5.7 Determine la raíz real de $f(x) = (0.8 - 0.3x)/x$:

a) Analíticamente
b) Gráficamente
c) Empleando tres iteraciones en el método de la falsa posición, con valores iniciales de 1 a 3. Calcule el error aproximado ε_a y el error verdadero ε_t en cada iteración.

---

a) Analíticamente

$$f(x) = \frac{0.8 - 0.3x}{x}$$

Igualamos la función a cero para obtener la raíz. El denominador no puede ser cero, por lo tanto:

$$0.8 - 0.3x = 0$$

$$x = \frac{0.8}{0.3}$$

In [1]:
```
x = 0.8/0.3
print(x)
```

```
2.666666666666667
```

b) Gráficamente

Tomamos valores con incrementos de 0.5 para x y graficamos:

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

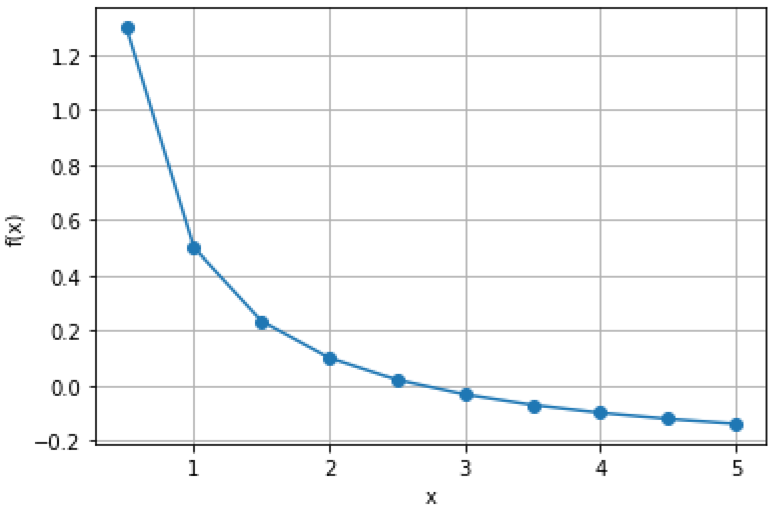
In [3]:
```
arrX = np.arange(0.5,5.1,0.5)#creando el arreglo de valores para x
#evaluamos en la función:
f_x = []
for i in arrX:
    f_x.append((0.8-0.3*i)/i)
d = {'x':arrX, 'f(x)':f_x}
df = pd.DataFrame(data=d)
df
```

Out[3]:

	x	f(x)
0	0.5	1.300000
1	1.0	0.500000
2	1.5	0.233333
3	2.0	0.100000
4	2.5	0.020000
5	3.0	-0.033333

	x	f(x)
6	3.5	-0.071429
7	4.0	-0.100000
8	4.5	-0.122222
9	5.0	-0.140000

In [4]:
```
plt.plot(arrX,f_x,'-o')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.grid(True)
plt.show()
```



De la gráfica, podemos observar que el signo de $f(x)$ cambia entre 2.5 y 3 (casi por la mitad), por lo que la raíz debe encontrarse en un valor cercano a **2.7**.

c) Método de la falsa posición:

Dados $x_l = 1$ y $x_u = 3$, si valuamos la función en estos valores tenemos que:

$$f(x_l) = 0.5$$

$$f(x_u) = -\frac{1}{30}$$

Recordando la expresión 5.7 del Chapra, tenemos que:

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

El valor de x_r calculado en la primera iteración, remplazará después a uno de los valores iniciales, en este caso usaremos x_u

In [5]:
```
xu = 3
xl = 1
xr = 8/3 #para el error comparamos con lo obtenido en a)
def f(x):
    return (0.8-0.3*x)/(x)
```

```python
def falsePos(xu, xl):
    xr = xu-((f(xu)*(xl-xu))/(f(xl)-f(xu)))
    return xr

xlArr = []
xuArr = []
xrArr = []
EaArr = []
EtArr = []
iteration = [1,2,3]

for i in range(3):
    xlArr.append(xl)
    xuArr.append(xu)
    xr = falsePos(xu,xl)
    xrArr.append(xr)
    if i == 0: EaArr.append(' ')
    else: EaArr.append(np.abs((xr-xu)*100/xr))
    xu = xr
    EtArr.append(np.abs((8/3 - xr)*100))

df = pd.DataFrame(columns=['iteration','xl','xu','xr','Ea(%)','Et(%)'])
df['iteration'] = iteration
df['xl'] = xlArr
df['xu'] = xuArr
df['xr'] = xrArr
df['Ea(%)'] = EaArr
df['Et(%)'] = EtArr
df.style.hide_index()
```

Out[5]:

iteration	xl	xu	xr	Ea(%)	Et(%)
1	1	3.000000	2.875000		20.833333
2	1	2.875000	2.796875	2.793296	13.020833
3	1	2.796875	2.748047	1.776830	8.138021

Observamos que luego de 3 iteraciones del Método de falsa posición, obtenemos una raíz $x_r = 2.748$ con un error aproximado del 1.77%. Si verificamos evaluando a esta raíz en $f(2.748)$ obtendríamos un valor de -0.0089, que es un valor relativamente cercano a cero.

Resumen:

a) De forma analítica se obtiene $x_r = 8/3 = 2.6\overline{6}$

b) Gráficamente podemos apreciar un valor cercano a $x_r = 2.7$

a) Con tres iteraciones del Método de falsa posición obtenemos que $x_r = 2.748$ con $\varepsilon_a = 1.78\%$ y $\varepsilon_t = 8.14\%$