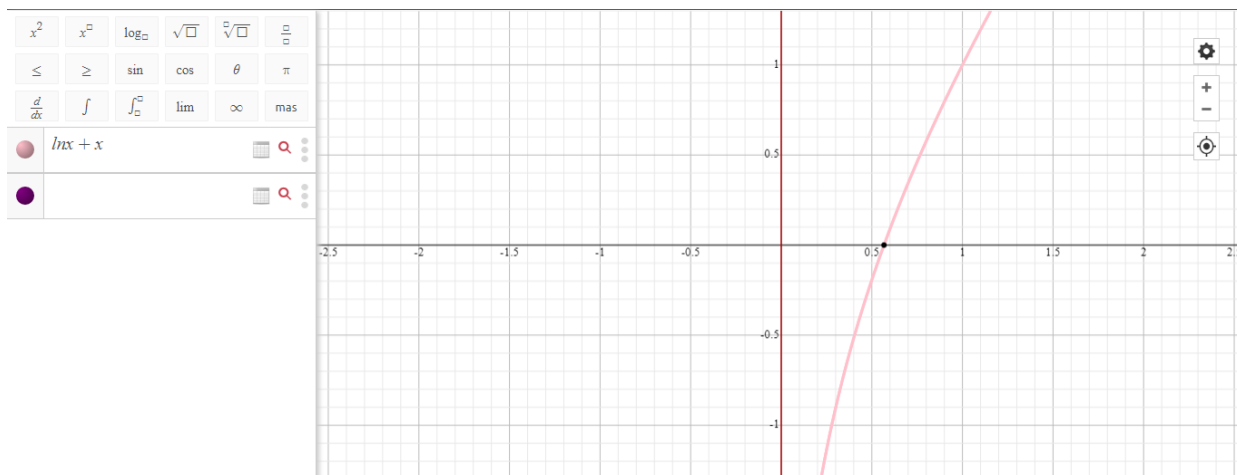


Cálculo de la raíz por el método de la bisección



```
In [1]: import numpy as np
def biseccion(f,a,b,tol,nmax):
    for i in range(1,nmax):
        if np.sign(f(a)) == np.sign(f(b)):
            print("La raíz no está en el intervalo especificado")
            return

        m = (a + b)/2

        if abs(f(m)) < tol:
            return m,i
        elif np.sign(f(a)) == np.sign(f(m)):
            a = m
        else:
            b = m
        i += i
```

```
In [2]: import math
f = lambda x: math.log(x+1e-15) + x # Añado un valor infinitesimal para evitar que se c
r1,cont = biseccion(f, 0, 1, 0.000001,100)
print("Con ",cont," iteraciones, la raíz es ",r1," y el error es ",0.575-r1) # 0.575 se
```

Con 20 iteraciones, la raíz es 0.567143440246582 y el error es 0.007856559753417924

Cálculo de la raíz por el método de la secante

```
In [3]: def secante(f,x0,x1,nmax,tol):
    for i in range(0,nmax):
        x2=(x0*f(x1)-x1*f(x0))/(f(x1)-f(x0));
        if abs(x2-x1) < tol:
            print("Con ",i," iteraciones, la raíz es ",x2," y el error es ",x2-x1)
            break
        else:
            x0=x1
```

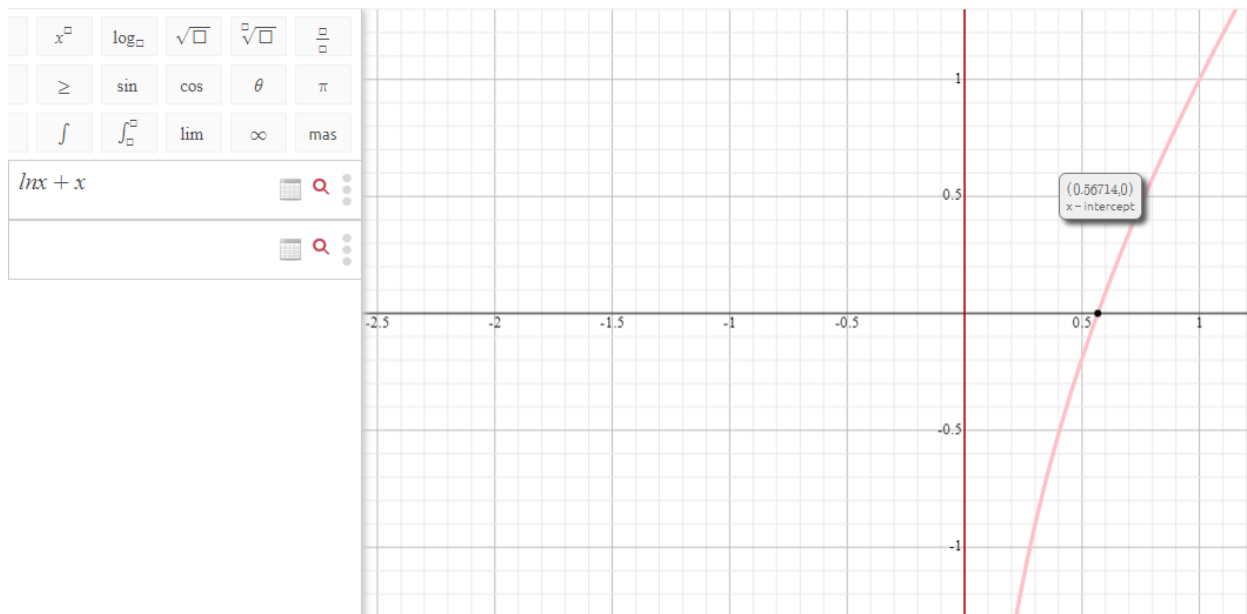
```
x1=x2  
return None
```

In [4]:

```
secante(f,0,1,20,1e-10)
```

Con 7 iteraciones, la raíz es 0.5671432904097832 y el error es -1.4432899320127035e-15

Comprobación gráfica



In []: