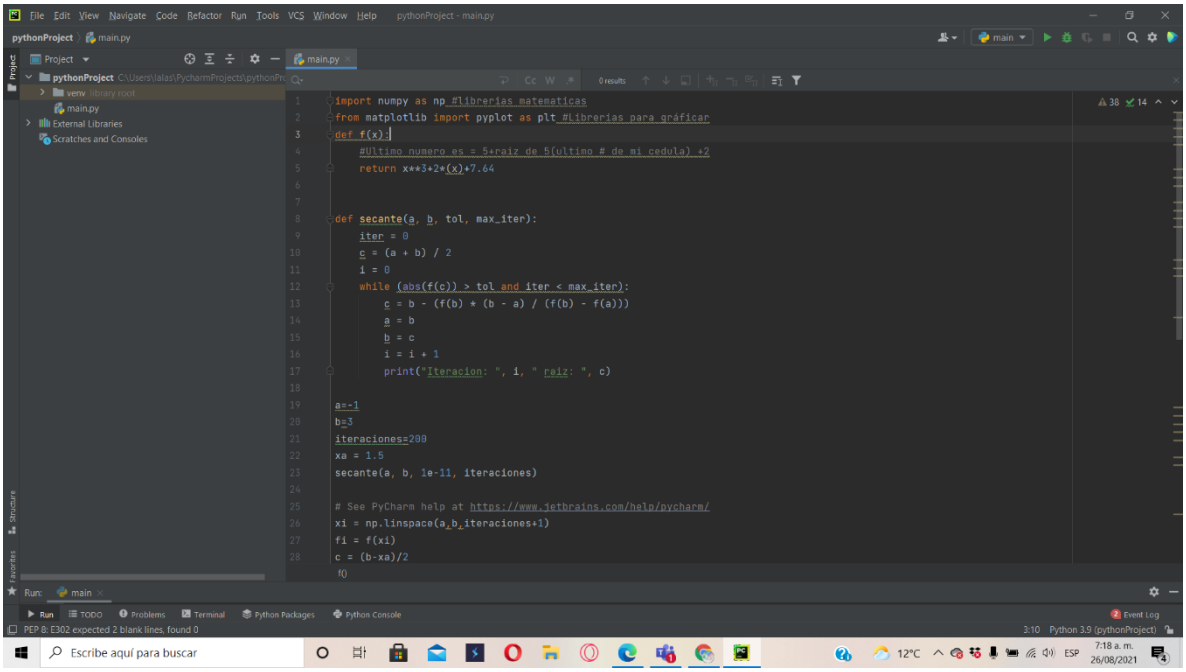
Laura Sofia Jiménez Ballén

1er parcial

1b. Método de la secante; $TOL < 10-11$

Ejercicio cuando k = 7.64



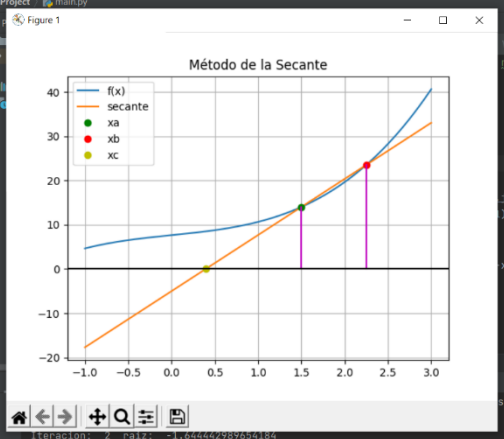
```
c = (b-xa)/2
pendiente = (f(xa+c)-f(xa))/(xa+c-xa)
b0 = f(xa) - pendiente*xa
tangentei = pendiente*xi+b0

fxa = f(xa)
xb = xa + c
fxb = f(xb)

plt.plot(xi,fi, label='f(x)')
plt.plot(xi,tangentei, label='secante')
plt.plot(xa,f(xa),'go', label='xa')
plt.plot(xa+c,f(xa+c),'ro', label='xb')
plt.plot((-b0/pendiente),0,'yo', label='xc')

plt.plot([xa,xa],[0,fxa],'m')
plt.plot([xb,xb],[0,fxb],'m')

plt.axhline(0, color='k')
plt.title('Método de la Secante')
plt.legend()
plt.grid()
plt.show()
```

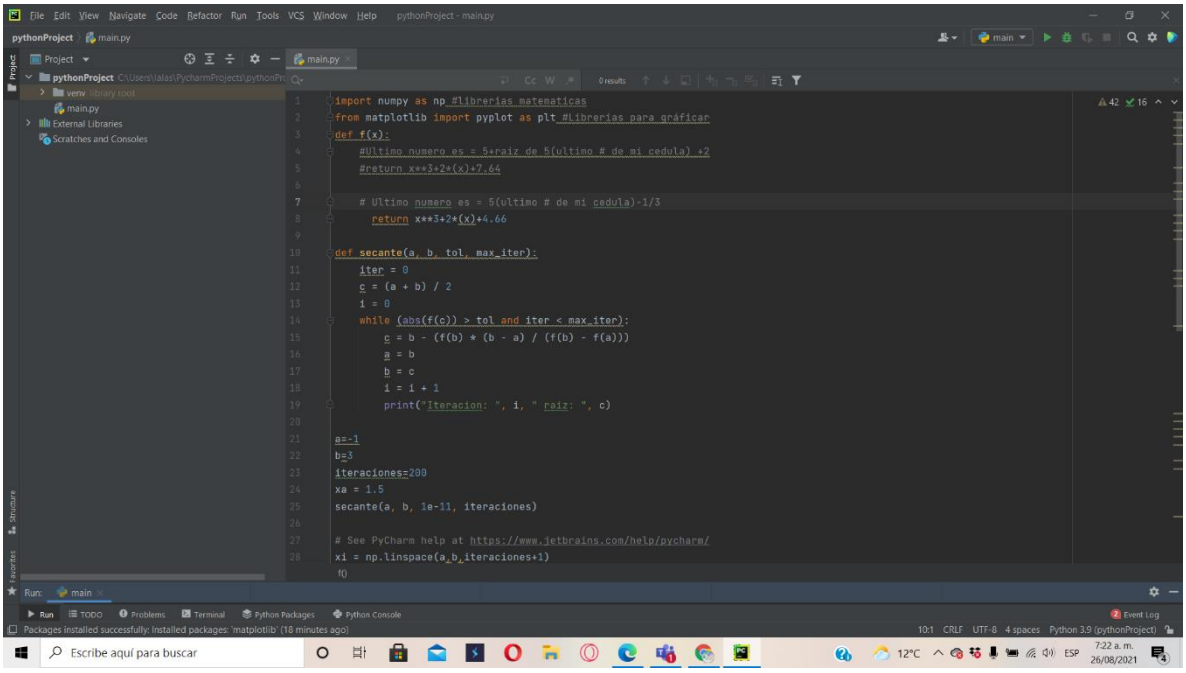


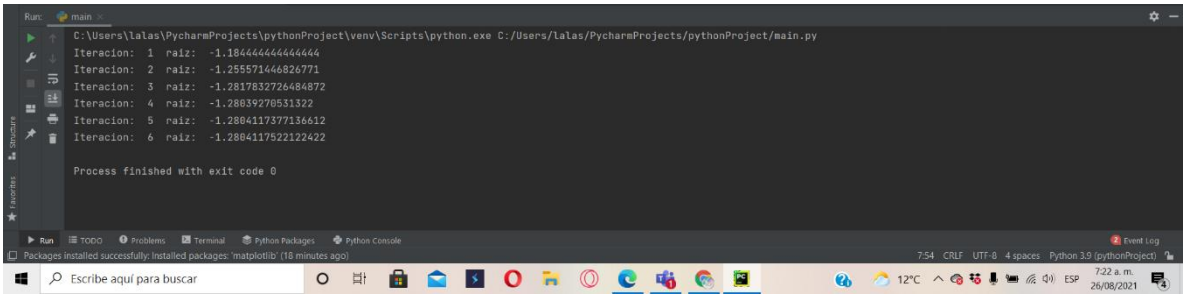

```
Iteracion:  2  raiz:  -1.644442989654184
Iteracion:  3  raiz:  -1.6343541566410515
Iteracion:  4  raiz:  -1.6349251863676604
Iteracion:  5  raiz:  -1.6349278526469482
Iteracion:  6  raiz:  -1.6349278518982033
```

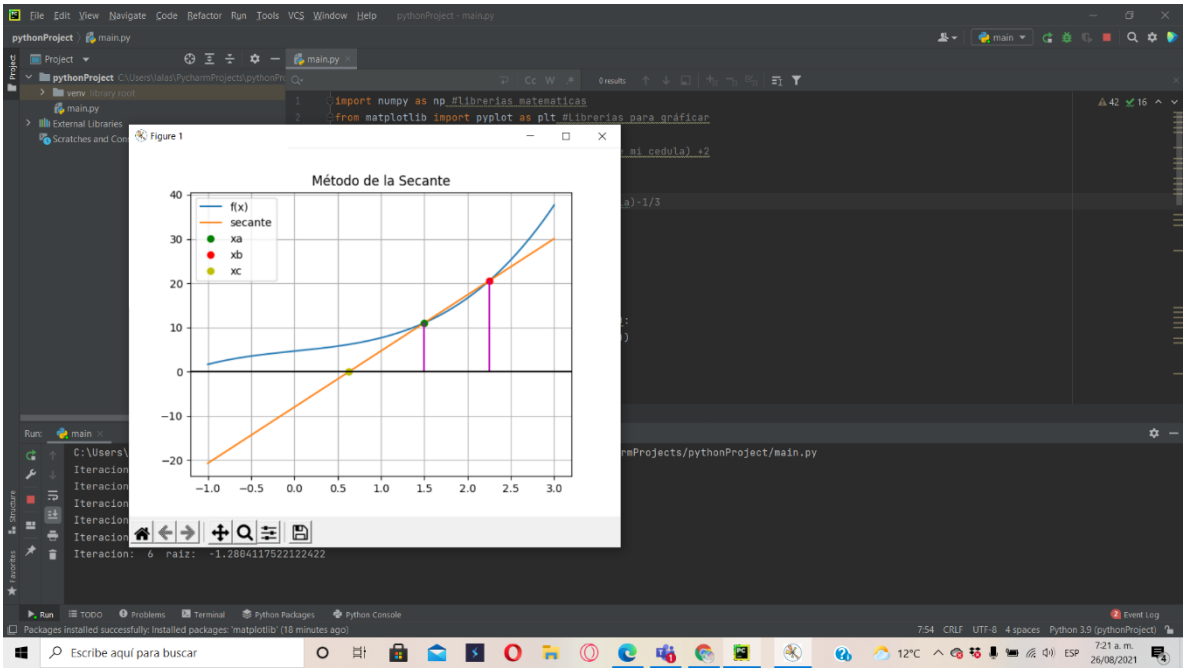
```
C:\Users\lalas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/lalas/PycharmProjects/pythonProject/main.py
Iteracion:  1  raiz:  -1.5155555555555553
Iteracion:  2  raiz:  -1.644442989654184
Iteracion:  3  raiz:  -1.6343541566410515
Iteracion:  4  raiz:  -1.6349251863676604
Iteracion:  5  raiz:  -1.6349278526469482
Iteracion:  6  raiz:  -1.6349278518982033

Process finished with exit code 0
```

Ejercicio cuando k =4.66

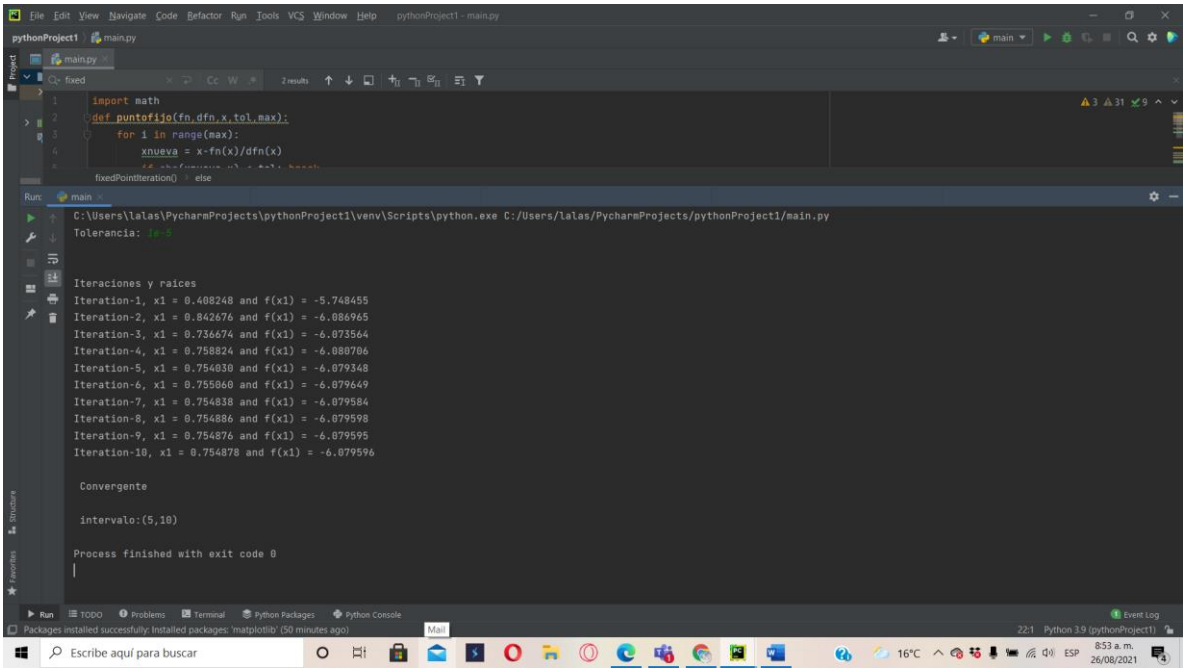






Punto 3:

c. $x\ 3 - 2x - 5$



Se necesitan 7 iteraciones para obtener aproximaciones precisas dentro de 10-5