Vectores con J23101-YFP

February 13, 2018

```
In [1]: import numpy as np
        import matplotlib
        import matplotlib.pyplot as plt
        %matplotlib inline
        from matplotlib import colors
        from scipy.interpolate import UnivariateSpline
        from scipy.optimize import curve_fit
        from scipy import stats
        import seaborn as sns

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

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

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

'''
print(cfp18211.shape)
print(rfp18211.shape)
print(yfp18211.shape)
print(od18211.shape)
print(cfp18212.shape)
print(rfp18212.shape)
print(yfp18212.shape)
print(od18212.shape)
print(cfp18213.shape)
print(rfp18213.shape)
print(yfp18213.shape)
print(od18213.shape)'''

cfp12211=np.fromfile('p1221gCFP1', sep=',')
rfp12211=np.fromfile('p1221gRFP1', sep=',')
yfp12211=np.fromfile('p1221gYFP1', sep=',')
od12211=np.fromfile('p1221gOD1', sep=',')
cfp12212=np.fromfile('p1221gCFP2', sep=',')
rfp12212=np.fromfile('p1221gRFP2', sep=',')
yfp12212=np.fromfile('p1221gYFP2', sep=',')
od12212=np.fromfile('p1221gOD2', sep=',')
cfp12213=np.fromfile('p1221gCFP3', sep=',')
rfp12213=np.fromfile('p1221gRFP3', sep=',')
yfp12213=np.fromfile('p1221gYFP3', sep=',')
od12213=np.fromfile('p1221gOD3', sep=',')

'''
print(cfp12211.shape)
print(rfp12211.shape)
print(yfp12211.shape)
print(od12211.shape)
print(cfp12212.shape)
```

```
print(rfp12212.shape)
print(yfp12212.shape)
print(od12212.shape)
print(cfp12213.shape)
print(rfp12213.shape)
print(yfp12213.shape)
print(od12213.shape)'''

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

'''
print(cfpcg151.shape)
print(rfpcg151.shape)
print(yfpcg151.shape)
print(odcg151.shape)
print(cfpcg151.shape)
print(rfpcg151.shape)
print(yfpcg151.shape)
print(odcg151.shape)
print(cfpcg151.shape)
print(rfpcg151.shape)
print(yfpcg151.shape)
print(odcg151.shape)'''

cfpcg181=np.fromfile('18pcgCFP1', sep=',')
rfpcg181=np.fromfile('18pcgRFP1', sep=',')
yfpcg181=np.fromfile('18pcgYFP1', sep=',')
odcg181=np.fromfile('18pcgOD1', sep=',')
cfpcg182=np.fromfile('18pcgCFP2', sep=',')
rfpcg182=np.fromfile('18pcgRFP2', sep=',')
yfpcg182=np.fromfile('18pcgYFP2', sep=',')
odcg182=np.fromfile ('18pcgOD2',sep=',')
cfpcg183=np.fromfile('18pcgCFP3', sep=',')
rfpcg183=np.fromfile('18pcgRFP3', sep=',')
yfpcg183=np.fromfile('18pcgYFP3', sep=',')
```

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

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

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

Out[2]: '\nprint(cfpcg121.shape)\nprint(rfpcg121.shape)\nprint(yfpcg121.shape)\nprint(odcg121.sh

In [3]:
```
#Promedios glicerol
#arrays replicas glicerol
cfp1521g1=np.fromfile('p1521glCFP1', sep=',')
rfp1521g1=np.fromfile('p1521glRFP1', sep=',')
```

```
yfp1521g1=np.fromfile('p1521glYFP1', sep=',')
od1521g1=np.fromfile('p1521glOD1', sep=',')
cfp1521g2=np.fromfile('p1521glCFP2', sep=',')
rfp1521g2=np.fromfile('p1521glRFP2', sep=',')
yfp1521g2=np.fromfile('p1521glYFP2', sep=',')
od1521g2=np.fromfile('p1521glOD2', sep=',')
cfp1521g3=np.fromfile('p1521glCFP3', sep=',')
rfp1521g3=np.fromfile('p1521glRFP3', sep=',')
yfp1521g3=np.fromfile('p1521glYFP3', sep=',')
od1521g3=np.fromfile('p1521glOD3', sep=',')
'''
print(cfp1521g1.shape)
print(rfp1521g1.shape)
print(yfp1521g1.shape)
print(od1521g1.shape)
print(cfp1521g2.shape)
print(rfp1521g2.shape)
print(yfp1521g2.shape)
print(od1521g2.shape)
print(cfp1521g3.shape)
print(rfp1521g3.shape)
print(yfp1521g3.shape)
print(od1521g3.shape)'''

cfp1821g1=np.fromfile('p1821glCFP1', sep=',')
rfp1821g1=np.fromfile('p1821glRFP1', sep=',')
yfp1821g1=np.fromfile('p1821glYFP1', sep=',')
od1821g1=np.fromfile('p1821glOD1', sep=',')
cfp1821g2=np.fromfile('p1821glCFP2', sep=',')
rfp1821g2=np.fromfile('p1821glRFP2', sep=',')
yfp1821g2=np.fromfile('p1821glYFP2', sep=',')
od1821g2=np.fromfile('p1821glOD2', sep=',')
cfp1821g3=np.fromfile('p1821glCFP3', sep=',')
rfp1821g3=np.fromfile('p1821glRFP3', sep=',')
yfp1821g3=np.fromfile('p1821glYFP3', sep=',')
od1821g3=np.fromfile('p1821glOD3', sep=',')
'''
print(cfp1821g1.shape)
print(rfp1821g1.shape)
print(yfp1821g1.shape)
print(od1821g1.shape)
print(cfp1821g2.shape)
print(rfp1821g2.shape)
print(yfp1821g2.shape)
print(od1821g2.shape)
print(cfp1821g3.shape)
print(rfp1821g3.shape)
print(yfp1821g3.shape)
```

```
print(od1821g3.shape)'''

cfp1221g1=np.fromfile('p1221glCFP1', sep=',')
rfp1221g1=np.fromfile('p1221glRFP1', sep=',')
yfp1221g1=np.fromfile('p1221glYFP1', sep=',')
od1221g1=np.fromfile('p1221glOD1', sep=',')
cfp1221g2=np.fromfile('p1221glCFP2', sep=',')
rfp1221g2=np.fromfile('p1221glRFP2', sep=',')
yfp1221g2=np.fromfile('p1221glYFP2', sep=',')
od1221g2=np.fromfile('p1221glOD2', sep=',')
cfp1221g3=np.fromfile('p1221glCFP3', sep=',')
rfp1221g3=np.fromfile('p1221glRFP3', sep=',')
yfp1221g3=np.fromfile('p1221glYFP3', sep=',')
od1221g3=np.fromfile('p1221glOD3', sep=',')
'''
print(cfp1221g1.shape)
print(rfp1221g1.shape)
print(yfp1221g1.shape)
print(od1221g1.shape)
print(cfp1221g2.shape)
print(rfp1221g2.shape)
print(yfp1221g2.shape)
print(od1221g2.shape)
print(cfp1221g3.shape)
print(rfp1221g3.shape)
print(yfp1221g3.shape)
print(od1221g3.shape)'''

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

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

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

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

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

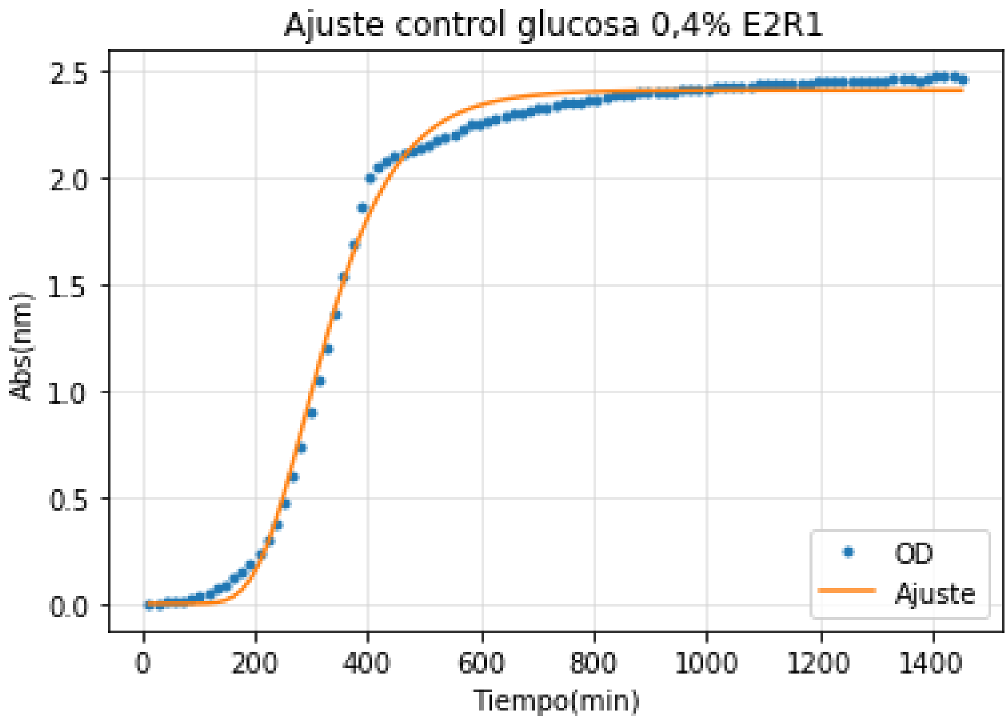
```
Out[3]: '\nprint(cfpcgl121.shape)\nprint(rfpcgl121.shape)\nprint(yfpcgl121.shape)\nprint(odcgl12
```

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

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

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

                print(z)

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

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

                Y_fit=evalF,z

                return(Y_fit)
```

```
In [5]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control 15 glucosa rep 1
        y1 = np.log(odcg151)-np.log(np.min(odcg151))
        print('Min OD = %e'%((np.min(odcg151))))
        evaly, params=Function_fit(tt,y1,0,-1,title = 'Ajuste control glucosa 0,4% E2R1')
        A1 = params[0]
```

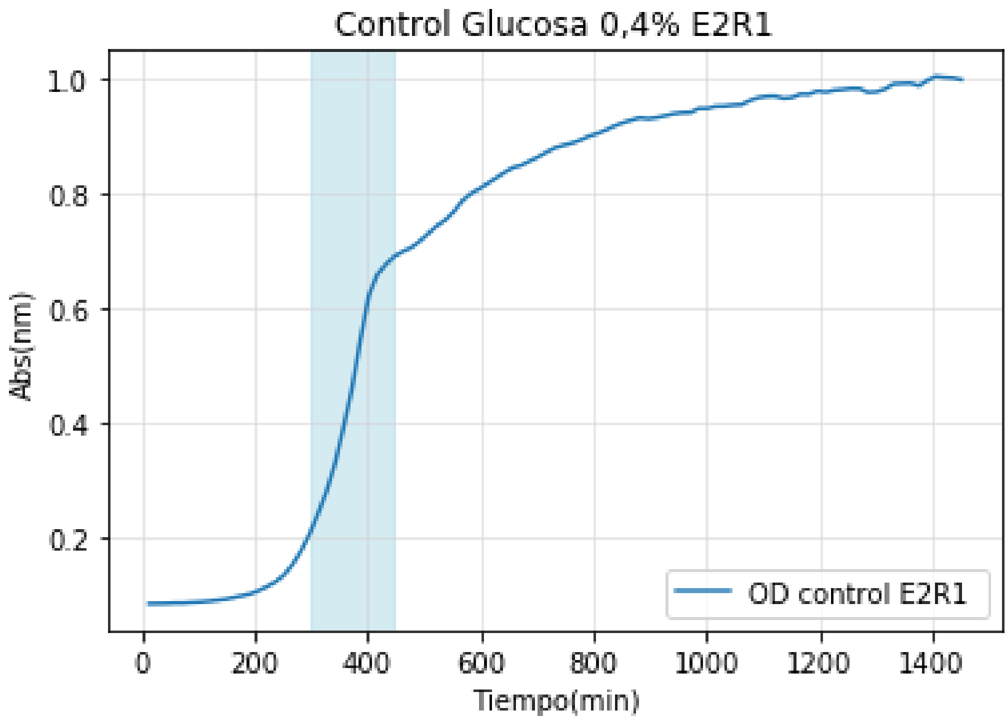
```python
um1=params[1]
l1=params[2]
print('A=%e'%(A1))
print('um=%e'%(um1))
print('l=%e'%(l1))

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

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

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

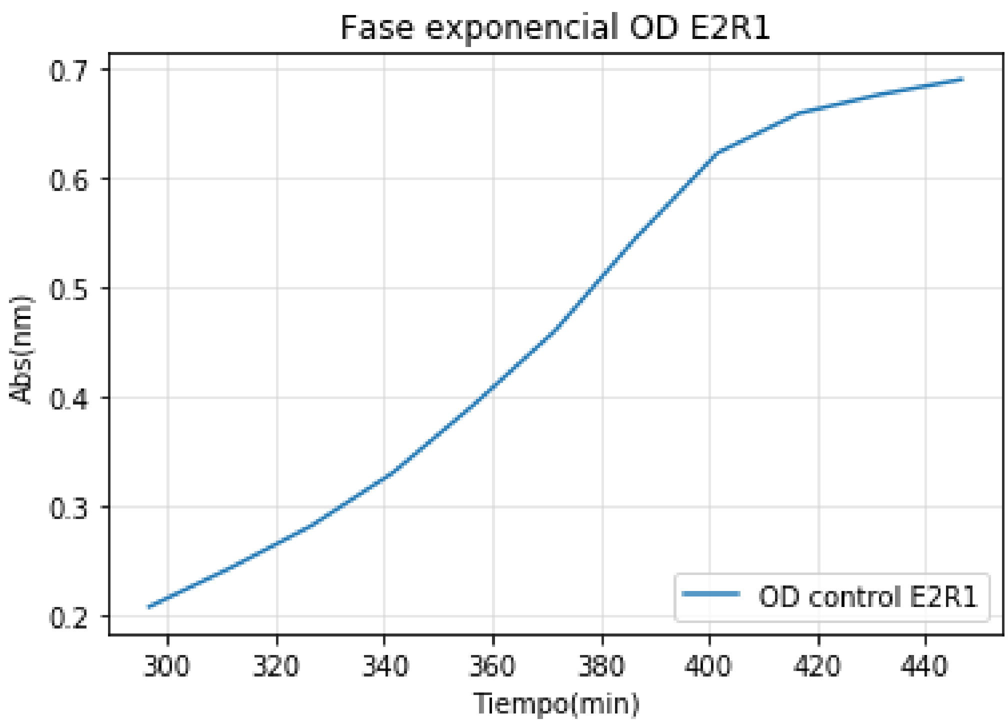
```
Min OD = 8.475000e-02
[  2.40766598e+00   1.00663667e-02   2.01163416e+02]
```




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

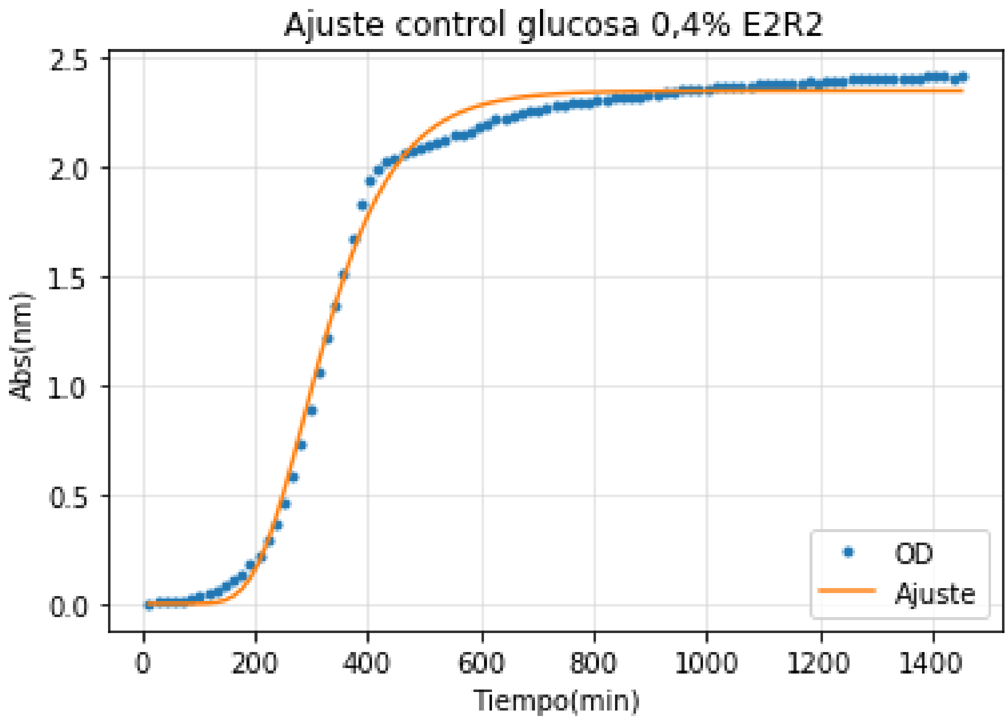
```
Out[5]: <matplotlib.legend.Legend at 0x2035de86da0>
```


Control Glucosa 0,4% E2R1
Abs(nm)
Tiempo(min)
OD control E2R1


Fase exponencial OD E2R1
Abs(nm)
Tiempo(min)
OD control E2R1

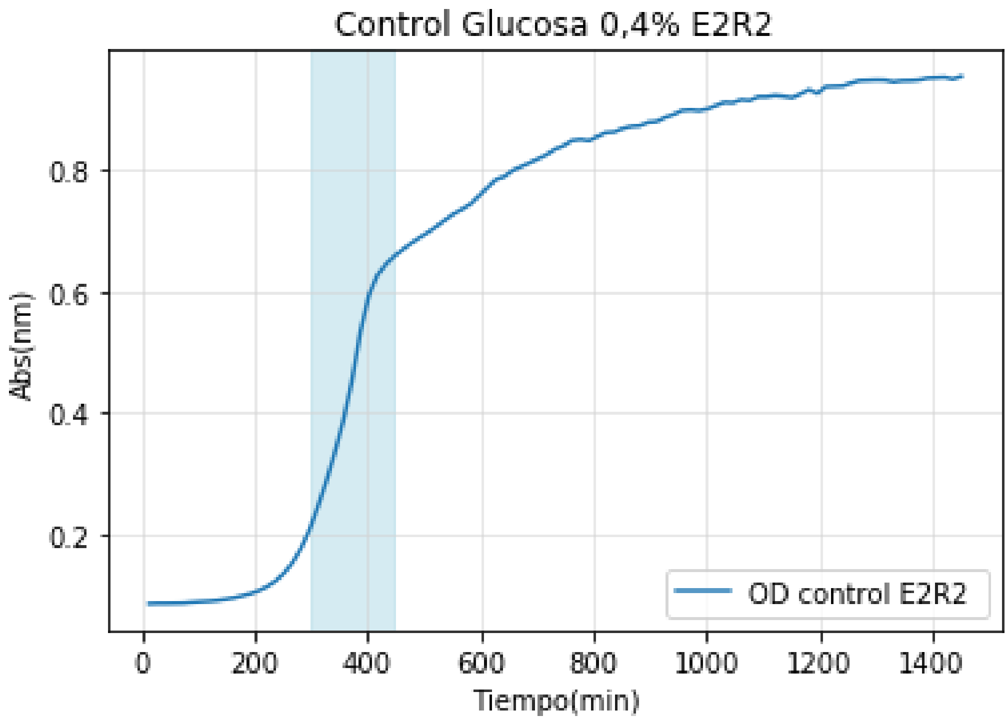
In [6]:
```python
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y2= np.log(odcg152)-np.log(np.min(odcg152))
print('Min OD = %e'%((np.min(odcg152))))
evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E2R2')
A2 = params[0]
um2=params[1]
l2=params[2]
print('A=%e'%(A2))
print('um=%e'%(um2))
print('l=%e'%(l2))

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

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

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

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

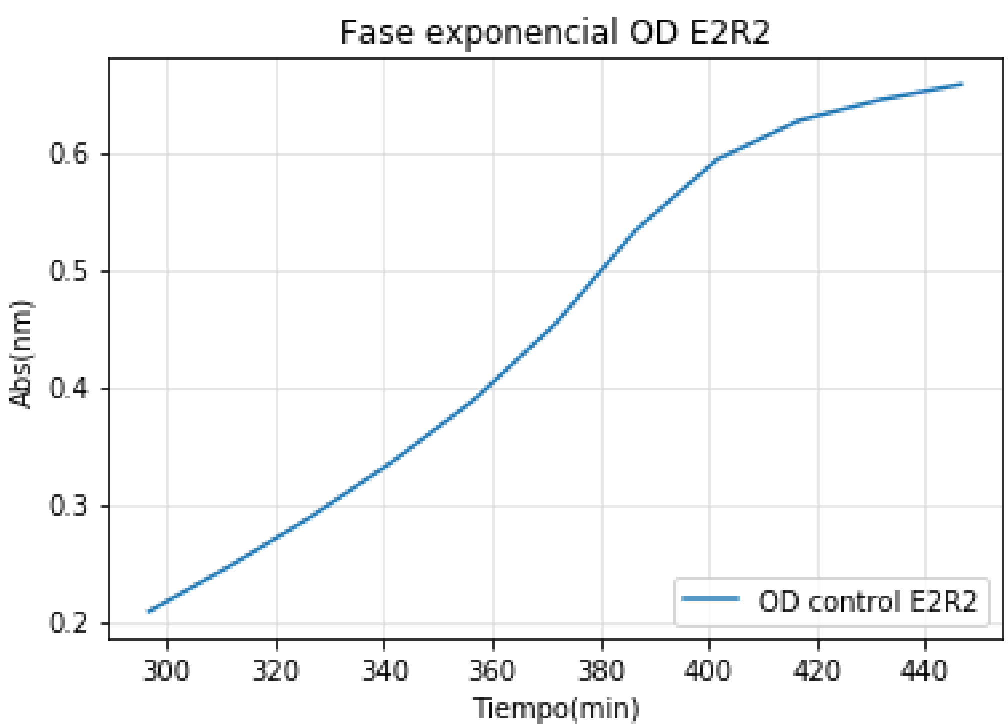



```
A=2.347018e+00
um=9.830048e-03
l=1.997845e+02
Tm=2.876192e+02
doubpe=7.051310e+01
ext=1.410262e+02
Tfinal=4.286455e+02
```

```
Out[6]: <matplotlib.legend.Legend at 0x2035f011438>
```



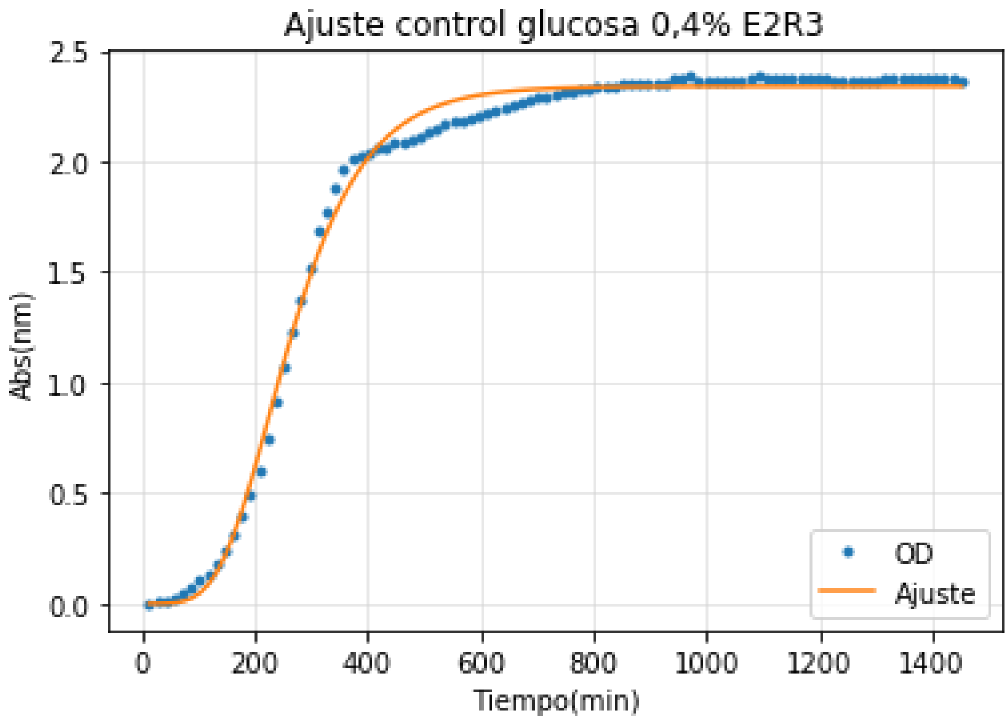
Control Glucosa 0,4% E2R2
Abs(nm)
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E2R2



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

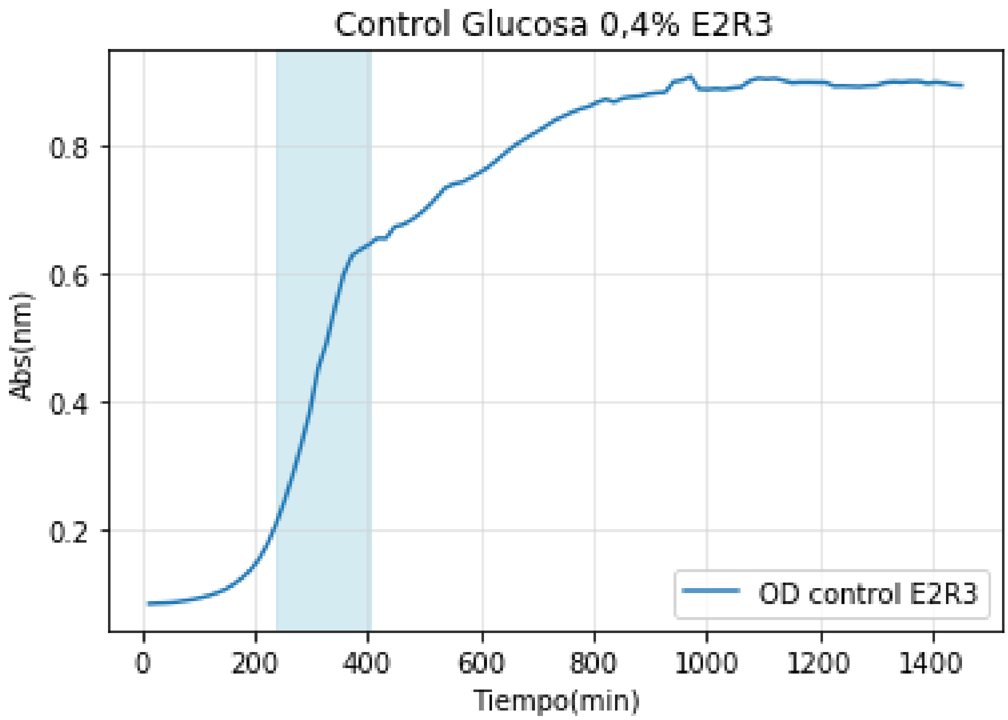
```
In [7]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control glucosa rep 3
        y3= np.log(odcg153)-np.log(np.min(odcg153))
        print('Min OD = %e'%((np.min(odcg153))))
        evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E2R3')
        A3= params[0]
        um3=params[1]
        l3=params[2]
        print('A=%e'%(A3))
        print('um=%e'%(um3))
        print('l=%e'%(l3))

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


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

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

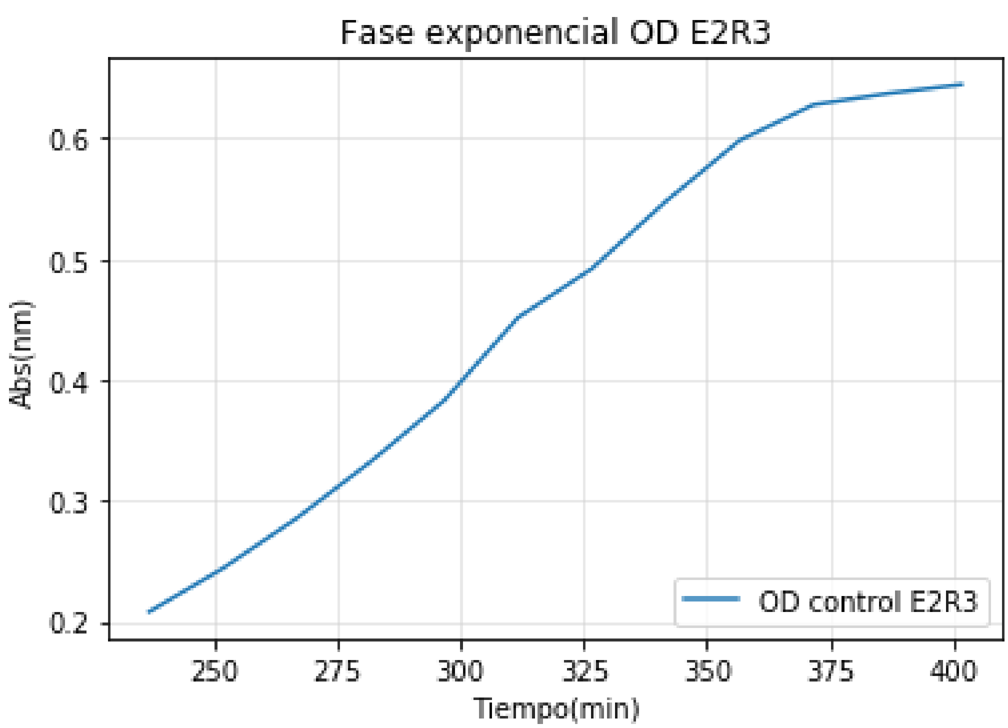
```
Min OD = 8.375000e-02
[  2.33929203e+00   9.40512328e-03   1.34262384e+02]
```




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

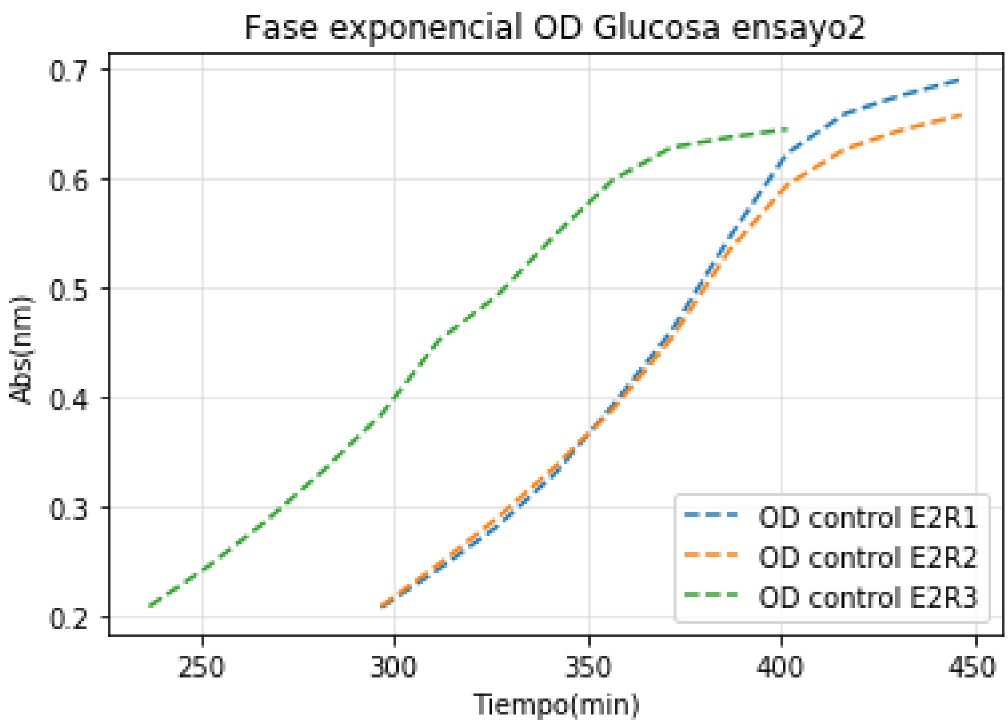
```
Out[7]: <matplotlib.legend.Legend at 0x2035f200a58>
```


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


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

```
In [8]: #Fase exponencial OD/tiempo
        plt.figure()
        plt.title('Fase exponencial OD Glucosa ensayo2')
        plt.xlabel('Tiempo(min)')
        plt.ylabel('Abs(nm)')
        plt.plot(tt[19:30],odcg151[19:30],'--',label='OD control E2R1')
        plt.plot(tt[19:30],odcg152[19:30],'--',label='OD control E2R2')
        plt.plot(tt[15:27],odcg153[15:27],'--',label='OD control E2R3')
        plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
        plt.legend(loc='lower right')
```

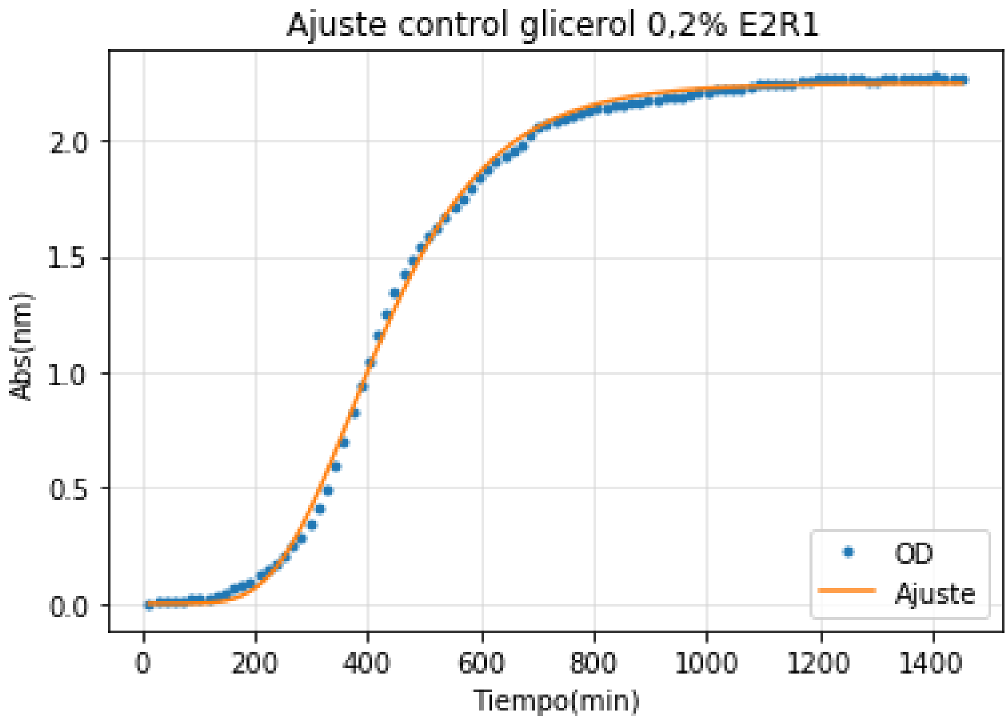
```
Out[8]: <matplotlib.legend.Legend at 0x2035f31df28>
```




```
In [9]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
        #control glicerl rep 1
        y4= np.log(odcgl151)-np.log(np.min(odcgl151))
        print('Min OD = %e'%((np.min(odcgl151))))
        evaly, params=Function_fit(tt,y4,0,-1, title = 'Ajuste control glicerol 0,2% E2R1')
        A4= params[0]
        um4=params[1]
        l4=params[2]
        print('A=%e'%(A4))
        print('um=%e'%(um4))
        print('l=%e'%(l4))
```

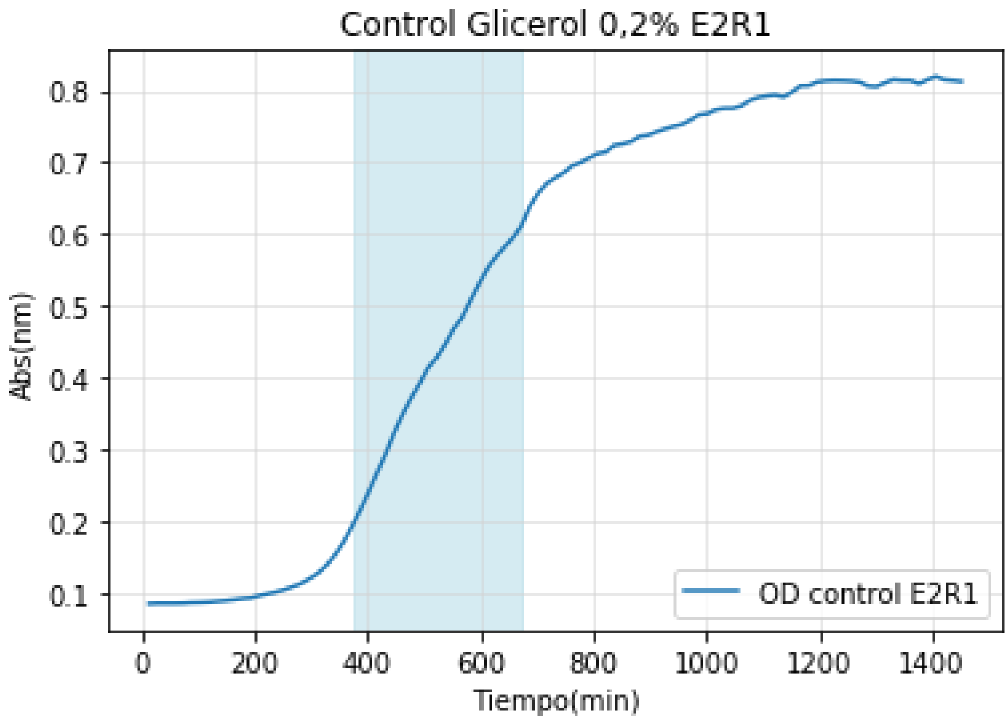
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+l4)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

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

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

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

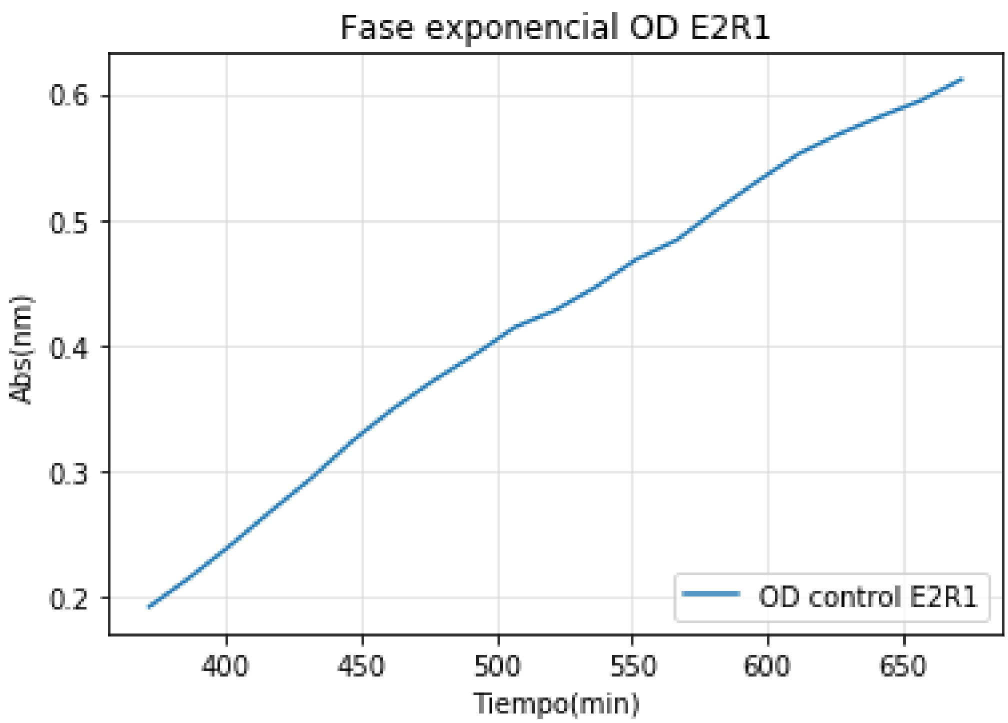



```
A=2.245957e+00
um=6.083361e-03
l=2.342486e+02
Tm=3.700685e+02
doubpe=1.139415e+02
ext=2.848537e+02
Tfinal=6.549222e+02
```

```
Out[9]: <matplotlib.legend.Legend at 0x2035f4aa9e8>
```



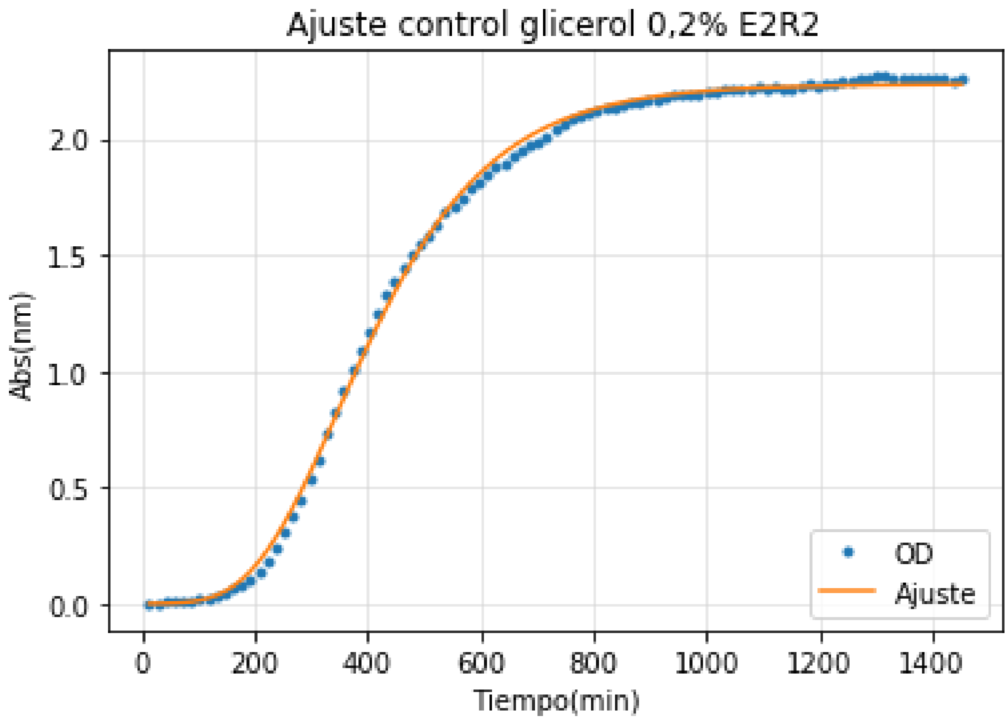
Control Glicerol 0,2% E2R1
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Abs(nm)
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E2R1



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

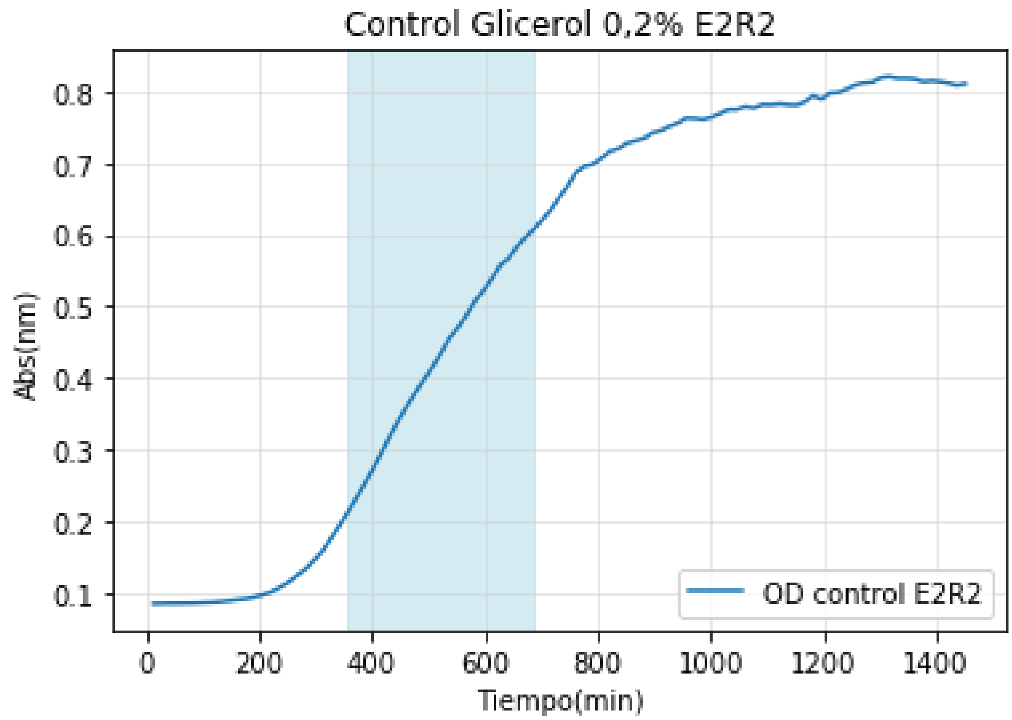
In [10]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 2
y5= np.log(odcgl152)-np.log(np.min(odcgl152))
print('Min OD = %e'%((np.min(odcgl152))))
evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 0,2% E2R2')
A5= params[0]
um5=params[1]
l5=params[2]
print('A=%e'%(A5))
print('um=%e'%(um5))
print('l=%e'%(l5))

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

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

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

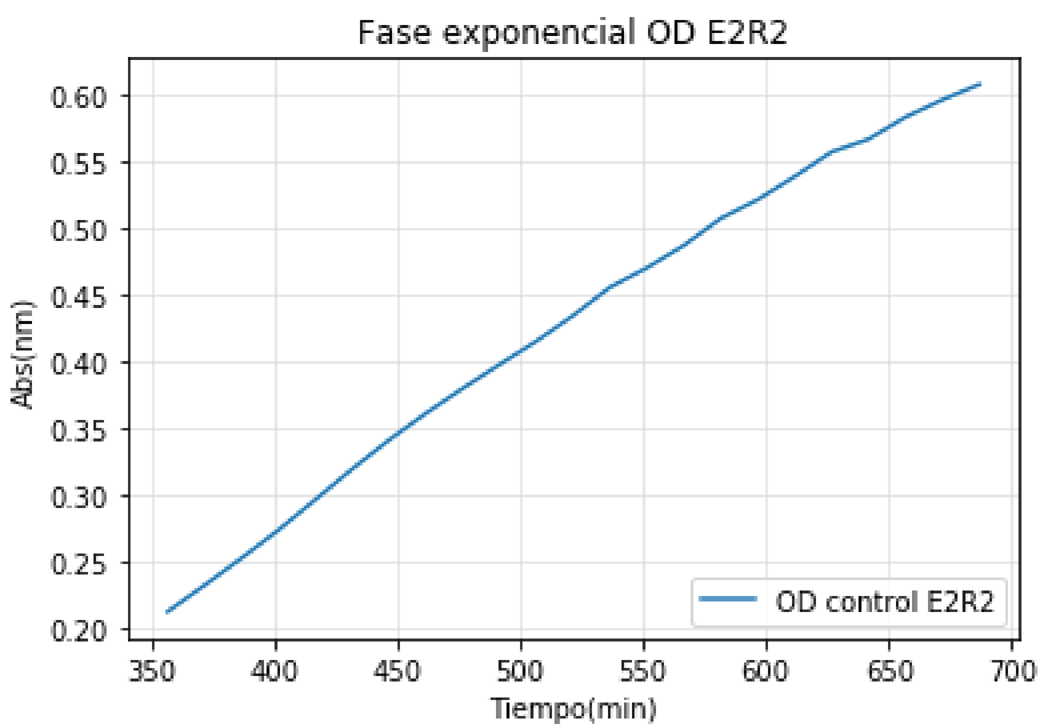
```
Min OD = 8.500000e-02
[  2.23521178e+00   5.43425313e-03   1.94280385e+02]
```




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

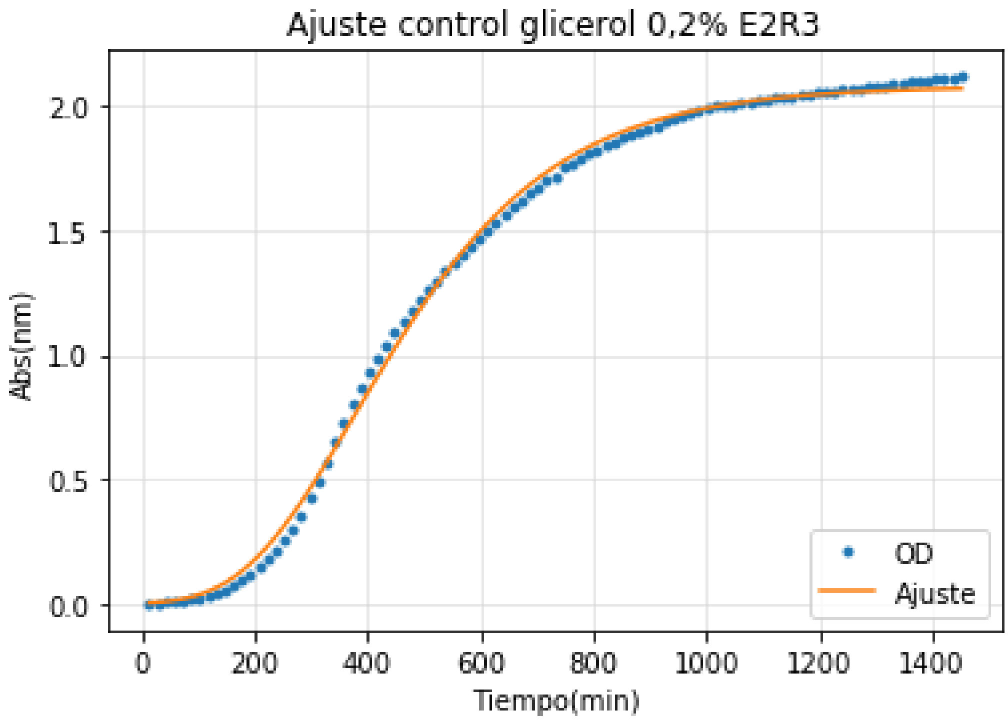
```
Out[10]: <matplotlib.legend.Legend at 0x2035f0752b0>
```


Control Glicerol 0,2% E2R2
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 E2R2


Fase exponencial OD E2R2
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
700
Tiempo(min)
OD control E2R2

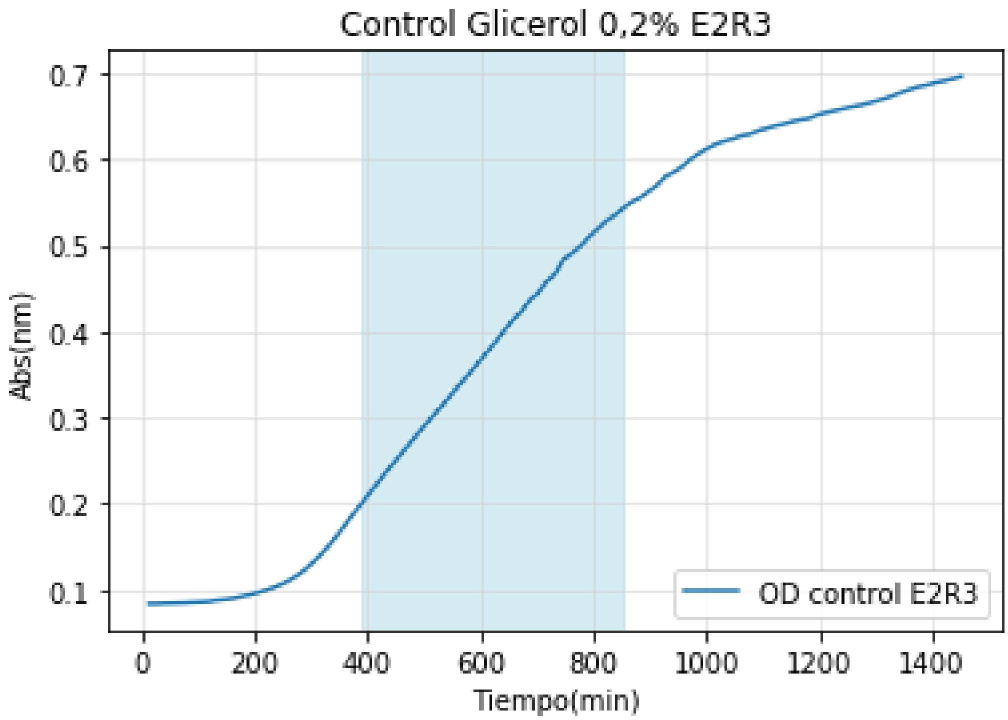
In [11]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerol rep 3
y6= np.log(odcgl153)-np.log(np.min(odcgl153))
print('Min OD = %e'%((np.min(odcgl153))))
evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 0,2% E2R3')
A6= params[0]
um6=params[1]
l6=params[2]
print('A=%e'%(A6))
print('um=%e'%(um6))
print('l=%e'%(l6))

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

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

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

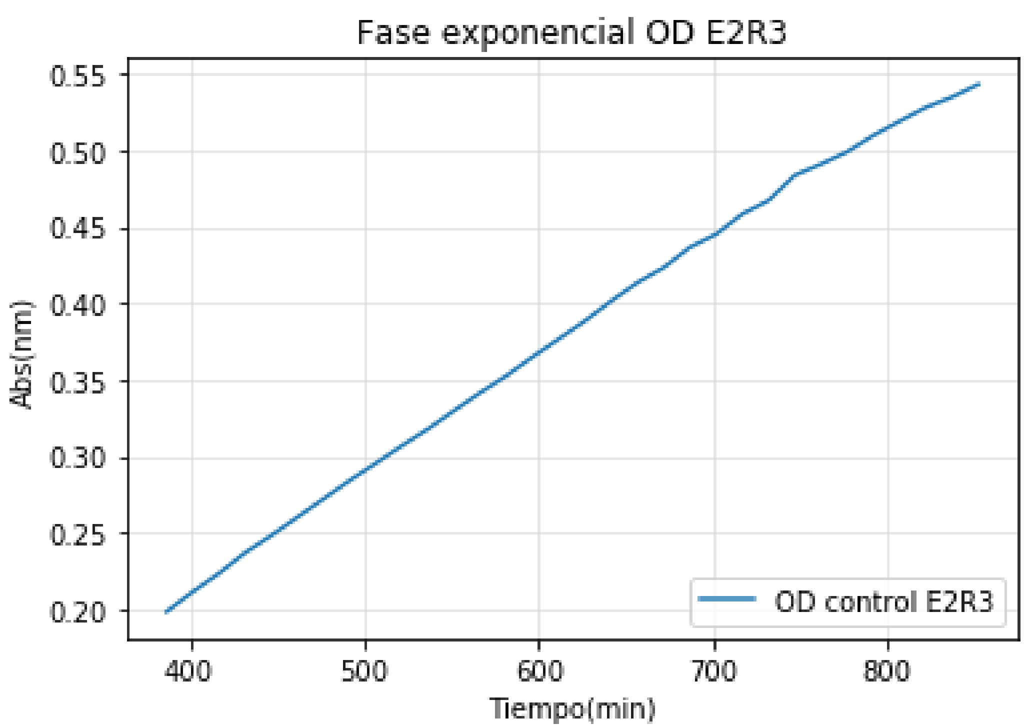
```
Min OD = 8.375000e-02
[  2.07943812e+00   3.85150055e-03   1.79490460e+02]
```




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

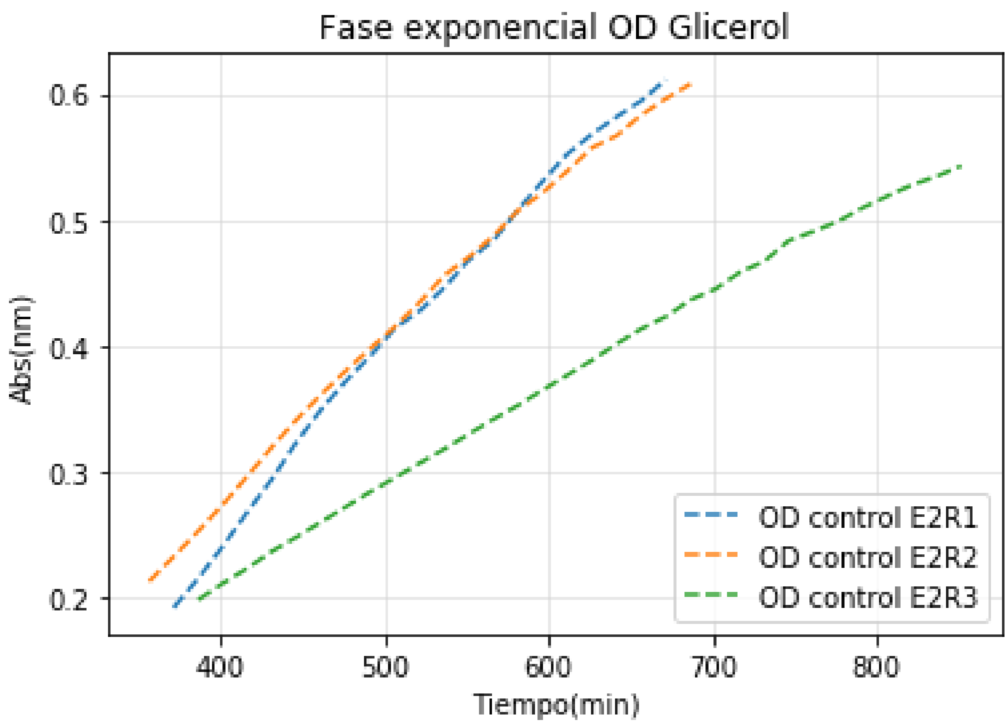
```
Out[11]: <matplotlib.legend.Legend at 0x2035f0feb00>
```


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


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

```
In [12]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glicerol')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[24:45],odcgl151[24:45],'--',label='OD control E2R1')
         plt.plot(tt[23:46],odcgl152[23:46],'--',label='OD control E2R2')
         plt.plot(tt[25:57],odcgl153[25:57],'--',label='OD control E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

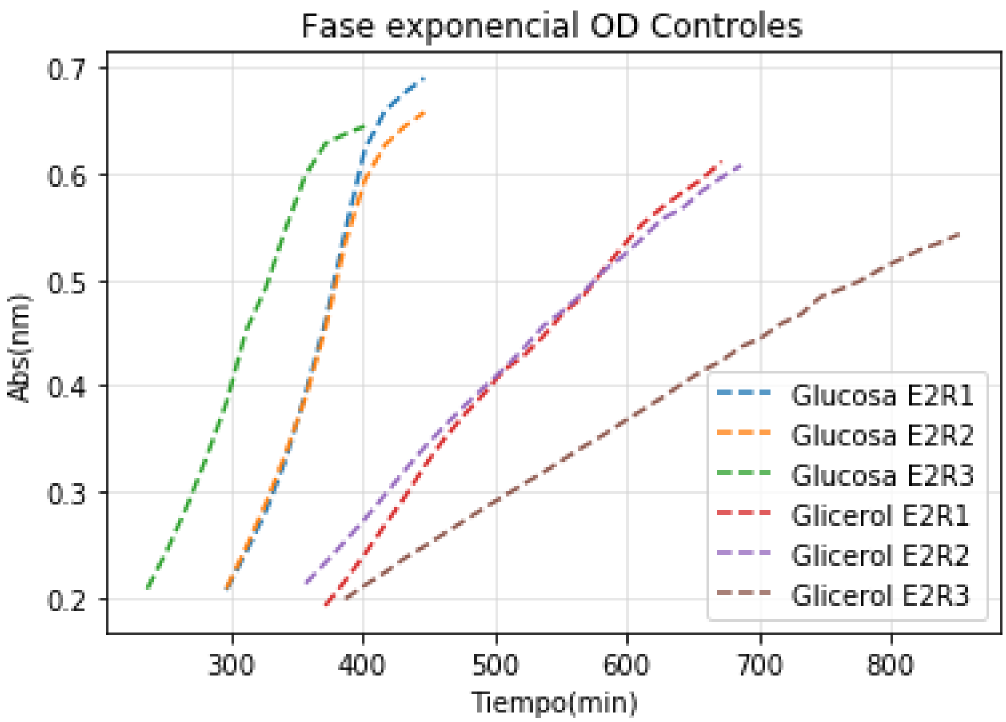
```
Out[12]: <matplotlib.legend.Legend at 0x2035f5dddd8>
```




```
In [13]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Controles')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:30],odcg151[19:30],'--',label='Glucosa E2R1')
         plt.plot(tt[19:30],odcg152[19:30],'--',label='Glucosa E2R2')
         plt.plot(tt[15:27],odcg153[15:27],'--',label='Glucosa E2R3')
         plt.plot(tt[24:45],odcgl151[24:45],'--',label='Glicerol E2R1')
         plt.plot(tt[23:46],odcgl152[23:46],'--',label='Glicerol E2R2')
         plt.plot(tt[25:57],odcgl153[25:57],'--',label='Glicerol E2R3')
```

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

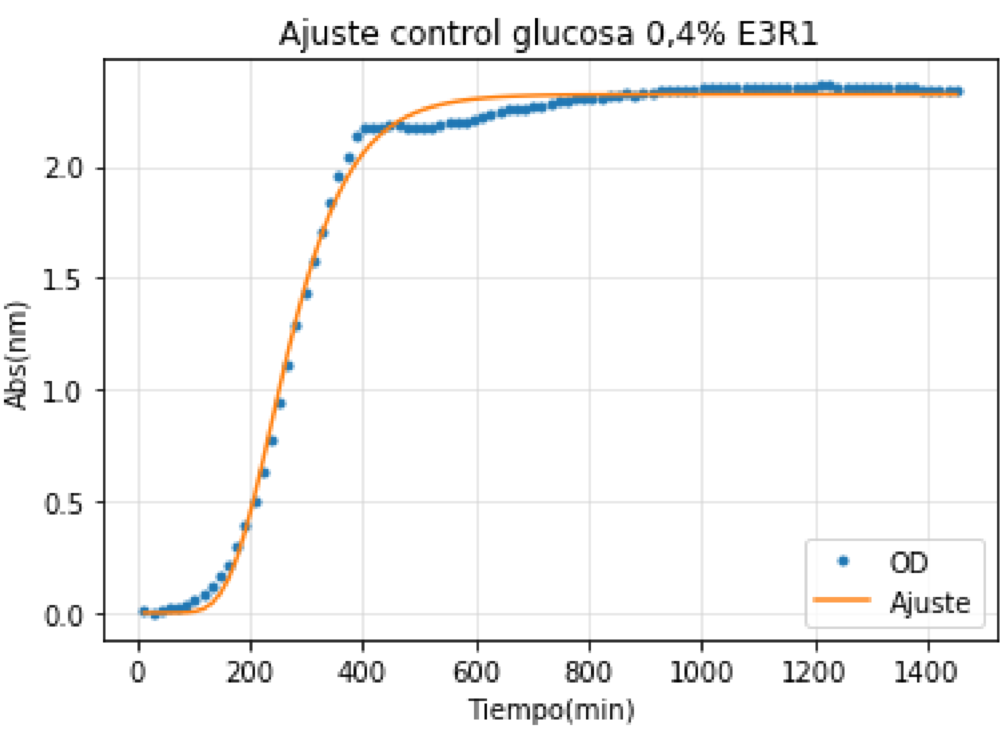
Out[13]: <matplotlib.legend.Legend at 0x2035f698b38>




```
In [14]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glucosa rep 1
         y7 = np.log(odcg181)-np.log(np.min(odcg181))
         print('Min OD = %e'%((np.min(odcg181))))
         evaly, params=Function_fit(tt,y7,0,-1,title = 'Ajuste control glucosa 0,4% E3R1')
         A7 = params[0]
         um7=params[1]
         l7=params[2]
         print('A=%e'%(A7))
         print('um=%e'%(um7))
         print('l=%e'%(l7))

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

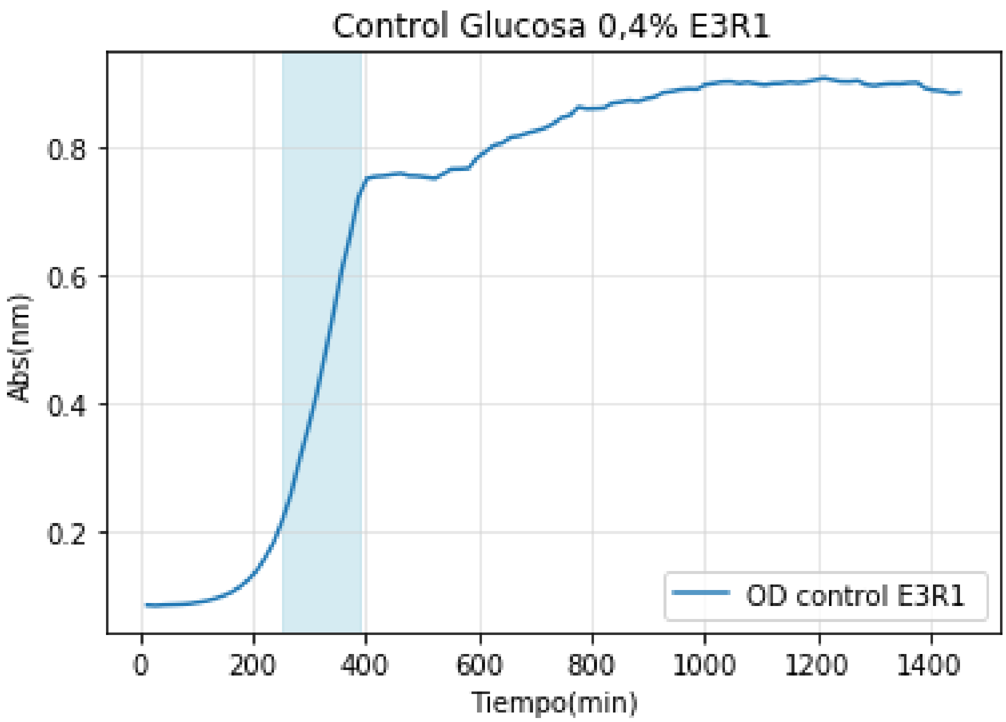
```python
ttot7=tm7+extdp7
print('Tfinal=%e'%ttot7)

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

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

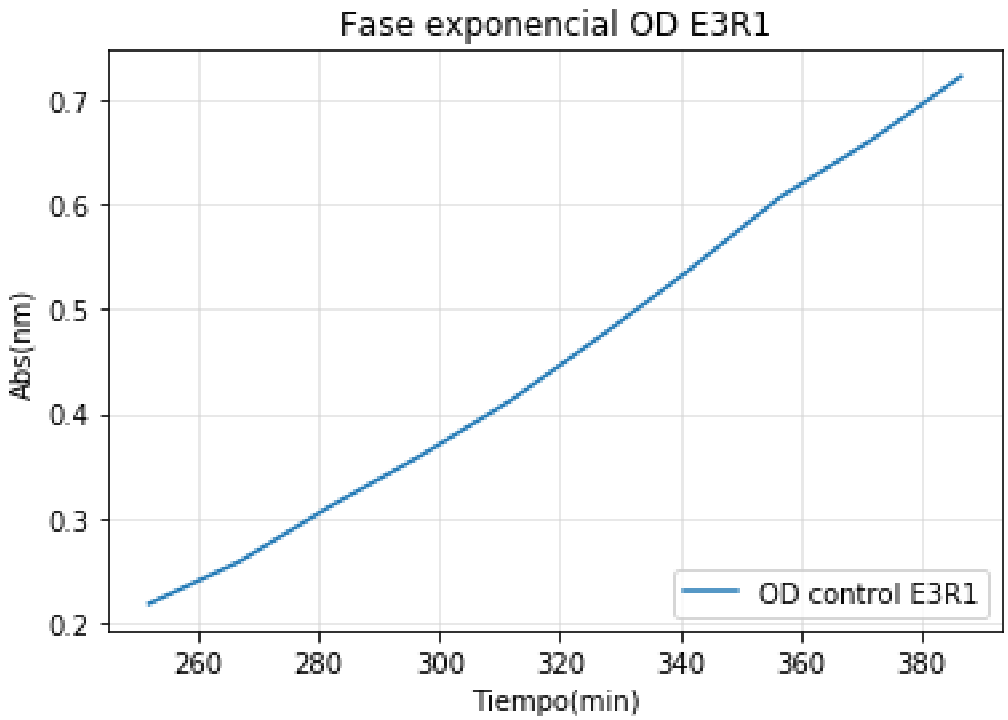
```
Min OD = 8.550000e-02
[  2.32234986e+00   1.11487857e-02   1.61538943e+02]
```



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

Out[14]: <matplotlib.legend.Legend at 0x2035efb9b38>

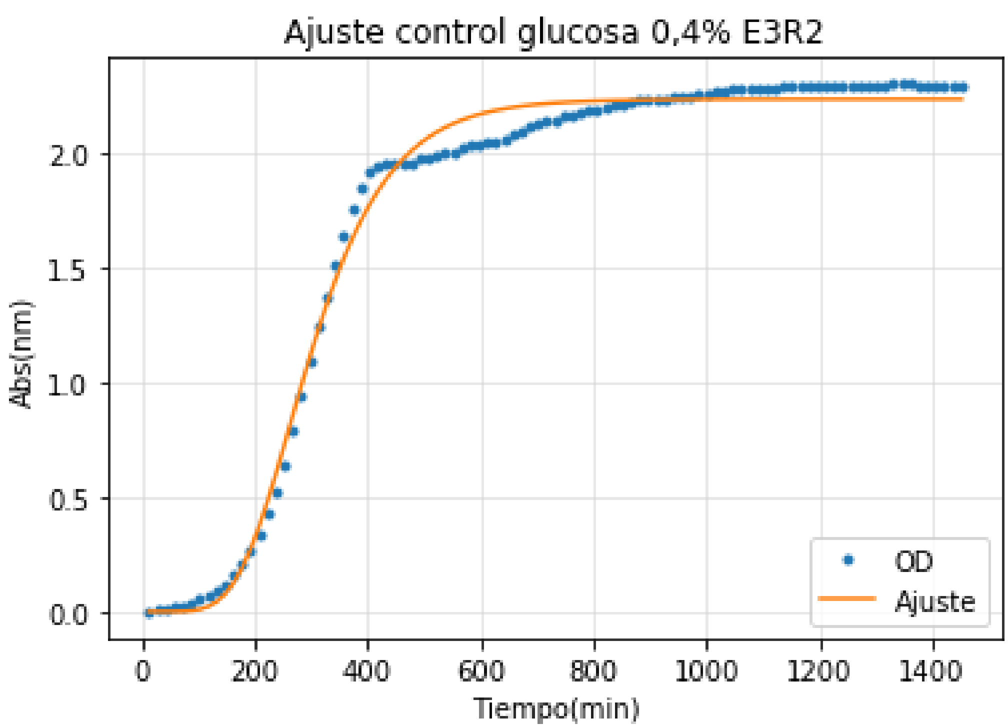




In [15]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y8= np.log(odcg182)-np.log(np.min(odcg182))
print('Min OD = %e'%((np.min(odcg182))))
evaly, params=Function_fit(tt,y8,0,-1, title = 'Ajuste control glucosa 0,4% E3R2')
A8 = params[0]
um8=params[1]
l8=params[2]
print('A=%e'%(A8))
print('um=%e'%(um8))
print('l=%e'%(l8))

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

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

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

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

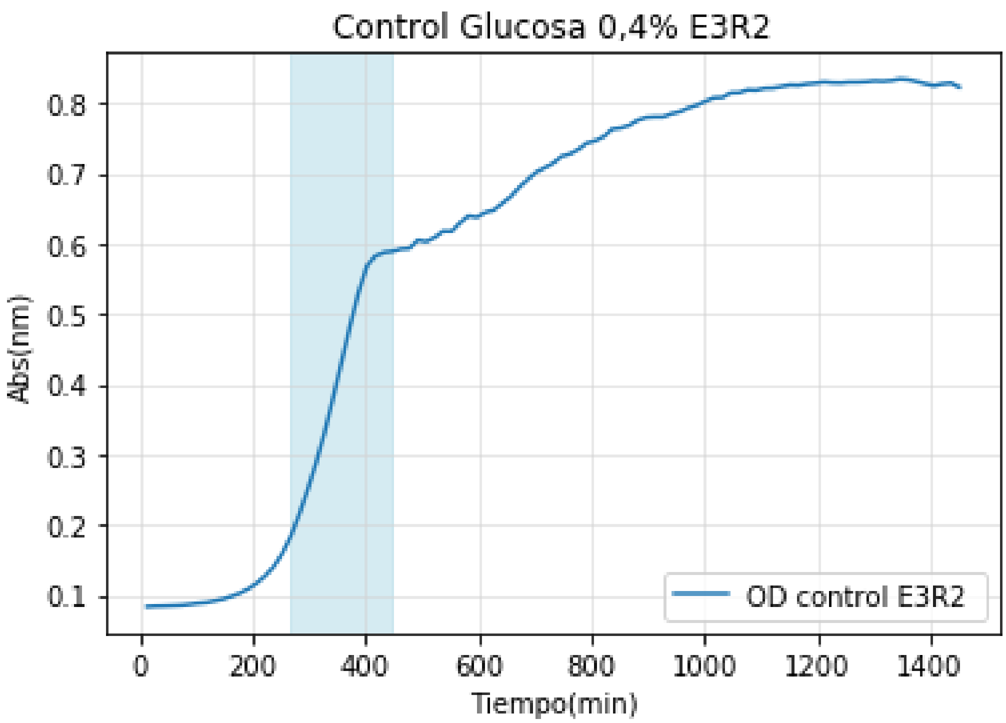
```
Min OD = 8.375000e-02
[  2.23365370e+00   8.60171922e-03   1.67065878e+02]
```

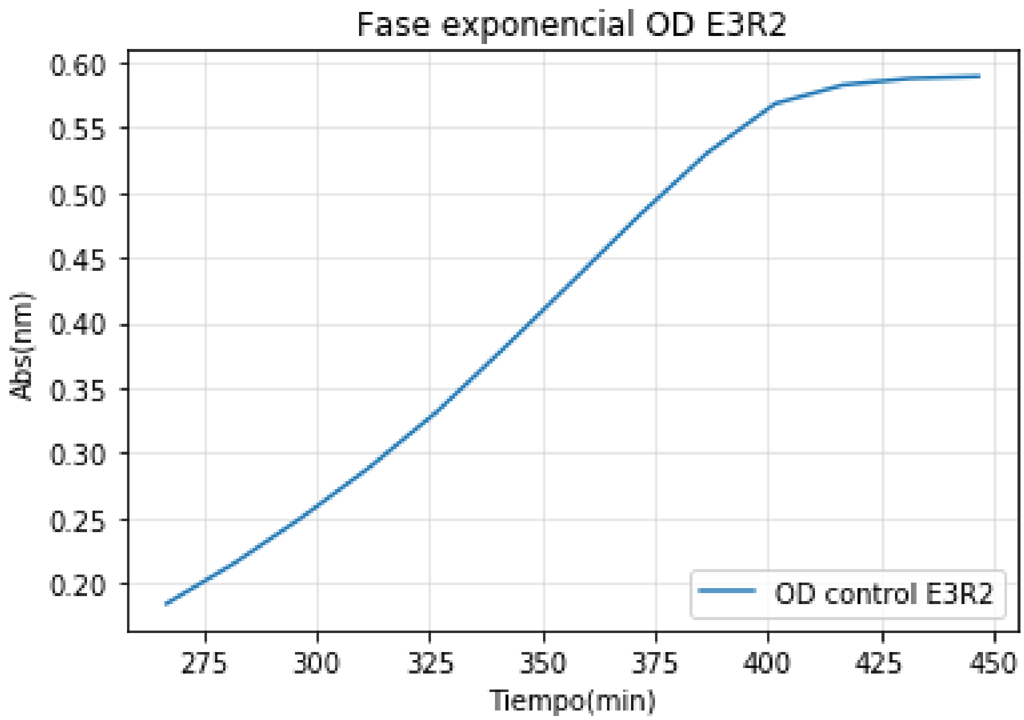



```
A=2.233654e+00
um=8.601719e-03
```

```
l=1.670659e+02
Tm=2.625951e+02
doubpe=8.058240e+01
ext=1.611648e+02
Tfinal=4.237599e+02
```

```
Out[15]: <matplotlib.legend.Legend at 0x2035f1ab240>
```

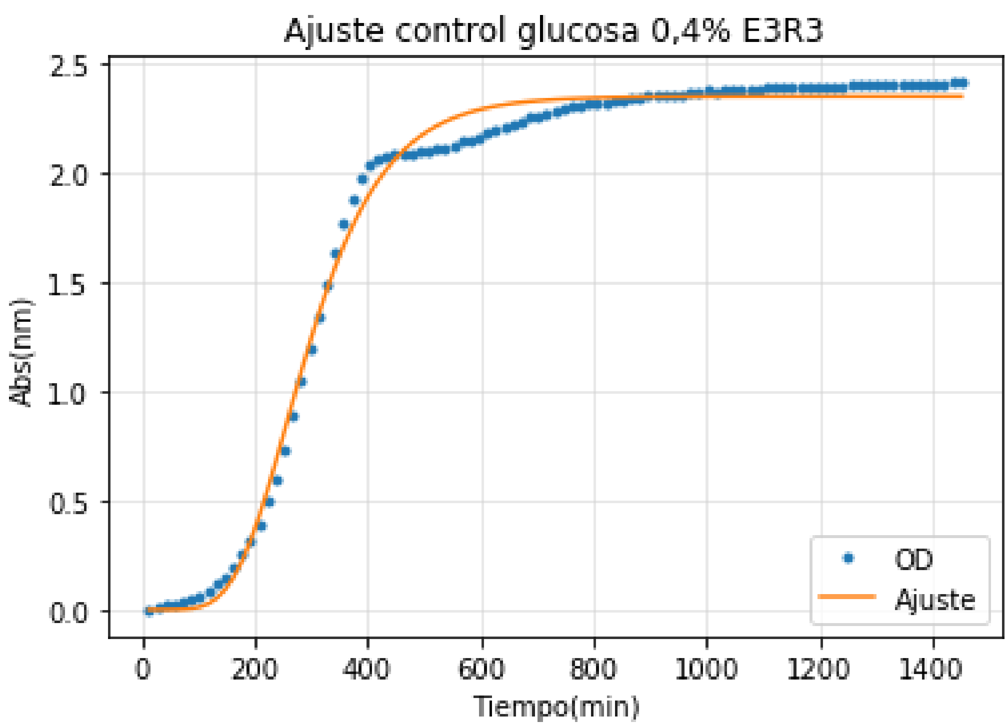




In [16]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y9= np.log(odcg183)-np.log(np.min(odcg183))
print('Min OD = %e'%((np.min(odcg183))))
evaly, params=Function_fit(tt,y9,0,-1, title = 'Ajuste control glucosa 0,4% E3R3')
A9= params[0]
um9=params[1]
l9=params[2]
print('A=%e'%(A9))
print('um=%e'%(um9))
print('l=%e'%(l9))

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


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

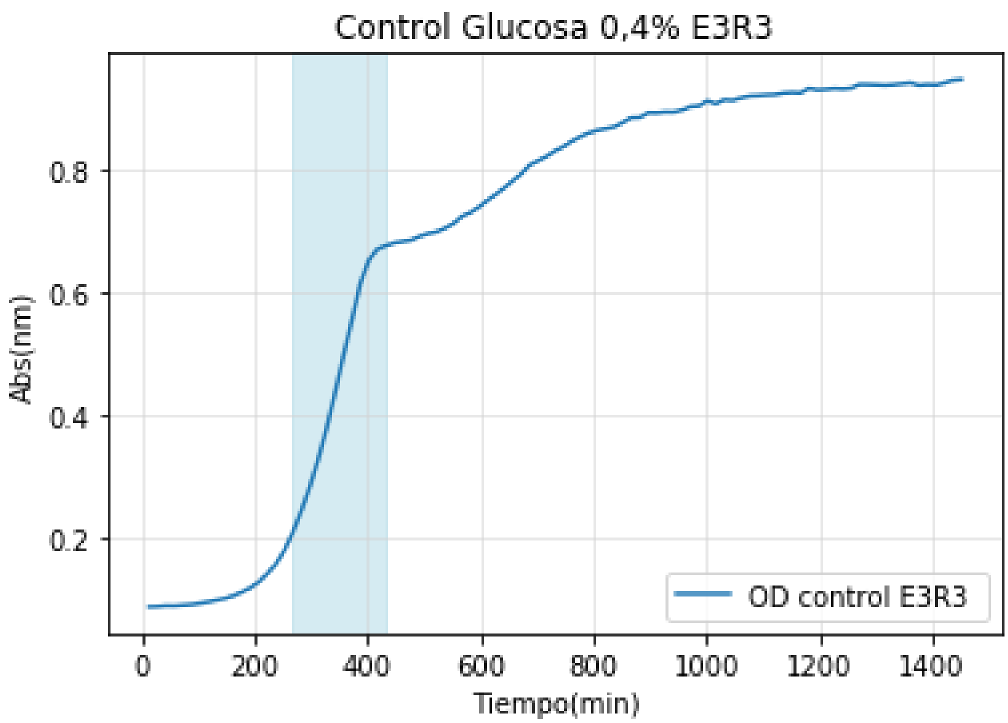
```python
y1=tt[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg183,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower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

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

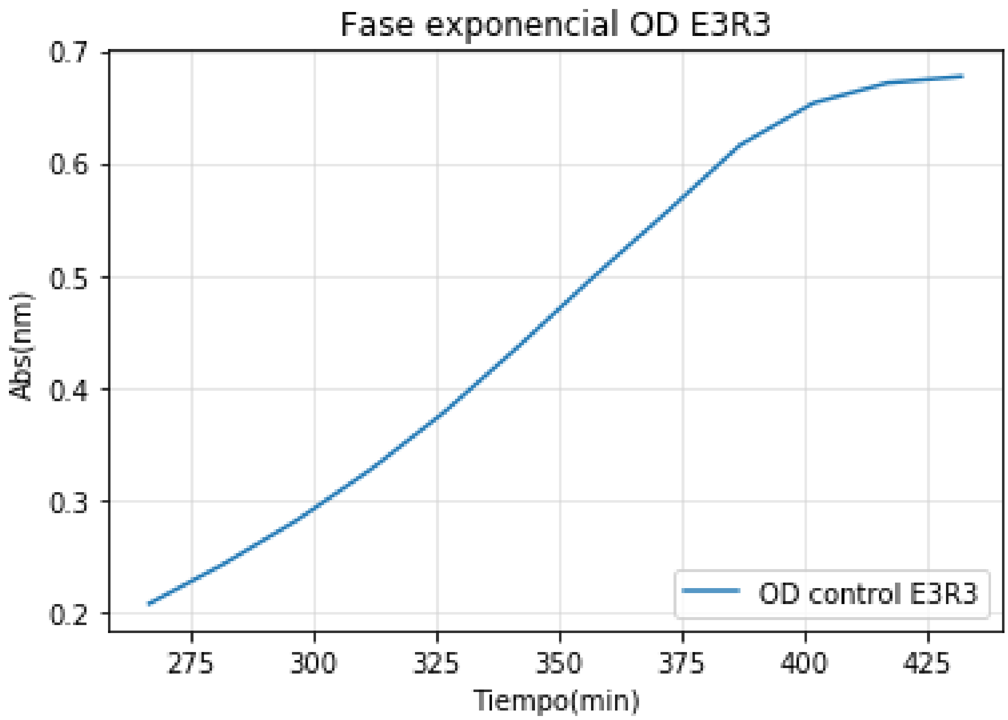
```
Min OD = 8.525000e-02
[  2.34758891e+00   9.20947423e-03   1.63131184e+02]
```



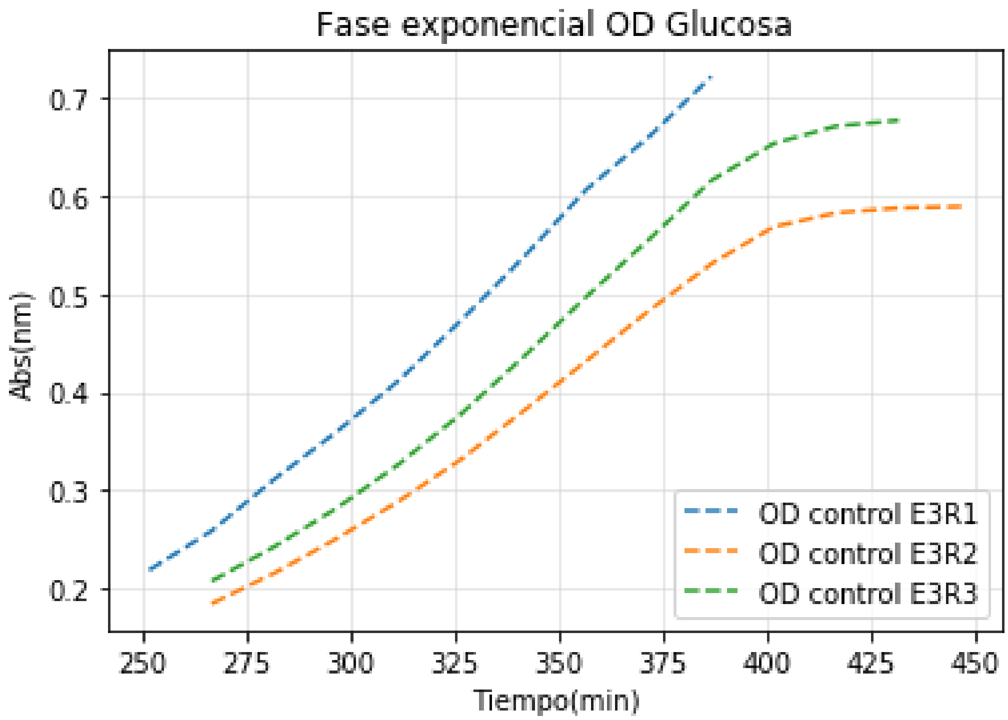
```
A=2.347589e+00
um=9.209474e-03
l=1.631312e+02
Tm=2.569074e+02
doubpe=7.526458e+01
ext=1.505292e+02
Tfinal=4.074366e+02
```

```
Out[16]: <matplotlib.legend.Legend at 0x2035f831128>
```





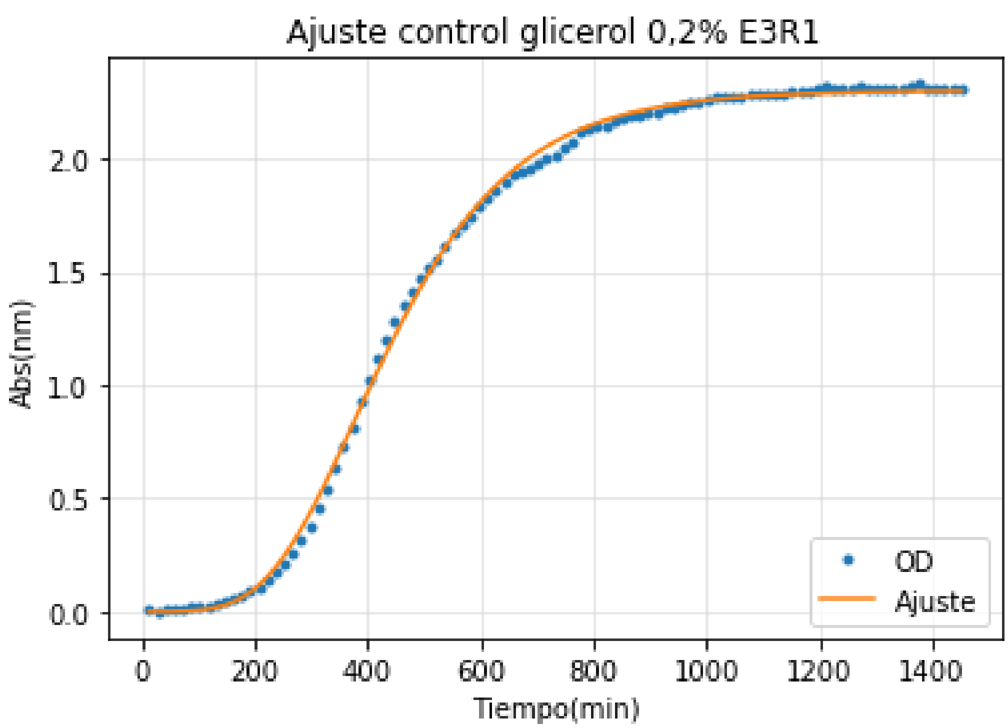

```
In [17]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glucosa')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],odcg181[16:26],'--',label='OD control E3R1')
         plt.plot(tt[17:30],odcg182[17:30],'--',label='OD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

Out[17]: <matplotlib.legend.Legend at 0x2035f945be0>
```



In [18]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 1
y10= np.log(odcgl181)-np.log(np.min(odcgl181))
print('Min OD = %e'%((np.min(odcgl181))))
evaly, params=Function_fit(tt,y10,0,-1, title = 'Ajuste control glicerol 0,2% E3R1')
A10= params[0]
um10=params[1]
l10=params[2]
print('A=%e'%(A10))
print('um=%e'%(um10))
print('l=%e'%(l10))

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

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

```
y2=tt[47]
plt.figure()
plt.title('Control Glicerol 0,2% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl181,label='OD control E3R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:48],odcgl181[25:48],label='OD control E3R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

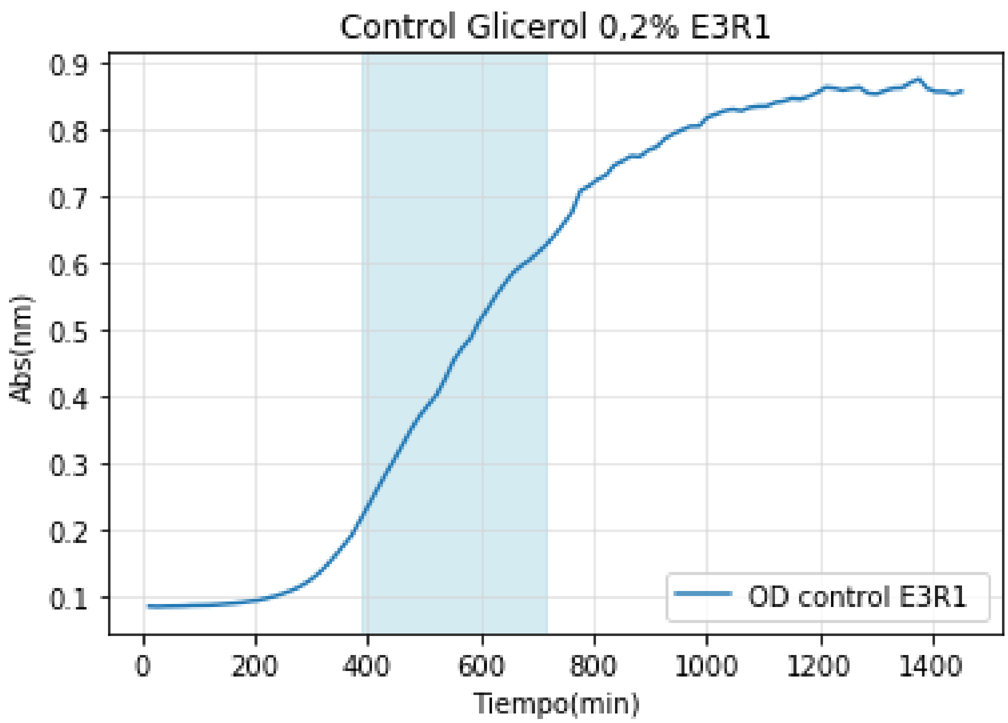
```
Min OD = 8.525000e-02
[  2.29993135e+00   5.42425453e-03   2.20607778e+02]
```

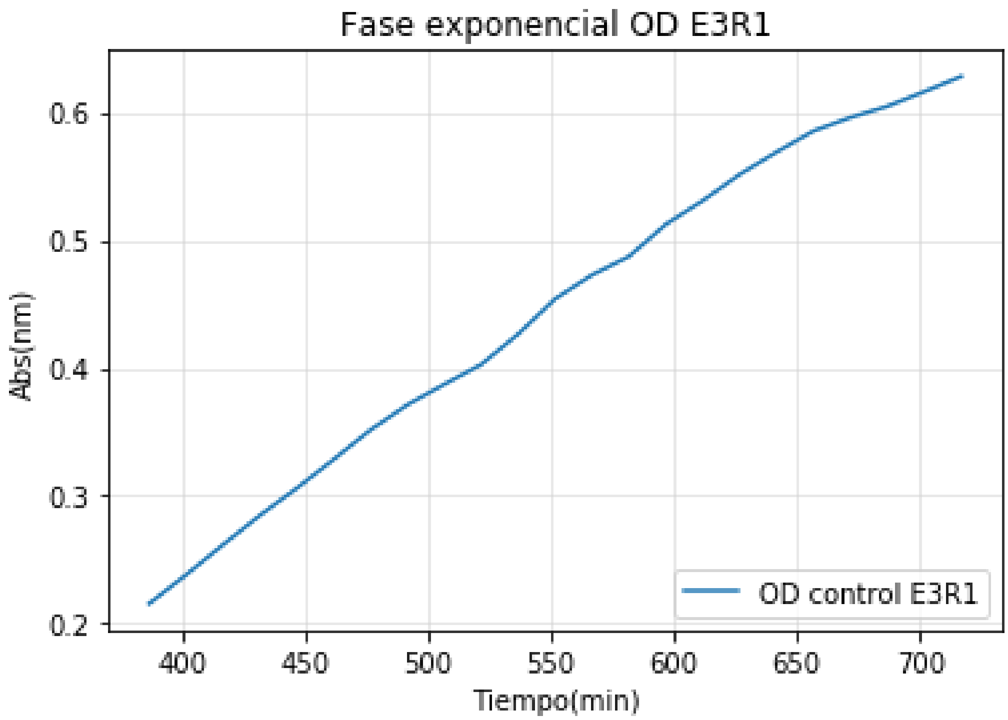



```
A=2.299931e+00
um=5.424255e-03
```

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

Out[18]: `<matplotlib.legend.Legend at 0x2035fae0630>`

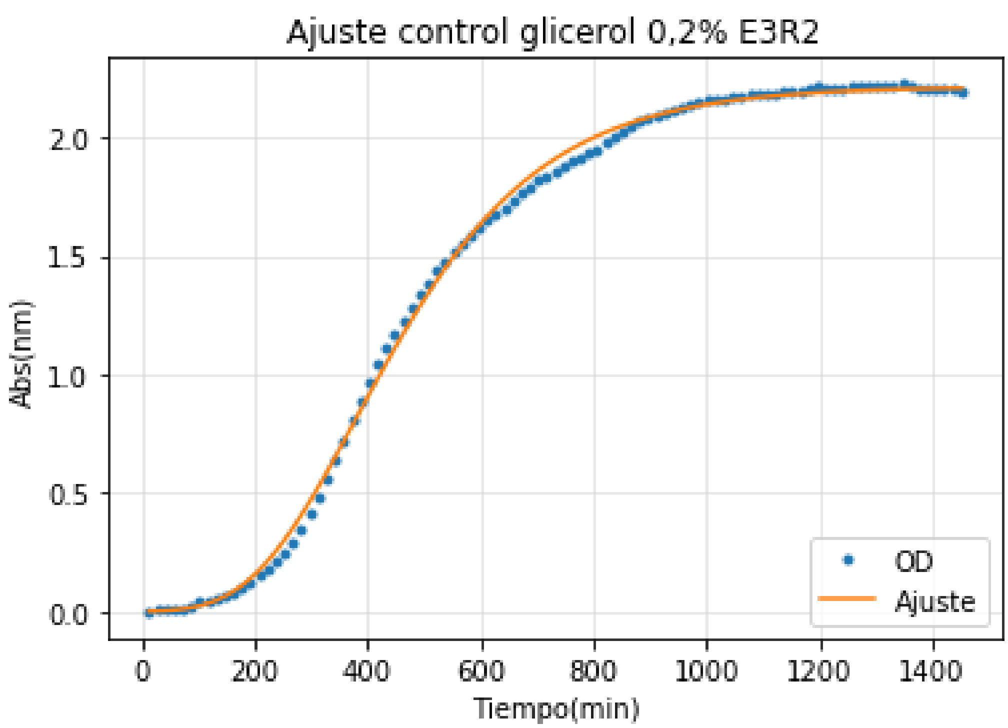





```
In [19]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 2
         y11= np.log(odcgl182)-np.log(np.min(odcgl182))
         print('Min OD = %e'%((np.min(odcgl182))))
         evaly, params=Function_fit(tt,y11,0,-1, title = 'Ajuste control glicerol 0,2% E3R2')
         A11= params[0]
         um11=params[1]
         l11=params[2]
         print('A=%e'%(A11))
         print('um=%e'%(um11))
         print('l=%e'%(l11))

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

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

```python
y2=tt[52]
plt.figure()
plt.title('Control Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl182,label='OD control E3R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],odcgl182[25:53],label='OD control E3R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

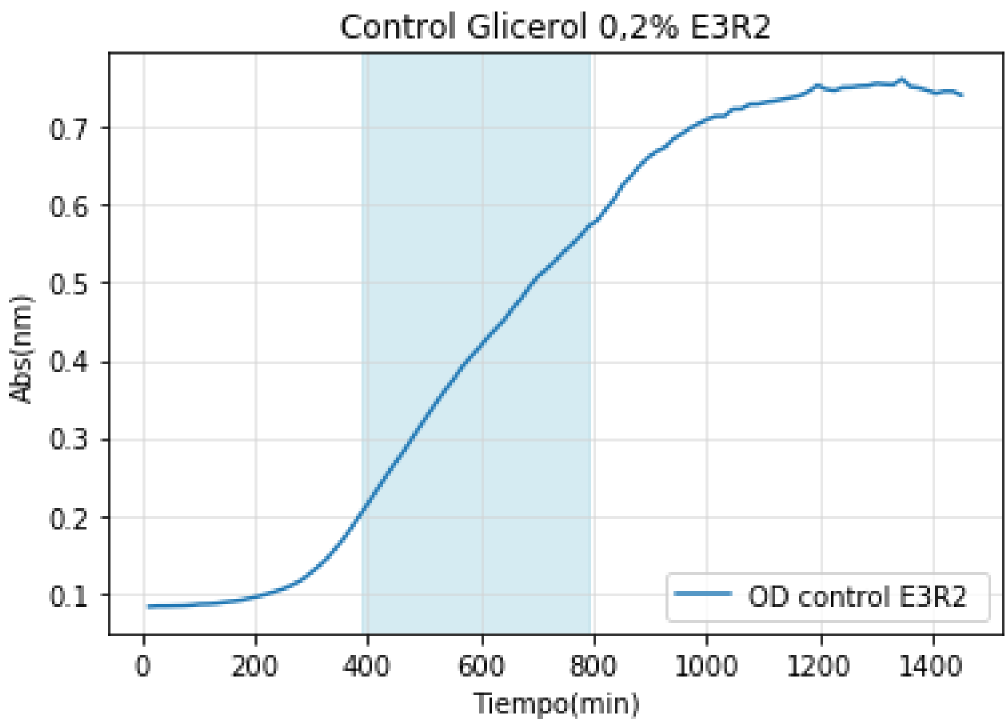
```
Min OD = 8.250000e-02
[  2.21283483e+00   4.40822880e-03   1.94027274e+02]
```

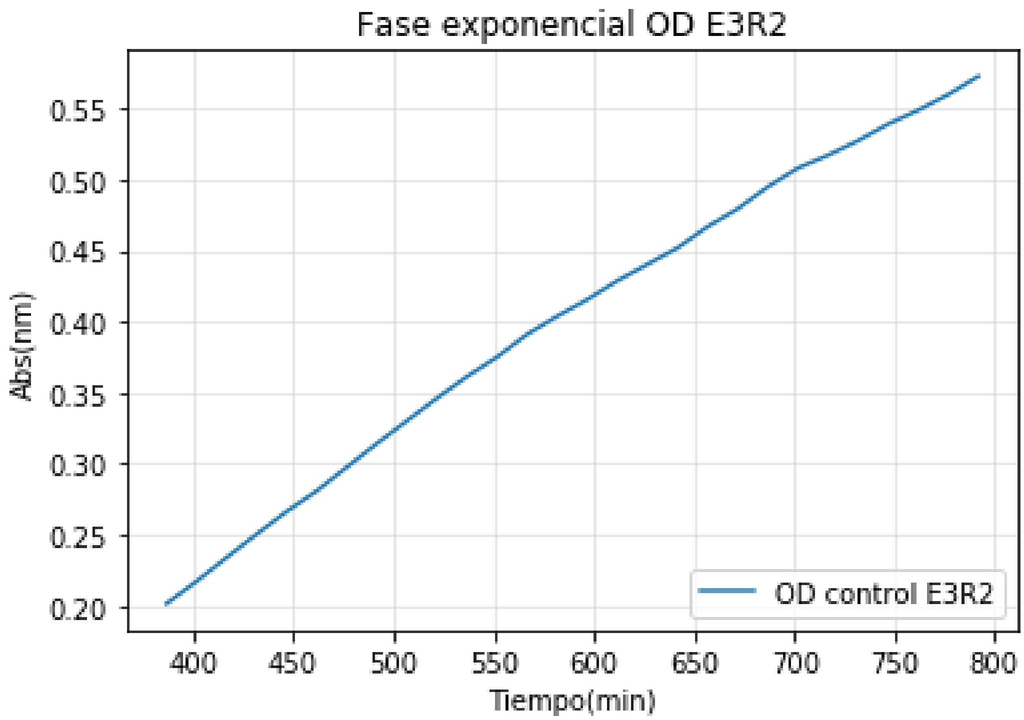



```
A=2.212835e+00
um=4.408229e-03
```

```
l=1.940273e+02
Tm=3.786947e+02
doubpe=1.572394e+02
ext=3.930985e+02
Tfinal=7.717932e+02
```

Out[19]: <matplotlib.legend.Legend at 0x2035f197b70>

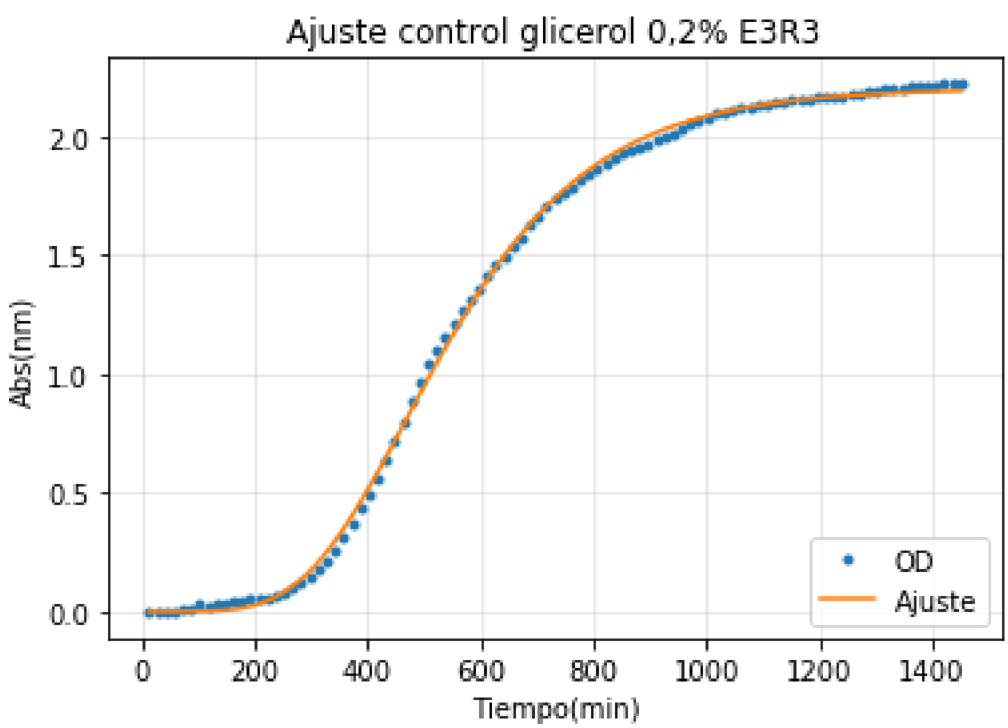




In [20]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 3
y12= np.log(odcgl183)-np.log(np.min(odcgl183))
print('Min OD = %e'%((np.min(odcgl183))))
evaly, params=Function_fit(tt,y12,0,-1, title = 'Ajuste control glicerol 0,2% E3R3')
A12= params[0]
um12=params[1]
l12=params[2]
print('A=%e'%(A12))
print('um=%e'%(um12))
print('l=%e'%(l12))

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

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

```python
y2=tt[50]
plt.figure()
plt.title('Control Glicerol 0,2% E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl183,label='OD control E3R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E3R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:51],odcgl183[24:51],label='OD control E3R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

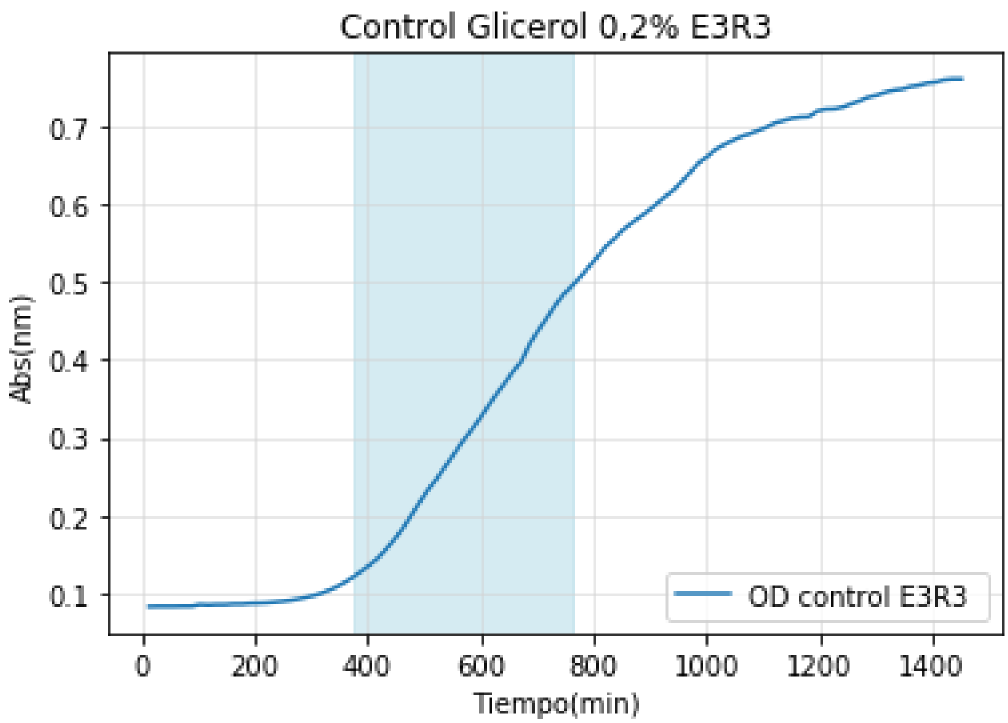
```
Min OD = 8.275000e-02
[  2.19895432e+00   4.44585642e-03   2.85665525e+02]
```

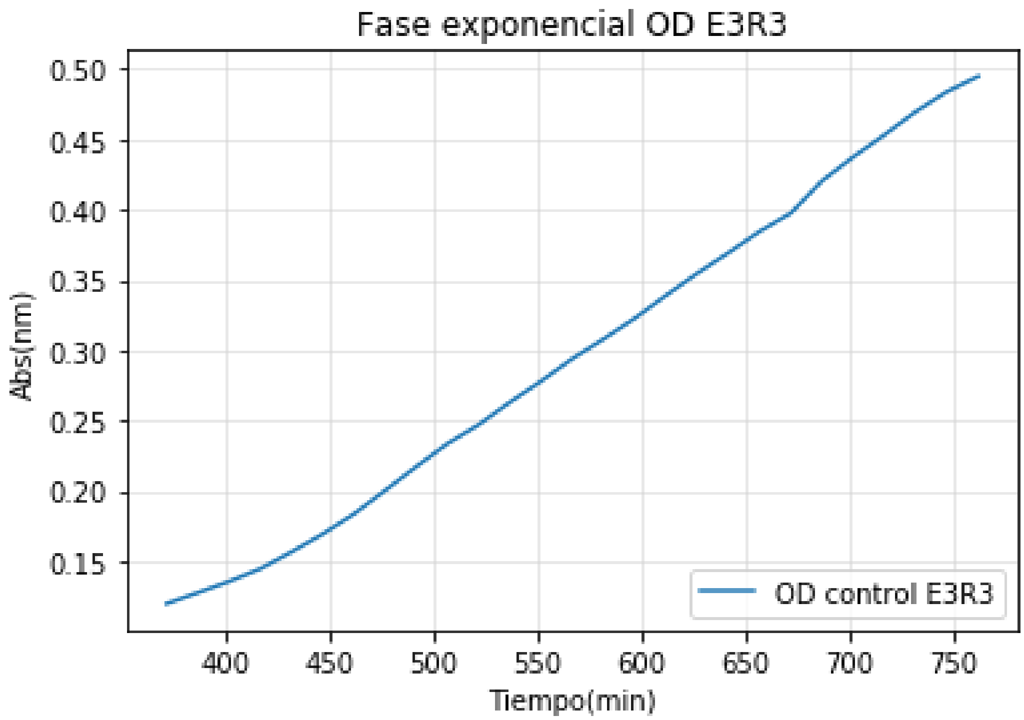



```
A=2.198954e+00
um=4.445856e-03
```

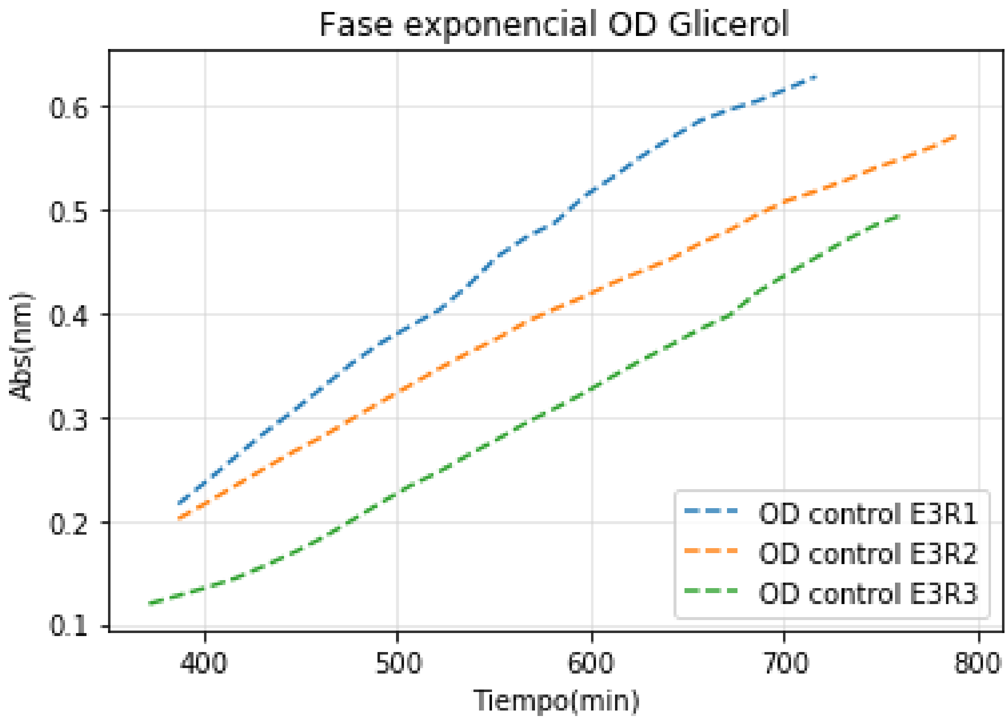
```
l=2.856655e+02
Tm=4.676215e+02
doubpe=1.559086e+02
ext=3.897715e+02
Tfinal=8.573929e+02
```

Out[20]: <matplotlib.legend.Legend at 0x2035f9ee908>

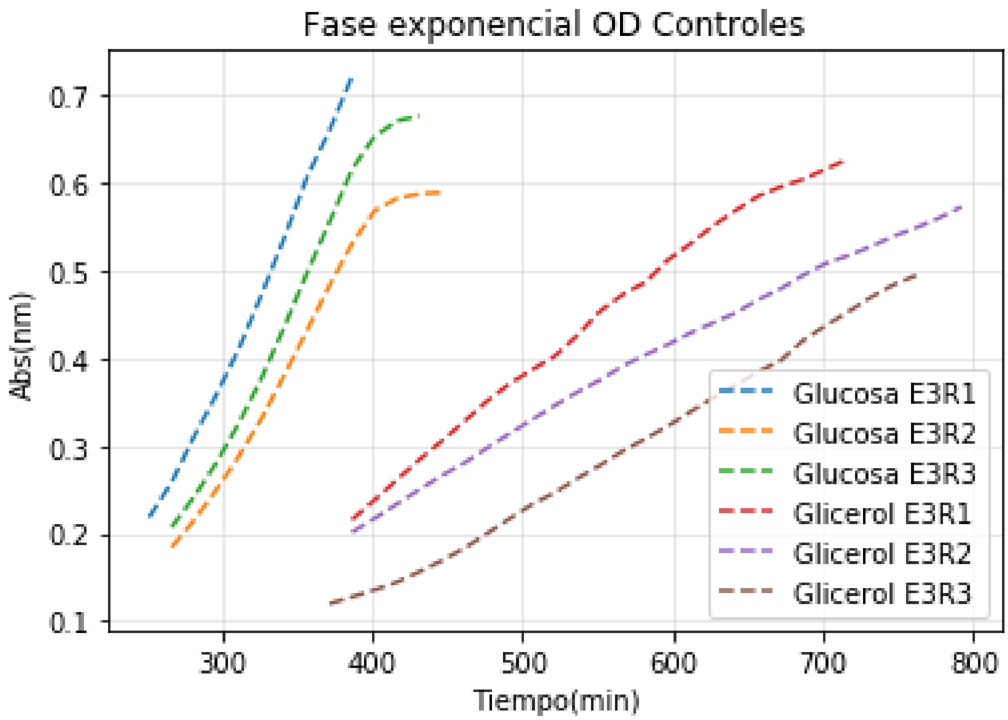





```
In [21]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glicerol')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[25:48],odcgl181[25:48],'--',label='OD control E3R1')
         plt.plot(tt[25:53],odcgl182[25:53],'--',label='OD control E3R2')
         plt.plot(tt[24:51],odcgl183[24:51],'--',label='OD control E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[21]: <matplotlib.legend.Legend at 0x2035fc1e6d8>
```



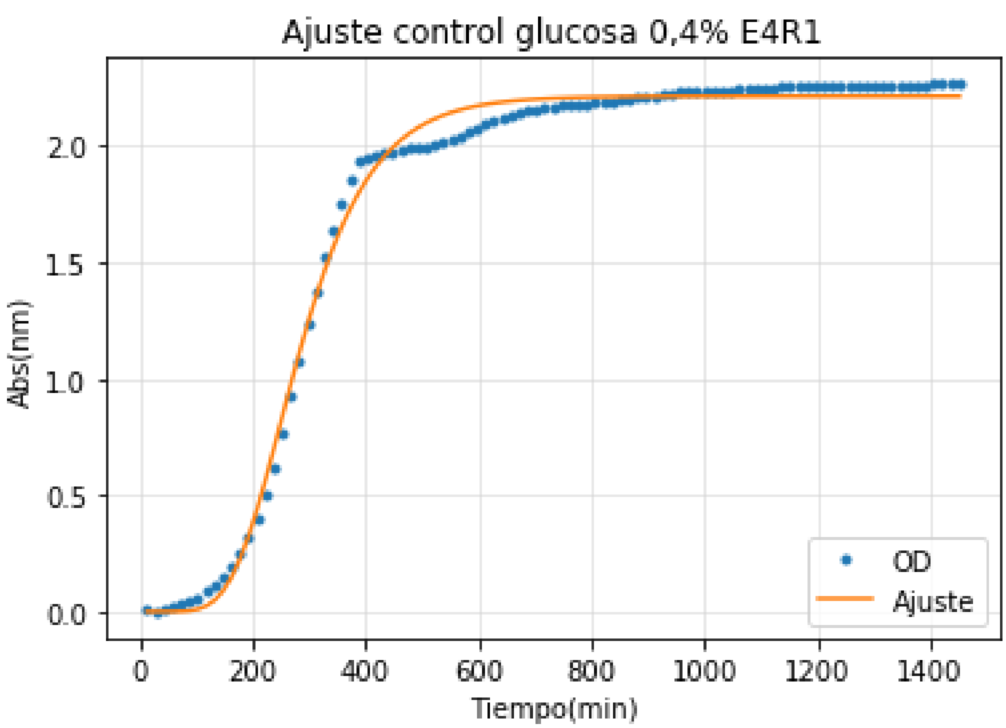

```
In [22]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Controles')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],odcg181[16:26],'--',label='Glucosa E3R1')
         plt.plot(tt[17:30],odcg182[17:30],'--',label='Glucosa E3R2')
         plt.plot(tt[17:29],odcg183[17:29],'--',label='Glucosa E3R3')
         plt.plot(tt[25:48],odcgl181[25:48],'--',label='Glicerol E3R1')
         plt.plot(tt[25:53],odcgl182[25:53],'--',label='Glicerol E3R2')
         plt.plot(tt[24:51],odcgl183[24:51],'--',label='Glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[22]: <matplotlib.legend.Legend at 0x2035fcd9c18>
```



In [23]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 1
y13 = np.log(odcg121)-np.log(np.min(odcg121))
print('Min OD = %e'%((np.min(odcg121))))
evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste control glucosa 0,4% E4R1')
A13 = params[0]
um13=params[1]
l13=params[2]
print('A=%e'%(A13))
print('um=%e'%(um13))
print('l=%e'%(l13))

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

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

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

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

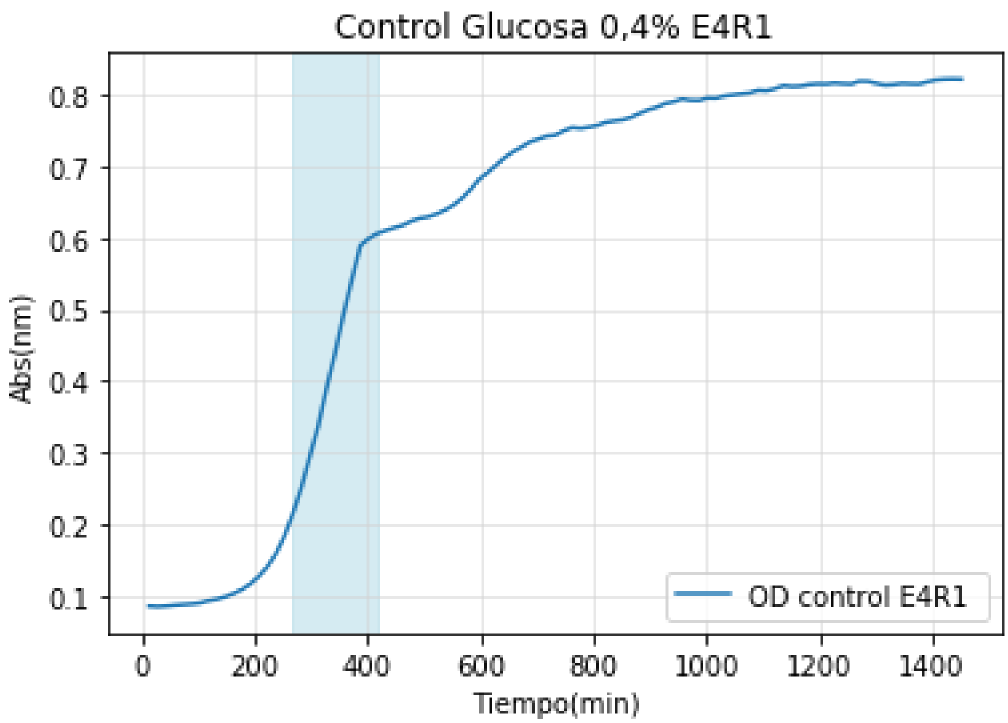
```
Min OD = 8.525000e-02
[  2.21400213e+00   9.27035018e-03   1.61117814e+02]
```

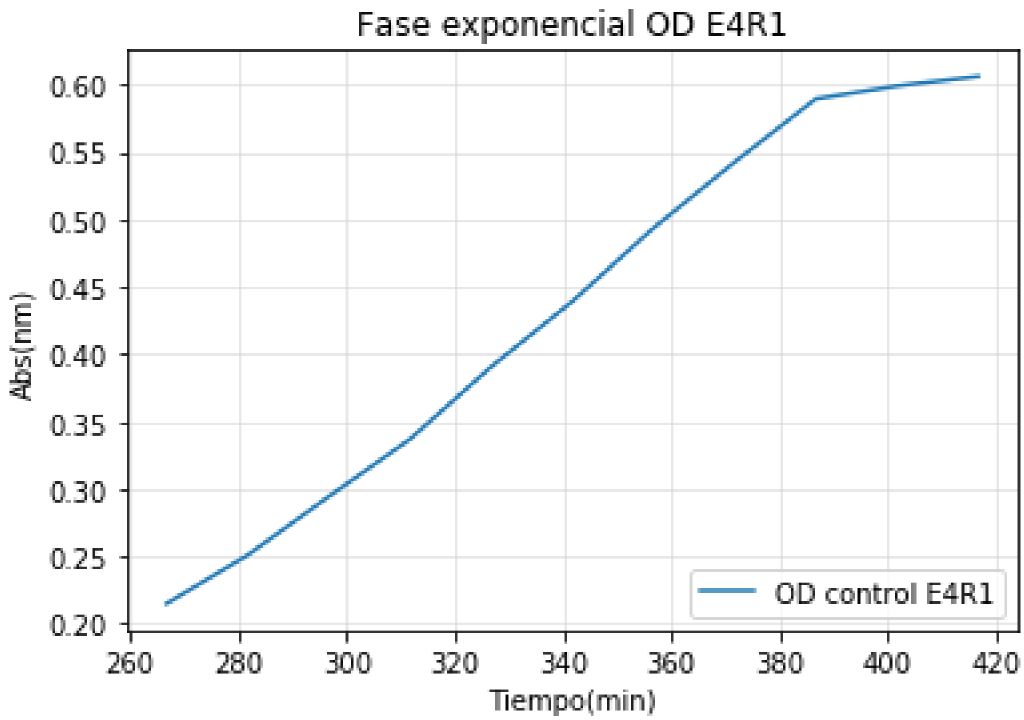



```
A=2.214002e+00
um=9.270350e-03
```

```
l=1.611178e+02
Tm=2.489770e+02
doubpe=7.477033e+01
ext=1.495407e+02
Tfinal=3.985177e+02
```

```
Out[23]: <matplotlib.legend.Legend at 0x2035fb56518>
```

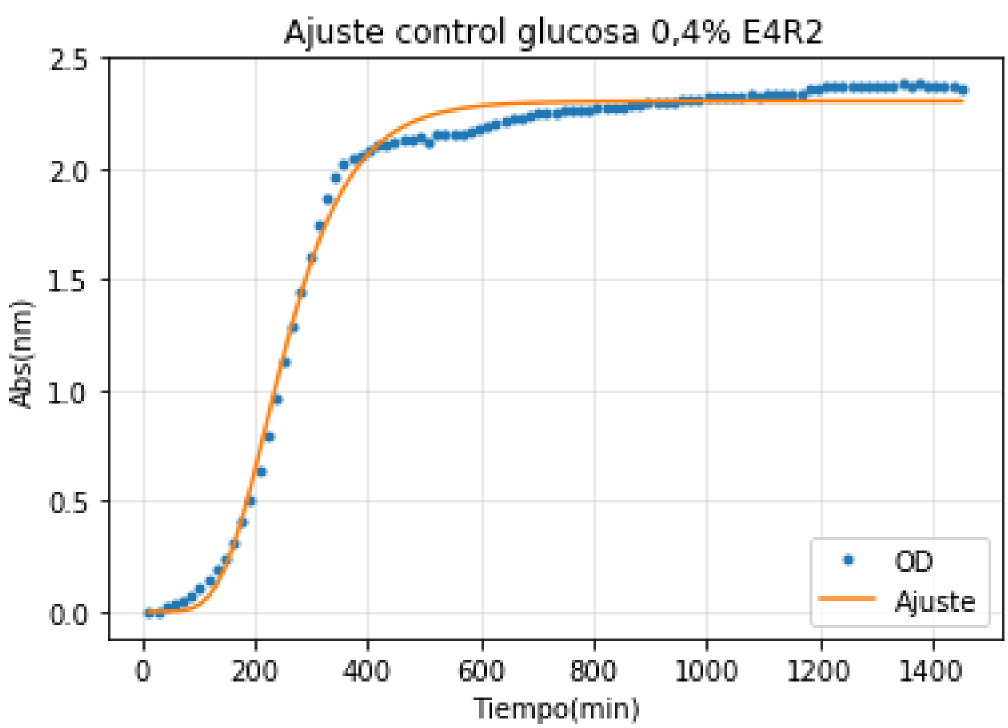




In [24]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y14= np.log(odcg122)-np.log(np.min(odcg122))
print('Min OD = %e'%((np.min(odcg122))))
evaly, params=Function_fit(tt,y14,0,-1, title = 'Ajuste control glucosa 0,4% E4R2')
A14= params[0]
um14=params[1]
l14=params[2]
print('A=%e'%(A14))
print('um=%e'%(um14))
print('l=%e'%(l14))

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

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

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

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

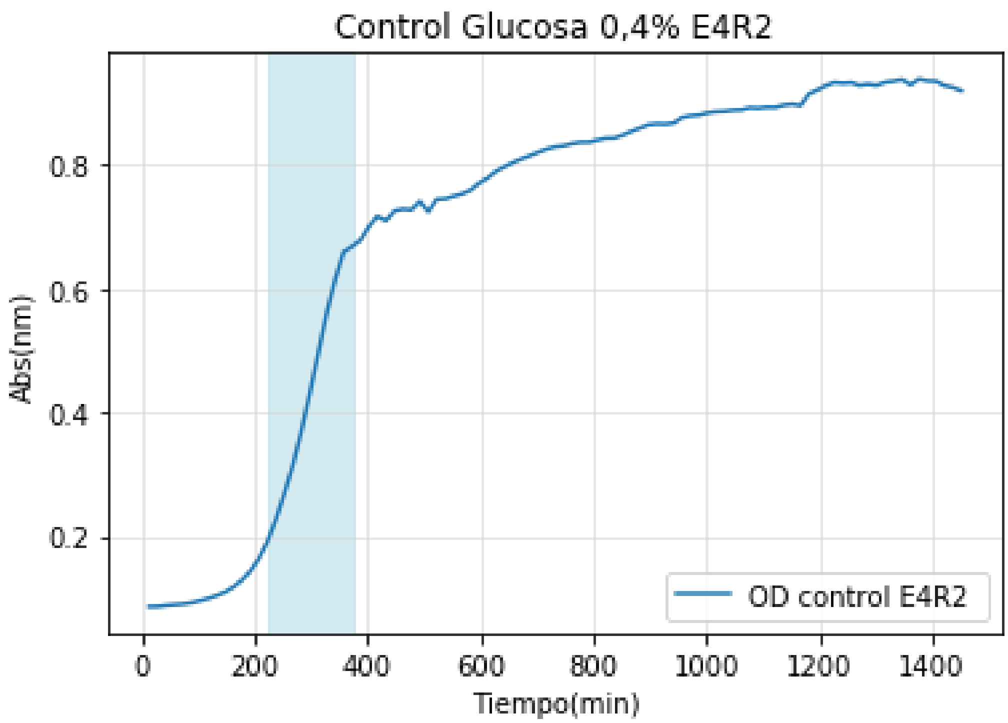
```
Min OD = 8.675000e-02
[  2.30555639e+00   1.03788860e-02   1.38136011e+02]
```

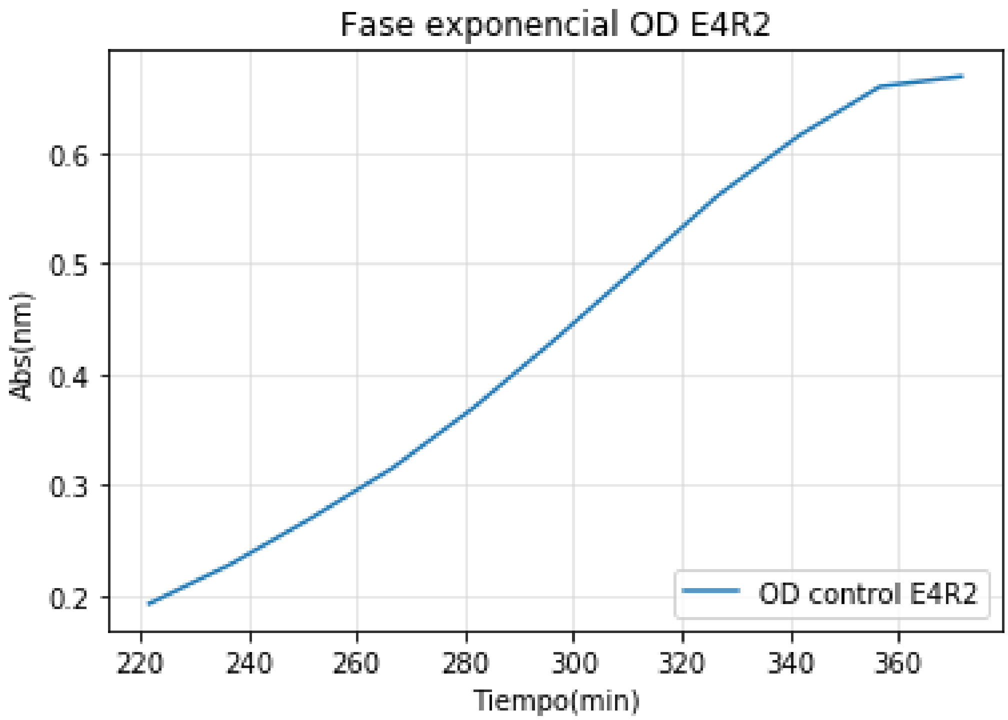



```
A=2.305556e+00
um=1.037889e-02
```

```
l=1.381360e+02
Tm=2.198564e+02
doubpe=6.678435e+01
ext=1.335687e+02
Tfinal=3.534251e+02
```

Out[24]: <matplotlib.legend.Legend at 0x2035dec2240>

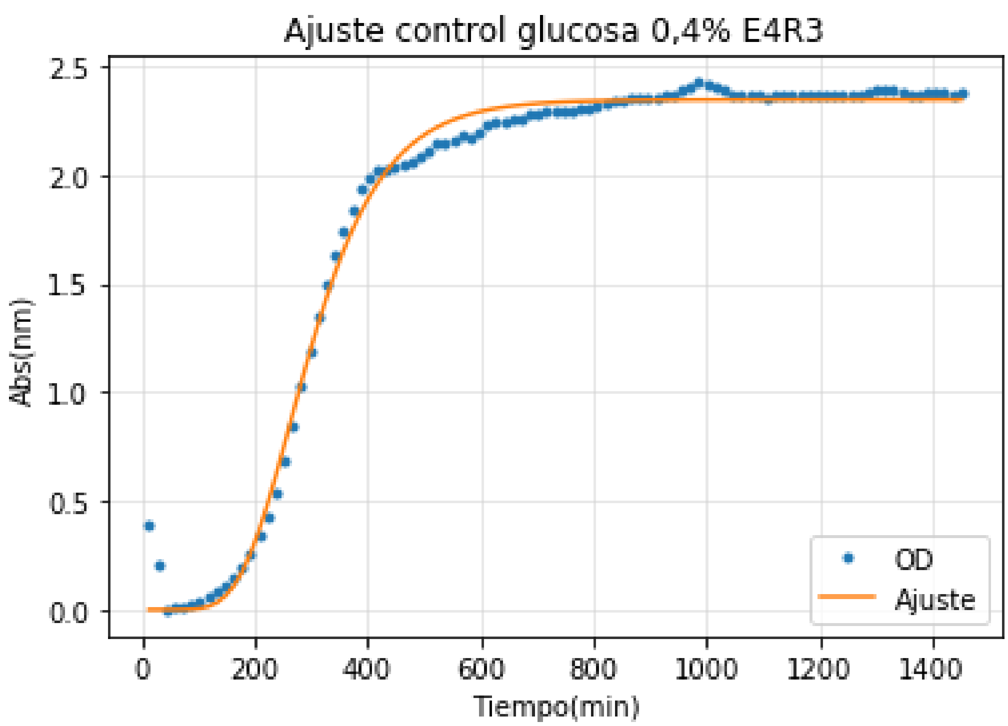




In [25]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y15= np.log(odcg123)-np.log(np.min(odcg123))
print('Min OD = %e'%((np.min(odcg123))))
evaly, params=Function_fit(tt,y15,0,-1, title = 'Ajuste control glucosa 0,4% E4R3')
A15= params[0]
um15=params[1]
l15=params[2]
print('A=%e'%(A15))
print('um=%e'%(um15))
print('l=%e'%(l15))

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


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

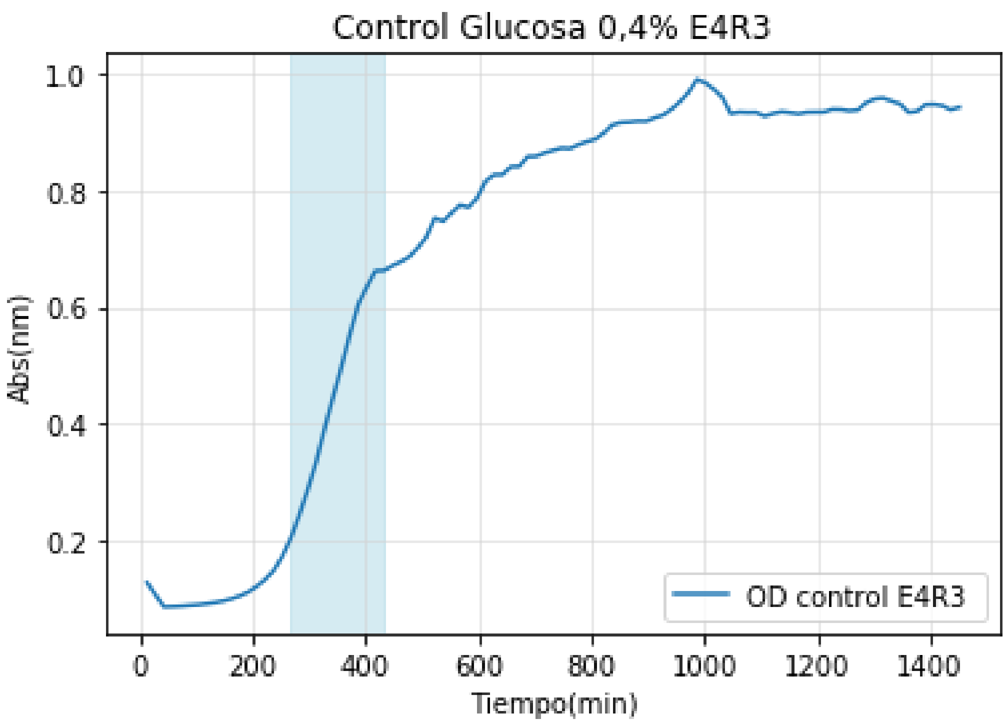
```
y1=tt[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg123,label='OD control E4R3 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

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

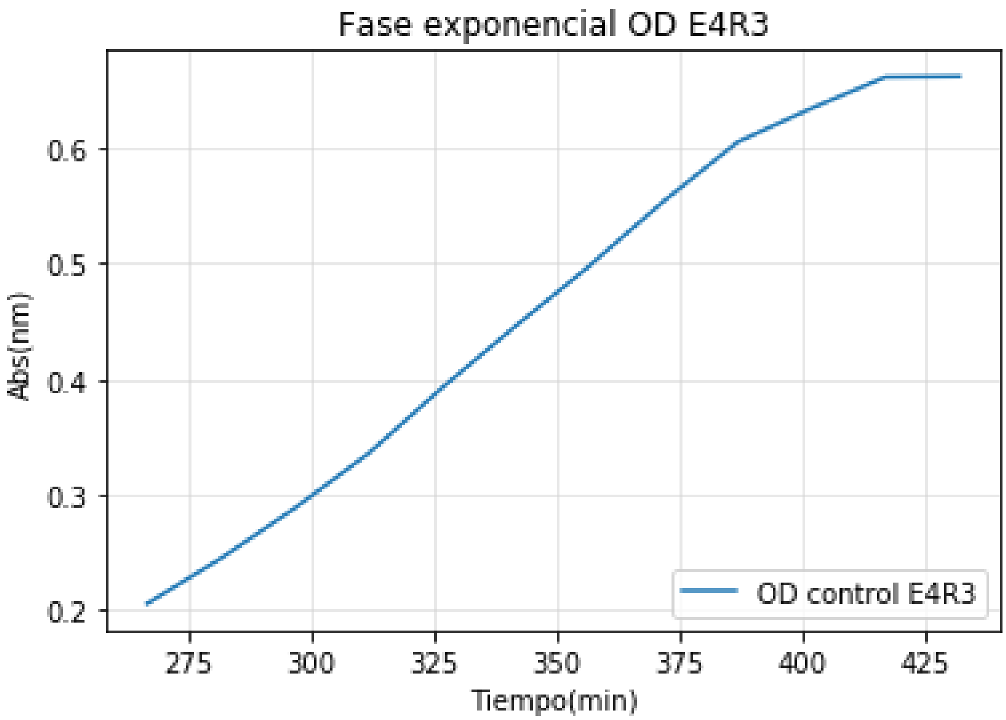
```
Min OD = 8.775000e-02
[  2.34400664e+00   9.59456675e-03   1.72702880e+02]
```



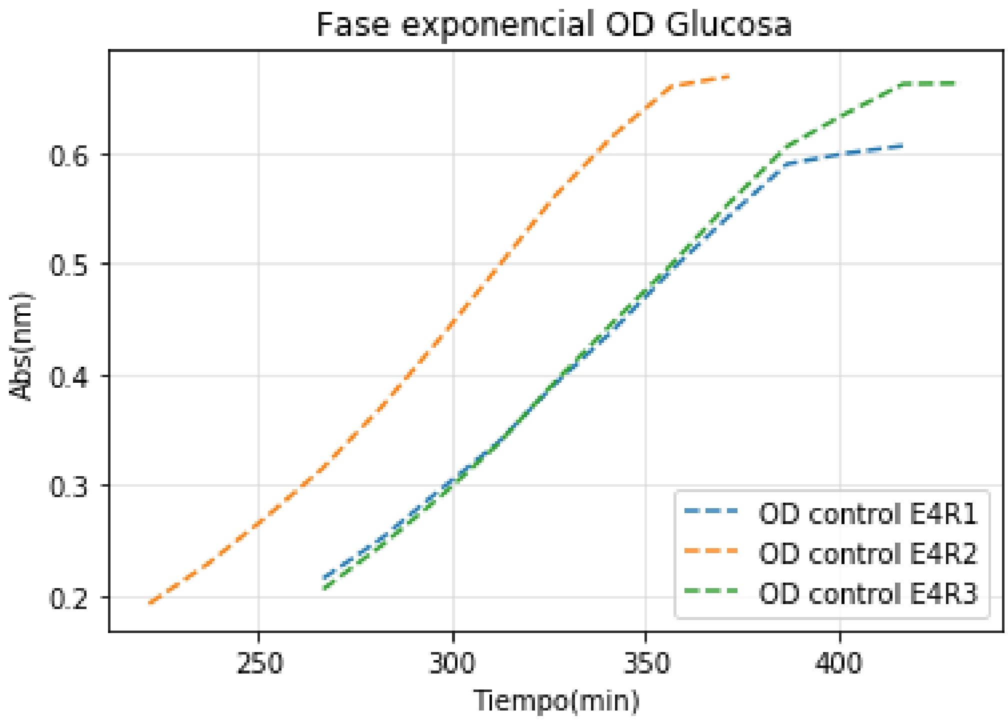
```
A=2.344007e+00
um=9.594567e-03
l=1.727029e+02
Tm=2.625779e+02
doubpe=7.224372e+01
ext=1.444874e+02
Tfinal=4.070653e+02
```

Out[25]: `<matplotlib.legend.Legend at 0x20360e43c88>`





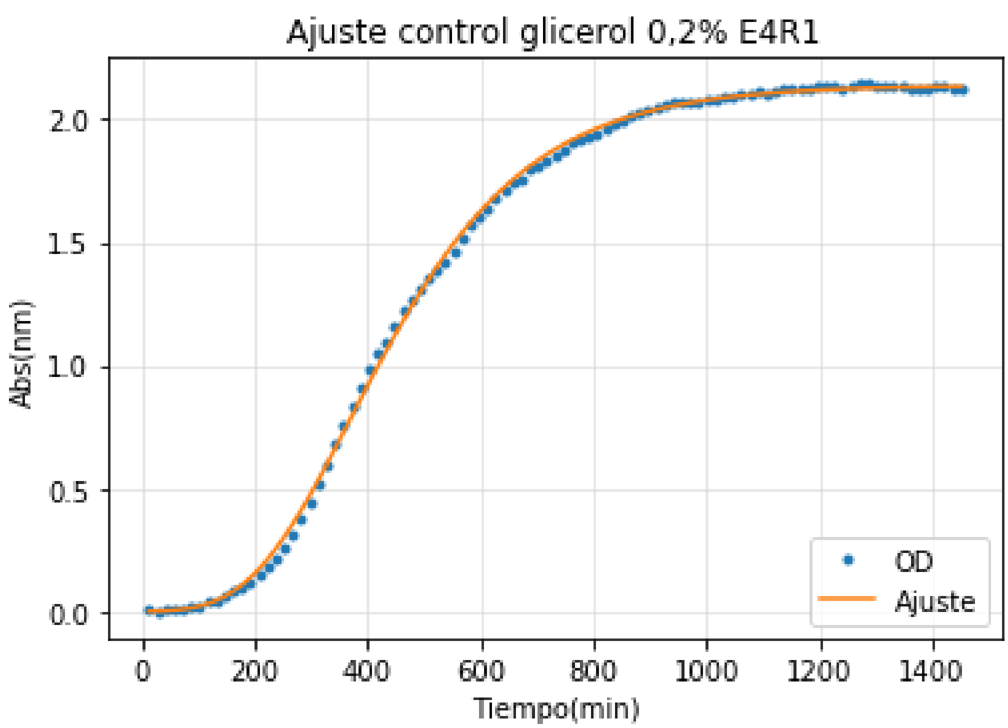

```
In [26]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glucosa')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:28],odcg121[17:28],'--',label='OD control E4R1')
         plt.plot(tt[14:25],odcg122[14:25],'--',label='OD control E4R2')
         plt.plot(tt[17:29],odcg123[17:29],'--',label='OD control E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[26]: <matplotlib.legend.Legend at 0x20360f5fe80>
```




```
In [27]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 1
         y16= np.log(odcgl121)-np.log(np.min(odcgl121))
         print('Min OD = %e'%((np.min(odcgl121))))
         evaly, params=Function_fit(tt,y16,0,-1, title = 'Ajuste control glicerol 0,2% E4R1')
         A16= params[0]
         um16=params[1]
         l16=params[2]
         print('A=%e'%(A16))
         print('um=%e'%(um16))
         print('l=%e'%(l16))

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

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

```python
y2=tt[51]
plt.figure()
plt.title('Control Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl121,label='OD control E4R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[24:52],odcgl121[24:52],label='OD control E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

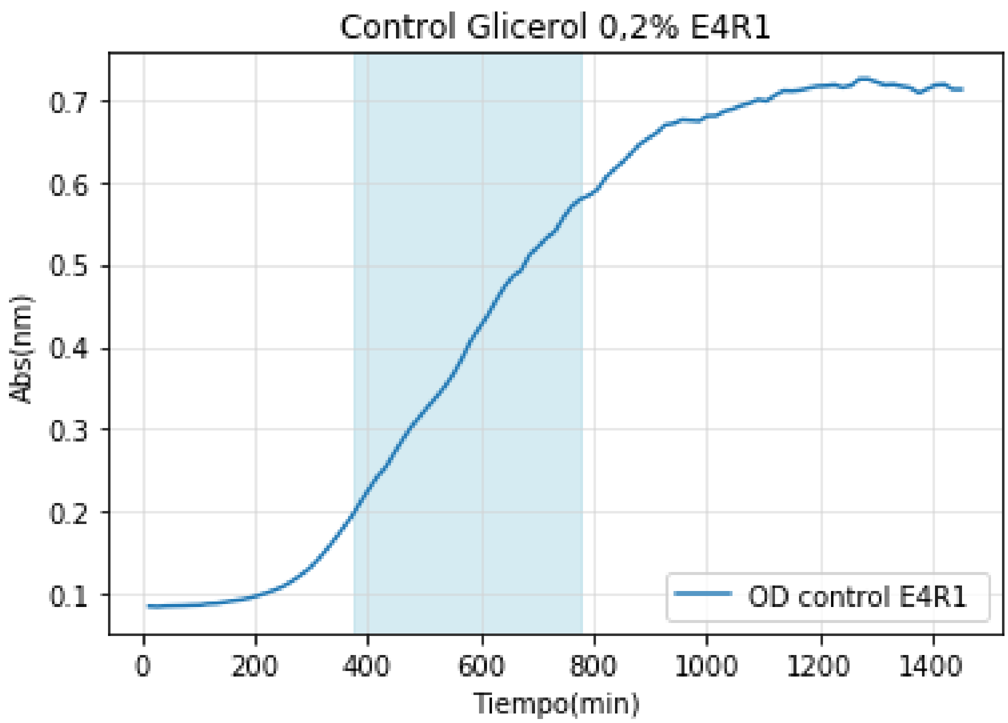
```
Min OD = 8.500000e-02
[  2.13735411e+00   4.44419084e-03   1.92719677e+02]
```

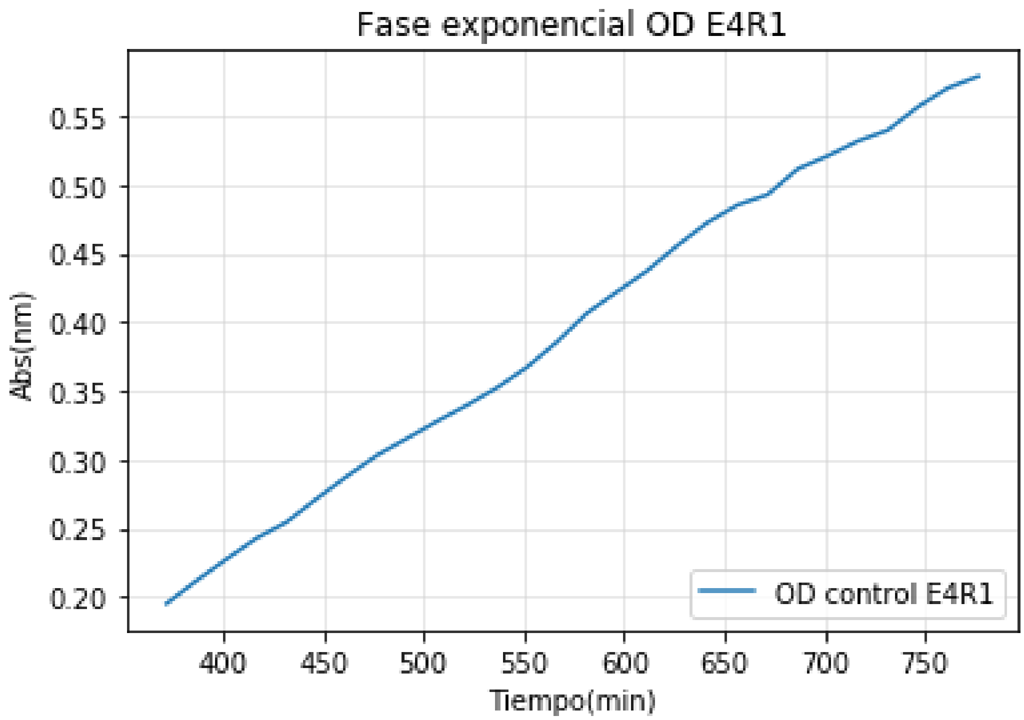



```
A=2.137354e+00
um=4.444191e-03
```

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

```
Out[27]: <matplotlib.legend.Legend at 0x203610e8588>
```

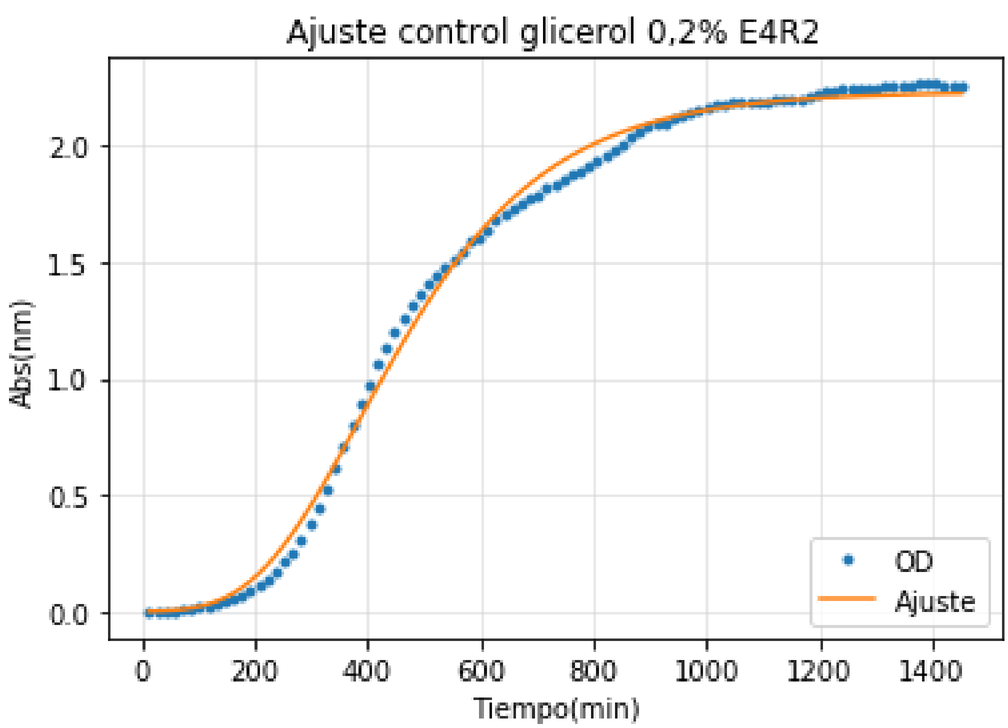




In [28]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 2
y17= np.log(odcgl122)-np.log(np.min(odcgl122))
print('Min OD = %e'%((np.min(odcgl122))))
evaly, params=Function_fit(tt,y17,0,-1, title = 'Ajuste control glicerol 0,2% E4R2')
A17= params[0]
um17=params[1]
l17=params[2]
print('A=%e'%(A17))
print('um=%e'%(um17))
print('l=%e'%(l17))

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

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

```python
y2=tt[52]
plt.figure()
plt.title('Control Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl122,label='OD control E4R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:53],odcgl122[25:53],label='OD control E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

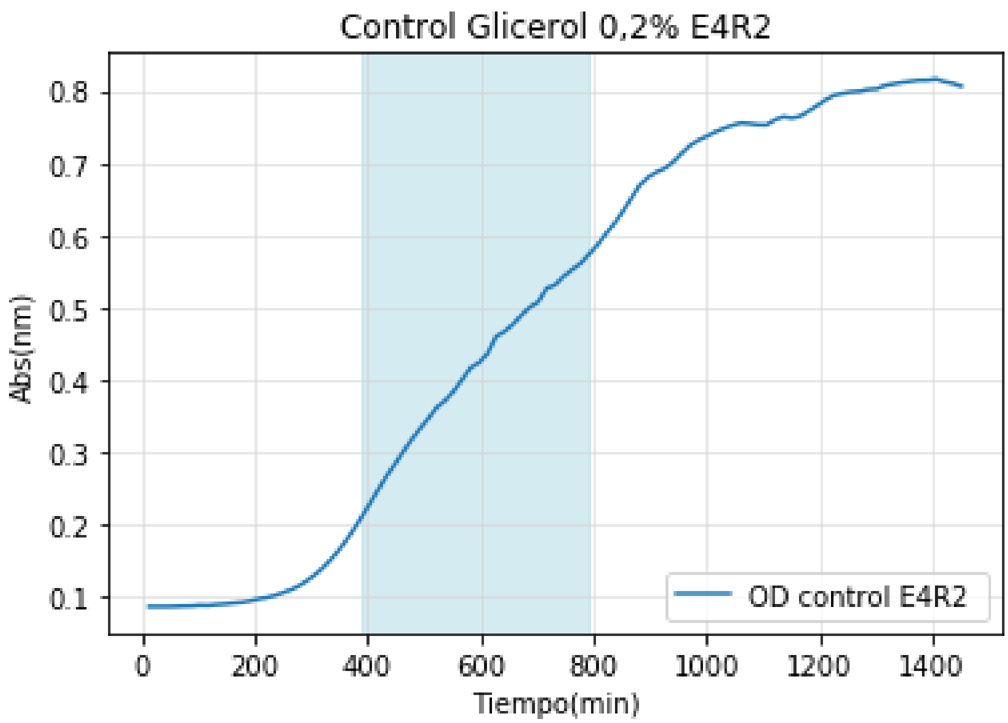
```
Min OD = 8.500000e-02
[  2.22944589e+00   4.42704053e-03   1.98978142e+02]
```

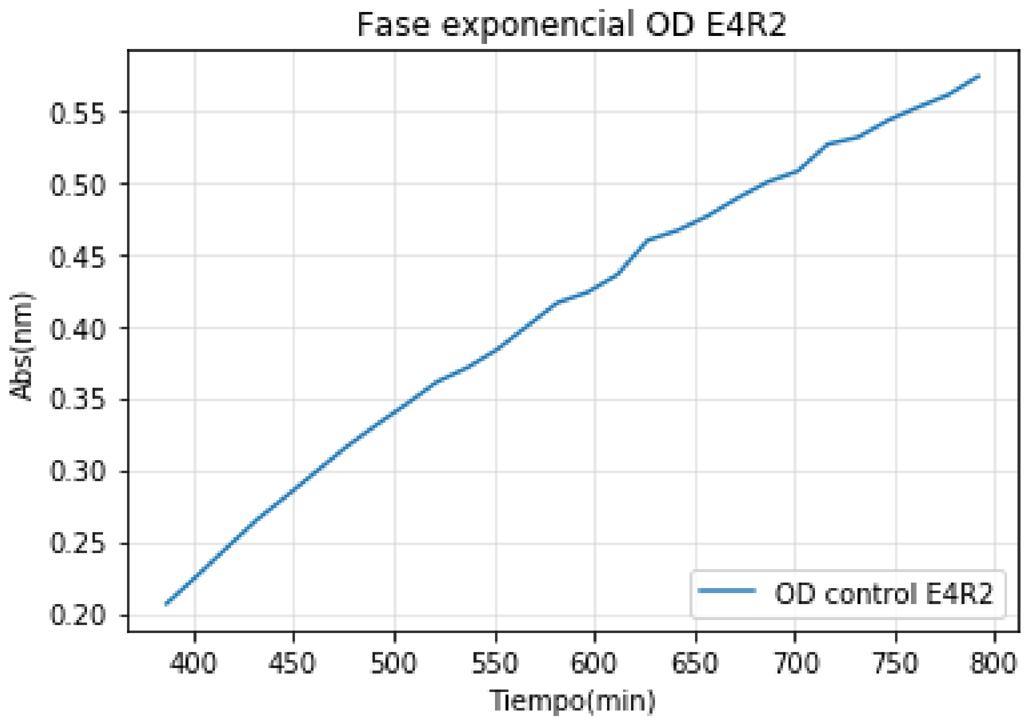



```
A=2.229446e+00
um=4.427041e-03
```

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

```
Out[28]: <matplotlib.legend.Legend at 0x2035f8a7278>
```

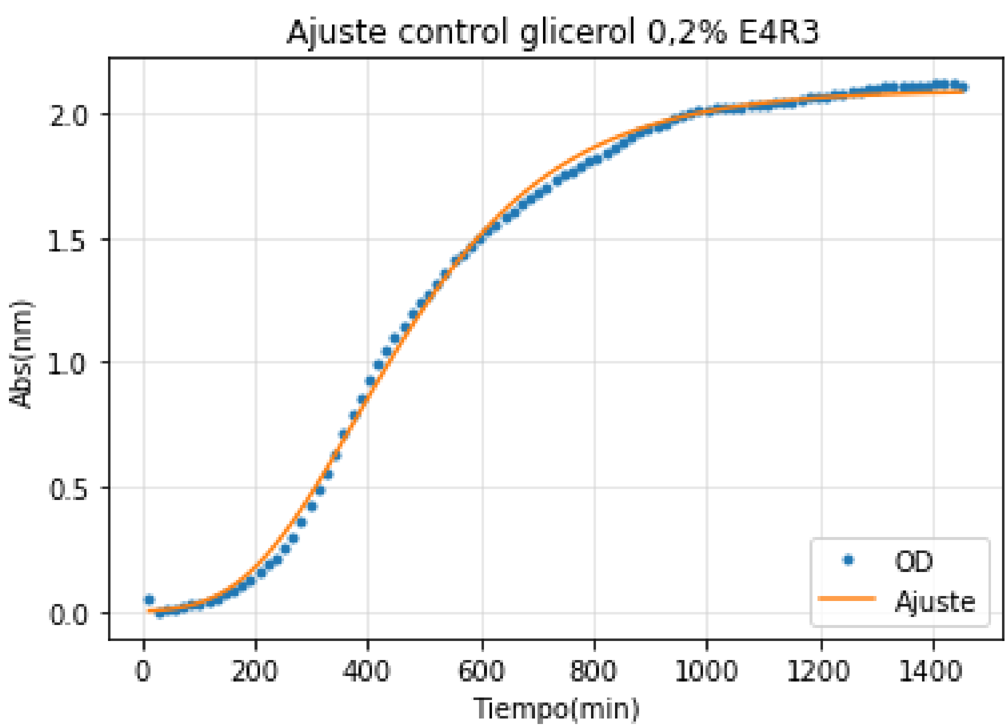





```
In [29]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 3
         y18= np.log(odcgl123)-np.log(np.min(odcgl123))
         print('Min OD = %e'%((np.min(odcgl123))))
         evaly, params=Function_fit(tt,y18,0,-1, title = 'Ajuste control glicerol 0,2% E4R3')
         A18= params[0]
         um18=params[1]
         l18=params[2]
         print('A=%e'%(A18))
         print('um=%e'%(um18))
         print('l=%e'%(l18))

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

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

```python
y2=tt[56]
plt.figure()
plt.title('Control Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl123,label='OD control E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[25:57],odcgl123[25:57],label='OD control E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

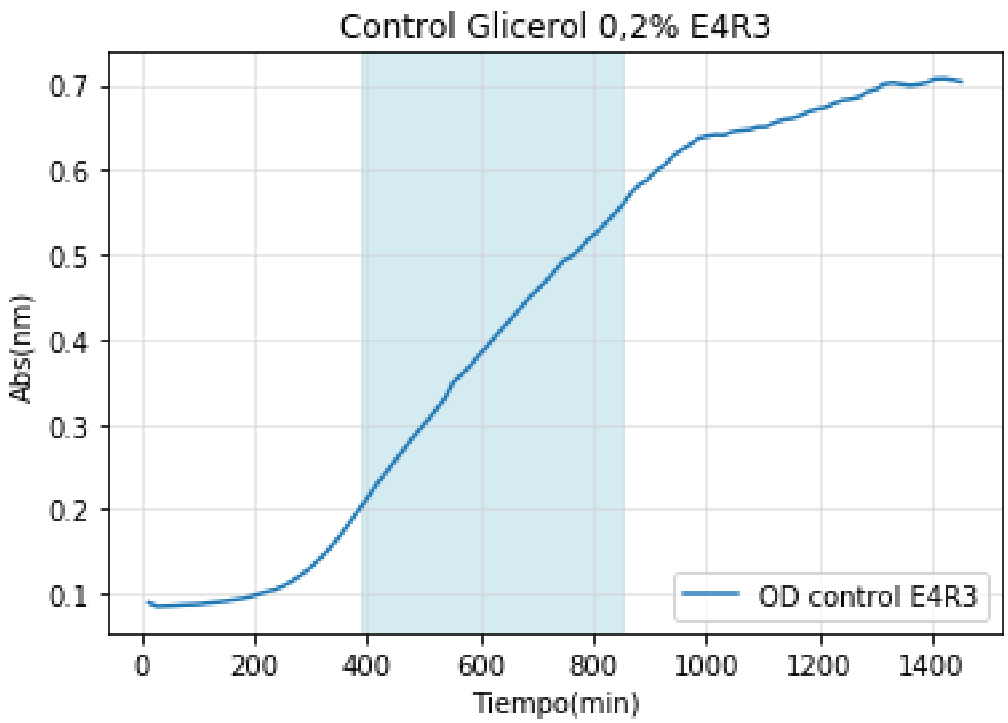
```
Min OD = 8.525000e-02
[  2.09342076e+00   3.90264068e-03   1.80268803e+02]
```

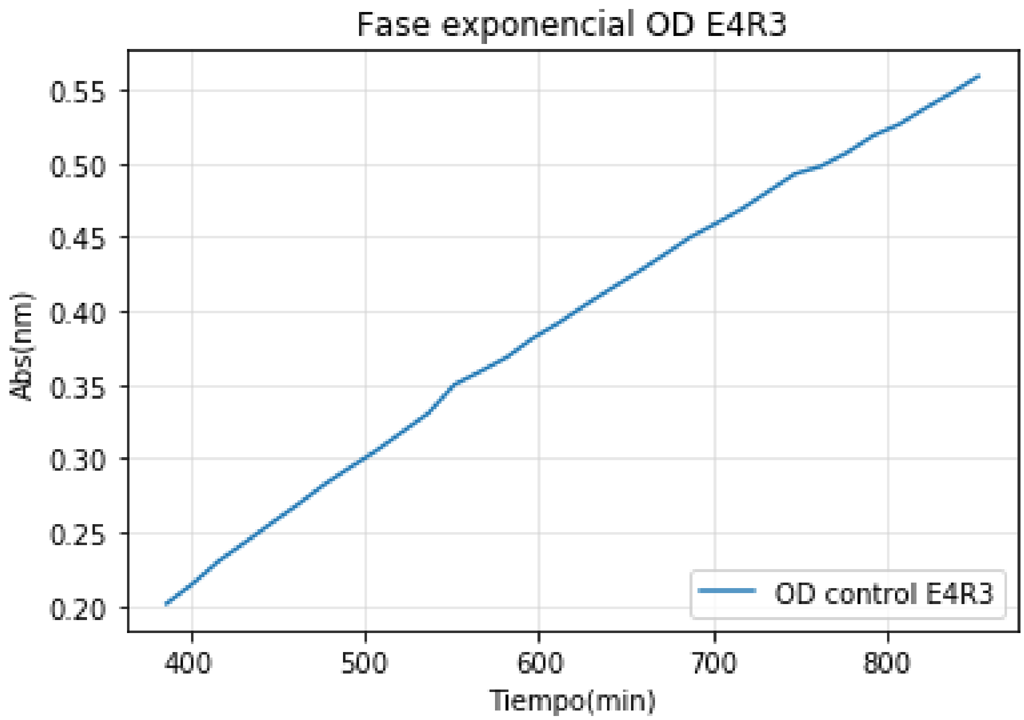



```
A=2.093421e+00
um=3.902641e-03
```

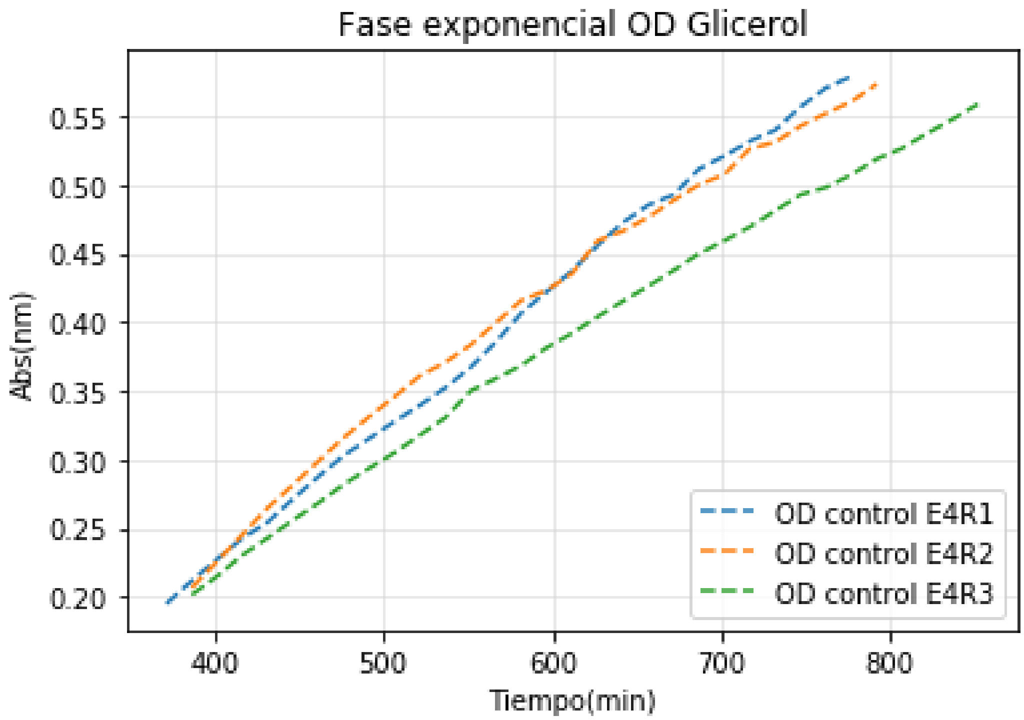
```
l=1.802688e+02
Tm=3.776035e+02
doubpe=1.776098e+02
ext=4.440245e+02
Tfinal=8.216280e+02
```

Out[29]: <matplotlib.legend.Legend at 0x2035fd61cc0>

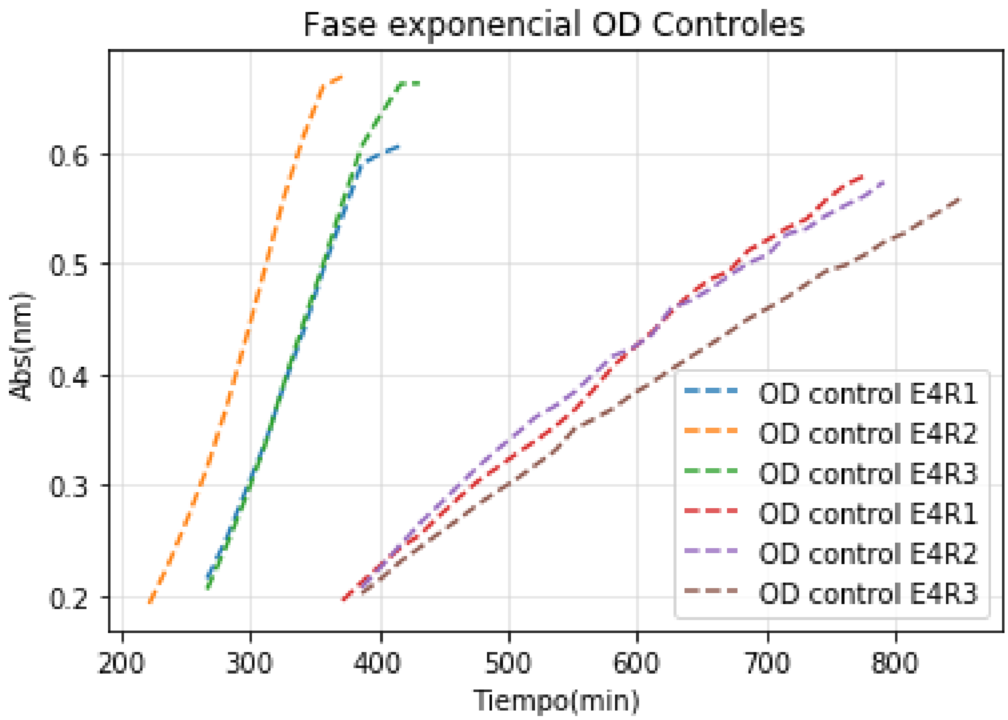





```
In [30]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Glicerol')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[24:52],odcgl121[24:52],'--',label='OD control E4R1')
         plt.plot(tt[25:53],odcgl122[25:53],'--',label='OD control E4R2')
         plt.plot(tt[25:57],odcgl123[25:57],'--',label='OD control E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[30]: <matplotlib.legend.Legend at 0x20361216fd0>
```



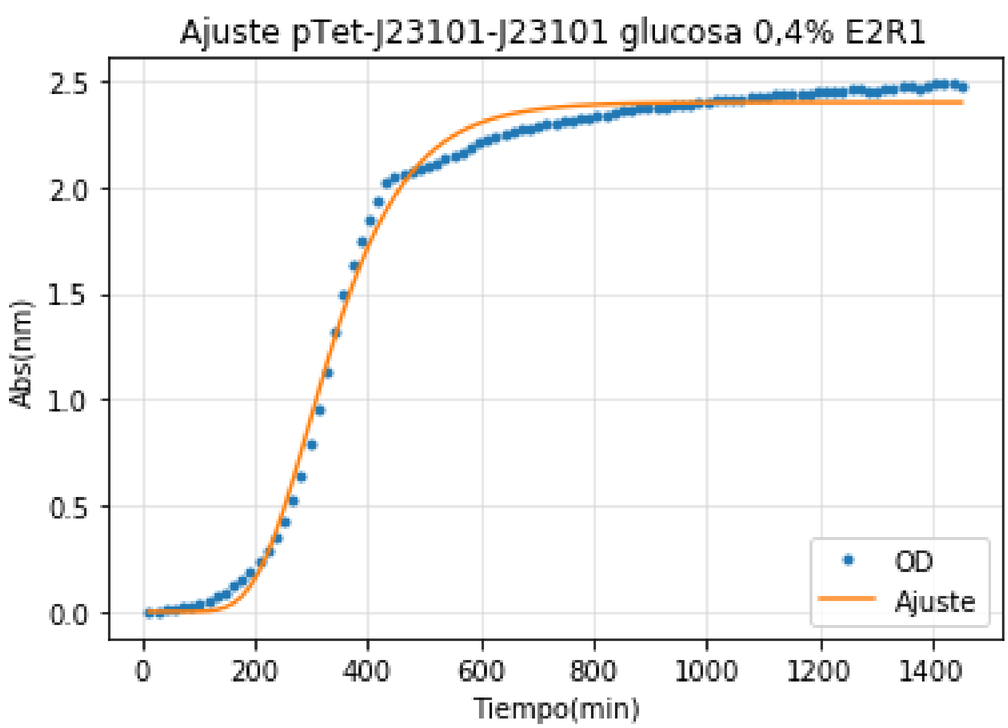

```
In [31]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD Controles')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:28],odcg121[17:28],'--',label='OD control E4R1')
         plt.plot(tt[14:25],odcg122[14:25],'--',label='OD control E4R2')
         plt.plot(tt[17:29],odcg123[17:29],'--',label='OD control E4R3')
         plt.plot(tt[24:52],odcgl121[24:52],'--',label='OD control E4R1')
         plt.plot(tt[25:53],odcgl122[25:53],'--',label='OD control E4R2')
         plt.plot(tt[25:57],odcgl123[25:57],'--',label='OD control E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[31]: <matplotlib.legend.Legend at 0x203612e1c50>
```



In [32]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-J23101-std glucosa rep 1
y19= np.log(od15211)-np.log(np.min(od15211))
print('Min OD = %e'%((np.min(od15211))))
evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste pTet-J23101-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]
```

```python
y2=tt[30]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15211,label='OD pTet-J23101-J23101 E2R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:31],od15211[19:31],label='OD pTet-J23101-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

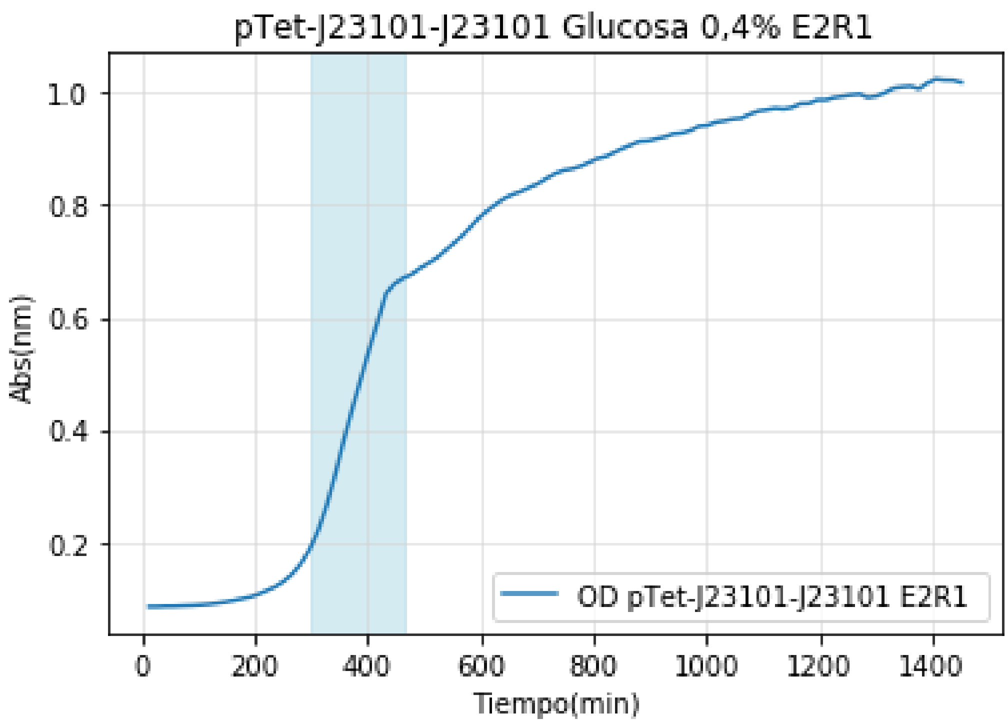
```
Min OD = 8.530000e-02
[  2.39988520e+00   9.27231338e-03   2.00377623e+02]
```

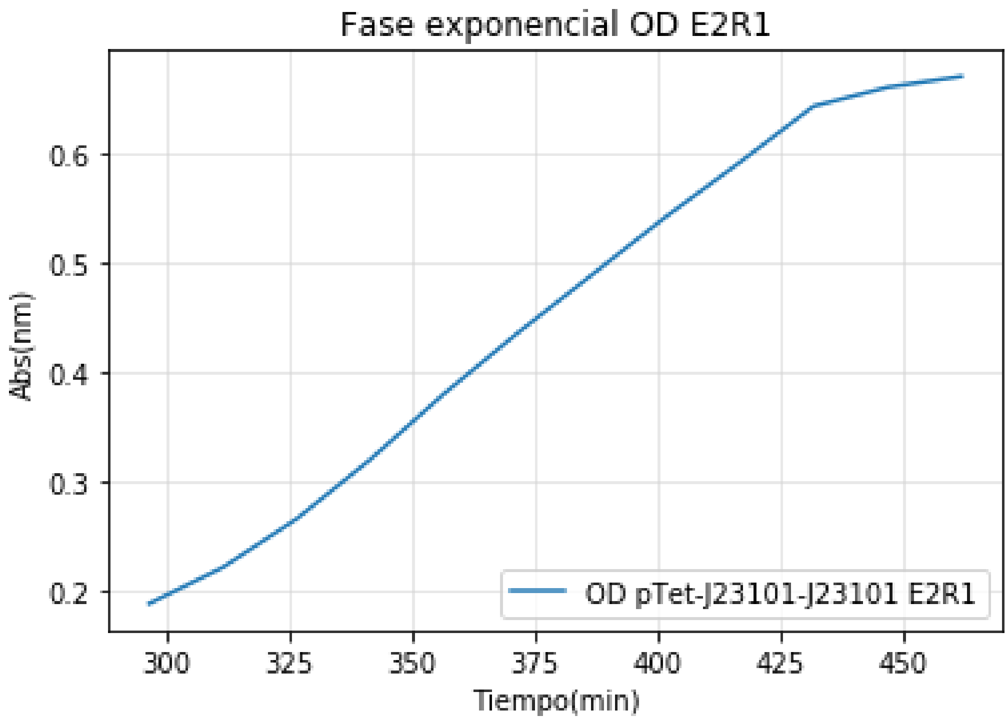



```
A=2.399885e+00
um=9.272313e-03
```

```
l=2.003776e+02
Tm=2.955932e+02
doubpe=7.475450e+01
ext=1.495090e+02
Tfinal=4.451022e+02
```

Out[32]: `<matplotlib.legend.Legend at 0x20360dad5c0>`

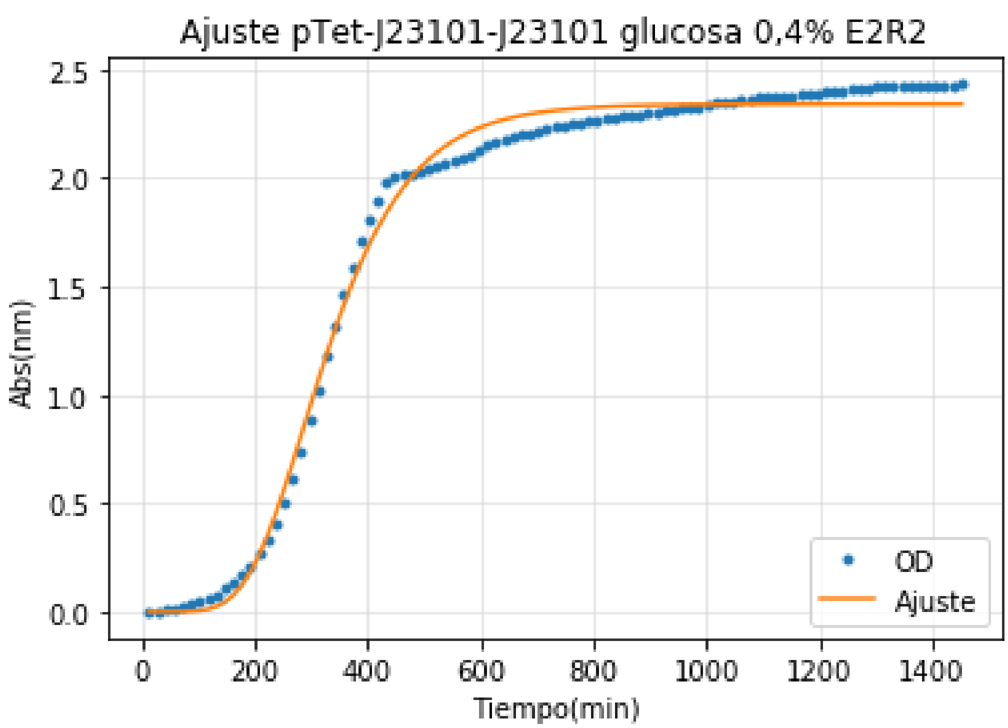




In [33]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 2
y20= np.log(od15212)-np.log(np.min(od15212))
print('Min OD = %e'%((np.min(od15212))))
evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste pTet-J23101-J23101 glucosa 0,4%
A20= params[0]
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[19]
```

```
y2=tt[31]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15212,label='OD pTet-J23101-J23101 E2R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[19:32],od15212[19:32],label='OD pTet-J23101-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

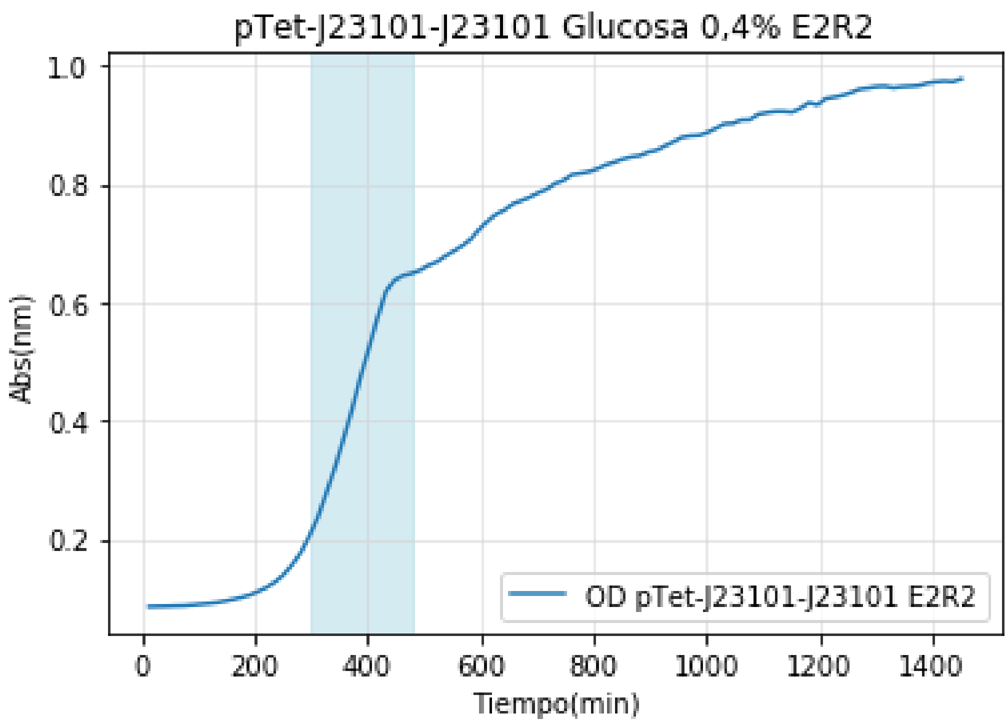
```
Min OD = 8.580000e-02
[  2.34140500e+00   8.43333849e-03   1.84992926e+02]
```

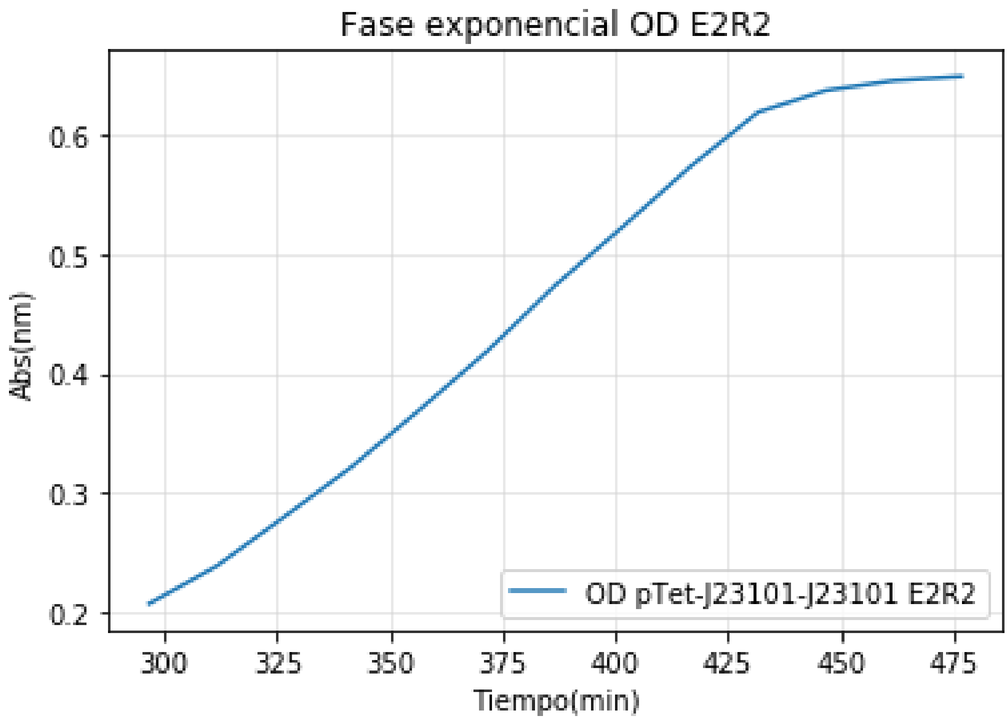



```
A=2.341405e+00
um=8.433338e-03
```

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

```
Out[33]: <matplotlib.legend.Legend at 0x20360de1e48>
```

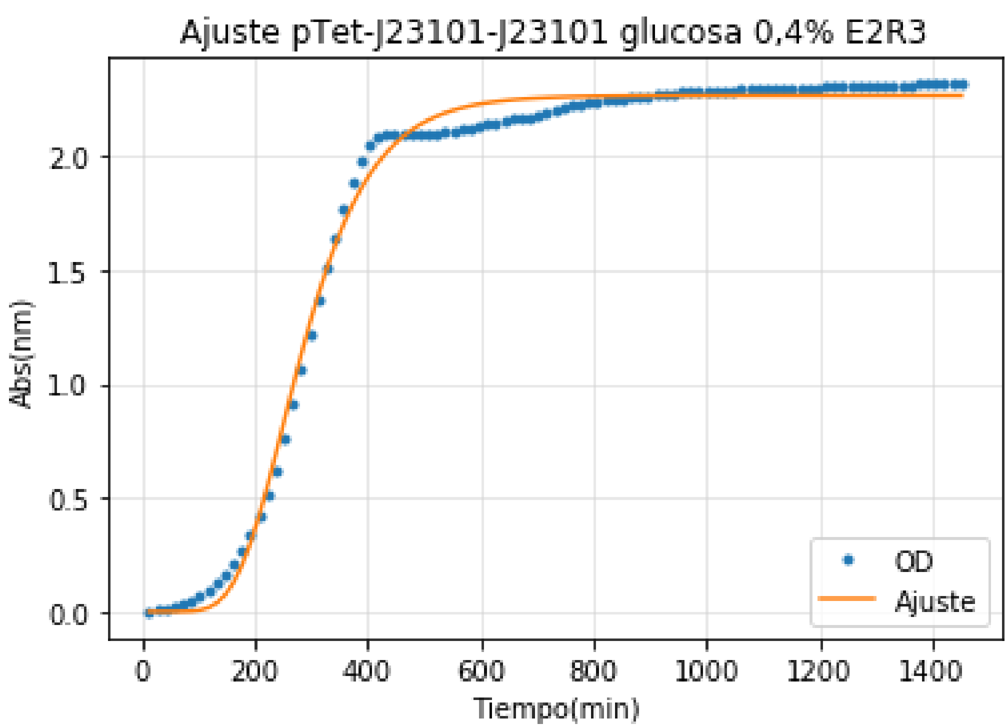




In [34]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glucosa rep 3
y21= np.log(od15213)-np.log(np.min(od15213))
print('Min OD = %e'%((np.min(od15213))))
evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste pTet-J23101-J23101 glucosa 0,4%
A21= params[0]
um21=params[1]
l21=params[2]
print('A=%e'%(A21))
print('um=%e'%(um21))
print('l=%e'%(l21))

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

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

```
y2=tt[27]
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od15213,label='OD pTet-J23101-J23101 E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:28],od15213[16:28],label='OD pTet-J23101-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

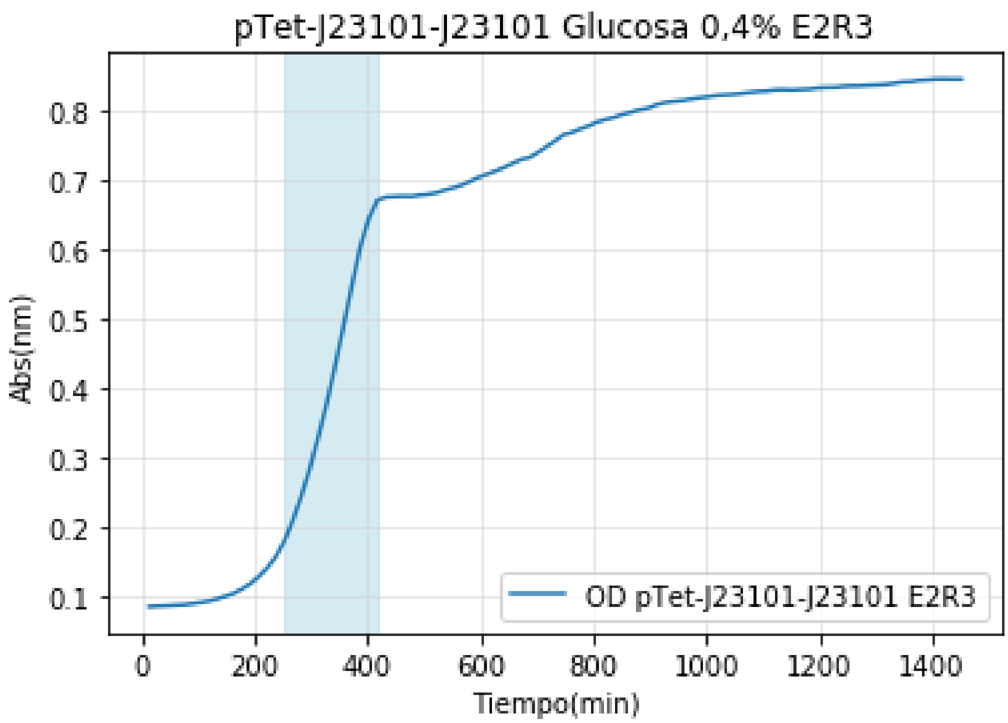
```
Min OD = 8.360000e-02
[  2.26457592e+00   9.83425423e-03   1.66237345e+02]
```

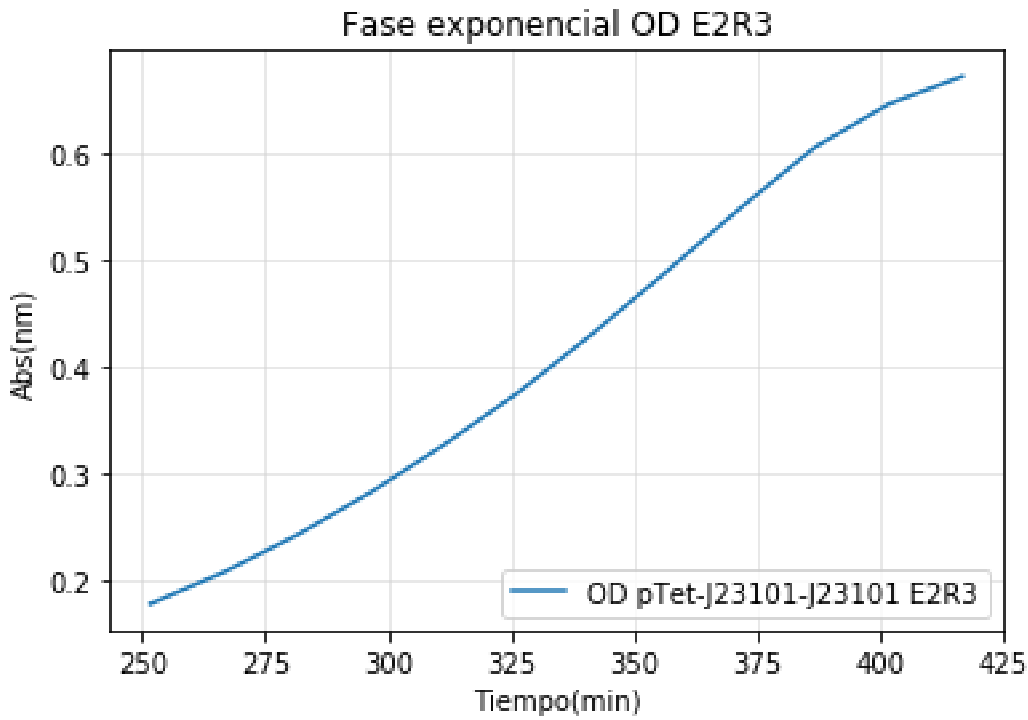



```
A=2.264576e+00
um=9.834254e-03
```

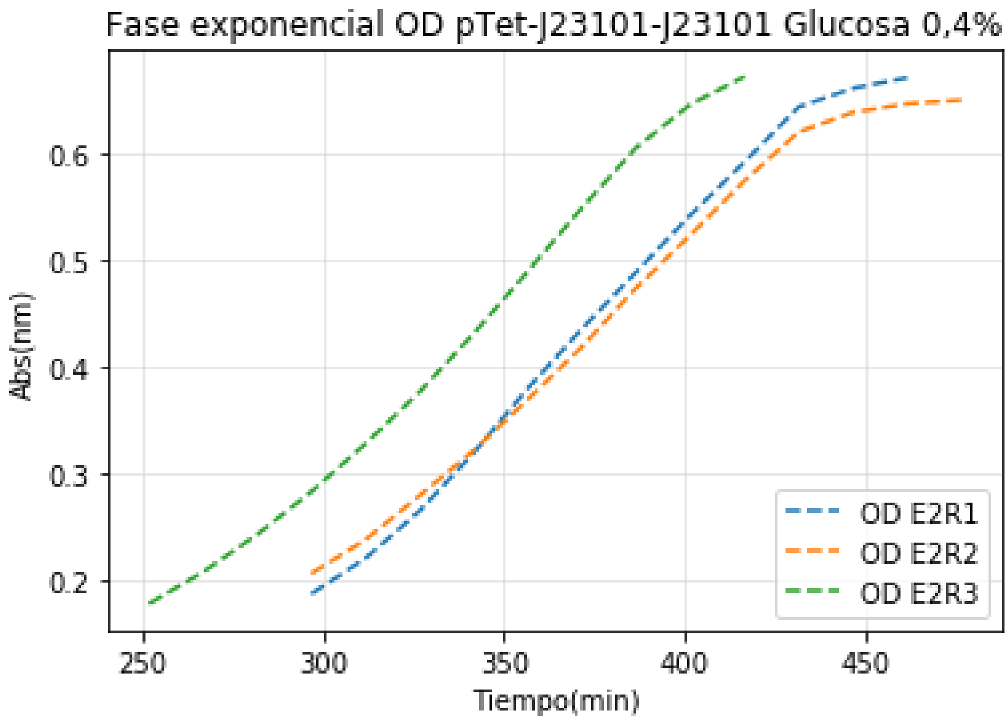
```
l=1.662373e+02
Tm=2.509505e+02
doubpe=7.048294e+01
ext=1.409659e+02
Tfinal=3.919164e+02
```

```
Out[34]: <matplotlib.legend.Legend at 0x20361455048>
```





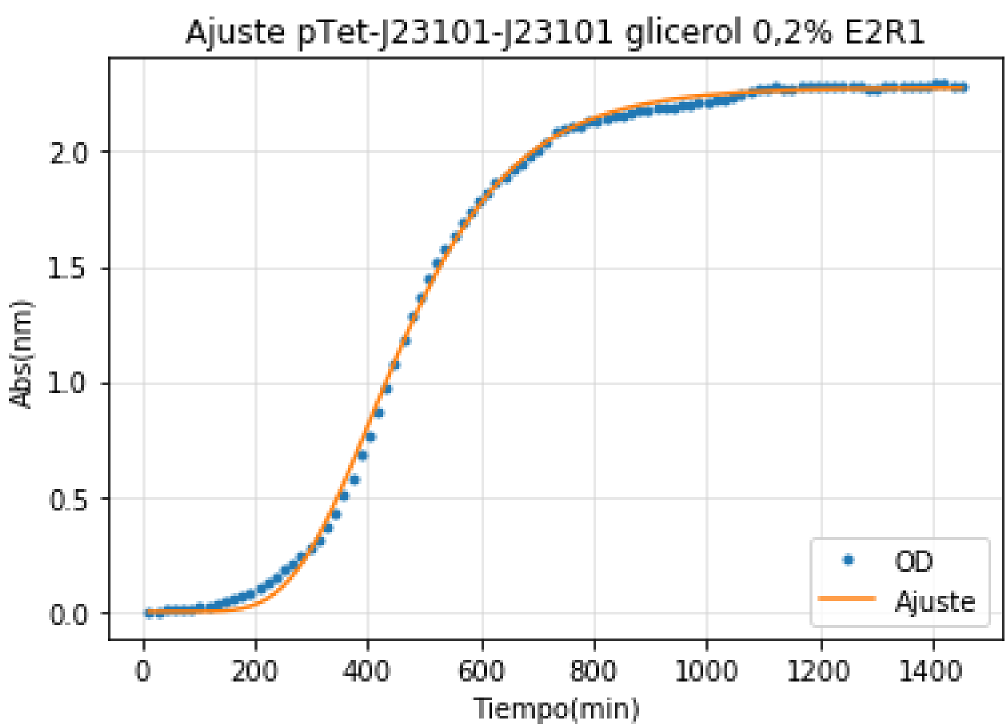

```
In [35]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:31],od15211[19:31],'--',label='OD E2R1')
         plt.plot(tt[19:32],od15212[19:32],'--',label='OD E2R2')
         plt.plot(tt[16:28],od15213[16:28],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[35]: <matplotlib.legend.Legend at 0x2036157ca90>
```




```
In [36]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #ptet-std-std glicerol rep 1
         y22= np.log(od1521g1)-np.log(np.min(od1521g1))
         print('Min OD = %e'%((np.min(od1521g1))))
         evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
         A22= params[0]
         um22=params[1]
         l22=params[2]
         print('A=%e'%(A22))
         print('um=%e'%(um22))
         print('l=%e'%(l22))

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

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

```python
y2=tt[47]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g1,label='OD pTet-J23101-J23101 E2R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:48],od1521g1[27:48],label='OD pTet-J23101-J23101 E2R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

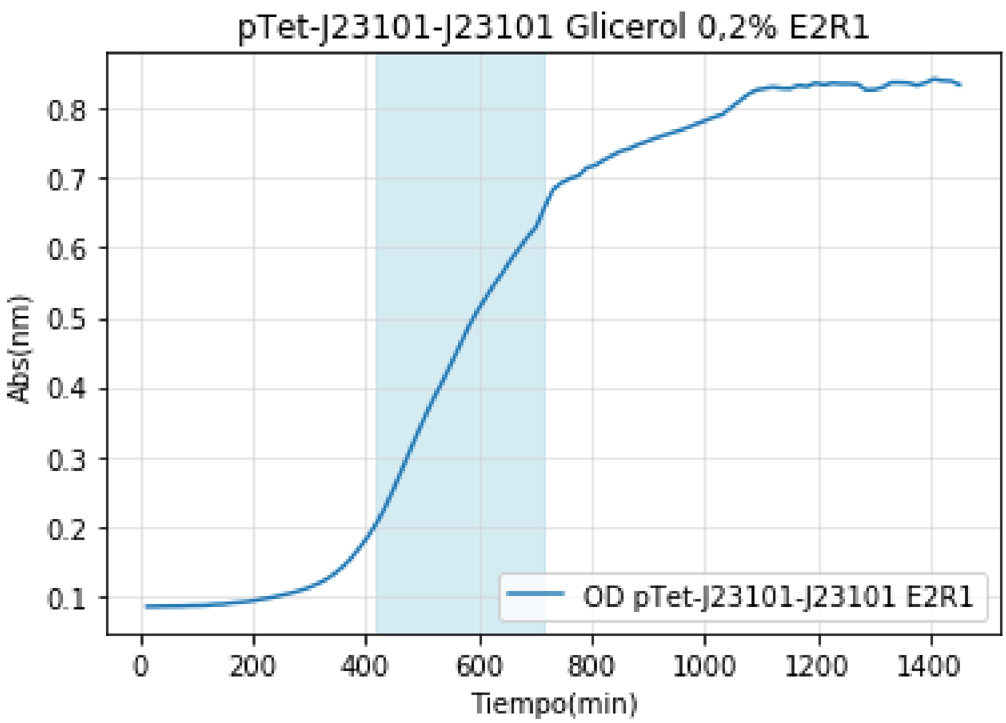
```
Min OD = 8.510000e-02
[  2.27552043e+00   5.96588621e-03   2.64362418e+02]
```

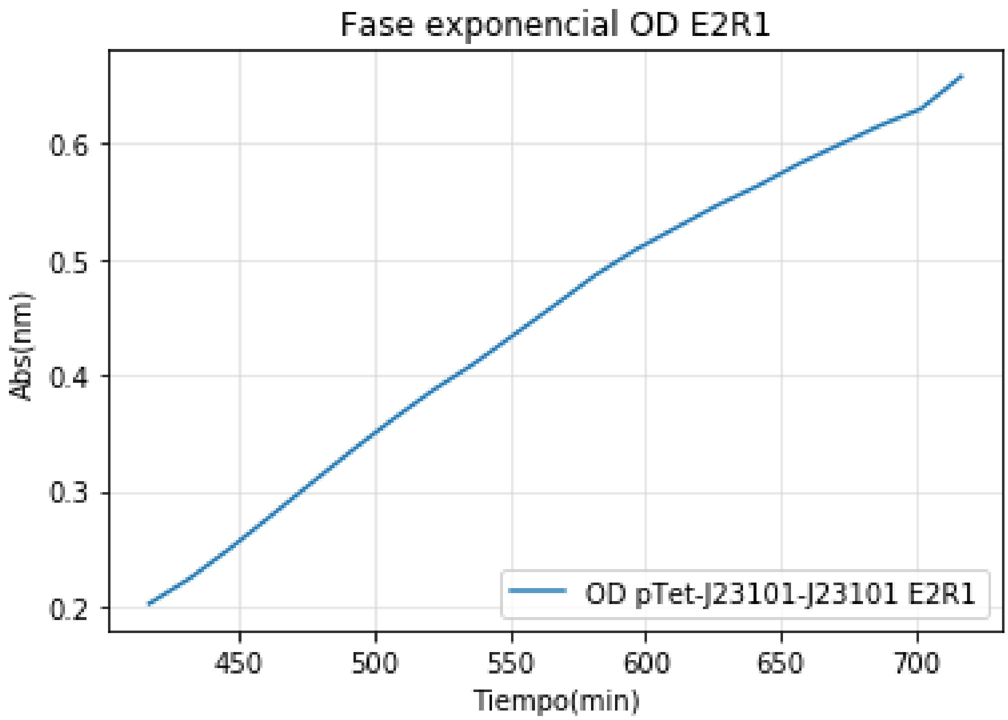



```
A=2.275520e+00
um=5.965886e-03
```

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

Out[36]: <matplotlib.legend.Legend at 0x203617112b0>

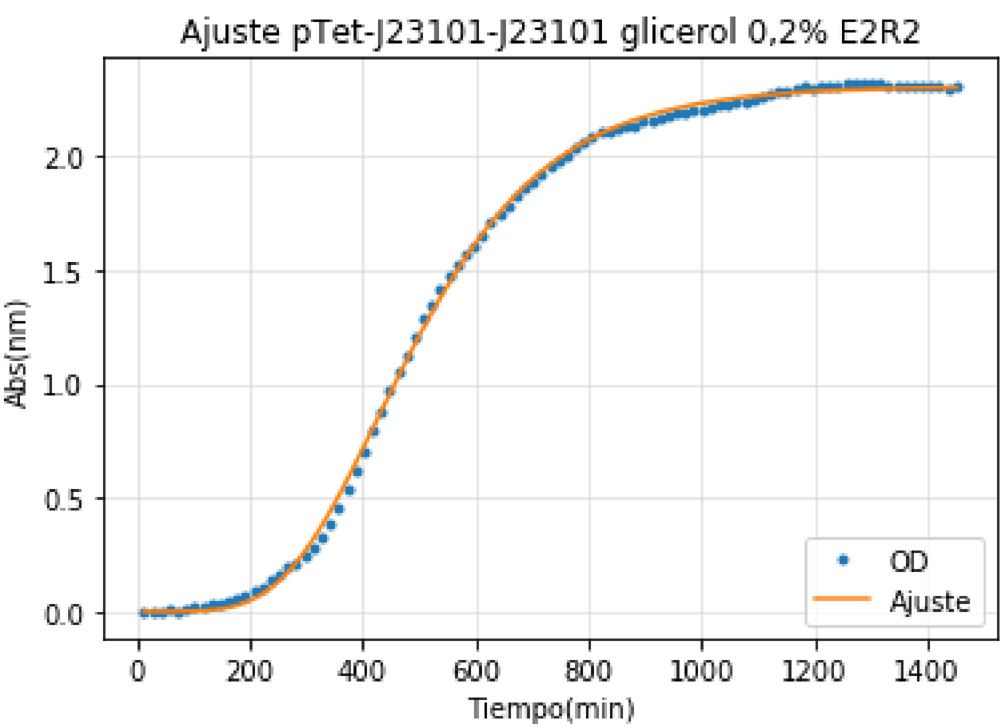




In [37]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 2
y23= np.log(od1521g2)-np.log(np.min(od1521g2))
print('Min OD = %e'%((np.min(od1521g2))))
evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
A23= 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[28]
```

```
y2=tt[52]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g2,label='OD pTet-J23101-J23101 E2R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:53],od1521g2[28:53],label='OD pTet-J23101-J23101 E2R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

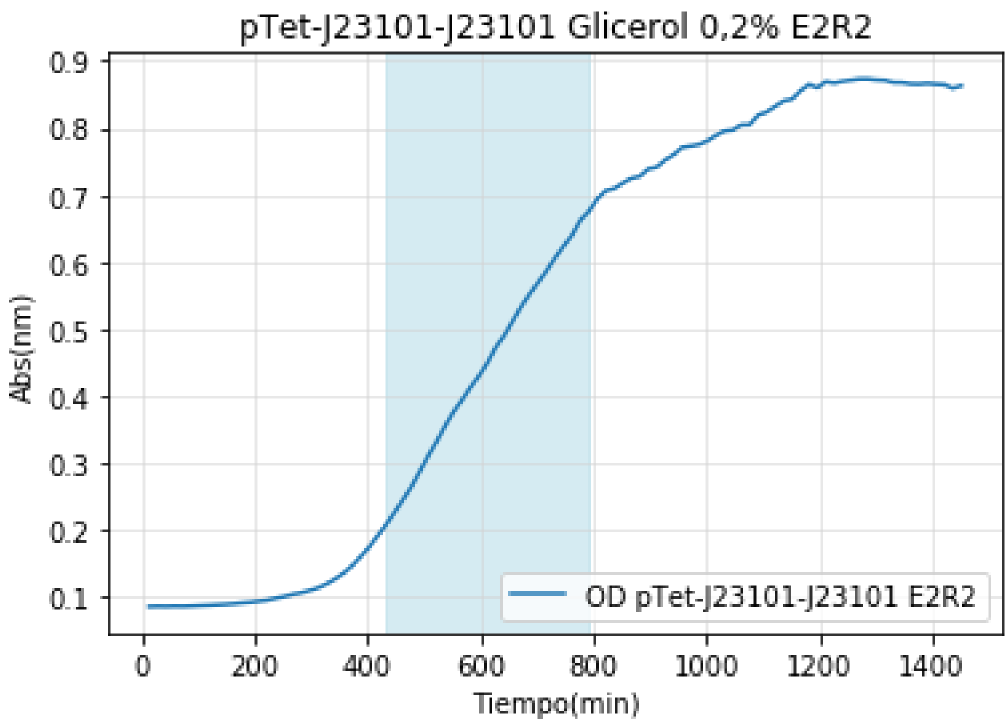
```
Min OD = 8.600000e-02
[  2.30705217e+00   5.07308656e-03   2.58634310e+02]
```

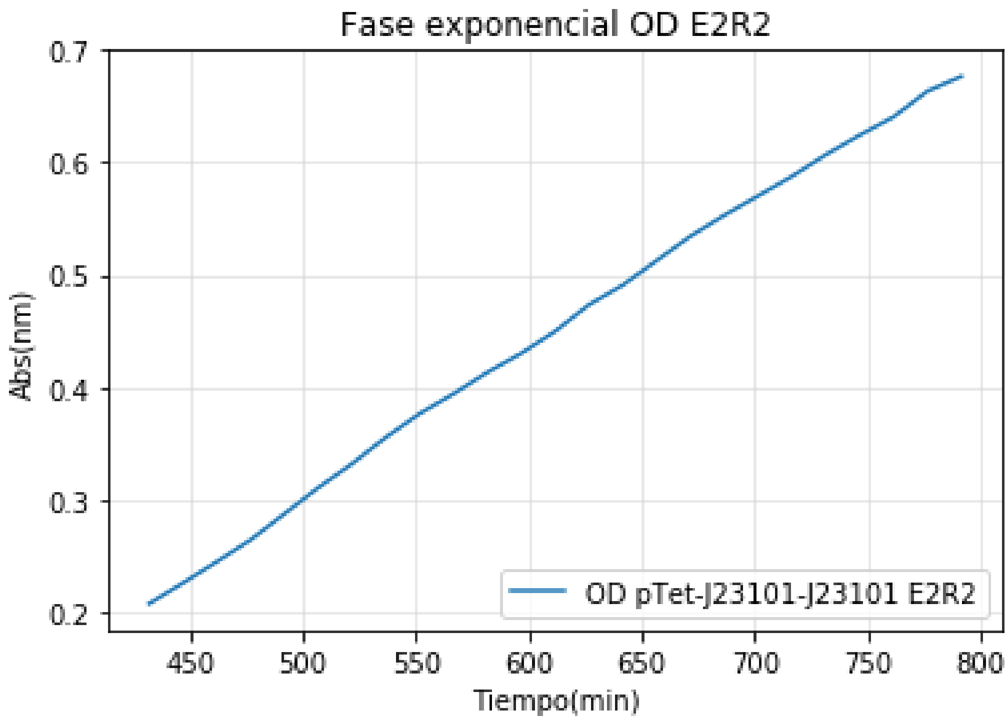



```
A=2.307052e+00
um=5.073087e-03
```

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

```
Out[37]: <matplotlib.legend.Legend at 0x203610fd198>
```

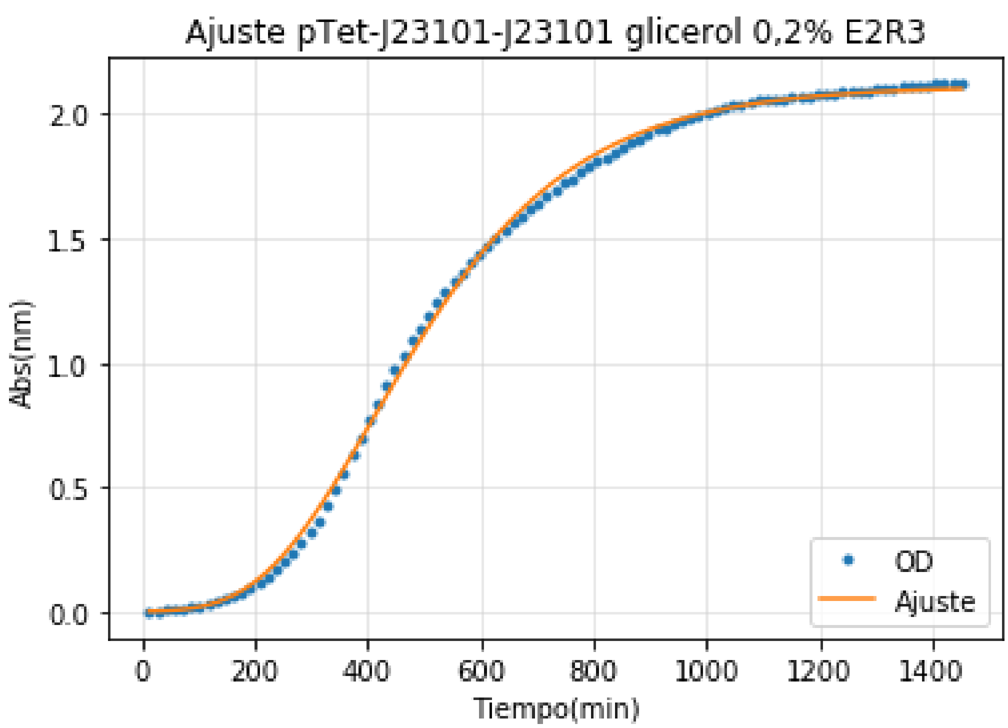




In [38]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#ptet-std-std glicerol rep 3
y24= np.log(od1521g3)-np.log(np.min(od1521g3))
print('Min OD = %e'%((np.min(od1521g3))))
evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste pTet-J23101-J23101 glicerol 0,2%
A24= params[0]
um24=params[1]
l24=params[2]
print('A=%e'%(A24))
print('um=%e'%(um24))
print('l=%e'%(l24))

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

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

```
y2=tt[52]
plt.figure()
plt.title('pTet-J23101-J23101 Glicerol 0,2% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1521g3,label='OD pTet-J23101-J23101 E2R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[27:53],od1521g3[27:53],label='OD pTet-J23101-J23101 E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

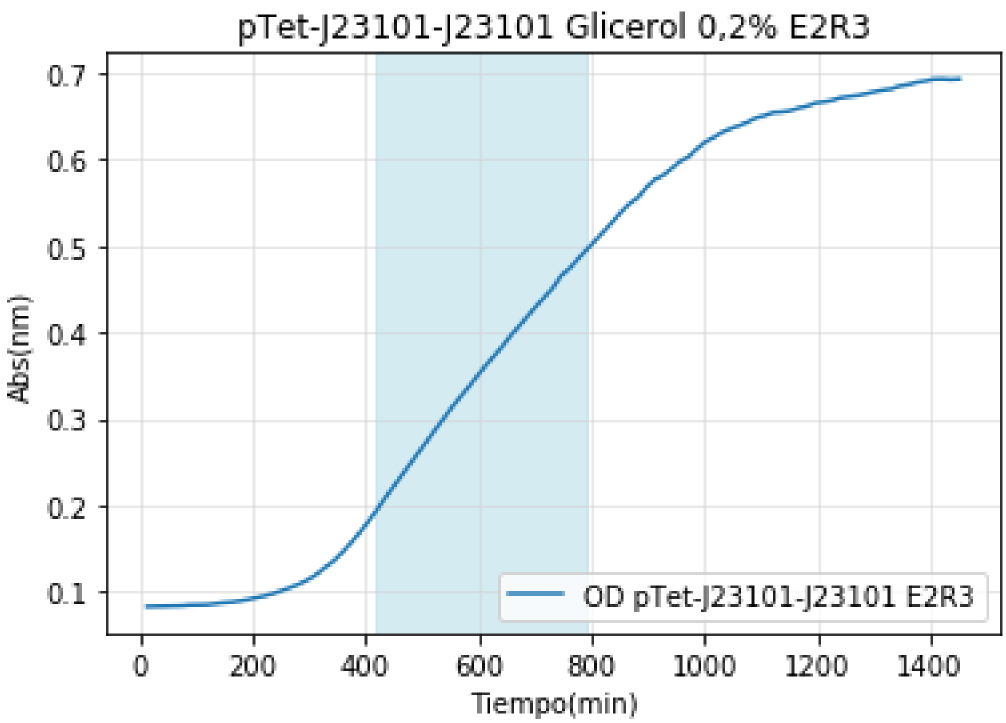
```
Min OD = 8.320000e-02
[  2.11194516e+00   3.89343431e-03   2.09853636e+02]
```

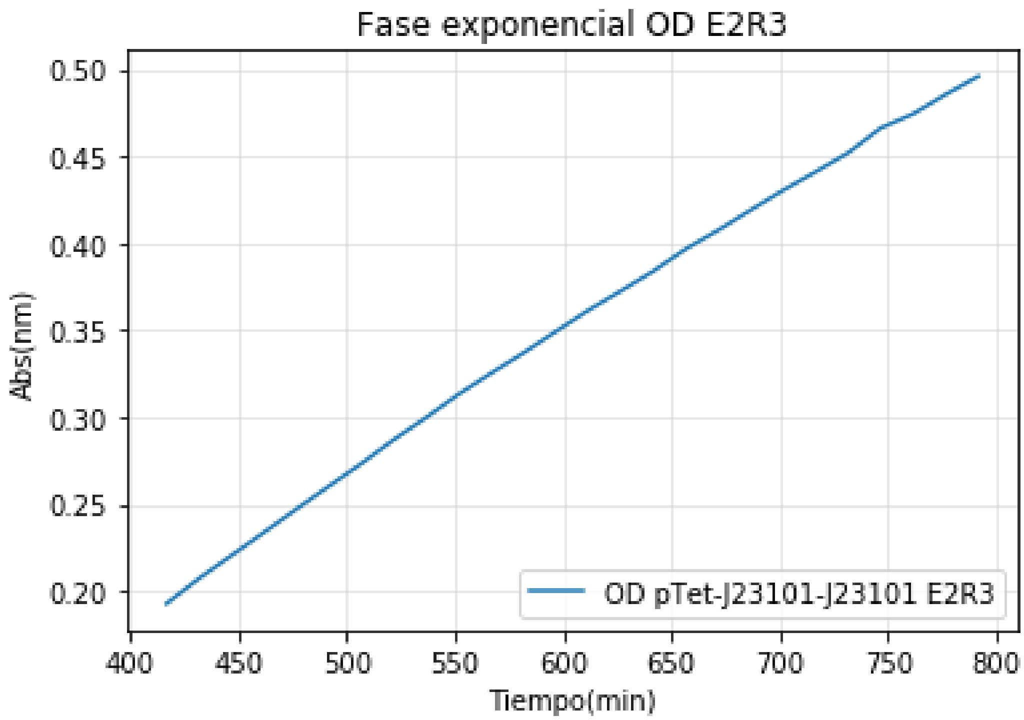



```
A=2.111945e+00
um=3.893434e-03
```

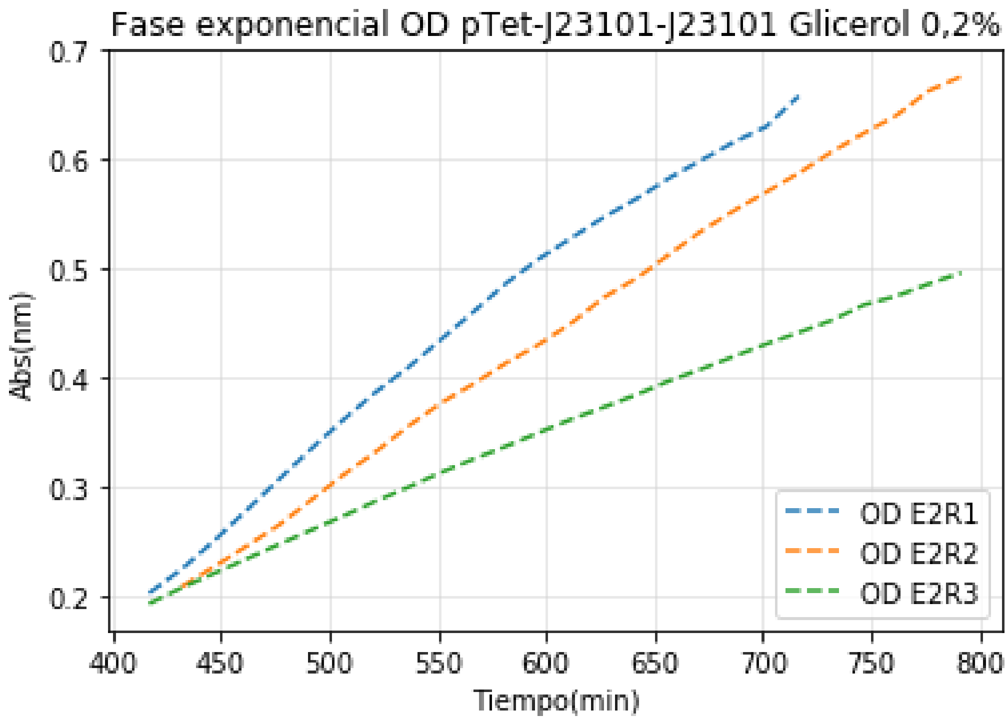
```
l=2.098536e+02
Tm=4.094053e+02
doubpe=1.780298e+02
ext=3.560595e+02
Tfinal=7.654648e+02
```

Out[38]: <matplotlib.legend.Legend at 0x20361076940>

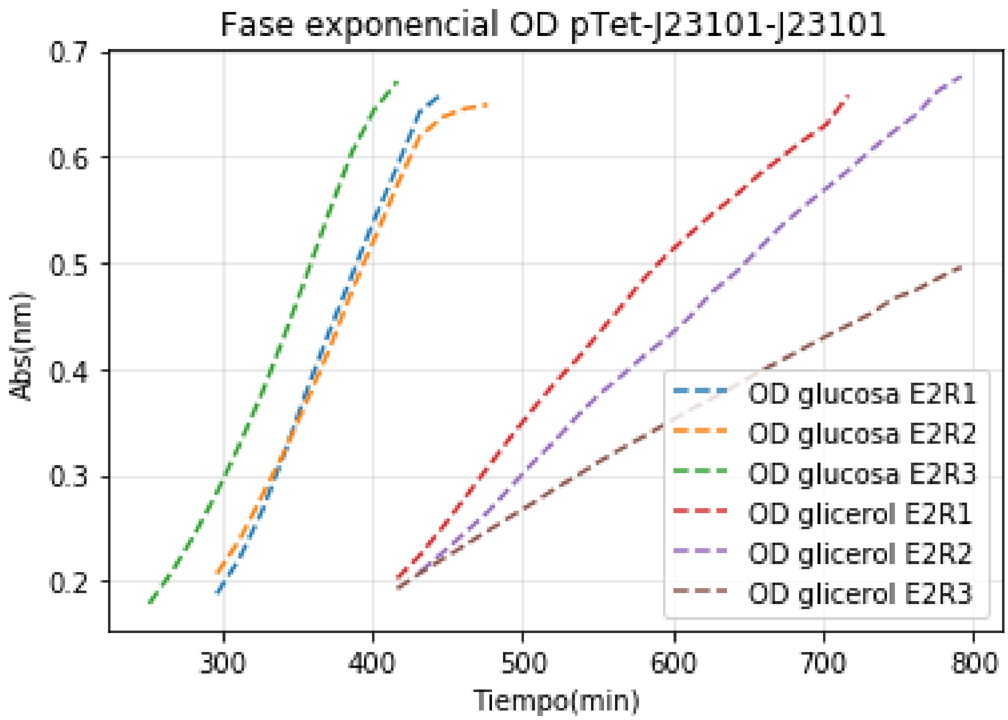





```
In [39]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[27:48],od1521g1[27:48],'--',label='OD E2R1')
         plt.plot(tt[28:53],od1521g2[28:53],'--',label='OD E2R2')
         plt.plot(tt[27:53],od1521g3[27:53],'--',label='OD E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[39]: <matplotlib.legend.Legend at 0x20361831710>
```



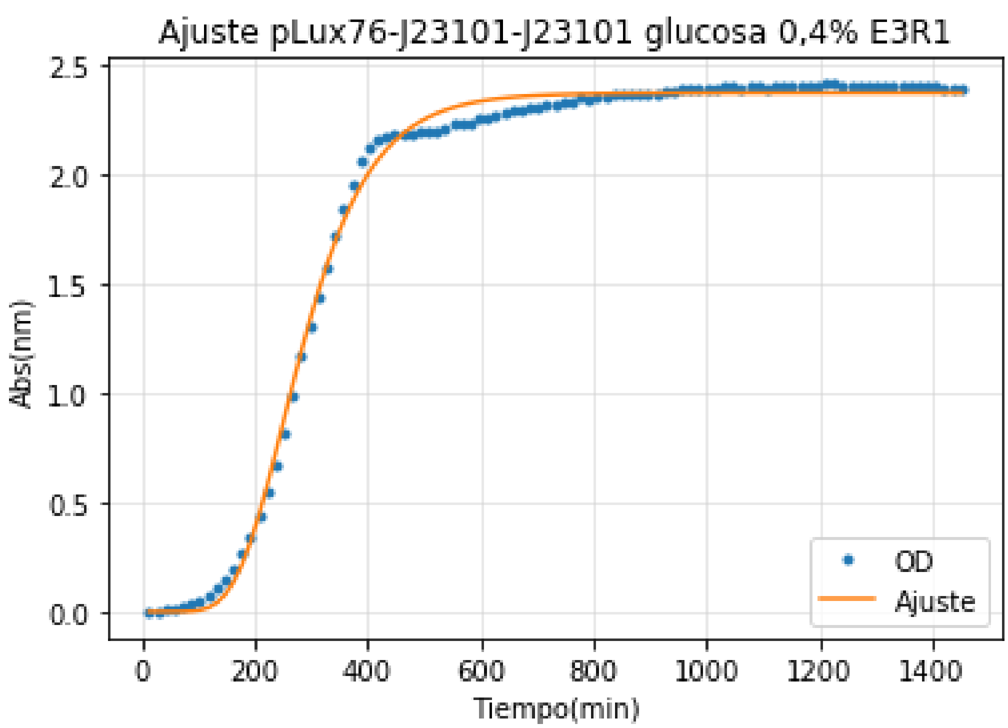

```
In [40]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pTet-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[19:30],od15211[19:30],'--',label='OD glucosa E2R1')
         plt.plot(tt[19:32],od15212[19:32],'--',label='OD glucosa E2R2')
         plt.plot(tt[16:28],od15213[16:28],'--',label='OD glucosa E2R3')
         plt.plot(tt[27:48],od1521g1[27:48],'--',label='OD glicerol E2R1')
         plt.plot(tt[28:53],od1521g2[28:53],'--',label='OD glicerol E2R2')
         plt.plot(tt[27:53],od1521g3[27:53],'--',label='OD glicerol E2R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[40]: <matplotlib.legend.Legend at 0x203618f9eb8>
```



In [41]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 1
y25= np.log(od18211)-np.log(np.min(od18211))
print('Min OD = %e'%((np.min(od18211))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste pLux76-J23101-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[16]
```

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

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

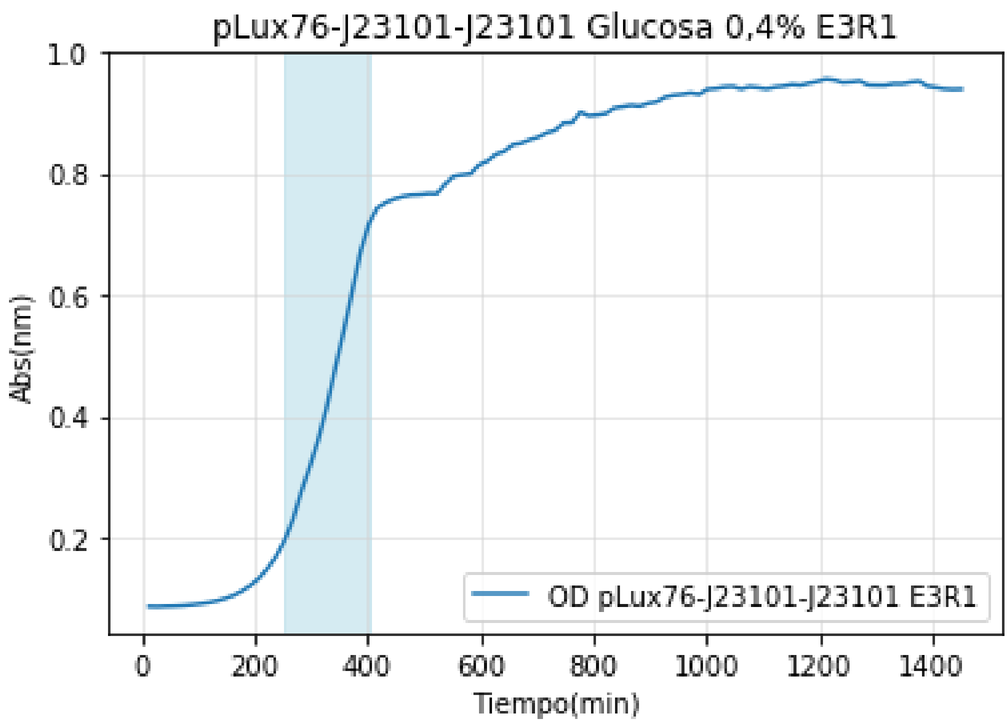
```
Min OD = 8.580000e-02
[  2.37037352e+00   1.03321767e-02   1.65952870e+02]
```

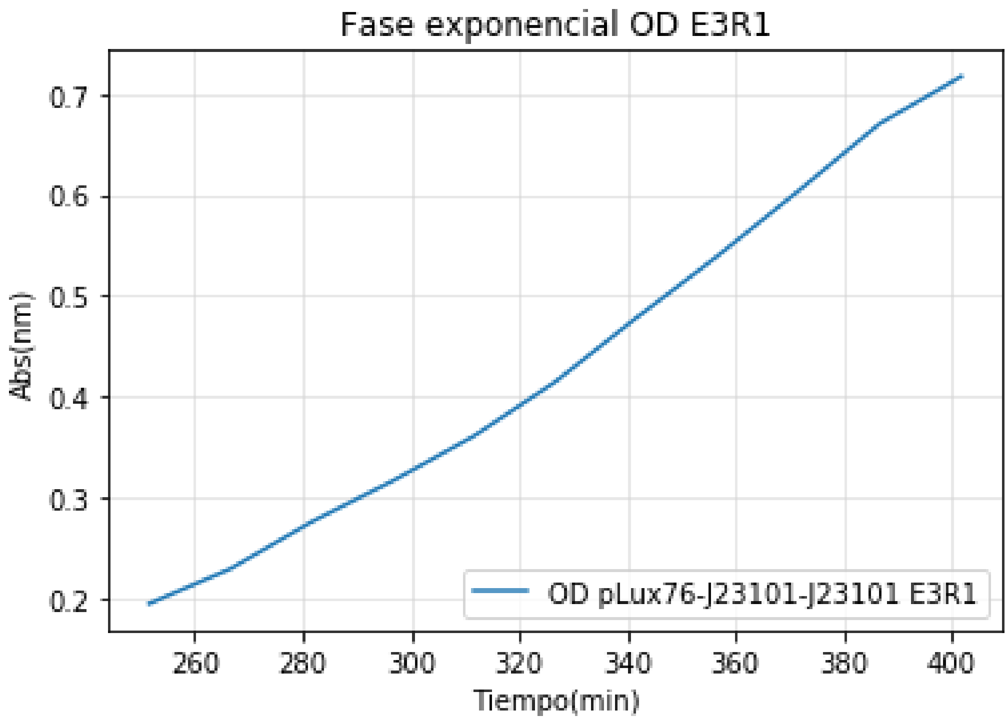



```
A=2.370374e+00
um=1.033218e-02
```

```
l=1.659529e+02
Tm=2.503505e+02
doubpe=6.708627e+01
ext=1.341725e+02
Tfinal=3.845231e+02
```

```
Out[41]: <matplotlib.legend.Legend at 0x20361aa2ba8>
```

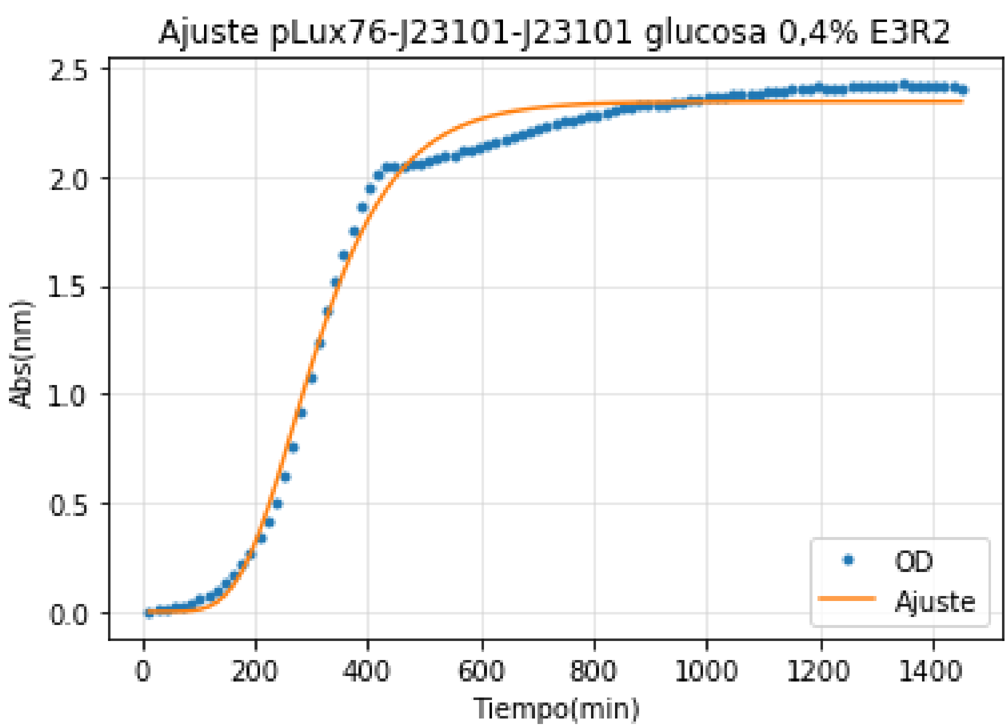




In [42]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 2
y26= np.log(od18212)-np.log(np.min(od18212))
print('Min OD = %e'%((np.min(od18212))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4
A26= params[0]
um26=params[1]
l26=params[2]
print('A=%e'%(A26))
print('um=%e'%(um26))
print('l=%e'%(l26))

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

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

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

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

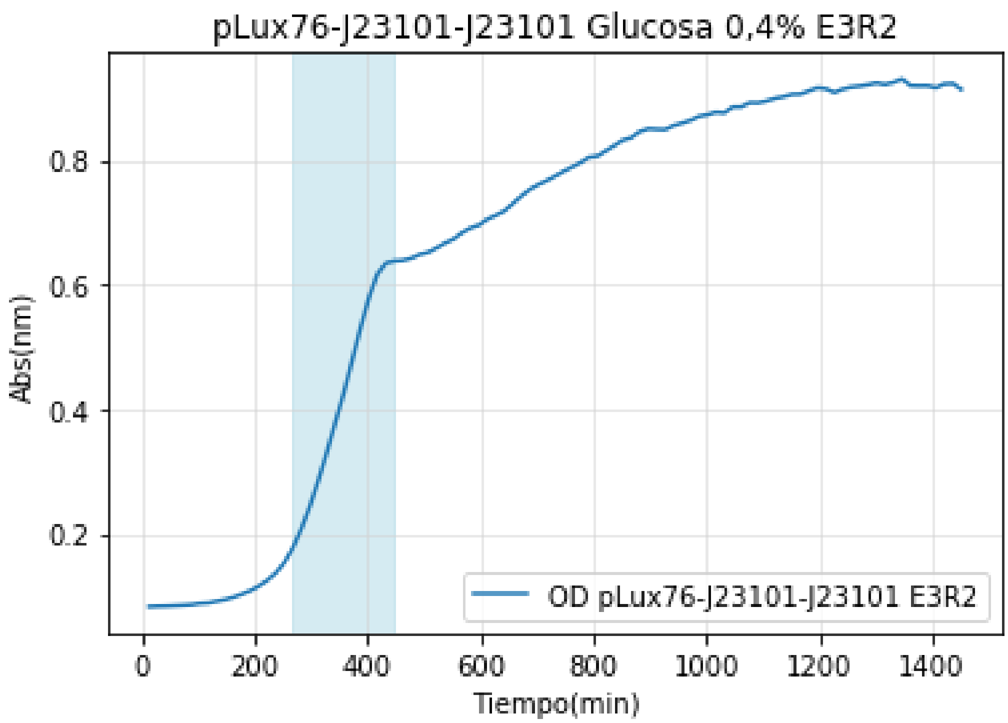
```
Min OD = 8.250000e-02
[  2.34308642e+00   8.73368317e-03   1.69674212e+02]
```

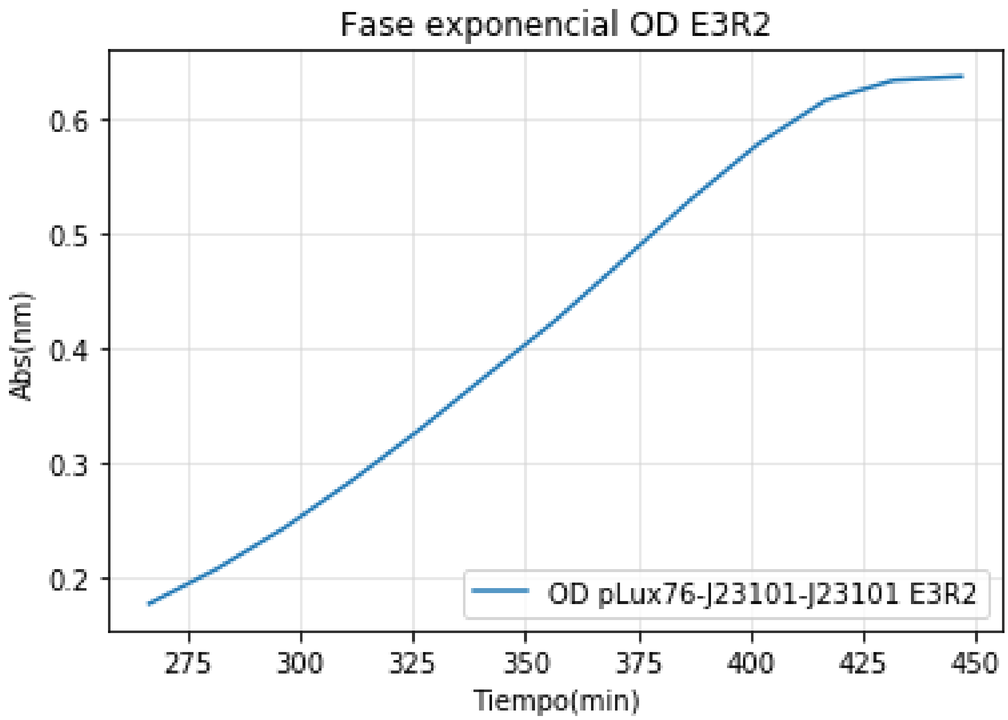



```
A=2.343086e+00
um=8.733683e-03
```

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

```
Out[42]: <matplotlib.legend.Legend at 0x20360d520f0>
```

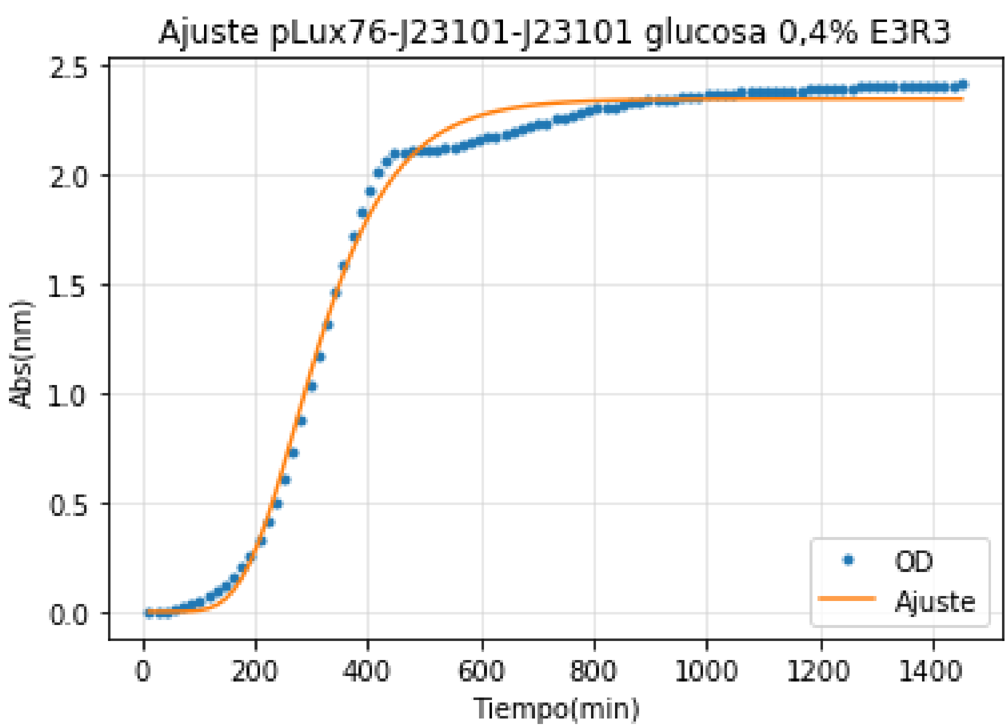




In [43]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glucosa rep 3
y27= np.log(od18213)-np.log(np.min(od18213))
print('Min OD = %e'%((np.min(od18213))))
evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste pLux76-J23101-J23101 glucosa 0,4
A27= params[0]
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[18]
```

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

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

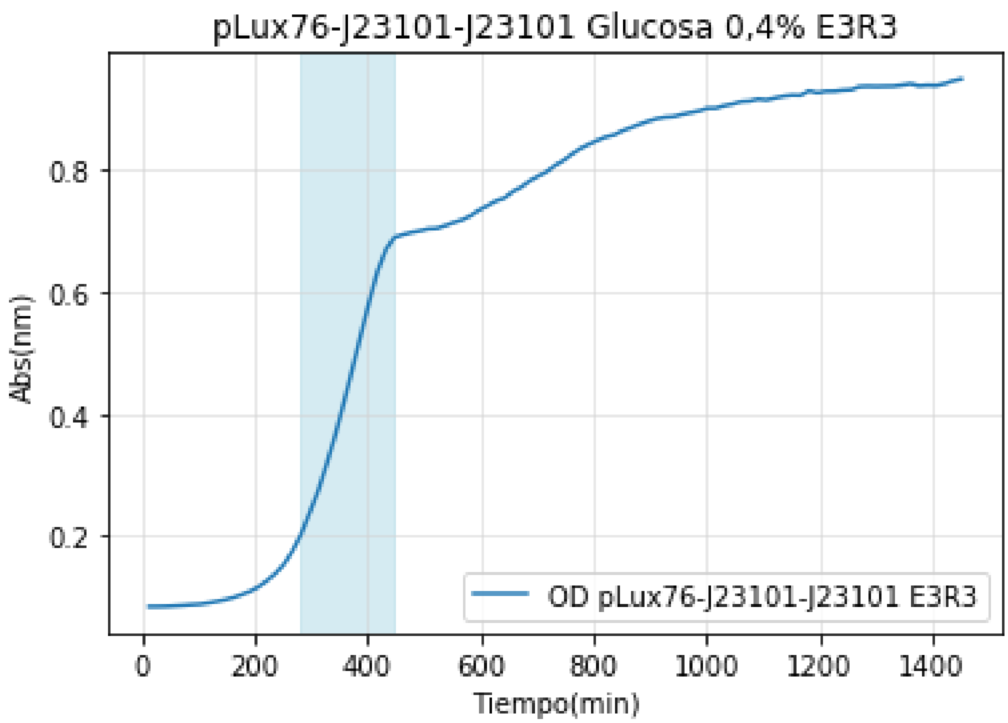
```
Min OD = 8.460000e-02
[  2.34879777e+00   9.01713157e-03   1.76447498e+02]
```

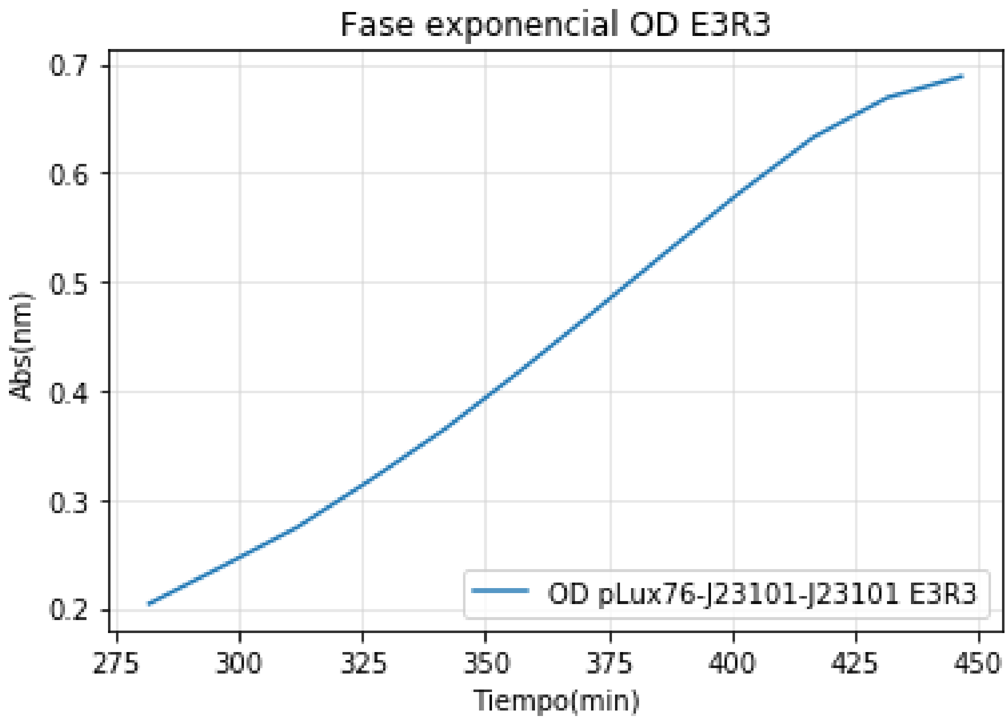



```
A=2.348798e+00
um=9.017132e-03
```

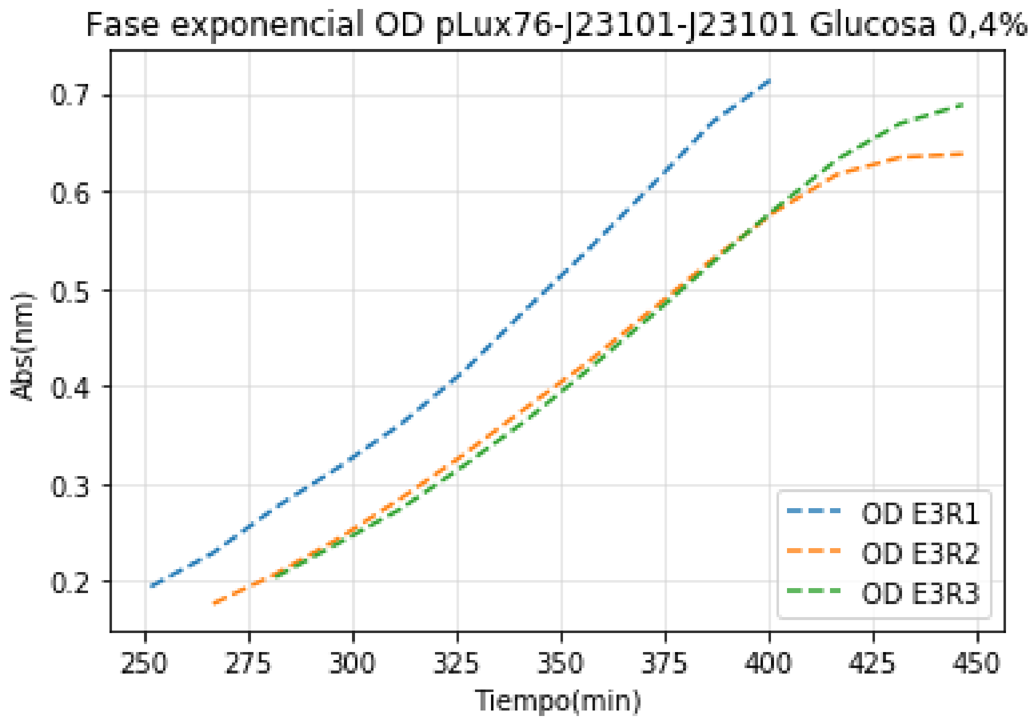
```
l=1.764475e+02
Tm=2.722734e+02
doubpe=7.687003e+01
ext=1.537401e+02
Tfinal=4.260134e+02
```

Out[43]: <matplotlib.legend.Legend at 0x20361502358>





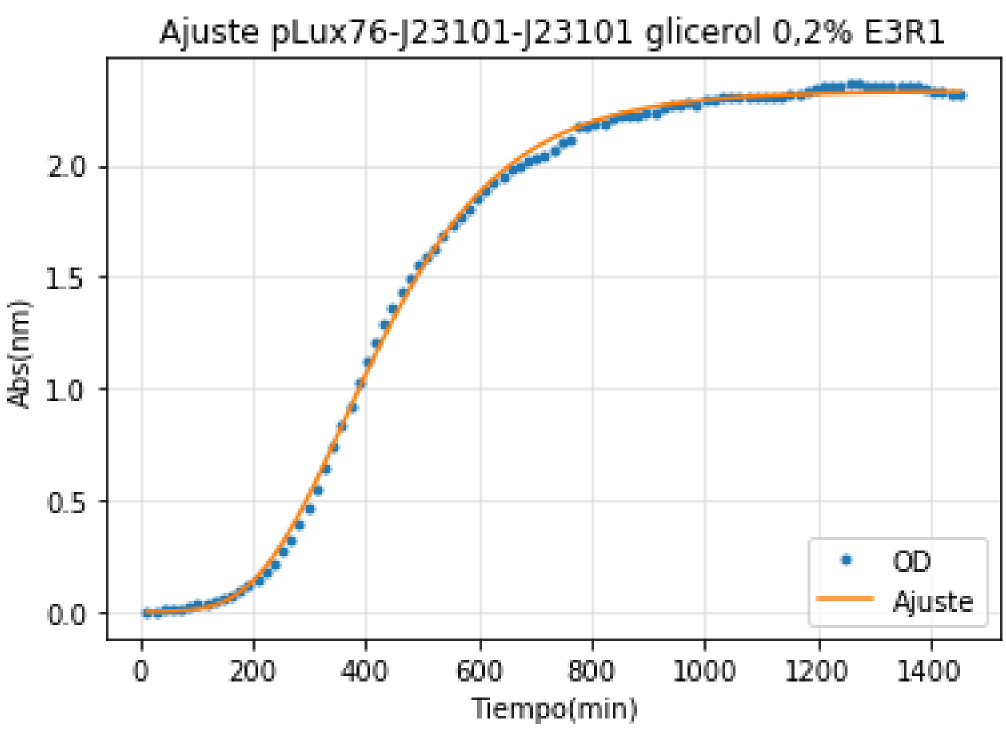

```
In [44]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],od18211[16:27],'--',label='OD E3R1')
         plt.plot(tt[17:30],od18212[17:30],'--',label='OD E3R2')
         plt.plot(tt[18:30],od18213[18:30],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[44]: <matplotlib.legend.Legend at 0x20361af2128>
```




```
In [45]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #plux-std-std glicerol rep 1
         y28= np.log(od1821g1)-np.log(np.min(od1821g1))
         print('Min OD = %e'%((np.min(od1821g1))))
         evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
         A28= params[0]
         um28=params[1]
         l28=params[2]
         print('A=%e'%(A28))
         print('um=%e'%(um28))
         print('l=%e'%(l28))

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

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

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

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

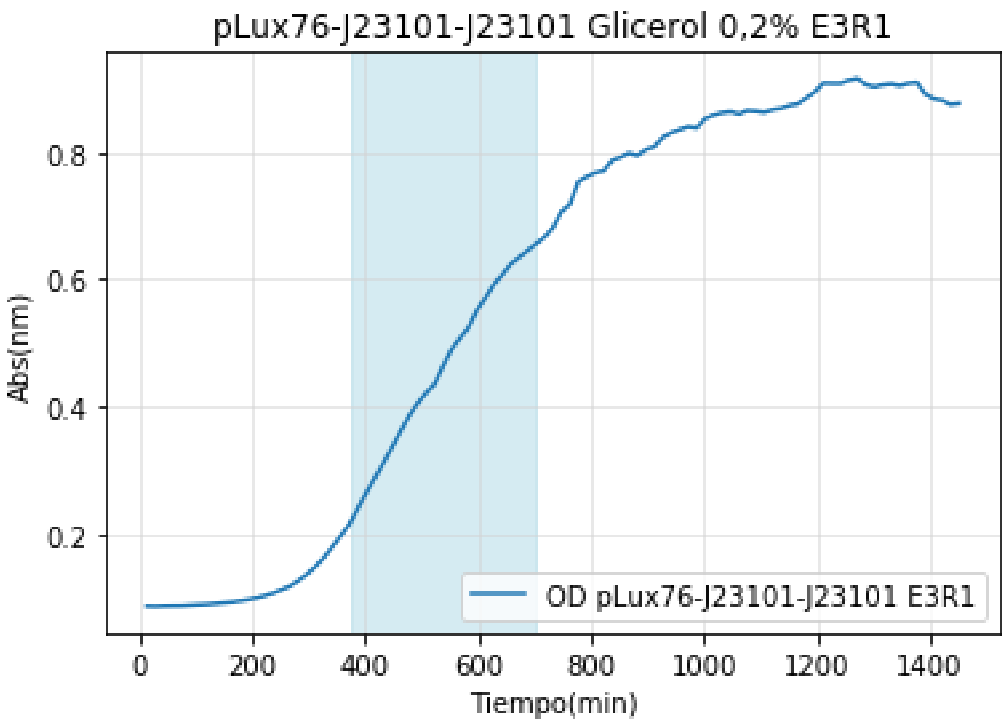
```
Min OD = 8.620000e-02
[  2.33021092e+00   5.48552233e-03   2.05388922e+02]
```

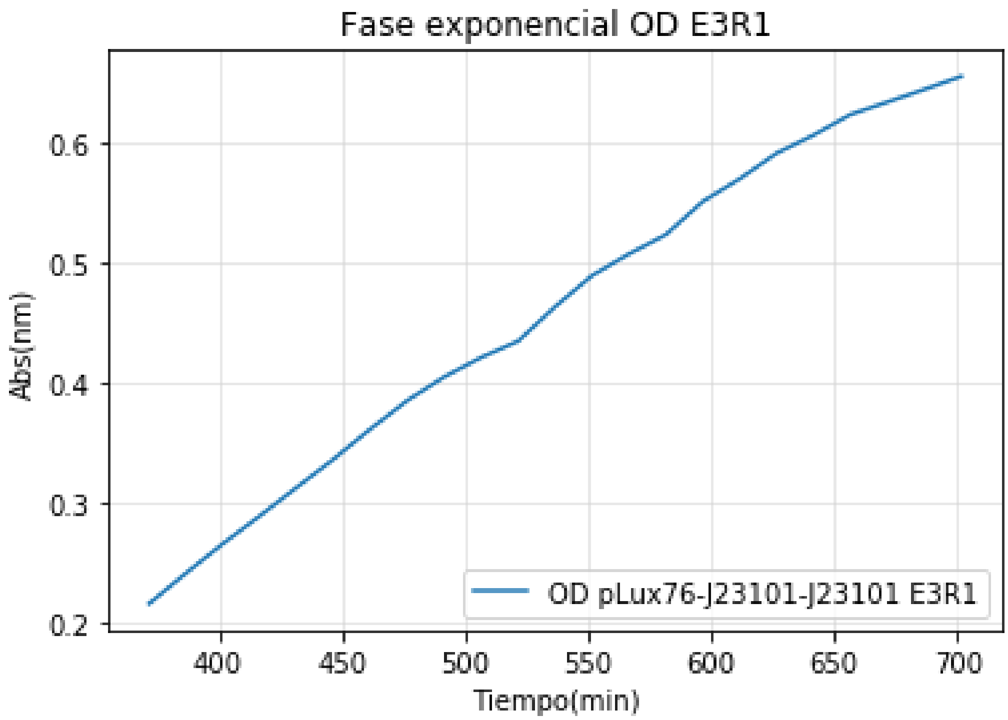



```
A=2.330211e+00
um=5.485522e-03
```

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

```
Out[45]: <matplotlib.legend.Legend at 0x20362c546a0>
```

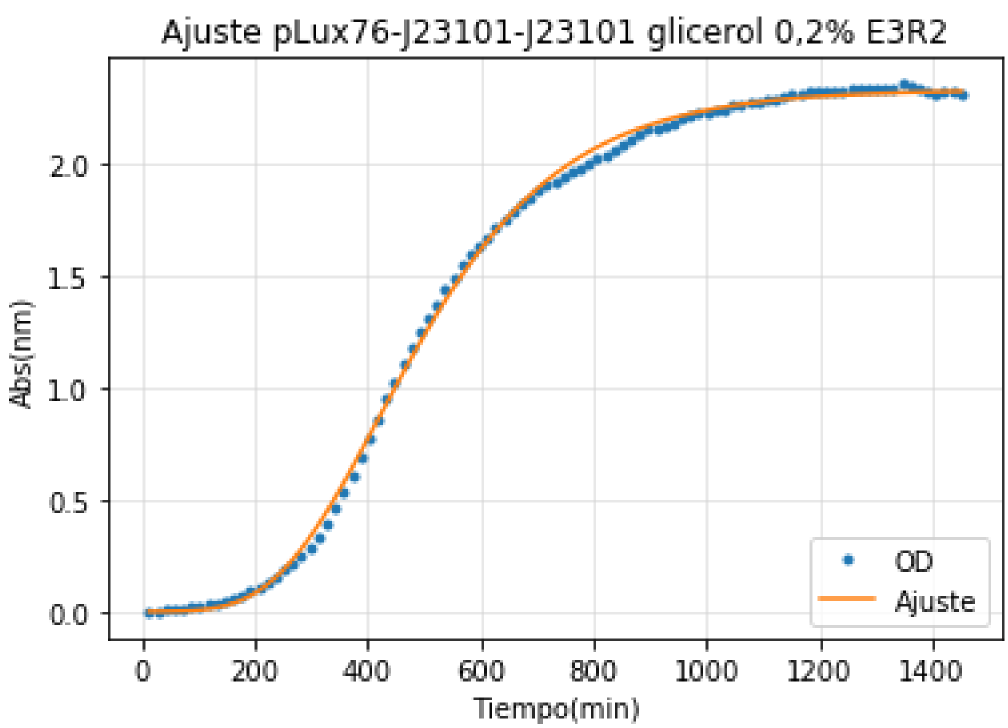




In [46]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glicerol rep 2
y29= np.log(od1821g2)-np.log(np.min(od1821g2))
print('Min OD = %e'%((np.min(od1821g2))))
evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
A29= params[0]
um29=params[1]
l29=params[2]
print('A=%e'%(A29))
print('um=%e'%(um29))
print('l=%e'%(l29))

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

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

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

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

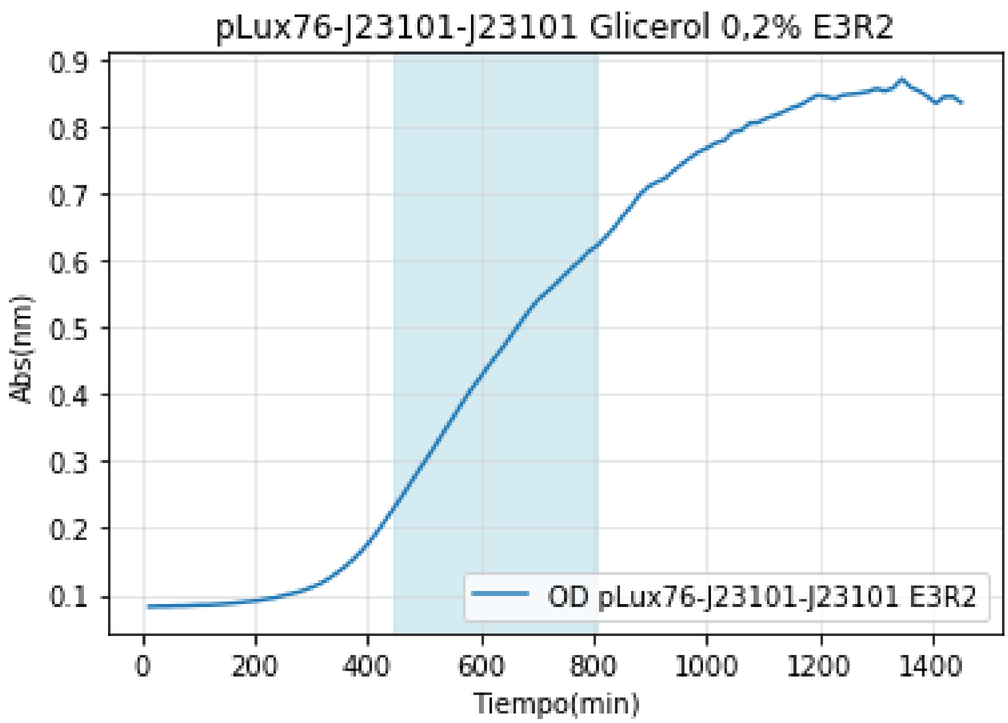
```
Min OD = 8.260000e-02
[  2.33071292e+00   4.73754815e-03   2.36596064e+02]
```

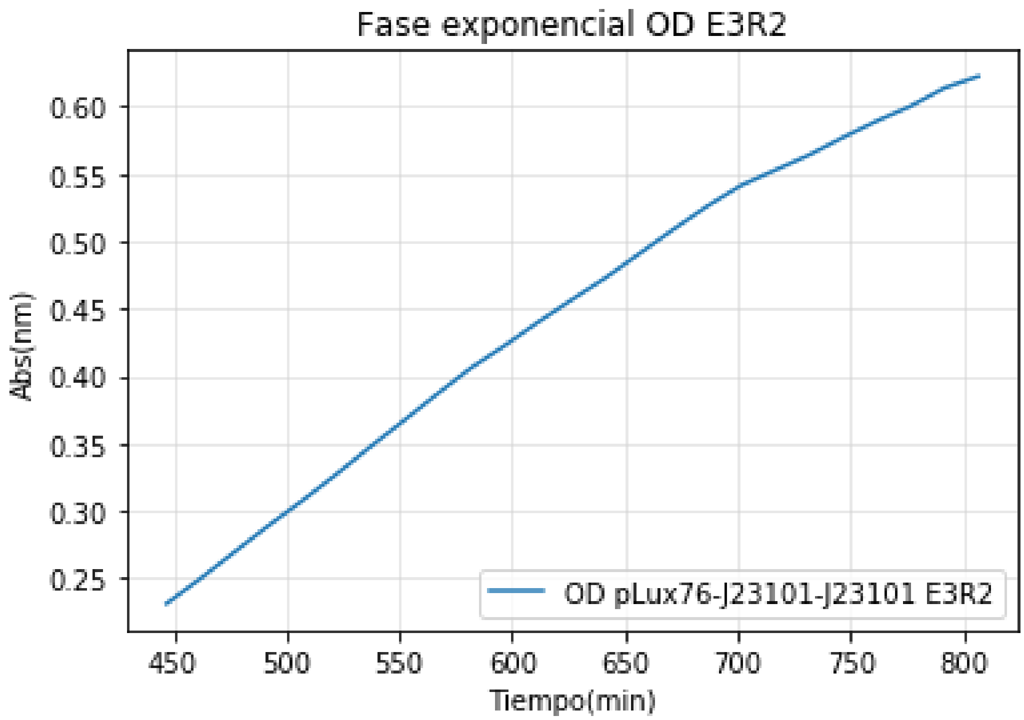



```
A=2.330713e+00
um=4.737548e-03
```

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

```
Out[46]: <matplotlib.legend.Legend at 0x20362e45ba8>
```

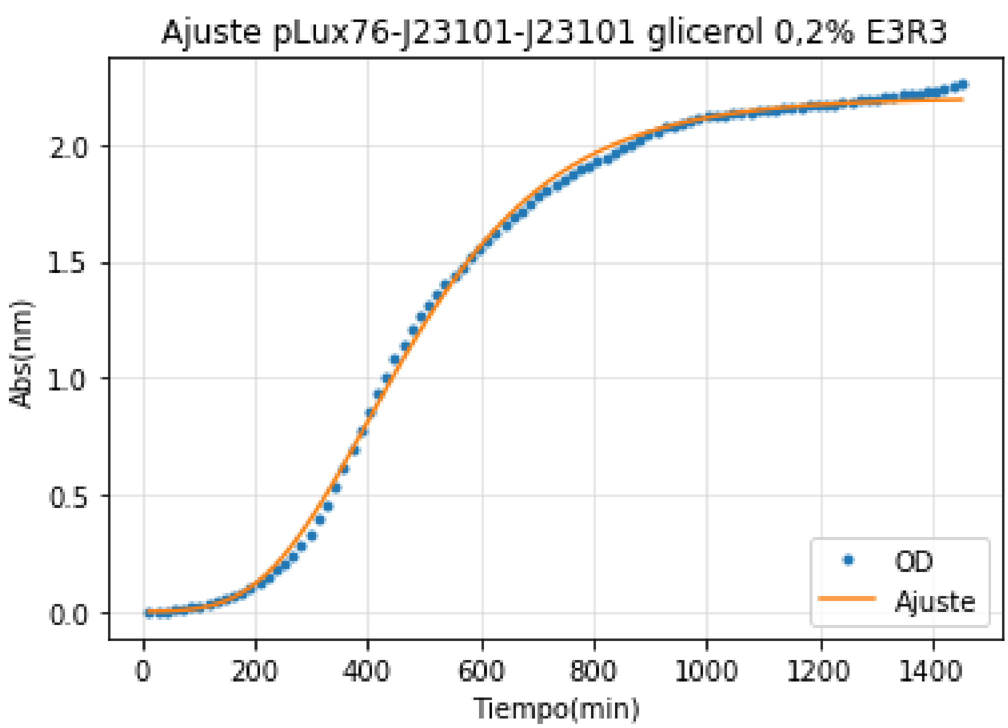




In [47]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#plux-std-std glicerol rep 3
y30= np.log(od1821g3)-np.log(np.min(od1821g3))
print('Min OD = %e'%((np.min(od1821g3))))
evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste pLux76-J23101-J23101 glicerol 0,
A30= params[0]
um30=params[1]
l30=params[2]
print('A=%e'%(A30))
print('um=%e'%(um30))
print('l=%e'%(l30))

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

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

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

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

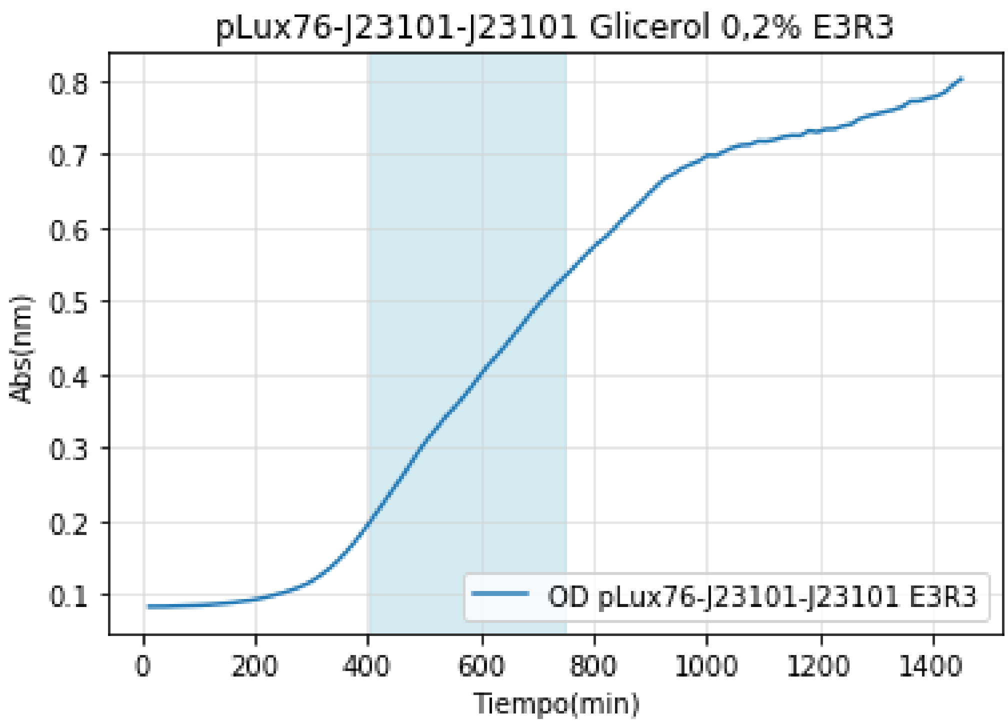
```
Min OD = 8.390000e-02
[  2.19954398e+00   4.34680854e-03   2.12218285e+02]
```

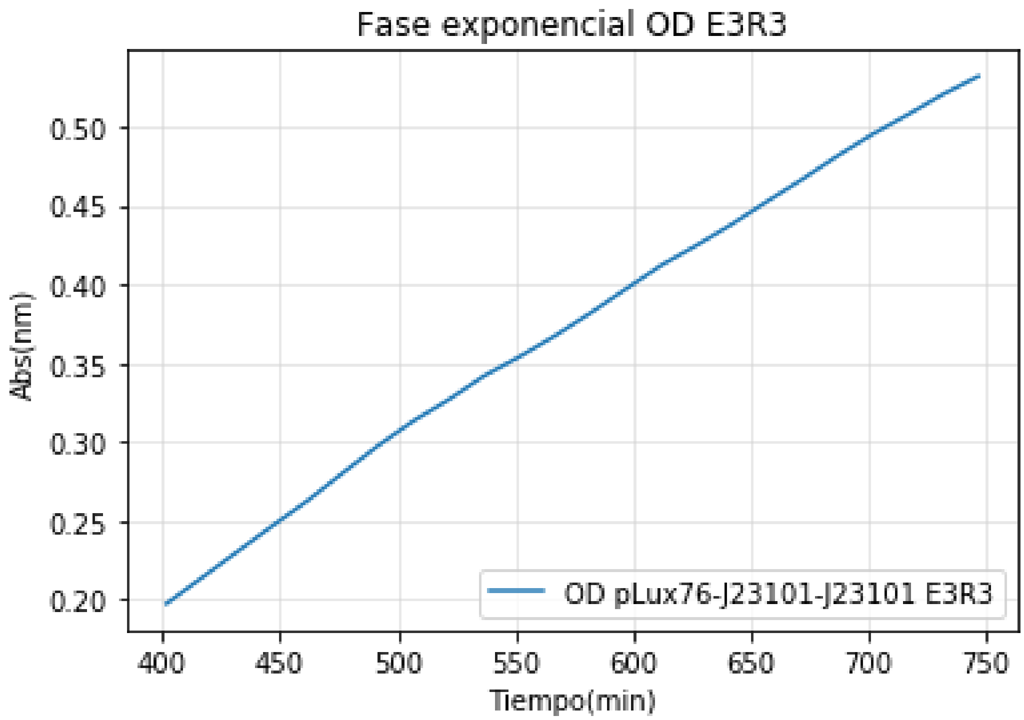



```
A=2.199544e+00
um=4.346809e-03
```

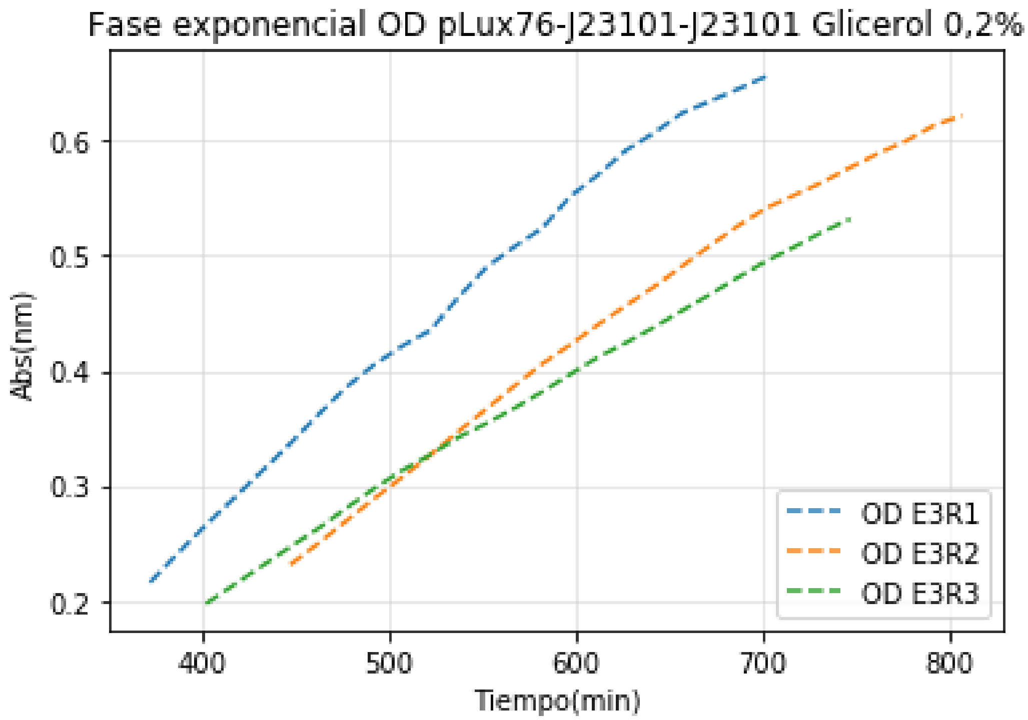
```
l=2.122183e+02
Tm=3.983703e+02
doubpe=1.594612e+02
ext=3.189223e+02
Tfinal=7.172926e+02
```

Out[47]: <matplotlib.legend.Legend at 0x20361982518>

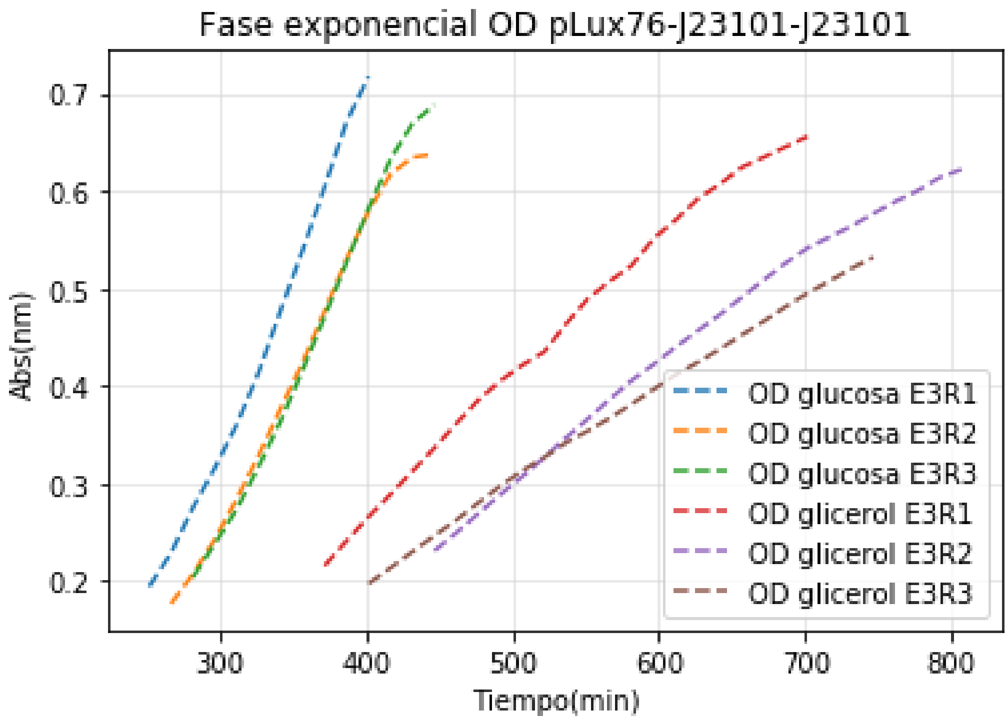





```
In [48]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[24:47],od1821g1[24:47],'--',label='OD E3R1')
         plt.plot(tt[29:54],od1821g2[29:54],'--',label='OD E3R2')
         plt.plot(tt[26:50],od1821g3[26:50],'--',label='OD E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[48]: <matplotlib.legend.Legend at 0x20360d94588>
```



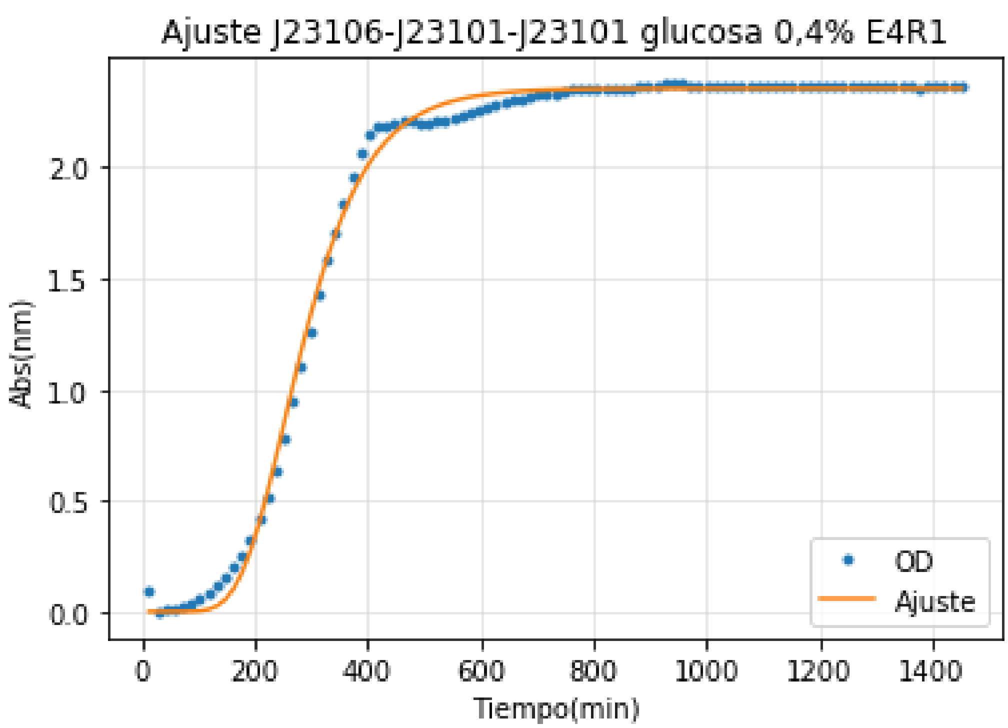

```
In [49]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:27],od18211[16:27],'--',label='OD glucosa E3R1')
         plt.plot(tt[17:30],od18212[17:30],'--',label='OD glucosa E3R2')
         plt.plot(tt[18:30],od18213[18:30],'--',label='OD glucosa E3R3')
         plt.plot(tt[24:47],od1821g1[24:47],'--',label='OD glicerol E3R1')
         plt.plot(tt[29:54],od1821g2[29:54],'--',label='OD glicerol E3R2')
         plt.plot(tt[26:50],od1821g3[26:50],'--',label='OD glicerol E3R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[49]: <matplotlib.legend.Legend at 0x203619e6160>
```



In [50]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 1
y31= np.log(od12211)-np.log(np.min(od12211))
print('Min OD = %e'%((np.min(od12211))))
evaly, params=Function_fit(tt,y31,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4
A31 = params[0]
um31=params[1]
l31=params[2]
print('A=%e'%(A31))
print('um=%e'%(um31))
print('l=%e'%(l31))

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

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

```python
y2=tt[26]
plt.figure()
plt.title('J23106-J23101-J23101 Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12211,label='OD J23106-J23101-J23101 E4R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:27],od12211[17:27],label='OD J23106-J23101-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

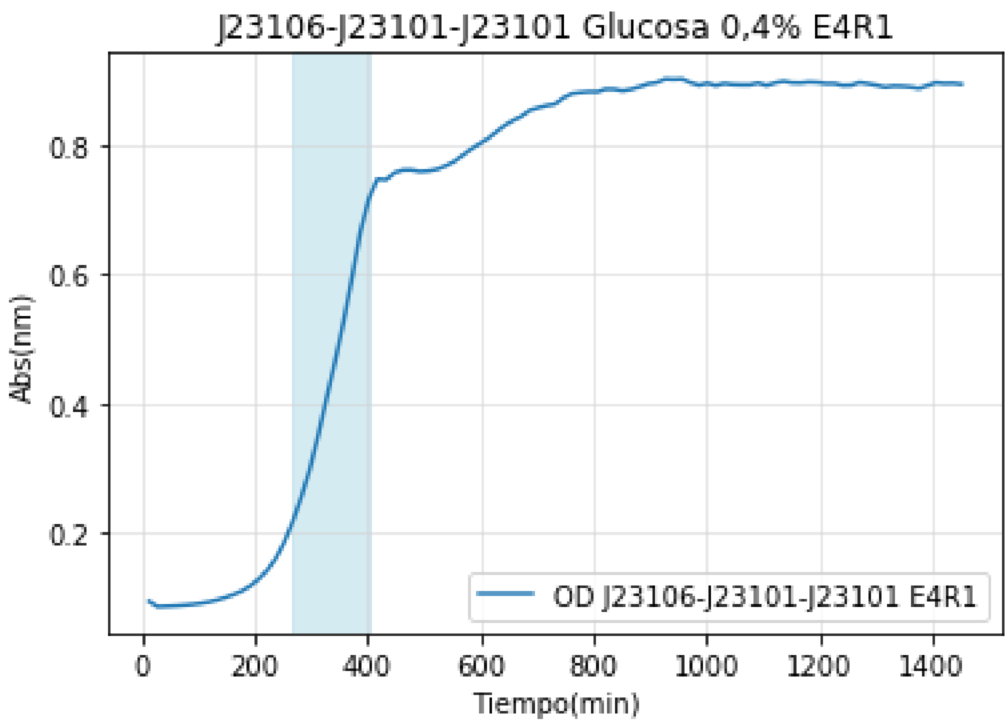
```
Min OD = 8.410000e-02
[  2.35417955e+00   1.08424351e-02   1.72836449e+02]
```

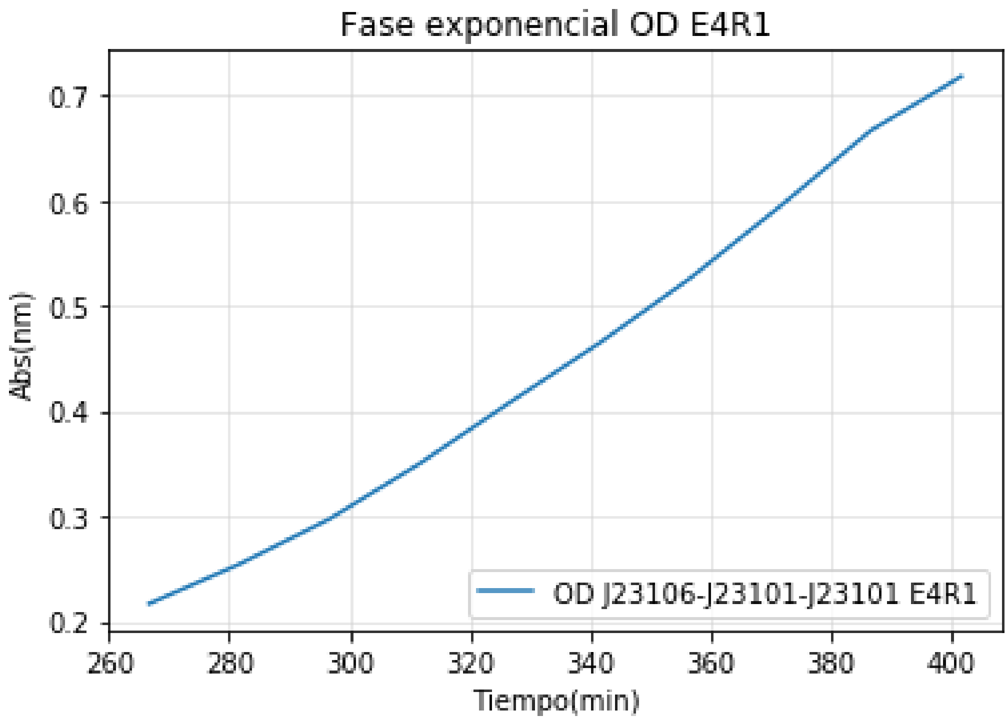



```
A=2.354180e+00
um=1.084244e-02
```

```
l=1.728364e+02
Tm=2.527128e+02
doubpe=6.392911e+01
ext=1.278582e+02
Tfinal=3.805710e+02
```

Out[50]: `<matplotlib.legend.Legend at 0x203619bb080>`

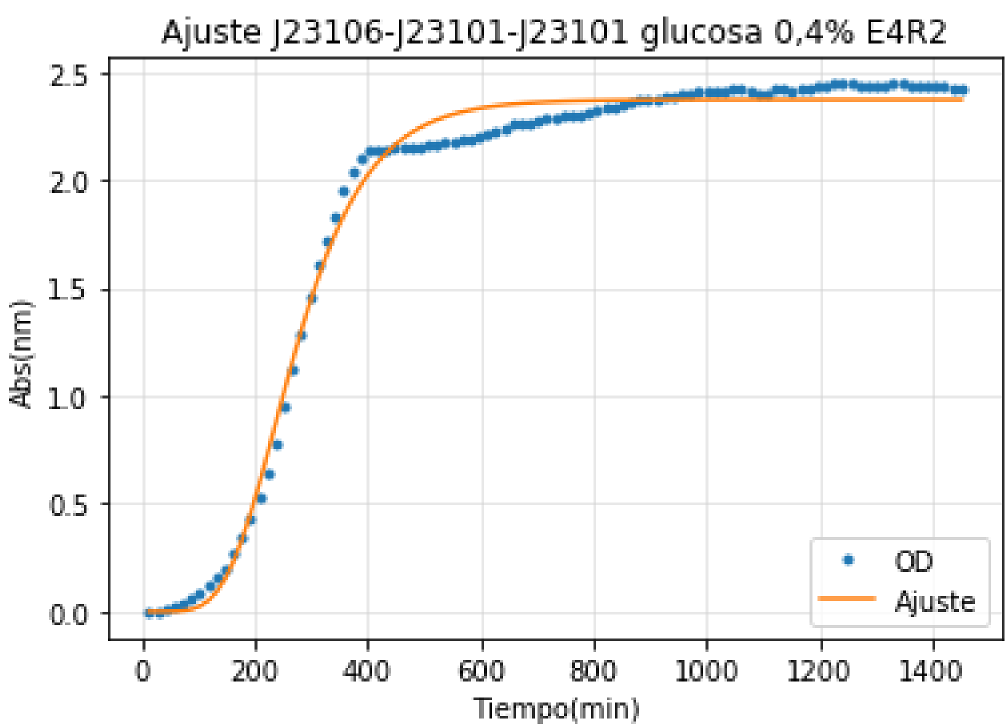





```
In [51]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-std-std glucosa rep 2
         y32= np.log(od12212)-np.log(np.min(od12212))
         print('Min OD = %e'%((np.min(od12212))))
         evaly, params=Function_fit(tt,y32,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4
         A32= params[0]
         um32=params[1]
         l32=params[2]
         print('A=%e'%(A32))
         print('um=%e'%(um32))
         print('l=%e'%(l32))

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

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

```python
y2=tt[26]
plt.figure()
plt.title('J23106-J23101-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12212,label='OD J23106-J23101-J23101 E4R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[15:27],od12212[15:27],label='OD pLux76-J23101-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

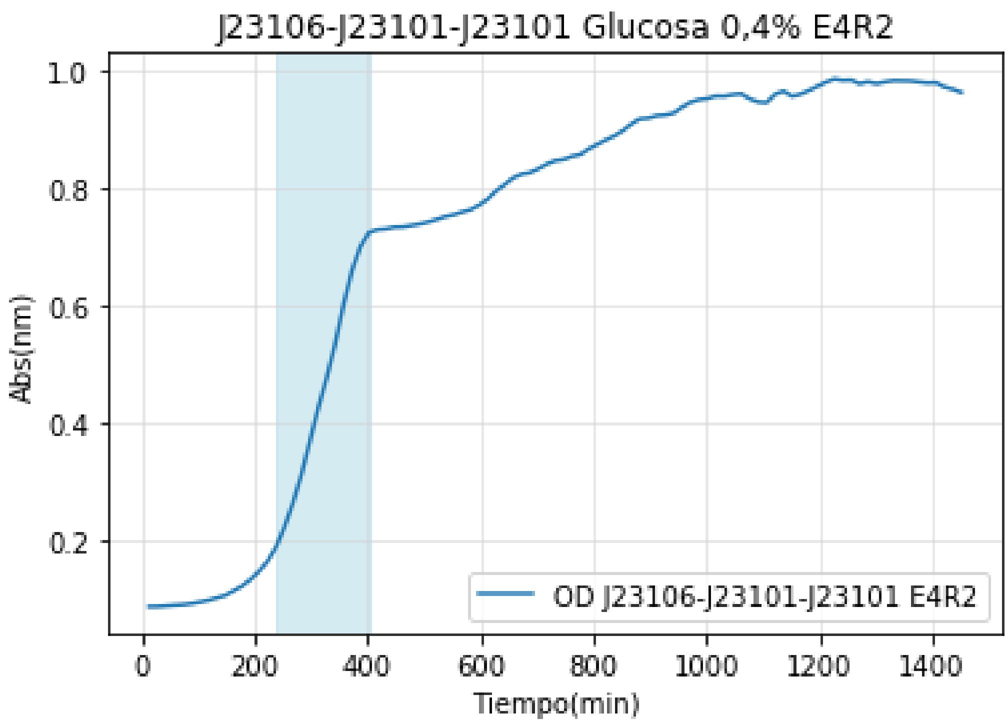
```
Min OD = 8.550000e-02
[  2.37271494e+00   9.82381102e-03   1.47415571e+02]
```

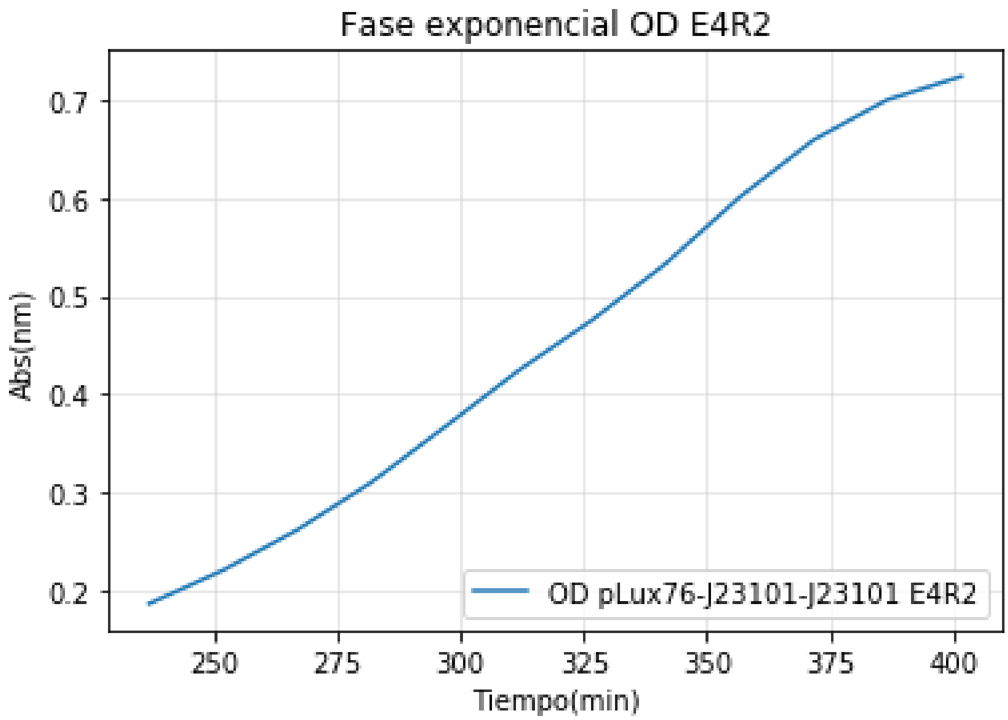



```
A=2.372715e+00
um=9.823811e-03
```

```
l=1.474156e+02
Tm=2.362684e+02
doubpe=7.055787e+01
ext=1.411157e+02
Tfinal=3.773841e+02
```

```
Out[51]: <matplotlib.legend.Legend at 0x20363083128>
```

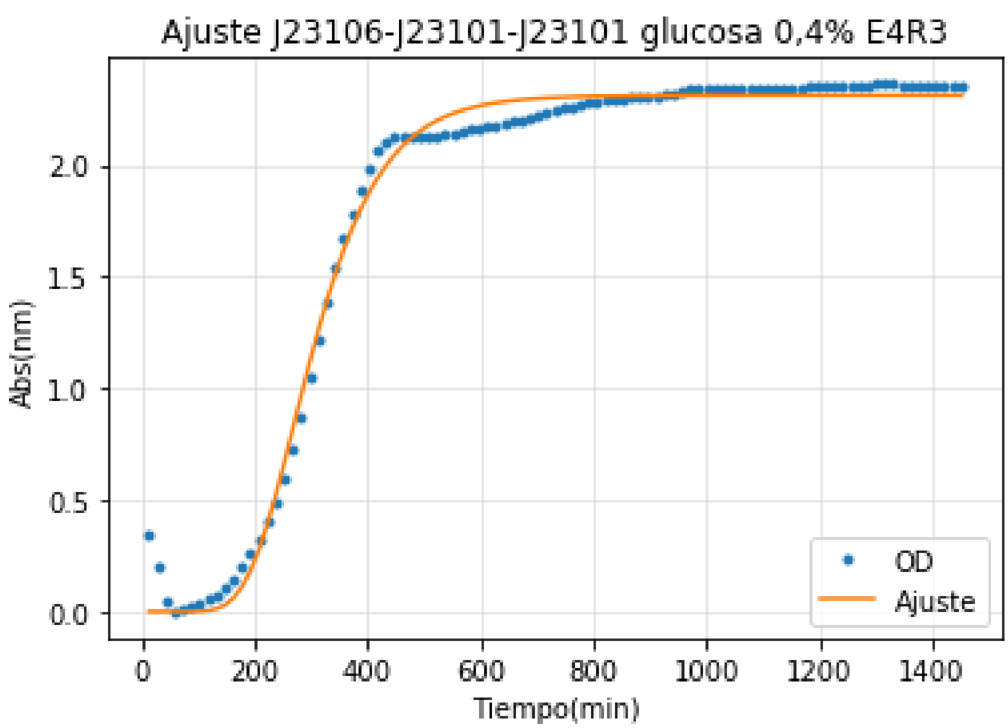





```
In [52]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-std-std glucosa rep 3
         y33= np.log(od12213)-np.log(np.min(od12213))
         print('Min OD = %e'%((np.min(od12213))))
         evaly, params=Function_fit(tt,y33,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4
         A33= params[0]
         um33=params[1]
         l33=params[2]
         print('A=%e'%(A33))
         print('um=%e'%(um33))
         print('l=%e'%(l33))

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

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

```
y2=tt[28]
plt.figure()
plt.title('J23106-J23101-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12213,label='OD J23106-J23101-J23101 E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[18:29],od12213[18:29],label='OD J23106-J23101-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

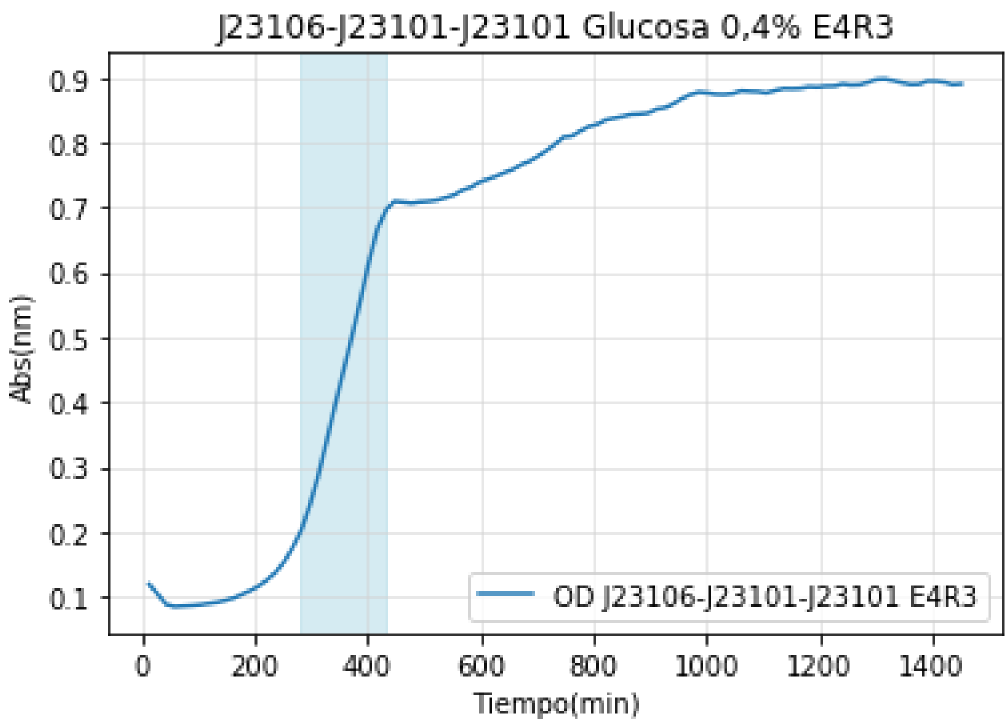
```
Min OD = 8.460000e-02
[  2.31183042e+00   1.00988565e-02   1.85588268e+02]
```

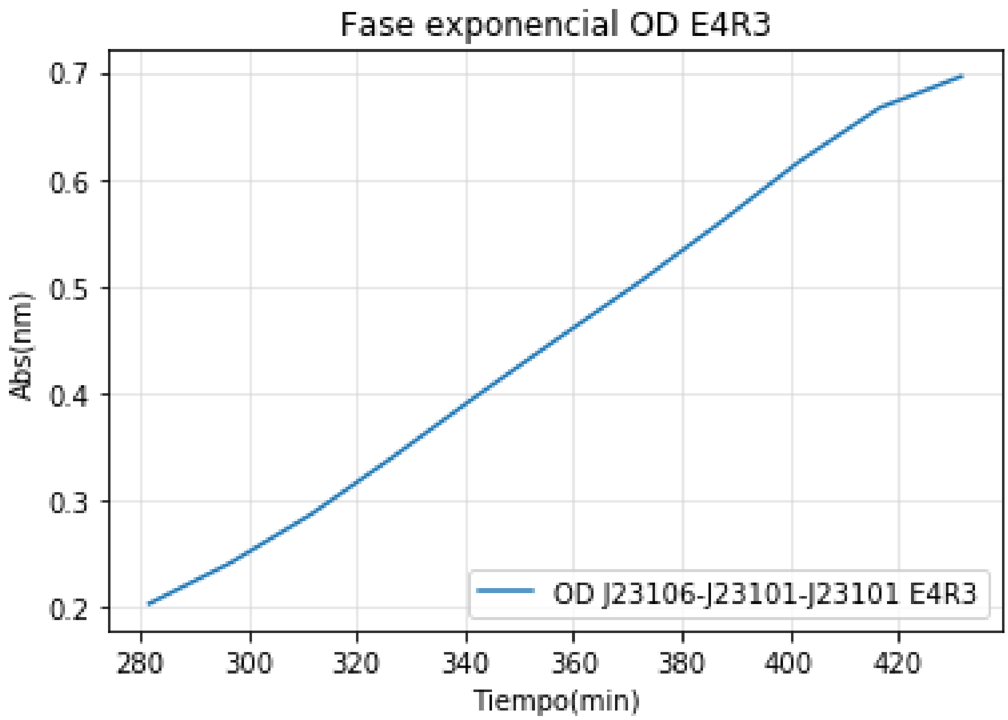



```
A=2.311830e+00
um=1.009886e-02
```

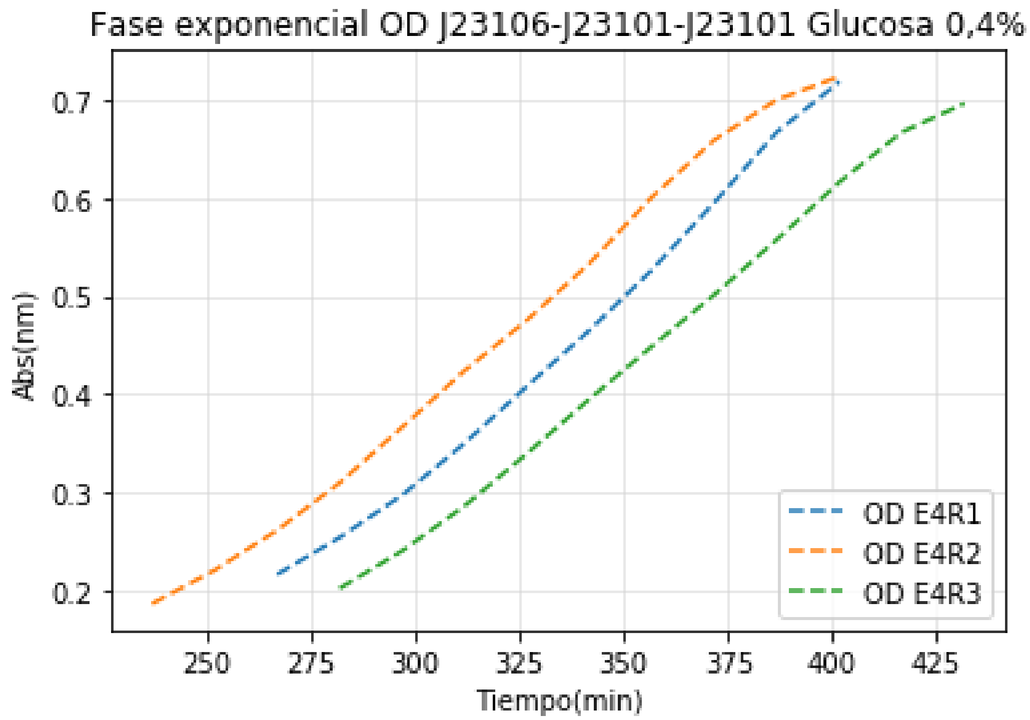
```
l=1.855883e+02
Tm=2.698032e+02
doubpe=6.863620e+01
ext=1.372724e+02
Tfinal=4.070756e+02
```

```
Out[52]: <matplotlib.legend.Legend at 0x2036327f2e8>
```





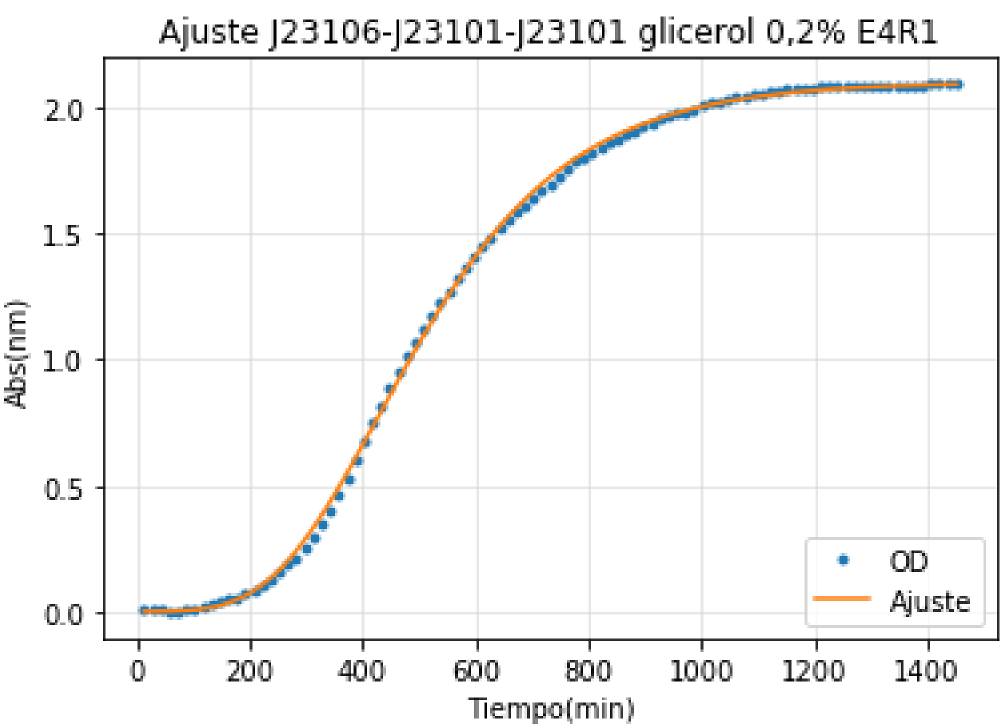

```
In [53]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:27],od12211[17:27],'--',label='OD E4R1')
         plt.plot(tt[15:27],od12212[15:27],'--',label='OD E4R2')
         plt.plot(tt[18:29],od12213[18:29],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[53]: <matplotlib.legend.Legend at 0x203631fe7b8>
```



In [54]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 1
y34= np.log(od1221g1)-np.log(np.min(od1221g1))
print('Min OD = %e'%((np.min(od1221g1))))
evaly, params=Function_fit(tt,y34,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A34= params[0]
um34=params[1]
l34=params[2]
print('A=%e'%(A34))
print('um=%e'%(um34))
print('l=%e'%(l34))

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

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

```python
y2=tt[57]
plt.figure()
plt.title('J23106-J23101-J23101 Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1221g1,label='OD J23106-J23101-J23101 E4R1')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[28:58],od1221g1[28:58],label='OD J23106-J23101-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

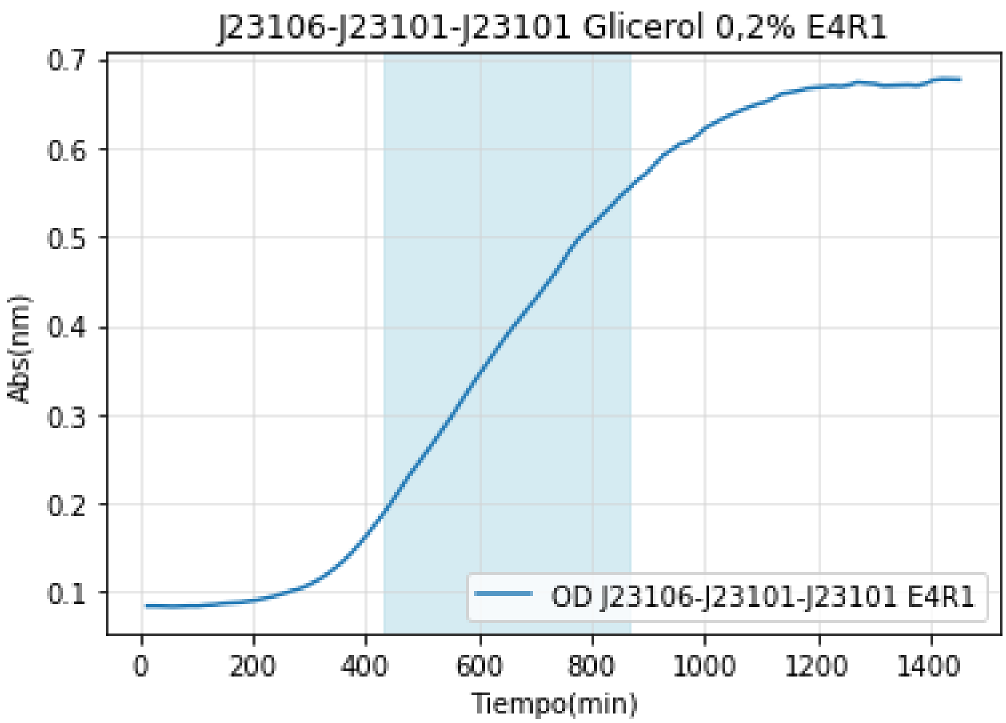
```
Min OD = 8.360000e-02
[  2.10005866e+00   4.10748730e-03   2.38628147e+02]
```

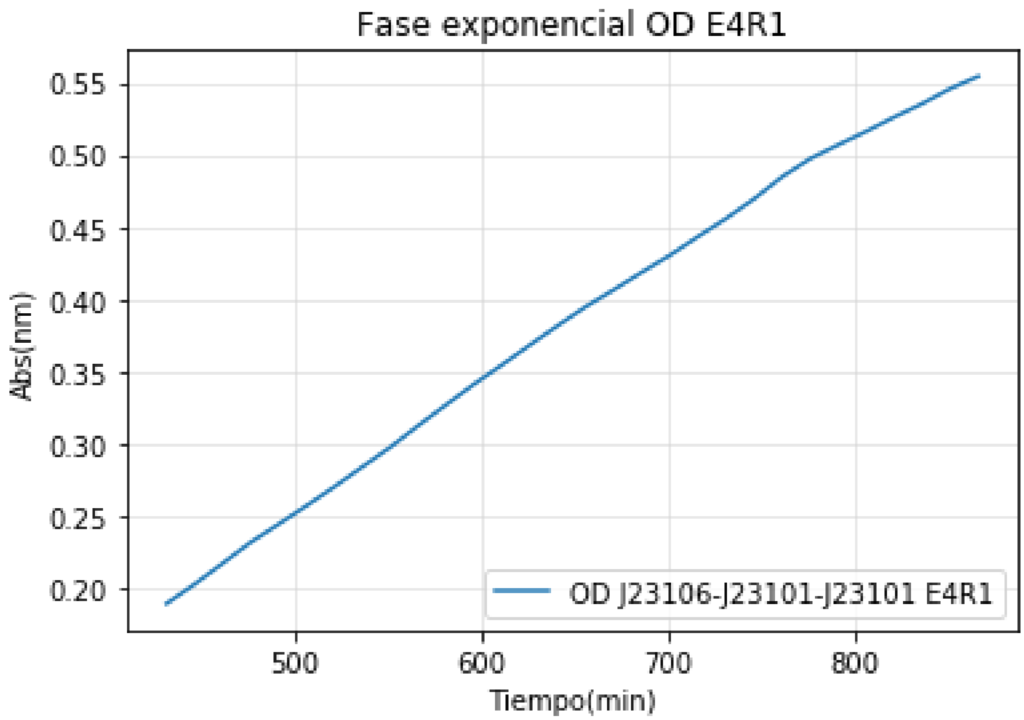



```
A=2.100059e+00
um=4.107487e-03
```

```
l=2.386281e+02
Tm=4.267160e+02
doubpe=1.687521e+02
ext=4.218803e+02
Tfinal=8.485963e+02
```

```
Out[54]: <matplotlib.legend.Legend at 0x2035f9f9550>
```

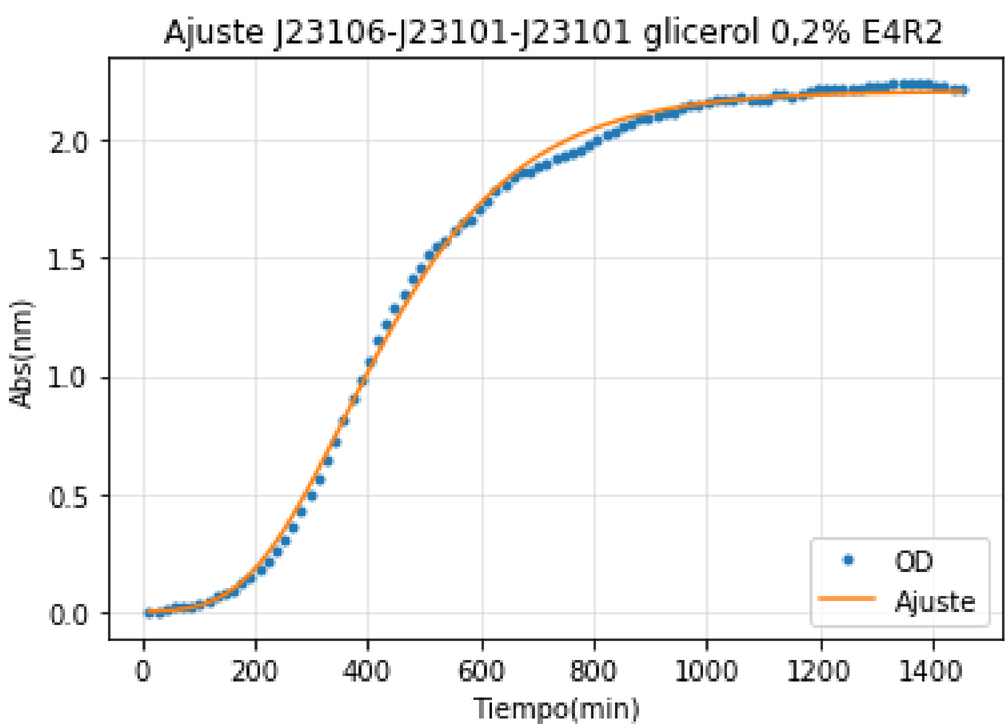




In [55]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 2
y35= np.log(od1221g2)-np.log(np.min(od1221g2))
print('Min OD = %e'%((np.min(od1221g2))))
evaly, params=Function_fit(tt,y35,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A35= params[0]
um35=params[1]
l35=params[2]
print('A=%e'%(A35))
print('um=%e'%(um35))
print('l=%e'%(l35))

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

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

```
y2=tt[49]
plt.figure()
plt.title('J23106-J23101-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1221g2,label='OD J23106-J23101-J23101 E4R2')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[23:50],od1221g2[23:50],label='OD J23106-J23101-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

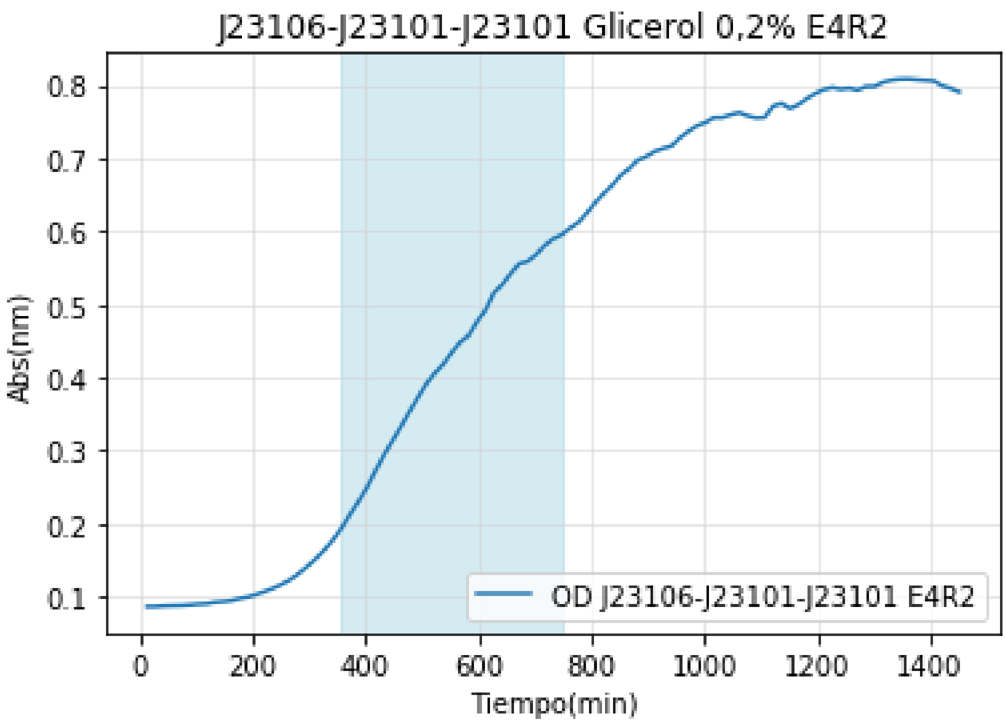
```
Min OD = 8.630000e-02
[  2.20913289e+00   4.73407774e-03   1.84774865e+02]
```

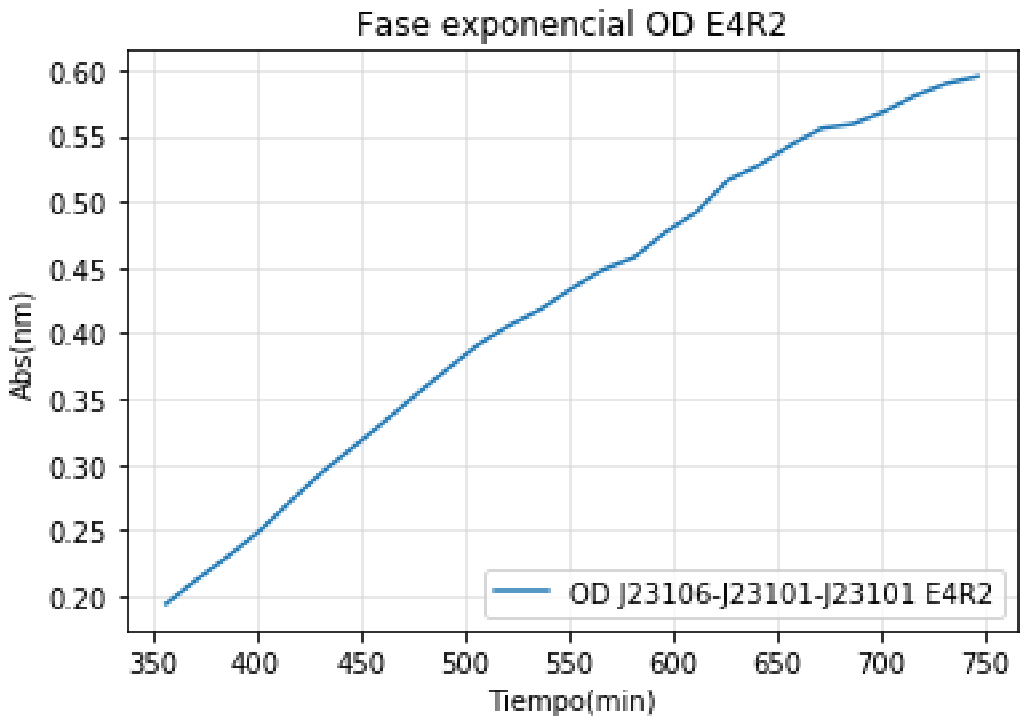



```
A=2.209133e+00
um=4.734078e-03
```

```
l=1.847749e+02
Tm=3.564439e+02
doubpe=1.464165e+02
ext=3.660413e+02
Tfinal=7.224852e+02
```

```
Out[55]: <matplotlib.legend.Legend at 0x203630177b8>
```

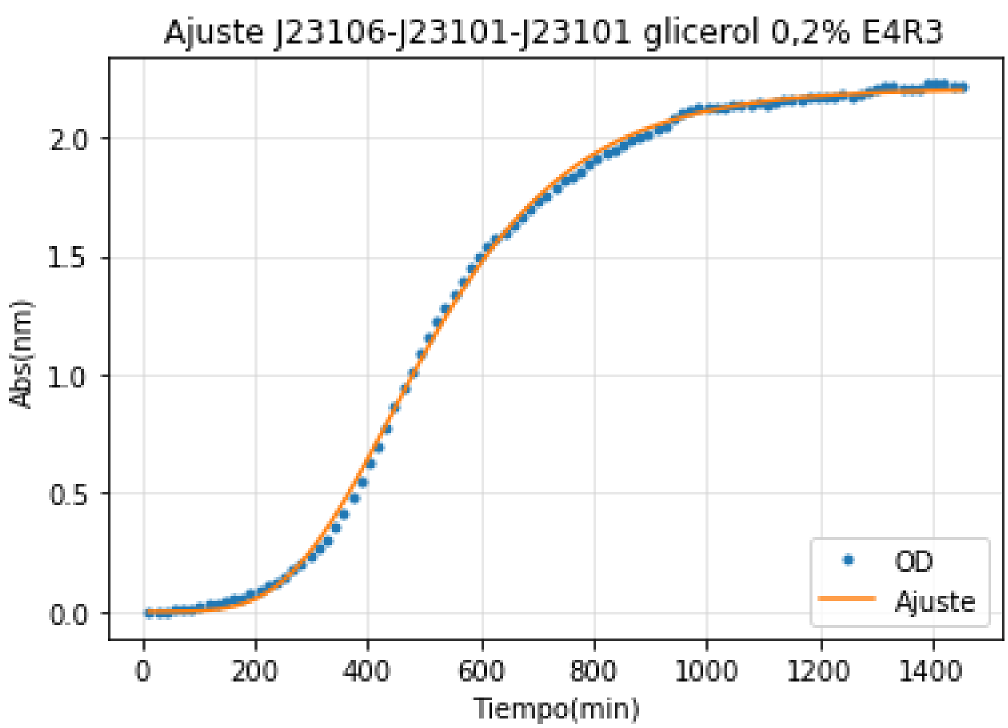




In [56]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 3
y36= np.log(od1221g3)-np.log(np.min(od1221g3))
print('Min OD = %e'%((np.min(od1221g3))))
evaly, params=Function_fit(tt,y36,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A36= params[0]
um36=params[1]
l36=params[2]
print('A=%e'%(A36))
print('um=%e'%(um36))
print('l=%e'%(l36))

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

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

```
y2=tt[51]
plt.figure()
plt.title('J23106-J23101-J23101 Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1221g3,label='OD J23106-J23101-J23101 E4R3')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[29:52],od1221g3[29:52],label='OD J23106-J23101-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

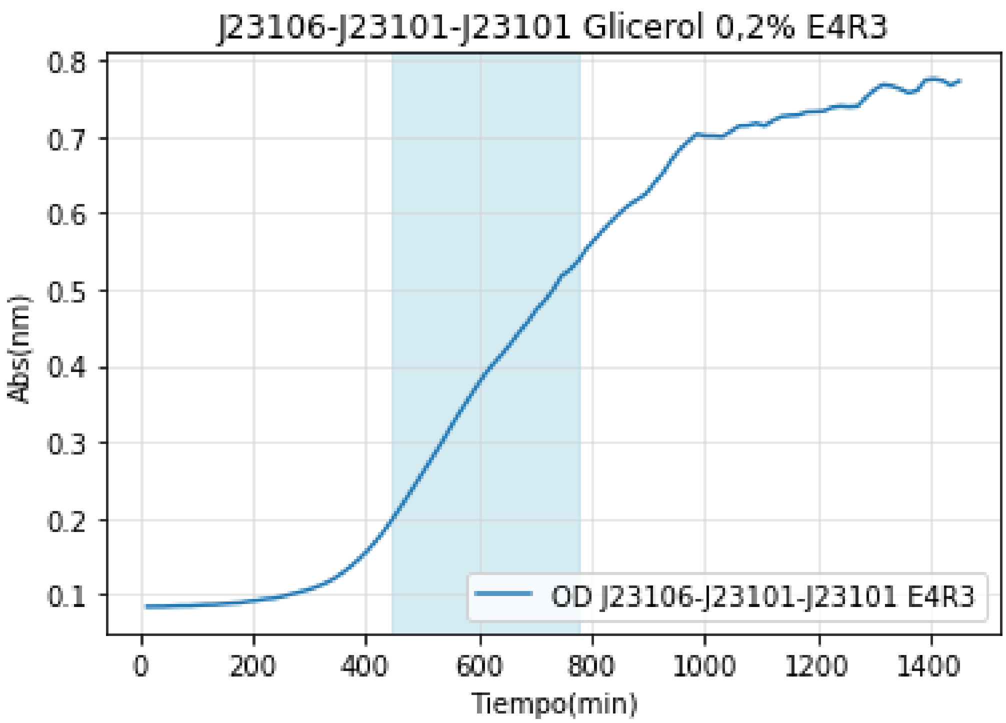
```
Min OD = 8.390000e-02
[  2.20537307e+00   4.48287691e-03   2.56148136e+02]
```

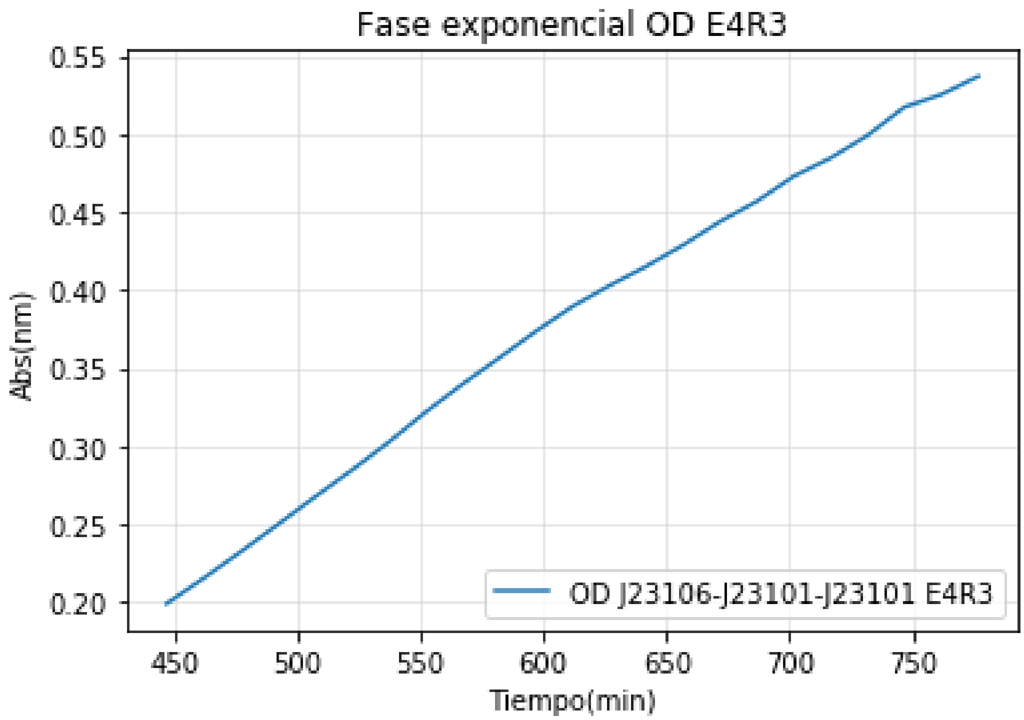



```
A=2.205373e+00
um=4.482877e-03
```

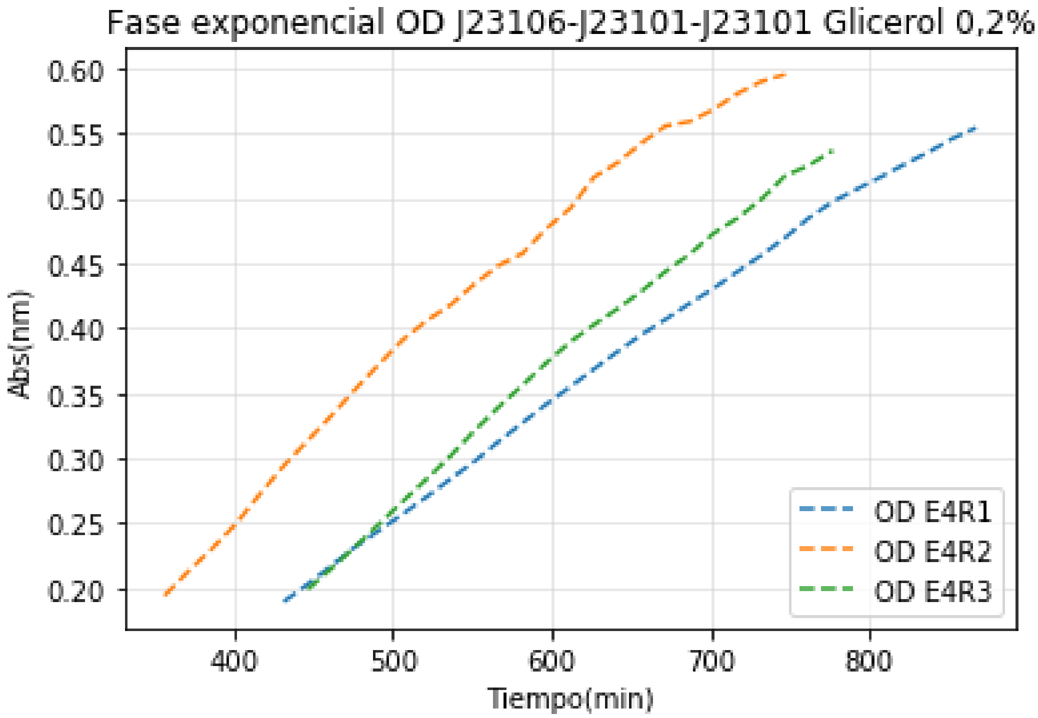
```
l=2.561481e+02
Tm=4.371282e+02
doubpe=1.546211e+02
ext=3.092421e+02
Tfinal=7.463703e+02
```

```
Out[56]: <matplotlib.legend.Legend at 0x203633eb0b8>
```

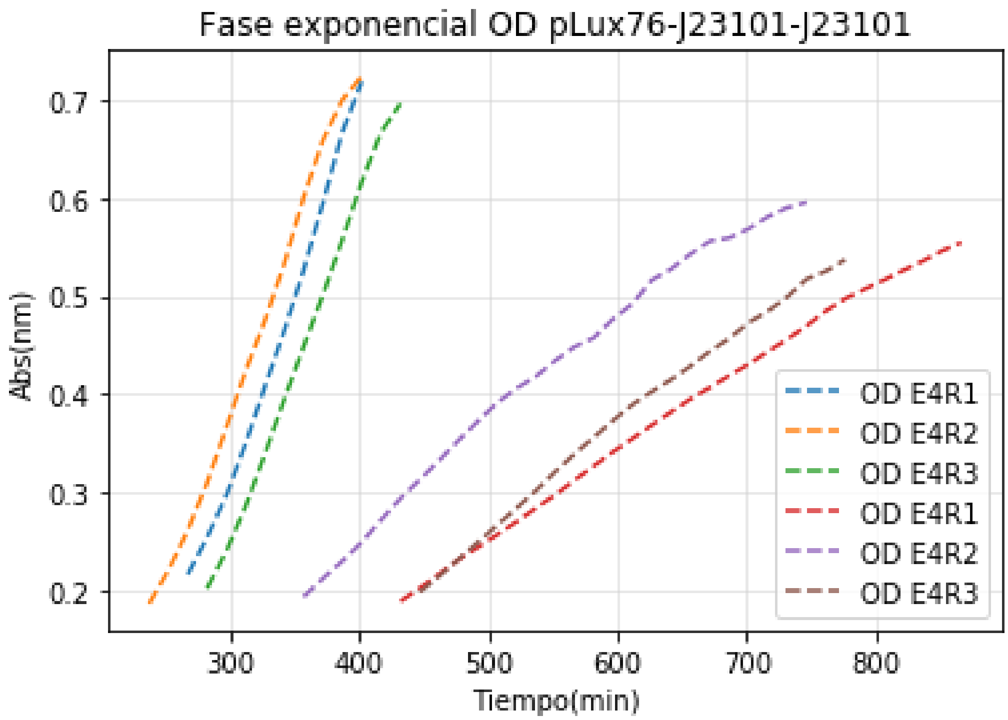





```
In [57]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[28:58],od1221g1[28:58],'--',label='OD E4R1')
         plt.plot(tt[23:50],od1221g2[23:50],'--',label='OD E4R2')
         plt.plot(tt[29:52],od1221g3[29:52],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[57]: <matplotlib.legend.Legend at 0x2036351bbe0>
```




```
In [58]: #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[17:27],od12211[17:27],'--',label='OD E4R1')
         plt.plot(tt[15:27],od12212[15:27],'--',label='OD E4R2')
         plt.plot(tt[18:29],od12213[18:29],'--',label='OD E4R3')
         plt.plot(tt[28:58],od1221g1[28:58],'--',label='OD E4R1')
         plt.plot(tt[23:50],od1221g2[23:50],'--',label='OD E4R2')
         plt.plot(tt[29:52],od1221g3[29:52],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[58]: <matplotlib.legend.Legend at 0x203635e2c18>
```



In [59]:
```
#Selección de datos en arrays, según lo determinado
#controles glucosa 15
o1=odcg151[19:30]
c1=cfpcg151[19:30]
r1=rfpcg151[19:30]
y1=yfpcg151[19:30]

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

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

#controles glicerol 15
o4=odcgl151[24:45]
c4=cfpcgl151[24:45]
r4=rfpcgl151[24:45]
y4=yfpcgl151[24:45]

o5=odcgl152[23:46]
```

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

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

#controles glucosa 18
o7=odcg181[16:26]
c7=cfpcg181[16:26]
r7=rfpcg181[16:26]
y7=yfpcg181[16:26]

o8=odcg182[17:30]
c8=cfpcg182[17:30]
r8=rfpcg182[17:30]
y8=yfpcg182[17:30]

o9=odcg183[17:29]
c9=cfpcg183[17:29]
r9=rfpcg183[17:29]
y9=yfpcg183[17:29]

#controles glicerol 18
o10=odcgl181[25:48]
c10=cfpcgl181[25:48]
r10=rfpcgl181[25:48]
y10=yfpcgl181[25:48]

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

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

#controles glucosa 12
o13=odcg121[17:28]
c13=cfpcg121[17:28]
r13=rfpcg121[17:28]
y13=yfpcg121[17:28]

o14=odcg122[14:25]
```

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

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

#controles glicerol 12
o16=odcgl121[24:52]
c16=cfpcgl121[24:52]
r16=rfpcgl121[24:52]
y16=yfpcgl121[24:52]

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

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

#ptet-std-std glucosa
o19=od15211[19:30]
c19=cfp15211[19:30]
r19=rfp15211[19:30]
y19=yfp15211[19:30]

o20=od15212[19:32]
c20=cfp15212[19:32]
r20=rfp15212[19:32]
y20=yfp15212[19:32]

o21=od15213[16:28]
c21=cfp15213[16:28]
r21=rfp15213[16:28]
y21=yfp15213[16:28]

#ptet-std-std glicerol
o22=od1521g1[27:48]
c22=cfp1521g1[27:48]
r22=rfp1521g1[27:48]
y22=yfp1521g1[27:48]

o23=od1521g2[28:53]
```

```
c23=cfp1521g2[28:53]
r23=rfp1521g2[28:53]
y23=yfp1521g2[28:53]

o24=od1521g3[27:53]
c24=cfp1521g3[27:53]
r24=rfp1521g3[27:53]
y24=yfp1521g3[27:53]

#pLux-std-std glucosa
o25=od18211[16:27]
c25=cfp18211[16:27]
r25=rfp18211[16:27]
y25=yfp18211[16:27]

o26=od18212[17:30]
c26=cfp18212[17:30]
r26=rfp18212[17:30]
y26=yfp18212[17:30]

o27=od18213[18:30]
c27=cfp18213[18:30]
r27=rfp18213[18:30]
y27=yfp18213[18:30]

#plux-std-std glicerol
o28=od1821g1[24:47]
c28=cfp1821g1[24:47]
r28=rfp1821g1[24:47]
y28=yfp1821g1[24:47]

o29=od1821g2[29:54]
c29=cfp1821g2[29:54]
r29=rfp1821g2[29:54]
y29=yfp1821g2[29:54]

o30=od1821g3[26:50]
c30=cfp1821g3[26:50]
r30=rfp1821g3[26:50]
y30=yfp1821g3[26:50]

#106-std-std glucosa
o31=od12211[17:27]
c31=cfp12211[17:27]
r31=rfp12211[17:27]
y31=yfp12211[17:27]

o32=od12212[15:27]
```

```
c32=cfp12212[15:27]
r32=rfp12212[15:27]
y32=yfp12212[15:27]

o33=od12213[18:29]
c33=cfp12213[18:29]
r33=rfp12213[18:29]
y33=yfp12213[18:29]

#106-std-std glicerol
o34=od1221g1[28:58]
c34=cfp1221g1[28:58]
r34=rfp1221g1[28:58]
y34=yfp1221g1[28:58]

o35=od1221g2[23:50]
c35=cfp1221g2[23:50]
r35=rfp1221g2[23:50]
y35=yfp1221g2[23:50]

o36=od1221g3[29:52]
c36=cfp1221g3[29:52]
r36=rfp1221g3[29:52]
y36=yfp1221g3[29:52]
```

In [60]:
```
#regresion lineal de replicas
#Controles glucosa 15
slope, intercept, r_value, p_value,std_err=stats.linregress(o1,c1)
slopec1=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o1,r1)
sloper1=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o1,y1)
slopey1=slope

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

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

```
#Controles glicerol 15
slope, intercept, r_value, p_value,std_err=stats.linregress(o4,c4)
slopec4=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o4,r4)
sloper4=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o4,y4)
slopey4=slope

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

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

#controles glucosa 18
slope, intercept, r_value, p_value,std_err=stats.linregress(o7,c7)
slopec7=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o7,r7)
sloper7=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o7,y7)
slopey7=slope

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

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

#controles glicerol 18
slope, intercept, r_value, p_value,std_err=stats.linregress(o10,c10)
slopec10=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o10,r10)
```

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

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

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

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

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

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

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

```python
slope, intercept, r_value, p_value,std_err=stats.linregress(o17,c17)
slopec17=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o17,r17)
sloper17=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o17,y17)
slopey17=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o18,c18)
slopec18=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o18,r18)
sloper18=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o18,y18)
slopey18=slope

#ptet-std-std glucosa
slope, intercept, r_value, p_value,std_err=stats.linregress(o19,c19)
slopec19=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o19,r19)
sloper19=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o19,y19)
slopey19=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o20,c20)
slopec20=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o20,r20)
sloper20=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o20,y20)
slopey20=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o21,c21)
slopec21=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o21,r21)
sloper21=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o21,y21)
slopey21=slope

#ptet-std-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
```

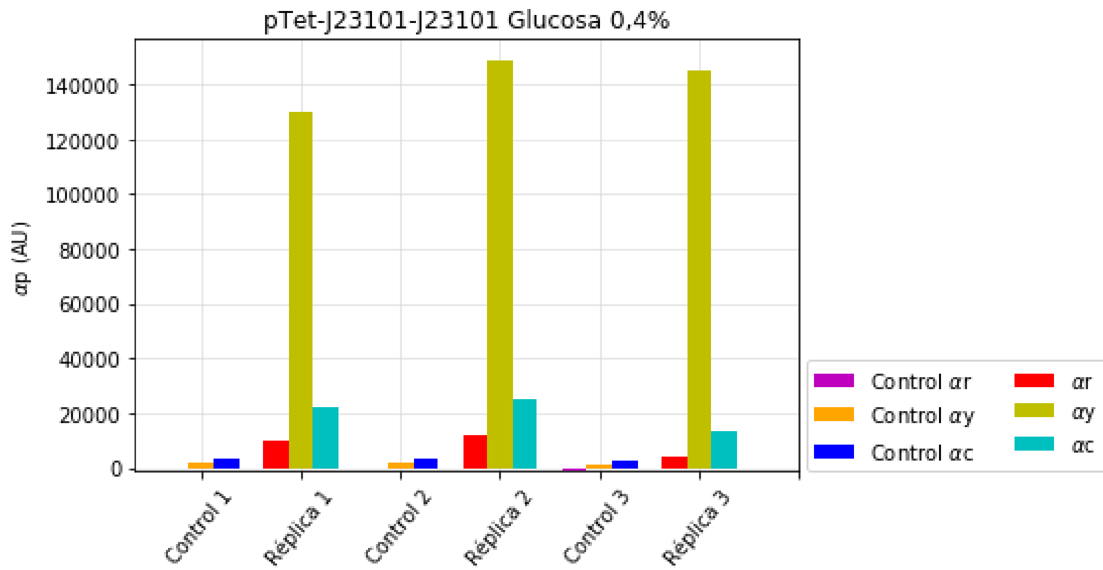
```
slope, intercept, r_value, p_value,std_err=stats.linregress(o23,y23)
slopey23=slope

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-std-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-std-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

#106-std-std glucosa
slope, intercept, r_value, p_value,std_err=stats.linregress(o31,c31)
slopec31=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o31,r31)
sloper31=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o31,y31)
slopey31=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o32,c32)
slopec32=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o32,r32)
sloper32=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o32,y32)
slopey32=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o33,c33)
slopec33=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o33,r33)
sloper33=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o33,y33)
slopey33=slope

#106-std-std glicerol
slope, intercept, r_value, p_value,std_err=stats.linregress(o34,c34)
slopec34=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o34,r34)
sloper34=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o34,y34)
slopey34=slope

slope, intercept, r_value, p_value,std_err=stats.linregress(o35,c35)
slopec35=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o35,r35)
sloper35=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o35,y35)
slopey35=slope

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

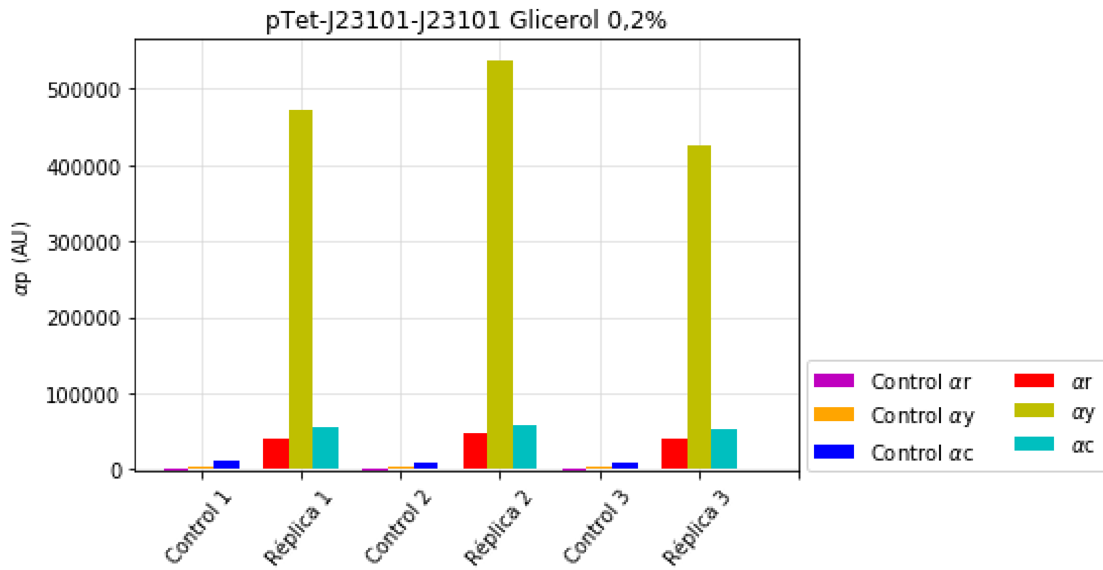
```
slopey36=slope
```

In [61]:
```
pendientescc=[slopec1,slopec2,slopec3,slopec4,slopec5,slopec6,slopec7,slopec8,slopec9,s
pendientesc=[slopec19,slopec20,slopec21,slopec22,slopec23,slopec24,slopec25,slopec26,sl
pendientescr=[sloper1,sloper2,sloper3,sloper4,sloper5,sloper6,sloper7,sloper8,sloper9,s
pendientesr=[sloper19,sloper20,sloper21,sloper22,sloper23,sloper24,sloper25,sloper26,sl
pendientescy=[slopey1,slopey2,slopey3,slopey4,slopey5,slopey6,slopey7,slopey8,slopey9,s
pendientesy=[slopey19,slopey20,slopey21,slopey22,slopey23,slopey24,slopey25,slopey26,sl
#Grafico pendientes ptet-std-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('pTet-J23101-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+' '+ r'$\al
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c
plt.bar(X[1]-0.25,pendientesr[0],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[0],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[0],color='c',width=0.25,label= r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[1],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[1],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[1],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[2],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[2],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[2],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[61]:
```
<matplotlib.legend.Legend at 0x2035cb529e8>
```



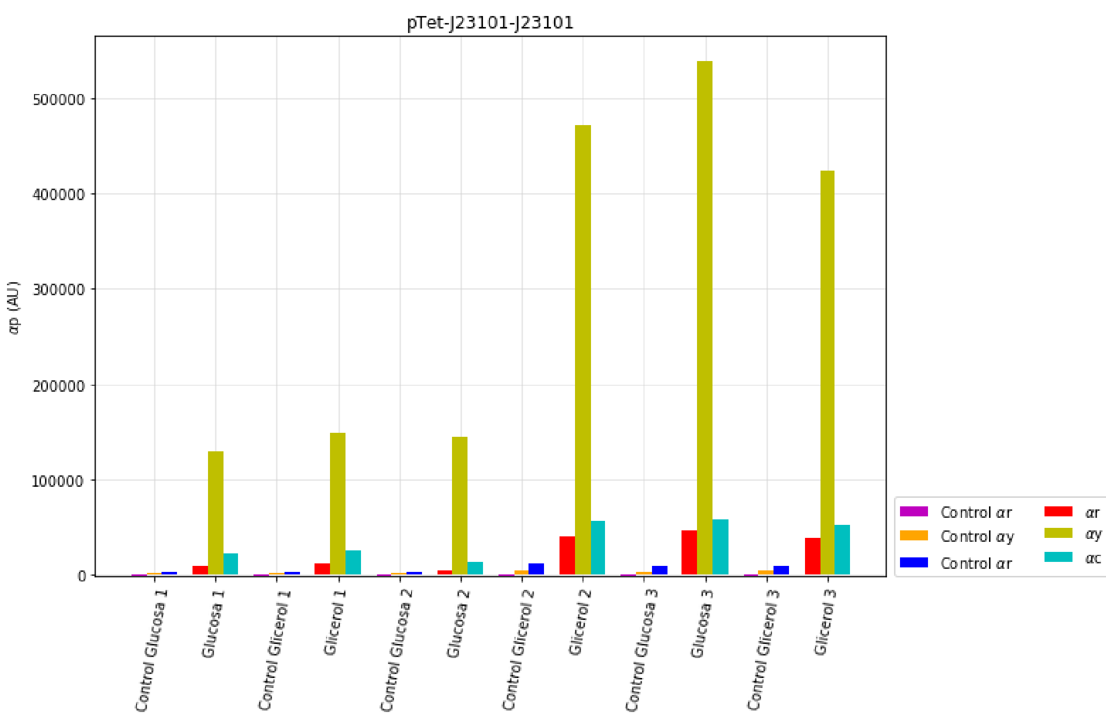

```
In [62]: #Grafico pendientes ptet-std-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('pTet-J23101-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientescr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
         plt.bar(X[0]+0.00,pendientescy[3],color='orange',width=0.25,label='Control'+' '+ r'$\al
         plt.bar(X[0]+0.25,pendientescc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c
         plt.bar(X[1]-0.25,pendientesr[3],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[3],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[3],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientescr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientescy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientescc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[4],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[4],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[4],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientescr[5],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientescy[5],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientescc[5],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[5],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[5],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[5],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]

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

Out[62]: <matplotlib.legend.Legend at 0x203631eb358>
```



In [63]:
```
#Grafico pendientes ptet-std-std
X = np.arange(12)
plt.figure(figsize=(10,7))
plt.title('pTet-J23101-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+' '+ r'$\al
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$r
plt.bar(X[1]-0.25,pendientesr[0],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[0],color='y',width=0.25,label= r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[0],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[1],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[1],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[1],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[2],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[2],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[2],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[3],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientescy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientescc[3],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[3],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[3],color='y',width=0.25,zorder=3)
```

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

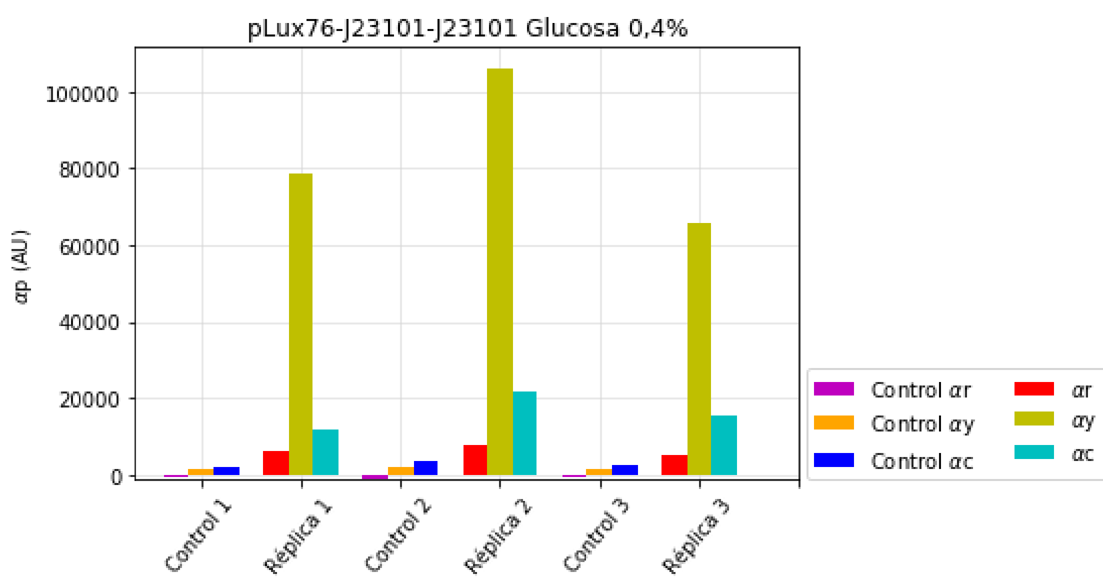
Out[63]: <matplotlib.legend.Legend at 0x20362c21780>




```
In [64]: #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,pendientescr[6],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
plt.bar(X[0]+0.00,pendientescy[6],color='orange',width=0.25,label='Control'+' '+ r'$\al
plt.bar(X[0]+0.25,pendientescc[6],color='b',width=0.25,label='Control'+' '+ r'$\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,pendientescr[7],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[7],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[7],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[8],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[8],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[8],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[8],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

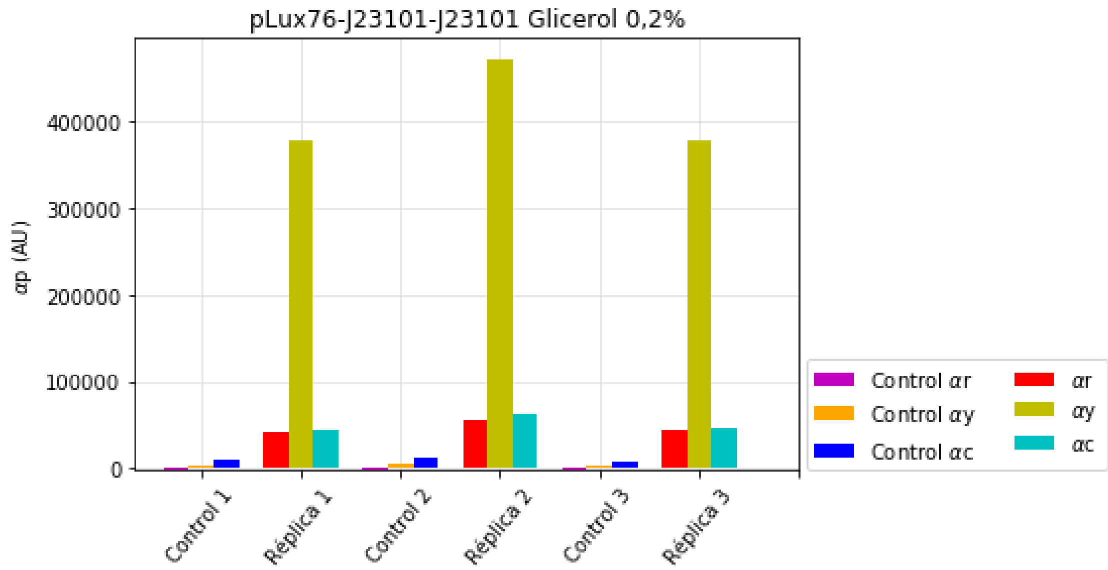
Out[64]: <matplotlib.legend.Legend at 0x203636f92b0>




```
In [65]: #Grafico pendientes plux-std-std Glicerol
X = np.arange(7)
plt.figure()
plt.title('pLux76-J23101-J23101 Glicerol 0,2%')
```

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

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

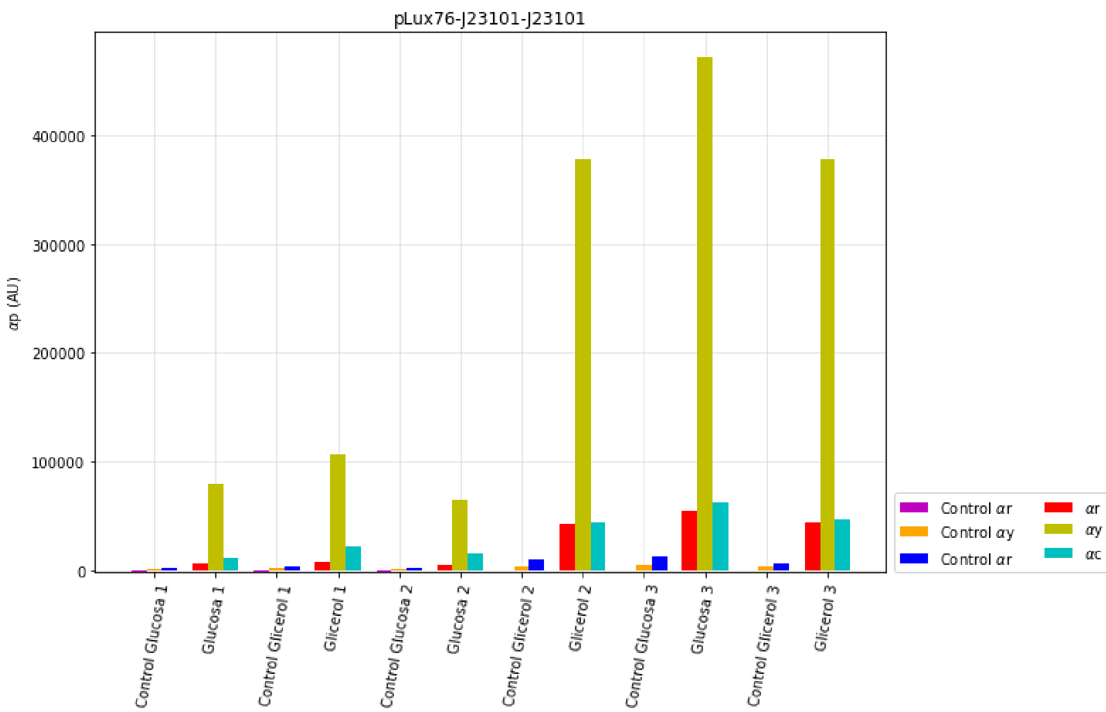
Out[65]: <matplotlib.legend.Legend at 0x2036502fb00>




```python
In [66]: #Grafico pendientes plux-std-std
         X = np.arange(12)
```

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

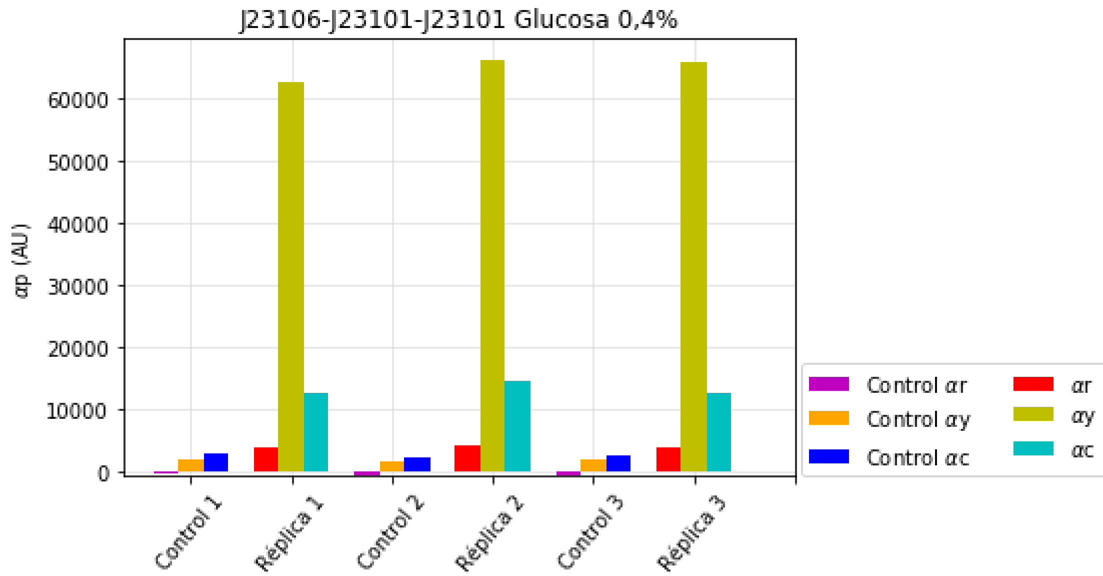
```
Out[66]: <matplotlib.legend.Legend at 0x203651d9be0>
```




```
In [67]: #Grafico pendientes 106-std-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-J23101-J23101 Glucosa 0,4%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientescr[12],color='m',width=0.25,label='Control'+' '+ r'$\alpha$
         plt.bar(X[0]+0.00,pendientescy[12],color='orange',width=0.25,label='Control'+' '+ r'$\a
         plt.bar(X[0]+0.25,pendientescc[12],color='b',width=0.25,label='Control'+' '+ r'$\alpha$
         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,pendientescr[13],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientescy[13],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientescc[13],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientescr[14],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientescy[14],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientescc[14],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
```

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

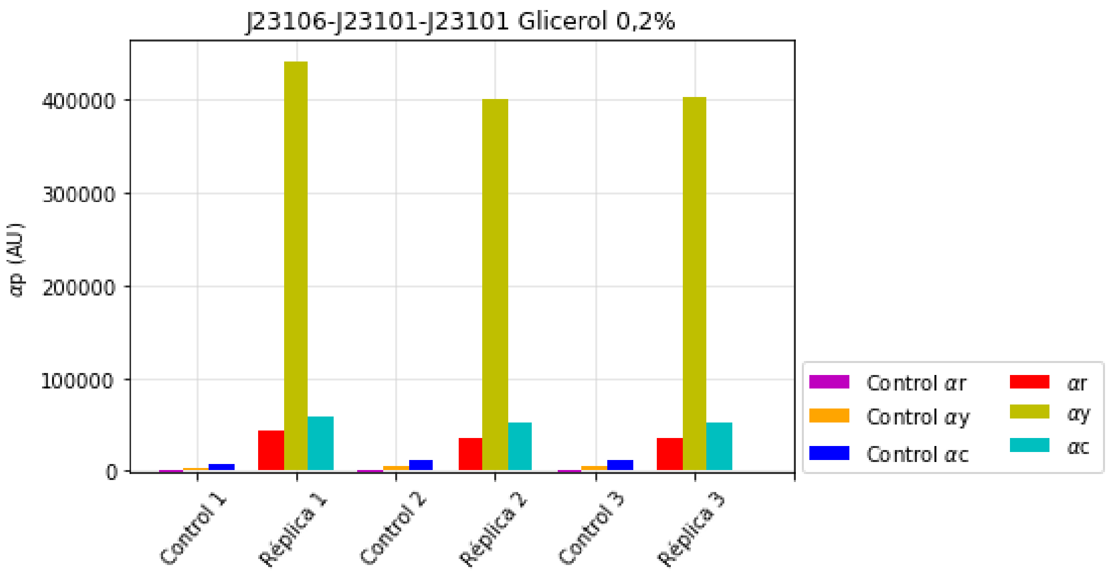
```
Out[67]: <matplotlib.legend.Legend at 0x2036562c4a8>
```




```
In [68]: #Grafico pendientes 106-std-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-J23101-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientescr[15],color='m',width=0.25,label='Control'+' '+ r'$\alpha$
         plt.bar(X[0]+0.00,pendientescy[15],color='orange',width=0.25,label='Control'+' '+ r'$\a
         plt.bar(X[0]+0.25,pendientescc[15],color='b',width=0.25,label='Control'+' '+ r'$\alpha$
         plt.bar(X[1]-0.25,pendientesr[15],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[15],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[15],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientescr[16],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientescy[16],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientescc[16],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientescr[17],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientescy[17],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientescc[17],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
```

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

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

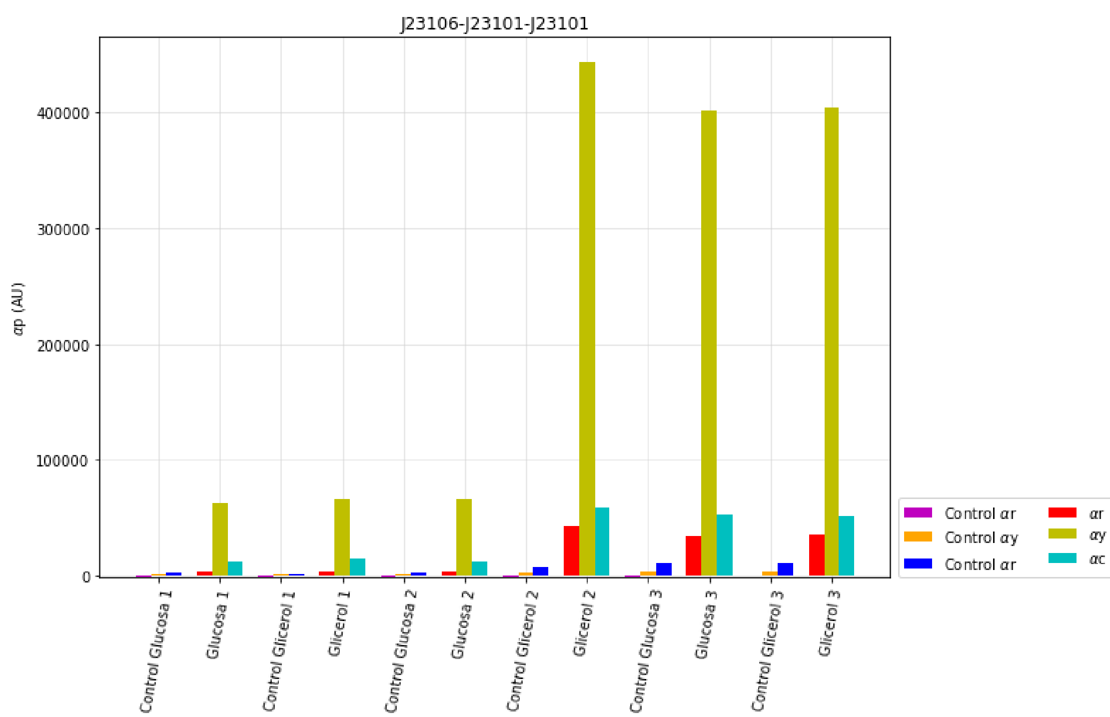
Out[68]: <matplotlib.legend.Legend at 0x20362f77748>



In [69]:
```
#Grafico pendientes plux-std-std
X = np.arange(12)
plt.figure(figsize=(10,7))
plt.title('J23106-J23101-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[12],color='m',width=0.25,label='Control'+' '+ r'$\alpha$
plt.bar(X[0]+0.00,pendientescy[12],color='orange',width=0.25,label='Control'+' '+ r'$\a
plt.bar(X[0]+0.25,pendientescc[12],color='b',width=0.25,label='Control'+' '+ r'$\alpha$
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,pendientescr[13],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[13],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[13],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[14],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[14],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[14],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
```

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

```
Out[69]: <matplotlib.legend.Legend at 0x20364cd34a8>
```

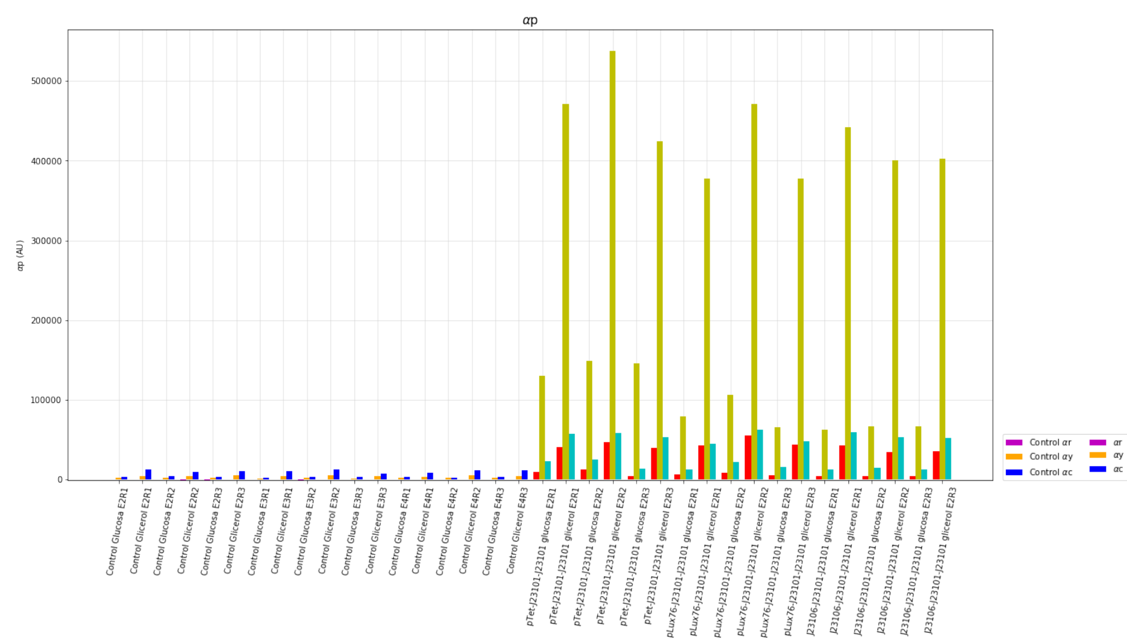


In [70]:
```
#Grafico pendientes todo
X = np.arange(36)
plt.figure(figsize=(20,10))
plt.title(r'$\alpha$p',fontsize=15.0)
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientescr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r
plt.bar(X[0]+0.00,pendientescy[0],color='orange',width=0.25,label='Control'+' '+ r'$\al
plt.bar(X[0]+0.25,pendientescc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c
plt.bar(X[1]-0.25,pendientescr[3],color='m',width=0.25,zorder=3)
plt.bar(X[1]+0.00,pendientescy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[1]+0.25,pendientescc[3],color='b',width=0.25,zorder=3)
plt.bar(X[2]-0.25,pendientescr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientescy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientescc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientescr[4],color='m',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientescy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientescc[4],color='b',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientescr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientescy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientescc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientescr[5],color='m',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientescy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientescc[5],color='b',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientescr[6],color='m',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[6]+0.00,pendientescy[6],color='orange',width=0.25,label=r'$\alpha$y',zorder=3
plt.bar(X[6]+0.25,pendientescc[6],color='blue',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[7]-0.25,pendientescr[9],color='m',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientescy[9],color='orange',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientescc[9],color='b',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientescr[7],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientescy[7],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientescc[7],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientescr[10],color='m',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientescy[10],color='orange',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientescc[10],color='b',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientescr[8],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientescy[8],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientescc[8],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientescr[11],color='m',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientescy[11],color='orange',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientescc[11],color='b',width=0.25,zorder=3)
plt.bar(X[12]-0.25,pendientescr[12],color='m',width=0.25,zorder=3)
plt.bar(X[12]+0.00,pendientescy[12],color='orange',width=0.25,zorder=3)
plt.bar(X[12]+0.25,pendientescc[12],color='b',width=0.25,zorder=3)
plt.bar(X[13]-0.25,pendientescr[15],color='m',width=0.25,zorder=3)
plt.bar(X[13]+0.00,pendientescy[15],color='orange',width=0.25,zorder=3)
plt.bar(X[13]+0.25,pendientescc[15],color='b',width=0.25,zorder=3)
plt.bar(X[14]-0.25,pendientescr[13],color='m',width=0.25,zorder=3)
```

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

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

Out[70]: <matplotlib.legend.Legend at 0x203652a6a58>

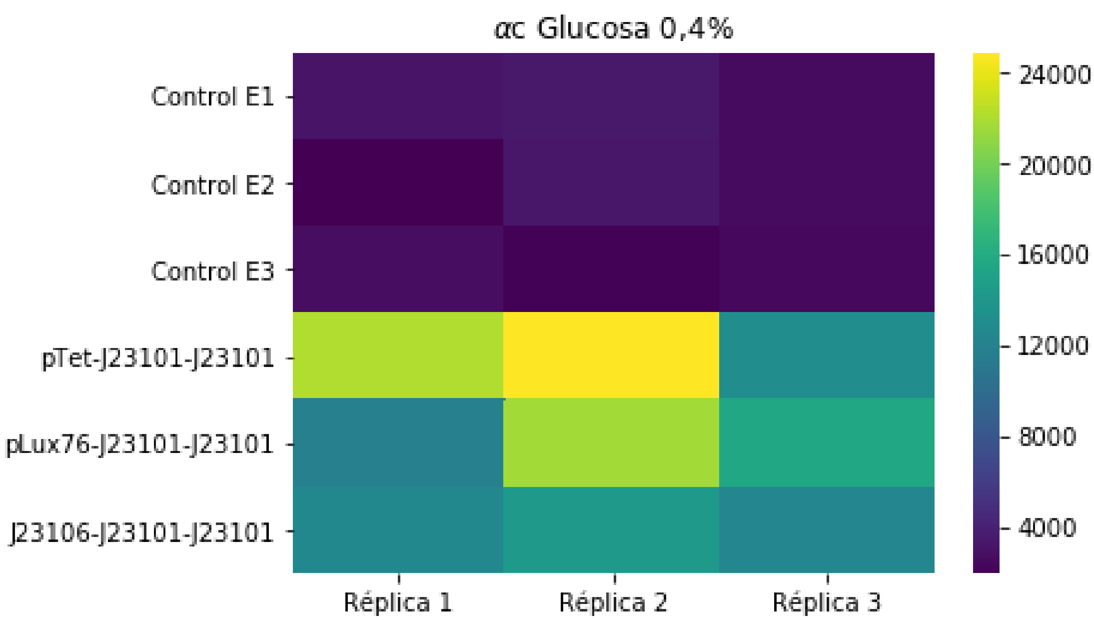


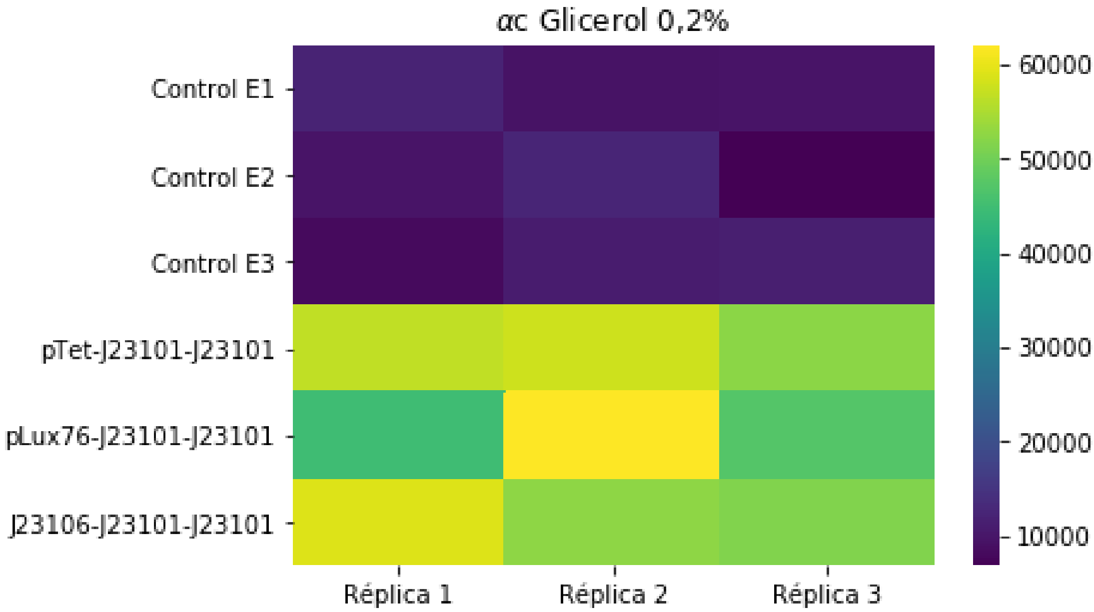

```
In [71]: cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15],
         cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec1

         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
```

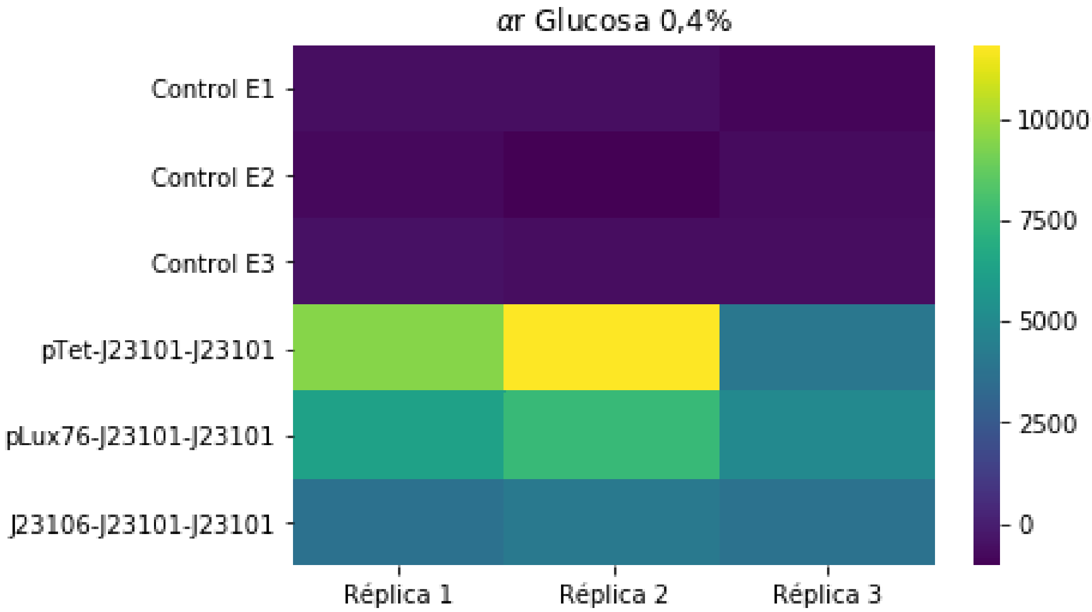
In [72]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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

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

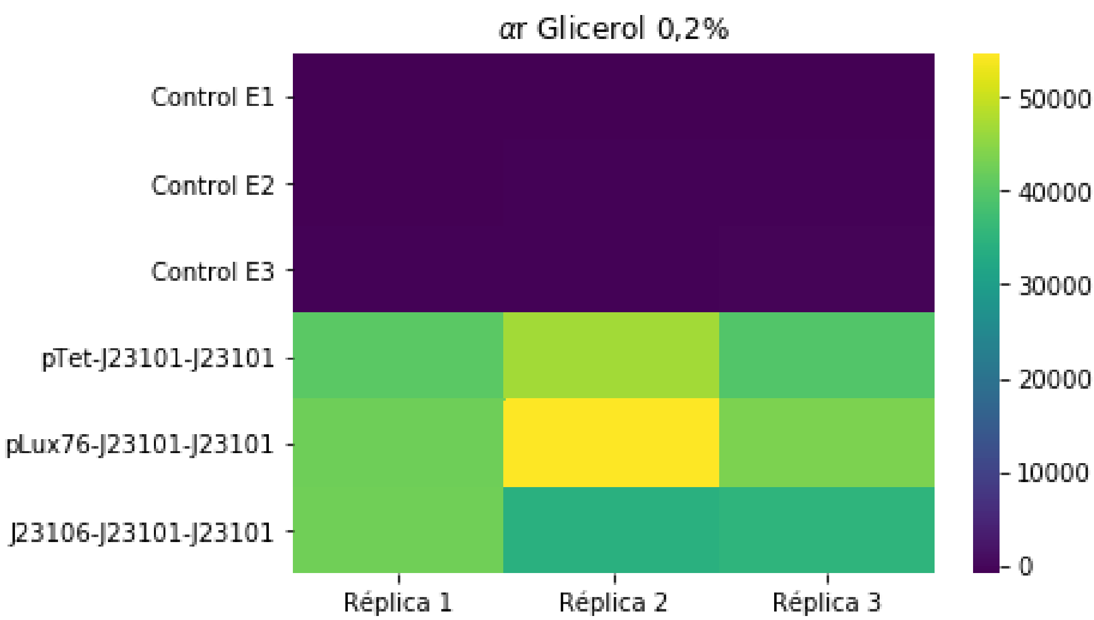
Out[72]: <matplotlib.axes._subplots.AxesSubplot at 0x20366583f28>






```
In [73]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231


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

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

Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x2036529f940>
```

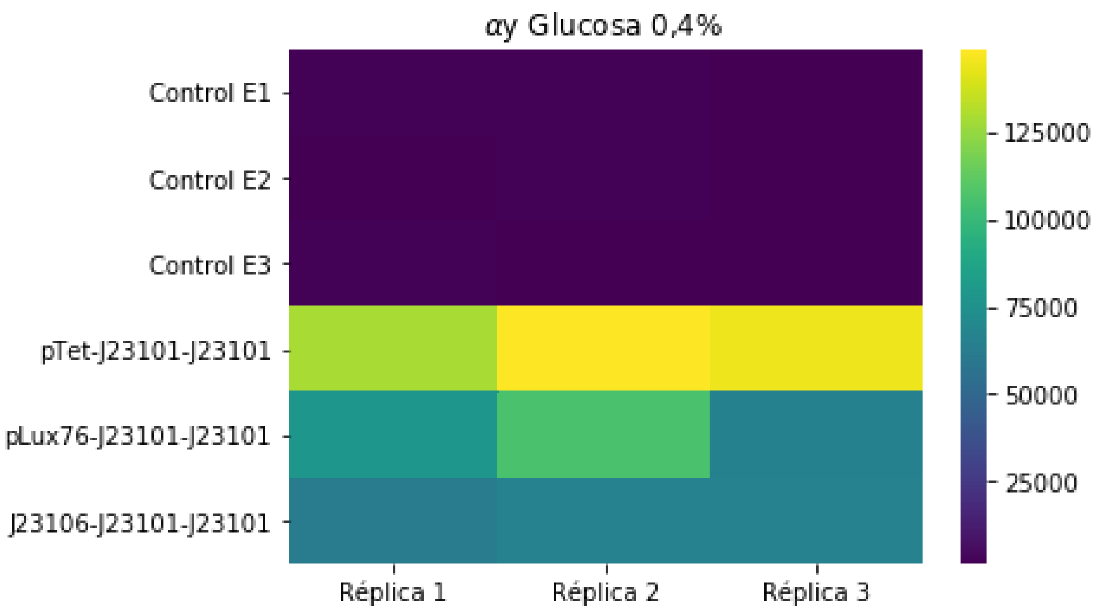


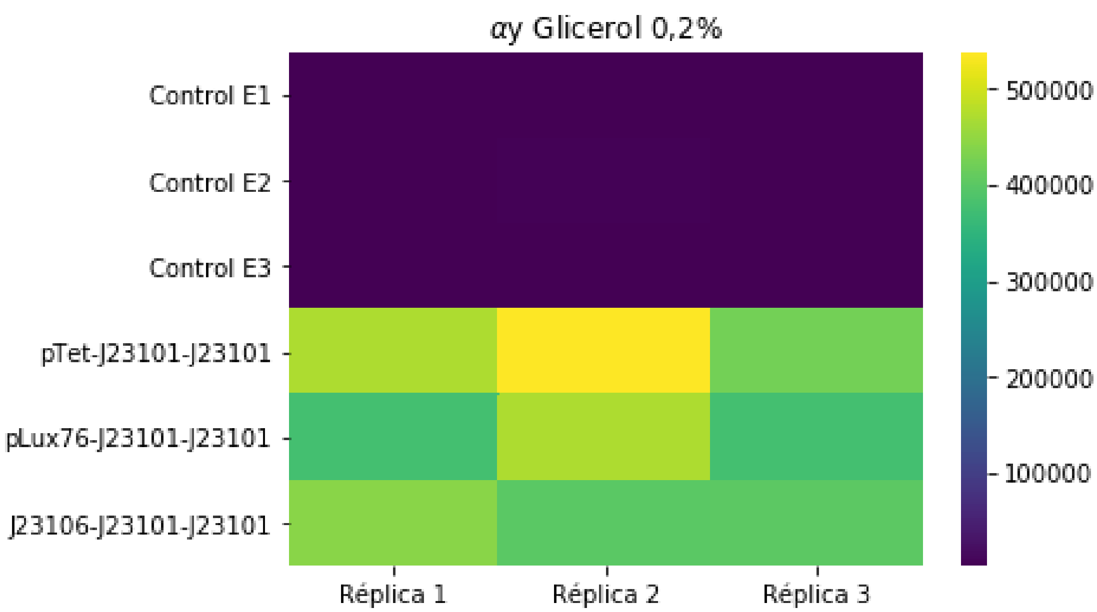



```
In [74]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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

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

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

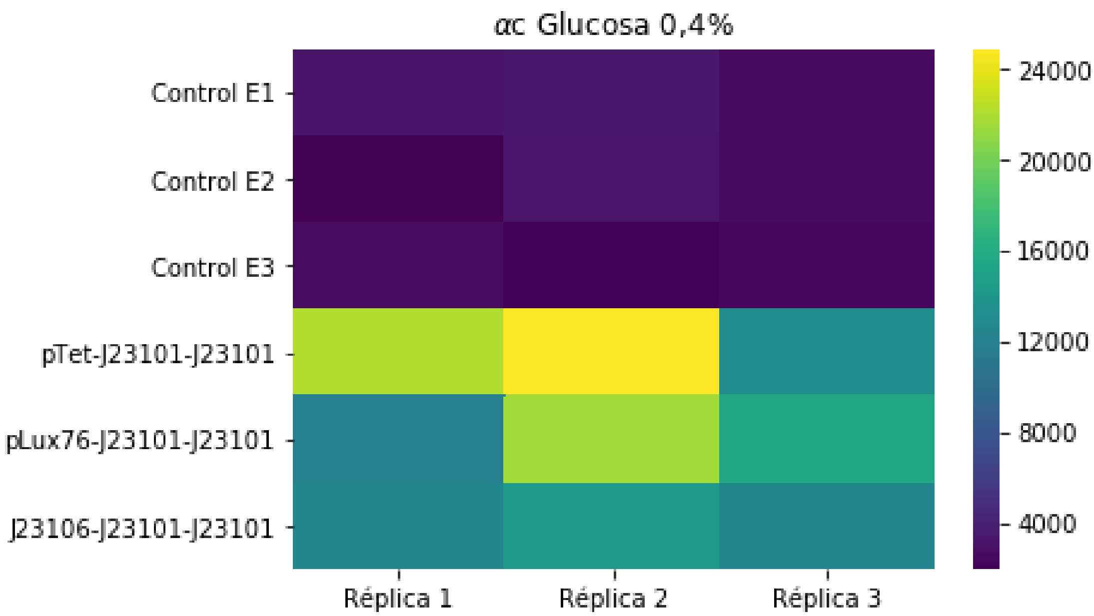
Out[74]: <matplotlib.axes._subplots.AxesSubplot at 0x20364d19550>

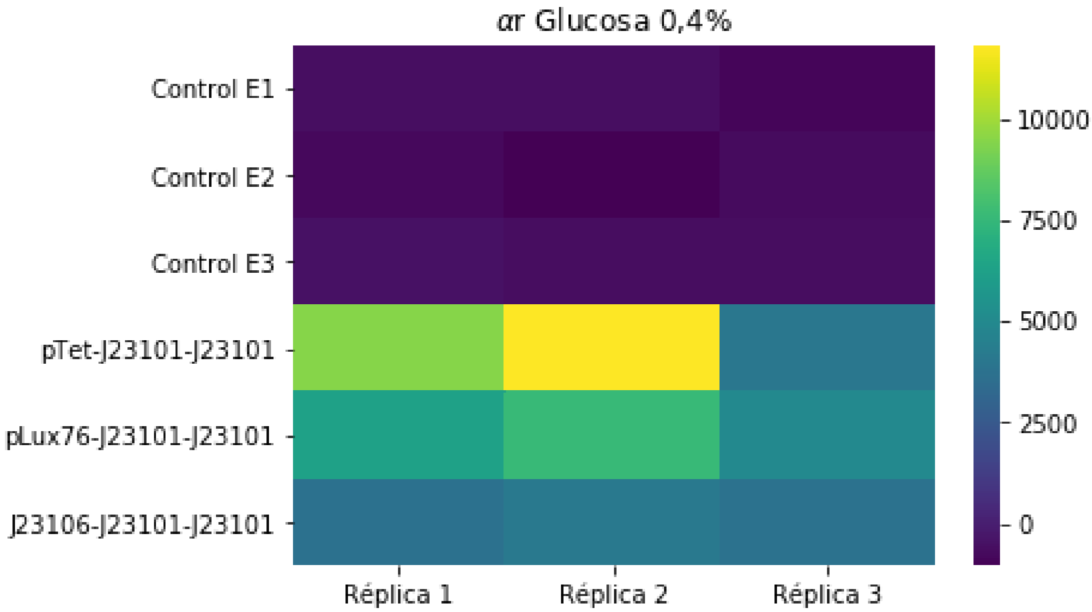


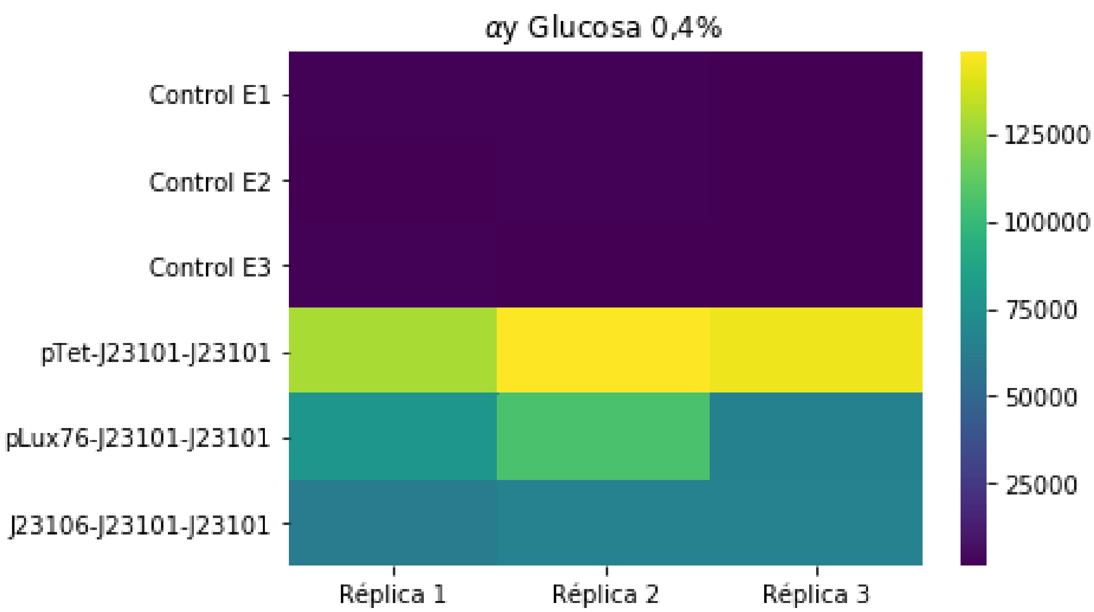


In [75]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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[75]: <matplotlib.axes._subplots.AxesSubplot at 0x20365c847f0>



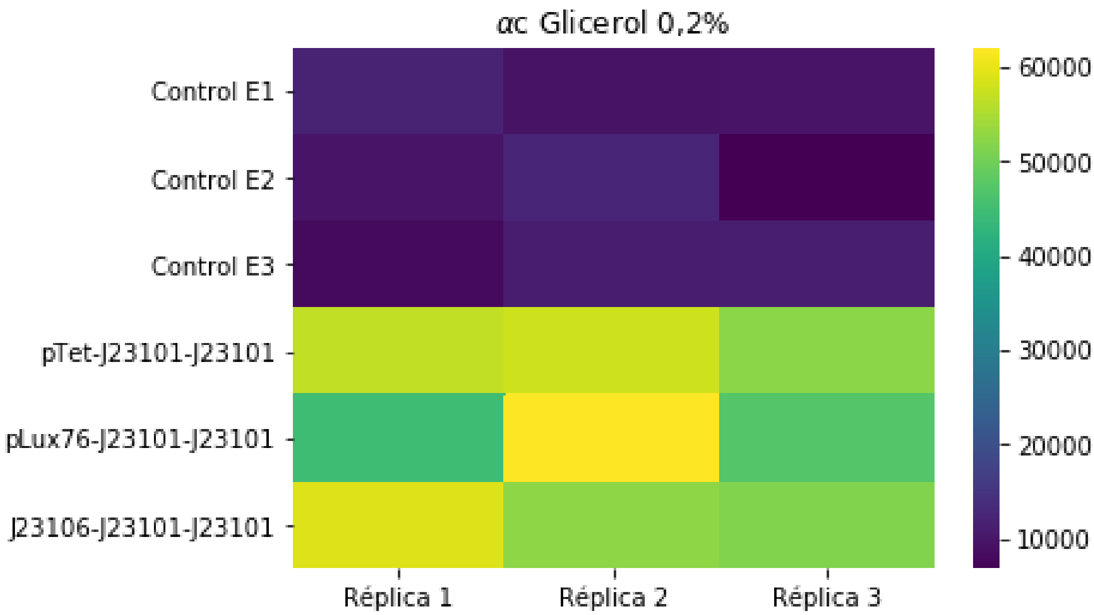


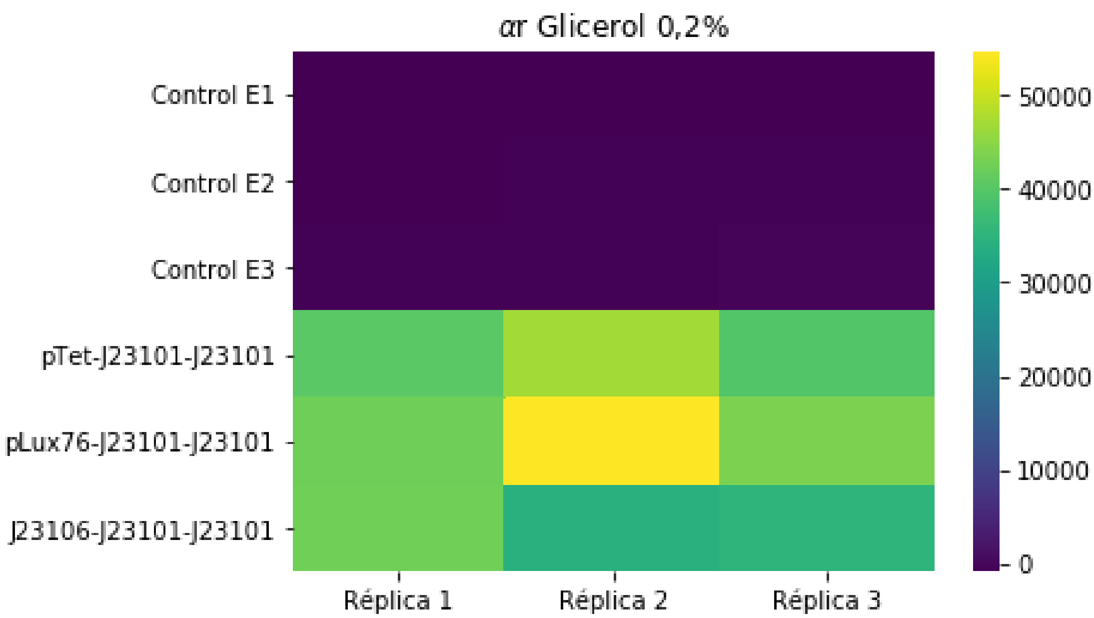


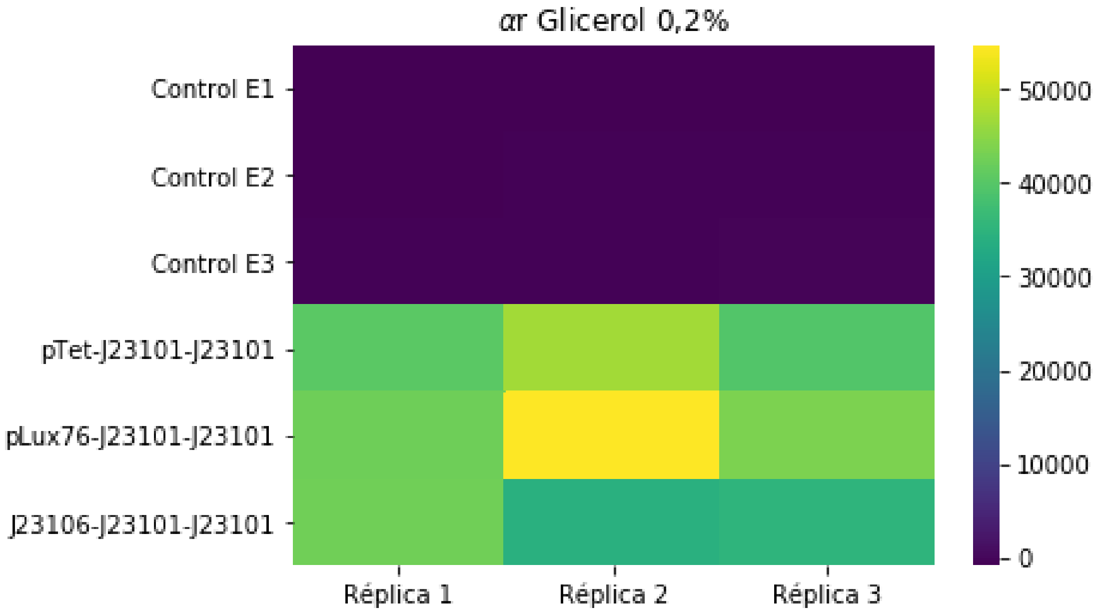
In [76]:
```
xlabel=['Réplica 1','Réplica 2','Réplica 3']
ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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[76]: <matplotlib.axes._subplots.AxesSubplot at 0x20365722d30>
```



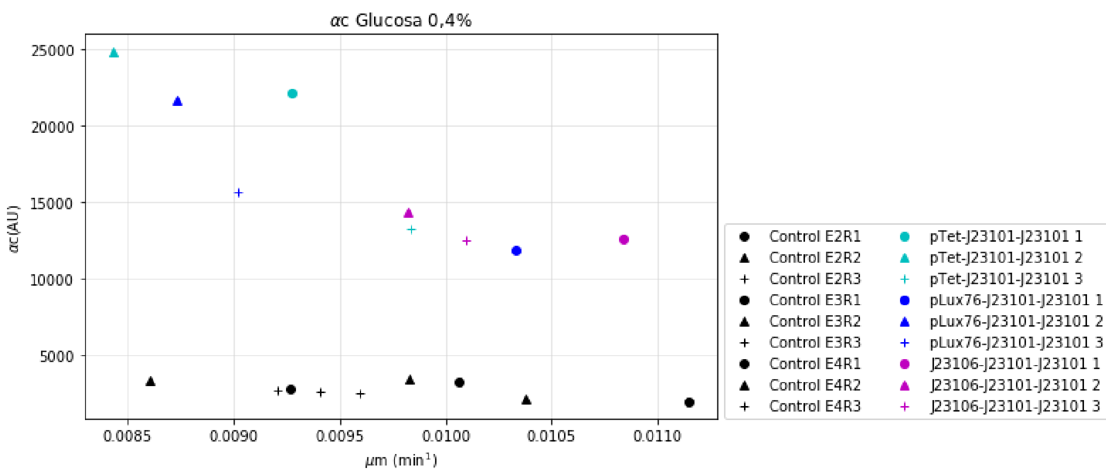


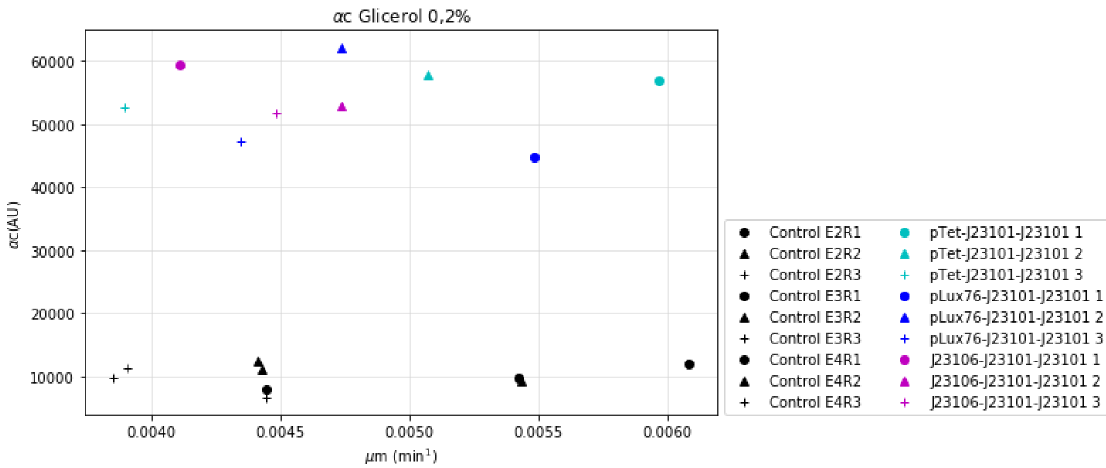



```
In [155]: #grafico de ac versus Um
          plt.figure(figsize=(8,5))
          plt.title(r'$\alpha$c Glucosa 0,4%')
          plt.xlabel(r'$\mu$m (min$^1$)')
          plt.ylabel(r'$\alpha$c(AU)')
          plt.plot(um1,slopec1,'ko',label='Control E2R1')
          plt.plot(um2,slopec2,'k^',label='Control E2R2')
          plt.plot(um3,slopec3,'k+',label='Control E2R3')
          plt.plot(um7,slopec7,'ko',label='Control E3R1')
          plt.plot(um8,slopec8,'k^',label='Control E3R2')
          plt.plot(um9,slopec9,'k+',label='Control E3R3')
          plt.plot(um13,slopec13,'ko',label='Control E4R1')
          plt.plot(um14,slopec14,'k^',label='Control E4R2')
          plt.plot(um15,slopec15,'k+',label='Control E4R3')
          plt.plot(um19,slopec19,'co',label='pTet-J23101-J23101 1')
          plt.plot(um20,slopec20,'c^',label='pTet-J23101-J23101 2')
          plt.plot(um21,slopec21,'c+',label='pTet-J23101-J23101 3')
          plt.plot(um25,slopec25,'bo',label='pLux76-J23101-J23101 1')
          plt.plot(um26,slopec26,'b^',label='pLux76-J23101-J23101 2')
          plt.plot(um27,slopec27,'b+',label='pLux76-J23101-J23101 3')
          plt.plot(um31,slopec31,'mo',label='J23106-J23101-J23101 1')
          plt.plot(um32,slopec32,'m^',label='J23106-J23101-J23101 2')
          plt.plot(um33,slopec33,'m+',label='J23106-J23101-J23101 3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
```

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

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

Out[155]: <matplotlib.legend.Legend at 0x2036e6999b0>

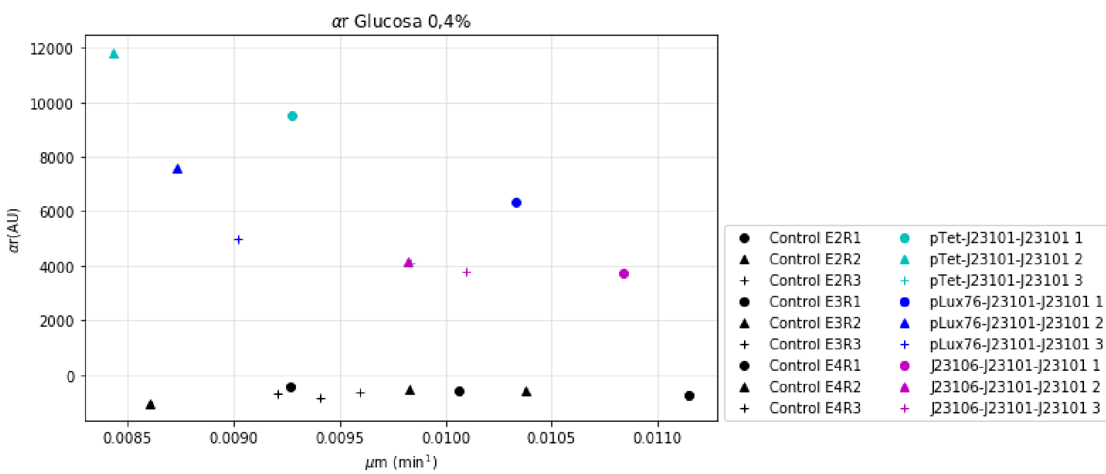


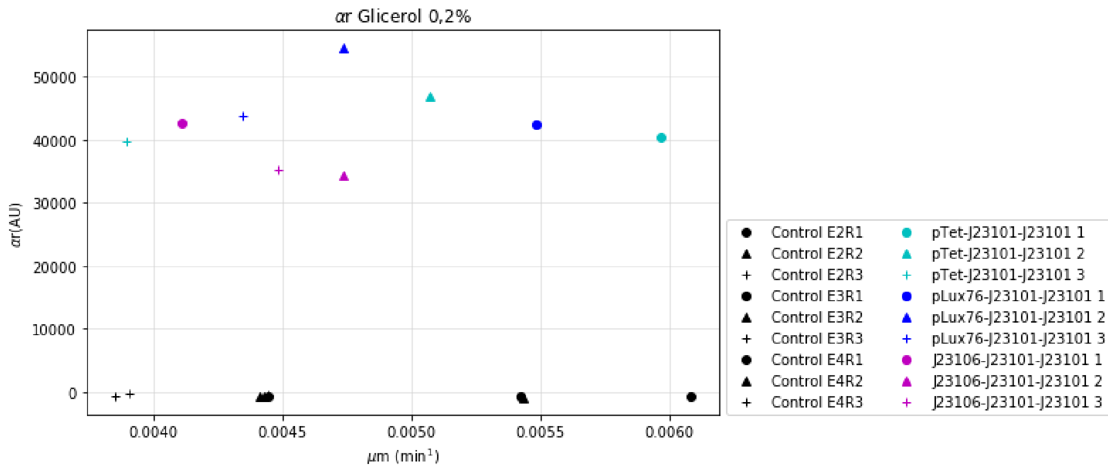


In [154]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'ko',label='Control E2R1')
plt.plot(um2,sloper2,'k^',label='Control E2R2')
plt.plot(um3,sloper3,'k+',label='Control E2R3')
plt.plot(um7,sloper7,'ko',label='Control E3R1')
plt.plot(um8,sloper8,'k^',label='Control E3R2')
plt.plot(um9,sloper9,'k+',label='Control E3R3')
plt.plot(um13,sloper13,'ko',label='Control E4R1')
plt.plot(um14,sloper14,'k^',label='Control E4R2')
plt.plot(um15,sloper15,'k+',label='Control E4R3')
plt.plot(um19,sloper19,'co',label='pTet-J23101-J23101 1')
plt.plot(um20,sloper20,'c^',label='pTet-J23101-J23101 2')
plt.plot(um21,sloper21,'c+',label='pTet-J23101-J23101 3')
plt.plot(um25,sloper25,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um26,sloper26,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um27,sloper27,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um31,sloper31,'mo',label='J23106-J23101-J23101 1')
plt.plot(um32,sloper32,'m^',label='J23106-J23101-J23101 2')
plt.plot(um33,sloper33,'m+',label='J23106-J23101-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

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

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

```
Out[154]: <matplotlib.legend.Legend at 0x20371b29978>
```

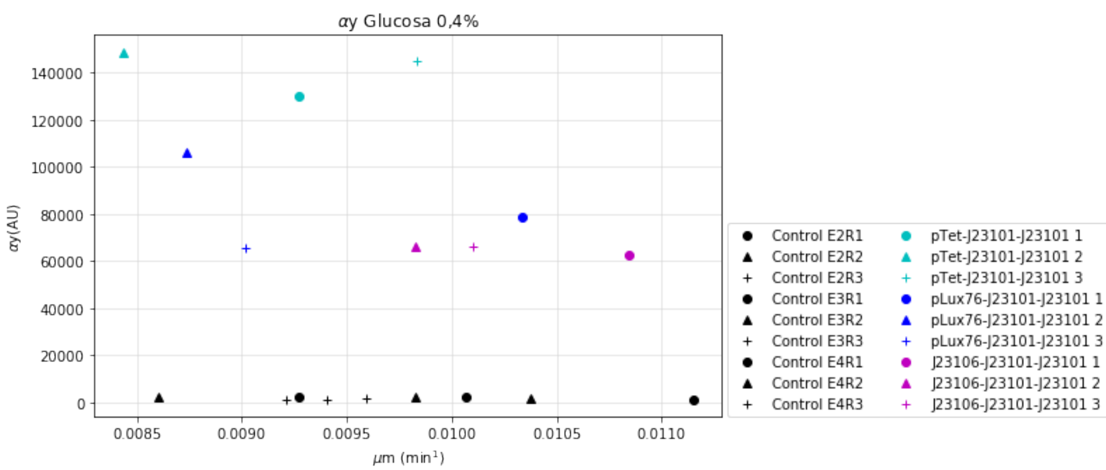


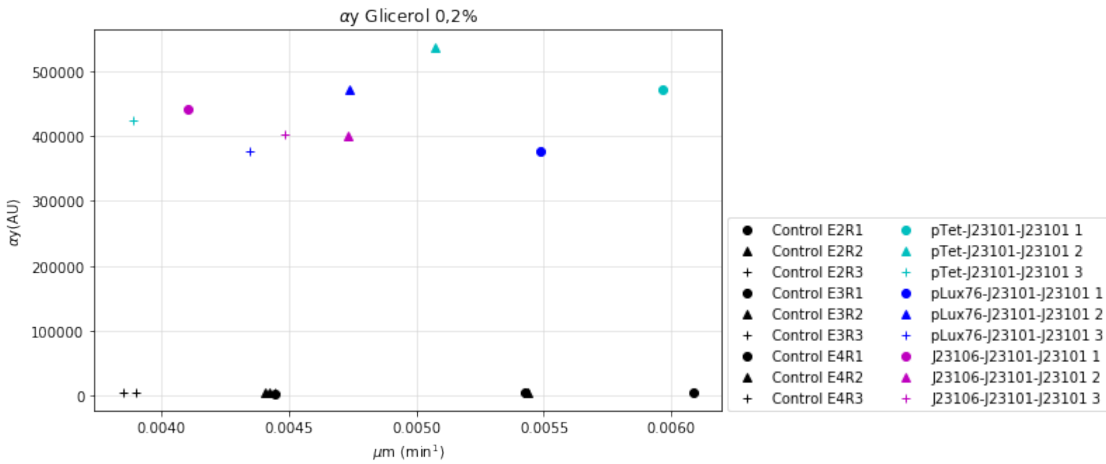


In [156]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'ko',label='Control E2R1')
plt.plot(um2,slopey2,'k^',label='Control E2R2')
plt.plot(um3,slopey3,'k+',label='Control E2R3')
plt.plot(um7,slopey7,'ko',label='Control E3R1')
plt.plot(um8,slopey8,'k^',label='Control E3R2')
plt.plot(um9,slopey9,'k+',label='Control E3R3')
plt.plot(um13,slopey13,'ko',label='Control E4R1')
plt.plot(um14,slopey14,'k^',label='Control E4R2')
plt.plot(um15,slopey15,'k+',label='Control E4R3')
plt.plot(um19,slopey19,'co',label='pTet-J23101-J23101 1')
plt.plot(um20,slopey20,'c^',label='pTet-J23101-J23101 2')
plt.plot(um21,slopey21,'c+',label='pTet-J23101-J23101 3')
plt.plot(um25,slopey25,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um26,slopey26,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um27,slopey27,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um31,slopey31,'mo',label='J23106-J23101-J23101 1')
plt.plot(um32,slopey32,'m^',label='J23106-J23101-J23101 2')
plt.plot(um33,slopey33,'m+',label='J23106-J23101-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

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

```
plt.plot(um4,slopey4,'ko',label='Control E2R1')
plt.plot(um5,slopey5,'k^',label='Control E2R2')
plt.plot(um6,slopey6,'k+',label='Control E2R3')
plt.plot(um10,slopey10,'ko',label='Control E3R1')
plt.plot(um11,slopey11,'k^',label='Control E3R2')
plt.plot(um12,slopey12,'k+',label='Control E3R3')
plt.plot(um16,slopey16,'ko',label='Control E4R1')
plt.plot(um17,slopey17,'k^',label='Control E4R2')
plt.plot(um18,slopey18,'k+',label='Control E4R3')
plt.plot(um22,slopey22,'co',label='pTet-J23101-J23101 1')
plt.plot(um23,slopey23,'c^',label='pTet-J23101-J23101 2')
plt.plot(um24,slopey24,'c+',label='pTet-J23101-J23101 3')
plt.plot(um28,slopey28,'bo',label='pLux76-J23101-J23101 1')
plt.plot(um29,slopey29,'b^',label='pLux76-J23101-J23101 2')
plt.plot(um30,slopey30,'b+',label='pLux76-J23101-J23101 3')
plt.plot(um34,slopey34,'mo',label='J23106-J23101-J23101 1')
plt.plot(um35,slopey35,'m^',label='J23106-J23101-J23101 2')
plt.plot(um36,slopey36,'m+',label='J23106-J23101-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[156]: <matplotlib.legend.Legend at 0x2037234c828>

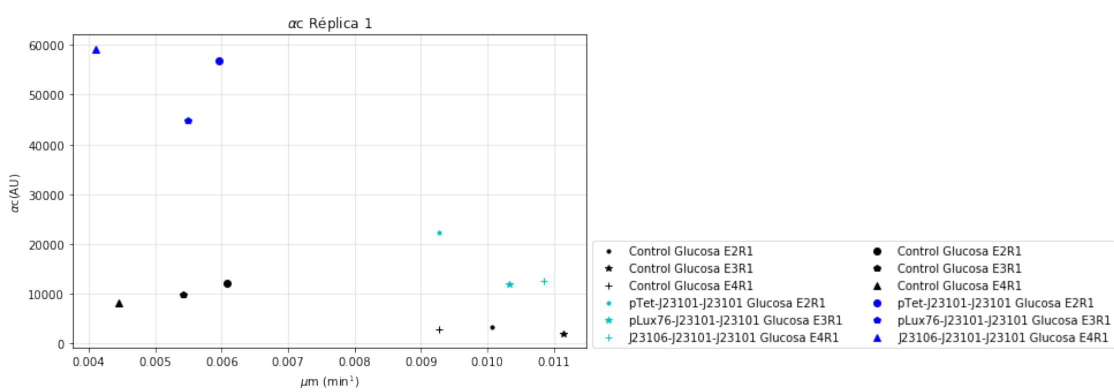


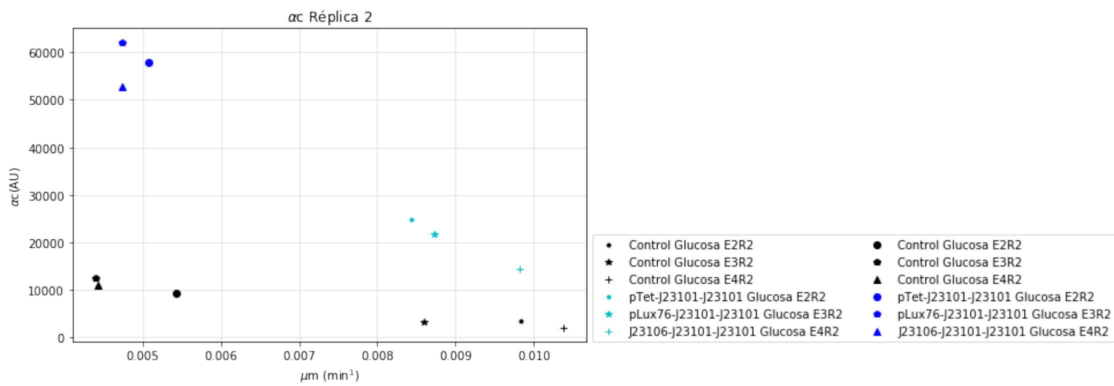


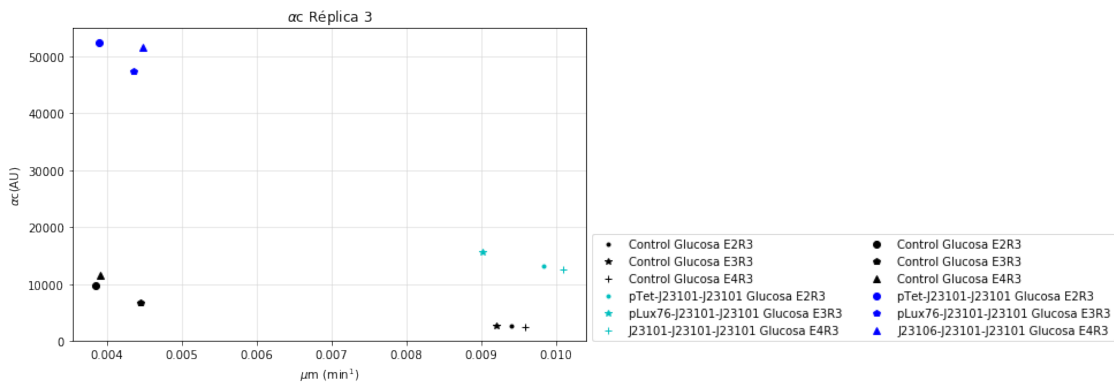
In [80]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um1,slopec1,'k.',label='Control Glucosa E2R1')
plt.plot(um7,slopec7,'k*',label='Control Glucosa E3R1')
plt.plot(um13,slopec13,'k+',label='Control Glucosa E4R1')
plt.plot(um19,slopec19,'c.',label='pTet-J23101-J23101 Glucosa E2R1')
plt.plot(um25,slopec25,'c*',label='pLux76-J23101-J23101 Glucosa E3R1')
plt.plot(um31,slopec31,'c+',label='J23106-J23101-J23101 Glucosa E4R1')
plt.plot(um4,slopec4,'ko',label='Control Glucosa E2R1')
plt.plot(um10,slopec10,'kp',label='Control Glucosa E3R1')
plt.plot(um16,slopec16,'k^',label='Control Glucosa E4R1')
plt.plot(um22,slopec22,'bo',label='pTet-J23101-J23101 Glucosa E2R1')
plt.plot(um28,slopec28,'bp',label='pLux76-J23101-J23101 Glucosa E3R1')
plt.plot(um34,slopec34,'b^',label='J23106-J23101-J23101 Glucosa E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

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

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

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

Out[80]: <matplotlib.legend.Legend at 0x20368130f28>



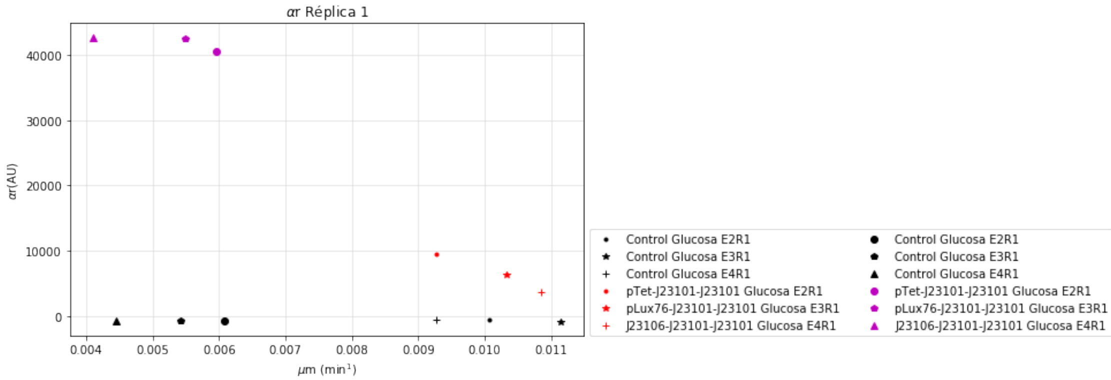


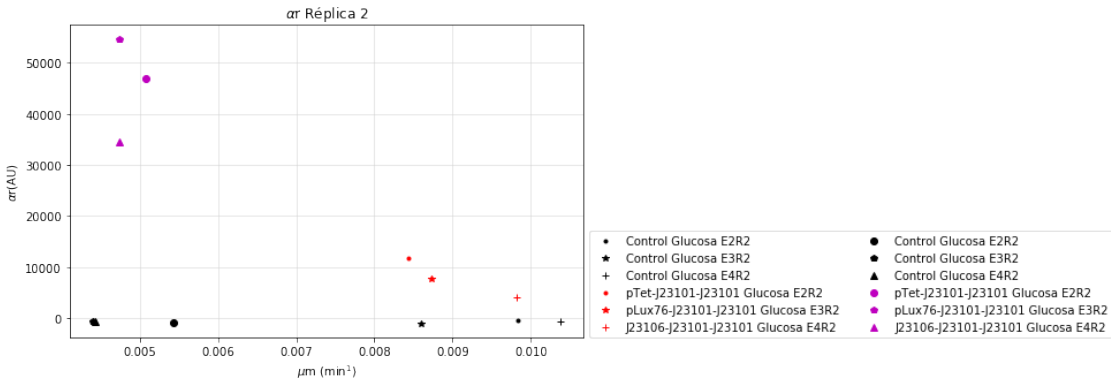


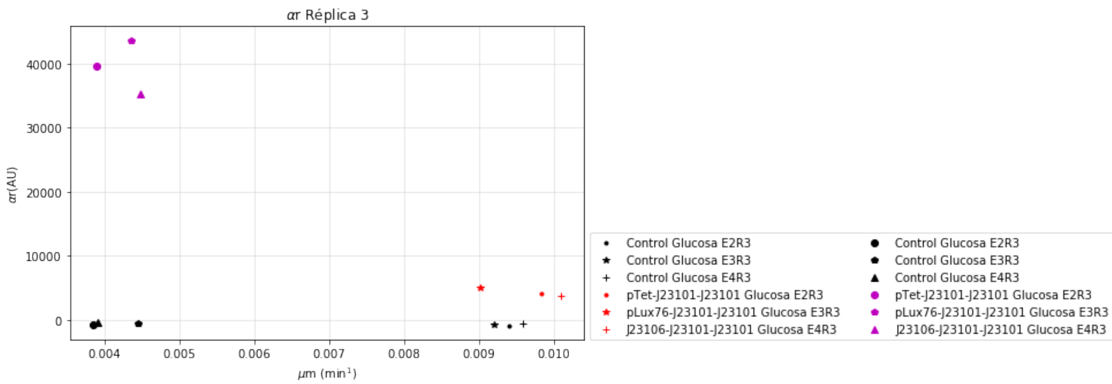

```
In [81]: #grafico de ar versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Réplica 1')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um1,sloper1,'k.',label='Control Glucosa E2R1')
         plt.plot(um7,sloper7,'k*',label='Control Glucosa E3R1')
         plt.plot(um13,sloper13,'k+',label='Control Glucosa E4R1')
         plt.plot(um19,sloper19,'r.',label='pTet-J23101-J23101 Glucosa E2R1')
         plt.plot(um25,sloper25,'r*',label='pLux76-J23101-J23101 Glucosa E3R1')
         plt.plot(um31,sloper31,'r+',label='J23106-J23101-J23101 Glucosa E4R1')
         plt.plot(um4,sloper4,'ko',label='Control Glucosa E2R1')
         plt.plot(um10,sloper10,'kp',label='Control Glucosa E3R1')
         plt.plot(um16,sloper16,'k^',label='Control Glucosa E4R1')
         plt.plot(um22,sloper22,'mo',label='pTet-J23101-J23101 Glucosa E2R1')
         plt.plot(um28,sloper28,'mp',label='pLux76-J23101-J23101 Glucosa E3R1')
         plt.plot(um34,sloper34,'m^',label='J23106-J23101-J23101 Glucosa E4R1')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)
```

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

#grafico de ar versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Réplica 3')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um3,sloper3,'k.',label='Control Glucosa E2R3')
plt.plot(um9,sloper9,'k*',label='Control Glucosa E3R3')
plt.plot(um15,sloper15,'k+',label='Control Glucosa E4R3')
plt.plot(um21,sloper21,'r.',label='pTet-J23101-J23101 Glucosa E2R3')
plt.plot(um27,sloper27,'r*',label='pLux76-J23101-J23101 Glucosa E3R3')
plt.plot(um33,sloper33,'r+',label='J23106-J23101-J23101 Glucosa E4R3')
plt.plot(um6,sloper6,'ko',label='Control Glucosa E2R3')
plt.plot(um12,sloper12,'kp',label='Control Glucosa E3R3')
plt.plot(um18,sloper18,'k^',label='Control Glucosa E4R3')
plt.plot(um24,sloper24,'mo',label='pTet-J23101-J23101 Glucosa E2R3')
plt.plot(um30,sloper30,'mp',label='pLux76-J23101-J23101 Glucosa E3R3')
plt.plot(um36,sloper36,'m^',label='J23106-J23101-J23101 Glucosa E4R3')
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 0x203688a4b38>
```





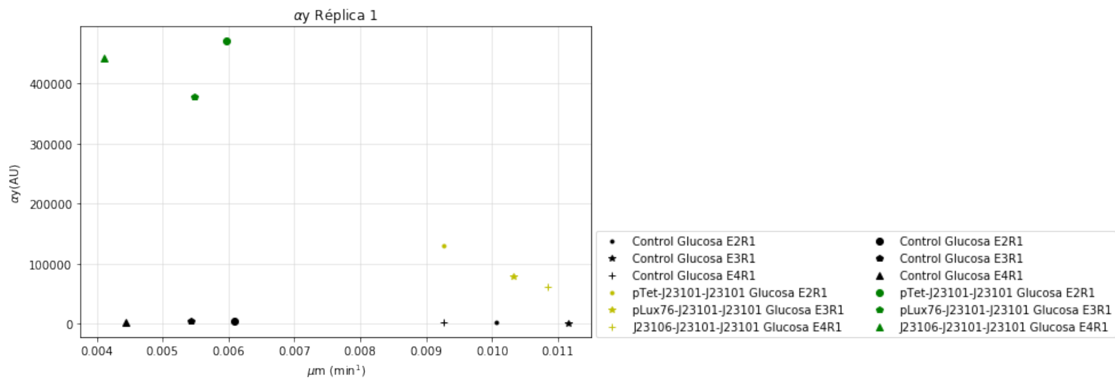


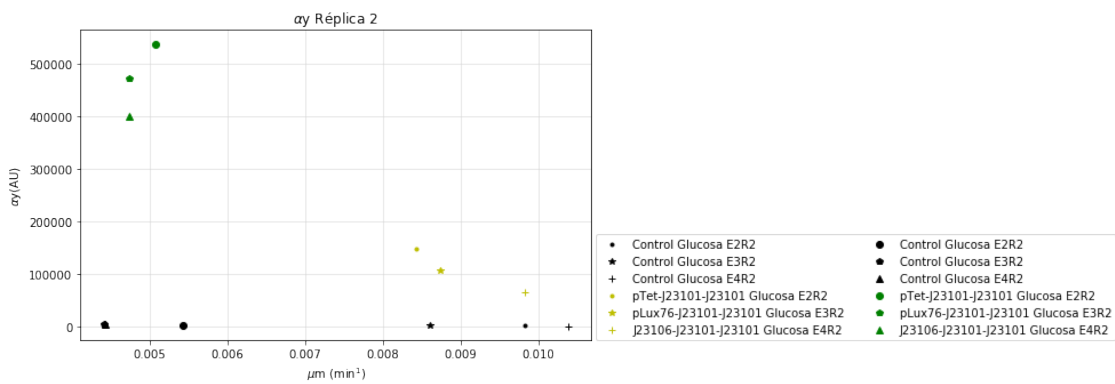
In [82]:
```
#grafico de ay versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
```

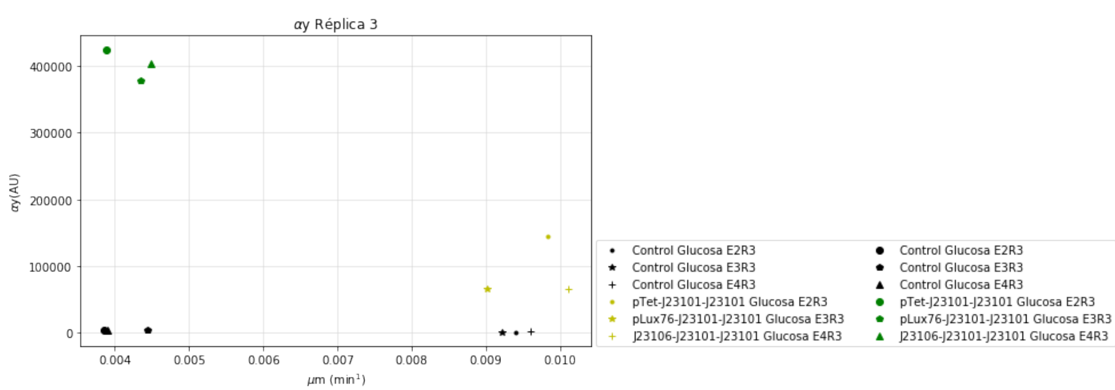
```
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'k.',label='Control Glucosa E2R1')
plt.plot(um7,slopey7,'k*',label='Control Glucosa E3R1')
plt.plot(um13,slopey13,'k+',label='Control Glucosa E4R1')
plt.plot(um19,slopey19,'y.',label='pTet-J23101-J23101 Glucosa E2R1')
plt.plot(um25,slopey25,'y*',label='pLux76-J23101-J23101 Glucosa E3R1')
plt.plot(um31,slopey31,'y+',label='J23106-J23101-J23101 Glucosa E4R1')
plt.plot(um4,slopey4,'ko',label='Control Glucosa E2R1')
plt.plot(um10,slopey10,'kp',label='Control Glucosa E3R1')
plt.plot(um16,slopey16,'k^',label='Control Glucosa E4R1')
plt.plot(um22,slopey22,'go',label='pTet-J23101-J23101 Glucosa E2R1')
plt.plot(um28,slopey28,'gp',label='pLux76-J23101-J23101 Glucosa E3R1')
plt.plot(um34,slopey34,'g^',label='J23106-J23101-J23101 Glucosa E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ay versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Réplica 2')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um2,slopey2,'k.',label='Control Glucosa E2R2')
plt.plot(um8,slopey8,'k*',label='Control Glucosa E3R2')
plt.plot(um14,slopey14,'k+',label='Control Glucosa E4R2')
plt.plot(um20,slopey20,'y.',label='pTet-J23101-J23101 Glucosa E2R2')
plt.plot(um26,slopey26,'y*',label='pLux76-J23101-J23101 Glucosa E3R2')
plt.plot(um32,slopey32,'y+',label='J23106-J23101-J23101 Glucosa E4R2')
plt.plot(um5,slopey5,'ko',label='Control Glucosa E2R2')
plt.plot(um11,slopey11,'kp',label='Control Glucosa E3R2')
plt.plot(um17,slopey17,'k^',label='Control Glucosa E4R2')
plt.plot(um23,slopey23,'go',label='pTet-J23101-J23101 Glucosa E2R2')
plt.plot(um29,slopey29,'gp',label='pLux76-J23101-J23101 Glucosa E3R2')
plt.plot(um35,slopey35,'g^',label='J23106-J23101-J23101 Glucosa E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ay  versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Réplica 3')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um3,slopey3,'k.',label='Control Glucosa E2R3')
plt.plot(um9,slopey9,'k*',label='Control Glucosa E3R3')
plt.plot(um15,slopey15,'k+',label='Control Glucosa E4R3')
plt.plot(um21,slopey21,'y.',label='pTet-J23101-J23101 Glucosa E2R3')
plt.plot(um27,slopey27,'y*',label='pLux76-J23101-J23101 Glucosa E3R3')
plt.plot(um33,slopey33,'y+',label='J23106-J23101-J23101 Glucosa E4R3')
plt.plot(um6,slopey6,'ko',label='Control Glucosa E2R3')
```

```
plt.plot(um12,slopey12,'kp',label='Control Glucosa E3R3')
plt.plot(um18,slopey18,'k^',label='Control Glucosa E4R3')
plt.plot(um24,slopey24,'go',label='pTet-J23101-J23101 Glucosa E2R3')
plt.plot(um30,slopey30,'gp',label='pLux76-J23101-J23101 Glucosa E3R3')
plt.plot(um36,slopey36,'g^',label='J23106-J23101-J23101 Glucosa E4R3')
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 0x2036906eba8>



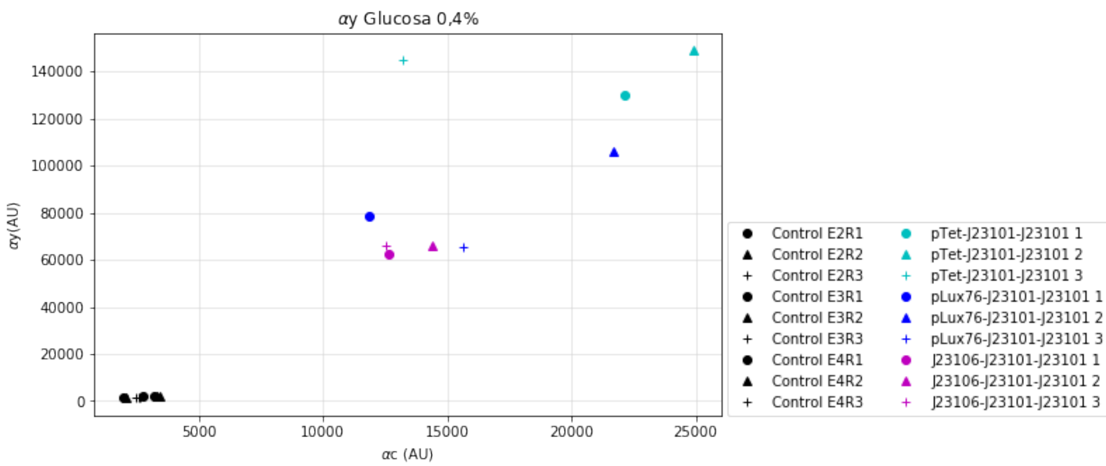


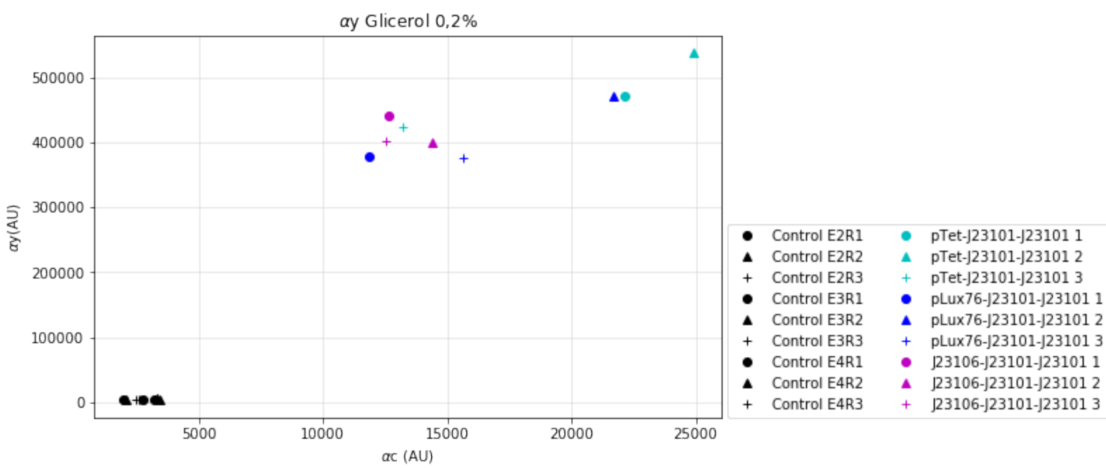


In [83]:
```
#grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopey1,'ko',label='Control E2R1')
plt.plot(slopec2,slopey2,'k^',label='Control E2R2')
plt.plot(slopec3,slopey3,'k+',label='Control E2R3')
plt.plot(slopec7,slopey7,'ko',label='Control E3R1')
plt.plot(slopec8,slopey8,'k^',label='Control E3R2')
plt.plot(slopec9,slopey9,'k+',label='Control E3R3')
plt.plot(slopec13,slopey13,'ko',label='Control E4R1')
plt.plot(slopec14,slopey14,'k^',label='Control E4R2')
plt.plot(slopec15,slopey15,'k+',label='Control E4R3')
plt.plot(slopec19,slopey19,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec20,slopey20,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec21,slopey21,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec25,slopey25,'bo',label='pLux76-J23101-J23101 1')
plt.plot(slopec26,slopey26,'b^',label='pLux76-J23101-J23101 2')
plt.plot(slopec27,slopey27,'b+',label='pLux76-J23101-J23101 3')
plt.plot(slopec31,slopey31,'mo',label='J23106-J23101-J23101 1')
plt.plot(slopec32,slopey32,'m^',label='J23106-J23101-J23101 2')
plt.plot(slopec33,slopey33,'m+',label='J23106-J23101-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ay vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopey4,'ko',label='Control E2R1')
plt.plot(slopec2,slopey5,'k^',label='Control E2R2')
plt.plot(slopec3,slopey6,'k+',label='Control E2R3')
plt.plot(slopec7,slopey10,'ko',label='Control E3R1')
plt.plot(slopec8,slopey11,'k^',label='Control E3R2')
plt.plot(slopec9,slopey12,'k+',label='Control E3R3')
plt.plot(slopec13,slopey16,'ko',label='Control E4R1')
plt.plot(slopec14,slopey17,'k^',label='Control E4R2')
plt.plot(slopec15,slopey18,'k+',label='Control E4R3')
plt.plot(slopec19,slopey22,'co',label='pTet-J23101-J23101 1')
plt.plot(slopec20,slopey23,'c^',label='pTet-J23101-J23101 2')
plt.plot(slopec21,slopey24,'c+',label='pTet-J23101-J23101 3')
plt.plot(slopec25,slopey28,'bo',label='pLux76-J23101-J23101 1')
plt.plot(slopec26,slopey29,'b^',label='pLux76-J23101-J23101 2')
```

```
plt.plot(slopec27,slopey30,'b+',label='pLux76-J23101-J23101 3')
plt.plot(slopec31,slopey34,'mo',label='J23106-J23101-J23101 1')
plt.plot(slopec32,slopey35,'m^',label='J23106-J23101-J23101 2')
plt.plot(slopec33,slopey36,'m+',label='J23106-J23101-J23101 3')
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 0x20368e9c048>

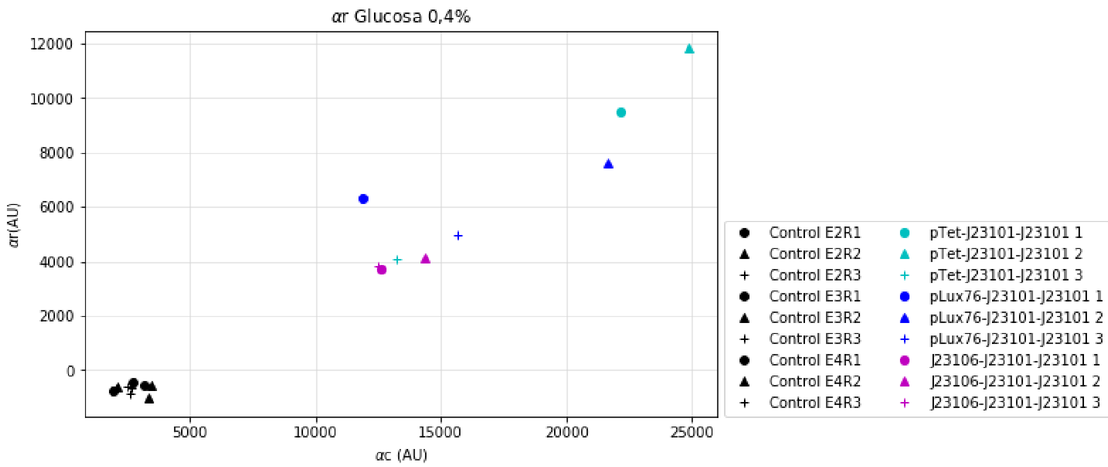


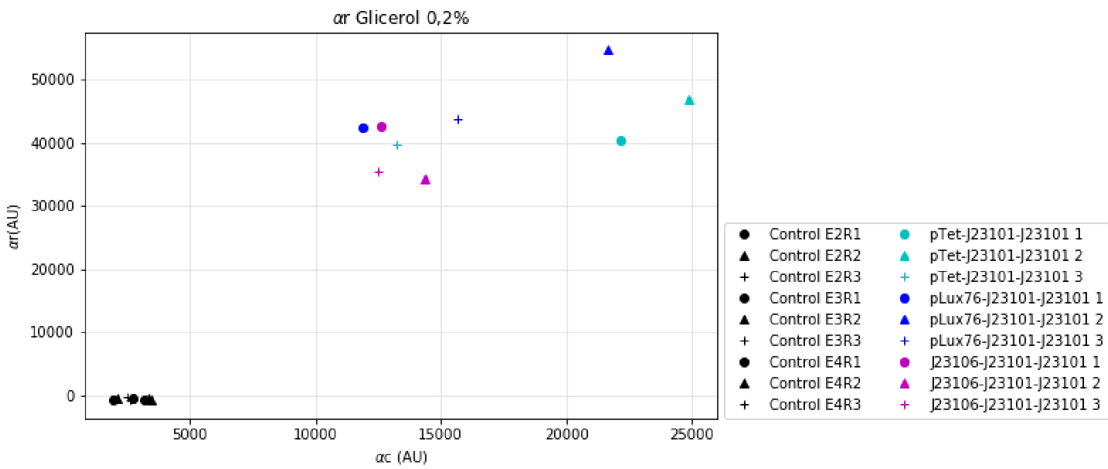



```
In [84]: #grafico de ar vs ac
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Glucosa 0,4%')
         plt.xlabel(r'$\alpha$c (AU)')
         plt.ylabel(r'$\alpha$r(AU)')
```

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

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

```
Out[84]: <matplotlib.legend.Legend at 0x203678b0390>
```

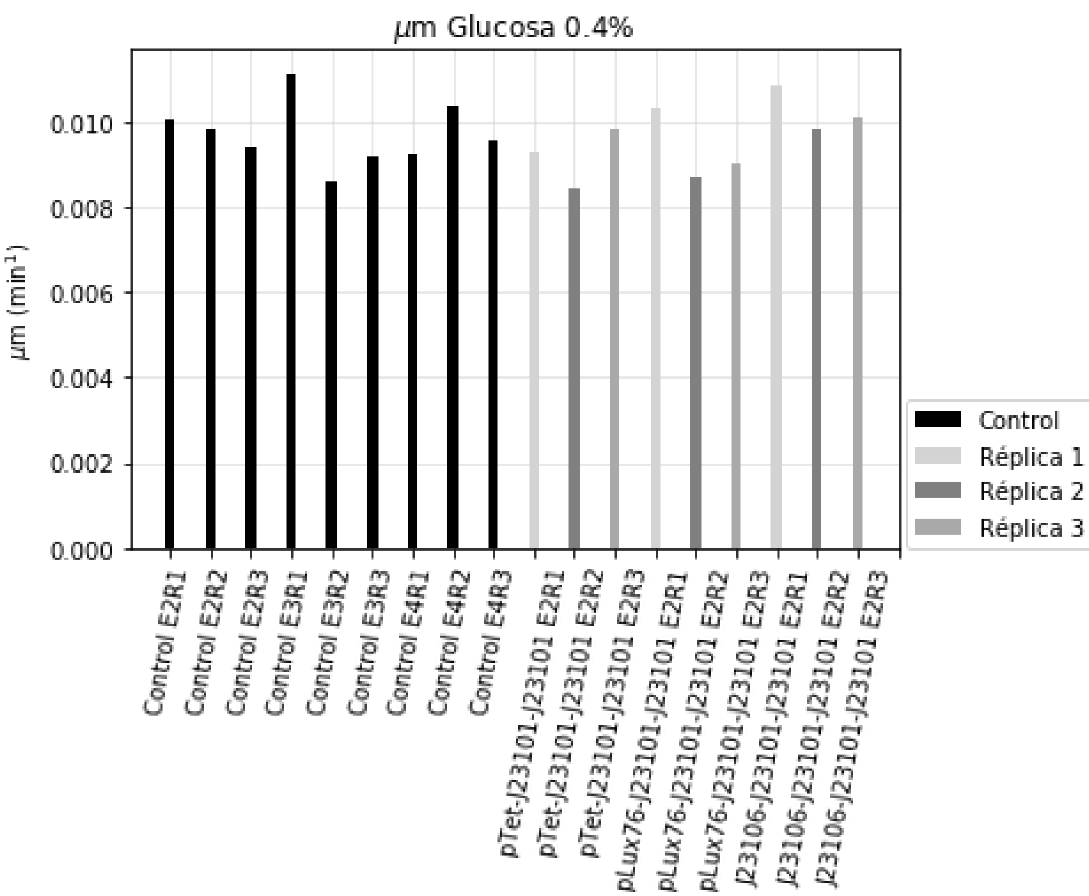




In [85]:
```
#Grafico de barras um de FPs
uglu=[um1,um2,um3,um7,um8,um9,um13,um14,um15,um19,um20,um21,um25,um26,um27,um31,um32,um
ugli=[um4,um5,um6,um10,um11,um12,um16,um17,um18,um22,um23,um24,um28,um29,um30,um34,um35

X = np.arange(19)
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='k',width=0.25,zorder=3)
plt.bar(X[4]+0.00,uglu[4],color='k',width=0.25,zorder=3)
plt.bar(X[5]+0.00,uglu[5],color='k',width=0.25,zorder=3)
```

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

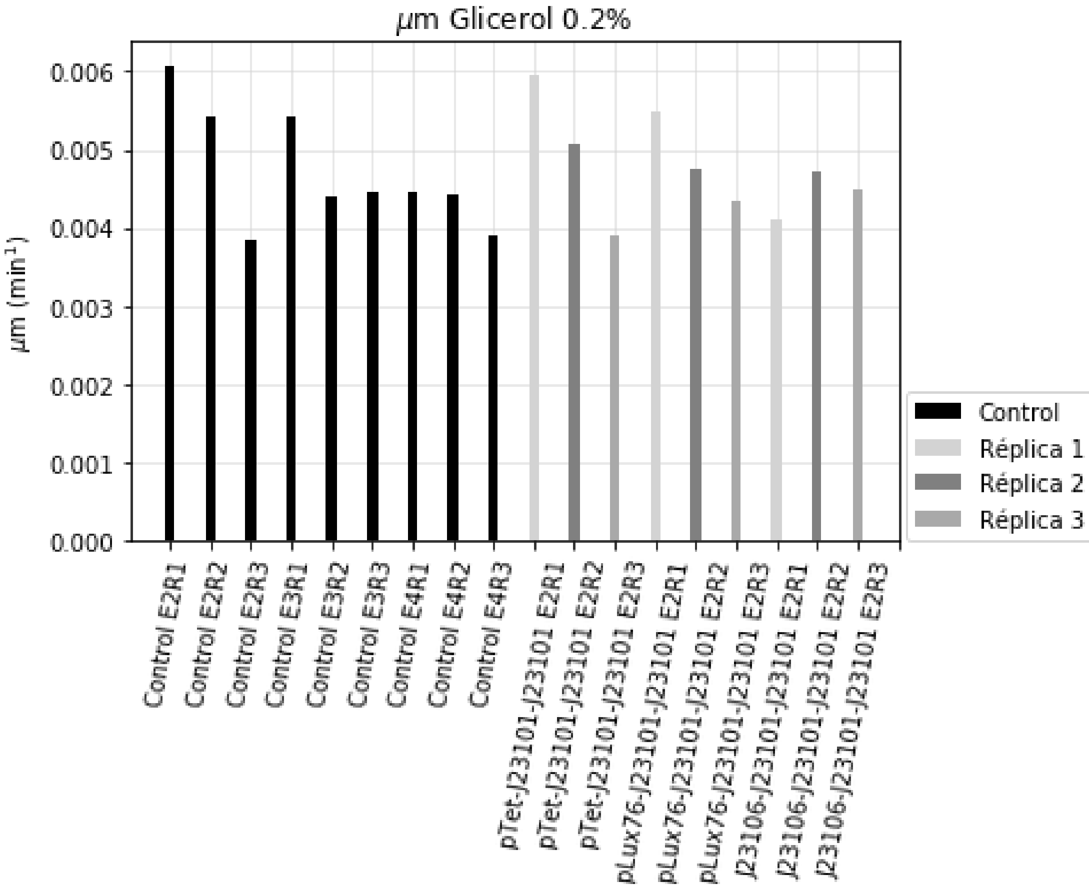
Out[85]: <matplotlib.legend.Legend at 0x203634716a0>



In [86]:
```
X = np.arange(19)
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='k',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='k',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='k',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='k',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='k',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='k',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ugli[15],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ugli[16],color='grey',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ugli[17],color='darkgrey',width=0.25,zorder=3)
plt.xticks(X,['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R2
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

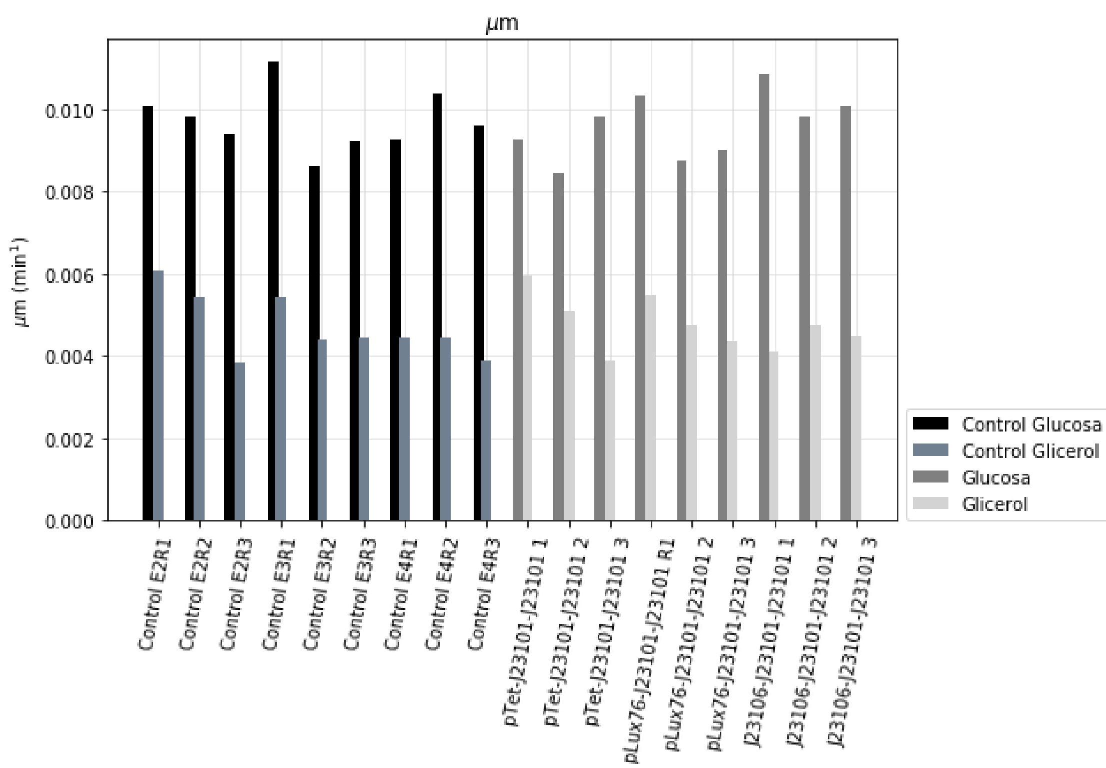
Out[86]: <matplotlib.legend.Legend at 0x203660634e0>




```
In [87]: X = np.arange(18)
         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]-0.20,uglu[3],color='k',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,ugli[3],color='slategrey',width=0.25,zorder=3)
         plt.bar(X[4]-0.20,uglu[4],color='k',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,ugli[4],color='slategrey',width=0.25,zorder=3)
         plt.bar(X[5]-0.20,uglu[5],color='k',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,ugli[5],color='slategrey',width=0.25,zorder=3)
         plt.bar(X[6]-0.20,uglu[6],color='k',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,ugli[6],color='slategrey',width=0.25,zorder=3)
         plt.bar(X[7]-0.20,uglu[7],color='k',width=0.25,zorder=3)
```

```
plt.bar(X[7]+0.00,ugli[7],color='slategrey',width=0.25,zorder=3)
plt.bar(X[8]-0.20,uglu[8],color='k',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='slategrey',width=0.25,zorder=3)
plt.bar(X[9]-0.25,uglu[9],color='grey',width=0.25,label='Glucosa',zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,label='Glicerol',zorder=3)
plt.bar(X[10]-0.25,uglu[10],color='grey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[11]-0.25,uglu[11],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[12]-0.25,uglu[12],color='grey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]-0.25,uglu[13],color='grey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[14]-0.25,uglu[14],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[15]-0.25,uglu[15],color='grey',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ugli[15],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[16]-0.25,uglu[16],color='grey',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ugli[16],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[17]-0.25,uglu[17],color='grey',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ugli[17],color='lightgrey',width=0.25,zorder=3)
plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

Out[87]: <matplotlib.legend.Legend at 0x20366528ac8>



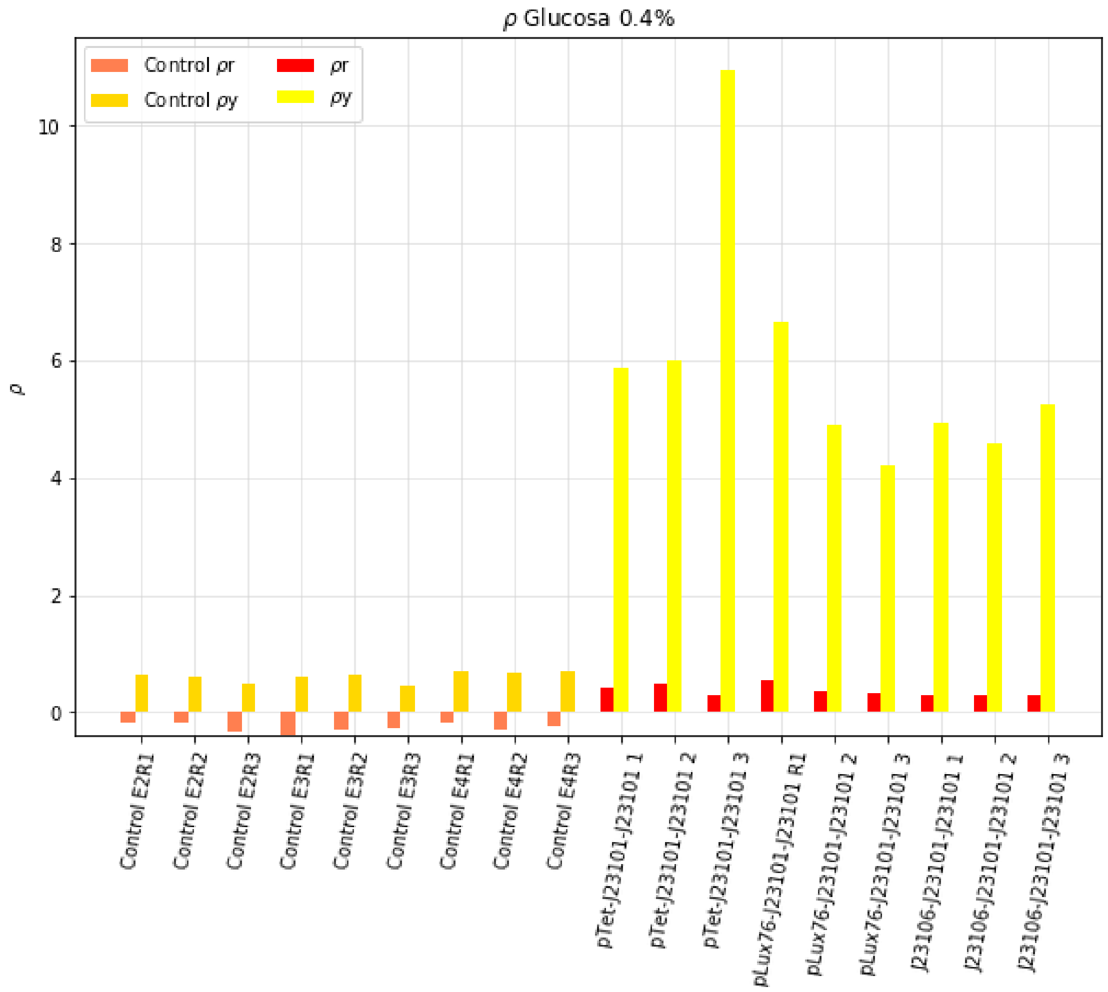
```
In [88]: #Ro RFP
         pr1=sloper1/slopec1
         pr2=sloper2/slopec2
         pr3=sloper3/slopec3
         pr4=sloper4/slopec4
         pr5=sloper5/slopec5
         pr6=sloper6/slopec6
         pr7=sloper7/slopec7
         pr8=sloper8/slopec8
         pr9=sloper9/slopec9
         pr10=sloper10/slopec10
         pr11=sloper11/slopec11
         pr12=sloper12/slopec12
         pr13=sloper13/slopec13
         pr14=sloper14/slopec14
         pr15=sloper15/slopec15
         pr16=sloper16/slopec16
         pr17=sloper17/slopec17
         pr18=sloper18/slopec18
         pr19=sloper19/slopec19
         pr20=sloper20/slopec20
         pr21=sloper21/slopec21
         pr22=sloper22/slopec22
         pr23=sloper23/slopec23
         pr24=sloper24/slopec24
         pr25=sloper25/slopec25
         pr26=sloper26/slopec26
         pr27=sloper27/slopec27
         pr28=sloper28/slopec28
         pr29=sloper29/slopec29
         pr30=sloper30/slopec30
         pr31=sloper31/slopec31
         pr32=sloper32/slopec32
         pr33=sloper33/slopec33
         pr34=sloper34/slopec34
         pr35=sloper35/slopec35
         pr36=sloper36/slopec36


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

In [89]: #Ro YFP
         py1=slopey1/slopec1
```

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


ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,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 [90]: X = np.arange(18)
         plt.figure(figsize=(10,7))
         plt.title(r'$\rho$ Glucosa 0.4%')
         plt.ylabel(r'$\rho$')
         plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r',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='coral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[6],color='gold',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[7],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[7],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[8],color='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[8],color='gold',width=0.25,zorder=3)
plt.bar(X[6]-0.25,ro_rfp[12],color='coral',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ro_yfp[12],color='gold',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[13],color='coral',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[13],color='gold',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[14],color='coral',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[14],color='gold',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[18],color='r',width=0.25,label=r'$\rho$r',zorder=3)
plt.bar(X[9]+0.00,ro_yfp[18],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
plt.bar(X[10]-0.25,ro_rfp[19],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[19],color='yellow',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[20],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[20],color='yellow',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[24],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[24],color='yellow',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[25],color='r',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[25],color='yellow',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[26],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[26],color='yellow',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[30],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[30],color='yellow',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[31],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[31],color='yellow',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[32],color='r',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[32],color='yellow',width=0.25,zorder=3)

plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

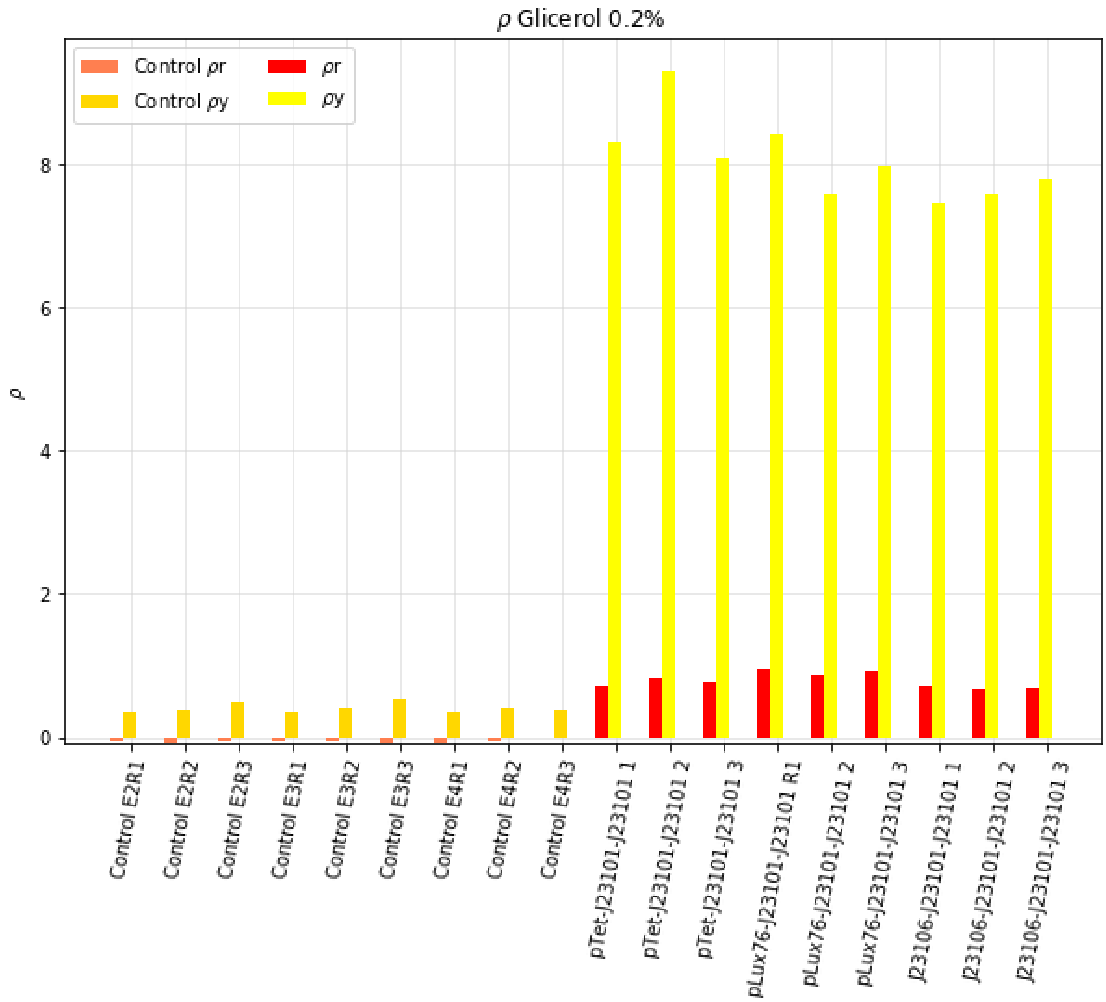
Out[90]: <matplotlib.legend.Legend at 0x20368fa50b8>



In [91]:
```
X = np.arange(18)
plt.figure(figsize=(10,7))
plt.title(r'$\rho$ Glicerol 0.2%')
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[3],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r',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='coral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[9],color='gold',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[10],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[10],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[11],color='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[11],color='gold',width=0.25,zorder=3)
```

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

plt.xticks(X, ['Control E2R1','Control E2R2','Control E2R3','Control E3R1','Control E3R
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

```
Out[91]: <matplotlib.legend.Legend at 0x20369171908>
```



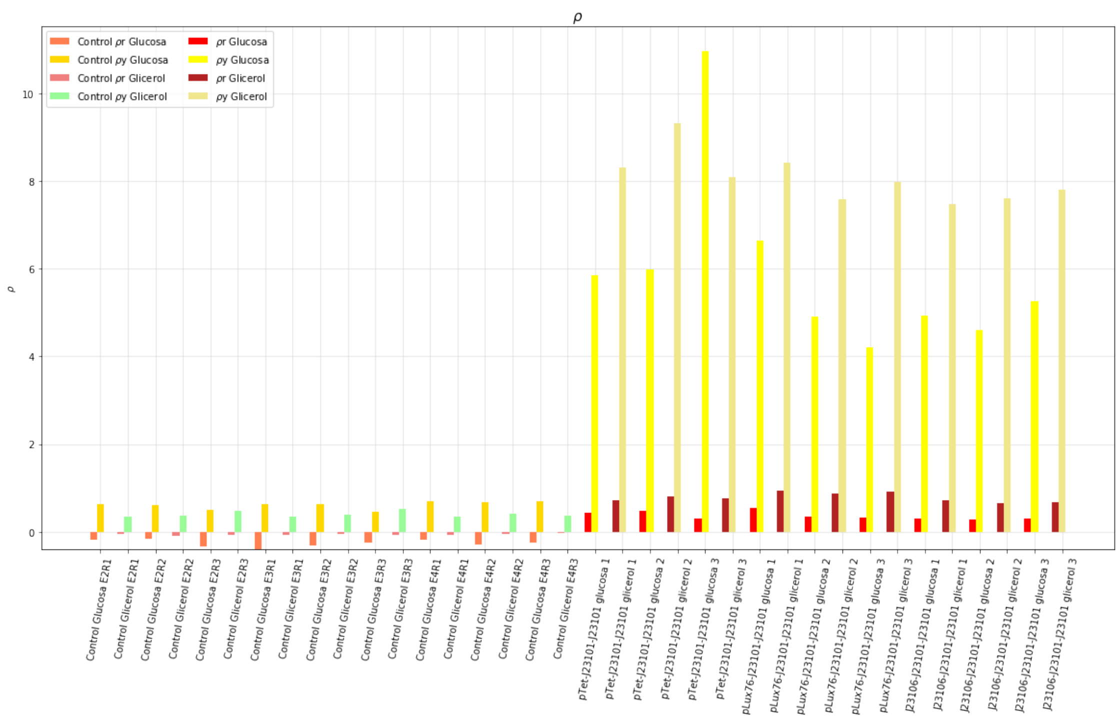

```
In [92]: X = np.arange(36)
         plt.figure(figsize=(20,10))
         plt.title(r'$\rho$',fontsize=15.0)
         plt.ylabel(r'$\rho$')
         plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r Glu
         plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y 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='coral',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ro_yfp[6],color='gold',width=0.25,zorder=3)
plt.bar(X[7]-0.25,ro_rfp[9],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ro_yfp[9],color='palegreen',width=0.25,zorder=3)
plt.bar(X[8]-0.25,ro_rfp[7],color='coral',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ro_yfp[7],color='gold',width=0.25,zorder=3)
plt.bar(X[9]-0.25,ro_rfp[10],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ro_yfp[10],color='palegreen',width=0.25,zorder=3)
plt.bar(X[10]-0.25,ro_rfp[8],color='coral',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ro_yfp[8],color='gold',width=0.25,zorder=3)
plt.bar(X[11]-0.25,ro_rfp[11],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ro_yfp[11],color='palegreen',width=0.25,zorder=3)
plt.bar(X[12]-0.25,ro_rfp[12],color='coral',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ro_yfp[12],color='gold',width=0.25,zorder=3)
plt.bar(X[13]-0.25,ro_rfp[15],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ro_yfp[15],color='palegreen',width=0.25,zorder=3)
plt.bar(X[14]-0.25,ro_rfp[13],color='coral',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ro_yfp[13],color='gold',width=0.25,zorder=3)
plt.bar(X[15]-0.25,ro_rfp[16],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[15]+0.00,ro_yfp[16],color='palegreen',width=0.25,zorder=3)
plt.bar(X[16]-0.25,ro_rfp[14],color='coral',width=0.25,zorder=3)
plt.bar(X[16]+0.00,ro_yfp[14],color='gold',width=0.25,zorder=3)
plt.bar(X[17]-0.25,ro_rfp[17],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[17]+0.00,ro_yfp[17],color='palegreen',width=0.25,zorder=3)
```

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

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

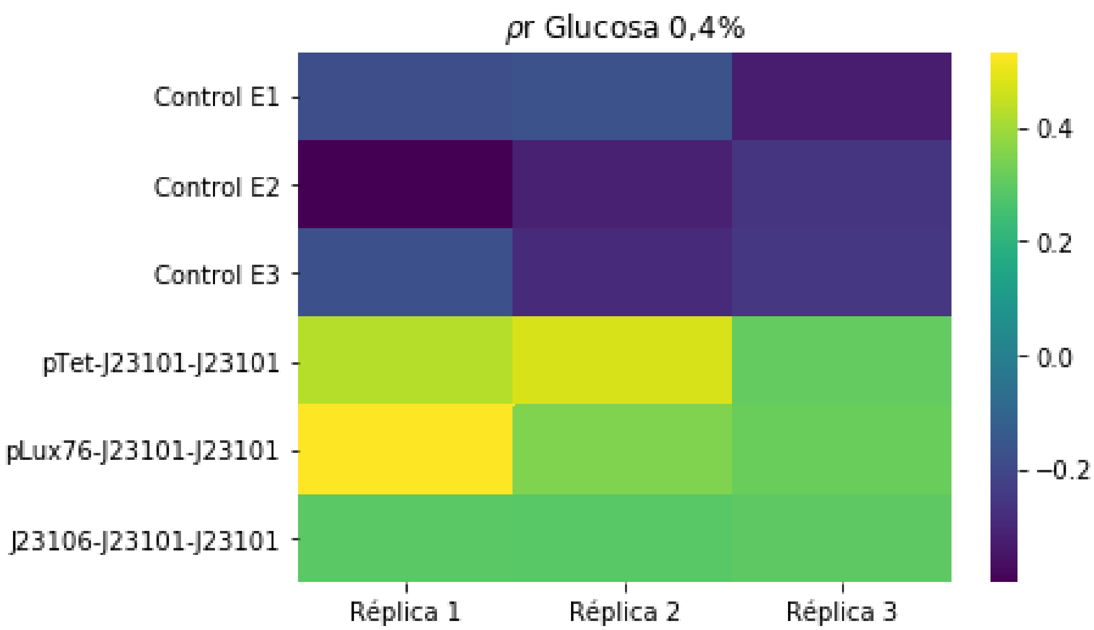
Out[92]: <matplotlib.legend.Legend at 0x20369bbb6a0>

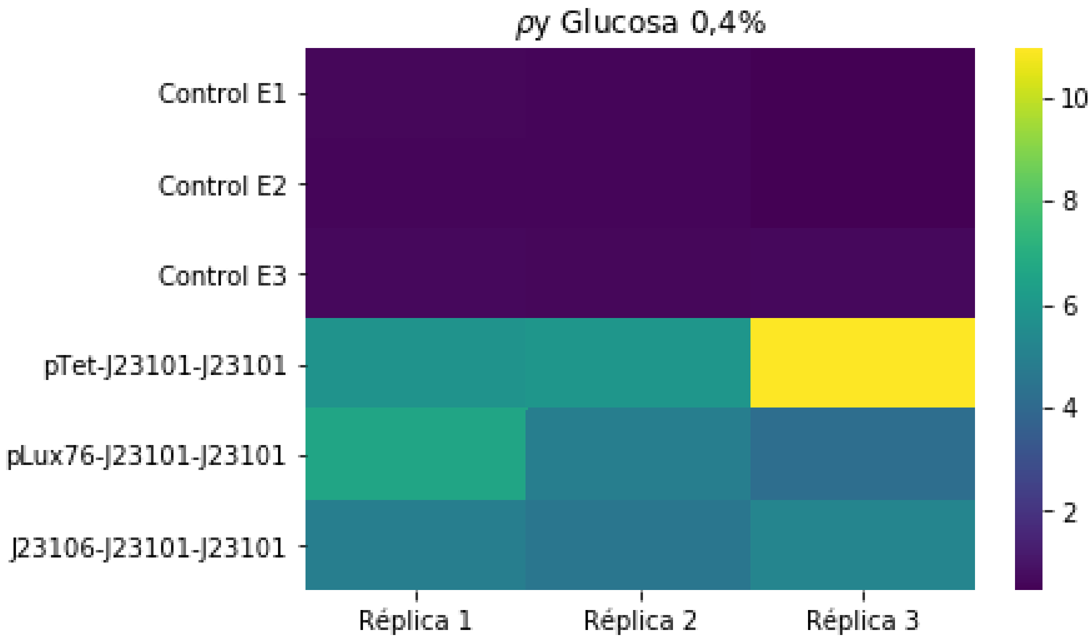


In [93]:
```
ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr2

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

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

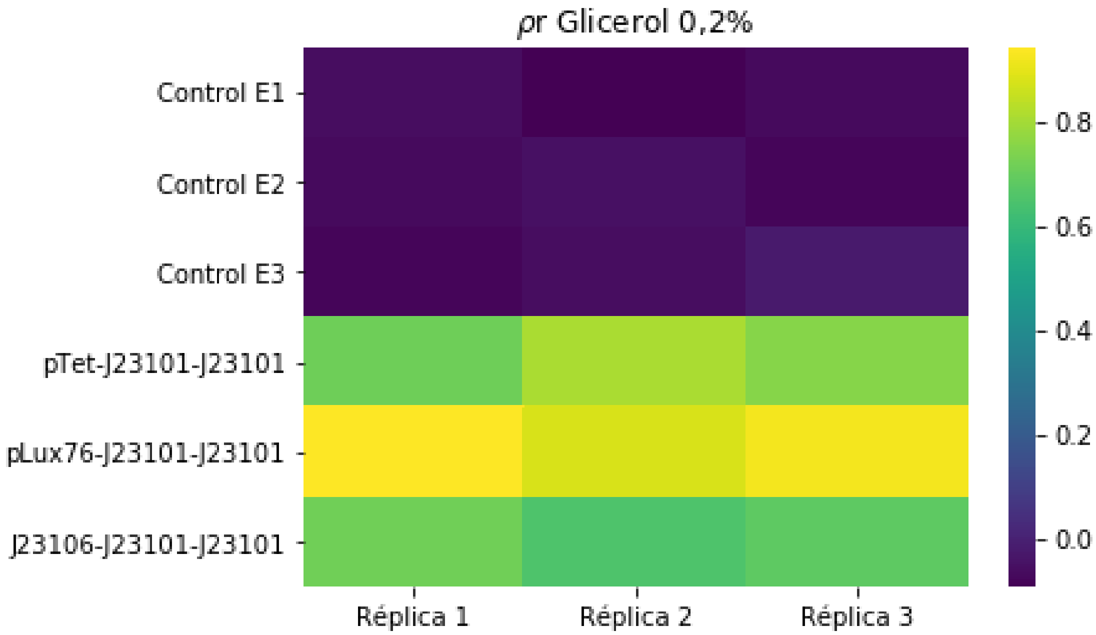
```
ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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

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

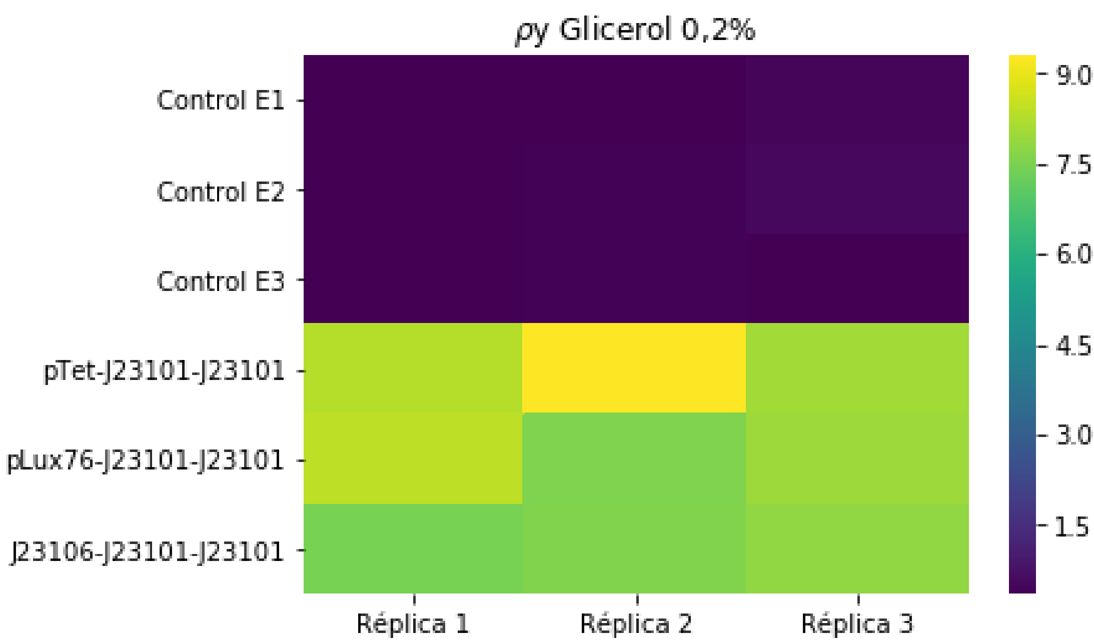
Out[93]: <matplotlib.axes._subplots.AxesSubplot at 0x2036785e0f0>






```
In [94]: 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 E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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

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

Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x2036671d978>
```

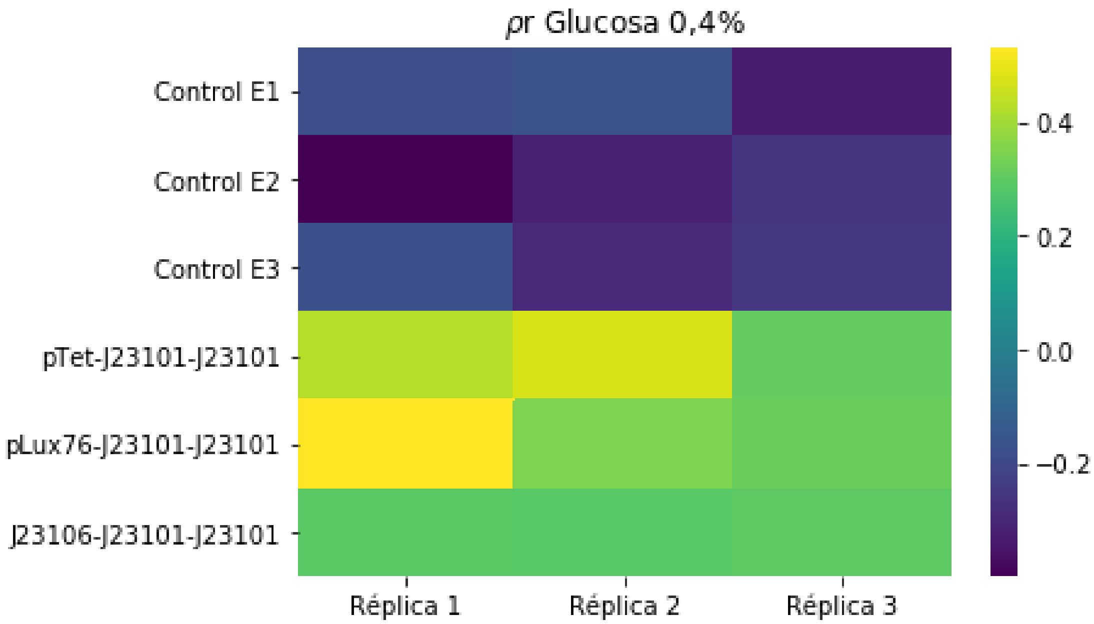


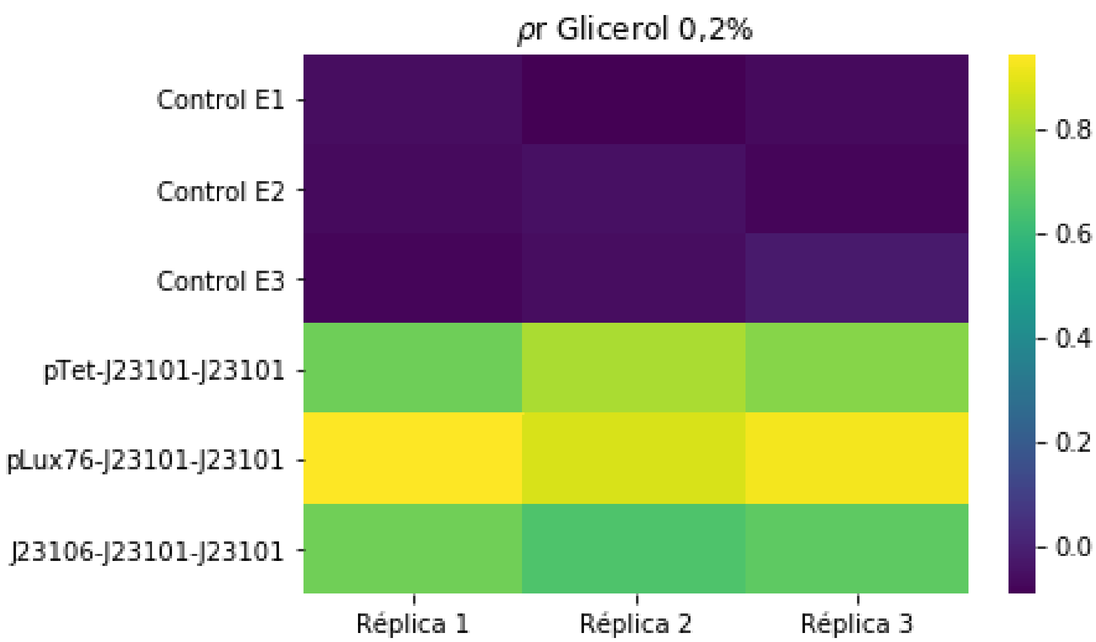



```
In [95]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

         plt.figure()
```

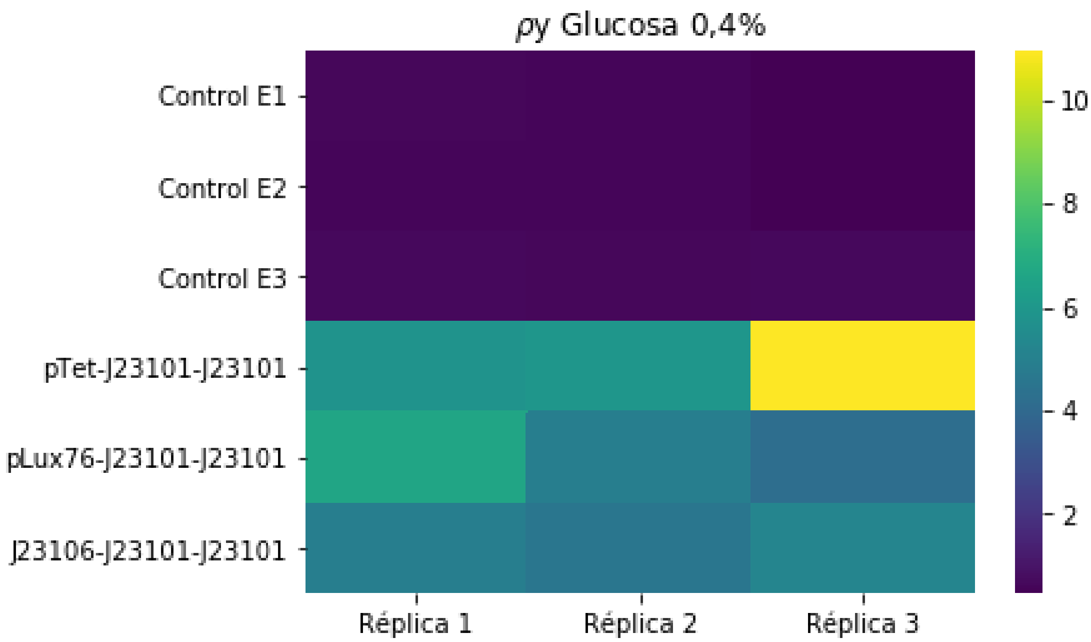
```
plt.title(r'$\rho$r Glucosa 0,4%')
sns.heatmap(ro_rfpglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

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

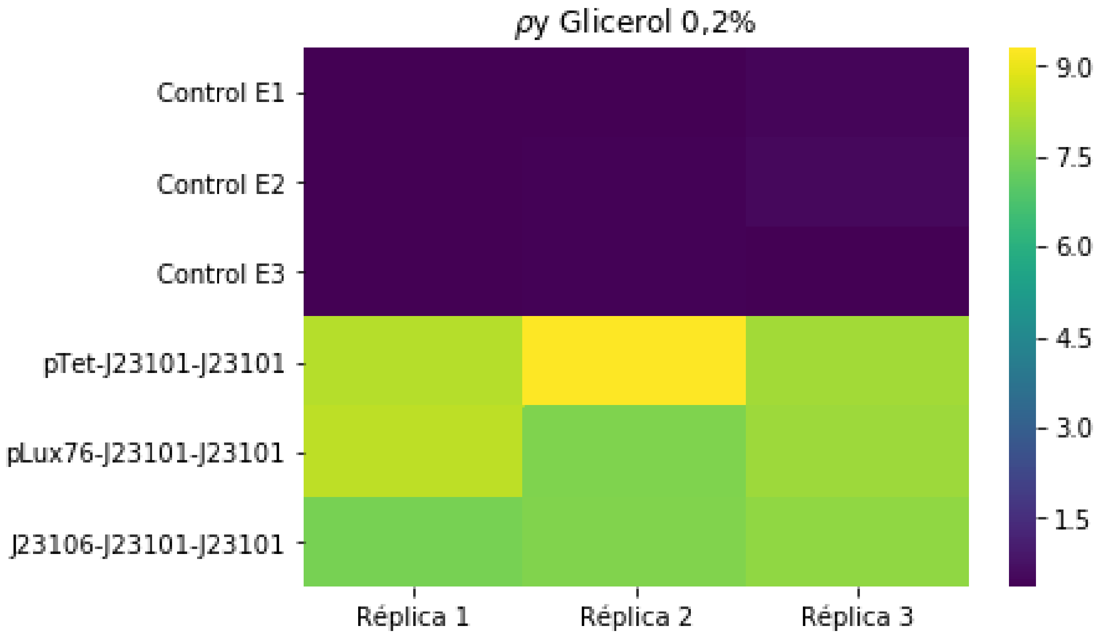
Out[95]: <matplotlib.axes._subplots.AxesSubplot at 0x20367ff3940>



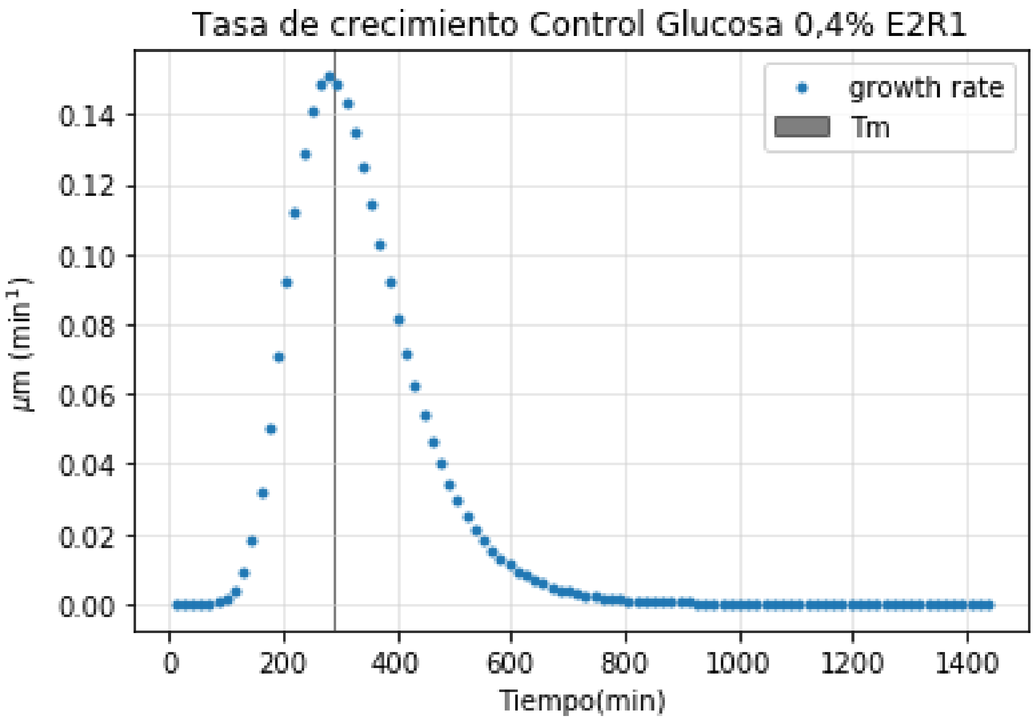


```
In [96]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control E1','Control E2','Control E3','pTet-J23101-J23101','pLux76-J23101-J231

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

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

Out[96]: <matplotlib.axes._subplots.AxesSubplot at 0x20368476d30>
```

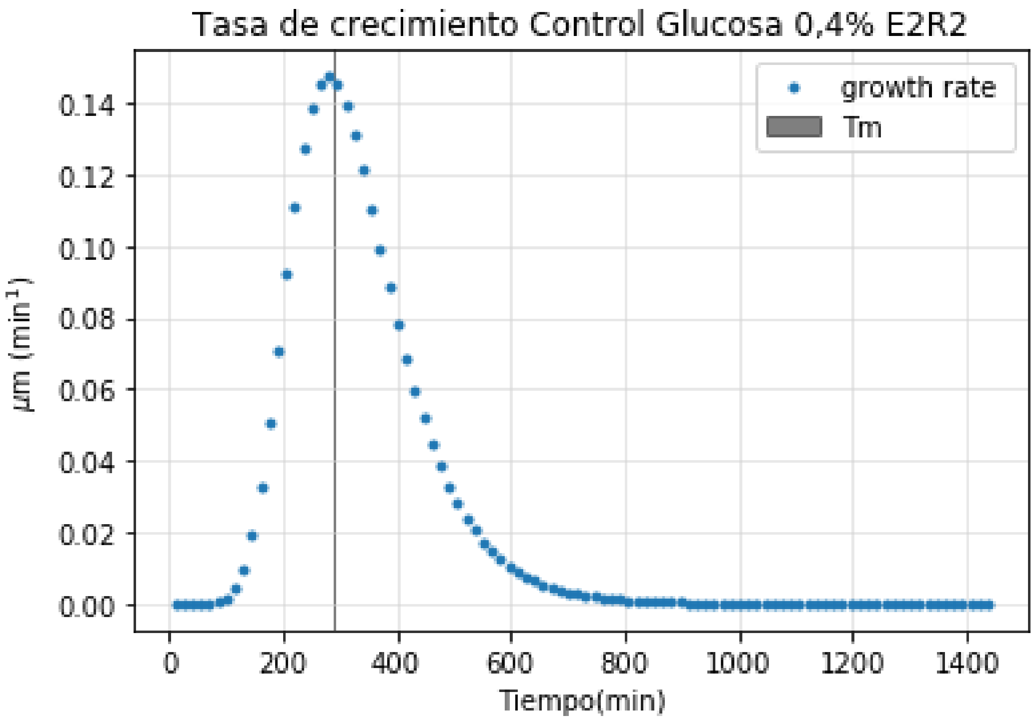





```
In [97]: #tasa de crecimiento
         ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(l1-tt))+1))))
         #Con diff
         dy1=(np.diff(ye1))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R1')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
         plt.plot(tt[:-1],dy1,'.',label='growth rate')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[97]: <matplotlib.legend.Legend at 0x203699139e8>
```

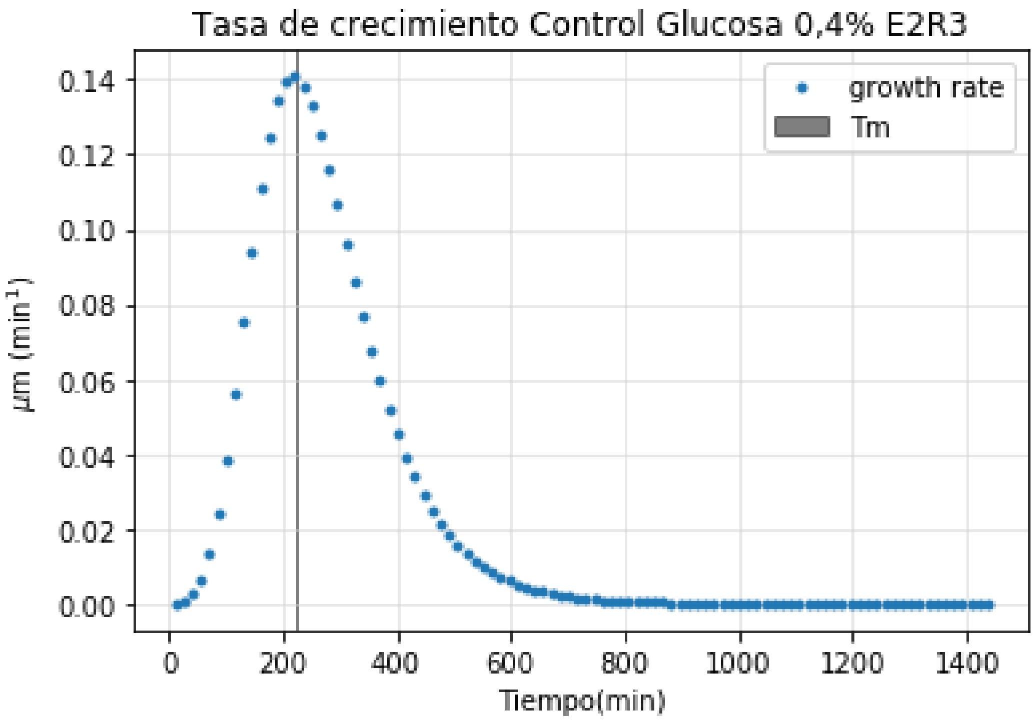



```
In [98]: #tasa de crecimiento

         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% E2R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.axvspan(tm2,tm2, color='k', alpha=0.5, label="Tm")
         plt.plot(tt[:-1],dy2,'.',label='growth rate ')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[98]: <matplotlib.legend.Legend at 0x2036a3bbef0>
```

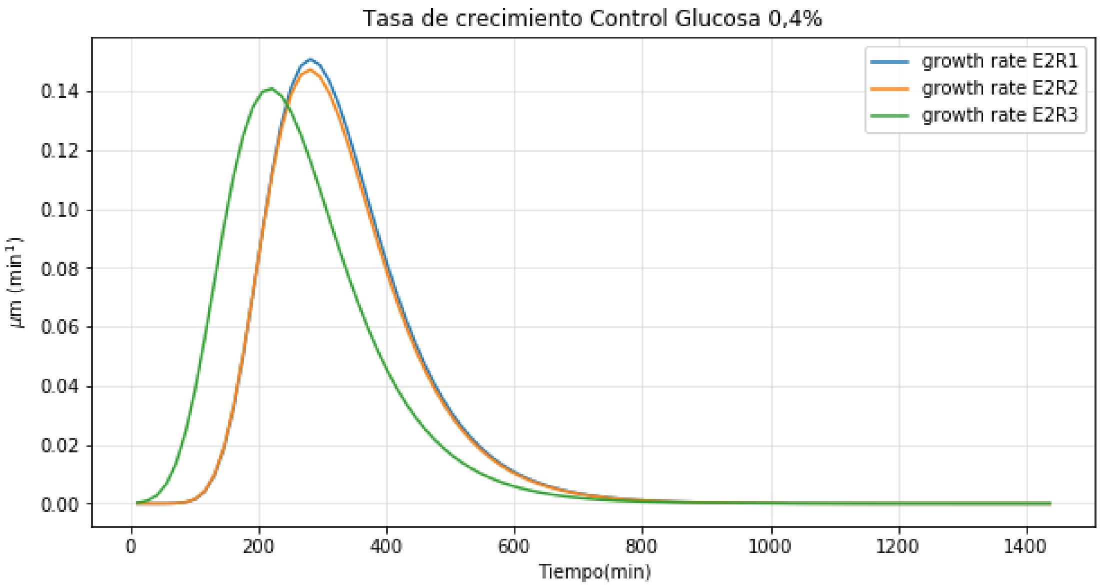


In [99]:
```
#tasa de crecimiento
ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(l3-tt))+1))))
#Con diff
dy3=(np.diff(ye3))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E2R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm3,tm3, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy3,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

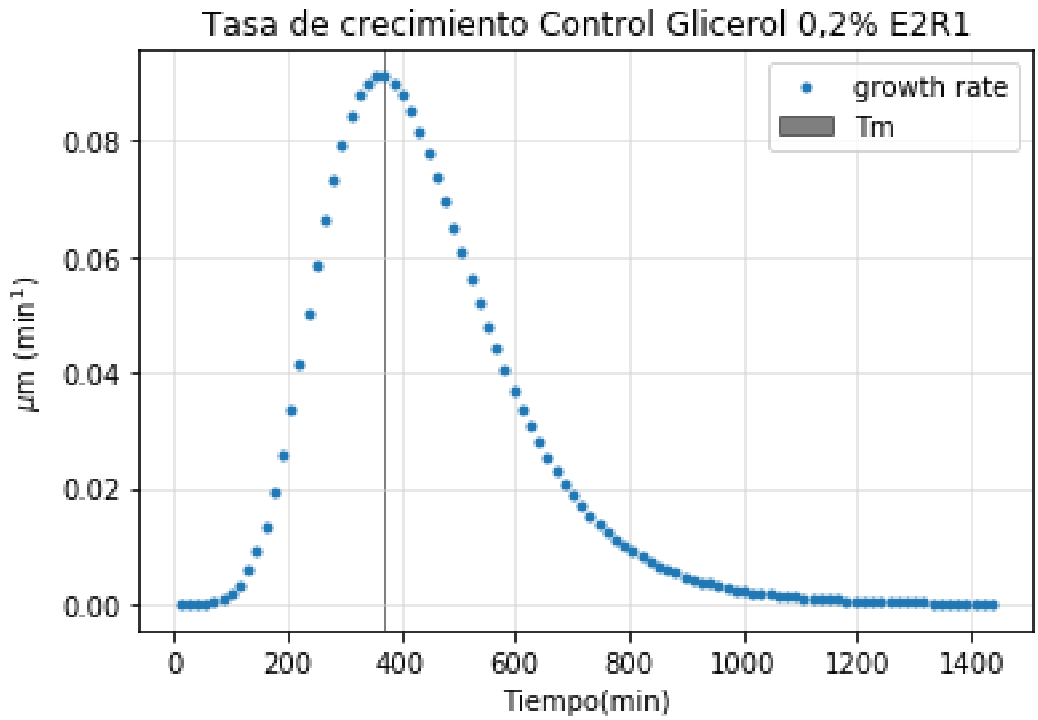
Out[99]:
```
<matplotlib.legend.Legend at 0x2036a48fdd8>
```



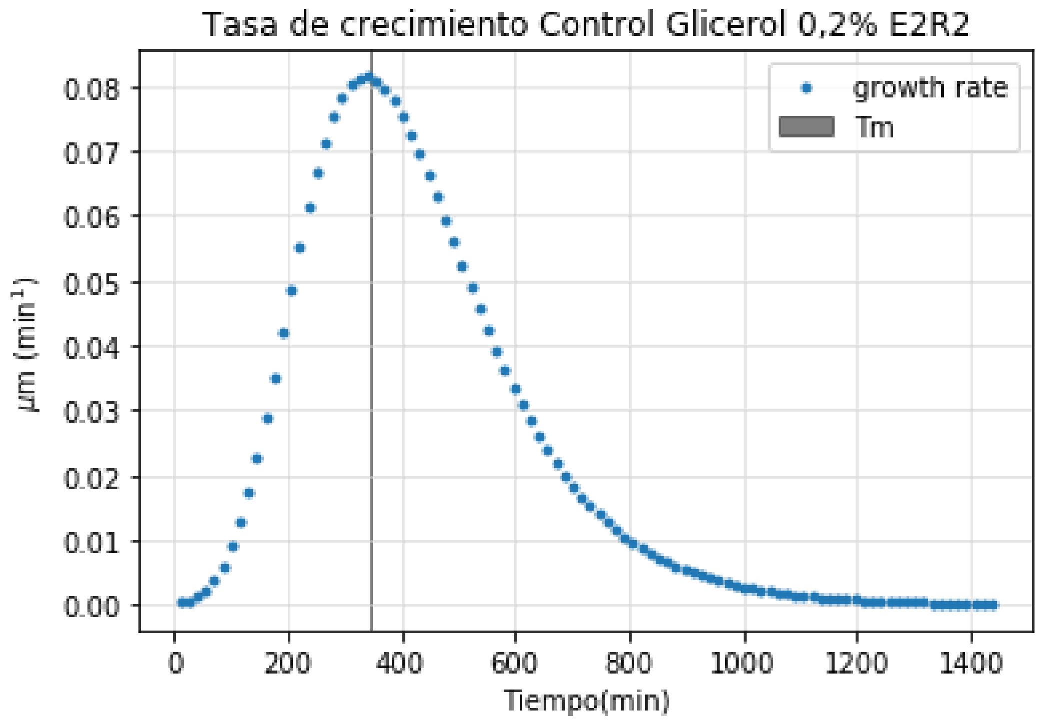

```
In [100]: #Tasas control réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,label='growth rate E2R1')
          plt.plot(tt[:-1],dy2,label='growth rate E2R2')
          plt.plot(tt[:-1],dy3,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[100]: <matplotlib.legend.Legend at 0x2036a56ae10>
```



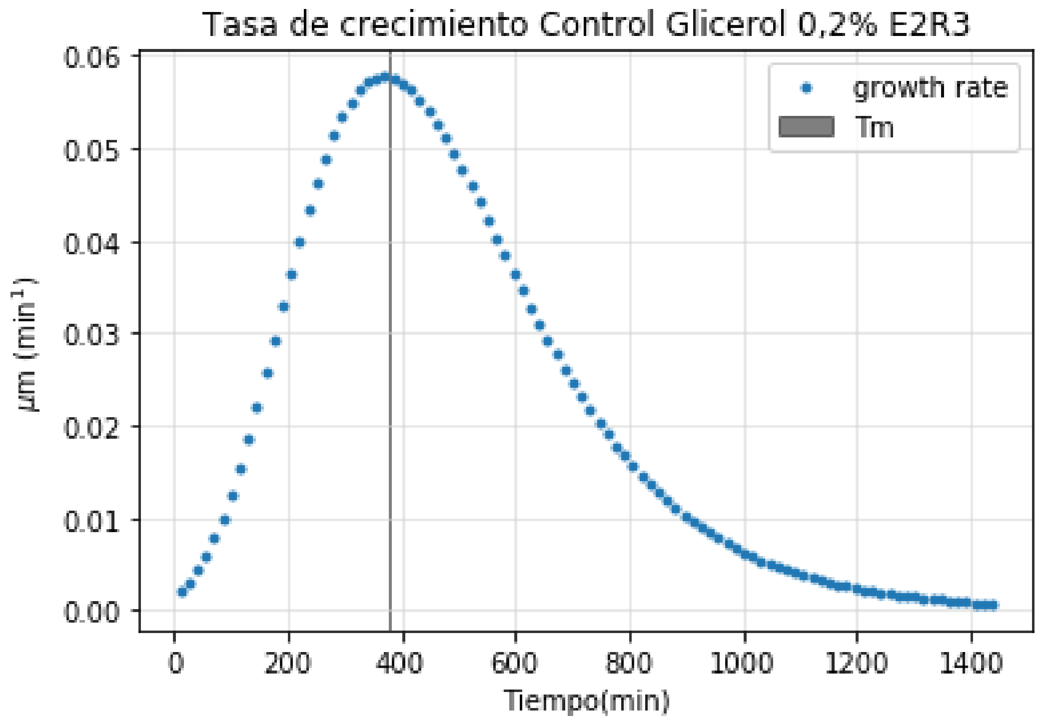

```
In [101]: #tasa de crecimiento
          ye4=((A4*np.exp(-np.exp((((um4*np.exp(1))/A4)*(l4-tt))+1))))
          #Con diff
          dy4=(np.diff(ye4))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm4,tm4, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy4,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[101]: <matplotlib.legend.Legend at 0x2036b613ba8>
```



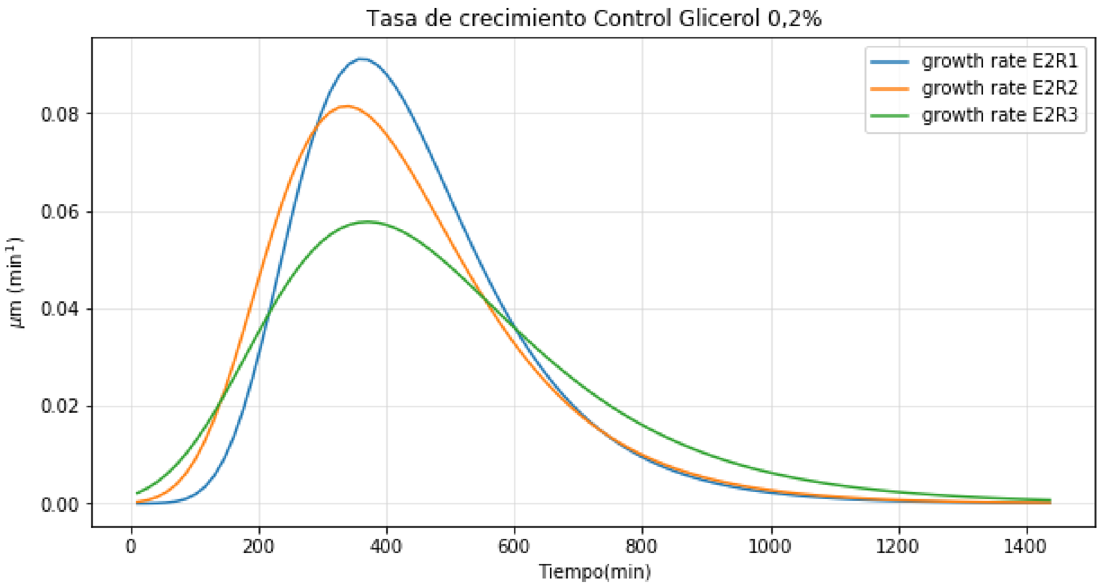

```
In [102]: #tasa de crecimiento
          ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(l5-tt))+1))))
          #Con diff
          dy5=(np.diff(ye5))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm5,tm5, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy5,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[102]: <matplotlib.legend.Legend at 0x2036b6e1eb8>
```



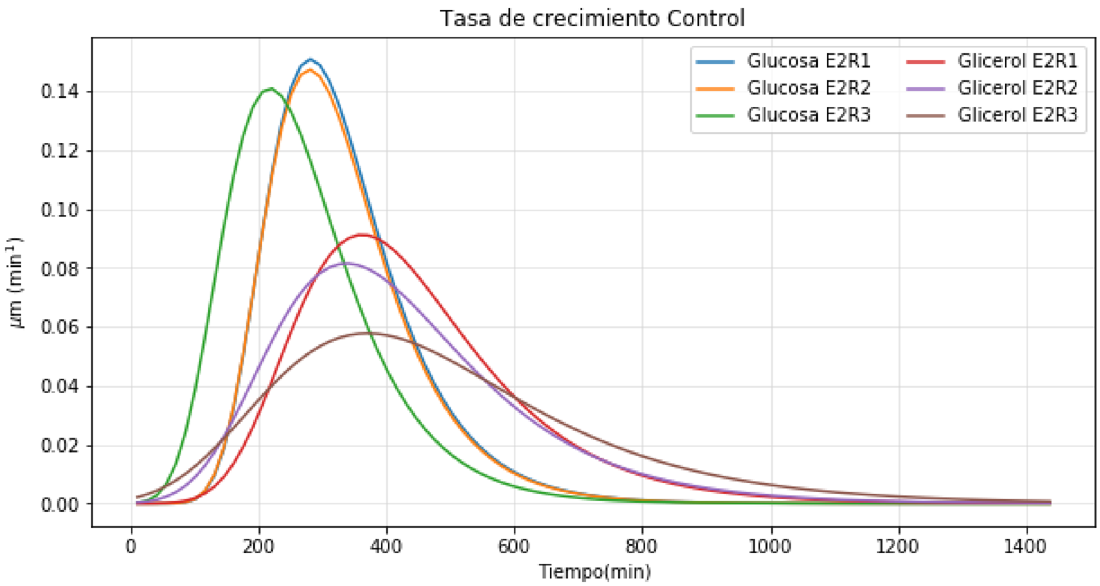

```
In [103]: #tasa de crecimiento
          ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(l6-tt))+1))))
          #Con diff
          dy6=(np.diff(ye6))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm6,tm6, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy6,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[103]: <matplotlib.legend.Legend at 0x2036b7c29b0>
```




```
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 E2R1')
          plt.plot(tt[:-1],dy5,label='growth rate E2R2')
          plt.plot(tt[:-1],dy6,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[104]: <matplotlib.legend.Legend at 0x2036b88c748>
```

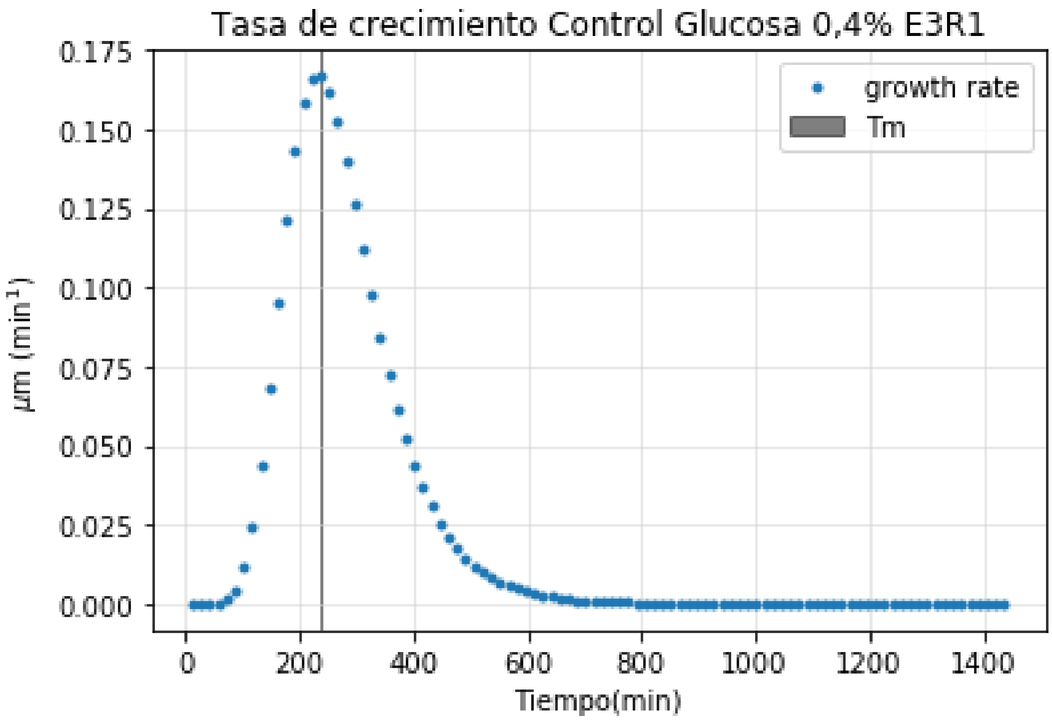



```
In [105]: #Tasas control réplicas controles
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,label='Glucosa E2R1')
          plt.plot(tt[:-1],dy2,label='Glucosa E2R2')
          plt.plot(tt[:-1],dy3,label='Glucosa E2R3')
          plt.plot(tt[:-1],dy4,label='Glicerol E2R1')
          plt.plot(tt[:-1],dy5,label='Glicerol E2R2')
          plt.plot(tt[:-1],dy6,label='Glicerol E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[105]: <matplotlib.legend.Legend at 0x2036b973dd8>
```

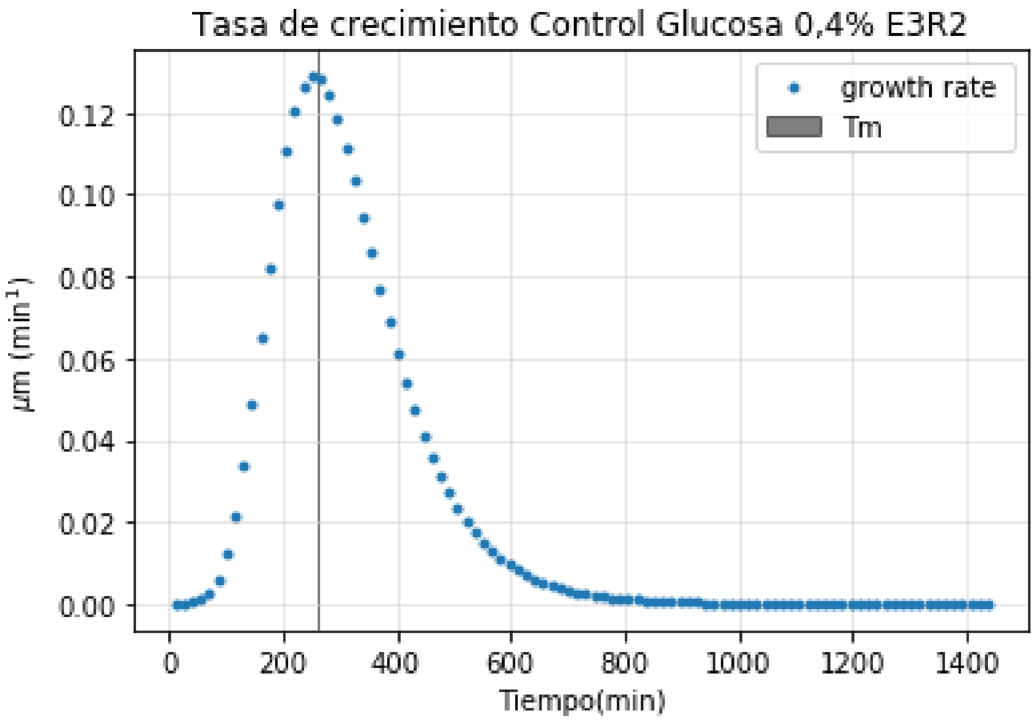


In [106]:
```
#tasa de crecimiento
ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(l7-tt))+1))))
#Con diff
dy7=(np.diff(ye7))
plt.figure()
plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm7,tm7, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy7,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

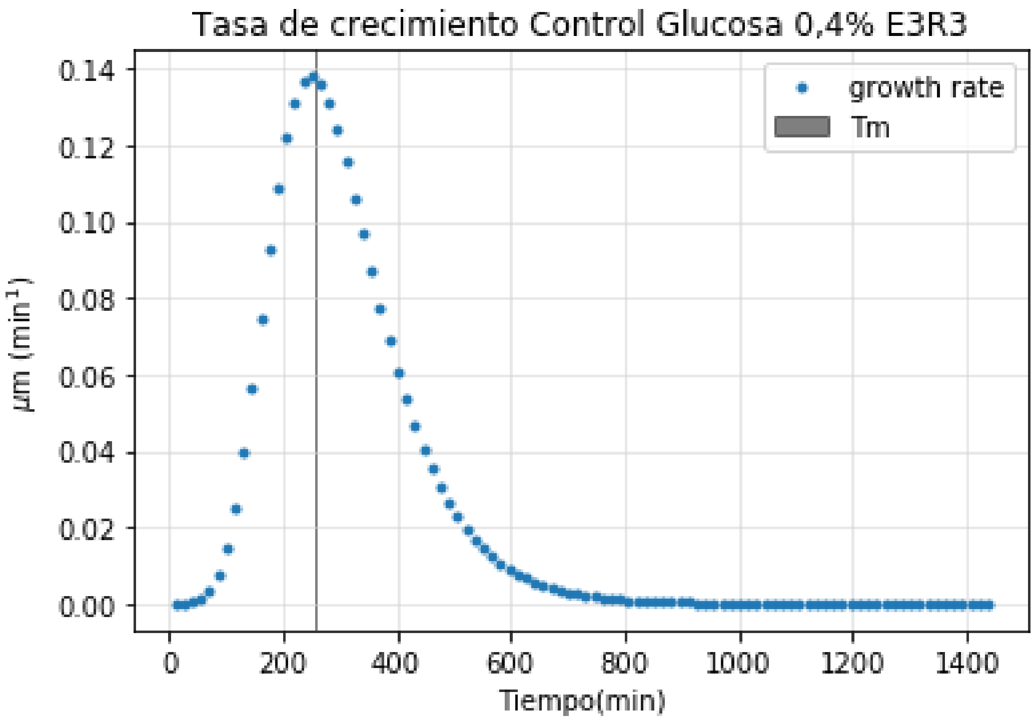
Out[106]:
```
<matplotlib.legend.Legend at 0x2036ba56b38>
```



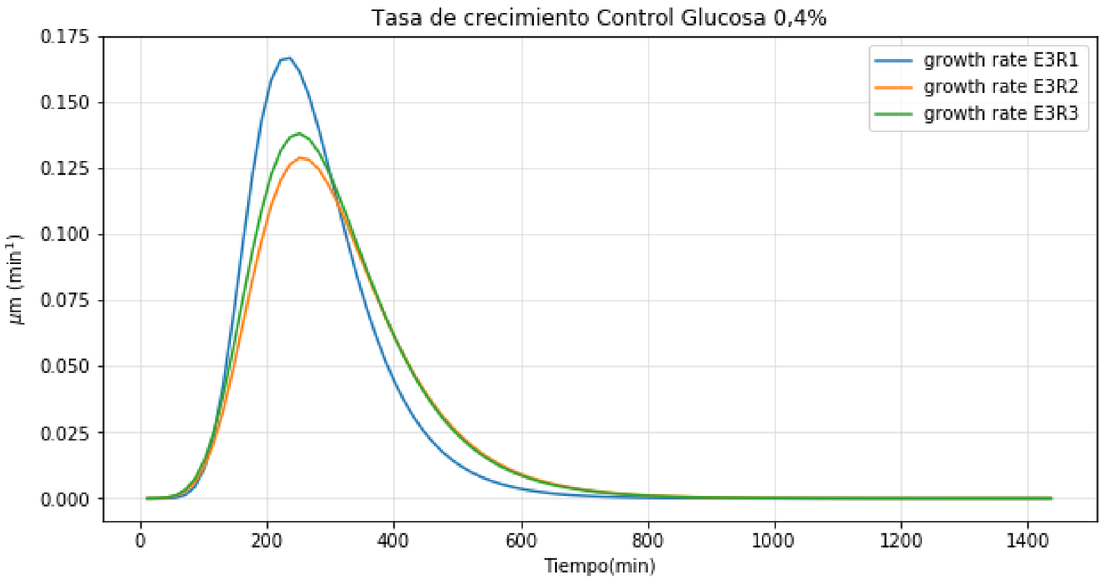

```
In [107]: #tasa de crecimiento

          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 Control Glucosa 0,4% E3R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm8,tm8, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy8,'.',label='growth rate ')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[107]: <matplotlib.legend.Legend at 0x2036bb28eb8>
```



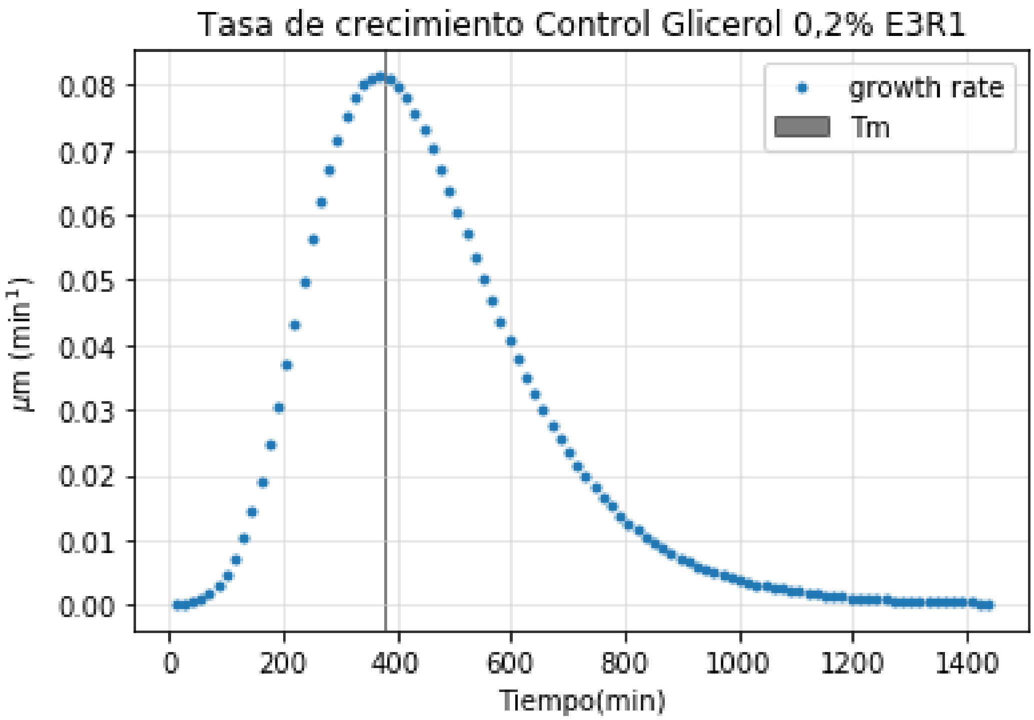

```
In [108]: #tasa de crecimiento
          ye9=((A9*np.exp(-np.exp((((um9*np.exp(1))/A9)*(l9-tt))+1))))
          #Con diff
          dy9=(np.diff(ye9))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glucosa 0,4% E3R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm9,tm9, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy9,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[108]: <matplotlib.legend.Legend at 0x2036a4516a0>
```



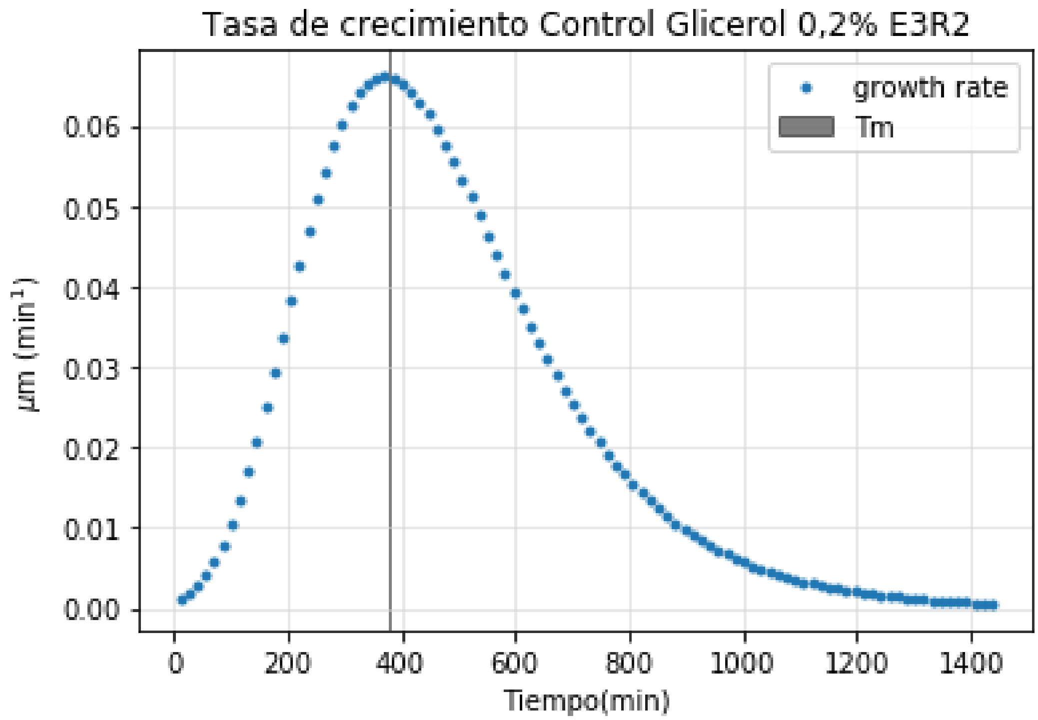

```
In [109]: #Tasas control réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='growth rate E3R1')
          plt.plot(tt[:-1],dy8,label='growth rate E3R2')
          plt.plot(tt[:-1],dy9,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[109]: <matplotlib.legend.Legend at 0x20369729a90>
```



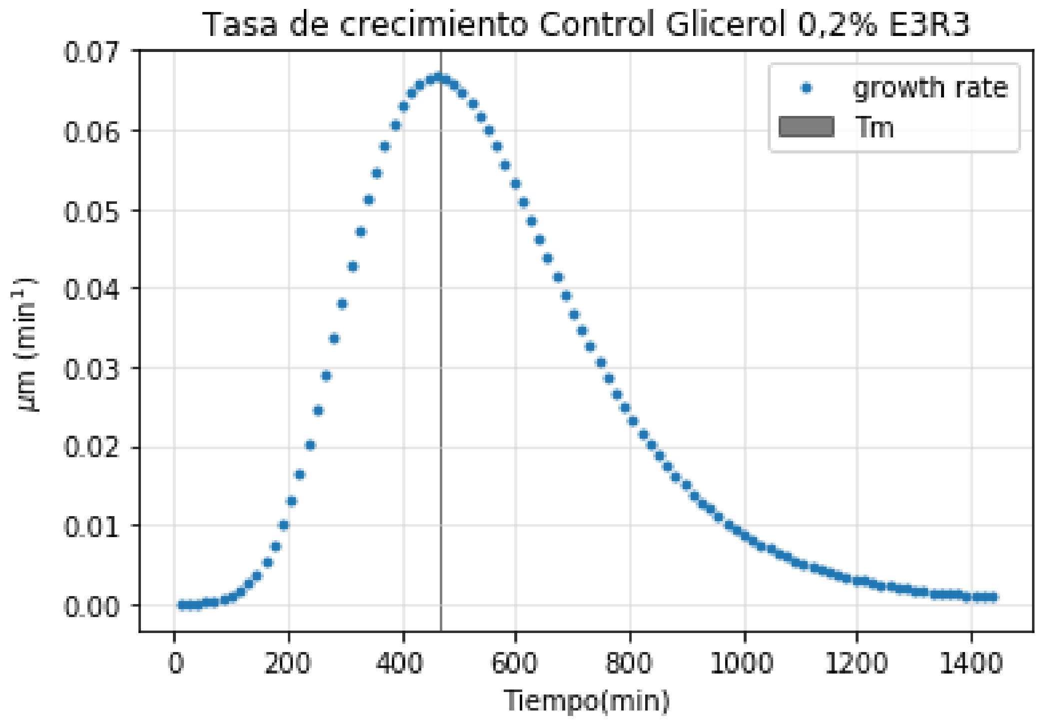

```
In [110]: #tasa de crecimiento
          ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(l10-tt))+1))))
          #Con diff
          dy10=(np.diff(ye10))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm10,tm10, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy10,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[110]: <matplotlib.legend.Legend at 0x2036884e518>
```



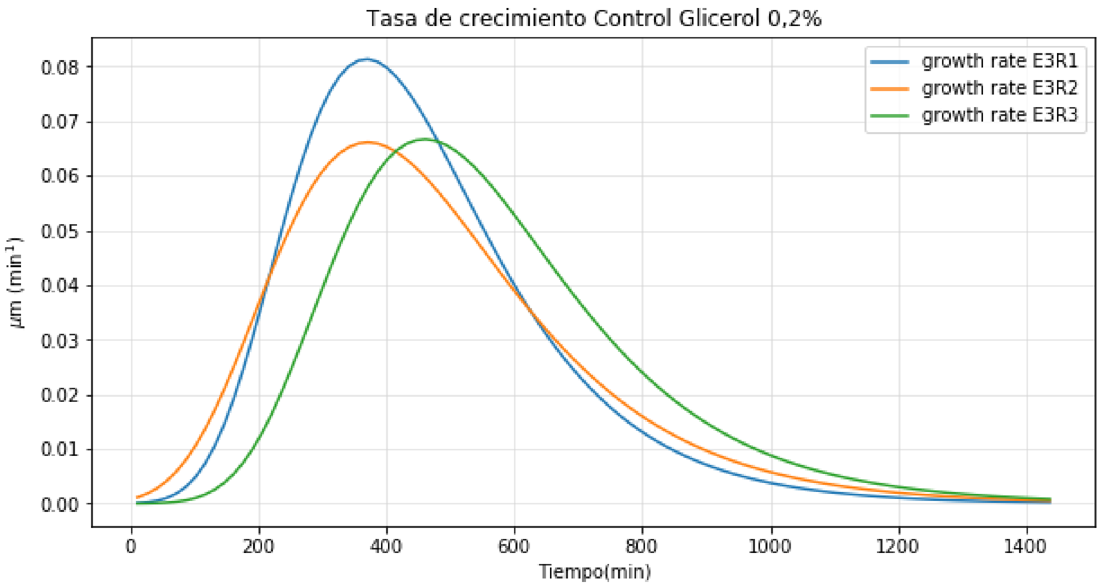

```
In [111]: #tasa de crecimiento
          ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(l11-tt))+1))))
          #Con diff
          dy11=(np.diff(ye11))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm11,tm11, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy11,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[111]: <matplotlib.legend.Legend at 0x20366548b00>
```



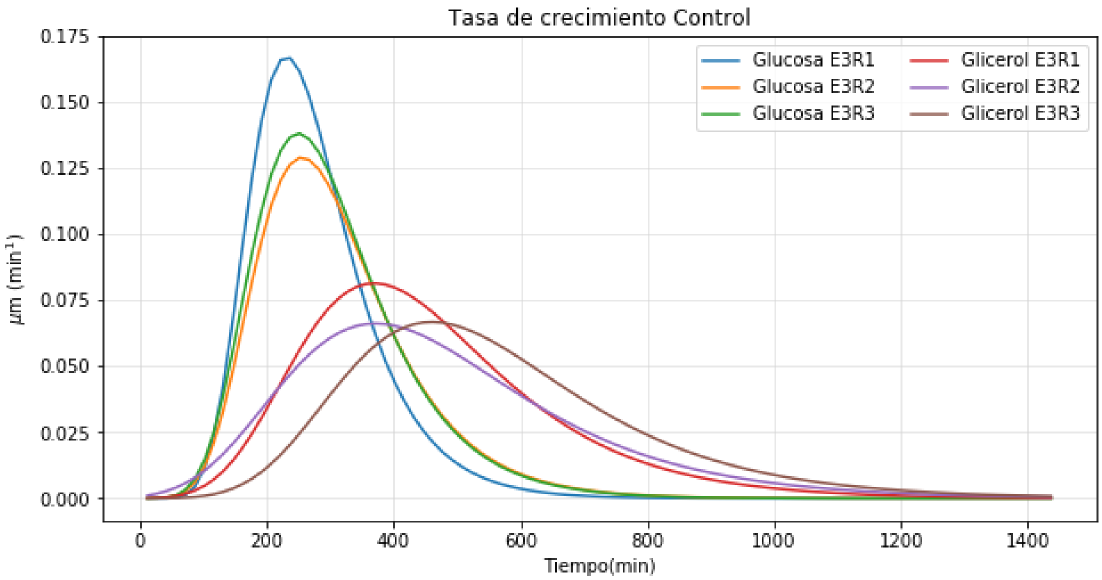

```
In [112]: #tasa de crecimiento
          ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(l12-tt))+1))))
          #Con diff
          dy12=(np.diff(ye12))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E3R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm12,tm12, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy12,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[112]: <matplotlib.legend.Legend at 0x20367a8af28>
```



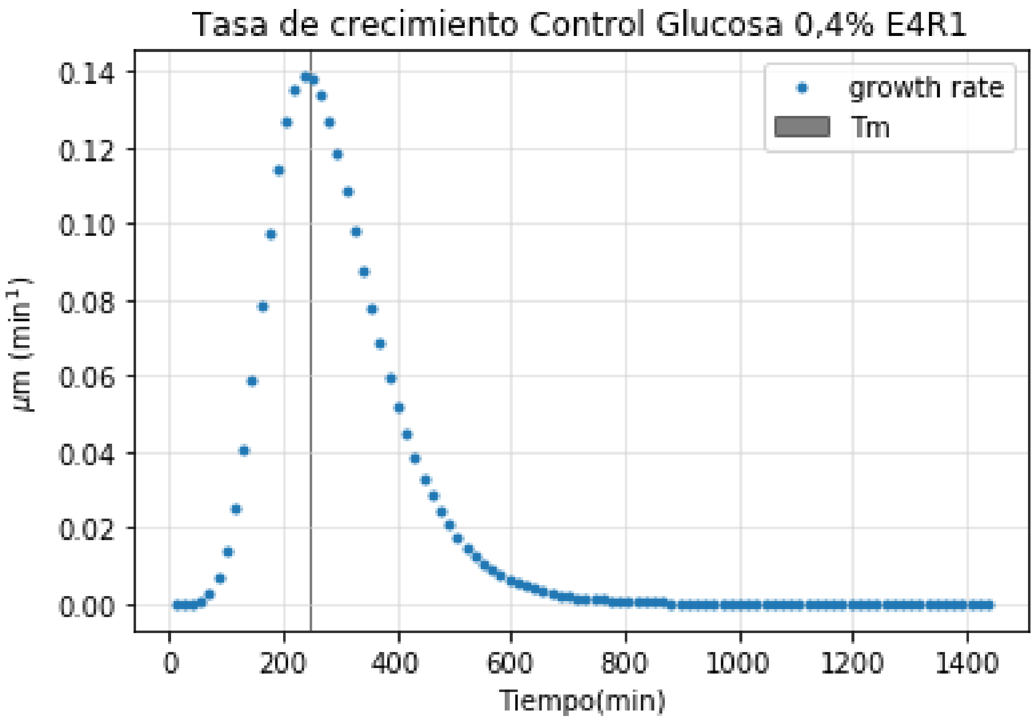

```
In [113]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy10,label='growth rate E3R1')
          plt.plot(tt[:-1],dy11,label='growth rate E3R2')
          plt.plot(tt[:-1],dy12,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[113]: <matplotlib.legend.Legend at 0x2036ba7f978>
```




```
In [114]: #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],dy7,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy8,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy9,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy10,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy11,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy12,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[114]: <matplotlib.legend.Legend at 0x20369775908>
```

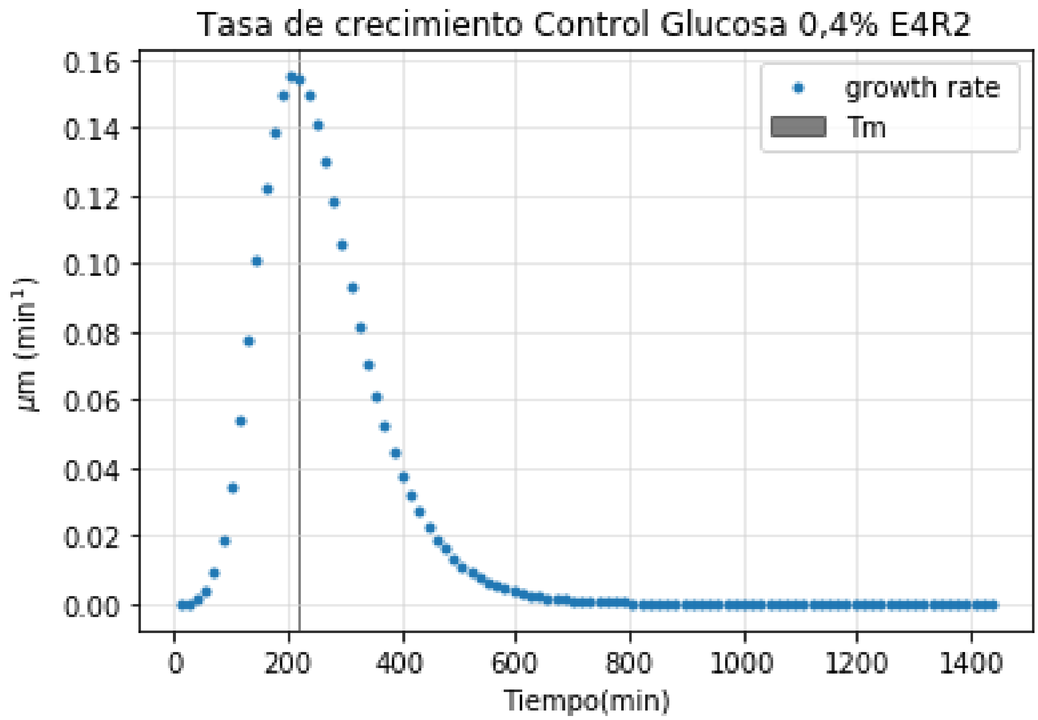



```
In [115]: #tasa de crecimiento
          ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(l13-tt))+1))))
          #Con diff
          dy13=(np.diff(ye13))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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 0x20369c95e80>
```

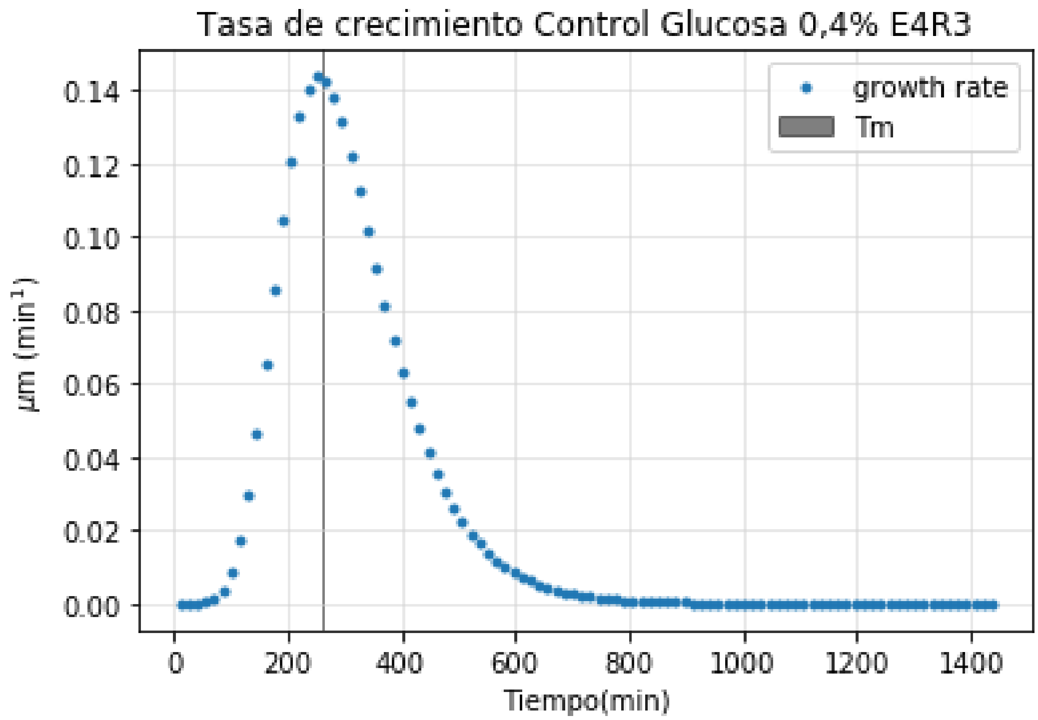


In [116]:
```
#tasa de crecimiento

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 Control Glucosa 0,4% E4R2')
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')
```

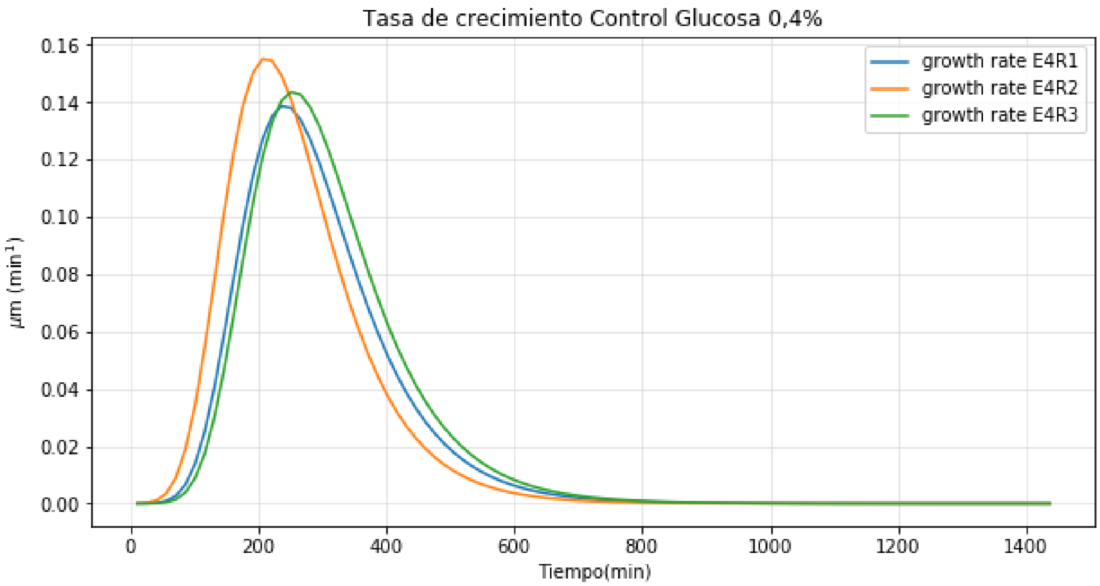
Out[116]: <matplotlib.legend.Legend at 0x20369d6e9e8>



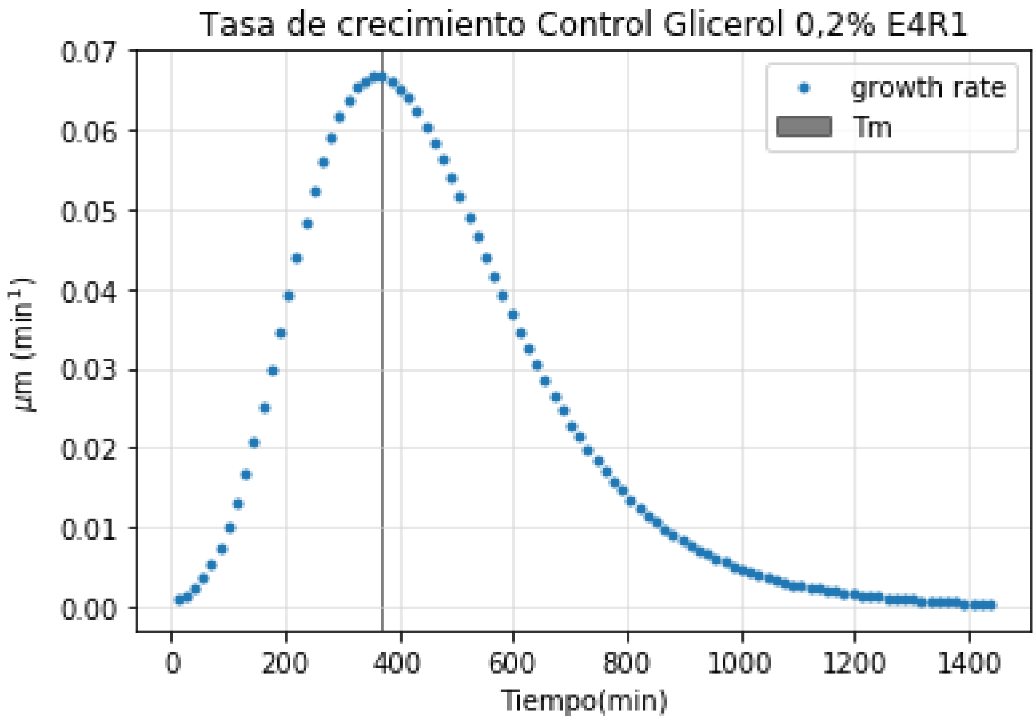

```
In [117]: #tasa de crecimiento
          ye15=((A15*np.exp(-np.exp((((um15*np.exp(1))/A15)*(l15-tt))+1))))
          #Con diff
          dy15=(np.diff(ye15))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (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 0x20369e27f60>
```



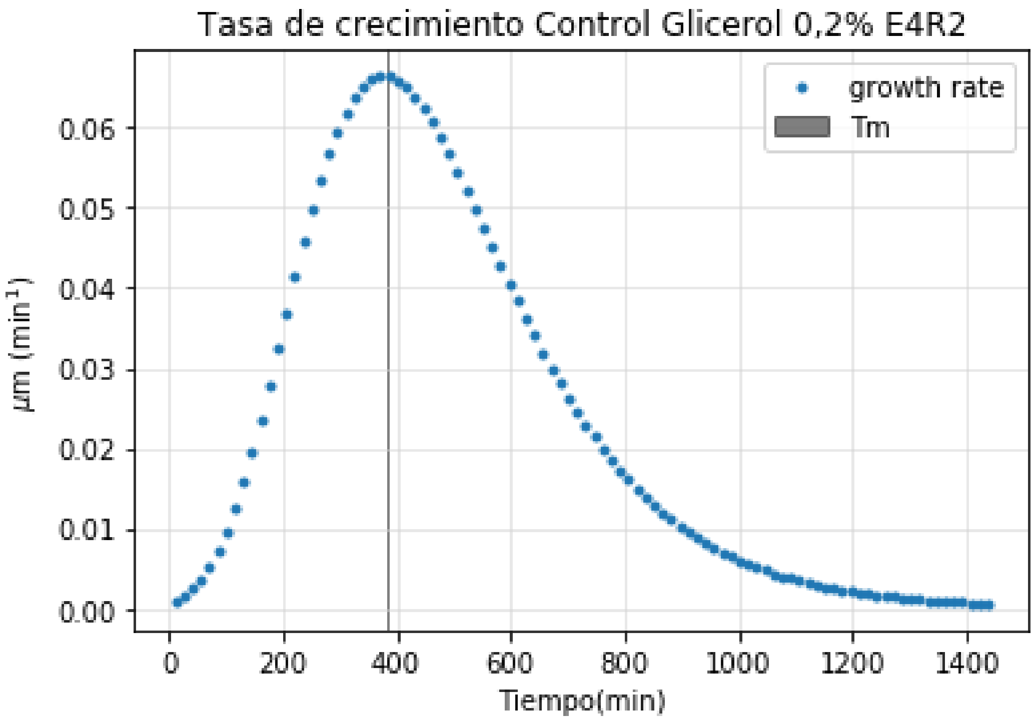

```
In [118]: #Tasas control réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='growth rate E4R1')
          plt.plot(tt[:-1],dy14,label='growth rate E4R2')
          plt.plot(tt[:-1],dy15,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[118]: <matplotlib.legend.Legend at 0x20369ef9b38>
```



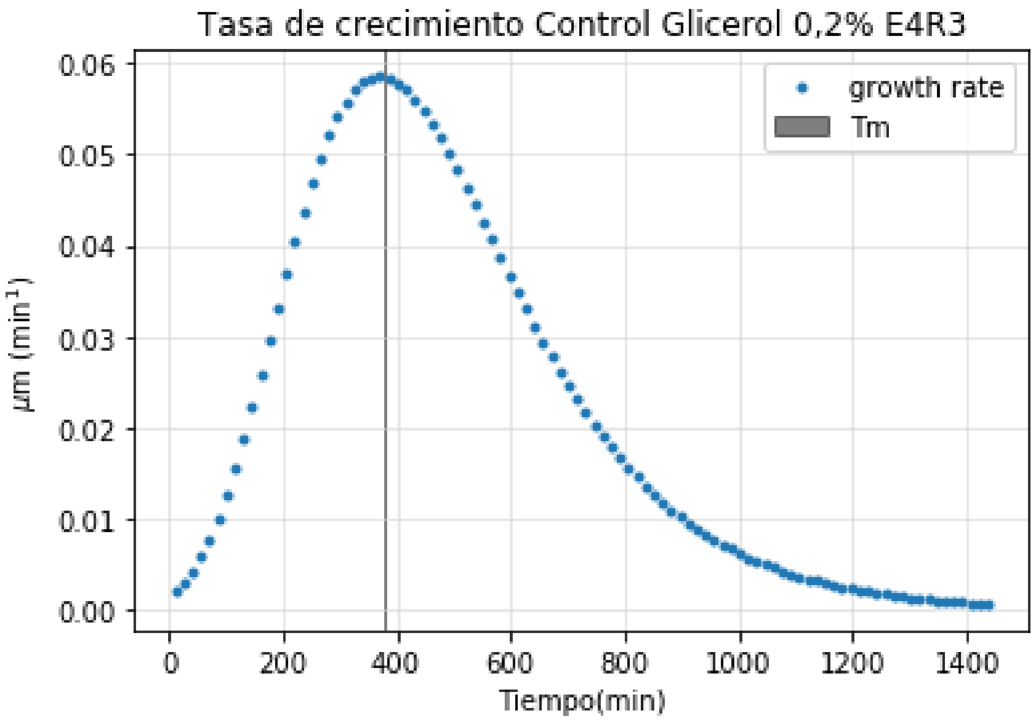

```
In [119]: #tasa de crecimiento
          ye16=((A16*np.exp(-np.exp((((um16*np.exp(1))/A16)*(l16-tt))+1))))
          #Con diff
          dy16=(np.diff(ye16))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          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 0x20369fc8fd0>
```



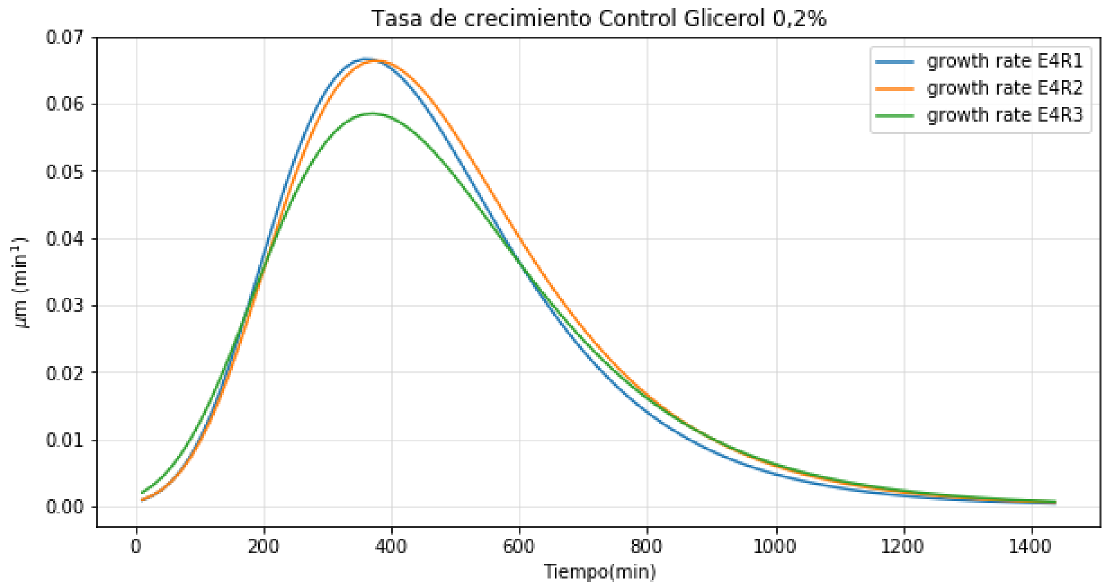

```
In [120]: #tasa de crecimiento
          ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(l17-tt))+1))))
          #Con diff
          dy17=(np.diff(ye17))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R2')
          plt.xlabel('Tiempo(min)')
          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 0x2036cbdfcf8>
```



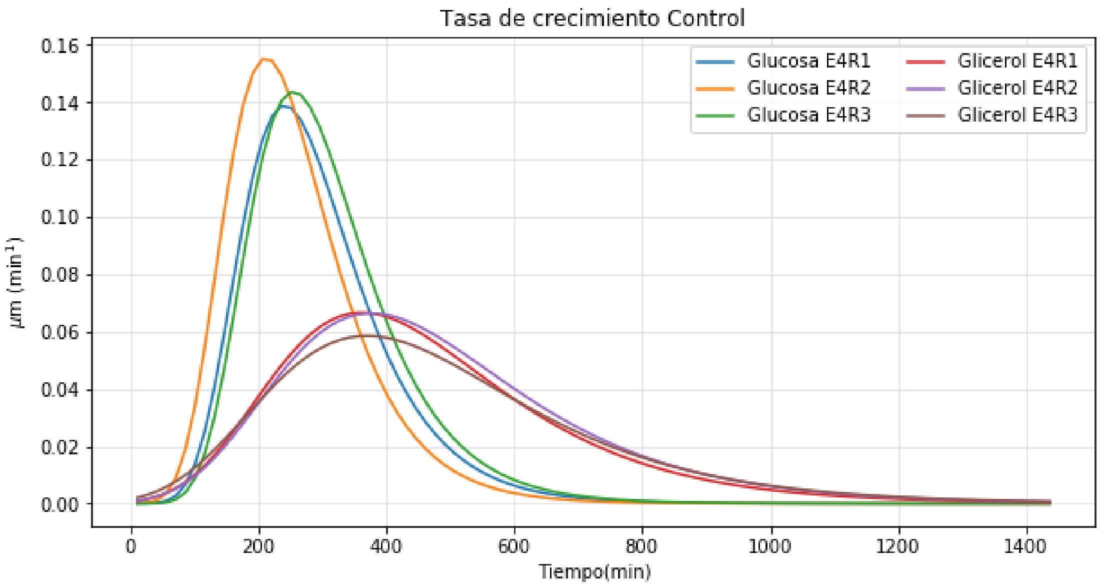

```
In [121]: #tasa de crecimiento
          ye18=((A18*np.exp(-np.exp((((um18*np.exp(1))/A18)*(l18-tt))+1))))
          #Con diff
          dy18=(np.diff(ye18))
          plt.figure()
          plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          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 0x2036ccb2b70>
```




```
In [122]: #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],dy16,label='growth rate E4R1')
          plt.plot(tt[:-1],dy17,label='growth rate E4R2')
          plt.plot(tt[:-1],dy18,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[122]: <matplotlib.legend.Legend at 0x2036cd7dbe0>
```

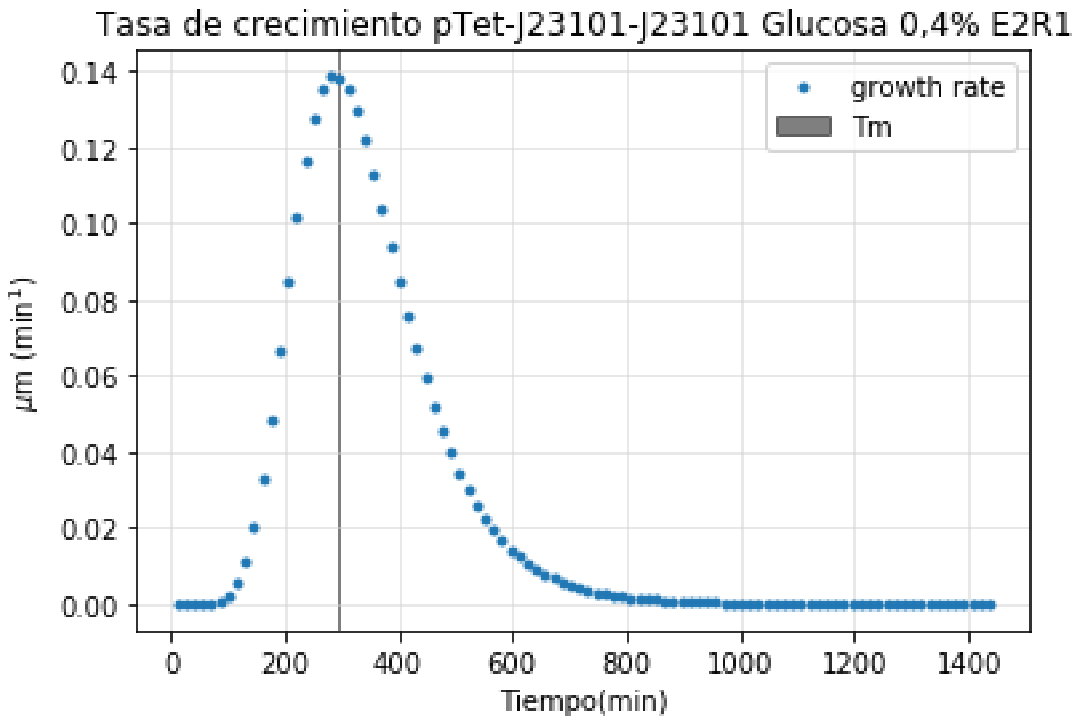



```
In [123]: #Tasas control réplicas controles
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento Control')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy14,label='Glucosa E4R2')
          plt.plot(tt[:-1],dy15,label='Glucosa E4R3')
          plt.plot(tt[:-1],dy16,label='Glicerol E4R1')
          plt.plot(tt[:-1],dy17,label='Glicerol E4R2')
          plt.plot(tt[:-1],dy18,label='Glicerol E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[123]: <matplotlib.legend.Legend at 0x2036ce6acf8>
```

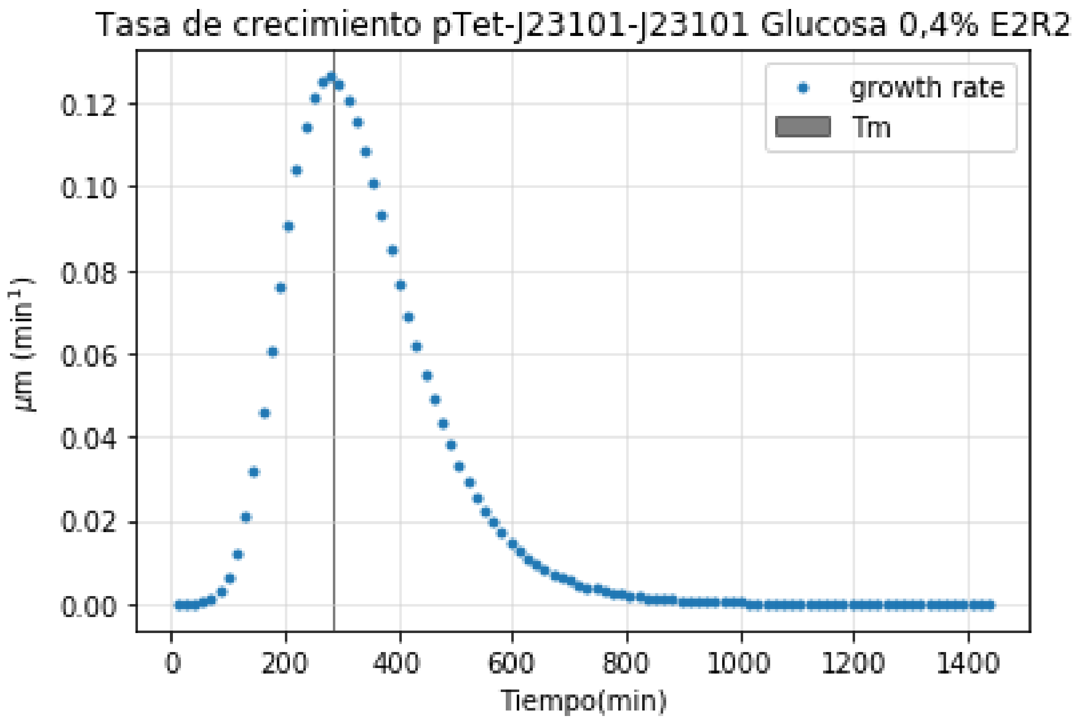


In [124]:
```
#tasa de crecimiento
ye19=((A19*np.exp(-np.exp((((um19*np.exp(1))/A19)*(l19-tt))+1))))
#Con diff
dy19=(np.diff(ye19))
plt.figure()
plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm19,tm19, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy19,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

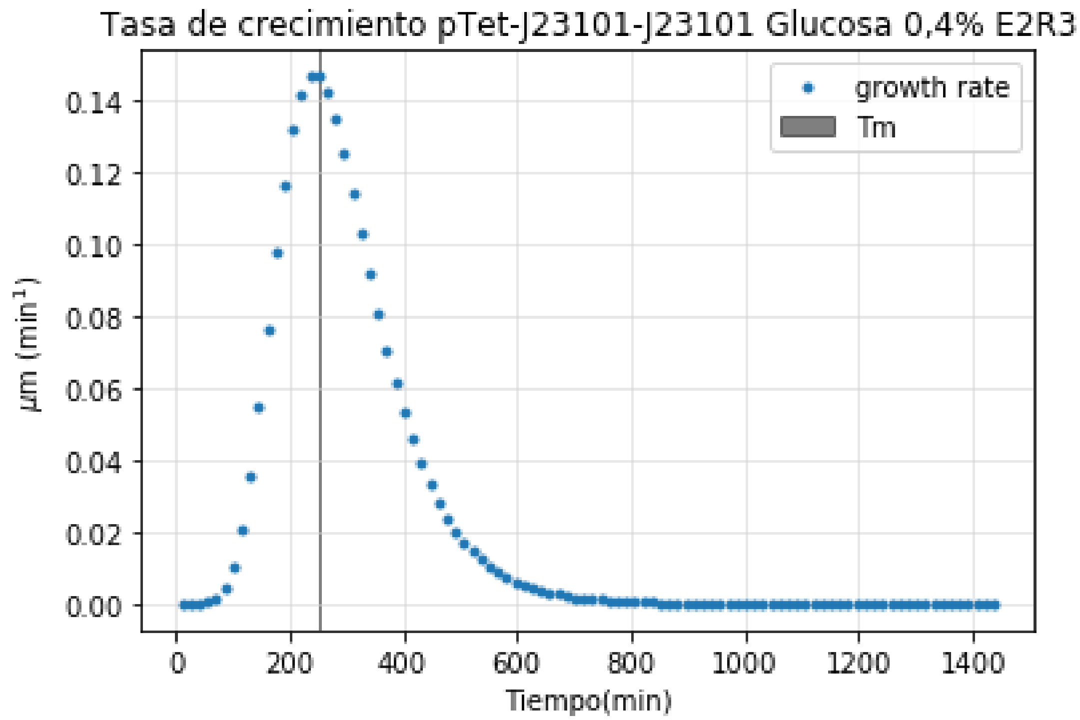
Out[124]:
```
<matplotlib.legend.Legend at 0x2036cf50dd8>
```



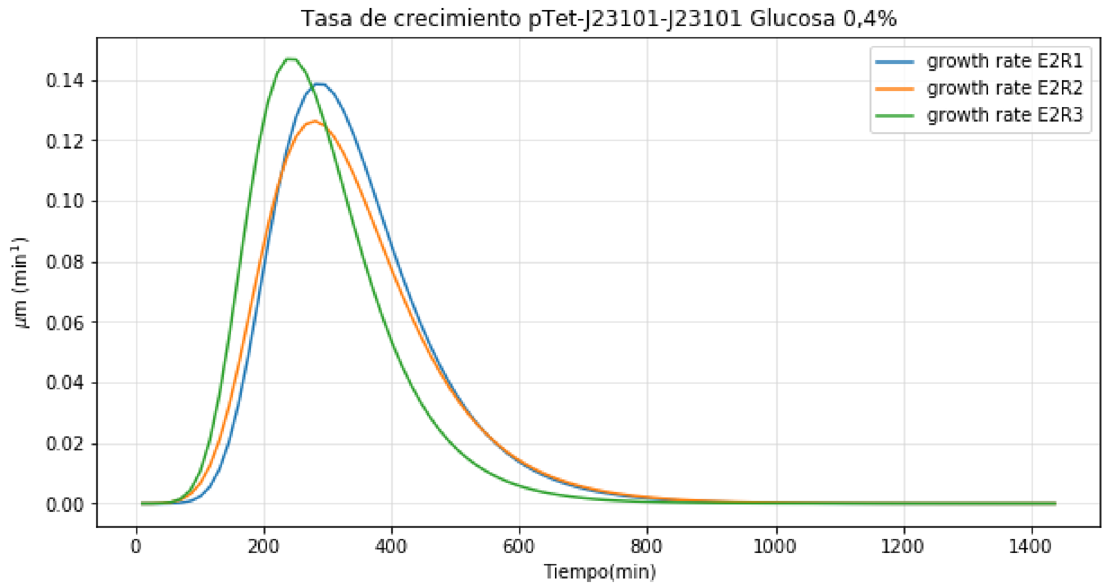

```
In [125]: #tasa de crecimiento
          ye20=((A20*np.exp(-np.exp((((um20*np.exp(1))/A20)*(l20-tt))+1))))
          #Con diff
          dy20=(np.diff(ye20))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm20,tm20, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy20,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[125]: <matplotlib.legend.Legend at 0x2036d02ba90>
```




```
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 pTet-J23101-J23101 Glucosa 0,4% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm21,tm21, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy21,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[126]: <matplotlib.legend.Legend at 0x2036d100fd0>
```

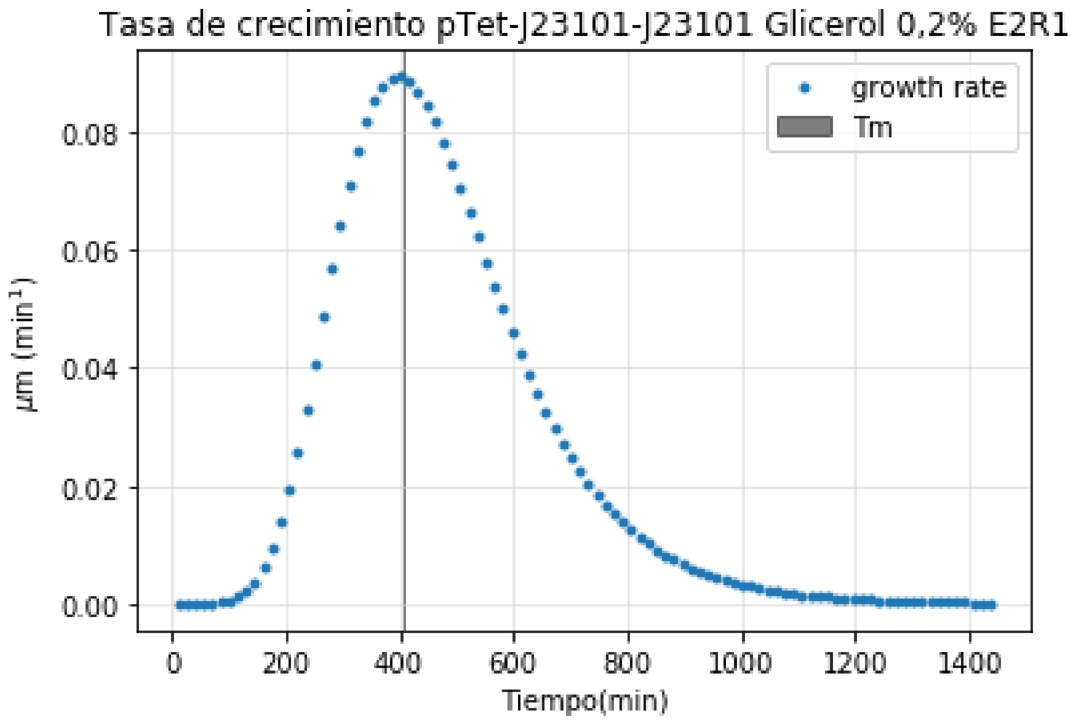



```
In [127]: #Tasas pLux-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='growth rate E2R1')
          plt.plot(tt[:-1],dy20,label='growth rate E2R2')
          plt.plot(tt[:-1],dy21,label='growth rate E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[127]: <matplotlib.legend.Legend at 0x2036d1d7f98>
```

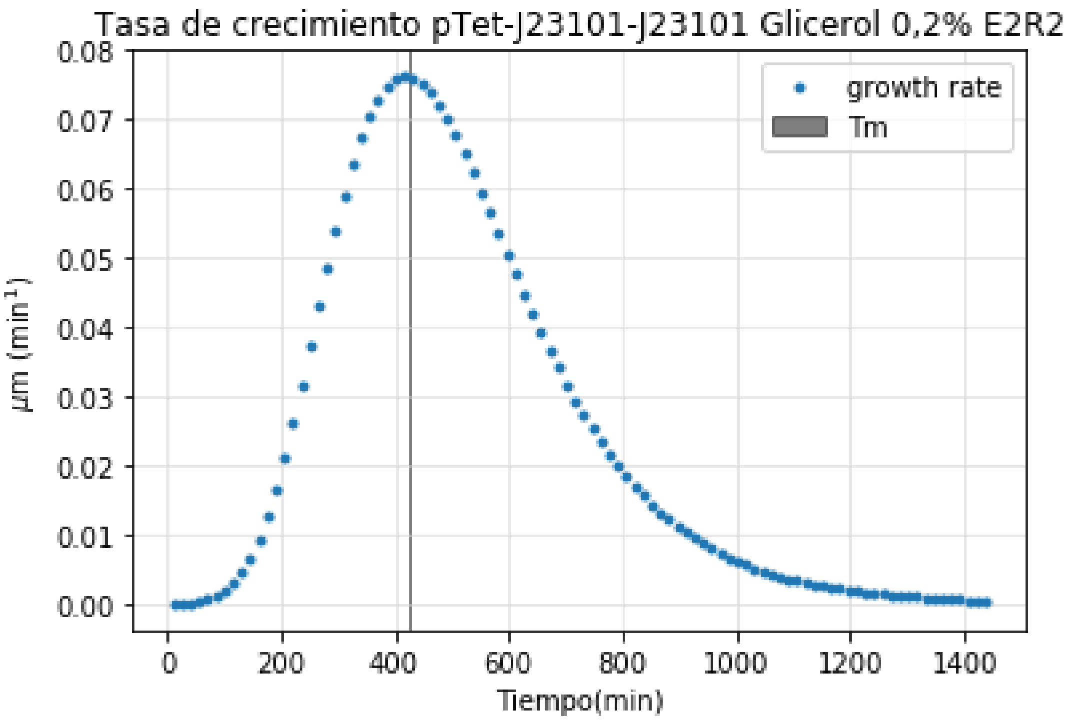


In [128]:
```
#tasa de crecimiento
ye22=((A22*np.exp(-np.exp((((um22*np.exp(1))/A22)*(l22-tt))+1))))
#Con diff
dy22=(np.diff(ye22))
plt.figure()
plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2% E2R1')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm22,tm22, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy22,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

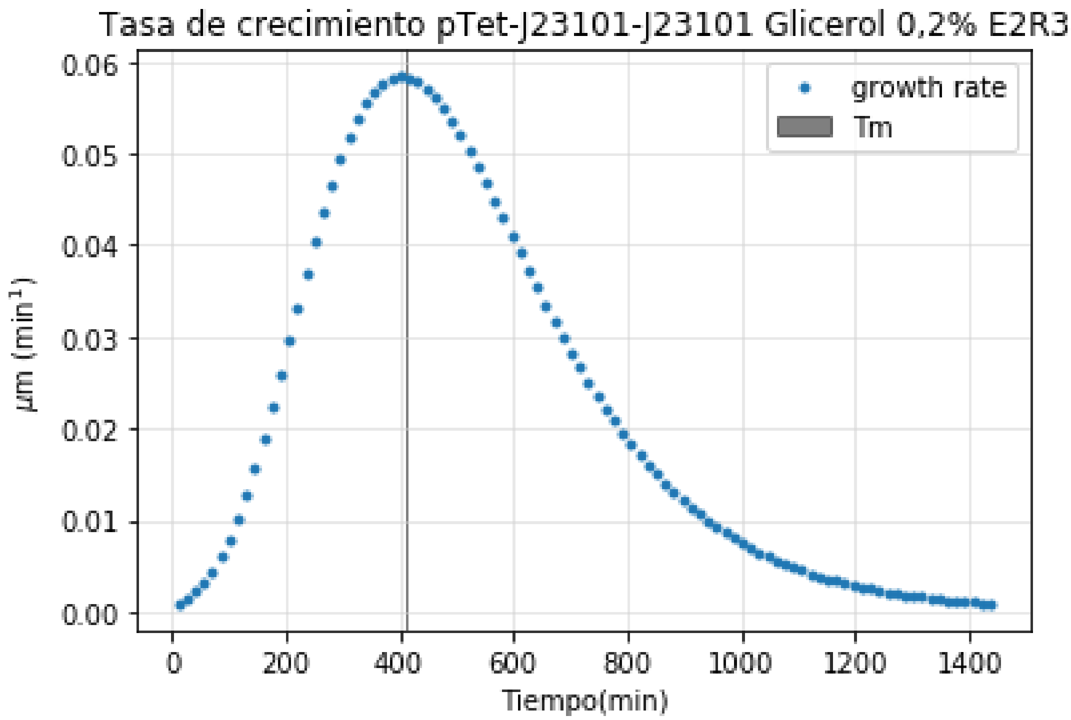
Out[128]:
```
<matplotlib.legend.Legend at 0x2036e28ed68>
```




```
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 pTet-J23101-J23101 Glicerol 0,2% E2R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm23,tm23, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy23,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[129]: <matplotlib.legend.Legend at 0x2036e35ca20>
```

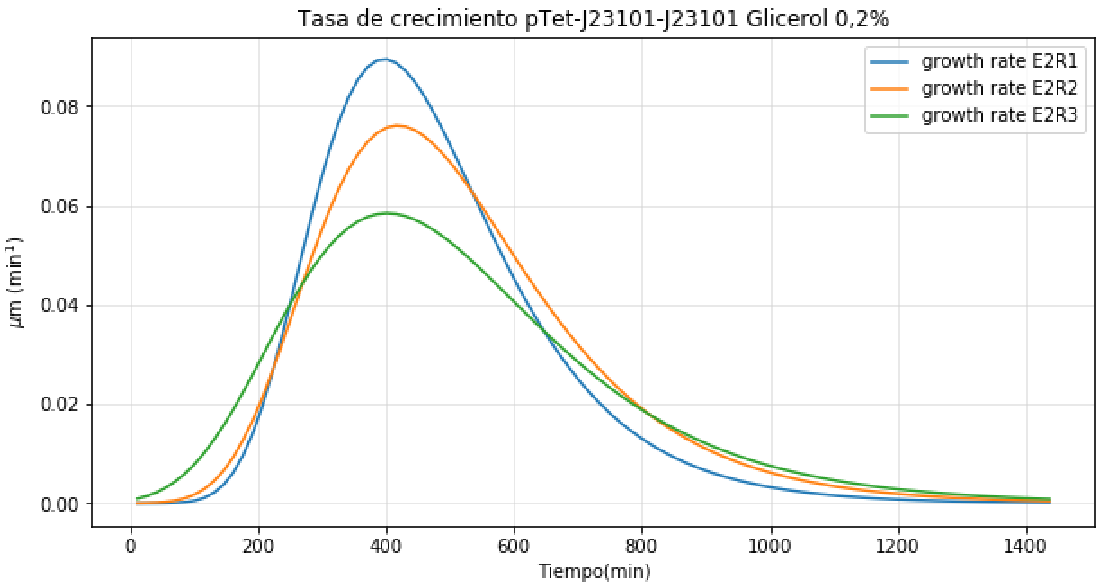



```
In [130]: #tasa de crecimiento
          ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(l24-tt))+1))))
          #Con diff
          dy24=(np.diff(ye24))
          plt.figure()
          plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2% E2R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm24,tm24, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy24,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[130]: <matplotlib.legend.Legend at 0x2036e437c88>
```

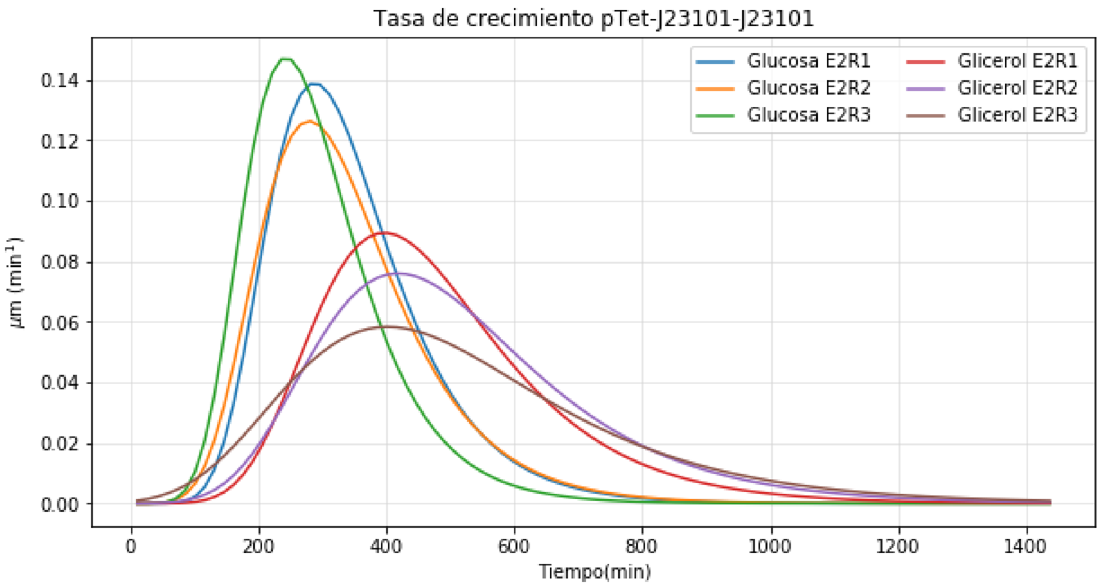


In [131]:
```
#Tasas control réplicas glicerol
plt.figure(figsize=(10,5))
plt.title('Tasa de crecimiento pTet-J23101-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy22,label='growth rate E2R1')
plt.plot(tt[:-1],dy23,label='growth rate E2R2')
plt.plot(tt[:-1],dy24,label='growth rate E2R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

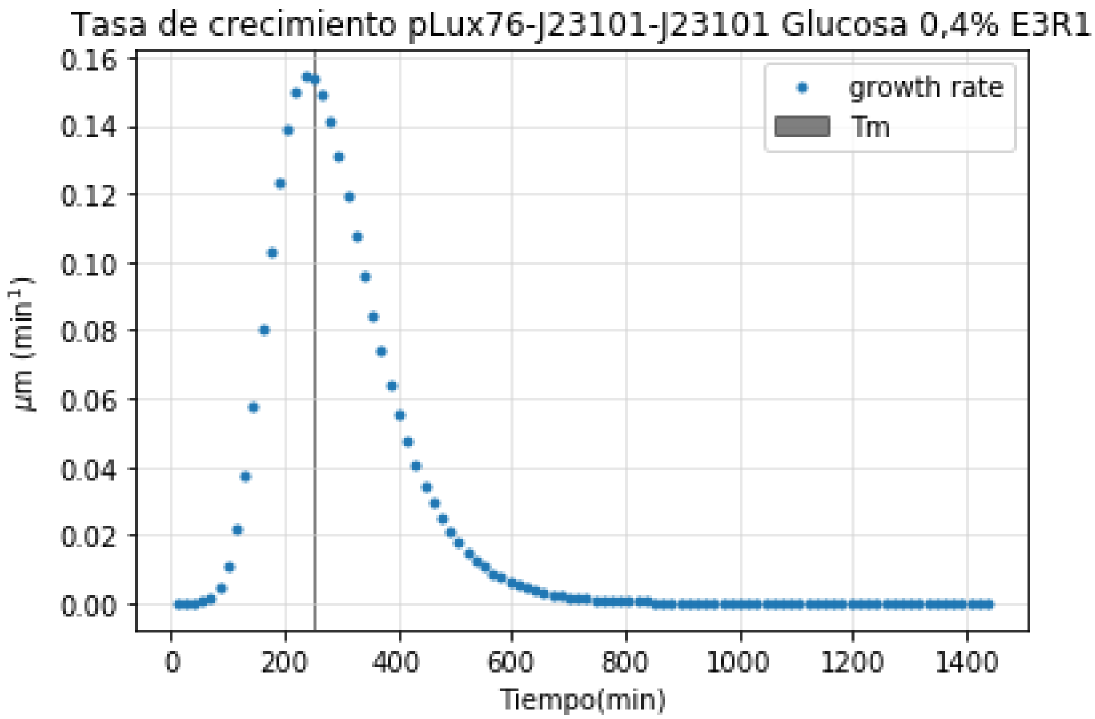
Out[131]:
```
<matplotlib.legend.Legend at 0x2036e501c88>
```



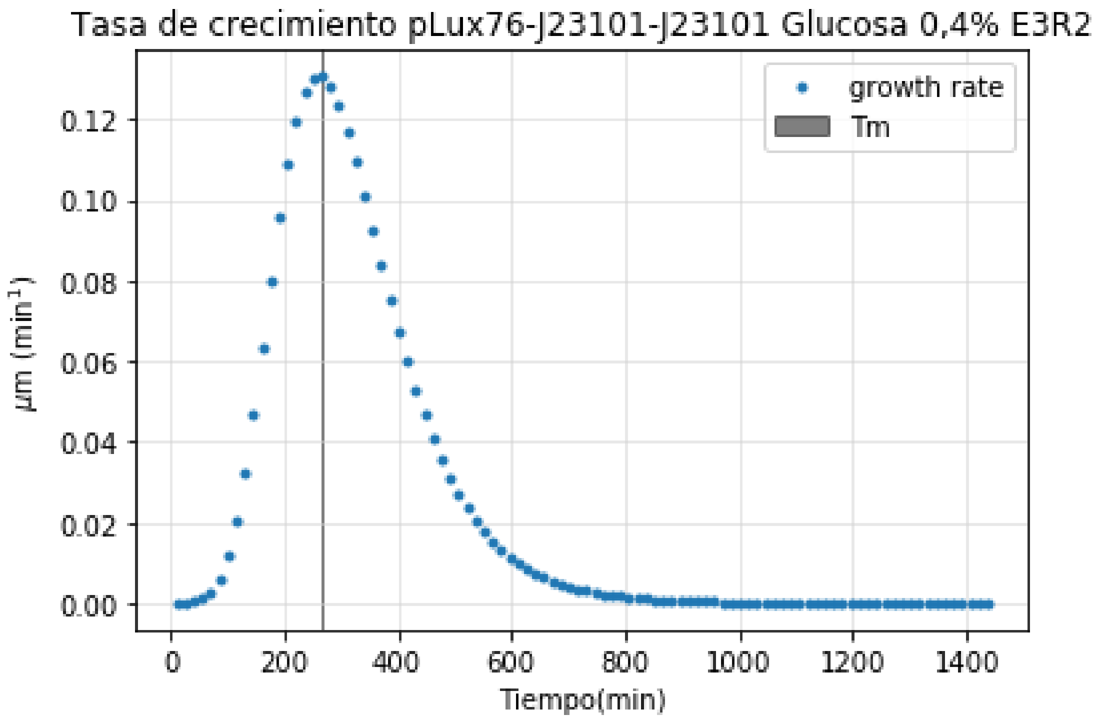

```
In [132]: #Tasas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pTet-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='Glucosa E2R1')
          plt.plot(tt[:-1],dy20,label='Glucosa E2R2')
          plt.plot(tt[:-1],dy21,label='Glucosa E2R3')
          plt.plot(tt[:-1],dy22,label='Glicerol E2R1')
          plt.plot(tt[:-1],dy23,label='Glicerol E2R2')
          plt.plot(tt[:-1],dy24,label='Glicerol E2R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[132]: <matplotlib.legend.Legend at 0x2036cf6bfd0>
```



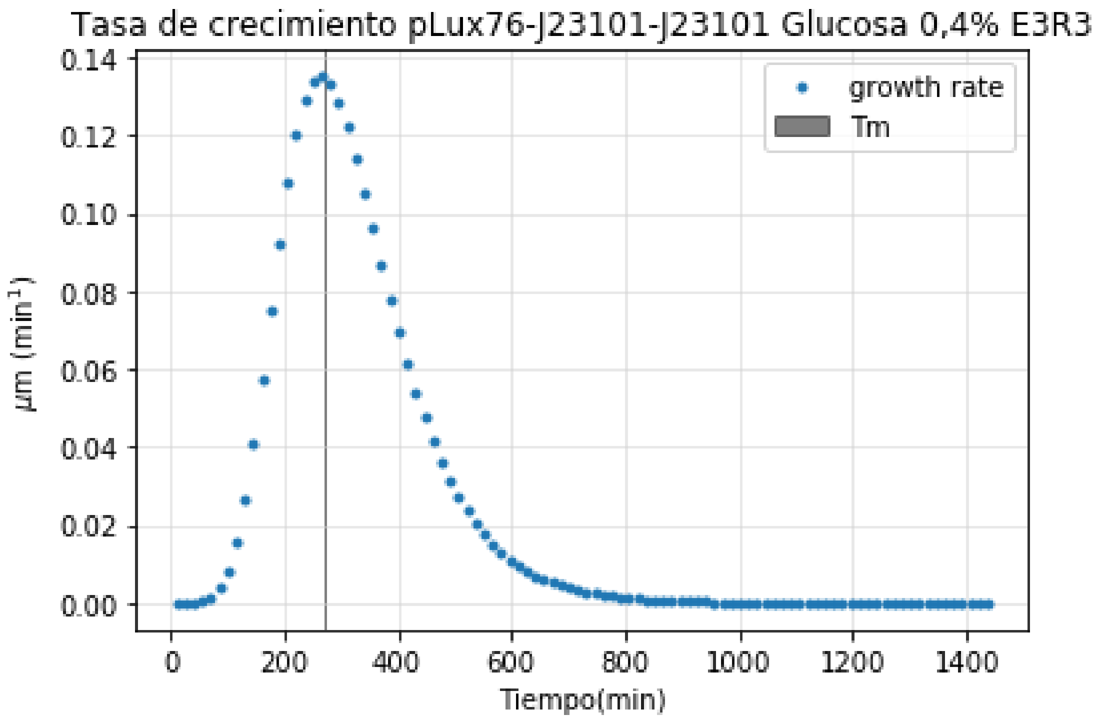

```
In [133]: #tasa de crecimiento
          ye25=((A25*np.exp(-np.exp((((um25*np.exp(1))/A25)*(l25-tt))+1))))
          #Con diff
          dy25=(np.diff(ye25))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E3R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm25,tm25, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy25,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[133]: <matplotlib.legend.Legend at 0x203691dc048>
```



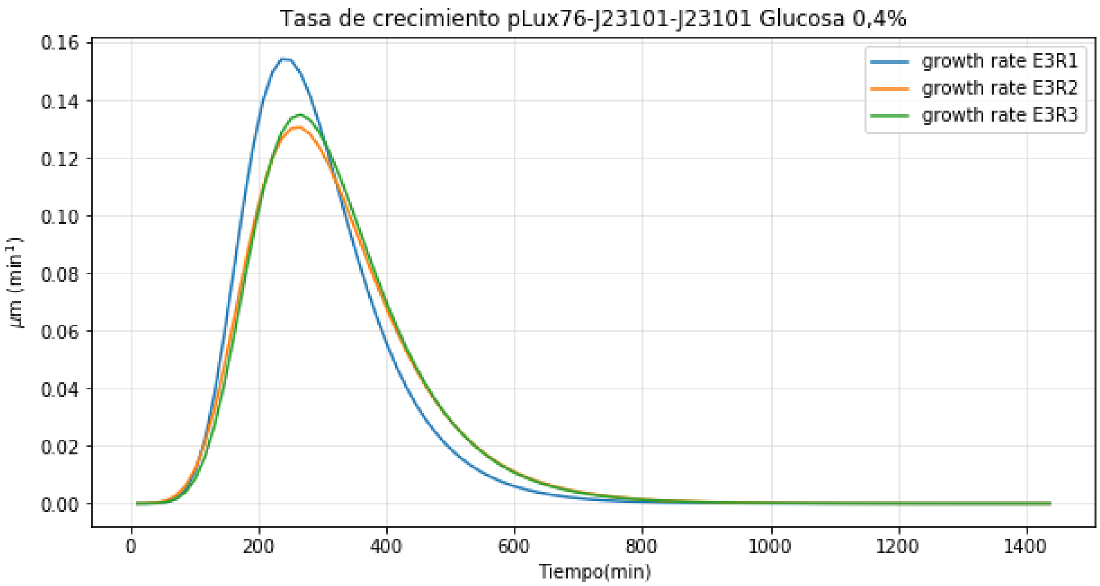

```
In [134]: #tasa de crecimiento
          ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(l26-tt))+1))))
          #Con diff
          dy26=(np.diff(ye26))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E3R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm26,tm26, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy26,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[134]: <matplotlib.legend.Legend at 0x2036e5bb828>
```



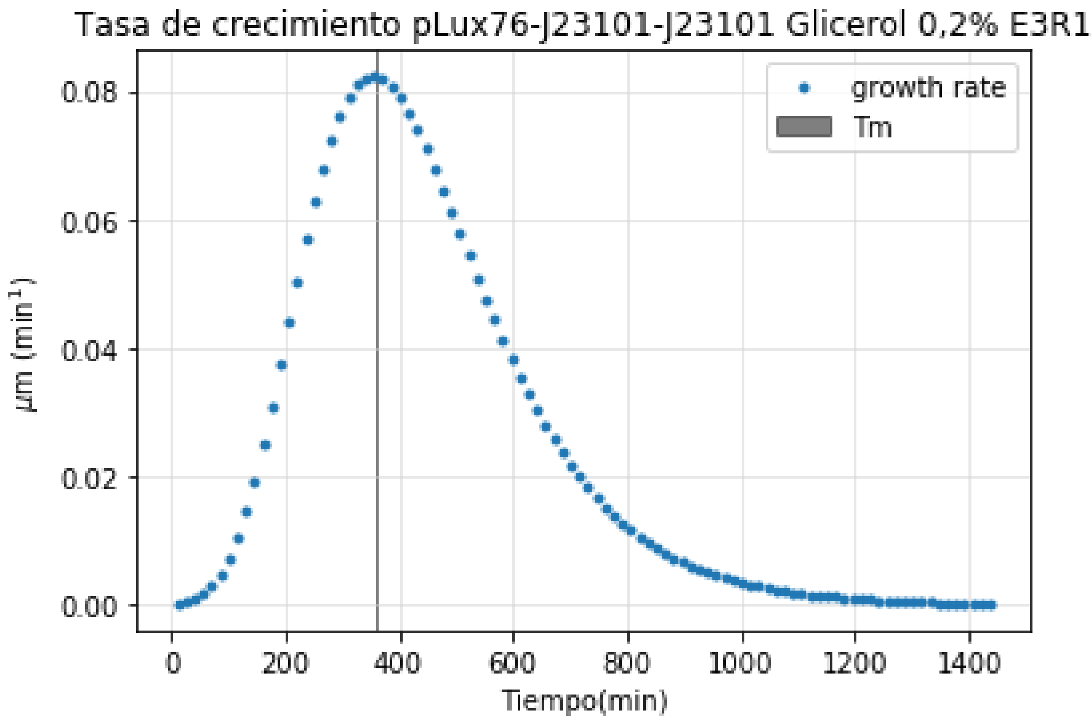

```
In [135]: #tasa de crecimiento
          ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(l27-tt))+1))))
          #Con diff
          dy27=(np.diff(ye27))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4% E3R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm27,tm27, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy27,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[135]: <matplotlib.legend.Legend at 0x2036e65deb8>
```




```
In [136]: #Tasas pLux-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='growth rate E3R1')
          plt.plot(tt[:-1],dy26,label='growth rate E3R2')
          plt.plot(tt[:-1],dy27,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[136]: <matplotlib.legend.Legend at 0x2036e71ef98>
```

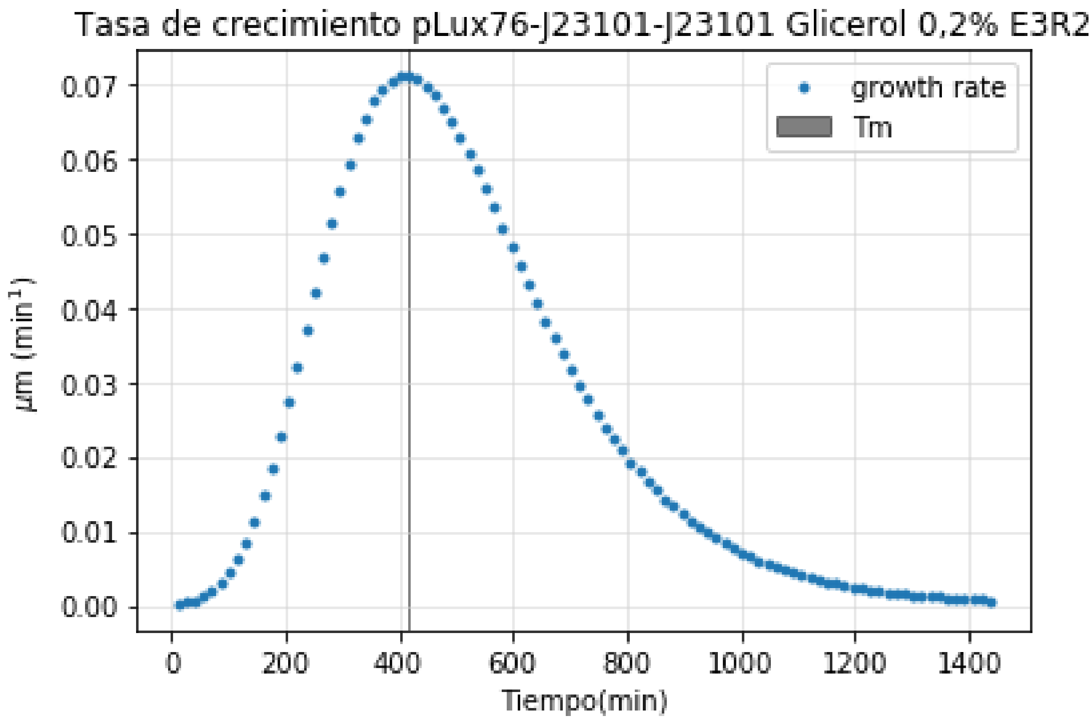



```
In [137]: #tasa de crecimiento
          ye28=((A28*np.exp(-np.exp((((um28*np.exp(1))/A28)*(l28-tt))+1))))
          #Con diff
          dy28=(np.diff(ye28))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm28,tm28, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy28,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[137]: <matplotlib.legend.Legend at 0x2036e7dfc88>
```

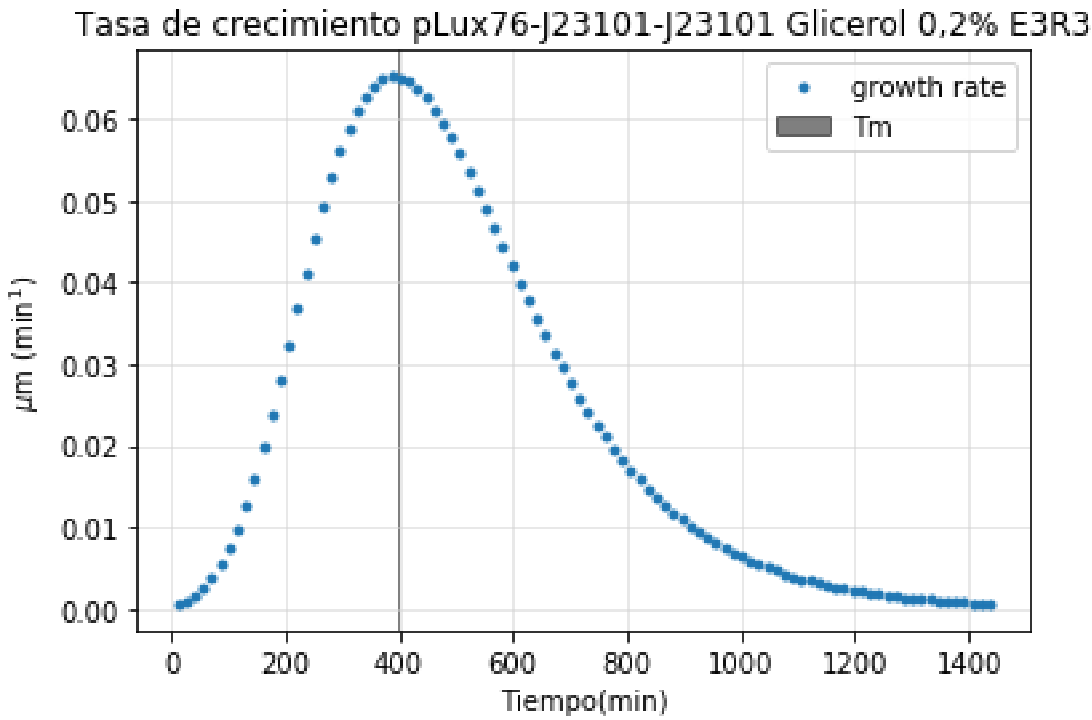


In [138]:
```python
#tasa de crecimiento
ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(l29-tt))+1))))
#Con diff
dy29=(np.diff(ye29))
plt.figure()
plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm29,tm29, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy29,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

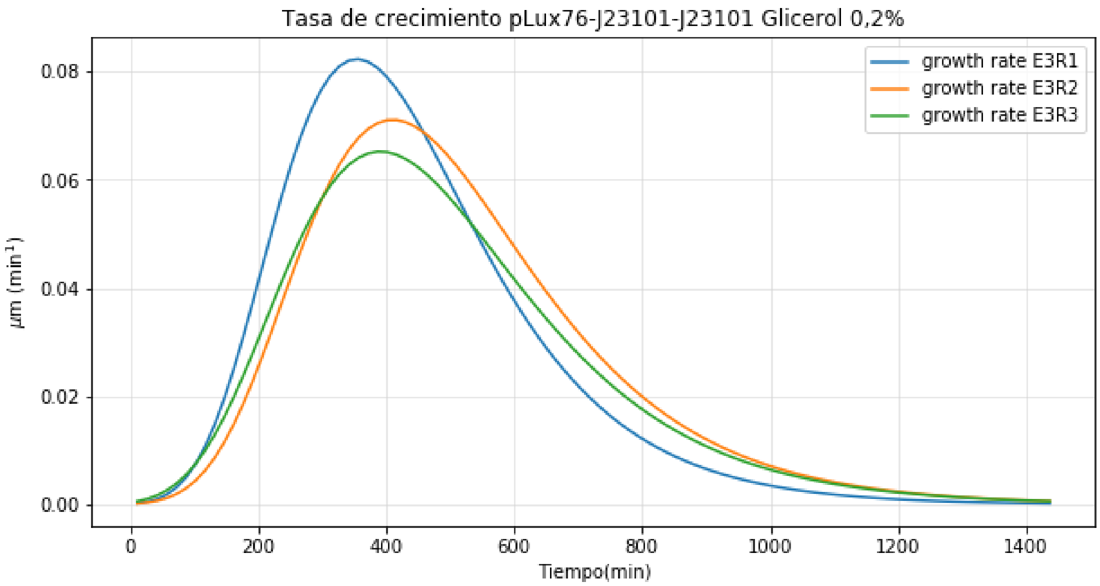
Out[138]: <matplotlib.legend.Legend at 0x2036e8a7908>



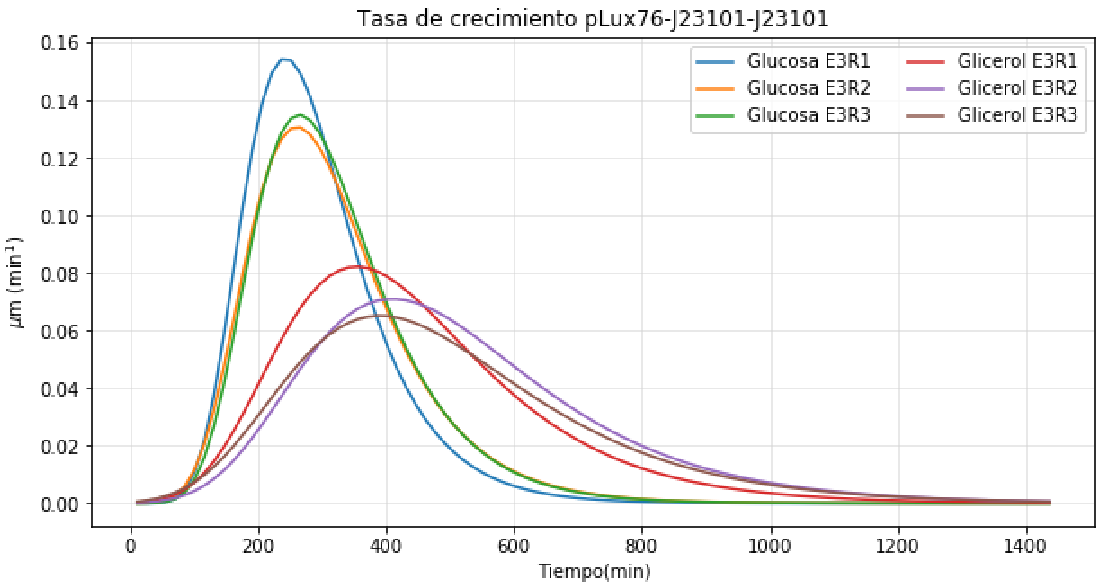

```
In [139]: #tasa de crecimiento
          ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(l30-tt))+1))))
          #Con diff
          dy30=(np.diff(ye30))
          plt.figure()
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2% E3R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm30,tm30, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy30,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[139]: <matplotlib.legend.Legend at 0x2036e97cac8>
```



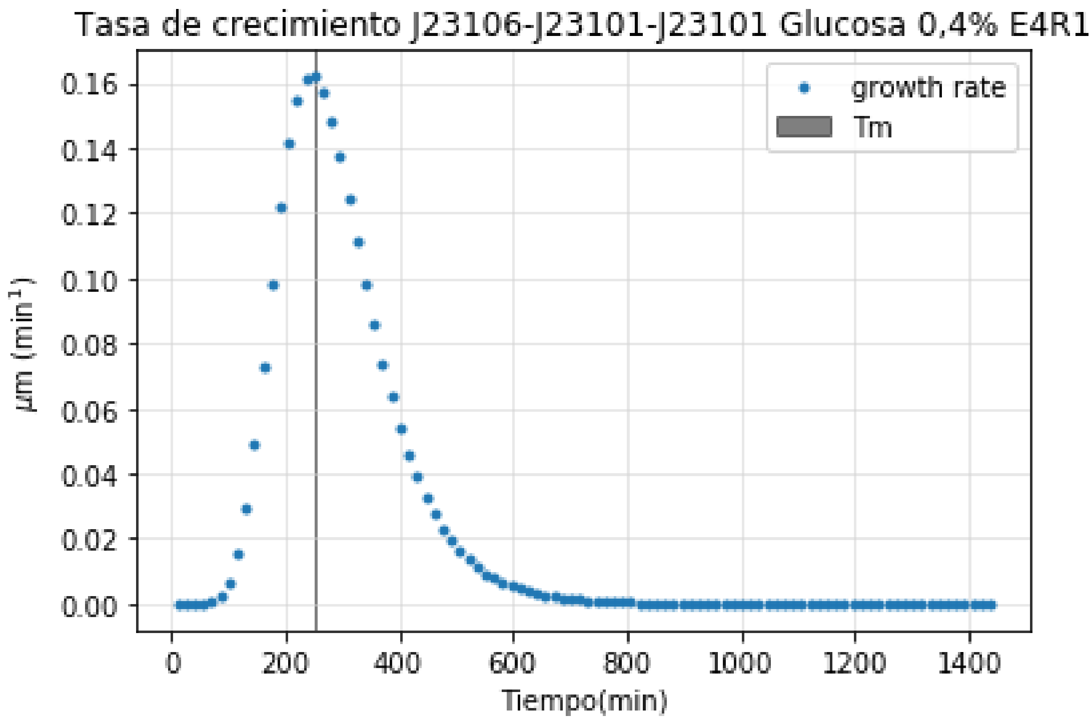

```
In [140]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy28,label='growth rate E3R1')
          plt.plot(tt[:-1],dy29,label='growth rate E3R2')
          plt.plot(tt[:-1],dy30,label='growth rate E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[140]: <matplotlib.legend.Legend at 0x2036ea45860>
```



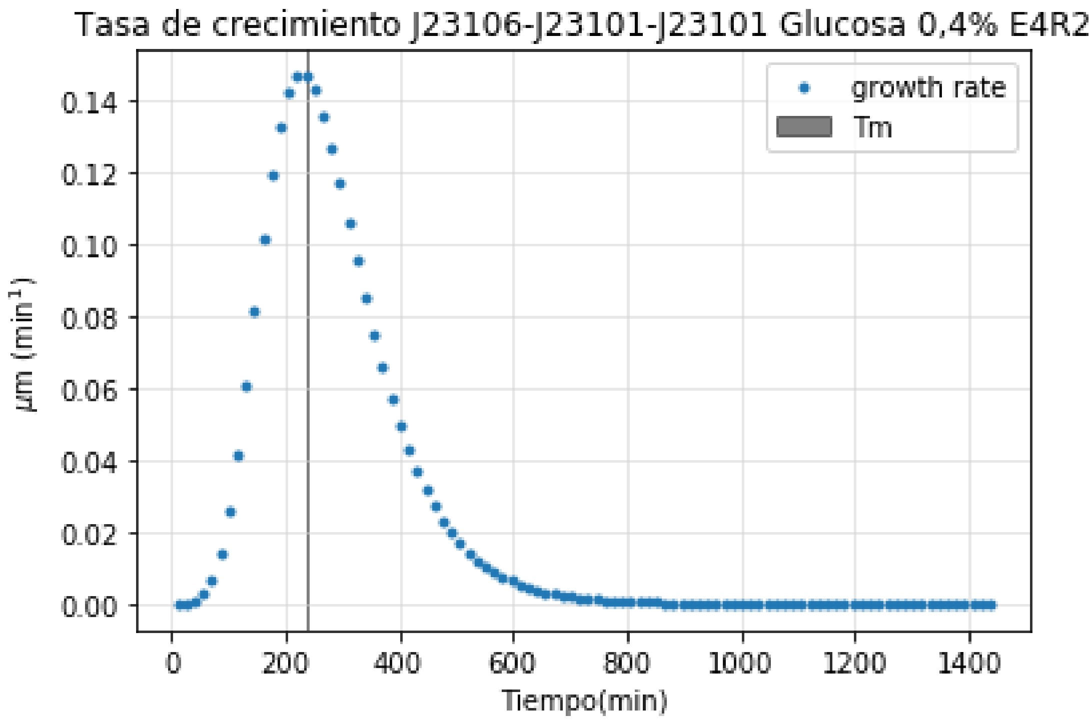

```
In [141]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento pLux76-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='Glucosa E3R1')
          plt.plot(tt[:-1],dy26,label='Glucosa E3R2')
          plt.plot(tt[:-1],dy27,label='Glucosa E3R3')
          plt.plot(tt[:-1],dy28,label='Glicerol E3R1')
          plt.plot(tt[:-1],dy29,label='Glicerol E3R2')
          plt.plot(tt[:-1],dy30,label='Glicerol E3R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[141]: <matplotlib.legend.Legend at 0x2036eb29ef0>
```



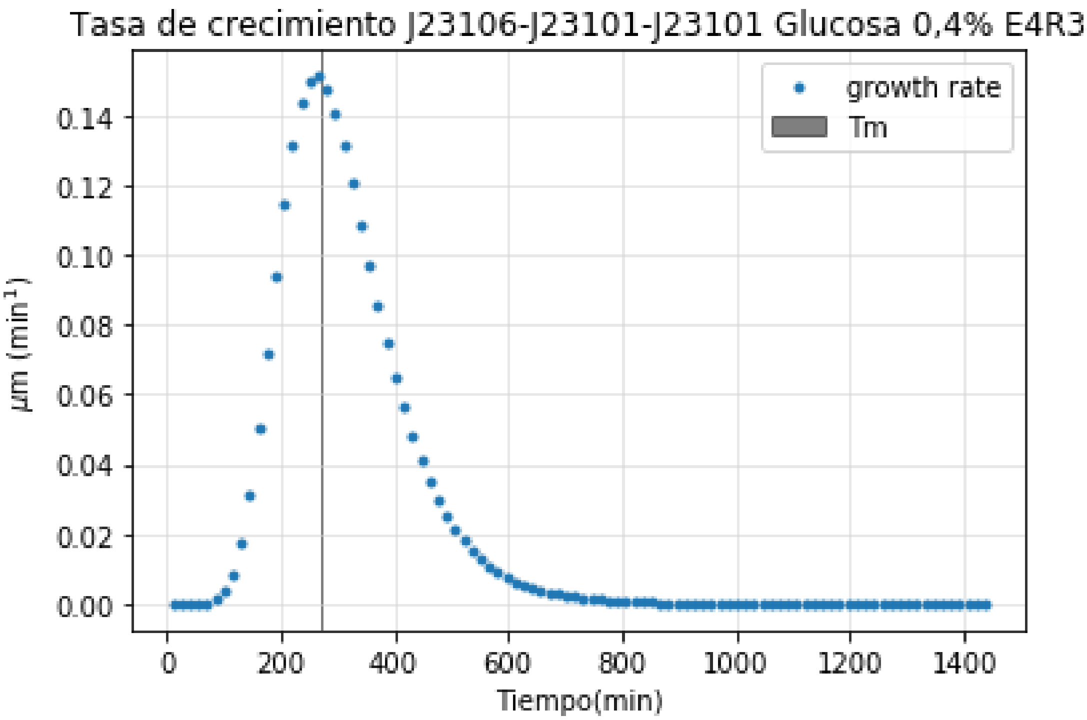

```
In [142]: #tasa de crecimiento
          ye31=((A31*np.exp(-np.exp((((um31*np.exp(1))/A31)*(l31-tt))+1))))
          #Con diff
          dy31=(np.diff(ye31))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm31,tm31, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy31,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[142]: <matplotlib.legend.Legend at 0x2036ec09f98>
```



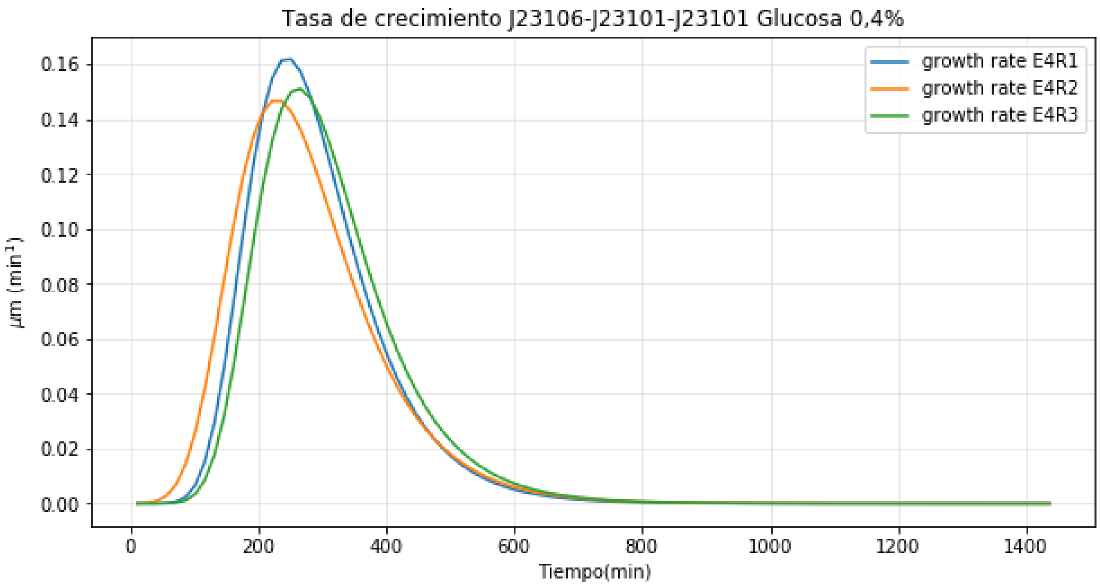

```
In [143]: #tasa de crecimiento
          ye32=((A32*np.exp(-np.exp((((um32*np.exp(1))/A32)*(l32-tt))+1))))
          #Con diff
          dy32=(np.diff(ye32))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm32,tm32, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy32,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[143]: <matplotlib.legend.Legend at 0x2036ece5b00>
```



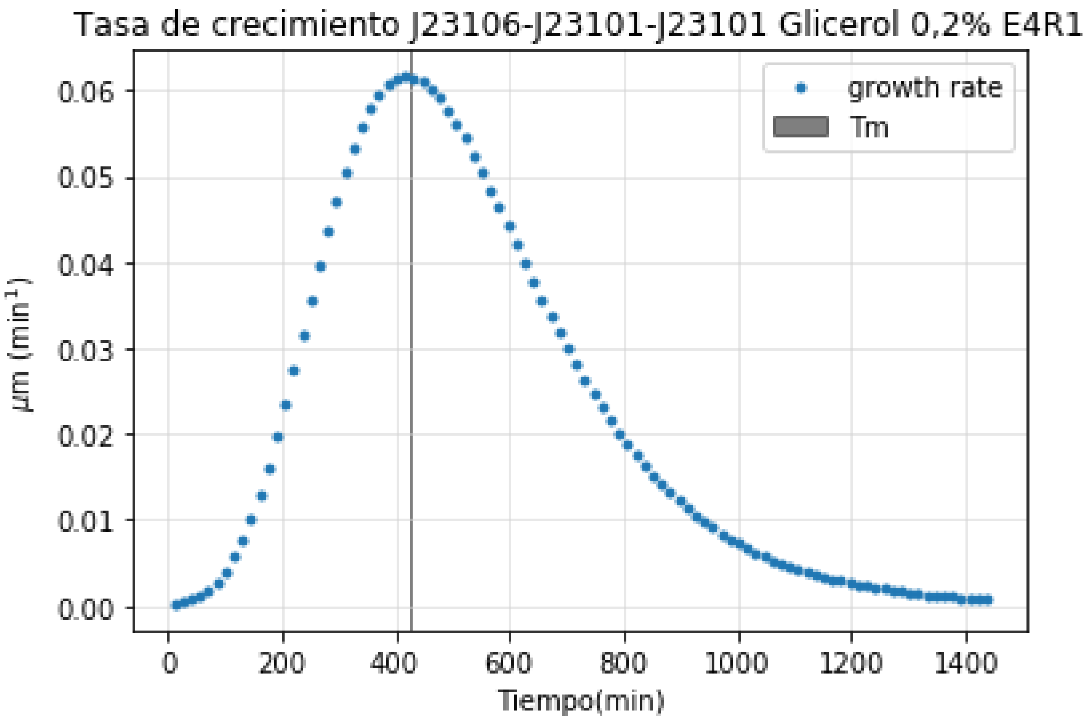

```
In [144]: #tasa de crecimiento
          ye33=((A33*np.exp(-np.exp((((um33*np.exp(1))/A33)*(l33-tt))+1))))
          #Con diff
          dy33=(np.diff(ye33))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm33,tm33, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy33,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[144]: <matplotlib.legend.Legend at 0x2036edbdb00>
```



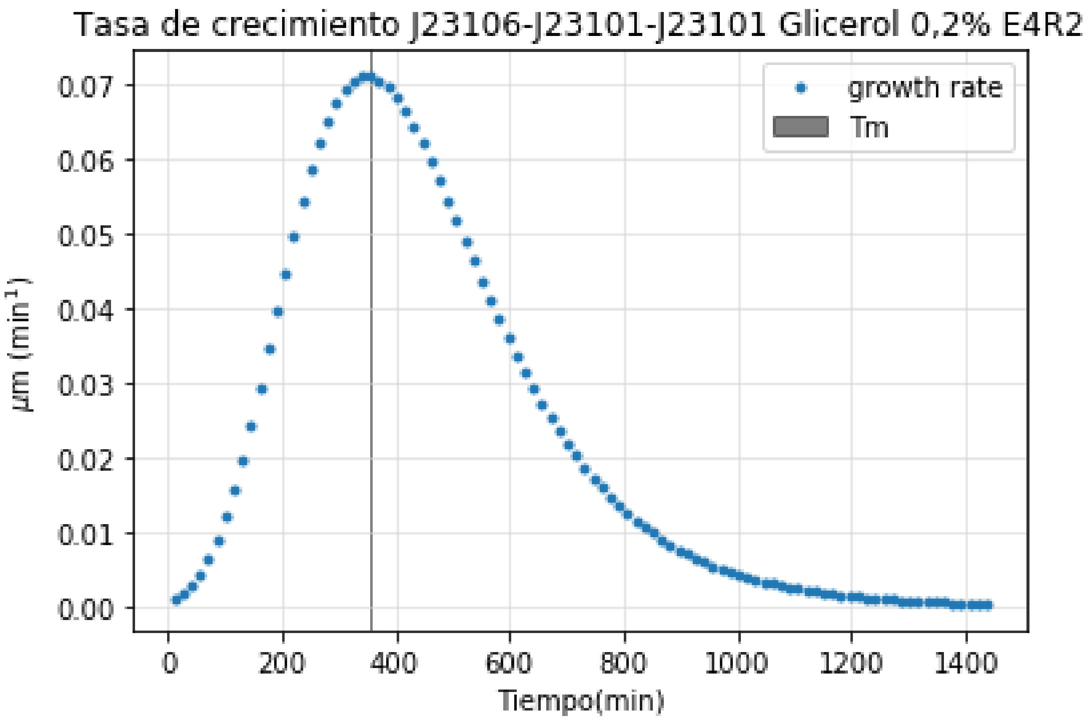

```
In [145]: #Tasas J23101-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy31,label='growth rate E4R1')
          plt.plot(tt[:-1],dy32,label='growth rate E4R2')
          plt.plot(tt[:-1],dy33,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[145]: <matplotlib.legend.Legend at 0x2036fe5cb70>
```



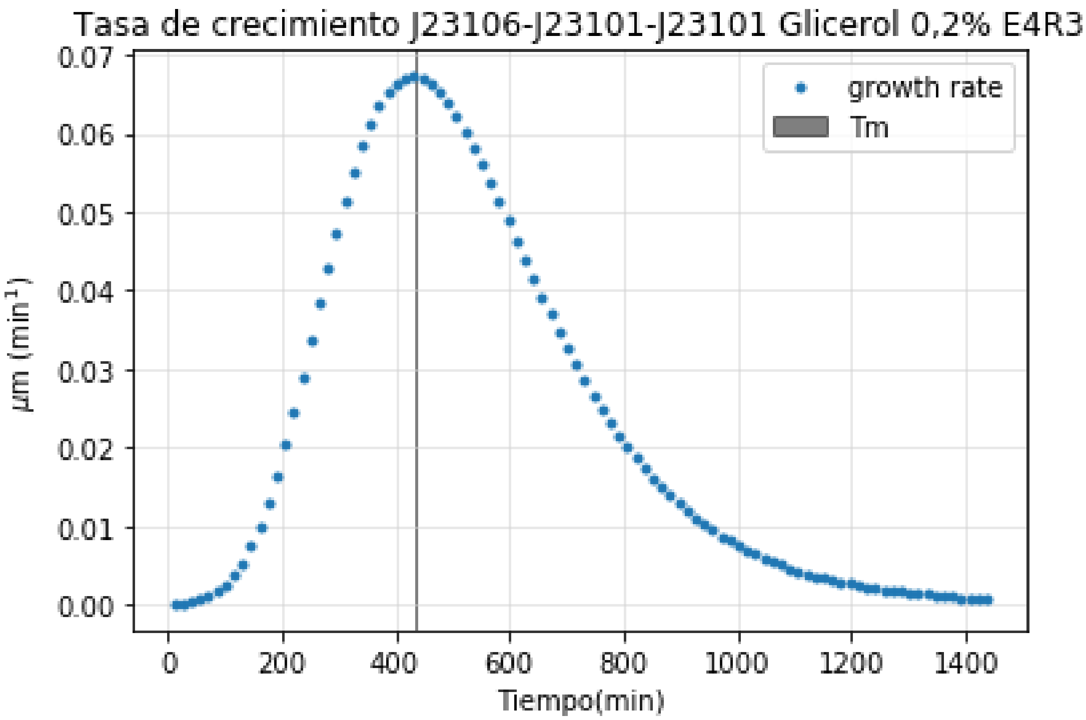

```
In [146]: #tasa de crecimiento
          ye34=((A34*np.exp(-np.exp((((um34*np.exp(1))/A34)*(l34-tt))+1))))
          #Con diff
          dy34=(np.diff(ye34))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm34,tm34, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy34,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[146]: <matplotlib.legend.Legend at 0x2036ff27fd0>
```



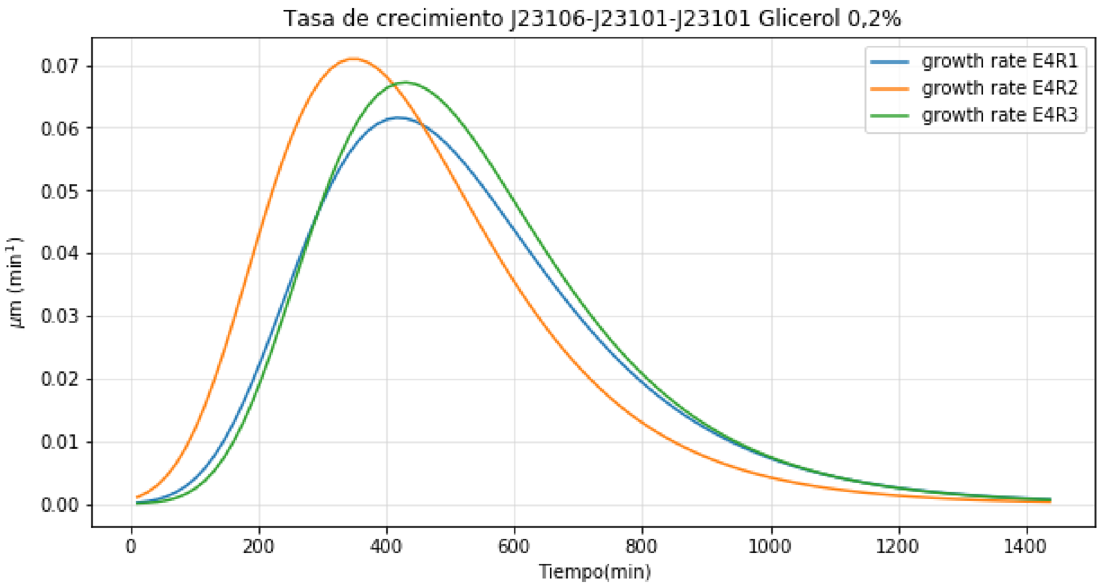

```
In [147]: #tasa de crecimiento
          ye35=((A35*np.exp(-np.exp((((um35*np.exp(1))/A35)*(l35-tt))+1))))
          #Con diff
          dy35=(np.diff(ye35))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R2')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm35,tm35, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy35,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[147]: <matplotlib.legend.Legend at 0x2036fffc978>
```



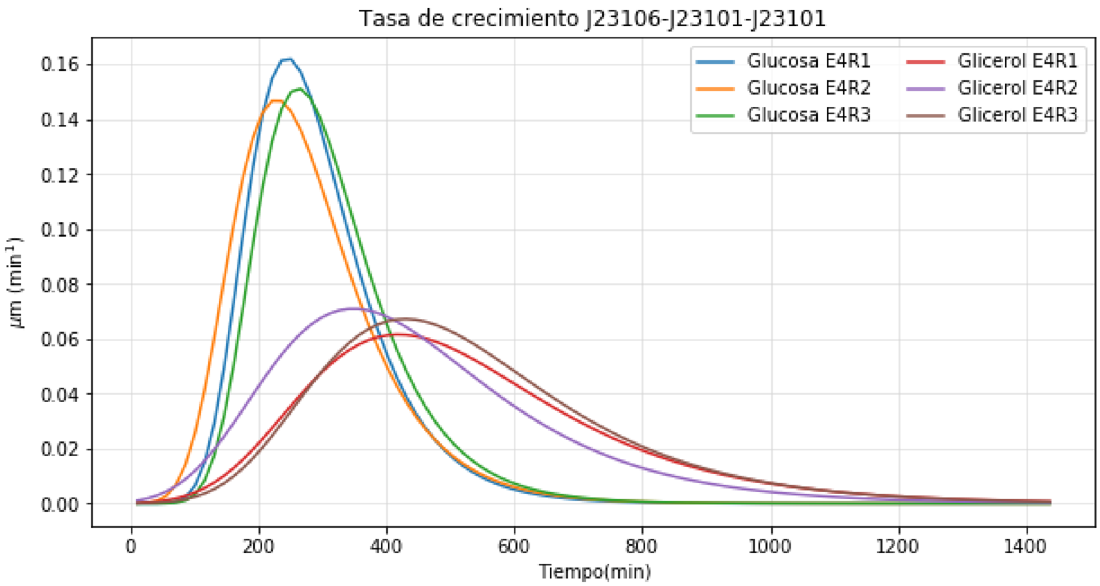

```
In [148]: #tasa de crecimiento
          ye36=((A36*np.exp(-np.exp((((um36*np.exp(1))/A36)*(l36-tt))+1))))
          #Con diff
          dy36=(np.diff(ye36))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.axvspan(tm36,tm36, color='k', alpha=0.5, label="Tm")
          plt.plot(tt[:-1],dy36,'.',label='growth rate')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[148]: <matplotlib.legend.Legend at 0x203700d19b0>
```



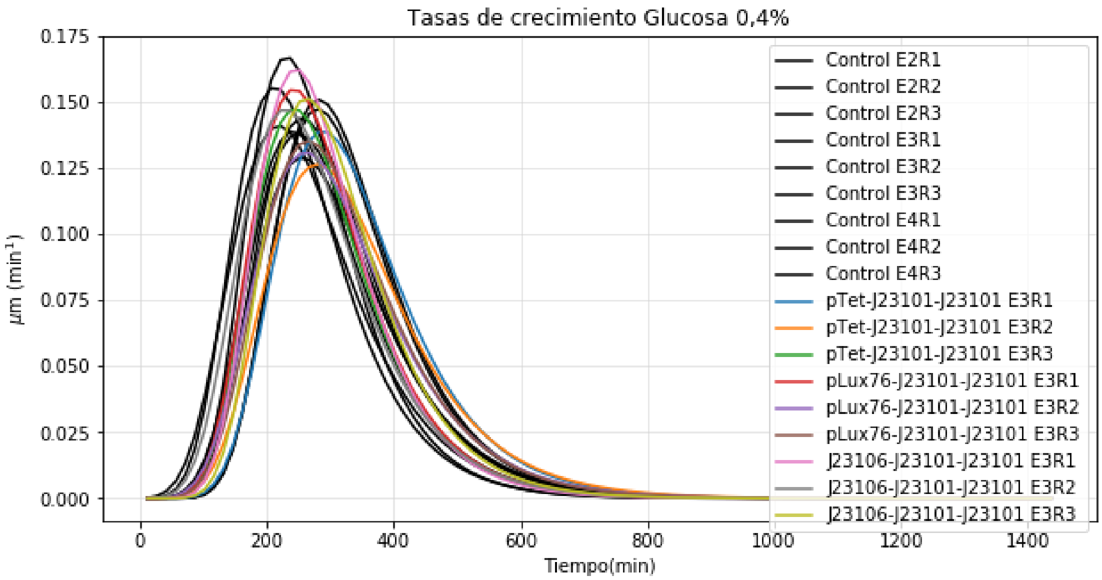

```
In [149]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy34,label='growth rate E4R1')
          plt.plot(tt[:-1],dy35,label='growth rate E4R2')
          plt.plot(tt[:-1],dy36,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[149]: <matplotlib.legend.Legend at 0x203701a6f98>
```




```
In [150]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23101-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy31,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy32,label='Glucosa E4R2')
          plt.plot(tt[:-1],dy33,label='Glucosa E4R3')
          plt.plot(tt[:-1],dy34,label='Glicerol E4R1')
          plt.plot(tt[:-1],dy35,label='Glicerol E4R2')
          plt.plot(tt[:-1],dy36,label='Glicerol E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[150]: <matplotlib.legend.Legend at 0x2037029c710>
```

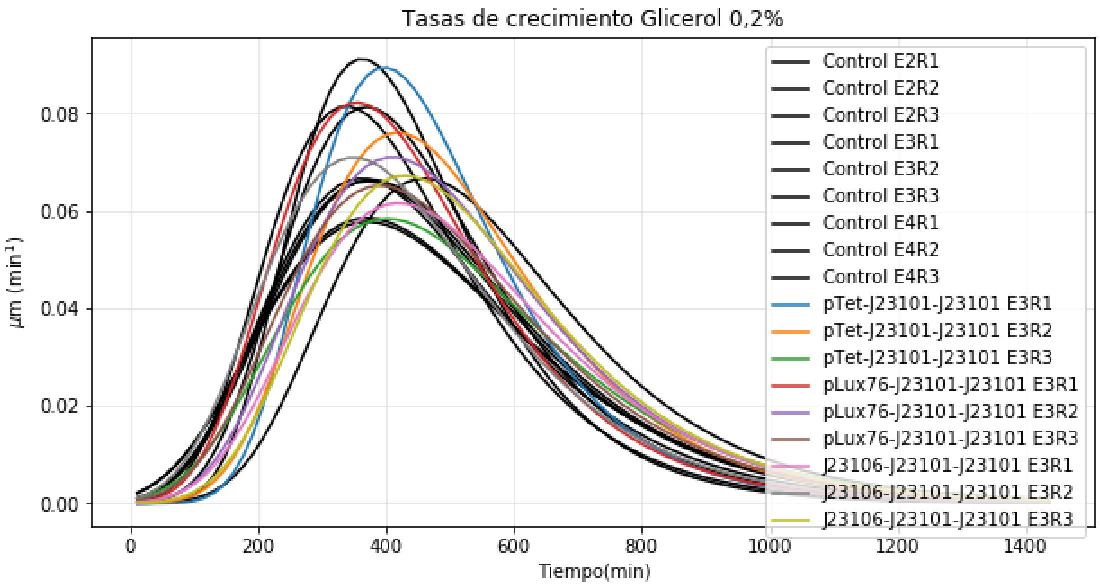



```
In [151]: #Tasas réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy1,'k',label='Control E2R1')
          plt.plot(tt[:-1],dy2,'k',label='Control E2R2')
          plt.plot(tt[:-1],dy3,'k',label='Control E2R3')
          plt.plot(tt[:-1],dy7,'k',label='Control E3R1')
          plt.plot(tt[:-1],dy8,'k',label='Control E3R2')
          plt.plot(tt[:-1],dy9,'k',label='Control E3R3')
          plt.plot(tt[:-1],dy13,'k',label='Control E4R1')
          plt.plot(tt[:-1],dy14,'k',label='Control E4R2')
          plt.plot(tt[:-1],dy15,'k',label='Control E4R3')
          plt.plot(tt[:-1],dy19,label='pTet-J23101-J23101 E3R1')
          plt.plot(tt[:-1],dy20,label='pTet-J23101-J23101 E3R2')
          plt.plot(tt[:-1],dy21,label='pTet-J23101-J23101 E3R3')
          plt.plot(tt[:-1],dy25,label='pLux76-J23101-J23101 E3R1')
          plt.plot(tt[:-1],dy26,label='pLux76-J23101-J23101 E3R2')
          plt.plot(tt[:-1],dy27,label='pLux76-J23101-J23101 E3R3')
          plt.plot(tt[:-1],dy31,label='J23106-J23101-J23101 E3R1')
          plt.plot(tt[:-1],dy32,label='J23106-J23101-J23101 E3R2')
          plt.plot(tt[:-1],dy33,label='J23106-J23101-J23101 E3R3')

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

Out[151]: <matplotlib.legend.Legend at 0x203703b3e10>
```



In [152]:
```
#Tasas réplicas glucosa
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 E2R1')
plt.plot(tt[:-1],dy5,'k',label='Control E2R2')
plt.plot(tt[:-1],dy6,'k',label='Control E2R3')
plt.plot(tt[:-1],dy10,'k',label='Control E3R1')
plt.plot(tt[:-1],dy11,'k',label='Control E3R2')
plt.plot(tt[:-1],dy12,'k',label='Control E3R3')
plt.plot(tt[:-1],dy16,'k',label='Control E4R1')
plt.plot(tt[:-1],dy17,'k',label='Control E4R2')
plt.plot(tt[:-1],dy18,'k',label='Control E4R3')
plt.plot(tt[:-1],dy22,label='pTet-J23101-J23101 E3R1')
plt.plot(tt[:-1],dy23,label='pTet-J23101-J23101 E3R2')
plt.plot(tt[:-1],dy24,label='pTet-J23101-J23101 E3R3')
plt.plot(tt[:-1],dy28,label='pLux76-J23101-J23101 E3R1')
plt.plot(tt[:-1],dy29,label='pLux76-J23101-J23101 E3R2')
plt.plot(tt[:-1],dy30,label='pLux76-J23101-J23101 E3R3')
plt.plot(tt[:-1],dy34,label='J23106-J23101-J23101 E3R1')
plt.plot(tt[:-1],dy35,label='J23106-J23101-J23101 E3R2')
plt.plot(tt[:-1],dy36,label='J23106-J23101-J23101 E3R3')

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

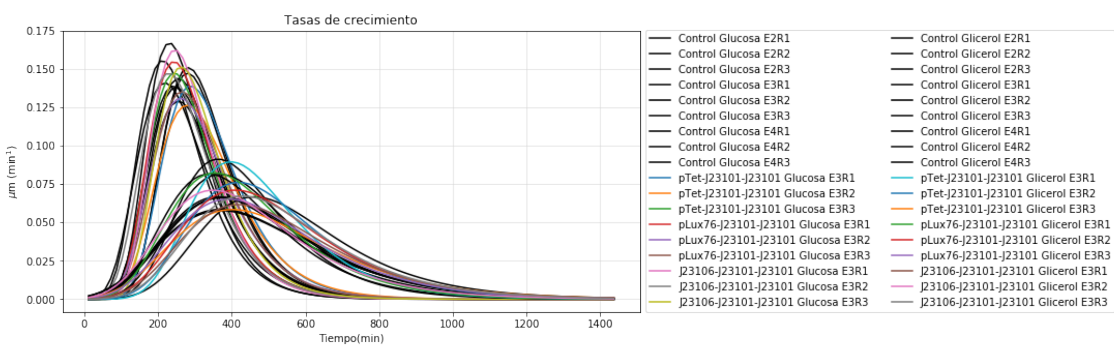
Out[152]: <matplotlib.legend.Legend at 0x20370520b70>



In [153]:
```
#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 Glucosa E2R1')
plt.plot(tt[:-1],dy2,'k',label='Control Glucosa E2R2')
plt.plot(tt[:-1],dy3,'k',label='Control Glucosa E2R3')
plt.plot(tt[:-1],dy7,'k',label='Control Glucosa E3R1')
plt.plot(tt[:-1],dy8,'k',label='Control Glucosa E3R2')
plt.plot(tt[:-1],dy9,'k',label='Control Glucosa E3R3')
plt.plot(tt[:-1],dy13,'k',label='Control Glucosa E4R1')
plt.plot(tt[:-1],dy14,'k',label='Control Glucosa E4R2')
plt.plot(tt[:-1],dy15,'k',label='Control Glucosa E4R3')
plt.plot(tt[:-1],dy19,label='pTet-J23101-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy20,label='pTet-J23101-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy21,label='pTet-J23101-J23101 Glucosa E3R3')
plt.plot(tt[:-1],dy25,label='pLux76-J23101-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy26,label='pLux76-J23101-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy27,label='pLux76-J23101-J23101 Glucosa E3R3')
plt.plot(tt[:-1],dy31,label='J23106-J23101-J23101 Glucosa E3R1')
plt.plot(tt[:-1],dy32,label='J23106-J23101-J23101 Glucosa E3R2')
plt.plot(tt[:-1],dy33,label='J23106-J23101-J23101 Glucosa E3R3')


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

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

Out[153]: <matplotlib.legend.Legend at 0x203706ed908>



In [ ]: