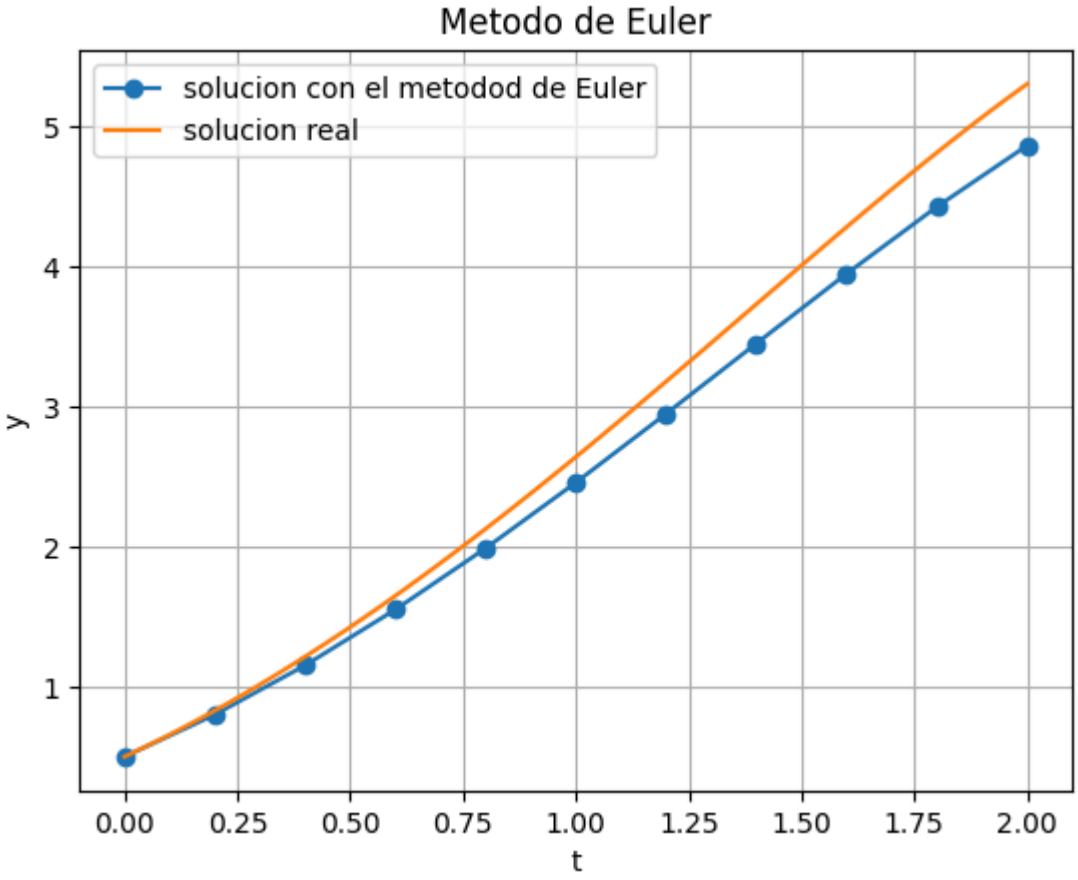
Taller Metodo de Euler

Nombre: Alexis Bautista
Fecha de entrega: 05 de febrero de 2025
Paralelo: GR1CC
Enlace de GitHub: https://github.com/alexis-bautista/Taller06-MN

Con N = 10

In [66]:
```
from src import ODE_euler

a = 0
b = 2
f = lambda t, y: y - t**2 + 1
y_t0 = 0.5
N = 10

y_values, t_values, h = ODE_euler(a=a, b=b, f=f, y_t0=y_t0, N=N)
```

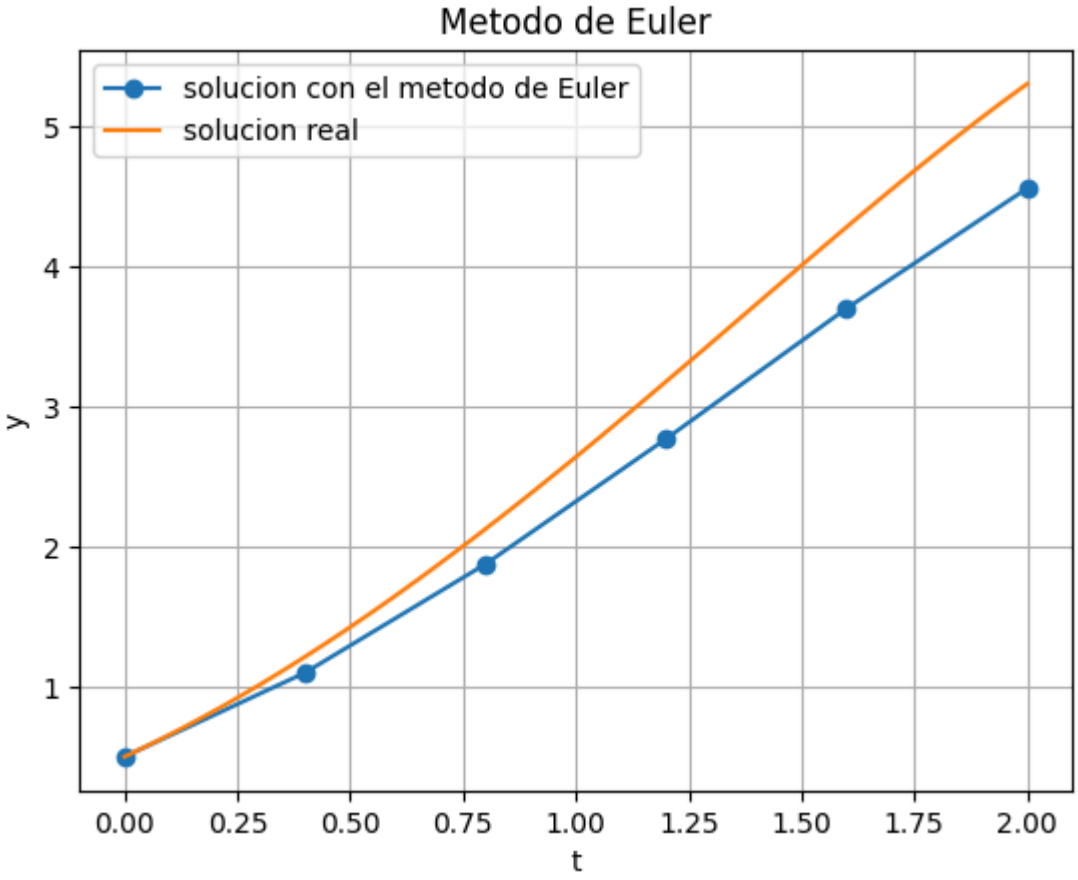
In [67]:
```
import matplotlib.pyplot as plt
import numpy as np

def y_real(t):
    return (t + 1)**2 - 0.5 * np.exp(t)

t_continuous = np.linspace(a, b, 100)
y_continuous = y_real(t_continuous)

# Graficade aproximacion y real
plt.plot(t_values, y_values, 'o-', label='solucion con el metodod de Euler')
plt.plot(t_continuous, y_continuous, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Con N = 5

In [68]:
```
a = 0
b = 2
f = lambda t, y: y - t**2 + 1
y_t0 = 0.5
N = 5

y_values_n5, t_values_n5, h_n5 = ODE_euler(a=a, b=b, f=f, y_t0=y_t0, N=N)
```

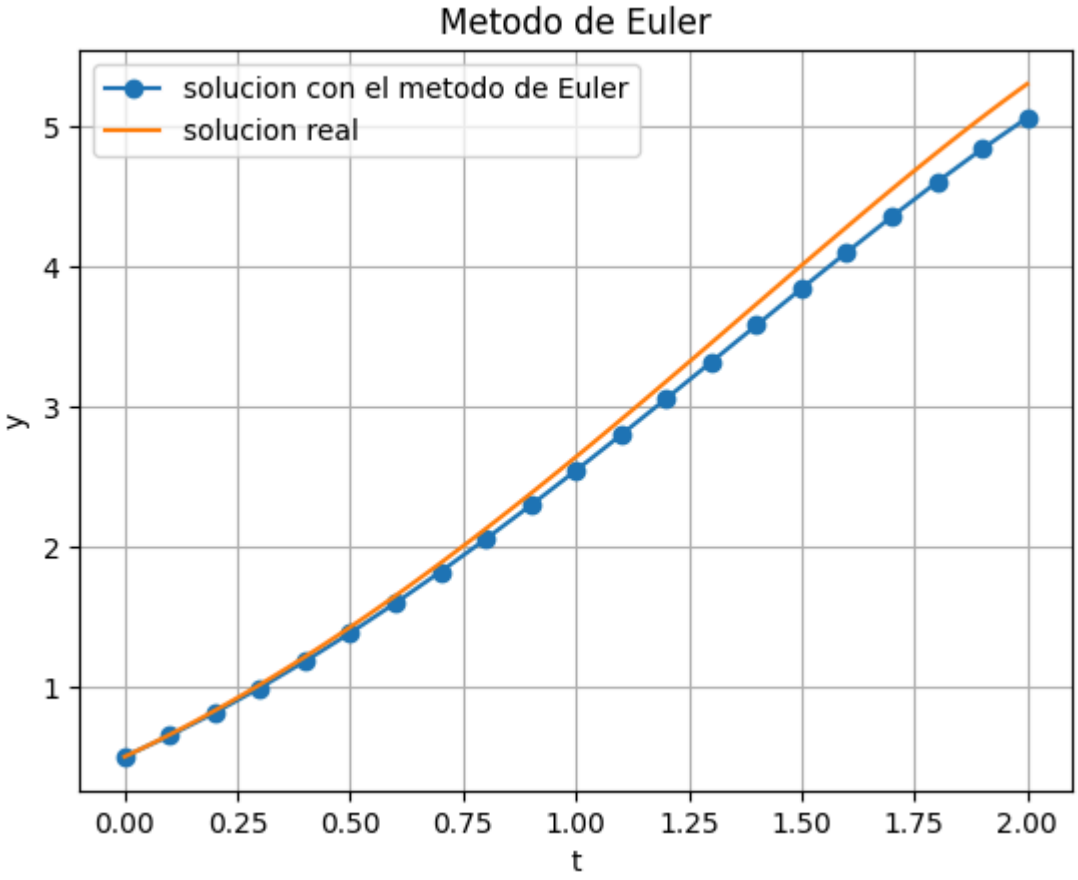
In [69]:
```
plt.plot(t_values_n5, y_values_n5, 'o-', label='solucion con el metodo de Euler'
plt.plot(t_continuous, y_continuous, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Con N = 20

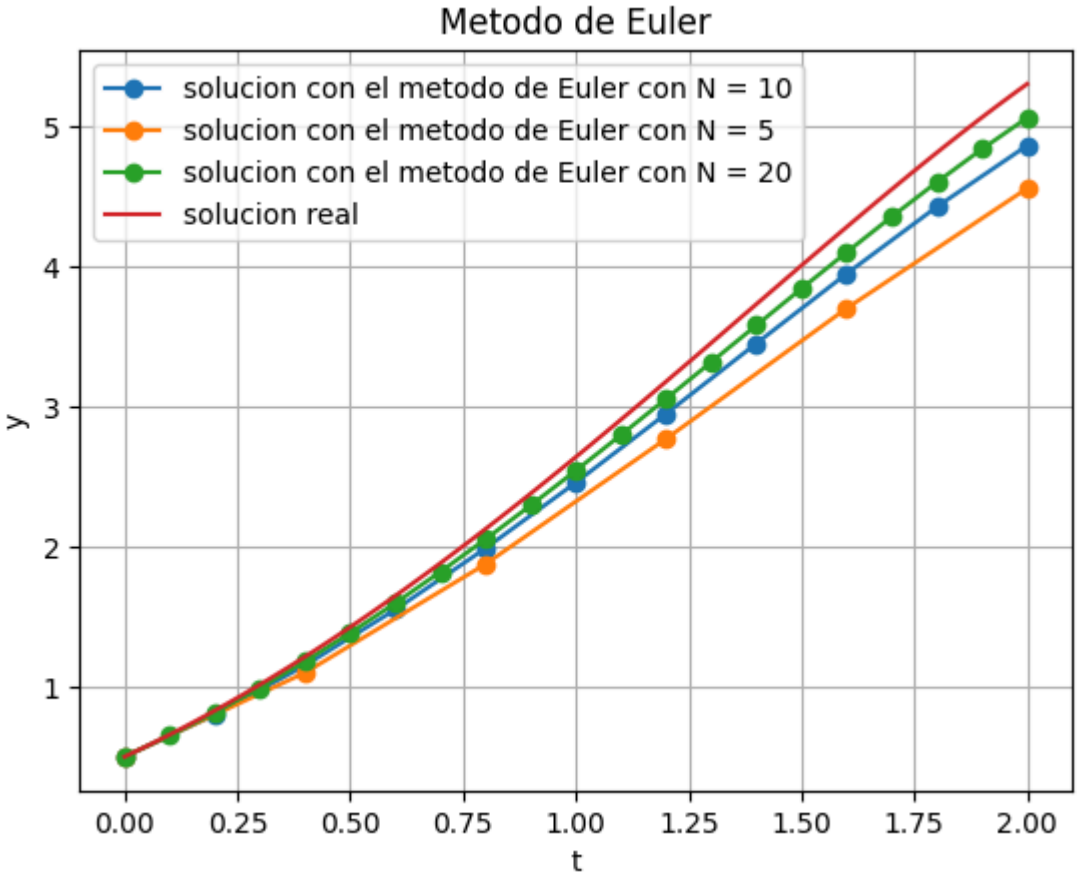
In [70]:
```
a = 0
b = 2
f = lambda t, y: y - t**2 + 1
y_t0 = 0.5
N = 20

y_values_n20, t_values_n20, h_n20 = ODE_euler(a=a, b=b, f=f, y_t0=y_t0, N=N)
```

In [71]:
```
plt.plot(t_values_n20, y_values_n20, 'o-', label='solucion con el metodo de Eule
plt.plot(t_continuous, y_continuous, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Graficas juntas

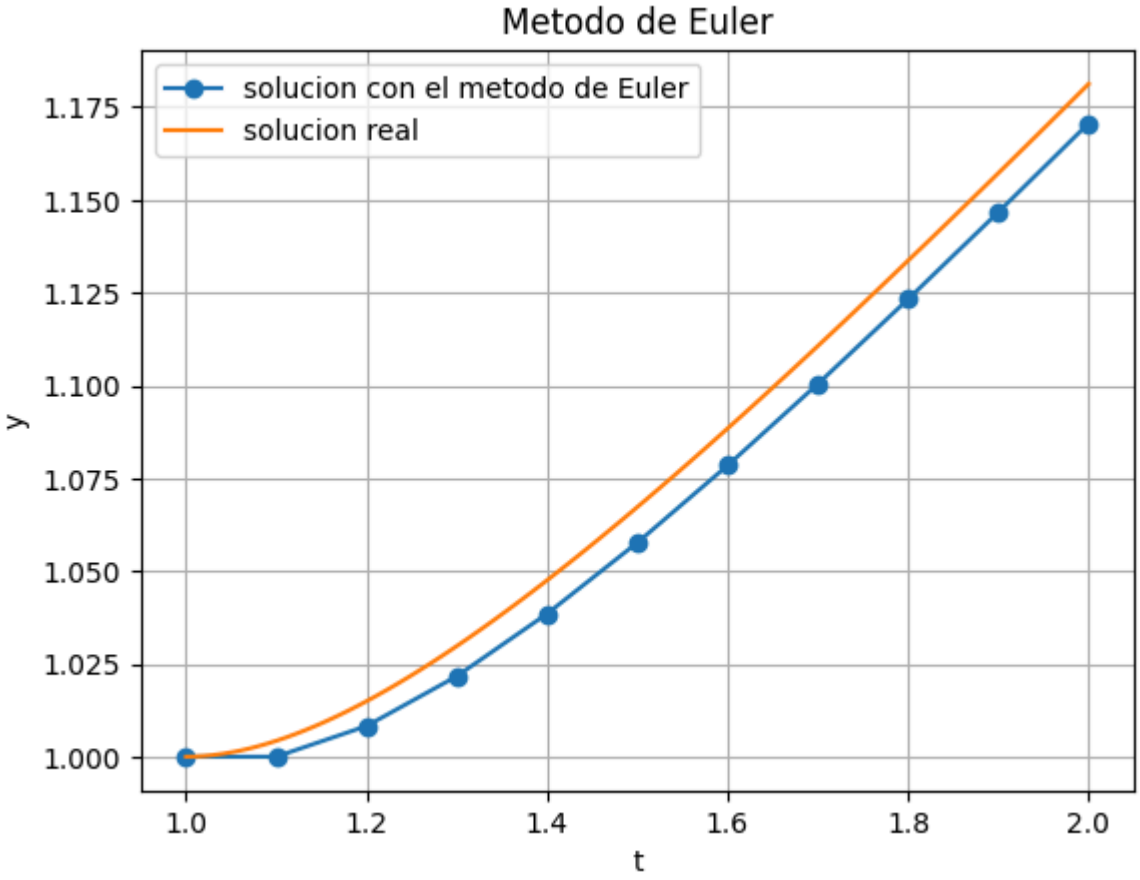
In [72]:
```
plt.plot(t_values, y_values, 'o-', label='solucion con el metodo de Euler con N
plt.plot(t_values_n5, y_values_n5, 'o-', label='solucion con el metodo de Euler
plt.plot(t_values_n20, y_values_n20, 'o-', label='solucion con el metodo de Eule
plt.plot(t_continuous, y_continuous, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Ejercicio 3

In [73]:
```
a = 1
b = 2
f = lambda t, y: y/t -(y/t)**2
y_t0 = 1
N = 10

y_values_ej3_n10, t_values_ej3_n10, h_ej3_n10 = ODE_euler(a=a, b=b, f=f, y_t0=y_
```

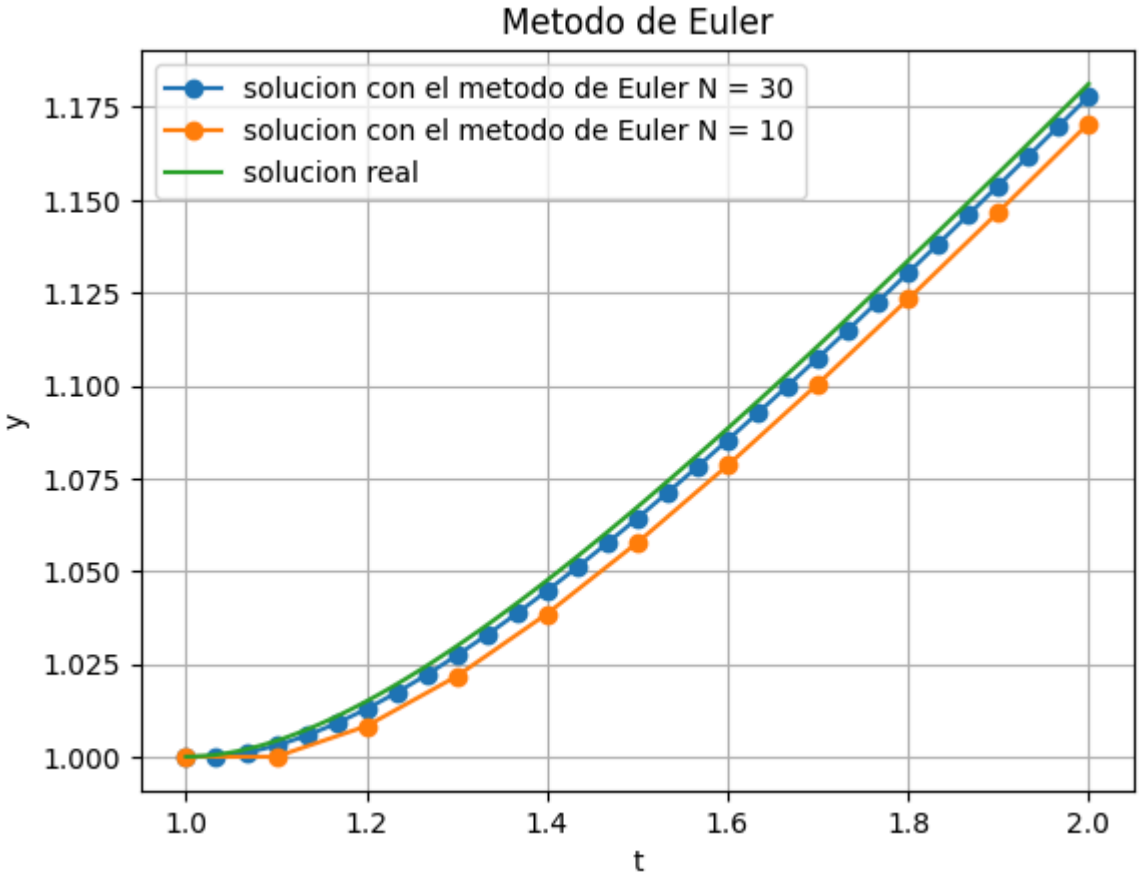
In [74]:
```
#sol real
def y_real_ej3(t):
    return t/(1+np.log(t))

t_continuous_ej3 = np.linspace(a, b, 100)
y_continuous_ej3 = y_real_ej3(t_continuous_ej3)

plt.plot(t_values_ej3_n10, y_values_ej3_n10, 'o-', label='solucion con el metodo
plt.plot(t_continuous_ej3, y_continuous_ej3, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Ejercicio 4

¿Que pasa al aumentar N?

In [76]:
```
a = 1
b = 2
f = lambda t, y: y/t -(y/t)**2
y_t0 = 1
N = 30

y_values_ej3_n30, t_values_ej3_n30, h_ej3_n30 = ODE_euler(a=a, b=b, f=f, y_t0=y_


#sol real
def y_real_ej3(t):
    return t/(1+np.log(t))

t_continuous_ej3 = np.linspace(a, b, 100)
y_continuous_ej3 = y_real_ej3(t_continuous_ej3)

plt.plot(t_values_ej3_n30, y_values_ej3_n30, 'o-', label='solucion con el metodo
plt.plot(t_values_ej3_n10, y_values_ej3_n10, 'o-', label='solucion con el metodo
plt.plot(t_continuous_ej3, y_continuous_ej3, label='solucion real')
plt.xlabel('t')
plt.ylabel('y')
plt.title('Metodo de Euler')
plt.legend()
plt.grid(True)
plt.show()
```



Al aumentar N la solucion es mas aproximada a la solucion real