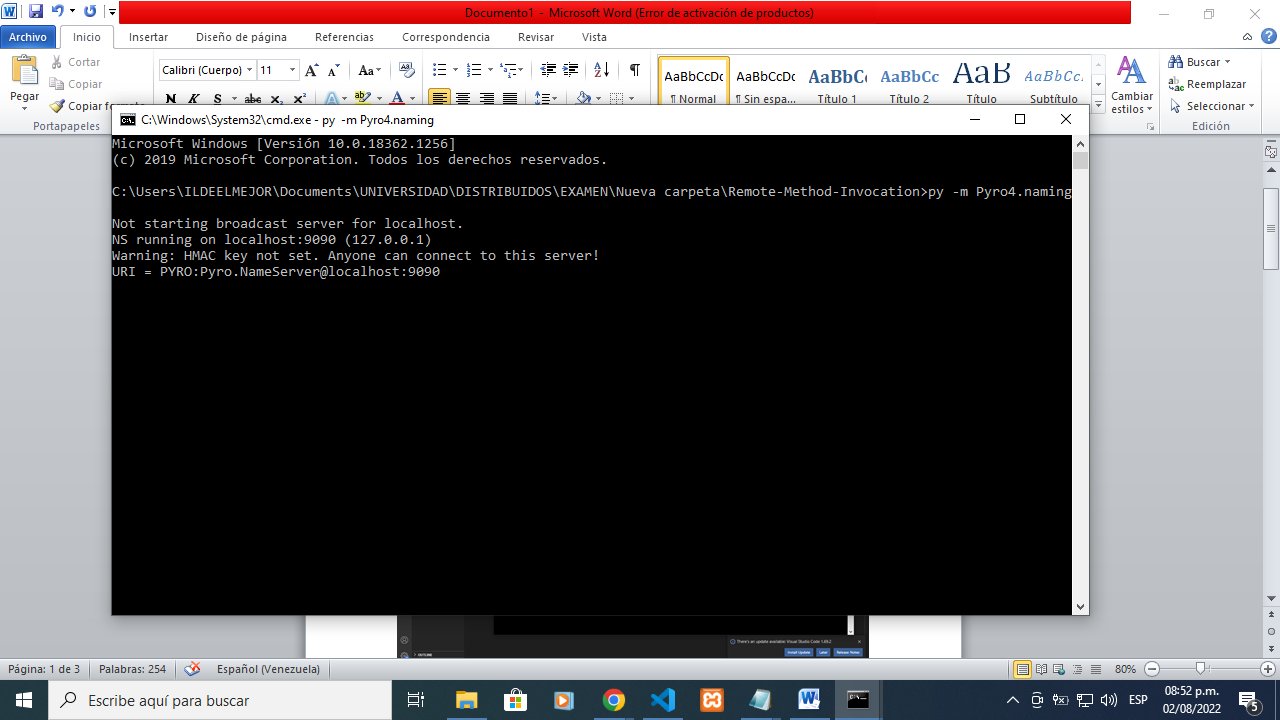
Implemente el algoritmo del Colector de Basura Distribuida. Puede usar python o Go. Entregue un documento donde explica paso a paso el funcionamiento del programa implementado.

Coloque los fuentes de su respuesta en Github o Gitlab. Incluya un audio con su explicación

Nota: la implementación que se solicita es simular o emular el comportamiento de este algoritmo. Debe usar una arquitectura cliente-servidor

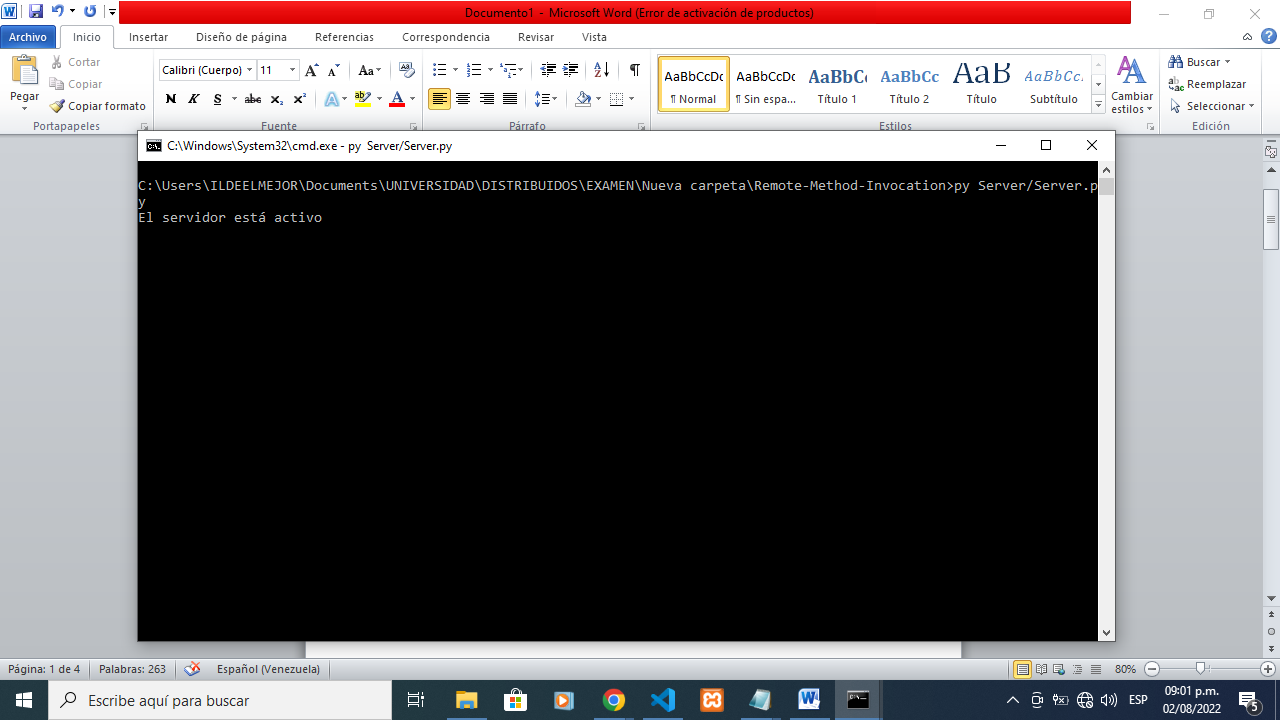
Paso 1:

Primero necesitamos correr el siguiente comando en una consola



Paso 2:

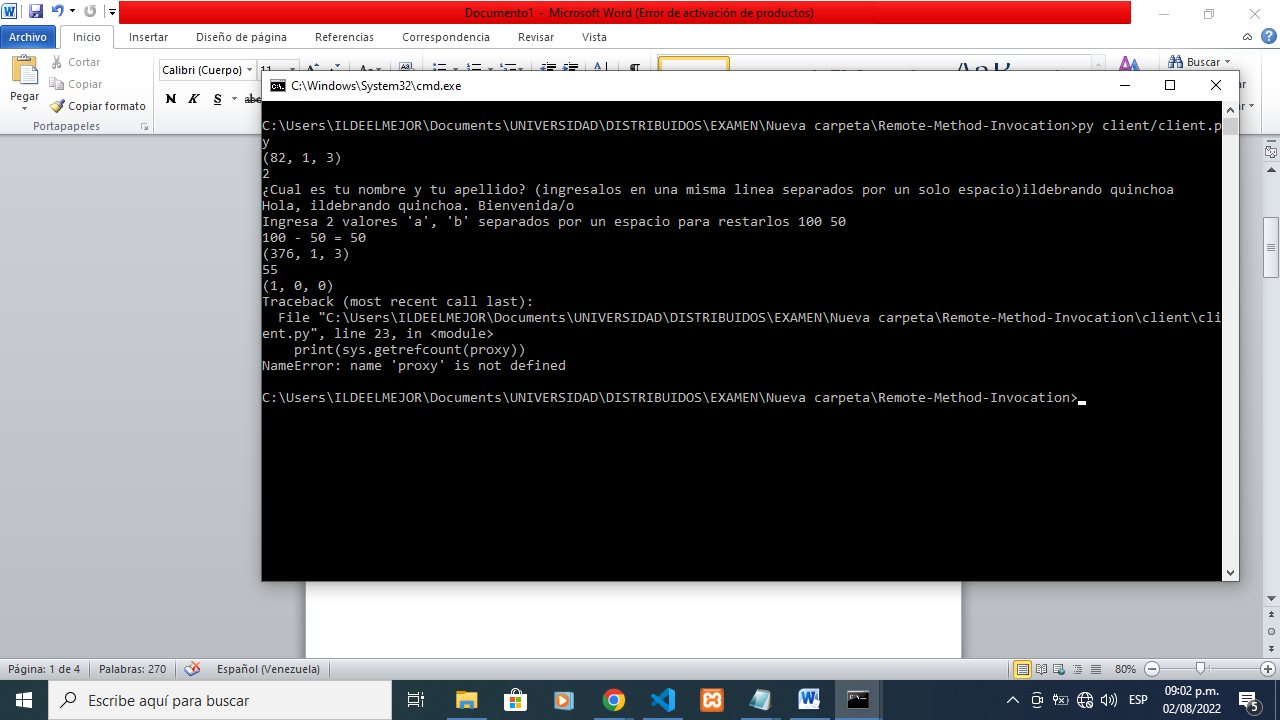
Iniciamos el servidor

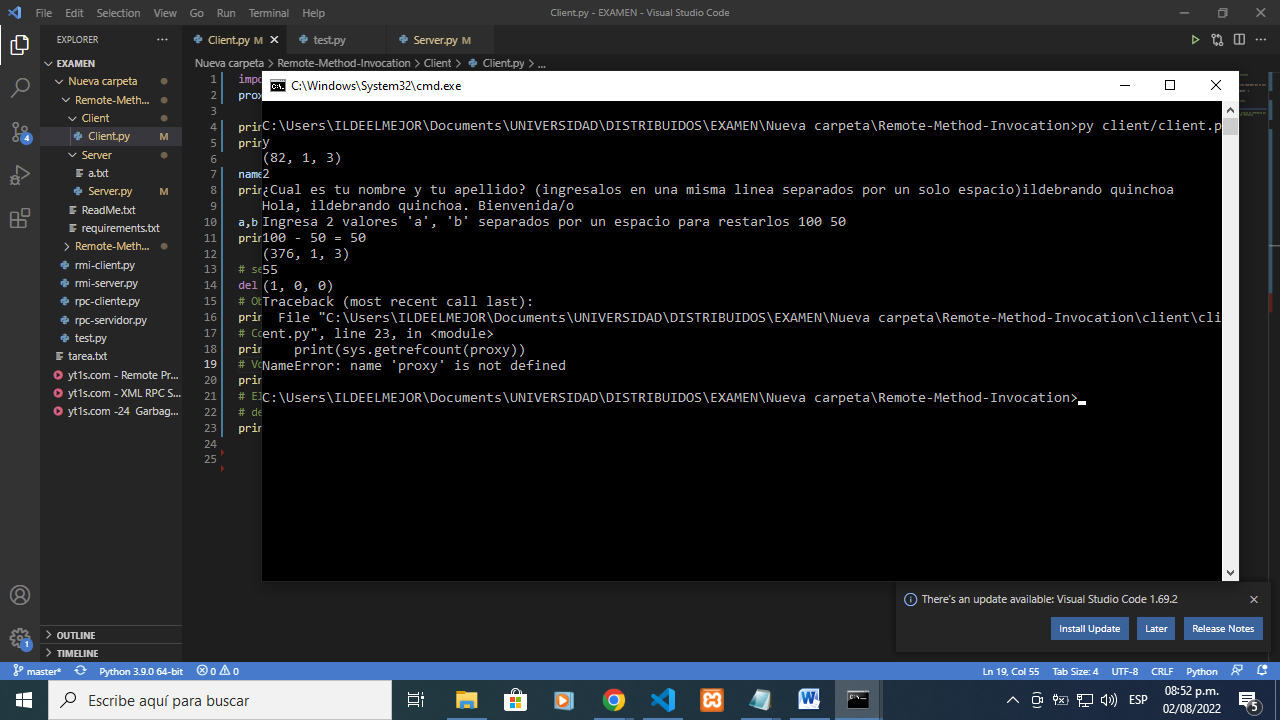


Paso 3:

Resultados. Aquí ingresamos los parámetros necesarios para las funciones en el servidor rmi.

Observamos la memoria ocupada y vaciamos usando el colector de basura. En la ultima línea se imprime proxy no esta definido ya que al borrar el proxy el colector de basura de django automáticamente borra la memoria asignada a este.





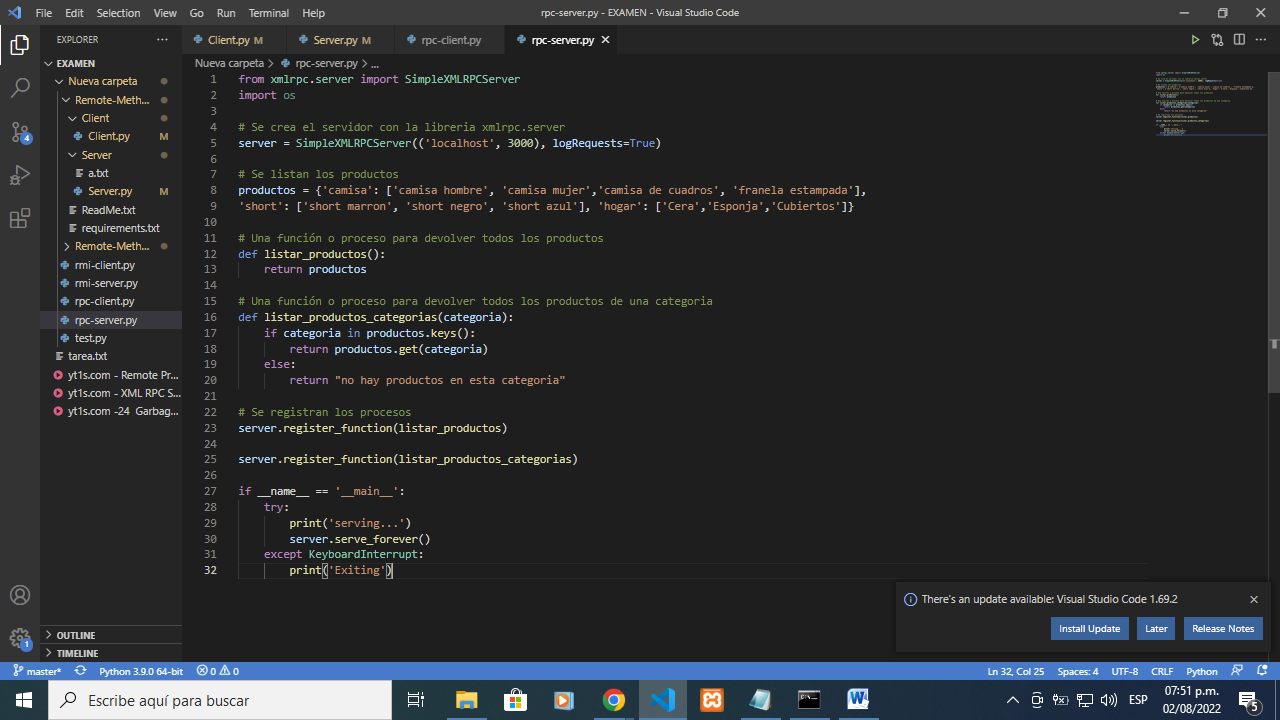
Implemente una aplicacion donde el cliente consulte un servicio de consulta de productos. Use RPC.

Identifique tanto en el cliente como en el servidor, los componentes de RPC. Entregue un documento donde explica paso a paso cada uno de las partes del protocolo.

Coloque las fuentes disponibles en Github o GitLab. Agregue una nota de voz con su explicación.

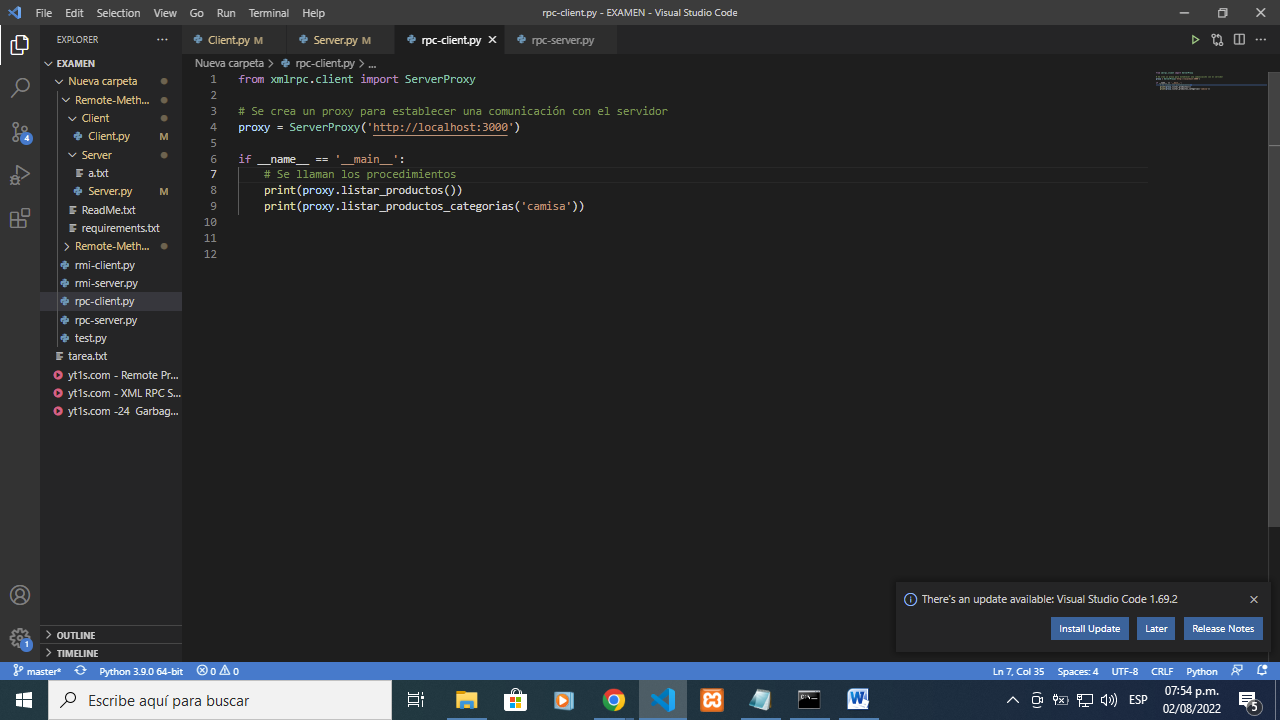
Paso 1:

Con el uso de la librería xmlrpc creamos un servidor RPC. En este archivo también listamos los productos y los procedimientos para retornar todos los productos y para retornarlos de acuerdo al parámetro que se les pase.



Paso 2:

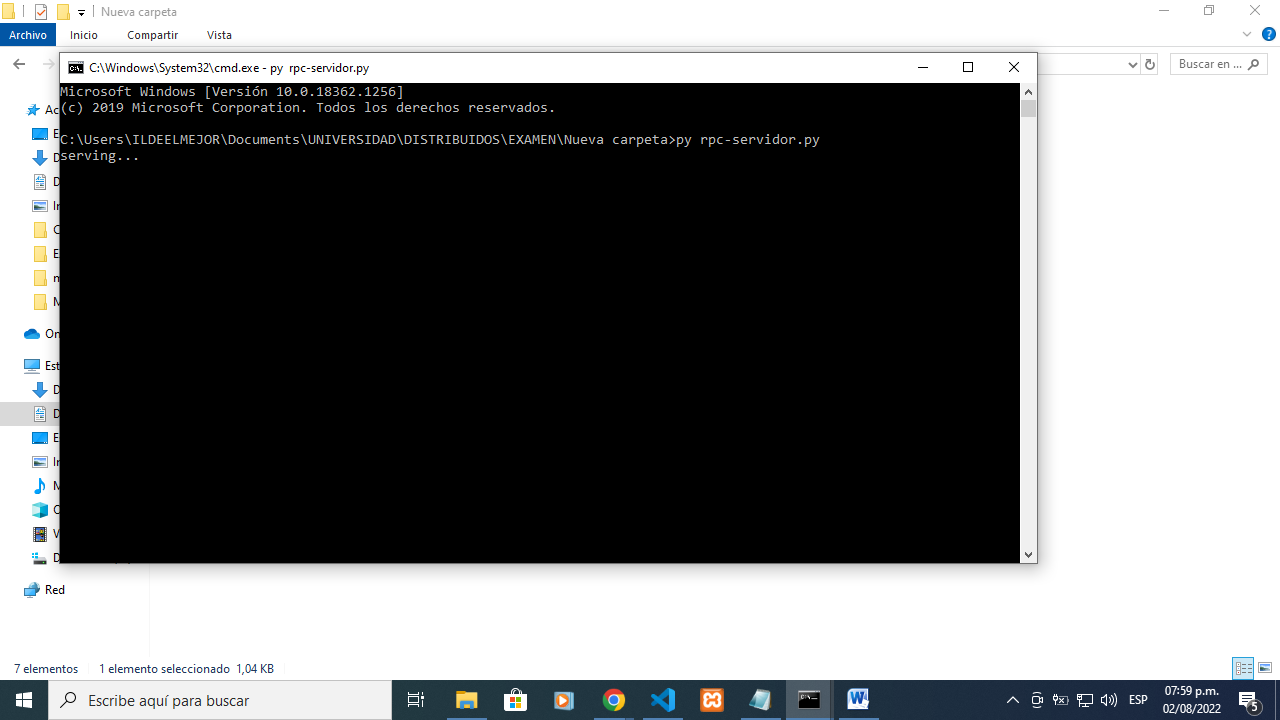
Con el uso de la misma librería se crea el lado del cliente. En este establecemos el proxy a la dirección del servidor. Luego imprimimos los resultados de los procedimientos haciendo los llamados de los procedimientos creados en el primer paso.



Paso 3:

Se inicia el servidor ubicándose en la carpeta y realizando el comando

py rpc-servidor.py



Paso 4:

Una vez el servidor haya sido inicado desde una consola de comando diferentes se inicia el archivo de cliente donde este llama las funciones del servidor y devuelve los resultados esperados.

