Funciones de transferencia: estabilidad

Determinar los ceros y los polos de las siguientes funciones de transferencia y graficarlos en el plano complejo S, decir si el sistema es Estable o Inestable.

In [1]:
```
# Dependencias a utilizar.
import control as ctrl
import matplotlib.pyplot as plt
from control.pzmap import pzmap
```

1) $\frac{(s+1)(s-1)}{s(s+2)(s+10)}$

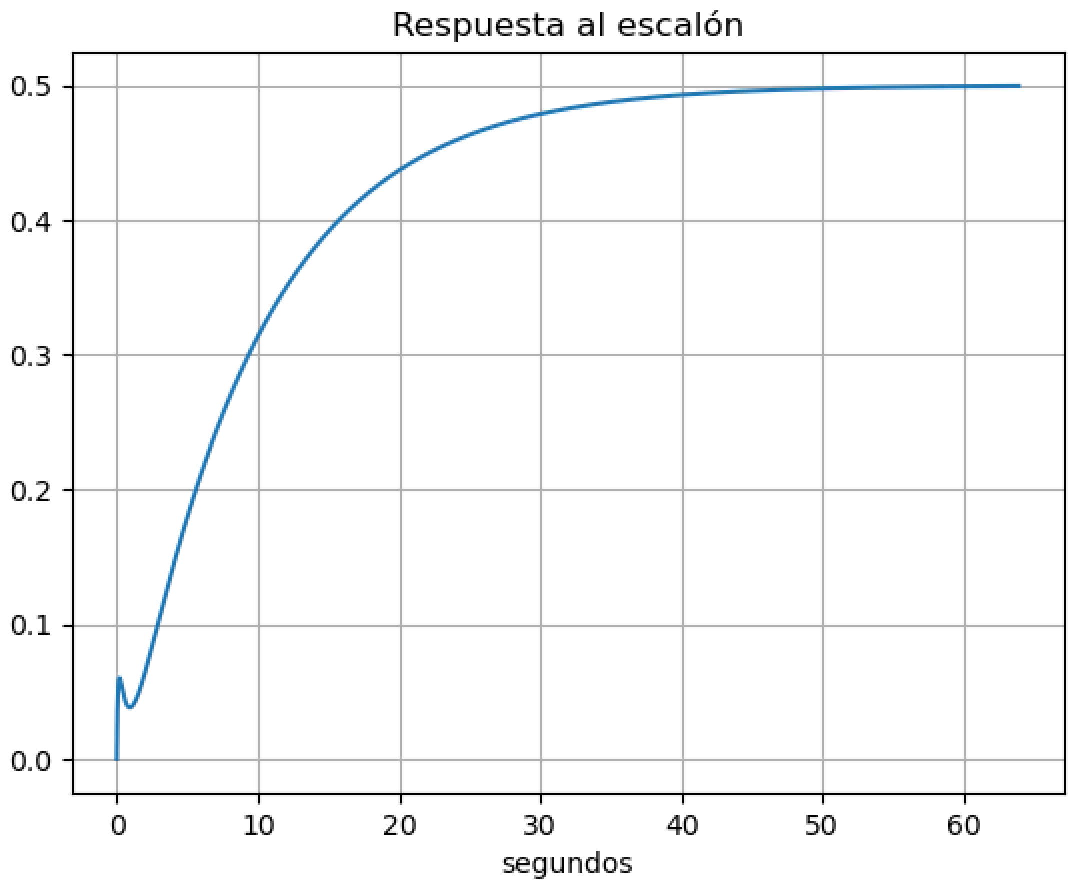
$(s+1)(s-1) = s^2 - 1$
$s(s+2)(s+10) = s^3 + 12s^2 + 20s$

In [2]:
```
num = [1,0,1]
den = [1,12,20,0]
sys = ctrl.tf(num, den)
print(sys)

sys_cl = ctrl.feedback(sys, 2)

#Respuesta al escalón
t, y = ctrl.step_response(sys_cl)
```

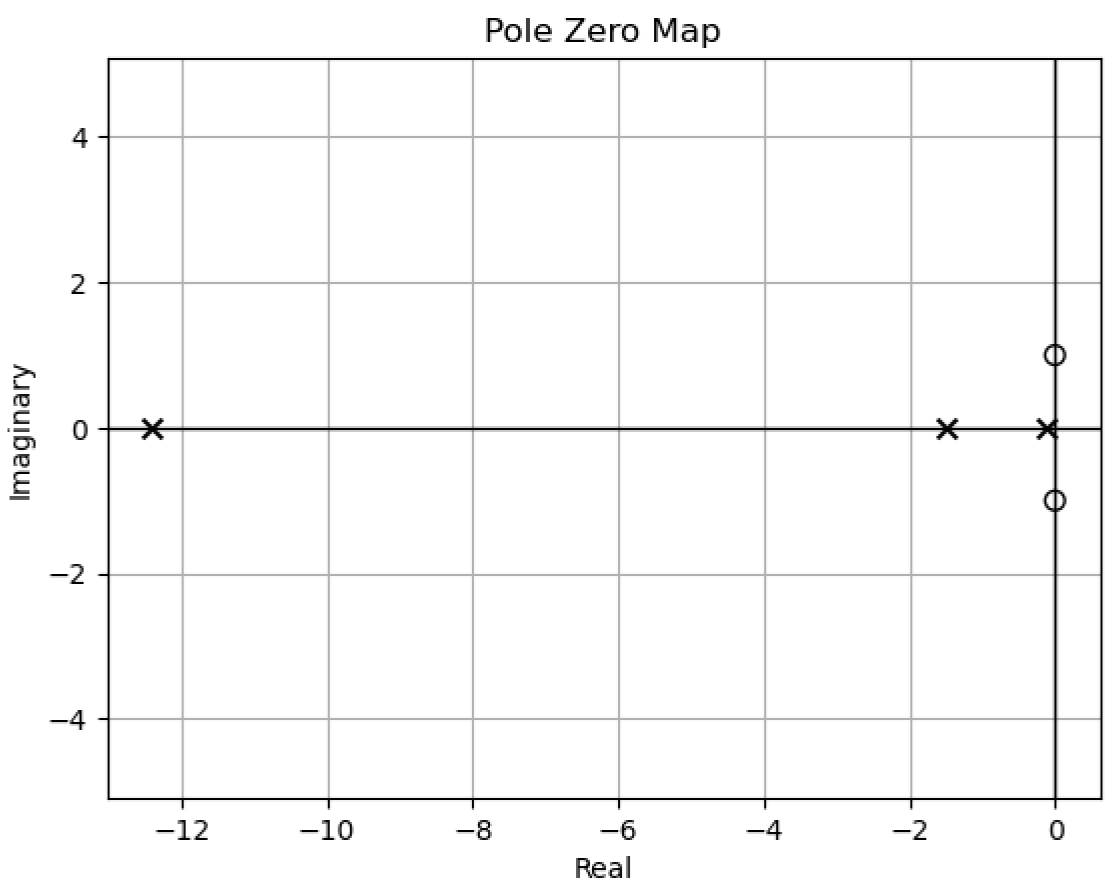
```
      s^2 + 1
-------------------
s^3 + 12 s^2 + 20 s
```

In [3]:
```
#Graficas
plt.plot(t, y)
plt.grid(True)
plt.title(u"Respuesta al escalón")
plt.xlabel("segundos")
```

Out[3]:
```
Text(0.5, 0, 'segundos')
```



In [4]:
```python
polos, zeros = pzmap(sys_cl), plt.grid(True)
```



El sistema es **estable**

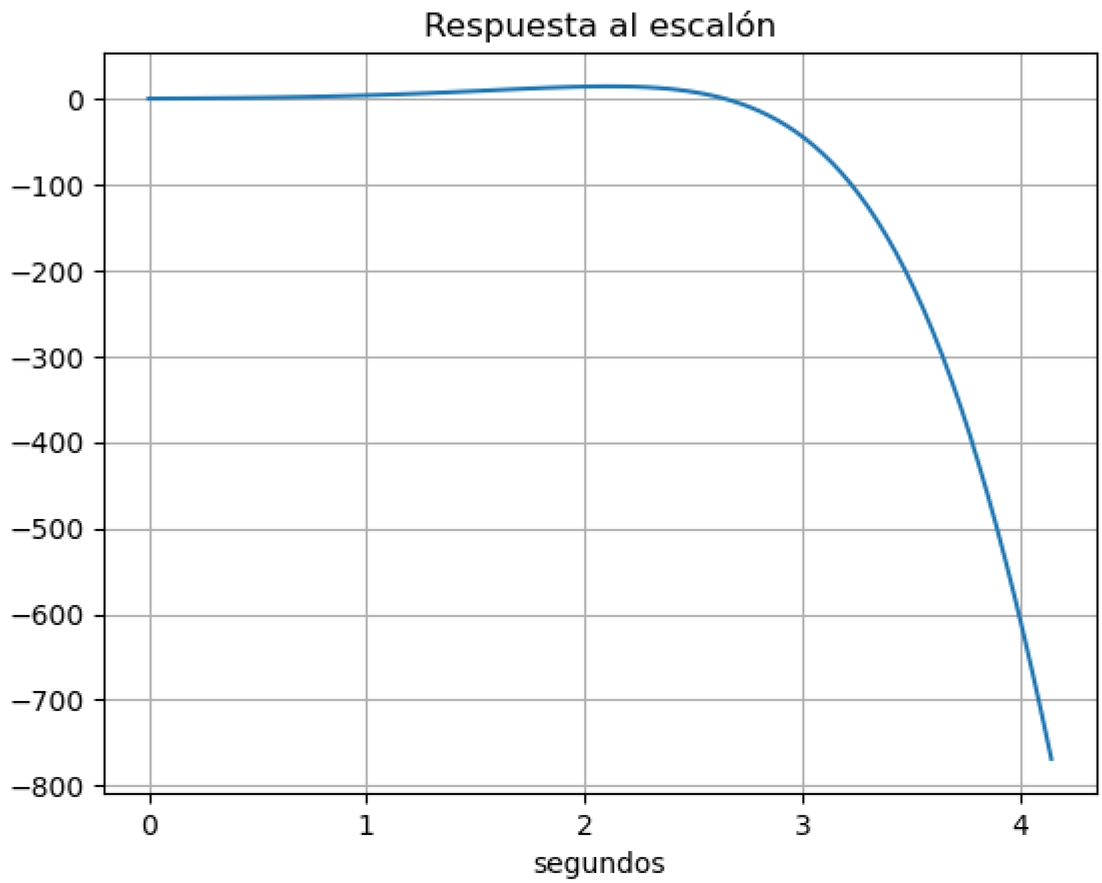
2) $\frac{s^2+2}{s^2-10s+8}$

In [5]:
```
num = [1,0,2]
den = [1,-10,8]
sys = ctrl.tf(num, den)
print(sys)

sys_cl = ctrl.feedback(sys, 2)

#Respuesta al escalón
t, y = ctrl.step_response(sys_cl)
```

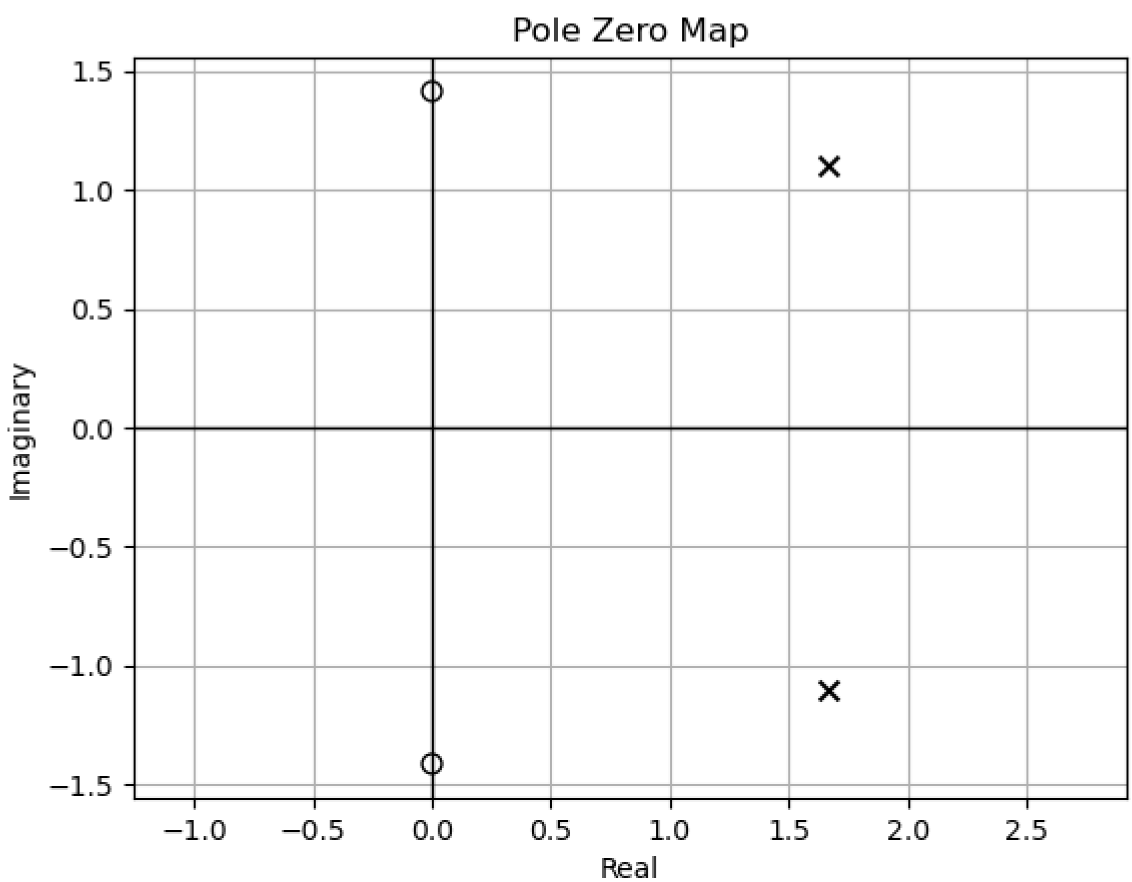
```
   s^2 + 2
--------------
s^2 - 10 s + 8
```

In [6]:
```
#Graficas
plt.plot(t, y)
plt.grid(True)
plt.title(u"Respuesta al escalón")
plt.xlabel("segundos")
```

Out[6]:
```
Text(0.5, 0, 'segundos')
```



In [7]:
```
polos, zeros = pzmap(sys_cl), plt.grid(True)
```



El sistema es **inestable**

3) $\frac{1}{(s+2)(s^2+10s+7)}$

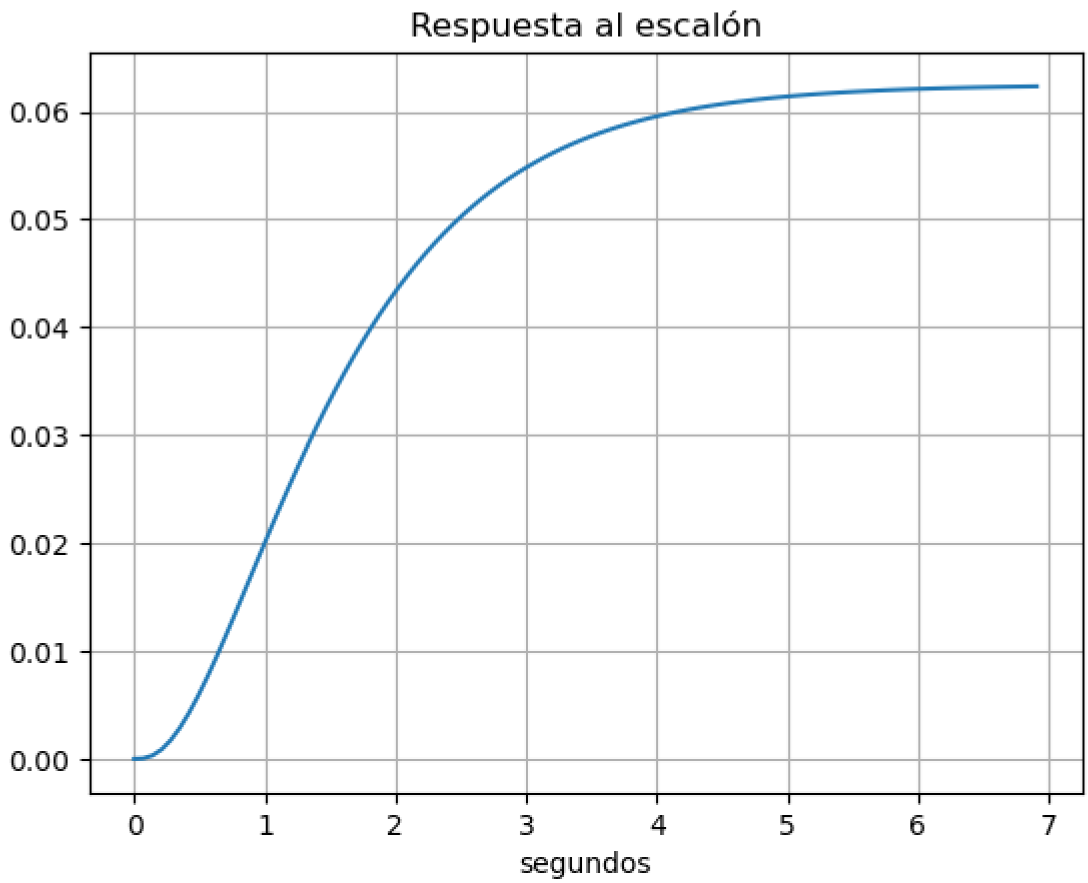
$$(s+2)(s^2+10s+7) = s^3 + 12s^2 + 27s + 14$$

In [8]:
```python
num = [1]
den = [1,12,27,14]
sys = ctrl.tf(num, den)
print(sys)

sys_cl = ctrl.feedback(sys, 2)

#Respuesta al escalón
t, y = ctrl.step_response(sys_cl)
```

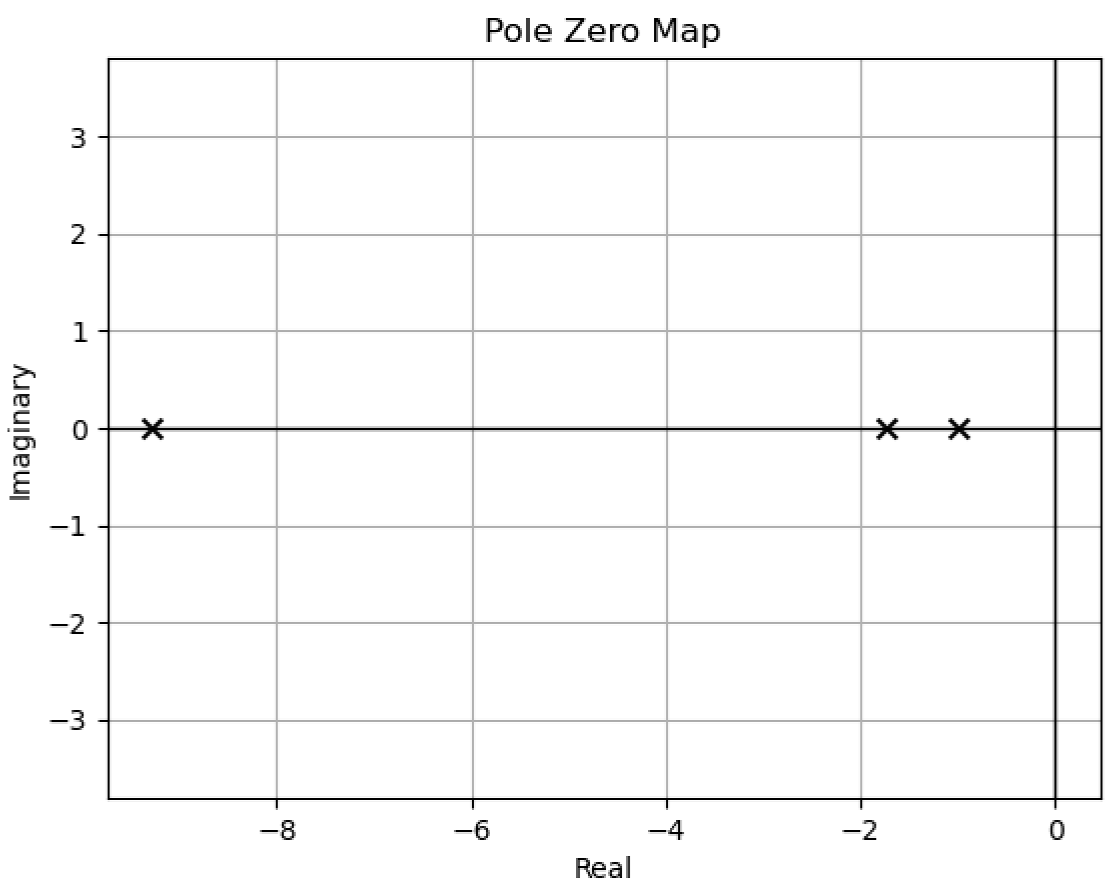
```
           1
------------------------
s^3 + 12 s^2 + 27 s + 14
```

In [9]:
```python
#Graficas
plt.plot(t, y)
plt.grid(True)
plt.title(u"Respuesta al escalón")
plt.xlabel("segundos")
```

Out[9]:
```
Text(0.5, 0, 'segundos')
```




```
In [10]: polos, zeros = pzmap(sys_cl), plt.grid(True)
```



El sistema es **estable**

4) $\frac{s+1}{(s+1)(10s+4)}$

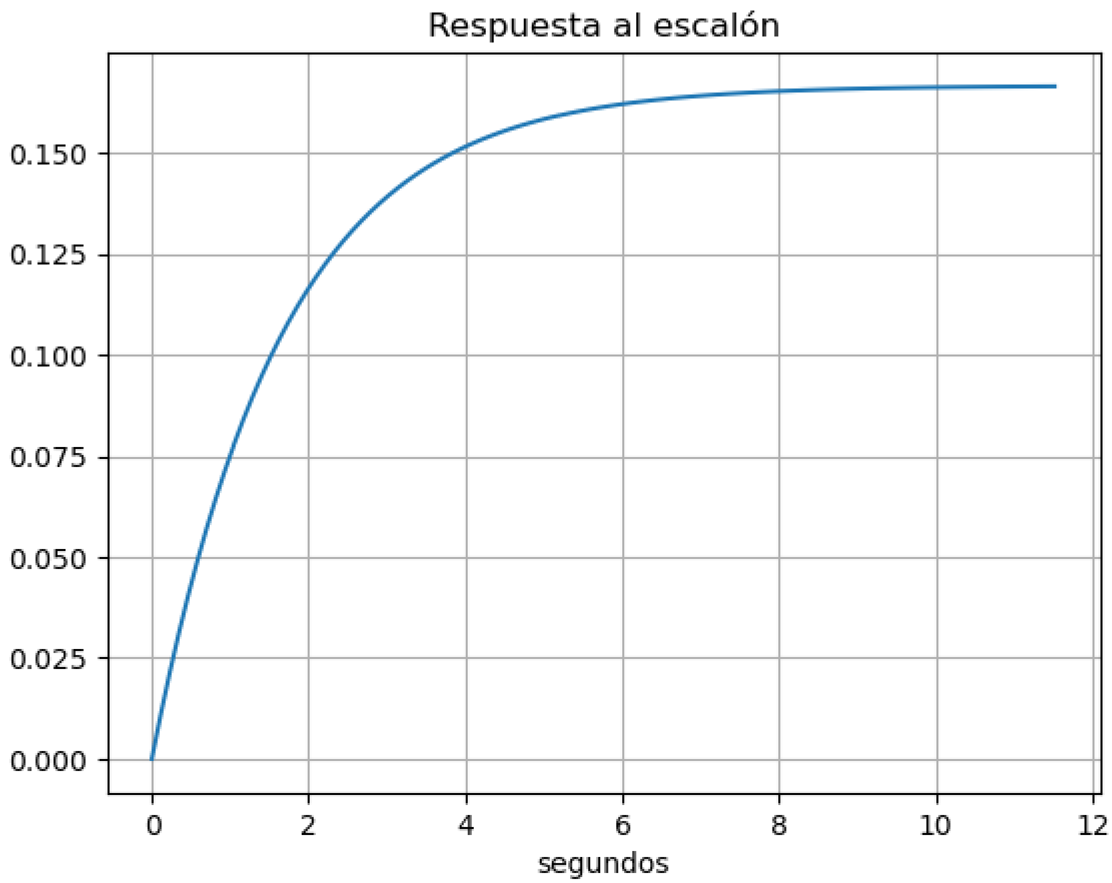
$$(s+1)(10s+4) = 10s^2 + 14s + 4$$

In [11]:
```
num = [1,1]
den = [10,14,4]
sys = ctrl.tf(num, den)
print(sys)

sys_cl = ctrl.feedback(sys, 2)

#Respuesta al escalón
t, y = ctrl.step_response(sys_cl)
```

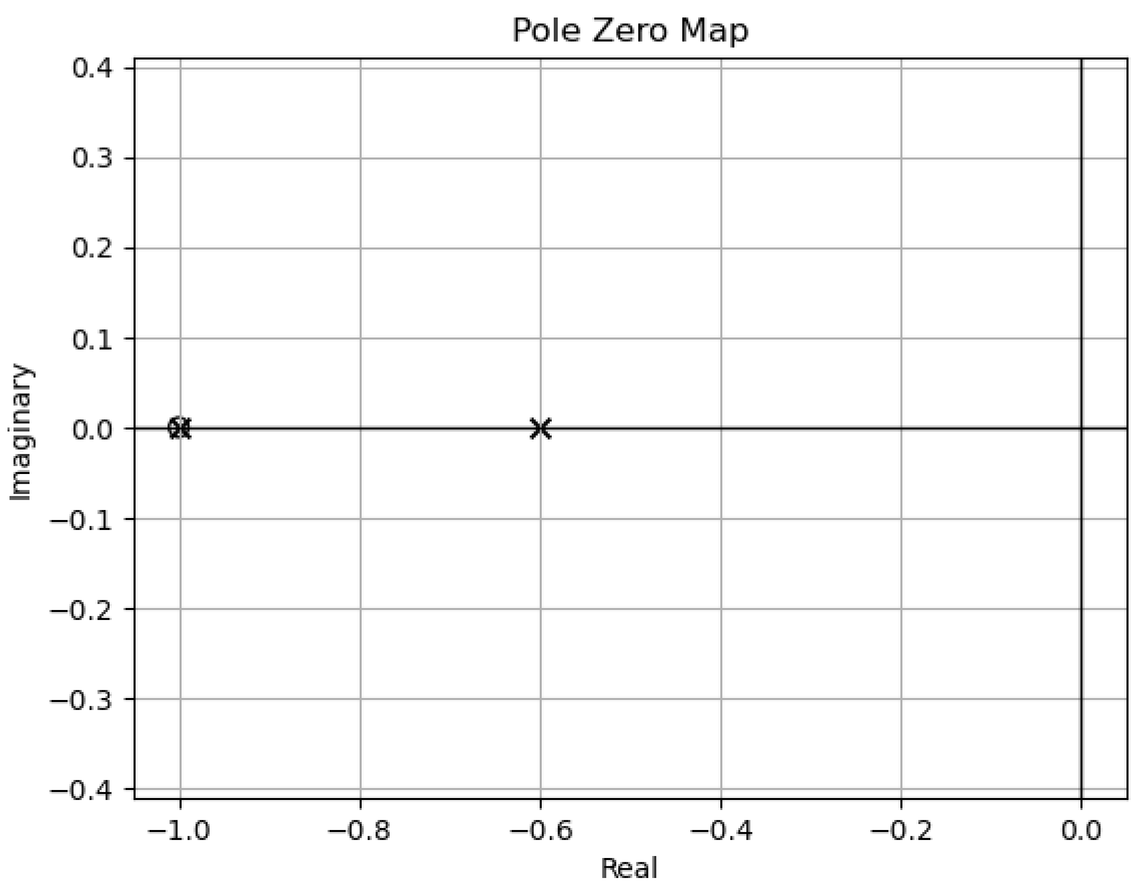
```
      s + 1
-----------------
10 s^2 + 14 s + 4
```

In [12]:
```
#Graficas
plt.plot(t, y)
plt.grid(True)
plt.title(u"Respuesta al escalón")
plt.xlabel("segundos")
```

Out[12]:
```
Text(0.5, 0, 'segundos')
```



In [13]:
```
polos, zeros = pzmap(sys_cl), plt.grid(True)
```



El sistema es **estable**

5) $\frac{1}{(s+2)(s^2+1)^2}$

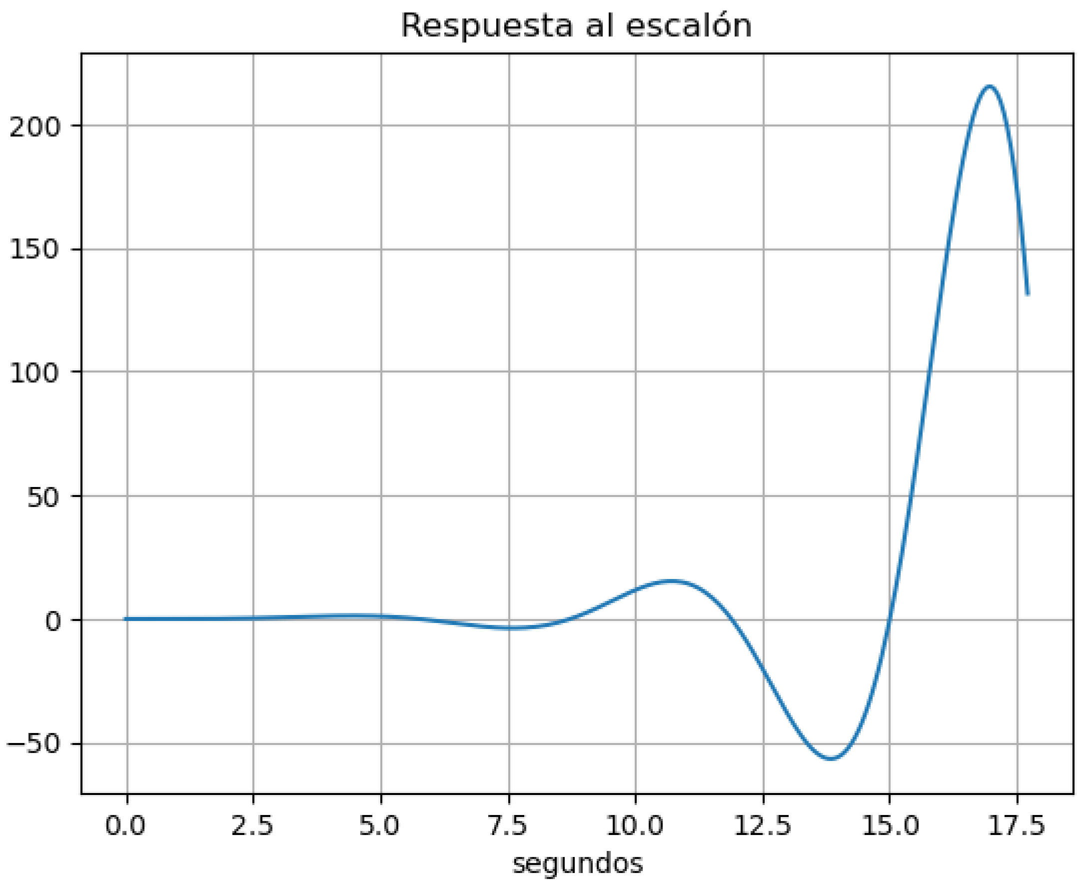
$(s+2)(s^2+1)^2 = s^5 + 2s^4 + 2s^3 + 4s^2 + s + 2$

In [14]:
```
num = [1]
den = [1,2,2,4,1,2]
sys = ctrl.tf(num, den)
print(sys)

sys_cl = ctrl.feedback(sys, 2)

#Respuesta al escalón
t, y = ctrl.step_response(sys_cl)
```

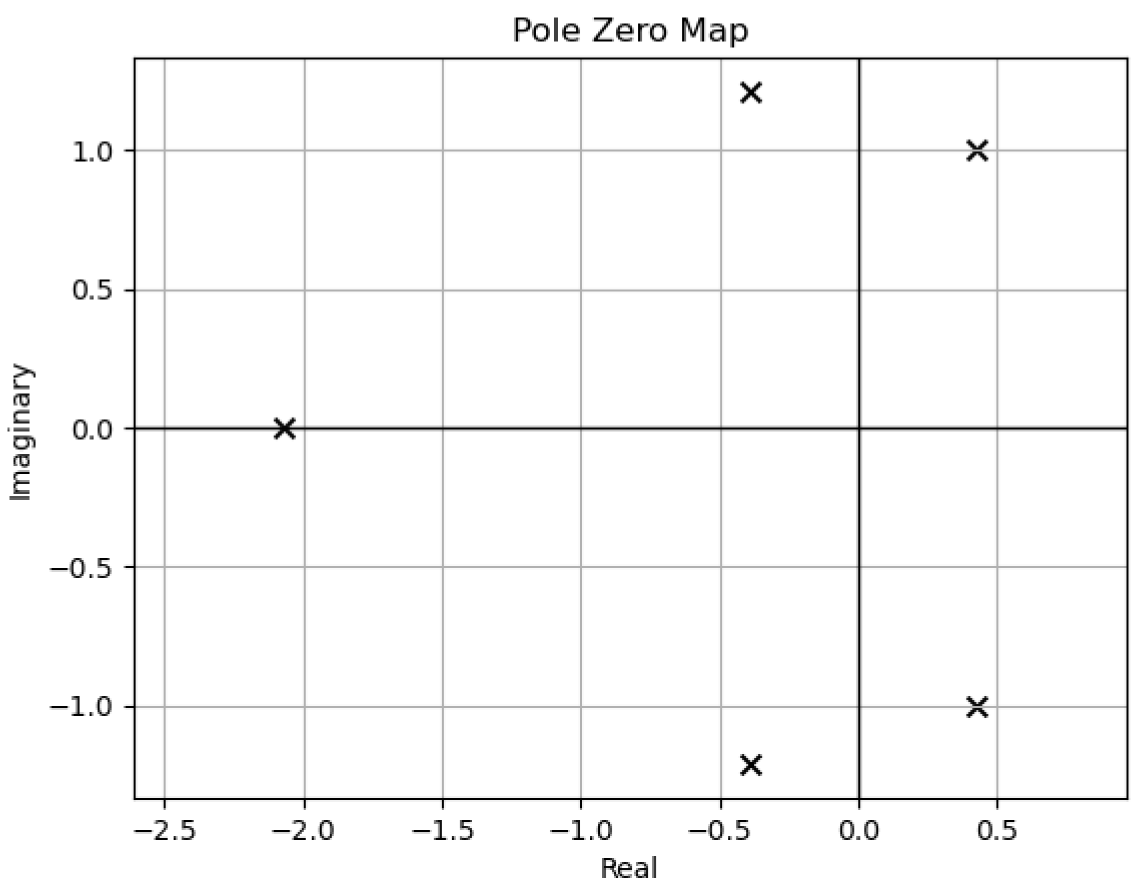
```
                 1
-----------------------------------
s^5 + 2 s^4 + 2 s^3 + 4 s^2 + s + 2
```

In [15]:
```
#Graficas
plt.plot(t, y)
plt.grid(True)
plt.title(u"Respuesta al escalón")
plt.xlabel("segundos")
```

Out[15]:
```
Text(0.5, 0, 'segundos')
```




```
In [16]: polos, zeros = pzmap(sys_cl), plt.grid(True)
```



El sistema es **inestable**