CODIGO PYTON

import numpy as np

# Definir las matrices y el vector del sistema

A = np.array([[52, 30, 18],

[20, 50, 30],

[25, 20, 55]])

b = np.array([4800, 5810, 5690])

# Inicialización

x = np.zeros(len(b)) # Vector de soluciones inicializado a cero

max\_iter = 100 # Número máximo de iteraciones

tolerance = 1e-5 # Tolerancia para la convergencia

# Método de Gauss-Seidel

for iteration in range(max\_iter):

x\_old = x.copy()

for i in range(len(b)):

sum1 = np.dot(A[i, :i], x[:i]) # Suma de A[i][j] \* x[j] para j < i

sum2 = np.dot(A[i, i+1:], x\_old[i+1:]) # Suma de A[i][j] \* x\_old[j] para j > i

x[i] = (b[i] - sum1 - sum2) / A[i, i] # Actualización del valor de x[i]

# Verificar la convergencia

if np.linalg.norm(x - x\_old, ord=np.inf) < tolerance:

break

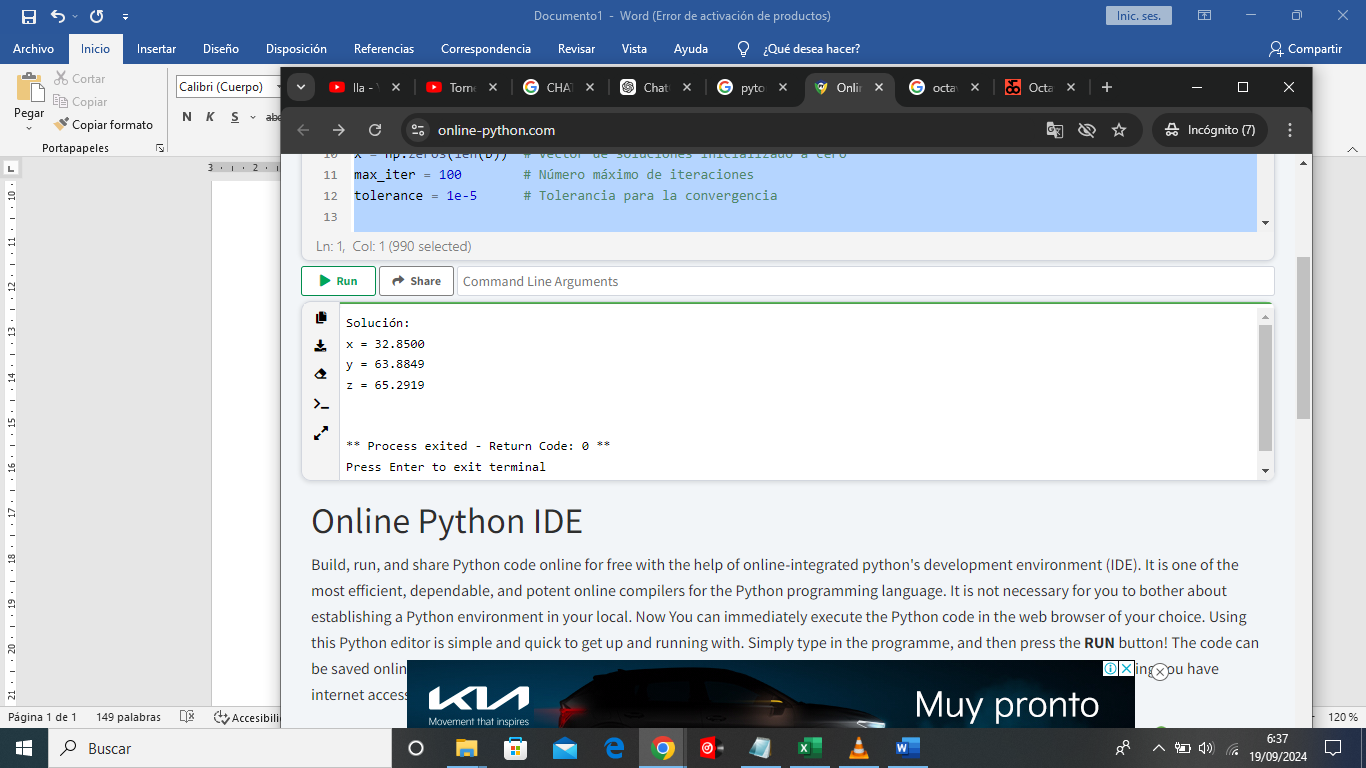
# Mostrar los resultados

print("Solución:")

print(f"x = {x[0]:.4f}")

print(f"y = {x[1]:.4f}")

print(f"z = {x[2]:.4f}")



CODIGO EN OCTAVE

A = [52, 30, 18;

20, 50, 30;

25, 20, 55];

b = [4800; 5810; 5690];

x = zeros(length(b), 1); % Vector de soluciones inicializado a cero

max\_iter = 100; % Número máximo de iteraciones

tolerance = 1e-5; % Tolerancia para la convergencia

% Método de Gauss-Seidel

for iteration = 1:max\_iter

x\_old = x; % Guardar los valores anteriores

for i = 1:length(b)

sum1 = A(i, 1:i-1) \* x(1:i-1); % Suma de A[i][j] \* x[j] para j < i

sum2 = A(i, i+1:end) \* x\_old(i+1:end); % Suma de A[i][j] \* x\_old[j] para j > i

x(i) = (b(i) - sum1 - sum2) / A(i, i); % Actualización de x[i]

end

% Verificar la convergencia

if max(abs(x - x\_old)) < tolerance

break;

end

end

% Mostrar los resultados

printf("Solución:\n");

printf("x = %.4f\n", x(1));

printf("y = %.4f\n", x(2));

printf("z = %.4f\n", x(3));

Solución:

x = 32.8500

y = 63.8849

z = 65.2919