Ensayo 4 todo(J23106)

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
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)'''

cfp12231=np.fromfile('p1223gCFP1', sep=',')
rfp12231=np.fromfile('p1223gRFP1', sep=',')
yfp12231=np.fromfile('p1223gYFP1', sep=',')
od12231=np.fromfile('p1223gOD1', sep=',')
cfp12232=np.fromfile('p1223gCFP2', sep=',')
rfp12232=np.fromfile('p1223gRFP2', sep=',')
yfp12232=np.fromfile('p1223gYFP2', sep=',')
od12232=np.fromfile('p1223gOD2', sep=',')
cfp12233=np.fromfile('p1223gCFP3', sep=',')
rfp12233=np.fromfile('p1223gRFP3', sep=',')
yfp12233=np.fromfile('p1223gYFP3', sep=',')
od12233=np.fromfile('p1223gOD3', sep=',')

'''
print(cfp12231.shape)
print(rfp12231.shape)
print(yfp12231.shape)
print(od12231.shape)
print(cfp12232.shape)
print(rfp12232.shape)
print(yfp12232.shape)
print(od12232.shape)
print(cfp12233.shape)
print(rfp12233.shape)
print(yfp12233.shape)
print(od12233.shape)'''

cfp12261=np.fromfile('p1226gCFP1', sep=',')
rfp12261=np.fromfile('p1226gRFP1', sep=',')
yfp12261=np.fromfile('p1226gYFP1', sep=',')
od12261=np.fromfile('p1226gOD1', sep=',')
cfp12262=np.fromfile('p1226gCFP2', sep=',')
rfp12262=np.fromfile('p1226gRFP2', sep=',')
yfp12262=np.fromfile('p1226gYFP2', sep=',')
od12262=np.fromfile('p1226gOD2', sep=',')
cfp12263=np.fromfile('p1226gCFP3', sep=',')
rfp12263=np.fromfile('p1226gRFP3', sep=',')
yfp12263=np.fromfile('p1226gYFP3', sep=',')
od12263=np.fromfile('p1226gOD3', sep=',')

'''
print(cfp12261.shape)
print(rfp12261.shape)
print(yfp12261.shape)
print(od12261.shape)
```

```
print(cfp12262.shape)
print(rfp12262.shape)
print(yfp12262.shape)
print(od12262.shape)
print(cfp12263.shape)
print(rfp12263.shape)
print(yfp12263.shape)
print(od12263.shape)'''

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

'''
print(cfp12271.shape)
print(rfp12271.shape)
print(yfp12271.shape)
print(od12271.shape)
print(cfp12272.shape)
print(rfp12272.shape)
print(yfp12272.shape)
print(od12272.shape)
print(cfp12273.shape)
print(rfp12273.shape)
print(yfp12273.shape)
print(od12273.shape)'''

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

```
yfpcg3=np.fromfile('pcgYFP3', sep=',')
odcg3=np.fromfile('pcgOD3', sep=',')
'''
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)
print(cfpcg1.shape)
print(rfpcg1.shape)
print(yfpcg1.shape)
print(odcg1.shape)'''
```

Out[2]: '\nprint(cfpcg1.shape)\nprint(rfpcg1.shape)\nprint(yfpcg1.shape)\nprint(odcg1.shape)\npr

```
In [3]: #Promedios glicerol
        #arrays replicas glicerol
        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)'''


        cfp1223g1=np.fromfile('p1223glCFP1', sep=',')
```

```
rfp1223g1=np.fromfile('p1223glRFP1', sep=',')
yfp1223g1=np.fromfile('p1223glYFP1', sep=',')
od1223g1=np.fromfile('p1223glOD1', sep=',')
cfp1223g2=np.fromfile('p1223glCFP2', sep=',')
rfp1223g2=np.fromfile('p1223glRFP2', sep=',')
yfp1223g2=np.fromfile('p1223glYFP2', sep=',')
od1223g2=np.fromfile('p1223glOD2', sep=',')
cfp1223g3=np.fromfile('p1223glCFP3', sep=',')
rfp1223g3=np.fromfile('p1223glRFP3', sep=',')
yfp1223g3=np.fromfile('p1223glYFP3', sep=',')
od1223g3=np.fromfile('p1223glOD3', sep=',')
'''
print(cfp1223g1.shape)
print(rfp1223g1.shape)
print(yfp1223g1.shape)
print(od1223g1.shape)
print(cfp1223g2.shape)
print(rfp1223g2.shape)
print(yfp1223g2.shape)
print(od1223g2.shape)
print(cfp1223g3.shape)
print(rfp1223g3.shape)
print(yfp1223g3.shape)
print(od1223g3.shape)'''


cfp1226g1=np.fromfile('p1226glCFP1', sep=',')
rfp1226g1=np.fromfile('p1226glRFP1', sep=',')
yfp1226g1=np.fromfile('p1226glYFP1', sep=',')
od1226g1=np.fromfile('p1226glOD1', sep=',')
cfp1226g2=np.fromfile('p1226glCFP2', sep=',')
rfp1226g2=np.fromfile('p1226glRFP2', sep=',')
yfp1226g2=np.fromfile('p1226glYFP2', sep=',')
od1226g2=np.fromfile('p1226glOD2', sep=',')
cfp1226g3=np.fromfile('p1226glCFP3', sep=',')
rfp1226g3=np.fromfile('p1226glRFP3', sep=',')
yfp1226g3=np.fromfile('p1226glYFP3', sep=',')
od1226g3=np.fromfile('p1226glOD3', sep=',')

'''
print(cfp1226g1.shape)
print(rfp1226g1.shape)
print(yfp1226g1.shape)
print(od1226g1.shape)
print(cfp1226g2.shape)
print(rfp1226g2.shape)
print(yfp1226g2.shape)
print(od1226g2.shape)
```

```
print(cfp1226g3.shape)
print(rfp1226g3.shape)
print(yfp1226g3.shape)
print(od1226g3.shape)'''

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

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


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

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

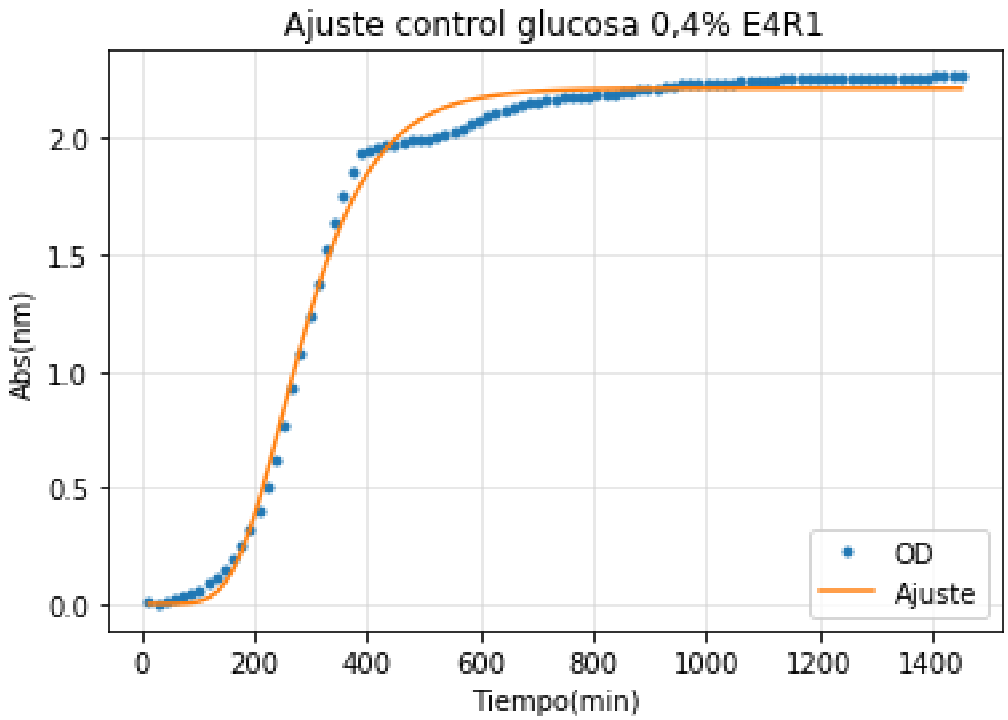
Out[3]: '\nprint(cfpcgl1.shape)\nprint(rfpcgl1.shape)\nprint(yfpcgl1.shape)\nprint(odcgl1.shape)

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

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

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

        print(z)

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

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

        Y_fit=evalF,z

        return(Y_fit)
```

In [5]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 1
y1 = np.log(odcg1)-np.log(np.min(odcg1))
print('Min OD = %e'%((np.min(odcg1))))
evaly, params=Function_fit(tt,y1,0,-1,title = 'Ajuste control glucosa 0,4% E4R1')
```

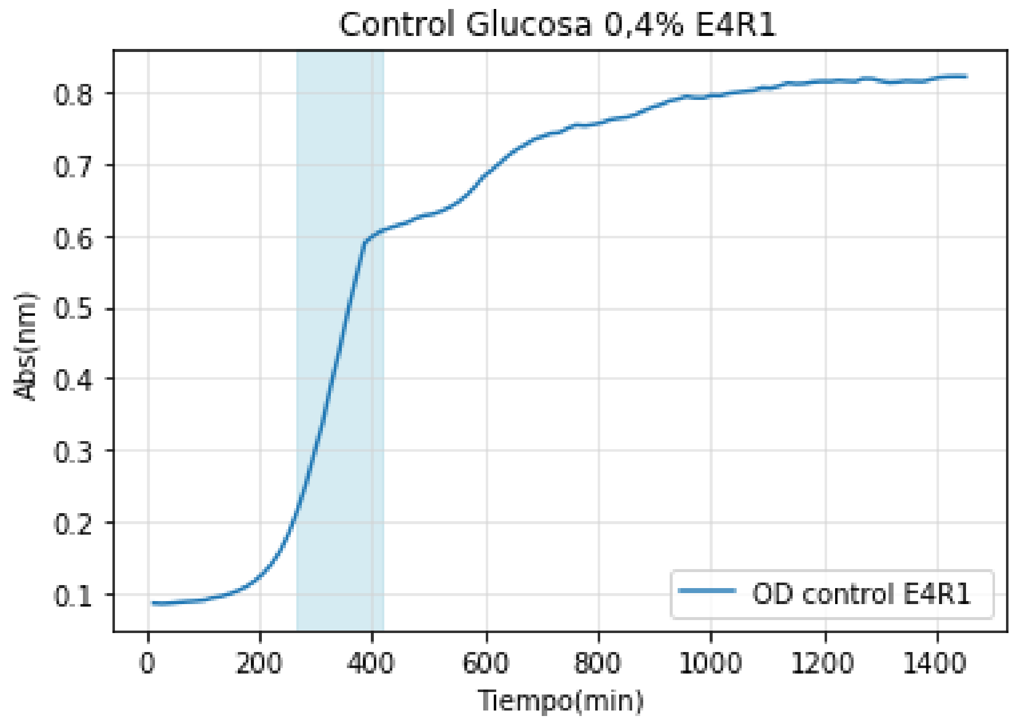
```
A1 = params[0]
um1=params[1]
l1=params[2]
print('A=%e'%(A1))
print('um=%e'%(um1))
print('l=%e'%(l1))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
y2=tt[27]
plt.figure()
plt.title('Control Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg1,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],odcg1[17:28],label='OD control E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

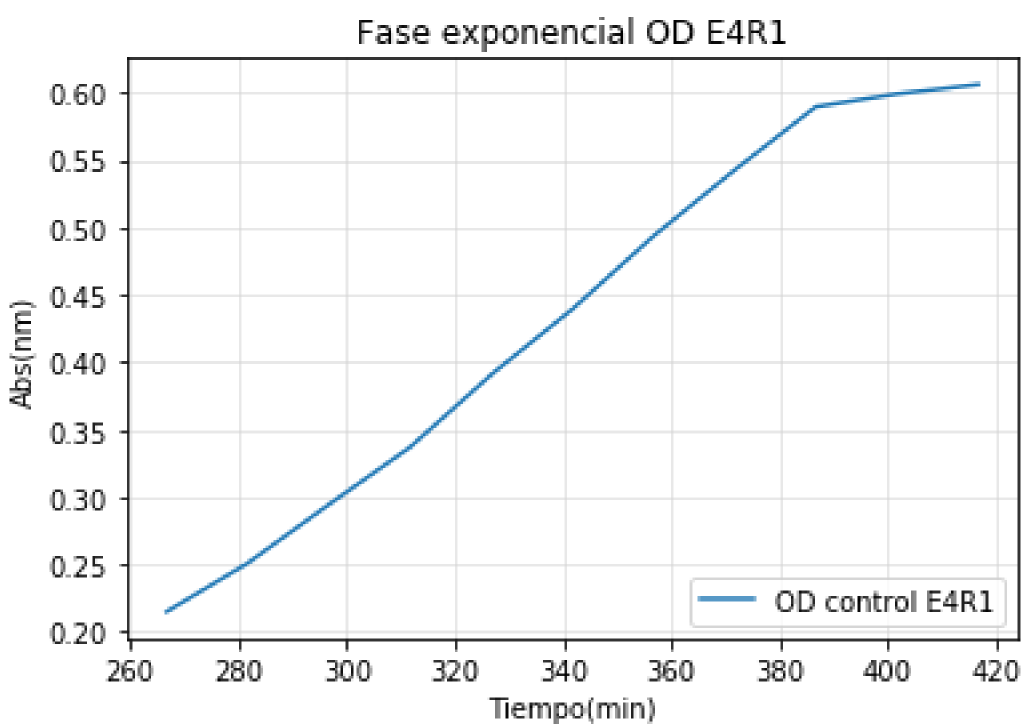
```
Min OD = 8.525000e-02
[  2.21400213e+00   9.27035018e-03   1.61117814e+02]
```




```
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
```

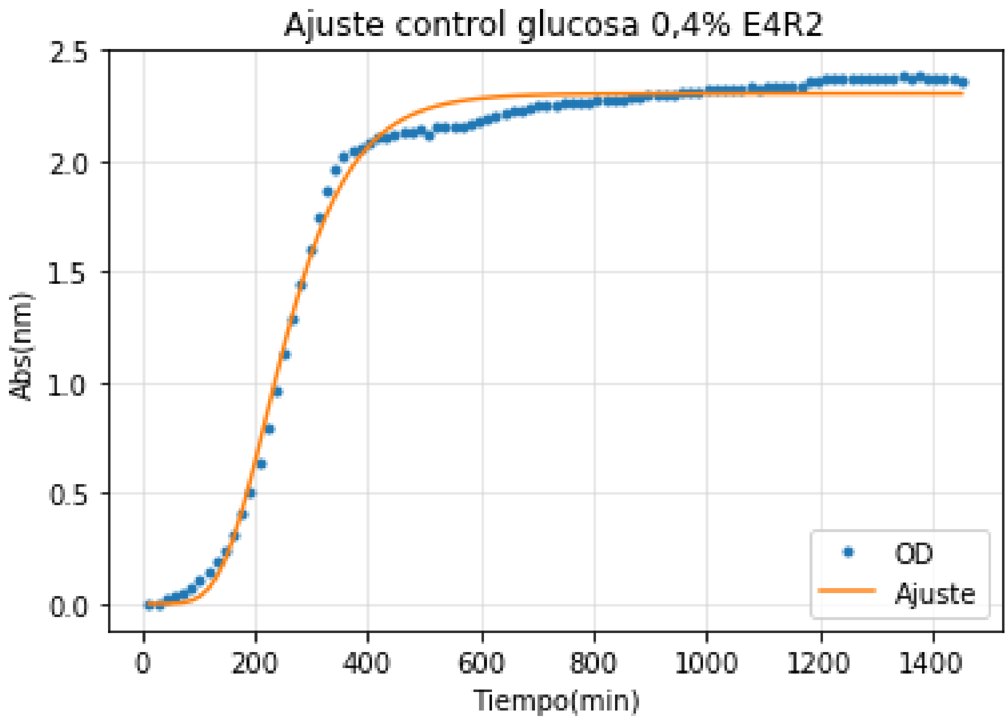
```
Out[5]: <matplotlib.legend.Legend at 0x1b203e54940>
```


Control Glucosa 0,4% E4R1
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 E4R1


Fase exponencial OD E4R1
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
260
280
300
320
340
360
380
400
420
Tiempo(min)
OD control E4R1

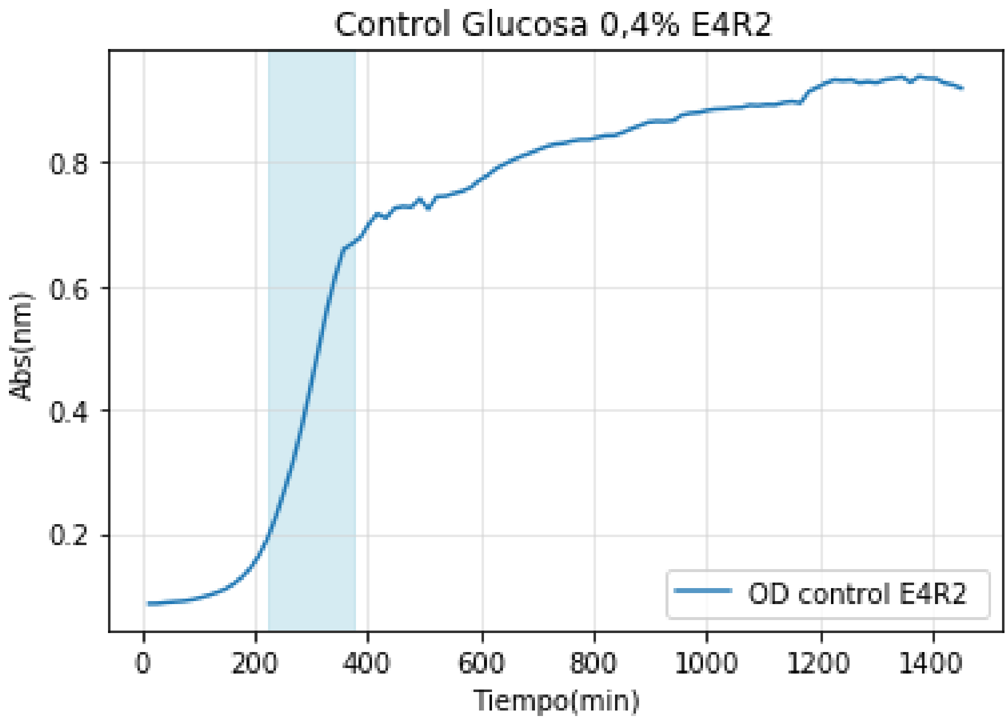
In [6]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 2
y2= np.log(odcg2)-np.log(np.min(odcg2))
print('Min OD = %e'%((np.min(odcg2))))
evaly, params=Function_fit(tt,y2,0,-1, title = 'Ajuste control glucosa 0,4% E4R2')
A2 = params[0]
um2=params[1]
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[14]
y2=tt[24]
plt.figure()
plt.title('Control Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg2,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],odcg2[14:25],label='OD control E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

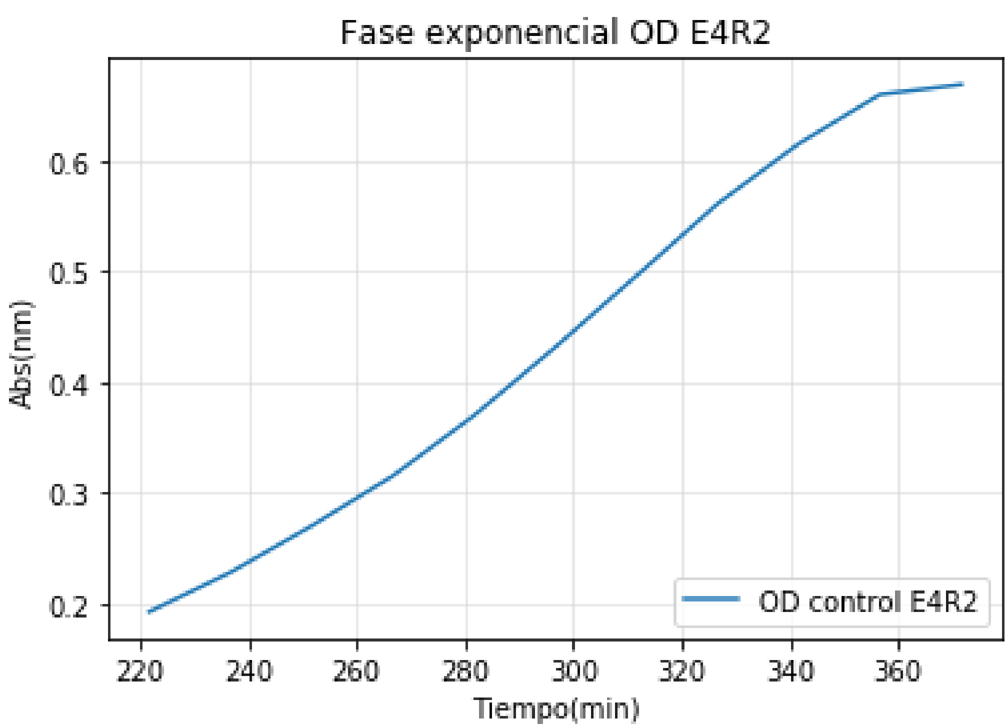
```
Min OD = 8.675000e-02
[  2.30555639e+00   1.03788860e-02   1.38136011e+02]
```




```
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
```

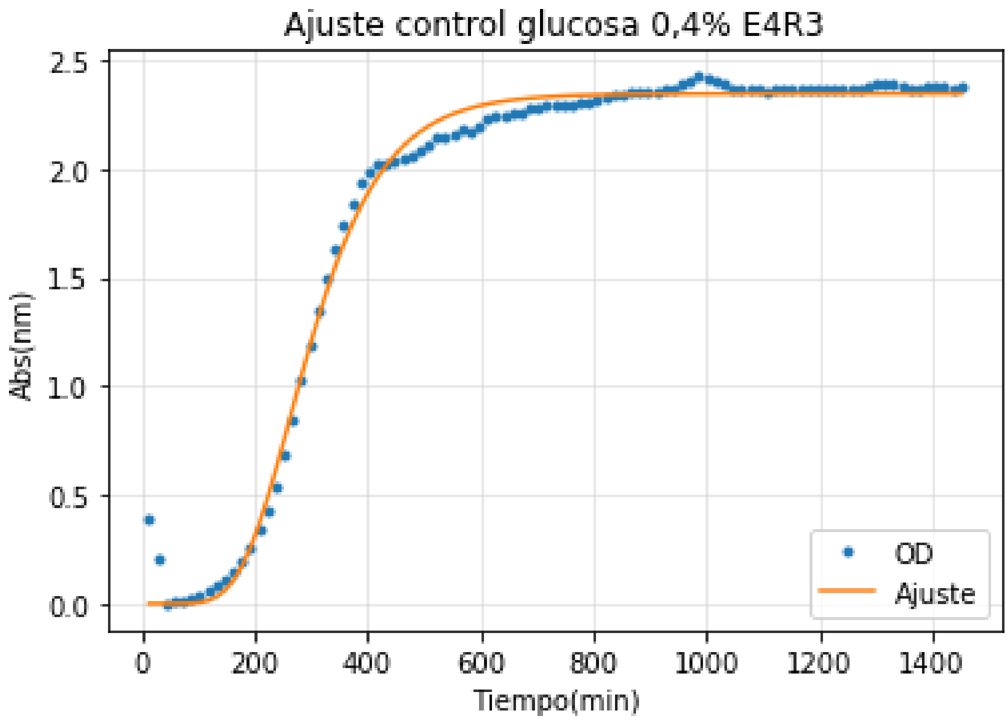
```
Out[6]: <matplotlib.legend.Legend at 0x1b204026a20>
```


Control Glucosa 0,4% E4R2
Abs(nm)
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
1200
1400
Tiempo(min)
OD control E4R2


Fase exponencial OD E4R2
Abs(nm)
0.2
0.3
0.4
0.5
0.6
220
240
260
280
300
320
340
360
Tiempo(min)
OD control E4R2

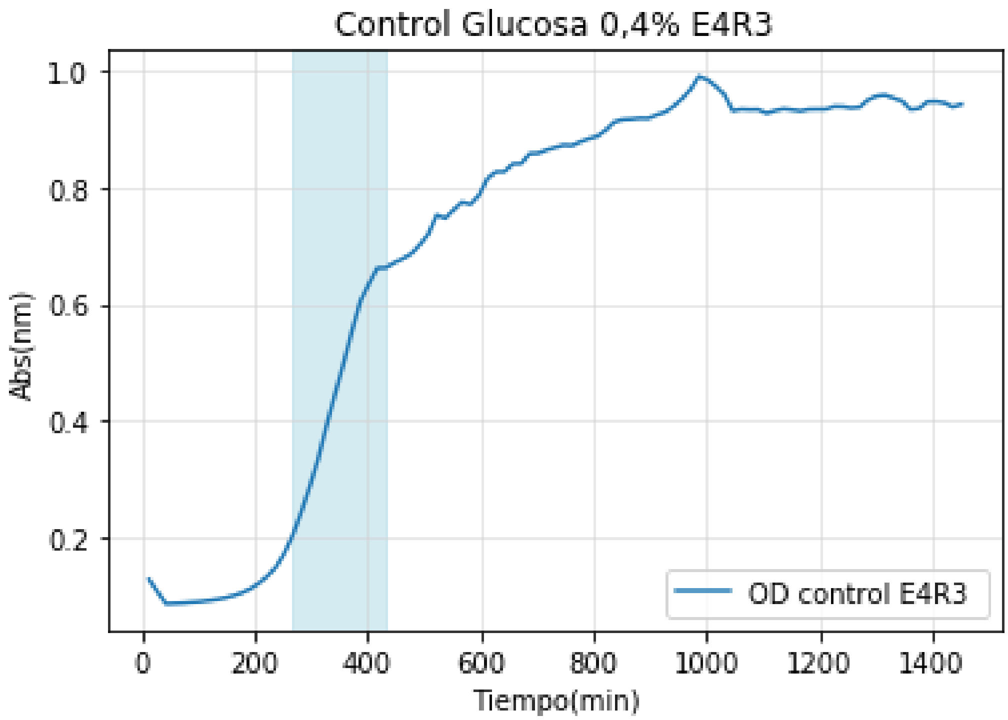
In [7]:
```python
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glucosa rep 3
y3= np.log(odcg3)-np.log(np.min(odcg3))
print('Min OD = %e'%((np.min(odcg3))))
evaly, params=Function_fit(tt,y3,0,-1, title = 'Ajuste control glucosa 0,4% E4R3')
A3= params[0]
um3=params[1]
l3=params[2]
print('A=%e'%(A3))
print('um=%e'%(um3))
print('l=%e'%(l3))

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


#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[17]
y2=tt[28]
plt.figure()
plt.title('Control Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcg3,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],odcg3[17:29],label='OD control E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

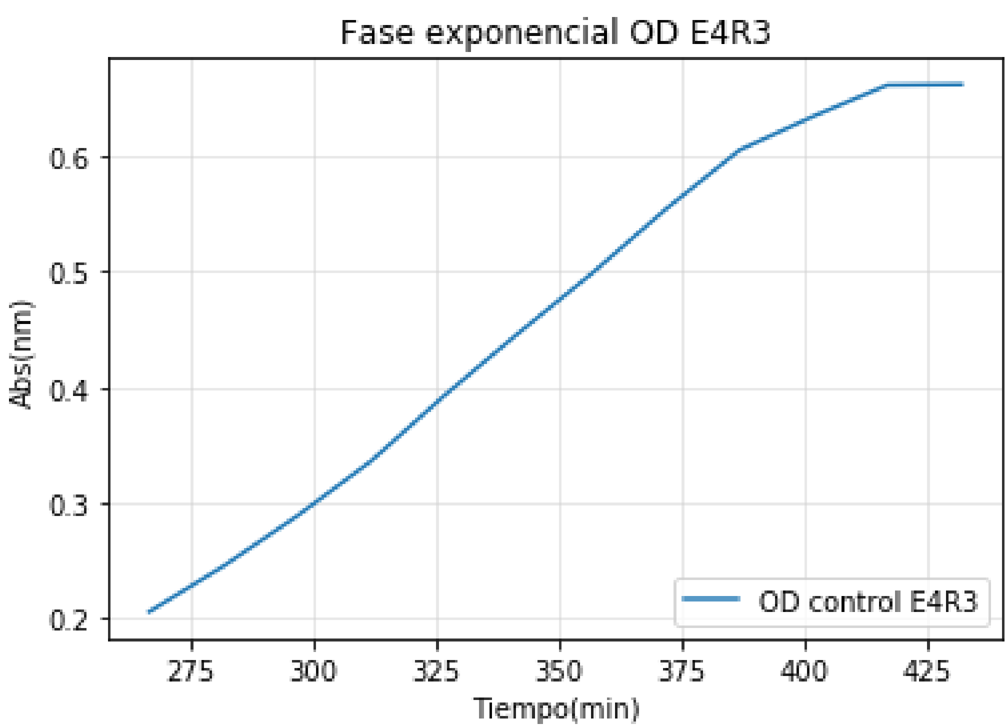
```
Min OD = 8.775000e-02
[  2.34400664e+00   9.59456675e-03   1.72702880e+02]
```




```
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
```

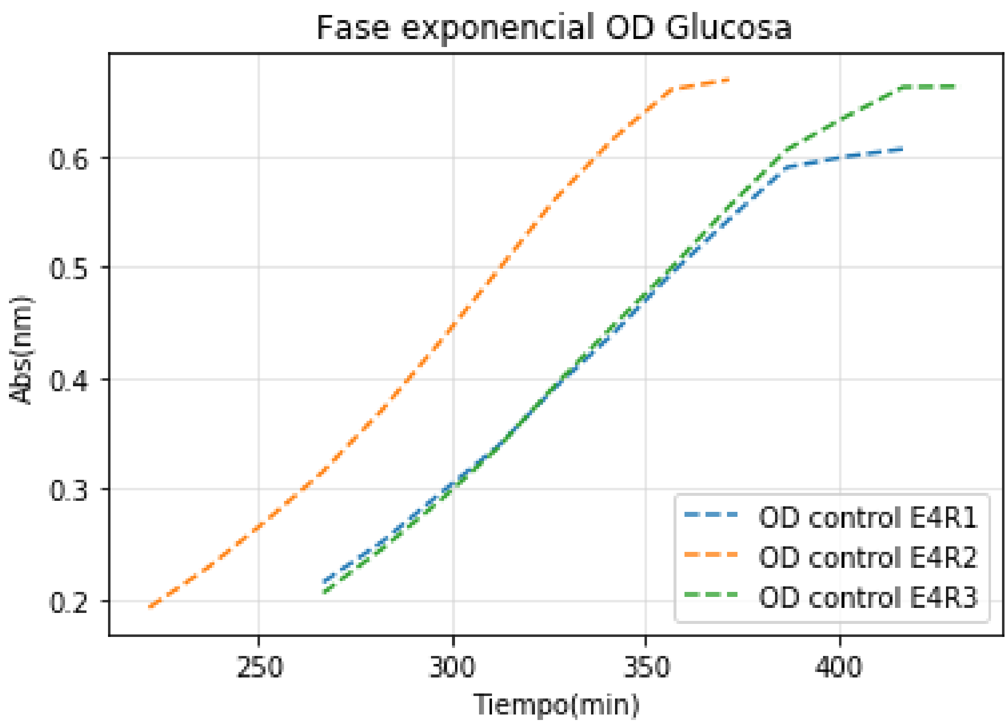
```
Out[7]: <matplotlib.legend.Legend at 0x1b2051f1828>
```


Control Glucosa 0,4% E4R3
Abs(nm)
Tiempo(min)
OD control E4R3
1.0
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
1200
1400


Fase exponencial OD E4R3
Abs(nm)
Tiempo(min)
OD control E4R3
0.6
0.5
0.4
0.3
0.2
275
300
325
350
375
400
425

In [8]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD Glucosa')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],odcg1[17:28],'--',label='OD control E4R1')
plt.plot(tt[14:25],odcg2[14:25],'--',label='OD control E4R2')
plt.plot(tt[17:29],odcg3[17:29],'--',label='OD control E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

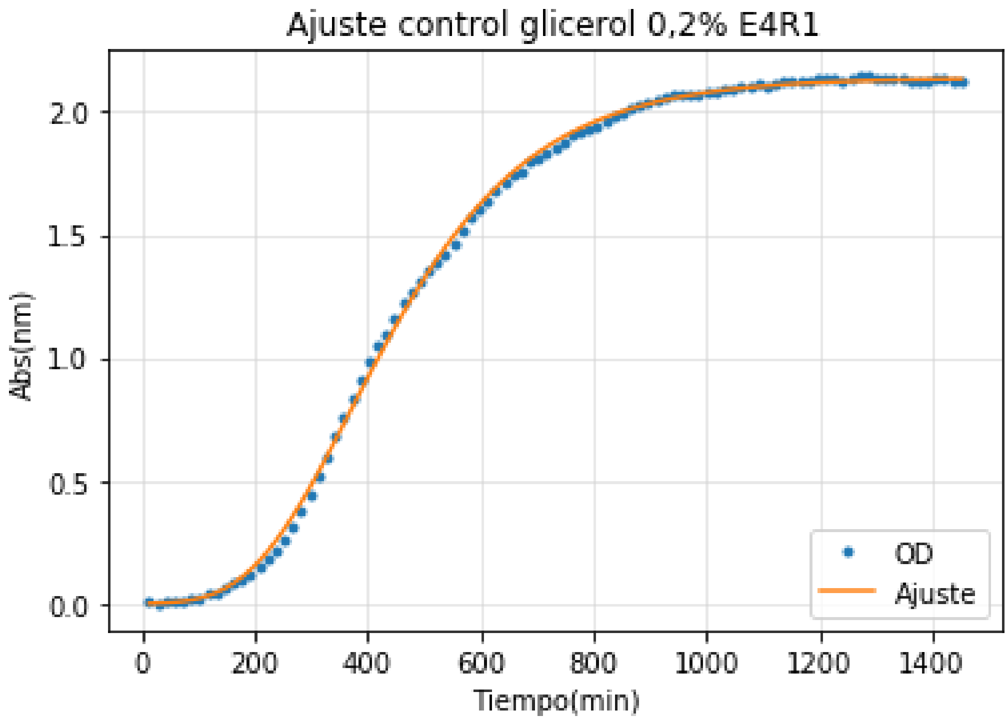
Out[8]: <matplotlib.legend.Legend at 0x1b205309ba8>



In [9]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 1
y4= np.log(odcgl1)-np.log(np.min(odcgl1))
print('Min OD = %e'%((np.min(odcgl1))))
evaly, params=Function_fit(tt,y4,0,-1, title = 'Ajuste control glicerol 0,2% E4R1')
A4= params[0]
um4=params[1]
l4=params[2]
print('A=%e'%(A4))
print('um=%e'%(um4))
print('l=%e'%(l4))
```

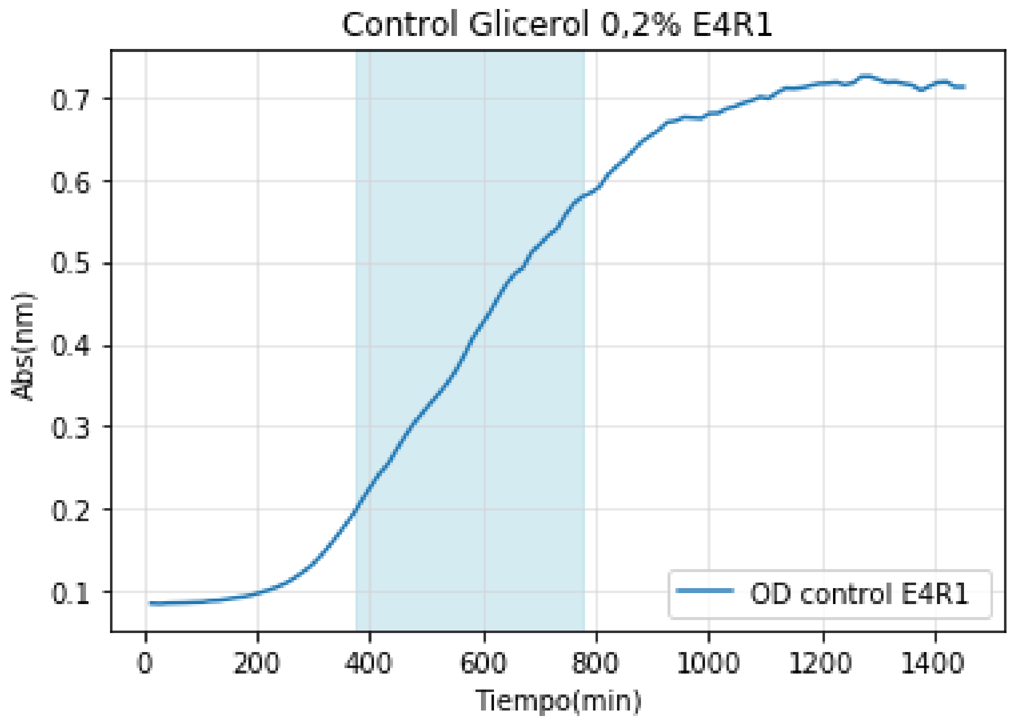
```python
#Cálculo datos para determinar extensión de la fase exponencial
tm4=((A4/(np.exp(1)*um4))+l4)
print('Tm=%e'%(tm4))
t24=((np.log(2))/um4)
print('doubpe=%e'%(t24))
extdp4=2.5*t24
print('ext=%e'%extdp4)
ttot4=tm4+extdp4
print('Tfinal=%e'%ttot4)

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[24]
y2=tt[51]
plt.figure()
plt.title('Control Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl1,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],odcgl1[24:52],label='OD control E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

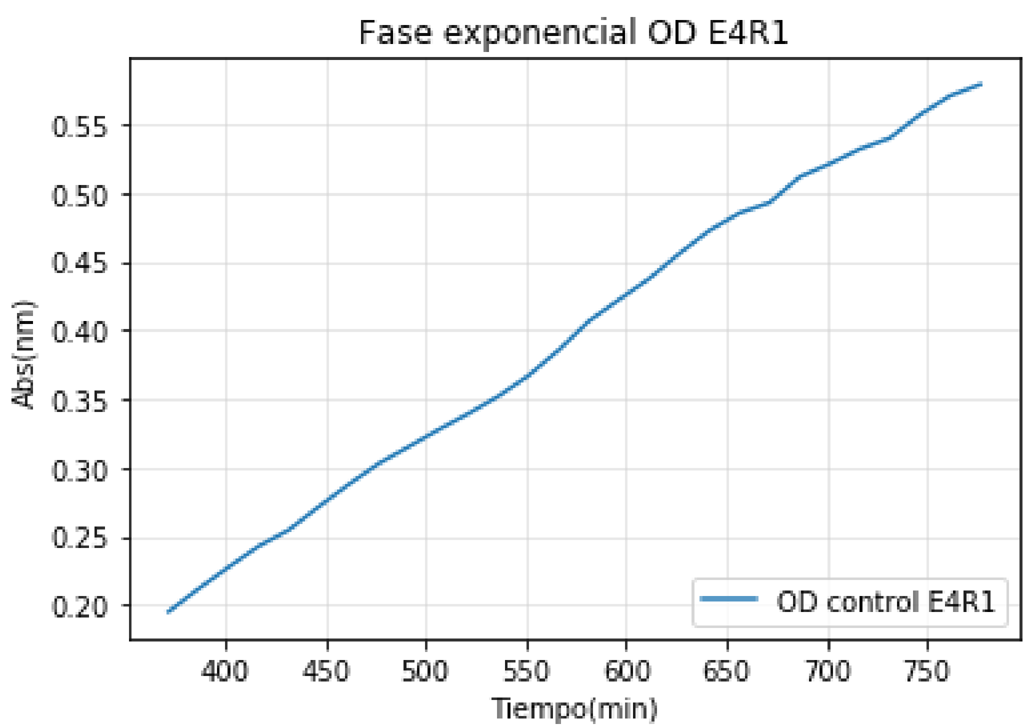
```
Min OD = 8.500000e-02
[  2.13735411e+00   4.44419084e-03   1.92719677e+02]
```




```
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
```

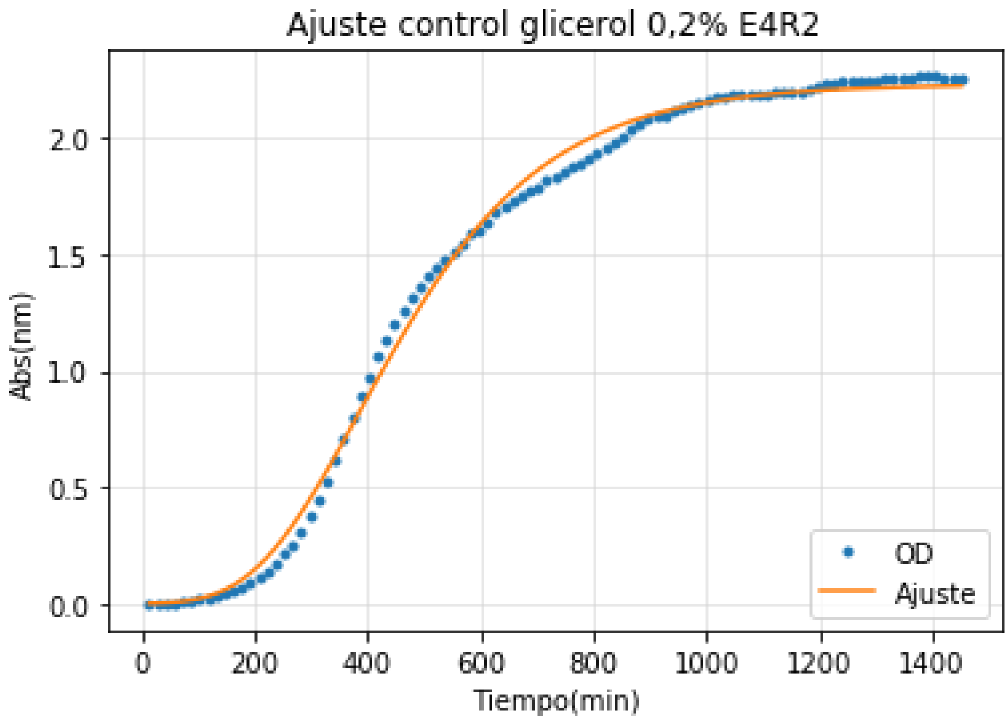
```
Out[9]: <matplotlib.legend.Legend at 0x1b20548e128>
```


Control Glicerol 0,2% E4R1
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 E4R1


Fase exponencial OD E4R1
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
400
450
500
550
600
650
700
750
Tiempo(min)
OD control E4R1

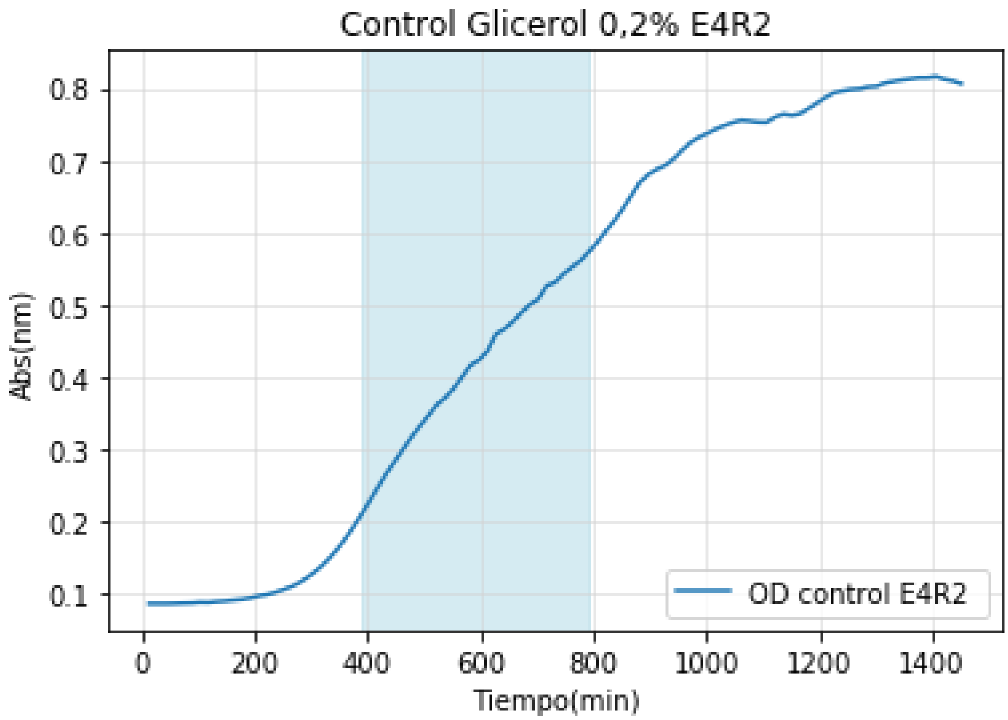
```
In [10]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #control glicerl rep 2
         y5= np.log(odcgl2)-np.log(np.min(odcgl2))
         print('Min OD = %e'%((np.min(odcgl2))))
         evaly, params=Function_fit(tt,y5,0,-1, title = 'Ajuste control glicerol 0,2% E4R2')
         A5= params[0]
         um5=params[1]
         l5=params[2]
         print('A=%e'%(A5))
         print('um=%e'%(um5))
         print('l=%e'%(l5))

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

         #Delimitación fase exponencial en grafico con OD/tiempo
         y1=tt[25]
         y2=tt[52]
         plt.figure()
         plt.title('Control Glicerol 0,2% E4R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt,odcgl2,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],odcgl2[25:53],label='OD control E4R2')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

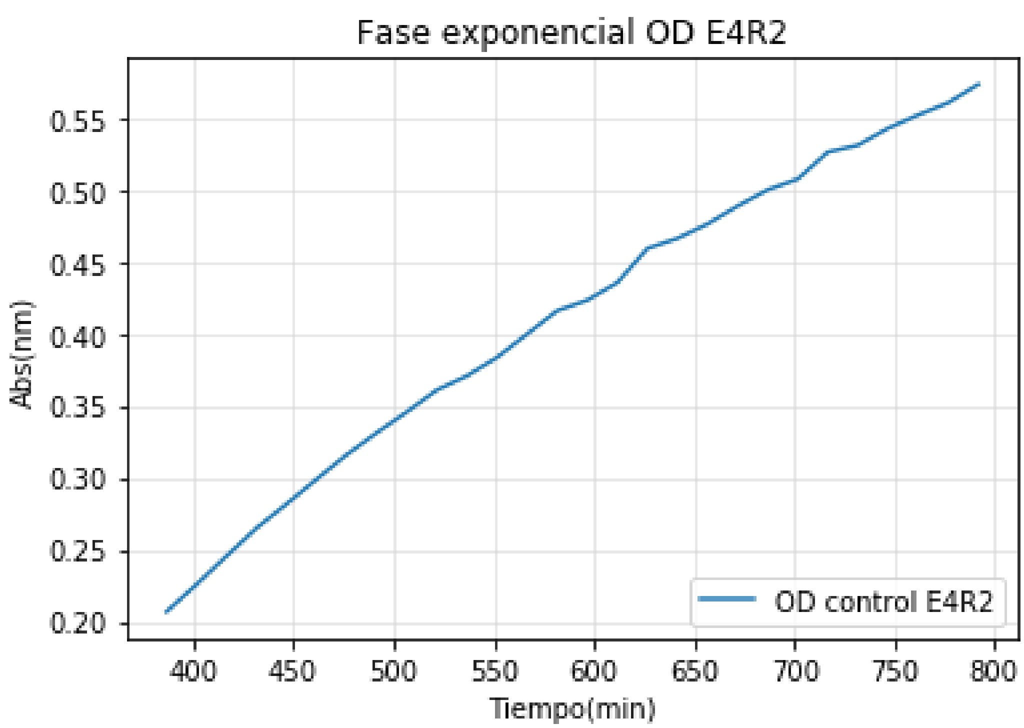
```
Min OD = 8.500000e-02
[  2.22944589e+00   4.42704053e-03   1.98978142e+02]
```




```
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
```

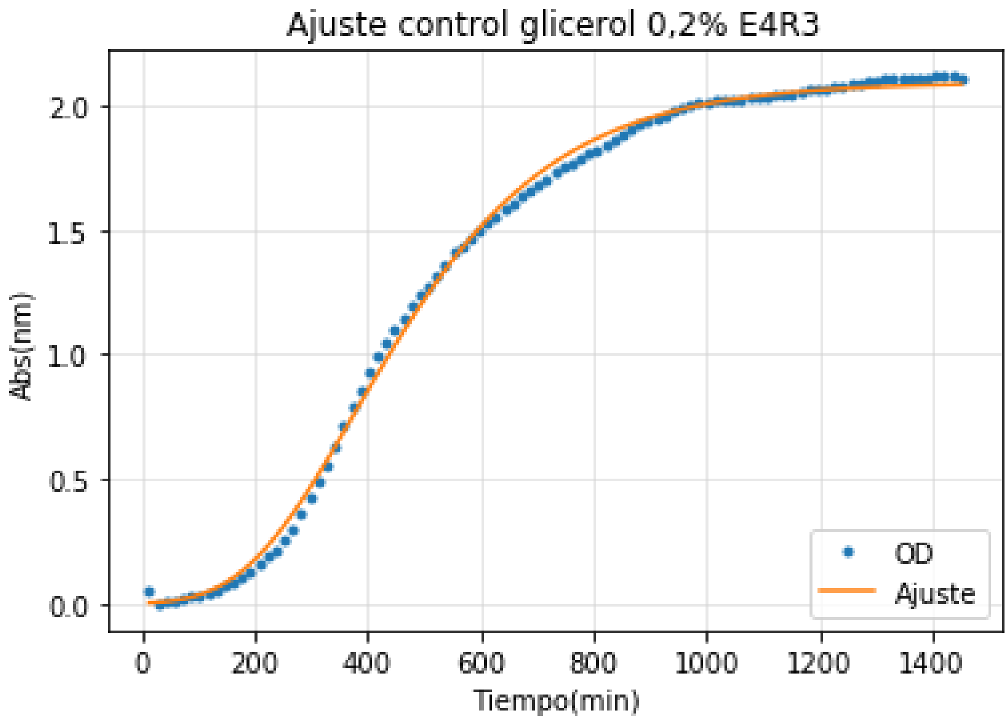
```
Out[10]: <matplotlib.legend.Legend at 0x1b203fc29e8>
```


Control Glicerol 0,2% E4R2
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 E4R2


Fase exponencial OD E4R2
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Abs(nm)
400
450
500
550
600
650
700
750
800
Tiempo(min)
OD control E4R2

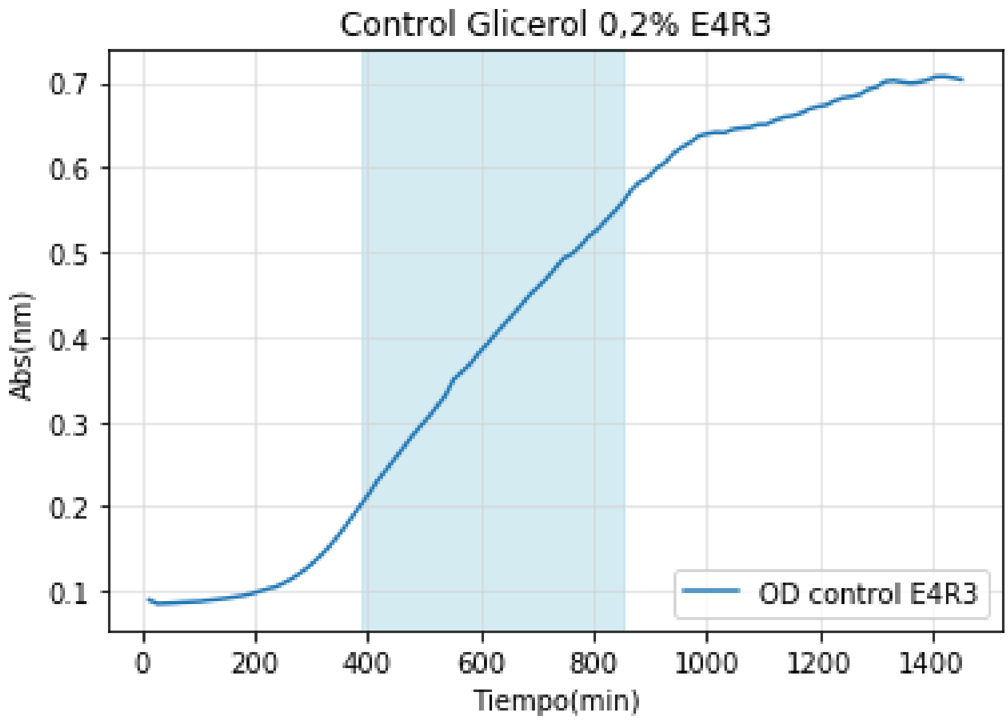
In [11]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#control glicerl rep 3
y6= np.log(odcgl3)-np.log(np.min(odcgl3))
print('Min OD = %e'%((np.min(odcgl3))))
evaly, params=Function_fit(tt,y6,0,-1, title = 'Ajuste control glicerol 0,2% E4R3')
A6= params[0]
um6=params[1]
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% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,odcgl3,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],odcgl3[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]
```

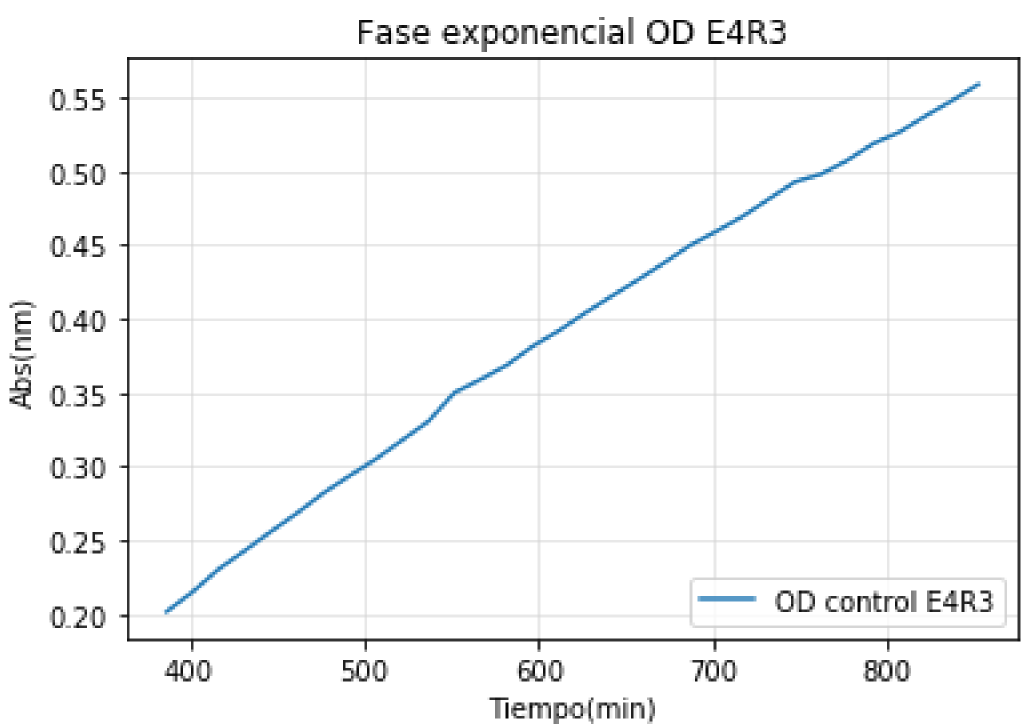



```
A=2.093421e+00
um=3.902641e-03
l=1.802688e+02
Tm=3.776035e+02
doubpe=1.776098e+02
ext=4.440245e+02
Tfinal=8.216280e+02
```

Out[11]: <matplotlib.legend.Legend at 0x1b203f29ba8>



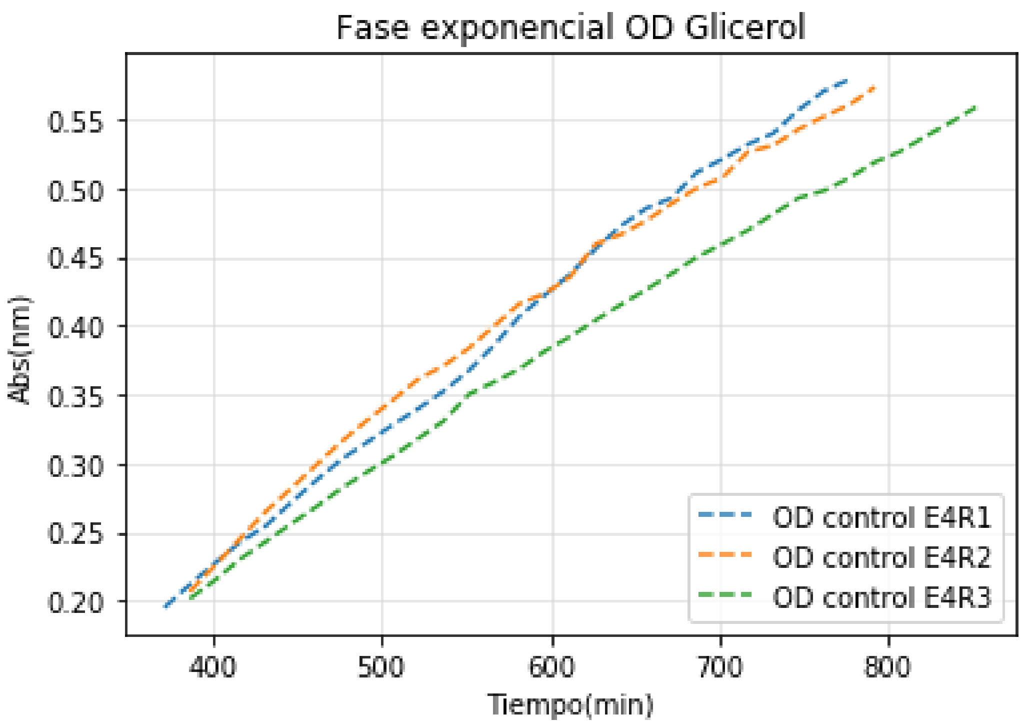
Control Glicerol 0,2% E4R3
Abs(nm)
Tiempo(min)
OD control E4R3



Fase exponencial OD E4R3
Abs(nm)
Tiempo(min)
OD control E4R3

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[24:52],odcgl1[24:52],'--',label='OD control E4R1')
plt.plot(tt[25:53],odcgl2[25:53],'--',label='OD control E4R2')
plt.plot(tt[25:57],odcgl3[25:57],'--',label='OD control E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

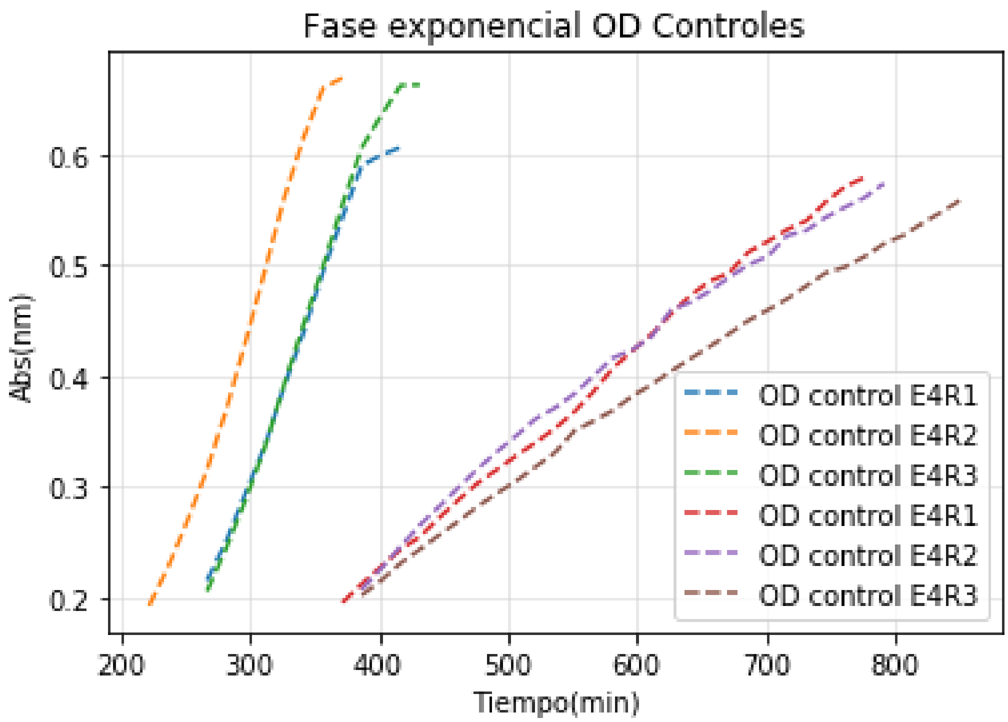
Out[12]: <matplotlib.legend.Legend at 0x1b2055c4a20>



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[17:28],odcg1[17:28],'--',label='OD control E4R1')
plt.plot(tt[14:25],odcg2[14:25],'--',label='OD control E4R2')
plt.plot(tt[17:29],odcg3[17:29],'--',label='OD control E4R3')
plt.plot(tt[24:52],odcgl1[24:52],'--',label='OD control E4R1')
plt.plot(tt[25:53],odcgl2[25:53],'--',label='OD control E4R2')
plt.plot(tt[25:57],odcgl3[25:57],'--',label='OD control E4R3')
```

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

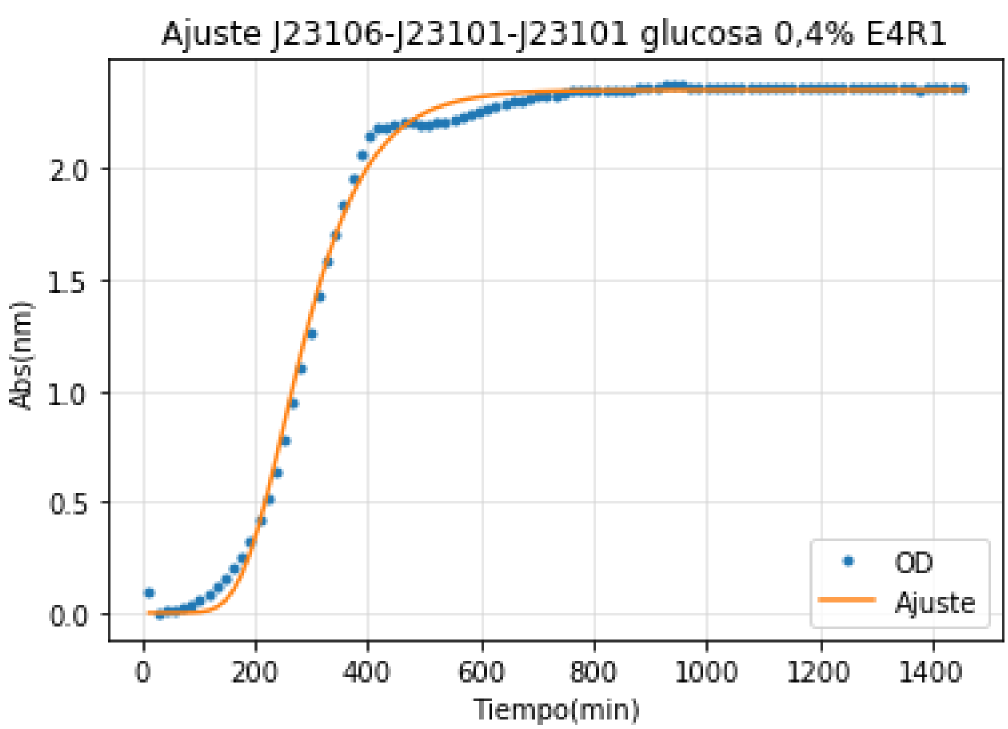
Out[13]: <matplotlib.legend.Legend at 0x1b20568f710>



In [14]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 1
y7= np.log(od12211)-np.log(np.min(od12211))
print('Min OD = %e'%((np.min(od12211))))
evaly, params=Function_fit(tt,y7,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4%
A7 = params[0]
um7=params[1]
l7=params[2]
print('A=%e'%(A7))
print('um=%e'%(um7))
print('l=%e'%(l7))

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

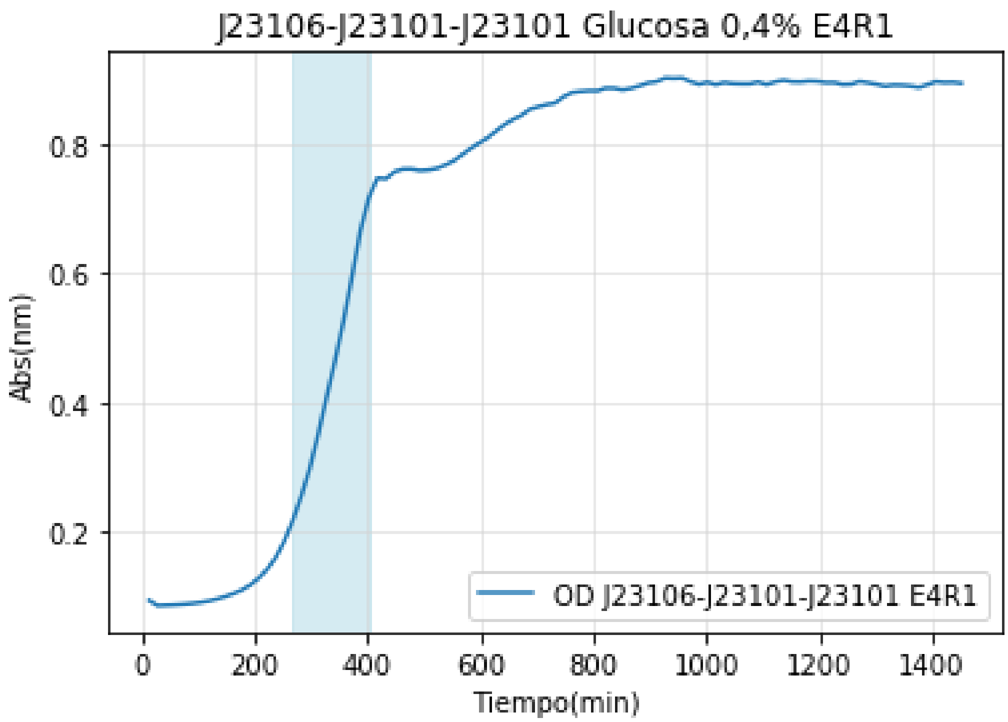
```
ttot7=tm7+extdp7
print('Tfinal=%e'%ttot7)

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

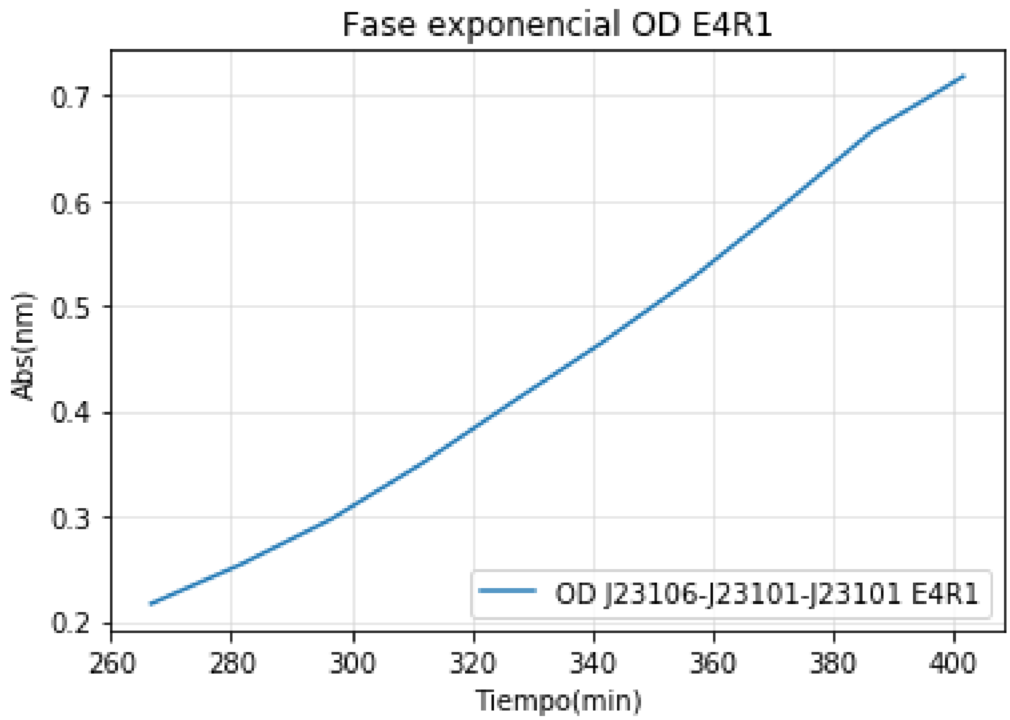
```
Min OD = 8.410000e-02
[  2.35417955e+00   1.08424351e-02   1.72836449e+02]
```



```
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[14]: <matplotlib.legend.Legend at 0x1b2050f8278>

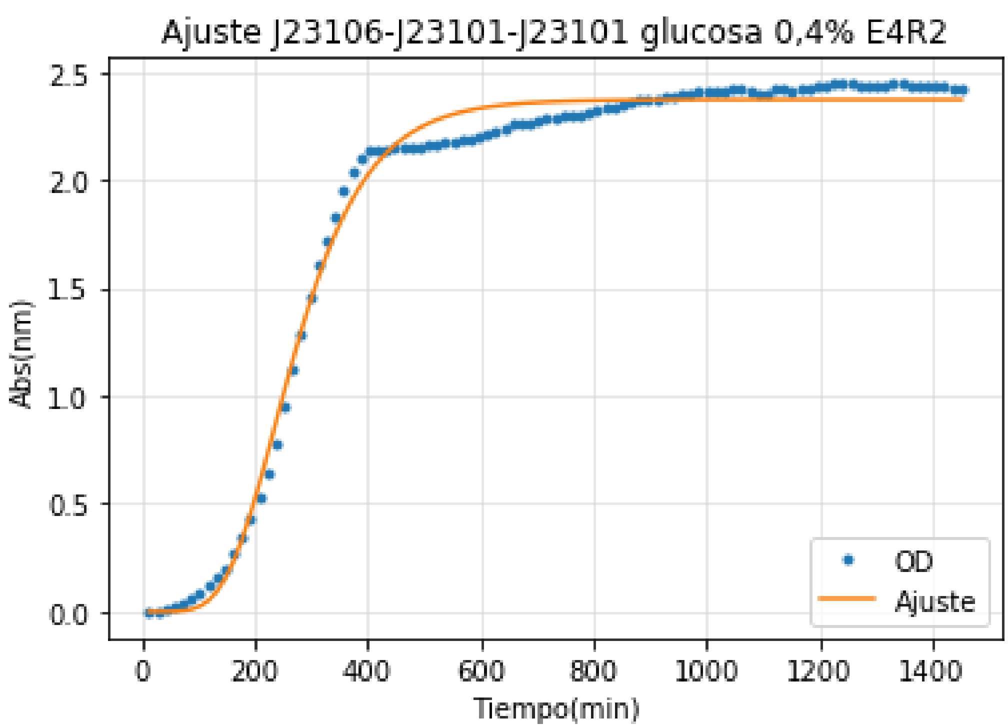




In [15]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 2
y8= np.log(od12212)-np.log(np.min(od12212))
print('Min OD = %e'%((np.min(od12212))))
evaly, params=Function_fit(tt,y8,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4%
A8= params[0]
um8=params[1]
l8=params[2]
print('A=%e'%(A8))
print('um=%e'%(um8))
print('l=%e'%(l8))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[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')
```

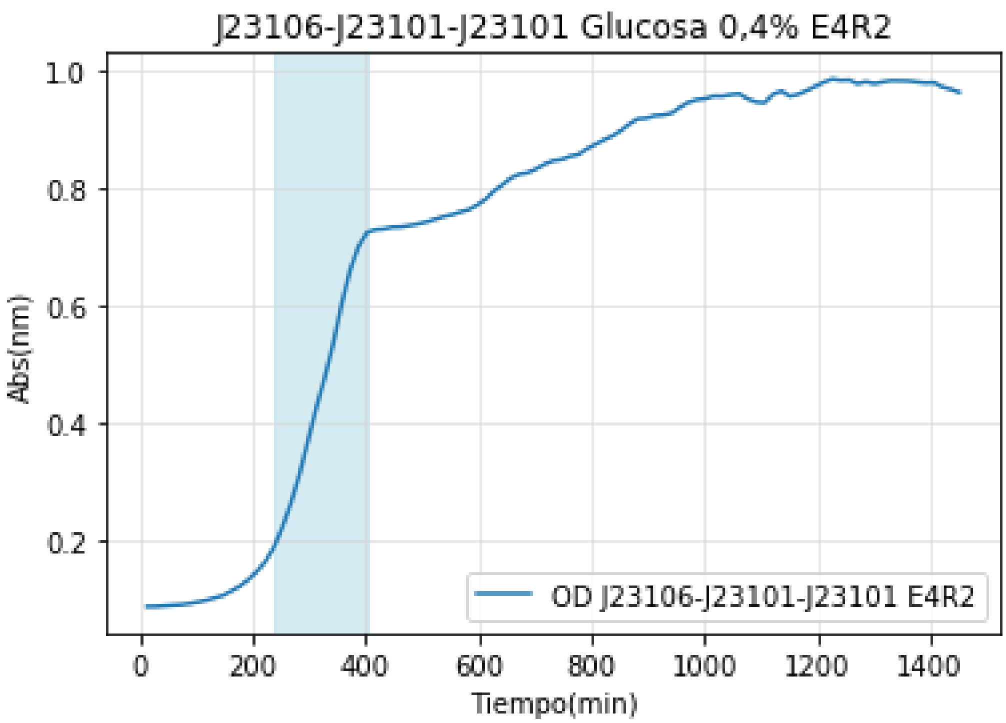
```
Min OD = 8.550000e-02
[  2.37271494e+00   9.82381102e-03   1.47415571e+02]
```

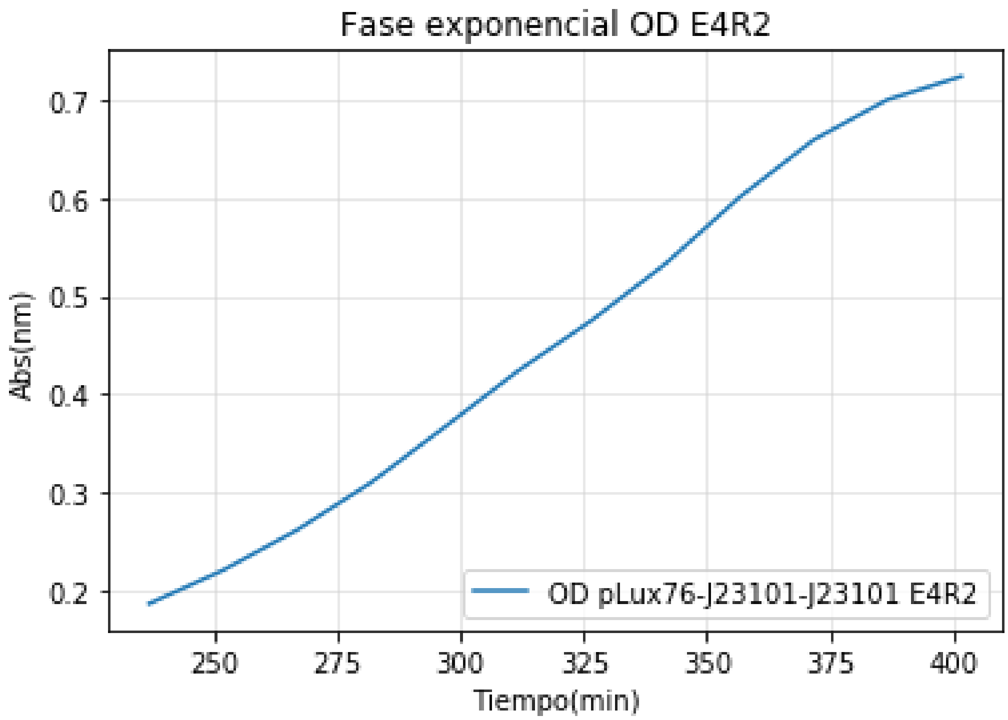



```
A=2.372715e+00
um=9.823811e-03
```

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

Out[15]: <matplotlib.legend.Legend at 0x1b20512e7f0>

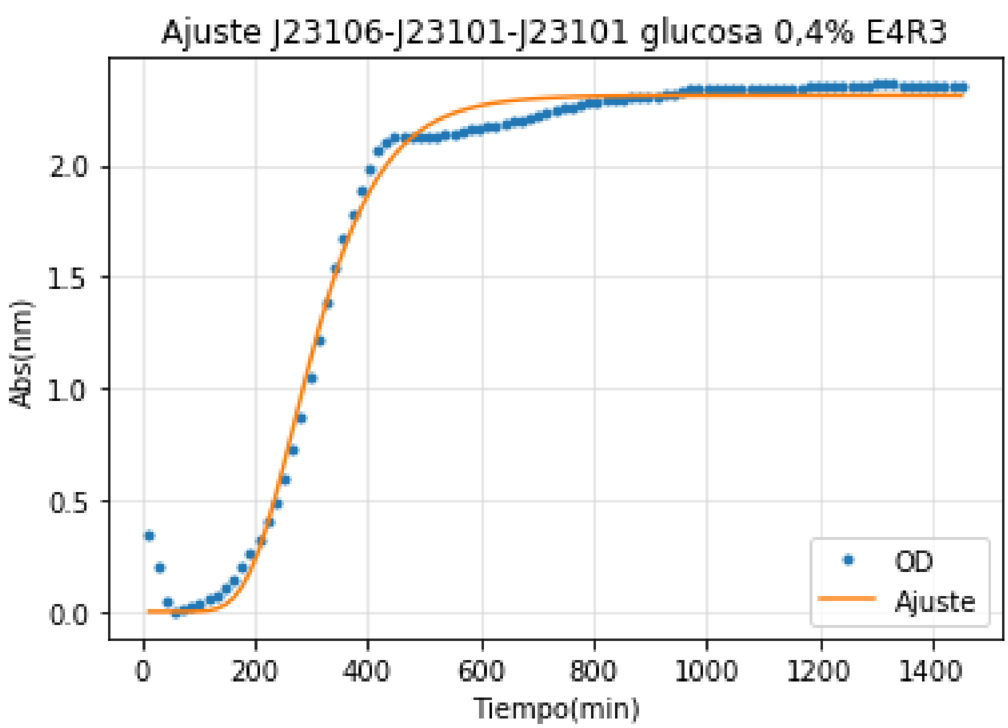




In [16]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glucosa rep 3
y9= np.log(od12213)-np.log(np.min(od12213))
print('Min OD = %e'%((np.min(od12213))))
evaly, params=Function_fit(tt,y9,0,-1,title = 'Ajuste J23106-J23101-J23101 glucosa 0,4%
A9= params[0]
um9=params[1]
l9=params[2]
print('A=%e'%(A9))
print('um=%e'%(um9))
print('l=%e'%(l9))

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

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

```python
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')
```

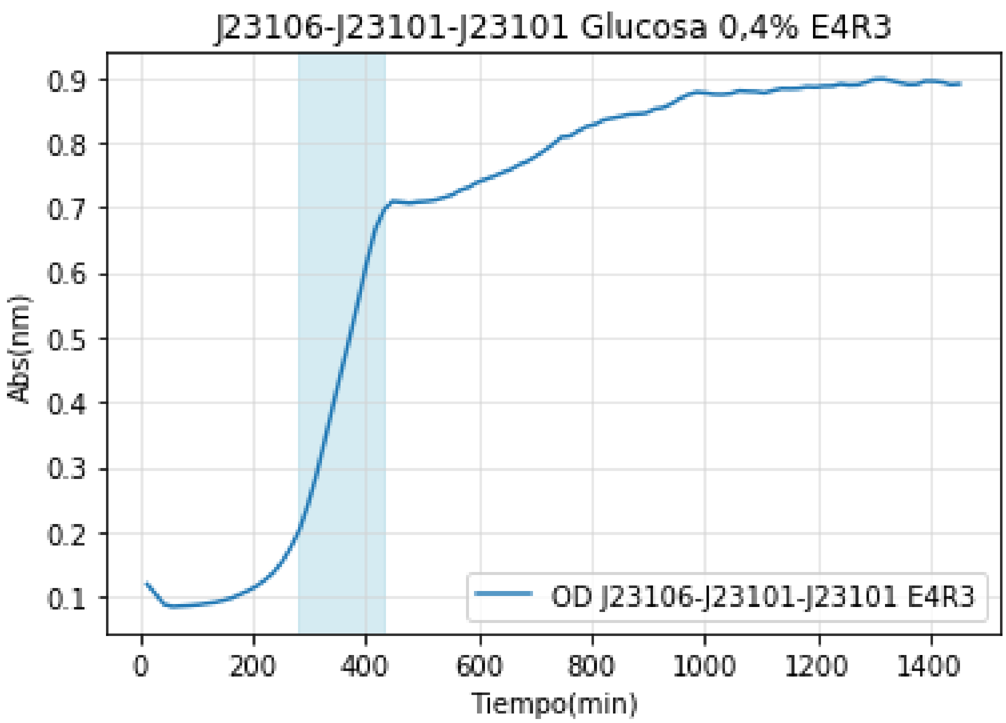
```
Min OD = 8.460000e-02
[  2.31183042e+00   1.00988565e-02   1.85588268e+02]
```

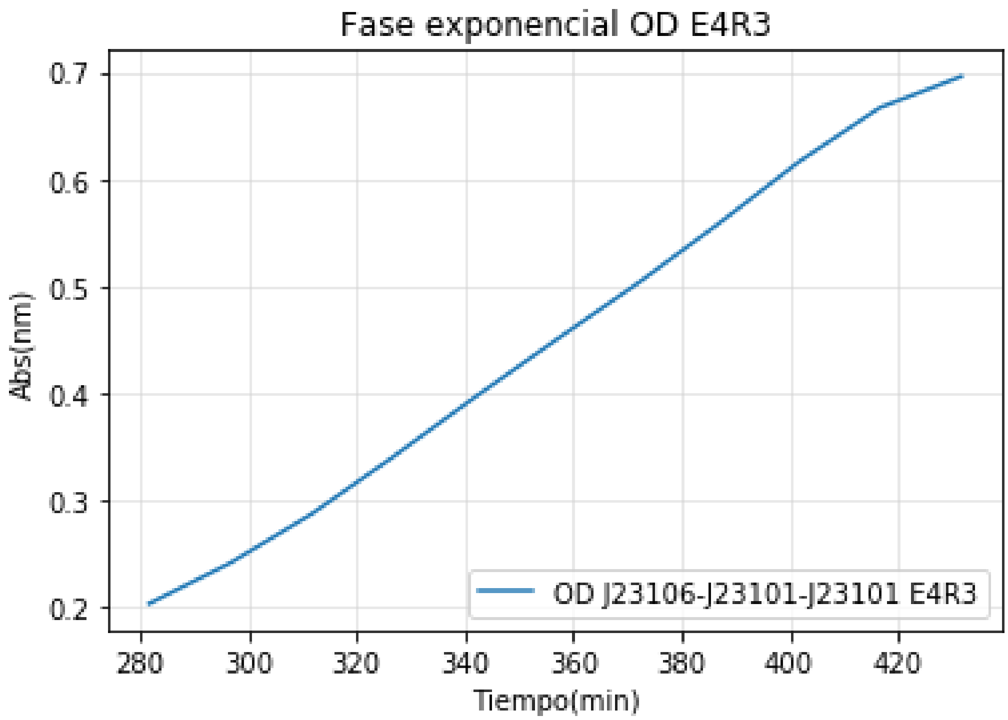



```
A=2.311830e+00
um=1.009886e-02
```

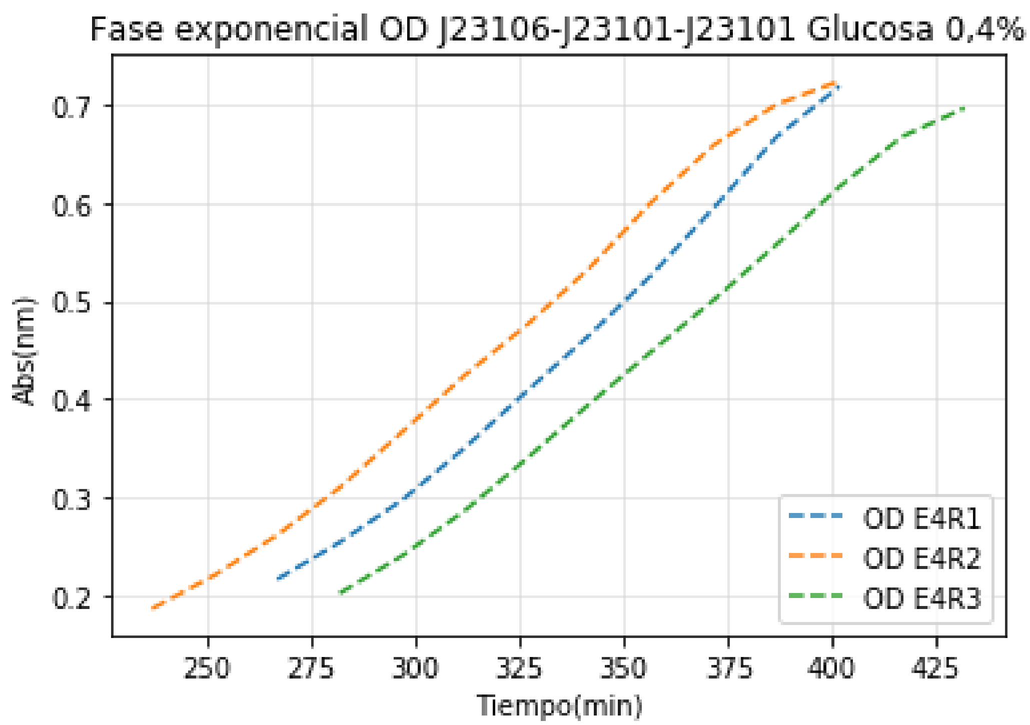
```
l=1.855883e+02
Tm=2.698032e+02
doubpe=6.863620e+01
ext=1.372724e+02
Tfinal=4.070756e+02
```

Out[16]: <matplotlib.legend.Legend at 0x1b205805e10>





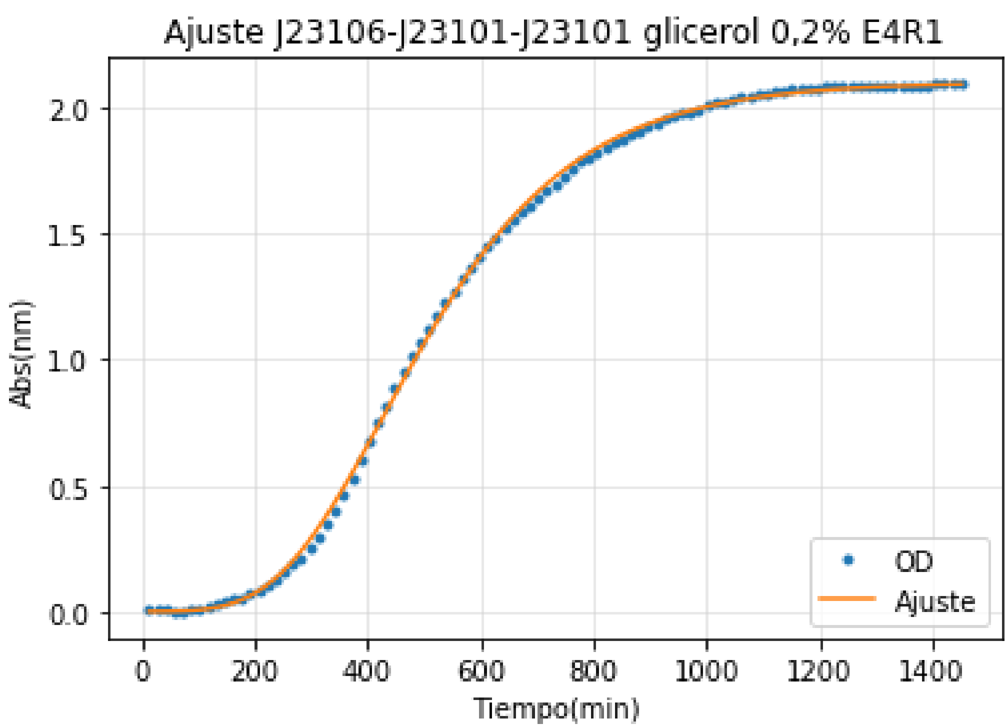

```
In [17]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23101-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:27],od12211[17:27],'--',label='OD E4R1')
         plt.plot(tt[15:27],od12212[15:27],'--',label='OD 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[17]: <matplotlib.legend.Legend at 0x1b20593a940>
```



In [18]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 1
y10= np.log(od1221g1)-np.log(np.min(od1221g1))
print('Min OD = %e'%((np.min(od1221g1))))
evaly, params=Function_fit(tt,y10,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A10= params[0]
um10=params[1]
l10=params[2]
print('A=%e'%(A10))
print('um=%e'%(um10))
print('l=%e'%(l10))

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

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

```
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')
```

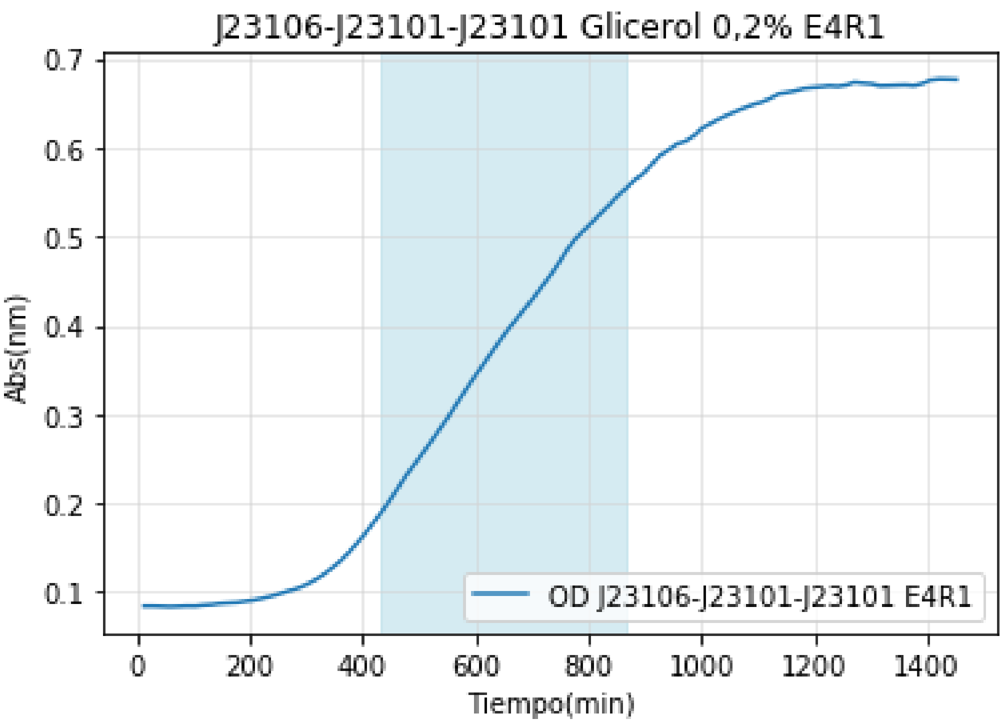
```
Min OD = 8.360000e-02
[  2.10005866e+00   4.10748730e-03   2.38628147e+02]
```

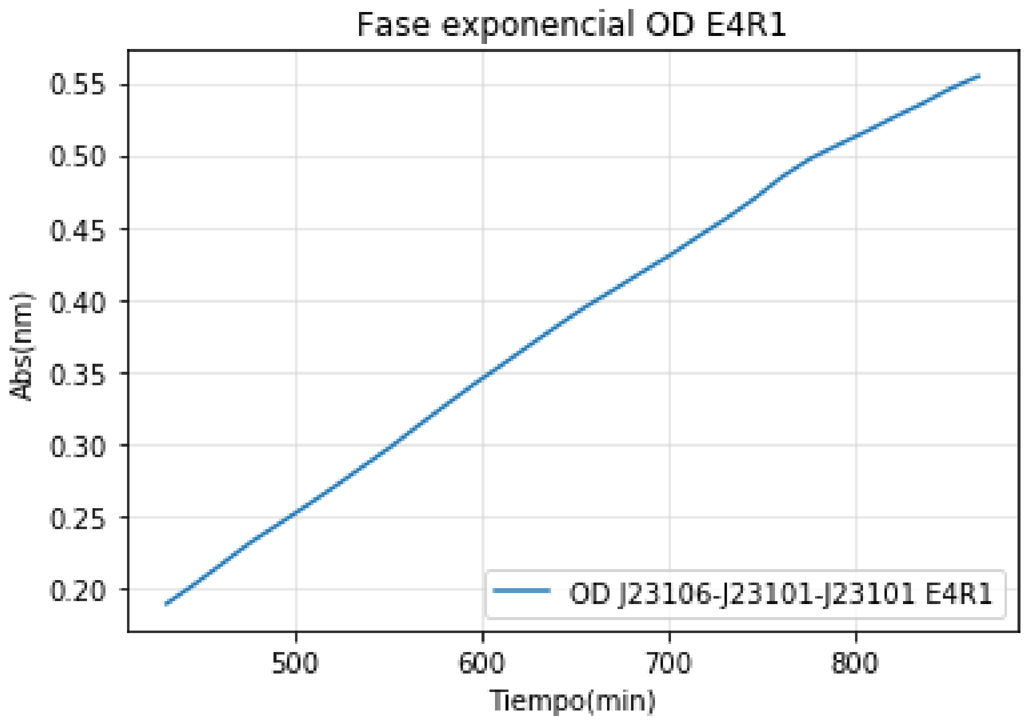



```
A=2.100059e+00
um=4.107487e-03
```

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

```
Out[18]: <matplotlib.legend.Legend at 0x1b205acdc18>
```

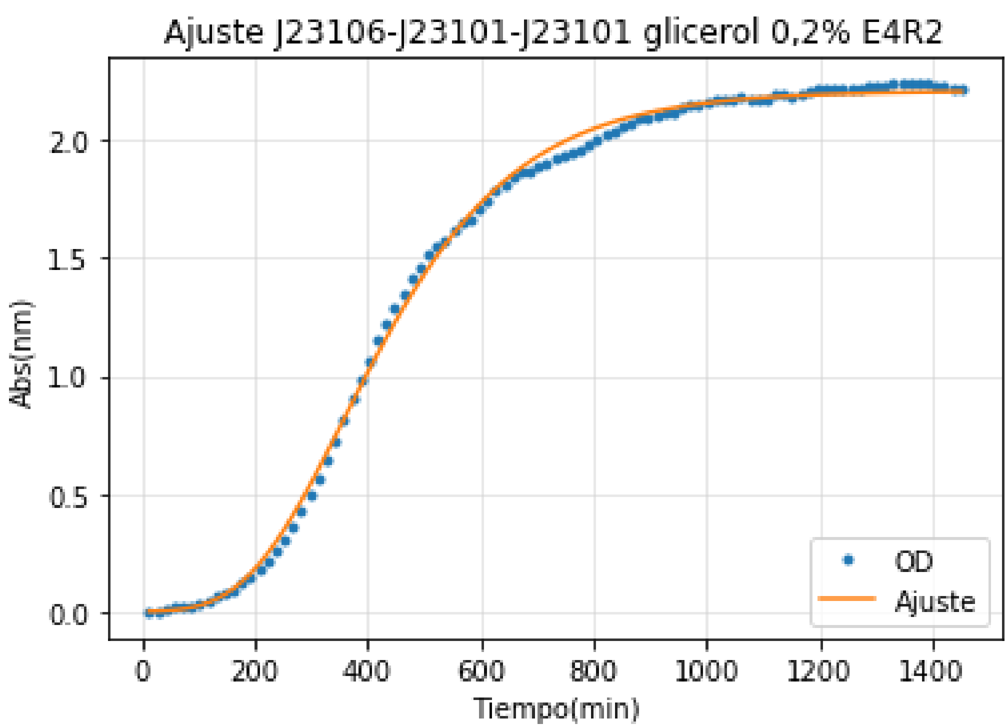




In [19]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 2
y11= np.log(od1221g2)-np.log(np.min(od1221g2))
print('Min OD = %e'%((np.min(od1221g2))))
evaly, params=Function_fit(tt,y11,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A11= params[0]
um11=params[1]
l11=params[2]
print('A=%e'%(A11))
print('um=%e'%(um11))
print('l=%e'%(l11))

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

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

```python
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')
```

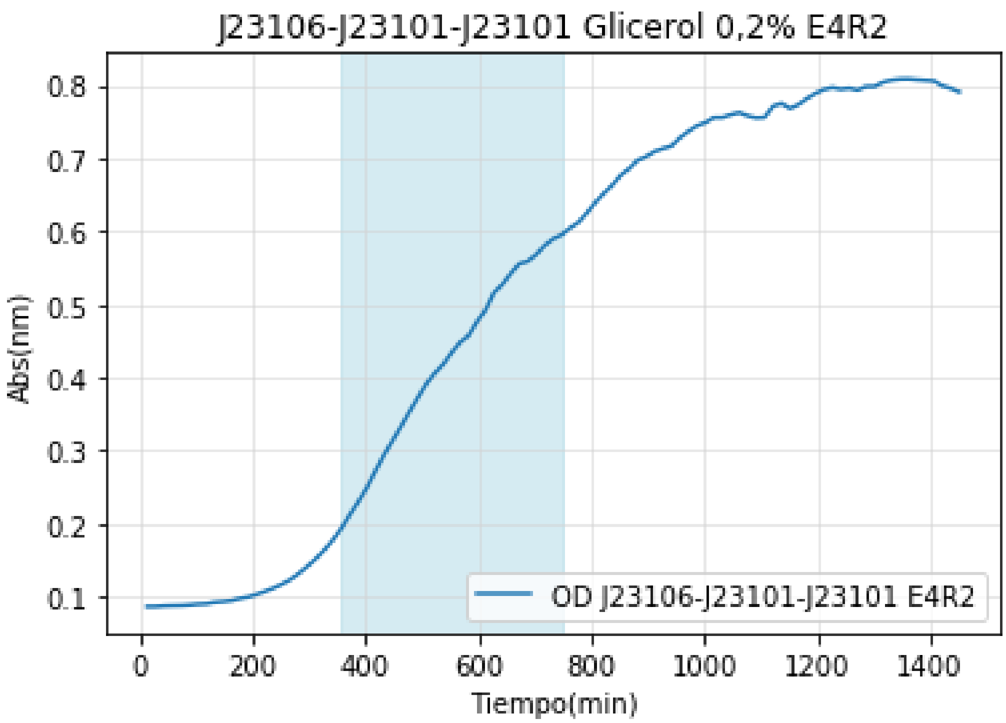
```
Min OD = 8.630000e-02
[  2.20913289e+00   4.73407774e-03   1.84774865e+02]
```

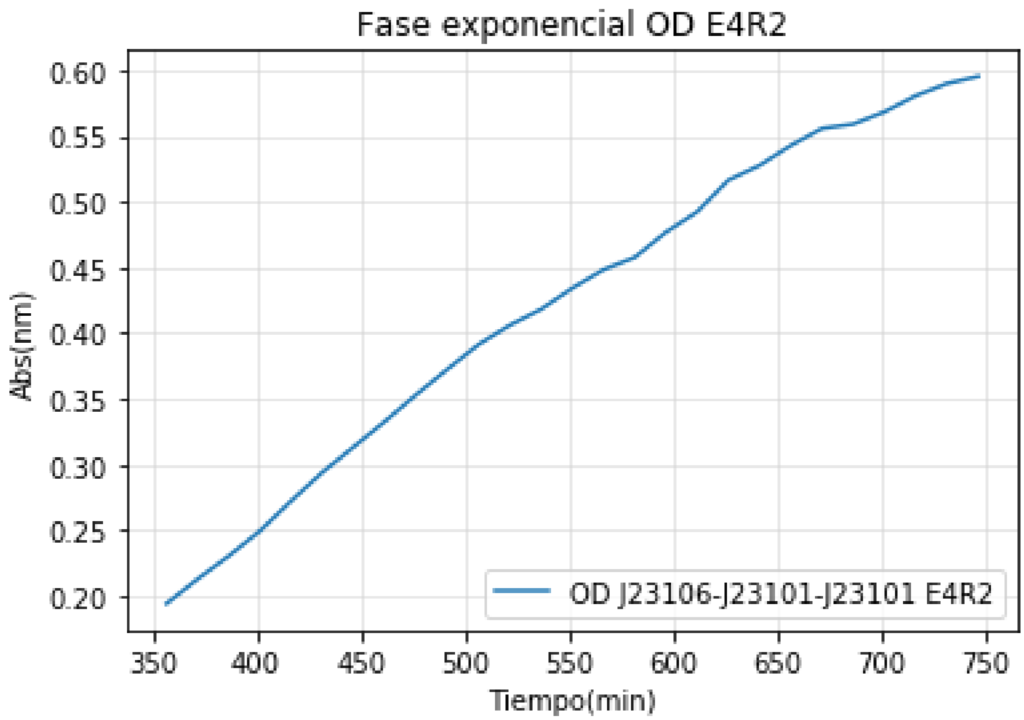



```
A=2.209133e+00
um=4.734078e-03
```

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

```
Out[19]: <matplotlib.legend.Legend at 0x1b203959eb8>
```

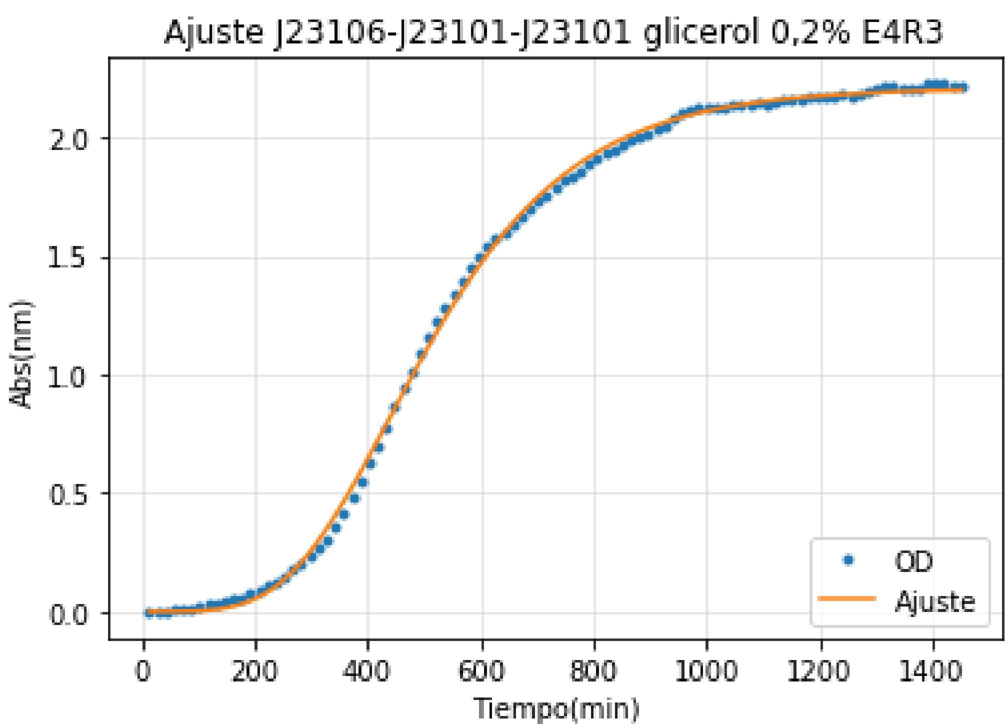




In [20]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-std-std glicerol rep 3
y12= np.log(od1221g3)-np.log(np.min(od1221g3))
print('Min OD = %e'%((np.min(od1221g3))))
evaly, params=Function_fit(tt,y12,0,-1,title = 'Ajuste J23106-J23101-J23101 glicerol 0,
A12= params[0]
um12=params[1]
l12=params[2]
print('A=%e'%(A12))
print('um=%e'%(um12))
print('l=%e'%(l12))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[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')
```

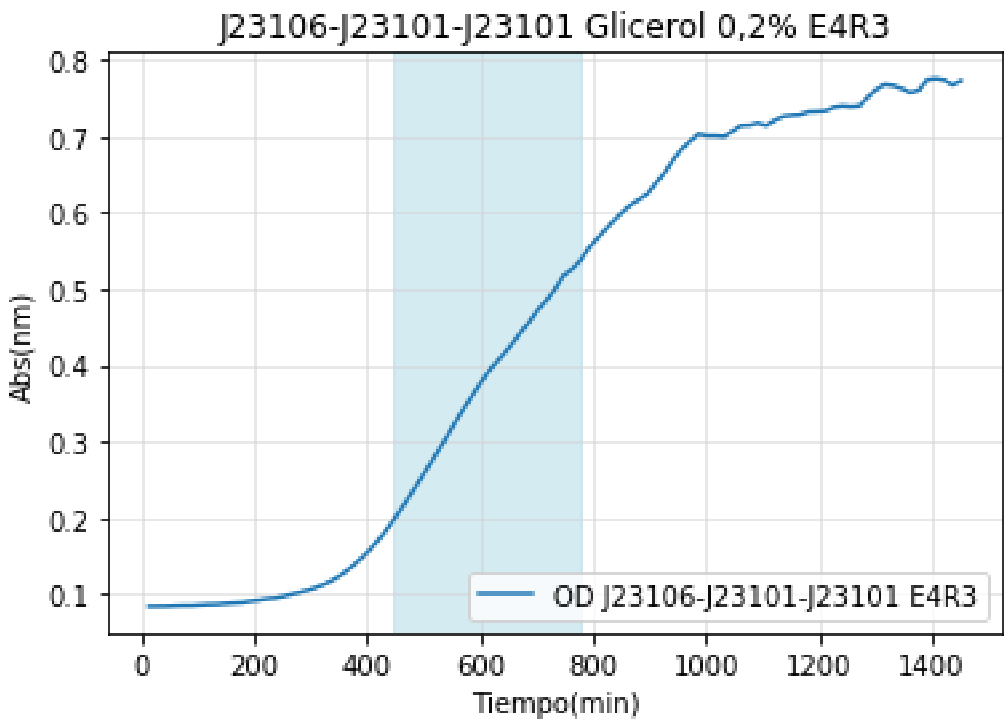
```
Min OD = 8.390000e-02
[  2.20537307e+00   4.48287691e-03   2.56148136e+02]
```

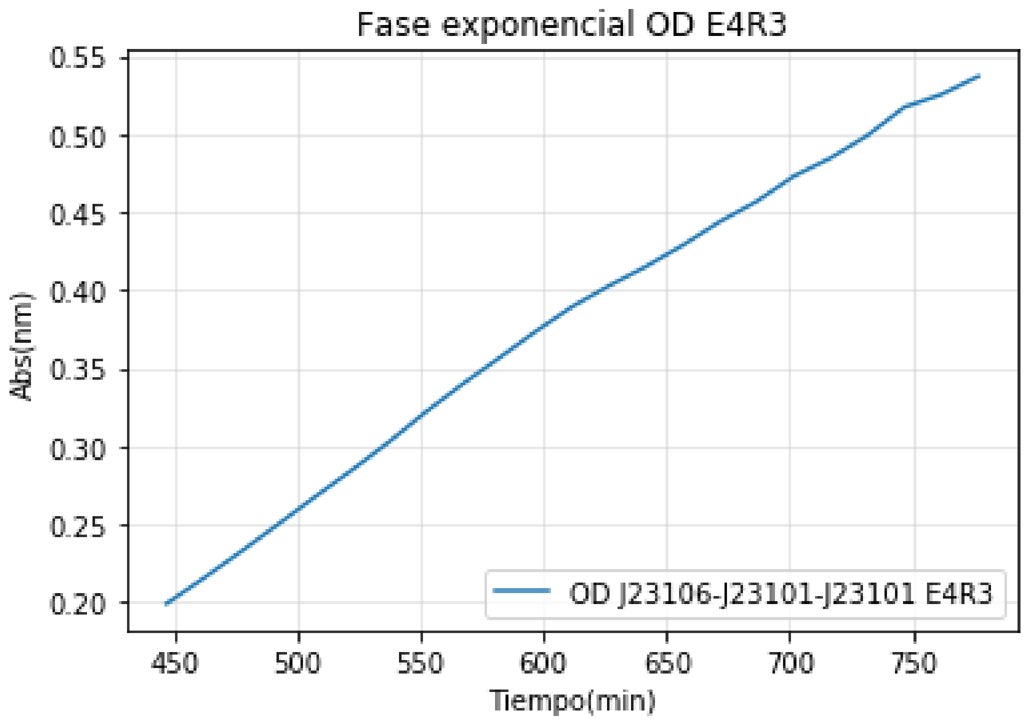



```
A=2.205373e+00
um=4.482877e-03
```

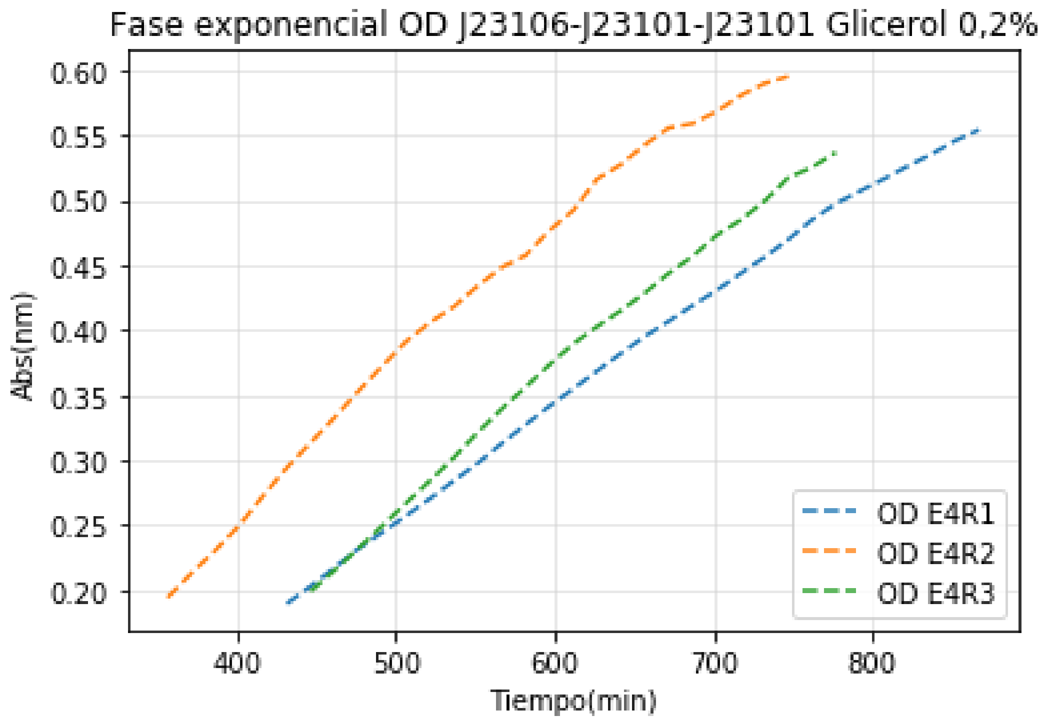
```
l=2.561481e+02
Tm=4.371282e+02
doubpe=1.546211e+02
ext=3.092421e+02
Tfinal=7.463703e+02
```

Out[20]: <matplotlib.legend.Legend at 0x1b20572a2e8>

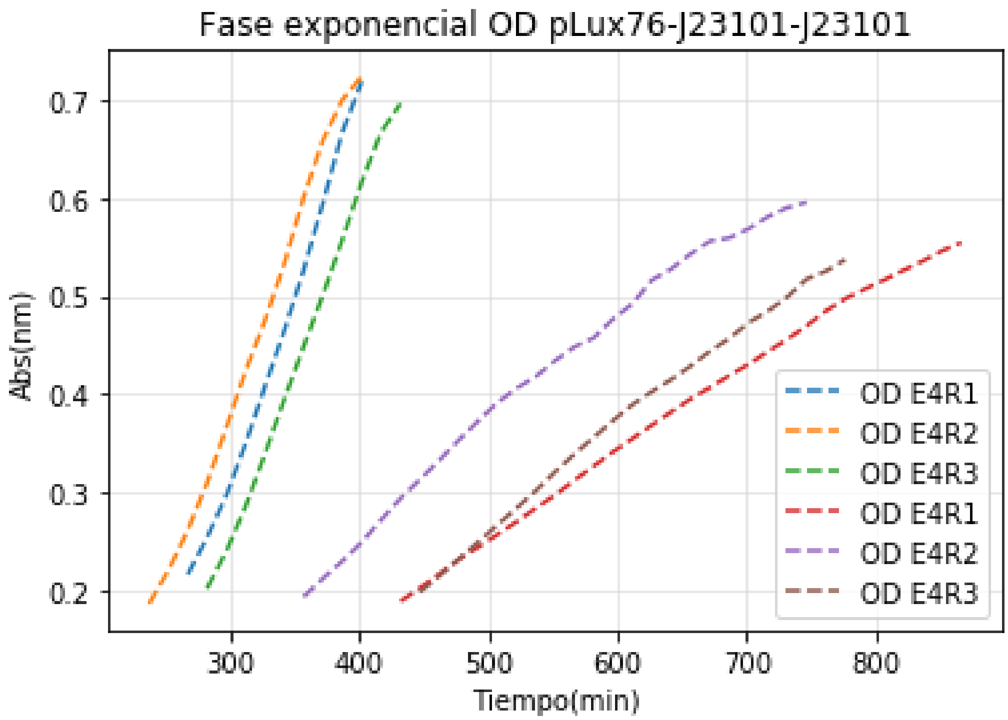





```
In [21]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23101-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[28:58],od1221g1[28:58],'--',label='OD E4R1')
         plt.plot(tt[23:50],od1221g2[23:50],'--',label='OD 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[21]: <matplotlib.legend.Legend at 0x1b205c00b70>
```



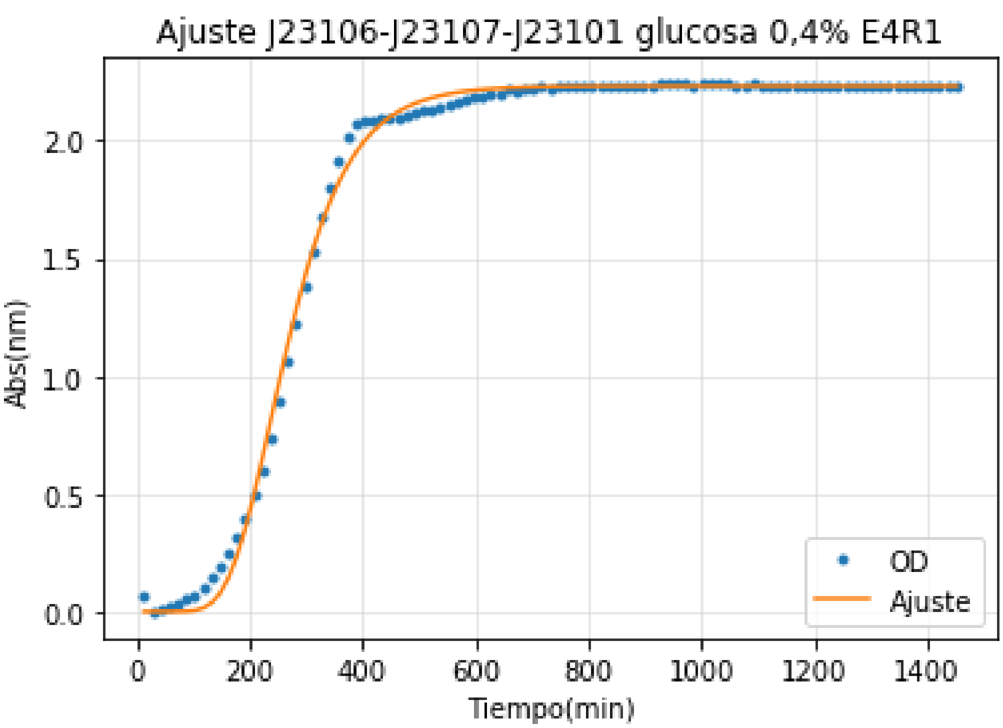

```
In [22]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD pLux76-J23101-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:27],od12211[17:27],'--',label='OD E4R1')
         plt.plot(tt[15:27],od12212[15:27],'--',label='OD 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[22]: <matplotlib.legend.Legend at 0x1b205cccba8>
```




```
In [23]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-107-std glucosa rep 1
         y13= np.log(od12231)-np.log(np.min(od12231))
         print('Min OD = %e'%((np.min(od12231))))
         evaly, params=Function_fit(tt,y13,0,-1,title = 'Ajuste J23106-J23107-J23101 glucosa 0,4
         A13 = params[0]
         um13=params[1]
         l13=params[2]
         print('A=%e'%(A13))
         print('um=%e'%(um13))
         print('l=%e'%(l13))

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

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

```
y2=tt[25]
plt.figure()
plt.title('J23106-J23107-J23101 Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12231,label='OD J23106-J23107-J23101 E4R1 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12231[16:26],label='OD J23106-J23107-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

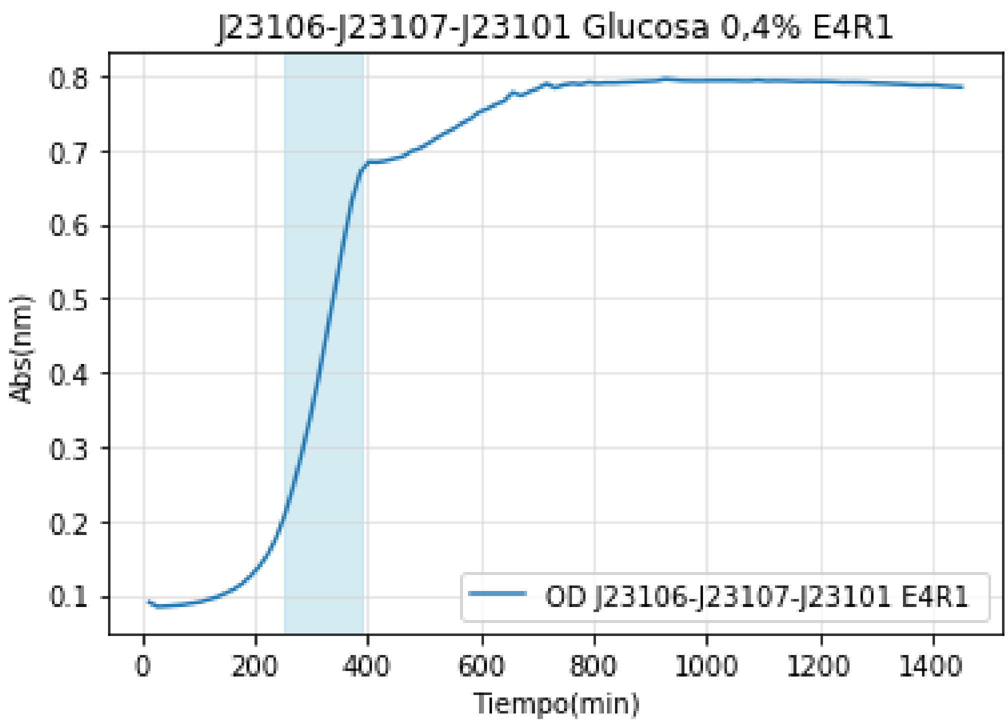
```
Min OD = 8.460000e-02
[  2.23226361e+00   1.09506306e-02   1.62099465e+02]
```

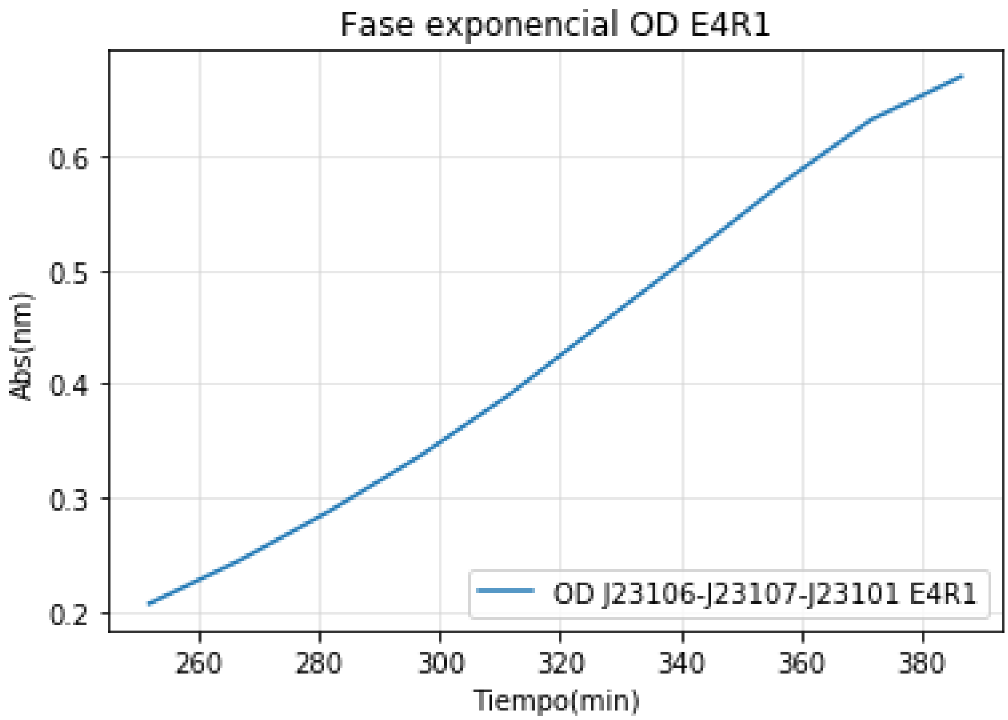



```
A=2.232264e+00
um=1.095063e-02
```

```
l=1.620995e+02
Tm=2.370909e+02
doubpe=6.329747e+01
ext=1.265949e+02
Tfinal=3.636859e+02
```

Out[23]: `<matplotlib.legend.Legend at 0x1b205b38a58>`

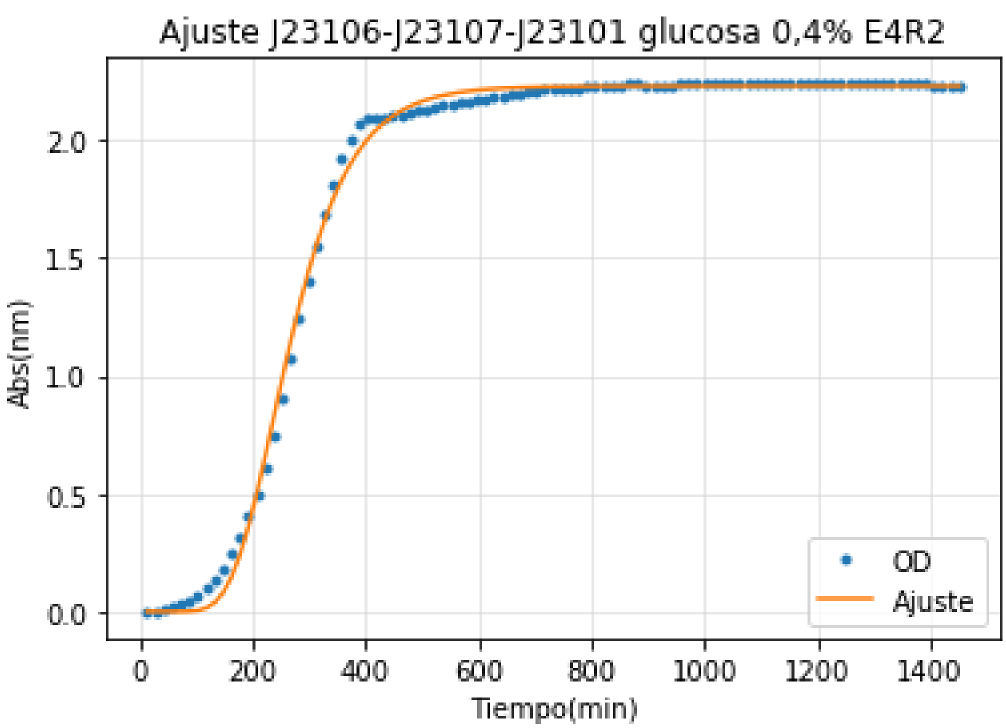




In [24]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-107-std glucosa rep 2
y14= np.log(od12232)-np.log(np.min(od12232))
print('Min OD = %e'%((np.min(od12232))))
evaly, params=Function_fit(tt,y14,0,-1,title = 'Ajuste J23106-J23107-J23101 glucosa 0,4
A14= params[0]
um14=params[1]
l14=params[2]
print('A=%e'%(A14))
print('um=%e'%(um14))
print('l=%e'%(l14))

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

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

```
y2=tt[25]
plt.figure()
plt.title('J23106-J23107-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12232,label='OD J23106-J23107-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[15:26],od12232[15:26],label='OD J23106-J23107-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

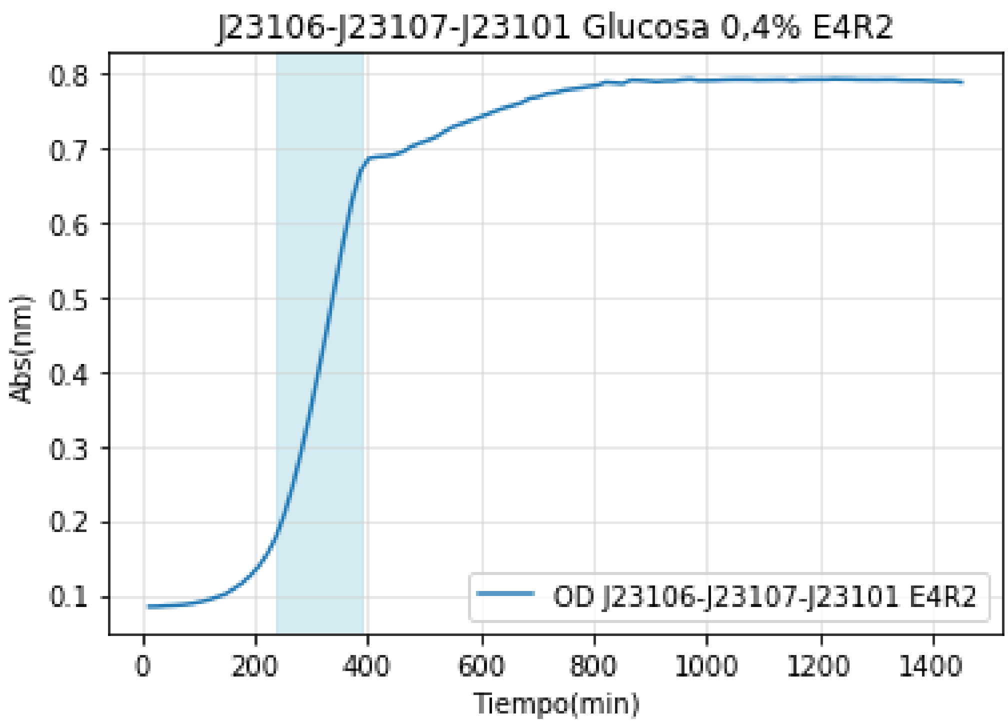
```
Min OD = 8.450000e-02
[  2.22943164e+00   1.10186847e-02   1.61533256e+02]
```

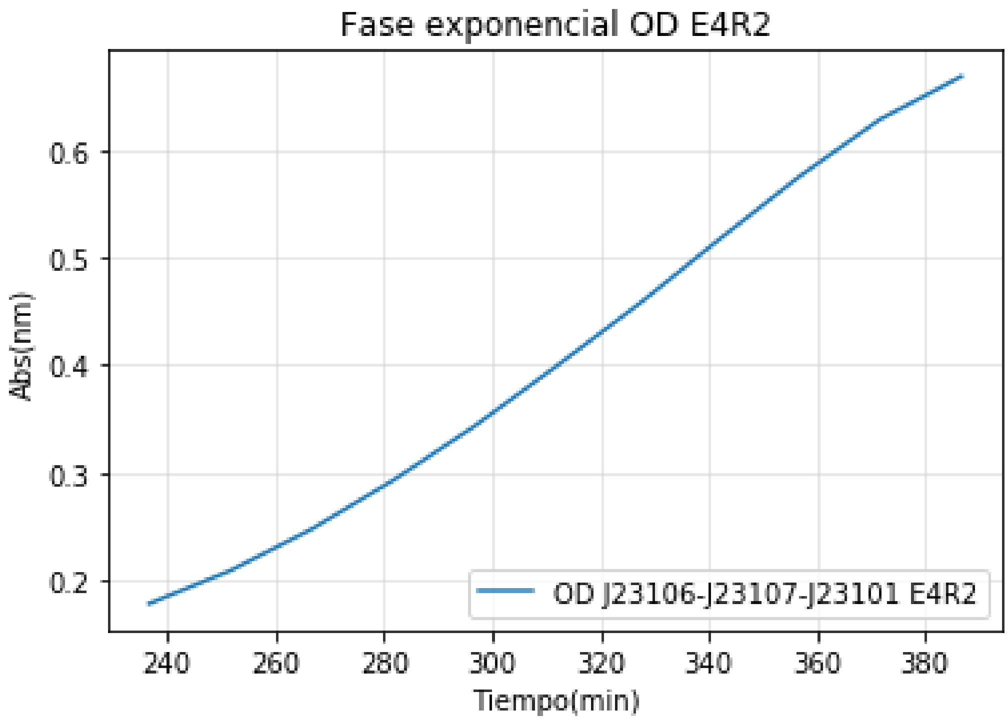



```
A=2.229432e+00
um=1.101868e-02
```

```
l=1.615333e+02
Tm=2.359670e+02
doubpe=6.290653e+01
ext=1.258131e+02
Tfinal=3.617801e+02
```

Out[24]: <matplotlib.legend.Legend at 0x1b205873cc0>

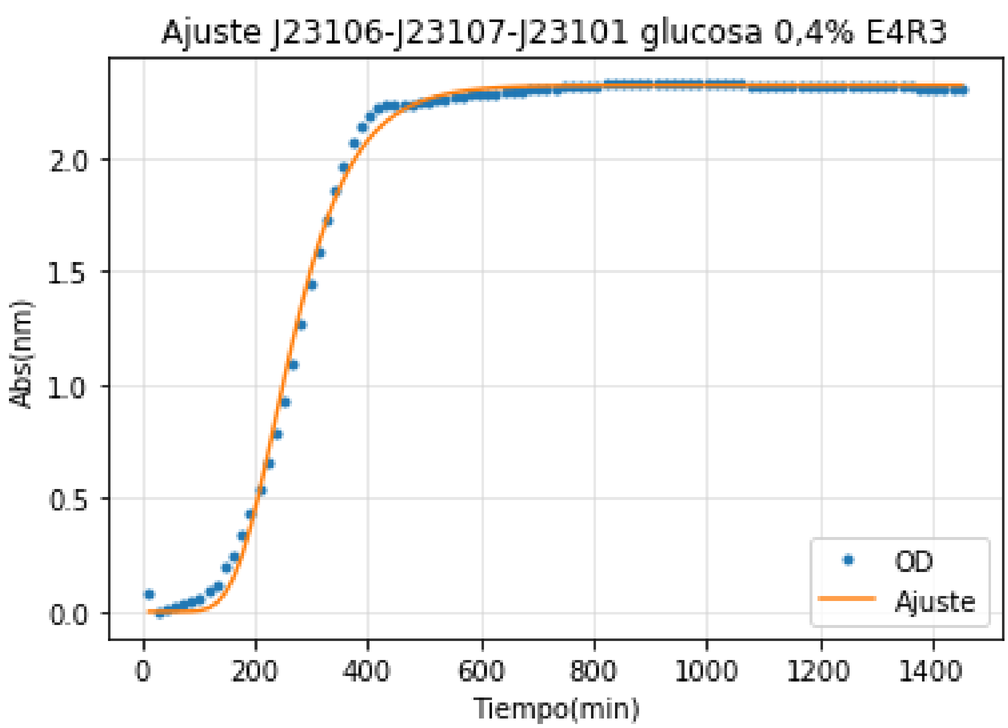




In [25]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-107-std glucosa rep 3
y15= np.log(od12233)-np.log(np.min(od12233))
print('Min OD = %e'%((np.min(od12233))))
evaly, params=Function_fit(tt,y15,0,-1,title = 'Ajuste J23106-J23107-J23101 glucosa 0,4
A15= params[0]
um15=params[1]
l15=params[2]
print('A=%e'%(A15))
print('um=%e'%(um15))
print('l=%e'%(l15))

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

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

```
y2=tt[25]
plt.figure()
plt.title('J23106-J23107-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12233,label='OD J23106-J23107-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[15:26],od12233[15:26],label='OD J23106-J23107-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

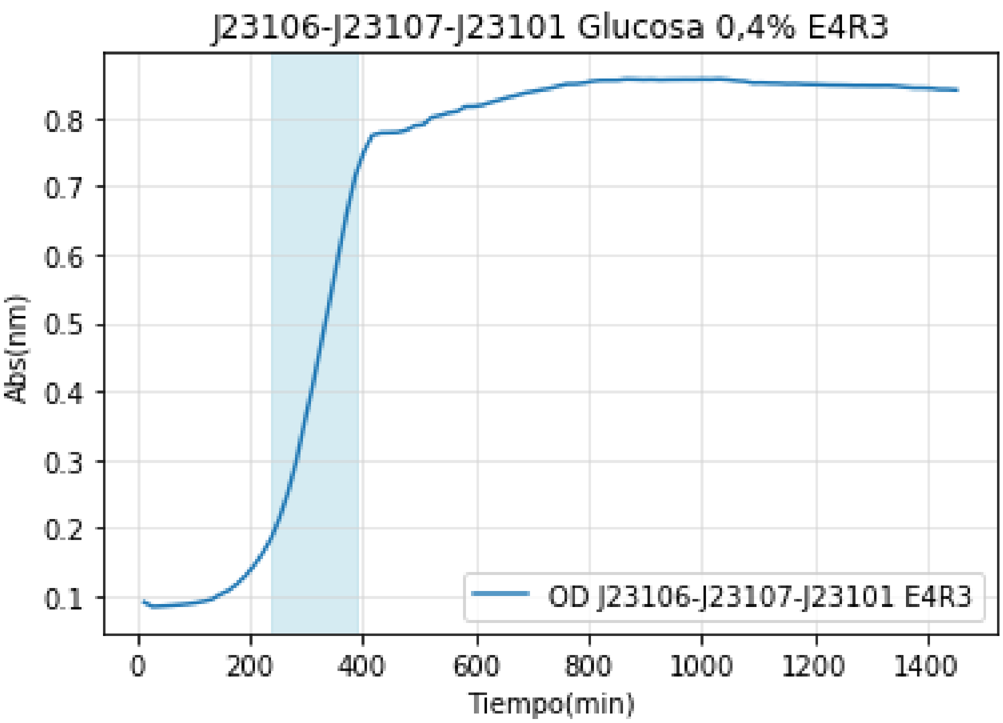
```
Min OD = 8.380000e-02
[  2.32067118e+00   1.15080749e-02   1.62138074e+02]
```

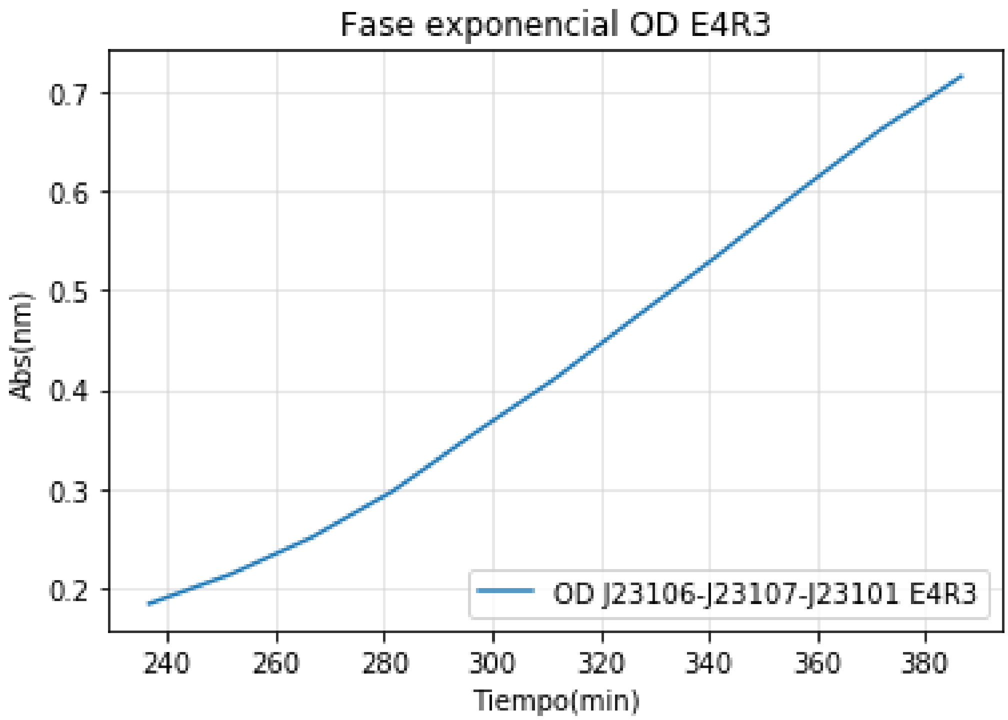



```
A=2.320671e+00
um=1.150807e-02
```

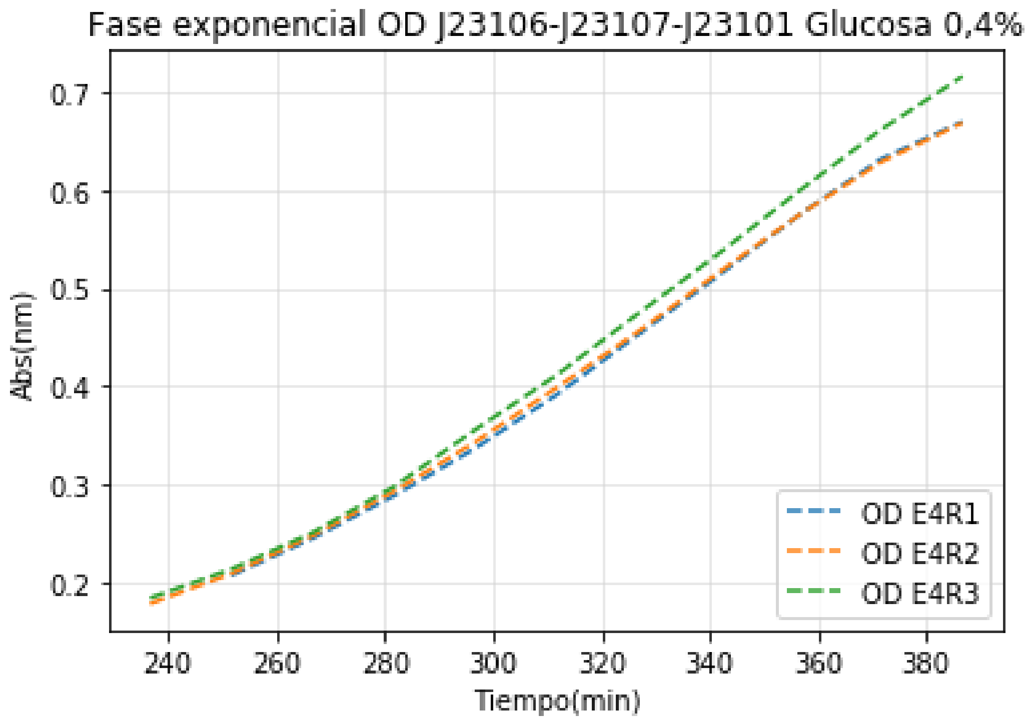
```
l=1.621381e+02
Tm=2.363231e+02
doubpe=6.023138e+01
ext=1.204628e+02
Tfinal=3.567859e+02
```

```
Out[25]: <matplotlib.legend.Legend at 0x1b205e39940>
```





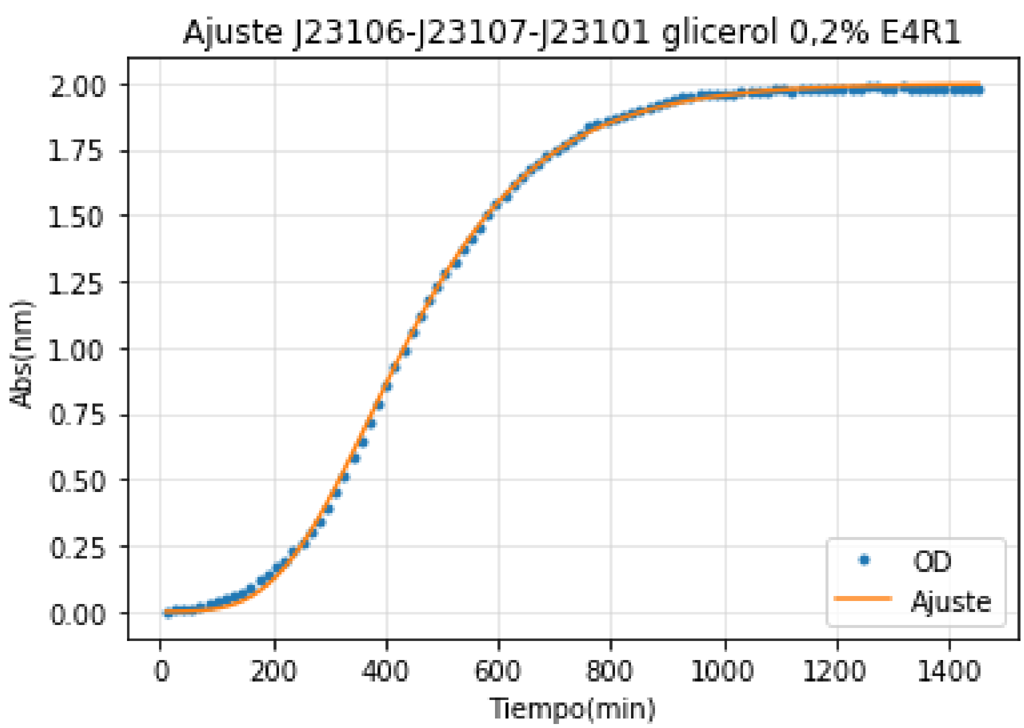

```
In [26]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23107-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],od12231[16:26],'--',label='OD E4R1')
         plt.plot(tt[15:26],od12232[15:26],'--',label='OD E4R2')
         plt.plot(tt[15:26],od12233[15:26],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[26]: <matplotlib.legend.Legend at 0x1b206f51f60>
```



In [27]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-107-std glicerol rep 1
y16= np.log(od1223g1)-np.log(np.min(od1223g1))
print('Min OD = %e'%((np.min(od1223g1))))
evaly, params=Function_fit(tt,y16,0,-1,title = 'Ajuste J23106-J23107-J23101 glicerol 0,
A16= params[0]
um16=params[1]
l16=params[2]
print('A=%e'%(A16))
print('um=%e'%(um16))
print('l=%e'%(l16))

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

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

```
y2=tt[52]
plt.figure()
plt.title('J23106-J23107-J23101 Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1223g1,label='OD J23106-J23107-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[25:53],od1223g1[25:53],label='OD J23106-J23107-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

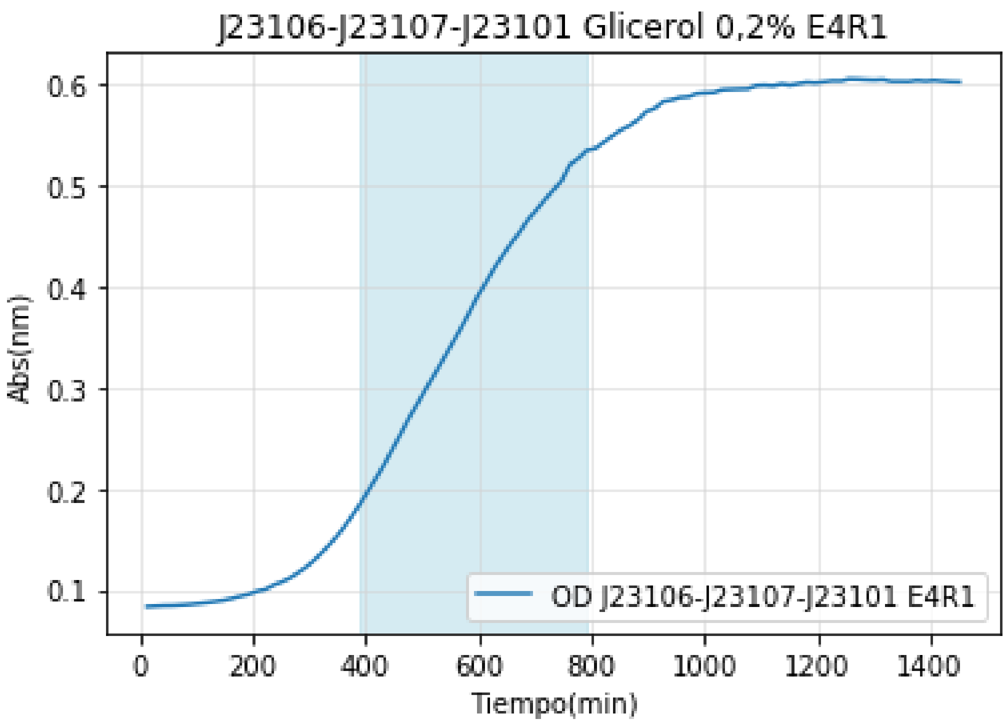
```
Min OD = 8.320000e-02
[  1.99917733e+00   4.41715628e-03   2.06073712e+02]
```

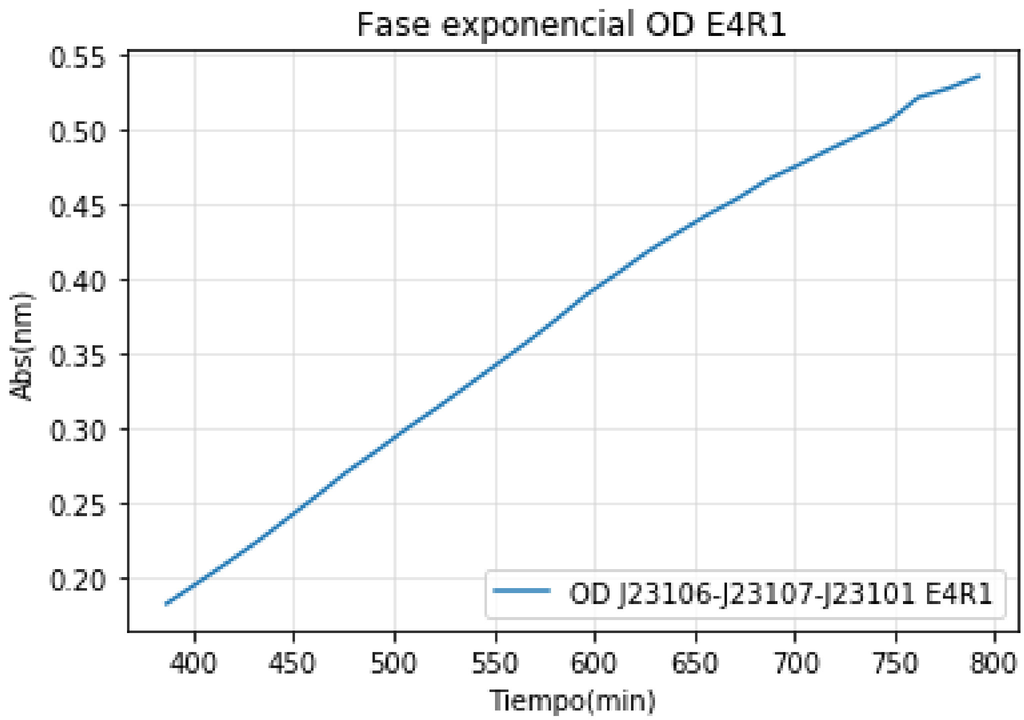



```
A=1.999177e+00
um=4.417156e-03
```

```
l=2.060737e+02
Tm=3.725736e+02
doubpe=1.569216e+02
ext=3.923040e+02
Tfinal=7.648776e+02
```

```
Out[27]: <matplotlib.legend.Legend at 0x1b205ef9518>
```

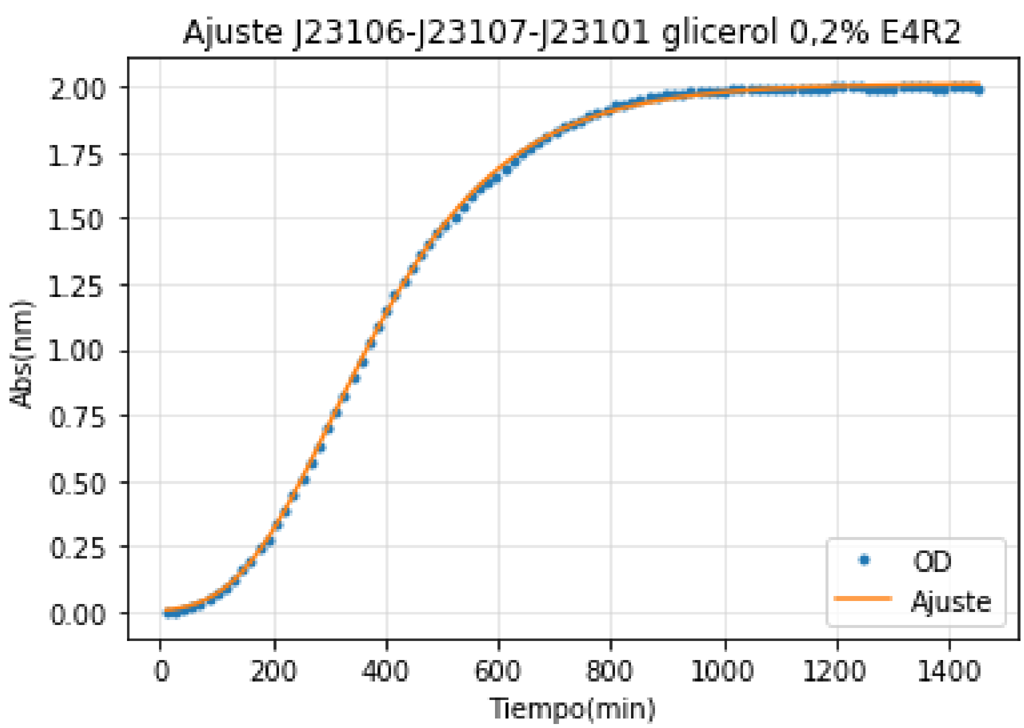




In [28]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-107-std glicerol rep 2
y17= np.log(od1223g2)-np.log(np.min(od1223g2))
print('Min OD = %e'%((np.min(od1223g2))))
evaly, params=Function_fit(tt,y17,0,-1,title = 'Ajuste J23106-J23107-J23101 glicerol 0,
A17= params[0]
um17=params[1]
l17=params[2]
print('A=%e'%(A17))
print('um=%e'%(um17))
print('l=%e'%(l17))

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

#Delimitación fase exponencial en grafico con OD/tiempo
y1=tt[20]
```

```
y2=tt[48]
plt.figure()
plt.title('J23106-J23107-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1223g2,label='OD J23106-J23107-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[20:49],od1223g2[20:49],label='OD J23106-J23107-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

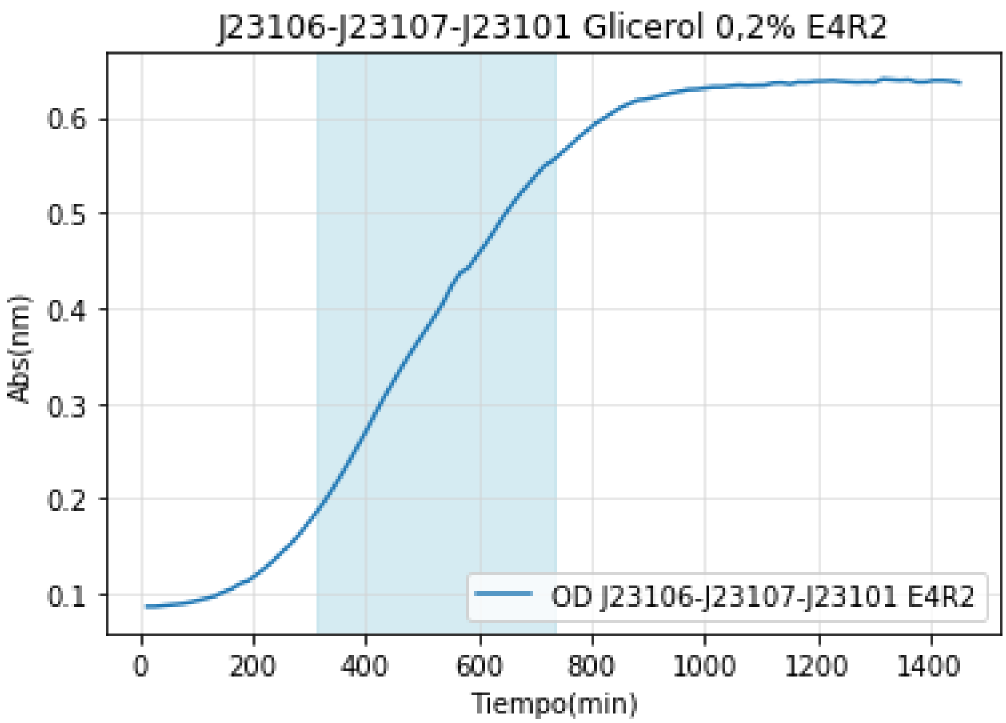
```
Min OD = 8.660000e-02
[  2.01084440e+00   4.37332701e-03   1.37075188e+02]
```

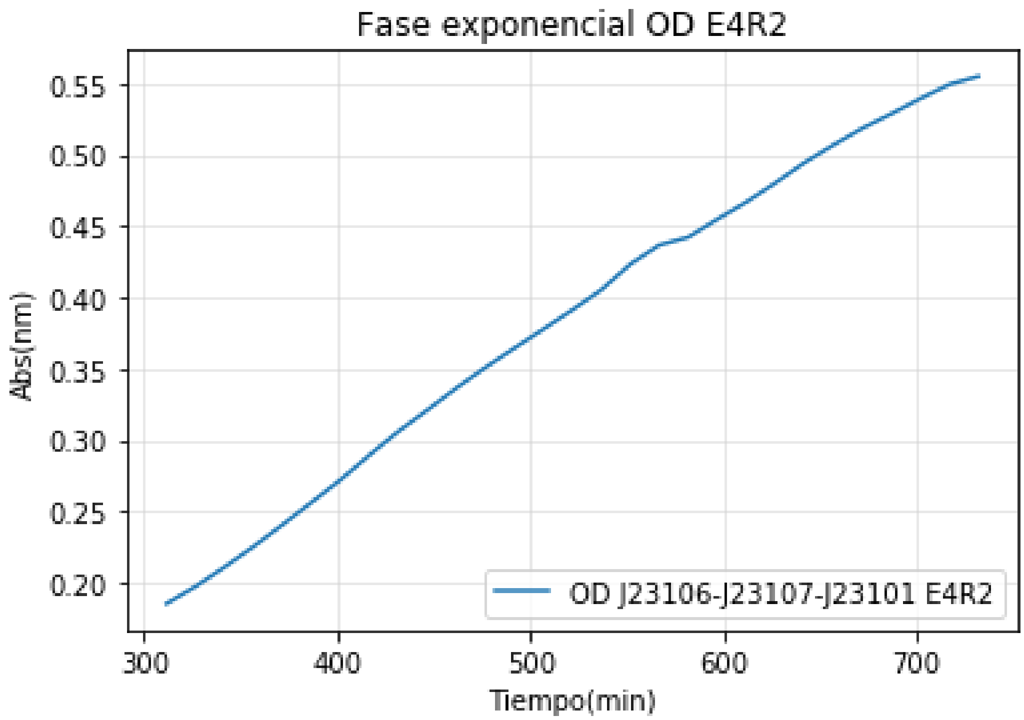



```
A=2.010844e+00
um=4.373327e-03
```

```
l=1.370752e+02
Tm=3.062252e+02
doubpe=1.584942e+02
ext=3.962356e+02
Tfinal=7.024608e+02
```

Out[28]: <matplotlib.legend.Legend at 0x1b203e66d30>

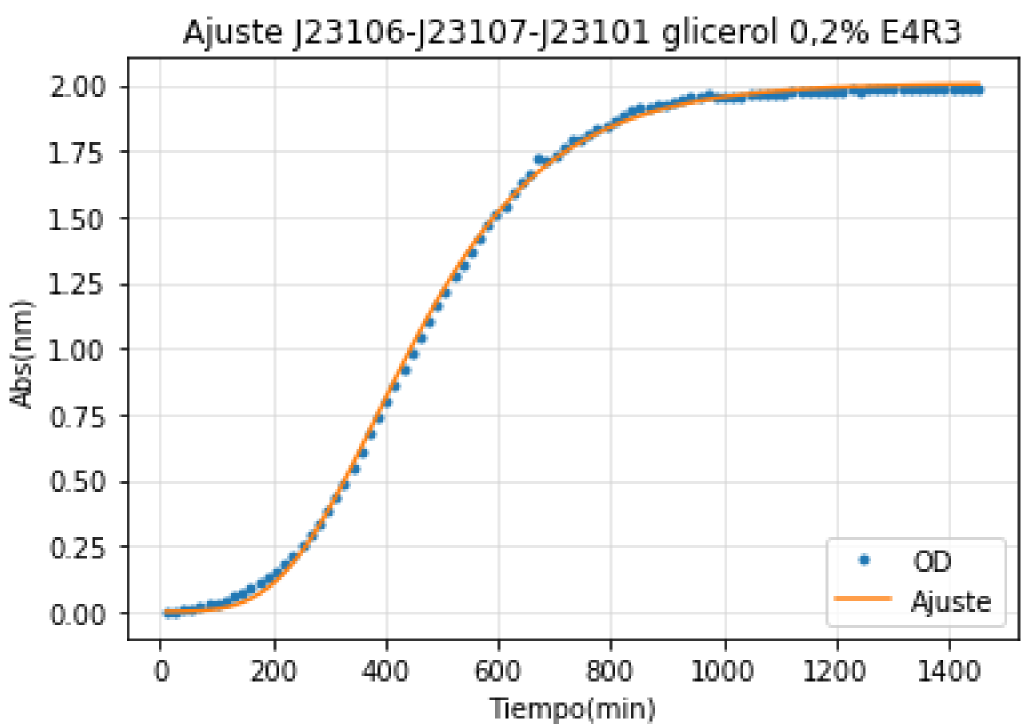




In [29]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-107-std glicerol rep 3
y18= np.log(od1223g3)-np.log(np.min(od1223g3))
print('Min OD = %e'%((np.min(od1223g3))))
evaly, params=Function_fit(tt,y18,0,-1,title = 'Ajuste J23106-J23107-J23101 glicerol 0,
A18= params[0]
um18=params[1]
l18=params[2]
print('A=%e'%(A18))
print('um=%e'%(um18))
print('l=%e'%(l18))

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

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

```
y2=tt[52]
plt.figure()
plt.title('J23106-J23107-J23101 Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1223g3,label='OD J23106-J23107-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[25:54],od1223g3[25:54],label='OD J23106-J23107-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

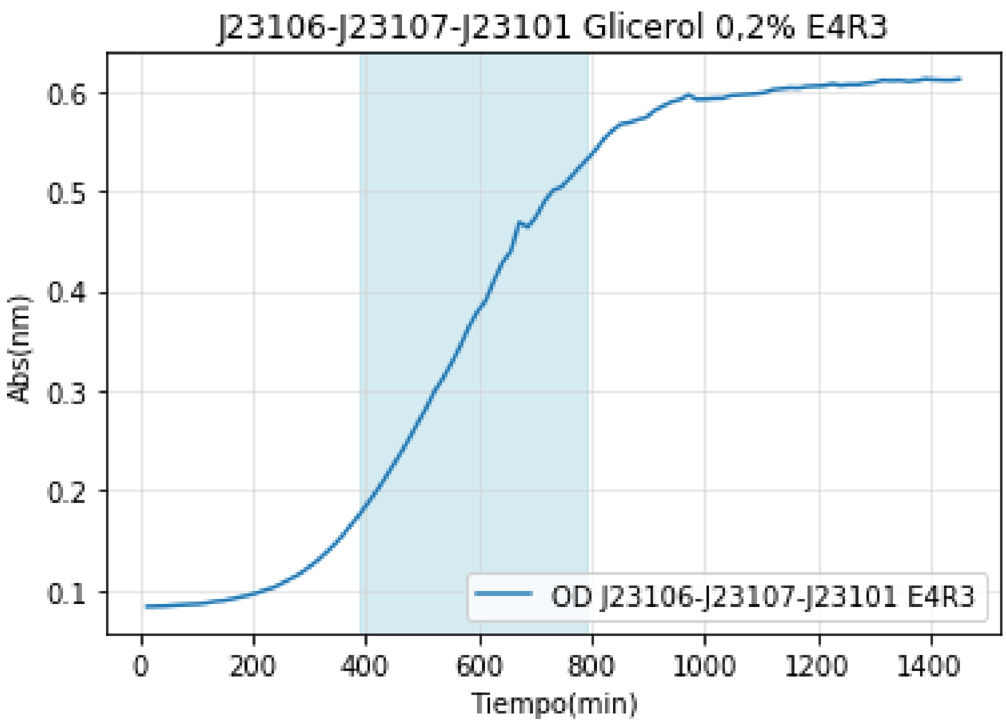
```
Min OD = 8.370000e-02
[  2.00705977e+00   4.34500193e-03   2.13821510e+02]
```

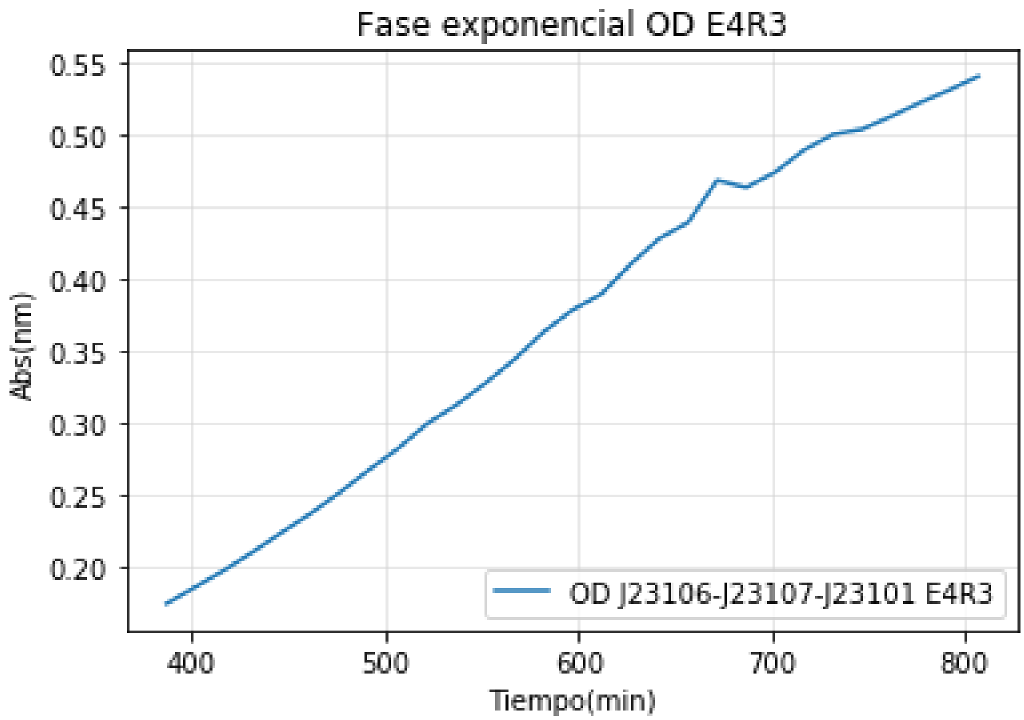



```
A=2.007060e+00
um=4.345002e-03
```

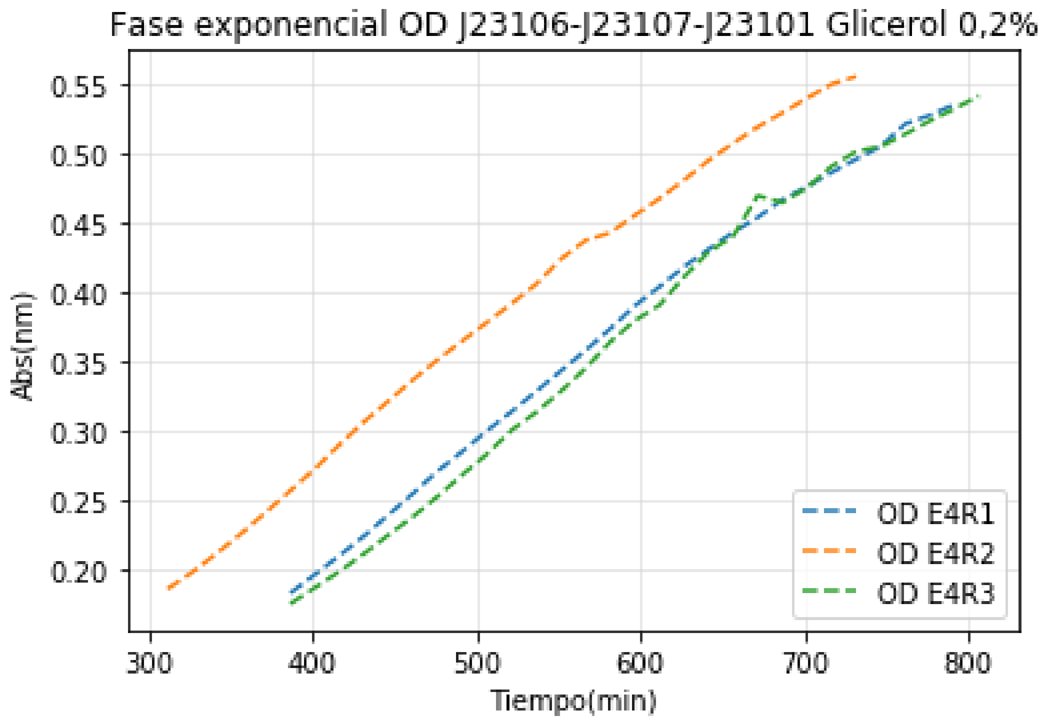
```
l=2.138215e+02
Tm=3.837538e+02
doubpe=1.595275e+02
ext=3.988187e+02
Tfinal=7.825725e+02
```

```
Out[29]: <matplotlib.legend.Legend at 0x1b2070f13c8>
```



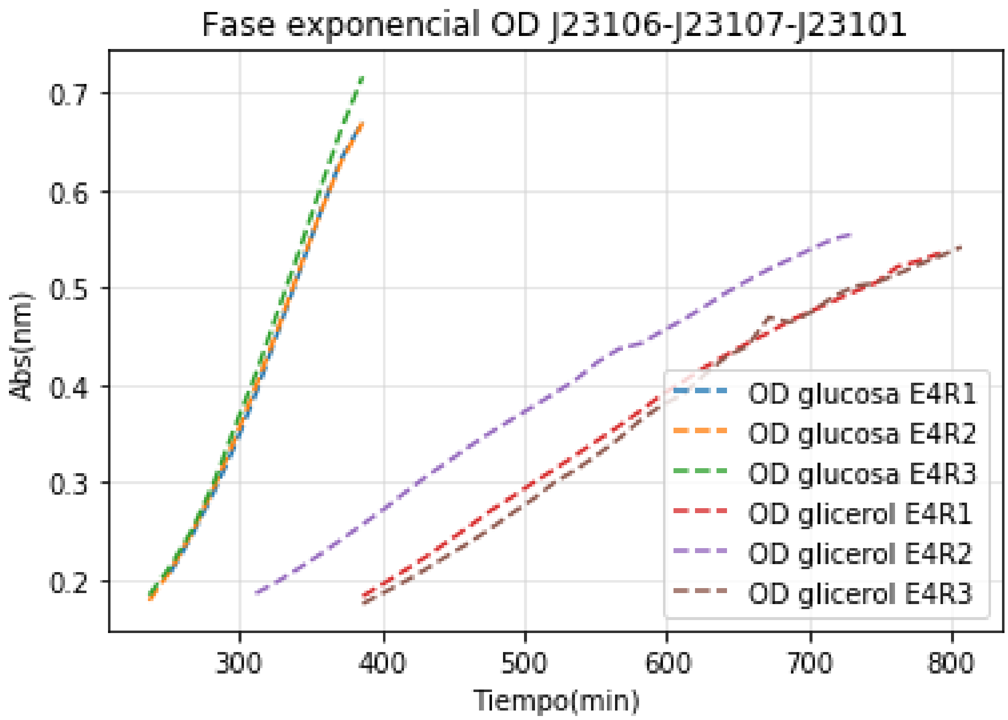



```
In [30]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23107-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[25:53],od1223g1[25:53],'--',label='OD E4R1')
         plt.plot(tt[20:49],od1223g2[20:49],'--',label='OD E4R2')
         plt.plot(tt[25:54],od1223g3[25:54],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[30]: <matplotlib.legend.Legend at 0x1b207215dd8>
```




```
In [31]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-J23107-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],od12231[16:26],'--',label='OD glucosa E4R1')
         plt.plot(tt[15:26],od12232[15:26],'--',label='OD glucosa E4R2')
         plt.plot(tt[15:26],od12233[15:26],'--',label='OD glucosa E4R3')
         plt.plot(tt[25:53],od1223g1[25:53],'--',label='OD glicerol E4R1')
         plt.plot(tt[20:49],od1223g2[20:49],'--',label='OD glicerol E4R2')
         plt.plot(tt[25:54],od1223g3[25:54],'--',label='OD glicerol E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')
```

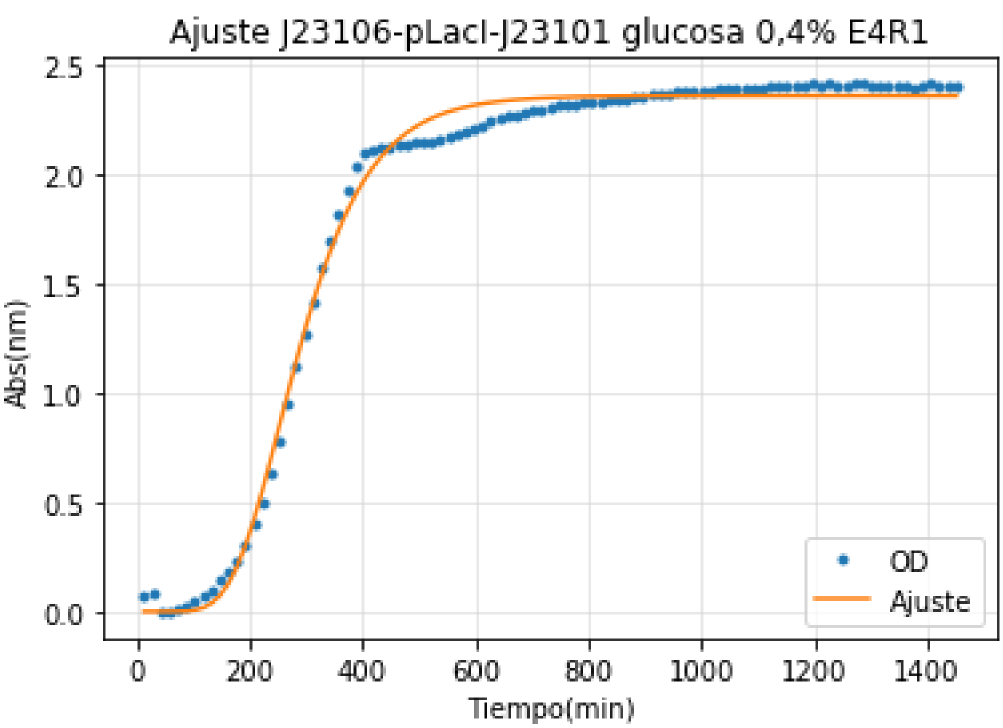
```
Out[31]: <matplotlib.legend.Legend at 0x1b2072ddcc0>
```



In [32]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-plac-std glucosa rep 1
y19= np.log(od12261)-np.log(np.min(od12261))
print('Min OD = %e'%((np.min(od12261))))
evaly, params=Function_fit(tt,y19,0,-1,title = 'Ajuste J23106-pLacI-J23101 glucosa 0,4%
A19= params[0]
um19=params[1]
l19=params[2]
print('A=%e'%(A19))
print('um=%e'%(um19))
print('l=%e'%(l19))

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

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

```
y2=tt[27]
plt.figure()
plt.title('J23106-pLacI-J23101 Glucosa 0,4% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12261,label='OD J23106-pLacI-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[17:28],od12261[17:28],label='OD J23106-pLacI-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

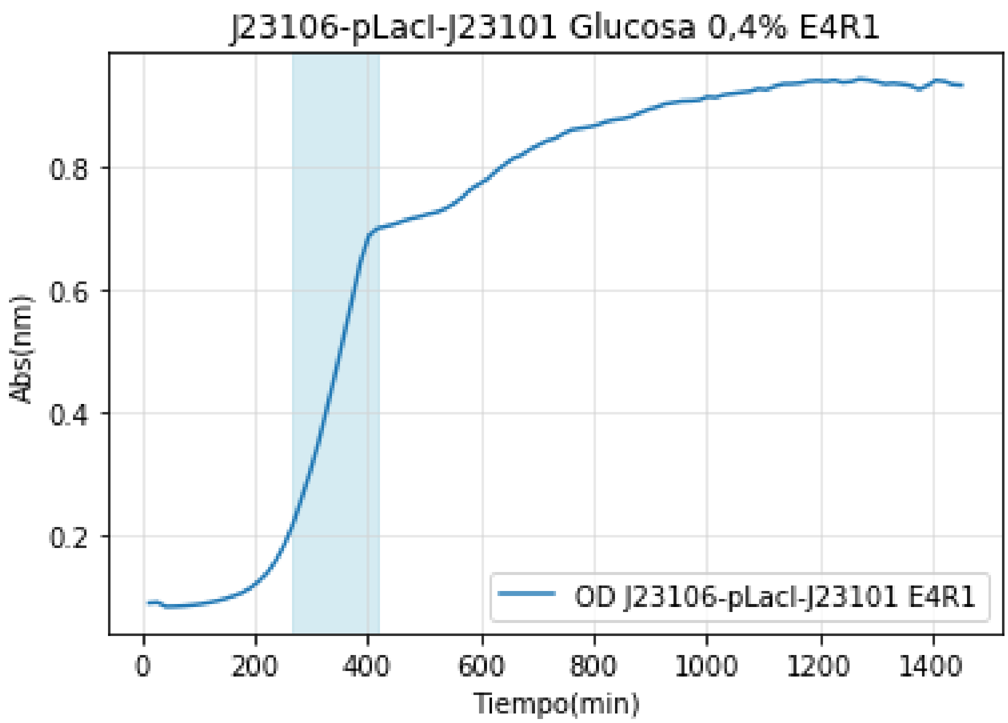
```
Min OD = 8.460000e-02
[  2.35733189e+00   1.00503143e-02   1.67202299e+02]
```

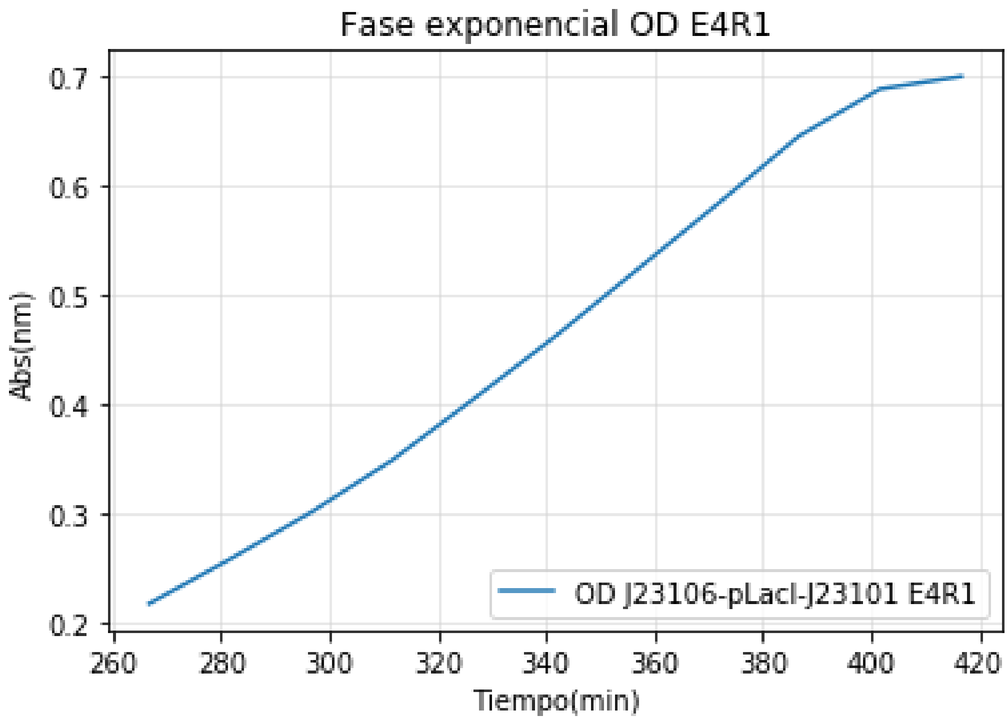



```
A=2.357332e+00
um=1.005031e-02
```

```
l=1.672023e+02
Tm=2.534895e+02
doubpe=6.896771e+01
ext=1.379354e+02
Tfinal=3.914250e+02
```

```
Out[32]: <matplotlib.legend.Legend at 0x1b2050eda20>
```

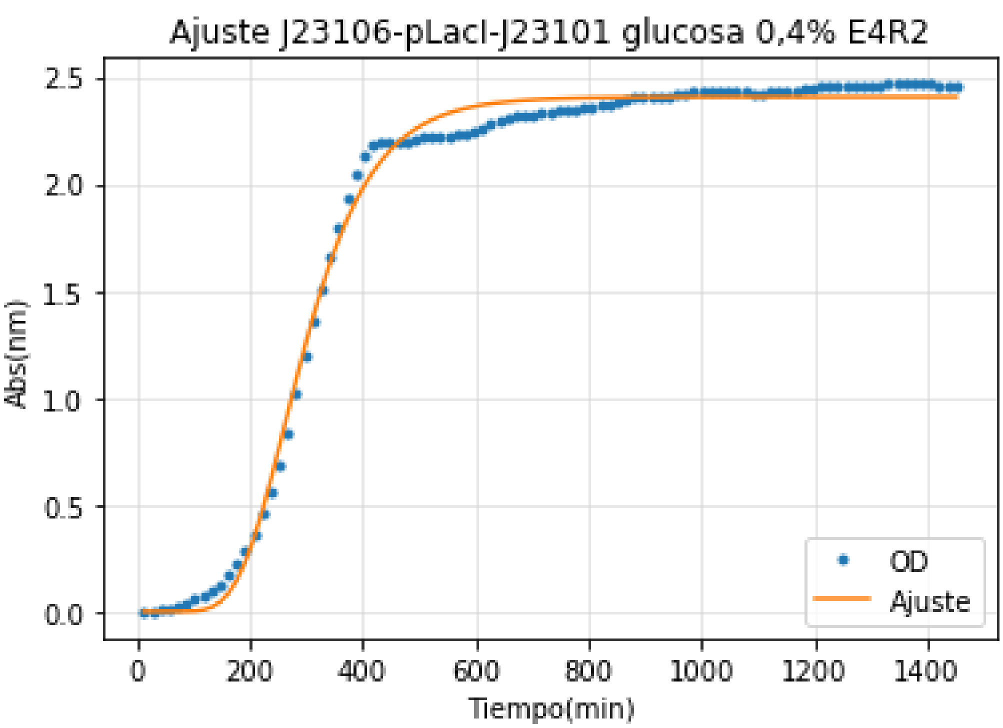




In [33]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-placI-std glucosa rep 2
y20= np.log(od12262)-np.log(np.min(od12262))
print('Min OD = %e'%((np.min(od12262))))
evaly, params=Function_fit(tt,y20,0,-1,title = 'Ajuste J23106-pLacI-J23101 glucosa 0,4%
A20= params[0]
um20=params[1]
l20=params[2]
print('A=%e'%(A20))
print('um=%e'%(um20))
print('l=%e'%(l20))

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

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

```
y2=tt[27]
plt.figure()
plt.title('J23106-pLacI-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12262,label='OD J23106-pLacI-J23101 E4R2 ')
plt.axvspan(y1,y2, color='lightblue', alpha=0.5)
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],od12262[17:28],label='OD J23106-pLacI-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

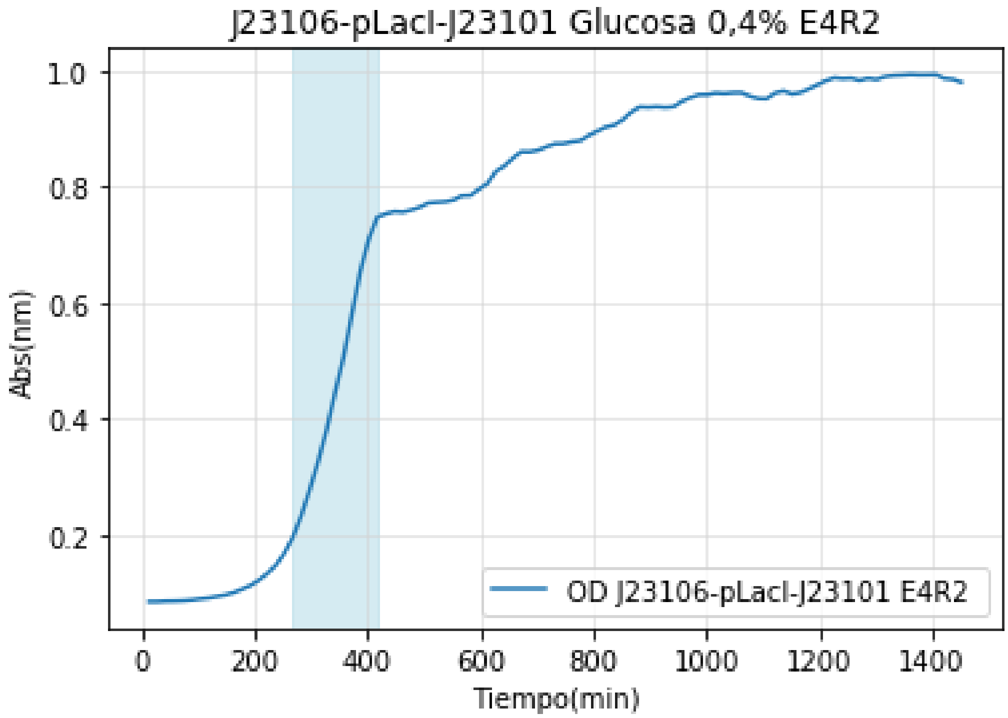
```
Min OD = 8.430000e-02
[  2.40664893e+00   1.05469249e-02   1.78463844e+02]
```

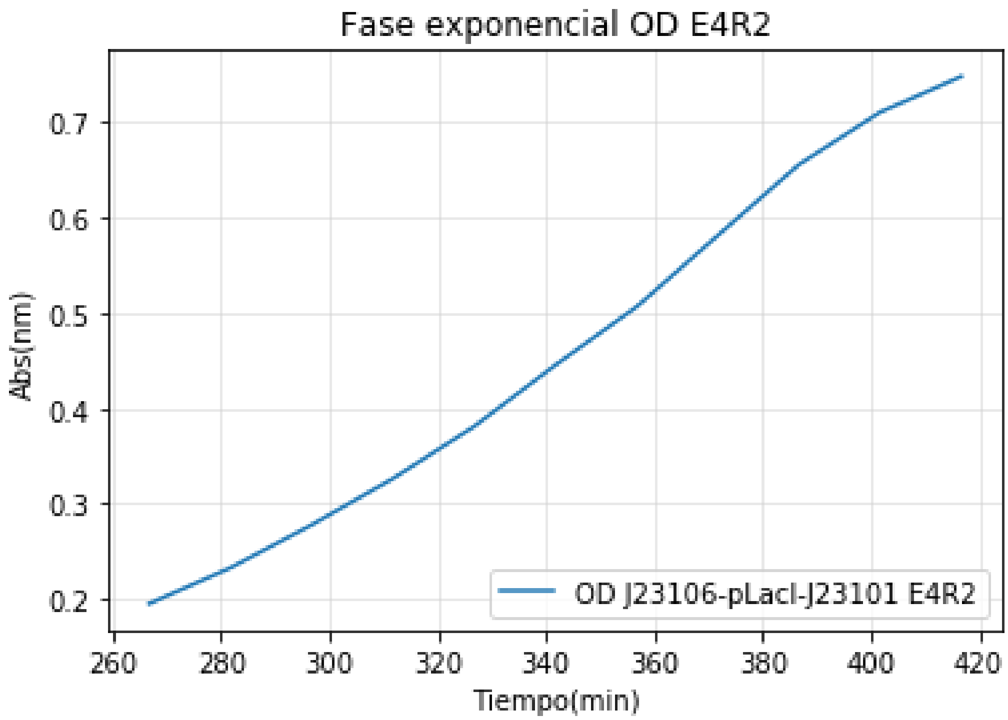



```
A=2.406649e+00
um=1.054692e-02
```

```
l=1.784638e+02
Tm=2.624084e+02
doubpe=6.572031e+01
ext=1.314406e+02
Tfinal=3.938490e+02
```

```
Out[33]: <matplotlib.legend.Legend at 0x1b2070094a8>
```

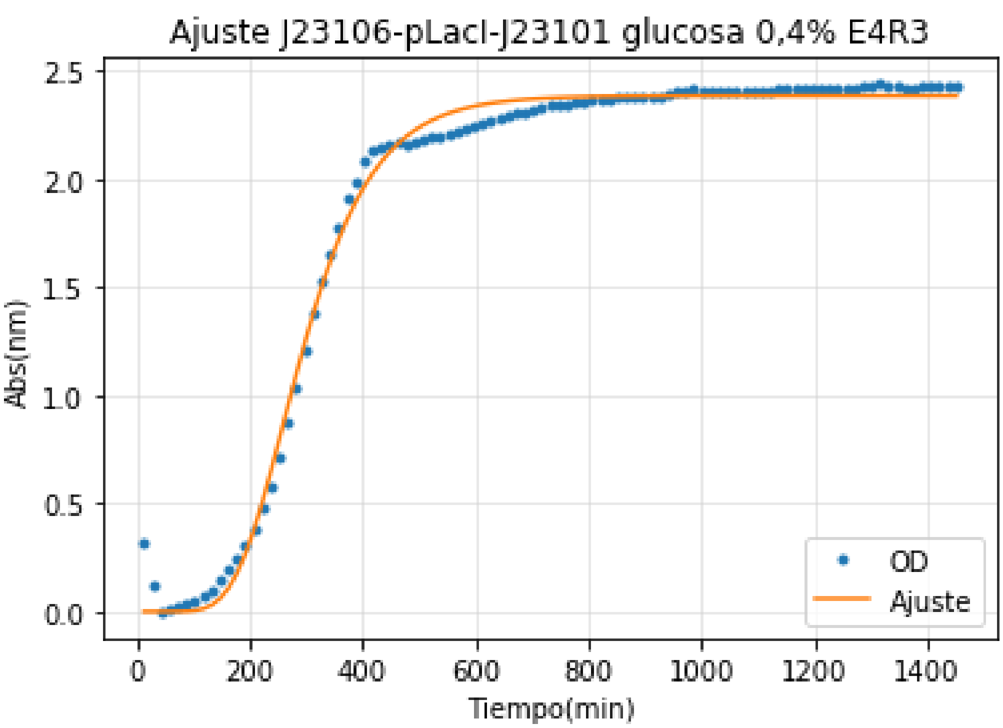





```
In [34]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-plac-std glucosa rep 3
         y21= np.log(od12263)-np.log(np.min(od12263))
         print('Min OD = %e'%((np.min(od12263))))
         evaly, params=Function_fit(tt,y21,0,-1,title = 'Ajuste J23106-pLacI-J23101 glucosa 0,4%
         A21= params[0]
         um21=params[1]
         l21=params[2]
         print('A=%e'%(A21))
         print('um=%e'%(um21))
         print('l=%e'%(l21))

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

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

```
y2=tt[27]
plt.figure()
plt.title('J23106-pLacI-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od12263,label='OD J23106-pLacI-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[17:28],od12263[17:28],label='OD J23106-pLacI-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

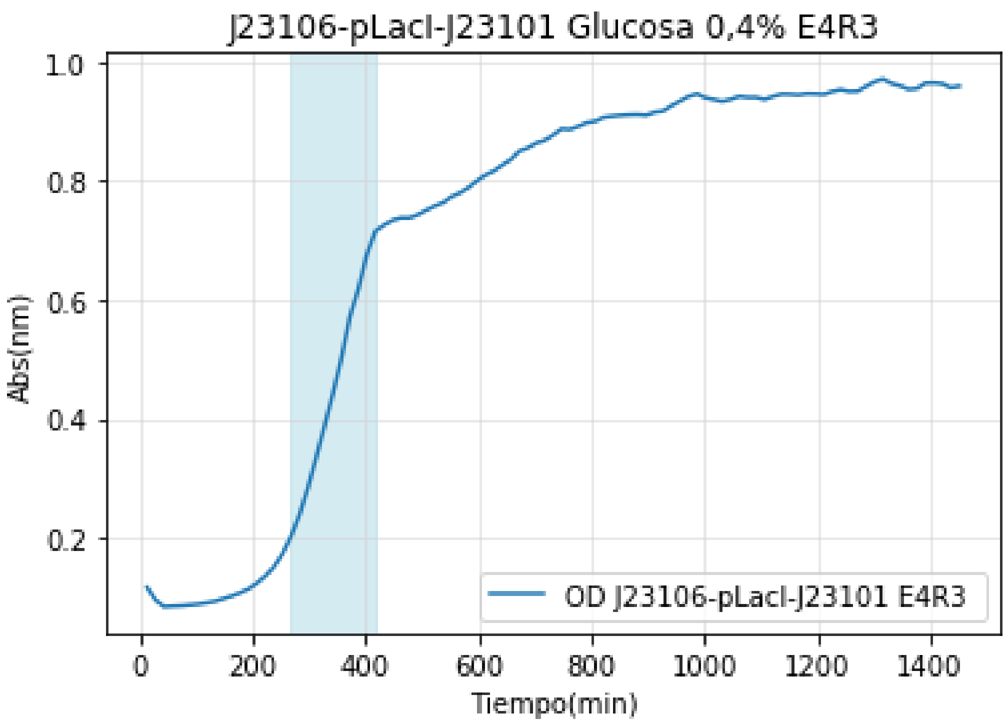
```
Min OD = 8.460000e-02
[  2.38737334e+00   1.01026915e-02   1.72612085e+02]
```

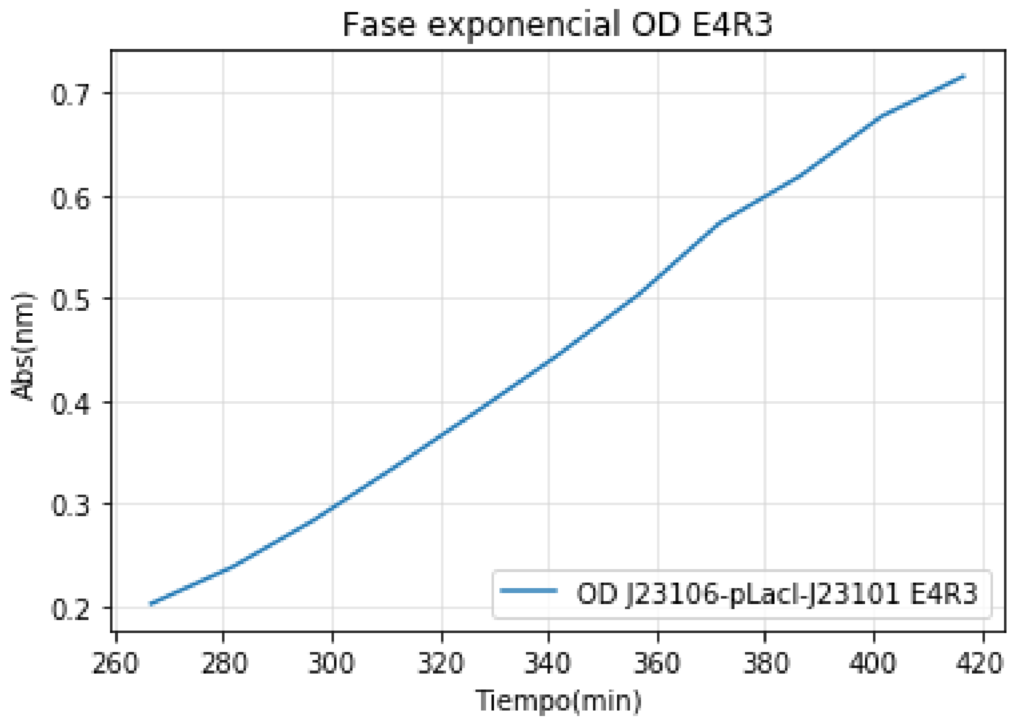



```
A=2.387373e+00
um=1.010269e-02
```

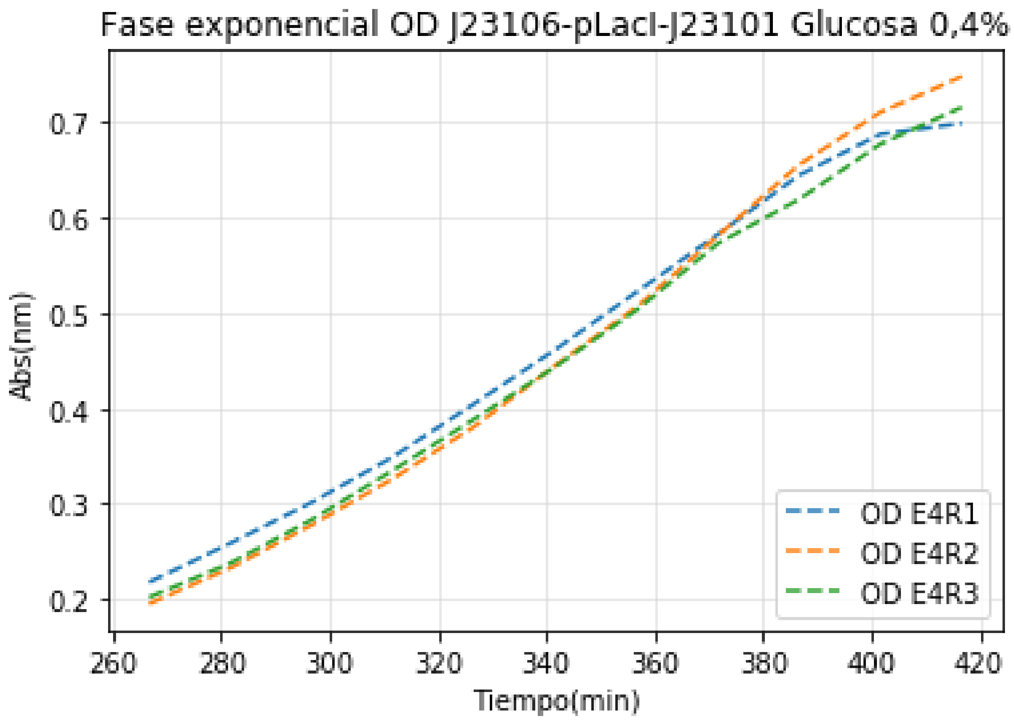
```
l=1.726121e+02
Tm=2.595459e+02
doubpe=6.861015e+01
ext=1.372203e+02
Tfinal=3.967662e+02
```

```
Out[34]: <matplotlib.legend.Legend at 0x1b2074ad080>
```





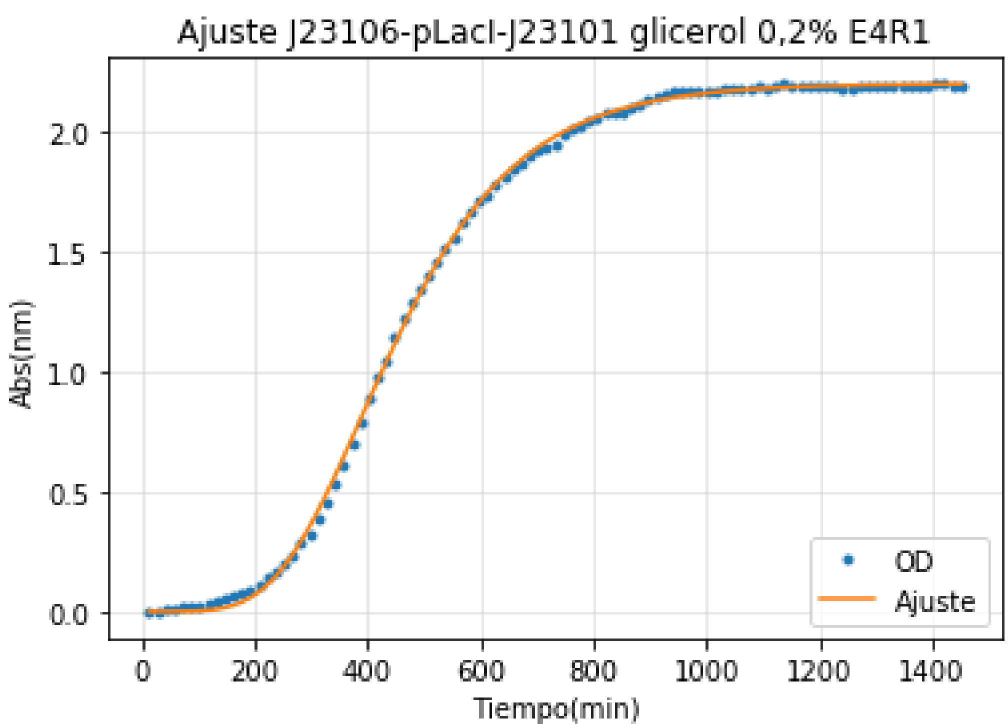

```
In [35]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-pLacI-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:28],od12261[17:28],'--',label='OD E4R1')
         plt.plot(tt[17:28],od12262[17:28],'--',label='OD E4R2')
         plt.plot(tt[17:28],od12263[17:28],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[35]: <matplotlib.legend.Legend at 0x1b20762afd0>
```




```
In [36]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-plac-std glicerol rep 1
         y22= np.log(od1226g1)-np.log(np.min(od1226g1))
         print('Min OD = %e'%((np.min(od1226g1))))
         evaly, params=Function_fit(tt,y22,0,-1,title = 'Ajuste J23106-pLacI-J23101 glicerol 0,2
         A22= params[0]
         um22=params[1]
         l22=params[2]
         print('A=%e'%(A22))
         print('um=%e'%(um22))
         print('l=%e'%(l22))

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

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

```
y2=tt[48]
plt.figure()
plt.title('J23106-pLacI-J23101 Glicerol 0,2% E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1226g1,label='OD J23106-pLacI-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[26:49],od1226g1[26:49],label='OD J23106-pLacI-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

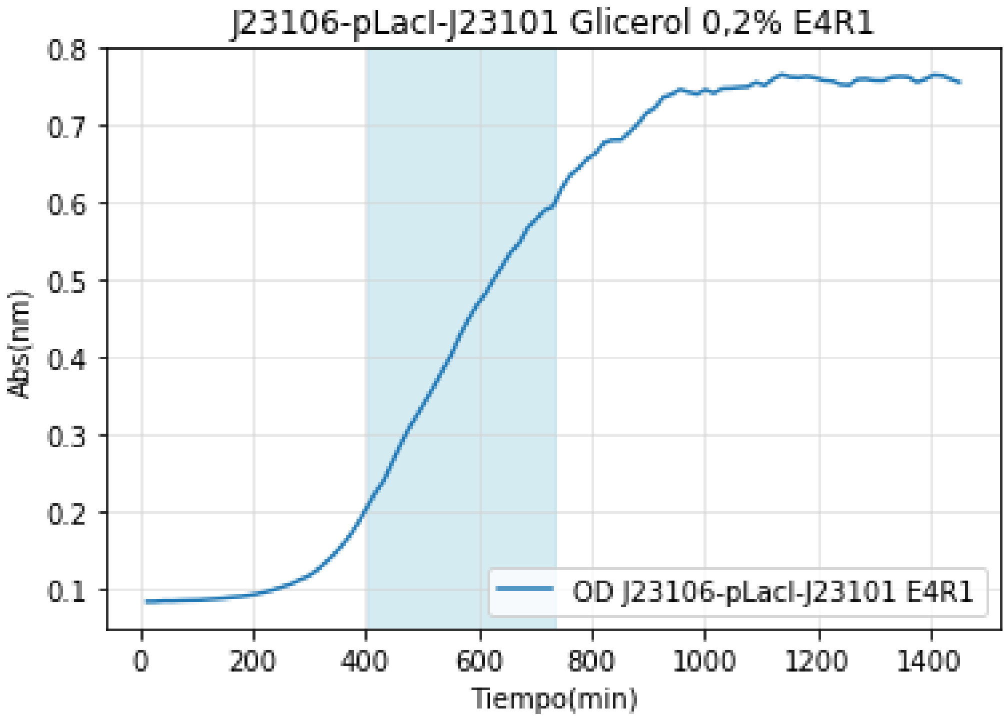
```
Min OD = 8.470000e-02
[  2.20277900e+00   5.30923238e-03   2.35467207e+02]
```

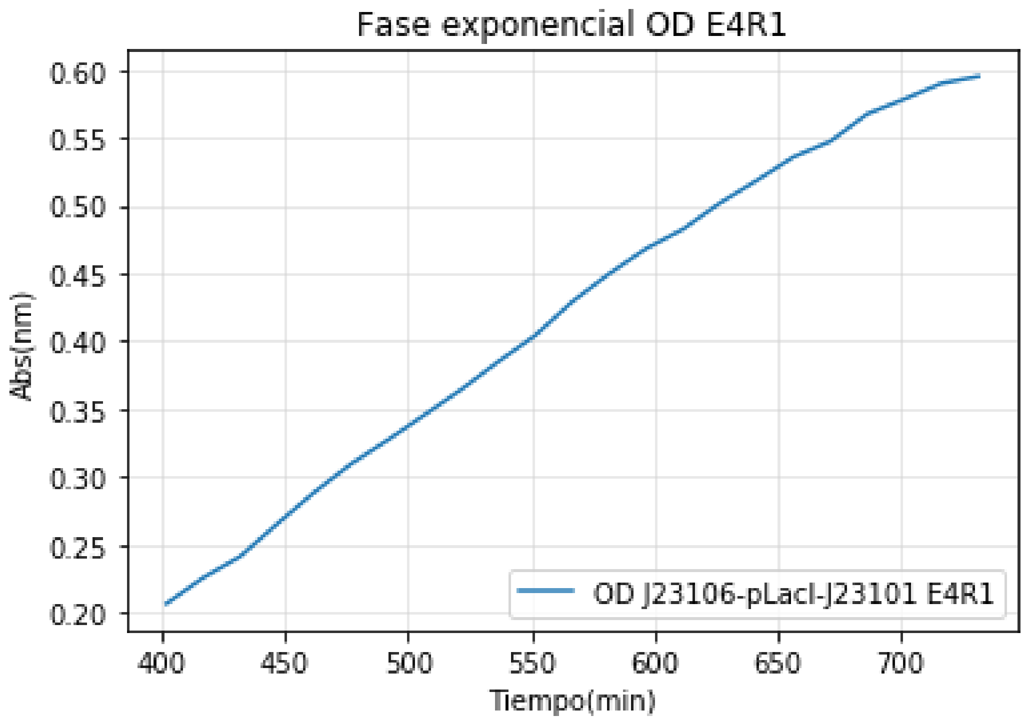



```
A=2.202779e+00
um=5.309232e-03
```

```
l=2.354672e+02
Tm=3.880989e+02
doubpe=1.305551e+02
ext=3.263877e+02
Tfinal=7.144866e+02
```

Out[36]: <matplotlib.legend.Legend at 0x1b2076fb8d0>

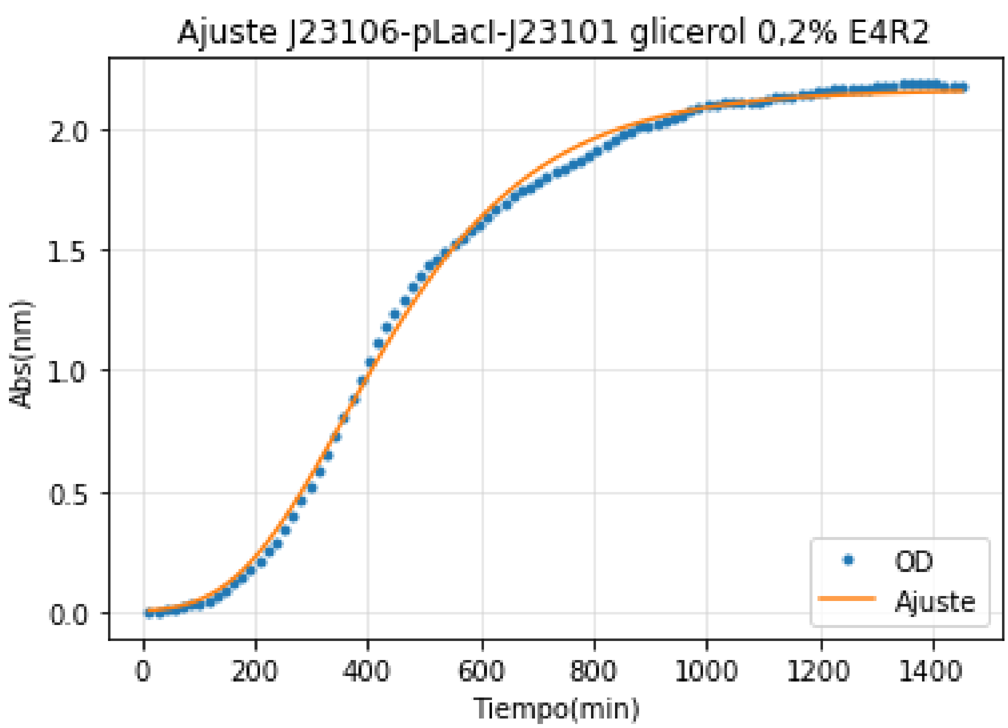





```
In [37]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-plac-std glicerol rep 2
         y23=np.log(od1226g2)-np.log(np.min(od1226g2))
         print('Min OD = %e'%((np.min(od1226g2))))
         evaly, params=Function_fit(tt,y23,0,-1,title = 'Ajuste J23106-pLacI-J23101 glicerol 0,2
         A23= params[0]
         um23=params[1]
         l23=params[2]
         print('A=%e'%(A23))
         print('um=%e'%(um23))
         print('l=%e'%(l23))

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

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

```
y2=tt[52]
plt.figure()
plt.title('J23106-pLacI-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1226g2,label='OD J23106-pLacI-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[23:53],od1226g2[23:53],label='OD J23106-pLacI-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

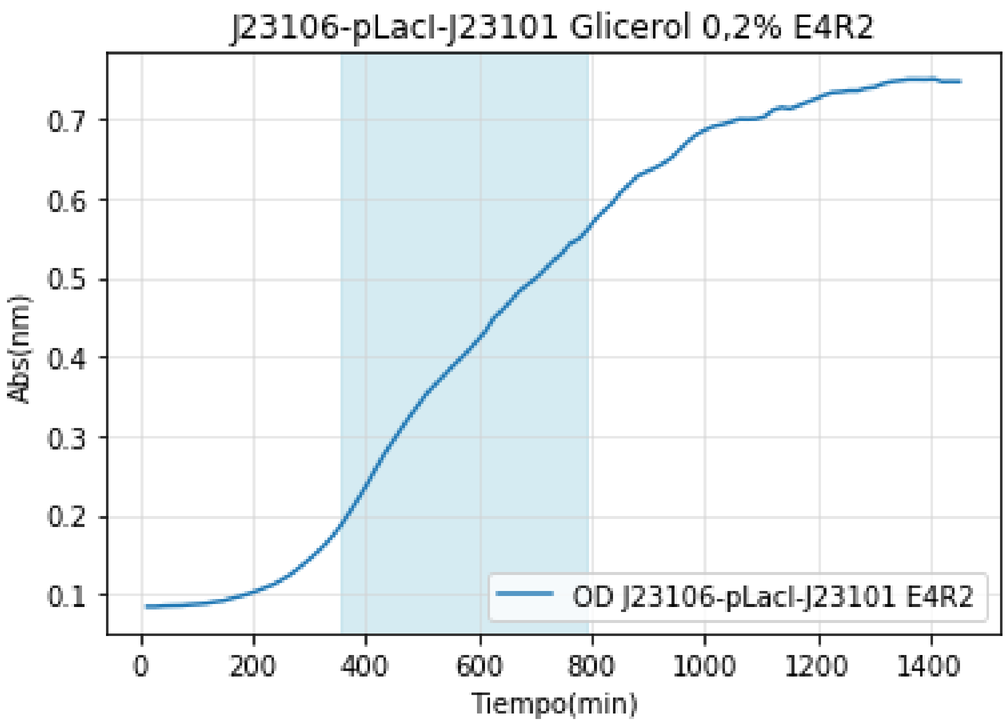
```
Min OD = 8.440000e-02
[  2.16301440e+00   4.14664738e-03   1.64063323e+02]
```

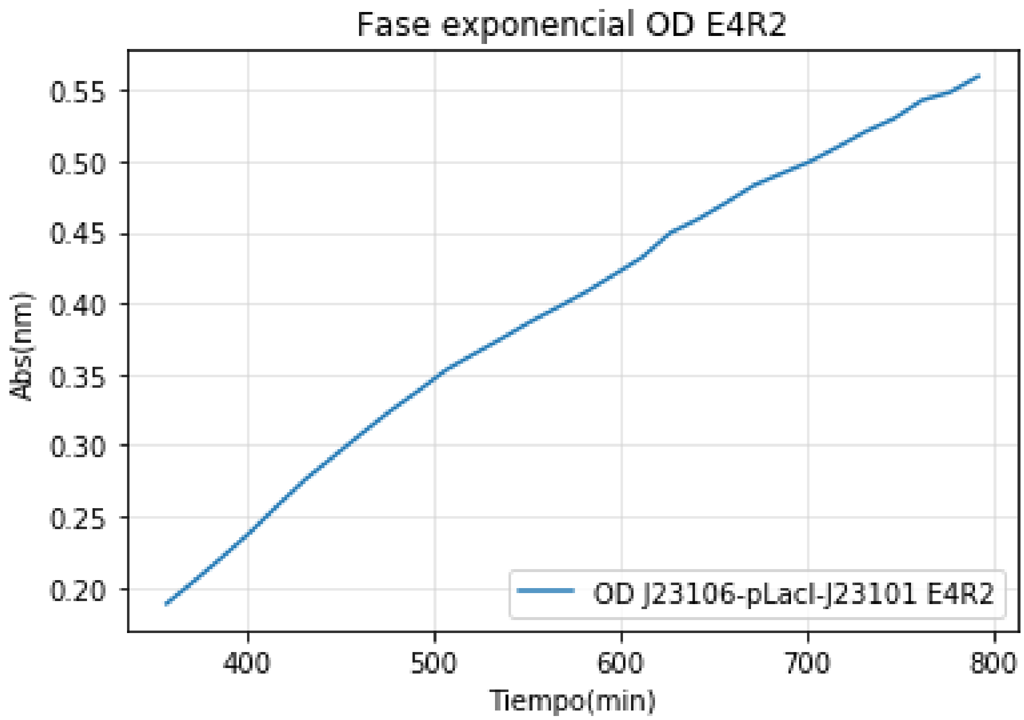



```
A=2.163014e+00
um=4.146647e-03
```

```
l=1.640633e+02
Tm=3.559602e+02
doubpe=1.671585e+02
ext=4.178961e+02
Tfinal=7.738563e+02
```

Out[37]: <matplotlib.legend.Legend at 0x1b205a7c518>

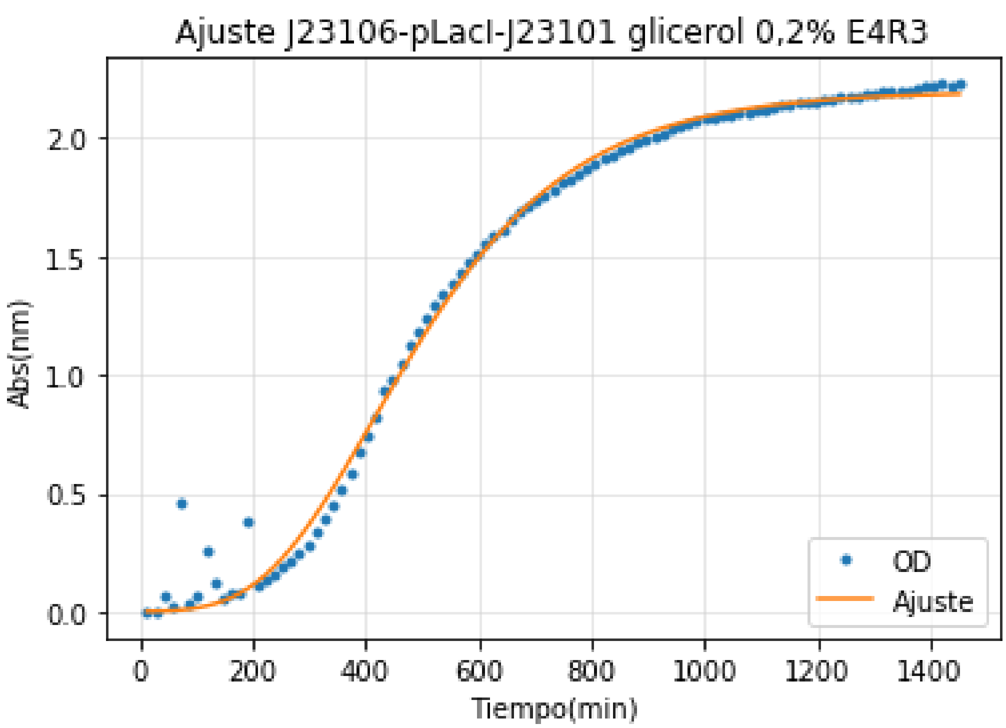





```
In [38]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-plac-std glicerol rep 3
         y24=np.log(od1226g3)-np.log(np.min(od1226g3))
         print('Min OD = %e'%((np.min(od1226g3))))
         evaly, params=Function_fit(tt,y24,0,-1,title = 'Ajuste J23106-pLacI-J23101 glicerol 0,2
         A24= params[0]
         um24=params[1]
         l24=params[2]
         print('A=%e'%(A24))
         print('um=%e'%(um24))
         print('l=%e'%(l24))

         #Cálculo datos para determinar extensión de la fase exponencial
         tm24=((A24/(np.exp(1)*um24))+l24)
         print('Tm=%e'%(tm24))
         t224=((np.log(2))/um24)
         print('doubpe=%e'%(t224))
         extdp24=2.5*t224
         print('ext=%e'%extdp24)
         ttot24=tm24+extdp24
         print('Tfinal=%e'%ttot24)

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

```
y2=tt[56]
plt.figure()
plt.title('J23106-pLacI-J23101 Glicerol 0,2% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt,od1226g3,label='OD J23106-pLacI-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[27:57],od1226g3[27:57],label='OD J23106-pLacI-J23101 E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

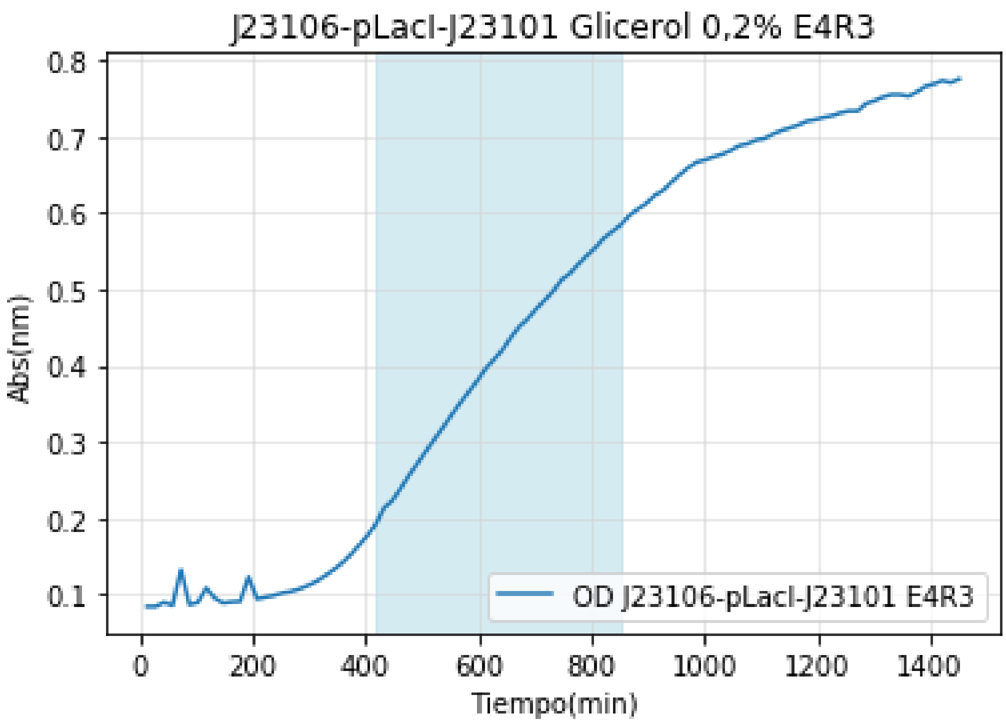
```
Min OD = 8.360000e-02
[  2.19762456e+00   4.12139949e-03   2.16414536e+02]
```

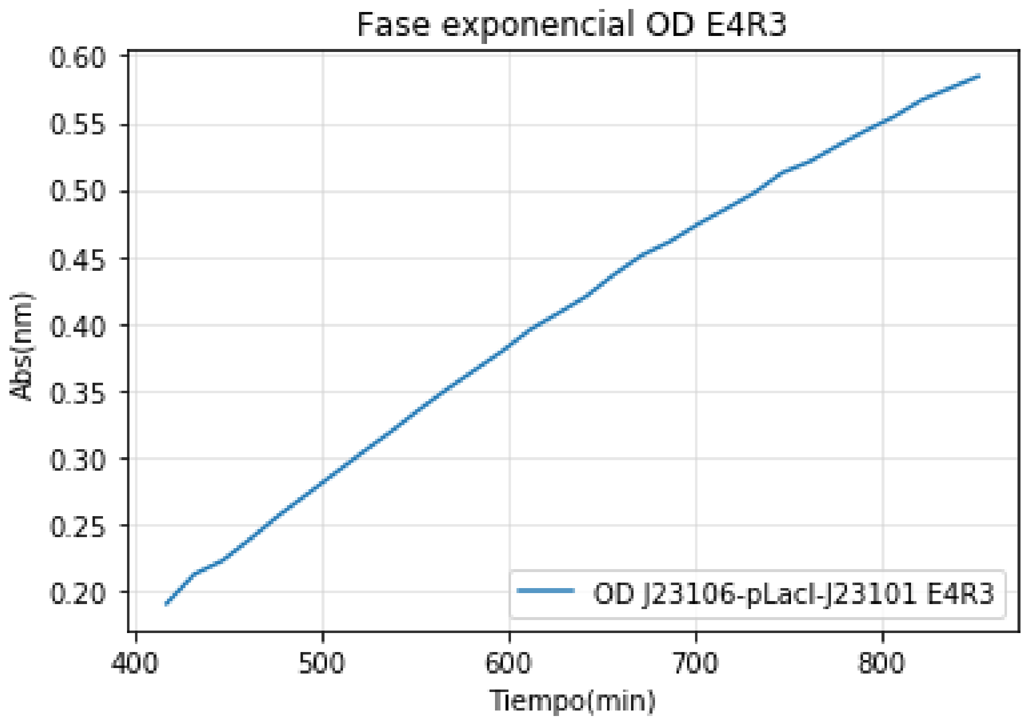



```
A=2.197625e+00
um=4.121399e-03
```

```
l=2.164145e+02
Tm=4.125763e+02
doubpe=1.681825e+02
ext=4.204562e+02
Tfinal=8.330325e+02
```

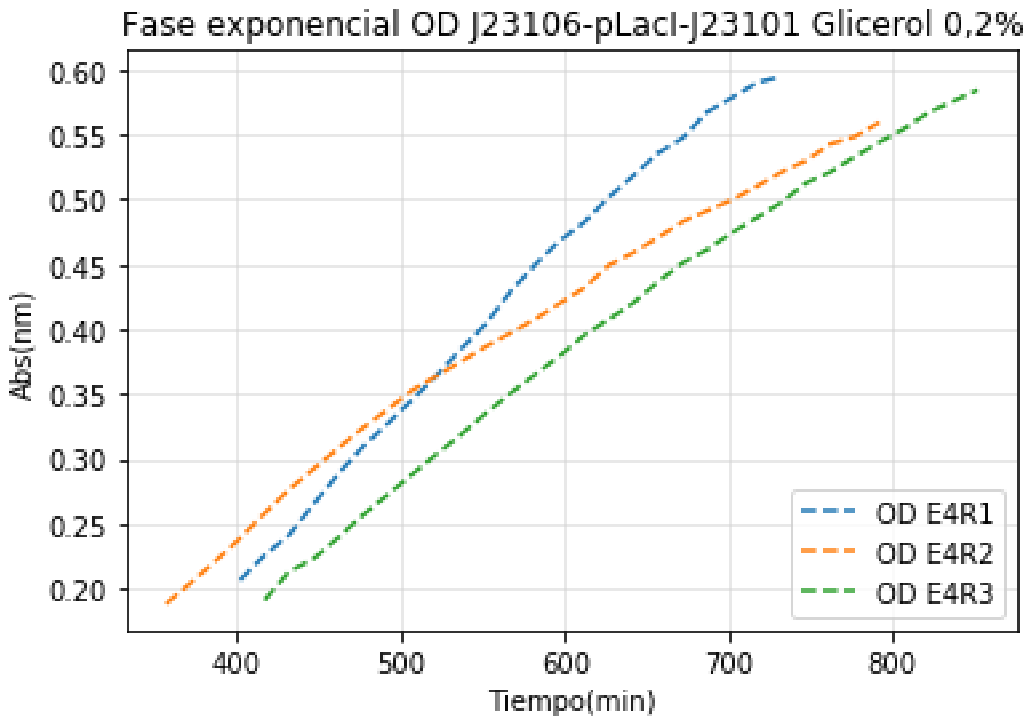
Out[38]: <matplotlib.legend.Legend at 0x1b2076dc9e8>



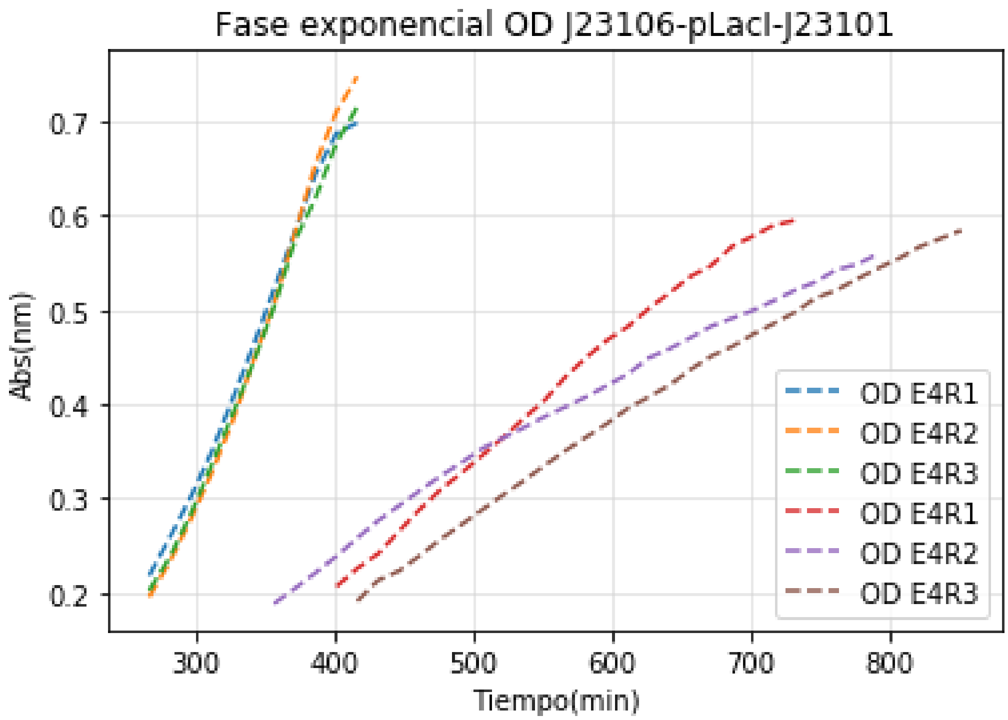


In [39]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23106-pLacI-J23101 Glicerol 0,2%')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[26:49],od1226g1[26:49],'--',label='OD E4R1')
plt.plot(tt[23:53],od1226g2[23:53],'--',label='OD E4R2')
plt.plot(tt[27:57],od1226g3[27:57],'--',label='OD E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[39]:
```
<matplotlib.legend.Legend at 0x1b20784fbe0>
```



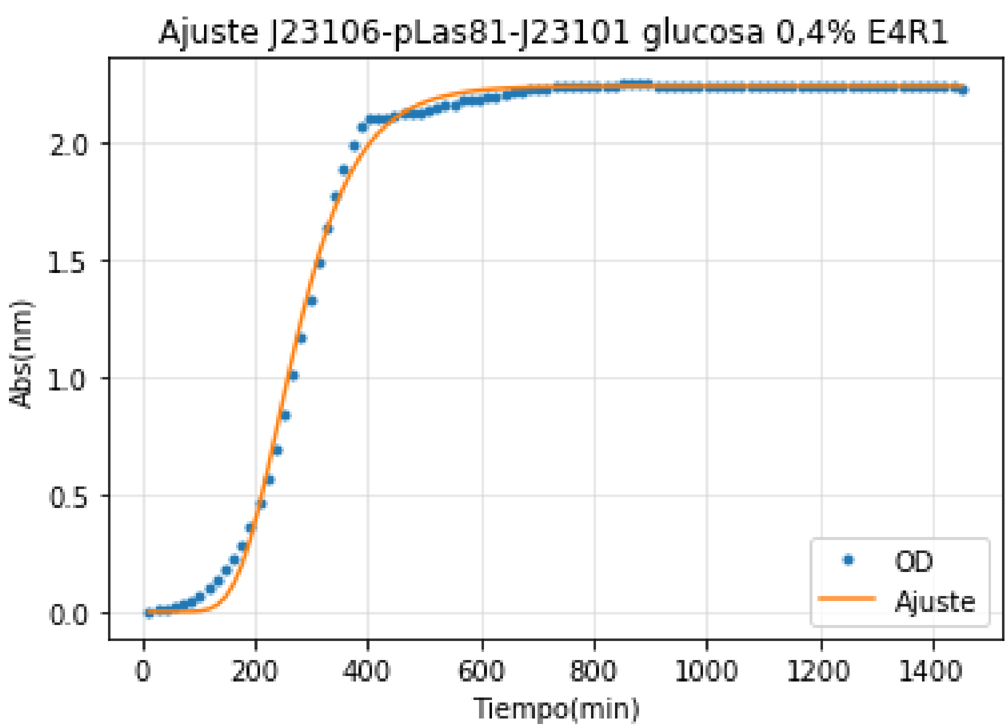

```
In [40]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-pLacI-J23101')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[17:28],od12261[17:28],'--',label='OD E4R1')
         plt.plot(tt[17:28],od12262[17:28],'--',label='OD E4R2')
         plt.plot(tt[17:28],od12263[17:28],'--',label='OD E4R3')
         plt.plot(tt[26:49],od1226g1[26:49],'--',label='OD E4R1')
         plt.plot(tt[23:53],od1226g2[23:53],'--',label='OD E4R2')
         plt.plot(tt[27:57],od1226g3[27:57],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[40]: <matplotlib.legend.Legend at 0x1b207919d30>
```



In [41]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-plas-std glucosa rep 1
y25= np.log(od12271)-np.log(np.min(od12271))
print('Min OD = %e'%((np.min(od12271))))
evaly, params=Function_fit(tt,y25,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,4
A25= params[0]
um25=params[1]
l25=params[2]
print('A=%e'%(A25))
print('um=%e'%(um25))
print('l=%e'%(l25))

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

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

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

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R1')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:26],od12271[17:26],label='OD J23106-pLas81-J23101 E4R1')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

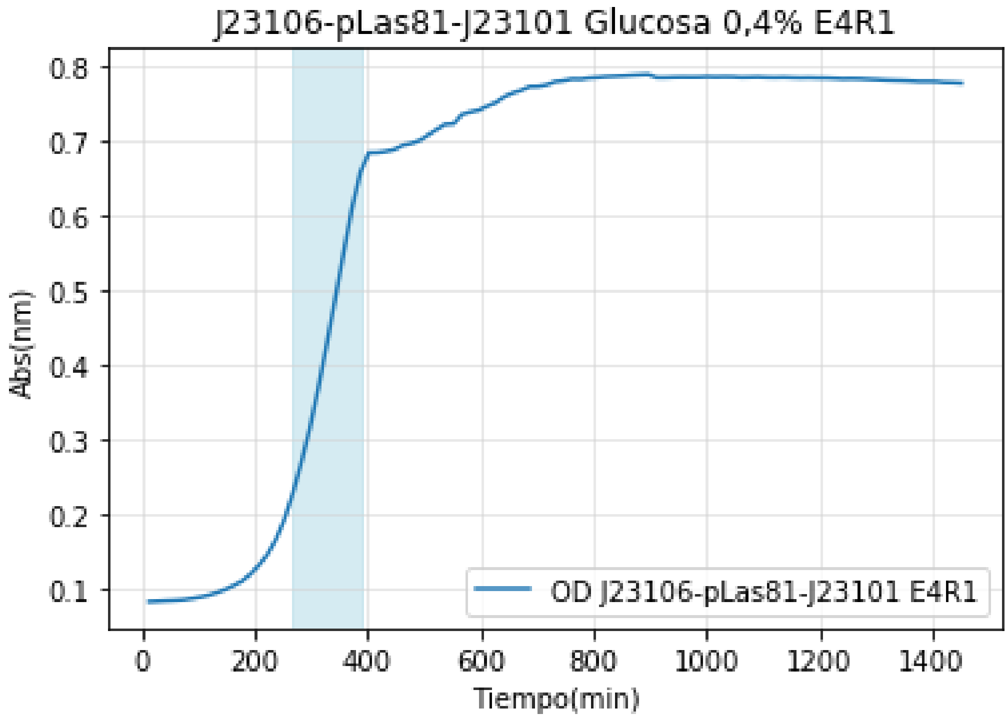
```
Min OD = 8.350000e-02
[  2.23907480e+00   1.10691843e-02   1.67672817e+02]
```

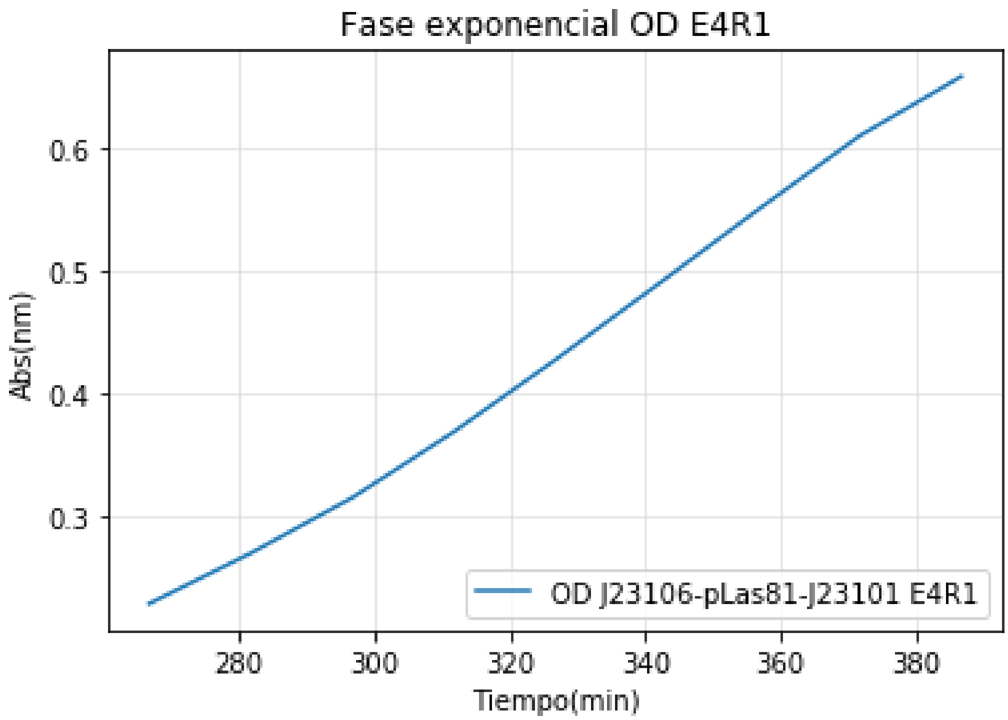



```
A=2.239075e+00
um=1.106918e-02
```

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

```
Out[41]: <matplotlib.legend.Legend at 0x1b207abb4a8>
```

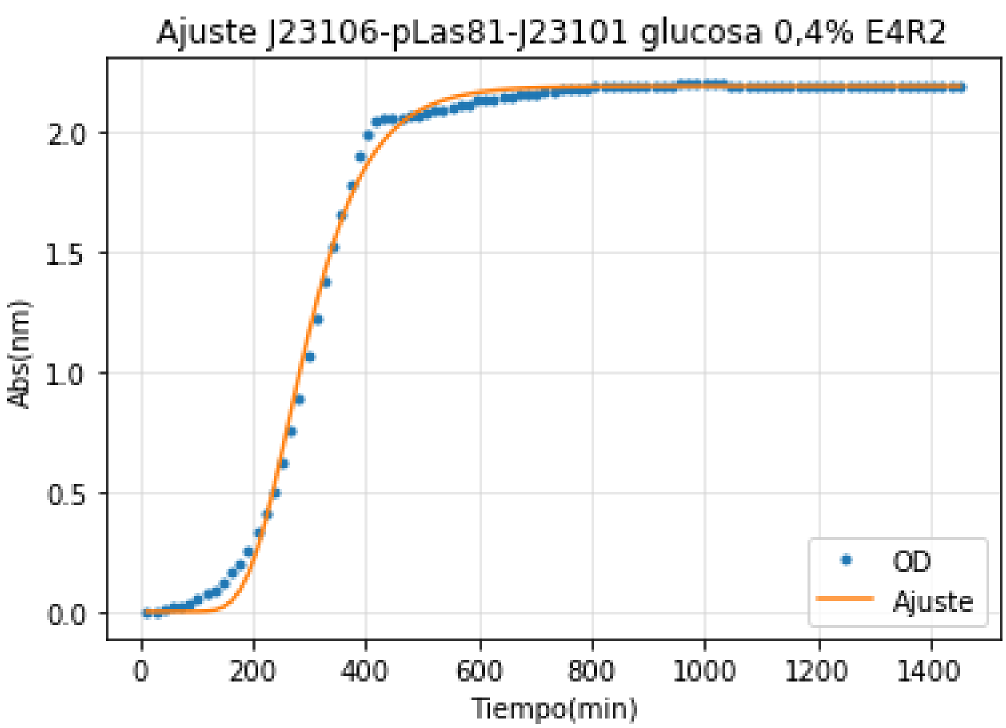




In [42]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-pLas-std glucosa rep 2
y26= np.log(od12272)-np.log(np.min(od12272))
print('Min OD = %e'%((np.min(od12272))))
evaly, params=Function_fit(tt,y26,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,4
A26= params[0]
um26=params[1]
l26=params[2]
print('A=%e'%(A26))
print('um=%e'%(um26))
print('l=%e'%(l26))

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

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

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

#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[17:28],od12272[17:28],label='OD J23106-pLas81-J23101 E4R2')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

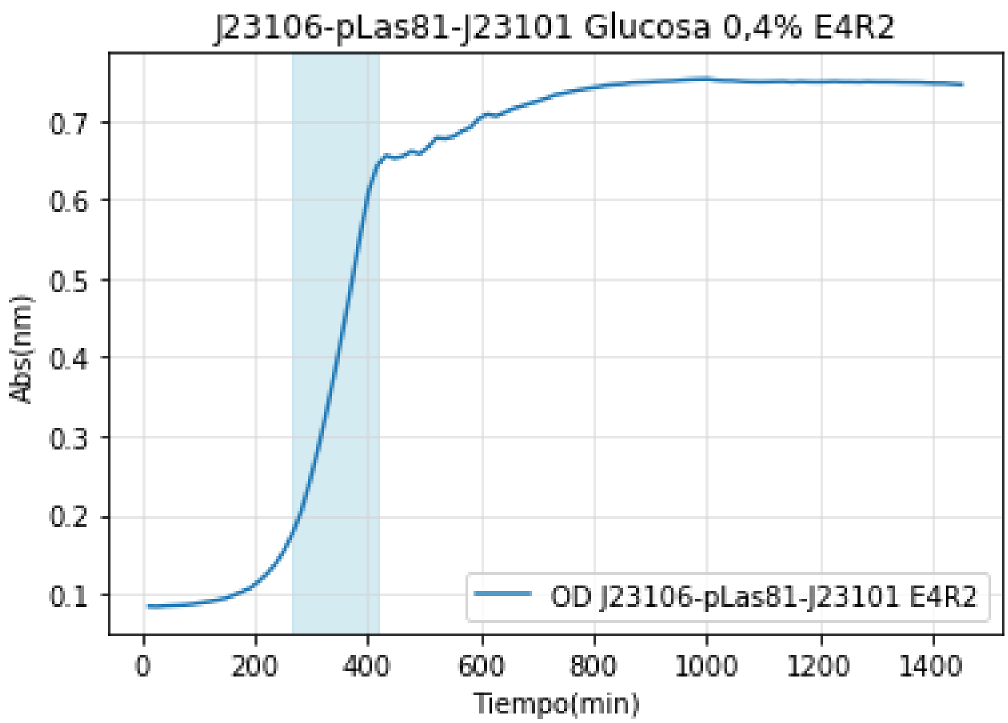
```
Min OD = 8.360000e-02
[  2.18901562e+00   1.06857798e-02   1.88770693e+02]
```

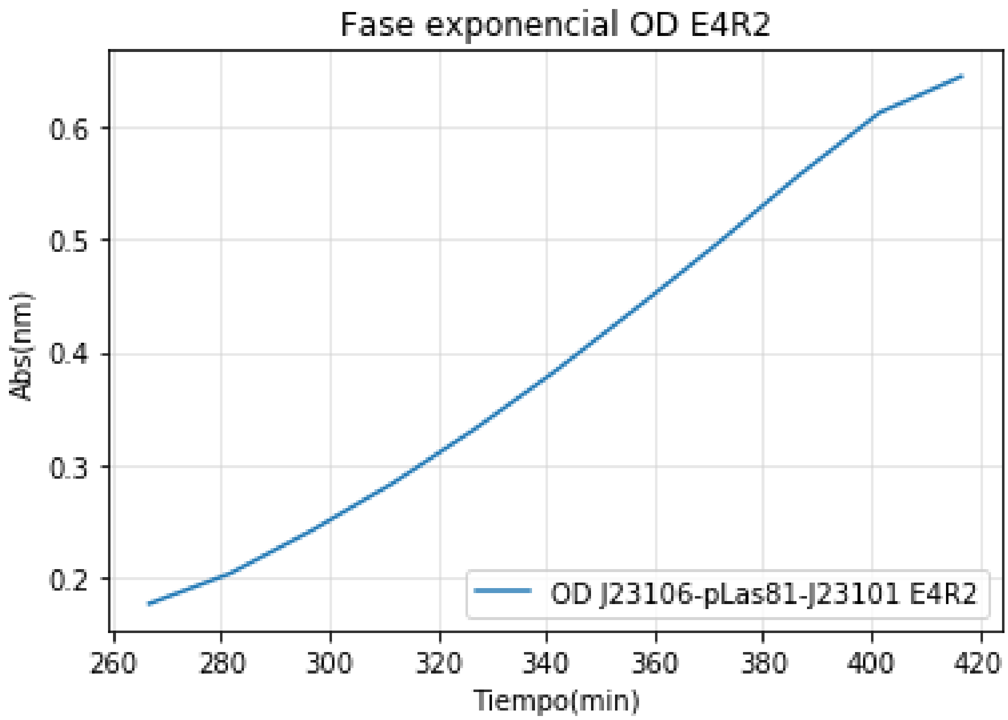



```
A=2.189016e+00
um=1.068578e-02
```

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

```
Out[42]: <matplotlib.legend.Legend at 0x1b207a08d30>
```

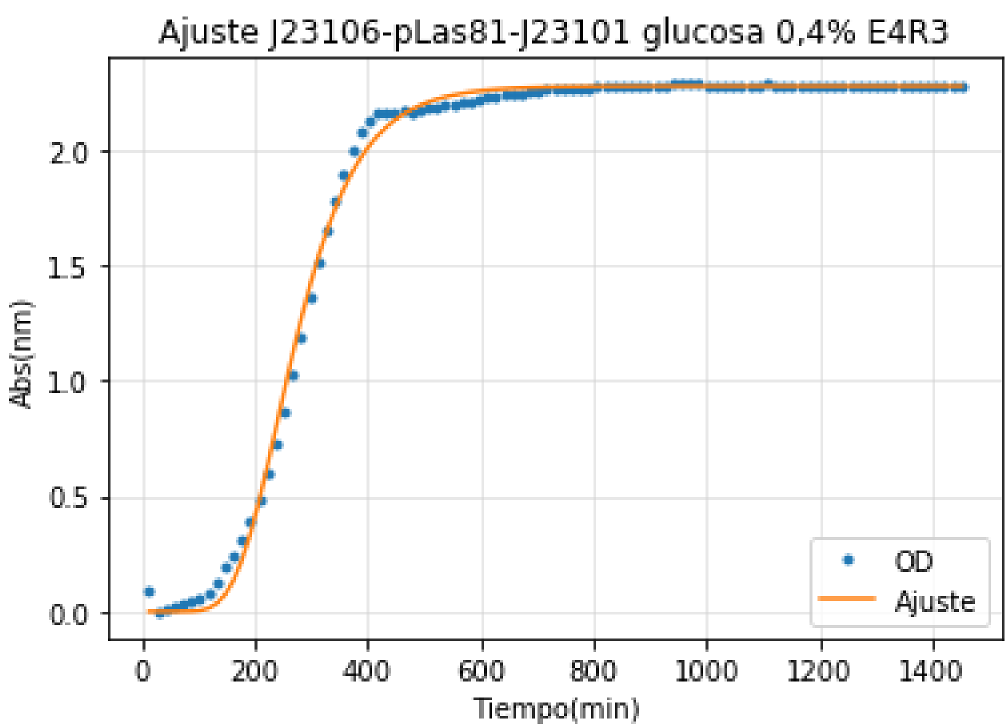





```
In [43]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-pLas-std glucosa rep 3
         y27= np.log(od12273)-np.log(np.min(od12273))
         print('Min OD = %e'%((np.min(od12273))))
         evaly, params=Function_fit(tt,y27,0,-1,title = 'Ajuste J23106-pLas81-J23101 glucosa 0,4
         A27= params[0]
         um27=params[1]
         l27=params[2]
         print('A=%e'%(A27))
         print('um=%e'%(um27))
         print('l=%e'%(l27))

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

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

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

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

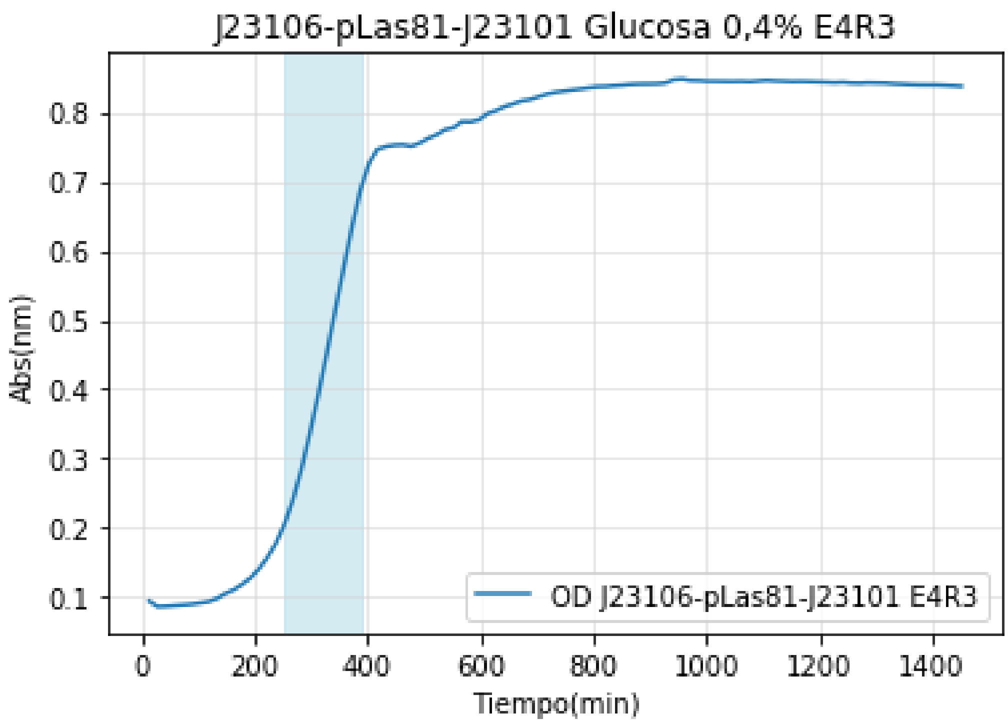
```
Min OD = 8.640000e-02
[  2.27424241e+00   1.09679411e-02   1.64283900e+02]
```

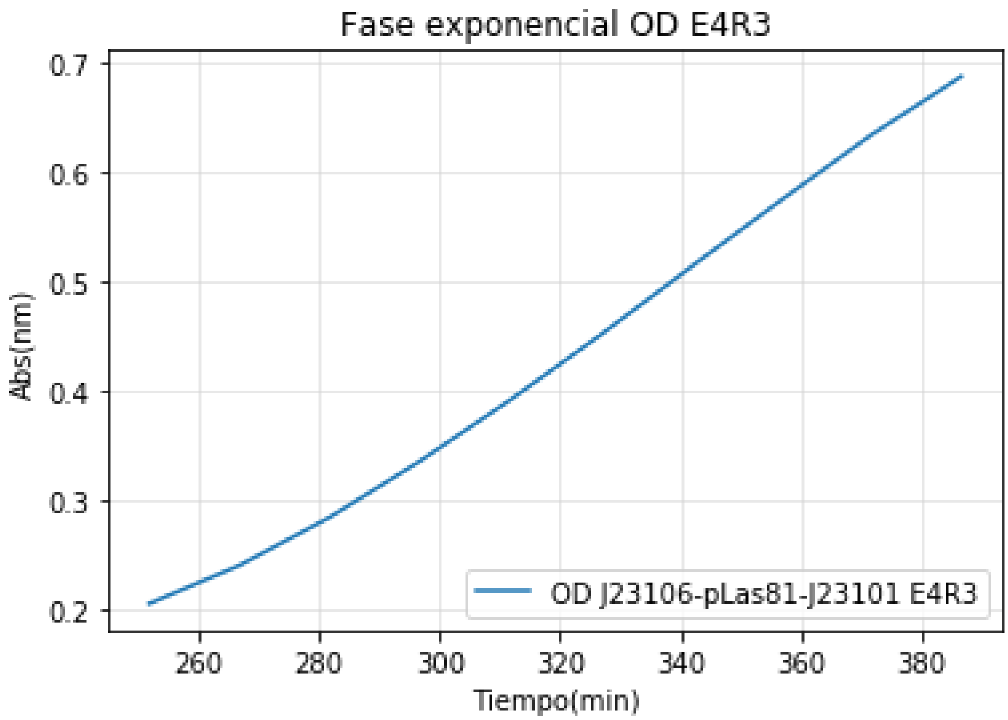



```
A=2.274242e+00
um=1.096794e-02
```

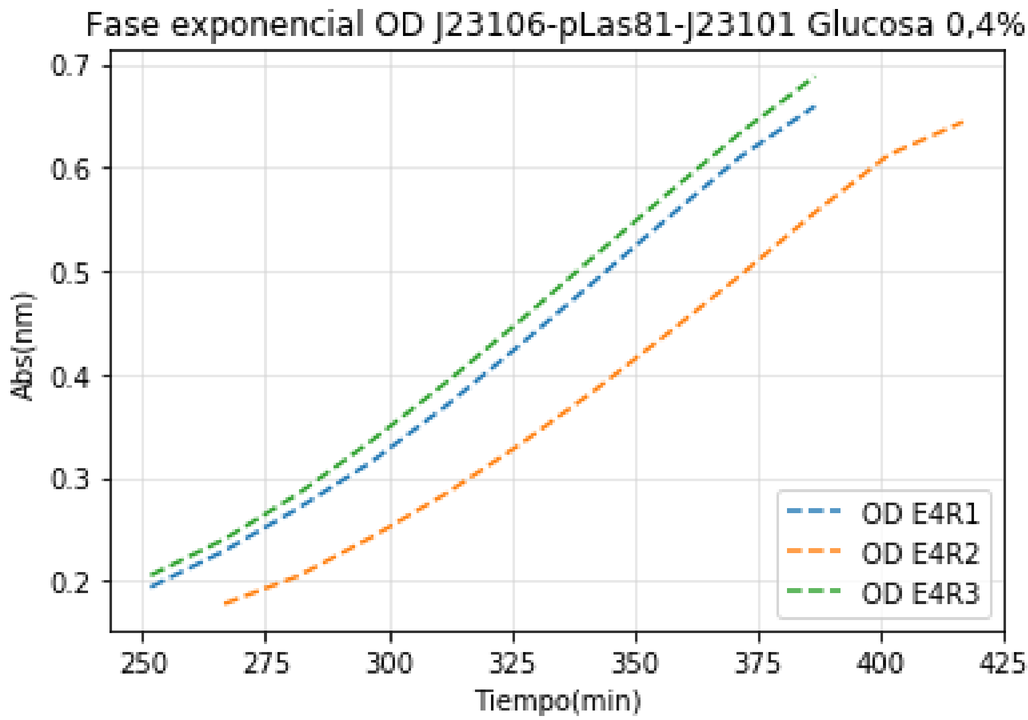
```
l=1.642839e+02
Tm=2.405650e+02
doubpe=6.319757e+01
ext=1.263951e+02
Tfinal=3.669602e+02
```

```
Out[43]: <matplotlib.legend.Legend at 0x1b207452dd8>
```





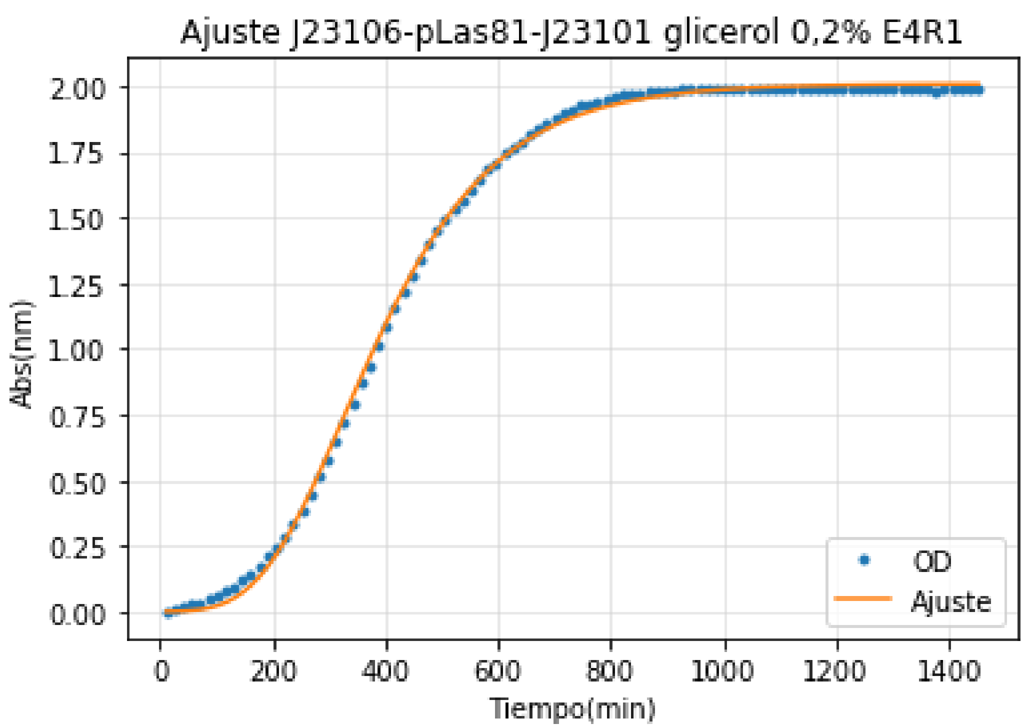

```
In [44]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-pLas81-J23101 Glucosa 0,4%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[16:26],od12271[16:26],'--',label='OD E4R1')
         plt.plot(tt[17:28],od12272[17:28],'--',label='OD E4R2')
         plt.plot(tt[16:26],od12273[16:26],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[44]: <matplotlib.legend.Legend at 0x1b208b987f0>
```



In [45]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-pLas-std glicerol rep 1
y28= np.log(od1227g1)-np.log(np.min(od1227g1))
print('Min OD = %e'%((np.min(od1227g1))))
evaly, params=Function_fit(tt,y28,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0,
A28= params[0]
um28=params[1]
l28=params[2]
print('A=%e'%(A28))
print('um=%e'%(um28))
print('l=%e'%(l28))

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

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

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

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

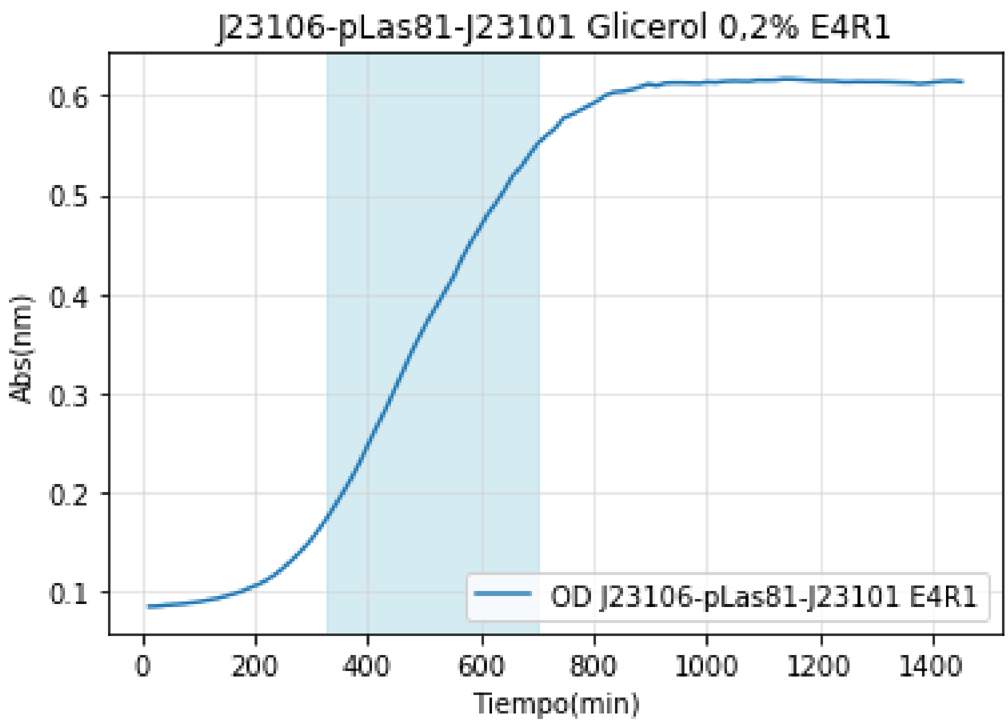
```
Min OD = 8.410000e-02
[  2.00880290e+00   4.95098248e-03   1.76571690e+02]
```

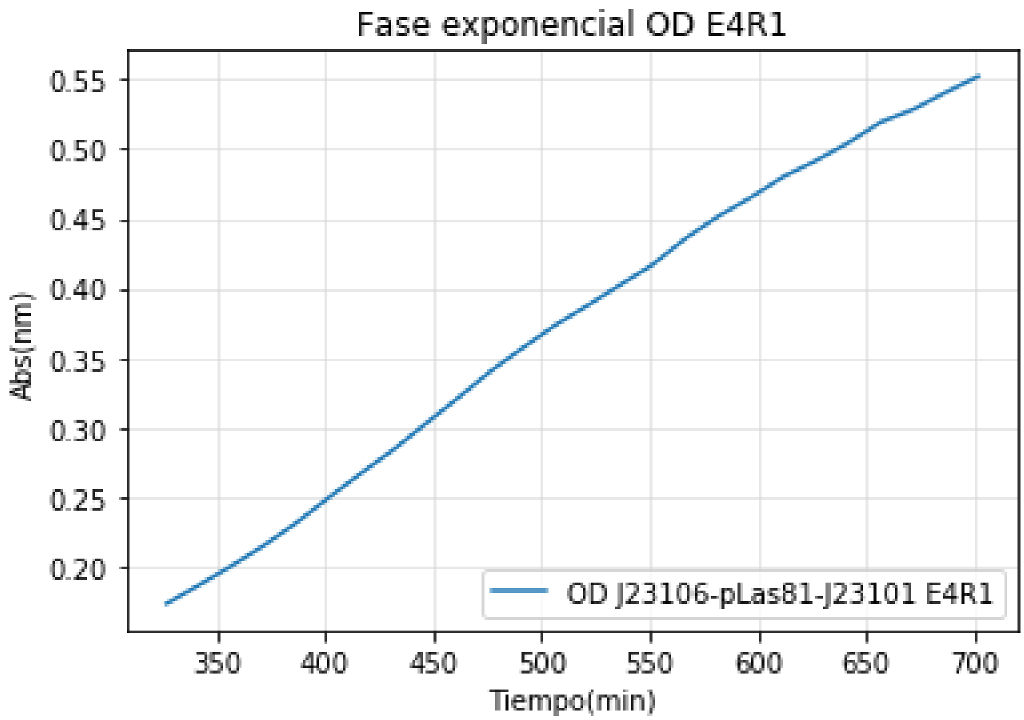



```
A=2.008803e+00
um=4.950982e-03
```

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

```
Out[45]: <matplotlib.legend.Legend at 0x1b208d36ac8>
```

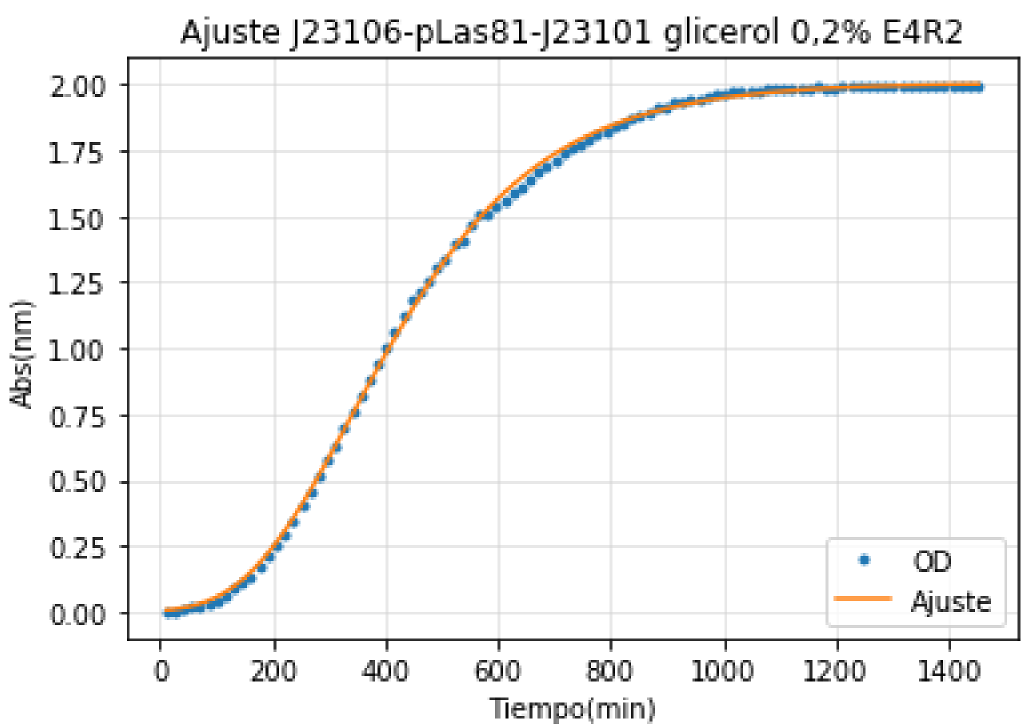




In [46]:
```
#Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
#106-pLas-std glicerol rep 2
y29= np.log(od1227g2)-np.log(np.min(od1227g2))
print('Min OD = %e'%((np.min(od1227g2))))
evaly, params=Function_fit(tt,y29,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0,
A29= params[0]
um29=params[1]
l29=params[2]
print('A=%e'%(A29))
print('um=%e'%(um29))
print('l=%e'%(l29))

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

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

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

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

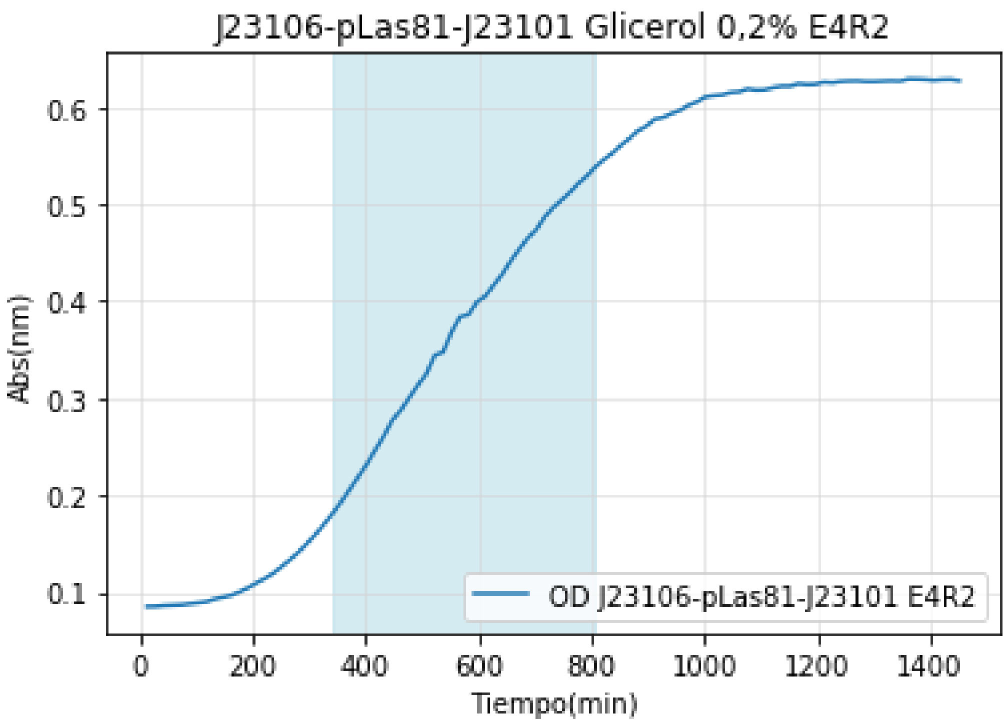
```
Min OD = 8.530000e-02
[  2.00719284e+00   3.95512057e-03   1.52315901e+02]
```

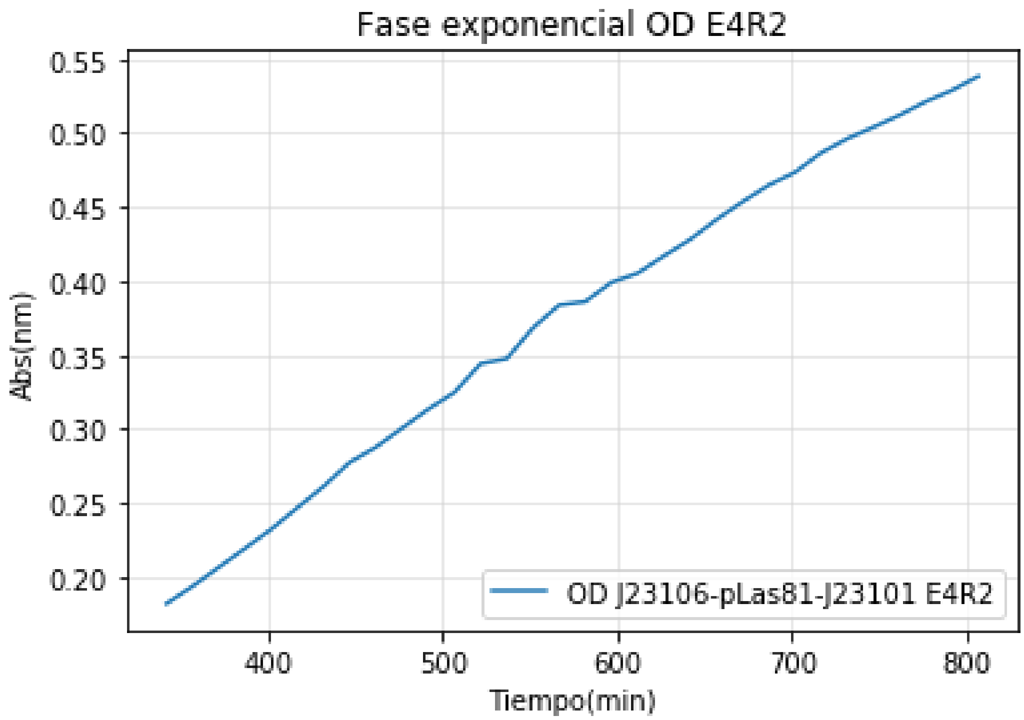



```
A=2.007193e+00
um=3.955121e-03
```

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

```
Out[46]: <matplotlib.legend.Legend at 0x1b208f50828>
```

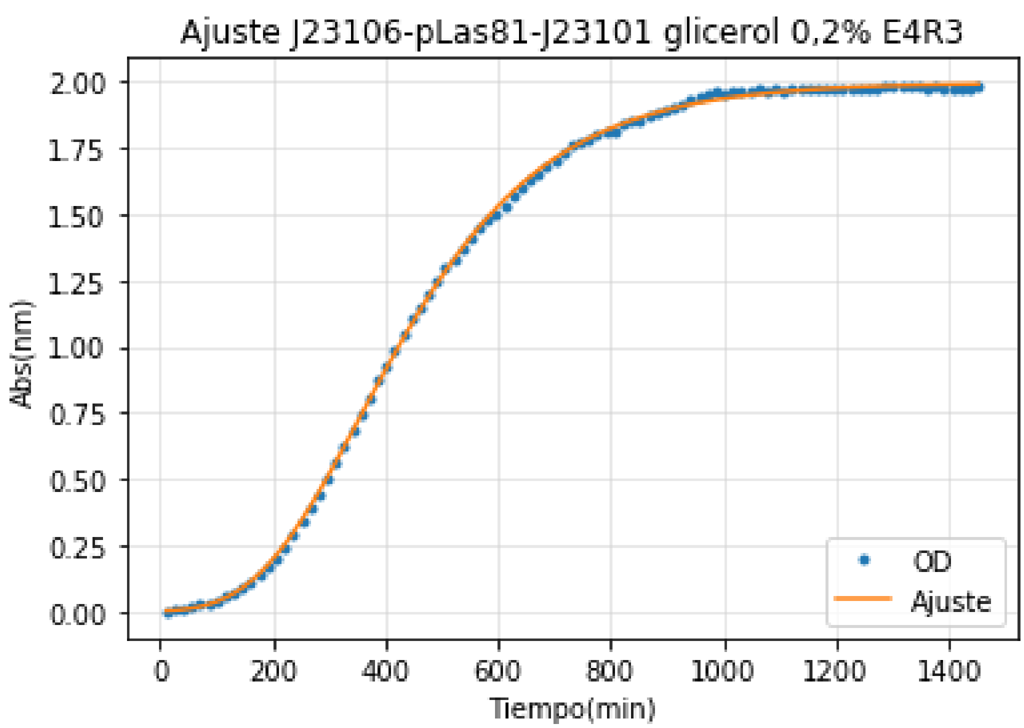





```
In [47]: #Ajuste de datos con modelo Gompertz y obtención de parámetros para el ajuste
         #106-pLas-std glicerol rep 3
         y30= np.log(od1227g3)-np.log(np.min(od1227g3))
         print('Min OD = %e'%((np.min(od1227g3))))
         evaly, params=Function_fit(tt,y30,0,-1,title = 'Ajuste J23106-pLas81-J23101 glicerol 0,
         A30= params[0]
         um30=params[1]
         l30=params[2]
         print('A=%e'%(A30))
         print('um=%e'%(um30))
         print('l=%e'%(l30))

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

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

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

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

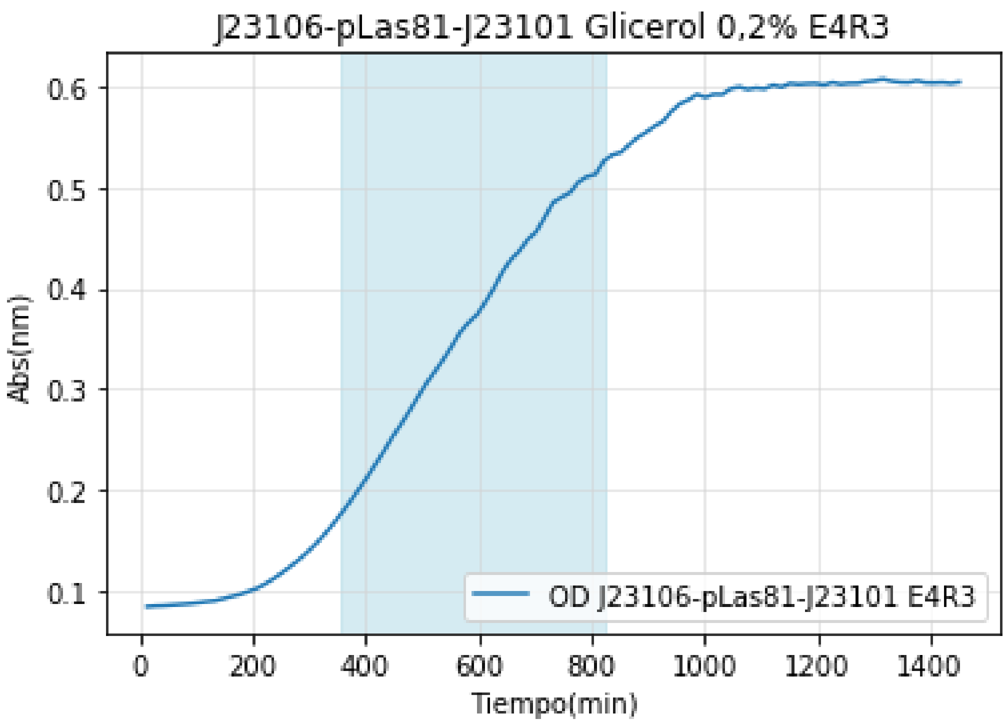
```
Min OD = 8.360000e-02
[  1.99609670e+00   3.97479297e-03   1.70376430e+02]
```

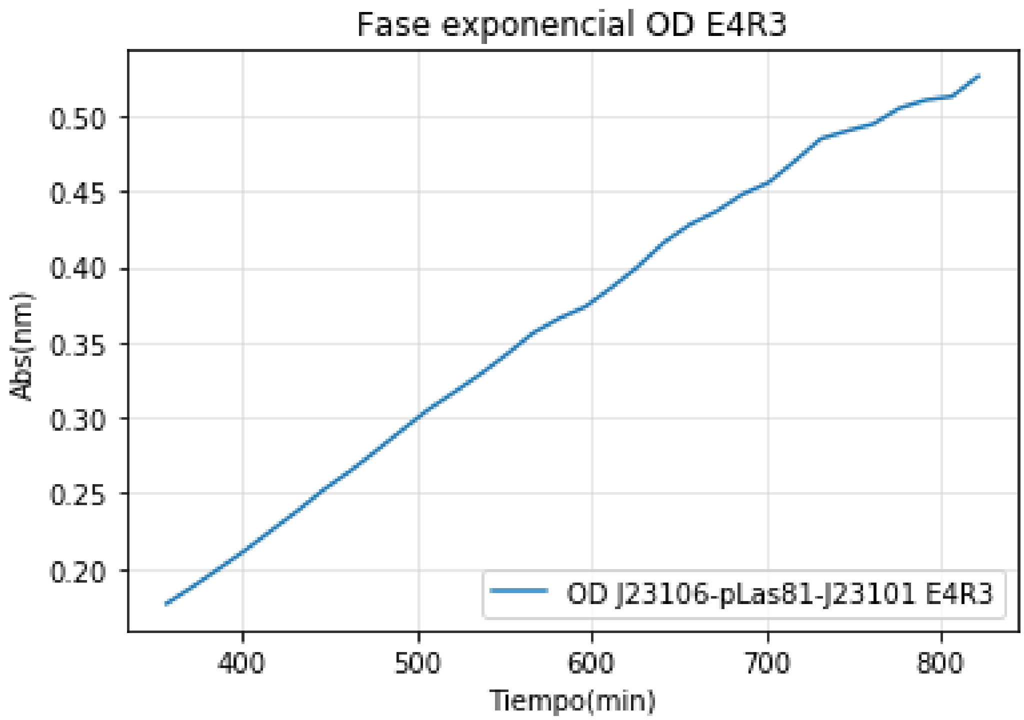



```
A=1.996097e+00
um=3.974793e-03
```

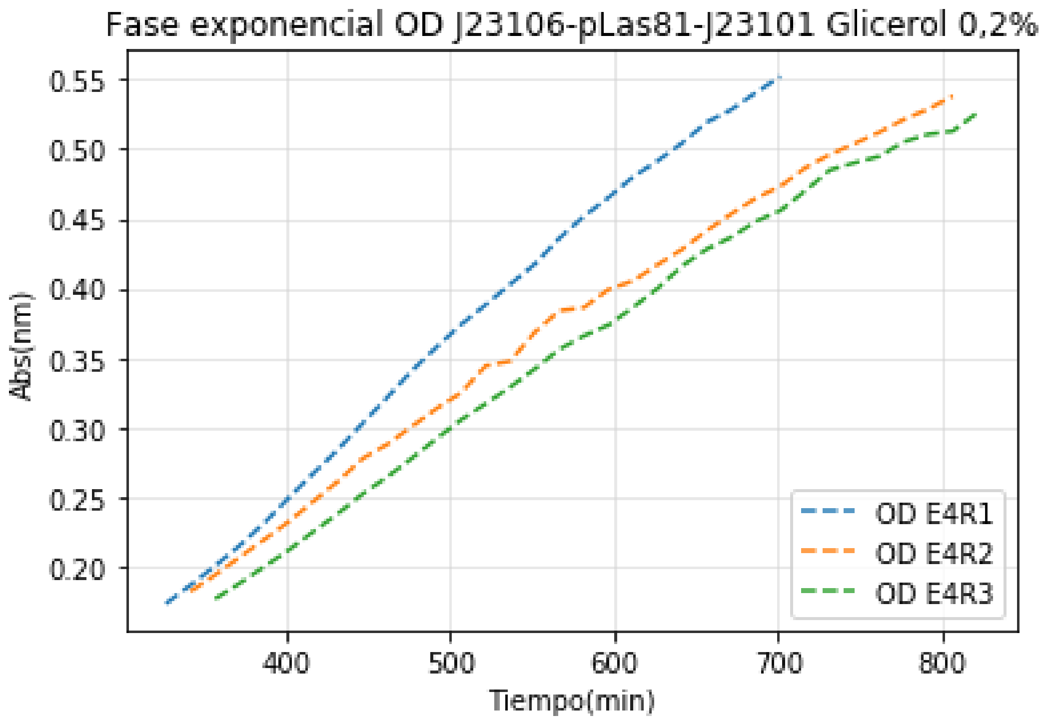
```
l=1.703764e+02
Tm=3.551214e+02
doubpe=1.743857e+02
ext=4.359643e+02
Tfinal=7.910857e+02
```

Out[47]: <matplotlib.legend.Legend at 0x1b208bb8908>



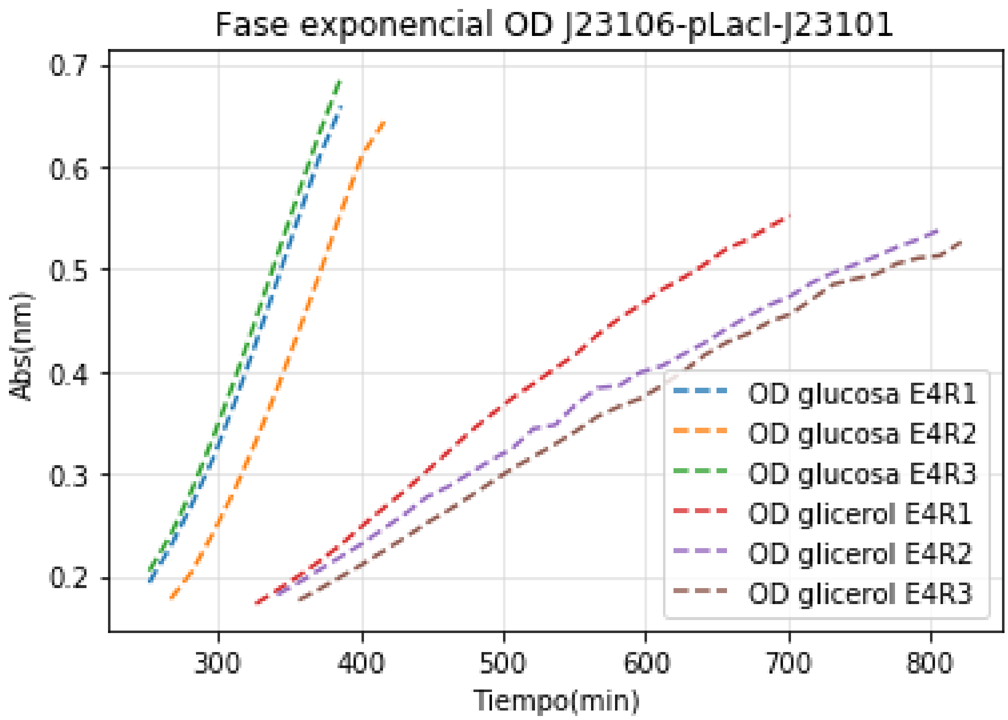



```
In [48]: #Fase exponencial OD/tiempo
         plt.figure()
         plt.title('Fase exponencial OD J23106-pLas81-J23101 Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel('Abs(nm)')
         plt.plot(tt[21:47],od1227g1[21:47],'--',label='OD E4R1')
         plt.plot(tt[22:54],od1227g2[22:54],'--',label='OD E4R2')
         plt.plot(tt[23:55],od1227g3[23:55],'--',label='OD E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
         plt.legend(loc='lower right')

Out[48]: <matplotlib.legend.Legend at 0x1b207095748>
```



In [49]:
```
#Fase exponencial OD/tiempo
plt.figure()
plt.title('Fase exponencial OD J23106-pLacI-J23101')
plt.xlabel('Tiempo(min)')
plt.ylabel('Abs(nm)')
plt.plot(tt[16:26],od12271[16:26],'--',label='OD glucosa E4R1')
plt.plot(tt[17:28],od12272[17:28],'--',label='OD glucosa E4R2')
plt.plot(tt[16:26],od12273[16:26],'--',label='OD glucosa E4R3')
plt.plot(tt[21:47],od1227g1[21:47],'--',label='OD glicerol E4R1')
plt.plot(tt[22:54],od1227g2[22:54],'--',label='OD glicerol E4R2')
plt.plot(tt[23:55],od1227g3[23:55],'--',label='OD glicerol E4R3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5)
plt.legend(loc='lower right')
```

Out[49]:
```
<matplotlib.legend.Legend at 0x1b20769f7f0>
```



In [50]:
```
#Selección de datos en arrays, según lo determinado
#controles glucosa
o1=odcg1[17:28]
c1=cfpcg1[17:28]
r1=rfpcg1[17:28]
y1=yfpcg1[17:28]

o2=odcg2[14:25]
c2=cfpcg2[14:25]
r2=rfpcg2[14:25]
y2=yfpcg2[14:25]

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

#controles glicerol
o4=odcgl1[24:52]
c4=cfpcgl1[24:52]
r4=rfpcgl1[24:52]
y4=yfpcgl1[24:52]

o5=odcgl2[25:53]
```

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

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

#106-std-std glucosa
o7=od12211[17:27]
c7=cfp12211[17:27]
r7=rfp12211[17:27]
y7=yfp12211[17:27]

o8=od12212[15:27]
c8=cfp12212[15:27]
r8=rfp12212[15:27]
y8=yfp12212[15:27]

o9=od12213[18:29]
c9=cfp12213[18:29]
r9=rfp12213[18:29]
y9=yfp12213[18:29]

#106-std-std glicerol
o10=od1221g1[28:58]
c10=cfp1221g1[28:58]
r10=rfp1221g1[28:58]
y10=yfp1221g1[28:58]

o11=od1221g2[23:50]
c11=cfp1221g2[23:50]
r11=rfp1221g2[23:50]
y11=yfp1221g2[23:50]

o12=od1221g3[29:52]
c12=cfp1221g3[29:52]
r12=rfp1221g3[29:52]
y12=yfp1221g3[29:52]

#106-107-std glucosa
o13=od12231[16:26]
c13=cfp12231[16:26]
r13=rfp12231[16:26]
y13=yfp12231[16:26]

o14=od12232[15:26]
```

```
c14=cfp12232[15:26]
r14=rfp12232[15:26]
y14=yfp12232[15:26]

o15=od12233[15:26]
c15=cfp12233[15:26]
r15=rfp12233[15:26]
y15=yfp12233[15:26]

#106-107-std glicerol
o16=od1223g1[25:53]
c16=cfp1223g1[25:53]
r16=rfp1223g1[25:53]
y16=yfp1223g1[25:53]

o17=od1223g2[20:49]
c17=cfp1223g2[20:49]
r17=rfp1223g2[20:49]
y17=yfp1223g2[20:49]

o18=od1223g3[25:54]
c18=cfp1223g3[25:54]
r18=rfp1223g3[25:54]
y18=yfp1223g3[25:54]

#106-pLac-std glucosa
o19=od12261[17:28]
c19=cfp12261[17:28]
r19=rfp12261[17:28]
y19=yfp12261[17:28]

o20=od12262[17:28]
c20=cfp12262[17:28]
r20=rfp12262[17:28]
y20=yfp12262[17:28]

o21=od12263[17:28]
c21=cfp12263[17:28]
r21=rfp12263[17:28]
y21=yfp12263[17:28]

#106-pLac-std glicerol
o22=od1226g1[26:49]
c22=cfp1226g1[26:49]
r22=rfp12261[26:49]
y22=yfp1226g1[26:49]

o23=od1226g2[23:53]
```

```
c23=cfp1226g2[23:53]
r23=rfp1226g2[23:53]
y23=yfp12262[23:53]

o24=od1226g3[27:57]
c24=cfp1226g3[27:57]
r24=rfp1226g3[27:57]
y24=yfp1226g3[27:57]

#106-pLas-std glucosa
o25=od12271[16:26]
c25=cfp12271[16:26]
r25=rfp12271[16:26]
y25=yfp12271[16:26]

o26=od12272[17:28]
c26=cfp12272[17:28]
r26=rfp12272[17:28]
y26=yfp12272[17:28]

o27=od12273[16:26]
c27=cfp12273[16:26]
r27=rfp12273[16:26]
y27=yfp12273[16:26]

#plux-pLas-std glicerol
o28=od1227g1[21:47]
c28=cfp1227g1[21:47]
r28=rfp1227g1[21:47]
y28=yfp1227g1[21:47]

o29=od1227g2[22:54]
c29=cfp1227g2[22:54]
r29=rfp1227g2[22:54]
y29=yfp1227g2[22:54]

o30=od1227g3[23:55]
c30=cfp1227g3[23:55]
r30=rfp1227g3[23:55]
y30=yfp1227g3[23:55]
```

```
In [51]: #regresion lineal de replicas
         #Controles glucosa
         slope, intercept, r_value, p_value,std_err=stats.linregress(o1,c1)
         slopec1=slope
         slope, intercept, r_value, p_value,std_err=stats.linregress(o1,r1)
         sloper1=slope
         slope, intercept, r_value, p_value,std_err=stats.linregress(o1,y1)
```

```
slopey1=slope

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

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

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

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

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

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

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

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

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

#106-std-std glicerol
slope, intercept, r_value, p_value,std_err=stats.linregress(o10,c10)
slopec10=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o10,r10)
sloper10=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o10,y10)
slopey10=slope

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

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

#106-107-std glucosa
slope, intercept, r_value, p_value,std_err=stats.linregress(o13,c13)
slopec13=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o13,r13)
sloper13=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o13,y13)
slopey13=slope

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

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

#106-107-std glicerol
slope, intercept, r_value, p_value,std_err=stats.linregress(o16,c16)
slopec16=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o16,r16)
sloper16=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o16,y16)
slopey16=slope

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

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

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

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

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

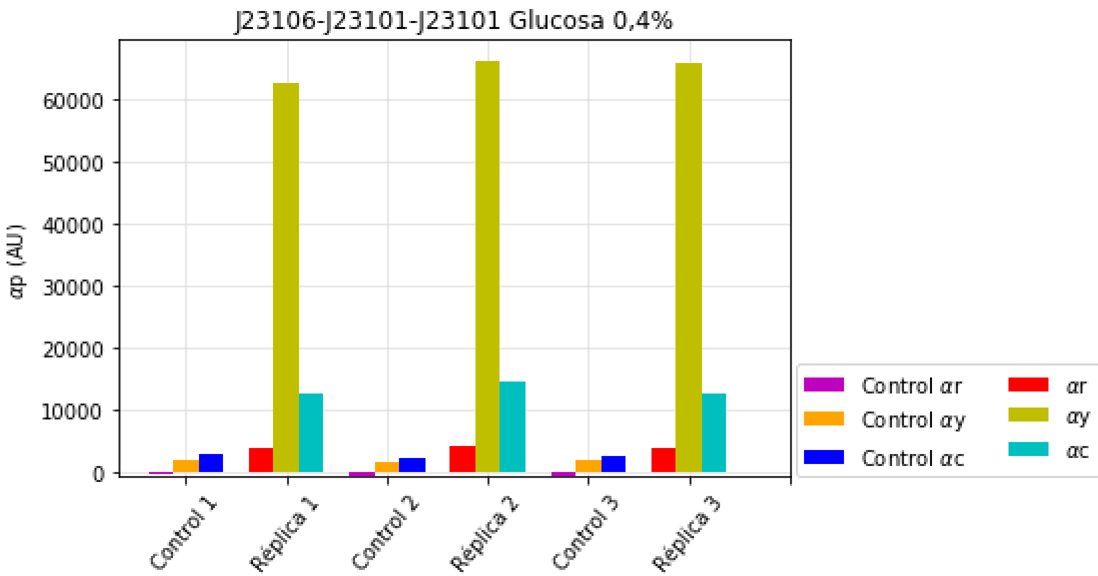
```python
sloper21=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o21,y21)
slopey21=slope

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

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

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

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

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

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

slope, intercept, r_value, p_value,std_err=stats.linregress(o30,c30)
slopec30=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o30,r30)
sloper30=slope
slope, intercept, r_value, p_value,std_err=stats.linregress(o30,y30)
slopey30=slope
```

```
In [52]: pendientesc=[slopec1,slopec2,slopec3,slopec4,slopec5,slopec6,slopec7,slopec8,slopec9,sl
pendientesr=[sloper1,sloper2,sloper3,sloper4,sloper5,sloper6,sloper7,sloper8,sloper9,sl
pendientesy=[slopey1,slopey2,slopey3,slopey4,slopey5,slopey6,slopey7,slopey8,slopey9,sl

#Grafico pendientes 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,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[6],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[6],color='c',width=0.25,label= r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
```

```
plt.bar(X[5]+0.25,pendientesc[8],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

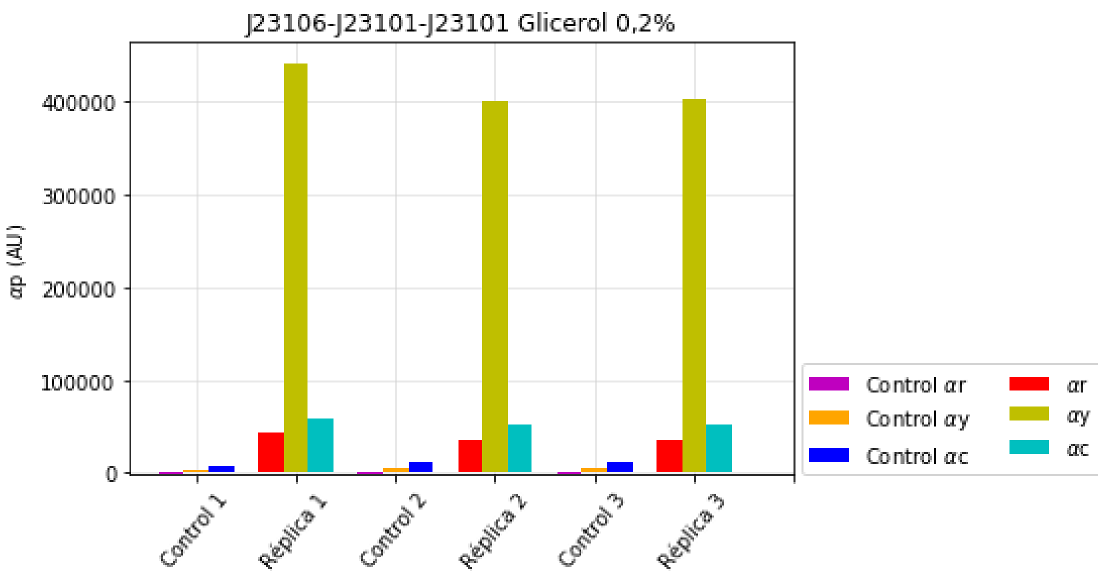
Out[52]: <matplotlib.legend.Legend at 0x1b207422668>




```
In [53]: #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,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[9],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[9],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[9],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
```

```
plt.bar(X[5]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[11],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]

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

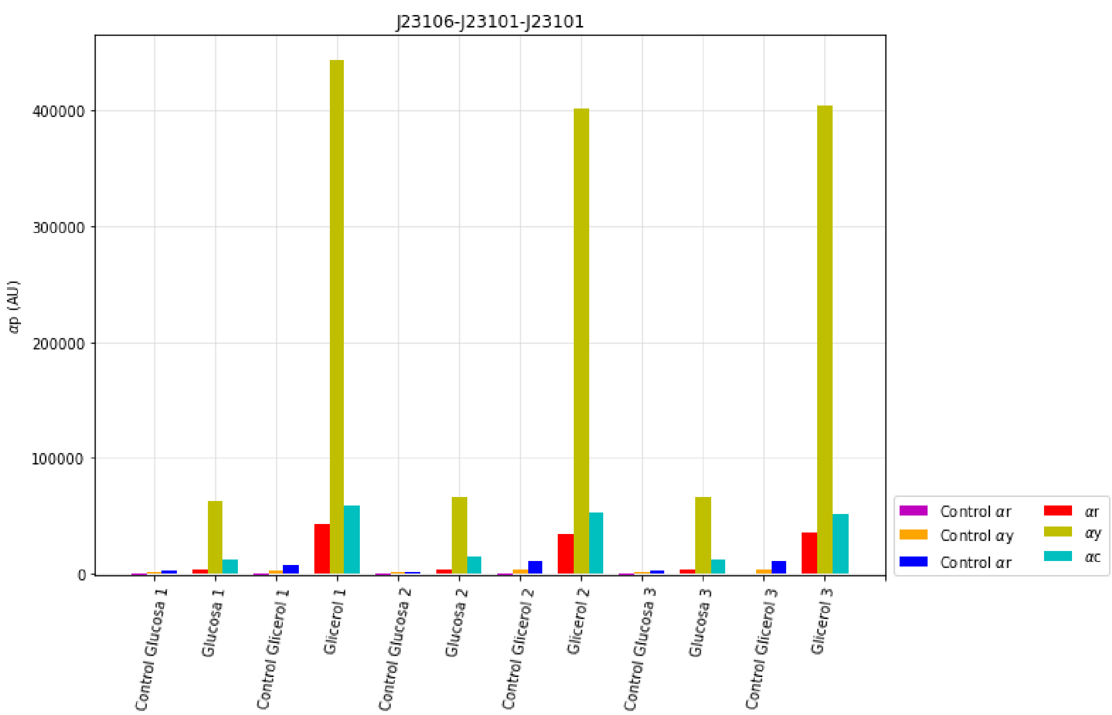
Out[53]: <matplotlib.legend.Legend at 0x1b20926bcf8>



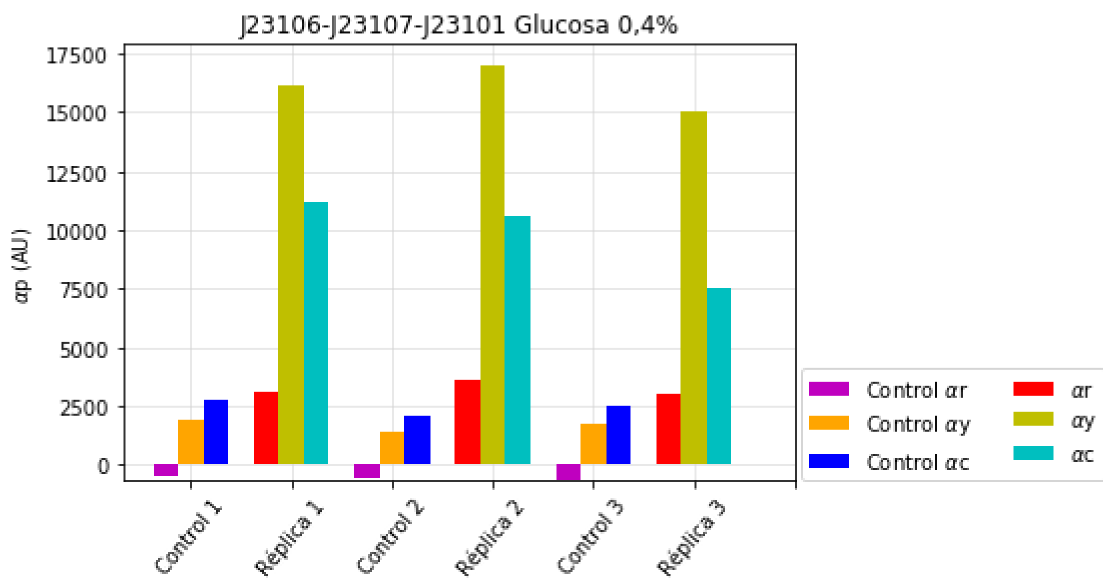

```
In [54]: #Grafico pendientes plux-std-std
X = np.arange(13)
plt.figure(figsize=(10,7))
plt.title('J23106-J23101-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[1]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[6],color='y',width=0.25,label= r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[6],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[9],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[9],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[9],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
```

```
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[8],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[11],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

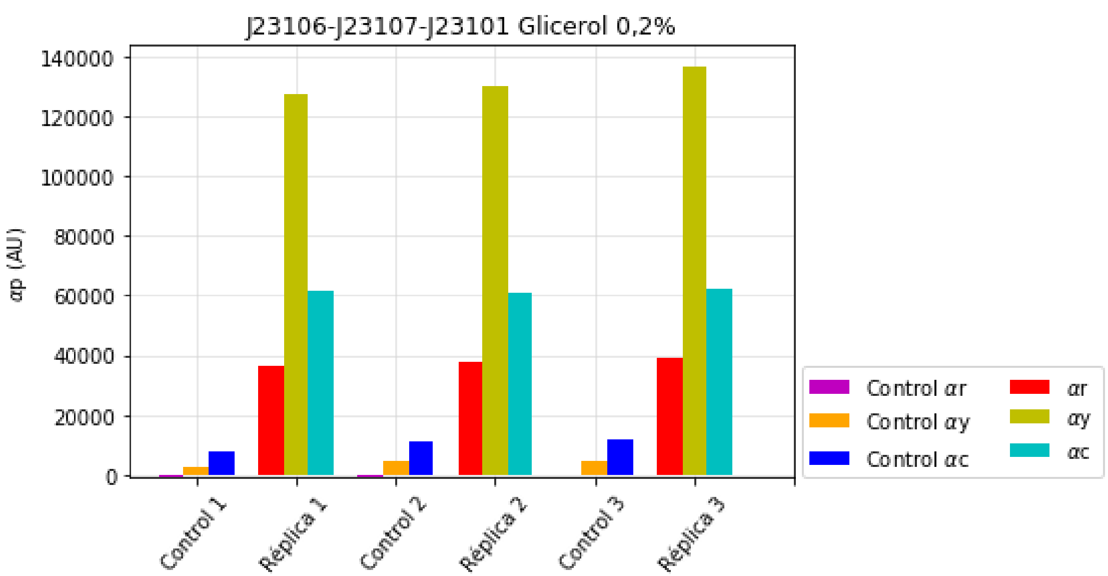
Out[54]: <matplotlib.legend.Legend at 0x1b20941b0b8>



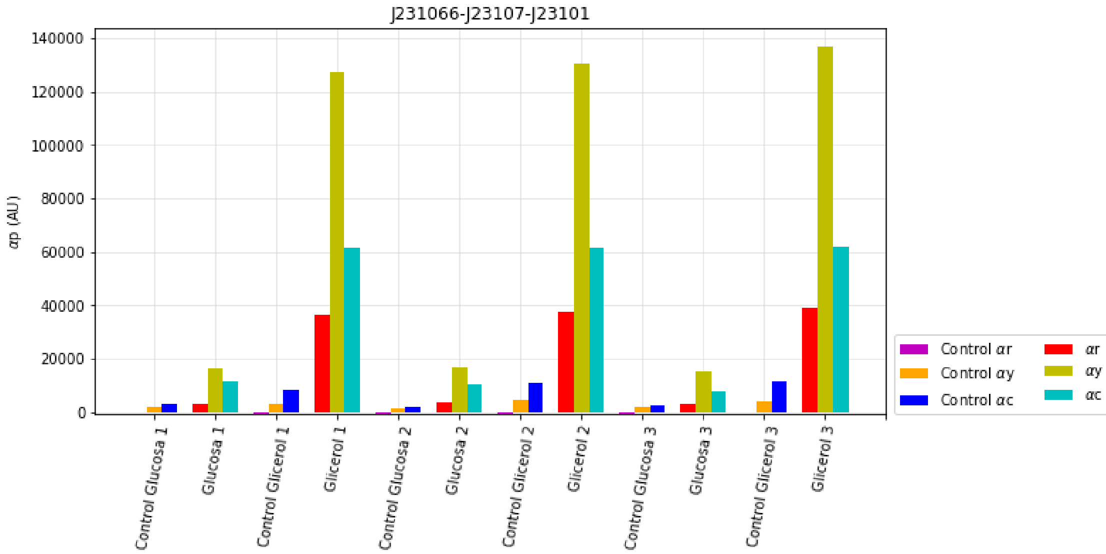
```
In [55]: #Grafico pendientes 106-107-std Glucosa
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-J23107-J23101 Glucosa 0,4%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[12],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[12],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

Out[55]: <matplotlib.legend.Legend at 0x1b208bdde48>
```



```
In [56]: #Grafico pendientes 106-107-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-J23107-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[15],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[15],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[15],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

Out[56]: <matplotlib.legend.Legend at 0x1b208e45978>
```

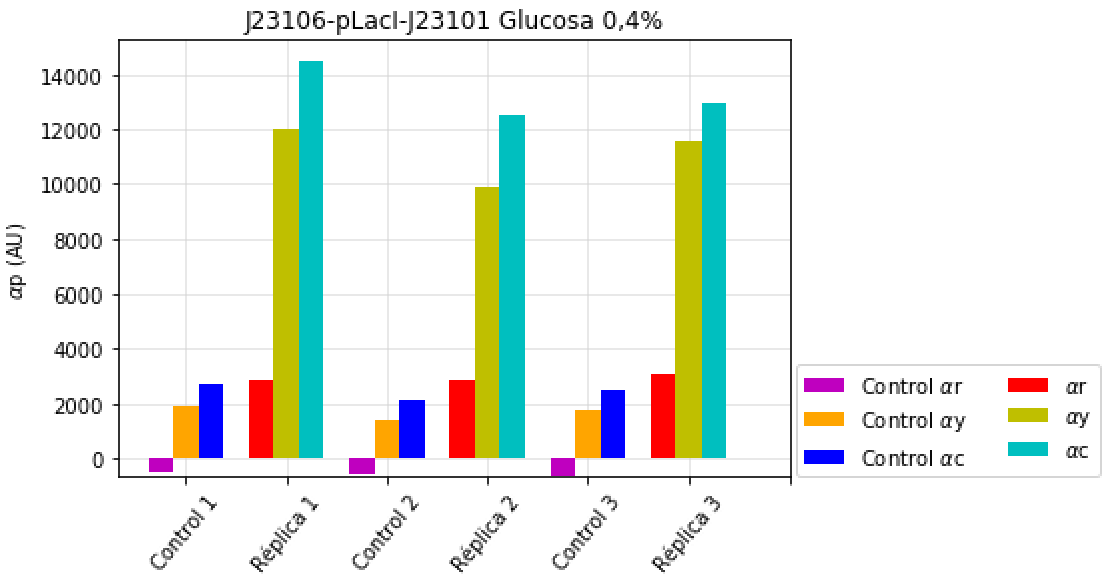


```
In [57]: #Grafico pendientes 106-107-std
         X = np.arange(13)
         plt.figure(figsize=(10,5))
         plt.title('J231066-J23107-J23101')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[12],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[12],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[12],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[15],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[7]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
         plt.bar(X[7]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
         plt.bar(X[7]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
         plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
         plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
         plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
         plt.bar(X[9]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
         plt.bar(X[9]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
         plt.bar(X[9]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
         plt.bar(X[10]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
         plt.bar(X[10]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
         plt.bar(X[10]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
         plt.bar(X[11]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
         plt.bar(X[11]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
         plt.bar(X[11]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Contr
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

Out[57]: <matplotlib.legend.Legend at 0x1b209593208>
```

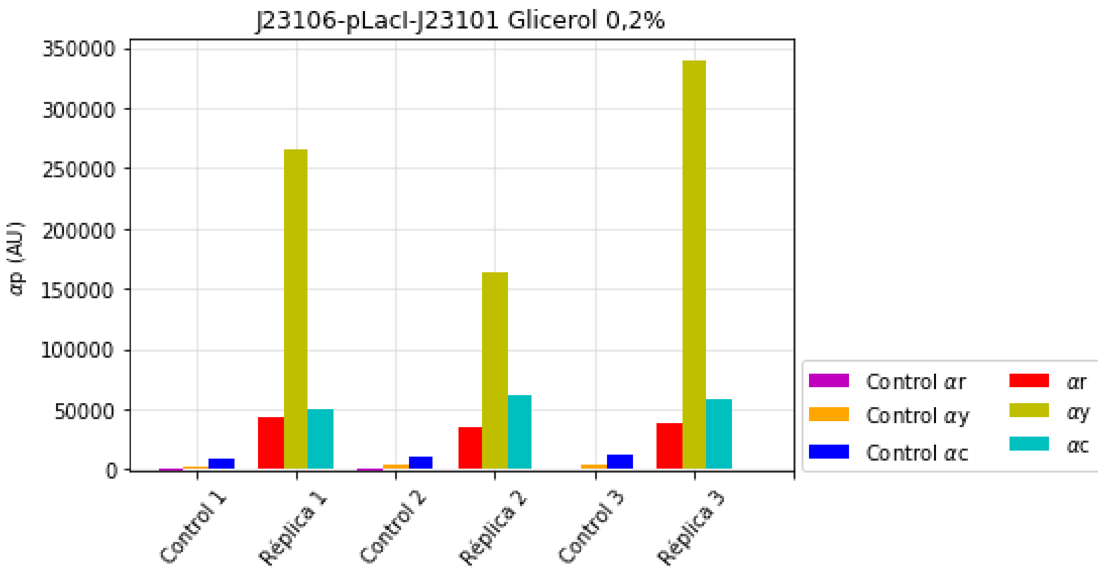


In [58]:
```
#Grafico pendientes 106-plac-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('J23106-pLacI-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[18],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[18],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[18],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[20],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[20],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[20],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[58]: <matplotlib.legend.Legend at 0x1b2099c3c88>



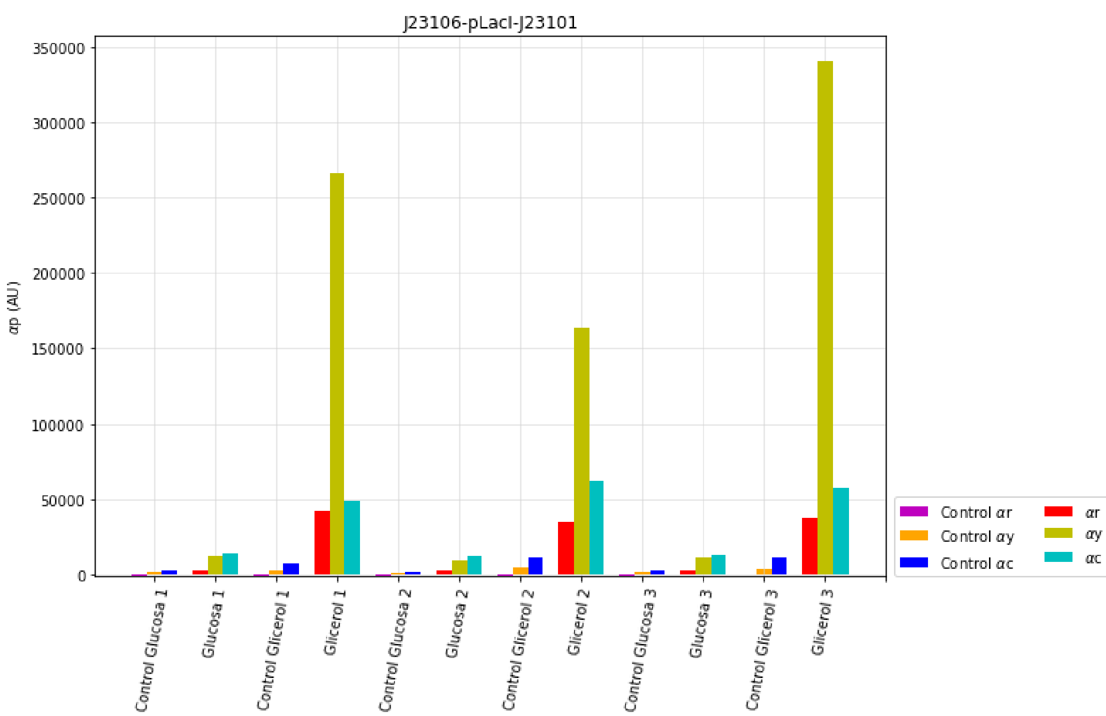

```
In [59]: #Grafico pendientes 106-plac-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-pLacI-J23101 Glicerol 0,2%')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[21],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[21],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[21],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[22],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[23],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[23],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[23],color='c',width=0.25,zorder=3)
         plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

Out[59]: <matplotlib.legend.Legend at 0x1b209b06ac8>
```




```
In [60]: #Grafico pendientes 106-plac-std
         X = np.arange(13)
         plt.figure(figsize=(10,7))
         plt.title('J23106-pLacI-J23101')
         plt.ylabel(r'$\alpha$p (AU)')
         plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
         plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
         plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
         plt.bar(X[1]-0.25,pendientesr[18],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
         plt.bar(X[1]+0.00,pendientesy[18],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
         plt.bar(X[1]+0.25,pendientesc[18],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
         plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
         plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,pendientesr[21],color='r',width=0.25,zorder=3)
         plt.bar(X[3]+0.00,pendientesy[21],color='y',width=0.25,zorder=3)
         plt.bar(X[3]+0.25,pendientesc[21],color='c',width=0.25,zorder=3)
         plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
         plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
         plt.bar(X[5]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
         plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
         plt.bar(X[7]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
         plt.bar(X[7]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
```

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

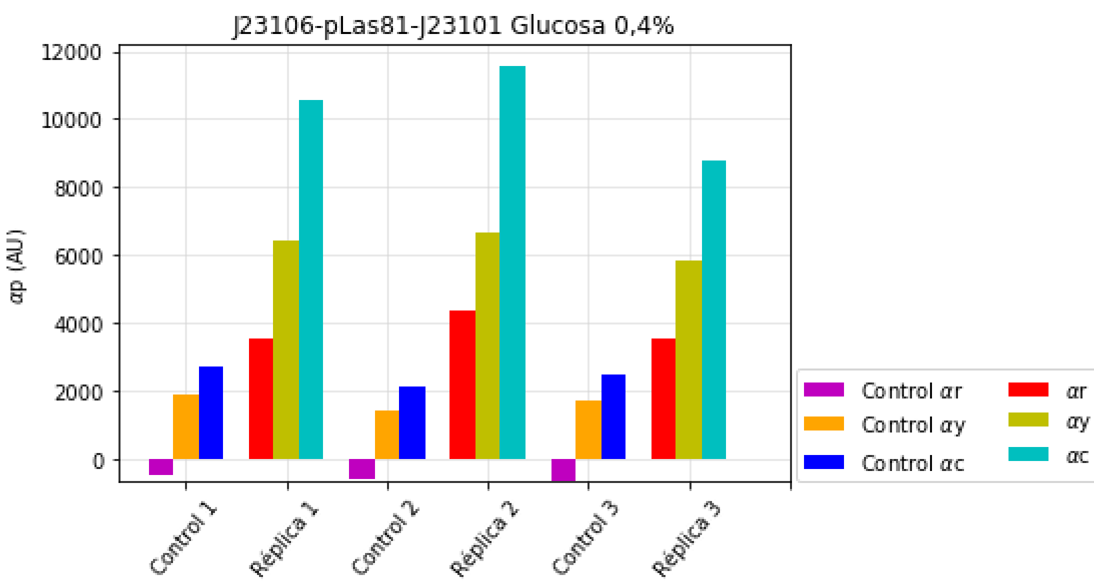
Out[60]: <matplotlib.legend.Legend at 0x1b209cc6dd8>



In [61]:
```
#Grafico pendientes 106-pLas-std Glucosa
X = np.arange(7)
plt.figure()
plt.title('J23106-pLas81-J23101 Glucosa 0,4%')
plt.ylabel(r'$\alpha$p (AU)')
```

```
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[24],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[24],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[26],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

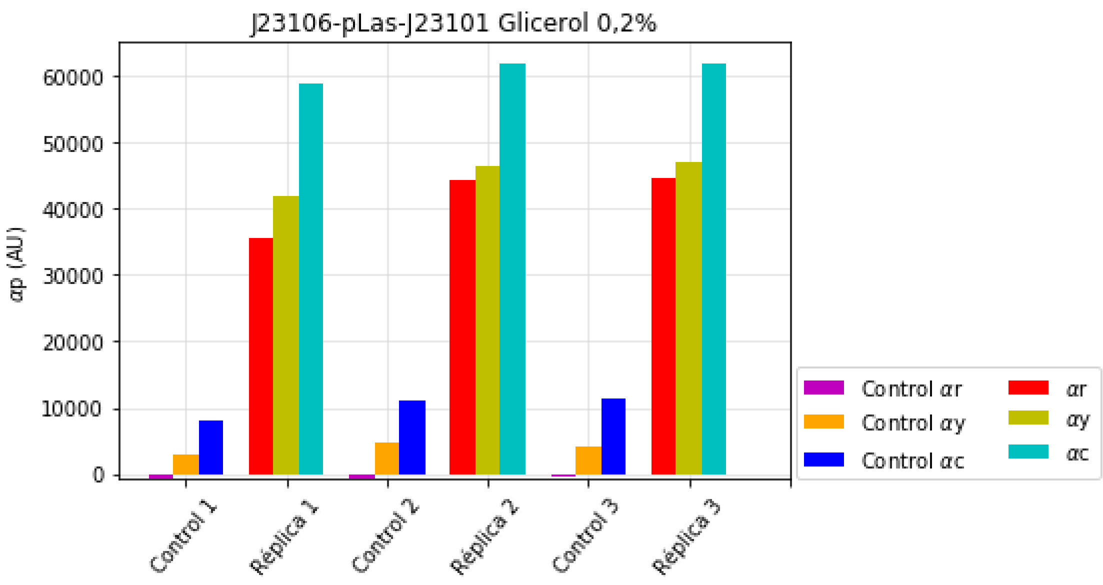
Out[61]: <matplotlib.legend.Legend at 0x1b209416d68>




```
In [62]: #Grafico pendientes 106-plas-std Glicerol
         X = np.arange(7)
         plt.figure()
         plt.title('J23106-pLas-J23101 Glicerol 0,2%')
```

```
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[3],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[3],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[3],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[27],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[27],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[27],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[29],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control 1',"Réplica 1",'Control 2',"Réplica 2",'Control 3',"Réplica 3"]
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

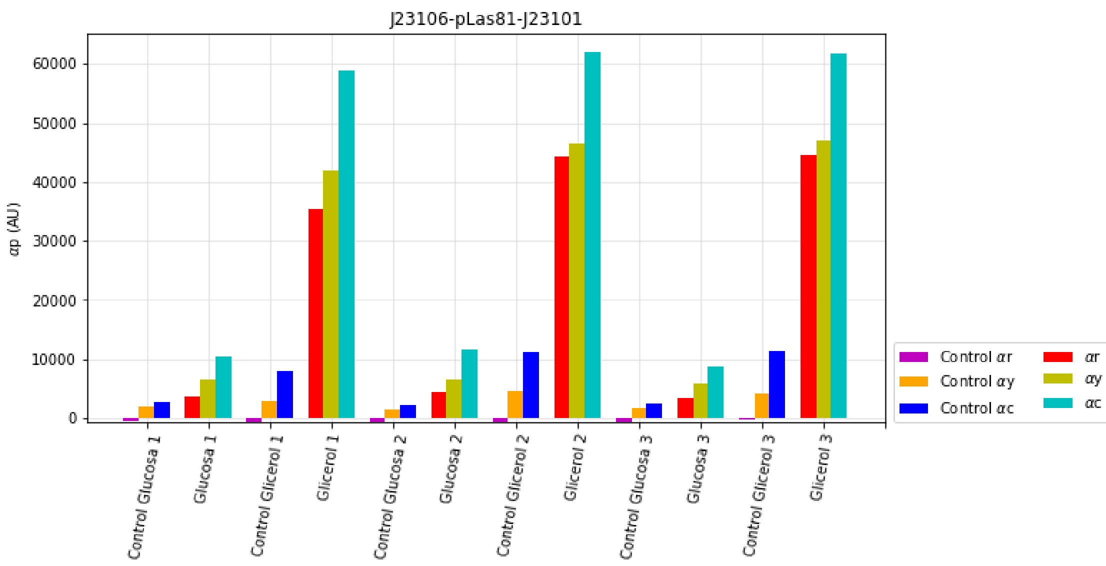
Out[62]: <matplotlib.legend.Legend at 0x1b20a182550>




```
In [63]: #Grafico pendientes 106-plas-std
         X = np.arange(13)
         plt.figure(figsize=(10,5))
```

```
plt.title('J23106-pLas81-J23101')
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[24],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[1]+0.00,pendientesy[24],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[1]+0.25,pendientesc[24],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[2]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[27],color='r',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[27],color='y',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[27],color='c',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[6]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[6]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[7]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[7]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[26],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[29],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1',"Glucosa 1",'Control Glicerol 1',"Glicerol 1",'Contr
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[63]: <matplotlib.legend.Legend at 0x1b20a32bb00>
```

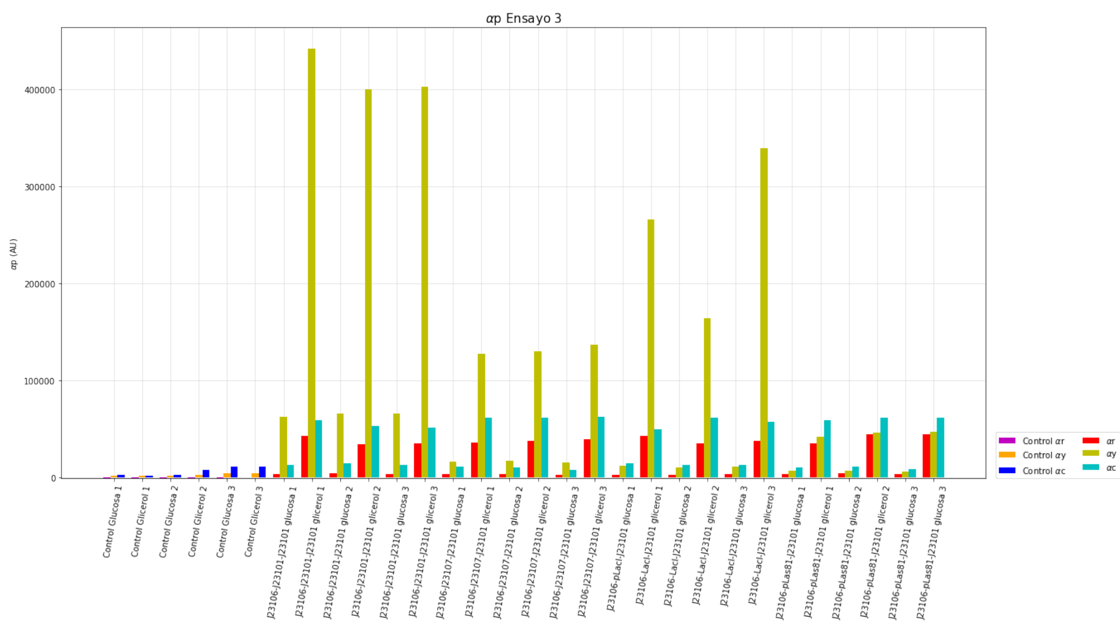


In [64]:
```
#Grafico pendientes todo
X = np.arange(30)
plt.figure(figsize=(20,10))
plt.title(r'$\alpha$p Ensayo 3',fontsize=15.0)
plt.ylabel(r'$\alpha$p (AU)')
plt.bar(X[0]-0.25,pendientesr[0],color='m',width=0.25,label='Control'+' '+ r'$\alpha$r'
plt.bar(X[0]+0.00,pendientesy[0],color='orange',width=0.25,label='Control'+' '+ r'$\alp
plt.bar(X[0]+0.25,pendientesc[0],color='b',width=0.25,label='Control'+' '+ r'$\alpha$c'
plt.bar(X[1]-0.25,pendientesr[1],color='m',width=0.25,zorder=3)
plt.bar(X[1]+0.00,pendientesy[1],color='orange',width=0.25,zorder=3)
plt.bar(X[1]+0.25,pendientesc[1],color='b',width=0.25,zorder=3)
plt.bar(X[2]-0.25,pendientesr[2],color='m',width=0.25,zorder=3)
plt.bar(X[2]+0.00,pendientesy[2],color='orange',width=0.25,zorder=3)
plt.bar(X[2]+0.25,pendientesc[2],color='b',width=0.25,zorder=3)
plt.bar(X[3]-0.25,pendientesr[3],color='m',width=0.25,zorder=3)
plt.bar(X[3]+0.00,pendientesy[3],color='orange',width=0.25,zorder=3)
plt.bar(X[3]+0.25,pendientesc[3],color='b',width=0.25,zorder=3)
plt.bar(X[4]-0.25,pendientesr[4],color='m',width=0.25,zorder=3)
plt.bar(X[4]+0.00,pendientesy[4],color='orange',width=0.25,zorder=3)
plt.bar(X[4]+0.25,pendientesc[4],color='b',width=0.25,zorder=3)
plt.bar(X[5]-0.25,pendientesr[5],color='m',width=0.25,zorder=3)
plt.bar(X[5]+0.00,pendientesy[5],color='orange',width=0.25,zorder=3)
plt.bar(X[5]+0.25,pendientesc[5],color='b',width=0.25,zorder=3)
plt.bar(X[6]-0.25,pendientesr[6],color='r',width=0.25,label=r'$\alpha$r',zorder=3)
plt.bar(X[6]+0.00,pendientesy[6],color='y',width=0.25,label=r'$\alpha$y',zorder=3)
plt.bar(X[6]+0.25,pendientesc[6],color='c',width=0.25,label=r'$\alpha$c',zorder=3)
plt.bar(X[7]-0.25,pendientesr[9],color='r',width=0.25,zorder=3)
plt.bar(X[7]+0.00,pendientesy[9],color='y',width=0.25,zorder=3)
```

```
plt.bar(X[7]+0.25,pendientesc[9],color='c',width=0.25,zorder=3)
plt.bar(X[8]-0.25,pendientesr[7],color='r',width=0.25,zorder=3)
plt.bar(X[8]+0.00,pendientesy[7],color='y',width=0.25,zorder=3)
plt.bar(X[8]+0.25,pendientesc[7],color='c',width=0.25,zorder=3)
plt.bar(X[9]-0.25,pendientesr[10],color='r',width=0.25,zorder=3)
plt.bar(X[9]+0.00,pendientesy[10],color='y',width=0.25,zorder=3)
plt.bar(X[9]+0.25,pendientesc[10],color='c',width=0.25,zorder=3)
plt.bar(X[10]-0.25,pendientesr[8],color='r',width=0.25,zorder=3)
plt.bar(X[10]+0.00,pendientesy[8],color='y',width=0.25,zorder=3)
plt.bar(X[10]+0.25,pendientesc[8],color='c',width=0.25,zorder=3)
plt.bar(X[11]-0.25,pendientesr[11],color='r',width=0.25,zorder=3)
plt.bar(X[11]+0.00,pendientesy[11],color='y',width=0.25,zorder=3)
plt.bar(X[11]+0.25,pendientesc[11],color='c',width=0.25,zorder=3)
plt.bar(X[12]-0.25,pendientesr[12],color='r',width=0.25,zorder=3)
plt.bar(X[12]+0.00,pendientesy[12],color='y',width=0.25,zorder=3)
plt.bar(X[12]+0.25,pendientesc[12],color='c',width=0.25,zorder=3)
plt.bar(X[13]-0.25,pendientesr[15],color='r',width=0.25,zorder=3)
plt.bar(X[13]+0.00,pendientesy[15],color='y',width=0.25,zorder=3)
plt.bar(X[13]+0.25,pendientesc[15],color='c',width=0.25,zorder=3)
plt.bar(X[14]-0.25,pendientesr[13],color='r',width=0.25,zorder=3)
plt.bar(X[14]+0.00,pendientesy[13],color='y',width=0.25,zorder=3)
plt.bar(X[14]+0.25,pendientesc[13],color='c',width=0.25,zorder=3)
plt.bar(X[15]-0.25,pendientesr[16],color='r',width=0.25,zorder=3)
plt.bar(X[15]+0.00,pendientesy[16],color='y',width=0.25,zorder=3)
plt.bar(X[15]+0.25,pendientesc[16],color='c',width=0.25,zorder=3)
plt.bar(X[16]-0.25,pendientesr[14],color='r',width=0.25,zorder=3)
plt.bar(X[16]+0.00,pendientesy[14],color='y',width=0.25,zorder=3)
plt.bar(X[16]+0.25,pendientesc[14],color='c',width=0.25,zorder=3)
plt.bar(X[17]-0.25,pendientesr[17],color='r',width=0.25,zorder=3)
plt.bar(X[17]+0.00,pendientesy[17],color='y',width=0.25,zorder=3)
plt.bar(X[17]+0.25,pendientesc[17],color='c',width=0.25,zorder=3)
plt.bar(X[18]-0.25,pendientesr[18],color='r',width=0.25,zorder=3)
plt.bar(X[18]+0.00,pendientesy[18],color='y',width=0.25,zorder=3)
plt.bar(X[18]+0.25,pendientesc[18],color='c',width=0.25,zorder=3)
plt.bar(X[19]-0.25,pendientesr[21],color='r',width=0.25,zorder=3)
plt.bar(X[19]+0.00,pendientesy[21],color='y',width=0.25,zorder=3)
plt.bar(X[19]+0.25,pendientesc[21],color='c',width=0.25,zorder=3)
plt.bar(X[20]-0.25,pendientesr[19],color='r',width=0.25,zorder=3)
plt.bar(X[20]+0.00,pendientesy[19],color='y',width=0.25,zorder=3)
plt.bar(X[20]+0.25,pendientesc[19],color='c',width=0.25,zorder=3)
plt.bar(X[21]-0.25,pendientesr[22],color='r',width=0.25,zorder=3)
plt.bar(X[21]+0.00,pendientesy[22],color='y',width=0.25,zorder=3)
plt.bar(X[21]+0.25,pendientesc[22],color='c',width=0.25,zorder=3)
plt.bar(X[22]-0.25,pendientesr[20],color='r',width=0.25,zorder=3)
plt.bar(X[22]+0.00,pendientesy[20],color='y',width=0.25,zorder=3)
plt.bar(X[22]+0.25,pendientesc[20],color='c',width=0.25,zorder=3)
plt.bar(X[23]-0.25,pendientesr[23],color='r',width=0.25,zorder=3)
plt.bar(X[23]+0.00,pendientesy[23],color='y',width=0.25,zorder=3)
```

```
plt.bar(X[23]+0.25,pendientesc[23],color='c',width=0.25,zorder=3)
plt.bar(X[24]-0.25,pendientesr[24],color='r',width=0.25,zorder=3)
plt.bar(X[24]+0.00,pendientesy[24],color='y',width=0.25,zorder=3)
plt.bar(X[24]+0.25,pendientesc[24],color='c',width=0.25,zorder=3)
plt.bar(X[25]-0.25,pendientesr[27],color='r',width=0.25,zorder=3)
plt.bar(X[25]+0.00,pendientesy[27],color='y',width=0.25,zorder=3)
plt.bar(X[25]+0.25,pendientesc[27],color='c',width=0.25,zorder=3)
plt.bar(X[26]-0.25,pendientesr[25],color='r',width=0.25,zorder=3)
plt.bar(X[26]+0.00,pendientesy[25],color='y',width=0.25,zorder=3)
plt.bar(X[26]+0.25,pendientesc[25],color='c',width=0.25,zorder=3)
plt.bar(X[27]-0.25,pendientesr[28],color='r',width=0.25,zorder=3)
plt.bar(X[27]+0.00,pendientesy[28],color='y',width=0.25,zorder=3)
plt.bar(X[27]+0.25,pendientesc[28],color='c',width=0.25,zorder=3)
plt.bar(X[28]-0.25,pendientesr[26],color='r',width=0.25,zorder=3)
plt.bar(X[28]+0.00,pendientesy[26],color='y',width=0.25,zorder=3)
plt.bar(X[28]+0.25,pendientesc[26],color='c',width=0.25,zorder=3)
plt.bar(X[29]-0.25,pendientesr[29],color='r',width=0.25,zorder=3)
plt.bar(X[29]+0.00,pendientesy[29],color='y',width=0.25,zorder=3)
plt.bar(X[29]+0.25,pendientesc[29],color='c',width=0.25,zorder=3)
plt.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control Gl
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

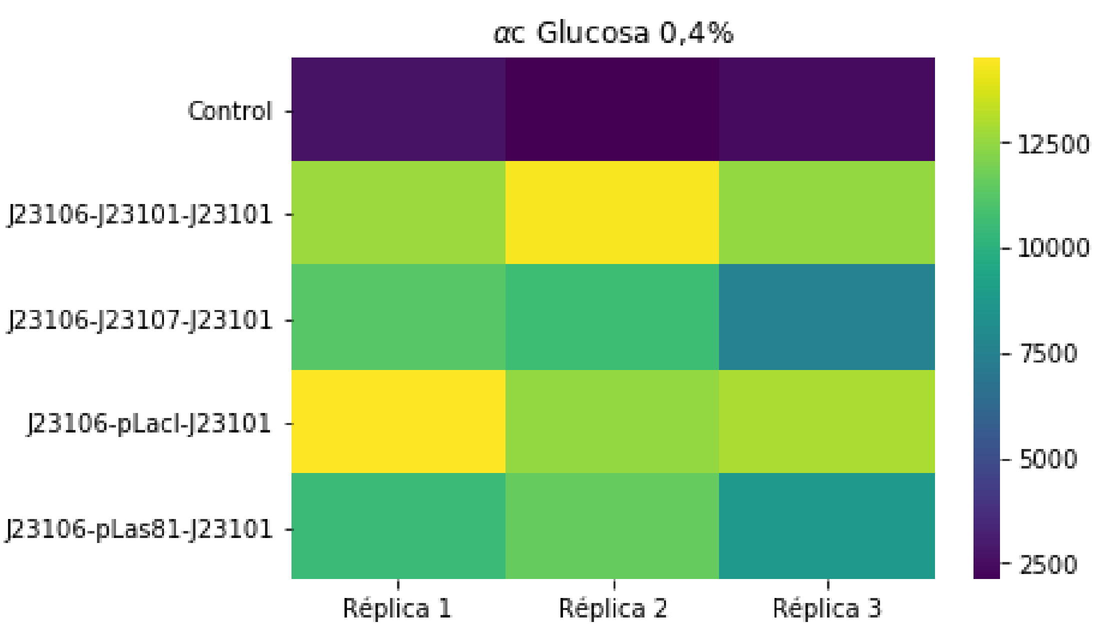
Out[64]: <matplotlib.legend.Legend at 0x1b20b55fb00>

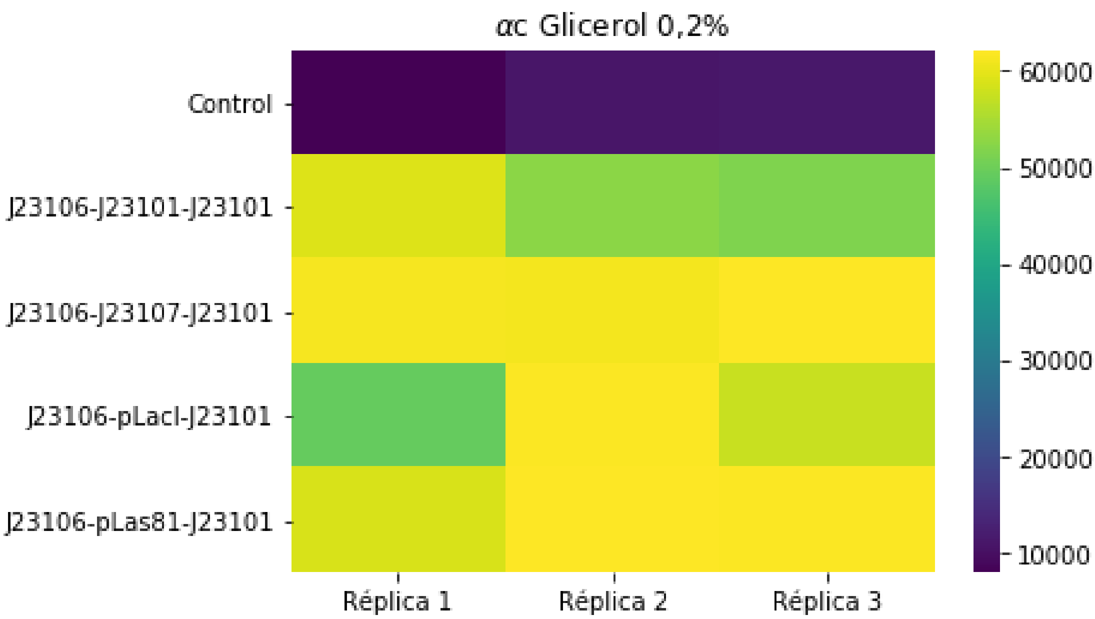



```
In [65]: cglu=[[slopec1,slopec2,slopec3],[slopec7,slopec8,slopec9],[slopec13,slopec14,slopec15],
         cgli=[[slopec4,slopec5,slopec6],[slopec10,slopec11,slopec12],[slopec16,slopec17,slopec1
```

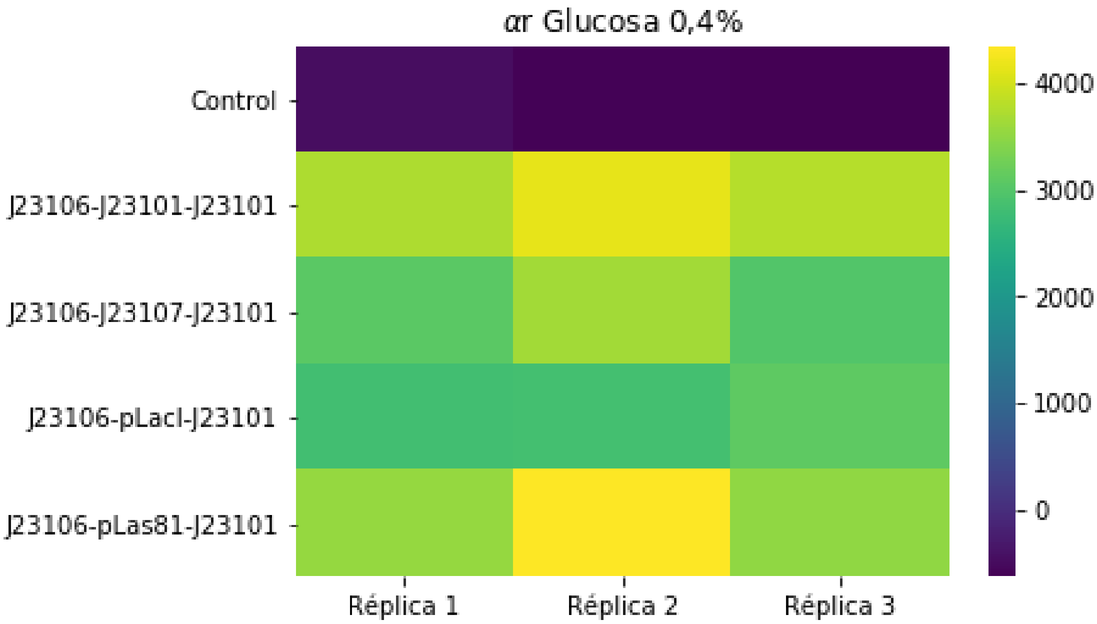
```
rglu=[[sloper1,sloper2,sloper3],[sloper7,sloper8,sloper9],[sloper13,sloper14,sloper15],
rgli=[[sloper4,sloper5,sloper6],[sloper10,sloper11,sloper12],[sloper16,sloper17,sloper1
yglu=[[slopey1,slopey2,slopey3],[slopey7,slopey8,slopey9],[slopey13,slopey14,slopey15],
ygli=[[slopey4,slopey5,slopey6],[slopey10,slopey11,slopey12],[slopey16,slopey17,slopey1
```

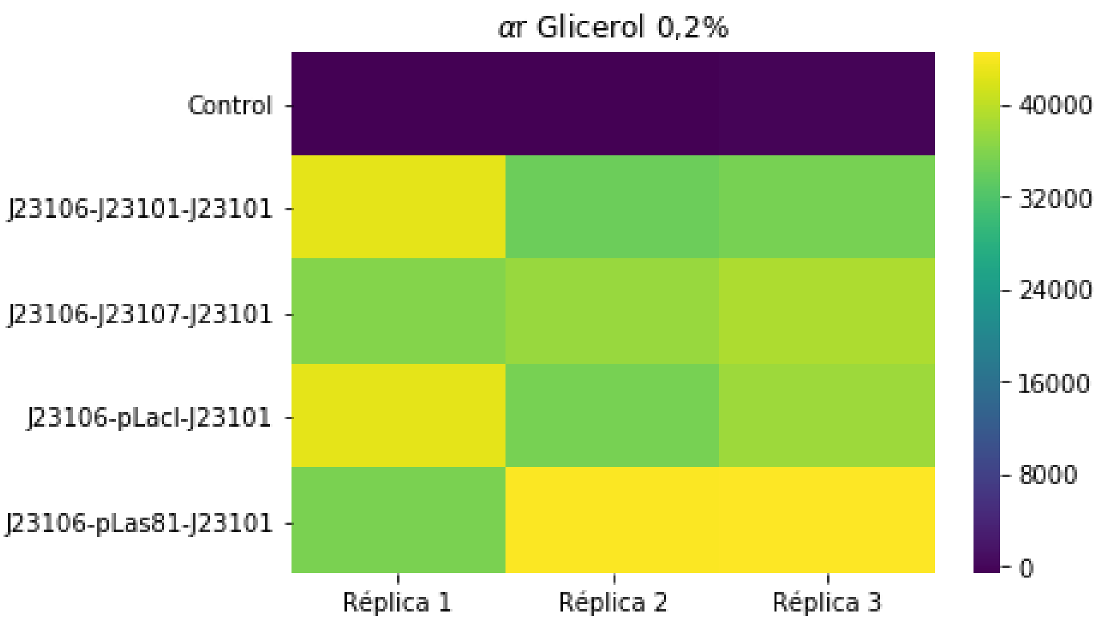
```
In [66]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','

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

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

Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20b7e60b8>
```






```
In [67]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','


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

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

Out[67]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20a1e0278>
```

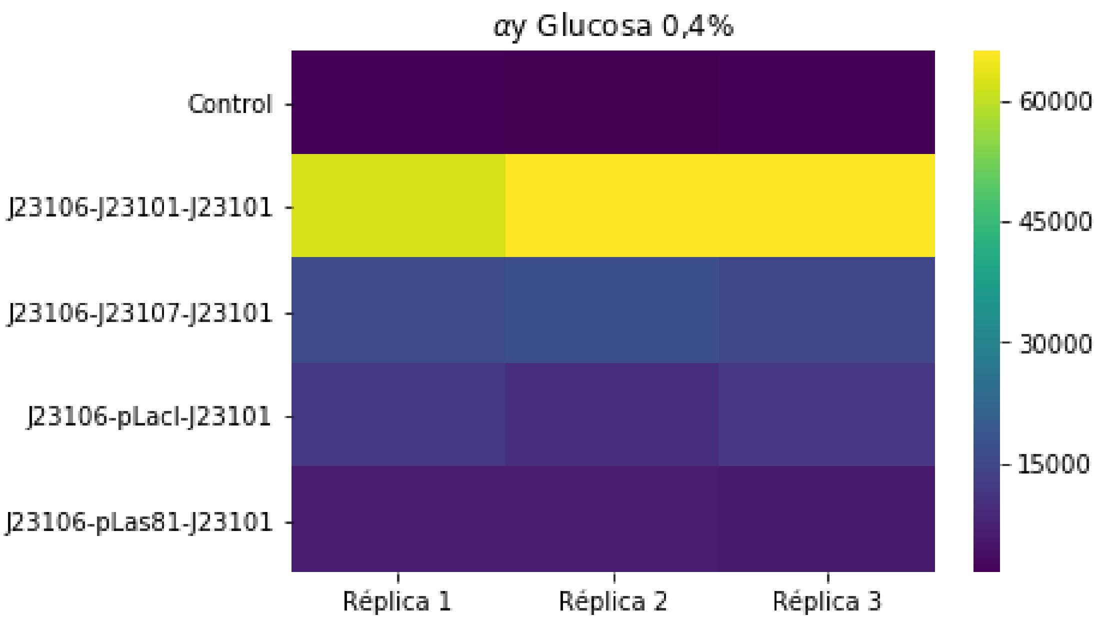


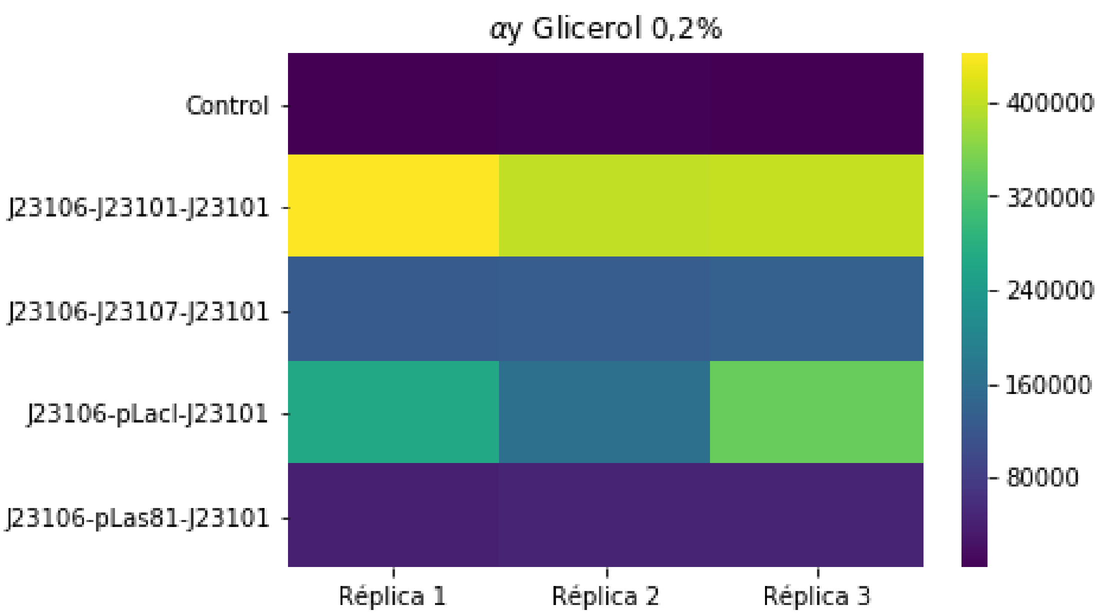



```
In [68]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','

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

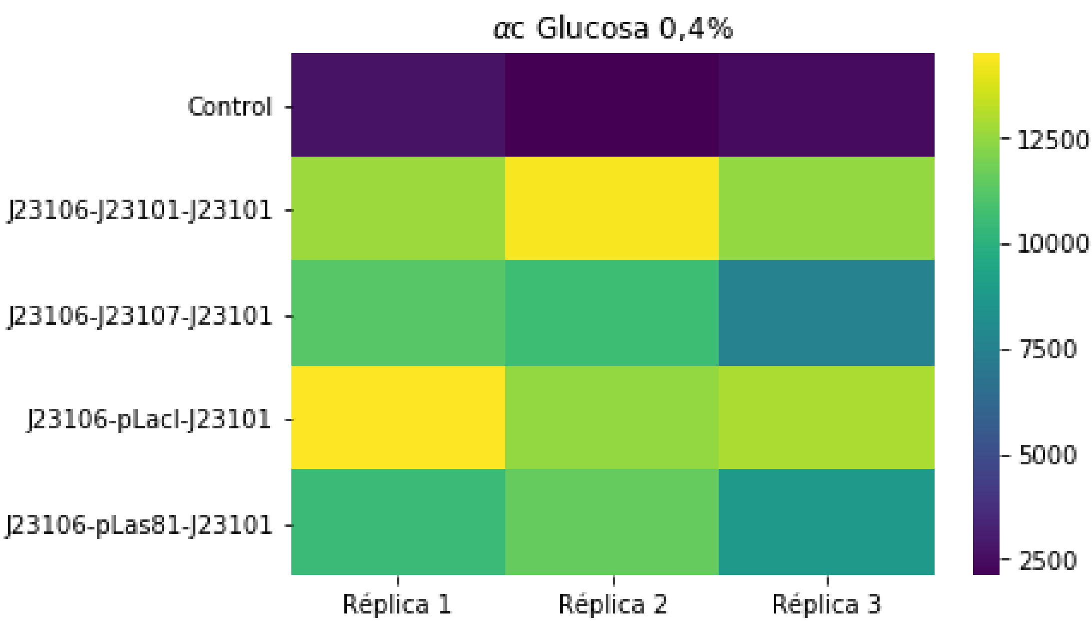
```
sns.heatmap(yglu,cmap='viridis',xticklabels=xlabel,yticklabels=ylabel)

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

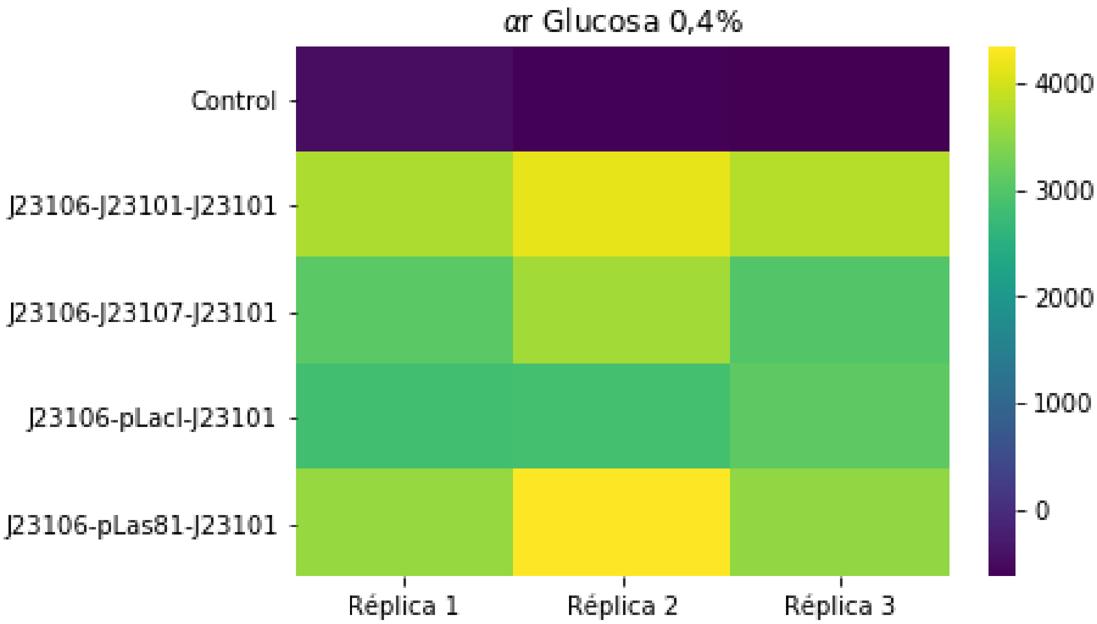
```
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x1b209e21198>
```

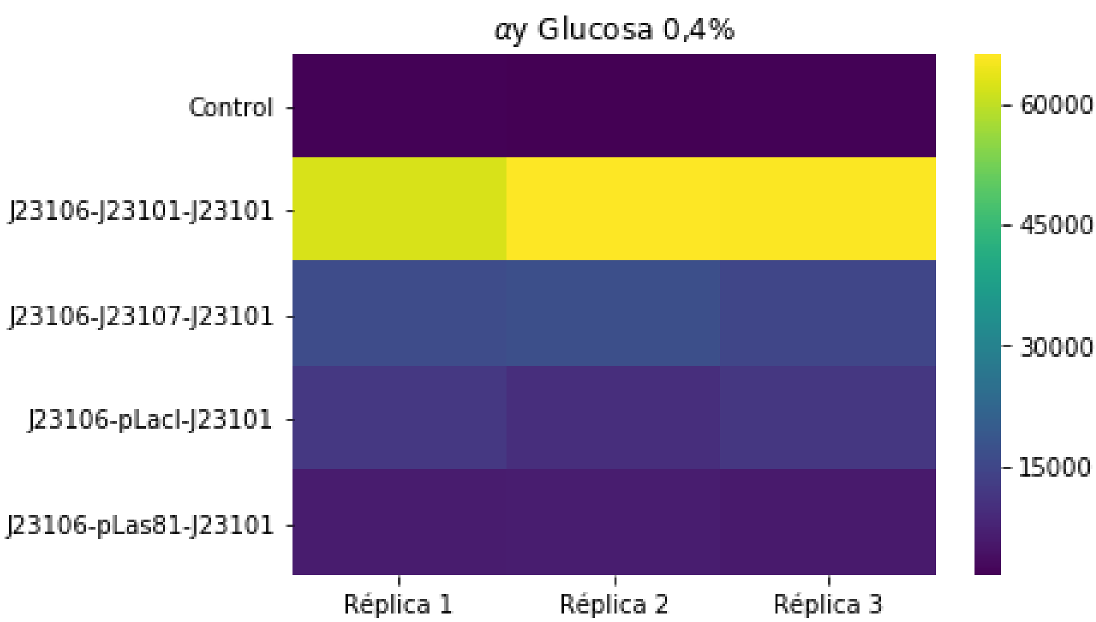




```
In [69]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','

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

Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x1b209d41828>
```



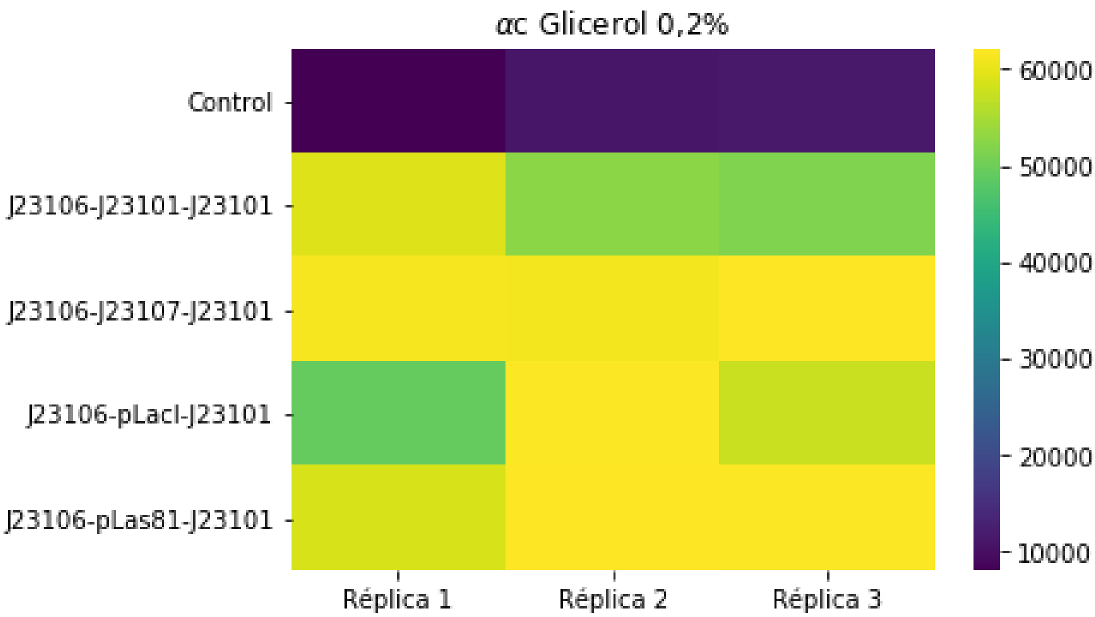


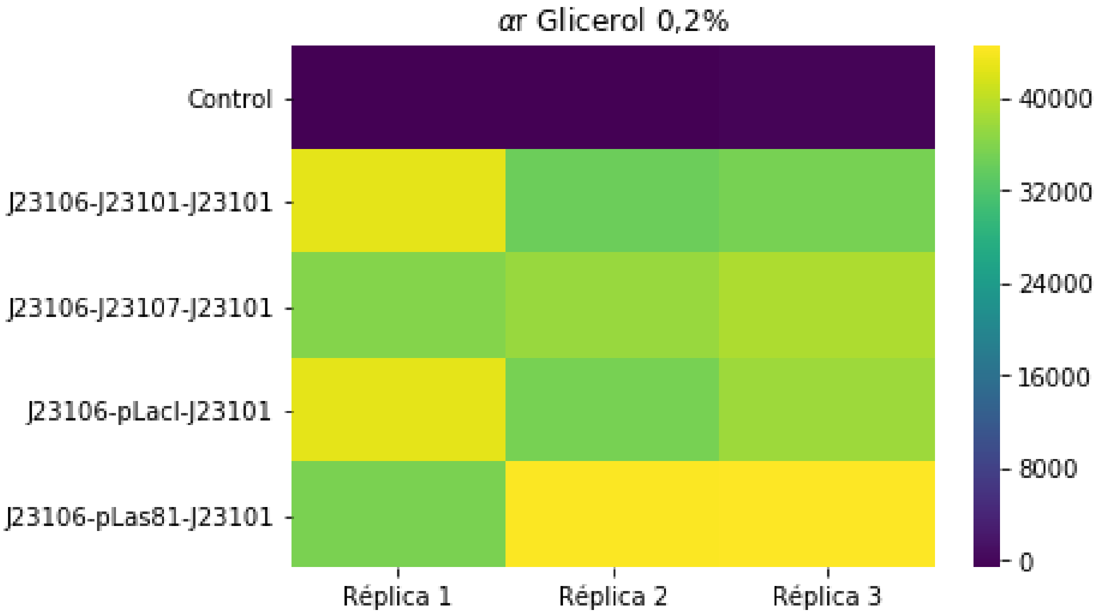


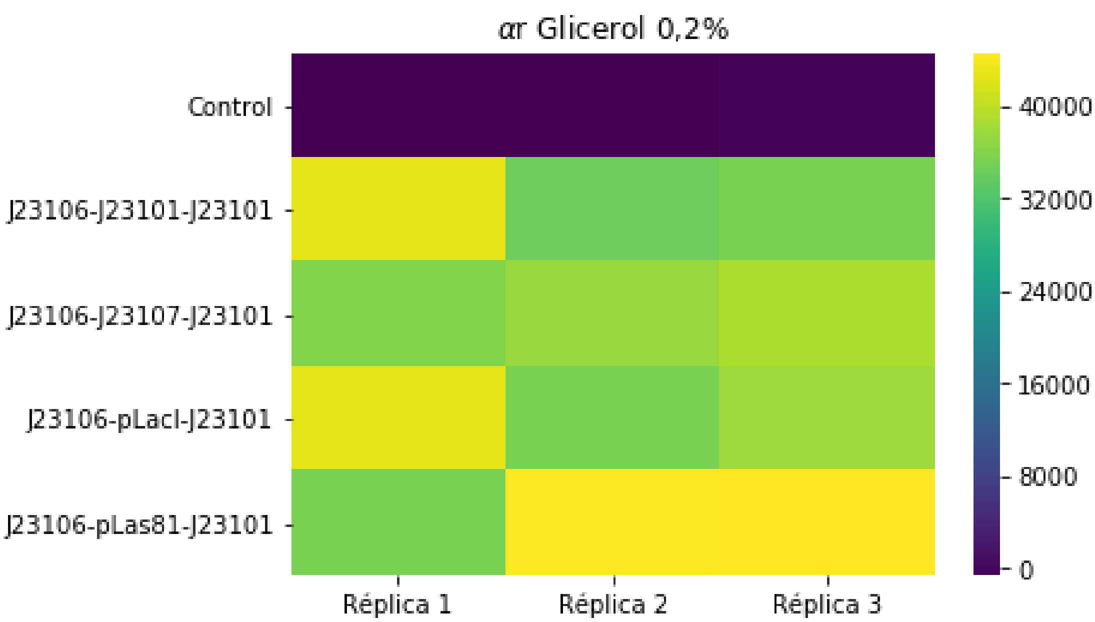

```
In [70]: xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','


         plt.figure()
```

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

```
Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20bc23748>
```



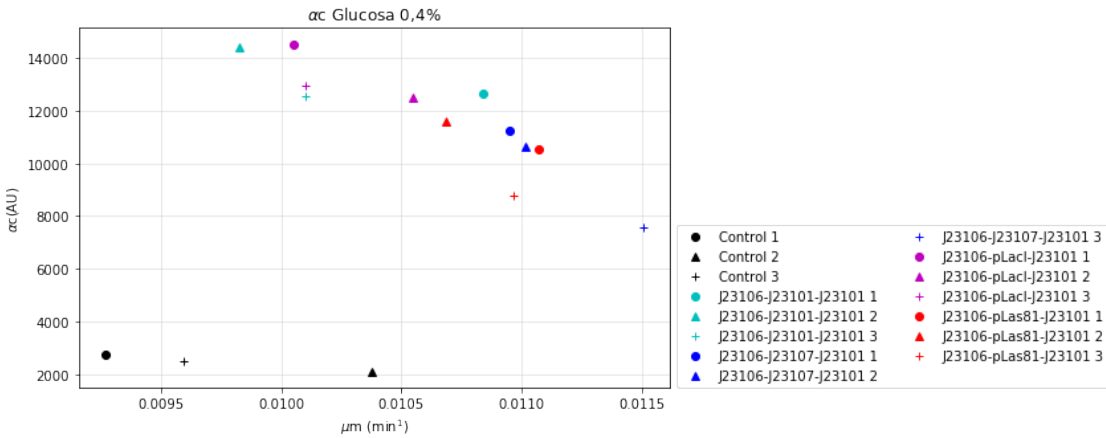


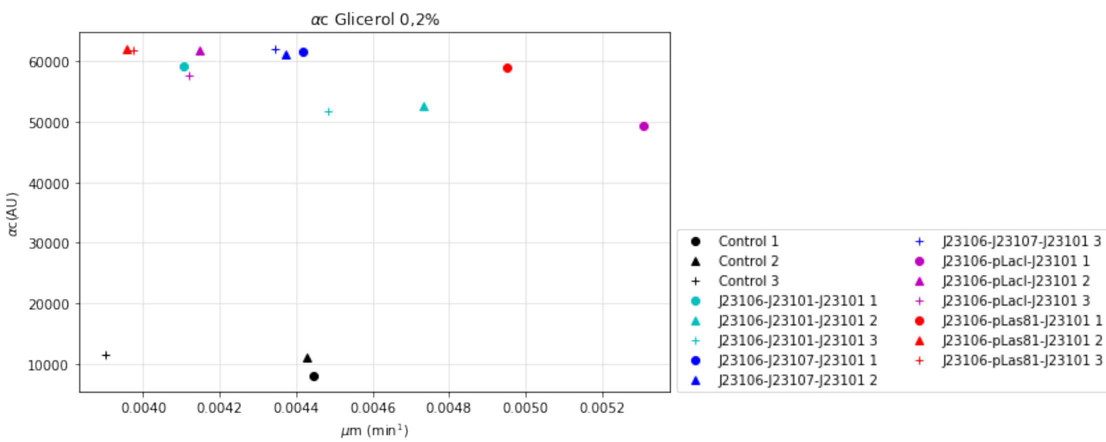



```
In [71]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$c Glucosa 0,4%')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$c(AU)')
```

```
plt.plot(um1,slopec1,'ko',label='Control 1')
plt.plot(um2,slopec2,'k^',label='Control 2')
plt.plot(um3,slopec3,'k+',label='Control 3')
plt.plot(um7,slopec7,'co',label='J23106-J23101-J23101 1')
plt.plot(um8,slopec8,'c^',label='J23106-J23101-J23101 2')
plt.plot(um9,slopec9,'c+',label='J23106-J23101-J23101 3')
plt.plot(um13,slopec13,'bo',label='J23106-J23107-J23101 1')
plt.plot(um14,slopec14,'b^',label='J23106-J23107-J23101 2')
plt.plot(um15,slopec15,'b+',label='J23106-J23107-J23101 3')
plt.plot(um19,slopec19,'mo',label='J23106-pLacI-J23101 1')
plt.plot(um20,slopec20,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um21,slopec21,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um25,slopec25,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um26,slopec26,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um27,slopec27,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um4,slopec4,'ko',label='Control 1')
plt.plot(um5,slopec5,'k^',label='Control 2')
plt.plot(um6,slopec6,'k+',label='Control 3')
plt.plot(um10,slopec10,'co',label='J23106-J23101-J23101 1')
plt.plot(um11,slopec11,'c^',label='J23106-J23101-J23101 2')
plt.plot(um12,slopec12,'c+',label='J23106-J23101-J23101 3')
plt.plot(um16,slopec16,'bo',label='J23106-J23107-J23101 1')
plt.plot(um17,slopec17,'b^',label='J23106-J23107-J23101 2')
plt.plot(um18,slopec18,'b+',label='J23106-J23107-J23101 3')
plt.plot(um22,slopec22,'mo',label='J23106-pLacI-J23101 1')
plt.plot(um23,slopec23,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um24,slopec24,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um28,slopec28,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um29,slopec29,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um30,slopec30,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[71]: <matplotlib.legend.Legend at 0x1b20c596c18>
```

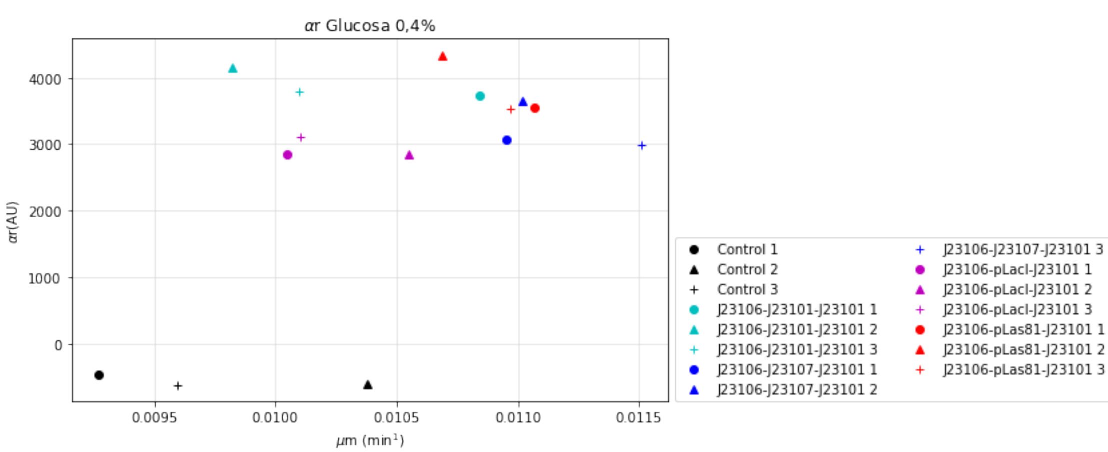


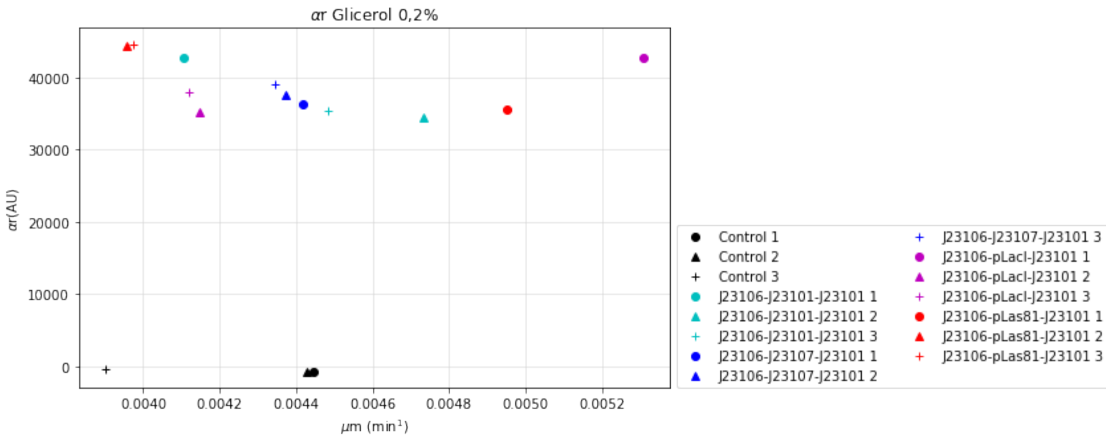


In [72]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um1,sloper1,'ko',label='Control 1')
plt.plot(um2,sloper2,'k^',label='Control 2')
plt.plot(um3,sloper3,'k+',label='Control 3')
plt.plot(um7,sloper7,'co',label='J23106-J23101-J23101 1')
plt.plot(um8,sloper8,'c^',label='J23106-J23101-J23101 2')
plt.plot(um9,sloper9,'c+',label='J23106-J23101-J23101 3')
plt.plot(um13,sloper13,'bo',label='J23106-J23107-J23101 1')
plt.plot(um14,sloper14,'b^',label='J23106-J23107-J23101 2')
plt.plot(um15,sloper15,'b+',label='J23106-J23107-J23101 3')
plt.plot(um19,sloper19,'mo',label='J23106-pLacI-J23101 1')
```

```python
plt.plot(um20,sloper20,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um21,sloper21,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um25,sloper25,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um26,sloper26,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um27,sloper27,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$r Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(um4,sloper4,'ko',label='Control 1')
plt.plot(um5,sloper5,'k^',label='Control 2')
plt.plot(um6,sloper6,'k+',label='Control 3')
plt.plot(um10,sloper10,'co',label='J23106-J23101-J23101 1')
plt.plot(um11,sloper11,'c^',label='J23106-J23101-J23101 2')
plt.plot(um12,sloper12,'c+',label='J23106-J23101-J23101 3')
plt.plot(um16,sloper16,'bo',label='J23106-J23107-J23101 1')
plt.plot(um17,sloper17,'b^',label='J23106-J23107-J23101 2')
plt.plot(um18,sloper18,'b+',label='J23106-J23107-J23101 3')
plt.plot(um22,sloper22,'mo',label='J23106-pLacI-J23101 1')
plt.plot(um23,sloper23,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um24,sloper24,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um28,sloper28,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um29,sloper29,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um30,sloper30,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

```
Out[72]: <matplotlib.legend.Legend at 0x1b20c3f64e0>
```

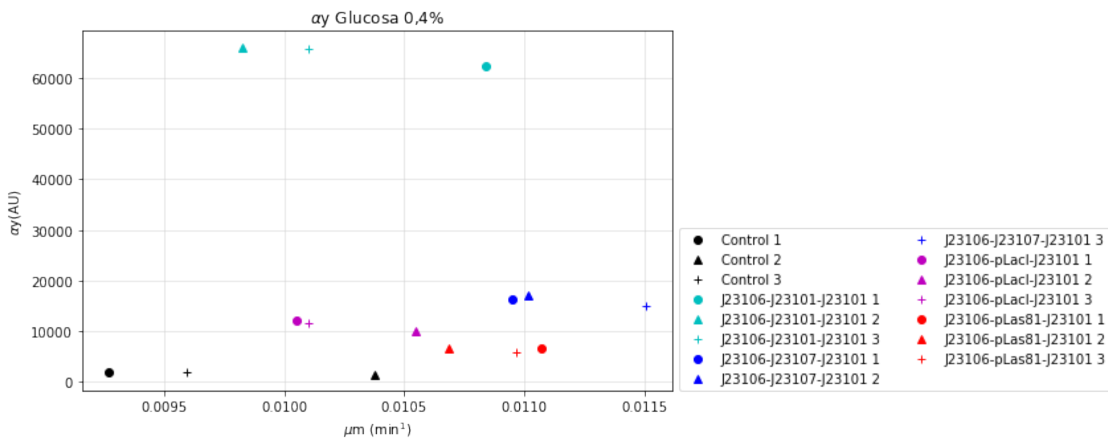


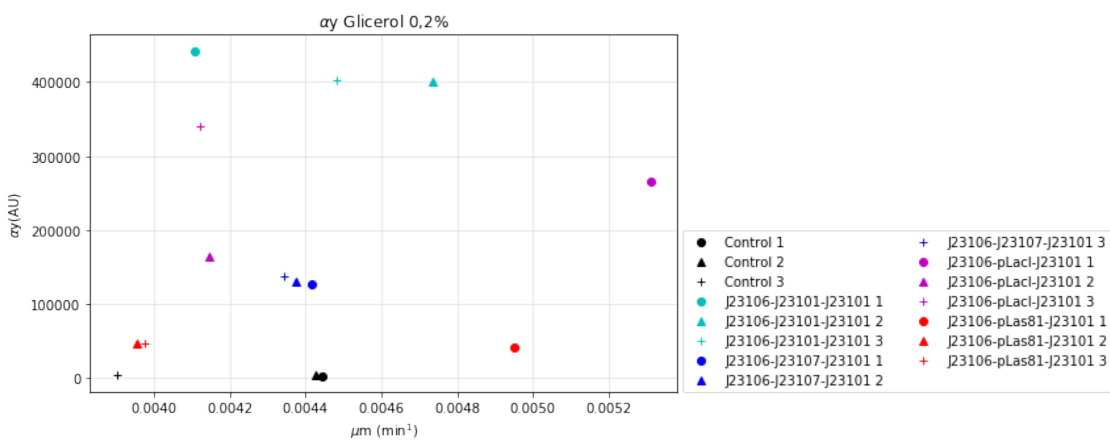


In [73]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'ko',label='Control 1')
plt.plot(um2,slopey2,'k^',label='Control 2')
plt.plot(um3,slopey3,'k+',label='Control 3')
plt.plot(um7,slopey7,'co',label='J23106-J23101-J23101 1')
plt.plot(um8,slopey8,'c^',label='J23106-J23101-J23101 2')
plt.plot(um9,slopey9,'c+',label='J23106-J23101-J23101 3')
plt.plot(um13,slopey13,'bo',label='J23106-J23107-J23101 1')
plt.plot(um14,slopey14,'b^',label='J23106-J23107-J23101 2')
plt.plot(um15,slopey15,'b+',label='J23106-J23107-J23101 3')
plt.plot(um19,slopey19,'mo',label='J23106-pLacI-J23101 1')
plt.plot(um20,slopey20,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um21,slopey21,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um25,slopey25,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um26,slopey26,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um27,slopey27,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glicerol 0,2%')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um4,slopey4,'ko',label='Control 1')
plt.plot(um5,slopey5,'k^',label='Control 2')
plt.plot(um6,slopey6,'k+',label='Control 3')
plt.plot(um10,slopey10,'co',label='J23106-J23101-J23101 1')
```

```
plt.plot(um11,slopey11,'c^',label='J23106-J23101-J23101 2')
plt.plot(um12,slopey12,'c+',label='J23106-J23101-J23101 3')
plt.plot(um16,slopey16,'bo',label='J23106-J23107-J23101 1')
plt.plot(um17,slopey17,'b^',label='J23106-J23107-J23101 2')
plt.plot(um18,slopey18,'b+',label='J23106-J23107-J23101 3')
plt.plot(um22,slopey22,'mo',label='J23106-pLacI-J23101 1')
plt.plot(um23,slopey23,'m^',label='J23106-pLacI-J23101 2')
plt.plot(um24,slopey24,'m+',label='J23106-pLacI-J23101 3')
plt.plot(um28,slopey28,'ro',label='J23106-pLas81-J23101 1')
plt.plot(um29,slopey29,'r^',label='J23106-pLas81-J23101 2')
plt.plot(um30,slopey30,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[73]: <matplotlib.legend.Legend at 0x1b20a3a0d30>

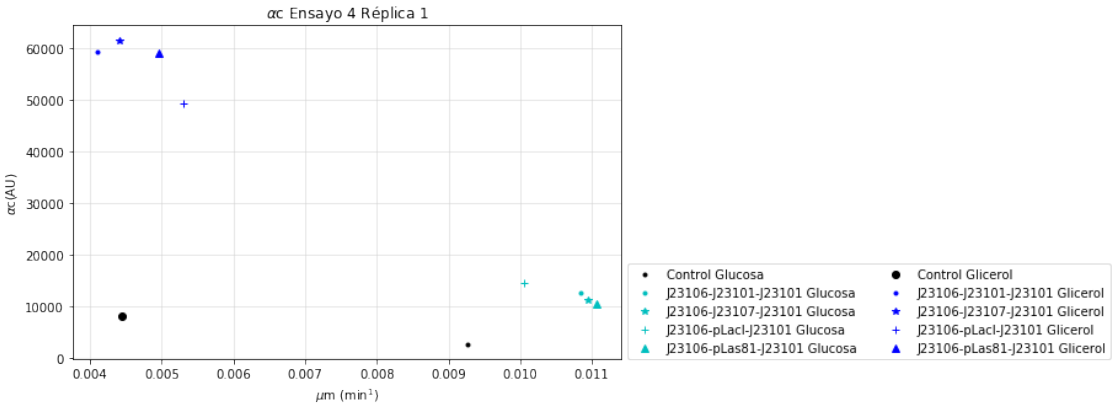


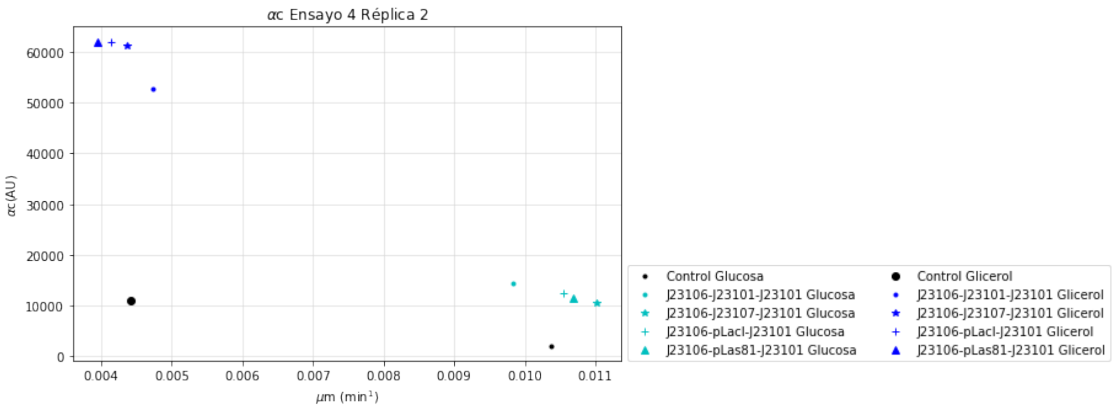


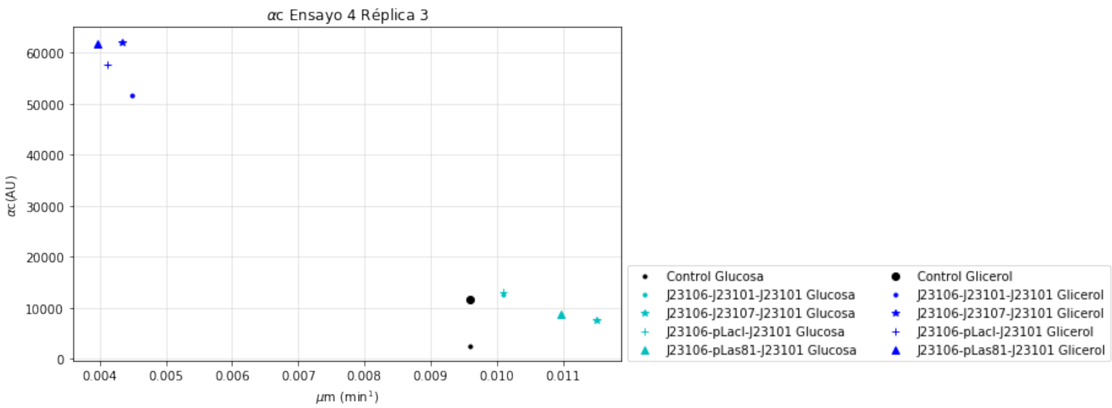
In [74]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 4 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um1,slopec1,'k.',label='Control Glucosa')
plt.plot(um7,slopec7,'c.',label='J23106-J23101-J23101 Glucosa')
plt.plot(um13,slopec13,'c*',label='J23106-J23107-J23101 Glucosa')
plt.plot(um19,slopec19,'c+',label='J23106-pLacI-J23101 Glucosa')
plt.plot(um25,slopec25,'c^',label='J23106-pLas81-J23101 Glucosa')
plt.plot(um4,slopec4,'ko',label='Control Glicerol')
plt.plot(um10,slopec10,'b.',label='J23106-J23101-J23101 Glicerol')
plt.plot(um16,slopec16,'b*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um22,slopec22,'b+',label='J23106-pLacI-J23101 Glicerol')
plt.plot(um28,slopec28,'b^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 4 Réplica 2')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um2,slopec2,'k.',label='Control Glucosa')
plt.plot(um8,slopec8,'c.',label='J23106-J23101-J23101 Glucosa')
plt.plot(um14,slopec14,'c*',label='J23106-J23107-J23101 Glucosa')
plt.plot(um20,slopec20,'c+',label='J23106-pLacI-J23101 Glucosa')
plt.plot(um26,slopec26,'c^',label='J23106-pLas81-J23101 Glucosa')
plt.plot(um5,slopec5,'ko',label='Control Glicerol')
plt.plot(um11,slopec11,'b.',label='J23106-J23101-J23101 Glicerol')
plt.plot(um17,slopec17,'b*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um23,slopec23,'b+',label='J23106-pLacI-J23101 Glicerol')
plt.plot(um29,slopec29,'b^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$c Ensayo 4 Réplica 3')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$c(AU)')
plt.plot(um3,slopec3,'k.',label='Control Glucosa')
plt.plot(um9,slopec9,'c.',label='J23106-J23101-J23101 Glucosa')
plt.plot(um15,slopec15,'c*',label='J23106-J23107-J23101 Glucosa')
plt.plot(um21,slopec21,'c+',label='J23106-pLacI-J23101 Glucosa')
plt.plot(um27,slopec27,'c^',label='J23106-pLas81-J23101 Glucosa')
plt.plot(um3,slopec6,'ko',label='Control Glicerol')
plt.plot(um12,slopec12,'b.',label='J23106-J23101-J23101 Glicerol')
plt.plot(um18,slopec18,'b*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um24,slopec24,'b+',label='J23106-pLacI-J23101 Glicerol')
```

```
plt.plot(um30,slopec30,'b^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[74]: <matplotlib.legend.Legend at 0x1b20bae8fd0>



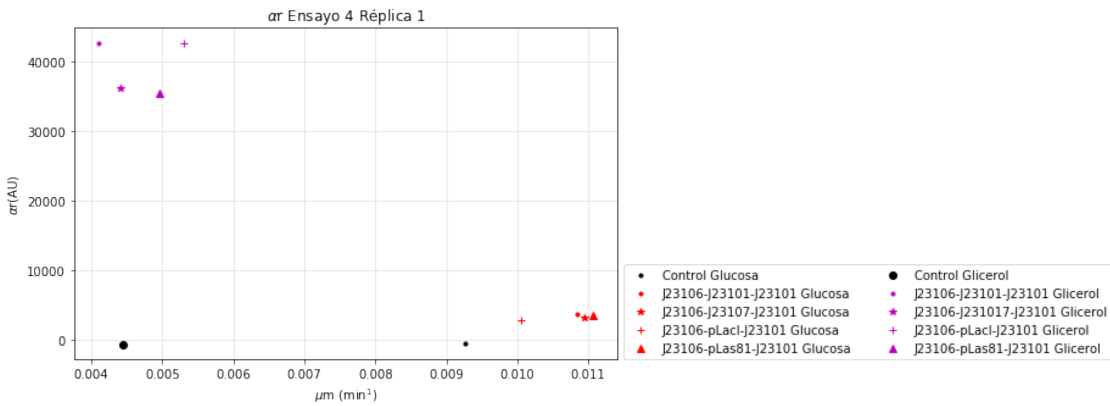


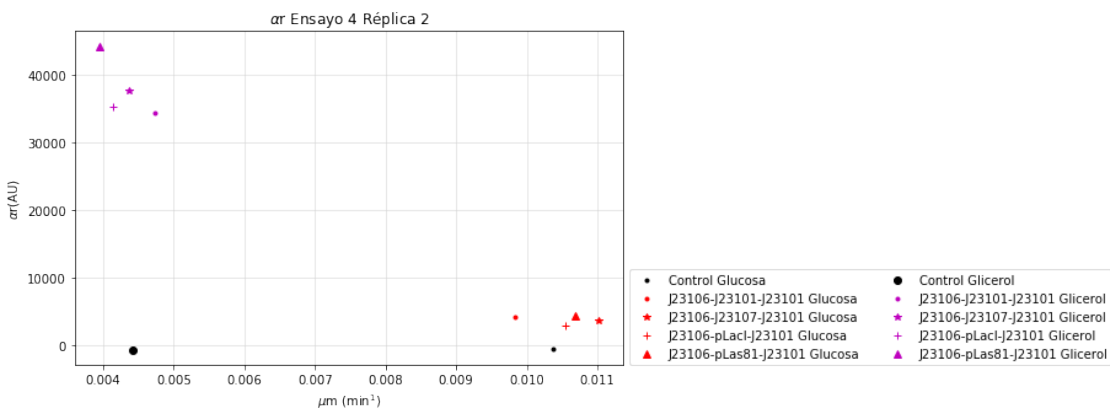


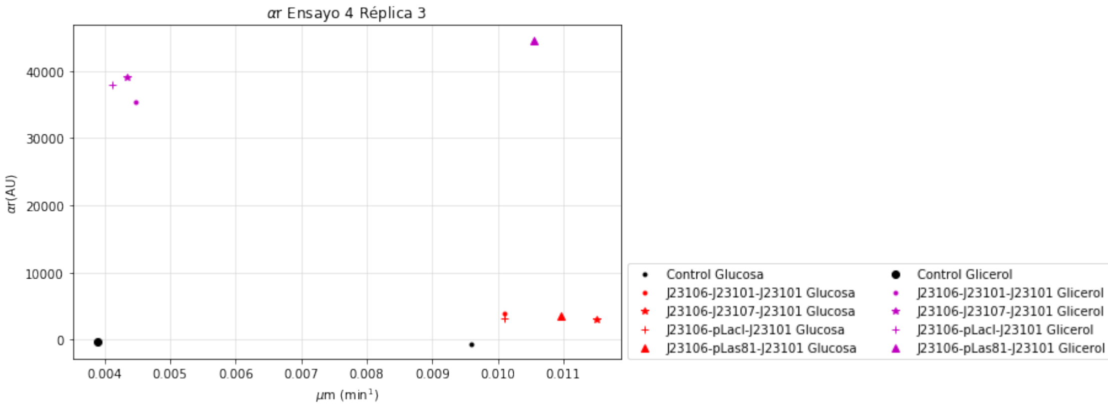
```
In [75]: #grafico de ac versus Um
         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 4 Réplica 1')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um1,sloper1,'k.',label='Control Glucosa ')
         plt.plot(um7,sloper7,'r.',label='J23106-J23101-J23101 Glucosa ')
         plt.plot(um13,sloper13,'r*',label='J23106-J23107-J23101 Glucosa ')
         plt.plot(um19,sloper19,'r+',label='J23106-pLacI-J23101 Glucosa ')
         plt.plot(um25,sloper25,'r^',label='J23106-pLas81-J23101 Glucosa ')
         plt.plot(um4,sloper4,'ko',label='Control Glicerol ')
         plt.plot(um10,sloper10,'m.',label='J23106-J23101-J23101 Glicerol ')
         plt.plot(um16,sloper16,'m*',label='J23106-J231017-J23101 Glicerol')
         plt.plot(um22,sloper22,'m+',label='J23106-pLacI-J23101 Glicerol ')
         plt.plot(um28,sloper28,'m^',label='J23106-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 4 Réplica 2')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um2,sloper2,'k.',label='Control Glucosa')
         plt.plot(um8,sloper8,'r.',label='J23106-J23101-J23101 Glucosa')
         plt.plot(um14,sloper14,'r*',label='J23106-J23107-J23101 Glucosa')
         plt.plot(um20,sloper20,'r+',label='J23106-pLacI-J23101 Glucosa')
         plt.plot(um26,sloper26,'r^',label='J23106-pLas81-J23101 Glucosa')
         plt.plot(um5,sloper5,'ko',label='Control Glicerol')
         plt.plot(um11,sloper11,'m.',label='J23106-J23101-J23101 Glicerol')
         plt.plot(um17,sloper17,'m*',label='J23106-J23107-J23101 Glicerol')
         plt.plot(um23,sloper23,'m+',label='J23106-pLacI-J23101 Glicerol')
         plt.plot(um29,sloper29,'m^',label='J23106-pLas81-J23101 Glicerol')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc=(1.01,0.0),ncol=2)

         plt.figure(figsize=(8,5))
         plt.title(r'$\alpha$r Ensayo 4 Réplica 3')
         plt.xlabel(r'$\mu$m (min$^1$)')
         plt.ylabel(r'$\alpha$r(AU)')
         plt.plot(um3,sloper3,'k.',label='Control Glucosa')
         plt.plot(um9,sloper9,'r.',label='J23106-J23101-J23101 Glucosa')
         plt.plot(um15,sloper15,'r*',label='J23106-J23107-J23101 Glucosa')
         plt.plot(um21,sloper21,'r+',label='J23106-pLacI-J23101 Glucosa')
         plt.plot(um27,sloper27,'r^',label='J23106-pLas81-J23101 Glucosa')
         plt.plot(um6,sloper6,'ko',label='Control Glicerol')
```

```
plt.plot(um12,sloper12,'m.',label='J23106-J23101-J23101 Glicerol')
plt.plot(um18,sloper18,'m*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um24,sloper24,'m+',label='J23106-pLacI-J23101 Glicerol')
plt.plot(um20,sloper30,'m^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[75]: <matplotlib.legend.Legend at 0x1b20e306eb8>



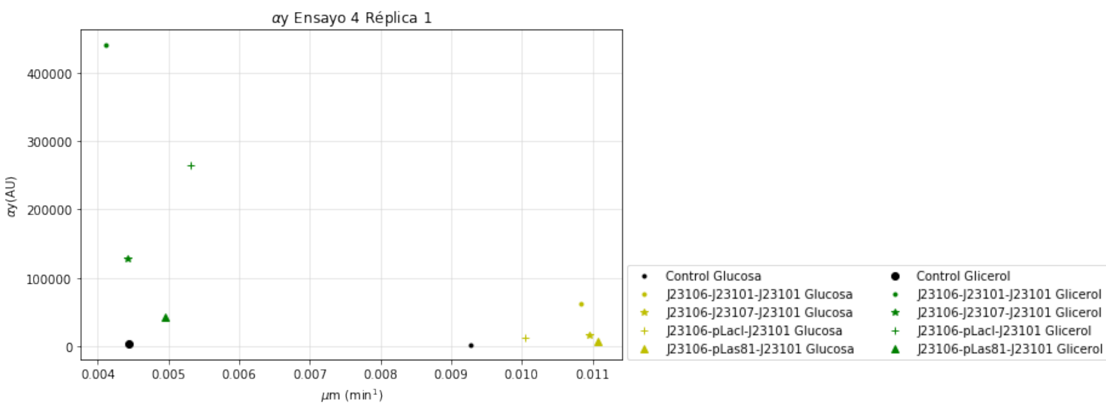


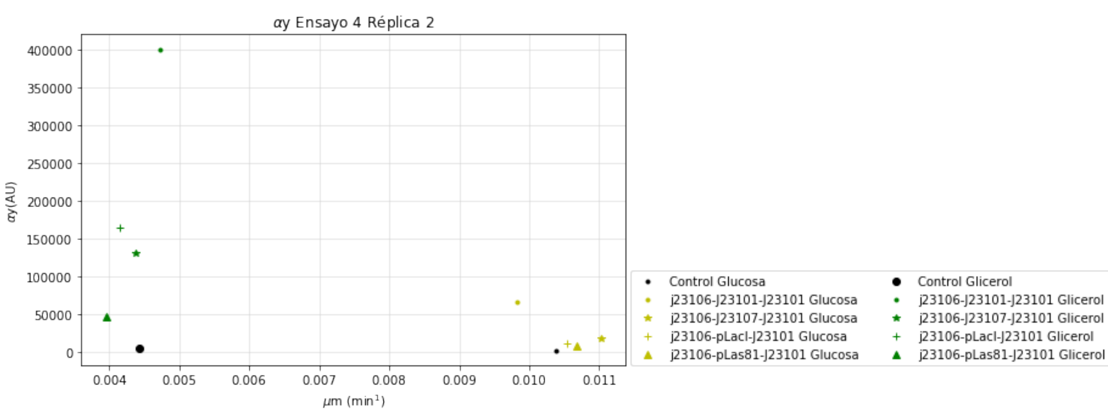


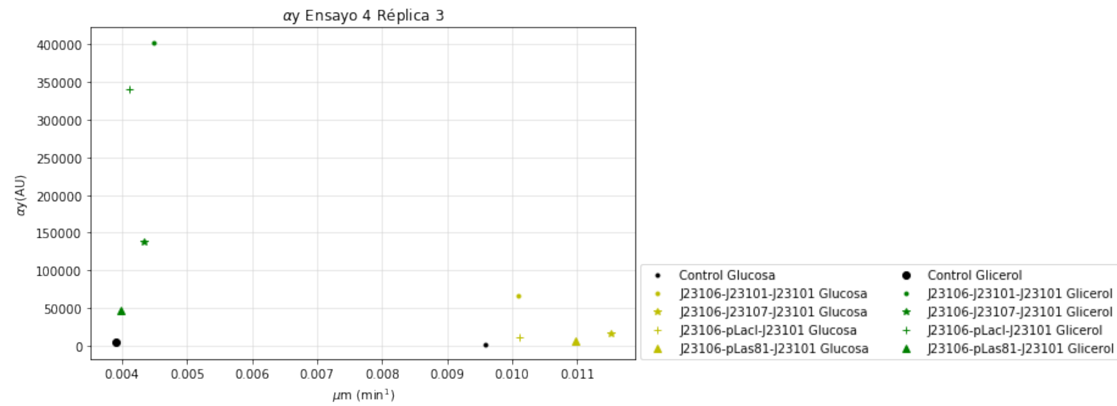
In [76]:
```
#grafico de ac versus Um
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 4 Réplica 1')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um1,slopey1,'k.',label='Control Glucosa ')
plt.plot(um7,slopey7,'y.',label='J23106-J23101-J23101 Glucosa ')
plt.plot(um13,slopey13,'y*',label='J23106-J23107-J23101 Glucosa ')
plt.plot(um19,slopey19,'y+',label='J23106-pLacI-J23101 Glucosa ')
plt.plot(um25,slopey25,'y^',label='J23106-pLas81-J23101 Glucosa ')
plt.plot(um4,slopey4,'ko',label='Control Glicerol ')
plt.plot(um10,slopey10,'g.',label='J23106-J23101-J23101 Glicerol ')
plt.plot(um16,slopey16,'g*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um22,slopey22,'g+',label='J23106-pLacI-J23101 Glicerol ')
plt.plot(um28,slopey28,'g^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 4 Réplica 2')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um2,slopey2,'k.',label='Control Glucosa')
plt.plot(um8,slopey8,'y.',label='j23106-J23101-J23101 Glucosa')
plt.plot(um14,slopey14,'y*',label='j23106-J23107-J23101 Glucosa')
plt.plot(um20,slopey20,'y+',label='j23106-pLacI-J23101 Glucosa')
plt.plot(um26,slopey26,'y^',label='j23106-pLas81-J23101 Glucosa')
plt.plot(um5,slopey5,'ko',label='Control Glicerol')
plt.plot(um11,slopey11,'g.',label='j23106-J23101-J23101 Glicerol')
plt.plot(um17,slopey17,'g*',label='j23106-J23107-J23101 Glicerol')
plt.plot(um23,slopey23,'g+',label='j23106-pLacI-J23101 Glicerol')
plt.plot(um29,slopey29,'g^',label='j23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
```

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

plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Ensayo 4 Réplica 3')
plt.xlabel(r'$\mu$m (min$^1$)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(um3,slopey3,'k.',label='Control Glucosa')
plt.plot(um9,slopey9,'y.',label='J23106-J23101-J23101 Glucosa')
plt.plot(um15,slopey15,'y*',label='J23106-J23107-J23101 Glucosa')
plt.plot(um21,slopey21,'y+',label='J23106-pLacI-J23101 Glucosa')
plt.plot(um27,slopey27,'y^',label='J23106-pLas81-J23101 Glucosa')
plt.plot(um6,slopey6,'ko',label='Control Glicerol')
plt.plot(um12,slopey12,'g.',label='J23106-J23101-J23101 Glicerol')
plt.plot(um18,slopey18,'g*',label='J23106-J23107-J23101 Glicerol')
plt.plot(um24,slopey24,'g+',label='J23106-pLacI-J23101 Glicerol')
plt.plot(um30,slopey30,'g^',label='J23106-pLas81-J23101 Glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[76]: <matplotlib.legend.Legend at 0x1b20e57a588>



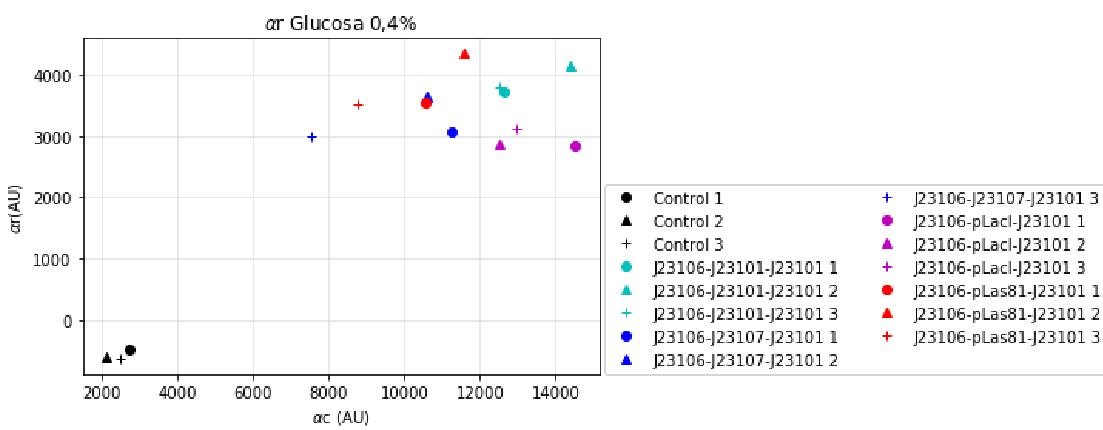


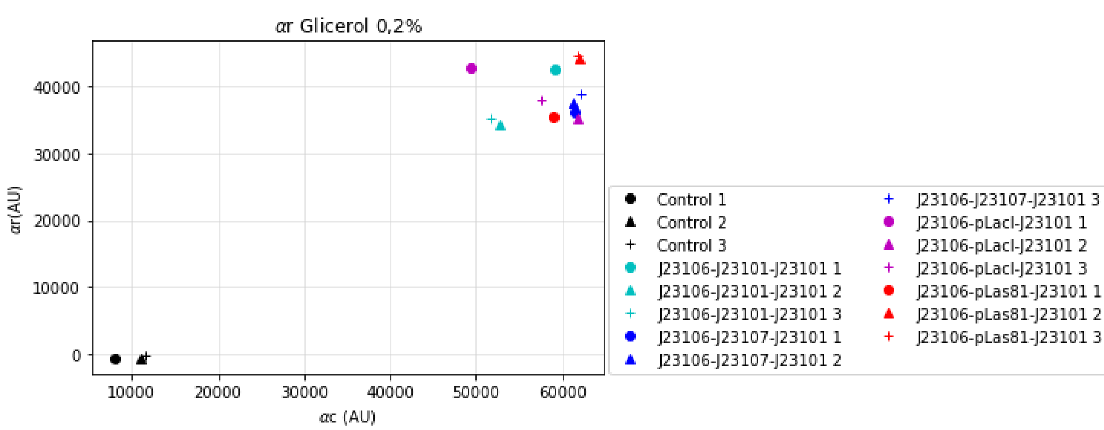


In [77]:
```
#grafico de ac versus ac
plt.figure()
plt.title(r'$\alpha$r Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(slopec1,sloper1,'ko',label='Control 1')
plt.plot(slopec2,sloper2,'k^',label='Control 2')
plt.plot(slopec3,sloper3,'k+',label='Control 3')
plt.plot(slopec7,sloper7,'co',label='J23106-J23101-J23101 1')
plt.plot(slopec8,sloper8,'c^',label='J23106-J23101-J23101 2')
plt.plot(slopec9,sloper9,'c+',label='J23106-J23101-J23101 3')
plt.plot(slopec13,sloper13,'bo',label='J23106-J23107-J23101 1')
plt.plot(slopec14,sloper14,'b^',label='J23106-J23107-J23101 2')
plt.plot(slopec15,sloper15,'b+',label='J23106-J23107-J23101 3')
plt.plot(slopec19,sloper19,'mo',label='J23106-pLacI-J23101 1')
plt.plot(slopec20,sloper20,'m^',label='J23106-pLacI-J23101 2')
plt.plot(slopec21,sloper21,'m+',label='J23106-pLacI-J23101 3')
plt.plot(slopec25,sloper25,'ro',label='J23106-pLas81-J23101 1')
plt.plot(slopec26,sloper26,'r^',label='J23106-pLas81-J23101 2')
plt.plot(slopec27,sloper27,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ac versus ac
plt.figure()
plt.title(r'$\alpha$r Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$r(AU)')
plt.plot(slopec4,sloper4,'ko',label='Control 1')
plt.plot(slopec5,sloper5,'k^',label='Control 2')
```

```
plt.plot(slopec6,sloper6,'k+',label='Control 3')
plt.plot(slopec10,sloper10,'co',label='J23106-J23101-J23101 1')
plt.plot(slopec11,sloper11,'c^',label='J23106-J23101-J23101 2')
plt.plot(slopec12,sloper12,'c+',label='J23106-J23101-J23101 3')
plt.plot(slopec16,sloper16,'bo',label='J23106-J23107-J23101 1')
plt.plot(slopec17,sloper17,'b^',label='J23106-J23107-J23101 2')
plt.plot(slopec18,sloper18,'b+',label='J23106-J23107-J23101 3')
plt.plot(slopec22,sloper22,'mo',label='J23106-pLacI-J23101 1')
plt.plot(slopec23,sloper23,'m^',label='J23106-pLacI-J23101 2')
plt.plot(slopec24,sloper24,'m+',label='J23106-pLacI-J23101 3')
plt.plot(slopec28,sloper28,'ro',label='J23106-pLas81-J23101 1')
plt.plot(slopec29,sloper29,'r^',label='J23106-pLas81-J23101 2')
plt.plot(slopec30,sloper30,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

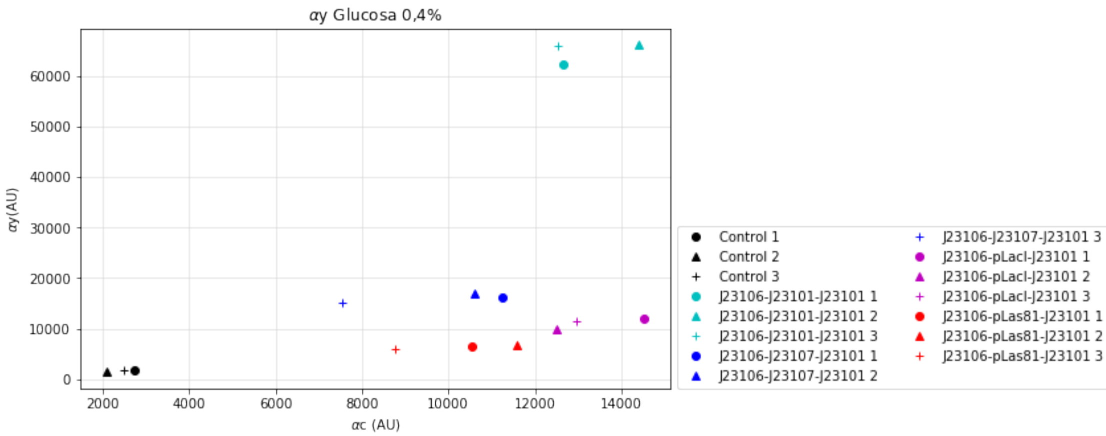
```
Out[77]: <matplotlib.legend.Legend at 0x1b20a20ada0>
```

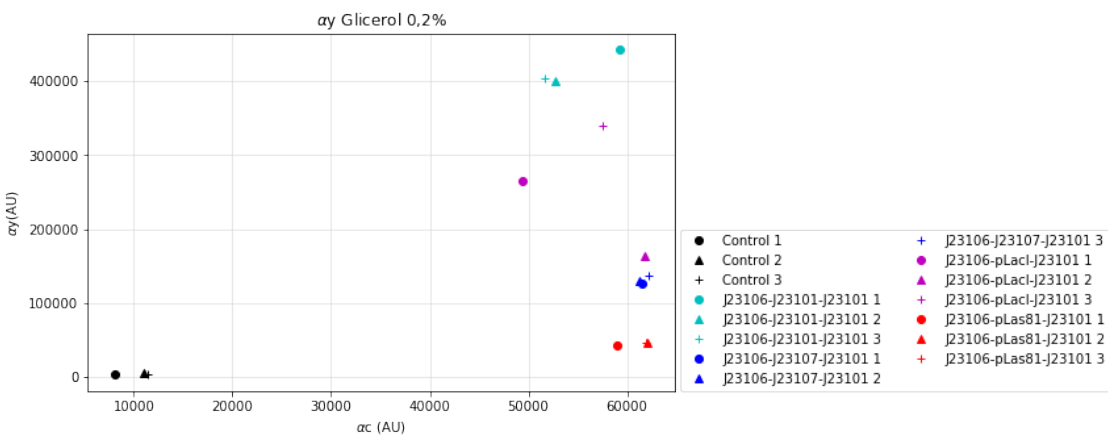




In [78]:
```
#grafico de ar vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glucosa 0,4%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec1,slopey1,'ko',label='Control 1')
plt.plot(slopec2,slopey2,'k^',label='Control 2')
plt.plot(slopec3,slopey3,'k+',label='Control 3')
plt.plot(slopec7,slopey7,'co',label='J23106-J23101-J23101 1')
plt.plot(slopec8,slopey8,'c^',label='J23106-J23101-J23101 2')
plt.plot(slopec9,slopey9,'c+',label='J23106-J23101-J23101 3')
plt.plot(slopec13,slopey13,'bo',label='J23106-J23107-J23101 1')
plt.plot(slopec14,slopey14,'b^',label='J23106-J23107-J23101 2')
plt.plot(slopec15,slopey15,'b+',label='J23106-J23107-J23101 3')
plt.plot(slopec19,slopey19,'mo',label='J23106-pLacI-J23101 1')
plt.plot(slopec20,slopey20,'m^',label='J23106-pLacI-J23101 2')
plt.plot(slopec21,slopey21,'m+',label='J23106-pLacI-J23101 3')
plt.plot(slopec25,slopey25,'ro',label='J23106-pLas81-J23101 1')
plt.plot(slopec26,slopey26,'r^',label='J23106-pLas81-J23101 2')
plt.plot(slopec27,slopey27,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)

#grafico de ay vs ac
plt.figure(figsize=(8,5))
plt.title(r'$\alpha$y Glicerol 0,2%')
plt.xlabel(r'$\alpha$c (AU)')
plt.ylabel(r'$\alpha$y(AU)')
plt.plot(slopec4,slopey4,'ko',label='Control 1')
plt.plot(slopec5,slopey5,'k^',label='Control 2')
plt.plot(slopec6,slopey6,'k+',label='Control 3')
plt.plot(slopec10,slopey10,'co',label='J23106-J23101-J23101 1')
plt.plot(slopec11,slopey11,'c^',label='J23106-J23101-J23101 2')
plt.plot(slopec12,slopey12,'c+',label='J23106-J23101-J23101 3')
plt.plot(slopec16,slopey16,'bo',label='J23106-J23107-J23101 1')
plt.plot(slopec17,slopey17,'b^',label='J23106-J23107-J23101 2')
plt.plot(slopec18,slopey18,'b+',label='J23106-J23107-J23101 3')
plt.plot(slopec22,slopey22,'mo',label='J23106-pLacI-J23101 1')
plt.plot(slopec23,slopey23,'m^',label='J23106-pLacI-J23101 2')
plt.plot(slopec24,slopey24,'m+',label='J23106-pLacI-J23101 3')
plt.plot(slopec28,slopey28,'ro',label='J23106-pLas81-J23101 1')
plt.plot(slopec29,slopey29,'r^',label='J23106-pLas81-J23101 2')
plt.plot(slopec30,slopey30,'r+',label='J23106-pLas81-J23101 3')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[78]: <matplotlib.legend.Legend at 0x1b20bb8b9e8>

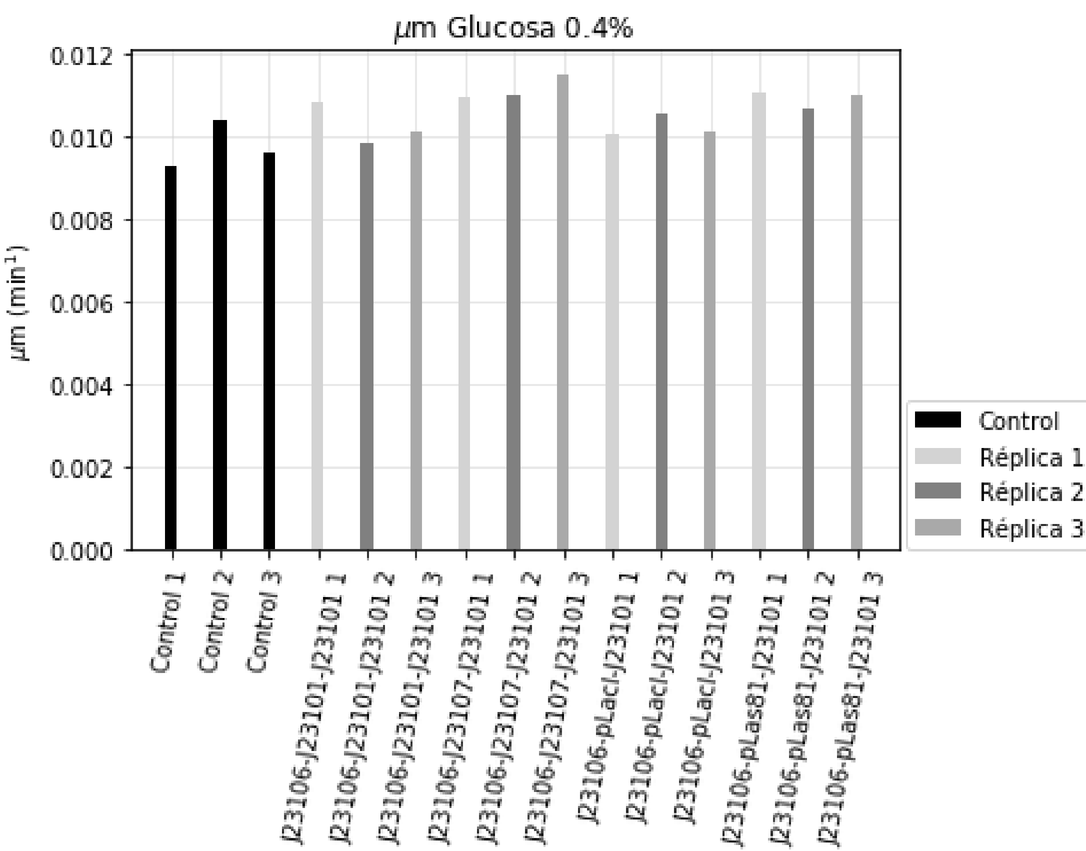




In [79]:
```
#Grafico de barras um de FPs
uglu=[um1,um2,um3,um7,um8,um9,um13,um14,um15,um19,um20,um21,um25,um26,um27]
ugli=[um4,um5,um6,um10,um11,um12,um16,um17,um18,um22,um23,um24,um28,um29,um30]

X = np.arange(15)
plt.figure()
plt.title(r'$\mu$m Glucosa 0.4%')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.bar(X[0]+0.00,uglu[0],color='k',width=0.25,label='Control',zorder=3)
plt.bar(X[1]+0.00,uglu[1],color='k',width=0.25,zorder=3)
plt.bar(X[2]+0.00,uglu[2],color='k',width=0.25,zorder=3)
plt.bar(X[3]+0.00,uglu[3],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[4]+0.00,uglu[4],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[5]+0.00,uglu[5],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[6]+0.00,uglu[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,uglu[7],color='grey',width=0.25,zorder=3)
```

```
plt.bar(X[8]+0.00,uglu[8],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,uglu[9],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,uglu[10],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,uglu[11],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,uglu[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,uglu[13],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,uglu[14],color='darkgrey',width=0.25,zorder=3)
plt.xticks(X,['Control 1','Control 2','Control 3','J23106-J23101-J23101 1','J23106-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

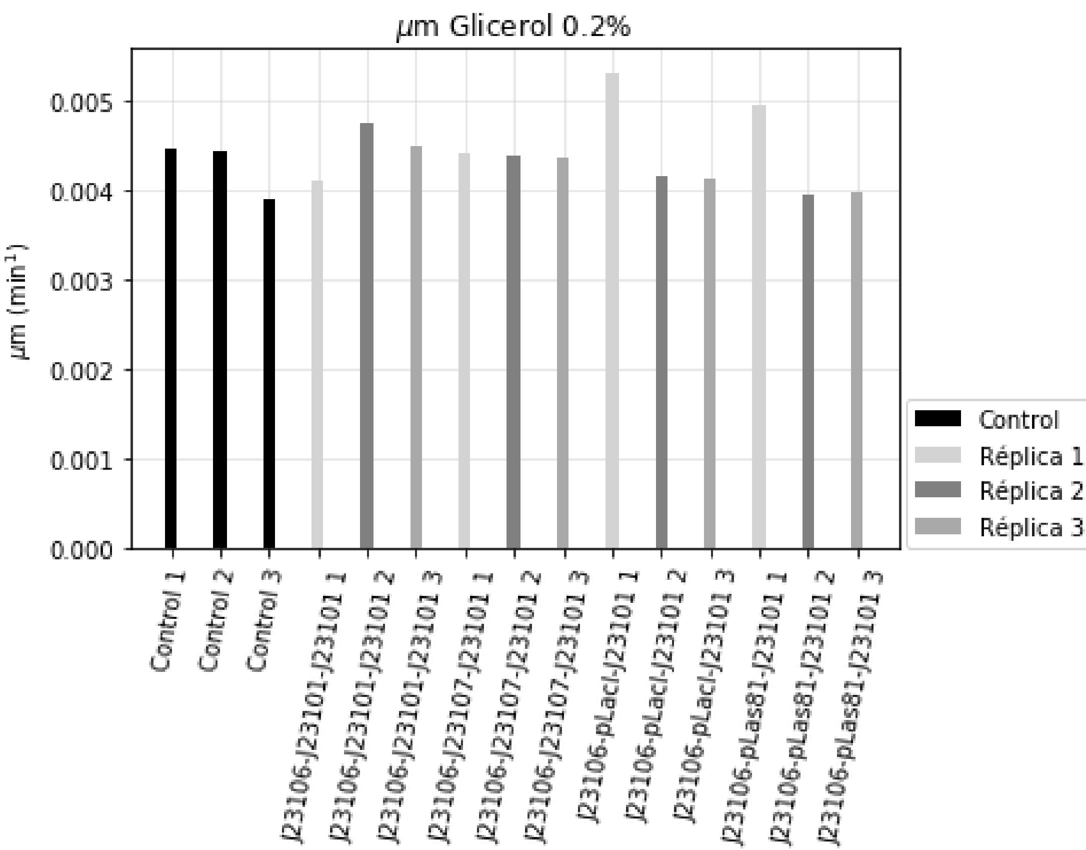
```
Out[79]: <matplotlib.legend.Legend at 0x1b209e38ba8>
```




```
In [80]: X = np.arange(15)
plt.figure()
plt.title(r'$\mu$m Glicerol 0.2%')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.bar(X[0]+0.00,ugli[0],color='k',width=0.25,label='Control',zorder=3)
plt.bar(X[1]+0.00,ugli[1],color='k',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ugli[2],color='k',width=0.25,zorder=3)
```

```
plt.bar(X[3]+0.00,ugli[3],color='lightgrey',width=0.25,label='Réplica 1',zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='grey',width=0.25,label='Réplica 2',zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='darkgrey',width=0.25,label='Réplica 3',zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='grey',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='darkgrey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='darkgrey',width=0.25,zorder=3)
plt.xticks(X,['Control 1','Control 2','Control 3','J23106-J23101-J23101 1','J23106-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

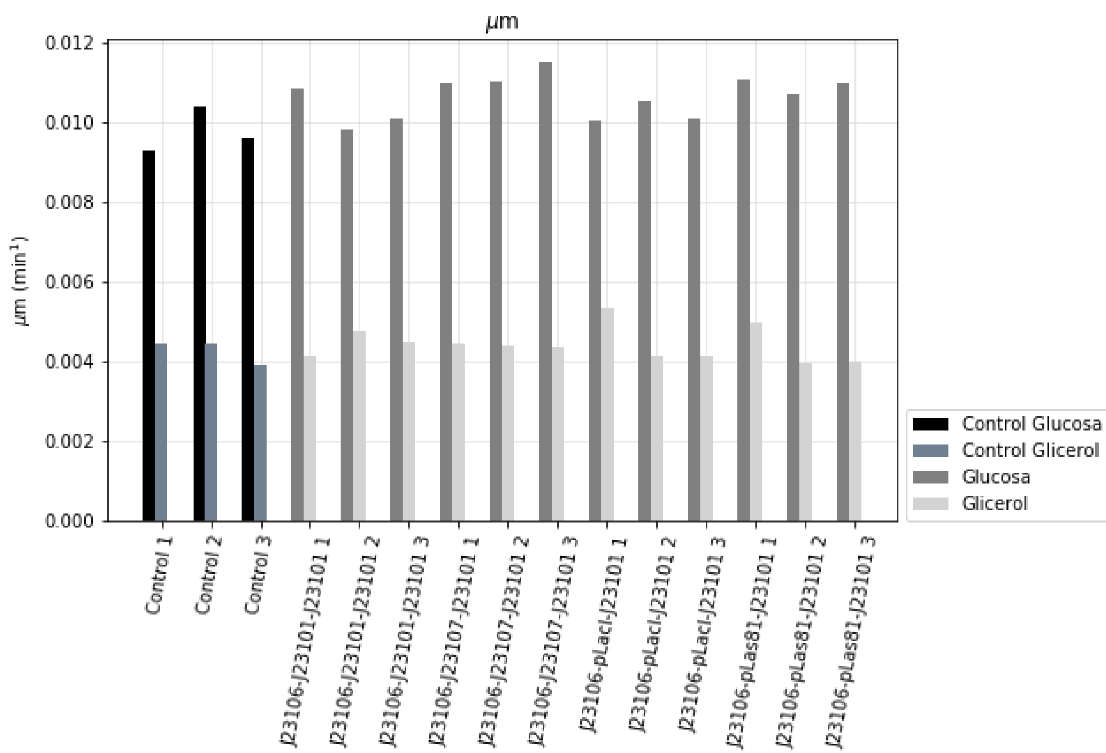
Out[80]: <matplotlib.legend.Legend at 0x1b20ea02a58>




```
In [81]: X = np.arange(15)
         plt.figure(figsize=(8,5))
```

```
plt.title(r'$\mu$m')
plt.ylabel(r'$\mu$m (min$^-1$)')
plt.bar(X[0]-0.25,uglu[0],color='k',width=0.25,label='Control Glucosa',zorder=3)
plt.bar(X[0]+0.00,ugli[0],color='slategrey',width=0.25,label='Control Glicerol',zorder=
plt.bar(X[1]-0.20,uglu[1],color='k',width=0.25,zorder=3)
plt.bar(X[1]+0.00,ugli[1],color='slategrey',width=0.25,zorder=3)
plt.bar(X[2]-0.25,uglu[2],color='k',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ugli[2],color='slategrey',width=0.25,zorder=3)
plt.bar(X[3]-0.25,uglu[3],color='grey',width=0.25,label='Glucosa',zorder=3)
plt.bar(X[3]+0.00,ugli[3],color='lightgrey',width=0.25,label='Glicerol',zorder=3)
plt.bar(X[4]-0.25,uglu[4],color='grey',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ugli[4],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[5]-0.25,uglu[5],color='grey',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ugli[5],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[6]-0.25,uglu[6],color='grey',width=0.25,zorder=3)
plt.bar(X[6]+0.00,ugli[6],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[7]-0.25,uglu[7],color='grey',width=0.25,zorder=3)
plt.bar(X[7]+0.00,ugli[7],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[8]-0.25,uglu[8],color='grey',width=0.25,zorder=3)
plt.bar(X[8]+0.00,ugli[8],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[9]-0.25,uglu[9],color='grey',width=0.25,zorder=3)
plt.bar(X[9]+0.00,ugli[9],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[10]-0.25,uglu[10],color='grey',width=0.25,zorder=3)
plt.bar(X[10]+0.00,ugli[10],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[11]-0.25,uglu[11],color='grey',width=0.25,zorder=3)
plt.bar(X[11]+0.00,ugli[11],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[12]-0.25,uglu[12],color='grey',width=0.25,zorder=3)
plt.bar(X[12]+0.00,ugli[12],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[13]-0.25,uglu[13],color='grey',width=0.25,zorder=3)
plt.bar(X[13]+0.00,ugli[13],color='lightgrey',width=0.25,zorder=3)
plt.bar(X[14]-0.25,uglu[14],color='grey',width=0.25,zorder=3)
plt.bar(X[14]+0.00,ugli[14],color='lightgrey',width=0.25,zorder=3)
plt.xticks(X,['Control 1','Control 2','Control 3','J23106-J23101-J23101 1','J23106-J231
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0))
```

```
Out[81]: <matplotlib.legend.Legend at 0x1b20ebb6128>
```



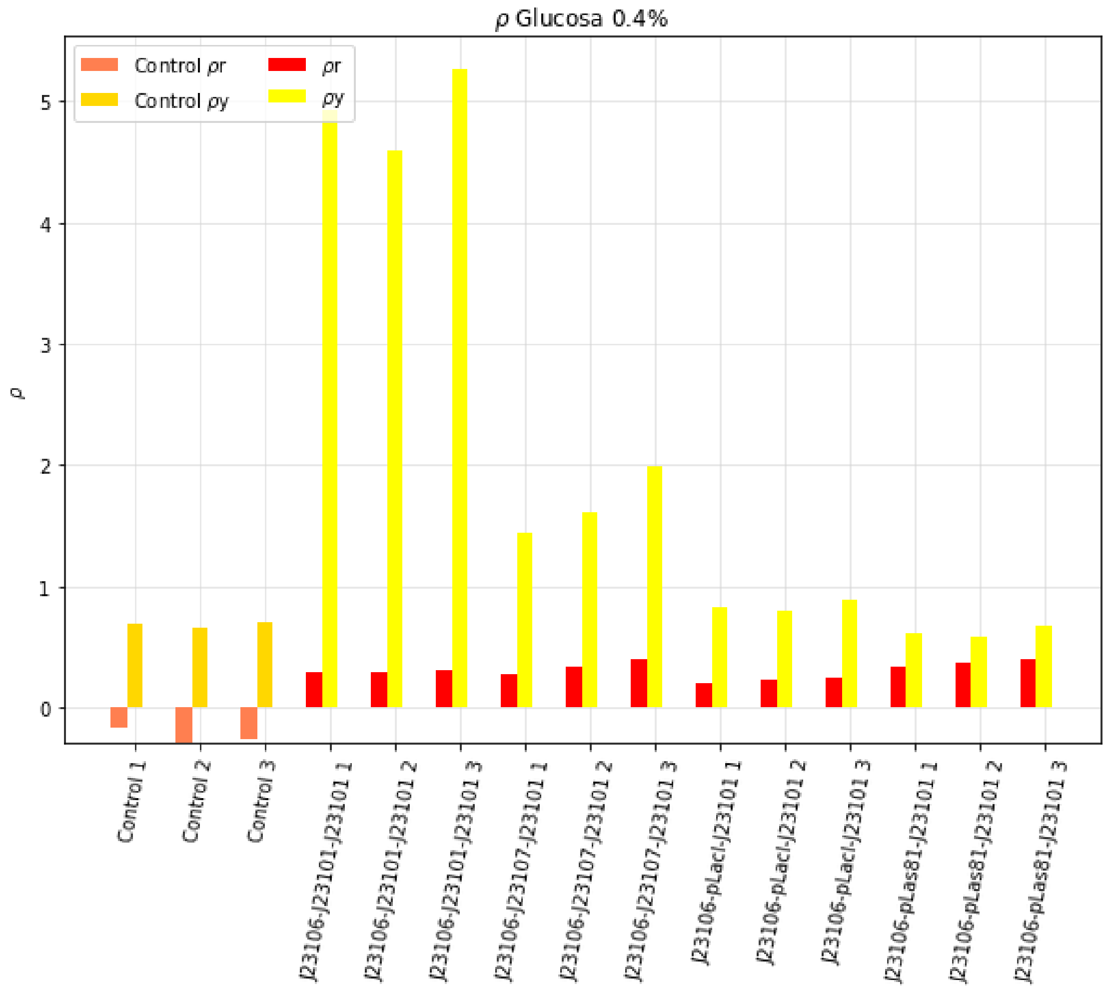
In [82]:
```
#Ro RFP
pr1=sloper1/slopec1
pr2=sloper2/slopec2
pr3=sloper3/slopec3
pr4=sloper4/slopec4
pr5=sloper5/slopec5
pr6=sloper6/slopec6
pr7=sloper7/slopec7
pr8=sloper8/slopec8
pr9=sloper9/slopec9
pr10=sloper10/slopec10
pr11=sloper11/slopec11
pr12=sloper12/slopec12
pr13=sloper13/slopec13
pr14=sloper14/slopec14
pr15=sloper15/slopec15
pr16=sloper16/slopec16
pr17=sloper17/slopec17
pr18=sloper18/slopec18
pr19=sloper19/slopec19
pr20=sloper20/slopec20
pr21=sloper21/slopec21
pr22=sloper22/slopec22
```

```
pr23=sloper23/slopec23
pr24=sloper24/slopec24
pr25=sloper25/slopec25
pr26=sloper26/slopec26
pr27=sloper27/slopec27
pr28=sloper28/slopec28
pr29=sloper29/slopec29
pr30=sloper30/slopec30


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

In [83]:
```
#Ro YFP
py1=slopey1/slopec1
py2=slopey2/slopec2
py3=slopey3/slopec3
py4=slopey4/slopec4
py5=slopey5/slopec5
py6=slopey6/slopec6
py7=slopey7/slopec7
py8=slopey8/slopec8
py9=slopey9/slopec9
py10=slopey10/slopec10
py11=slopey11/slopec11
py12=slopey12/slopec12
py13=slopey13/slopec13
py14=slopey14/slopec14
py15=slopey15/slopec15
py16=slopey16/slopec16
py17=slopey17/slopec17
py18=slopey18/slopec18
py19=slopey19/slopec19
py20=slopey20/slopec20
py21=slopey21/slopec21
py22=slopey22/slopec22
py23=slopey23/slopec23
py24=slopey24/slopec24
py25=slopey25/slopec25
py26=slopey26/slopec26
py27=slopey27/slopec27
py28=slopey28/slopec28
py29=slopey29/slopec29
py30=slopey30/slopec30


ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py1
```

```
ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py2
ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,
```

```
In [84]: #grafico ro de yfp y de rfp
         ro_rfp=[pr1,pr2,pr3,pr4,pr5,pr6,pr7,pr8,pr9,pr10,pr11,pr12,pr13,pr14,pr15,pr16,pr17,pr1
         ro_yfp=[py1,py2,py3,py4,py5,py6,py7,py8,py9,py10,py11,py12,py13,py14,py15,py16,py17,py1


         X = np.arange(15)
         plt.figure(figsize=(10,7))
         plt.title(r'$\rho$ Glucosa 0.4%')
         plt.ylabel(r'$\rho$')
         plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r',zo
         plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y',zor
         plt.bar(X[1]-0.25,ro_rfp[1],color='coral',width=0.25,zorder=3)
         plt.bar(X[1]+0.00,ro_yfp[1],color='gold',width=0.25,zorder=3)
         plt.bar(X[2]-0.25,ro_rfp[2],color='coral',width=0.25,zorder=3)
         plt.bar(X[2]+0.00,ro_yfp[2],color='gold',width=0.25,zorder=3)
         plt.bar(X[3]-0.25,ro_rfp[6],color='r',width=0.25,label=r'$\rho$r',zorder=3)
         plt.bar(X[3]+0.00,ro_yfp[6],color='yellow',width=0.25,label=r'$\rho$y',zorder=3)
         plt.bar(X[4]-0.25,ro_rfp[7],color='r',width=0.25,zorder=3)
         plt.bar(X[4]+0.00,ro_yfp[7],color='yellow',width=0.25,zorder=3)
         plt.bar(X[5]-0.25,ro_rfp[8],color='r',width=0.25,zorder=3)
         plt.bar(X[5]+0.00,ro_yfp[8],color='yellow',width=0.25,zorder=3)
         plt.bar(X[6]-0.25,ro_rfp[12],color='r',width=0.25,zorder=3)
         plt.bar(X[6]+0.00,ro_yfp[12],color='yellow',width=0.25,zorder=3)
         plt.bar(X[7]-0.25,ro_rfp[13],color='r',width=0.25,zorder=3)
         plt.bar(X[7]+0.00,ro_yfp[13],color='yellow',width=0.25,zorder=3)
         plt.bar(X[8]-0.25,ro_rfp[14],color='r',width=0.25,zorder=3)
         plt.bar(X[8]+0.00,ro_yfp[14],color='yellow',width=0.25,zorder=3)
         plt.bar(X[9]-0.25,ro_rfp[18],color='r',width=0.25,zorder=3)
         plt.bar(X[9]+0.00,ro_yfp[18],color='yellow',width=0.25,zorder=3)
         plt.bar(X[10]-0.25,ro_rfp[19],color='r',width=0.25,zorder=3)
         plt.bar(X[10]+0.00,ro_yfp[19],color='yellow',width=0.25,zorder=3)
         plt.bar(X[11]-0.25,ro_rfp[20],color='r',width=0.25,zorder=3)
         plt.bar(X[11]+0.00,ro_yfp[20],color='yellow',width=0.25,zorder=3)
         plt.bar(X[12]-0.25,ro_rfp[24],color='r',width=0.25,zorder=3)
         plt.bar(X[12]+0.00,ro_yfp[24],color='yellow',width=0.25,zorder=3)
         plt.bar(X[13]-0.25,ro_rfp[25],color='r',width=0.25,zorder=3)
         plt.bar(X[13]+0.00,ro_yfp[25],color='yellow',width=0.25,zorder=3)
         plt.bar(X[14]-0.25,ro_rfp[26],color='r',width=0.25,zorder=3)
         plt.bar(X[14]+0.00,ro_yfp[26],color='yellow',width=0.25,zorder=3)
         plt.xticks(X,['Control 1','Control 2','Control 3','J23106-J23101-J23101 1','J23106-J231
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper left',ncol=2)

Out[84]: <matplotlib.legend.Legend at 0x1b20ee824a8>
```

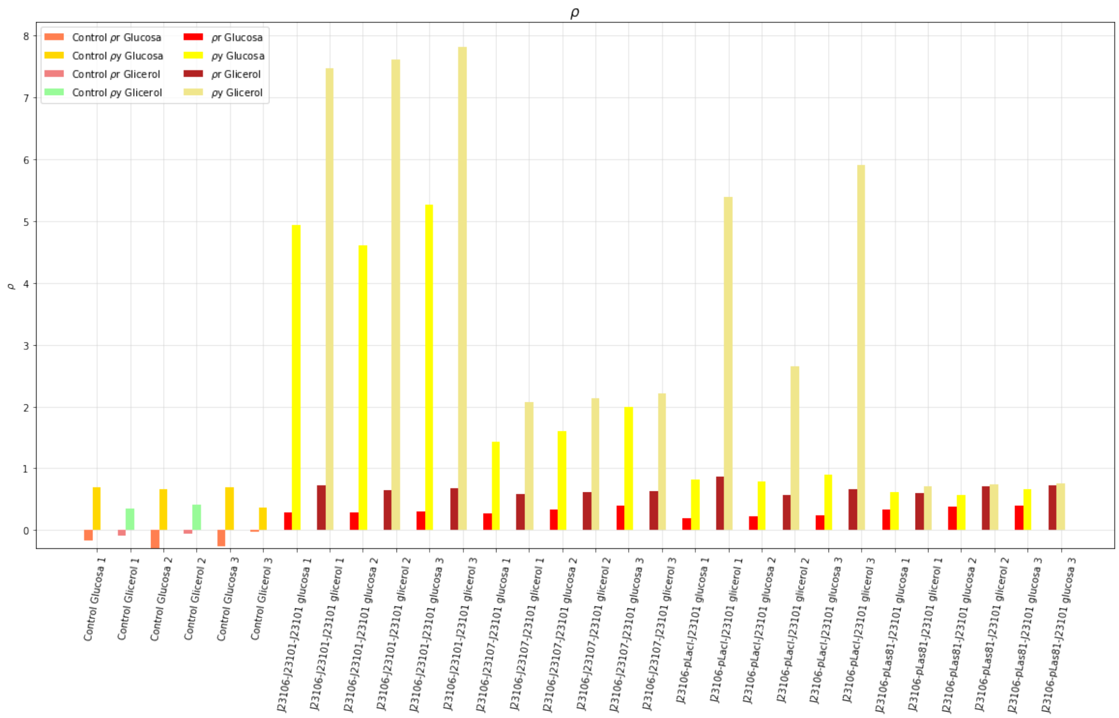


In [85]:
```
X = np.arange(30)
plt.figure(figsize=(20,10))
plt.title(r'$\rho$',fontsize=15.0)
plt.ylabel(r'$\rho$')
plt.bar(X[0]-0.25,ro_rfp[0],color='coral',width=0.25,label= 'Control'+' '+r'$\rho$r Glu
plt.bar(X[0]+0.00,ro_yfp[0],color='gold',width=0.25,label= 'Control'+' '+r'$\rho$y Gluc
plt.bar(X[1]-0.25,ro_rfp[3],color='lightcoral',width=0.25,label= 'Control'+' '+r'$\rho$
plt.bar(X[1]+0.00,ro_yfp[3],color='palegreen',width=0.25,label= 'Control'+' '+r'$\rho$y
plt.bar(X[2]-0.25,ro_rfp[1],color='coral',width=0.25,zorder=3)
plt.bar(X[2]+0.00,ro_yfp[1],color='gold',width=0.25,zorder=3)
plt.bar(X[3]-0.25,ro_rfp[4],color='lightcoral',width=0.25,zorder=3)
plt.bar(X[3]+0.00,ro_yfp[4],color='palegreen',width=0.25,zorder=3)
plt.bar(X[4]-0.25,ro_rfp[2],color='coral',width=0.25,zorder=3)
plt.bar(X[4]+0.00,ro_yfp[2],color='gold',width=0.25,zorder=3)
plt.bar(X[5]-0.25,ro_rfp[5],color='coral',width=0.25,zorder=3)
plt.bar(X[5]+0.00,ro_yfp[5],color='gold',width=0.25,zorder=3)
```

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

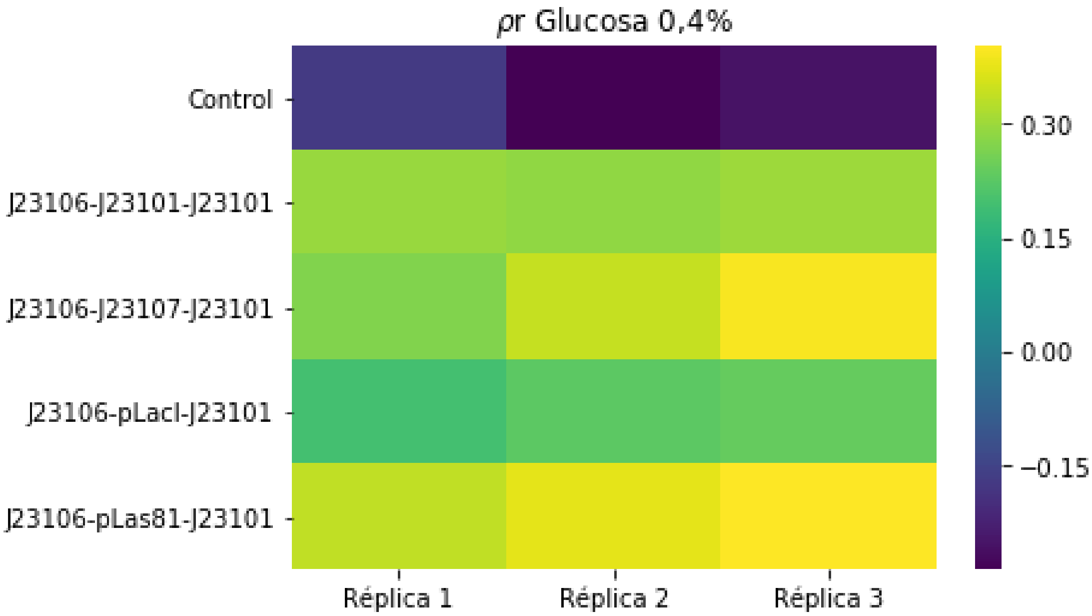
```
plt.xticks(X, ['Control Glucosa 1','Control Glicerol 1','Control Glucosa 2','Control Gl
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper left',ncol=2)
```

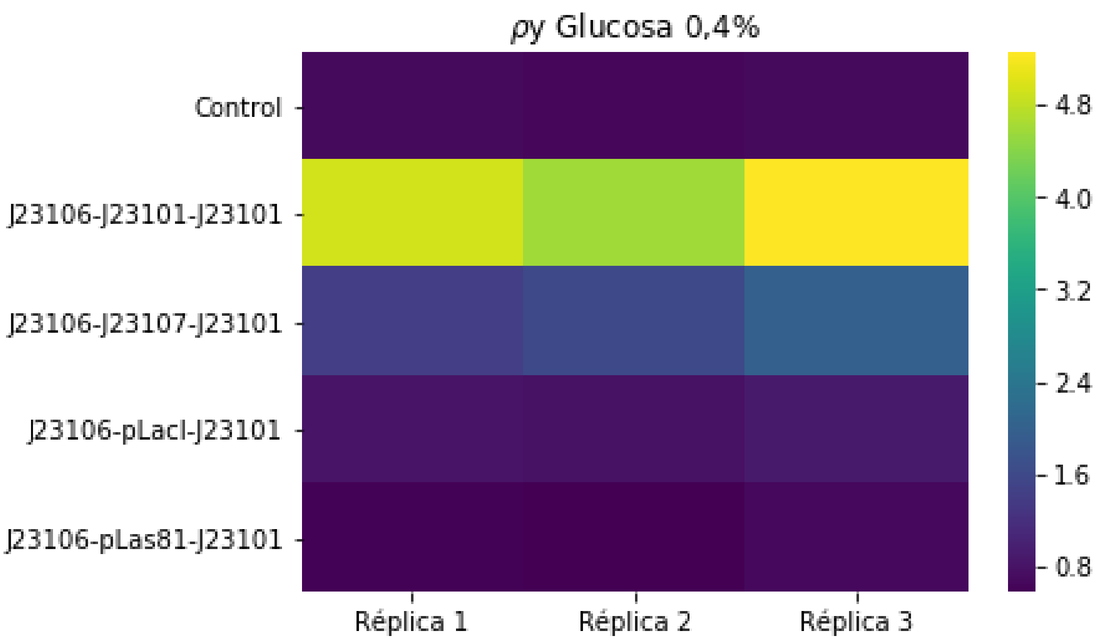
Out[85]: <matplotlib.legend.Legend at 0x1b20ec27e80>



In [86]:
```
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','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','

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

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

Out[86]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20bb0c080>

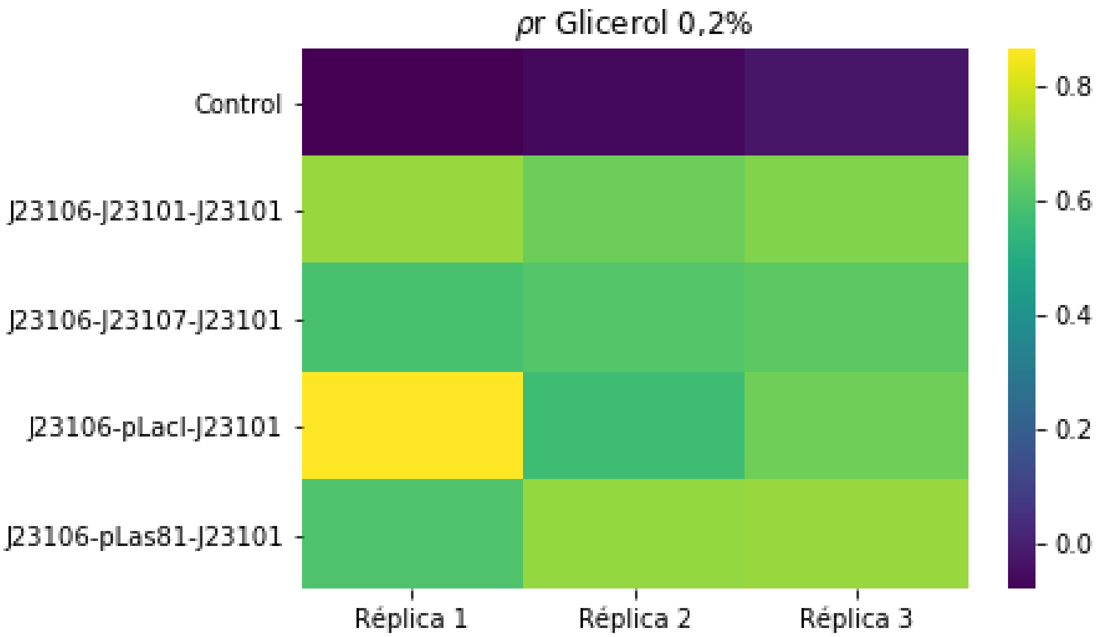


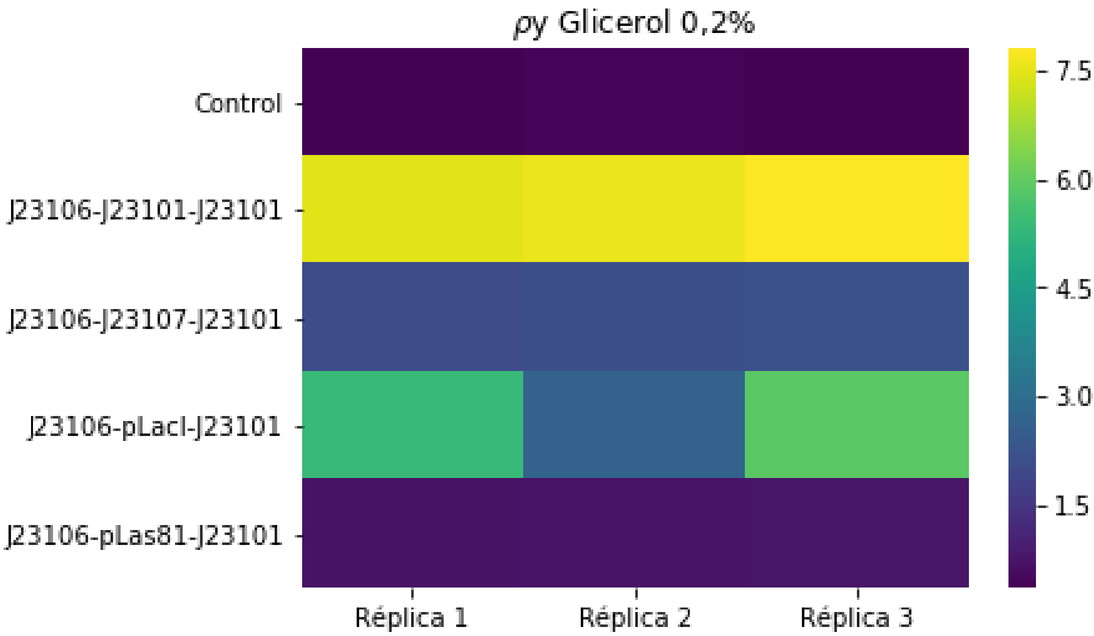



```
In [87]: ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
         ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,

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

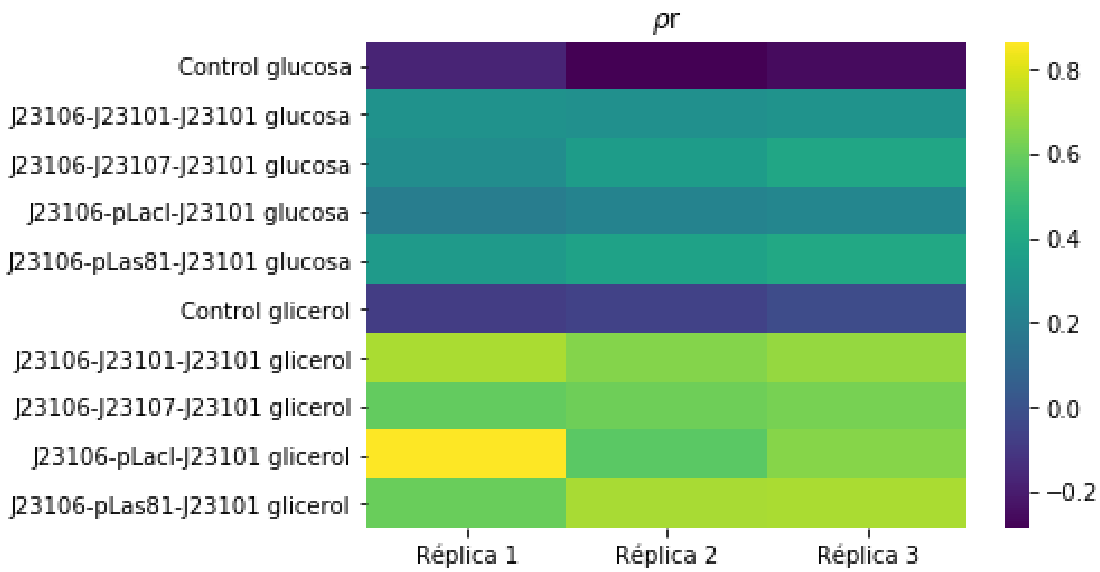
```
ylabel=['Control','J23106-J23101-J23101','J23106-J23107-J23101','J23106-pLacI-J23101','

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

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

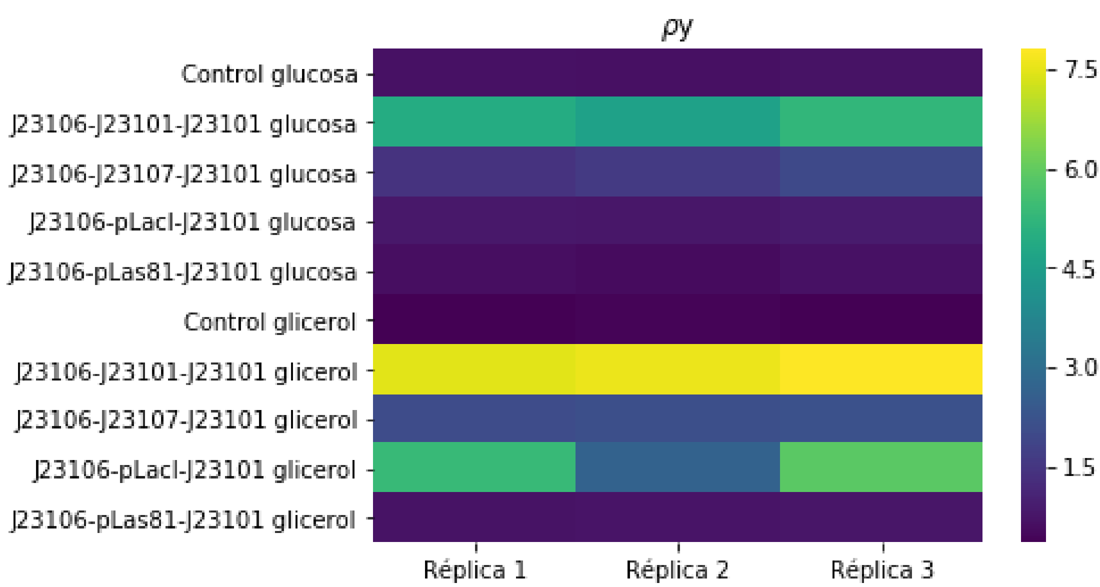
Out[87]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20b80def0>






```
In [88]: ro_rfpgli=[[pr4,pr5,pr6],[pr10,pr11,pr12],[pr16,pr17,pr18],[pr22,pr23,pr24],[pr28,pr29,
         ro_rfpglu=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr2
         bothrfp=[[pr1,pr2,pr3],[pr7,pr8,pr9],[pr13,pr14,pr15],[pr19,pr20,pr21],[pr25,pr26,pr27]
         ro_yfpglu=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py2
         ro_yfpgli=[[py4,py5,py6],[py10,py11,py12],[py16,py17,py18],[py22,py23,py24],[py28,py29,
         bothyfp=[[py1,py2,py3],[py7,py8,py9],[py13,py14,py15],[py19,py20,py21],[py25,py26,py27]

         xlabel=['Réplica 1','Réplica 2','Réplica 3']
         ylabel=['Control glucosa','J23106-J23101-J23101 glucosa','J23106-J23107-J23101 glucosa'

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

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

Out[88]: <matplotlib.axes._subplots.AxesSubplot at 0x1b20c19c358>
```

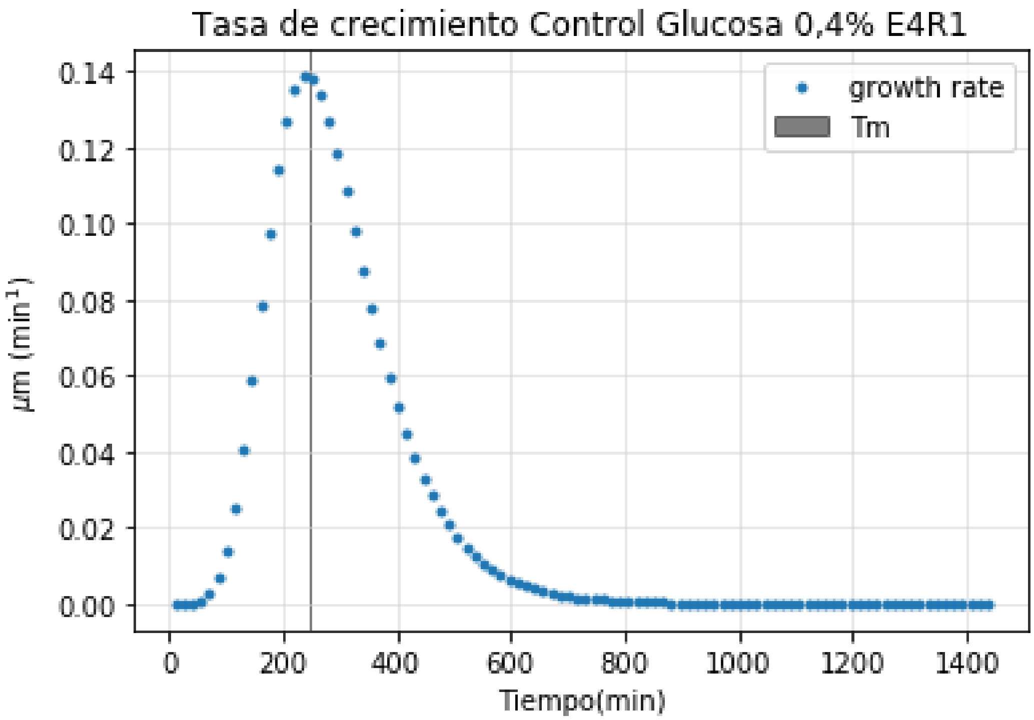





```
In [89]: #tasa de crecimiento
         ye1=((A1*np.exp(-np.exp((((um1*np.exp(1))/A1)*(l1-tt))+1))))
         #Con diff
         dy1=(np.diff(ye1))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R1')
         plt.xlabel('Tiempo(min)')
```

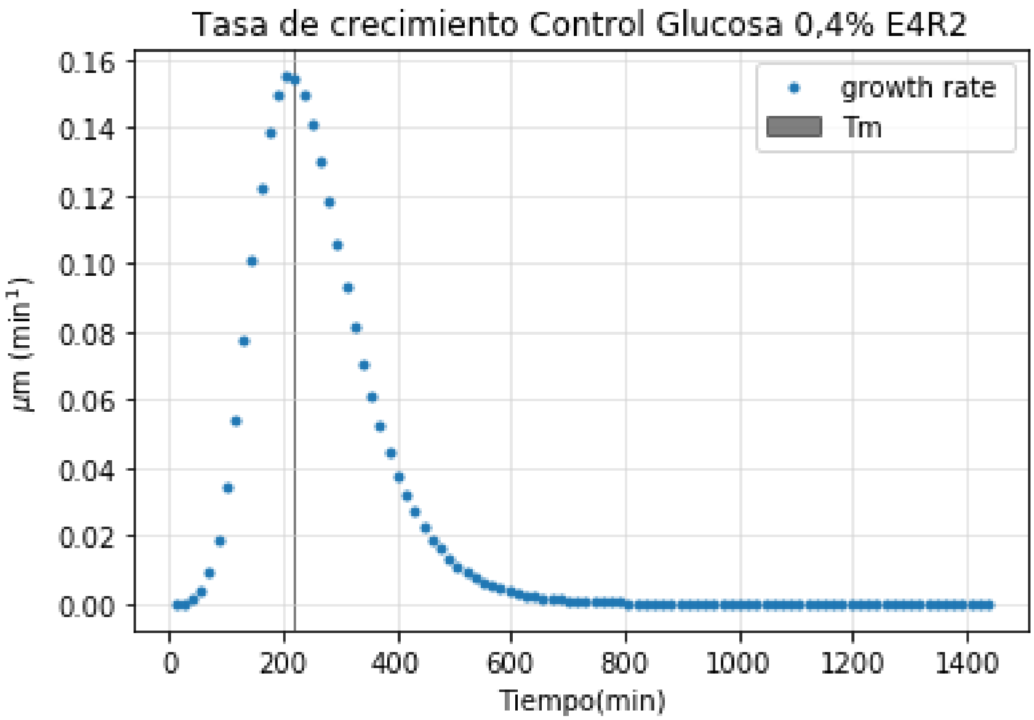
```
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm1,tm1, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy1,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

Out[89]: <matplotlib.legend.Legend at 0x1b20cd67f98>

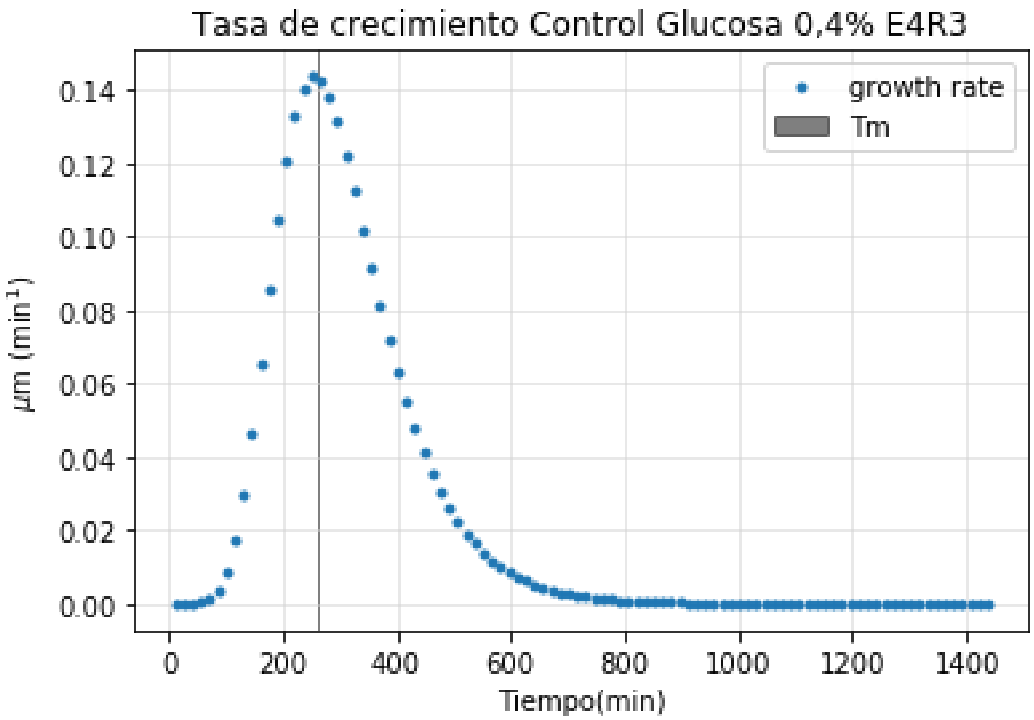


In [90]:
```
#tasa de crecimiento

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% E4R2')
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')
```

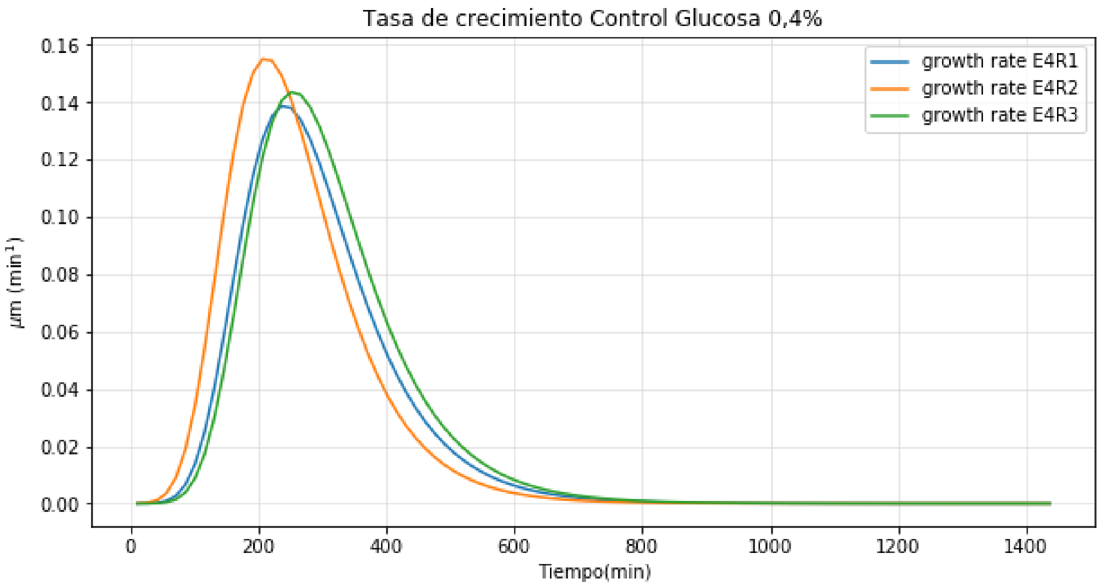
Out[90]: <matplotlib.legend.Legend at 0x1b20de529b0>



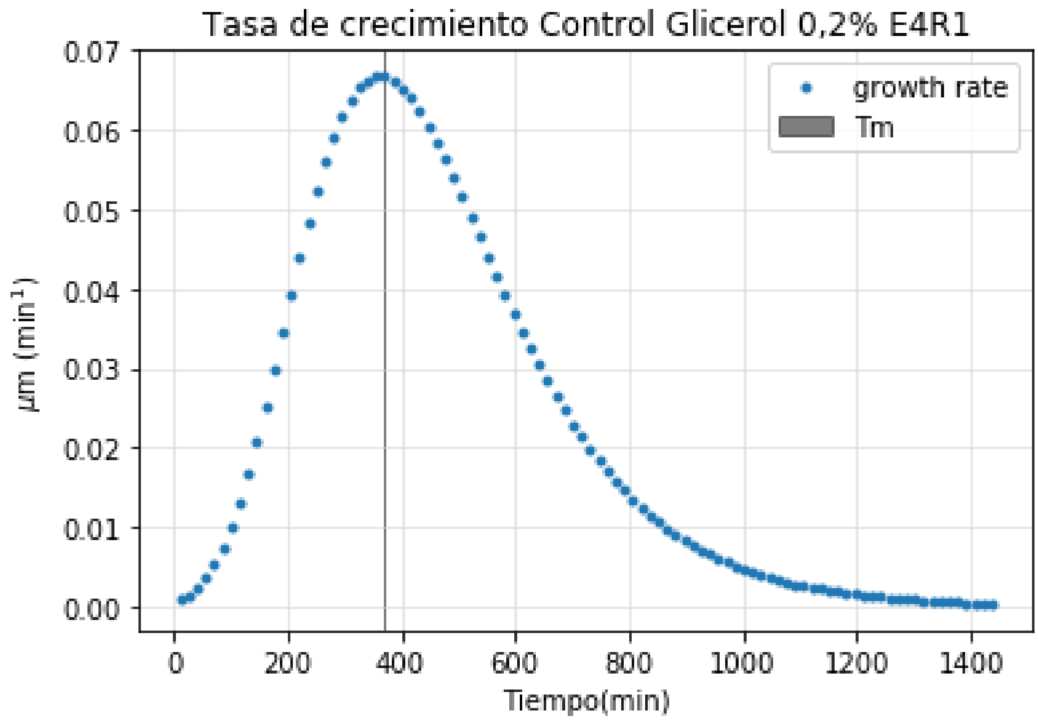

```
In [91]: #tasa de crecimiento
         ye3=((A3*np.exp(-np.exp((((um3*np.exp(1))/A3)*(l3-tt))+1))))
         #Con diff
         dy3=(np.diff(ye3))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glucosa 0,4% E4R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[91]: <matplotlib.legend.Legend at 0x1b20df2cac8>
```




```
In [92]: #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 E4R1')
         plt.plot(tt[:-1],dy2,label='growth rate E4R2')
         plt.plot(tt[:-1],dy3,label='growth rate E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[92]: <matplotlib.legend.Legend at 0x1b20e5f6b00>
```

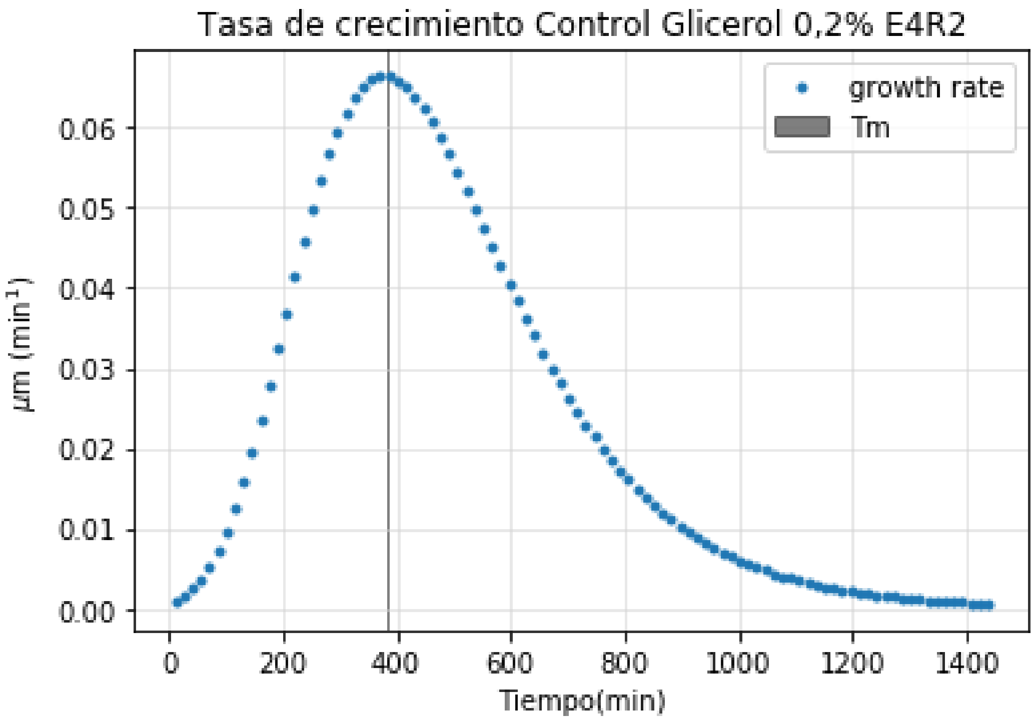



```
In [93]: #tasa de crecimiento
         ye4=((A4*np.exp(-np.exp((((um4*np.exp(1))/A4)*(l4-tt))+1))))
         #Con diff
         dy4=(np.diff(ye4))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R1')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[93]: <matplotlib.legend.Legend at 0x1b20e6c8f60>
```

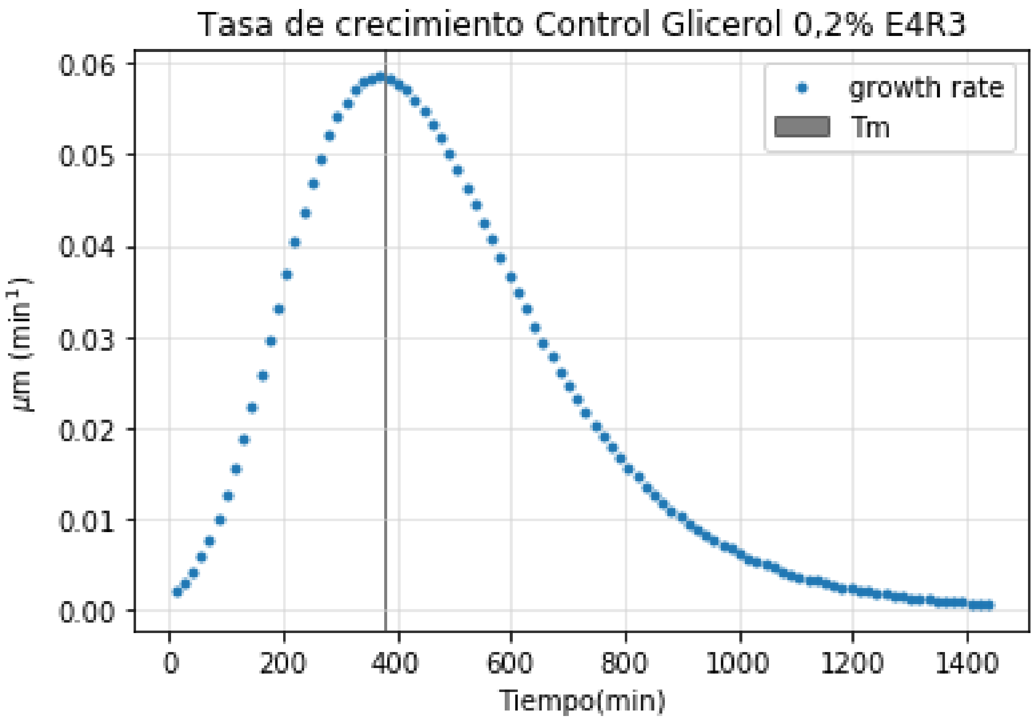


In [94]:
```
#tasa de crecimiento
ye5=((A5*np.exp(-np.exp((((um5*np.exp(1))/A5)*(l5-tt))+1))))
#Con diff
dy5=(np.diff(ye5))
plt.figure()
plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm5,tm5, color='k', alpha=0.5, label="Tm")
plt.plot(tt[:-1],dy5,'.',label='growth rate')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc='upper right')
```

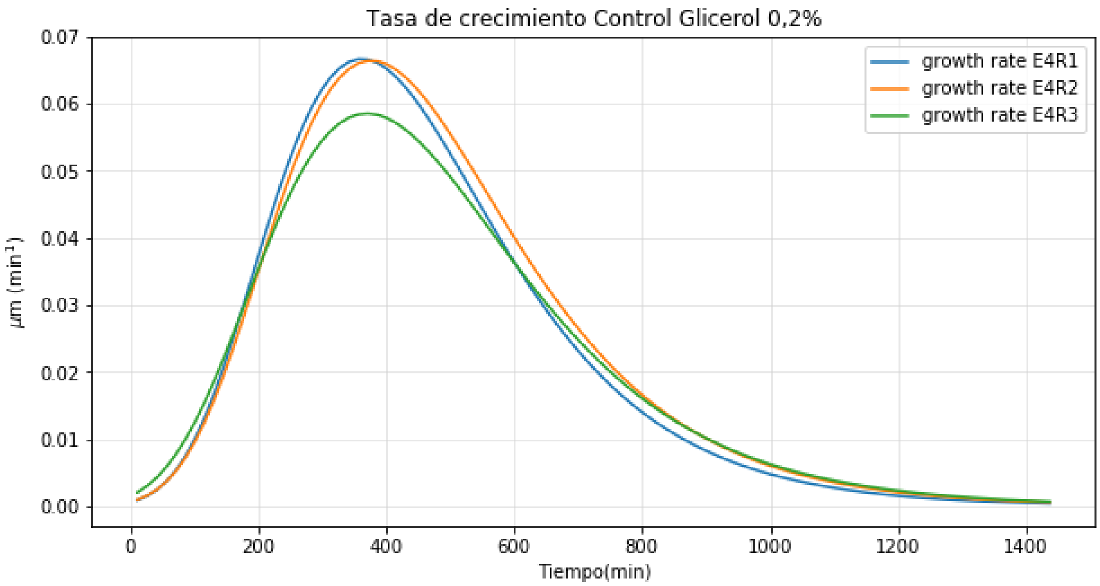
Out[94]:
```
<matplotlib.legend.Legend at 0x1b20efbfcc0>
```



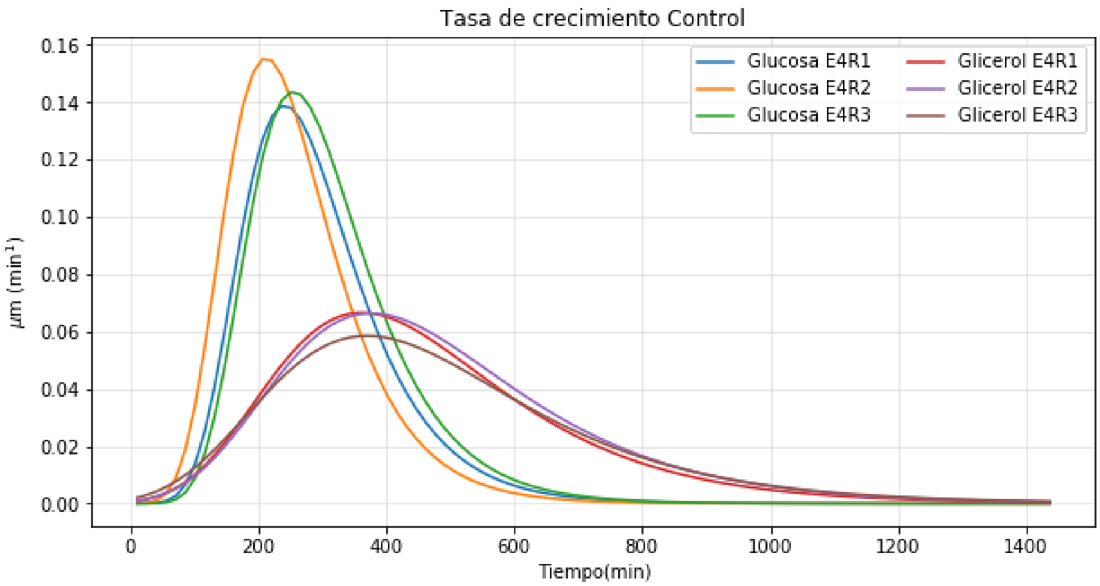

```
In [95]: #tasa de crecimiento
         ye6=((A6*np.exp(-np.exp((((um6*np.exp(1))/A6)*(l6-tt))+1))))
         #Con diff
         dy6=(np.diff(ye6))
         plt.figure()
         plt.title('Tasa de crecimiento Control Glicerol 0,2% E4R3')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[95]: <matplotlib.legend.Legend at 0x1b20fbc1ba8>
```



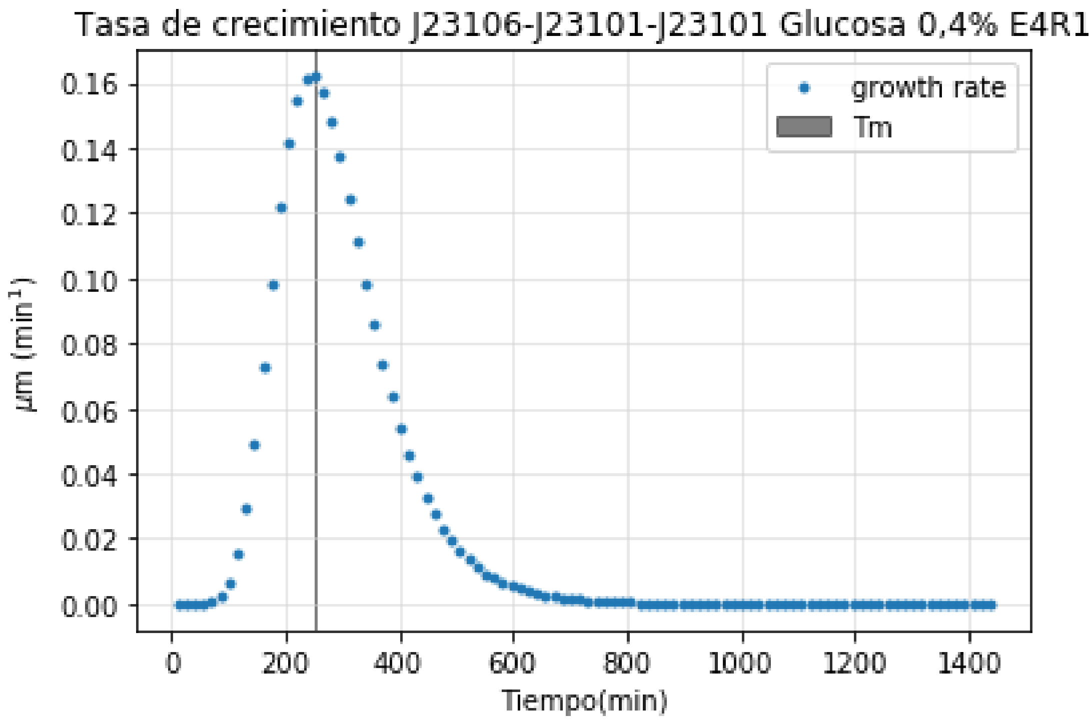

```
In [96]: #Tasas control réplicas glicerol
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control Glicerol 0,2%')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy4,label='growth rate E4R1')
         plt.plot(tt[:-1],dy5,label='growth rate E4R2')
         plt.plot(tt[:-1],dy6,label='growth rate E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right')

Out[96]: <matplotlib.legend.Legend at 0x1b20fc97c88>
```



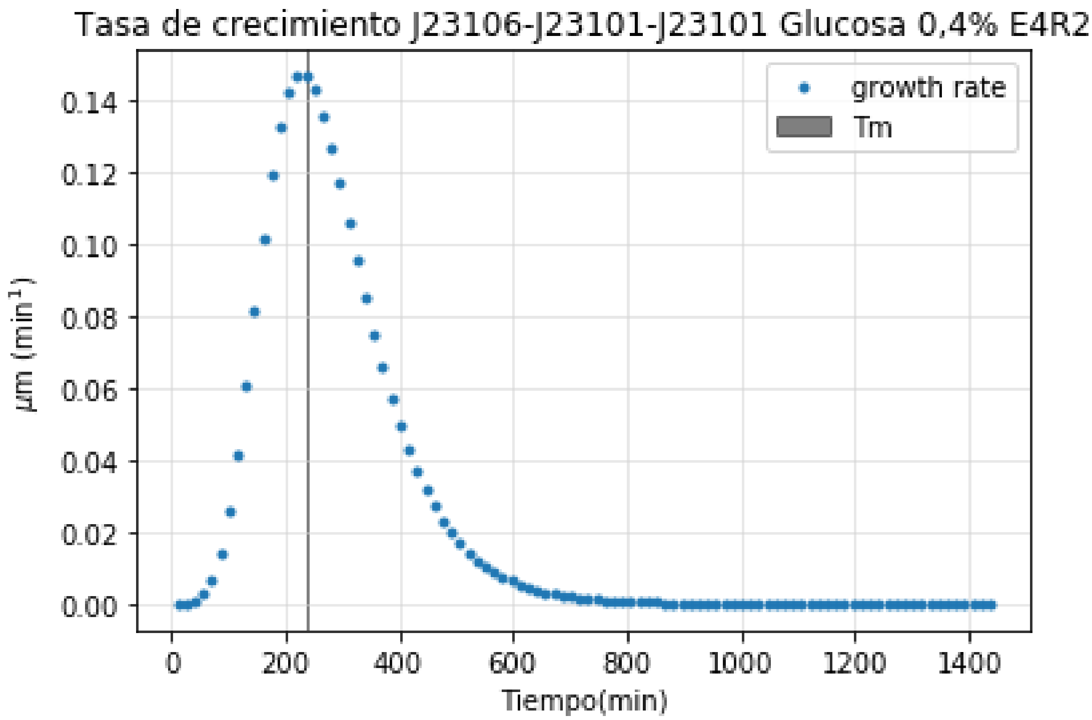

```
In [97]: #Tasas control réplicas controles
         plt.figure(figsize=(10,5))
         plt.title('Tasa de crecimiento Control')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (min$^1$)')
         plt.plot(tt[:-1],dy1,label='Glucosa E4R1')
         plt.plot(tt[:-1],dy2,label='Glucosa E4R2')
         plt.plot(tt[:-1],dy3,label='Glucosa E4R3')
         plt.plot(tt[:-1],dy4,label='Glicerol E4R1')
         plt.plot(tt[:-1],dy5,label='Glicerol E4R2')
         plt.plot(tt[:-1],dy6,label='Glicerol E4R3')
         plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
         plt.legend(loc='upper right',ncol=2)

Out[97]: <matplotlib.legend.Legend at 0x1b210d51da0>
```



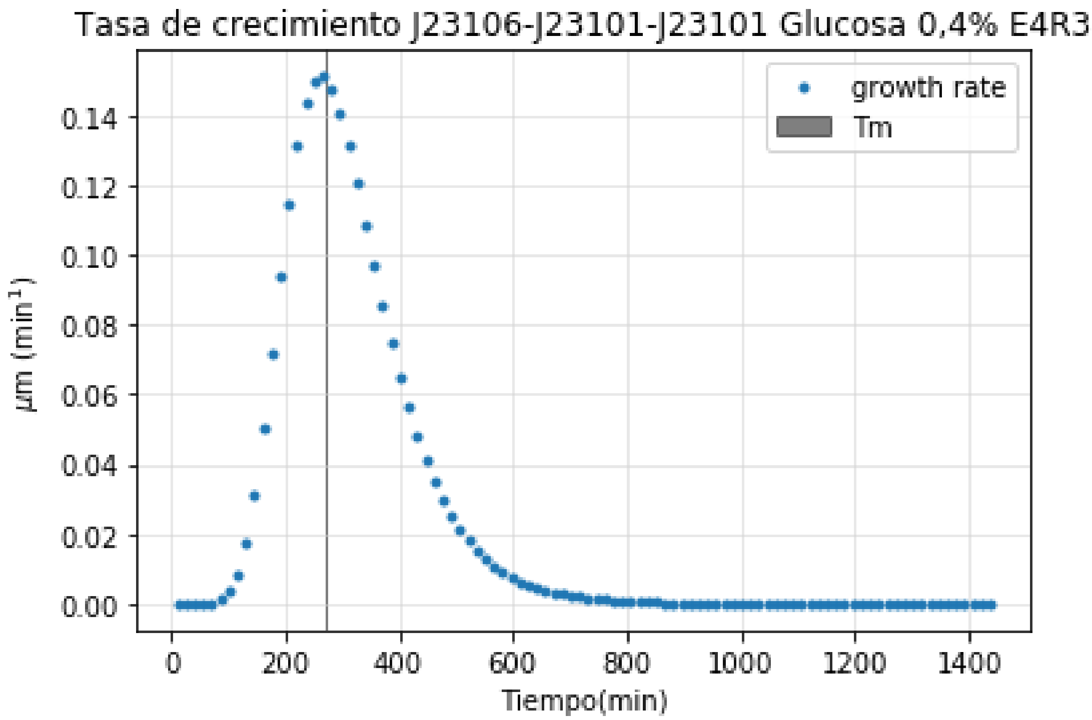

```
In [98]: #tasa de crecimiento
         ye7=((A7*np.exp(-np.exp((((um7*np.exp(1))/A7)*(l7-tt))+1))))
         #Con diff
         dy7=(np.diff(ye7))
         plt.figure()
         plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R1')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[98]: <matplotlib.legend.Legend at 0x1b210e30940>
```



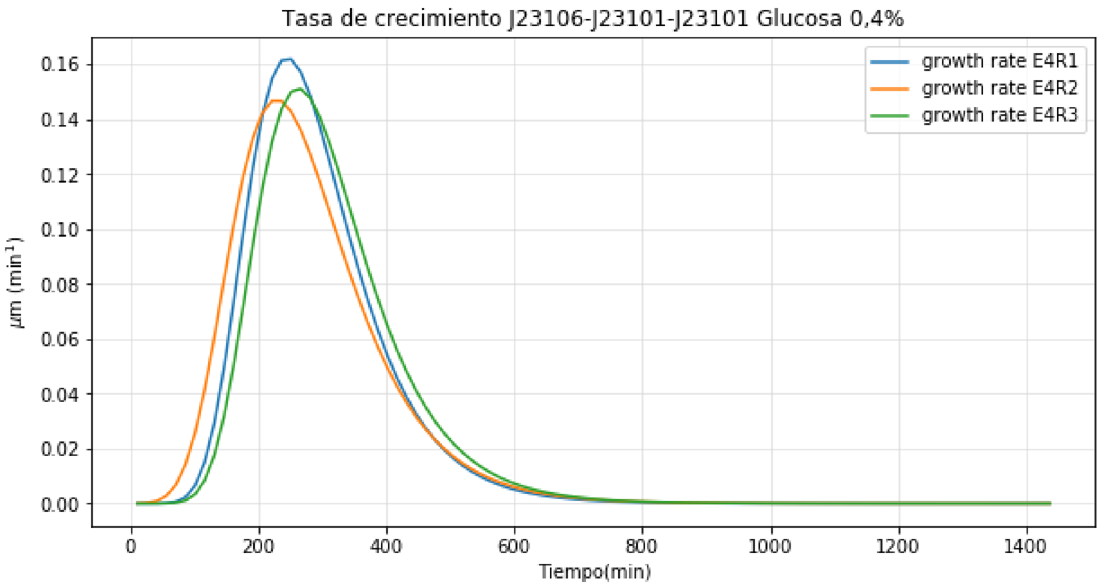

```
In [99]: #tasa de crecimiento
         ye8=((A8*np.exp(-np.exp((((um8*np.exp(1))/A8)*(l8-tt))+1))))
         #Con diff
         dy8=(np.diff(ye8))
         plt.figure()
         plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R2')
         plt.xlabel('Tiempo(min)')
         plt.ylabel(r'$\mu$m (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[99]: <matplotlib.legend.Legend at 0x1b210f13cf8>
```



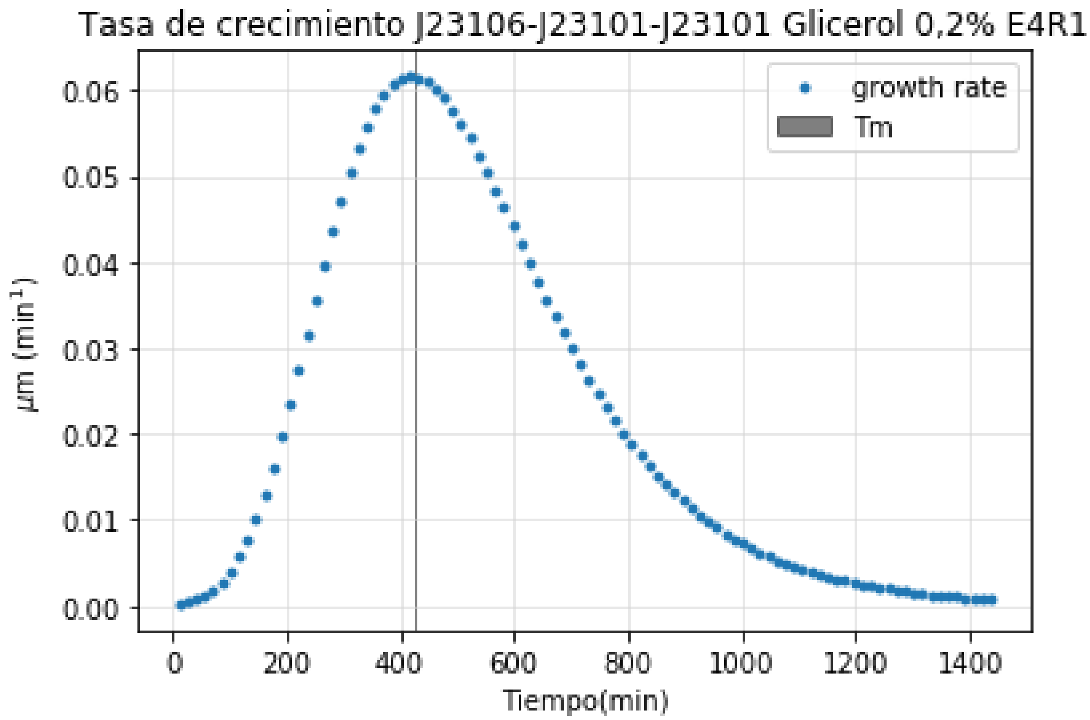

```
In [100]: #tasa de crecimiento
          ye9=((A9*np.exp(-np.exp((((um9*np.exp(1))/A9)*(l9-tt))+1))))
          #Con diff
          dy9=(np.diff(ye9))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4% E4R3')
          plt.xlabel('Tiempo(min)')
          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[100]: <matplotlib.legend.Legend at 0x1b20ce15160>
```



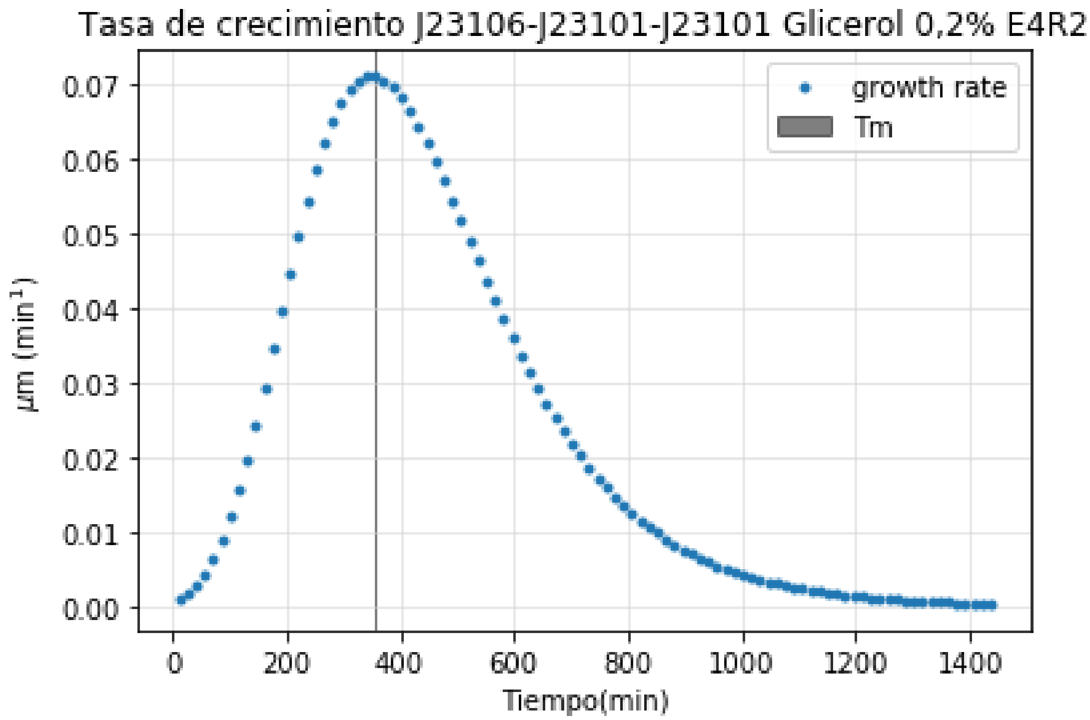

```
In [101]: #Tasas J23101-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy7,label='growth rate E4R1')
          plt.plot(tt[:-1],dy8,label='growth rate E4R2')
          plt.plot(tt[:-1],dy9,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[101]: <matplotlib.legend.Legend at 0x1b20e128da0>
```



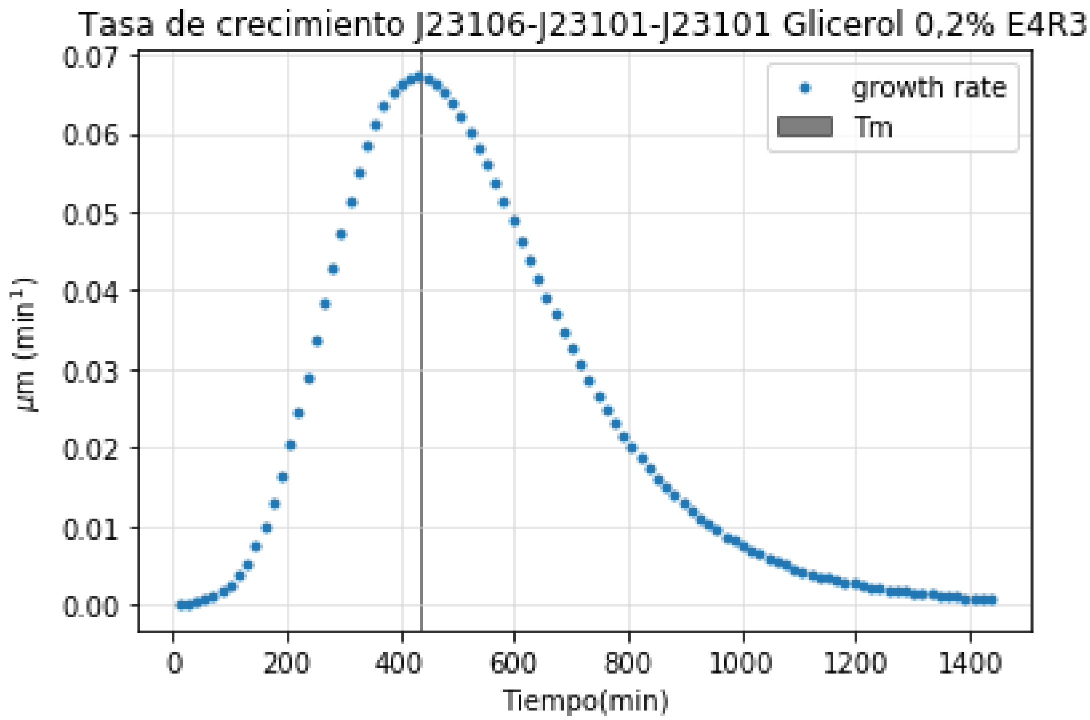

```
In [102]: #tasa de crecimiento
          ye10=((A10*np.exp(-np.exp((((um10*np.exp(1))/A10)*(l10-tt))+1))))
          #Con diff
          dy10=(np.diff(ye10))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          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[102]: <matplotlib.legend.Legend at 0x1b209cb4a90>
```



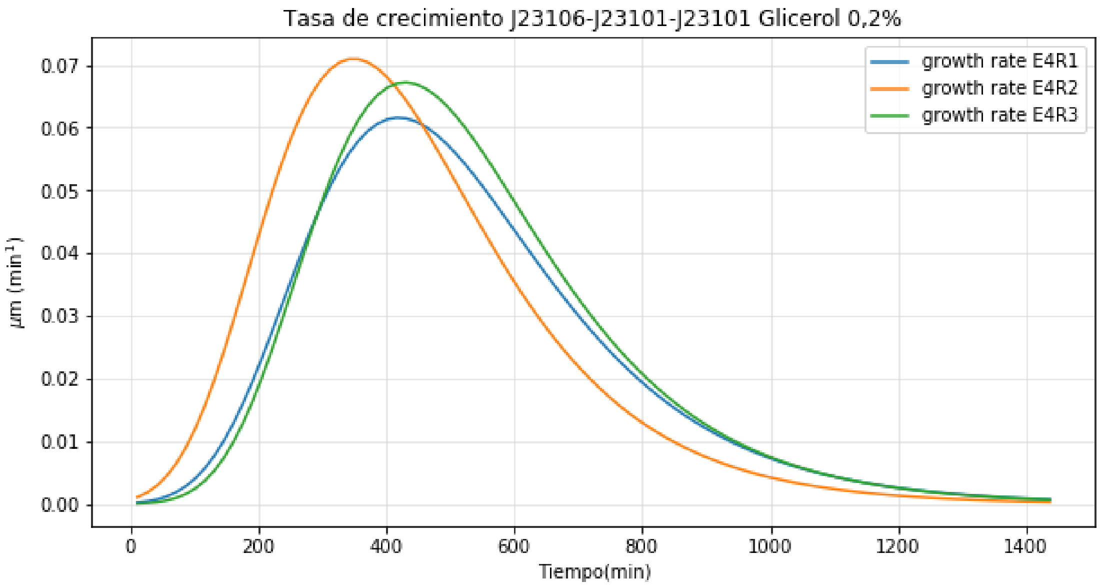

```
In [103]: #tasa de crecimiento
          ye11=((A11*np.exp(-np.exp((((um11*np.exp(1))/A11)*(l11-tt))+1))))
          #Con diff
          dy11=(np.diff(ye11))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R2')
          plt.xlabel('Tiempo(min)')
          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[103]: <matplotlib.legend.Legend at 0x1b20bb5e978>
```



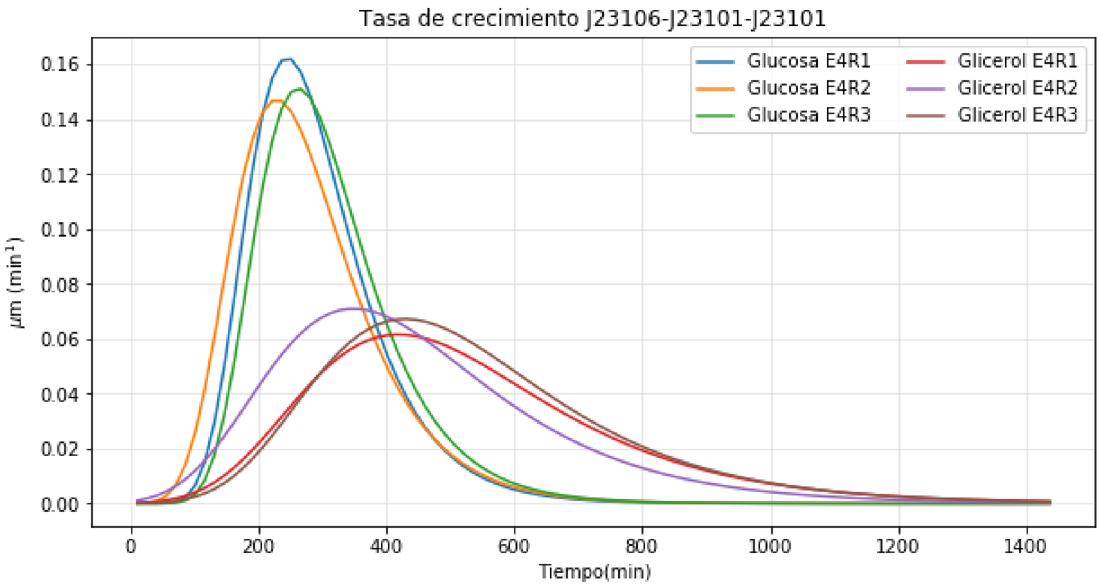

```
In [104]: #tasa de crecimiento
          ye12=((A12*np.exp(-np.exp((((um12*np.exp(1))/A12)*(l12-tt))+1))))
          #Con diff
          dy12=(np.diff(ye12))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          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[104]: <matplotlib.legend.Legend at 0x1b20f390eb8>
```



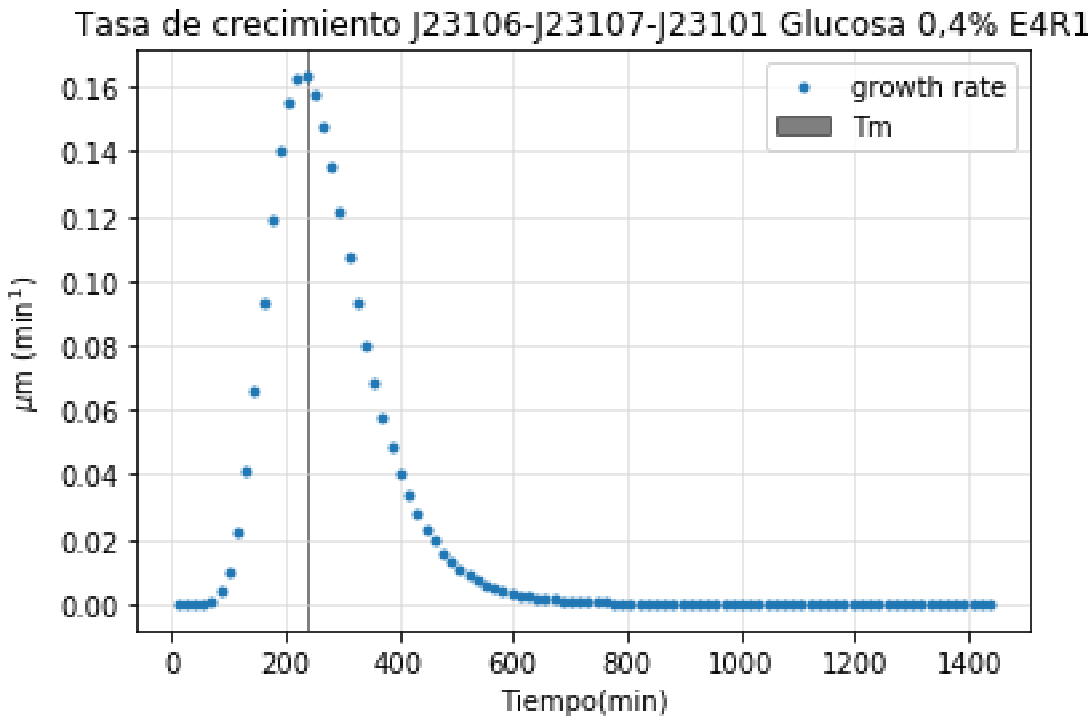

```
In [105]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23101-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy10,label='growth rate E4R1')
          plt.plot(tt[:-1],dy11,label='growth rate E4R2')
          plt.plot(tt[:-1],dy12,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[105]: <matplotlib.legend.Legend at 0x1b20f226780>
```




```
In [106]: #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],dy7,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy8,label='Glucosa E4R2')
          plt.plot(tt[:-1],dy9,label='Glucosa E4R3')
          plt.plot(tt[:-1],dy10,label='Glicerol E4R1')
          plt.plot(tt[:-1],dy11,label='Glicerol E4R2')
          plt.plot(tt[:-1],dy12,label='Glicerol E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[106]: <matplotlib.legend.Legend at 0x1b20f46ad68>
```

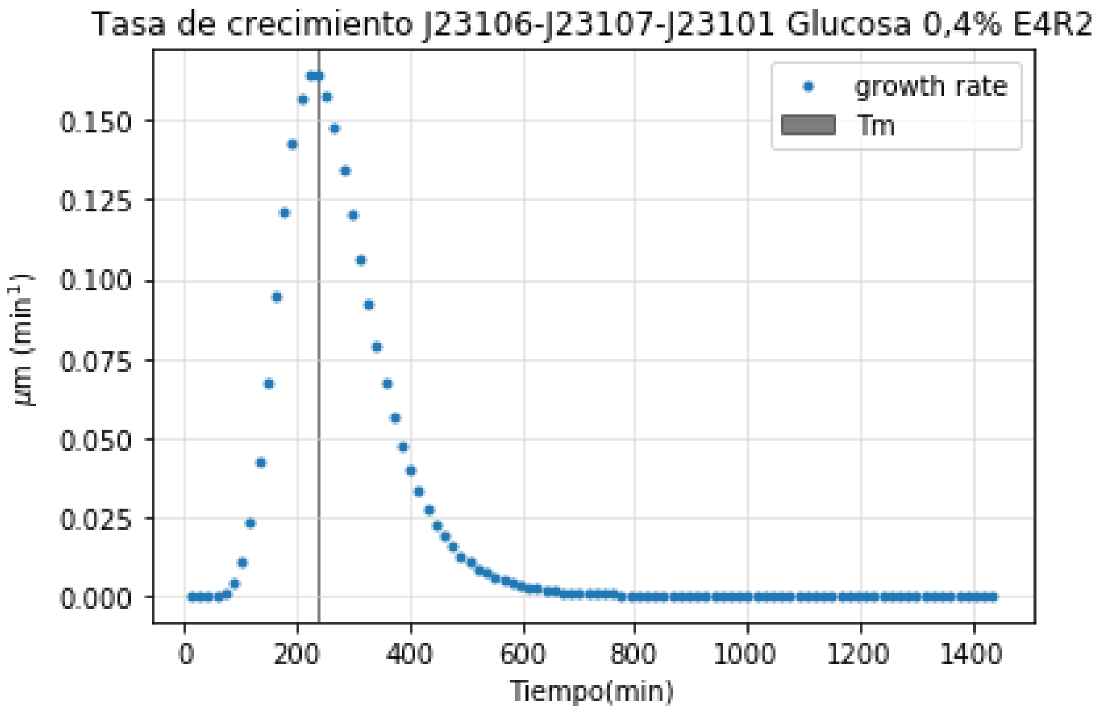



```
In [107]: #tasa de crecimiento
          ye13=((A13*np.exp(-np.exp((((um13*np.exp(1))/A13)*(l13-tt))+1))))
          #Con diff
          dy13=(np.diff(ye13))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glucosa 0,4% E4R1')
          plt.xlabel('Tiempo(min)')
          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[107]: <matplotlib.legend.Legend at 0x1b20f54ccf8>
```

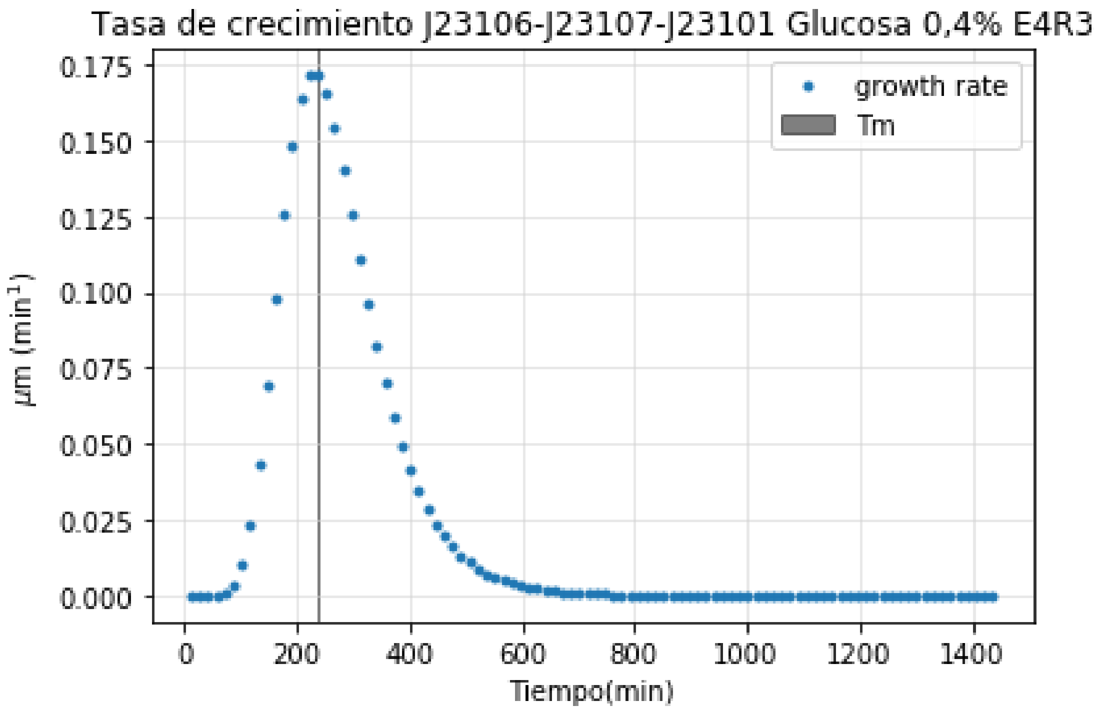


In [108]:
```
#tasa de crecimiento
ye14=((A14*np.exp(-np.exp((((um14*np.exp(1))/A14)*(l14-tt))+1))))
#Con diff
dy14=(np.diff(ye14))
plt.figure()
plt.title('Tasa de crecimiento J23106-J23107-J23101 Glucosa 0,4% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm14,tm14, color='k', 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')
```

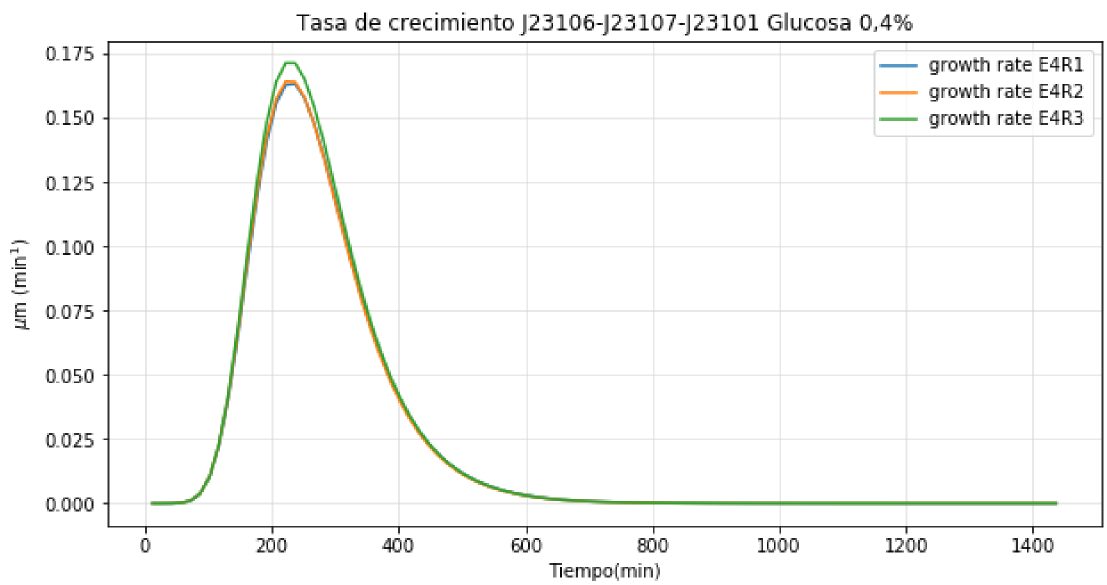
Out[108]: <matplotlib.legend.Legend at 0x1b20f606a90>



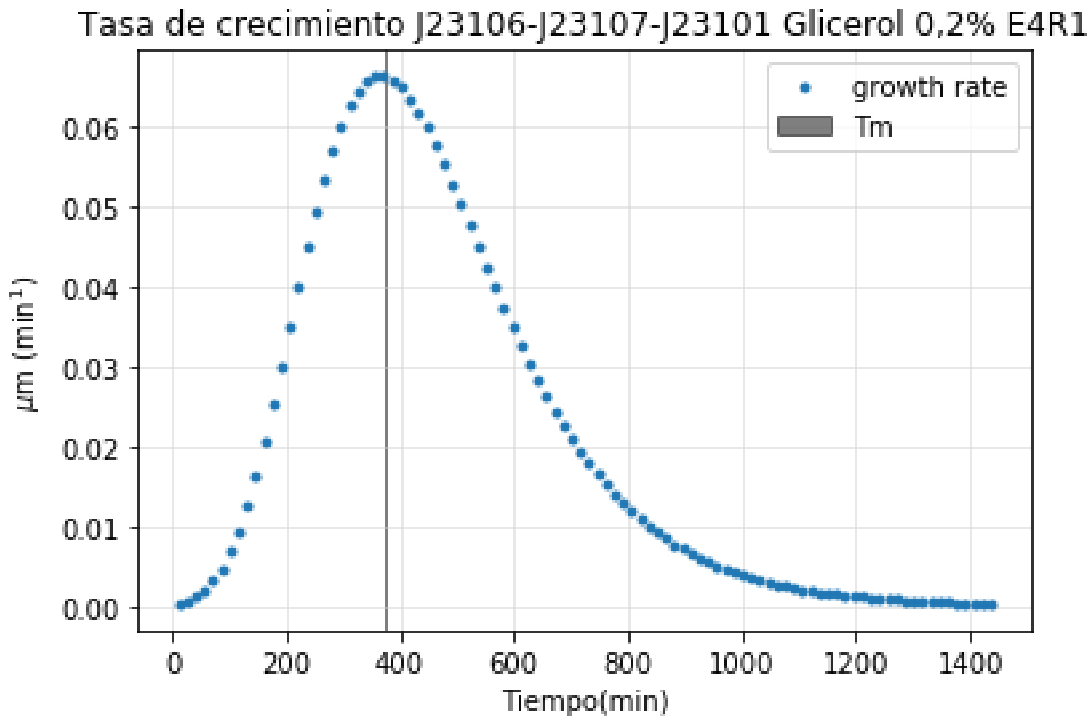

```
In [109]: #tasa de crecimiento
          ye15=((A15*np.exp(-np.exp((((um15*np.exp(1))/A15)*(l15-tt))+1))))
          #Con diff
          dy15=(np.diff(ye15))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glucosa 0,4% E4R3')
          plt.xlabel('Tiempo(min)')
          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[109]: <matplotlib.legend.Legend at 0x1b20f6d7cc0>
```



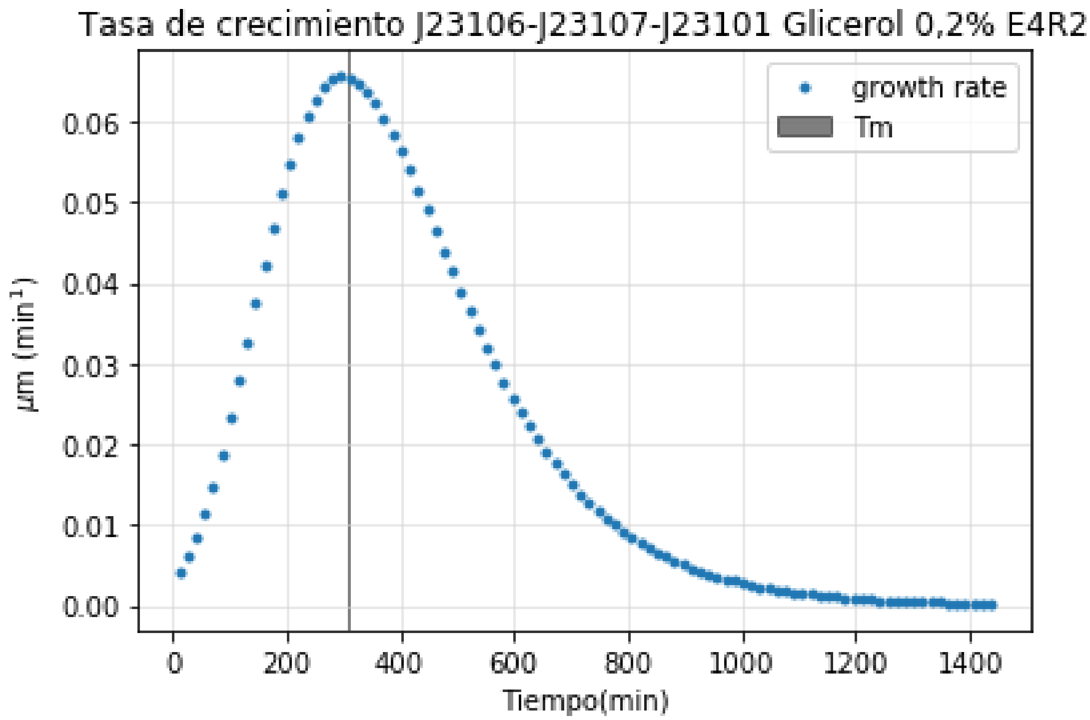

```
In [110]: #Tasas plux76-J23101-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='growth rate 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[110]: <matplotlib.legend.Legend at 0x1b20f7a8cf8>
```



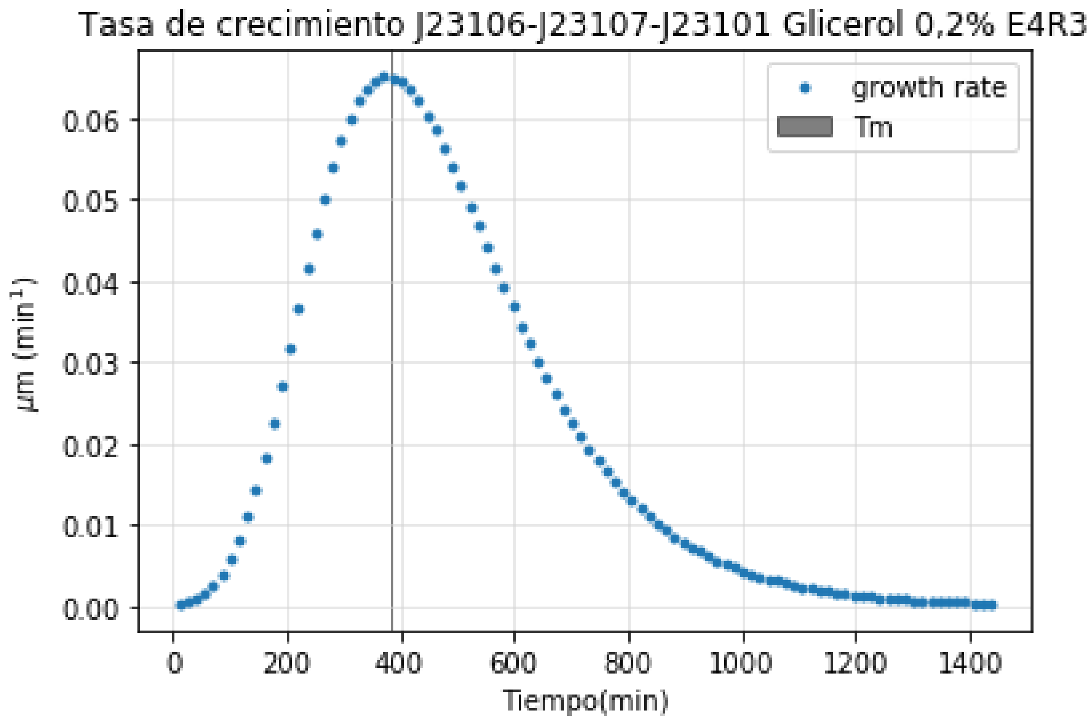

```
In [111]: #tasa de crecimiento
          ye16=((A16*np.exp(-np.exp((((um16*np.exp(1))/A16)*(l16-tt))+1))))
          #Con diff
          dy16=(np.diff(ye16))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          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[111]: <matplotlib.legend.Legend at 0x1b21104b860>
```



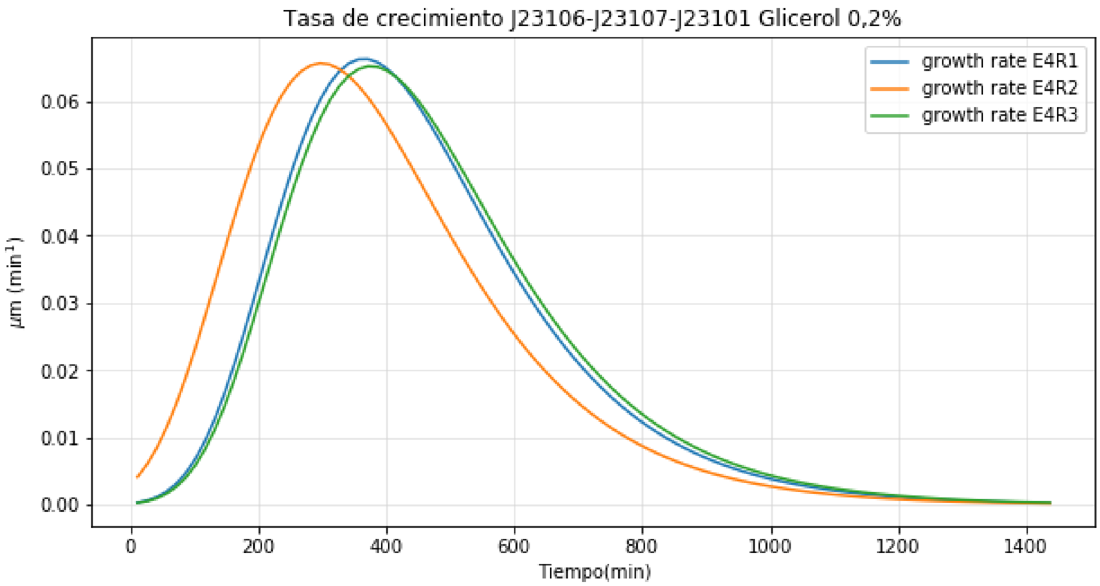

```
In [112]: #tasa de crecimiento
          ye17=((A17*np.exp(-np.exp((((um17*np.exp(1))/A17)*(l17-tt))+1))))
          #Con diff
          dy17=(np.diff(ye17))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glicerol 0,2% E4R2')
          plt.xlabel('Tiempo(min)')
          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[112]: <matplotlib.legend.Legend at 0x1b2120e7f98>
```



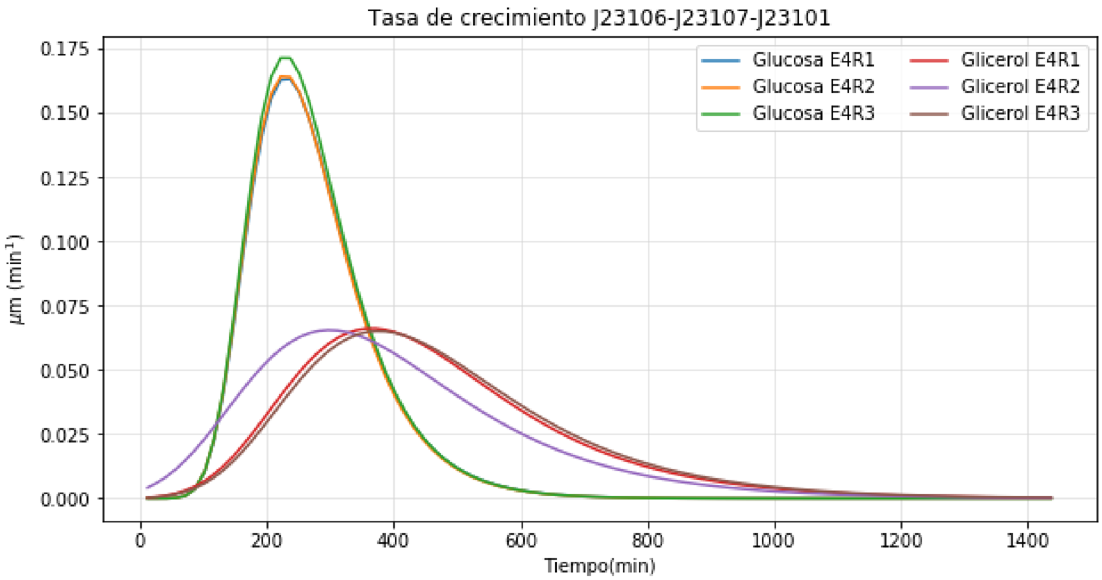

```
In [113]: #tasa de crecimiento
          ye18=((A18*np.exp(-np.exp((((um18*np.exp(1))/A18)*(l18-tt))+1))))
          #Con diff
          dy18=(np.diff(ye18))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          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[113]: <matplotlib.legend.Legend at 0x1b2121bd908>
```



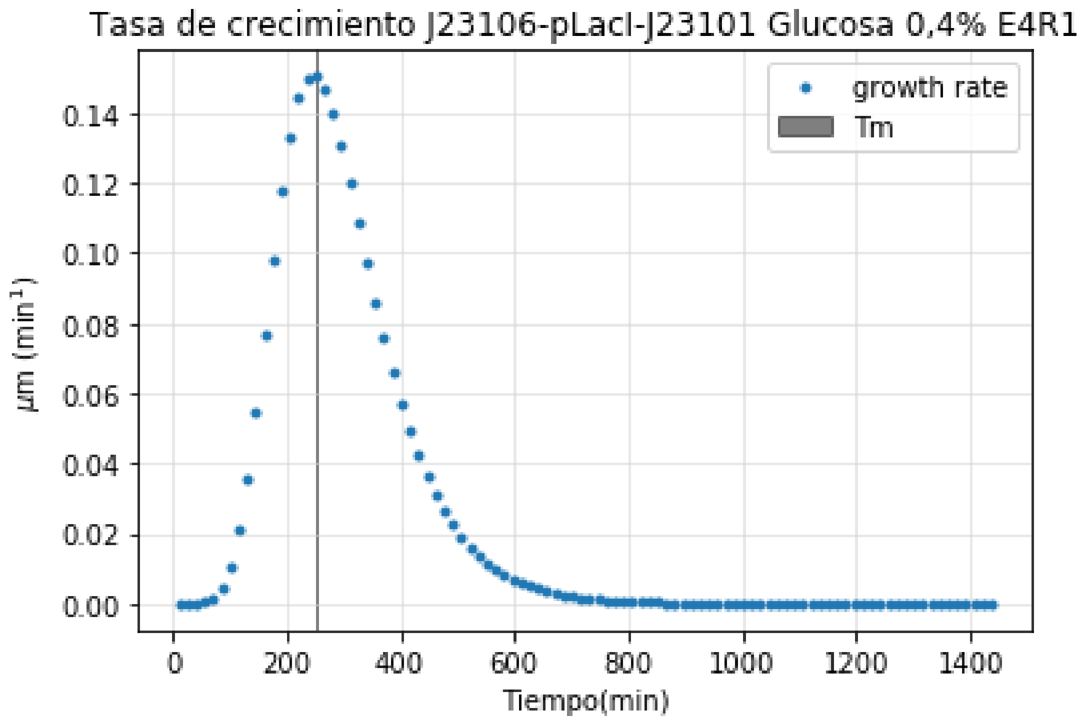

```
In [114]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23107-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy16,label='growth rate E4R1')
          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[114]: <matplotlib.legend.Legend at 0x1b212296c88>
```



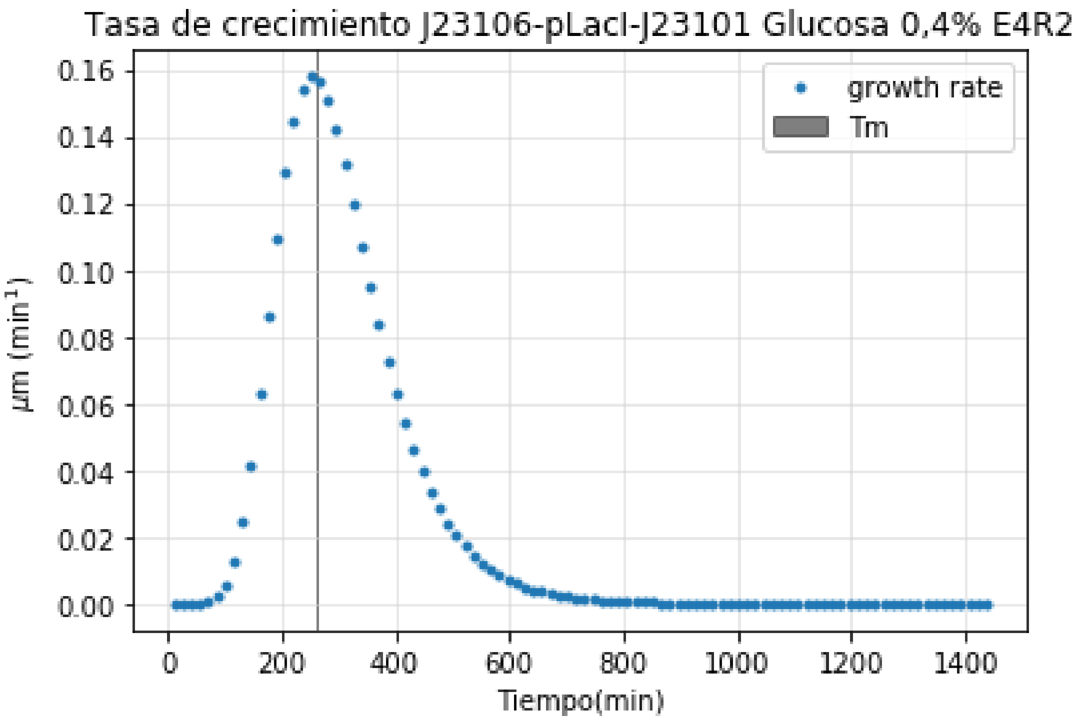

```
In [115]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-J23107-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy13,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy14,label='Glucosa 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[115]: <matplotlib.legend.Legend at 0x1b212381b38>
```



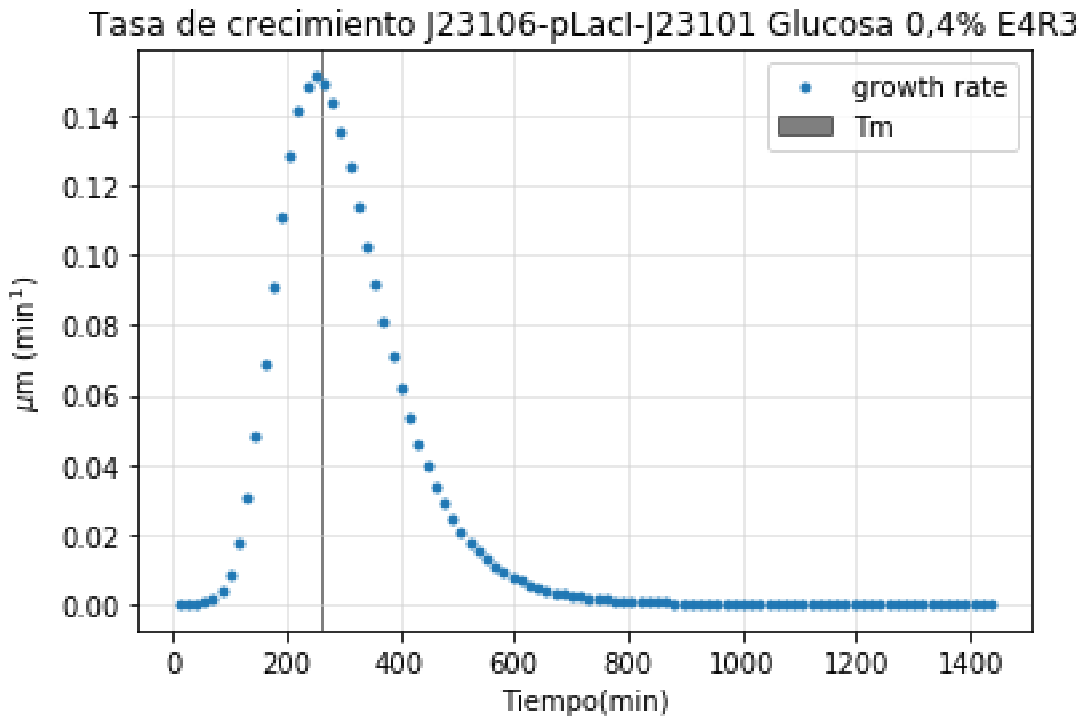

```
In [116]: #tasa de crecimiento
          ye19=((A19*np.exp(-np.exp((((um19*np.exp(1))/A19)*(l19-tt))+1))))
          #Con diff
          dy19=(np.diff(ye19))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glucosa 0,4% E4R1')
          plt.xlabel('Tiempo(min)')
          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[116]: <matplotlib.legend.Legend at 0x1b21245abe0>
```



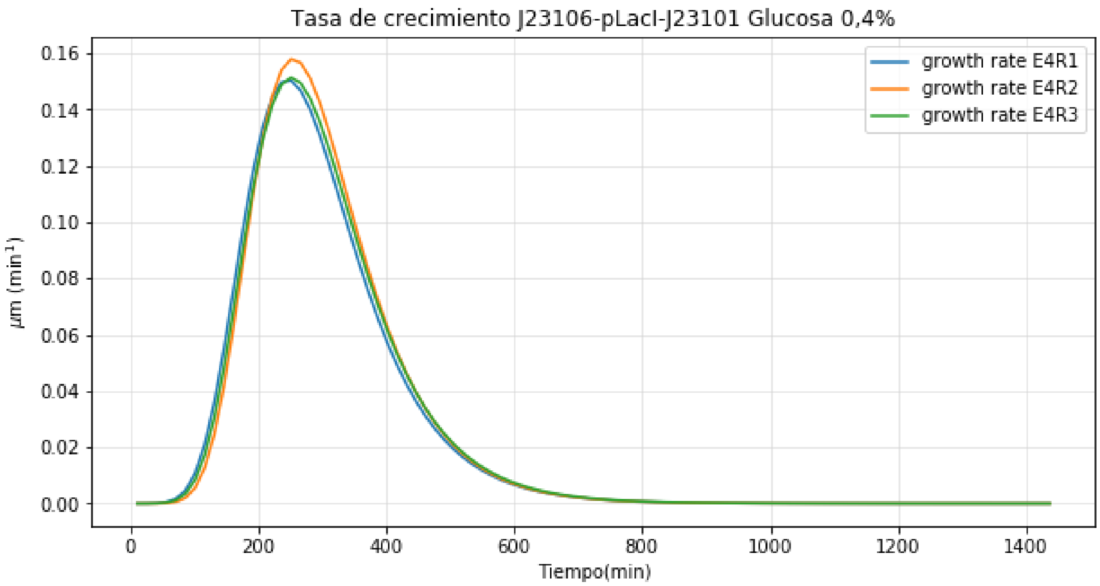

```
In [117]: #tasa de crecimiento
          ye20=((A20*np.exp(-np.exp((((um20*np.exp(1))/A20)*(l20-tt))+1))))
          #Con diff
          dy20=(np.diff(ye20))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glucosa 0,4% E4R2')
          plt.xlabel('Tiempo(min)')
          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[117]: <matplotlib.legend.Legend at 0x1b21253dfd0>
```



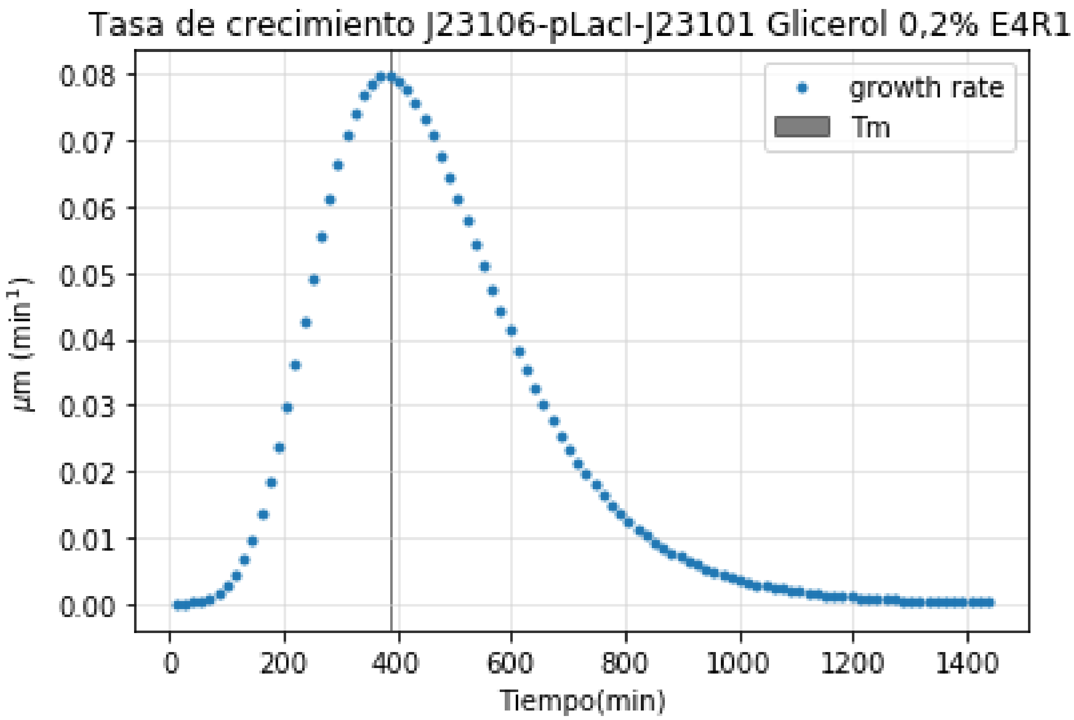

```
In [118]: #tasa de crecimiento
          ye21=((A21*np.exp(-np.exp((((um21*np.exp(1))/A21)*(l21-tt))+1))))
          #Con diff
          dy21=(np.diff(ye21))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glucosa 0,4% E4R3')
          plt.xlabel('Tiempo(min)')
          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[118]: <matplotlib.legend.Legend at 0x1b2125fccc0>
```



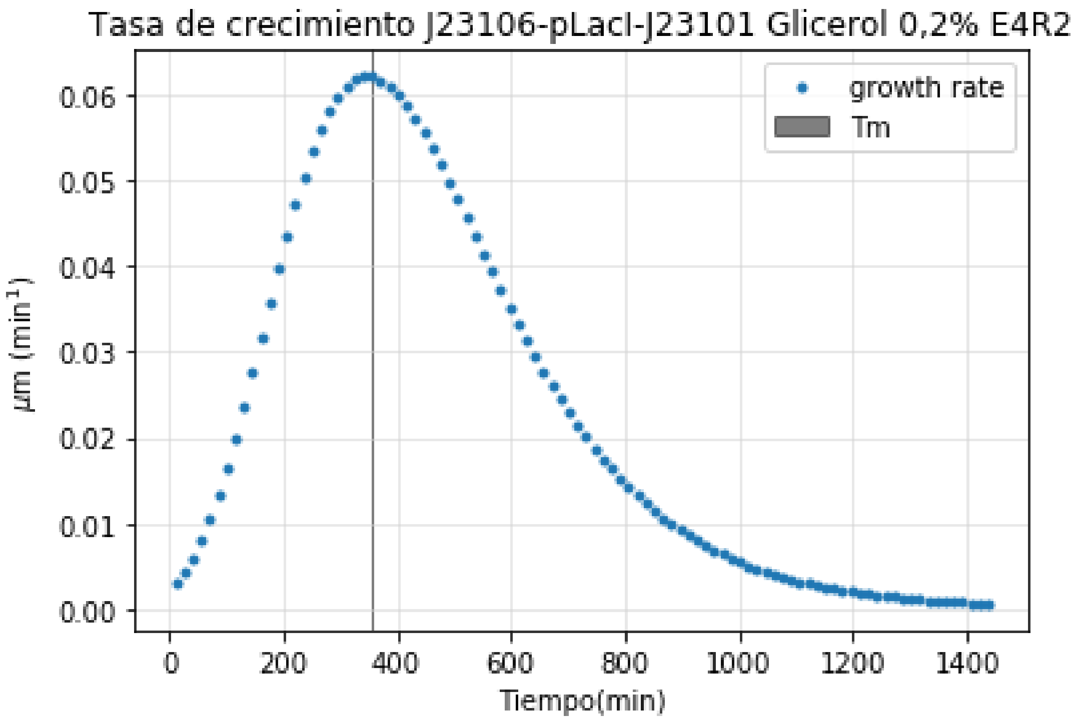

```
In [119]: #Tasas plux76-ptet-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='growth rate E4R1')
          plt.plot(tt[:-1],dy20,label='growth rate E4R2')
          plt.plot(tt[:-1],dy21,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[119]: <matplotlib.legend.Legend at 0x1b2126d57b8>
```



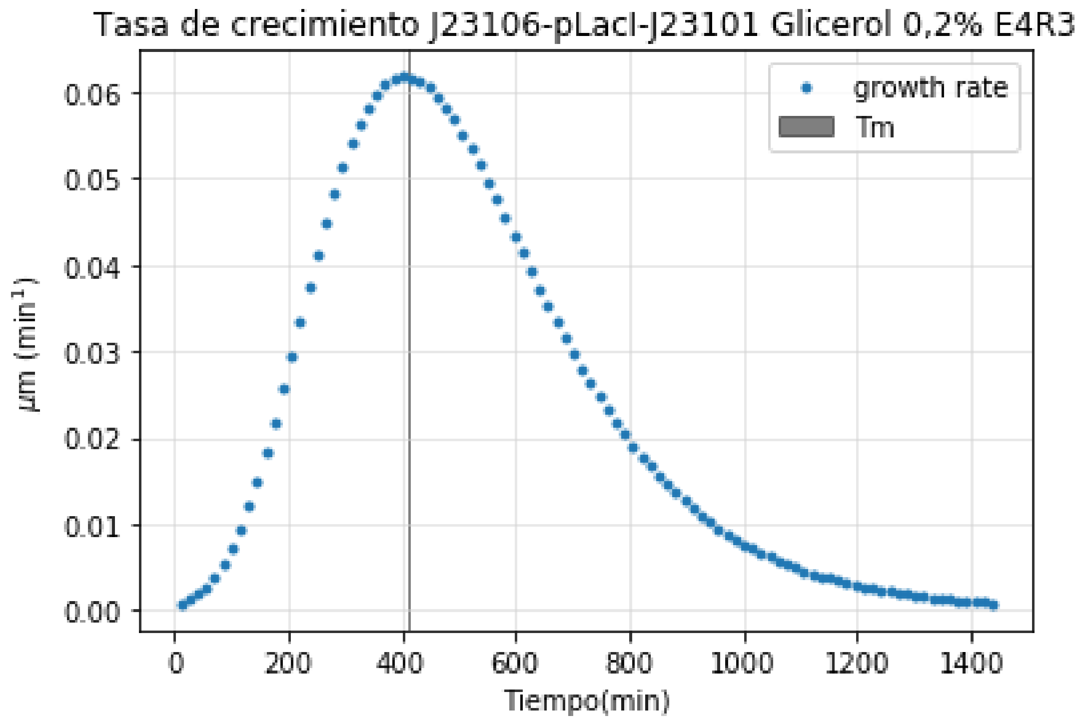

```
In [120]: #tasa de crecimiento
          ye22=((A22*np.exp(-np.exp((((um22*np.exp(1))/A22)*(l22-tt))+1))))
          #Con diff
          dy22=(np.diff(ye22))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          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[120]: <matplotlib.legend.Legend at 0x1b212799e80>
```



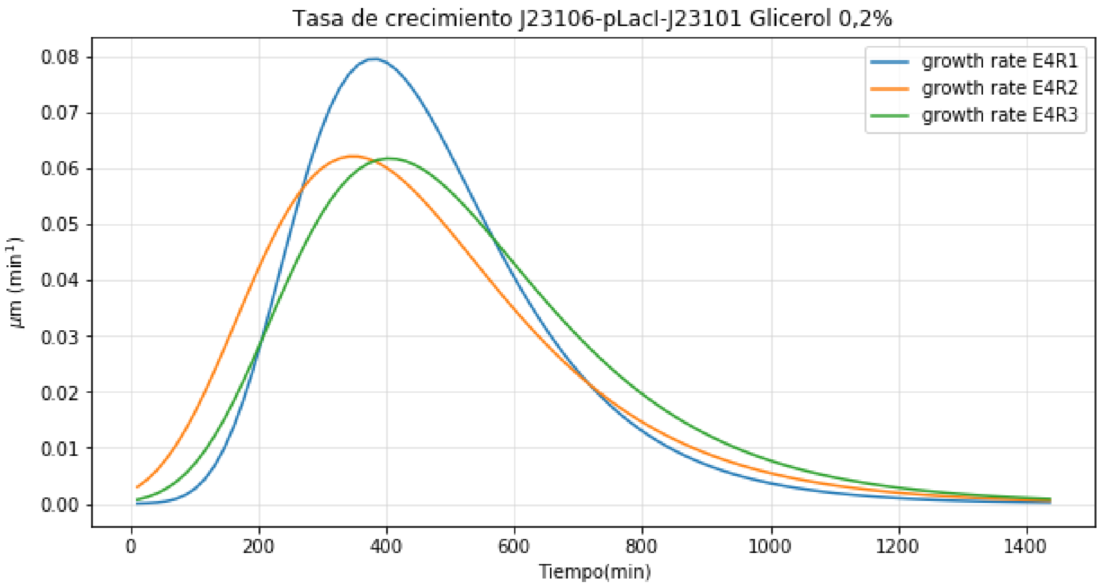

```
In [121]: #tasa de crecimiento
          ye23=((A23*np.exp(-np.exp((((um23*np.exp(1))/A23)*(l23-tt))+1))))
          #Con diff
          dy23=(np.diff(ye23))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glicerol 0,2% E4R2')
          plt.xlabel('Tiempo(min)')
          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[121]: <matplotlib.legend.Legend at 0x1b2128779e8>
```



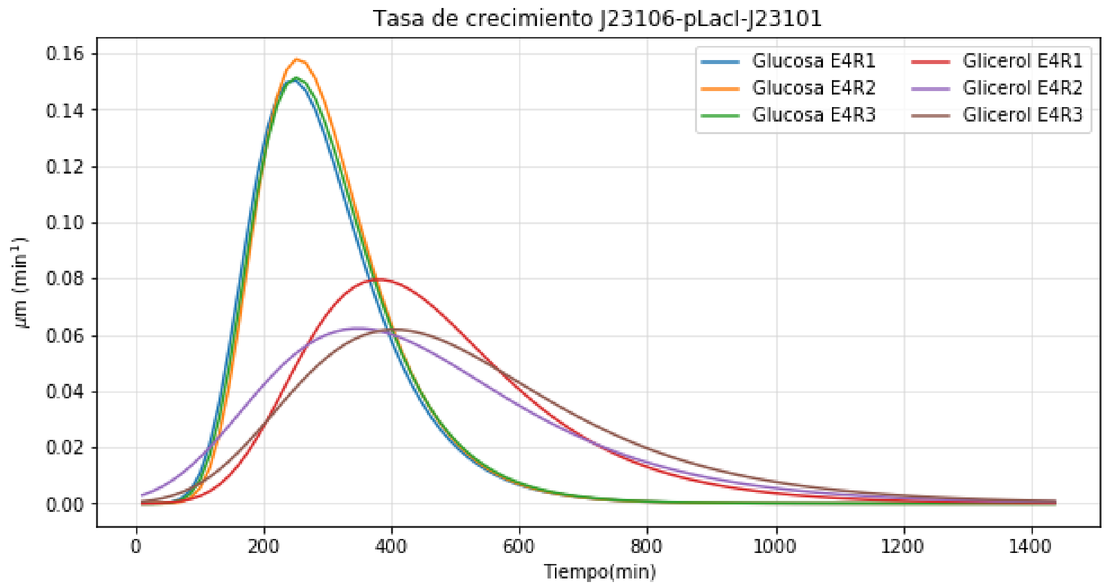

```
In [122]: #tasa de crecimiento
          ye24=((A24*np.exp(-np.exp((((um24*np.exp(1))/A24)*(l24-tt))+1))))
          #Con diff
          dy24=(np.diff(ye24))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          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[122]: <matplotlib.legend.Legend at 0x1b213912b70>
```



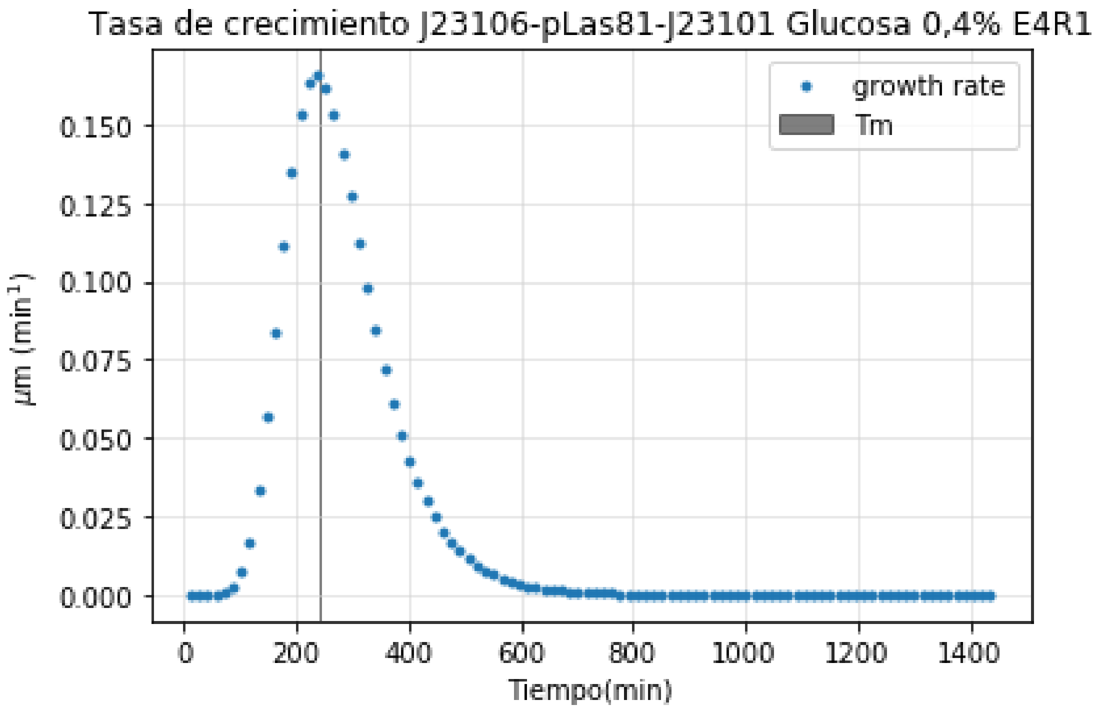

```
In [123]: #Tasas control réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLacI-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy22,label='growth rate E4R1')
          plt.plot(tt[:-1],dy23,label='growth rate E4R2')
          plt.plot(tt[:-1],dy24,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[123]: <matplotlib.legend.Legend at 0x1b2121f72e8>
```



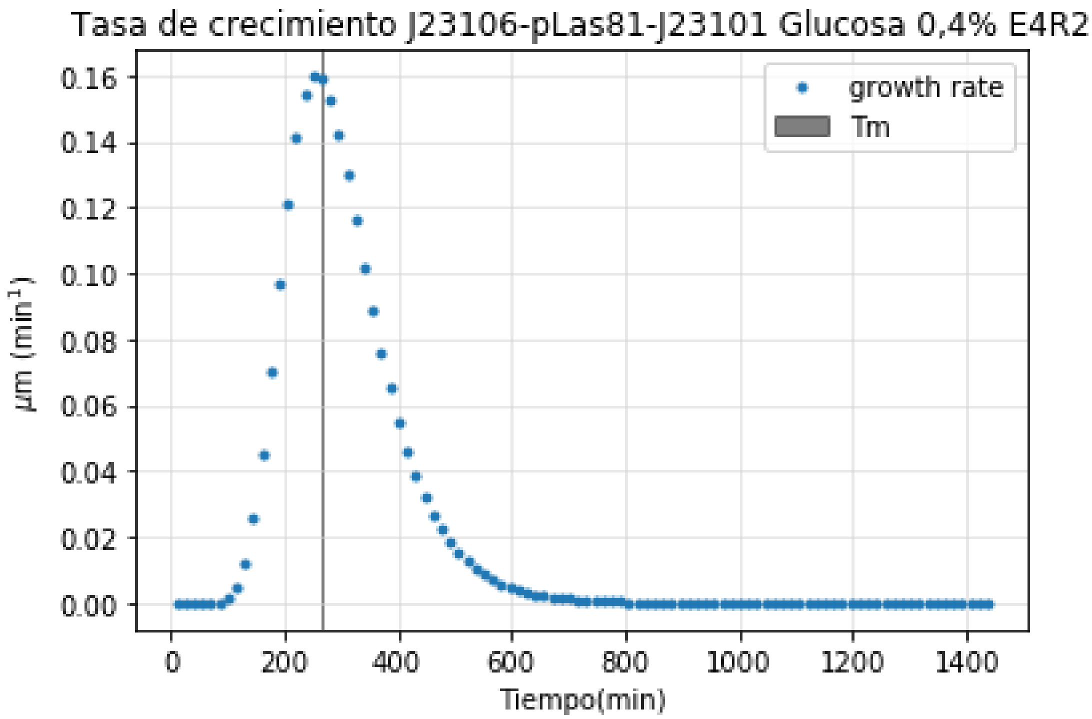

```
In [124]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLacI-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy19,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy20,label='Glucosa E4R2')
          plt.plot(tt[:-1],dy21,label='Glucosa E4R3')
          plt.plot(tt[:-1],dy22,label='Glicerol E4R1')
          plt.plot(tt[:-1],dy23,label='Glicerol E4R2')
          plt.plot(tt[:-1],dy24,label='Glicerol E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[124]: <matplotlib.legend.Legend at 0x1b20ba47ef0>
```




```
In [125]: #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 J23106-pLas81-J23101 Glucosa 0,4% E4R1')
          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[125]: <matplotlib.legend.Legend at 0x1b2139c8d30>
```

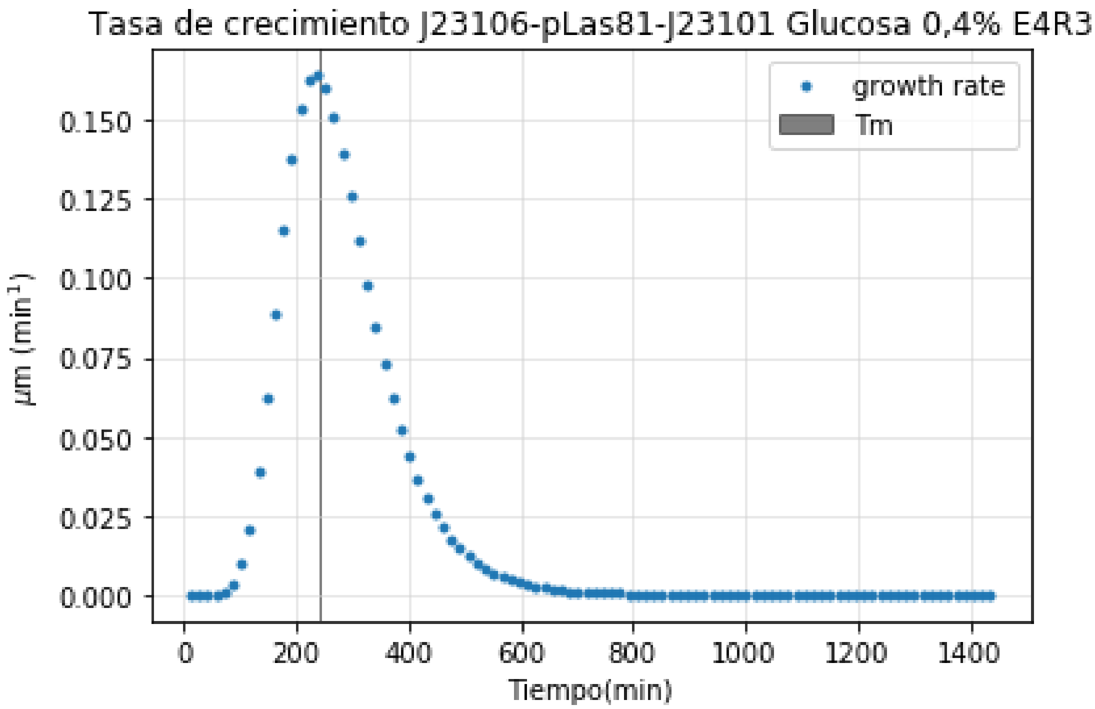



```
In [126]: #tasa de crecimiento
          ye26=((A26*np.exp(-np.exp((((um26*np.exp(1))/A26)*(l26-tt))+1))))
          #Con diff
          dy26=(np.diff(ye26))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4% E4R2')
          plt.xlabel('Tiempo(min)')
          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[126]: <matplotlib.legend.Legend at 0x1b202a0e860>
```

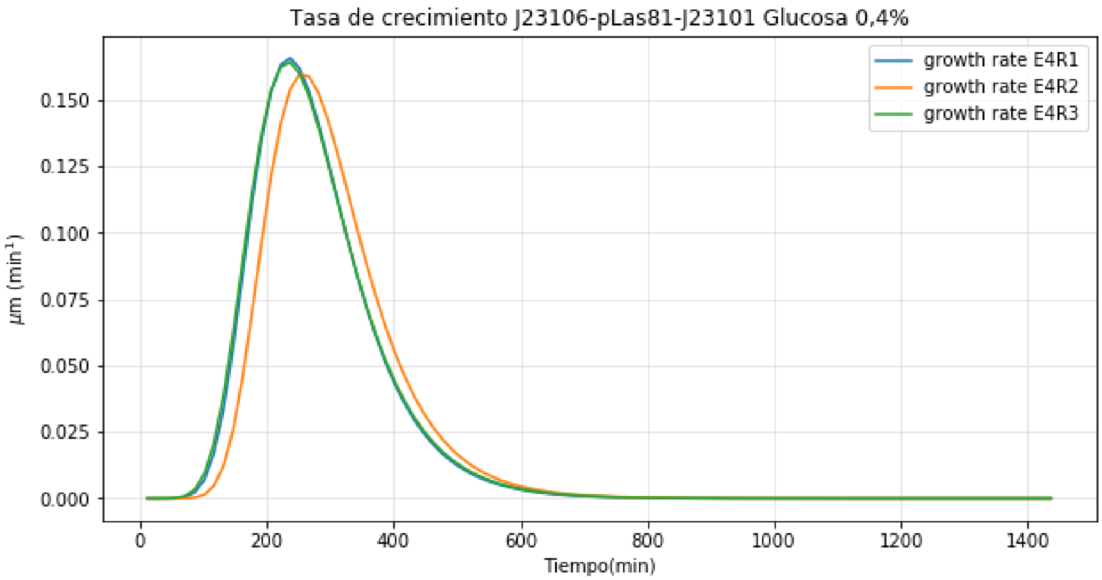


In [127]:
```
#tasa de crecimiento
ye27=((A27*np.exp(-np.exp((((um27*np.exp(1))/A27)*(l27-tt))+1))))
#Con diff
dy27=(np.diff(ye27))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4% E4R3')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm27,tm27, color='k', 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')
```

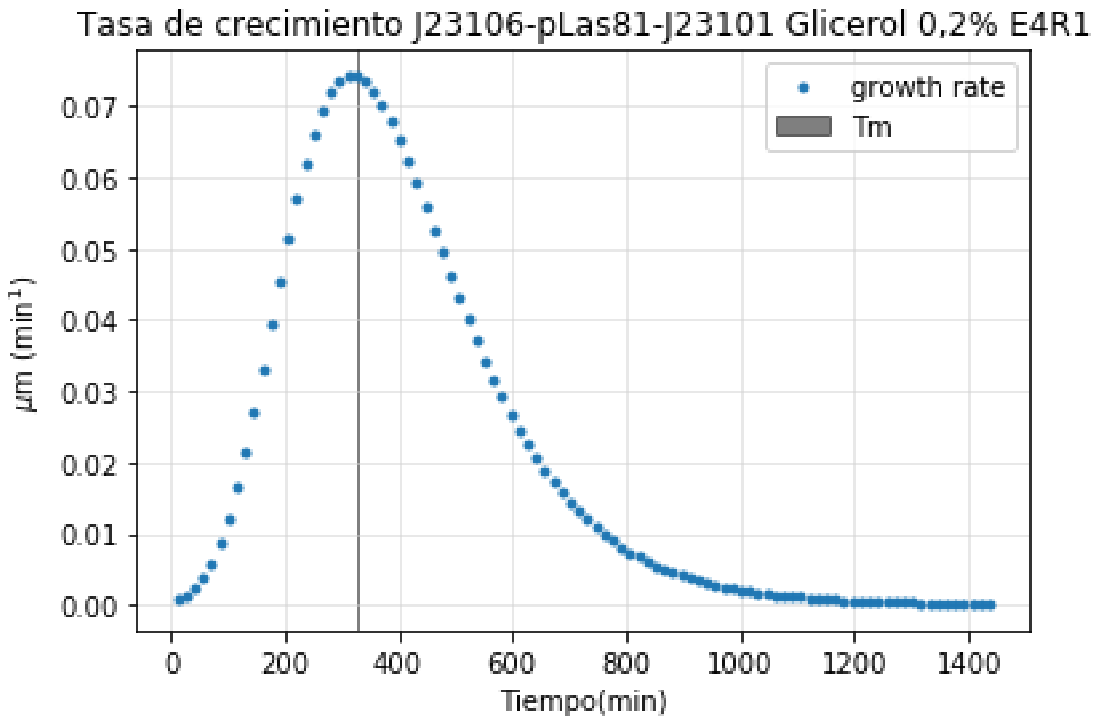
Out[127]: <matplotlib.legend.Legend at 0x1b202ab7e48>




```
In [128]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glucosa 0,4%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='growth rate E4R1')
          plt.plot(tt[:-1],dy26,label='growth rate E4R2')
          plt.plot(tt[:-1],dy27,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[128]: <matplotlib.legend.Legend at 0x1b202b90710>
```

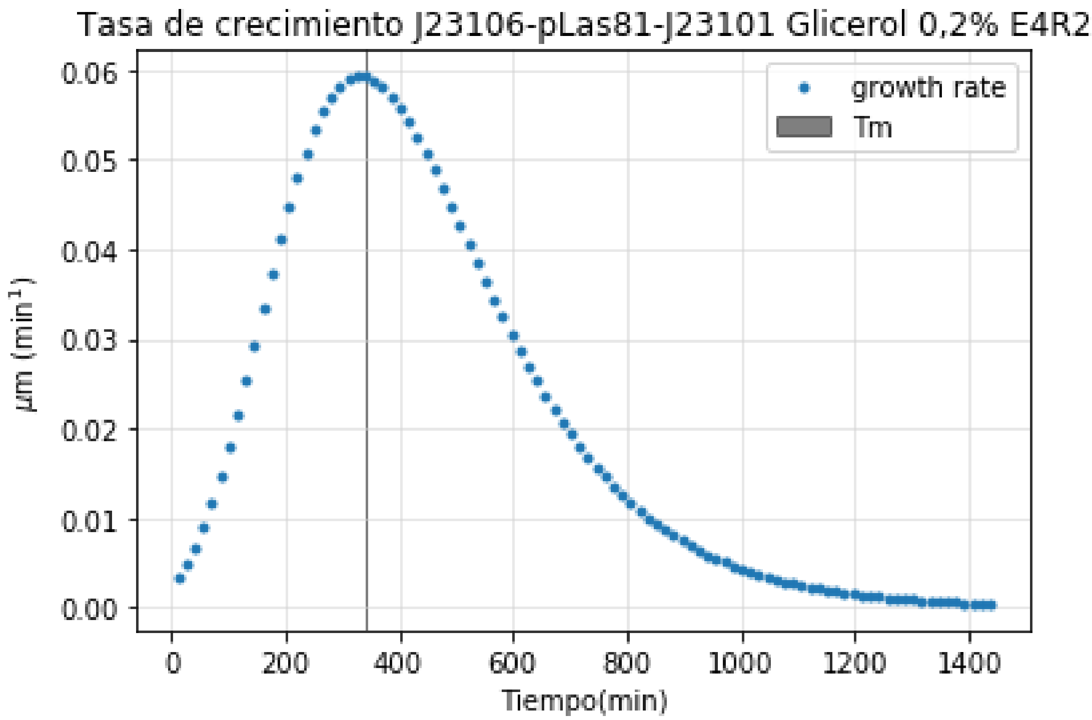



```
In [129]: #tasa de crecimiento
          ye28=((A28*np.exp(-np.exp((((um28*np.exp(1))/A28)*(l28-tt))+1))))
          #Con diff
          dy28=(np.diff(ye28))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R1')
          plt.xlabel('Tiempo(min)')
          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[129]: <matplotlib.legend.Legend at 0x1b213f50b70>
```

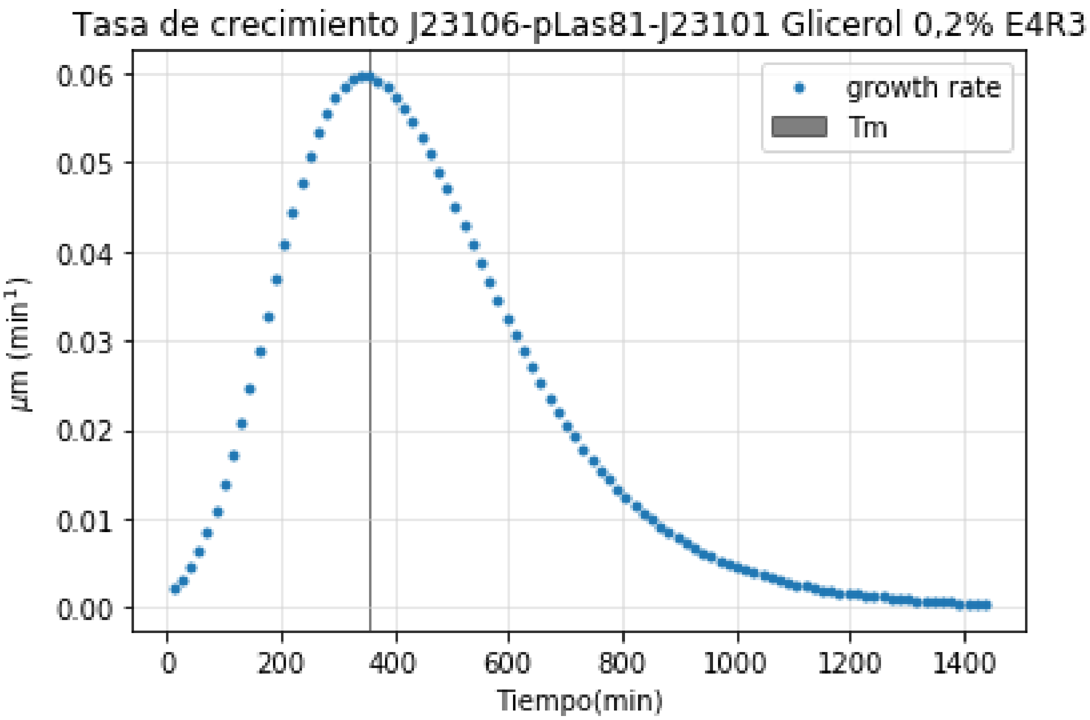


In [130]:
```
#tasa de crecimiento
ye29=((A29*np.exp(-np.exp((((um29*np.exp(1))/A29)*(l29-tt))+1))))
#Con diff
dy29=(np.diff(ye29))
plt.figure()
plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R2')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.axvspan(tm29,tm29, color='k', 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')
```

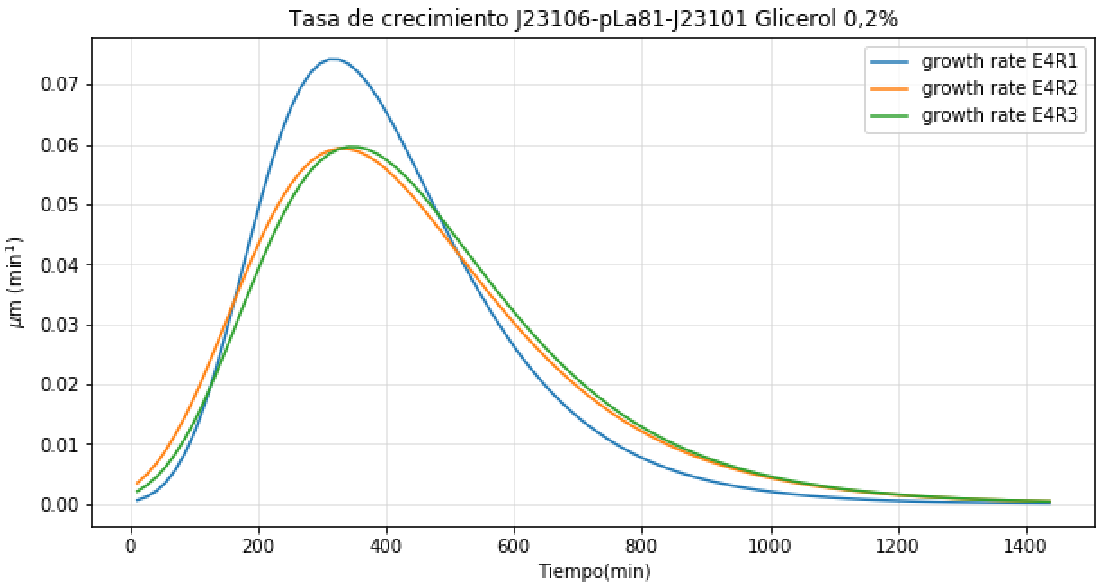
Out[130]:
```
<matplotlib.legend.Legend at 0x1b21402ca90>
```



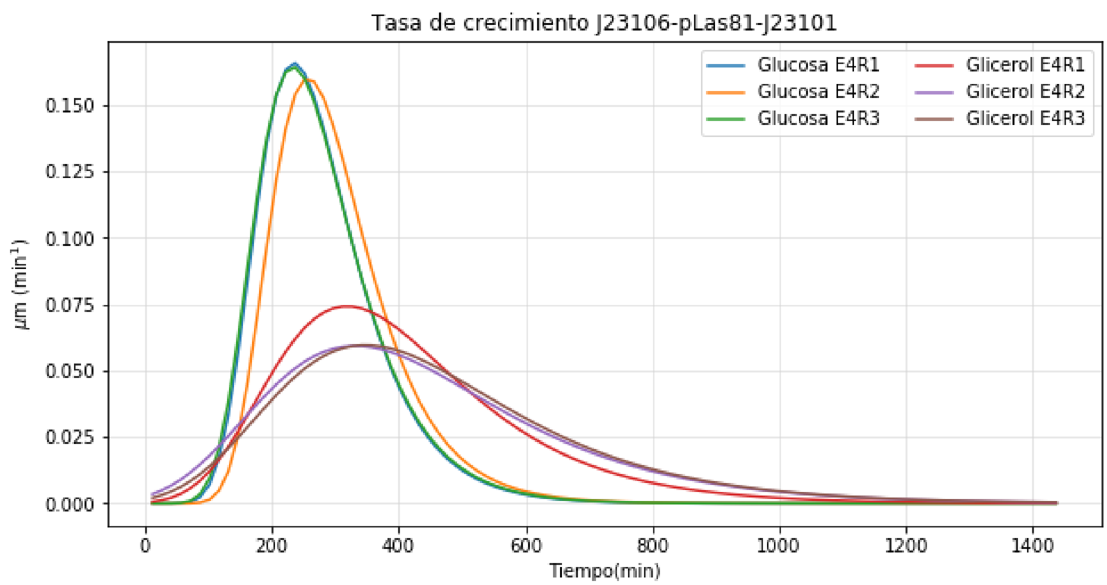

```
In [131]: #tasa de crecimiento
          ye30=((A30*np.exp(-np.exp((((um30*np.exp(1))/A30)*(l30-tt))+1))))
          #Con diff
          dy30=(np.diff(ye30))
          plt.figure()
          plt.title('Tasa de crecimiento J23106-pLas81-J23101 Glicerol 0,2% E4R3')
          plt.xlabel('Tiempo(min)')
          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[131]: <matplotlib.legend.Legend at 0x1b2140fa940>
```



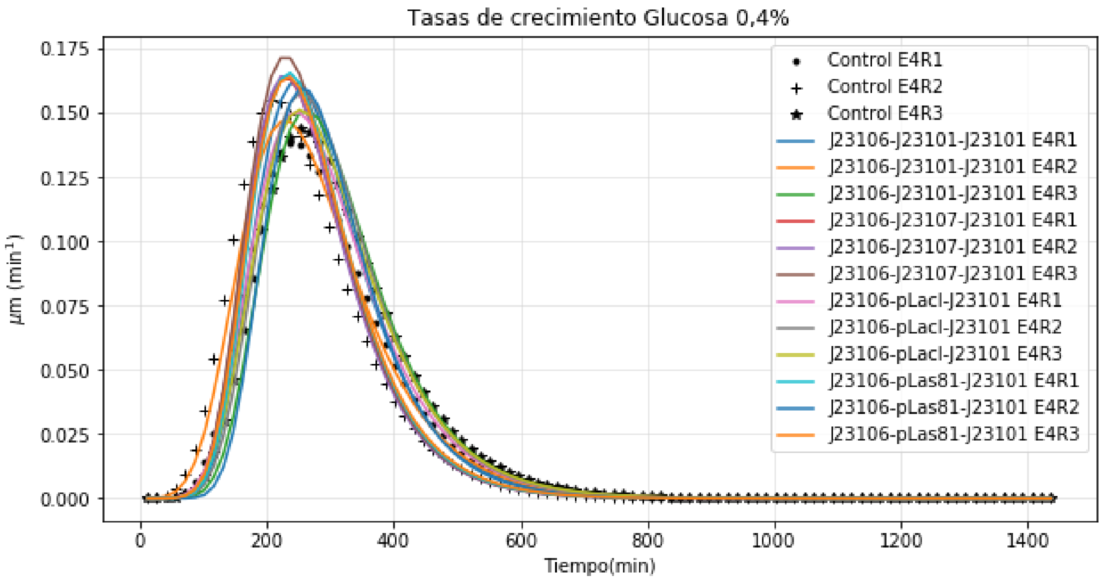

```
In [132]: #Tasas plux76-plac-J23101 réplicas glucosa
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLa81-J23101 Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy28,label='growth rate E4R1')
          plt.plot(tt[:-1],dy29,label='growth rate E4R2')
          plt.plot(tt[:-1],dy30,label='growth rate E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[132]: <matplotlib.legend.Legend at 0x1b2141c6fd0>
```



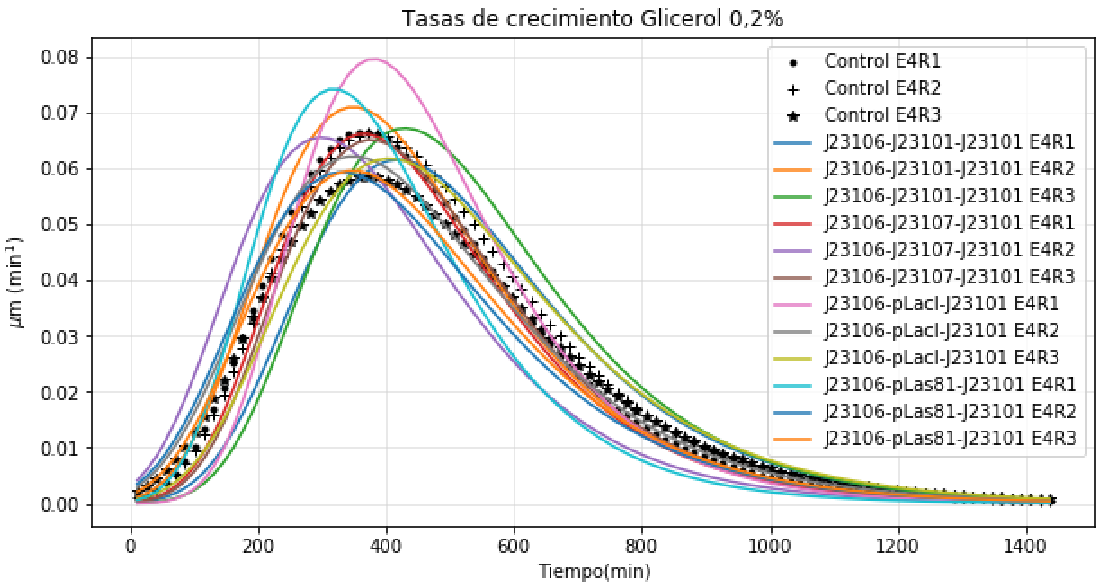

```
In [133]: #Tasas control réplicas
          plt.figure(figsize=(10,5))
          plt.title('Tasa de crecimiento J23106-pLas81-J23101')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy25,label='Glucosa E4R1')
          plt.plot(tt[:-1],dy26,label='Glucosa E4R2')
          plt.plot(tt[:-1],dy27,label='Glucosa E4R3')
          plt.plot(tt[:-1],dy28,label='Glicerol E4R1')
          plt.plot(tt[:-1],dy29,label='Glicerol E4R2')
          plt.plot(tt[:-1],dy30,label='Glicerol E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right',ncol=2)

Out[133]: <matplotlib.legend.Legend at 0x1b2142bc6a0>
```




```
In [134]: #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 E4R1')
          plt.plot(tt[:-1],dy2,'k+',label='Control E4R2')
          plt.plot(tt[:-1],dy3,'k*',label='Control E4R3')
          plt.plot(tt[:-1],dy7,label='J23106-J23101-J23101 E4R1')
          plt.plot(tt[:-1],dy8,label='J23106-J23101-J23101 E4R2')
          plt.plot(tt[:-1],dy9,label='J23106-J23101-J23101 E4R3')
          plt.plot(tt[:-1],dy13,label='J23106-J23107-J23101 E4R1')
          plt.plot(tt[:-1],dy14,label='J23106-J23107-J23101 E4R2')
          plt.plot(tt[:-1],dy15,label='J23106-J23107-J23101 E4R3')
          plt.plot(tt[:-1],dy19,label='J23106-pLacI-J23101 E4R1')
          plt.plot(tt[:-1],dy20,label='J23106-pLacI-J23101 E4R2')
          plt.plot(tt[:-1],dy21,label='J23106-pLacI-J23101 E4R3')
          plt.plot(tt[:-1],dy25,label='J23106-pLas81-J23101 E4R1')
          plt.plot(tt[:-1],dy26,label='J23106-pLas81-J23101 E4R2')
          plt.plot(tt[:-1],dy27,label='J23106-pLas81-J23101 E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[134]: <matplotlib.legend.Legend at 0x1b2143bedd8>
```



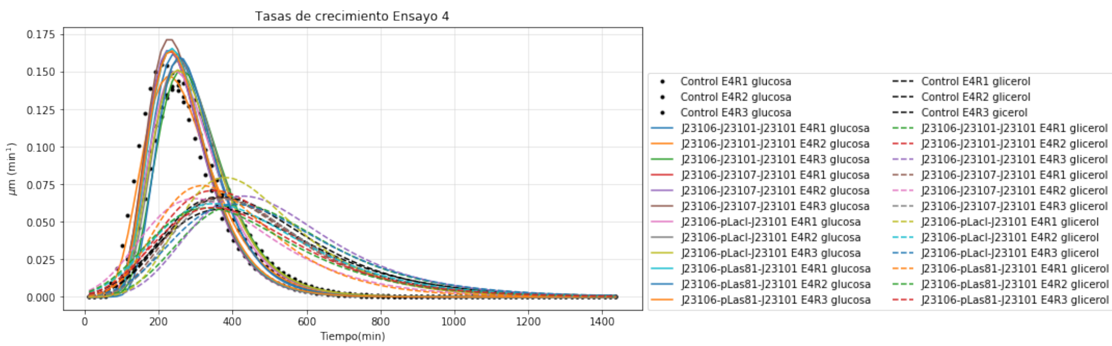

```
In [135]: #Tasas réplicas glicerol
          plt.figure(figsize=(10,5))
          plt.title('Tasas de crecimiento Glicerol 0,2%')
          plt.xlabel('Tiempo(min)')
          plt.ylabel(r'$\mu$m (min$^1$)')
          plt.plot(tt[:-1],dy4,'k.',label='Control E4R1')
          plt.plot(tt[:-1],dy5,'k+',label='Control E4R2')
          plt.plot(tt[:-1],dy6,'k*',label='Control E4R3')
          plt.plot(tt[:-1],dy10,label='J23106-J23101-J23101 E4R1')
          plt.plot(tt[:-1],dy11,label='J23106-J23101-J23101 E4R2')
          plt.plot(tt[:-1],dy12,label='J23106-J23101-J23101 E4R3')
          plt.plot(tt[:-1],dy16,label='J23106-J23107-J23101 E4R1')
          plt.plot(tt[:-1],dy17,label='J23106-J23107-J23101 E4R2')
          plt.plot(tt[:-1],dy18,label='J23106-J23107-J23101 E4R3')
          plt.plot(tt[:-1],dy22,label='J23106-pLacI-J23101 E4R1')
          plt.plot(tt[:-1],dy23,label='J23106-pLacI-J23101 E4R2')
          plt.plot(tt[:-1],dy24,label='J23106-pLacI-J23101 E4R3')
          plt.plot(tt[:-1],dy28,label='J23106-pLas81-J23101 E4R1')
          plt.plot(tt[:-1],dy29,label='J23106-pLas81-J23101 E4R2')
          plt.plot(tt[:-1],dy30,label='J23106-pLas81-J23101 E4R3')
          plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
          plt.legend(loc='upper right')

Out[135]: <matplotlib.legend.Legend at 0x1b214530d68>
```



In [136]:
```
#Tasas réplicas glucosa
plt.figure(figsize=(10,5))
plt.title('Tasas de crecimiento Ensayo 4')
plt.xlabel('Tiempo(min)')
plt.ylabel(r'$\mu$m (min$^1$)')
plt.plot(tt[:-1],dy1,'k.',label='Control E4R1 glucosa')
plt.plot(tt[:-1],dy2,'k.',label='Control E4R2 glucosa')
plt.plot(tt[:-1],dy3,'k.',label='Control E4R3 glucosa')
plt.plot(tt[:-1],dy7,label='J23106-J23101-J23101 E4R1 glucosa')
plt.plot(tt[:-1],dy8,label='J23106-J23101-J23101 E4R2 glucosa')
plt.plot(tt[:-1],dy9,label='J23106-J23101-J23101 E4R3 glucosa')
plt.plot(tt[:-1],dy13,label='J23106-J23107-J23101 E4R1 glucosa')
plt.plot(tt[:-1],dy14,label='J23106-J23107-J23101 E4R2 glucosa')
plt.plot(tt[:-1],dy15,label='J23106-J23107-J23101 E4R3 glucosa')
plt.plot(tt[:-1],dy19,label='J23106-pLacI-J23101 E4R1 glucosa')
plt.plot(tt[:-1],dy20,label='J23106-pLacI-J23101 E4R2 glucosa')
plt.plot(tt[:-1],dy21,label='J23106-pLacI-J23101 E4R3 glucosa')
plt.plot(tt[:-1],dy25,label='J23106-pLas81-J23101 E4R1 glucosa')
plt.plot(tt[:-1],dy26,label='J23106-pLas81-J23101 E4R2 glucosa')
plt.plot(tt[:-1],dy27,label='J23106-pLas81-J23101 E4R3 glucosa')
plt.plot(tt[:-1],dy4,'k--',label='Control E4R1 glicerol')
plt.plot(tt[:-1],dy5,'k--',label='Control E4R2 glicerol')
plt.plot(tt[:-1],dy6,'k--',label='Control E4R3 gicerol')
plt.plot(tt[:-1],dy10,'--',label='J23106-J23101-J23101 E4R1 glicerol')
plt.plot(tt[:-1],dy11,'--',label='J23106-J23101-J23101 E4R2 glicerol')
plt.plot(tt[:-1],dy12,'--',label='J23106-J23101-J23101 E4R3 glicerol')
plt.plot(tt[:-1],dy16,'--',label='J23106-J23107-J23101 E4R1 glicerol')
plt.plot(tt[:-1],dy17,'--',label='J23106-J23107-J23101 E4R2 glicerol')
```

```python
plt.plot(tt[:-1],dy18,'--',label='J23106-J23107-J23101 E4R3 glicerol')
plt.plot(tt[:-1],dy22,'--',label='J23106-pLacI-J23101 E4R1 glicerol')
plt.plot(tt[:-1],dy23,'--',label='J23106-pLacI-J23101 E4R2 glicerol')
plt.plot(tt[:-1],dy24,'--',label='J23106-pLacI-J23101 E4R3 glicerol')
plt.plot(tt[:-1],dy28,'--',label='J23106-pLas81-J23101 E4R1 glicerol')
plt.plot(tt[:-1],dy29,'--',label='J23106-pLas81-J23101 E4R2 glicerol')
plt.plot(tt[:-1],dy30,'--',label='J23106-pLas81-J23101 E4R3 glicerol')
plt.grid(color='lightgray', linestyle='-', linewidth=0.5,zorder=0)
plt.legend(loc=(1.01,0.0),ncol=2)
```

Out[136]: <matplotlib.legend.Legend at 0x1b214643ba8>



In [ ]:

In [ ]: