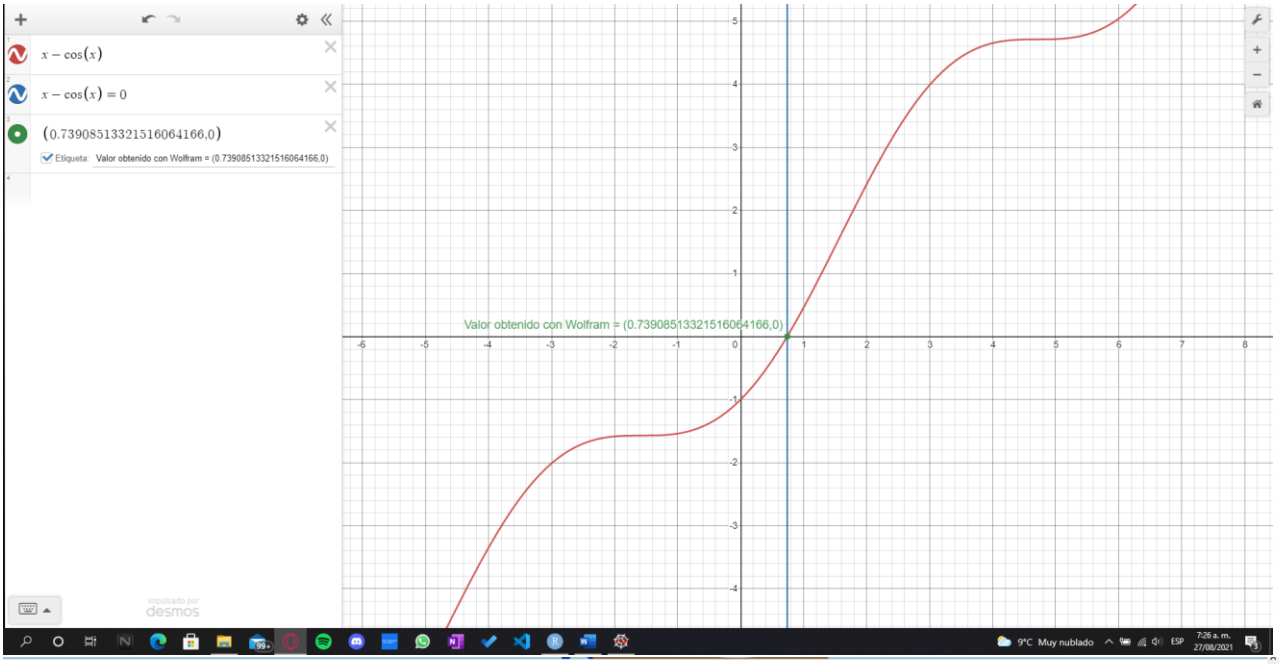
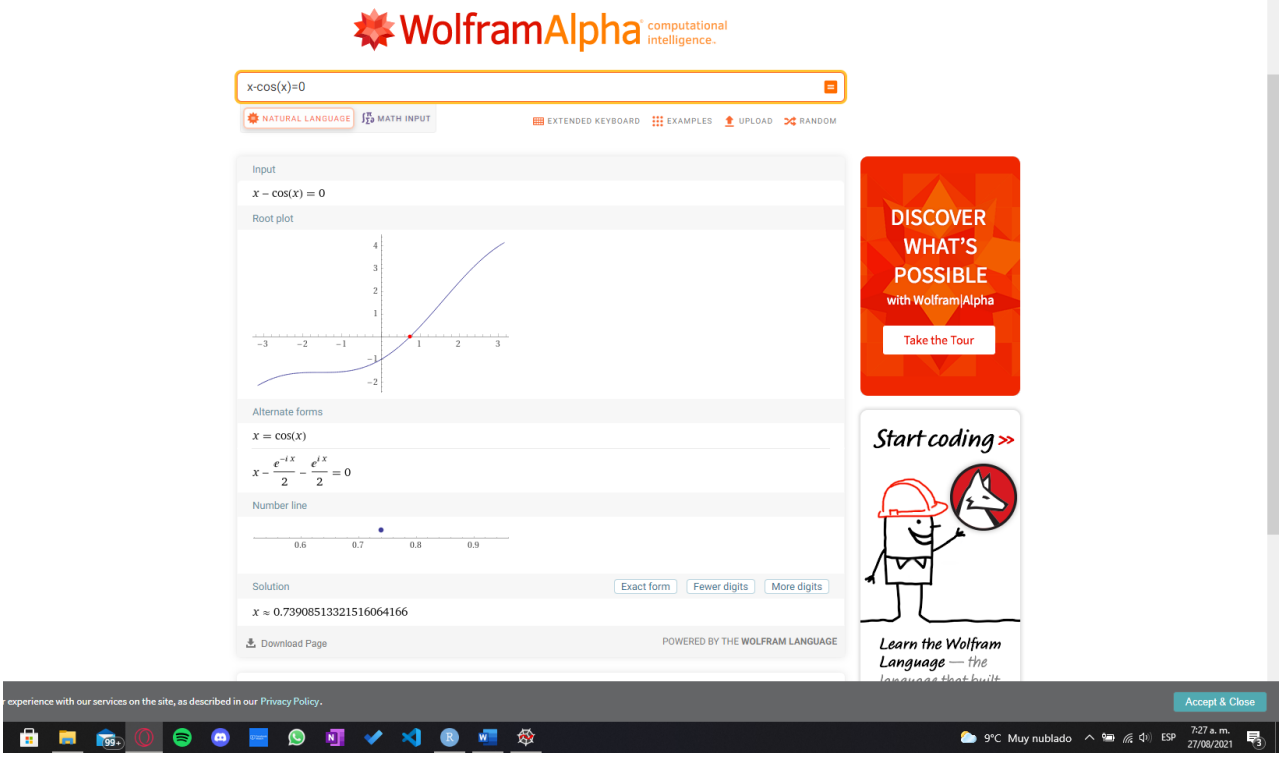
Parcial N°1 – Análisis numérico
Estudiante: Harold Duván Pinilla Salinas
Subnumeral seleccionado: 3b

Realizando una solución inicial mediante las herramientas WolframAlpha y Desmos para el subnumeral 3b del parcial se encontraron los siguientes resultados:

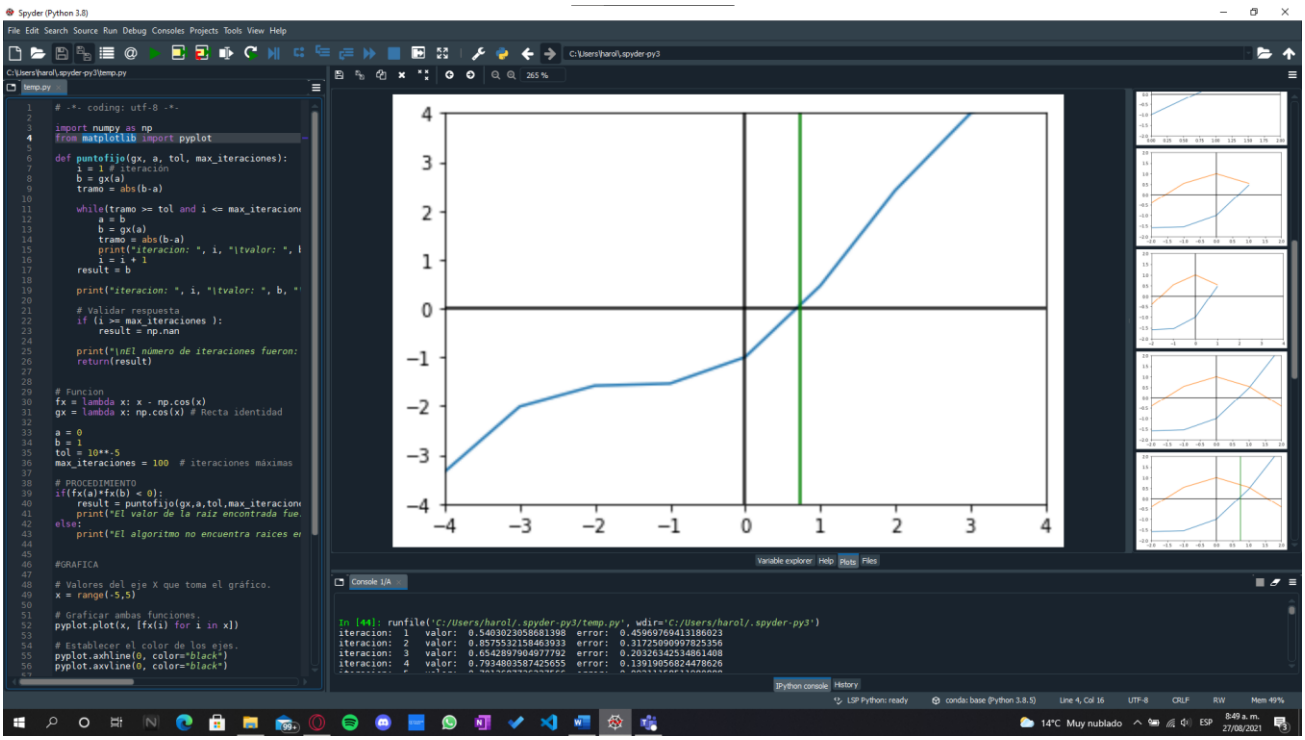




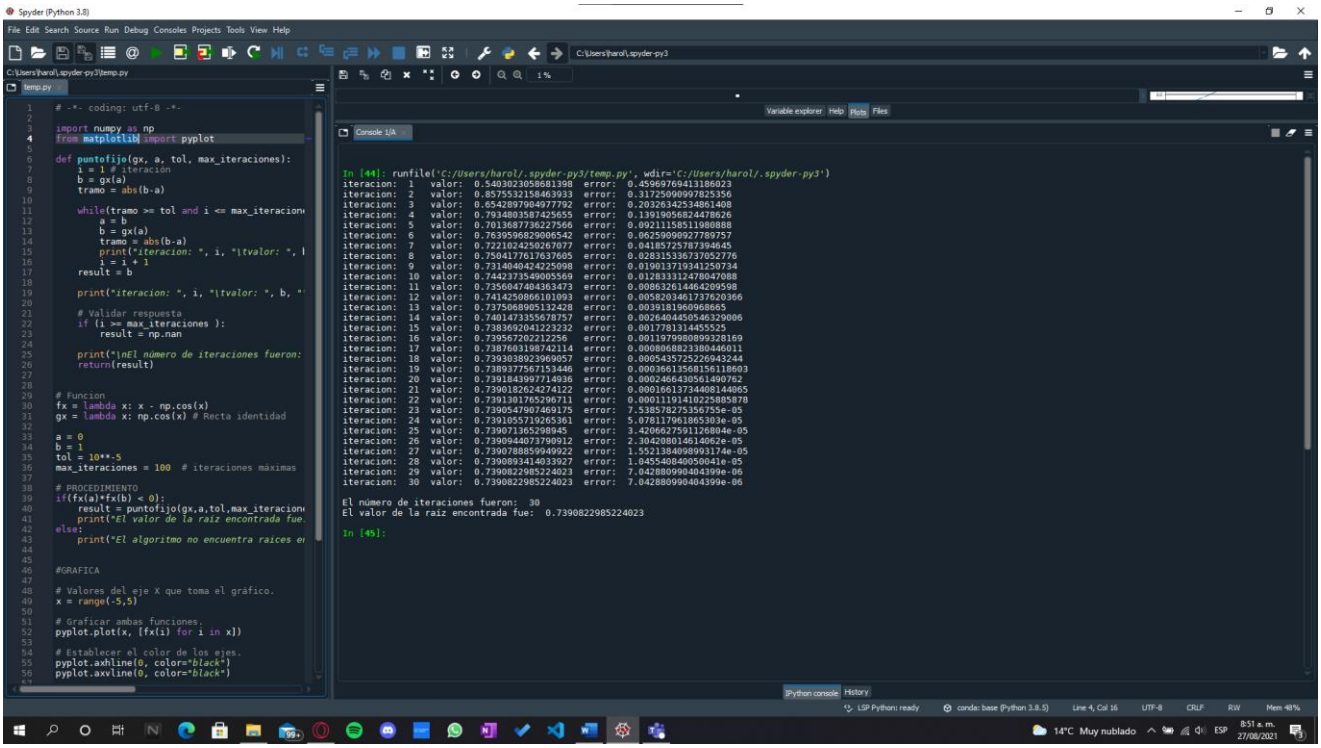
La implementación del método del punto fijo la realicé en Python y el código se encuentra adjunto en este repositorio.

Gracias a las aproximaciones dadas por las herramientas anteriores se pudo determinar que la función converge en todo intervalo que tenga el valor de la raíz real en él. Si en el intervalo dado no se halla este valor, el algoritmo realiza una impresión por pantalla advirtiendo de este hecho.

La grafica obtenida mediante la librería matplotlib de Python fue:



Luego mediante la impresión de las iteraciones, el valor temporal hallado de la raíz y el error se pudo obtener una tabla con los siguientes resultados:



```
# -*- coding: utf-8 -*-

import numpy as np
from matplotlib import pyplot

def puntofijo(gx, a, tol, max_iteraciones):
    i = 1 # iteración
    b = gx(a)
    tramo = abs(b-a)

    while(tramo >= tol and i <= max_iteracion
        a = b
        b = gx(a)
        tramo = abs(b-a)
        print("iteracion: ", i, "|tvalor: ", b, "
        i = i + 1
    result = b

    print("iteracion: ", i, "|tvalor: ", b, "

    # Validar respuesta
    if (i >= max_iteraciones ):
        result = np.nan

    print("\nEl número de iteraciones fueron:
    return(result)


# Funcion
fx = lambda x: x - np.cos(x)
gx = lambda x: np.cos(x) # Recta identidad

a = 0
b = 1
tol = 10**-5
max_iteraciones = 100  # iteraciones máximas

# PROCEDIMIENTO
if(fx(a)*fx(b) < 0):
    result = puntofijo(gx,a,tol,max_iteracion
    print("El valor de la raiz encontrada fue.
else:
    print("El algoritmo no encuentra raices en


#GRAFICA

# Valores del eje X que toma el gráfico.
x = range(-5,5)

# Graficar ambas funciones.
pyplot.plot(x, [fx(i) for i in x])

# Establecer el color de los ejes.
pyplot.axhline(0, color="black")
pyplot.axvline(0, color="black")
```

```
In [44]: runfile('C:/Users/harol/.spyder-py3/temp.py', wdir='C:/Users/harol/.spyder-py3')
iteracion:  1   valor:  0.5403023058681398  error:  0.45969769413186023
iteracion:  2   valor:  0.8575532158463933  error:  0.3172509099782535
iteracion:  3   valor:  0.6542897904977792  error:  0.2032634253486140
iteracion:  4   valor:  0.7934803587425655  error:  0.1391905682447862
iteracion:  5   valor:  0.7013687736227566  error:  0.0921115851980888
iteracion:  6   valor:  0.7639596829006542  error:  0.0625909092778975
iteracion:  7   valor:  0.7221024250267077  error:  0.0418572578739464
iteracion:  8   valor:  0.7504177617637605  error:  0.0283153367370527
iteracion:  9   valor:  0.7314040424225098  error:  0.0190137193412507
iteracion:  10  valor:  0.7442373549005569  error:  0.0128333124780470
iteracion:  11  valor:  0.7356047404363473  error:  0.0086326144662095
iteracion:  12  valor:  0.7414250866101093  error:  0.0058203461737620
iteracion:  13  valor:  0.7375068905132428  error:  0.0039181960968665
iteracion:  14  valor:  0.7401473355678757  error:  0.0026404455054632
iteracion:  15  valor:  0.7383692041223232  error:  0.0017781314455525
iteracion:  16  valor:  0.7395672022122562  error:  0.0011979980899328
iteracion:  17  valor:  0.7387603198742114  error:  0.0008068823380446
iteracion:  18  valor:  0.7393038923969657  error:  0.0005435725226943
iteracion:  19  valor:  0.7389377567153446  error:  0.0003661356815611
iteracion:  20  valor:  0.7391843997714936  error:  0.0002466430561490
iteracion:  21  valor:  0.7390182624274122  error:  0.0001661373440814
iteracion:  22  valor:  0.7391301765296711  error:  0.0001119141022588
iteracion:  23  valor:  0.7390547907469175  error:  7.538578275356755e-05
iteracion:  24  valor:  0.7391055719265361  error:  5.078117961865303e-05
iteracion:  25  valor:  0.7390713652980945  error:  3.420662759112680
iteracion:  26  valor:  0.7390944073790912  error:  2.304208014614062e-05
iteracion:  27  valor:  0.7390788859949922  error:  1.552138409899317
iteracion:  28  valor:  0.7390893414033927  error:  1.045540840050041e-05
iteracion:  29  valor:  0.7390822985224023  error:  7.042880990404399e-06
iteracion:  30  valor:  0.7390822985224023  error:  7.042880990404399e-06

El número de iteraciones fueron:  30
El valor de la raiz encontrada fue:  0.7390822985224023

In [45]:
```

Finalmente, con este algoritmo se pudo hallar una aproximación convergente al valor de la raíz verdadero con una tolerancia de 10^{-5} , en el intervalo [0, 1] y con un total de iteraciones de 30