Alumno: Daniel Ricardo Ramírez Molina

Parcial 1

Análisis Numérico

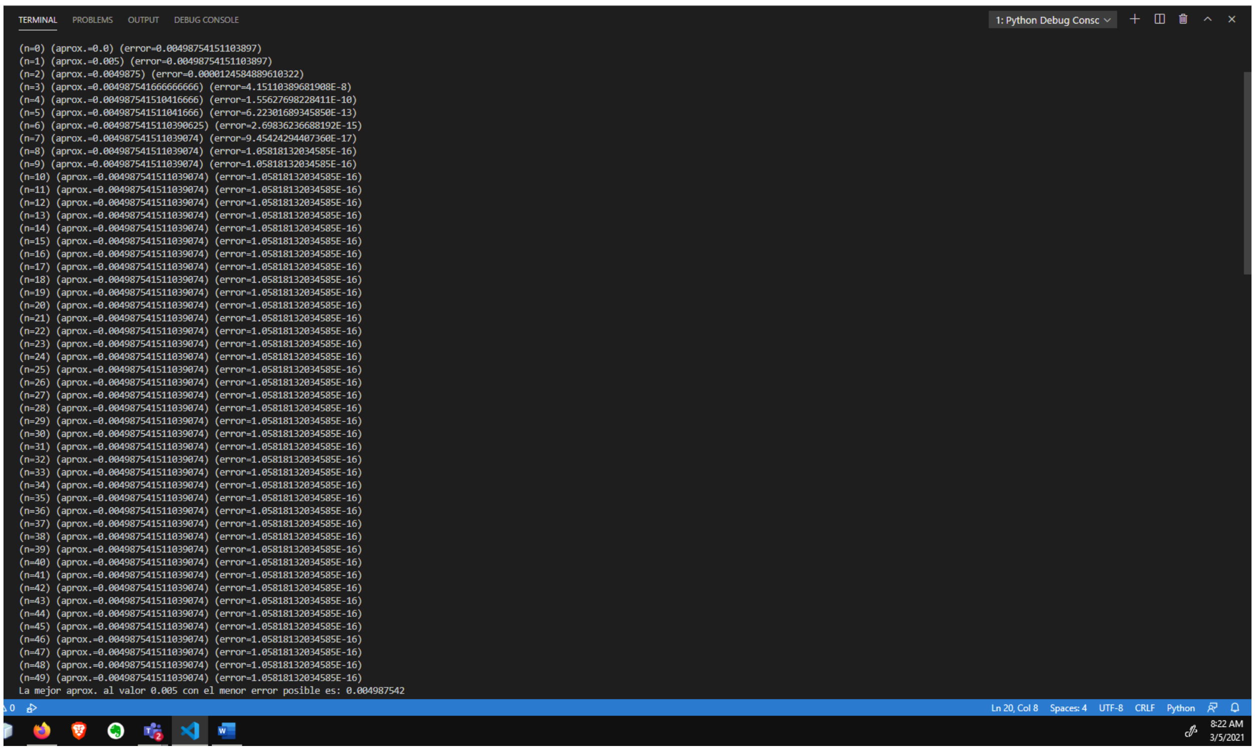
Ejercicio escogido para realizar: **3.C**

3. En los siguientes ejercicios aplicar el Teorema de Taylor para aproximar la función *f(x)* con un polinomio de Taylor alrededor de a= 0 (de menor error), estimar el error para cada *x*, realizar una gráfica que muestre el polinomio de aproximación. Implemente en R o Python, utilizar 9 cifras significativas

c) f(x) = ln(1 +x) en [−0.5,0.5] para x= 0.005, 0.0001 , 0.499999999

La implementación del algoritmo fue realizada en Python y a continuación se muestran los resultados:

Para x=0.005 (El resultado se muestra al final de la imagen)

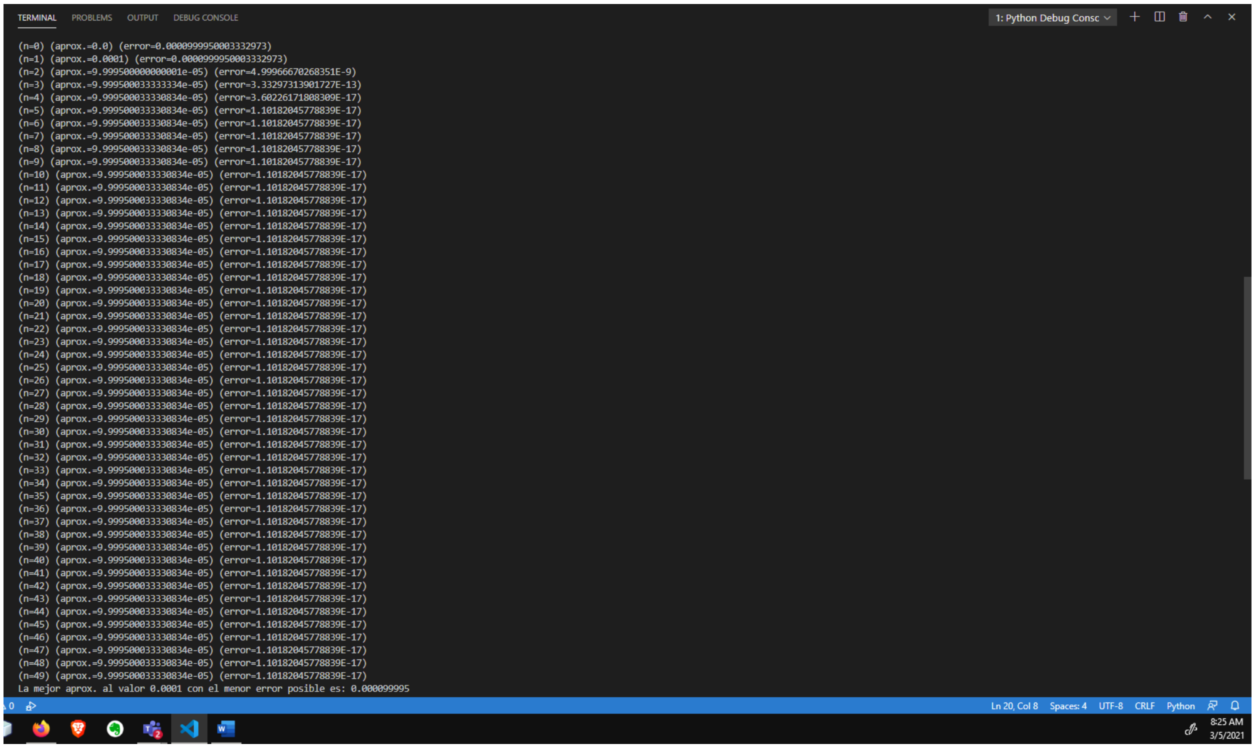


```
(n=0) (aprox.=0.0) (error=0.00498754151103897)
(n=1) (aprox.=0.005) (error=0.00498754151103897)
(n=2) (aprox.=0.0049875) (error=0.0000124584889610322)
(n=3) (aprox.=0.004987541666666666) (error=4.15110389681908E-8)
(n=4) (aprox.=0.004987541510416666) (error=1.55627698228411E-10)
(n=5) (aprox.=0.004987541511041666) (error=6.22301689345850E-13)
(n=6) (aprox.=0.0049875415110390625) (error=2.69836236688192E-15)
(n=7) (aprox.=0.004987541511039074) (error=9.45424294407360E-17)
(n=8) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=9) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=10) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=11) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=12) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=13) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=14) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=15) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=16) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=17) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=18) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=19) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=20) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=21) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=22) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=23) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=24) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=25) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=26) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=27) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=28) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=29) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=30) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=31) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=32) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=33) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=34) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=35) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=36) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=37) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=38) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=39) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=40) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=41) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=42) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=43) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=44) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=45) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=46) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=47) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=48) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
(n=49) (aprox.=0.004987541511039074) (error=1.05818132034585E-16)
La mejor aprox. al valor 0.005 con el menor error posible es: 0.004987542
```

Comprobación de respuesta con Wolfram:

Alumno: Daniel Ricardo Ramírez Molina

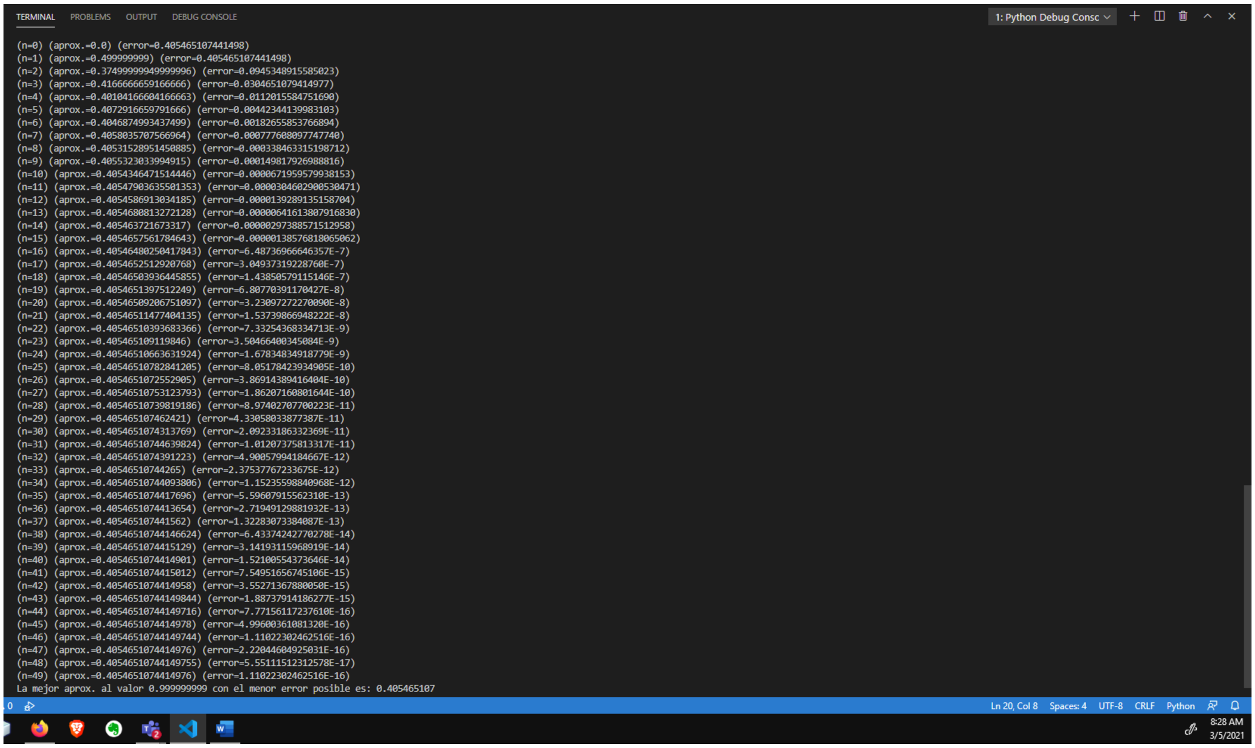
Para x=0.0001 (El resultado se muestra al final de la imagen)



Comprobación de respuesta con Wolfram:

Alumno: Daniel Ricardo Ramírez Molina

Para x=0. 0.499999999 (El resultado se muestra al final de la imagen)



Comprobación de respuesta con Wolfram:

Alumno: Daniel Ricardo Ramírez Molina

La grafica de la función de aproximación se programó en Python y se muestra a continuación (También se puede visualizar al correr el código)