In [1]:

```
%load_ext autoreload
```

In [2]:

```
%autoreload 2
from src import gauss_jacobi, gauss_seidel
```

```
[01-25 15:37:41][INFO] 2025-01-25 15:37:41.991927
[01-25 15:37:42][INFO] 2025-01-25 15:37:42.472141
```

Taller Gauss Jacobi - Gauss Seidel

Nombre: Alexis Bautista

Fecha de entrega: 25 de enero de 2025

Curso: GR1CC

Link de GitHub: https://github.com/alexis-bautista/Talller-Gauss-Jacobi-Seidel-MN

Metodo Gauss Jacobi

Con estimacion inicial x_0 = (1, 1)

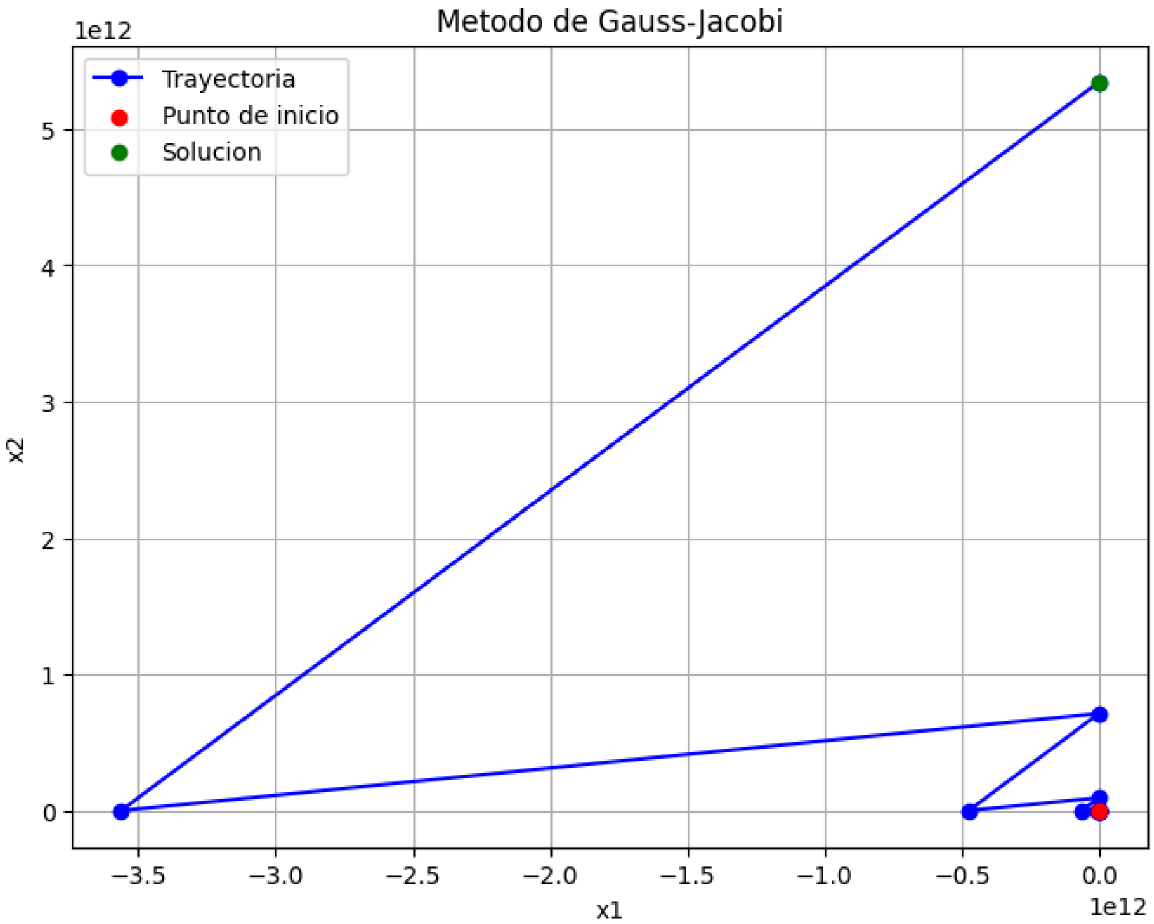
In [3]:

```
import numpy as np
import matplotlib.pyplot as plt

A = np.array([[2, 10], [3, 2]])
b = np.array([16, 11])
x0 = np.array([1,1])
tol = 1e-6
max_iter = 30

solucion_jacobi_1,tray_jacobi_1 = gauss_jacobi(A=A, b=b, x0=x0, tol=tol, max_ite

print("Solución: ", solucion_jacobi_1)
```

```
[01-25 15:37:45][INFO] i= 0 x: [1 1]
[01-25 15:37:45][INFO] i= 1 x: [[3. 4.]]
[01-25 15:37:45][INFO] i= 2 x: [[-12.   1.]]
[01-25 15:37:45][INFO] i= 1 x: [[3. 4.]]
[01-25 15:37:45][INFO] i= 2 x: [[-12.   1.]]
[01-25 15:37:45][INFO] i= 3 x: [[ 3.  23.5]]
[01-25 15:37:45][INFO] i= 4 x: [[-109.5    1. ]]
[01-25 15:37:45][INFO] i= 5 x: [[  3.   169.75]]
[01-25 15:37:45][INFO] i= 6 x: [[-840.75    1.  ]]
[01-25 15:37:45][INFO] i= 7 x: [[   3.    1266.625]]
[01-25 15:37:45][INFO] i= 8 x: [[-6.325125e+03  1.000000e+00]]
[01-25 15:37:45][INFO] i= 9 x: [[3.0000000e+00 9.4931875e+03]]
[01-25 15:37:45][INFO] i= 10 x: [[-4.74579375e+04  1.00000000e+00]]
[01-25 15:37:45][INFO] i= 11 x: [[3.00000000e+00 7.11924062e+04]]
[01-25 15:37:45][INFO] i= 12 x: [[-3.55954031e+05  1.00000000e+00]]
[01-25 15:37:45][INFO] i= 13 x: [[3.00000000e+00 5.33936547e+05]]
[01-25 15:37:45][INFO] i= 14 x: [[-2.66967473e+06  1.00000000e+00]]
[01-25 15:37:45][INFO] i= 15 x: [[3.0000000e+00 4.0045176e+06]]
[01-25 15:37:45][INFO] i= 16 x: [[-2.002258e+07  1.000000e+00]]
[01-25 15:37:45][INFO] i= 17 x: [[3.00000000e+00 3.00338755e+07]]
[01-25 15:37:45][INFO] i= 18 x: [[-1.5016937e+08  1.0000000e+00]]
[01-25 15:37:45][INFO] i= 19 x: [[3.0000000e+00 2.2525406e+08]]
[01-25 15:37:45][INFO] i= 20 x: [[-1.12627029e+09  1.00000000e+00]]
[01-25 15:37:45][INFO] i= 21 x: [[3.00000000e+00 1.68940544e+09]]
[01-25 15:37:45][INFO] i= 22 x: [[-8.4470272e+09  1.0000000e+00]]
[01-25 15:37:45][INFO] i= 23 x: [[3.00000000e+00 1.26705408e+10]]
[01-25 15:37:45][INFO] i= 24 x: [[-6.3352704e+10  1.0000000e+00]]
[01-25 15:37:45][INFO] i= 25 x: [[3.00000000e+00 9.50290561e+10]]
[01-25 15:37:45][INFO] i= 26 x: [[-4.7514528e+11  1.0000000e+00]]
[01-25 15:37:45][INFO] i= 27 x: [[3.00000000e+00 7.12717921e+11]]
[01-25 15:37:45][INFO] i= 28 x: [[-3.5635896e+12  1.0000000e+00]]
[01-25 15:37:45][INFO] i= 29 x: [[3.0000000e+00 5.3453844e+12]]
Solución:  [[3.0000000e+00]
 [5.3453844e+12]]
```

In [4]:

```python
tray_jacobi_1 = np.array([x.flatten() for x in tray_jacobi_1])
# Graficar
plt.figure(figsize=(8, 6))
plt.plot(tray_jacobi_1[:, 0], tray_jacobi_1[:, 1], marker='o', color='b', label=
plt.scatter(x0[0], x0[1], color='r', label='Punto de inicio', zorder=5)
plt.scatter(solucion_jacobi_1[0], solucion_jacobi_1[1], color='g', label='Soluci

plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Metodo de Gauss-Jacobi')

plt.legend()
plt.grid(True)
plt.show()
```



Con estimacion inicial x_0 = (1, 1) y cambio de fila

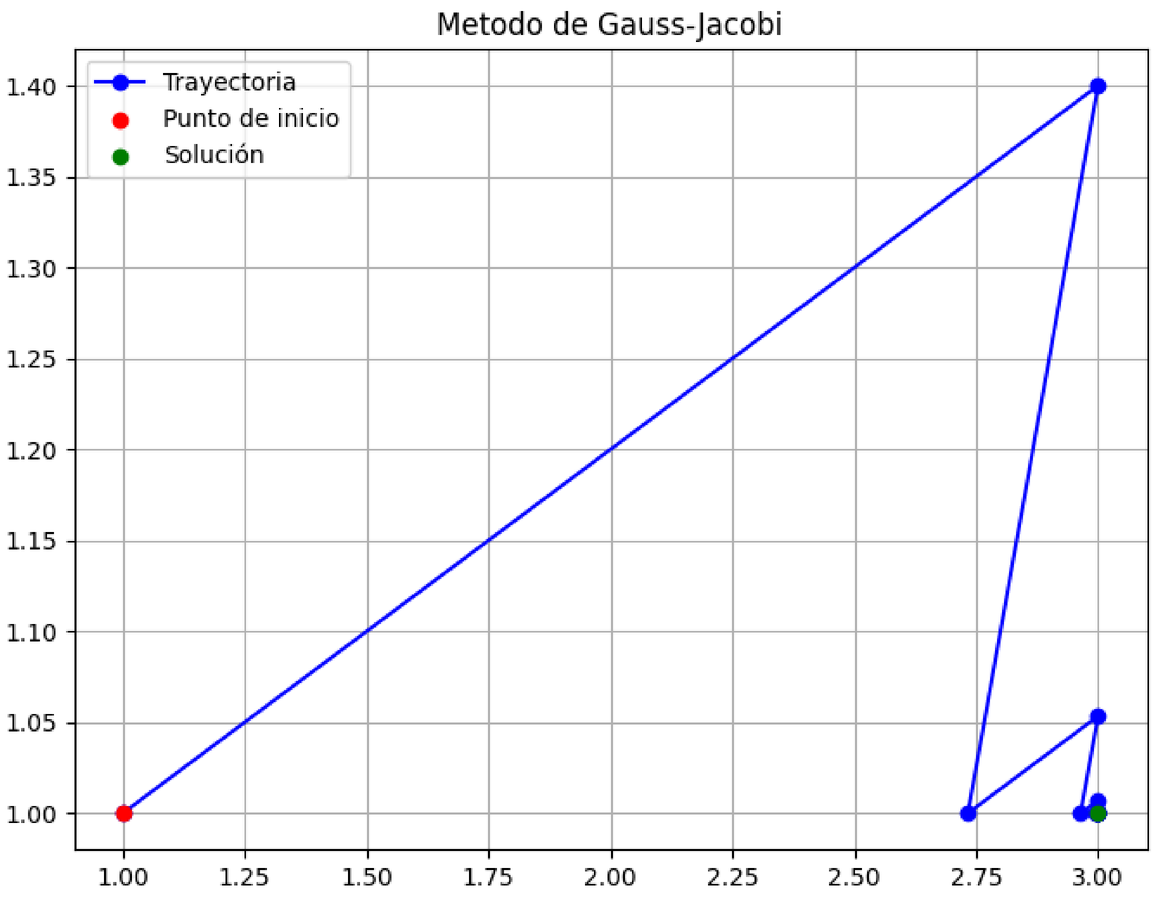
In [5]:

```
A = np.array([[3, 2], [2, 10]])
b = np.array([11, 16])
x0 = np.array([1,1])
tol = 1e-6
max_iter = 30

solucion_jacobi_2, tray_jacobi_2 = gauss_jacobi(A= A, b= b,x0 = x0, tol= tol, ma

print("Solucion:", solucion_jacobi_2)
```

```
[01-25 15:37:47][INFO] i= 0 x: [1 1]
[01-25 15:37:47][INFO] i= 1 x: [[3.  1.4]]
[01-25 15:37:47][INFO] i= 2 x: [[2.73333333 1.        ]]
[01-25 15:37:47][INFO] i= 3 x: [[3.         1.05333333]]
[01-25 15:37:47][INFO] i= 1 x: [[3.  1.4]]
[01-25 15:37:47][INFO] i= 2 x: [[2.73333333 1.        ]]
[01-25 15:37:47][INFO] i= 3 x: [[3.         1.05333333]]
[01-25 15:37:47][INFO] i= 4 x: [[2.96444444 1.        ]]
[01-25 15:37:47][INFO] i= 5 x: [[3.         1.00711111]]
[01-25 15:37:47][INFO] i= 6 x: [[2.99525926 1.        ]]
[01-25 15:37:47][INFO] i= 7 x: [[3.         1.00094815]]
[01-25 15:37:47][INFO] i= 8 x: [[2.9993679 1.       ]]
[01-25 15:37:47][INFO] i= 9 x: [[3.         1.00012642]]
[01-25 15:37:47][INFO] i= 10 x: [[2.99991572 1.        ]]
[01-25 15:37:47][INFO] i= 11 x: [[3.         1.00001686]]
[01-25 15:37:47][INFO] i= 12 x: [[2.99998876 1.        ]]
[01-25 15:37:47][INFO] i= 13 x: [[3.         1.00000225]]
[01-25 15:37:47][INFO] i= 14 x: [[2.9999985 1.       ]]
[01-25 15:37:47][INFO] i= 15 x: [[3.        1.0000003]]
Solucion: [[2.9999998]
 [1.       ]]
```

In [6]:

```
tray_jacobi_2 = np.array([x.flatten() for x in tray_jacobi_2])
# Graficar
plt.figure(figsize=(8, 6))
plt.plot(tray_jacobi_2[:, 0], tray_jacobi_2[:, 1], marker='o', color='b', label=
plt.scatter(x0[0], x0[1], color='r', label='Punto de inicio', zorder=5)
plt.scatter(solucion_jacobi_2[0], solucion_jacobi_2[1], color='g', label='Soluci

plt.title('Metodo de Gauss-Jacobi')
plt.legend()
plt.grid(True)
plt.show()
```



Con estimacion inicial x_0 = (5, -2)

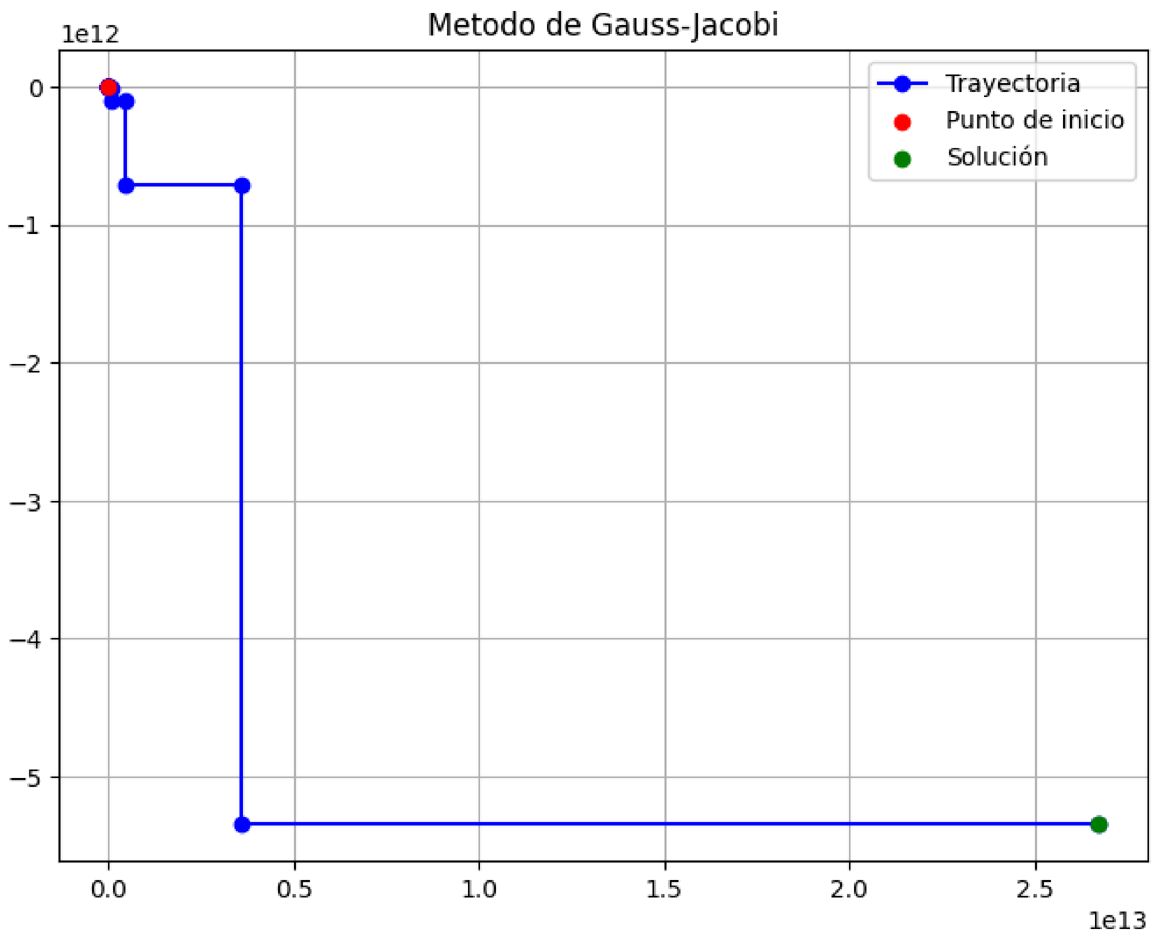
In [25]:

```
import numpy as np
import matplotlib.pyplot as plt

A = np.array([[2, 10], [3, 2]])
b = np.array([16, 11])
x0 = np.array([5,-2])
tol = 1e-6
max_iter = 30

solucion_jacobi_3,tray_jacobi_3 = gauss_jacobi(A=A, b=b, x0=x0, tol=tol, max_ite

print("Solución: ", solucion_jacobi_3)
```

```
[01-25 15:42:41][INFO] i= 0 x: [ 5 -2]
[01-25 15:42:41][INFO] i= 1 x: [[18. -2.]]
[01-25 15:42:41][INFO] i= 2 x: [[ 18.  -21.5]]
[01-25 15:42:41][INFO] i= 1 x: [[18. -2.]]
[01-25 15:42:41][INFO] i= 2 x: [[ 18.  -21.5]]
[01-25 15:42:41][INFO] i= 3 x: [[115.5 -21.5]]
[01-25 15:42:41][INFO] i= 4 x: [[ 115.5  -167.75]]
[01-25 15:42:41][INFO] i= 5 x: [[ 846.75 -167.75]]
[01-25 15:42:41][INFO] i= 6 x: [[  846.75  -1264.625]]
[01-25 15:42:41][INFO] i= 7 x: [[ 6331.125 -1264.625]]
[01-25 15:42:41][INFO] i= 8 x: [[ 6331.125  -9491.1875]]
[01-25 15:42:41][INFO] i= 9 x: [[47463.9375 -9491.1875]]
[01-25 15:42:41][INFO] i= 10 x: [[ 47463.9375  -71190.40625]]
[01-25 15:42:41][INFO] i= 11 x: [[355960.03125 -71190.40625]]
[01-25 15:42:41][INFO] i= 12 x: [[ 355960.03125  -533934.546875]]
[01-25 15:42:41][INFO] i= 13 x: [[2669680.734375 -533934.546875]]
[01-25 15:42:41][INFO] i= 14 x: [[ 2669680.734375  -4004515.6015625]]
[01-25 15:42:41][INFO] i= 15 x: [[20022586.0078125 -4004515.6015625]]
[01-25 15:42:41][INFO] i= 16 x: [[ 20022586.0078125  -30033873.51171875]]
[01-25 15:42:41][INFO] i= 17 x: [[ 1.50169376e+08 -3.00338735e+07]]
[01-25 15:42:41][INFO] i= 18 x: [[ 1.50169376e+08 -2.25254058e+08]]
[01-25 15:42:41][INFO] i= 19 x: [[ 1.12627030e+09 -2.25254058e+08]]
[01-25 15:42:41][INFO] i= 20 x: [[ 1.12627030e+09 -1.68940544e+09]]
[01-25 15:42:41][INFO] i= 21 x: [[ 8.44702721e+09 -1.68940544e+09]]
[01-25 15:42:41][INFO] i= 22 x: [[ 8.44702721e+09 -1.26705408e+10]]
[01-25 15:42:41][INFO] i= 23 x: [[ 6.33527041e+10 -1.26705408e+10]]
[01-25 15:42:41][INFO] i= 24 x: [[ 6.33527041e+10 -9.50290561e+10]]
[01-25 15:42:41][INFO] i= 25 x: [[ 4.75145280e+11 -9.50290561e+10]]
[01-25 15:42:41][INFO] i= 26 x: [[ 4.75145280e+11 -7.12717921e+11]]
[01-25 15:42:41][INFO] i= 27 x: [[ 3.56358960e+12 -7.12717921e+11]]
[01-25 15:42:41][INFO] i= 28 x: [[ 3.5635896e+12 -5.3453844e+12]]
[01-25 15:42:41][INFO] i= 29 x: [[ 2.6726922e+13 -5.3453844e+12]]
Solución:  [[ 2.6726922e+13]
 [-5.3453844e+12]]
```

In [26]:

```
tray_jacobi_3 = np.array([x.flatten() for x in tray_jacobi_3])
# Graficar
plt.figure(figsize=(8, 6))
plt.plot(tray_jacobi_3[:, 0], tray_jacobi_3[:, 1], marker='o', color='b', label=
plt.scatter(x0[0], x0[1], color='r', label='Punto de inicio', zorder=5)
plt.scatter(solucion_jacobi_3[0], solucion_jacobi_3[1], color='g', label='Soluci


plt.title('Metodo de Gauss-Jacobi')
plt.legend()
plt.grid(True)
plt.show()
```



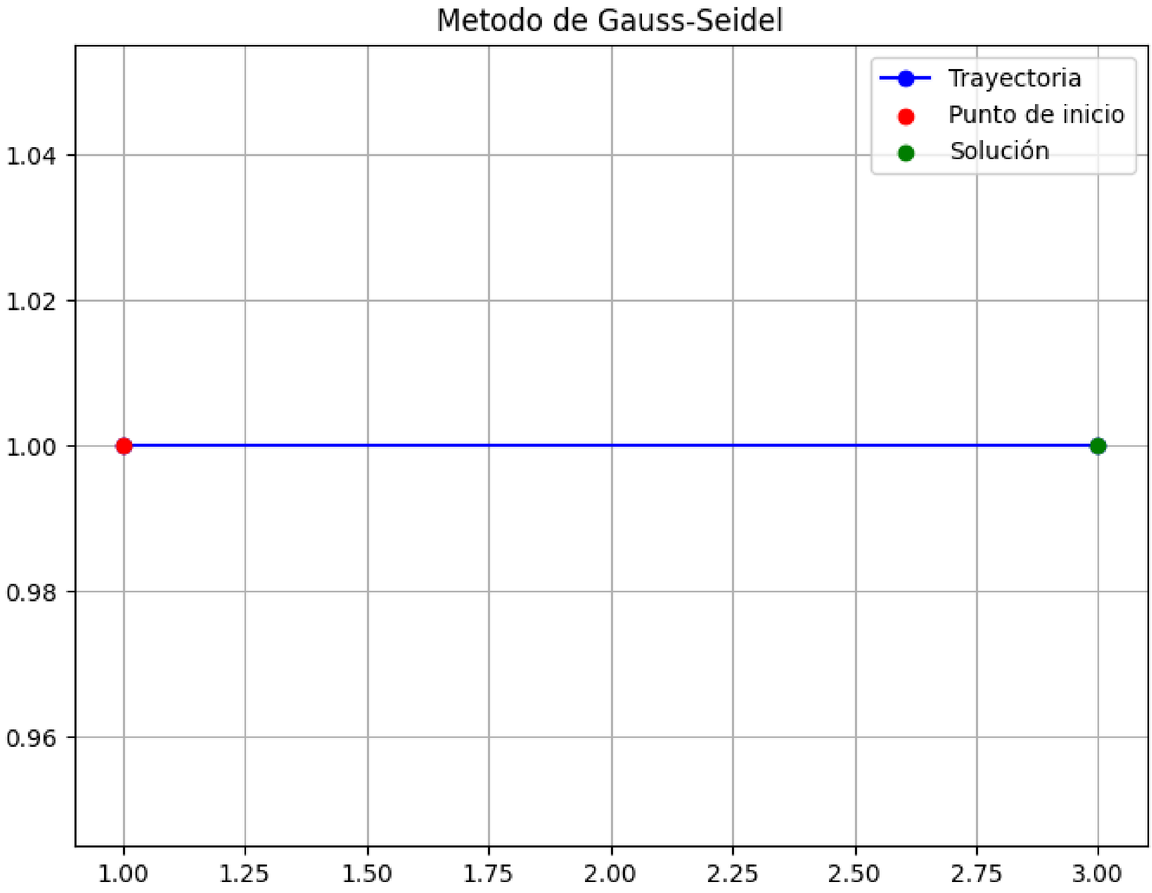
Metodo Gauss Seidel

Con estimacion inicial x_0 = (1, 1)

```
In [9]:  A = np.array([[2, 10], [3, 2]])
         b = np.array([16, 11])
         x0 = np.array([1,1])
         tol = 1e-6
         max_iter = 30

         solucion_seidel_1, tray_seidel_1 = gauss_seidel(A=A, b=b, x0=x0, tol=tol, max_it

         print("solucion: ", solucion_seidel_1)
```

```
[01-25 15:37:51][INFO] i= 0 x: [1 1]
[01-25 15:37:51][INFO] i= 1 x: [[3. 1.]]
solucion:  [[3.]
 [1.]]
[01-25 15:37:51][INFO] i= 1 x: [[3. 1.]]
solucion:  [[3.]
 [1.]]
```

```
In [10]: tray_seidel_1 = np.array([x.flatten() for x in tray_seidel_1])
         # Graficar
         plt.figure(figsize=(8, 6))
         plt.plot(tray_seidel_1[:, 0], tray_seidel_1[:, 1], marker='o', color='b', label=
         plt.scatter(x0[0], x0[1], color='r', label='Punto de inicio', zorder=5)
         plt.scatter(solucion_seidel_1[0], solucion_seidel_1[1], color='g', label='Soluci
         plt.title('Metodo de Gauss-Seidel')
         plt.legend()
         plt.grid(True)
         plt.show()
```



Con estimacion inicial x_0 = (5, -2)

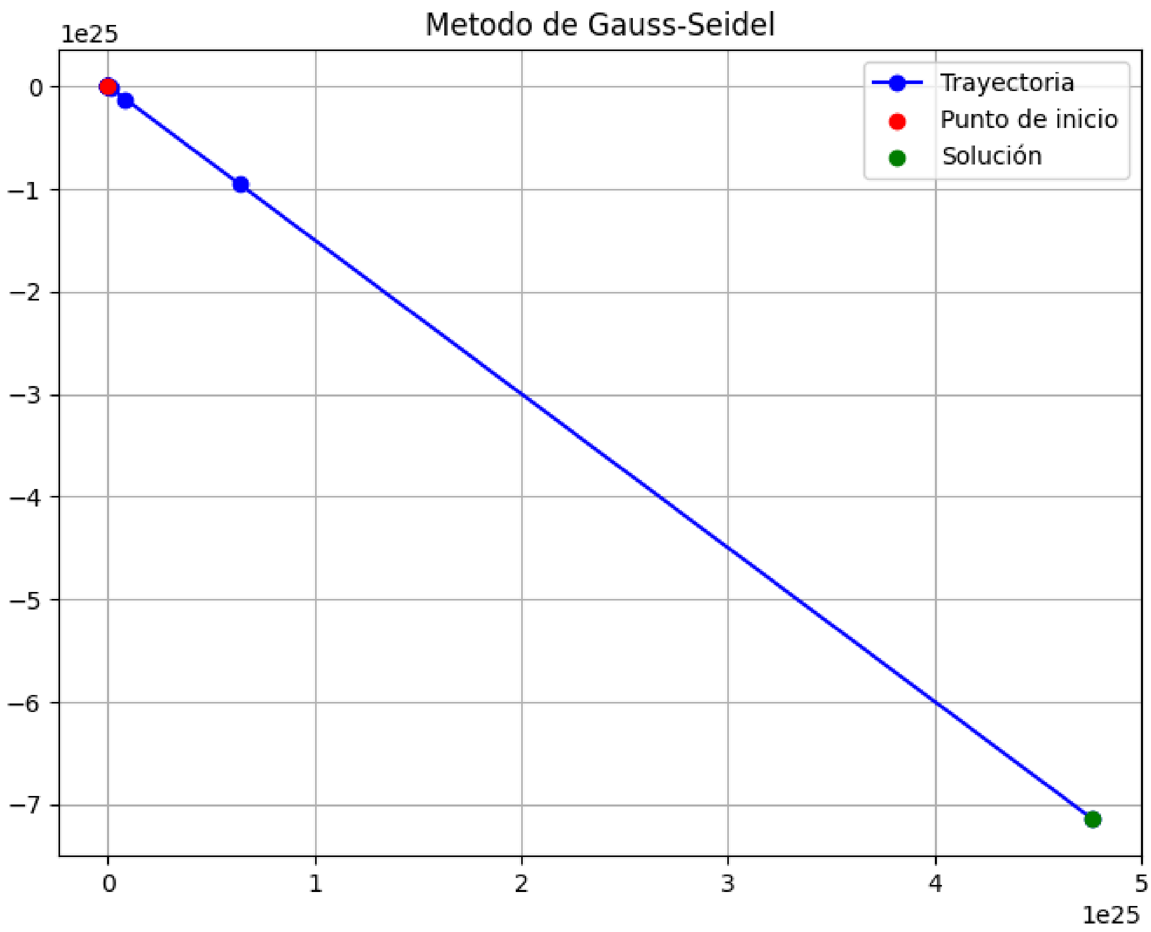
In [23]:

```
A = np.array([[2, 10], [3, 2]])
b = np.array([16, 11])
x0 = np.array([5,-2])
tol = 1e-6
max_iter = 30

solucion_seidel_2,tray_seidel_2 = gauss_seidel(A=A, b=b, x0=x0, tol=tol, max_ite

print("Solución: ", solucion_seidel_2)
```

```
[01-25 15:42:22][INFO] i= 0 x: [ 5 -2]
[01-25 15:42:22][INFO] i= 1 x: [[ 18.   -21.5]]
[01-25 15:42:22][INFO] i= 2 x: [[ 115.5   -167.75]]
[01-25 15:42:22][INFO] i= 3 x: [[  846.75   -1264.625]]
[01-25 15:42:22][INFO] i= 1 x: [[ 18.   -21.5]]
[01-25 15:42:22][INFO] i= 2 x: [[ 115.5   -167.75]]
[01-25 15:42:22][INFO] i= 3 x: [[  846.75   -1264.625]]
[01-25 15:42:22][INFO] i= 4 x: [[ 6331.125   -9491.1875]]
[01-25 15:42:22][INFO] i= 5 x: [[ 47463.9375   -71190.40625]]
[01-25 15:42:22][INFO] i= 6 x: [[ 355960.03125   -533934.546875]]
[01-25 15:42:22][INFO] i= 7 x: [[ 2669680.734375   -4004515.6015625]]
[01-25 15:42:22][INFO] i= 8 x: [[ 20022586.0078125   -30033873.51171875]]
[01-25 15:42:22][INFO] i= 9 x: [[ 1.50169376e+08 -2.25254058e+08]]
[01-25 15:42:22][INFO] i= 10 x: [[ 1.12627030e+09 -1.68940544e+09]]
[01-25 15:42:22][INFO] i= 11 x: [[ 8.44702721e+09 -1.26705408e+10]]
[01-25 15:42:22][INFO] i= 12 x: [[ 6.33527041e+10 -9.50290561e+10]]
[01-25 15:42:22][INFO] i= 13 x: [[ 4.75145280e+11 -7.12717921e+11]]
[01-25 15:42:22][INFO] i= 14 x: [[ 3.5635896e+12 -5.3453844e+12]]
[01-25 15:42:22][INFO] i= 15 x: [[ 2.6726922e+13 -4.0090383e+13]]
[01-25 15:42:22][INFO] i= 16 x: [[ 2.00451915e+14 -3.00677873e+14]]
[01-25 15:42:22][INFO] i= 17 x: [[ 1.50338936e+15 -2.25508405e+15]]
[01-25 15:42:22][INFO] i= 18 x: [[ 1.12754202e+16 -1.69131303e+16]]
[01-25 15:42:22][INFO] i= 19 x: [[ 8.45656517e+16 -1.26848478e+17]]
[01-25 15:42:22][INFO] i= 20 x: [[ 6.34242388e+17 -9.51363582e+17]]
[01-25 15:42:22][INFO] i= 21 x: [[ 4.75681791e+18 -7.13522686e+18]]
[01-25 15:42:22][INFO] i= 22 x: [[ 3.56761343e+19 -5.35142015e+19]]
[01-25 15:42:22][INFO] i= 23 x: [[ 2.67571007e+20 -4.01356511e+20]]
[01-25 15:42:22][INFO] i= 24 x: [[ 2.00678256e+21 -3.01017383e+21]]
[01-25 15:42:22][INFO] i= 25 x: [[ 1.50508692e+22 -2.25763037e+22]]
[01-25 15:42:22][INFO] i= 26 x: [[ 1.12881519e+23 -1.69322278e+23]]
[01-25 15:42:22][INFO] i= 27 x: [[ 8.46611390e+23 -1.26991709e+24]]
[01-25 15:42:22][INFO] i= 28 x: [[ 6.34958543e+24 -9.52437814e+24]]
[01-25 15:42:22][INFO] i= 29 x: [[ 4.76218907e+25 -7.14328361e+25]]
Solución:  [[ 4.76218907e+25]
 [-7.14328361e+25]]
```

In [24]:

```
tray_seidel_2 = np.array([x.flatten() for x in tray_seidel_2])
# Graficar
plt.figure(figsize=(8, 6))
plt.plot(tray_seidel_2[:, 0], tray_seidel_2[:, 1], marker='o', color='b', label=
plt.scatter(x0[0], x0[1], color='r', label='Punto de inicio', zorder=5)
plt.scatter(solucion_seidel_2[0], solucion_seidel_2[1], color='g', label='Soluci

plt.title('Metodo de Gauss-Seidel')
plt.legend()
plt.grid(True)
plt.show()
```

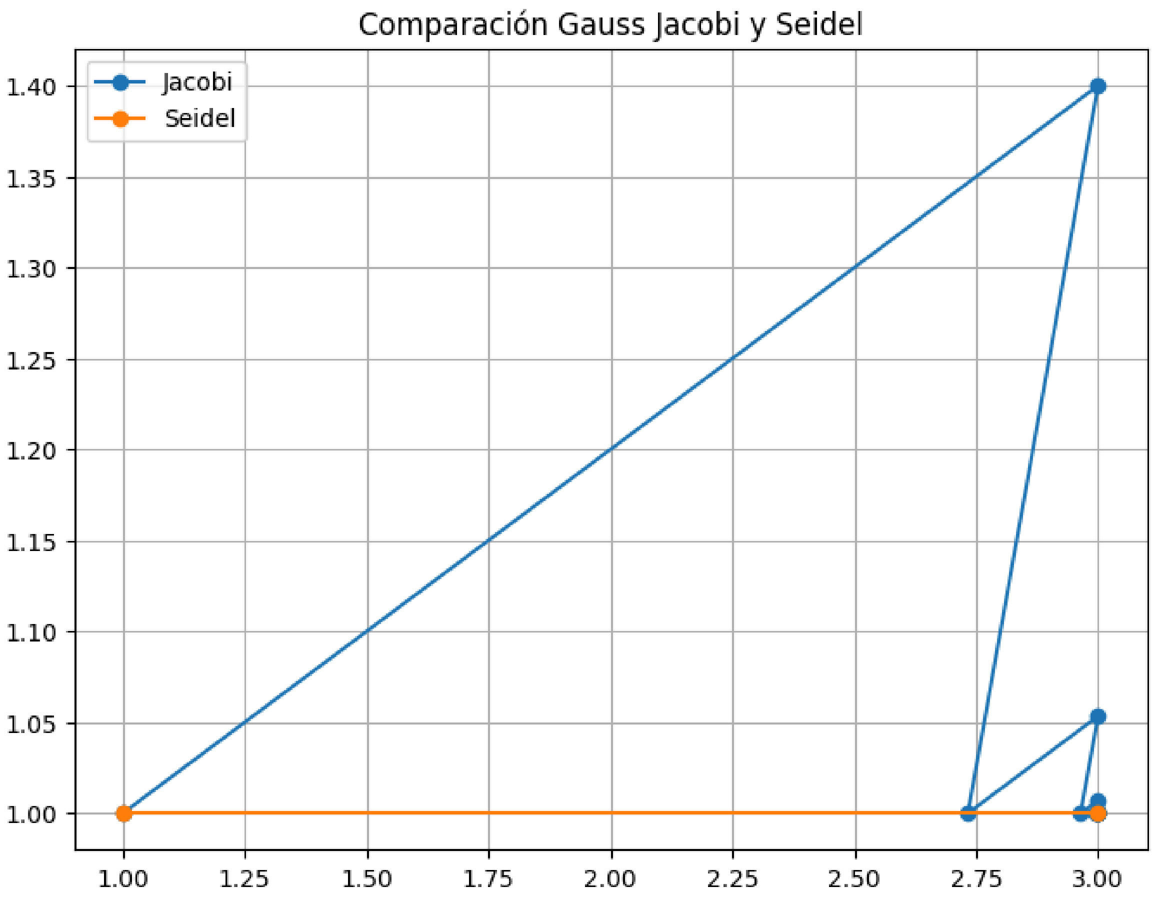


Comparacion Metodos

Con estimacion inicial x0 = (1,1)

In [13]:

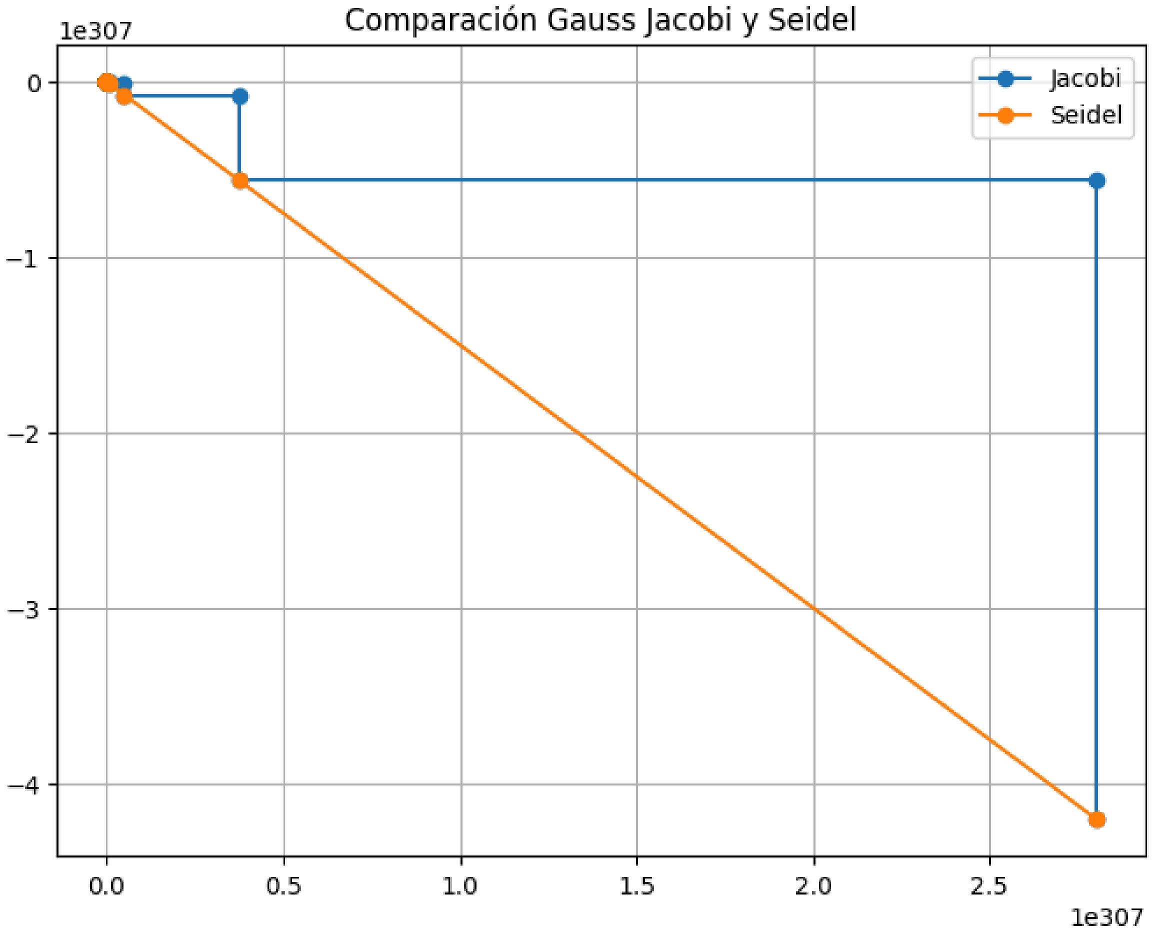
```python
fig, ax = plt.subplots(figsize=(8, 6))
ax.plot(tray_jacobi_2[:, 0], tray_jacobi_2[:, 1], marker='o', label='Jacobi')
ax.plot(tray_seidel_1[:, 0], tray_seidel_1[:, 1], marker='o', label='Seidel')

ax.set_title('Comparación Gauss Jacobi y Seidel')
ax.legend()
ax.grid(True)
plt.show()
```



Con estimacion inicial x0 = (5,-2)

In [22]:

```
fig, ax = plt.subplots(figsize=(8, 6))
ax.plot(tray_jacobi_3[:, 0], tray_jacobi_3[:, 1], marker='o', label='Jacobi')
ax.plot(tray_seidel_2[:, 0], tray_seidel_2[:, 1], marker='o', label='Seidel')

ax.set_title('Comparación Gauss Jacobi y Seidel')
ax.legend()
ax.grid(True)
plt.show()
```


Comparación Gauss Jacobi y Seidel
1e307
0
−1
−2
−3
−4
0.0
0.5
1.0
1.5
2.0
2.5
1e307
Jacobi
Seidel