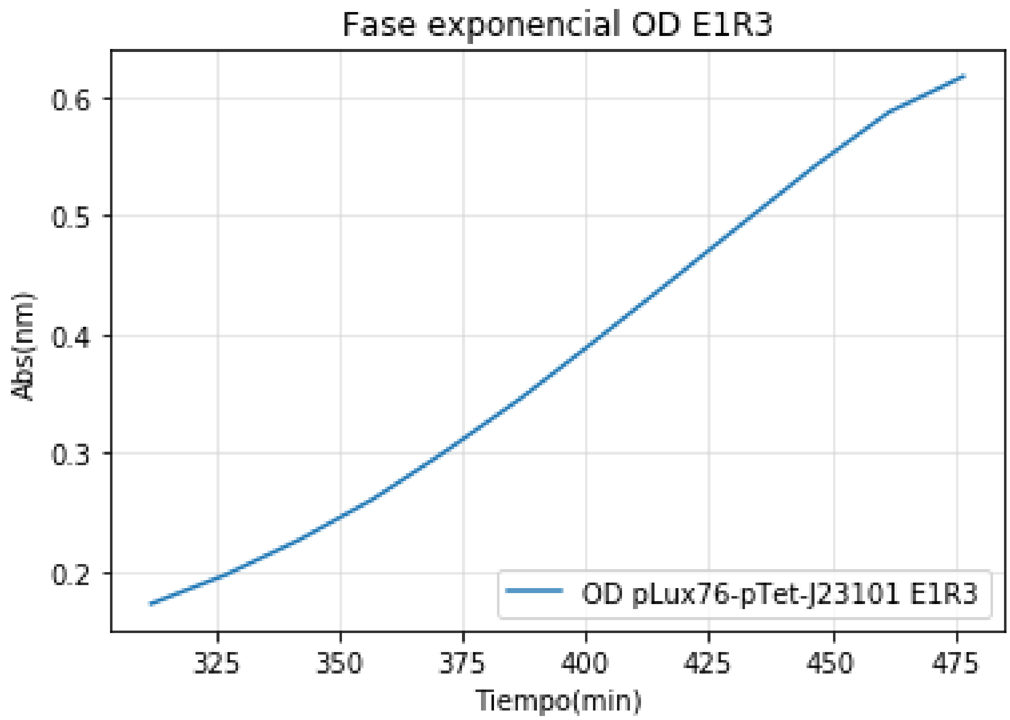



```
A=2.148345e+00
um=9.386984e-03
```

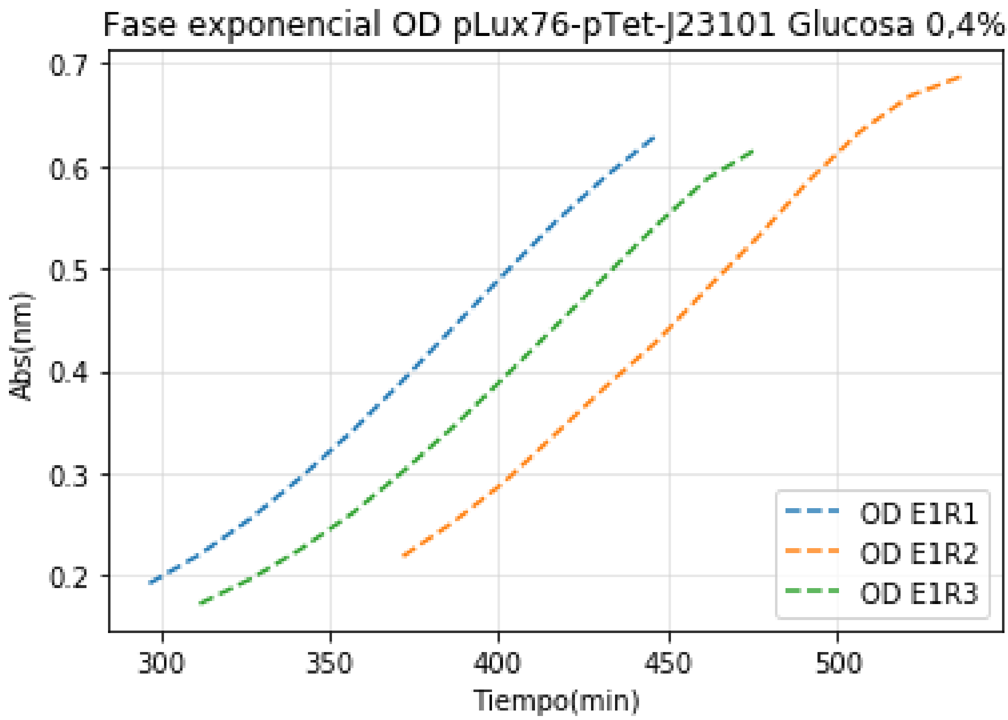
```
l=2.263371e+02
Tm=3.105316e+02
doubpe=7.384131e+01
ext=1.476826e+02
Tfinal=4.582142e+02
```

```
Out[41]: <matplotlib.legend.Legend at 0x1aba997ea20>
```





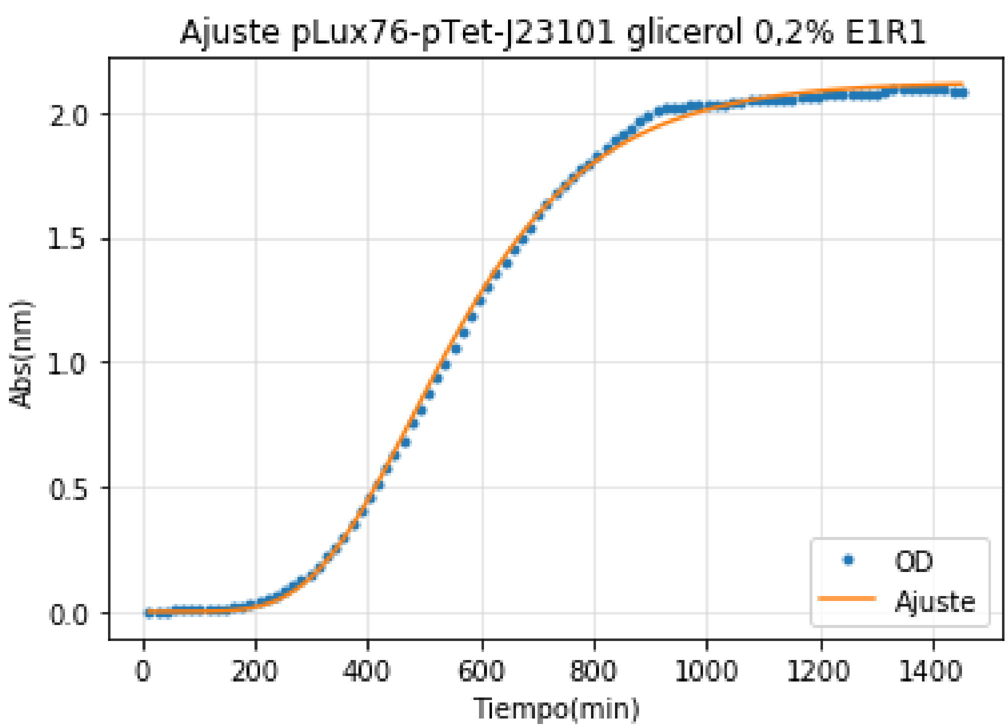

```
In [42]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pTet-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:30],odpts1[19:30],'--',label='OD E1R1')
         plt.plot(tt[24:36],odpts2[24:36],'--',label='OD E1R2')
         plt.plot(tt[20:32],odpts3[20:32],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[42]: <matplotlib.legend.Legend at 0x1aba97e4278>
```




```
In [43]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #pLux-pTet-std glicerol rep 1
         y22= np.log(odptsg1)-np.log(np.min(odptsg1))
         print('Min OD = %e'%((np.min(odptsg1))))
         evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste pLux76-pTet-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[32]
```

```python
y2=tt[59]
plt.figure()
plt.title('pLux76-pTet-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odptsg1,label='OD pLux76-pTet-J23101 E1R1')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[32:60],odptsg1[32:60],label='OD pLux76-pTet-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

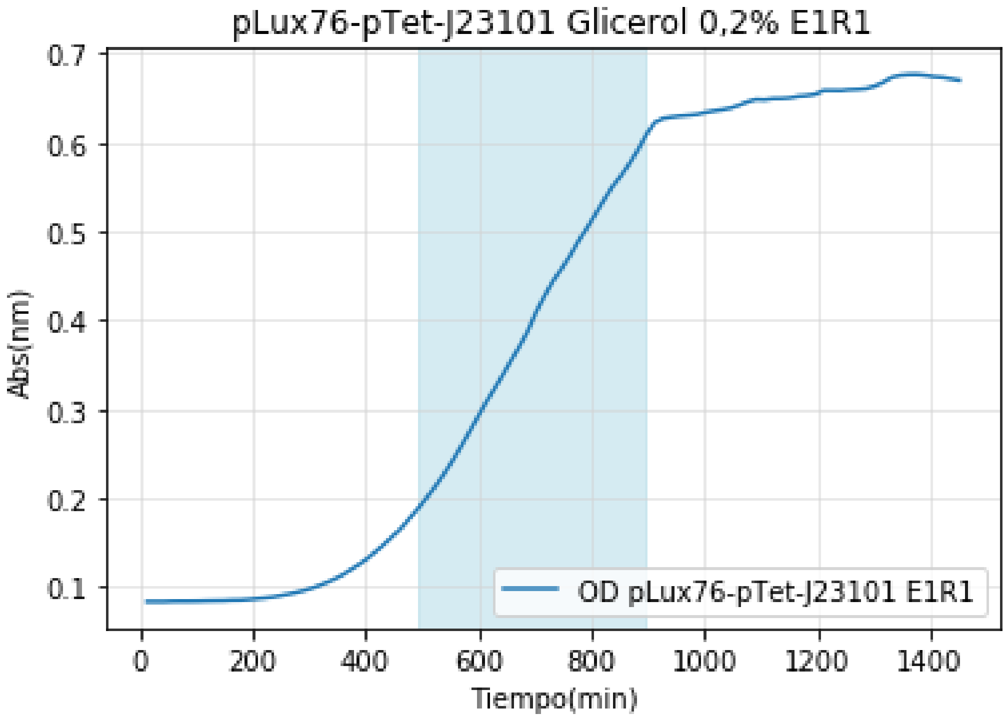
```
Min OD = 8.330000e-02
[  2.12107601e+00   4.36676099e-03   3.00000000e+02]
```

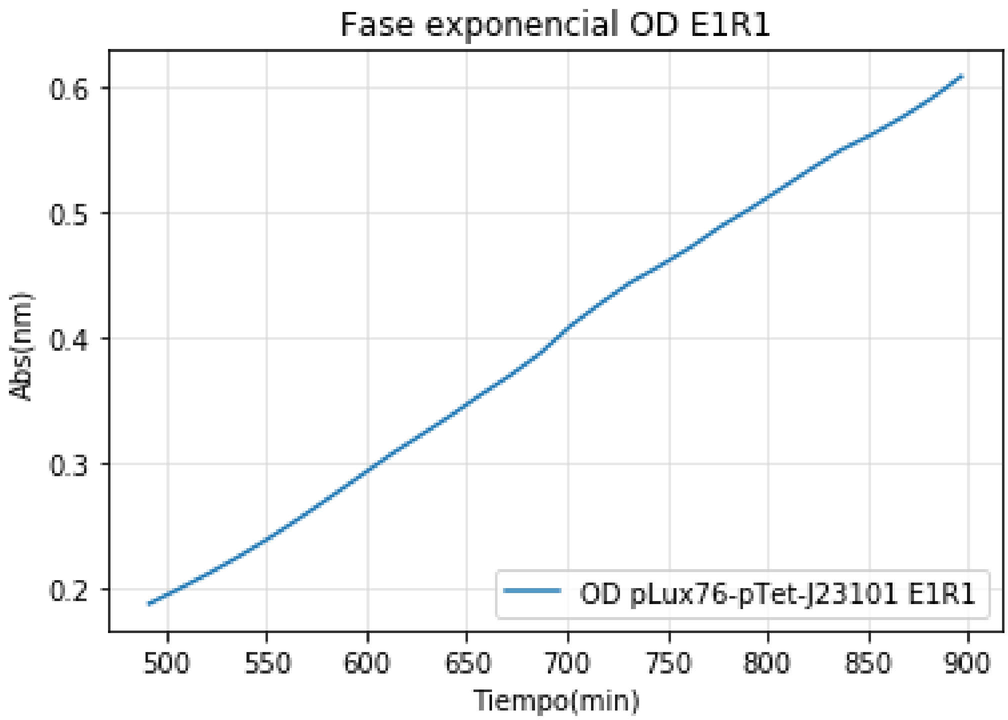



```
A=2.121076e+00
um=4.366761e-03
```

```
l=3.000000e+02
Tm=4.786909e+02
doubpe=1.587326e+02
ext=3.968314e+02
Tfinal=8.755223e+02
```

Out[43]: <matplotlib.legend.Legend at 0x1aba7900048>

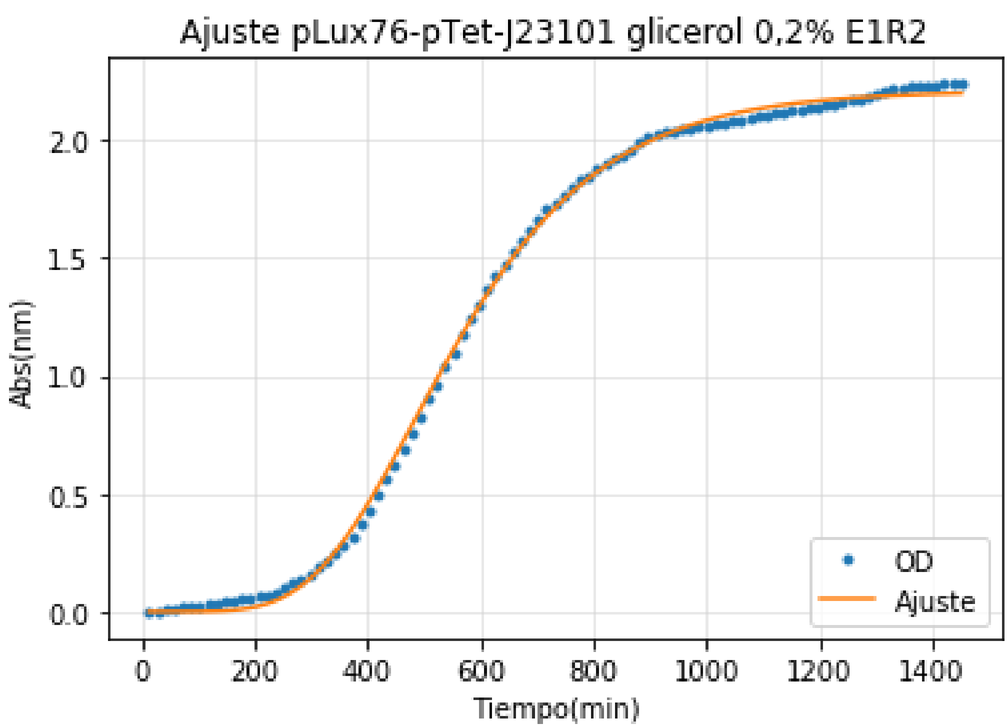




In [44]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pTet-std glicerol rep 2
y23= np.log(odptsg2)-np.log(np.min(odptsg2))
print('Min OD = %e'%((np.min(odptsg2))))
evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste pLux76-pTet-J23101 glicerol 0,2%
A23= params[0]
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[32]
```

```
y2=tt[59]
plt.figure()
plt.title('pLux76-pTet-J23101 Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odptsg2,label='OD pLux76-pTet-J23101 E1R2')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[32:60],odptsg2[32:60],label='OD pLux76-pTet-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

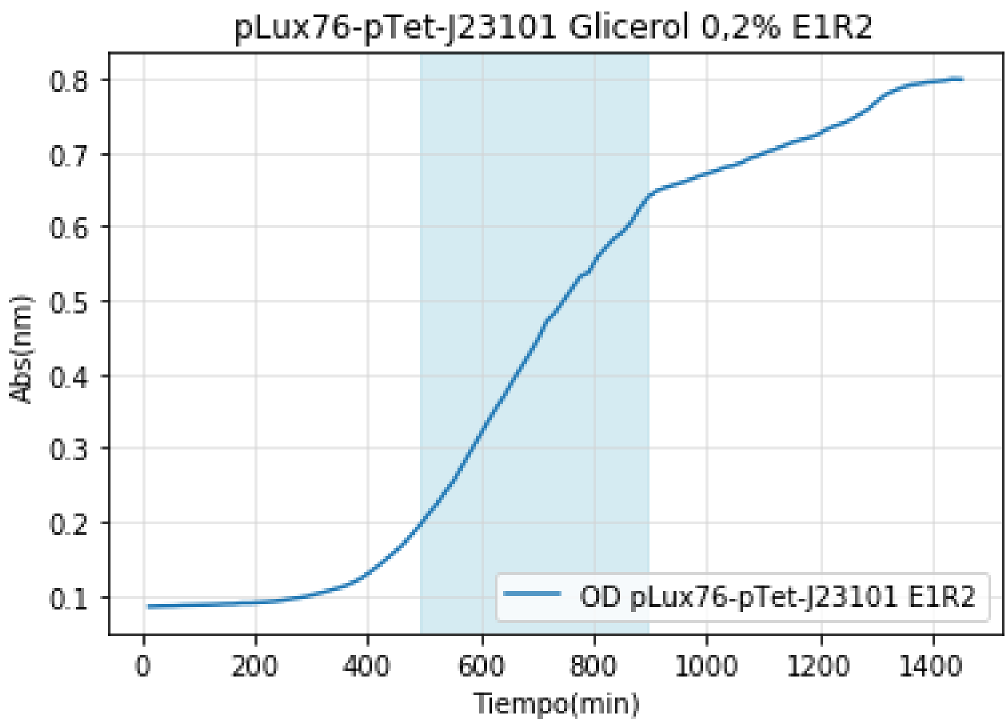
```
Min OD = 8.520000e-02
[  2.21178147e+00   4.45495834e-03   3.00000000e+02]
```

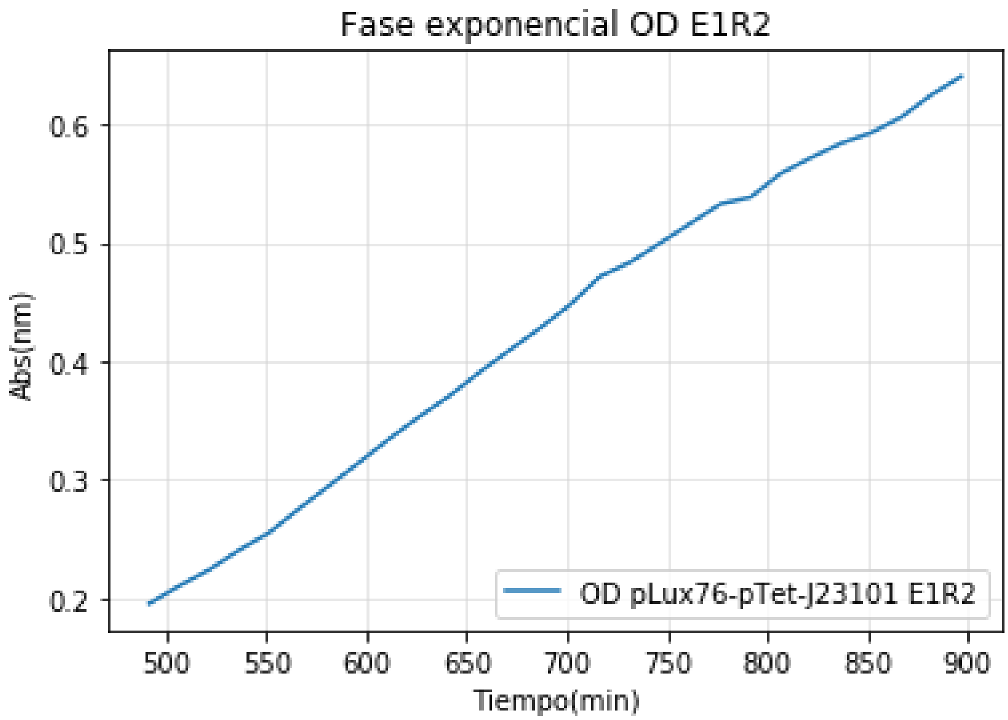



```
A=2.211781e+00
um=4.454958e-03
```

```
l=3.000000e+02
Tm=4.826434e+02
doubpe=1.555900e+02
ext=3.889751e+02
Tfinal=8.716186e+02
```

Out[44]: <matplotlib.legend.Legend at 0x1aba9abb240>

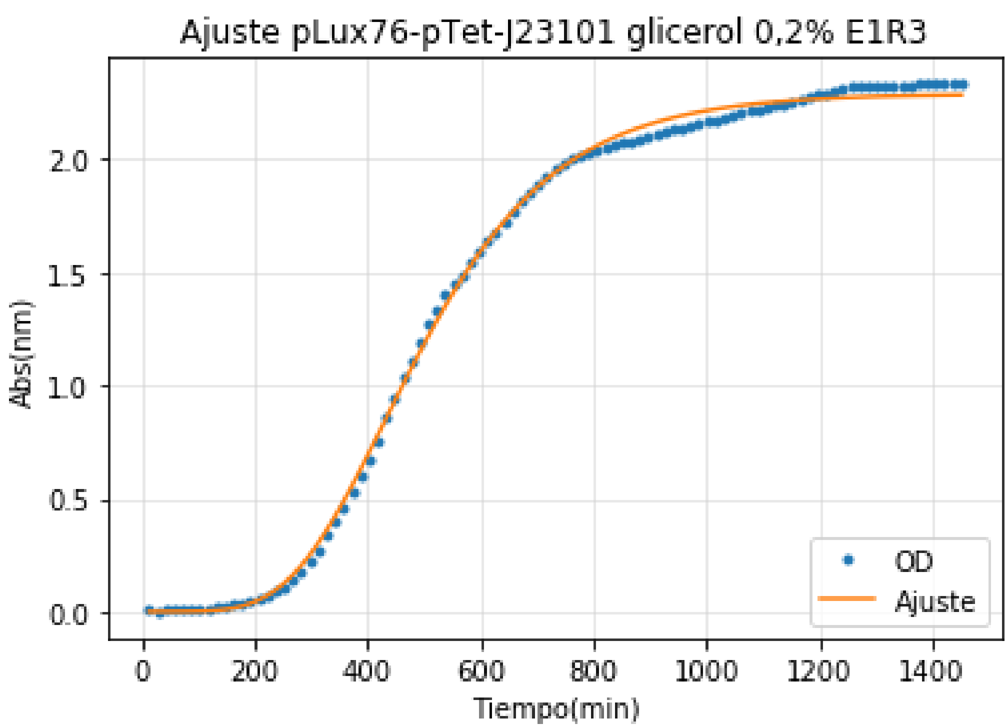




In [45]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pTet-std glicerol rep 3
y24= np.log(odptsg3)-np.log(np.min(odptsg3))
print('Min OD = %e'%((np.min(odptsg3))))
evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste pLux76-pTet-J23101 glicerol 0,2%
A24= params[0]
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[28]
```

```
y2=tt[52]
plt.figure()
plt.title('pLux76-pTet-J23101 Glicerol 0,2% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odptsg3,label='OD pLux76-pTet-J23101 E1R3')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:53],odptsg3[28:53],label='OD pLux76-pTet-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

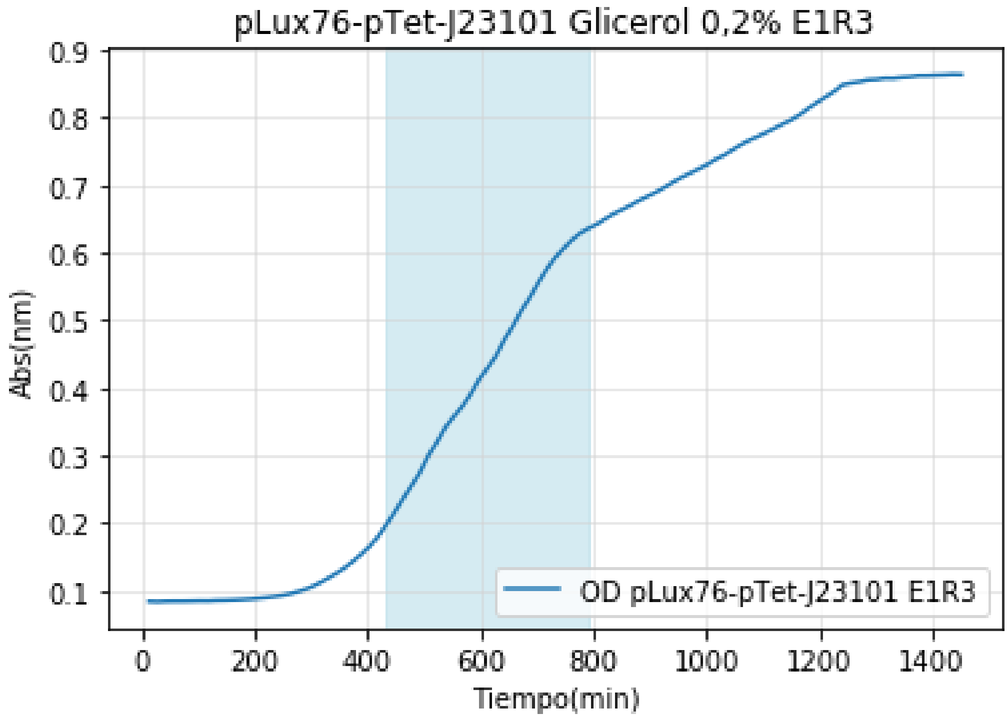
```
Min OD = 8.380000e-02
[  2.29047087e+00   5.02597702e-03   2.61934082e+02]
```

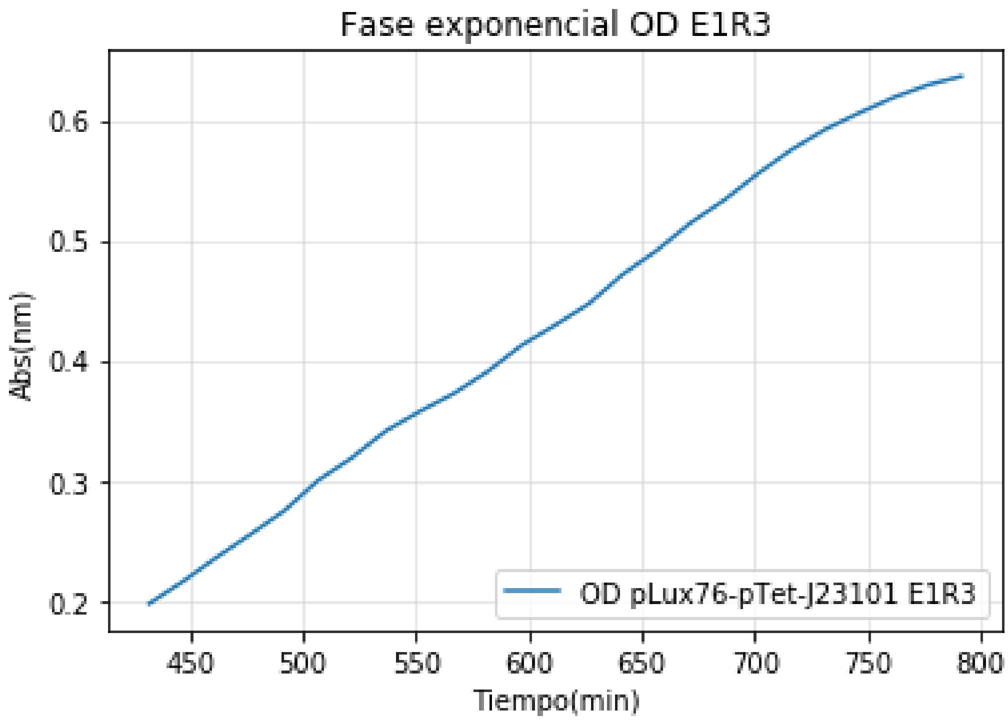



```
A=2.290471e+00
um=5.025977e-03
```

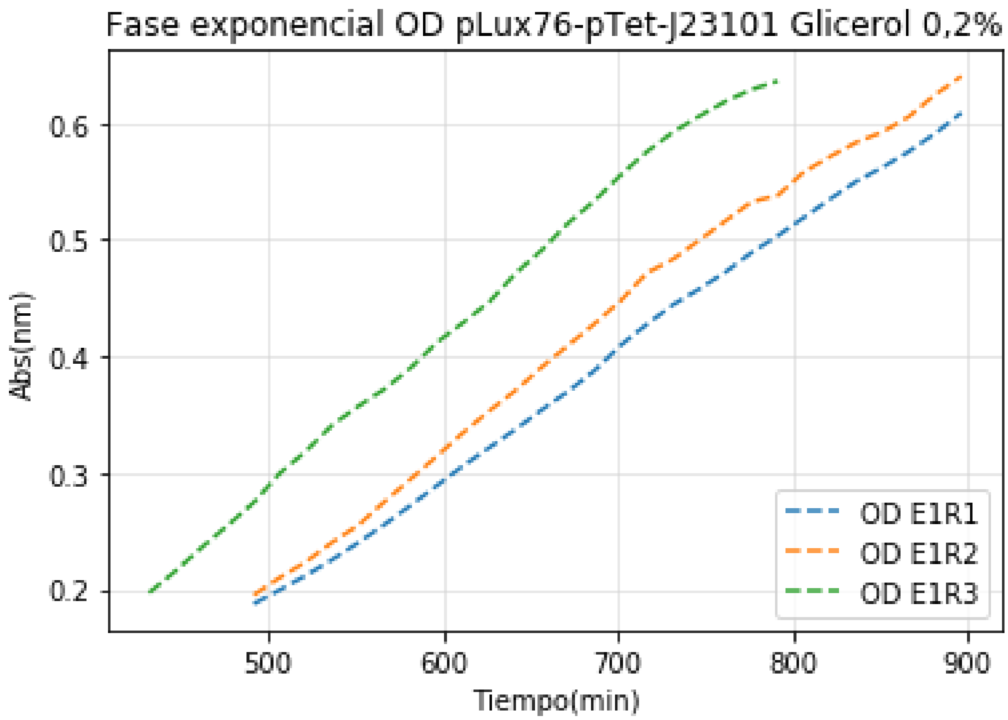
```
l=2.619341e+02
Tm=4.295865e+02
doubpe=1.379129e+02
ext=3.447823e+02
Tfinal=7.743688e+02
```

Out[45]: <matplotlib.legend.Legend at 0x1aba9ccb748>

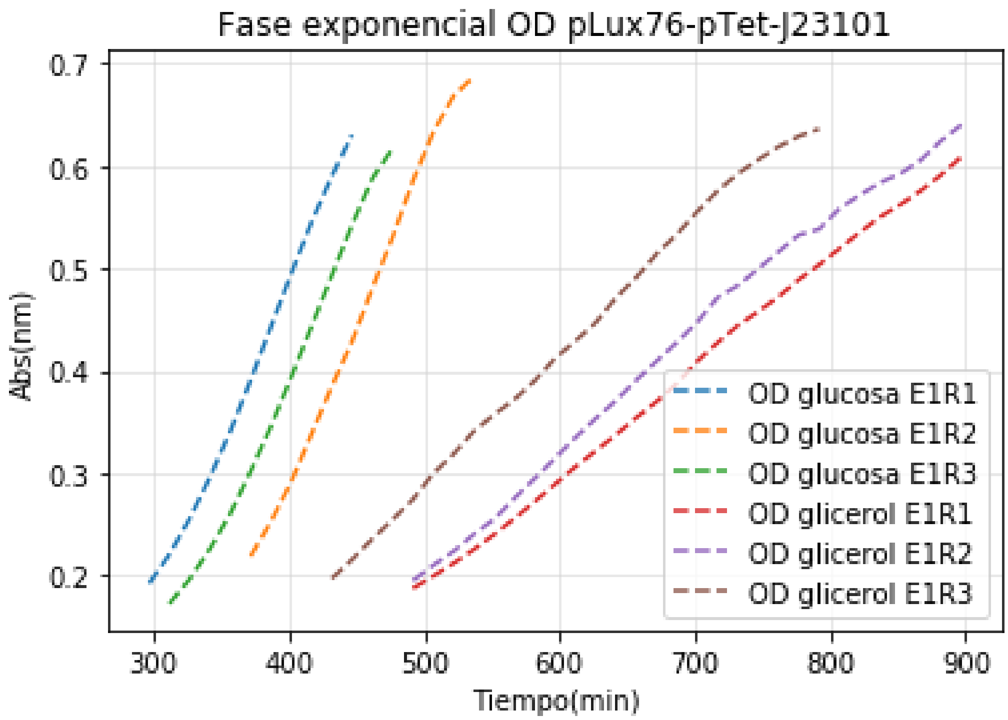





```
In [46]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pTet-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[32:60],odptsg1[32:60],'--',label='OD E1R1')
         plt.plot(tt[32:60],odptsg2[32:60],'--',label='OD E1R2')
         plt.plot(tt[28:53],odptsg3[28:53],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[46]: <matplotlib.legend.Legend at 0x1aba9df3b00>
```



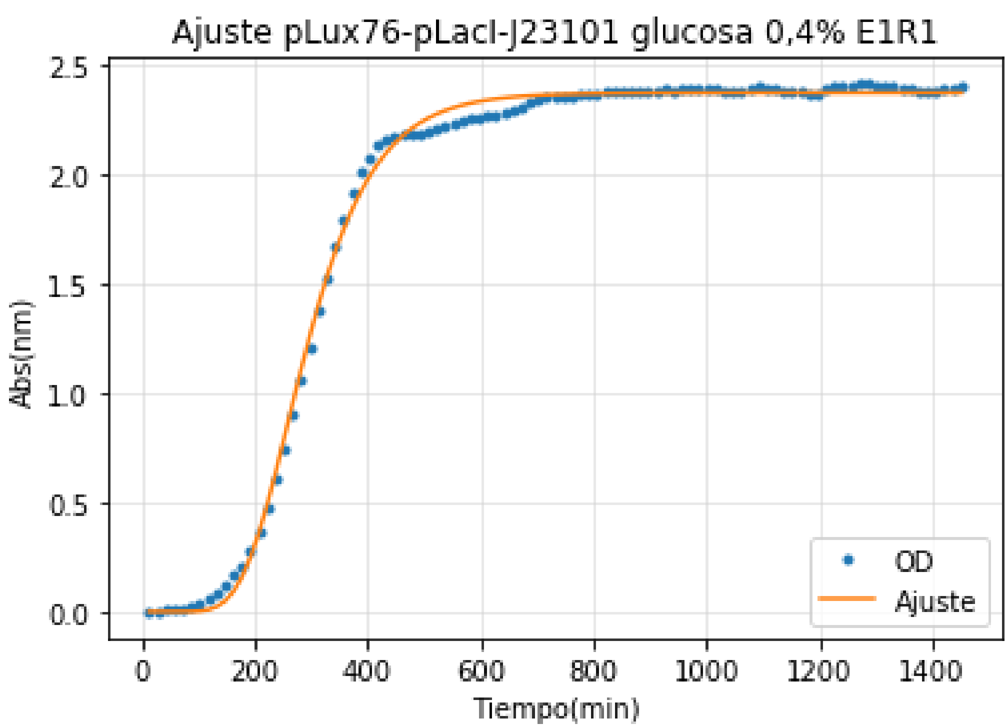

```
In [47]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pTet-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:30],odpts1[19:30],'--',label='OD glucosa E1R1')
         plt.plot(tt[24:36],odpts2[24:36],'--',label='OD glucosa E1R2')
         plt.plot(tt[20:32],odpts3[20:32],'--',label='OD glucosa E1R3')
         plt.plot(tt[32:60],odptsg1[32:60],'--',label='OD glicerol E1R1')
         plt.plot(tt[32:60],odptsg2[32:60],'--',label='OD glicerol E1R2')
         plt.plot(tt[28:53],odptsg3[28:53],'--',label='OD glicerol E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[47]: <matplotlib.legend.Legend at 0x1aba9bc79b0>
```



In [48]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-pLacI-std glucosa rep 1
y25= np.log(odpls1)-np.log(np.min(odpls1))
print('Min OD = %e'%((np.min(odpls1))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste pLux76-pLacI-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[27]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpls1,label='OD pLux76-pLacI-J23101 E1R1 ')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],odpls1[17:28],label='OD pLux76-pLacI-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

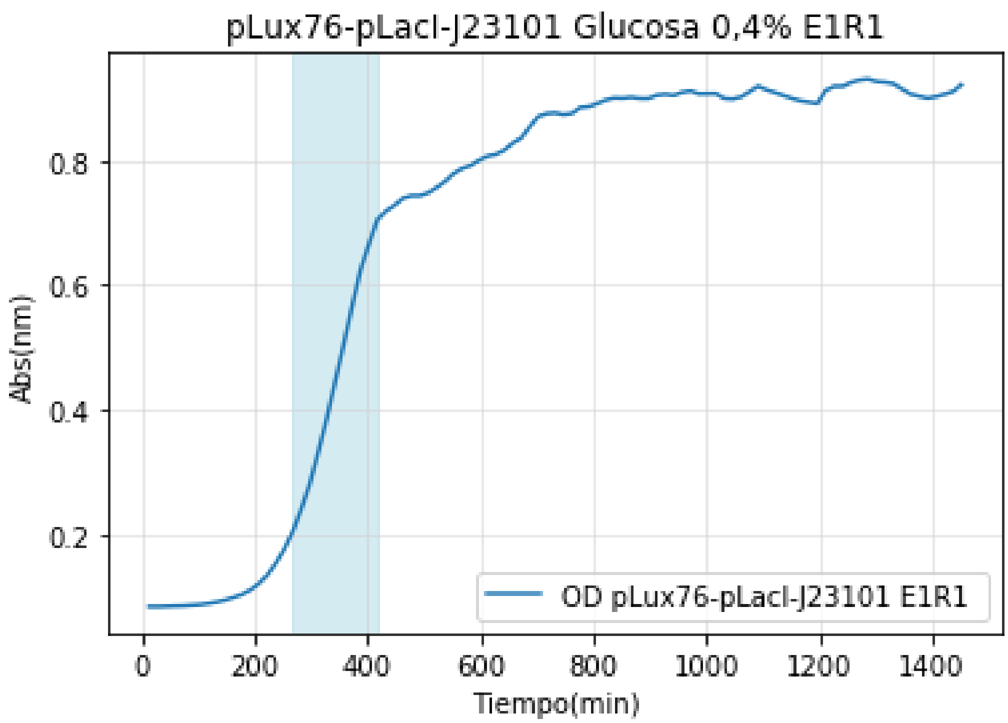
```
Min OD = 8.340000e-02
[  2.37254921e+00   1.05578438e-02   1.76137357e+02]
```

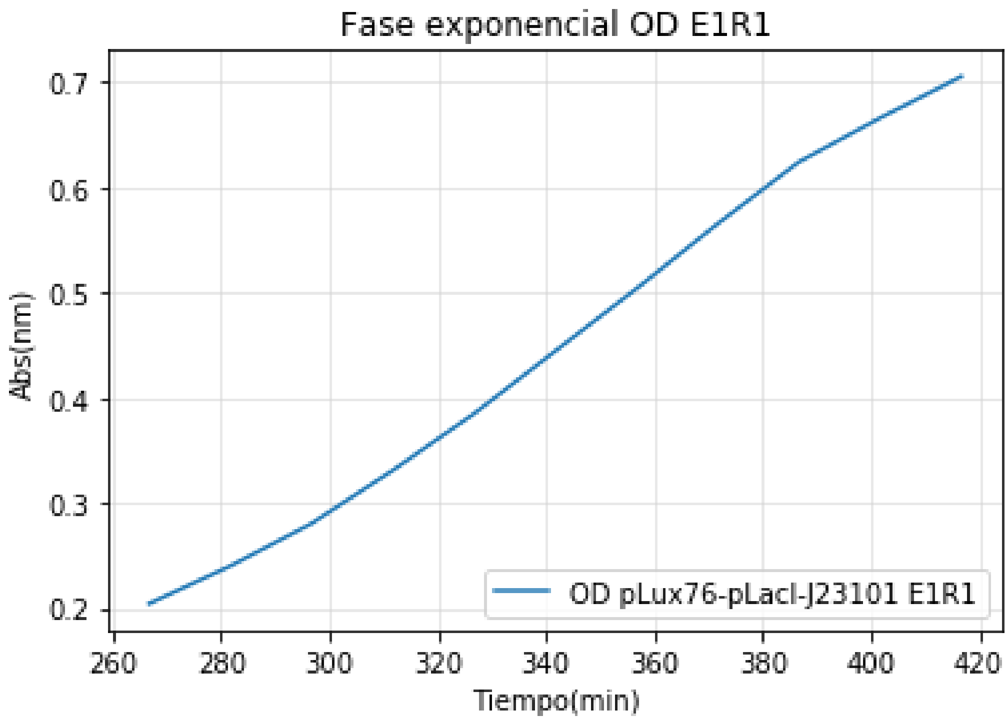



```
A=2.372549e+00
um=1.055784e-02
```

```
l=1.761374e+02
Tm=2.588069e+02
doubpe=6.565234e+01
ext=1.313047e+02
Tfinal=3.901116e+02
```

```
Out[48]: <matplotlib.legend.Legend at 0x1aba95e04e0>
```

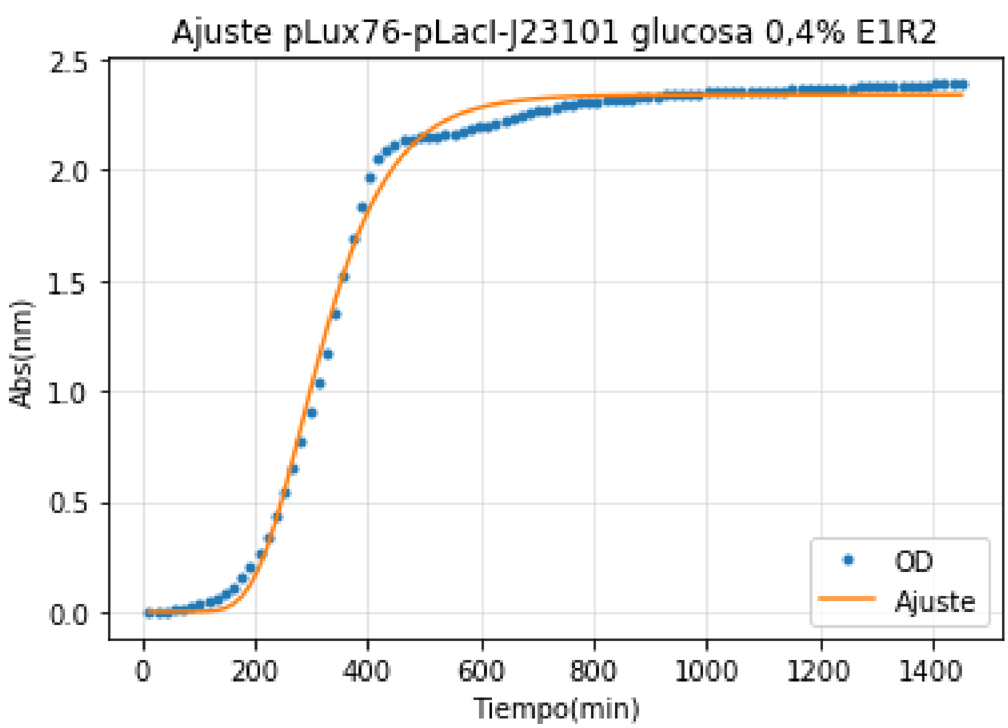




In [49]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-pLacI-std glucosa rep 2
y26= np.log(odpls2)-np.log(np.min(odpls2))
print('Min OD = %e'%((np.min(odpls2))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glucosa 0,4%
A26= params[0]
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[19]
```

```
y2=tt[29]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpls2,label='OD pLux76-pLacI-J23101 E1R2 ')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:30],odpls2[19:30],label='OD pLux76-pLacI-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

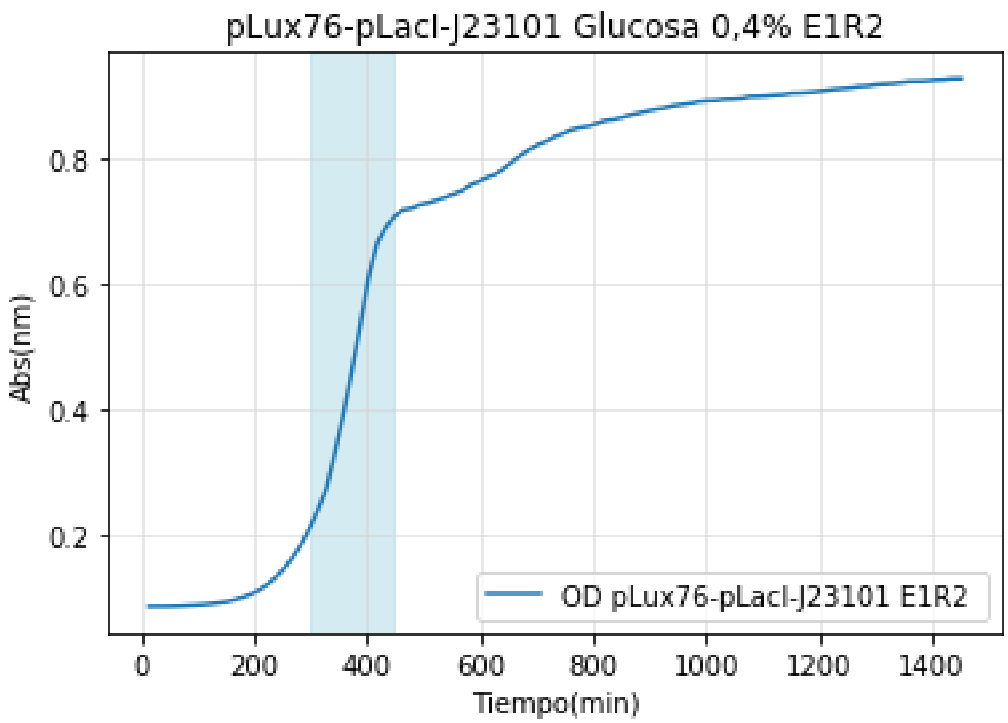
```
Min OD = 8.530000e-02
[  2.33931585e+00   1.00807393e-02   1.98362726e+02]
```

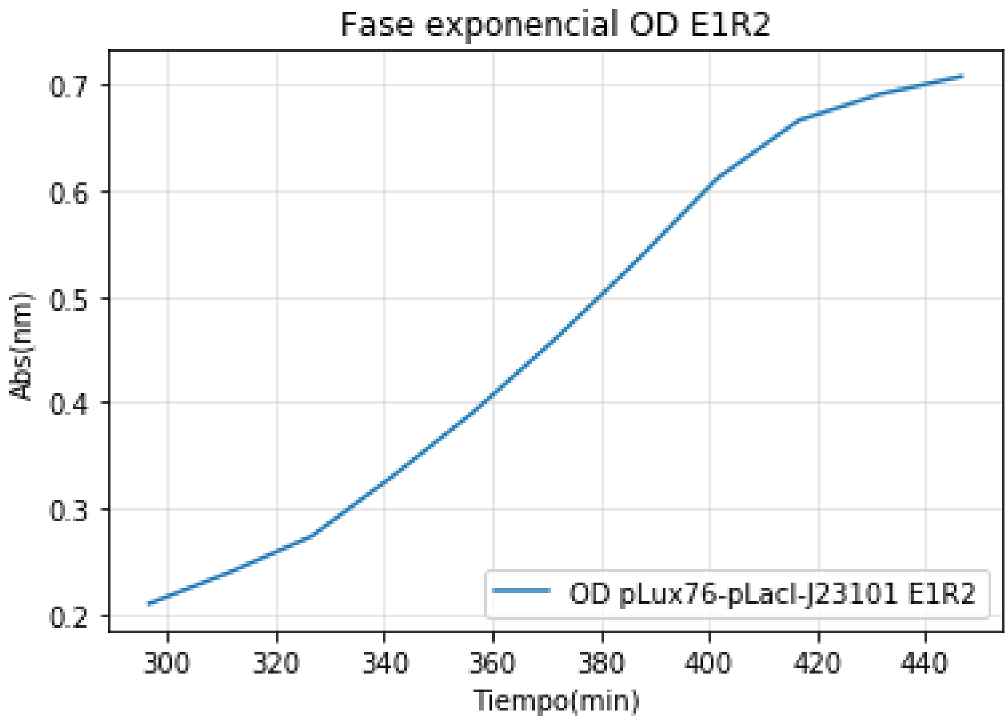



```
A=2.339316e+00
um=1.008074e-02
```

```
l=1.983627e+02
Tm=2.837321e+02
doubpe=6.875956e+01
ext=1.375191e+02
Tfinal=4.212512e+02
```

Out[49]: <matplotlib.legend.Legend at 0x1aba95c20b8>

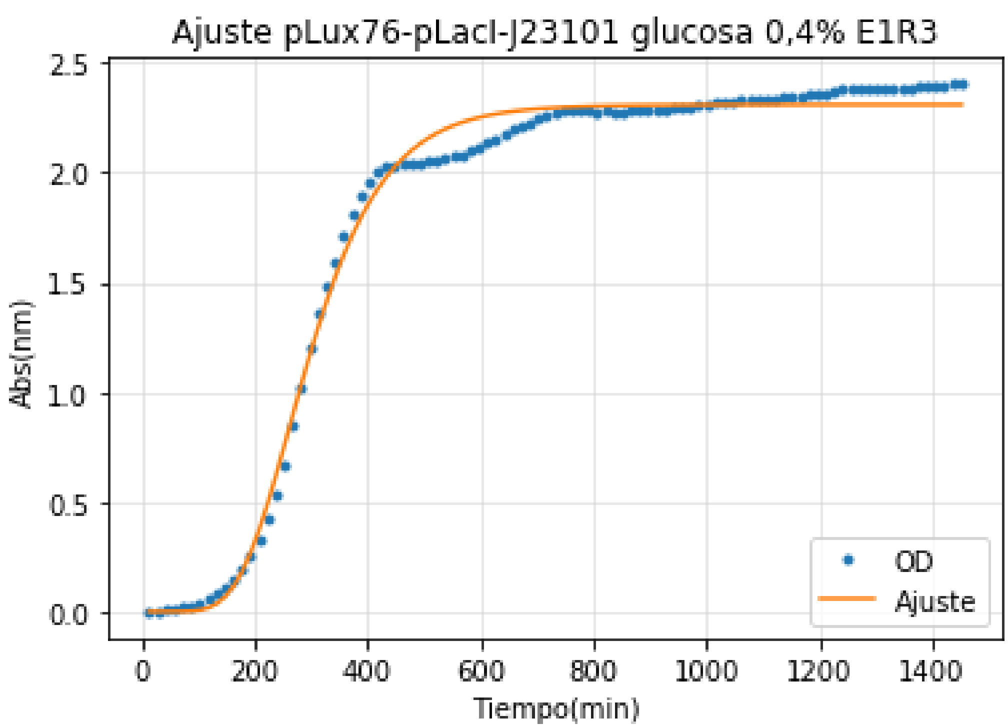





```
In [50]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-pLacI-std glucosa rep 3
         y27= np.log(odpls3)-np.log(np.min(odpls3))
         print('Min OD = %e'%((np.min(odpls3))))
         evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste pLux76-pLacI-J23101 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[29]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glucosa 0,4% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odpls3,label='OD pLux76-pLacI-J23101 E1R3 ')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:29],odpls3[17:29],label='OD pLux76-pLacI-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

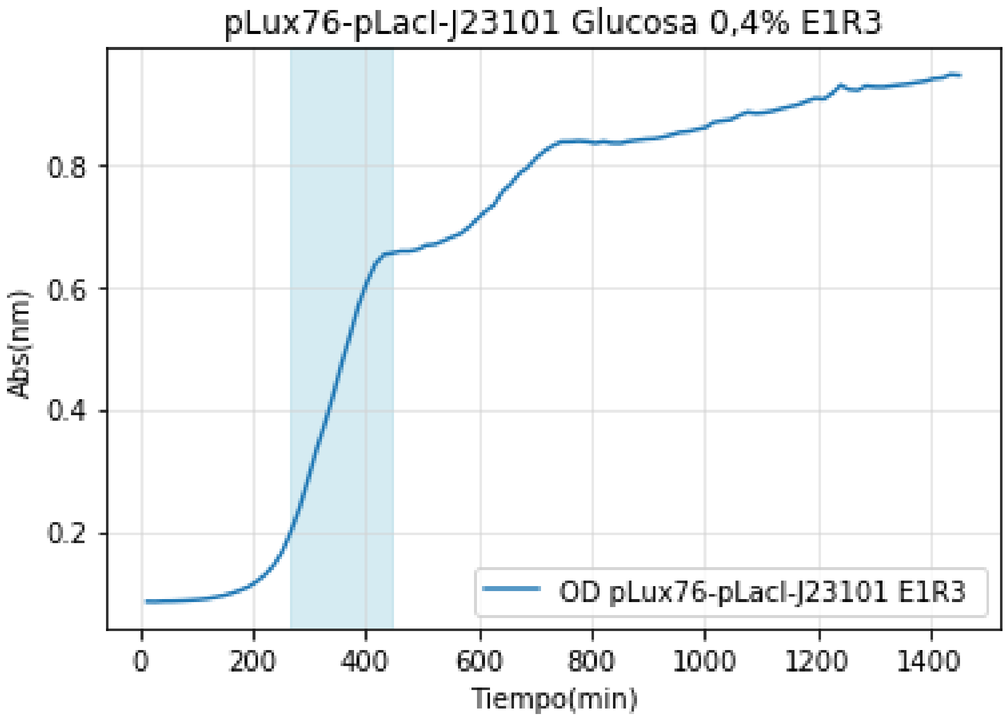
```
Min OD = 8.550000e-02
[  2.30955971e+00   9.31231591e-03   1.70378995e+02]
```

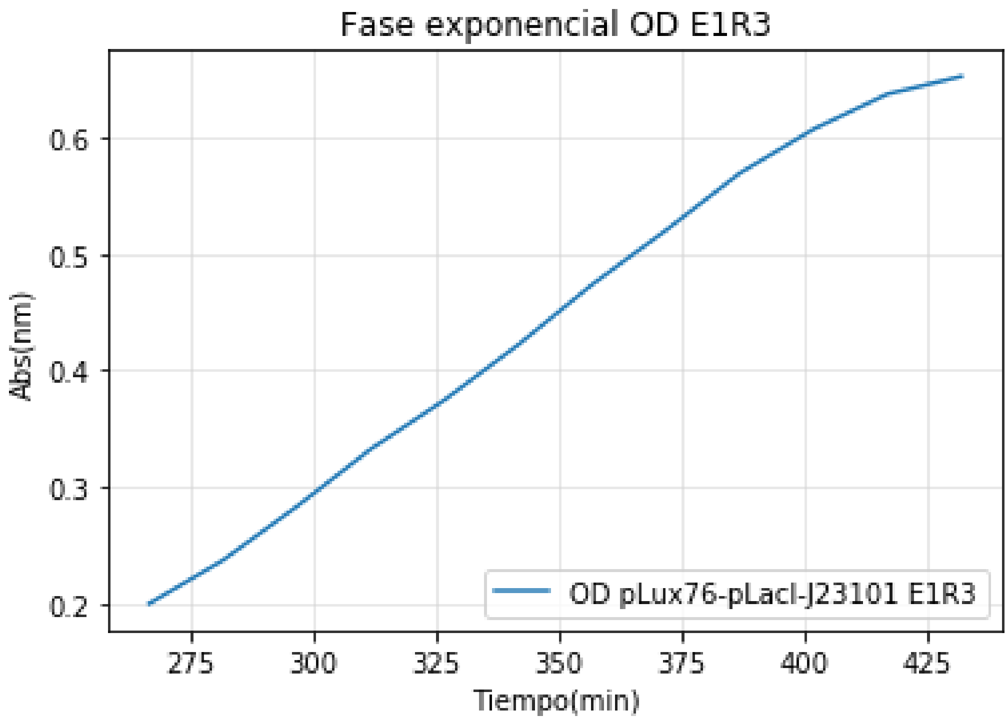



```
A=2.309560e+00
um=9.312316e-03
```

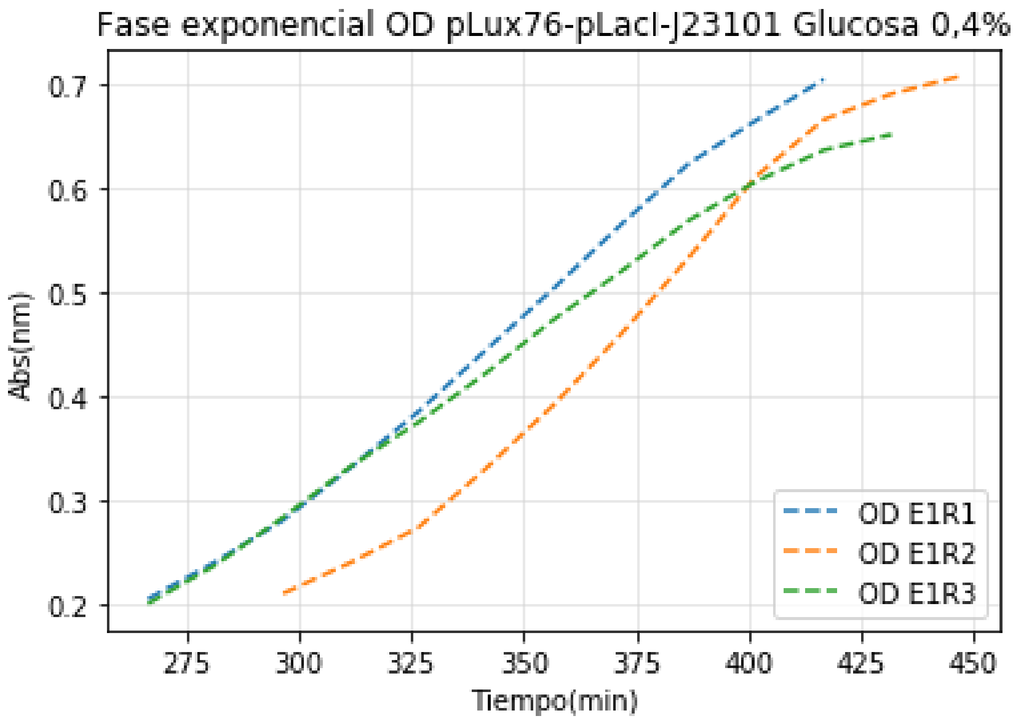
```
l=1.703790e+02
Tm=2.616173e+02
doubpe=7.443338e+01
ext=1.488668e+02
Tfinal=4.104840e+02
```

```
Out[50]: <matplotlib.legend.Legend at 0x1abaaf4f358>
```





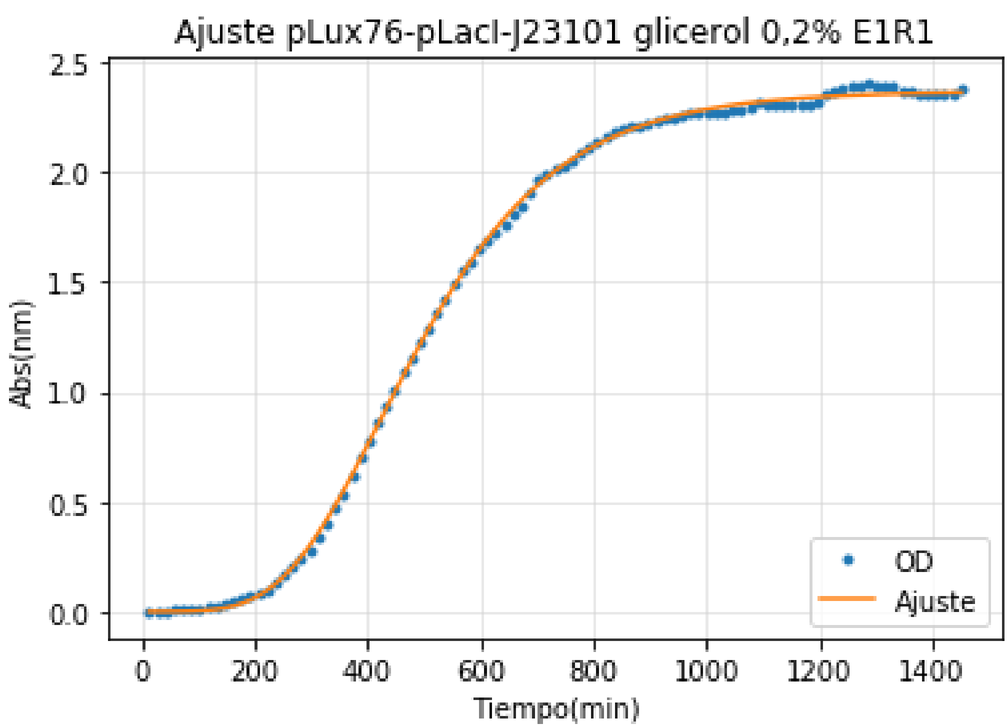

```
In [51]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLacI-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:28],odpls1[17:28],'--',label='OD E1R1')
         plt.plot(tt[19:30],odpls2[19:30],'--',label='OD E1R2')
         plt.plot(tt[17:29],odpls3[17:29],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[51]: <matplotlib.legend.Legend at 0x1abab069940>
```




```
In [52]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #pLux-pLacI-std glicerol rep 1
         y28= np.log(odplsg1)-np.log(np.min(odplsg1))
         print('Min OD = %e'%((np.min(odplsg1))))
         evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glicerol 0,2
         A28= params[0]
         um28=params[1]
         l28=params[2]
         print('A=%e'%(A28))
         print('um=%e'%(um28))
         print('l=%e'%(l28))

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

         #Delimitación fase exponencial en grafico con OD/tiempo
         y1=tt[28]
```

```
y2=tt[52]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odplsg1,label='OD pLux76-pLacI-J23101 E1R1')
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 E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:53],odplsg1[28:53],label='OD pLux76-pLacI-J23101 E1R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

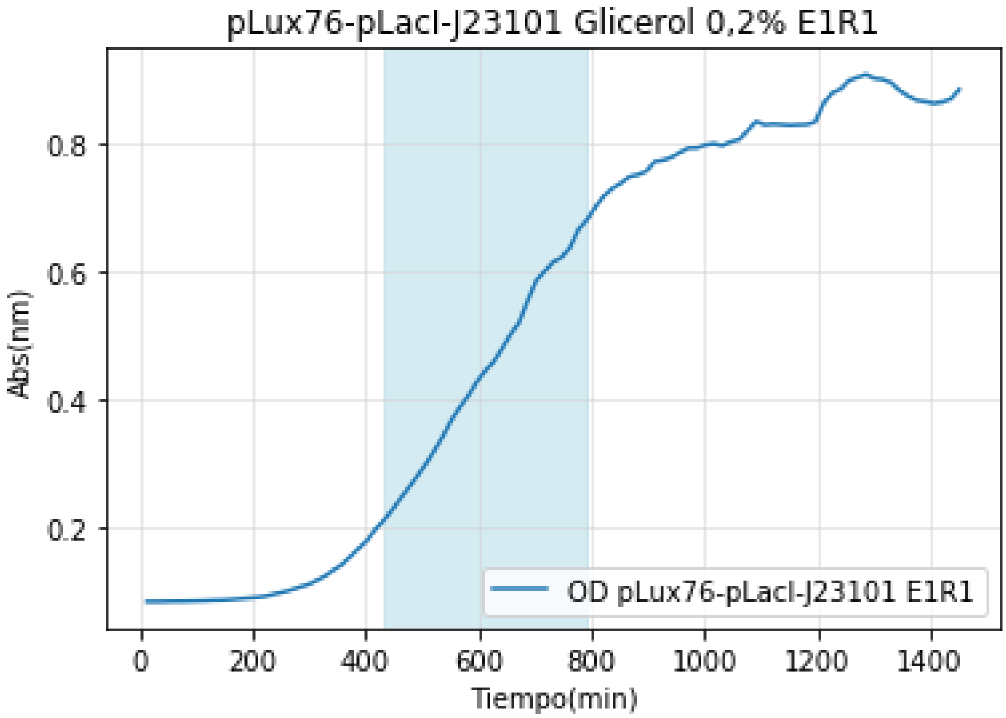
```
Min OD = 8.220000e-02
[  2.36820398e+00   5.03878308e-03   2.48816297e+02]
```

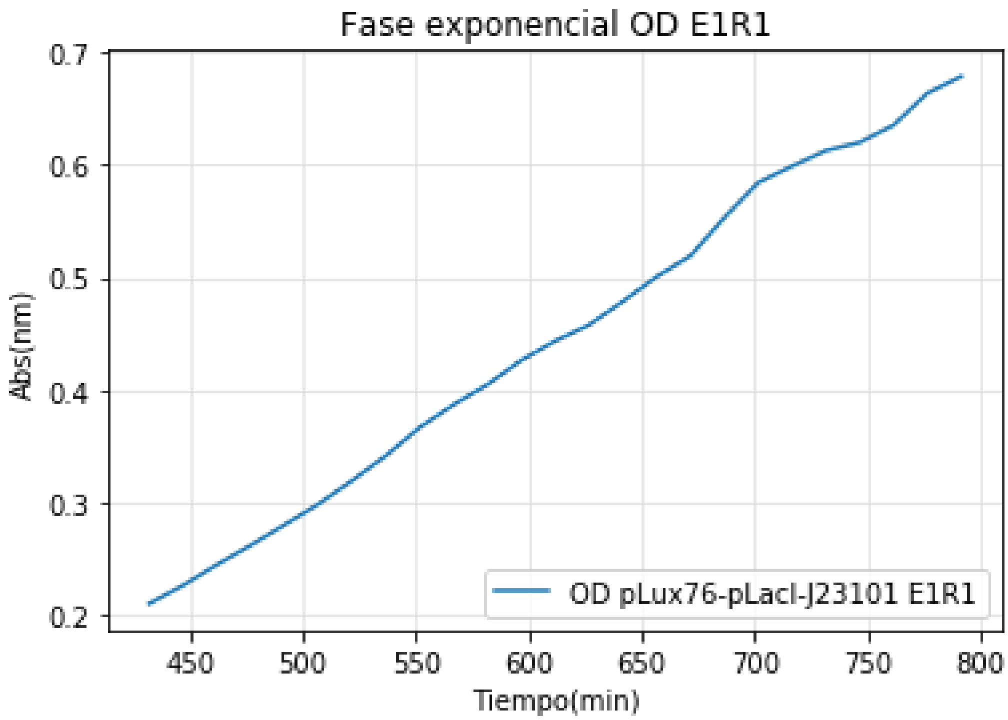



```
A=2.368204e+00
um=5.038783e-03
```

```
l=2.488163e+02
Tm=4.217179e+02
doubpe=1.375624e+02
ext=3.439060e+02
Tfinal=7.656239e+02
```

```
Out[52]: <matplotlib.legend.Legend at 0x1aba9f1aef0>
```

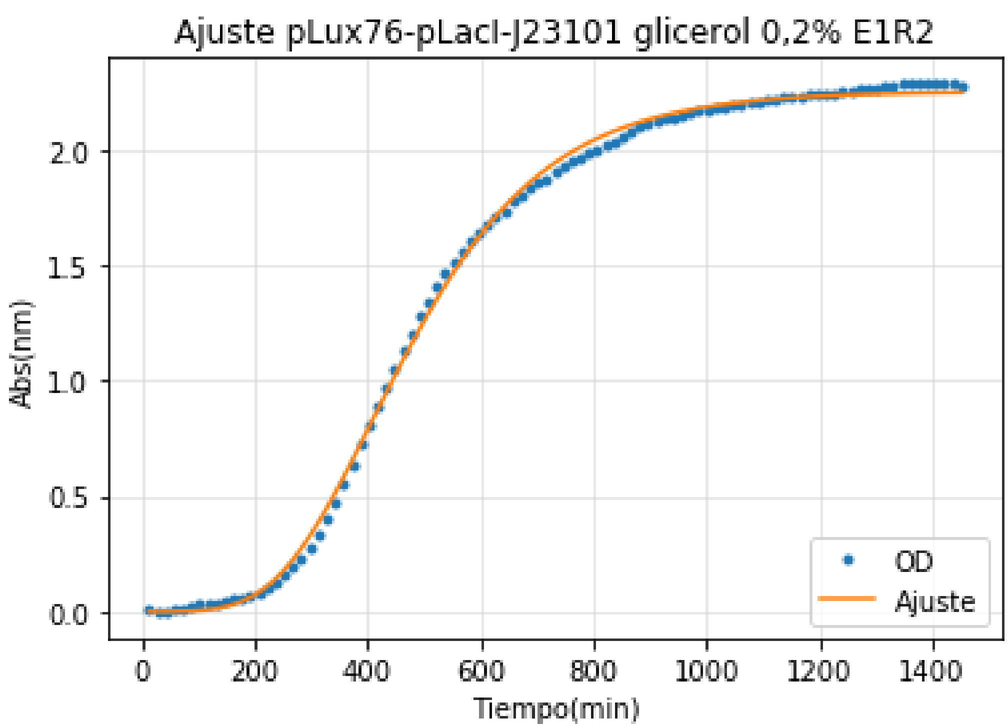




In [53]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pLacI-std glicerol rep 2
y29= np.log(odplsg2)-np.log(np.min(odplsg2))
print('Min OD = %e'%((np.min(odplsg2))))
evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glicerol 0,2
A29= params[0]
um29=params[1]
l29=params[2]
print('A=%e'%(A29))
print('um=%e'%(um29))
print('l=%e'%(l29))

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

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

```
y2=tt[52]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odplsg2,label='OD pLux76-pLacI-J23101 E1R2')
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 E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:53],odplsg2[27:53],label='OD pLux76-pLacI-J23101 E1R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

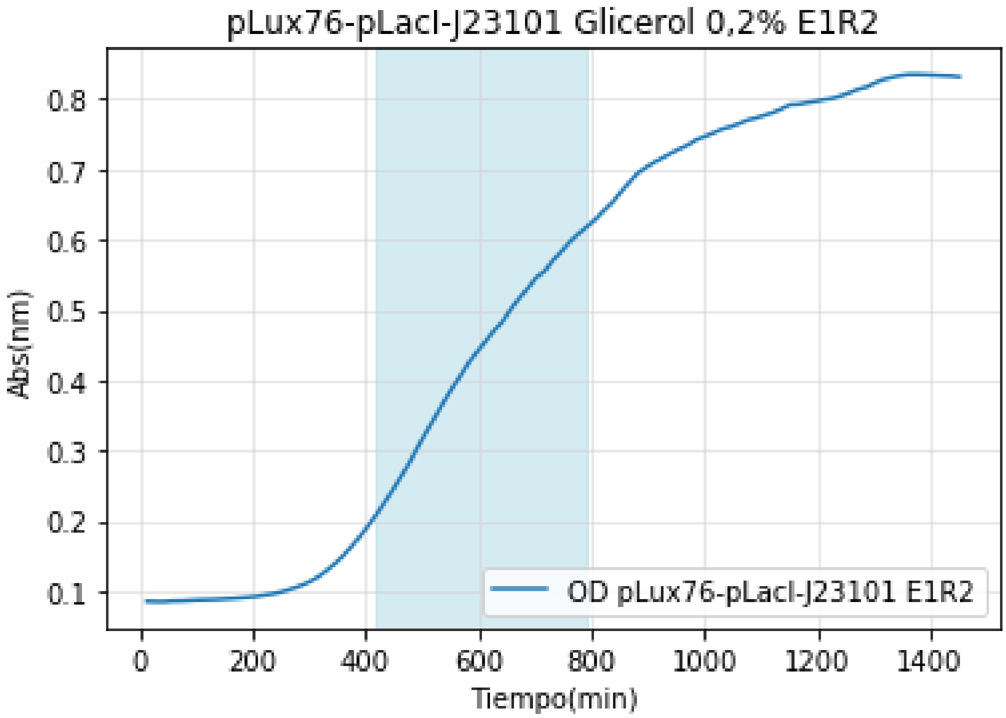
```
Min OD = 8.500000e-02
[  2.25292316e+00   4.90000175e-03   2.38514687e+02]
```

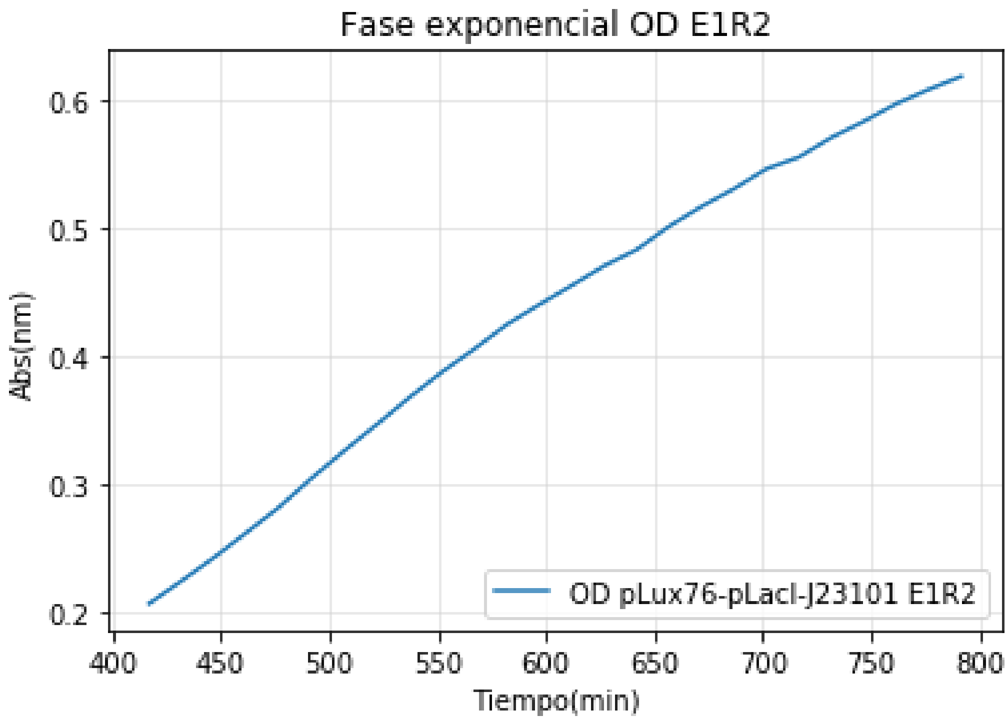



```
A=2.252923e+00
um=4.900002e-03
```

```
l=2.385147e+02
Tm=4.076583e+02
doubpe=1.414586e+02
ext=3.536464e+02
Tfinal=7.613047e+02
```

Out[53]: <matplotlib.legend.Legend at 0x1aba931acc0>

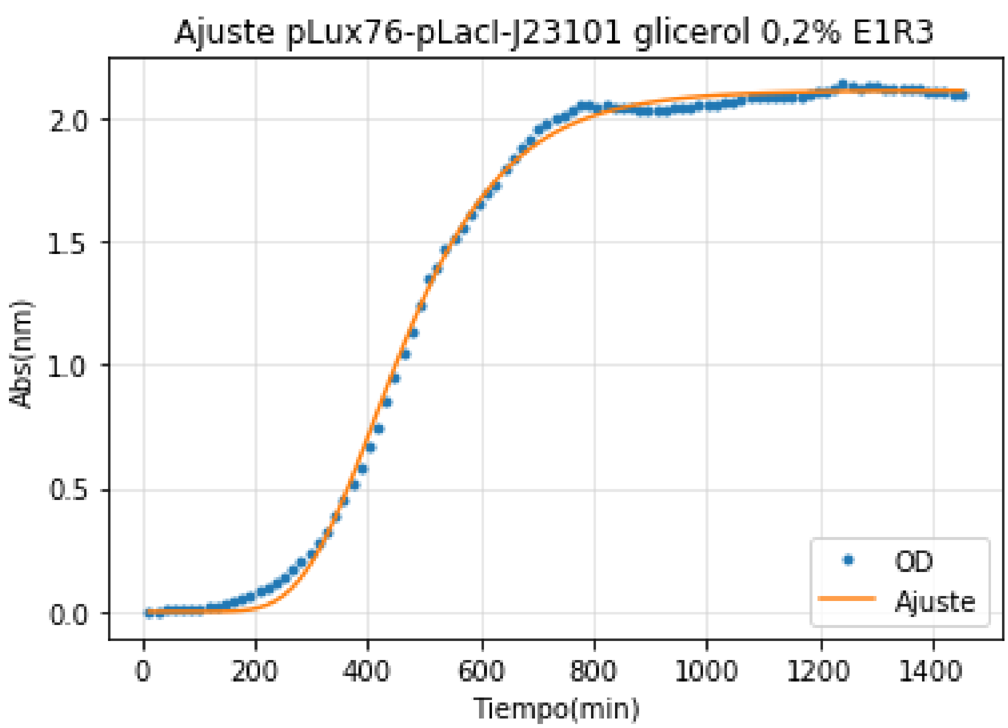




In [54]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#pLux-pLacI-std glicerol rep 3
y30= np.log(odplsg3)-np.log(np.min(odplsg3))
print('Min OD = %e'%((np.min(odplsg3))))
evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste pLux76-pLacI-J23101 glicerol 0,2
A30= params[0]
um30=params[1]
l30=params[2]
print('A=%e'%(A30))
print('um=%e'%(um30))
print('l=%e'%(l30))

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

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

```python
y2=tt[47]
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odplsg3,label='OD pLux76-pLacI-J23101 E1R3')
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 E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:48],odplsg3[27:48],label='OD pLux76-pLacI-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

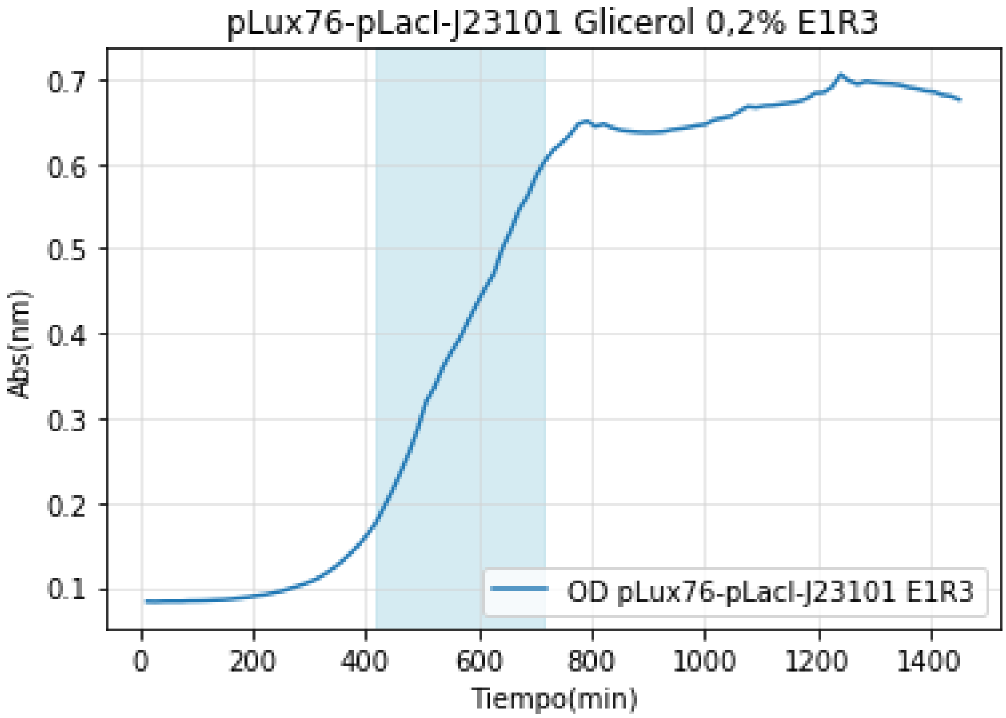
```
Min OD = 8.340000e-02
[  2.10857574e+00   6.00163219e-03   2.81867611e+02]
```

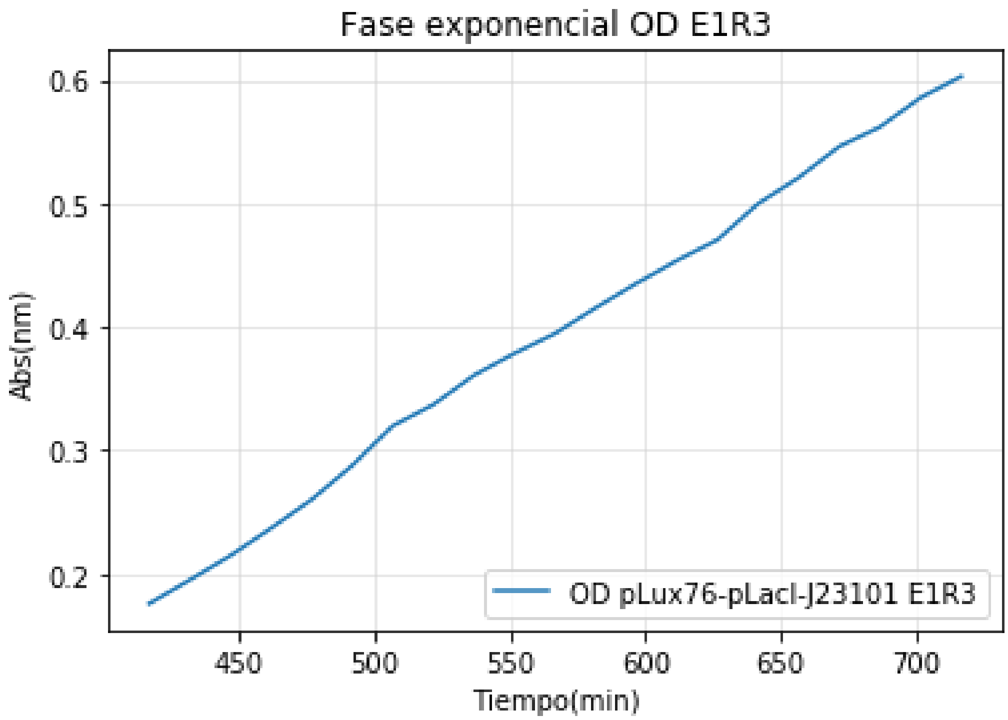



```
A=2.108576e+00
um=6.001632e-03
```

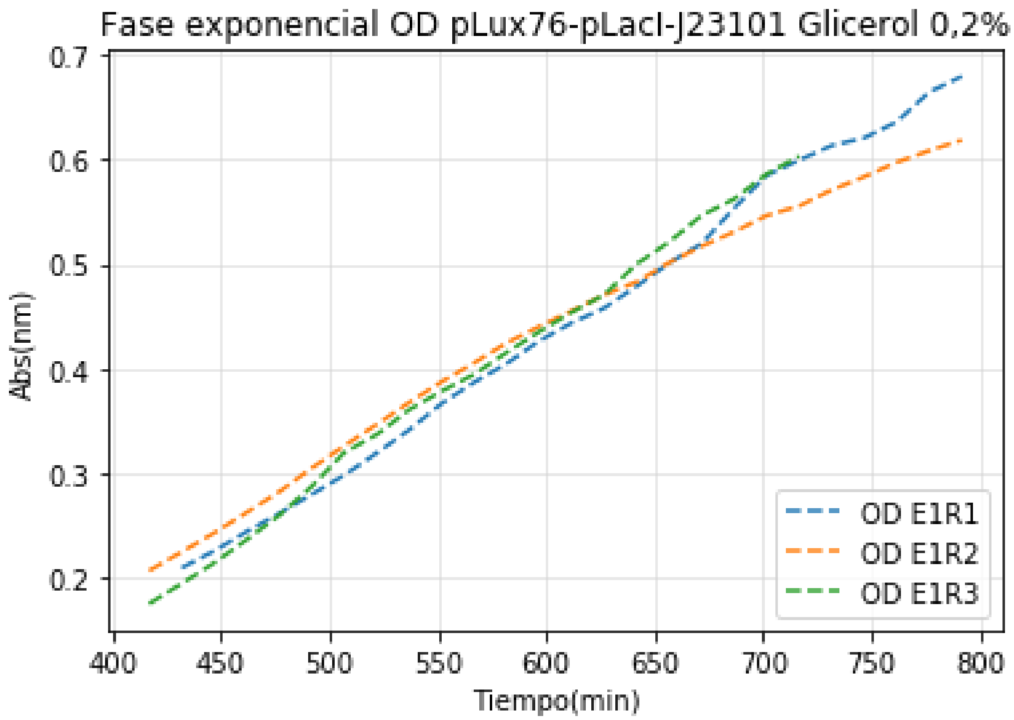
```
l=2.818676e+02
Tm=4.111161e+02
doubpe=1.154931e+02
ext=2.887328e+02
Tfinal=6.998488e+02
```

```
Out[54]: <matplotlib.legend.Legend at 0x1abab10b780>
```



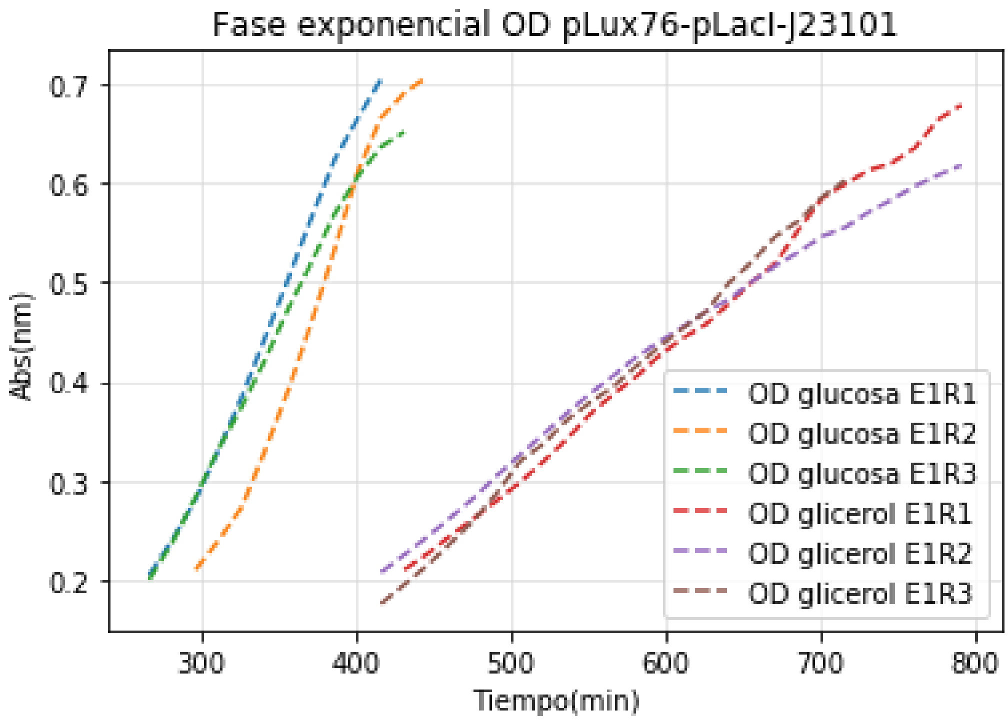



```
In [55]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-pLacI-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[28:53],odplsg1[28:53],'--',label='OD E1R1')
         plt.plot(tt[27:53],odplsg2[27:53],'--',label='OD E1R2')
         plt.plot(tt[27:48],odplsg3[27:48],'--',label='OD E1R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[55]: <matplotlib.legend.Legend at 0x1abab273c18>
```



In [56]:
```
#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[17:28],odpls1[17:28],'--',label='OD glucosa E1R1')
plt.plot(tt[19:30],odpls2[19:30],'--',label='OD glucosa E1R2')
plt.plot(tt[17:29],odpls3[17:29],'--',label='OD glucosa E1R3')
plt.plot(tt[28:53],odplsg1[28:53],'--',label='OD glicerol E1R1')
plt.plot(tt[27:53],odplsg2[27:53],'--',label='OD glicerol E1R2')
plt.plot(tt[27:48],odplsg3[27:48],'--',label='OD glicerol E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[56]:
```
<matplotlib.legend.Legend at 0x1abab33bef0>
```



In [57]:
```
#Selección de datos en arrays, según lo determinado
#controles glucosa
o1=odcg1[16:27]
c1=cfpcg1[16:27]
r1=rfpcg1[16:27]
y1=yfpcg1[16:27]

o2=odcg2[16:26]
c2=cfpcg2[16:26]
r2=rfpcg2[16:26]
y2=yfpcg2[16:26]

o3=odcg3[16:26]
c3=cfpcg3[16:26]
r3=rfpcg3[16:26]
y3=yfpcg3[16:26]

#controles glicerol
o4=odcgl1[26:50]
c4=cfpcgl1[26:50]
r4=rfpcgl1[26:50]
y4=yfpcgl1[26:50]

o5=odcgl2[22:46]
```

```
c5=cfpcgl2[22:46]
r5=rfpcgl2[22:46]
y5=yfpcgl2[22:46]

o6=odcgl3[23:45]
c6=cfpcgl3[23:45]
r6=rfpcgl3[23:45]
y6=yfpcgl3[23:45]

#std-std-std glucosa
o7=odsss1[18:29]
c7=cfpsss1[18:29]
r7=rfpsss1[18:29]
y7=yfpsss1[18:29]

o8=odsss2[22:34]
c8=cfpsss2[22:34]
r8=rfpsss2[22:34]
y8=yfpsss2[22:34]

o9=odsss3[18:27]
c9=cfpsss3[18:27]
r9=rfpsss3[18:27]
y9=yfpsss3[18:27]

#std-std-std glicerol
o10=odsssg1[31:56]
c10=cfpsssg1[31:56]
r10=rfpsssg1[31:56]
y10=yfpsssg1[31:56]

o11=odsssg2[32:60]
c11=cfpsssg2[32:60]
r11=rfpsssg2[32:60]
y11=yfpsssg2[32:60]

o12=odsssg3[30:51]
c12=cfpsssg3[30:51]
r12=rfpsssg3[30:51]
y12=yfpsssg3[30:51]

#plux-std-std glucosa
o13=odpss1[18:29]
c13=cfppss1[18:29]
r13=rfppss1[18:29]
y13=yfppss1[18:29]

o14=odpss2[19:30]
```

```
c14=cfppss2[19:30]
r14=rfppss2[19:30]
y14=yfppss2[19:30]

o15=odpss3[17:29]
c15=cfppss3[17:29]
r15=rfppss3[17:29]
y15=yfppss3[17:29]

#plux-std-std glicerol
o16=odpssg1[31:55]
c16=cfppssg1[31:55]
r16=rfppssg1[31:55]
y16=yfppssg1[31:55]

o17=odpssg2[27:52]
c17=cfppssg2[27:52]
r17=rfppssg2[27:52]
y17=yfppssg2[27:52]

o18=odpssg3[28:50]
c18=cfppssg3[28:50]
r18=rfppssg3[28:50]
y18=yfppssg3[28:50]

#plux-pTet-std glucosa
o19=odpts1[19:30]
c19=cfppts1[19:30]
r19=rfppts1[19:30]
y19=yfppts1[19:30]

o20=odpts2[24:36]
c20=cfppts2[24:36]
r20=rfppts2[24:36]
y20=yfppts2[24:36]

o21=odpts3[20:32]
c21=cfppts3[20:32]
r21=rfppts3[20:32]
y21=yfppts3[20:32]

#plux-pTet-std glicerol
o22=odptsg1[32:60]
c22=cfpptsg1[32:60]
r22=rfpptsg1[32:60]
y22=yfpptsg1[32:60]

o23=odptsg2[32:60]
```

```
c23=cfpptsg2[32:60]
r23=rfpptsg2[32:60]
y23=yfpptsg2[32:60]

o24=odptsg3[28:53]
c24=cfpptsg3[28:53]
r24=rfpptsg3[28:53]
y24=yfpptsg3[28:53]

#plux-pLacI-std glucosa
o25=odpls1[17:28]
c25=cfppls1[17:28]
r25=rfppls1[17:28]
y25=yfppls1[17:28]

o26=odpls2[19:30]
c26=cfppls2[19:30]
r26=rfppls2[19:30]
y26=yfppls2[19:30]

o27=odpls3[17:29]
c27=cfppls3[17:29]
r27=rfppls3[17:29]
y27=yfppls3[17:29]

#plux-pLacI-std glicerol
o28=odplsg1[28:53]
c28=cfpplsg1[28:53]
r28=rfpplsg1[28:53]
y28=yfpplsg1[28:53]

o29=odplsg2[27:53]
c29=cfpplsg2[27:53]
r29=rfpplsg2[27:53]
y29=yfpplsg2[27:53]

o30=odplsg3[27:48]
c30=cfpplsg3[27:48]
r30=rfpplsg3[27:48]
y30=yfpplsg3[27:48]
```

```
In [58]: #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

#std-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

#std-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-std-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
```

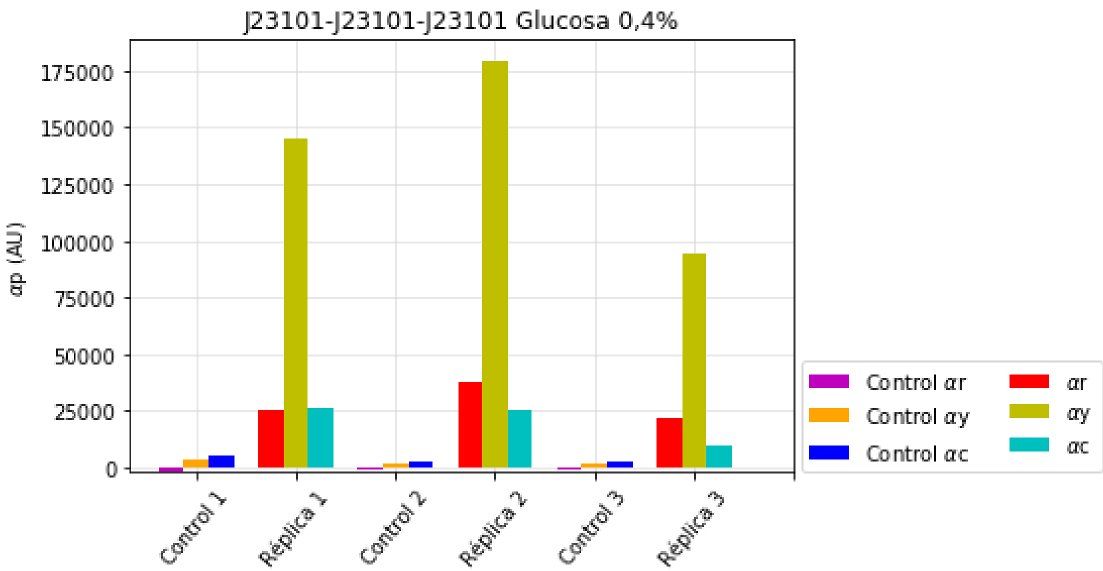
```python
slope, intercept, r_value, p_value,std_err=stats.linregress(o15,c15)
slopec15=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o15,r15)
sloper15=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o15,y15)
slopey15=slope

#plux-std-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-pTet-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-pTet-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-pLacI-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-pLacI-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
```

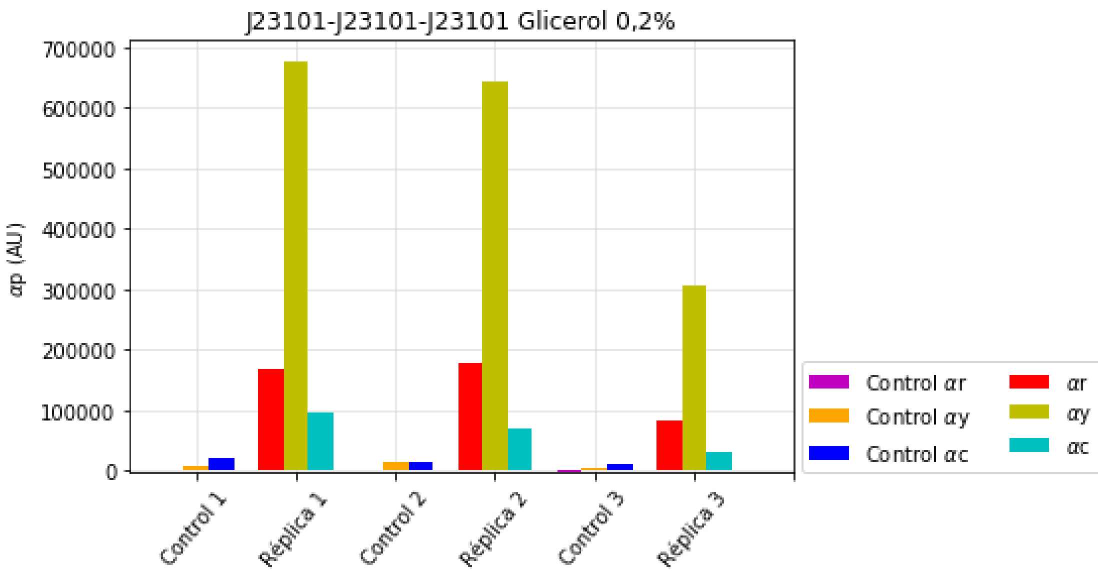
```
In [59]: pendientesc=[slopec1,slopec2,slopec3,slopec4,slopec5,slopec6,slopec7,slopec8,slopec9,sl
         pendientesr=[sloper1,sloper2,sloper3,sloper4,sloper5,sloper6,sloper7,sloper8,sloper9,sl
         pendientesy=[slopey1,slopey2,slopey3,slopey4,slopey5,slopey6,slopey7,slopey8,slopey9,sl

         #Grafico pendientes std-std-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('J23101-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)
plt.savefig(filename='alfap stdx3 glucosa.png', dpi=300, facecolor='w', edgecolor='w',b
```



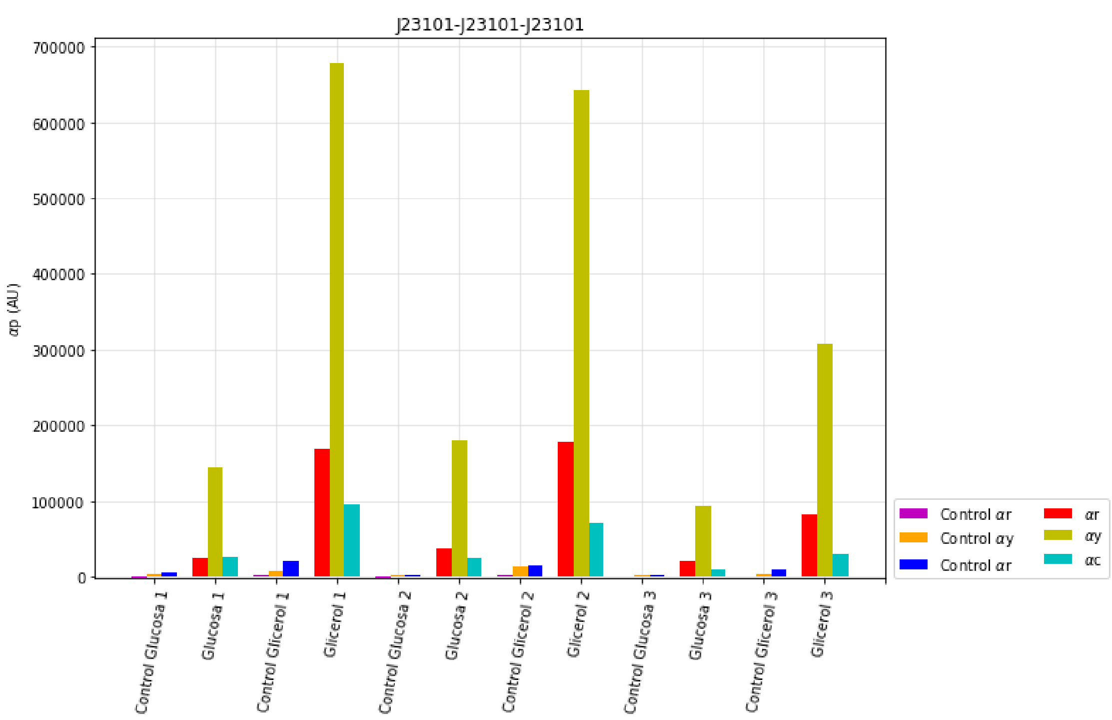

```
In [60]: #Grafico pendientes std-std-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('J23101-J23101-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         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)
plt.savefig(filename='alfap stdx3 glicerol.png', dpi=300, facecolor='w', edgecolor='w',
```

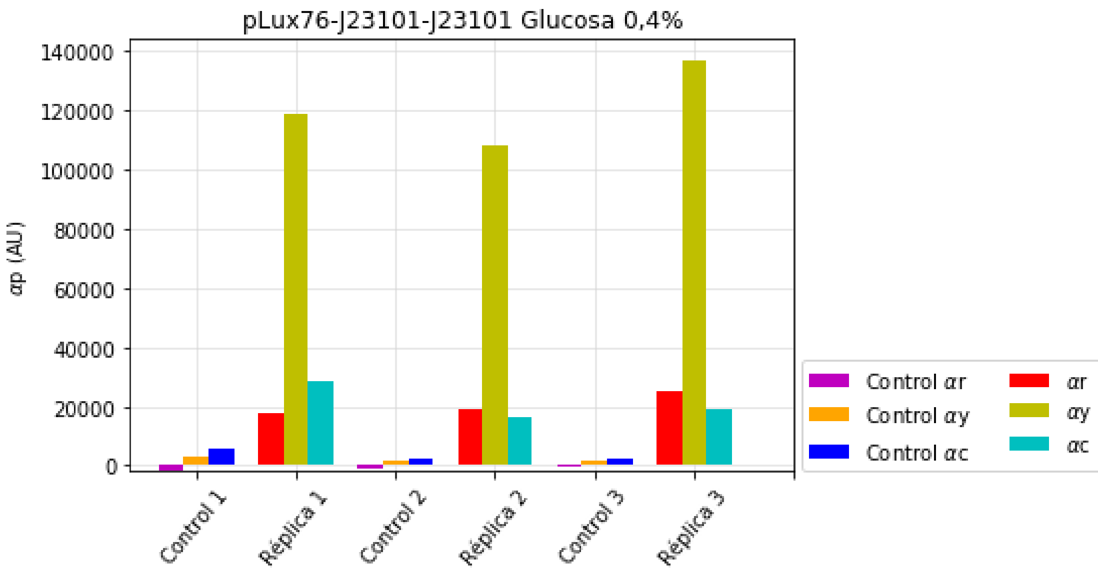



```
In [61]: #Grafico pendientes std-std-std
         X = np.arange(13)
         plt.figure(figsize=(10,7))
         plt.title('J23101-J23101-J23101')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ 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)
```

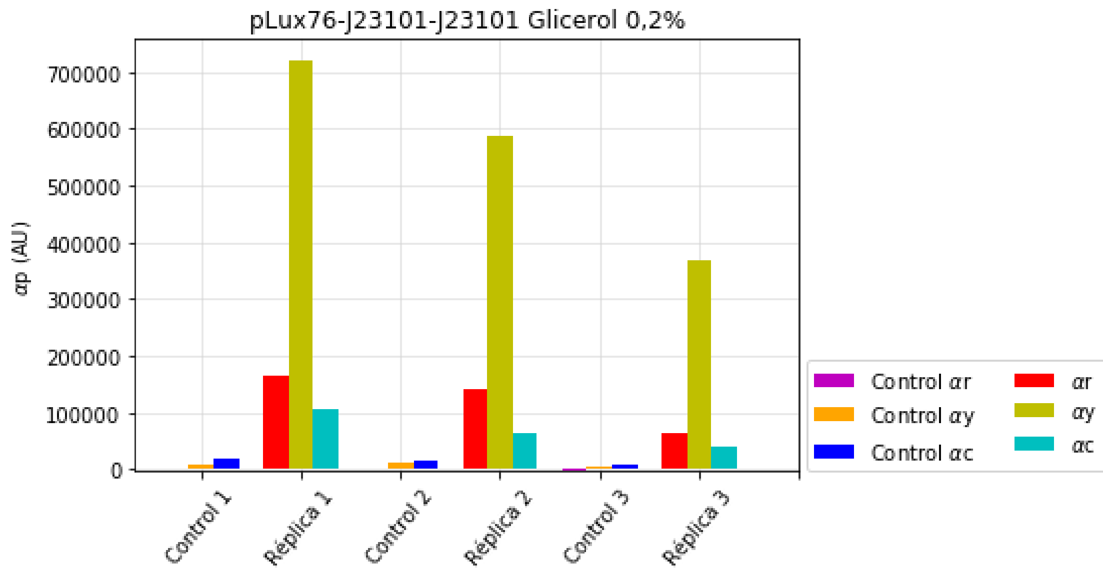
```
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
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)
plt.savefig(filename='alfap stdx3 ambos.png', dpi=300, facecolor='w', edgecolor='w',bbo
```



```
In [62]: #print('ar',pendientesr)
         #print('ay',pendientesy)
         #print('ac',pendientesc)

         #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$c'
         plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\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)
         plt.savefig(filename='alfap plux-std-std glucosa.png', dpi=300, facecolor='w', edgecolo
```

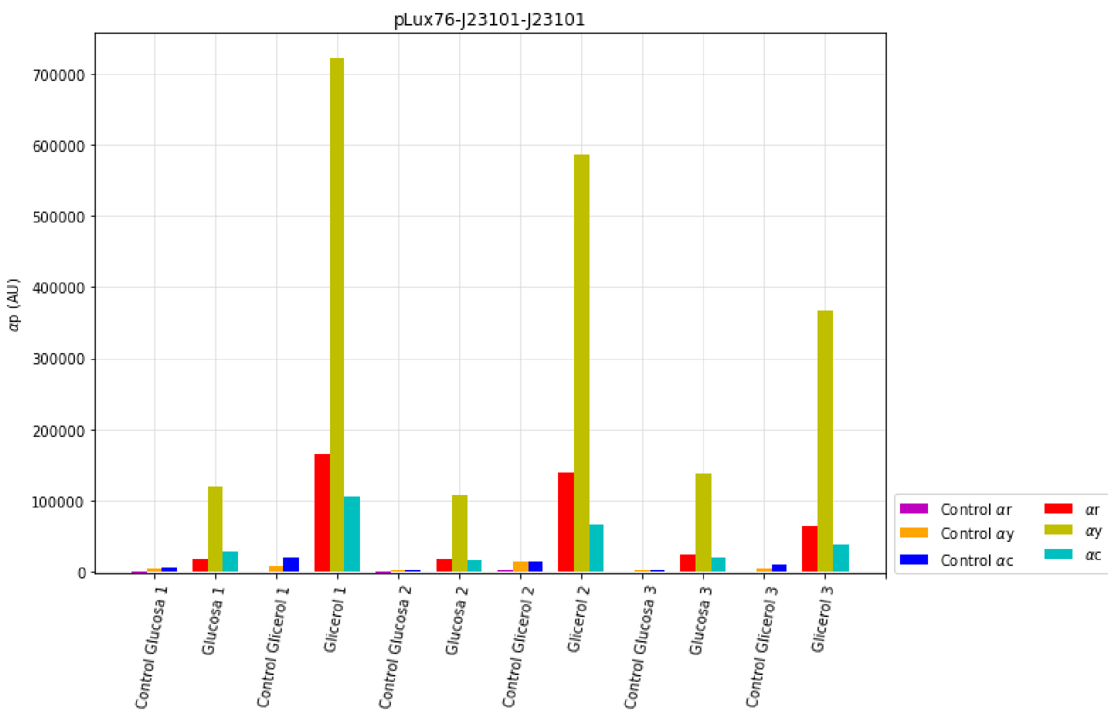



```
In [63]: #Grafico pendientes plux-std-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-J23101-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         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$c'
         plt.bar(X[1]-0.25,pendientesr[15],color='r',width=0.25,label=r'$\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)
         plt.savefig(filename='alfap plux-std-std glicerol.png', dpi=300, facecolor='w', edgecol
```



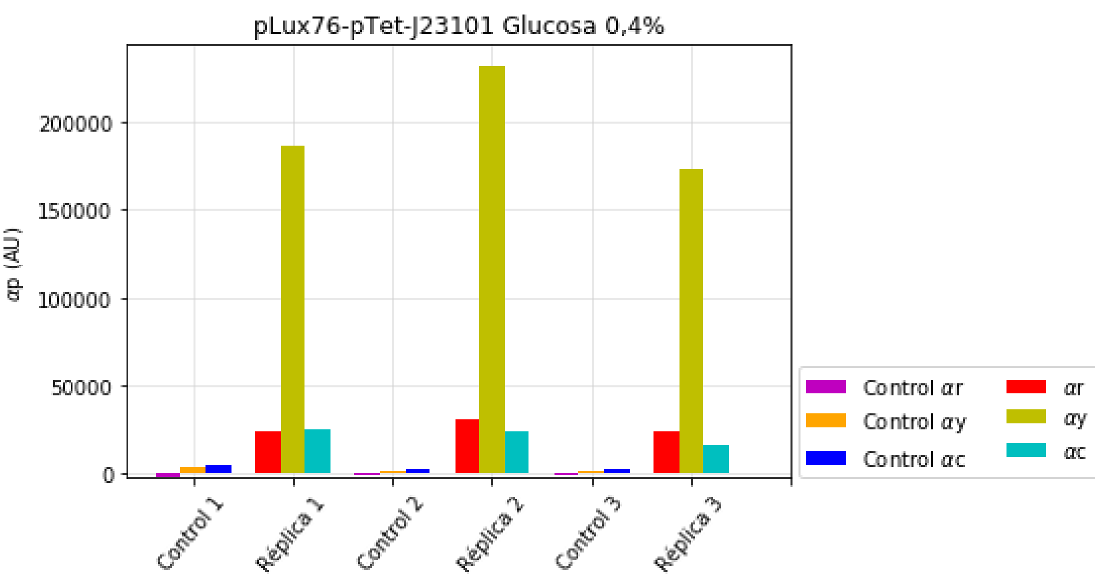

```
In [64]: #Grafico pendientes plux-std-std
         X = np.arange(13)
         plt.figure(figsize=(10,7))
         plt.title('pLux76-J23101-J23101')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\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)
plt.savefig(filename='alfap plux-std-std ambos.png', dpi=300, facecolor='w', edgecolor=
```



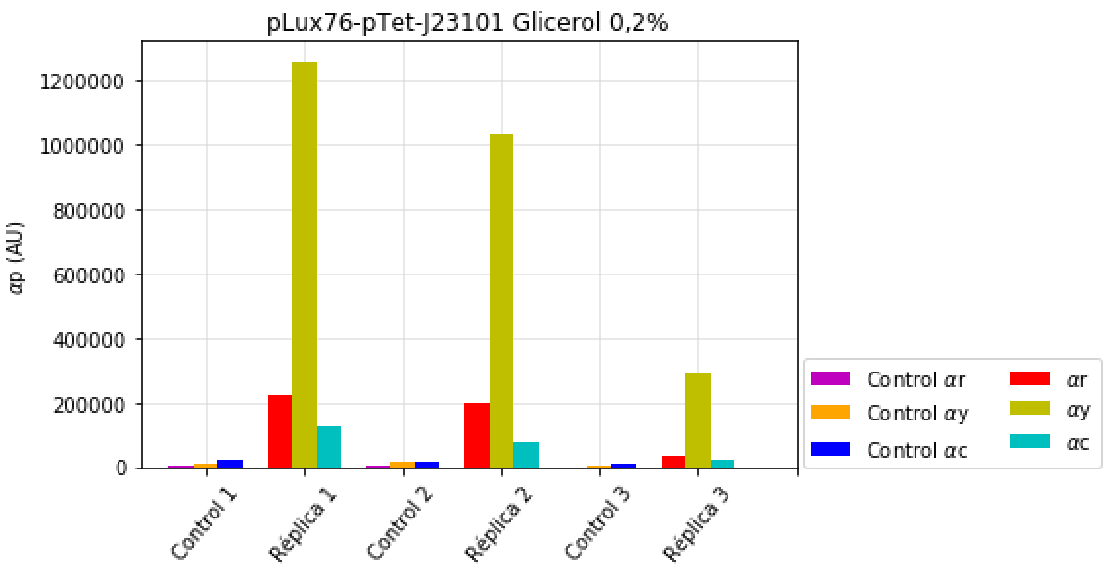
In [65]:
```
#Grafico pendientes plux-pTet-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pLux76-pTet-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
```

```
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)
plt.savefig(filename='alfap plux-ptet-std glucosa.png', dpi=300, facecolor='w', edgecol
```




```
In [66]: #Grafico pendientes plux-ptet-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-pTet-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'
```

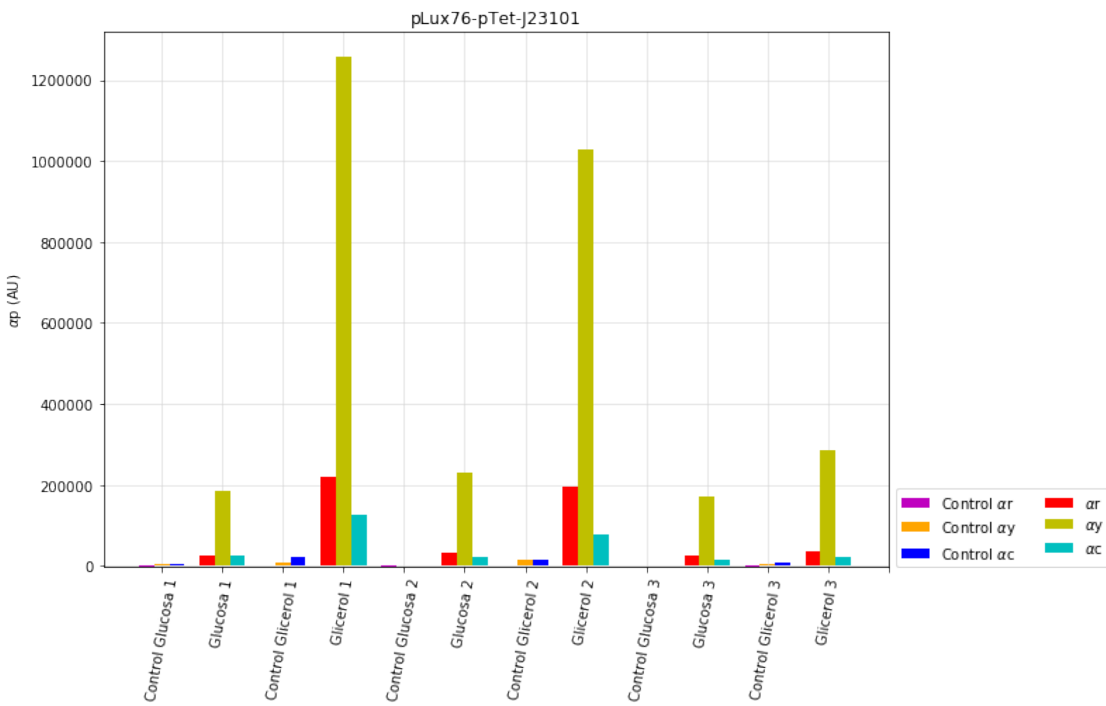
```
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[21],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[21],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[21],color='c',width=0.25,label=r'$\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)
plt.savefig(filename='alfap plux-ptet-std glicerol.png', dpi=300, facecolor='w', edgeco
```



In [67]:

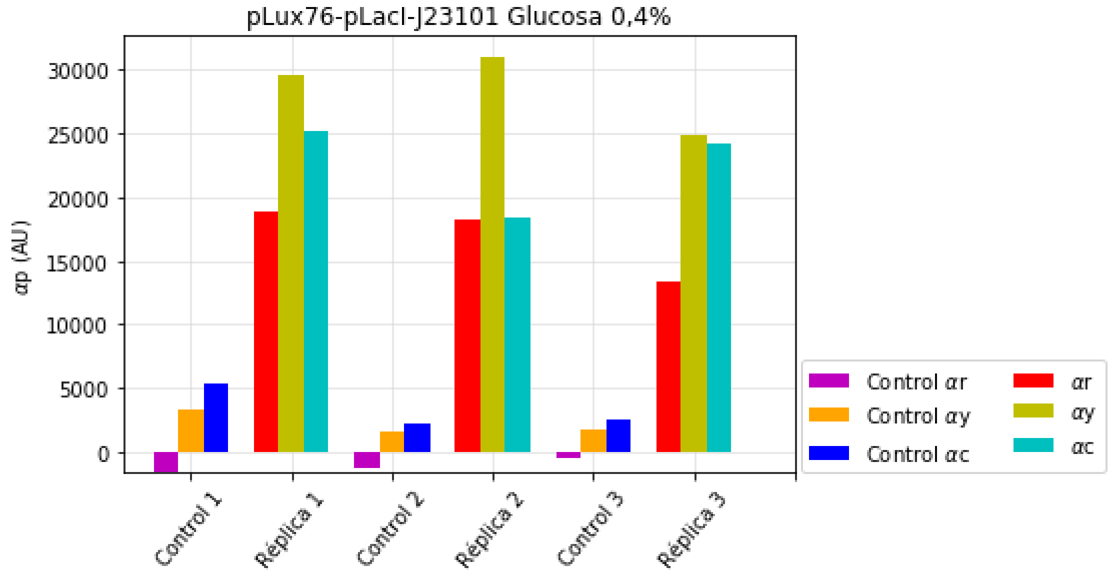
```
#Grafico pendientes plux-ptet-std
X = np.arange(13)
plt.figure(figsize=(10,7))
plt.title('pLux76-pTet-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
```

```
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)
plt.savefig(filename='alfap plux-ptet-std ambos.png', dpi=300, facecolor='w', edgecolor
```



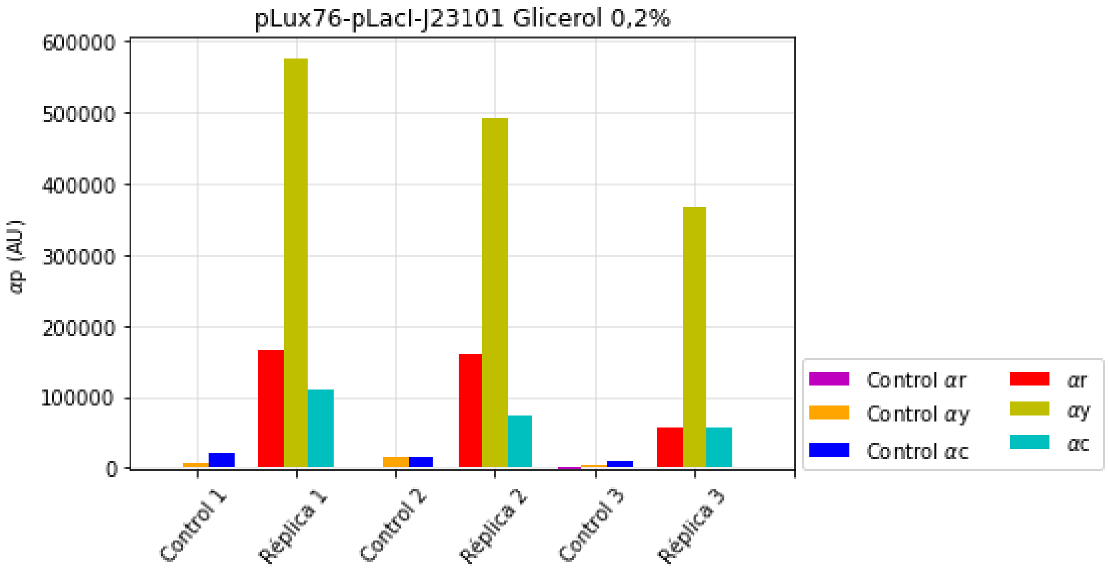

```
In [68]: #Grafico pendientes plux-pLacI-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('pLux76-pLacI-J23101 Glucosa 0,4%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         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$c'
         plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\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)
plt.savefig(filename='alfap plux-plac-std glucosa.png', dpi=300, facecolor='w', edgecol
```



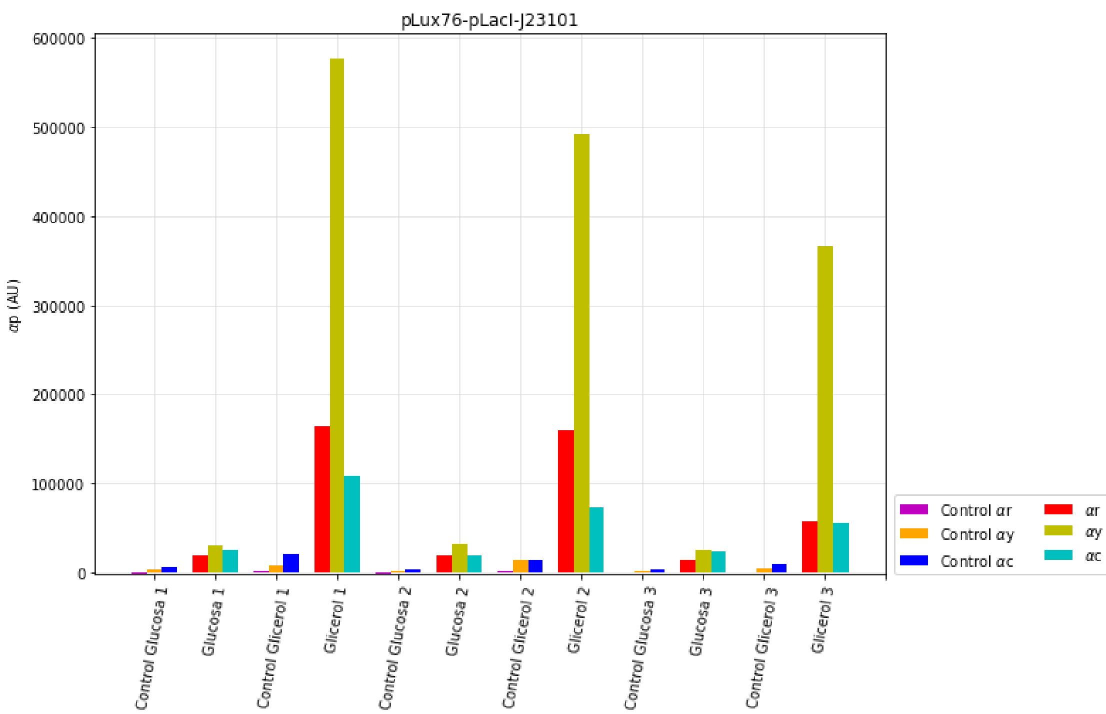

```
In [69]: #Grafico pendientes plux-plac-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pLux76-pLacI-J23101 Glicerol 0,2%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[27],color='r',width=0.25,label=r'$\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)
plt.savefig(filename='alfap plux-plac-std glicerol.png', dpi=300, facecolor='w', edgeco
```




```
In [70]: #Grafico pendientes plux-plac-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$c'
         plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\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)
plt.savefig(filename='alfap plux-plac-std ambos.png', dpi=300, facecolor='w', edgecolor
```

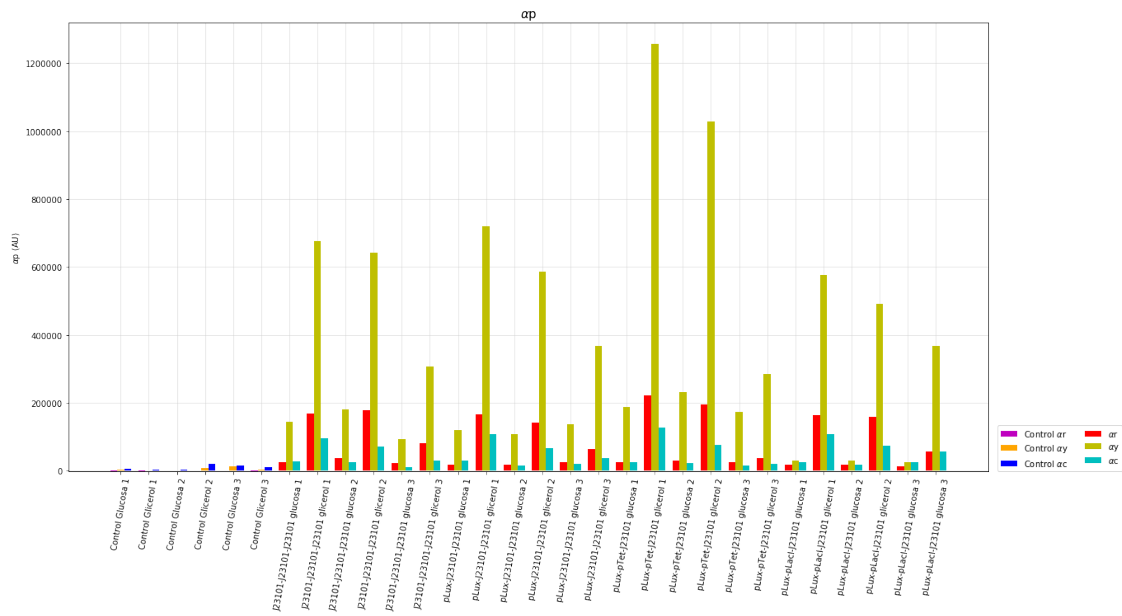



```
In [71]: #Grafico pendientes todo
         X = np.arange(30)
```

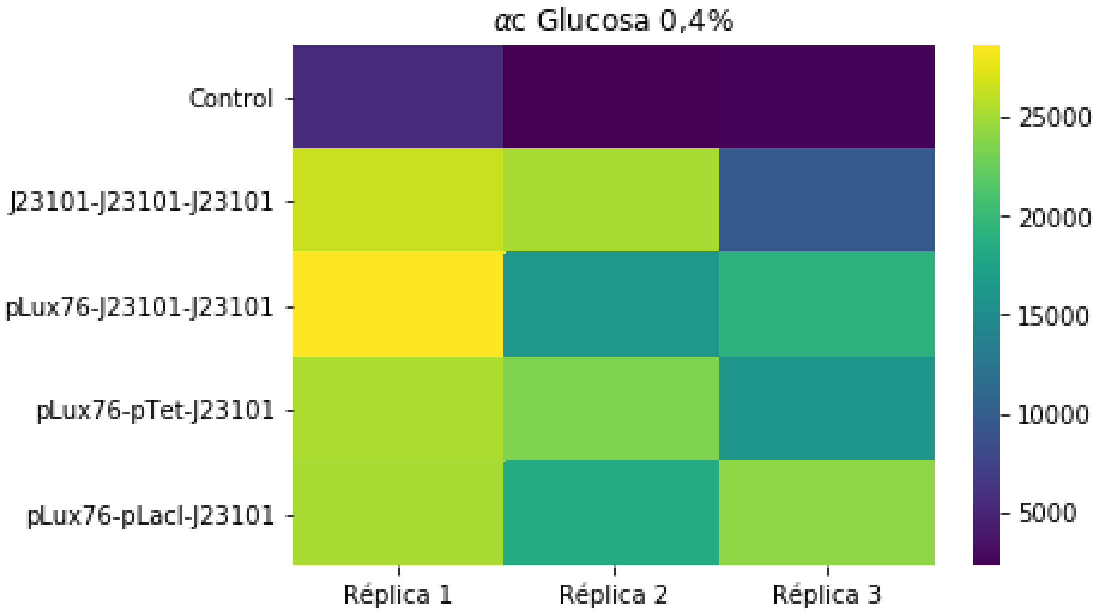
```
plt.figure(figsize=(20,10))
plt.title(r'$\alpha$p',fontsize=15.0)
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[1]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[1]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[2]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\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)
```

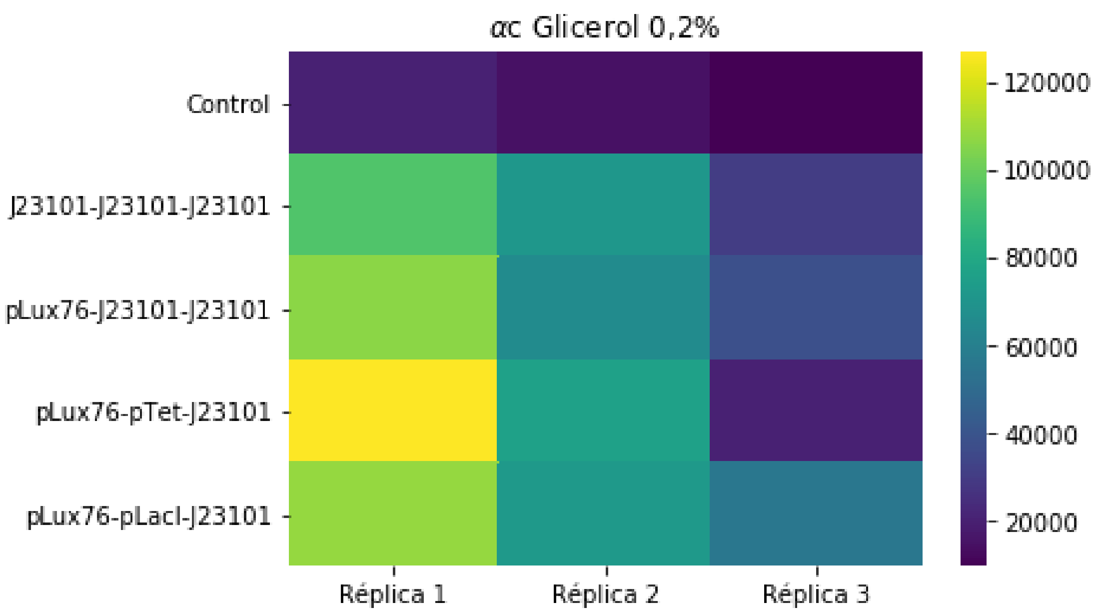
```
plt.bar(X[15]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[15]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[16]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
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 Gl
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
plt.savefig(filename='alfa-p todo.png', dpi=300, facecolor='w', edgecolor='w',bbox_inch
```



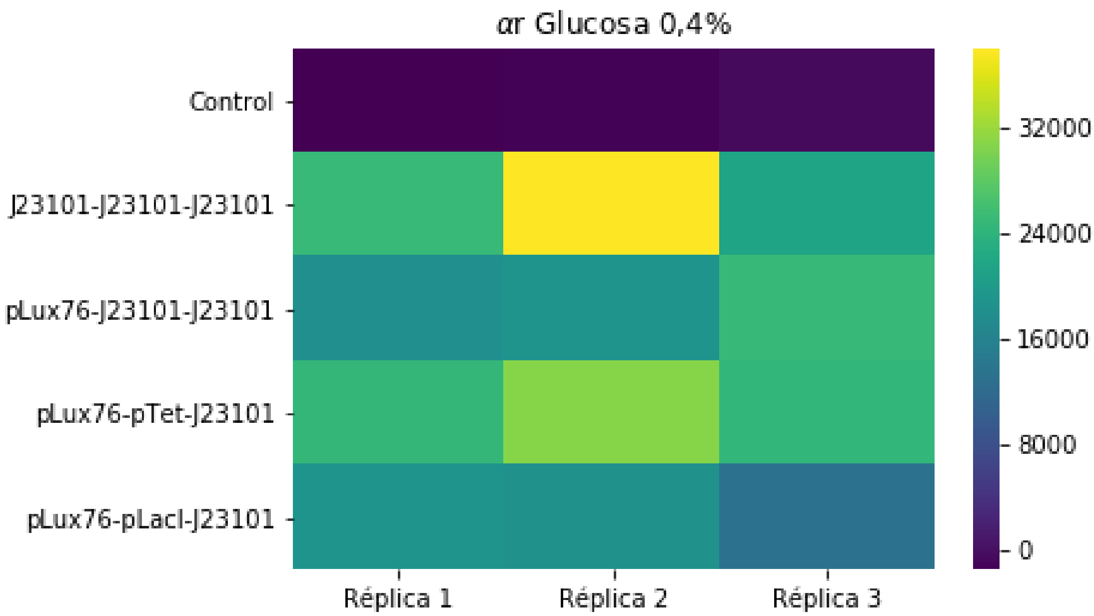

```
In [72]: cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15],
         cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec1
         rglu=[[sloper1,sloper2,sloper3],[sloper7,sloper8,sloper9],[sloper13,sloper14,sloper15],
         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 [73]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

         plt.figure()
         plt.title(r'$\alpha$c Glucosa 0,4%')
         sns.heatmap(cglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='hm alfa-c glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbo

         plt.figure()
         plt.title(r'$\alpha$c Glicerol 0,2%')
         sns.heatmap(cgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='hm alfa-c glicerol.png', dpi=300, facecolor='w', edgecolor='w',bb
```

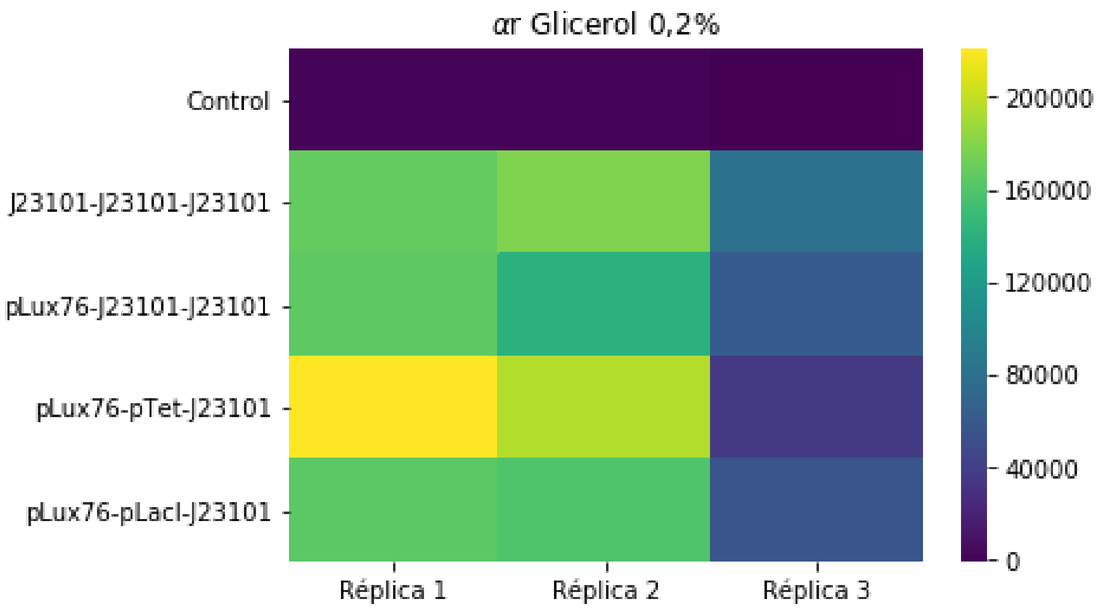




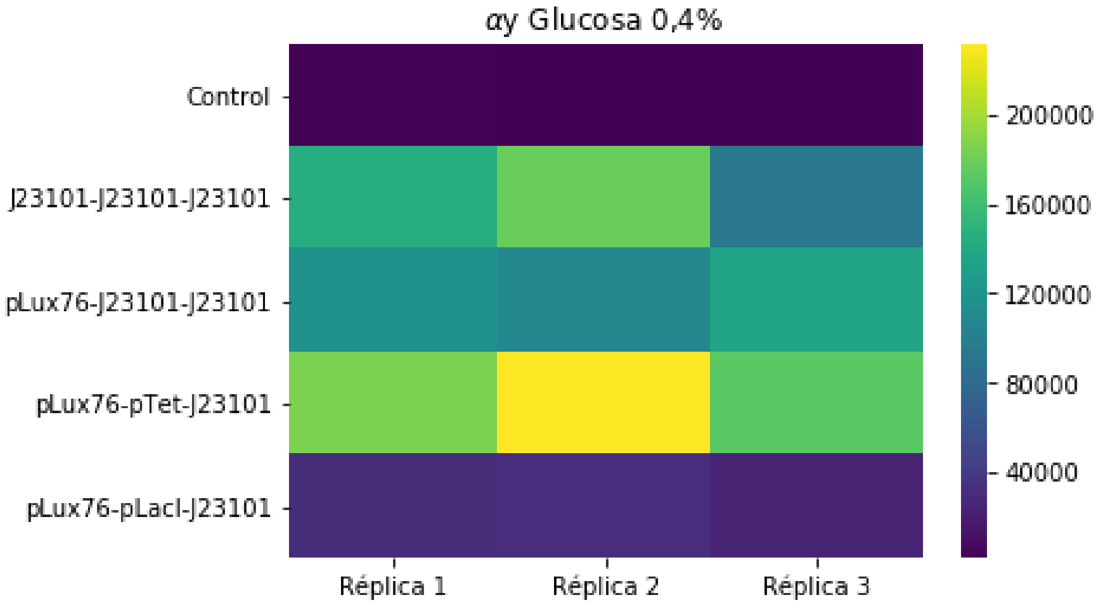
In [74]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

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

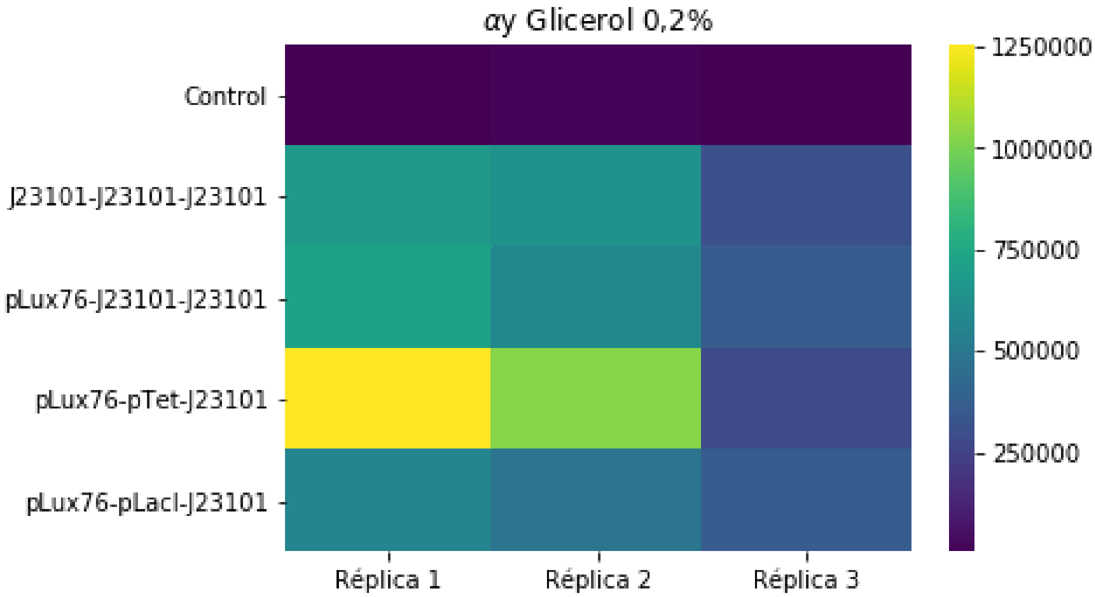
```
sns.heatmap(rglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.savefig(filename='hm alfa-r glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbo


plt.figure()
plt.title(r'$\alpha$r Glicerol 0,2%')
sns.heatmap(rgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.savefig(filename='hm alfa-r glicerol.png', dpi=300, facecolor='w', edgecolor='w',bb
```



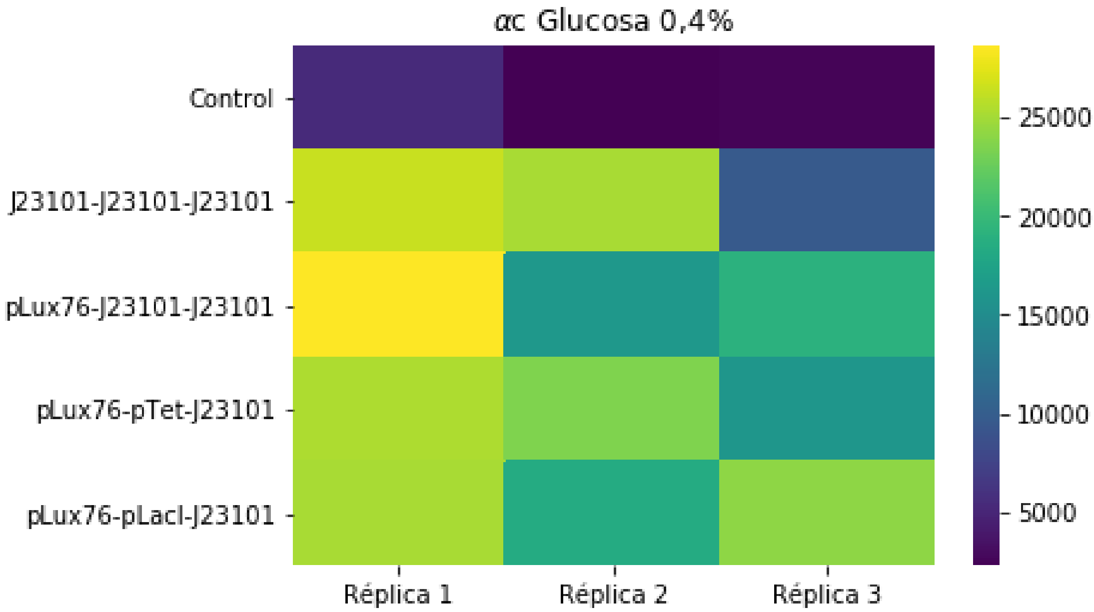


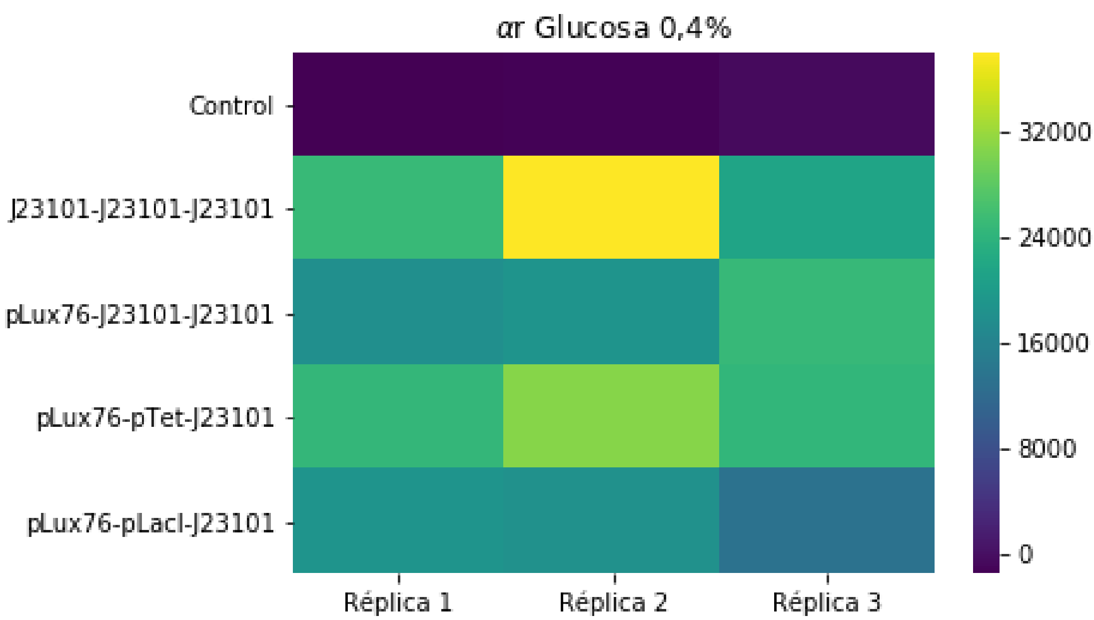
```
In [75]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

         plt.figure()
         plt.title(r'$\alpha$y Glucosa 0,4%')
         sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='hm alfa-y glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbo


         plt.figure()
         plt.title(r'$\alpha$y Glicerol 0,2%')
         sns.heatmap(ygli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='hm alfa-y glicerol.png', dpi=300, facecolor='w', edgecolor='w',bb
```

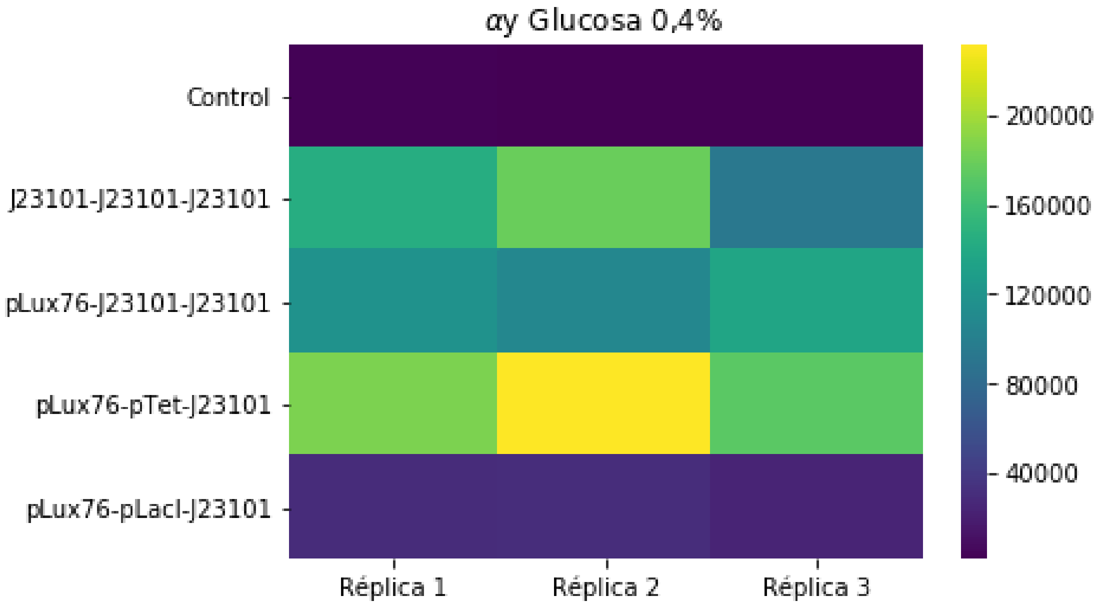




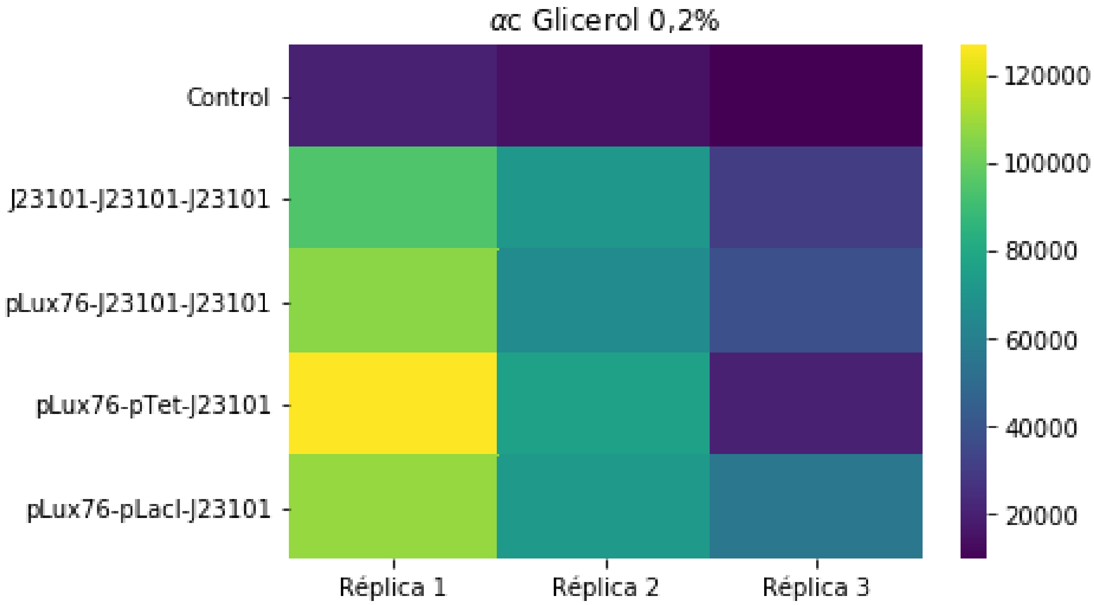

```
In [76]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

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

Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x1abac5bdc50>
```

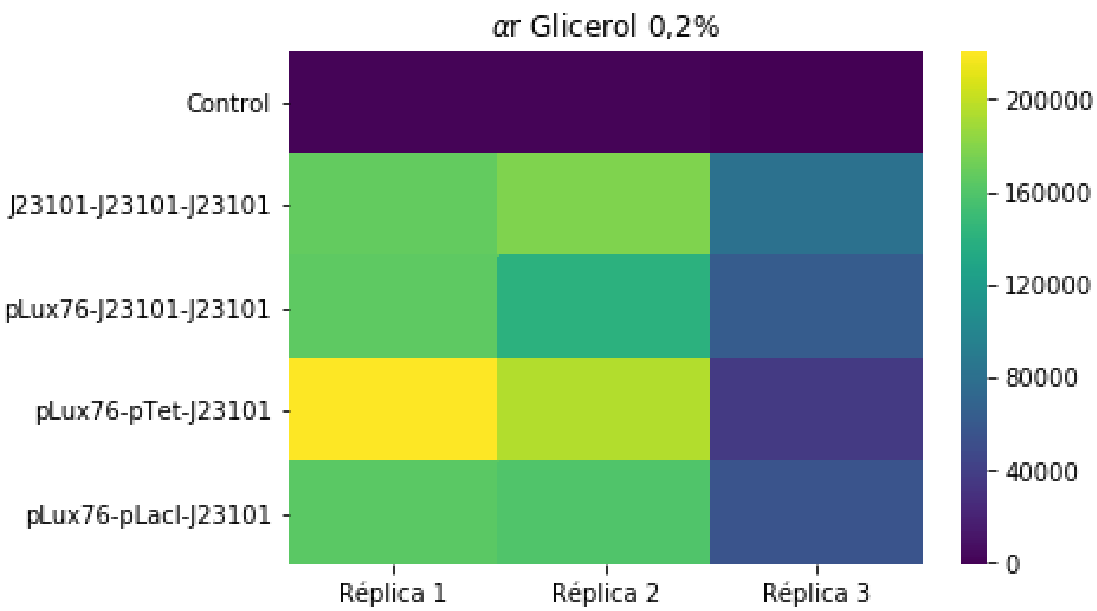

αc Glucosa 0,4%
Control
J23101-J23101-J23101
pLux76-J23101-J23101
pLux76-pTet-J23101
pLux76-pLacI-J23101
Réplica 1
Réplica 2
Réplica 3
25000
20000
15000
10000
5000

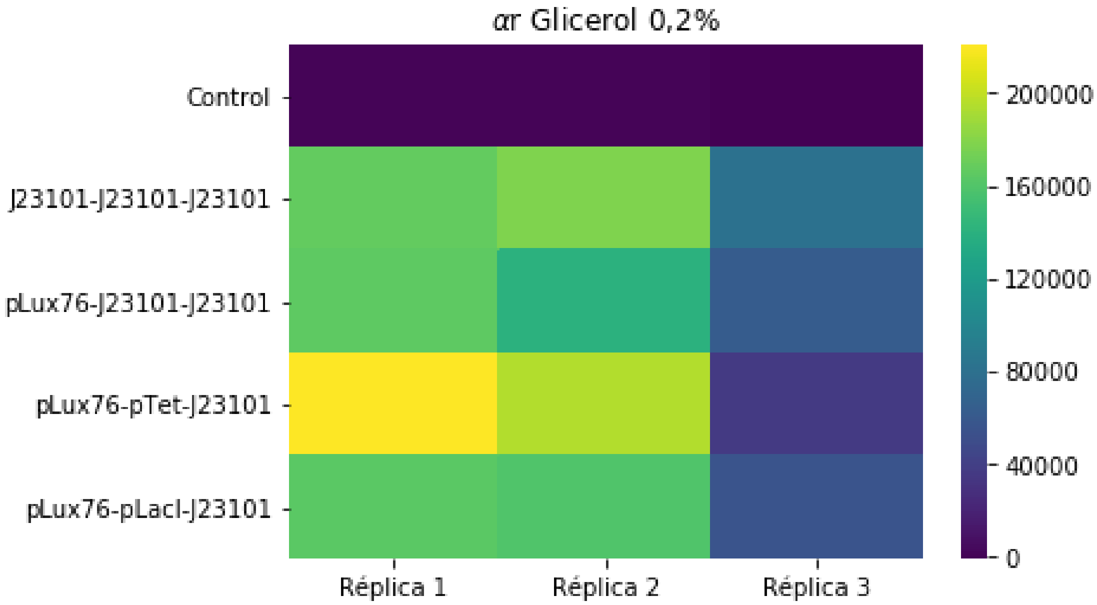

αr Glucosa 0,4%
Control
J23101-J23101-J23101
pLux76-J23101-J23101
pLux76-pTet-J23101
pLux76-pLacI-J23101
Réplica 1
Réplica 2
Réplica 3
32000
24000
16000
8000
0




```
In [77]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

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

Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x1aba9a9c588>
```

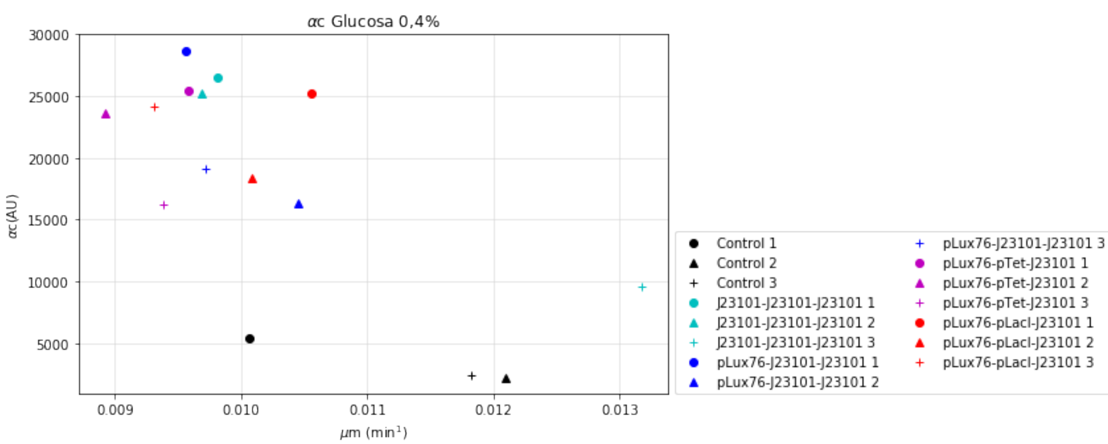

αc Glicerol 0,2%
Control
J23101-J23101-J23101
pLux76-J23101-J23101
pLux76-pTet-J23101
pLux76-pLacI-J23101
Réplica 1
Réplica 2
Réplica 3
120000
100000
80000
60000
40000
20000

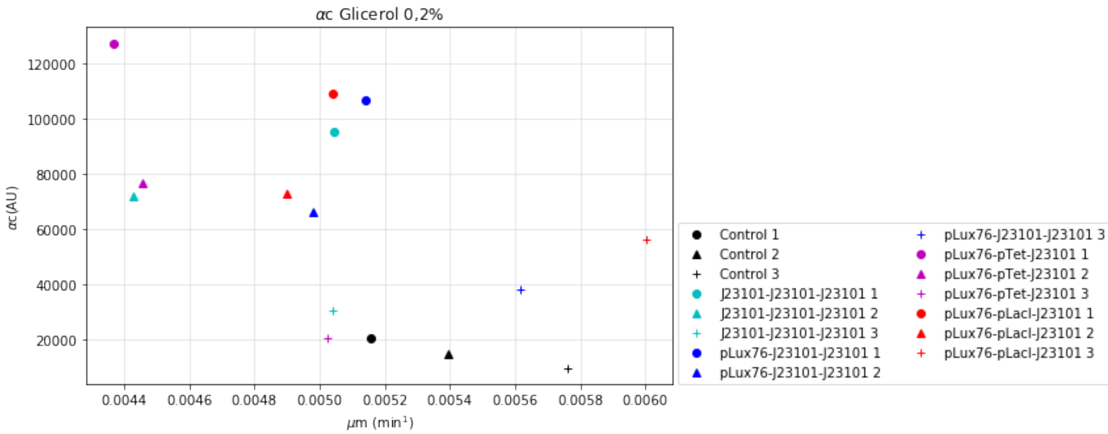

αr Glicerol 0,2%
Control
J23101-J23101-J23101
pLux76-J23101-J23101
pLux76-pTet-J23101
pLux76-pLacI-J23101
Réplica 1
Réplica 2
Réplica 3
200000
160000
120000
80000
40000
0




```
In [78]: #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='J23101-J23101-J23101 1')
         plt.plot(um8,slopec8,'c^',label='J23101-J23101-J23101 2')
         plt.plot(um9,slopec9,'c+',label='J23101-J23101-J23101 3')
         plt.plot(um13,slopec13,'bo',label='pLux76-J23101-J23101 1')
         plt.plot(um14,slopec14,'b^',label='pLux76-J23101-J23101 2')
         plt.plot(um15,slopec15,'b+',label='pLux76-J23101-J23101 3')
         plt.plot(um19,slopec19,'mo',label='pLux76-pTet-J23101 1')
         plt.plot(um20,slopec20,'m^',label='pLux76-pTet-J23101 2')
         plt.plot(um21,slopec21,'m+',label='pLux76-pTet-J23101 3')
         plt.plot(um25,slopec25,'ro',label='pLux76-pLacI-J23101 1')
         plt.plot(um26,slopec26,'r^',label='pLux76-pLacI-J23101 2')
         plt.plot(um27,slopec27,'r+',label='pLux76-pLacI-J23101 3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)
         plt.savefig(filename='alfa-c glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_i


         #grafico de ac versus Um
         plt.figure(figsize=(8,5))
```

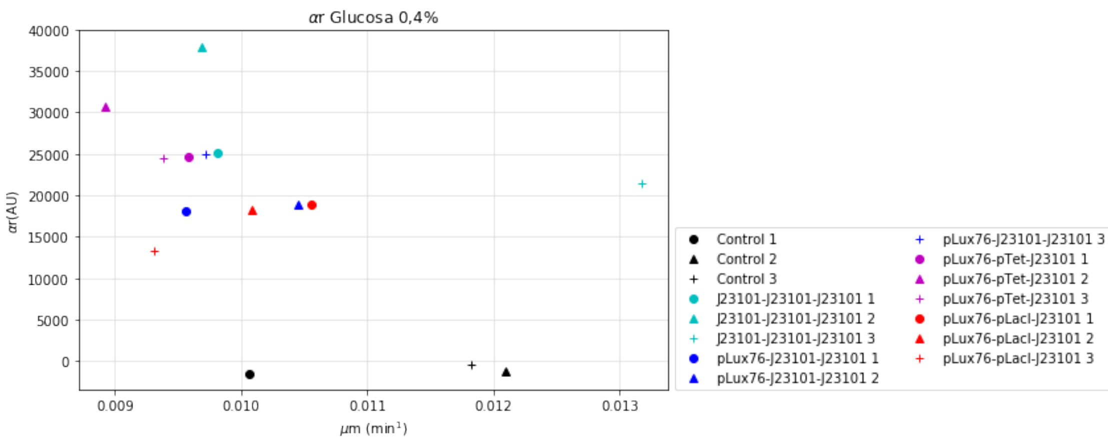
```
plt.title(r'$\alpha$c Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um4,slopec4,'ko',label='Control 1')
plt.plot(um5,slopec5,'k^',label='Control 2')
plt.plot(um6,slopec6,'k+',label='Control 3')
plt.plot(um10,slopec10,'co',label='J23101-J23101-J23101 1')
plt.plot(um11,slopec11,'c^',label='J23101-J23101-J23101 2')
plt.plot(um12,slopec12,'c+',label='J23101-J23101-J23101 3')
plt.plot(um16,slopec16,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um17,slopec17,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um18,slopec18,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um22,slopec22,'mo',label='pLux76-pTet-J23101 1')
plt.plot(um23,slopec23,'m^',label='pLux76-pTet-J23101 2')
plt.plot(um24,slopec24,'m+',label='pLux76-pTet-J23101 3')
plt.plot(um28,slopec28,'ro',label='pLux76-pLacI-J23101 1')
plt.plot(um29,slopec29,'r^',label='pLux76-pLacI-J23101 2')
plt.plot(um30,slopec30,'r+',label='pLux76-pLacI-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-c glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_
```

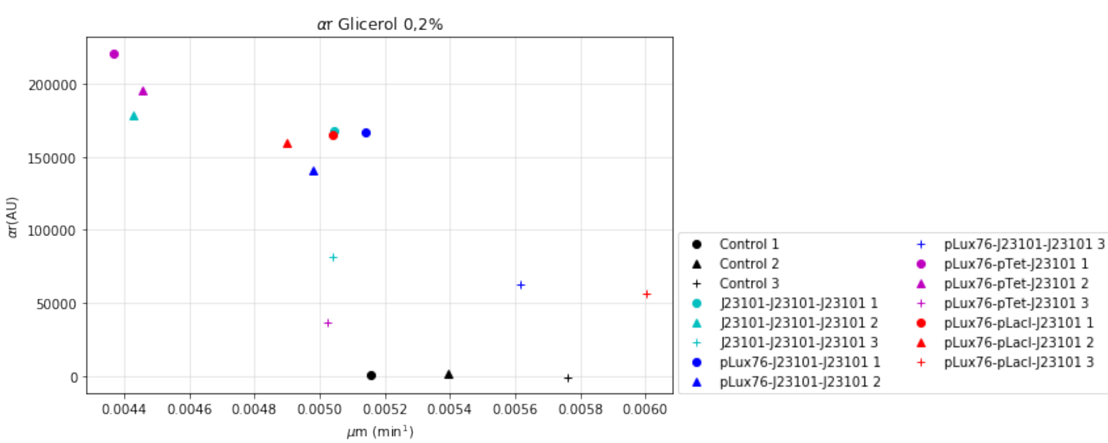




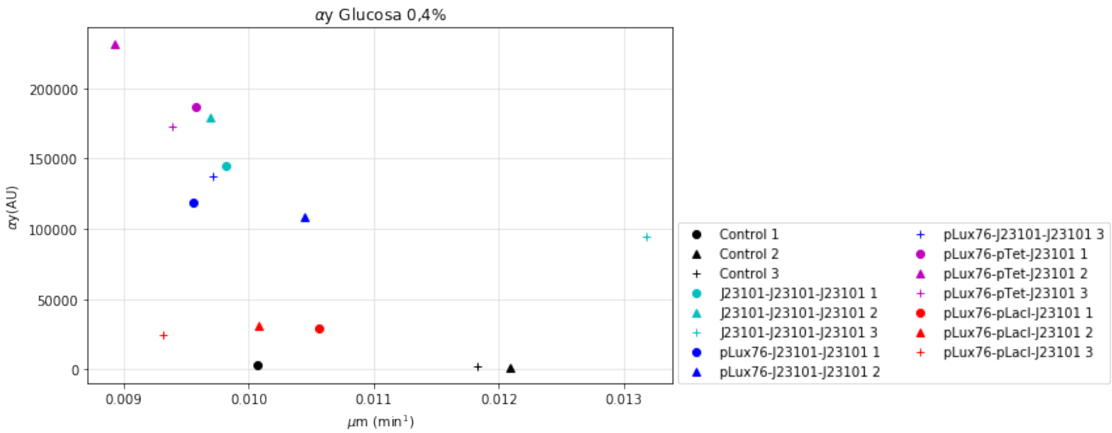
In [79]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'ko',label='Control 1')
plt.plot(um2,sloper2,'k^',label='Control 2')
plt.plot(um3,sloper3,'k+',label='Control 3')
plt.plot(um7,sloper7,'co',label='J23101-J23101-J23101 1')
plt.plot(um8,sloper8,'c^',label='J23101-J23101-J23101 2')
plt.plot(um9,sloper9,'c+',label='J23101-J23101-J23101 3')
plt.plot(um13,sloper13,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um14,sloper14,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um15,sloper15,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um19,sloper19,'mo',label='pLux76-pTet-J23101 1')
plt.plot(um20,sloper20,'m^',label='pLux76-pTet-J23101 2')
plt.plot(um21,sloper21,'m+',label='pLux76-pTet-J23101 3')
plt.plot(um25,sloper25,'ro',label='pLux76-pLacI-J23101 1')
plt.plot(um26,sloper26,'r^',label='pLux76-pLacI-J23101 2')
plt.plot(um27,sloper27,'r+',label='pLux76-pLacI-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-r glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_i


#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')
```

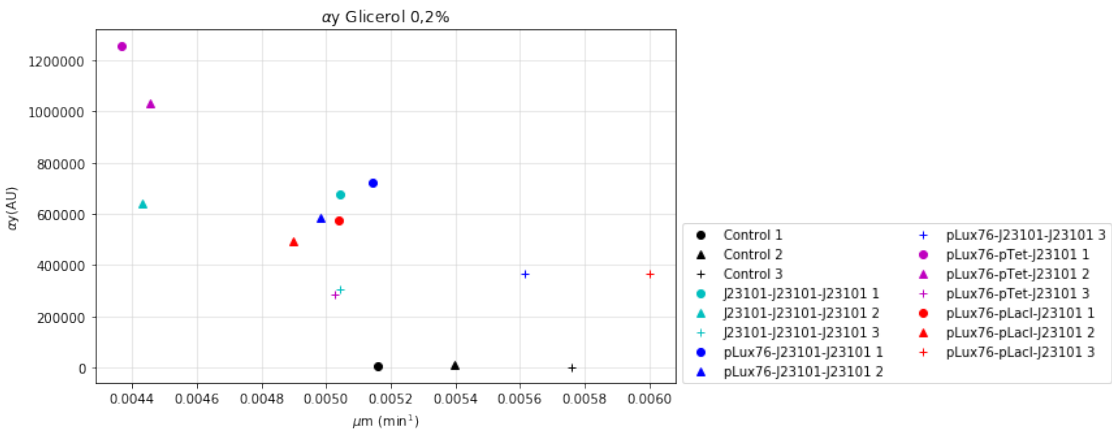
```
plt.plot(um6,sloper6,'k+',label='Control 3')
plt.plot(um10,sloper10,'co',label='J23101-J23101-J23101 1')
plt.plot(um11,sloper11,'c^',label='J23101-J23101-J23101 2')
plt.plot(um12,sloper12,'c+',label='J23101-J23101-J23101 3')
plt.plot(um16,sloper16,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um17,sloper17,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um18,sloper18,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um22,sloper22,'mo',label='pLux76-pTet-J23101 1')
plt.plot(um23,sloper23,'m^',label='pLux76-pTet-J23101 2')
plt.plot(um24,sloper24,'m+',label='pLux76-pTet-J23101 3')
plt.plot(um28,sloper28,'ro',label='pLux76-pLacI-J23101 1')
plt.plot(um29,sloper29,'r^',label='pLux76-pLacI-J23101 2')
plt.plot(um30,sloper30,'r+',label='pLux76-pLacI-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-r glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_
```





```
In [80]: #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='J23101-J23101-J23101 1')
         plt.plot(um8,slopey8,'c^',label='J23101-J23101-J23101 2')
         plt.plot(um9,slopey9,'c+',label='J23101-J23101-J23101 3')
         plt.plot(um13,slopey13,'bo',label='pLux76-J23101-J23101 1')
         plt.plot(um14,slopey14,'b^',label='pLux76-J23101-J23101 2')
         plt.plot(um15,slopey15,'b+',label='pLux76-J23101-J23101 3')
         plt.plot(um19,slopey19,'mo',label='pLux76-pTet-J23101 1')
         plt.plot(um20,slopey20,'m^',label='pLux76-pTet-J23101 2')
         plt.plot(um21,slopey21,'m+',label='pLux76-pTet-J23101 3')
         plt.plot(um25,slopey25,'ro',label='pLux76-pLacI-J23101 1')
         plt.plot(um26,slopey26,'r^',label='pLux76-pLacI-J23101 2')
         plt.plot(um27,slopey27,'r+',label='pLux76-pLacI-J23101 3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)
         plt.savefig(filename='alfa-y glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_i


         #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='J23101-J23101-J23101 1')
         plt.plot(um11,slopey11,'c^',label='J23101-J23101-J23101 2')
         plt.plot(um12,slopey12,'c+',label='J23101-J23101-J23101 3')
         plt.plot(um16,slopey16,'bo',label='pLux76-J23101-J23101 1')
         plt.plot(um17,slopey17,'b^',label='pLux76-J23101-J23101 2')
         plt.plot(um18,slopey18,'b+',label='pLux76-J23101-J23101 3')
         plt.plot(um22,slopey22,'mo',label='pLux76-pTet-J23101 1')
         plt.plot(um23,slopey23,'m^',label='pLux76-pTet-J23101 2')
         plt.plot(um24,slopey24,'m+',label='pLux76-pTet-J23101 3')
         plt.plot(um28,slopey28,'ro',label='pLux76-pLacI-J23101 1')
         plt.plot(um29,slopey29,'r^',label='pLux76-pLacI-J23101 2')
         plt.plot(um30,slopey30,'r+',label='pLux76-pLacI-J23101 3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)
         plt.savefig(filename='alfa-y glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_
```

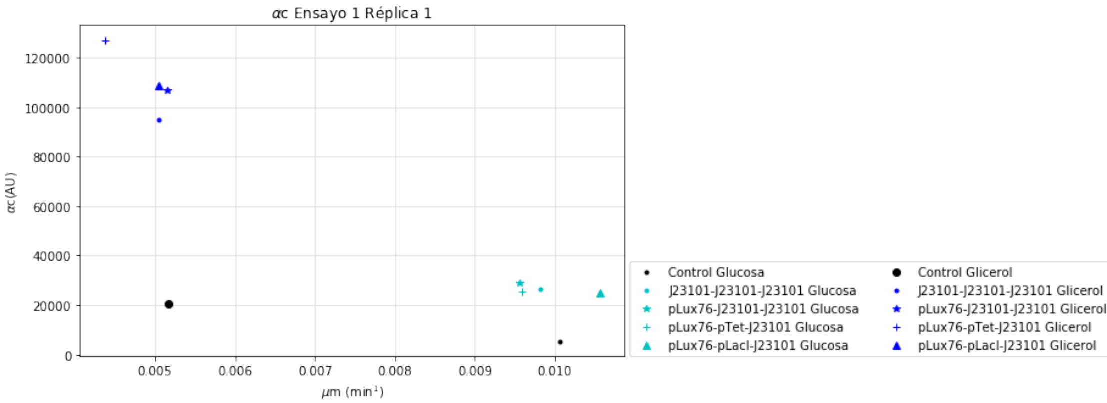


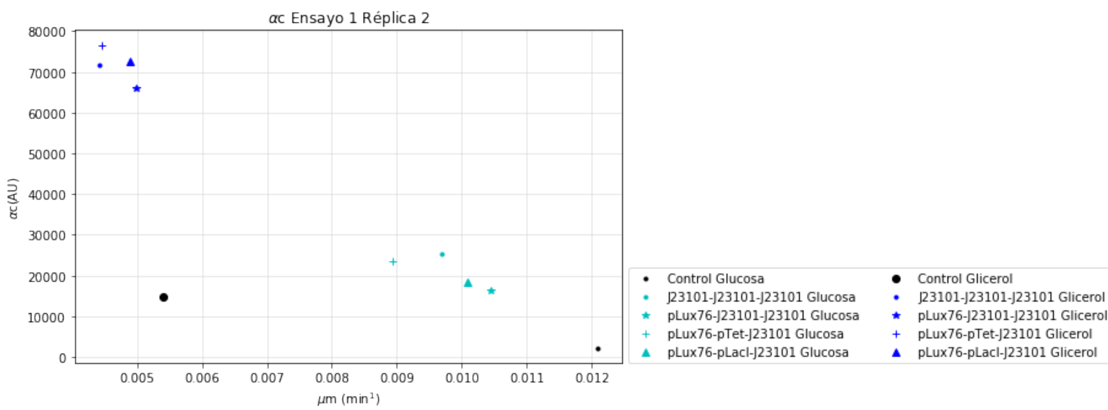


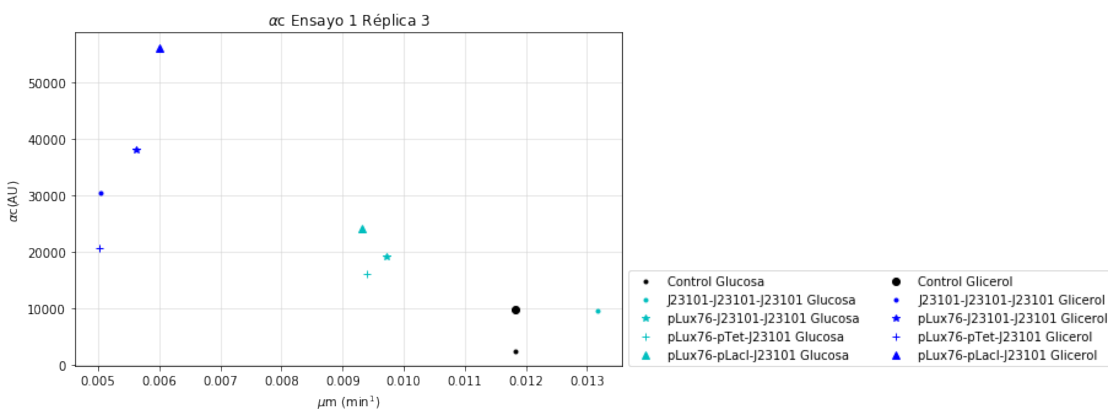
In [81]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 1 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um1,slopec1,'k.',label='Control Glucosa')
plt.plot(um7,slopec7,'c.',label='J23101-J23101-J23101 Glucosa')
plt.plot(um13,slopec13,'c*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um19,slopec19,'c+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um25,slopec25,'c^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um4,slopec4,'ko',label='Control Glicerol')
plt.plot(um10,slopec10,'b.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um16,slopec16,'b*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um22,slopec22,'b+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um28,slopec28,'b^',label='pLux76-pLacI-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
```

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

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um14,slopec14,'c*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um20,slopec20,'c+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um26,slopec26,'c^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um5,slopec5,'ko',label='Control Glicerol')
plt.plot(um11,slopec11,'b.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um17,slopec17,'b*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um23,slopec23,'b+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um29,slopec29,'b^',label='pLux76-pLacI-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 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um15,slopec15,'c*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um21,slopec21,'c+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um27,slopec27,'c^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um3,slopec6,'ko',label='Control Glicerol')
plt.plot(um12,slopec12,'b.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um18,slopec18,'b*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um24,slopec24,'b+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um30,slopec30,'b^',label='pLux76-pLacI-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[81]: <matplotlib.legend.Legend at 0x1ababdeab00>





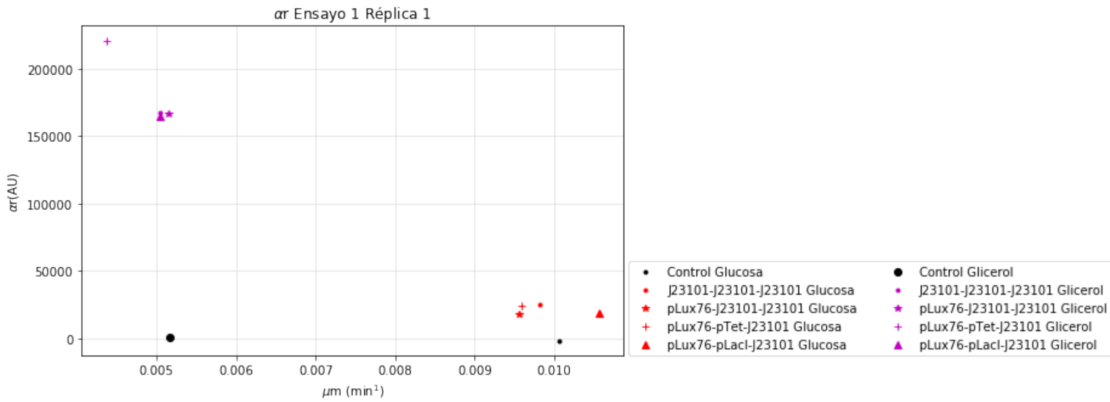


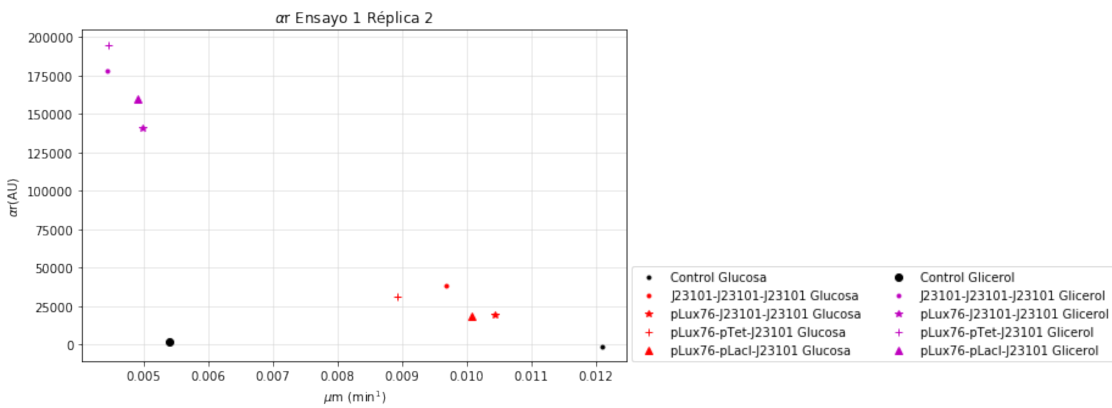
In [82]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
```

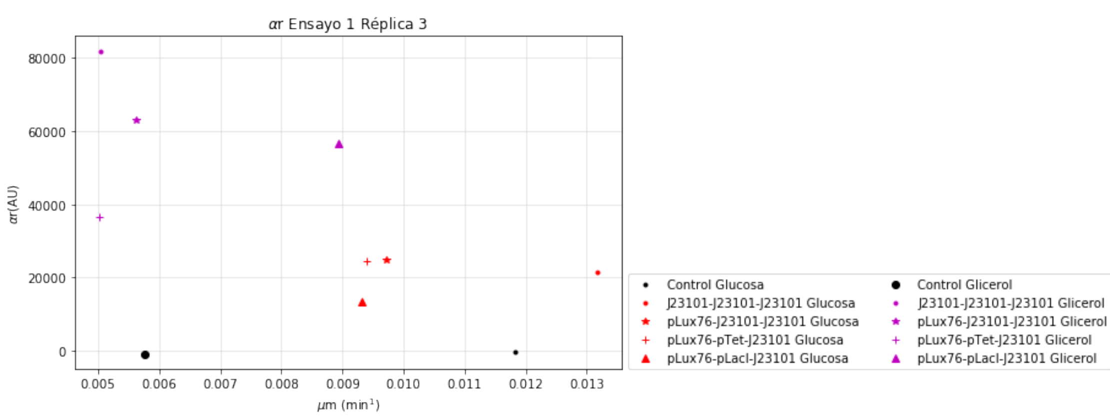
```
plt.title(r'$\alpha$r Ensayo 1 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'k.',label='Control Glucosa ')
plt.plot(um7,sloper7,'r.',label='J23101-J23101-J23101 Glucosa ')
plt.plot(um13,sloper13,'r*',label='pLux76-J23101-J23101 Glucosa ')
plt.plot(um19,sloper19,'r+',label='pLux76-pTet-J23101 Glucosa ')
plt.plot(um25,sloper25,'r^',label='pLux76-pLacI-J23101 Glucosa ')
plt.plot(um4,sloper4,'ko',label='Control Glicerol ')
plt.plot(um10,sloper10,'m.',label='J23101-J23101-J23101 Glicerol ')
plt.plot(um16,sloper16,'m*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um22,sloper22,'m+',label='pLux76-pTet-J23101 Glicerol ')
plt.plot(um28,sloper28,'m^',label='pLux76-pLacI-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 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um14,sloper14,'r*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um20,sloper20,'r+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um26,sloper26,'r^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um5,sloper5,'ko',label='Control Glicerol')
plt.plot(um11,sloper11,'m.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um17,sloper17,'m*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um23,sloper23,'m+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um29,sloper29,'m^',label='pLux76-pLacI-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 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um15,sloper15,'r*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um21,sloper21,'r+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um27,sloper27,'r^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um6,sloper6,'ko',label='Control Glicerol')
plt.plot(um12,sloper12,'m.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um18,sloper18,'m*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um24,sloper24,'m+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um20,sloper30,'m^',label='pLux76-pLacI-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
```

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

Out[82]: <matplotlib.legend.Legend at 0x1ababd14dd8>



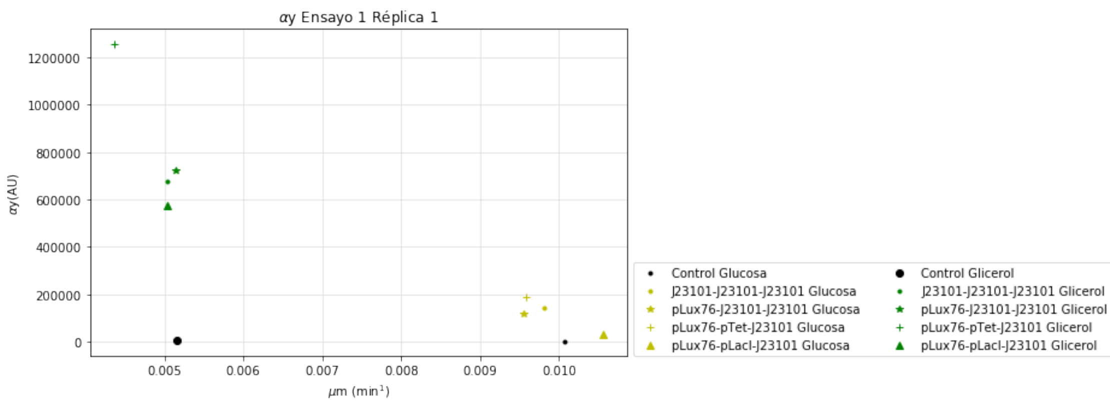


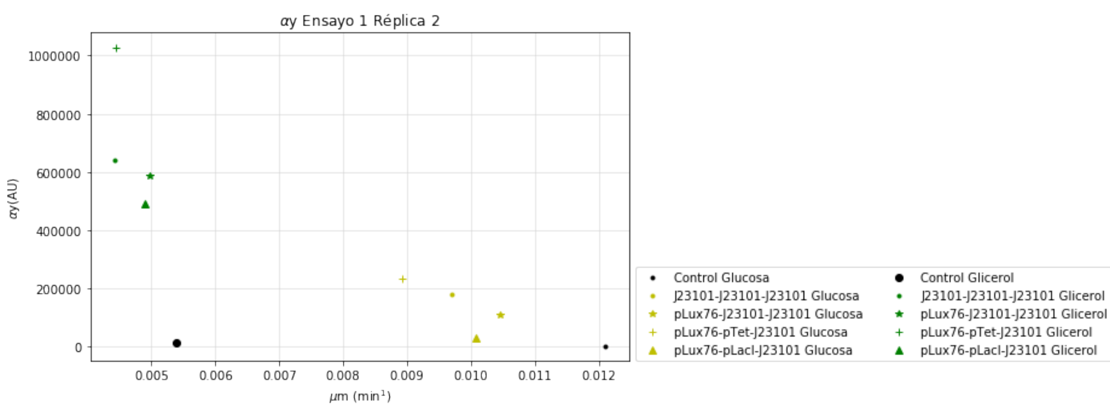


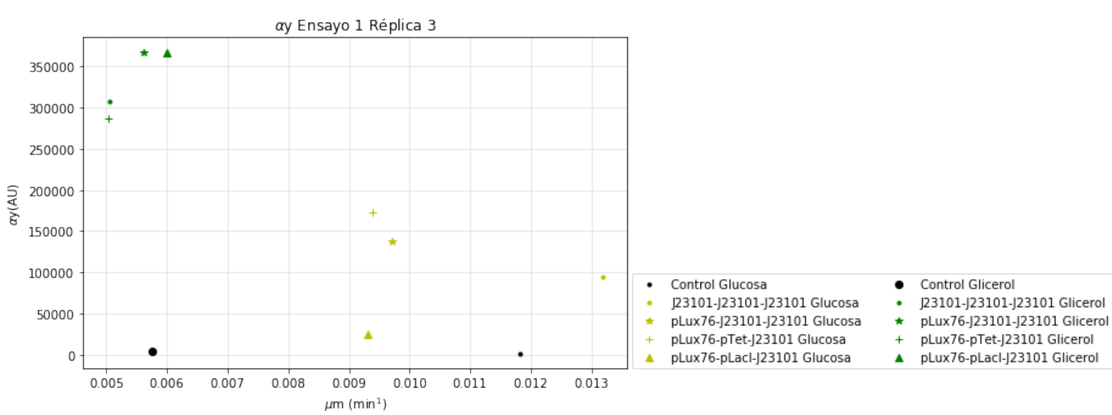
In [83]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 1 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'k.',label='Control Glucosa ')
plt.plot(um7,slopey7,'y.',label='J23101-J23101-J23101 Glucosa ')
plt.plot(um13,slopey13,'y*',label='pLux76-J23101-J23101 Glucosa ')
plt.plot(um19,slopey19,'y+',label='pLux76-pTet-J23101 Glucosa ')
plt.plot(um25,slopey25,'y^',label='pLux76-pLacI-J23101 Glucosa ')
plt.plot(um4,slopey4,'ko',label='Control Glicerol ')
plt.plot(um10,slopey10,'g.',label='J23101-J23101-J23101 Glicerol ')
plt.plot(um16,slopey16,'g*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um22,slopey22,'g+',label='pLux76-pTet-J23101 Glicerol ')
plt.plot(um28,slopey28,'g^',label='pLux76-pLacI-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 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um14,slopey14,'y*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um20,slopey20,'y+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um26,slopey26,'y^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um5,slopey5,'ko',label='Control Glicerol')
plt.plot(um11,slopey11,'g.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um17,slopey17,'g*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um23,slopey23,'g+',label='pLux76-pTet-J23101 Glicerol')
plt.plot(um29,slopey29,'g^',label='pLux76-pLacI-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 1 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='J23101-J23101-J23101 Glucosa')
plt.plot(um15,slopey15,'y*',label='pLux76-J23101-J23101 Glucosa')
plt.plot(um21,slopey21,'y+',label='pLux76-pTet-J23101 Glucosa')
plt.plot(um27,slopey27,'y^',label='pLux76-pLacI-J23101 Glucosa')
plt.plot(um6,slopey6,'ko',label='Control Glicerol')
plt.plot(um12,slopey12,'g.',label='J23101-J23101-J23101 Glicerol')
plt.plot(um18,slopey18,'g*',label='pLux76-J23101-J23101 Glicerol')
plt.plot(um24,slopey24,'g+',label='pLux76-pTet-J23101 Glicerol')
```

```python
plt.plot(um30,slopey30,'g^',label='pLux76-pLacI-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[83]: <matplotlib.legend.Legend at 0x1ababe55fd0>

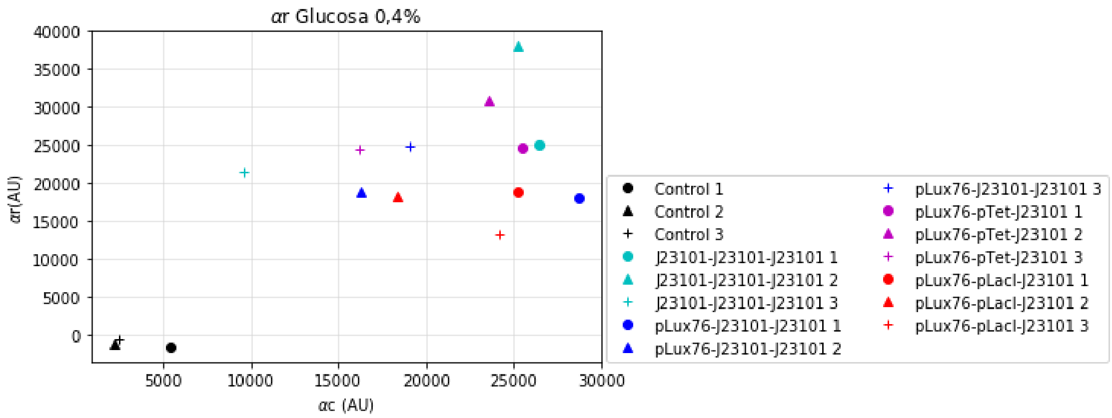


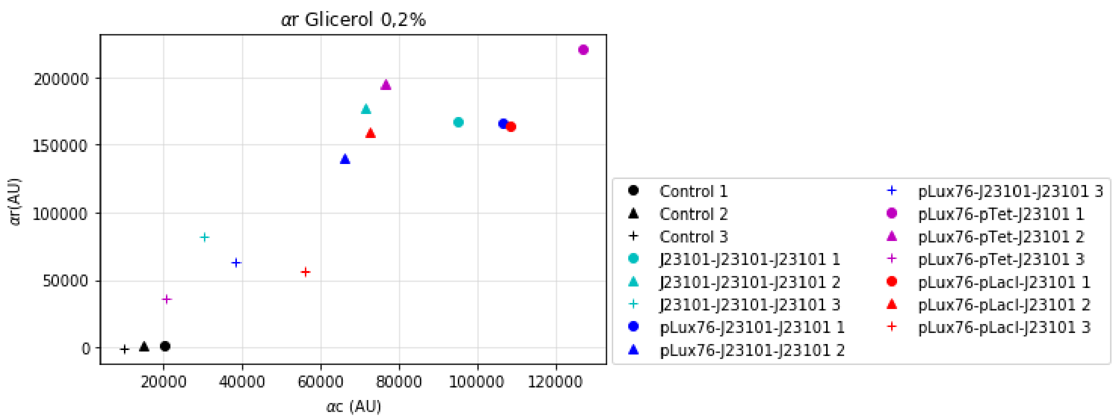




```
In [84]: #grafico de ac versus Um
         plt.figure()
         plt.title(r'$\alpha$r Glucosa 0,4%')
         plt.xlabel(r'$\alpha$c (AU)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(slopec1,sloper1,'ko',label='Control 1')
         plt.plot(slopec2,sloper2,'k^',label='Control 2')
         plt.plot(slopec3,sloper3,'k+',label='Control 3')
         plt.plot(slopec7,sloper7,'co',label='J23101-J23101-J23101 1')
         plt.plot(slopec8,sloper8,'c^',label='J23101-J23101-J23101 2')
         plt.plot(slopec9,sloper9,'c+',label='J23101-J23101-J23101 3')
         plt.plot(slopec13,sloper13,'bo',label='pLux76-J23101-J23101 1')
         plt.plot(slopec14,sloper14,'b^',label='pLux76-J23101-J23101 2')
         plt.plot(slopec15,sloper15,'b+',label='pLux76-J23101-J23101 3')
         plt.plot(slopec19,sloper19,'mo',label='pLux76-pTet-J23101 1')
         plt.plot(slopec20,sloper20,'m^',label='pLux76-pTet-J23101 2')
         plt.plot(slopec21,sloper21,'m+',label='pLux76-pTet-J23101 3')
         plt.plot(slopec25,sloper25,'ro',label='pLux76-pLacI-J23101 1')
         plt.plot(slopec26,sloper26,'r^',label='pLux76-pLacI-J23101 2')
         plt.plot(slopec27,sloper27,'r+',label='pLux76-pLacI-J23101 3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)
         plt.savefig(filename='alfa-r y c glucosa.png', dpi=300, facecolor='w', edgecolor='w',bb


         #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='J23101-J23101-J23101 1')
         plt.plot(slopec11,sloper11,'c^',label='J23101-J23101-J23101 2')
         plt.plot(slopec12,sloper12,'c+',label='J23101-J23101-J23101 3')
         plt.plot(slopec16,sloper16,'bo',label='pLux76-J23101-J23101 1')
         plt.plot(slopec17,sloper17,'b^',label='pLux76-J23101-J23101 2')
         plt.plot(slopec18,sloper18,'b+',label='pLux76-J23101-J23101 3')
         plt.plot(slopec22,sloper22,'mo',label='pLux76-pTet-J23101 1')
         plt.plot(slopec23,sloper23,'m^',label='pLux76-pTet-J23101 2')
         plt.plot(slopec24,sloper24,'m+',label='pLux76-pTet-J23101 3')
         plt.plot(slopec28,sloper28,'ro',label='pLux76-pLacI-J23101 1')
         plt.plot(slopec29,sloper29,'r^',label='pLux76-pLacI-J23101 2')
         plt.plot(slopec30,sloper30,'r+',label='pLux76-pLacI-J23101 3')
```

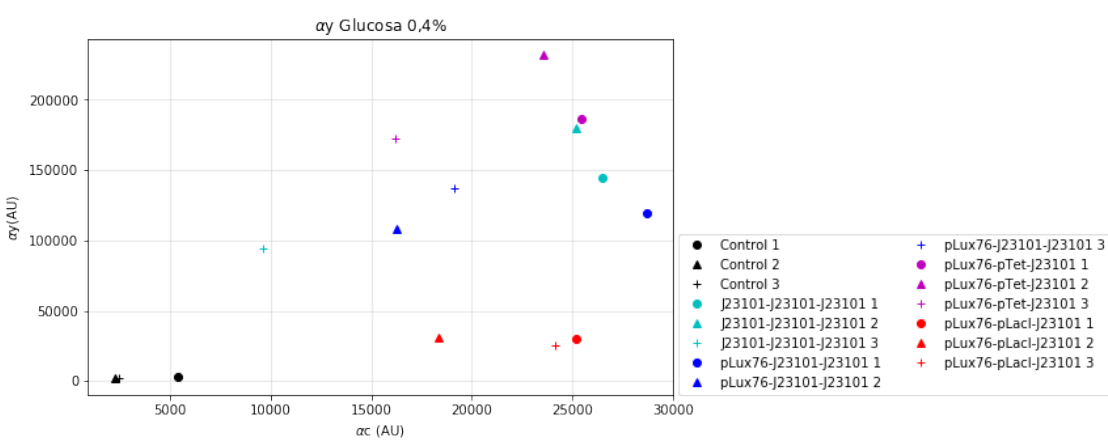
```
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-r y c glicerol.png', dpi=300, facecolor='w', edgecolor='w',b
```

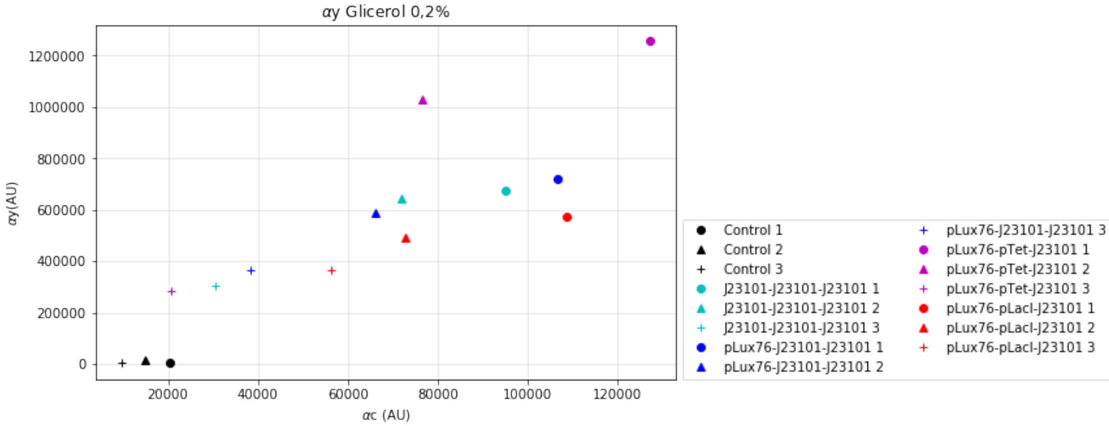




In [85]:
```
#grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopey1,'ko',label='Control 1')
plt.plot(slopec2,slopey2,'k^',label='Control 2')
plt.plot(slopec3,slopey3,'k+',label='Control 3')
plt.plot(slopec7,slopey7,'co',label='J23101-J23101-J23101 1')
plt.plot(slopec8,slopey8,'c^',label='J23101-J23101-J23101 2')
plt.plot(slopec9,slopey9,'c+',label='J23101-J23101-J23101 3')
plt.plot(slopec13,slopey13,'bo',label='pLux76-J23101-J23101 1')
plt.plot(slopec14,slopey14,'b^',label='pLux76-J23101-J23101 2')
```

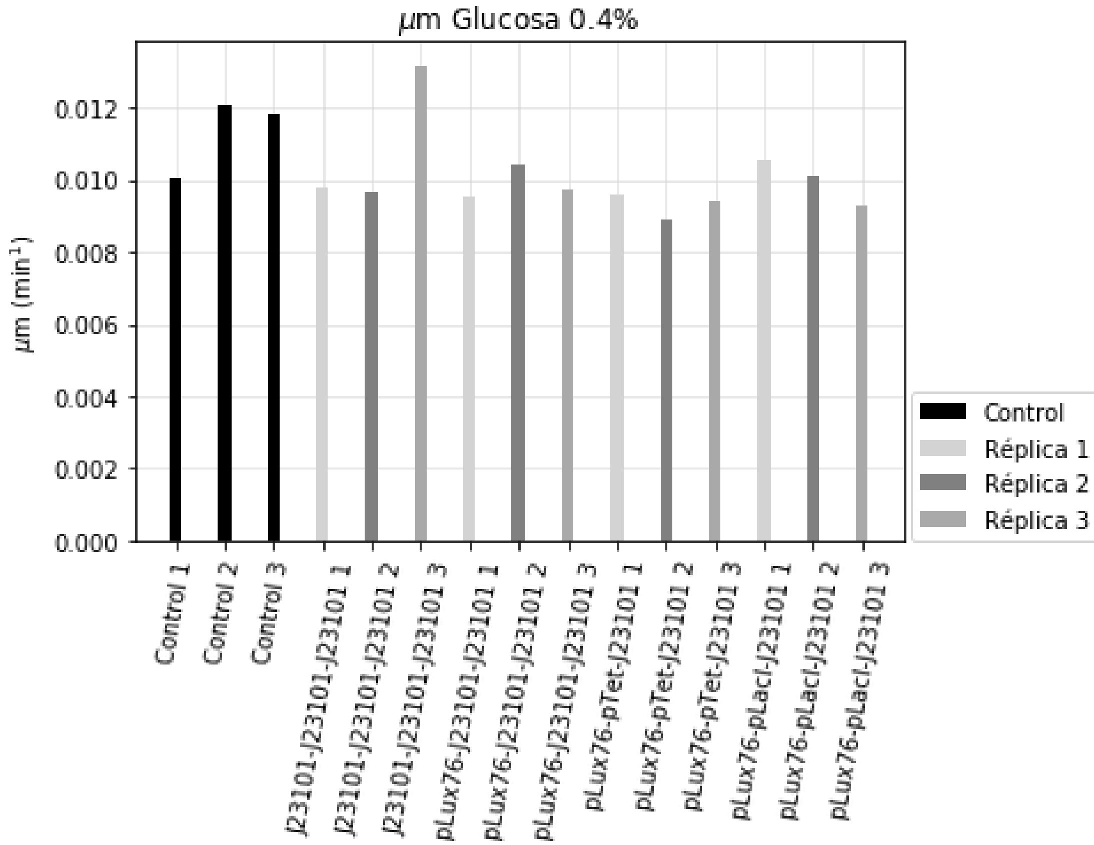
```
plt.plot(slopec15,slopey15,'b+',label='pLux76-J23101-J23101 3')
plt.plot(slopec19,slopey19,'mo',label='pLux76-pTet-J23101 1')
plt.plot(slopec20,slopey20,'m^',label='pLux76-pTet-J23101 2')
plt.plot(slopec21,slopey21,'m+',label='pLux76-pTet-J23101 3')
plt.plot(slopec25,slopey25,'ro',label='pLux76-pLacI-J23101 1')
plt.plot(slopec26,slopey26,'r^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec27,slopey27,'r+',label='pLux76-pLacI-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-y y c glucosa.png', dpi=300, facecolor='w', edgecolor='w',bb


#grafico de ay vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec4,slopey4,'ko',label='Control 1')
plt.plot(slopec5,slopey5,'k^',label='Control 2')
plt.plot(slopec6,slopey6,'k+',label='Control 3')
plt.plot(slopec10,slopey10,'co',label='J23101-J23101-J23101 1')
plt.plot(slopec11,slopey11,'c^',label='J23101-J23101-J23101 2')
plt.plot(slopec12,slopey12,'c+',label='J23101-J23101-J23101 3')
plt.plot(slopec16,slopey16,'bo',label='pLux76-J23101-J23101 1')
plt.plot(slopec17,slopey17,'b^',label='pLux76-J23101-J23101 2')
plt.plot(slopec18,slopey18,'b+',label='pLux76-J23101-J23101 3')
plt.plot(slopec22,slopey22,'mo',label='pLux76-pTet-J23101 1')
plt.plot(slopec23,slopey23,'m^',label='pLux76-pTet-J23101 2')
plt.plot(slopec24,slopey24,'m+',label='pLux76-pTet-J23101 3')
plt.plot(slopec28,slopey28,'ro',label='pLux76-pLacI-J23101 1')
plt.plot(slopec29,slopey29,'r^',label='pLux76-pLacI-J23101 2')
plt.plot(slopec30,slopey30,'r+',label='pLux76-pLacI-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='alfa-y y c glicerol.png', dpi=300, facecolor='w', edgecolor='w',b
```





In [86]:
```
#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','J23101-J23101-J23101 1','J23101-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

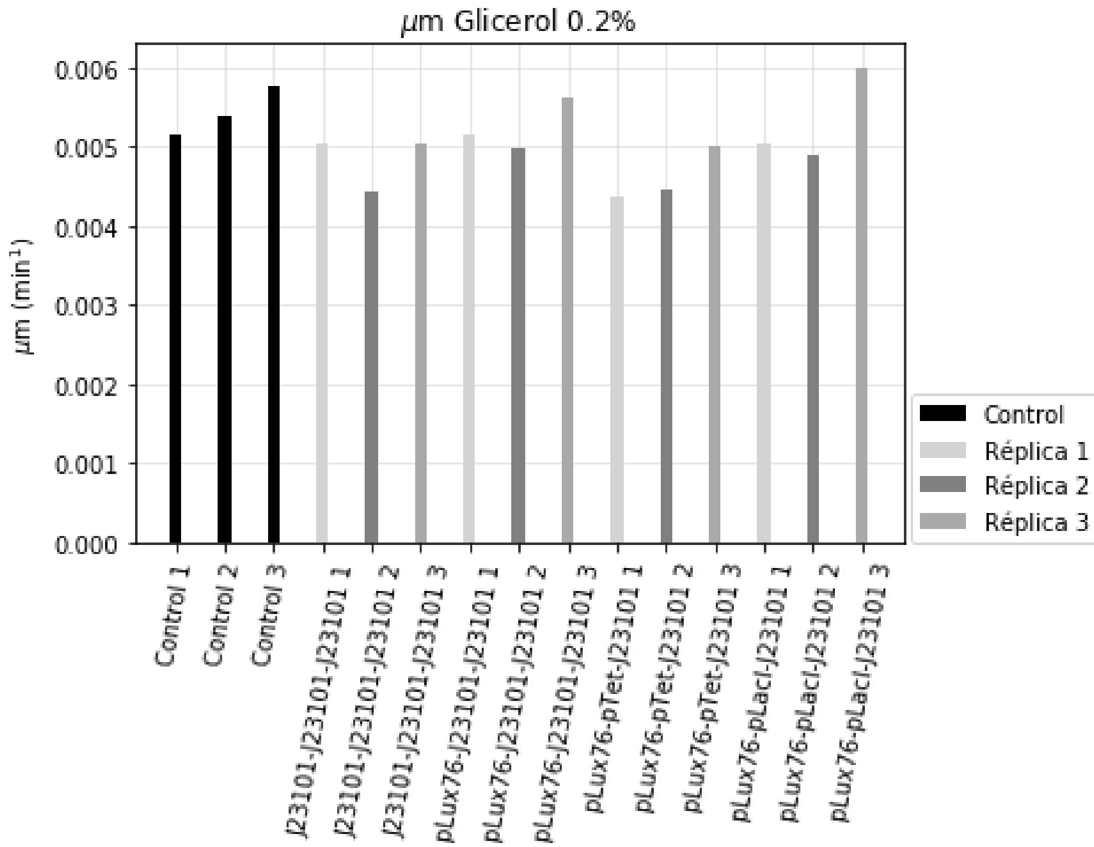
Out[86]: <matplotlib.legend.Legend at 0x1abac4e63c8>




```
In [87]: 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','J23101-J23101-J23101 1','J23101-J231
```

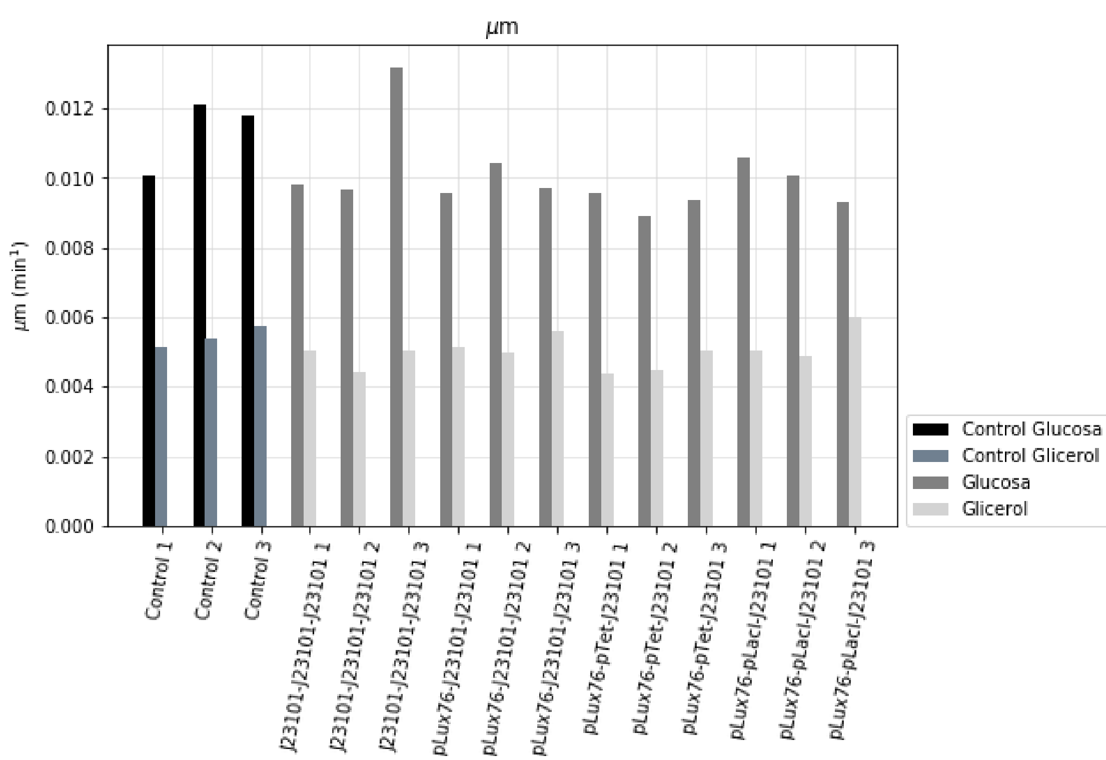
```
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
        plt.legend(loc=(1.01,0.0))
```

Out[87]: <matplotlib.legend.Legend at 0x1abacfa3940>




```
In [88]: 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','J23101-J23101-J23101 1','J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
plt.savefig(filename='um todo.png', dpi=300, facecolor='w', edgecolor='w',bbox_inches='
```



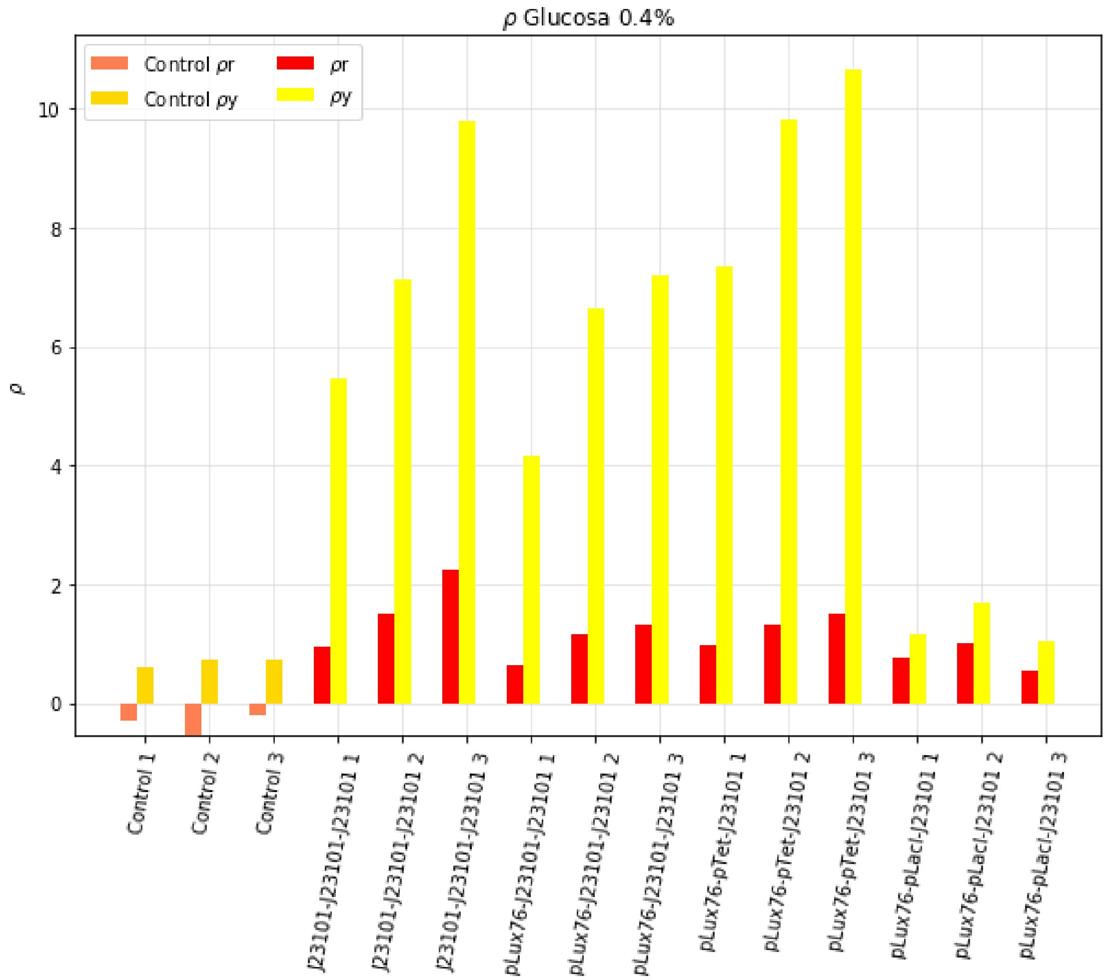
In [89]:
```
#Ro RFP
pr1=sloper1/slopec1
pr2=sloper2/slopec2
pr3=sloper3/slopec3
pr4=sloper4/slopec4
pr5=sloper5/slopec5
pr6=sloper6/slopec6
pr7=sloper7/slopec7
pr8=sloper8/slopec8
pr9=sloper9/slopec9
pr10=sloper10/slopec10
pr11=sloper11/slopec11
pr12=sloper12/slopec12
pr13=sloper13/slopec13
pr14=sloper14/slopec14
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 [90]:
```
#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,
```

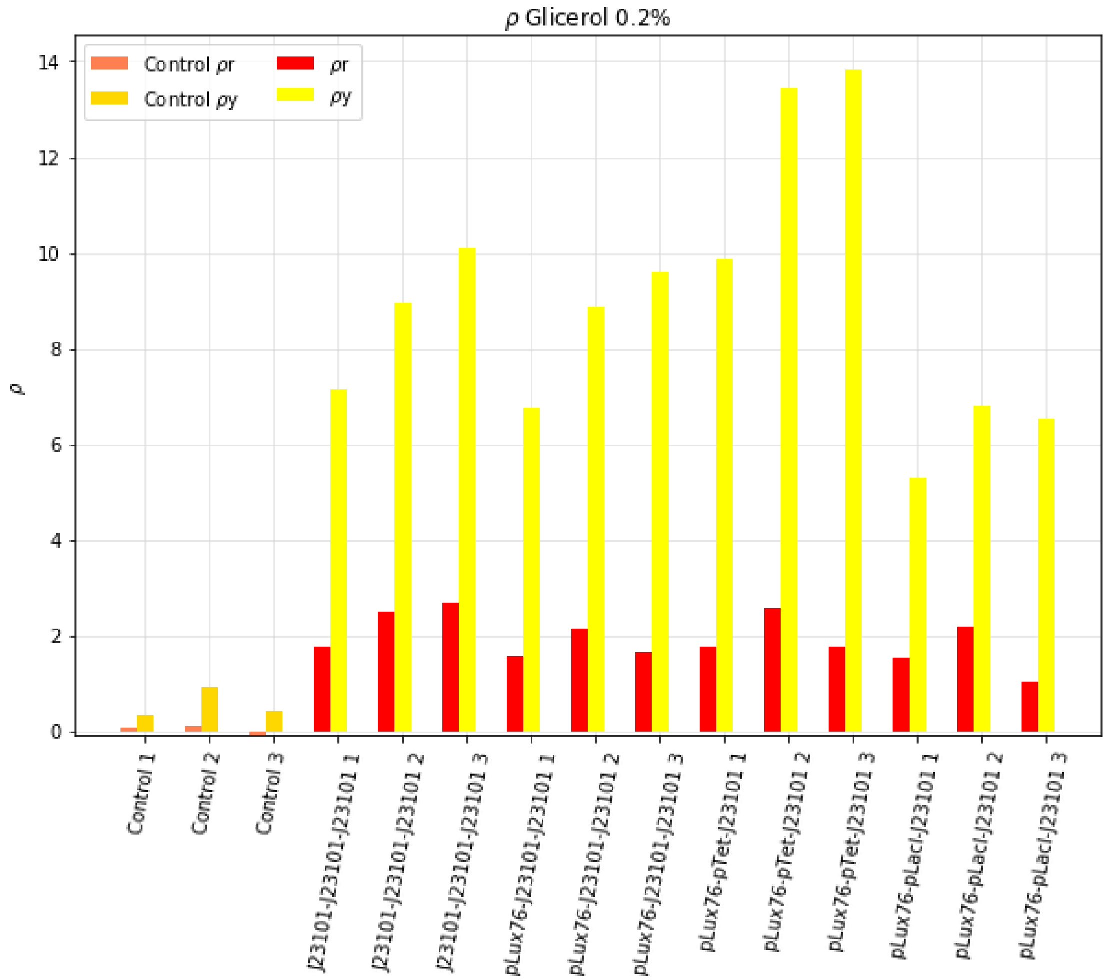
In [91]:
```
#grafico ro de yfp y de rfp
ro_rfp=[pr1,pr2,pr3,pr4,pr5,pr6,pr7,pr8,pr9,pr10,pr11,pr12,pr13,pr14,pr15,pr16,pr17,pr1
ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py1


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','J23101-J23101-J23101 1','J23101-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
plt.savefig(filename='rho glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_inch
```




```
In [92]: 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)
```

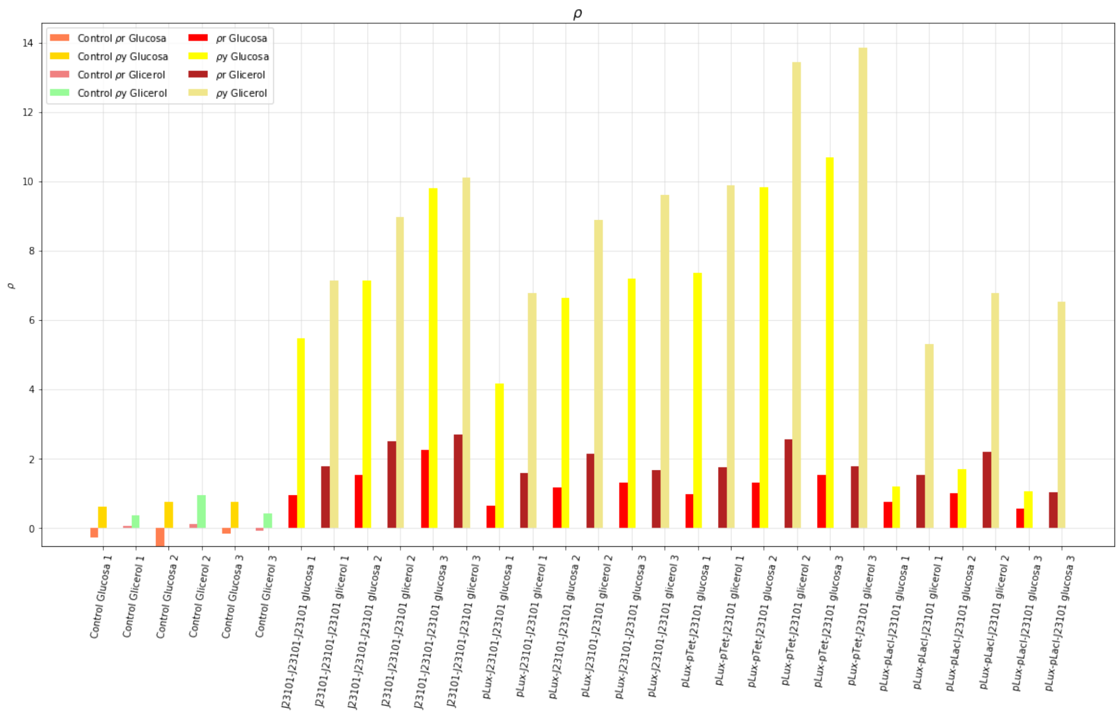
```
plt.bar(X[6]+0.00,ro_yfp[15],color='yellow',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[16],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[16],color='yellow',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[17],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[17],color='yellow',width=0.25,zorder=3)
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','J23101-J23101-J23101 1','J23101-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
plt.savefig(filename='rho glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_inc
```



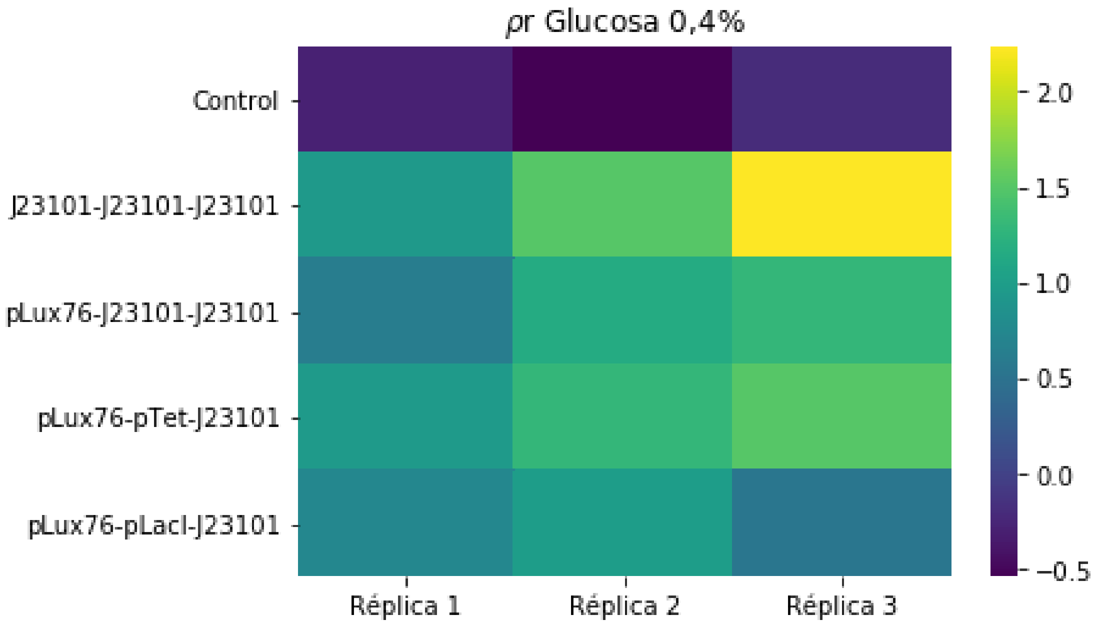

```
In [93]: X = np.arange(30)
         plt.figure(figsize=(20,10))
         plt.title(r'$\rho$',fontsize=15.0)
         plt.ylabel(r'$\rho$')
         plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r Glu
         plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y Gluc
         plt.bar(X[1]-0.25,ro_rfp[3],color='lightcoral',width=0.25,label= 'Control'+' '+r'$\rho$
         plt.bar(X[1]+0.00,ro_yfp[3],color='palegreen',width=0.25,label= 'Control'+' '+r'$\rho$y
         plt.bar(X[2]-0.25,ro_rfp[1],color='coral',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,ro_yfp[1],color='gold',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,ro_rfp[4],color='lightcoral',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,ro_yfp[4],color='palegreen',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,ro_rfp[2],color='coral',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,ro_yfp[2],color='gold',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,ro_rfp[5],color='lightcoral',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,ro_yfp[5],color='palegreen',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,ro_rfp[6],color='r',width=0.25,label=r'$\rho$r Glucosa',zorder=3)
```

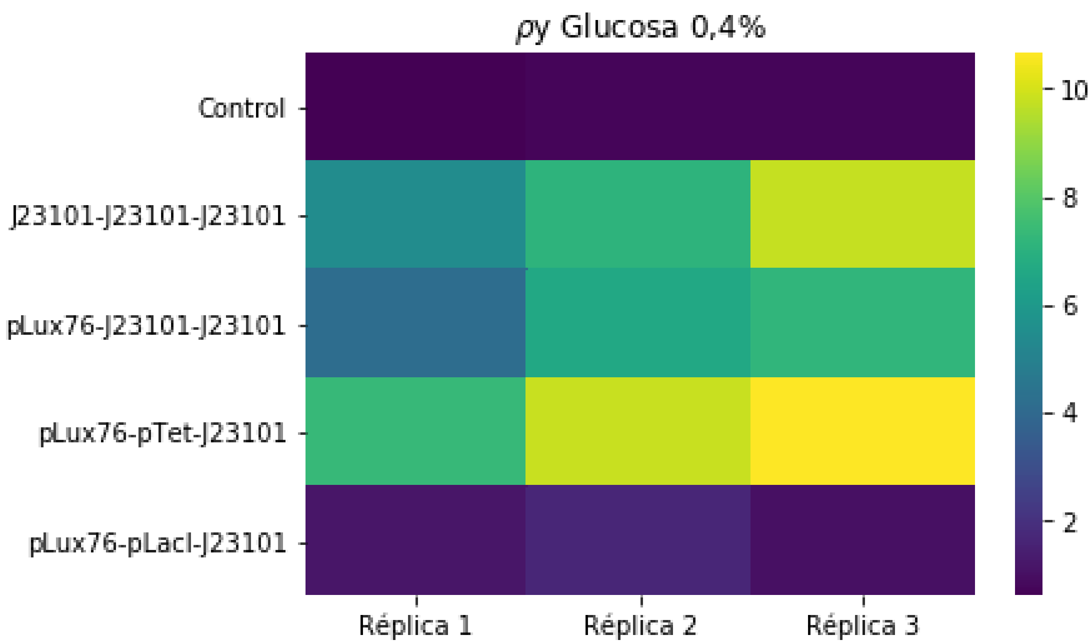
```
plt.bar(X[6]+0.00,ro_yfp[6],color='yellow',width=0.25,label=r'$\rho$y Glucosa',zorder=3
plt.bar(X[7]-0.25,ro_rfp[9],color='firebrick',width=0.25,label=r'$\rho$r Glicerol',zord
plt.bar(X[7]+0.00,ro_yfp[9],color='khaki',width=0.25,label=r'$\rho$y Glicerol',zorder=3
plt.bar(X[8]-0.25,ro_rfp[7],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[7],color='yellow',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[10],color='firebrick',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ro_yfp[10],color='khaki',width=0.25,zorder=3)
plt.bar(X[10]-0.25,ro_rfp[8],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[8],color='yellow',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[11],color='firebrick',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[11],color='khaki',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[12],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[12],color='yellow',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[15],color='firebrick',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[15],color='khaki',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[13],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[13],color='yellow',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[16],color='firebrick',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[16],color='khaki',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[14],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[14],color='yellow',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[17],color='firebrick',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[17],color='khaki',width=0.25,zorder=3)
plt.bar(X[18]-0.25,ro_rfp[18],color='r',width=0.25,zorder=3)
plt.bar(X[18]+0.00,ro_yfp[18],color='yellow',width=0.25,zorder=3)
plt.bar(X[19]-0.25,ro_rfp[21],color='firebrick',width=0.25,zorder=3)
plt.bar(X[19]+0.00,ro_yfp[21],color='khaki',width=0.25,zorder=3)
plt.bar(X[20]-0.25,ro_rfp[19],color='r',width=0.25,zorder=3)
plt.bar(X[20]+0.00,ro_yfp[19],color='yellow',width=0.25,zorder=3)
plt.bar(X[21]-0.25,ro_rfp[22],color='firebrick',width=0.25,zorder=3)
plt.bar(X[21]+0.00,ro_yfp[22],color='khaki',width=0.25,zorder=3)
plt.bar(X[22]-0.25,ro_rfp[20],color='r',width=0.25,zorder=3)
plt.bar(X[22]+0.00,ro_yfp[20],color='yellow',width=0.25,zorder=3)
plt.bar(X[23]-0.25,ro_rfp[23],color='firebrick',width=0.25,zorder=3)
plt.bar(X[23]+0.00,ro_yfp[23],color='khaki',width=0.25,zorder=3)
plt.bar(X[24]-0.25,ro_rfp[24],color='r',width=0.25,zorder=3)
plt.bar(X[24]+0.00,ro_yfp[24],color='yellow',width=0.25,zorder=3)
plt.bar(X[25]-0.25,ro_rfp[27],color='firebrick',width=0.25,zorder=3)
plt.bar(X[25]+0.00,ro_yfp[27],color='khaki',width=0.25,zorder=3)
plt.bar(X[26]-0.25,ro_rfp[25],color='r',width=0.25,zorder=3)
plt.bar(X[26]+0.00,ro_yfp[25],color='yellow',width=0.25,zorder=3)
plt.bar(X[27]-0.25,ro_rfp[28],color='firebrick',width=0.25,zorder=3)
plt.bar(X[27]+0.00,ro_yfp[28],color='khaki',width=0.25,zorder=3)
plt.bar(X[28]-0.25,ro_rfp[26],color='r',width=0.25,zorder=3)
plt.bar(X[28]+0.00,ro_yfp[26],color='yellow',width=0.25,zorder=3)
plt.bar(X[29]-0.25,ro_rfp[29],color='firebrick',width=0.25,zorder=3)
plt.bar(X[29]+0.00,ro_yfp[29],color='khaki',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control Gl
```

```
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
plt.savefig(filename='rho todo.png', dpi=300, facecolor='w', edgecolor='w',bbox_inches=
```



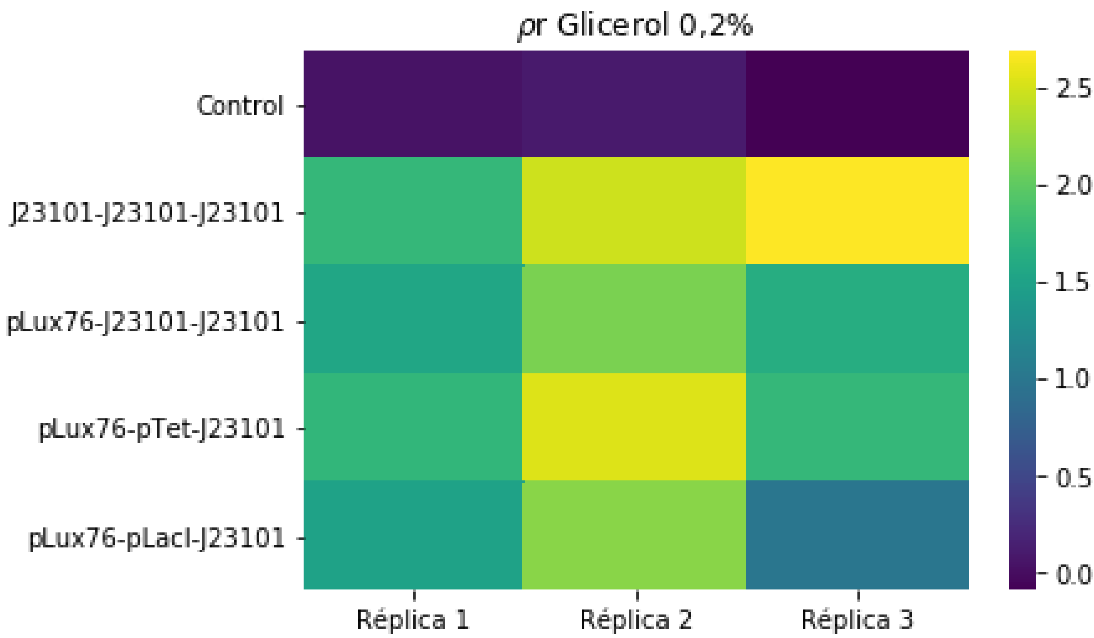

```
In [94]: ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr2

         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','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

         plt.figure()
         plt.title(r'$\rho$r Glucosa 0,4%')
         sns.heatmap(ro_rfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='rho-r glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_in


         plt.figure()
         plt.title(r'$\rho$y Glucosa 0,4%')
         sns.heatmap(ro_yfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
         plt.savefig(filename='rho-y glucosa.png', dpi=300, facecolor='w', edgecolor='w',bbox_in
```

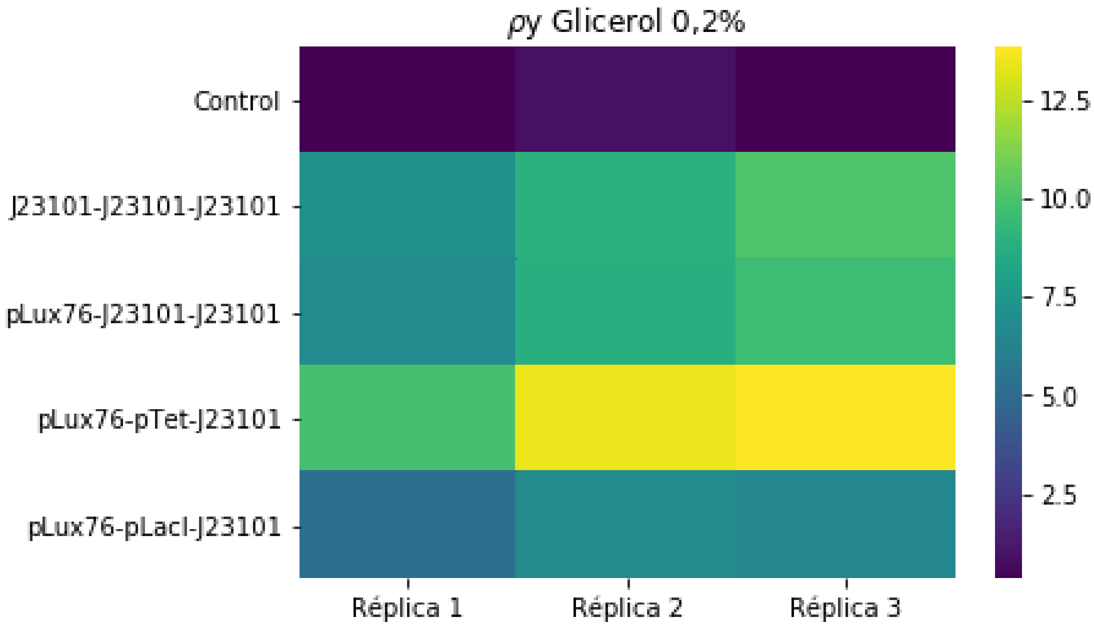




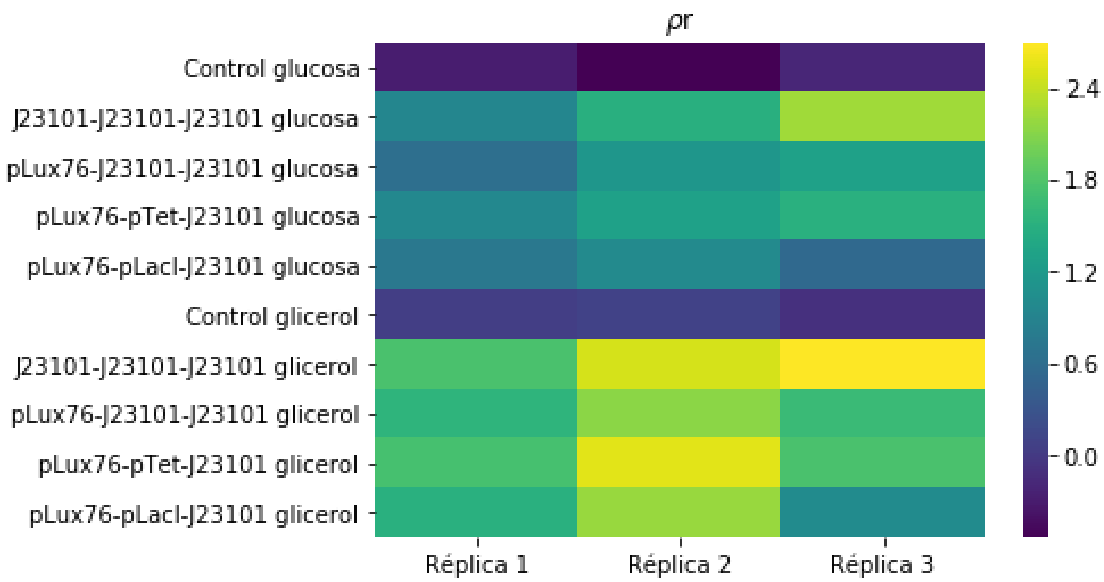

```
In [95]: ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
         ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,

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

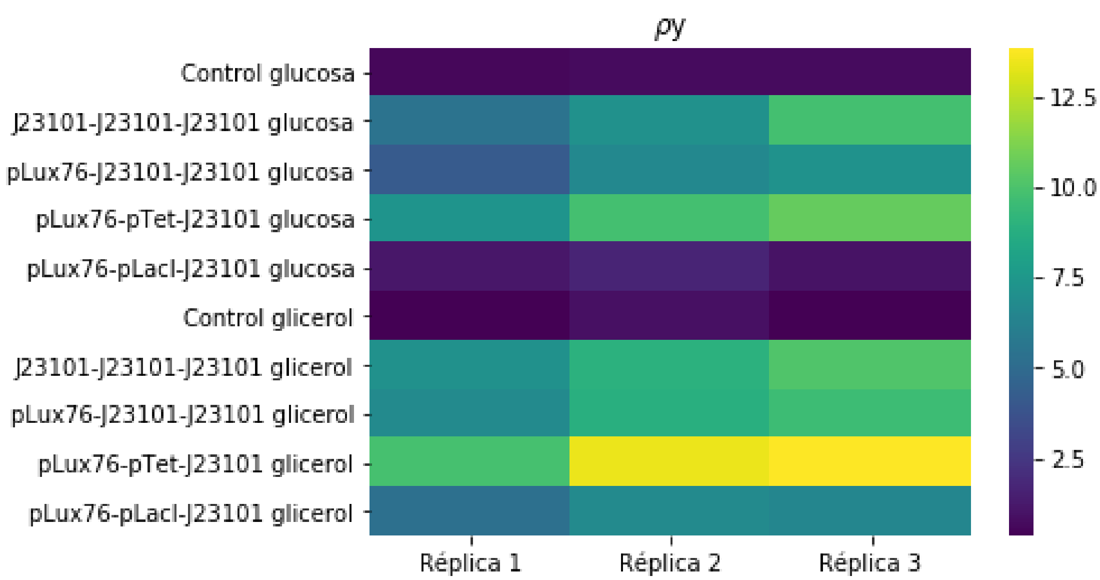
```
ylabel=['Control','J23101-J23101-J23101','pLux76-J23101-J23101','pLux76-pTet-J23101','p

plt.figure()
plt.title(r'$\rho$r Glicerol 0,2%')
sns.heatmap(ro_rfpgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.savefig(filename='rho-r glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_i


plt.figure()
plt.title(r'$\rho$y Glicerol 0,2%')
sns.heatmap(ro_yfpgli,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)
plt.savefig(filename='rho-y glicerol.png', dpi=300, facecolor='w', edgecolor='w',bbox_i
```






```
In [96]: 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','J23101-J23101-J23101 glucosa','pLux76-J23101-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[96]: <matplotlib.axes._subplots.AxesSubplot at 0x1abacebd438>
```

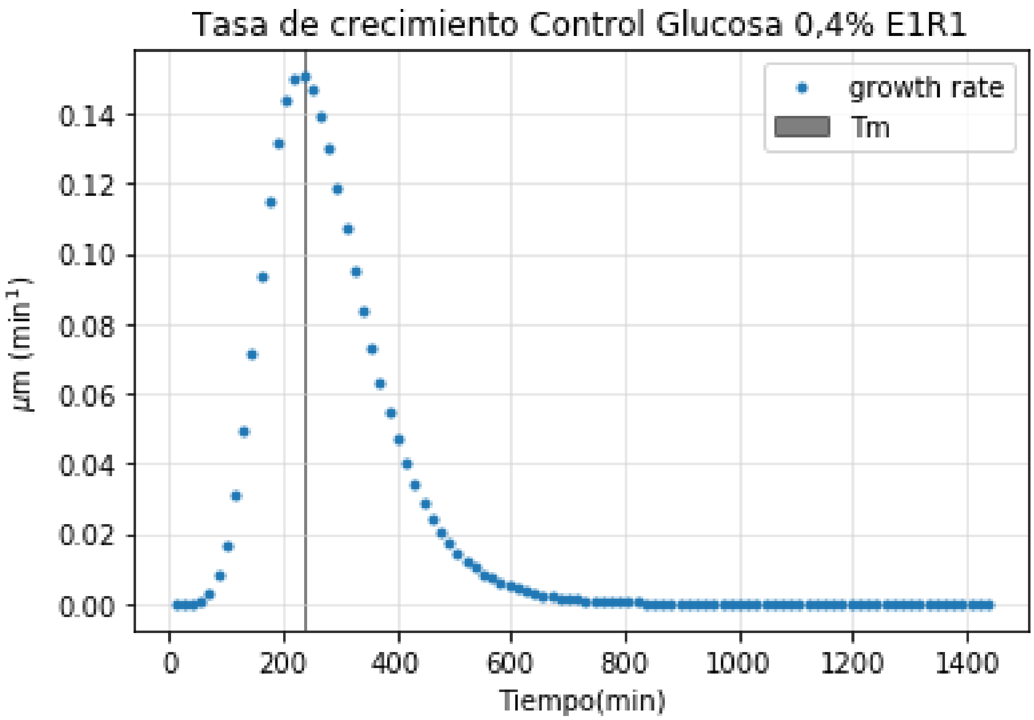





```
In [97]: #tasa de crecimiento
         ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(l1-tt))+1))))
         #Con diff
         dy1=(np.diff(ye1))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E1R1')
         plt.xlabel('Tiempo(min)')
```

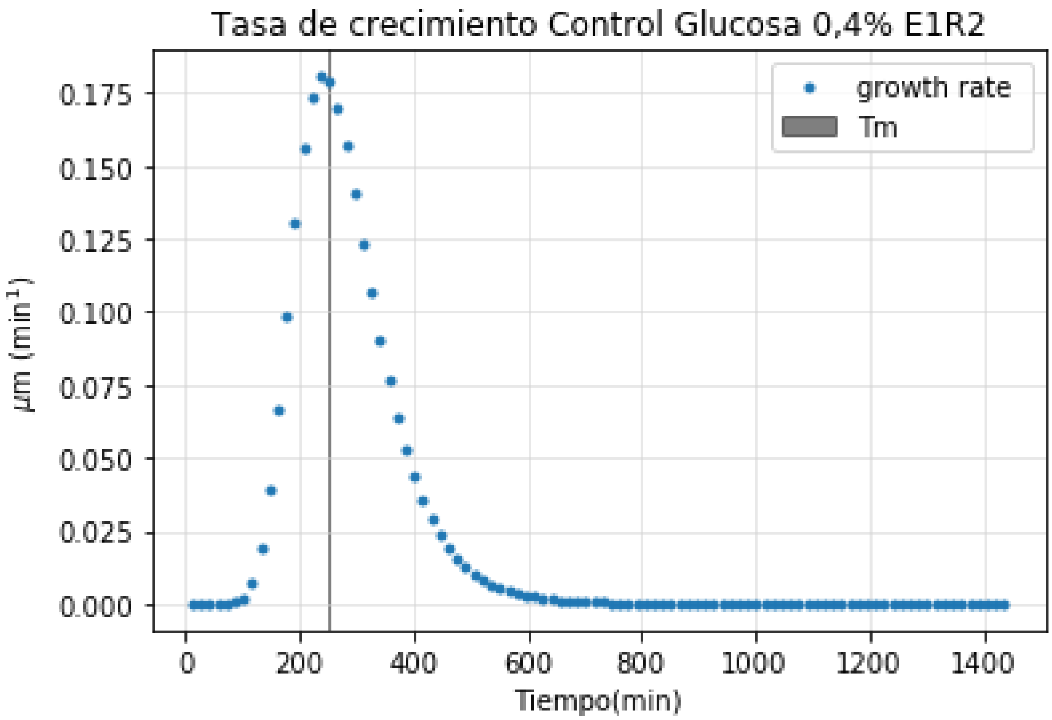
```
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy1,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[97]: <matplotlib.legend.Legend at 0x1ababc4fc18>

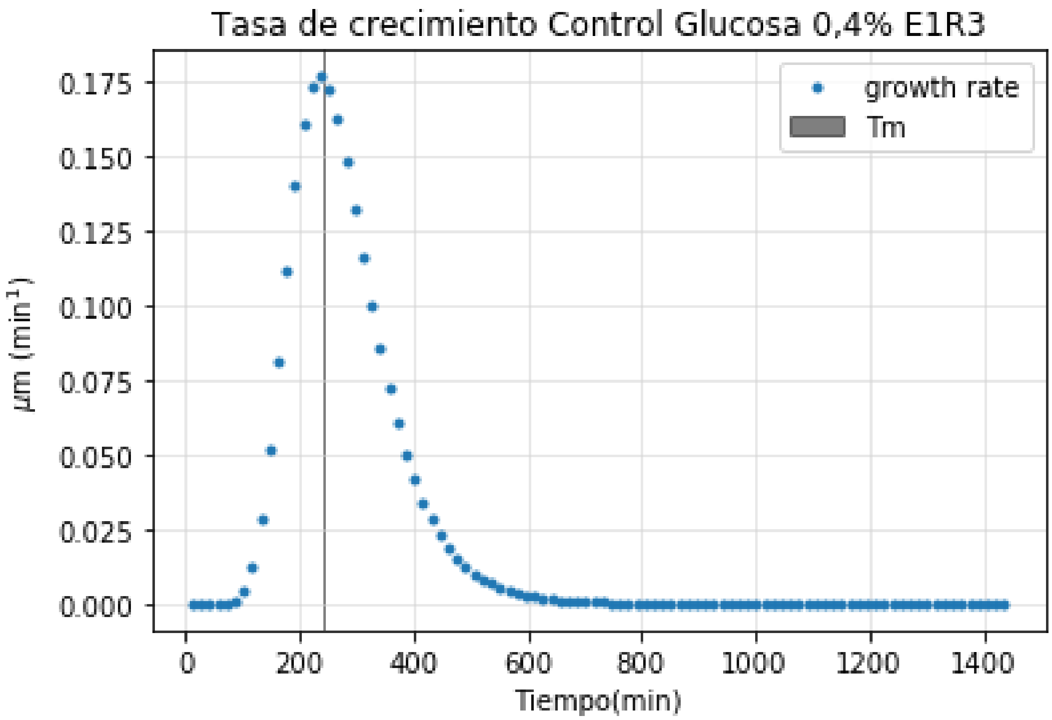


In [98]:
```
#tasa de crecimiento

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% E1R2')
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')
```

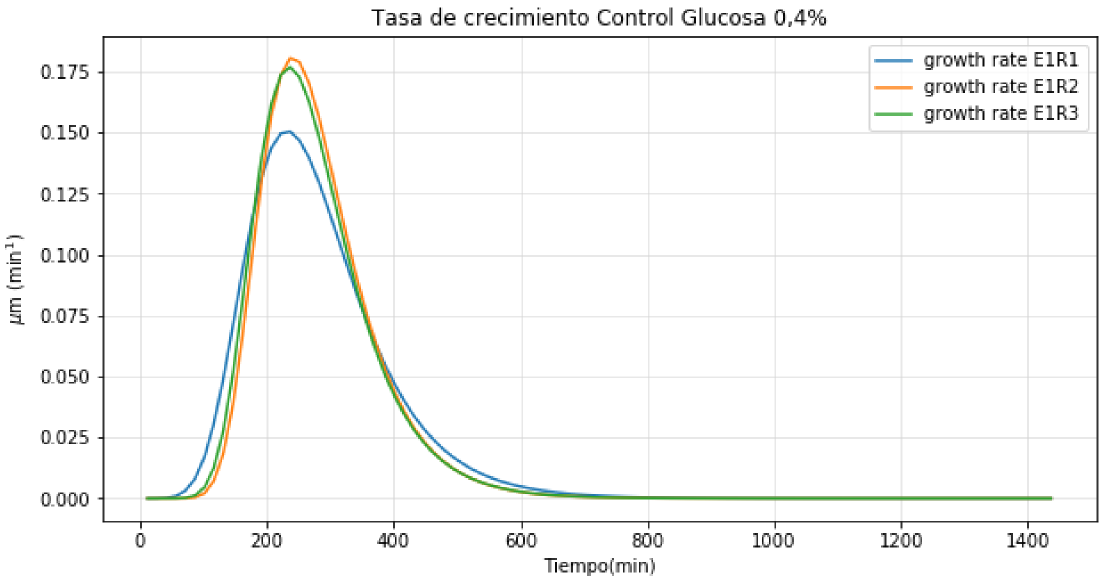
Out[98]: <matplotlib.legend.Legend at 0x1abac530550>



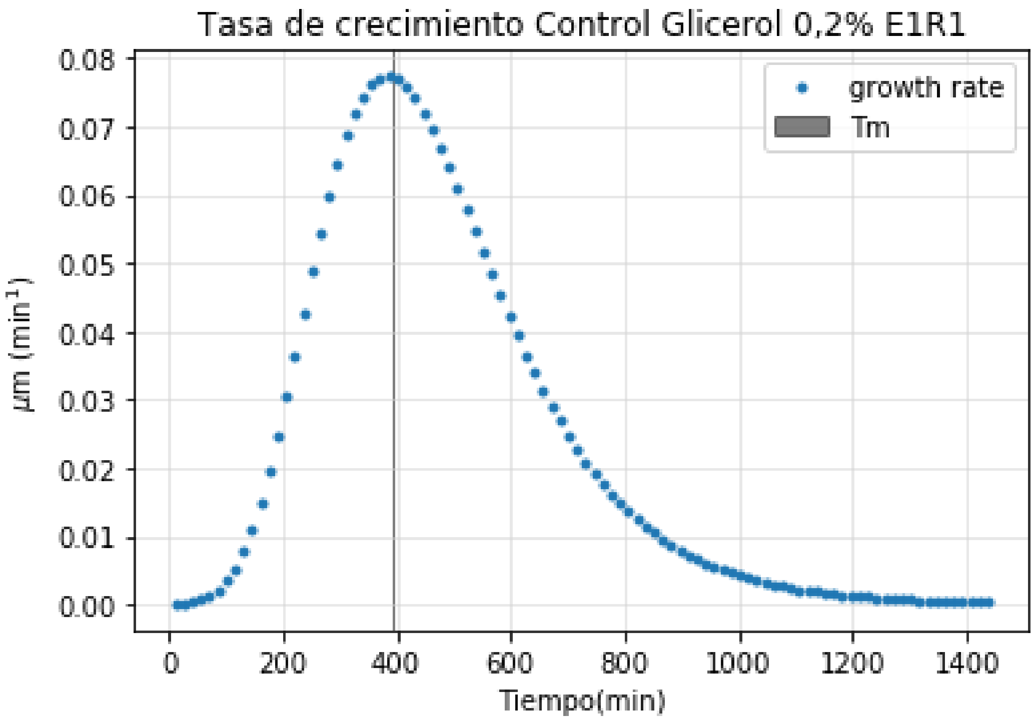

```
In [99]: #tasa de crecimiento
         ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(l3-tt))+1))))
         #Con diff
         dy3=(np.diff(ye3))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E1R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[99]: <matplotlib.legend.Legend at 0x1abaca51e10>
```



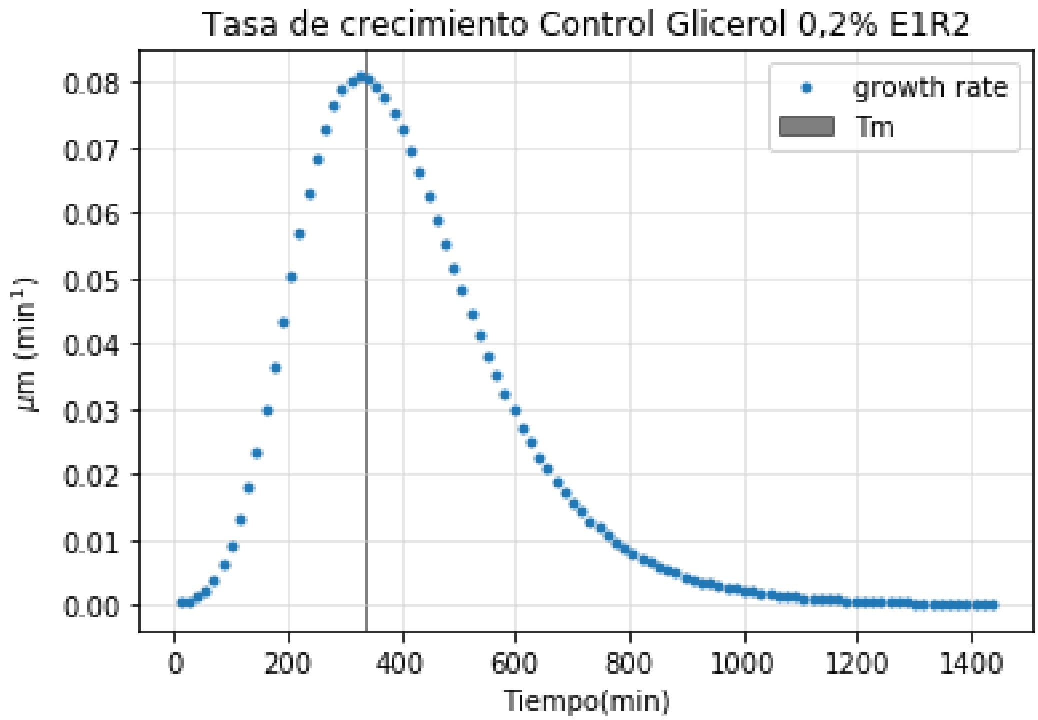

```
In [100]: #Tasas control réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,label='growth rate E1R1')
          plt.plot(tt[:-1],dy2,label='growth rate E1R2')
          plt.plot(tt[:-1],dy3,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[100]: <matplotlib.legend.Legend at 0x1abacfabc50>
```




```
In [101]: #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% E1R1')
          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[101]: <matplotlib.legend.Legend at 0x1abac9f7a20>
```

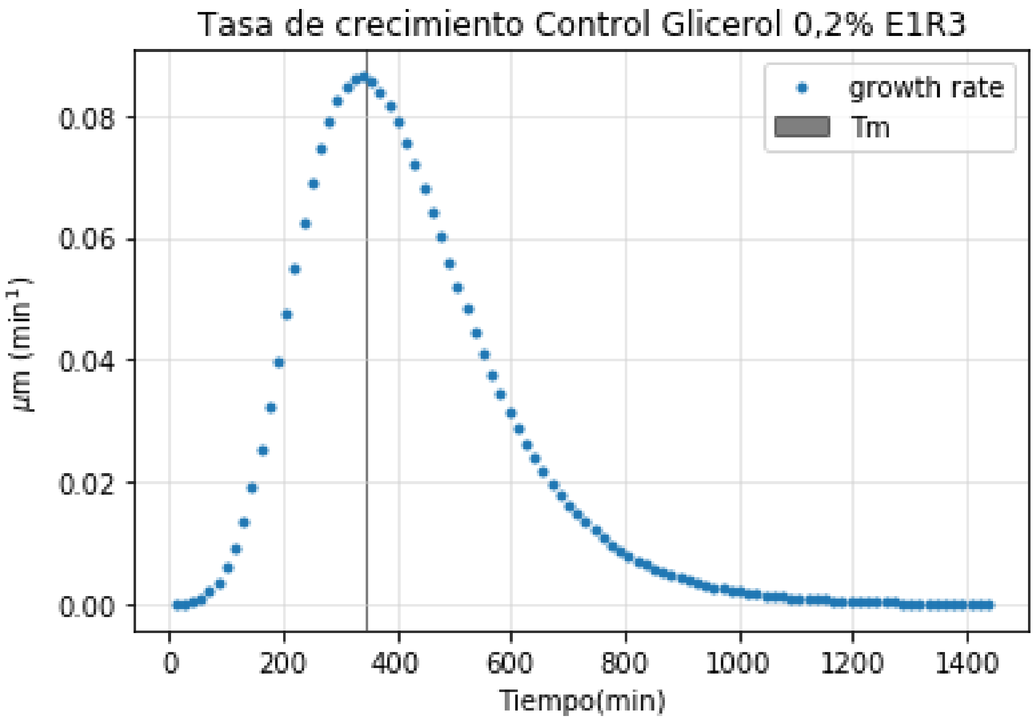



```
In [102]: #tasa de crecimiento
          ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(l5-tt))+1))))
          #Con diff
          dy5=(np.diff(ye5))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E1R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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[102]: <matplotlib.legend.Legend at 0x1ababf4fa90>
```

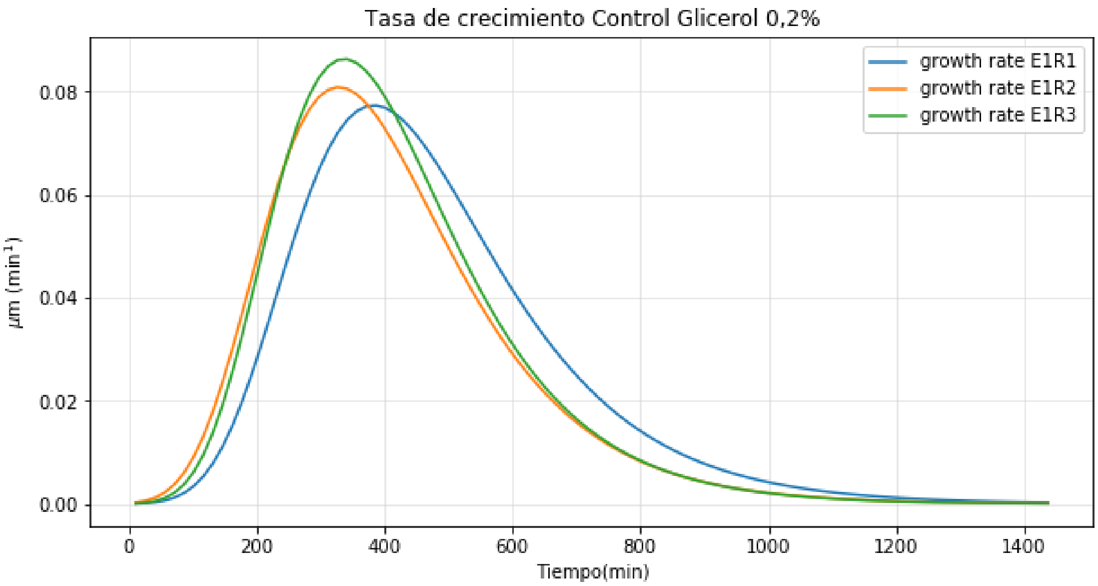


In [103]:
```
#tasa de crecimiento
ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(l6-tt))+1))))
#Con diff
dy6=(np.diff(ye6))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm6,tm6, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy6,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

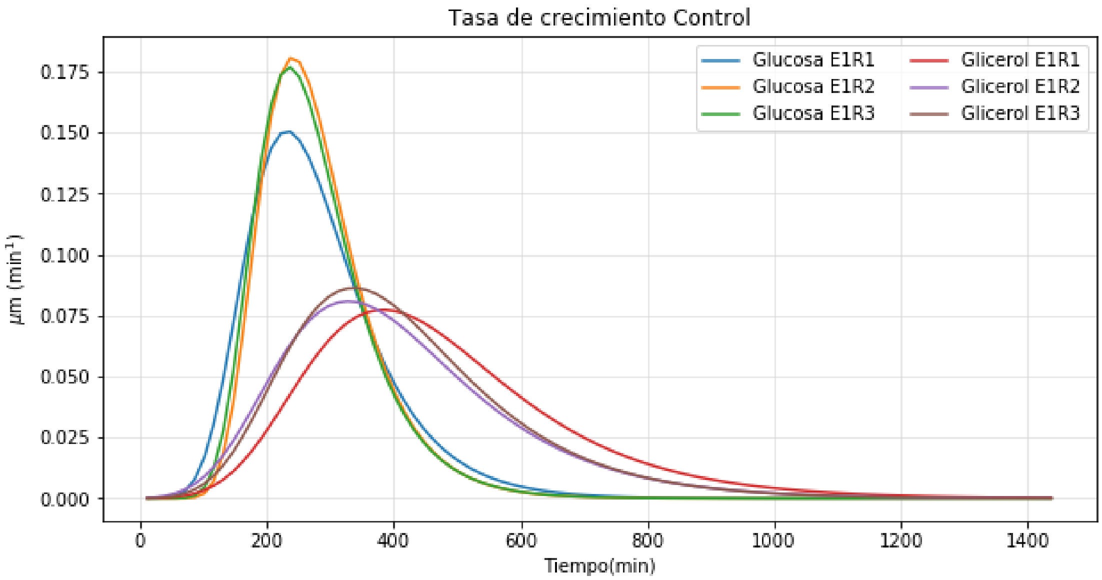
Out[103]: <matplotlib.legend.Legend at 0x1abac4b2ac8>



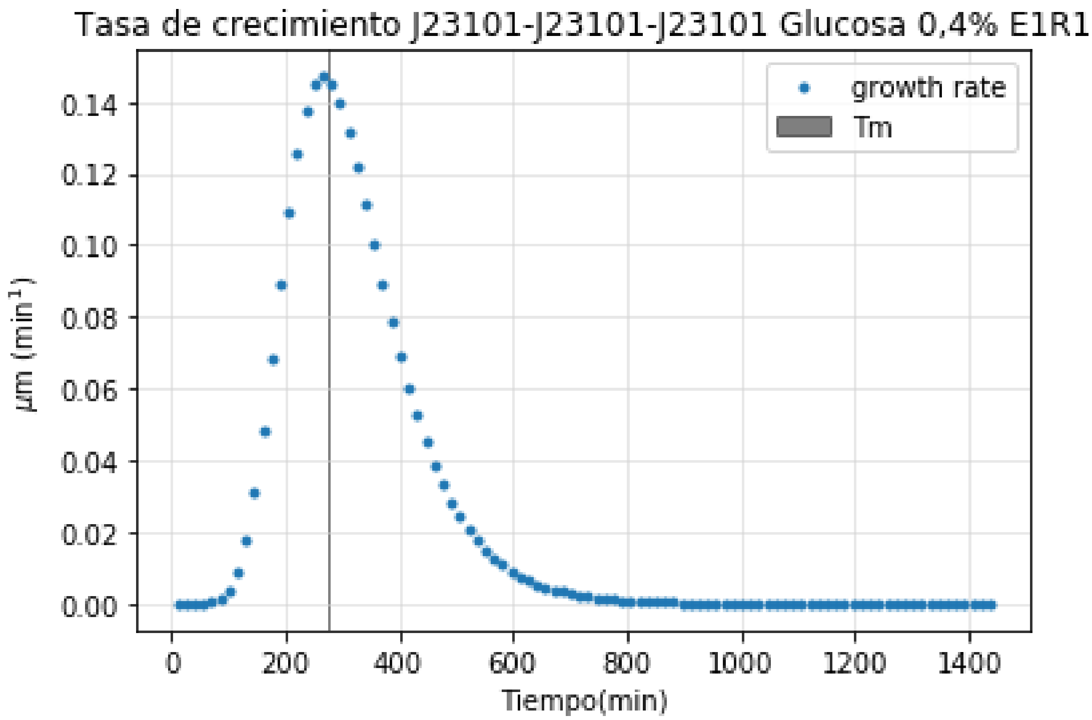

```
In [104]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy4,label='growth rate E1R1')
          plt.plot(tt[:-1],dy5,label='growth rate E1R2')
          plt.plot(tt[:-1],dy6,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[104]: <matplotlib.legend.Legend at 0x1ababa306a0>
```



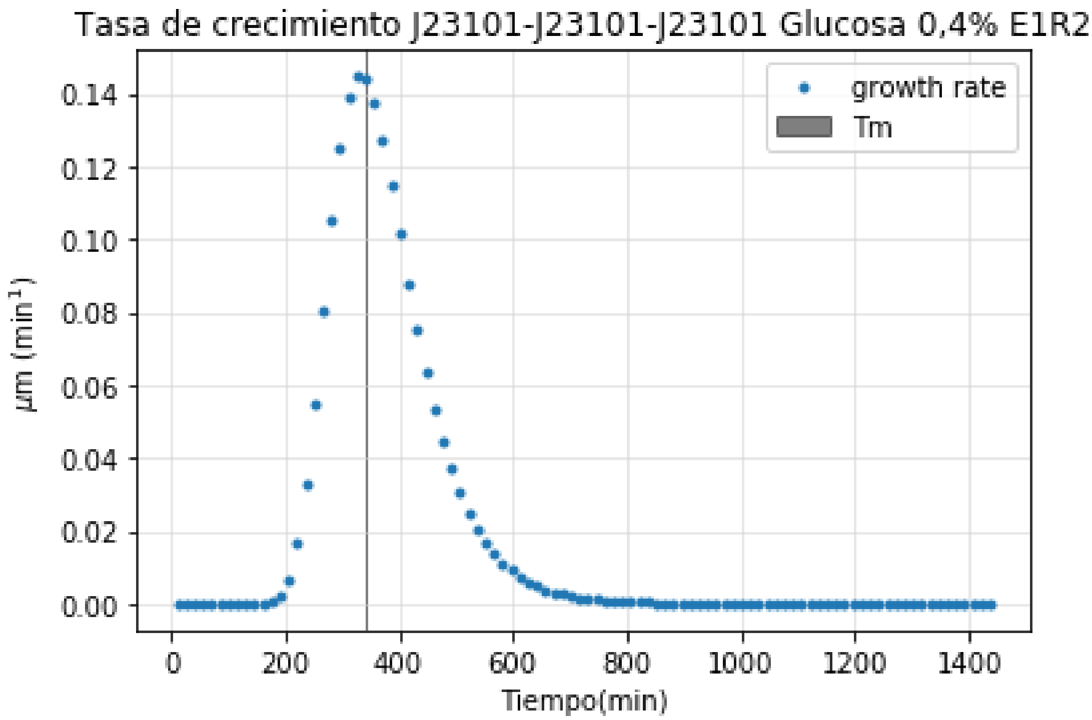

```
In [105]: #Tasas control réplicas controles
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,label='Glucosa E1R1')
          plt.plot(tt[:-1],dy2,label='Glucosa E1R2')
          plt.plot(tt[:-1],dy3,label='Glucosa E1R3')
          plt.plot(tt[:-1],dy4,label='Glicerol E1R1')
          plt.plot(tt[:-1],dy5,label='Glicerol E1R2')
          plt.plot(tt[:-1],dy6,label='Glicerol E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[105]: <matplotlib.legend.Legend at 0x1ababb11da0>
```



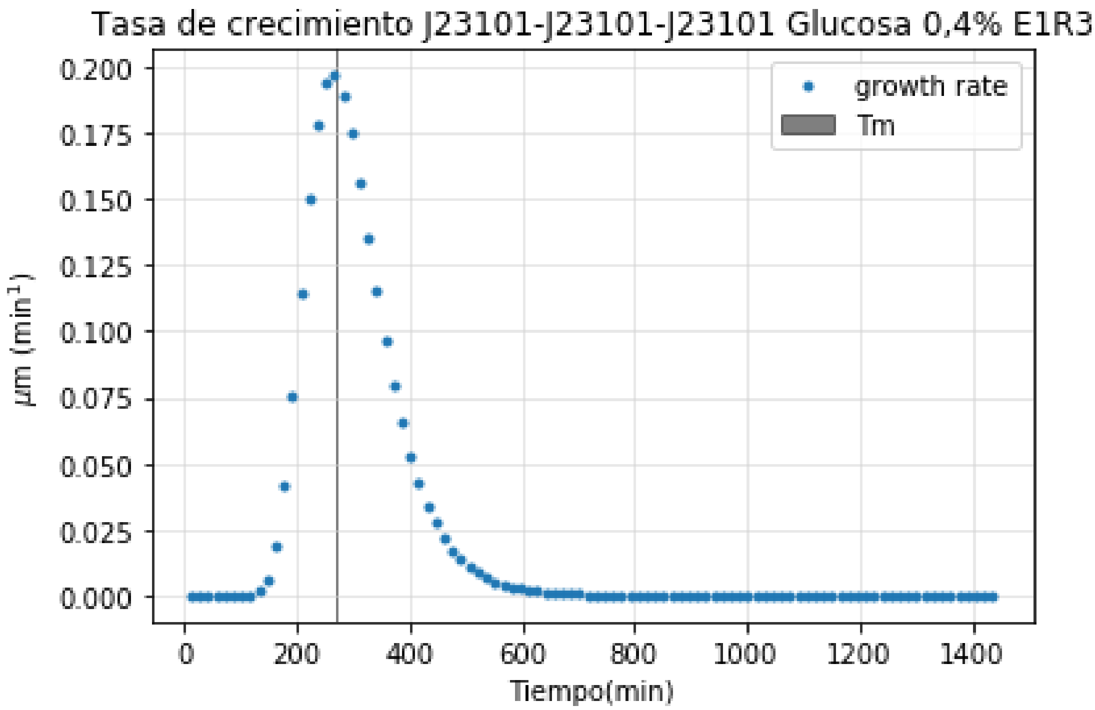

```
In [106]: #tasa de crecimiento
          ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(l7-tt))+1))))
          #Con diff
          dy7=(np.diff(ye7))
          plt.figure()
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glucosa 0,4% E1R1')
          plt.xlabel('Tiempo(min)')
          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[106]: <matplotlib.legend.Legend at 0x1abac1aaf60>
```



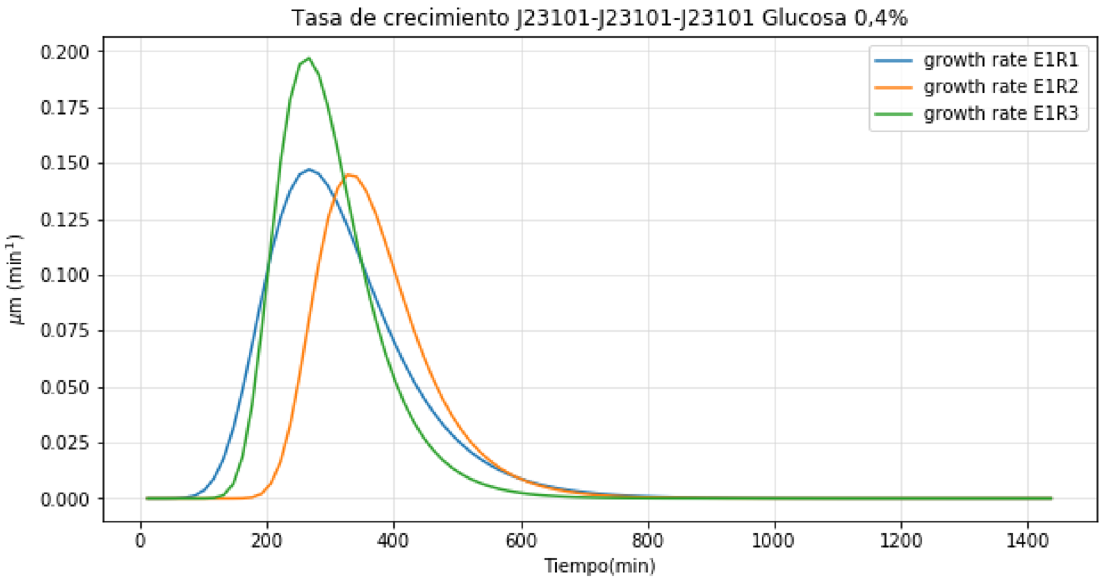

```
In [107]: #tasa de crecimiento
          ye8=((A8*np.exp(-np.exp((((um8*np.exp(1))/A8)*(l8-tt))+1))))
          #Con diff
          dy8=(np.diff(ye8))
          plt.figure()
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glucosa 0,4% E1R2')
          plt.xlabel('Tiempo(min)')
          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[107]: <matplotlib.legend.Legend at 0x1abad09d080>
```



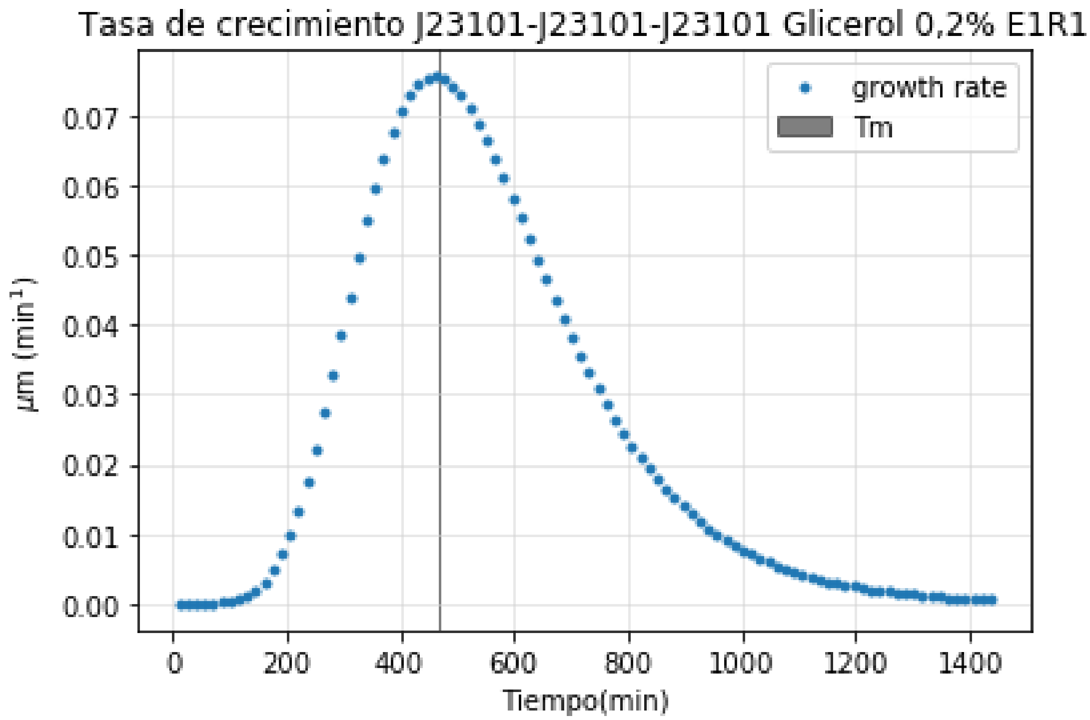

```
In [108]: #tasa de crecimiento
          ye9=((A9*np.exp(-np.exp((((um9*np.exp(1))/A9)*(l9-tt))+1))))
          #Con diff
          dy9=(np.diff(ye9))
          plt.figure()
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glucosa 0,4% E1R3')
          plt.xlabel('Tiempo(min)')
          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[108]: <matplotlib.legend.Legend at 0x1aba4fa1eb8>
```




```
In [109]: #Tasas J23101-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='growth rate E1R1')
          plt.plot(tt[:-1],dy8,label='growth rate E1R2')
          plt.plot(tt[:-1],dy9,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[109]: <matplotlib.legend.Legend at 0x1abacf98dd8>
```

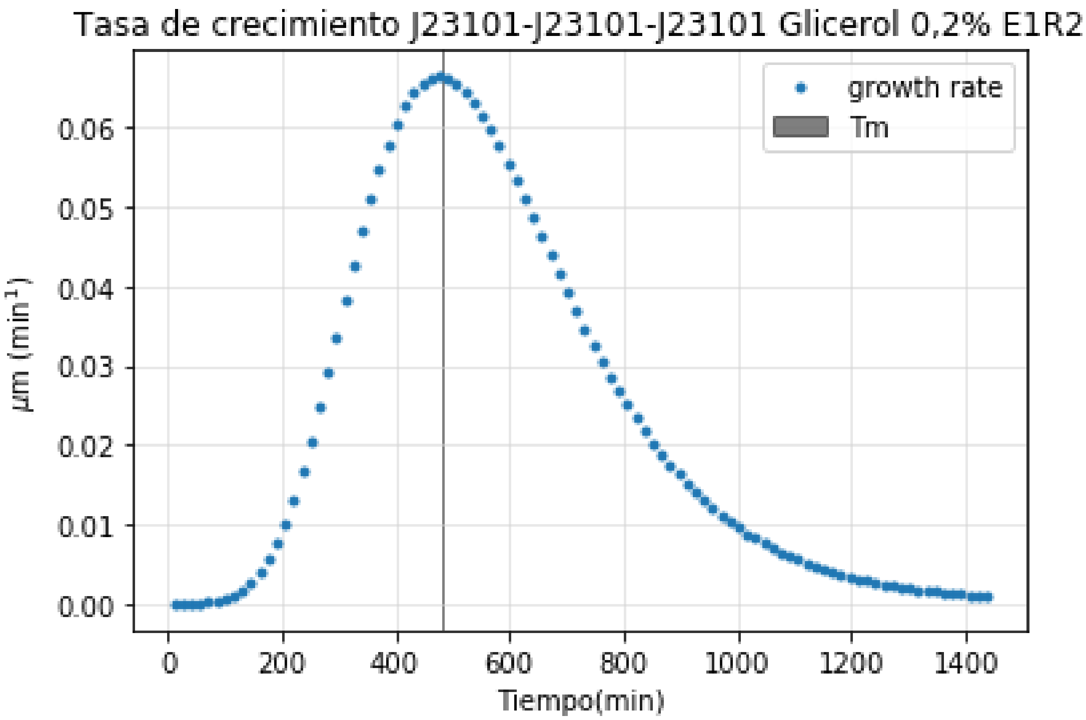



```
In [110]: #tasa de crecimiento
          ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(l10-tt))+1))))
          #Con diff
          dy10=(np.diff(ye10))
          plt.figure()
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glicerol 0,2% E1R1')
          plt.xlabel('Tiempo(min)')
          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[110]: <matplotlib.legend.Legend at 0x1abac3d39b0>
```

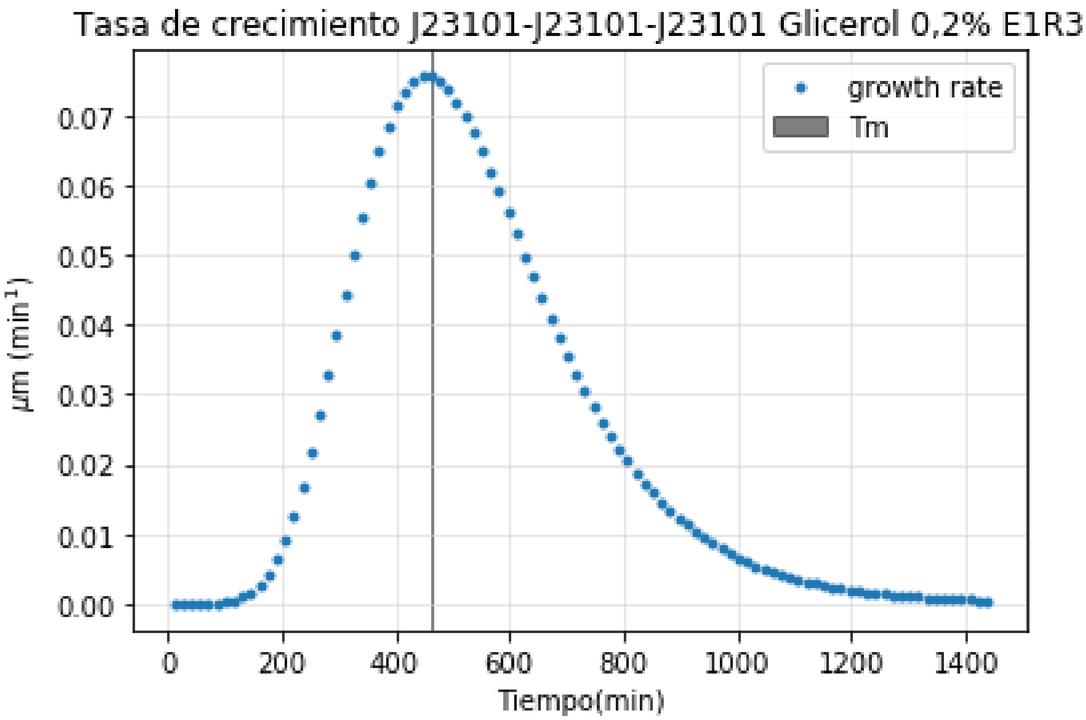


In [111]:
```
#tasa de crecimiento
ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(l11-tt))+1))))
#Con diff
dy11=(np.diff(ye11))
plt.figure()
plt.title('Tasa de crecimiento J23101-J23101-J23101 Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm11,tm11, color='k', 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[111]:
```
<matplotlib.legend.Legend at 0x1ababc0a780>
```

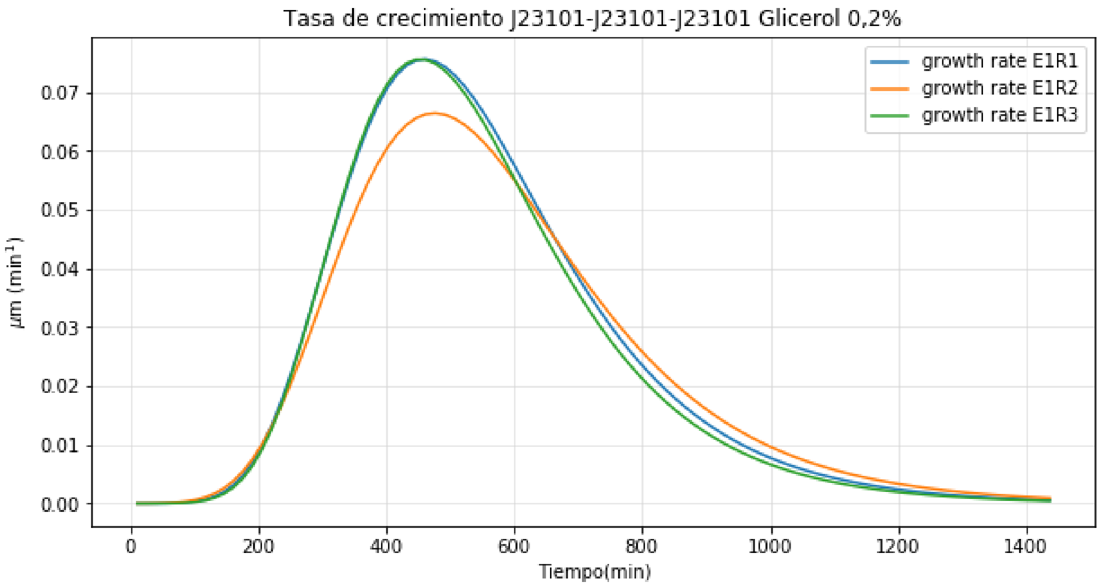



```
In [112]: #tasa de crecimiento
          ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(l12-tt))+1))))
          #Con diff
          dy12=(np.diff(ye12))
          plt.figure()
          plt.title('Tasa de crecimiento J23101-J23101-J23101 Glicerol 0,2% E1R3')
          plt.xlabel('Tiempo(min)')
          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[112]: <matplotlib.legend.Legend at 0x1abac7dfe48>
```

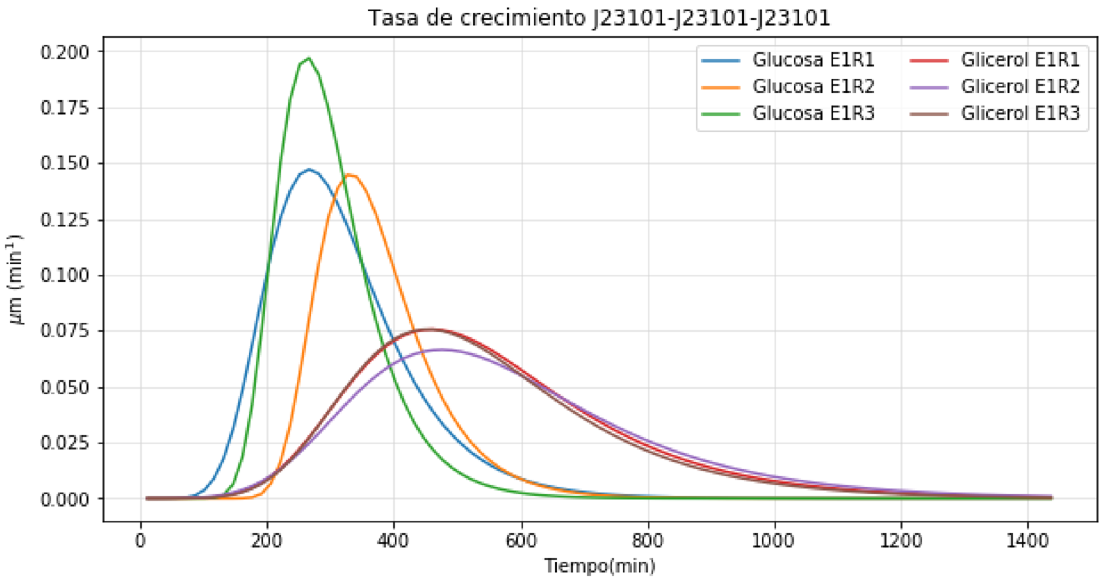


In [113]:
```
#Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento J23101-J23101-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy10,label='growth rate E1R1')
plt.plot(tt[:-1],dy11,label='growth rate E1R2')
plt.plot(tt[:-1],dy12,label='growth rate E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

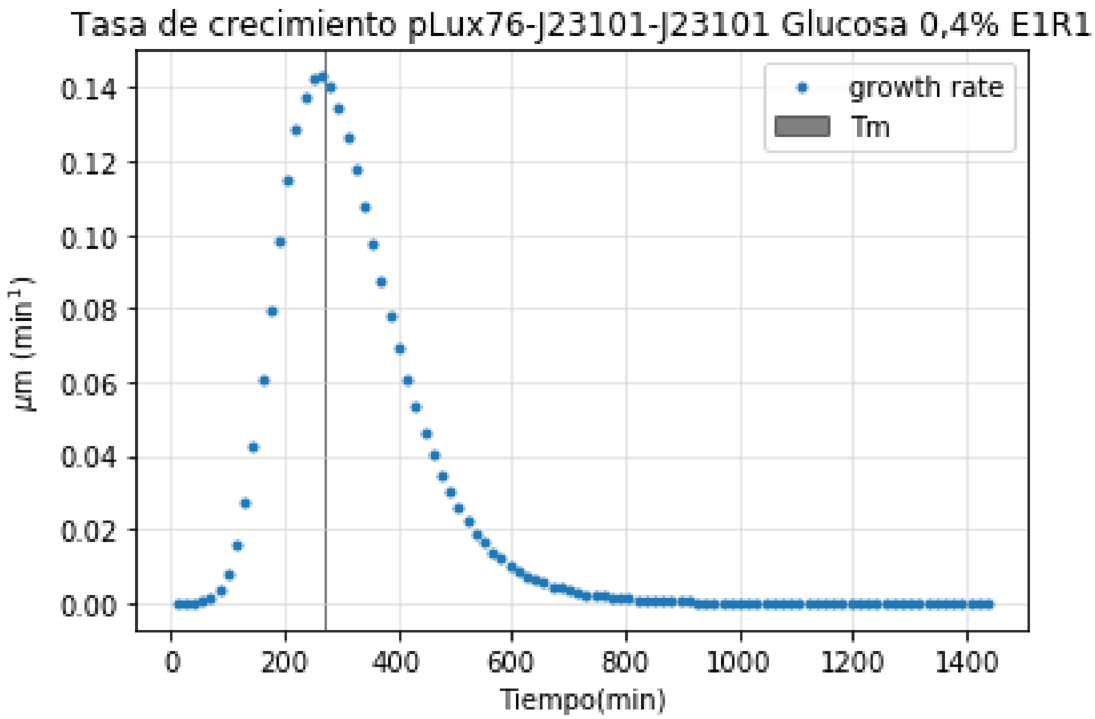
Out[113]:
```
<matplotlib.legend.Legend at 0x1abacb75f28>
```



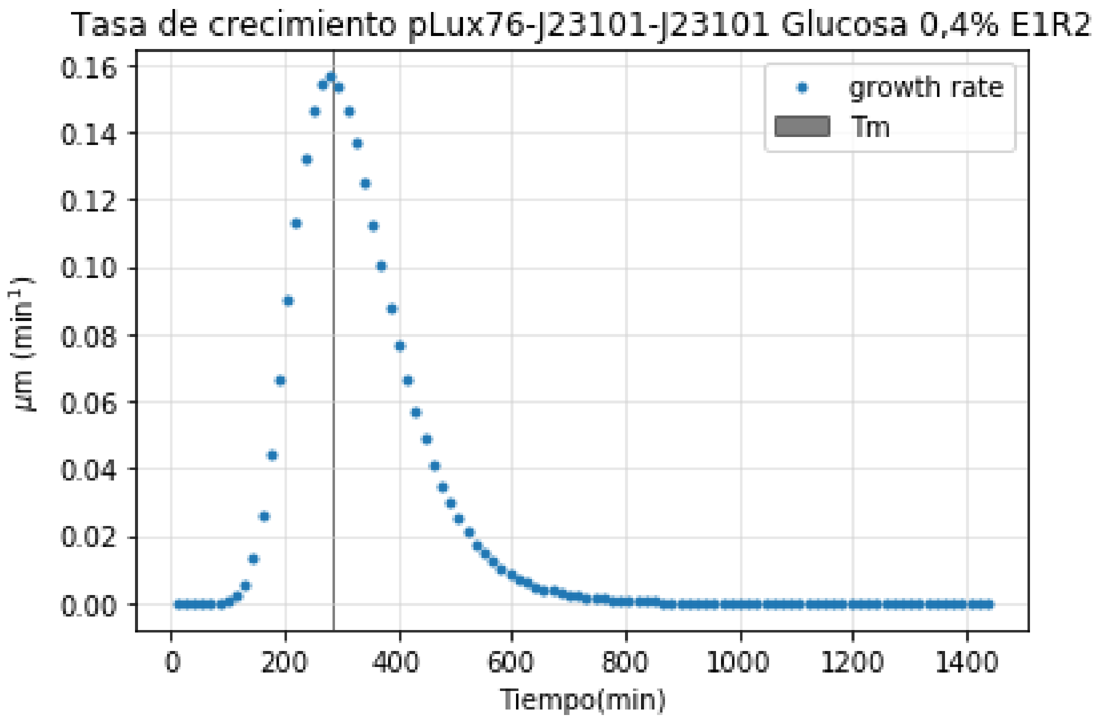

```
In [114]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23101-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='Glucosa E1R1')
          plt.plot(tt[:-1],dy8,label='Glucosa E1R2')
          plt.plot(tt[:-1],dy9,label='Glucosa E1R3')
          plt.plot(tt[:-1],dy10,label='Glicerol E1R1')
          plt.plot(tt[:-1],dy11,label='Glicerol E1R2')
          plt.plot(tt[:-1],dy12,label='Glicerol E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[114]: <matplotlib.legend.Legend at 0x1abacc64ba8>
```



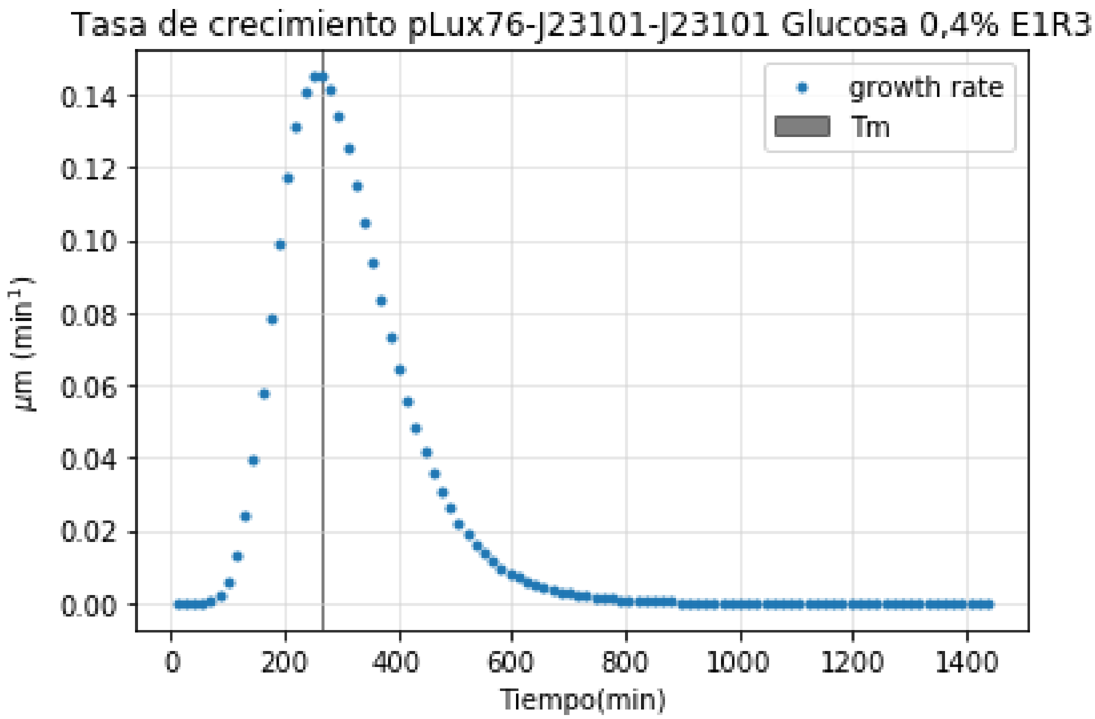

```
In [115]: #tasa de crecimiento
          ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(l13-tt))+1))))
          #Con diff
          dy13=(np.diff(ye13))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E1R1')
          plt.xlabel('Tiempo(min)')
          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[115]: <matplotlib.legend.Legend at 0x1abacd3beb8>
```



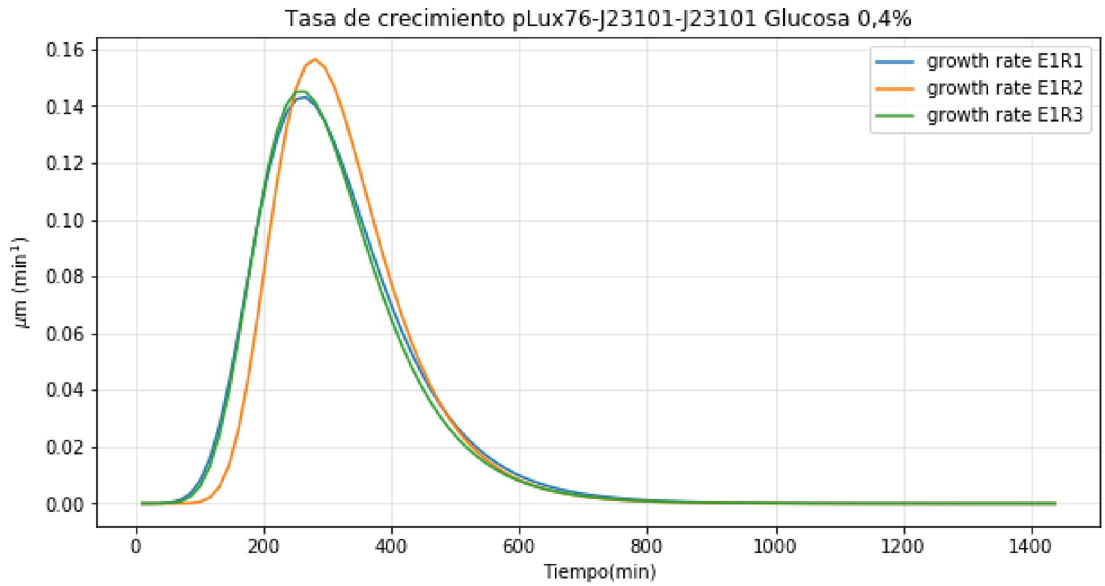

```
In [116]: #tasa de crecimiento
          ye14=((A14*np.exp(-np.exp((((um14*np.exp(1))/A14)*(l14-tt))+1))))
          #Con diff
          dy14=(np.diff(ye14))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E1R2')
          plt.xlabel('Tiempo(min)')
          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[116]: <matplotlib.legend.Legend at 0x1abace1b940>
```




```
In [117]: #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-J23101-J23101 Glucosa 0,4% E1R3')
          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[117]: <matplotlib.legend.Legend at 0x1abad1a4eb8>
```

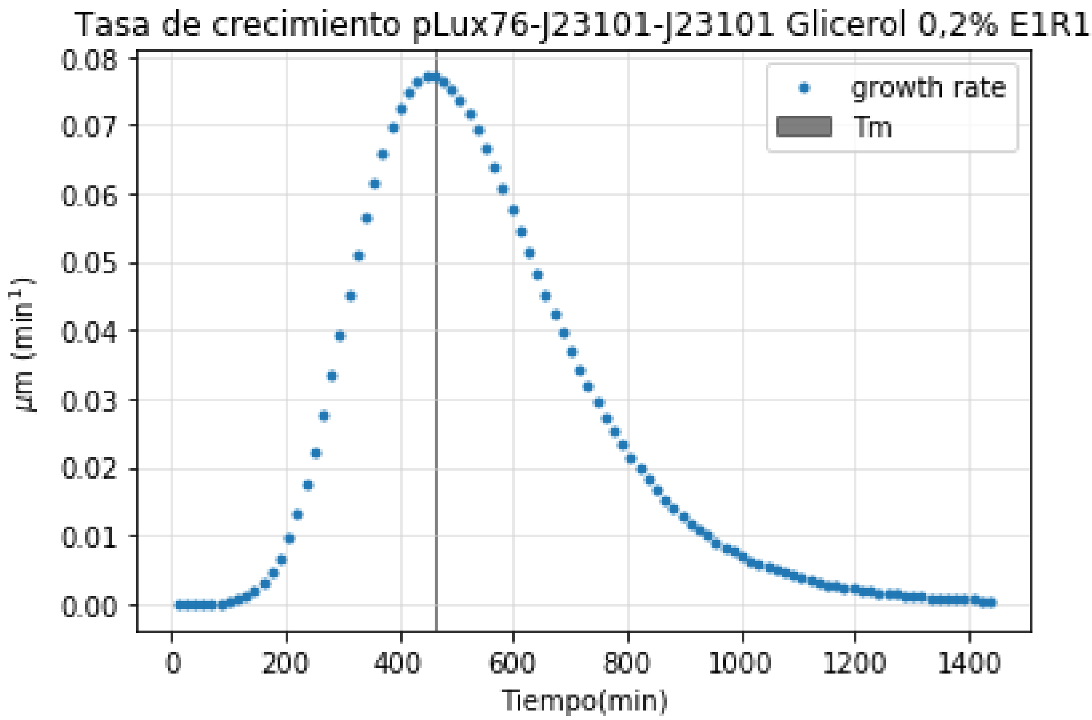



```
In [118]: #Tasas plux76-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='growth rate E1R1')
          plt.plot(tt[:-1],dy14,label='growth rate E1R2')
          plt.plot(tt[:-1],dy15,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[118]: <matplotlib.legend.Legend at 0x1abad27ba90>
```



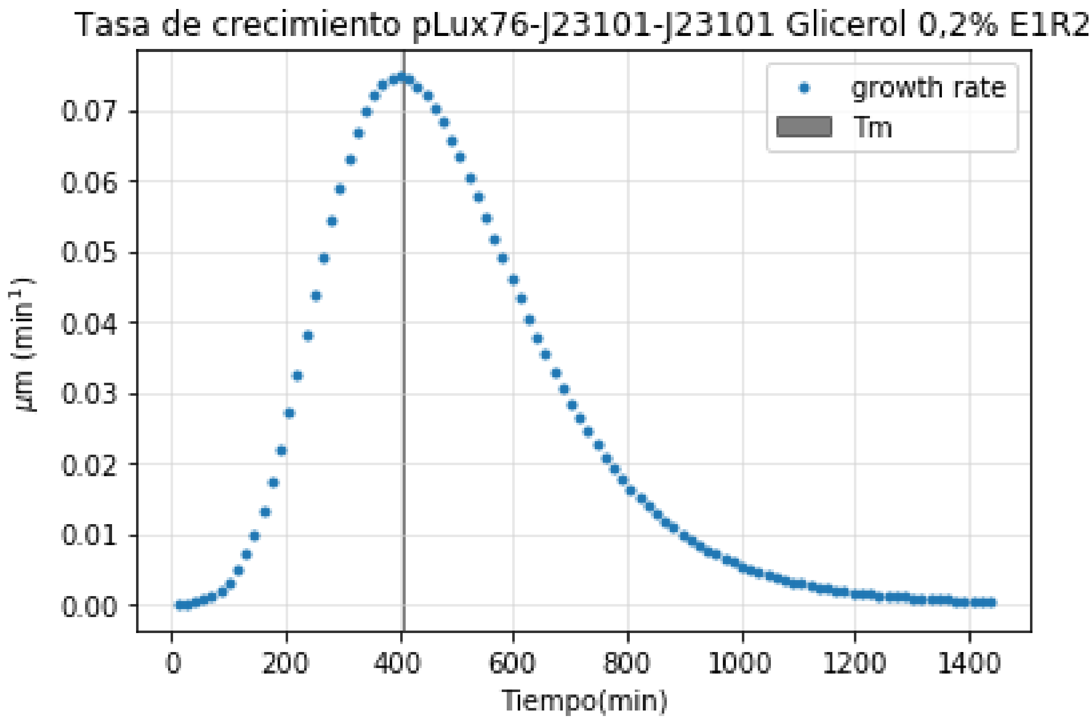
In [119]:
```
#tasa de crecimiento
ye16=((A16*np.exp(-np.exp((((um16*np.exp(1))/A16)*(l16-tt))+1))))
#Con diff
dy16=(np.diff(ye16))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm16,tm16, color='k', 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[119]: <matplotlib.legend.Legend at 0x1abad230ef0>

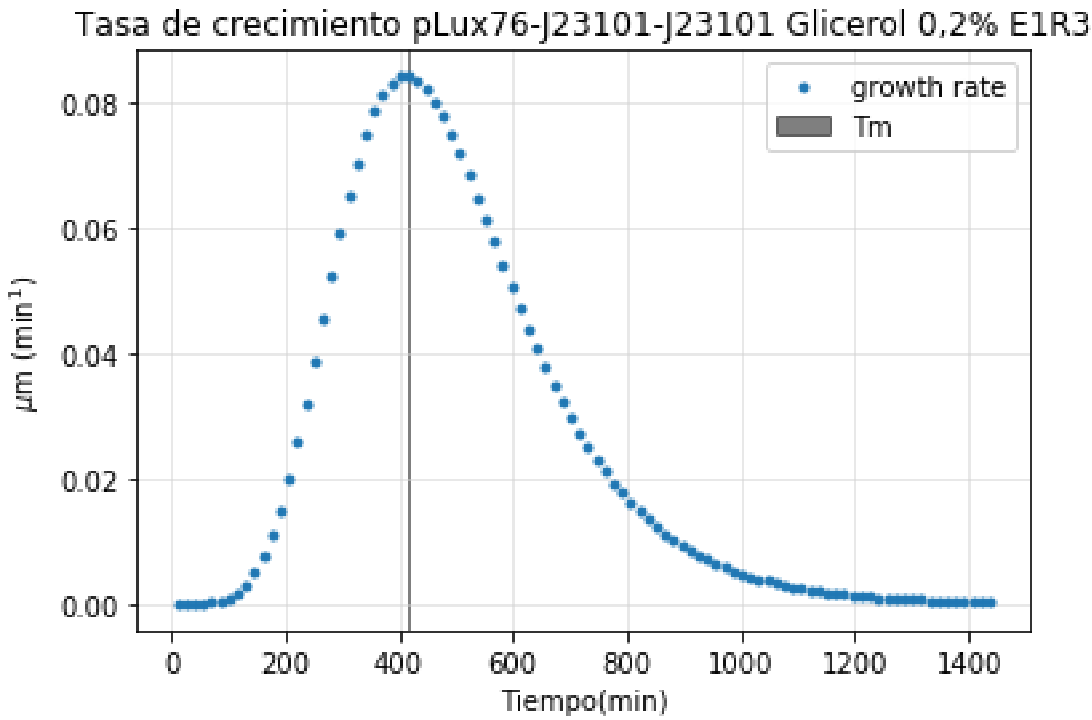


In [120]:
```
#tasa de crecimiento
ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(l17-tt))+1))))
#Con diff
dy17=(np.diff(ye17))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E1R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm17,tm17, color='k', 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')
```

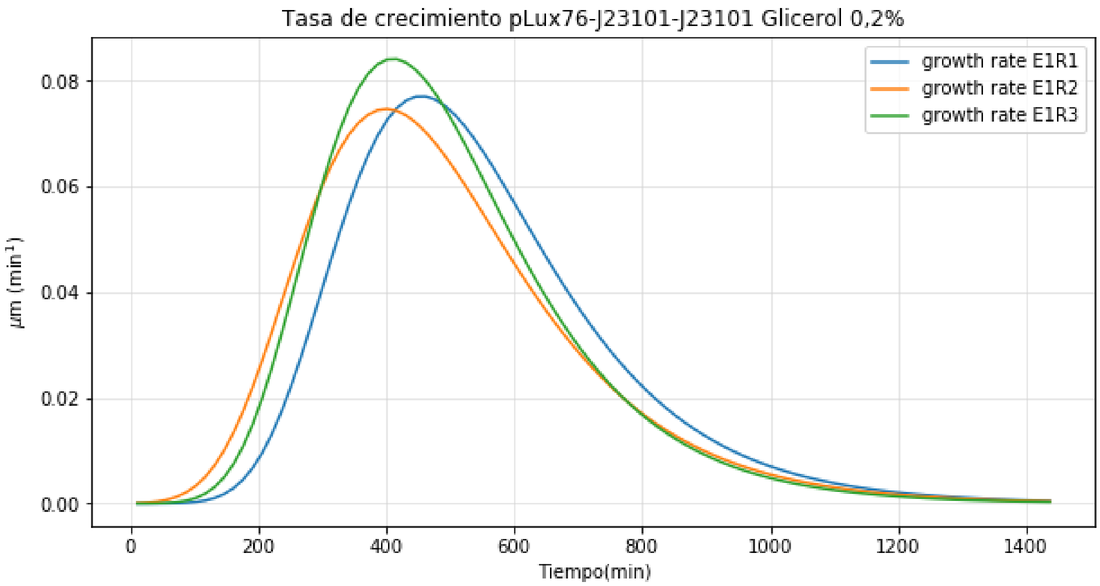
Out[120]:
```
<matplotlib.legend.Legend at 0x1abad427a58>
```




```
In [121]: #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-J23101-J23101 Glicerol 0,2% E1R3')
          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[121]: <matplotlib.legend.Legend at 0x1ababed5550>
```

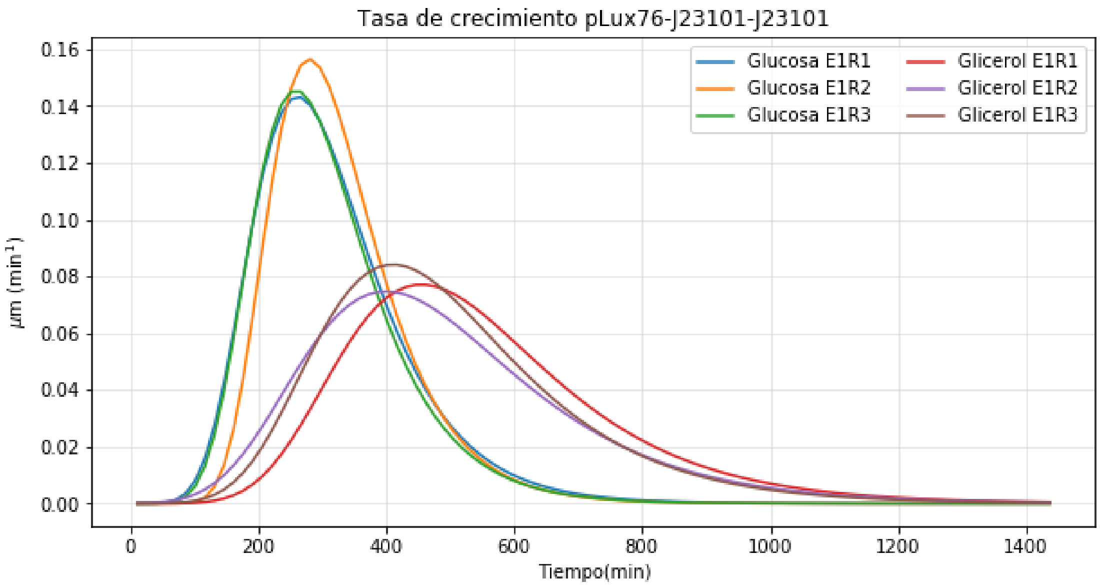



```
In [122]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy16,label='growth rate E1R1')
          plt.plot(tt[:-1],dy17,label='growth rate E1R2')
          plt.plot(tt[:-1],dy18,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[122]: <matplotlib.legend.Legend at 0x1abad4b4ba8>
```

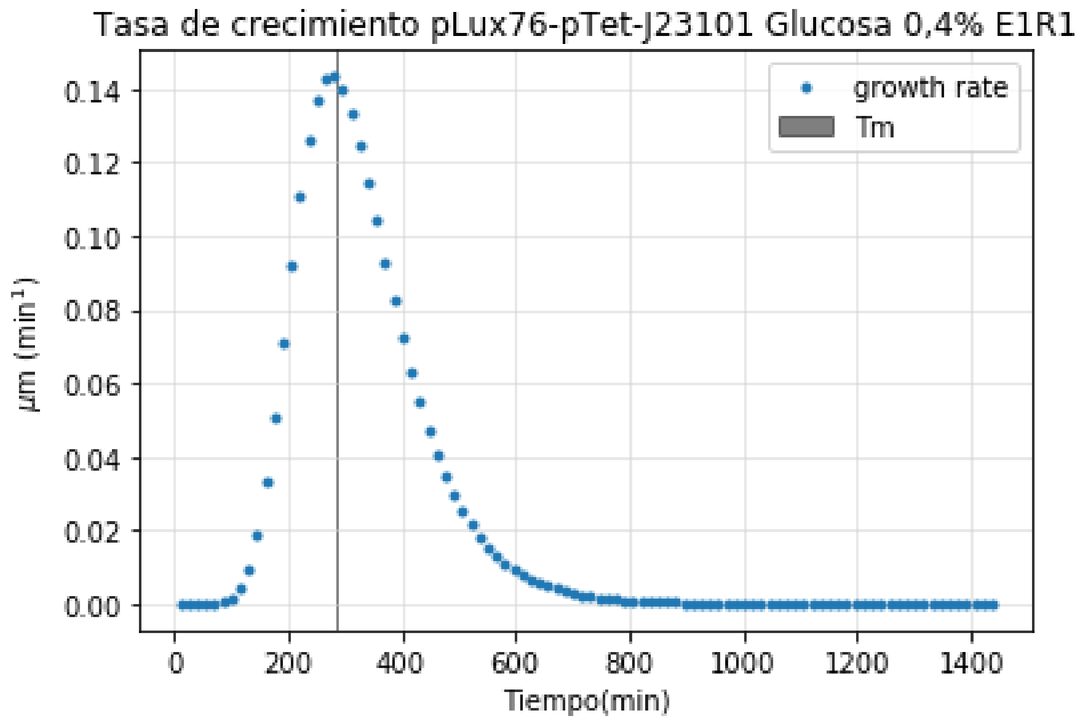


In [123]:
```python
#Tasas control réplicas
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-J23101-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy13,label='Glucosa E1R1')
plt.plot(tt[:-1],dy14,label='Glucosa E1R2')
plt.plot(tt[:-1],dy15,label='Glucosa E1R3')
plt.plot(tt[:-1],dy16,label='Glicerol E1R1')
plt.plot(tt[:-1],dy17,label='Glicerol E1R2')
plt.plot(tt[:-1],dy18,label='Glicerol E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

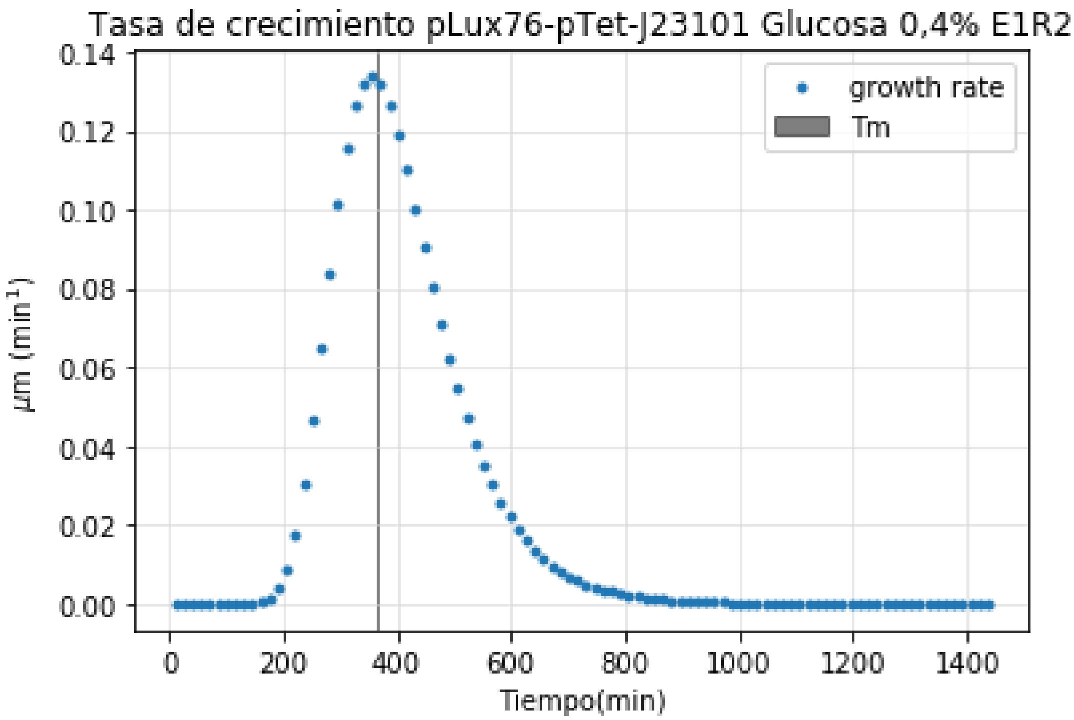
Out[123]:
```
<matplotlib.legend.Legend at 0x1abad577e48>
```



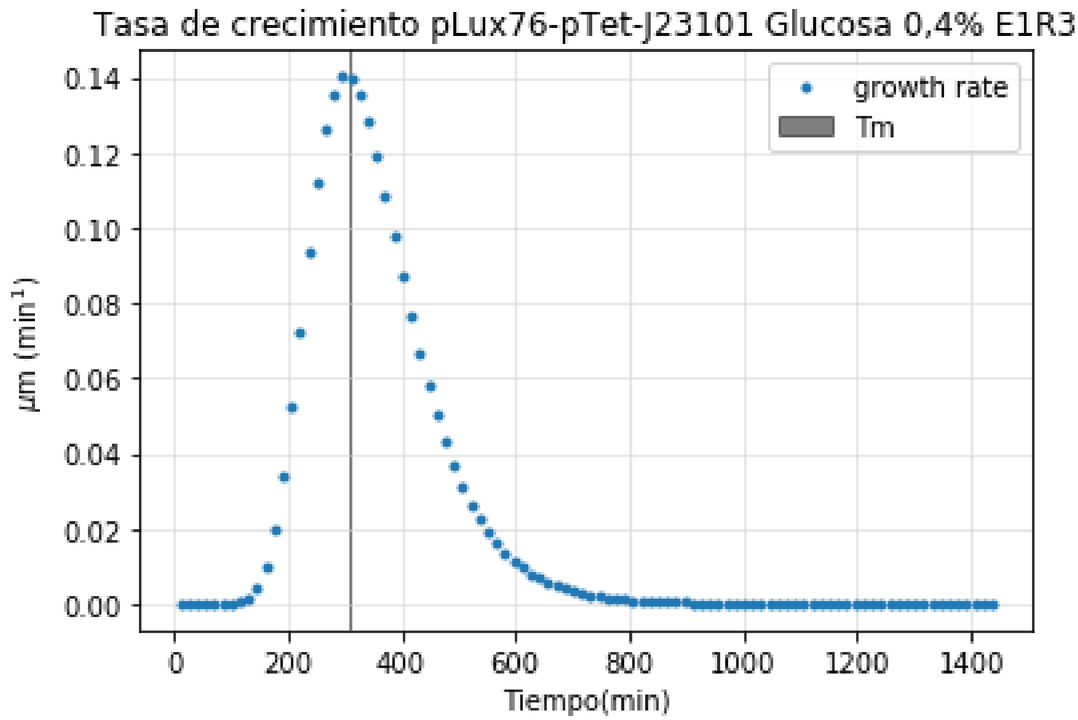

```
In [124]: #tasa de crecimiento
          ye19=((A19*np.exp(-np.exp((((um19*np.exp(1))/A19)*(l19-tt))+1))))
          #Con diff
          dy19=(np.diff(ye19))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glucosa 0,4% E1R1')
          plt.xlabel('Tiempo(min)')
          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[124]: <matplotlib.legend.Legend at 0x1abae667940>
```



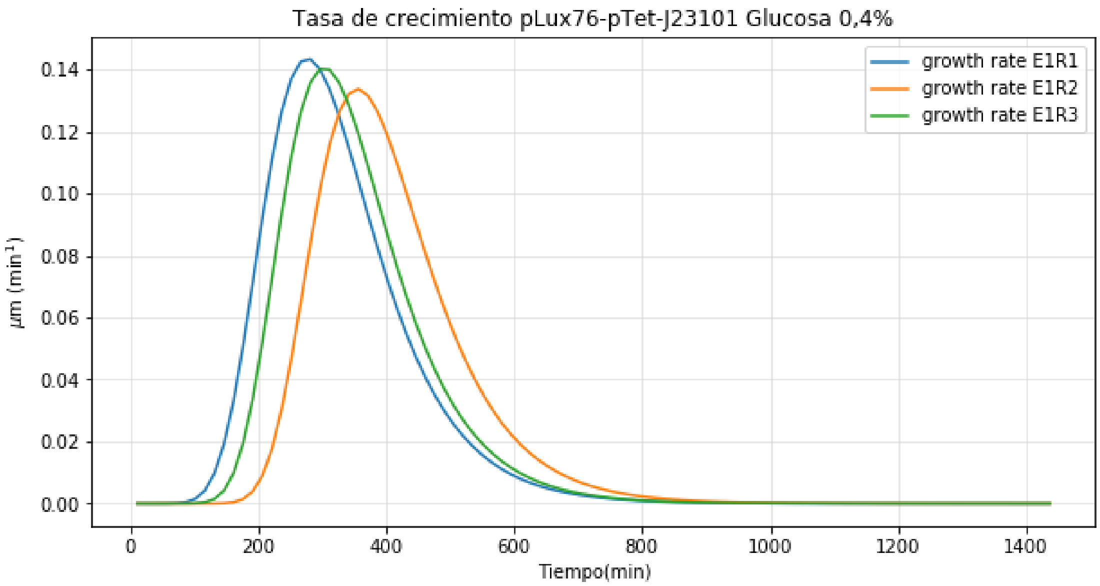

```
In [125]: #tasa de crecimiento
          ye20=((A20*np.exp(-np.exp((((um20*np.exp(1))/A20)*(l20-tt))+1))))
          #Con diff
          dy20=(np.diff(ye20))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glucosa 0,4% E1R2')
          plt.xlabel('Tiempo(min)')
          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[125]: <matplotlib.legend.Legend at 0x1abae73fc50>
```



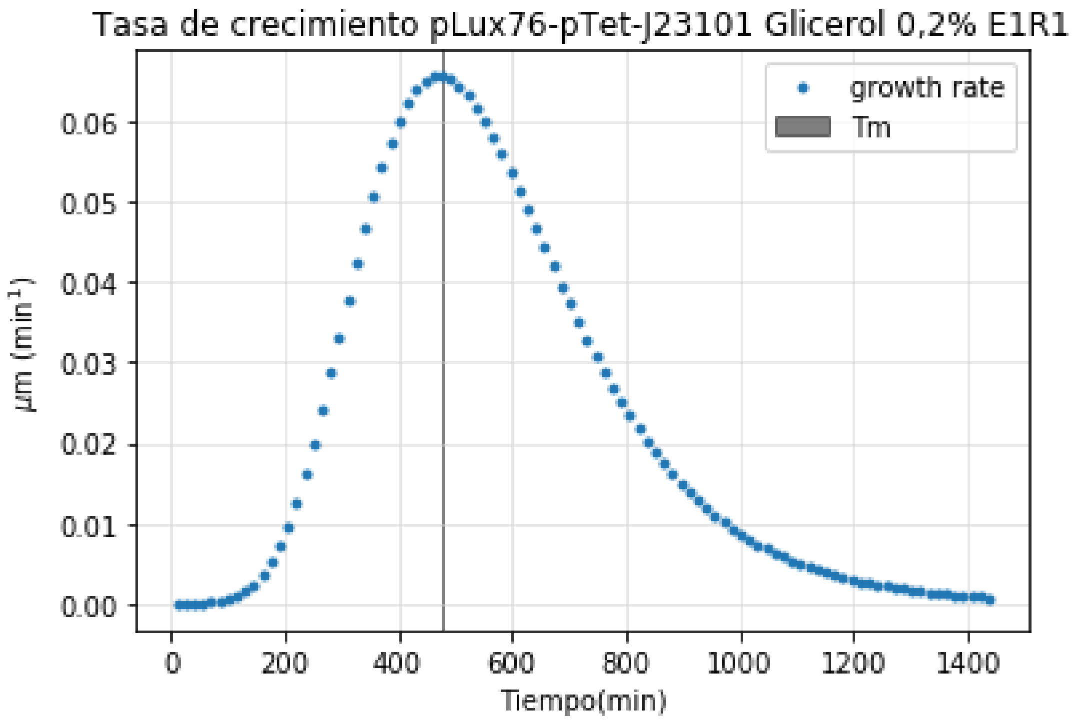

```
In [126]: #tasa de crecimiento
          ye21=((A21*np.exp(-np.exp((((um21*np.exp(1))/A21)*(l21-tt))+1))))
          #Con diff
          dy21=(np.diff(ye21))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glucosa 0,4% E1R3')
          plt.xlabel('Tiempo(min)')
          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[126]: <matplotlib.legend.Legend at 0x1abae812ba8>
```



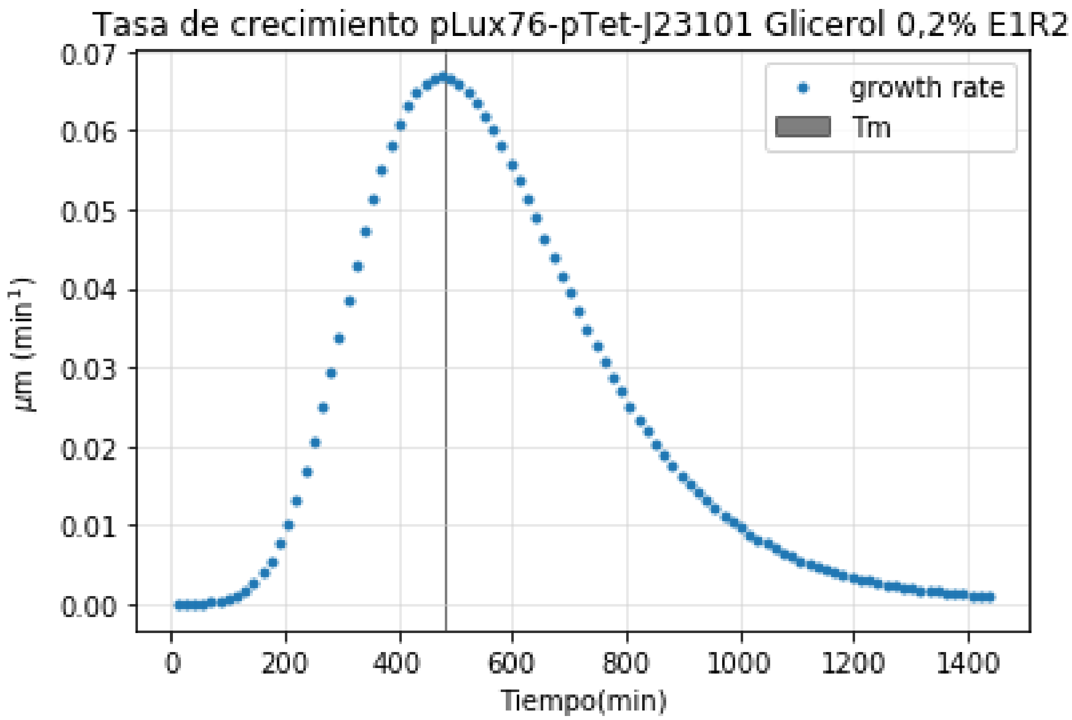

```
In [127]: #Tasas plux76-ptet-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='growth rate E1R1')
          plt.plot(tt[:-1],dy20,label='growth rate E1R2')
          plt.plot(tt[:-1],dy21,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[127]: <matplotlib.legend.Legend at 0x1abaf8b6c18>
```



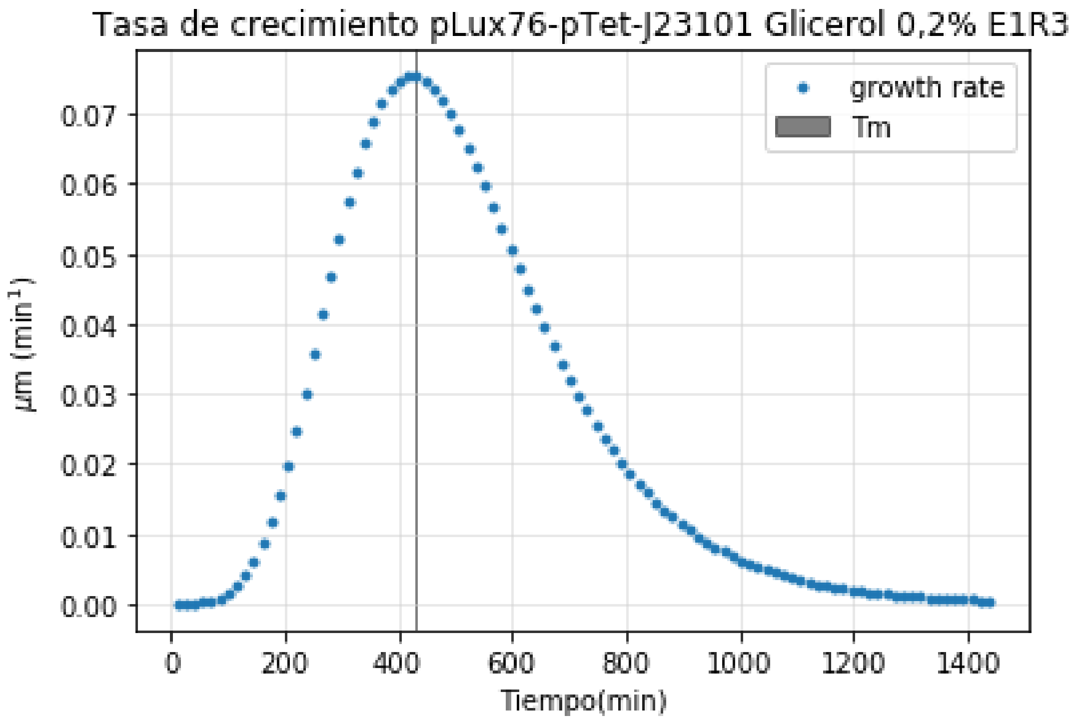

```
In [128]: #tasa de crecimiento
          ye22=((A22*np.exp(-np.exp((((um22*np.exp(1))/A22)*(l22-tt))+1))))
          #Con diff
          dy22=(np.diff(ye22))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glicerol 0,2% E1R1')
          plt.xlabel('Tiempo(min)')
          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[128]: <matplotlib.legend.Legend at 0x1abaf99e7f0>
```




```
In [129]: #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-pTet-J23101 Glicerol 0,2% E1R2')
          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[129]: <matplotlib.legend.Legend at 0x1abafa63eb8>
```

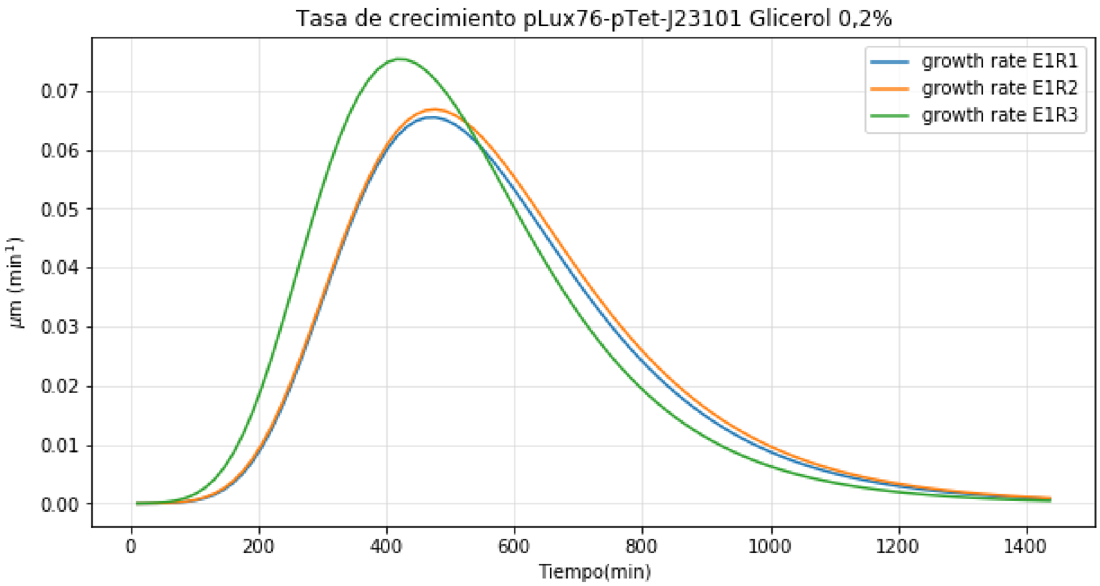



```
In [130]: #tasa de crecimiento
          ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(l24-tt))+1))))
          #Con diff
          dy24=(np.diff(ye24))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glicerol 0,2% E1R3')
          plt.xlabel('Tiempo(min)')
          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[130]: <matplotlib.legend.Legend at 0x1abafb4ada0>
```



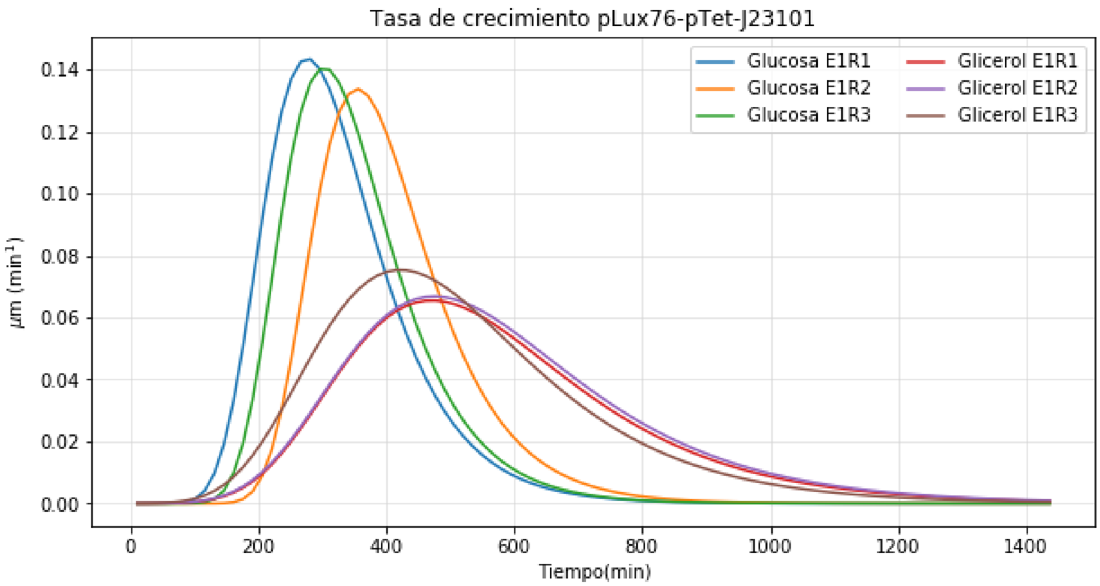
In [131]:
```
#Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pTet-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy22,label='growth rate E1R1')
plt.plot(tt[:-1],dy23,label='growth rate E1R2')
plt.plot(tt[:-1],dy24,label='growth rate E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[131]:
```
<matplotlib.legend.Legend at 0x1abafc1acf8>
```

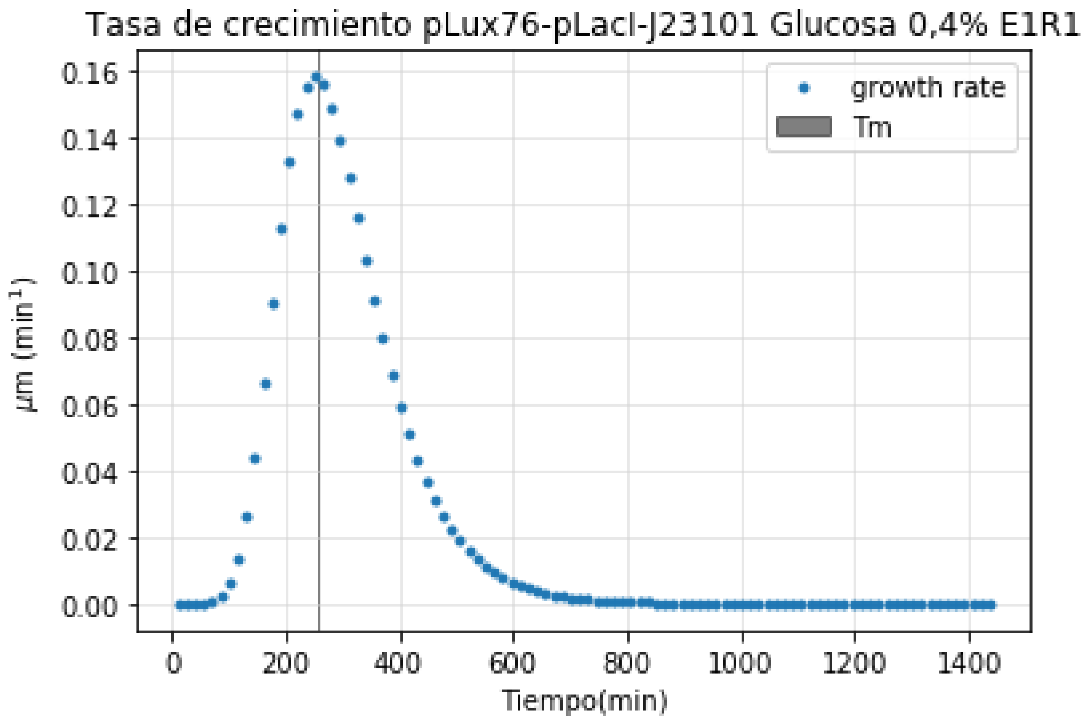


In [132]:
```python
#Tasas control réplicas
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pTet-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy19,label='Glucosa E1R1')
plt.plot(tt[:-1],dy20,label='Glucosa E1R2')
plt.plot(tt[:-1],dy21,label='Glucosa E1R3')
plt.plot(tt[:-1],dy22,label='Glicerol E1R1')
plt.plot(tt[:-1],dy23,label='Glicerol E1R2')
plt.plot(tt[:-1],dy24,label='Glicerol E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right',ncol=2)
```

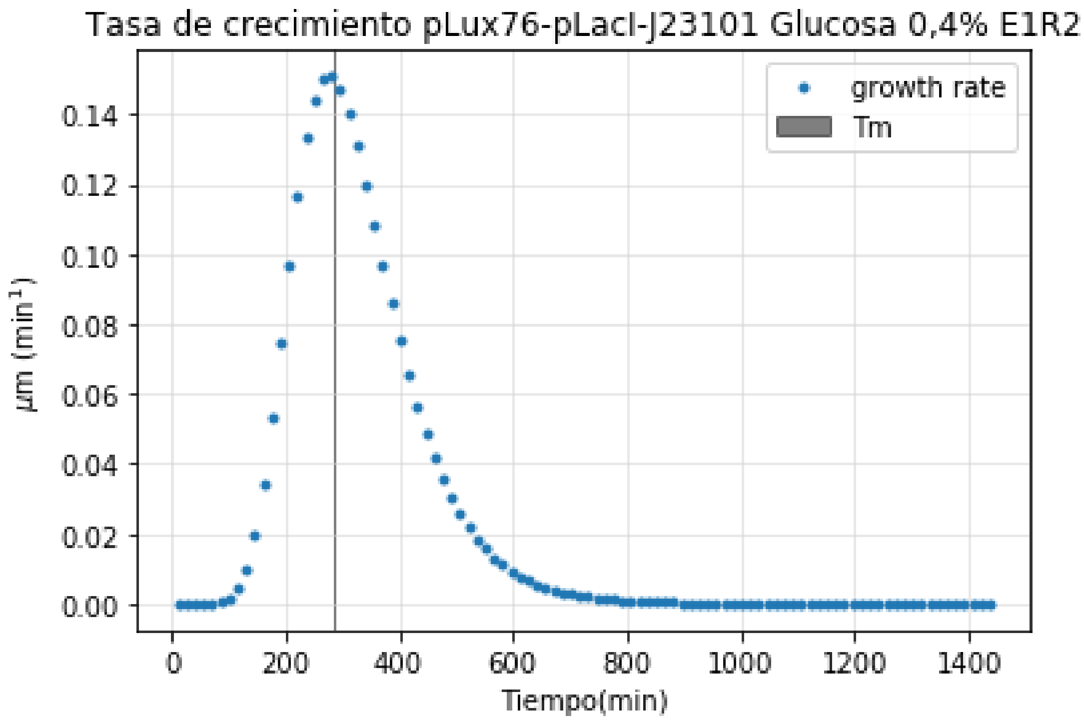
Out[132]:
```
<matplotlib.legend.Legend at 0x1abafd0cda0>
```




```
In [133]: #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-pLacI-J23101 Glucosa 0,4% E1R1')
          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[133]: <matplotlib.legend.Legend at 0x1abafdeaa58>
```

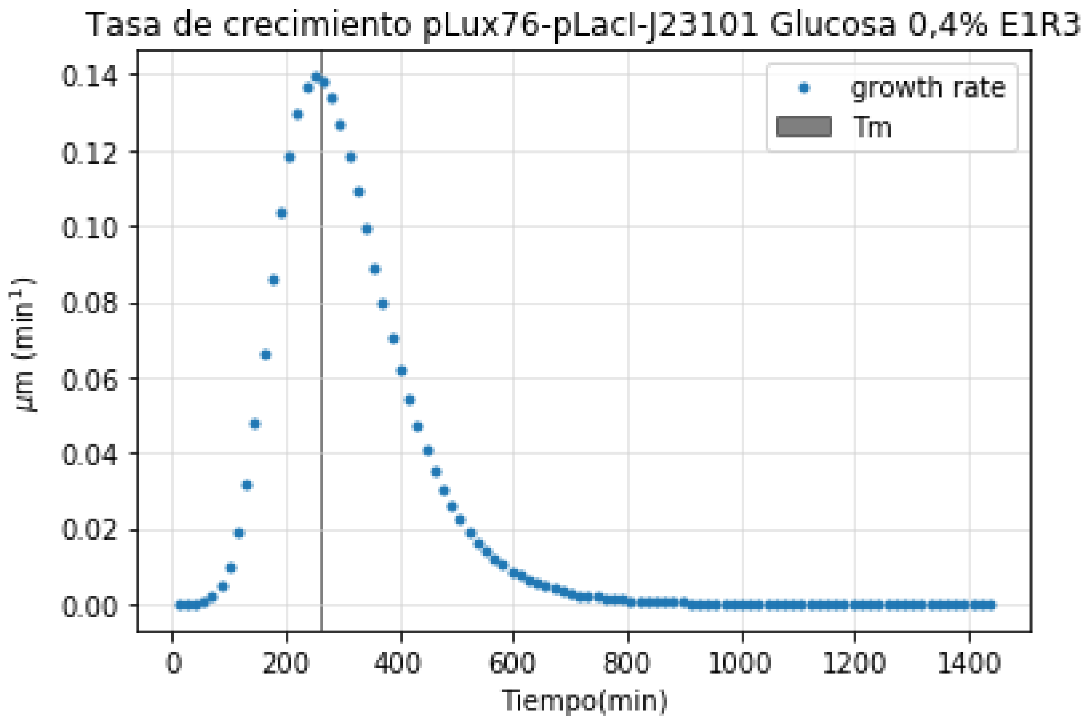



```
In [134]: #tasa de crecimiento
          ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(l26-tt))+1))))
          #Con diff
          dy26=(np.diff(ye26))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4% E1R2')
          plt.xlabel('Tiempo(min)')
          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[134]: <matplotlib.legend.Legend at 0x1abafeab8d0>
```



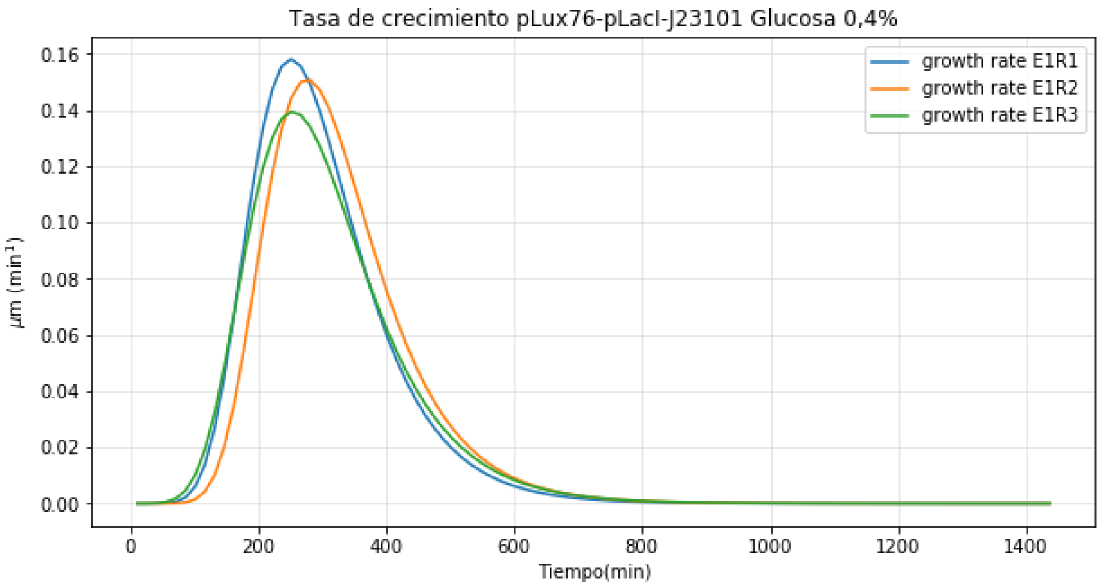
In [135]:
```
#tasa de crecimiento
ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(l27-tt))+1))))
#Con diff
dy27=(np.diff(ye27))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4% E1R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm27,tm27, color='k', 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[135]:
```
<matplotlib.legend.Legend at 0x1abaff82b70>
```



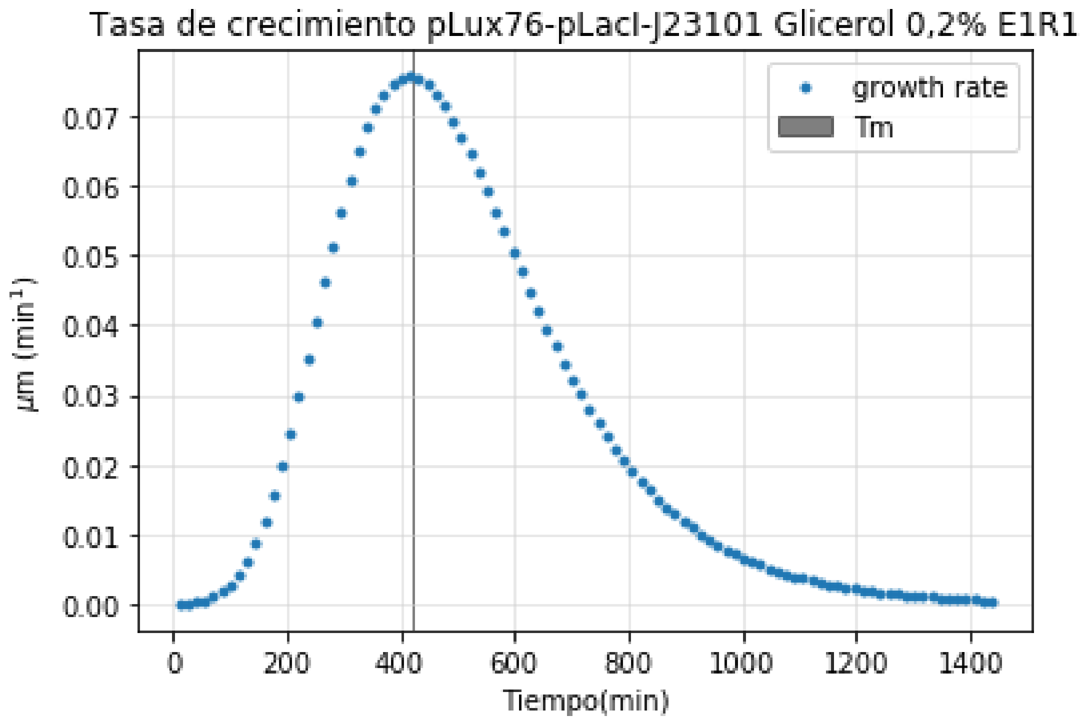
In [136]:
```python
#Tasas plux76-plac-J23101 réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glucosa 0,4%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy25,label='growth rate E1R1')
plt.plot(tt[:-1],dy26,label='growth rate E1R2')
plt.plot(tt[:-1],dy27,label='growth rate E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[136]:
```
<matplotlib.legend.Legend at 0x1abb1025b00>
```

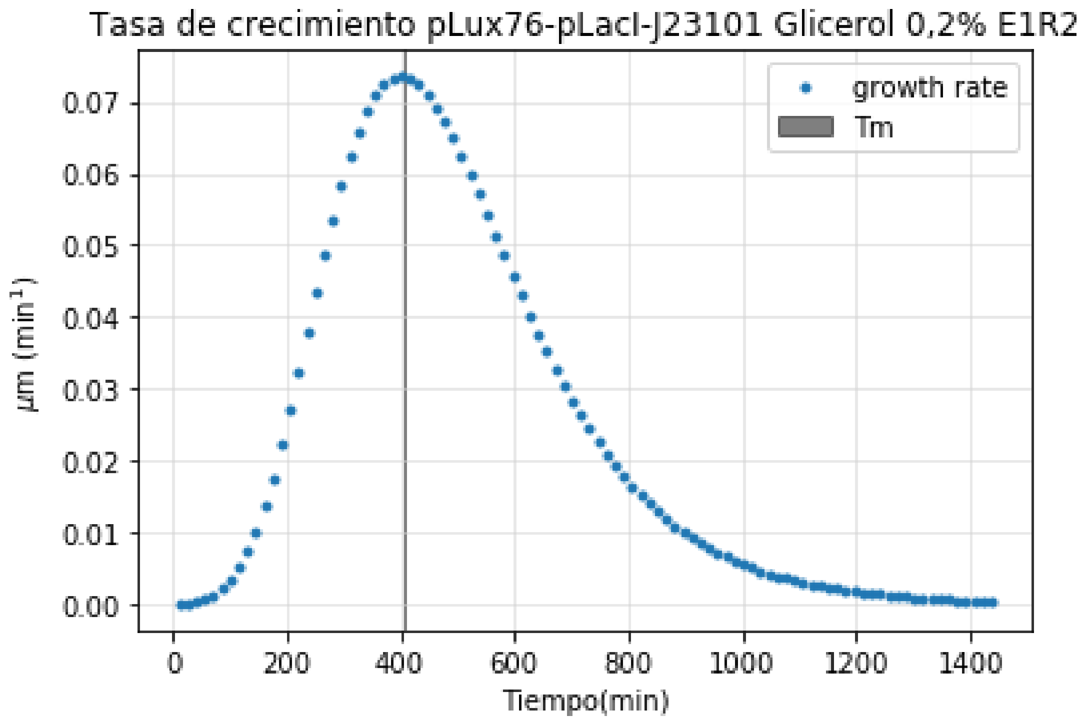


In [137]:
```
#tasa de crecimiento
ye28=((A28*np.exp(-np.exp((((um28*np.exp(1))/A28)*(l28-tt))+1))))
#Con diff
dy28=(np.diff(ye28))
plt.figure()
plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2% E1R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm28,tm28, color='k', 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')
```

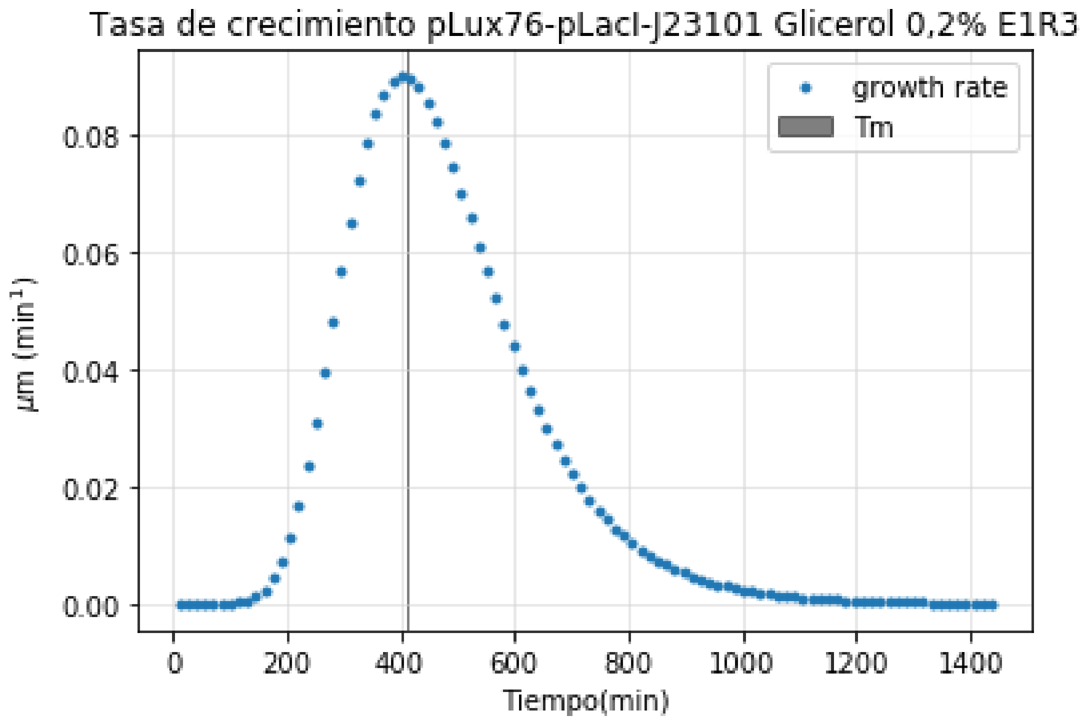
Out[137]:
```
<matplotlib.legend.Legend at 0x1abb10f1940>
```



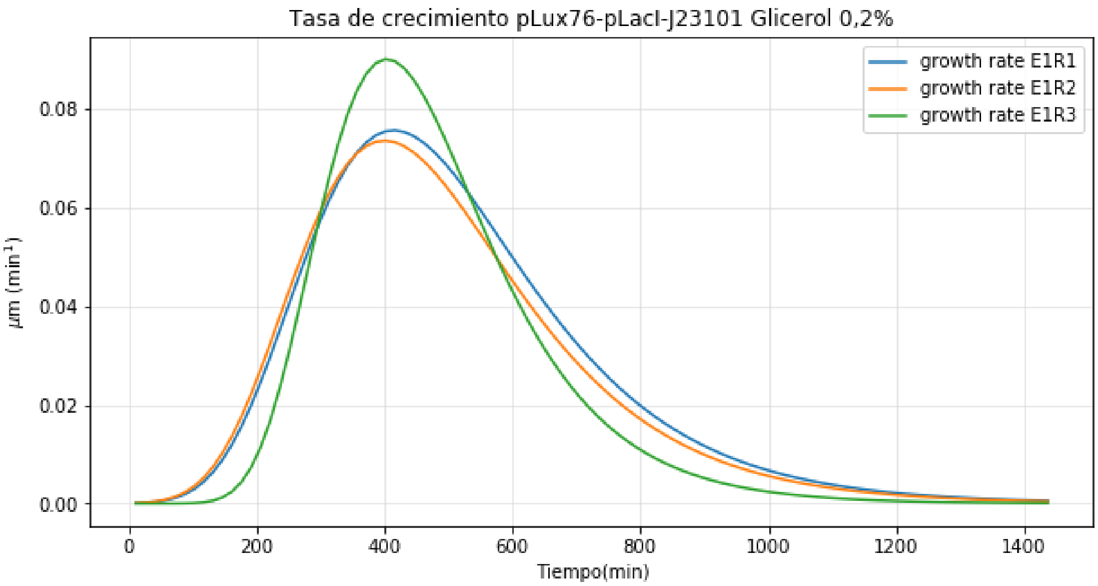

```
In [138]: #tasa de crecimiento
          ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(l29-tt))+1))))
          #Con diff
          dy29=(np.diff(ye29))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2% E1R2')
          plt.xlabel('Tiempo(min)')
          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[138]: <matplotlib.legend.Legend at 0x1abaf8be2b0>
```



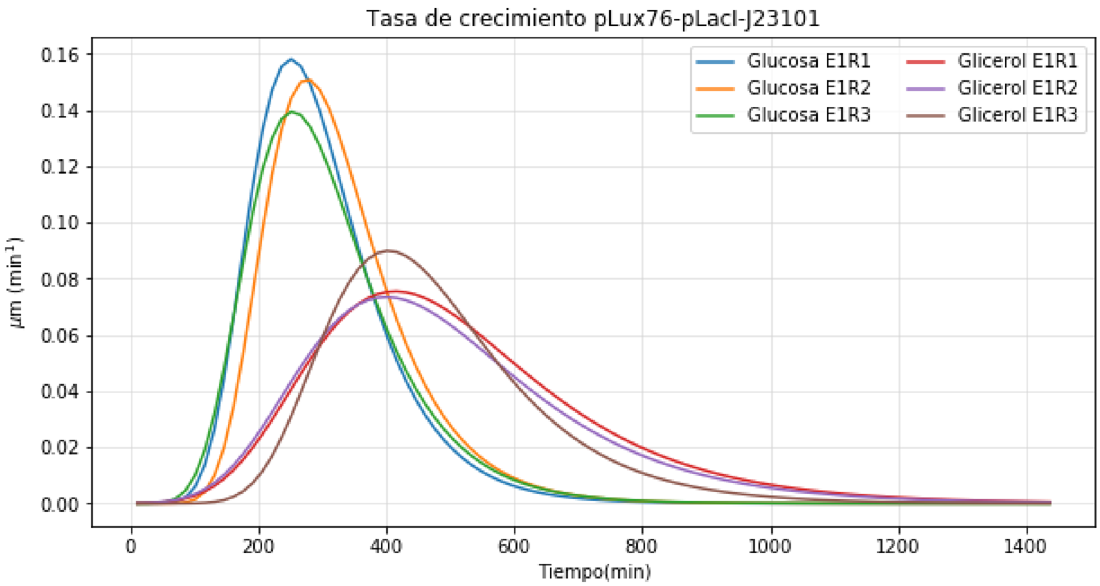

```
In [139]: #tasa de crecimiento
          ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(l30-tt))+1))))
          #Con diff
          dy30=(np.diff(ye30))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2% E1R3')
          plt.xlabel('Tiempo(min)')
          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[139]: <matplotlib.legend.Legend at 0x1abad365f60>
```



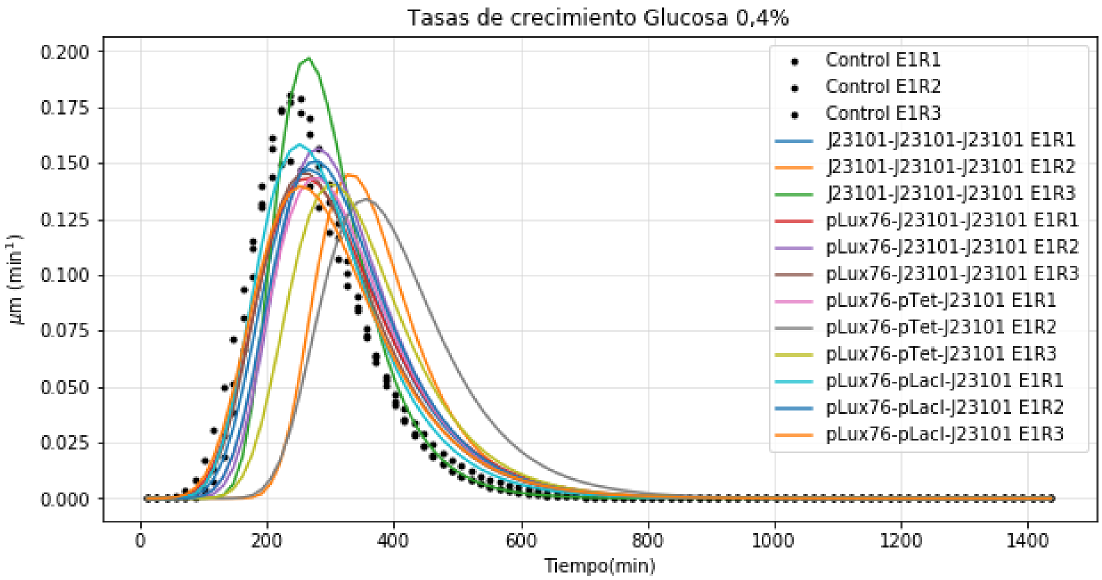

```
In [140]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-pLacI-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy28,label='growth rate E1R1')
          plt.plot(tt[:-1],dy29,label='growth rate E1R2')
          plt.plot(tt[:-1],dy30,label='growth rate E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[140]: <matplotlib.legend.Legend at 0x1abb11c3da0>
```




```
In [141]: #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],dy25,label='Glucosa E1R1')
          plt.plot(tt[:-1],dy26,label='Glucosa E1R2')
          plt.plot(tt[:-1],dy27,label='Glucosa E1R3')
          plt.plot(tt[:-1],dy28,label='Glicerol E1R1')
          plt.plot(tt[:-1],dy29,label='Glicerol E1R2')
          plt.plot(tt[:-1],dy30,label='Glicerol E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[141]: <matplotlib.legend.Legend at 0x1abb128fb00>
```

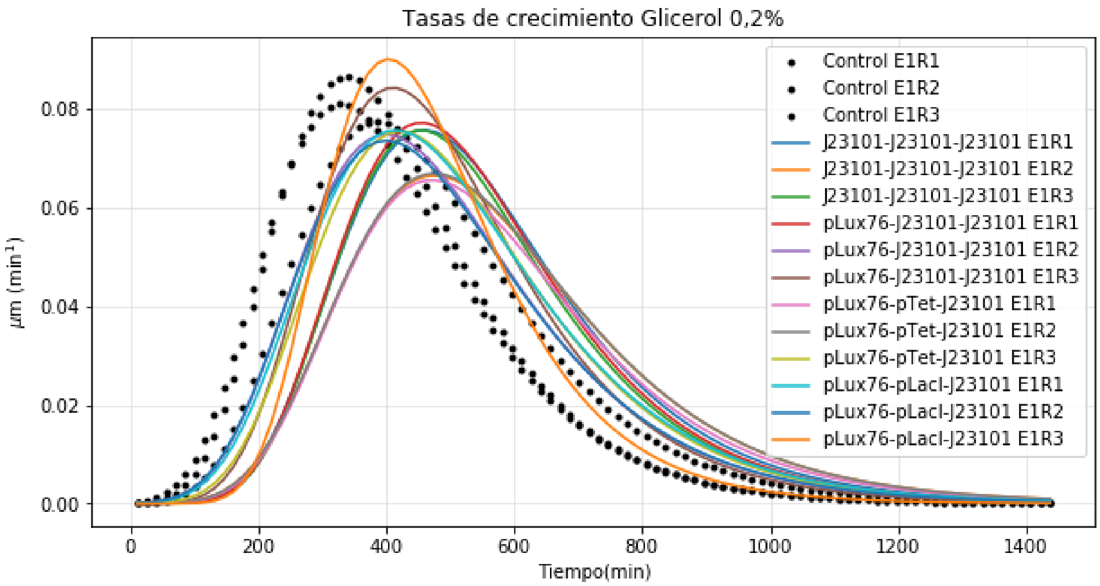



```
In [142]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,'k.',label='Control E1R1')
          plt.plot(tt[:-1],dy2,'k.',label='Control E1R2')
          plt.plot(tt[:-1],dy3,'k.',label='Control E1R3')
          plt.plot(tt[:-1],dy7,label='J23101-J23101-J23101 E1R1')
          plt.plot(tt[:-1],dy8,label='J23101-J23101-J23101 E1R2')
          plt.plot(tt[:-1],dy9,label='J23101-J23101-J23101 E1R3')
          plt.plot(tt[:-1],dy13,label='pLux76-J23101-J23101 E1R1')
          plt.plot(tt[:-1],dy14,label='pLux76-J23101-J23101 E1R2')
          plt.plot(tt[:-1],dy15,label='pLux76-J23101-J23101 E1R3')
          plt.plot(tt[:-1],dy19,label='pLux76-pTet-J23101 E1R1')
          plt.plot(tt[:-1],dy20,label='pLux76-pTet-J23101 E1R2')
          plt.plot(tt[:-1],dy21,label='pLux76-pTet-J23101 E1R3')
          plt.plot(tt[:-1],dy25,label='pLux76-pLacI-J23101 E1R1')
          plt.plot(tt[:-1],dy26,label='pLux76-pLacI-J23101 E1R2')
          plt.plot(tt[:-1],dy27,label='pLux76-pLacI-J23101 E1R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[142]: <matplotlib.legend.Legend at 0x1abb139ab00>
```



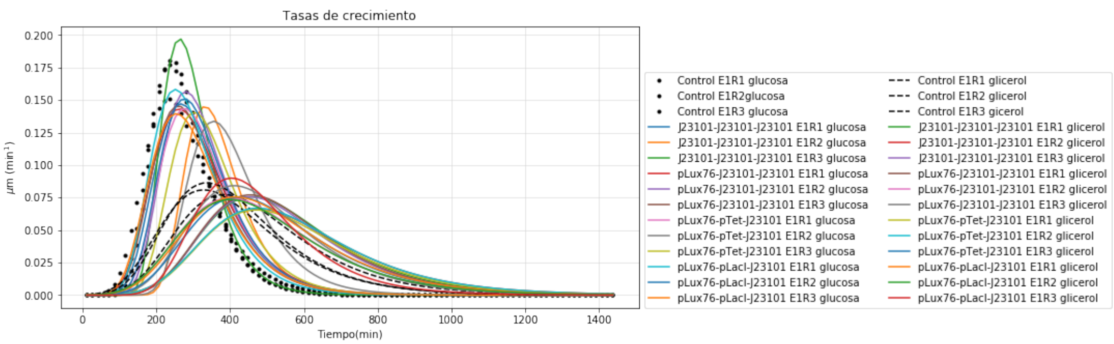
In [143]:
```
#Tasas réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy4,'k.',label='Control E1R1')
plt.plot(tt[:-1],dy5,'k.',label='Control E1R2')
plt.plot(tt[:-1],dy6,'k.',label='Control E1R3')
plt.plot(tt[:-1],dy10,label='J23101-J23101-J23101 E1R1')
plt.plot(tt[:-1],dy11,label='J23101-J23101-J23101 E1R2')
plt.plot(tt[:-1],dy12,label='J23101-J23101-J23101 E1R3')
plt.plot(tt[:-1],dy16,label='pLux76-J23101-J23101 E1R1')
plt.plot(tt[:-1],dy17,label='pLux76-J23101-J23101 E1R2')
plt.plot(tt[:-1],dy18,label='pLux76-J23101-J23101 E1R3')
plt.plot(tt[:-1],dy22,label='pLux76-pTet-J23101 E1R1')
plt.plot(tt[:-1],dy23,label='pLux76-pTet-J23101 E1R2')
plt.plot(tt[:-1],dy24,label='pLux76-pTet-J23101 E1R3')
plt.plot(tt[:-1],dy28,label='pLux76-pLacI-J23101 E1R1')
plt.plot(tt[:-1],dy29,label='pLux76-pLacI-J23101 E1R2')
plt.plot(tt[:-1],dy30,label='pLux76-pLacI-J23101 E1R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[143]: <matplotlib.legend.Legend at 0x1abb150eb00>



In [144]:
```
#Tasas réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy1,'k.',label='Control E1R1 glucosa')
plt.plot(tt[:-1],dy2,'k.',label='Control E1R2glucosa')
plt.plot(tt[:-1],dy3,'k.',label='Control E1R3 glucosa')
plt.plot(tt[:-1],dy7,label='J23101-J23101-J23101 E1R1 glucosa')
plt.plot(tt[:-1],dy8,label='J23101-J23101-J23101 E1R2 glucosa')
plt.plot(tt[:-1],dy9,label='J23101-J23101-J23101 E1R3 glucosa')
plt.plot(tt[:-1],dy13,label='pLux76-J23101-J23101 E1R1 glucosa')
plt.plot(tt[:-1],dy14,label='pLux76-J23101-J23101 E1R2 glucosa')
plt.plot(tt[:-1],dy15,label='pLux76-J23101-J23101 E1R3 glucosa')
plt.plot(tt[:-1],dy19,label='pLux76-pTet-J23101 E1R1 glucosa')
plt.plot(tt[:-1],dy20,label='pLux76-pTet-J23101 E1R2 glucosa')
plt.plot(tt[:-1],dy21,label='pLux76-pTet-J23101 E1R3 glucosa')
plt.plot(tt[:-1],dy25,label='pLux76-pLacI-J23101 E1R1 glucosa')
plt.plot(tt[:-1],dy26,label='pLux76-pLacI-J23101 E1R2 glucosa')
plt.plot(tt[:-1],dy27,label='pLux76-pLacI-J23101 E1R3 glucosa')
plt.plot(tt[:-1],dy4,'k--',label='Control E1R1 glicerol')
plt.plot(tt[:-1],dy5,'k--',label='Control E1R2 glicerol')
plt.plot(tt[:-1],dy6,'k--',label='Control E1R3 gicerol')
plt.plot(tt[:-1],dy10,label='J23101-J23101-J23101 E1R1 glicerol')
plt.plot(tt[:-1],dy11,label='J23101-J23101-J23101 E1R2 glicerol')
plt.plot(tt[:-1],dy12,label='J23101-J23101-J23101 E1R3 glicerol')
plt.plot(tt[:-1],dy16,label='pLux76-J23101-J23101 E1R1 glicerol')
plt.plot(tt[:-1],dy17,label='pLux76-J23101-J23101 E1R2 glicerol')
```

```
plt.plot(tt[:-1],dy18,label='pLux76-J23101-J23101 E1R3 glicerol')
plt.plot(tt[:-1],dy22,label='pLux76-pTet-J23101 E1R1 glicerol')
plt.plot(tt[:-1],dy23,label='pLux76-pTet-J23101 E1R2 glicerol')
plt.plot(tt[:-1],dy24,label='pLux76-pTet-J23101 E1R3 glicerol')
plt.plot(tt[:-1],dy28,label='pLux76-pLacI-J23101 E1R1 glicerol')
plt.plot(tt[:-1],dy29,label='pLux76-pLacI-J23101 E1R2 glicerol')
plt.plot(tt[:-1],dy30,label='pLux76-pLacI-J23101 E1R3 glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
plt.savefig(filename='tasas crecimiento.png', dpi=300, facecolor='w', edgecolor='w',bb
```



In [ ]:

In [ ]: