UNIVERSIDAD MAYOR DE SAN SIMÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA INFORMATICA

**AGENTES INTELIGENTES**

**PROYECTO FINAL**

ELABORADO POR: Gerson Egues Choque Loreño.

DOCENTE: Msc. Lic. Montoya Burgos Yony Richard

PERIODO: II-2019

COCHABAMBA - BOLIVIA

INFORME FINAL

Funcionamiento:

* Ejecutar antes de todo el martillito, para que todo quede compilado, mejor si compilamos archivo por archivo.
* Ejecutar el archivo “Main”
* Tenemos 1 servidor y 2 clientes
* Los clientes leen el video frame por frame envían el nombre de la imagen al servidor cuando agarran una imagen si esta no fue procesada por ningún otro cliente, para que el servidor diga a los demás clientes que ya no pueden procesar esa imagen.
* Los clientes envían un mensaje al servidor cuando terminan de procesar una imagen, envían el resultado de dicha imagen y se imprime en la terminal
* Las imágenes se escriben para que se pueda comparar los resultados escritos en la terminal por el servidor
* Cuando un cliente termina de leer el video el programa cliente termina, pero no el servidor y tampoco el programa principal “Main”, pero como como nosotros sabemos cuántas imágenes distintas tiene el video, una vez obtenido todos los resultados, terminamos el programa por fuerza bruta.
* En la línea 57, en el método “resolverProblema”, podemos modificar el nombre del método al que llama y el color para obtener las distintas figuras geométricas.

def resolverProblema(self,img):  
 return self.getCuadrado(img,'azul') #azul, amarillo, rojo

tenemos los siguientes métodos:

def getTriangulo(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,3,0.03)  
  
def getCuadrado(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,4,0.03)  
  
def getCirculo(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,8,0.03) - self.get2Circulos(imagen,color) - self.get2cuadrados(imagen,color)  
  
def getEstrella(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,10,0.03)

#figuras complejas

def get2cuadrados(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,8,0.012)  
  
def get2Circulos(self,imagen,color):  
 return self.getFiguraGeometrica(imagen,color,11,0.014)

tenemos los siguientes colores: azul, rojo, amarillo

* Los resultados parciales de cada imagen salen escritos en la terminal de la siguiente forma:

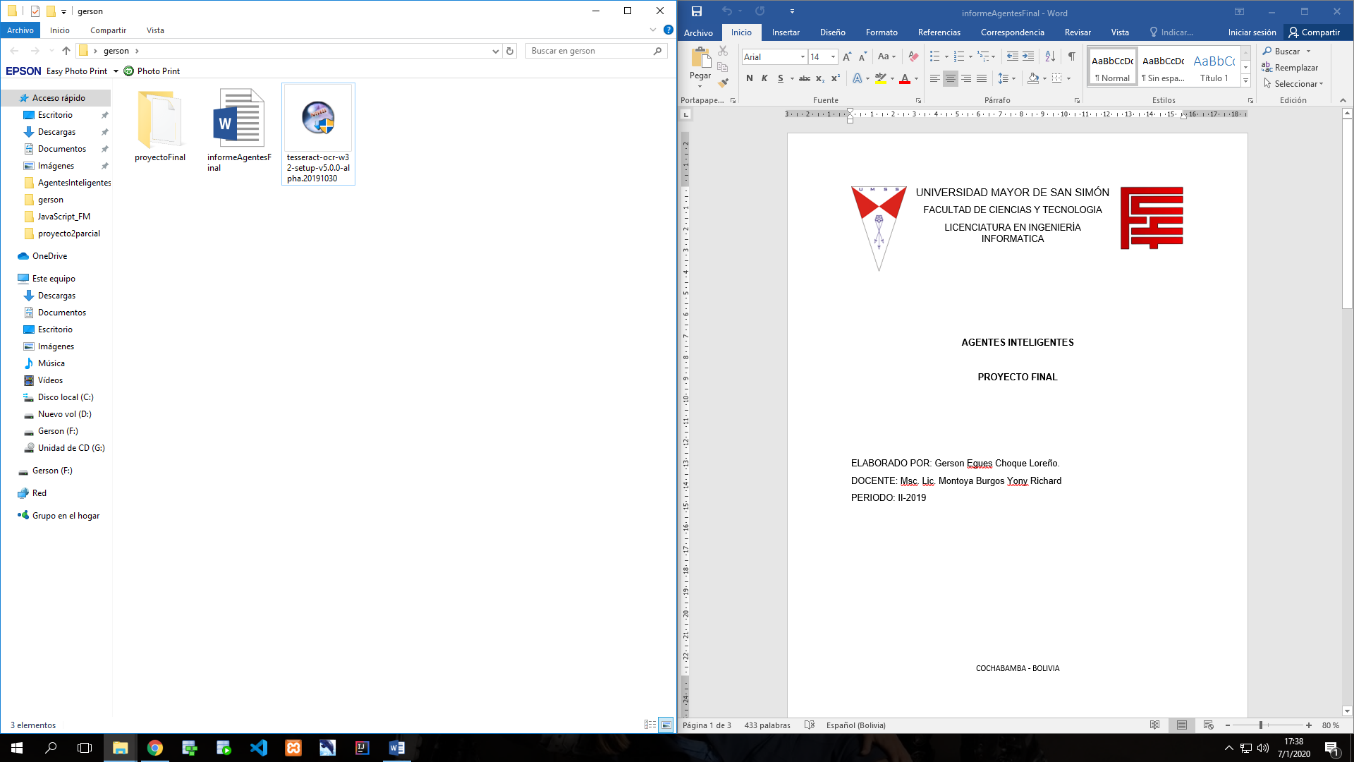
\*\* server \*\* resultado de procesar imagen por id cliente …………. --> 1

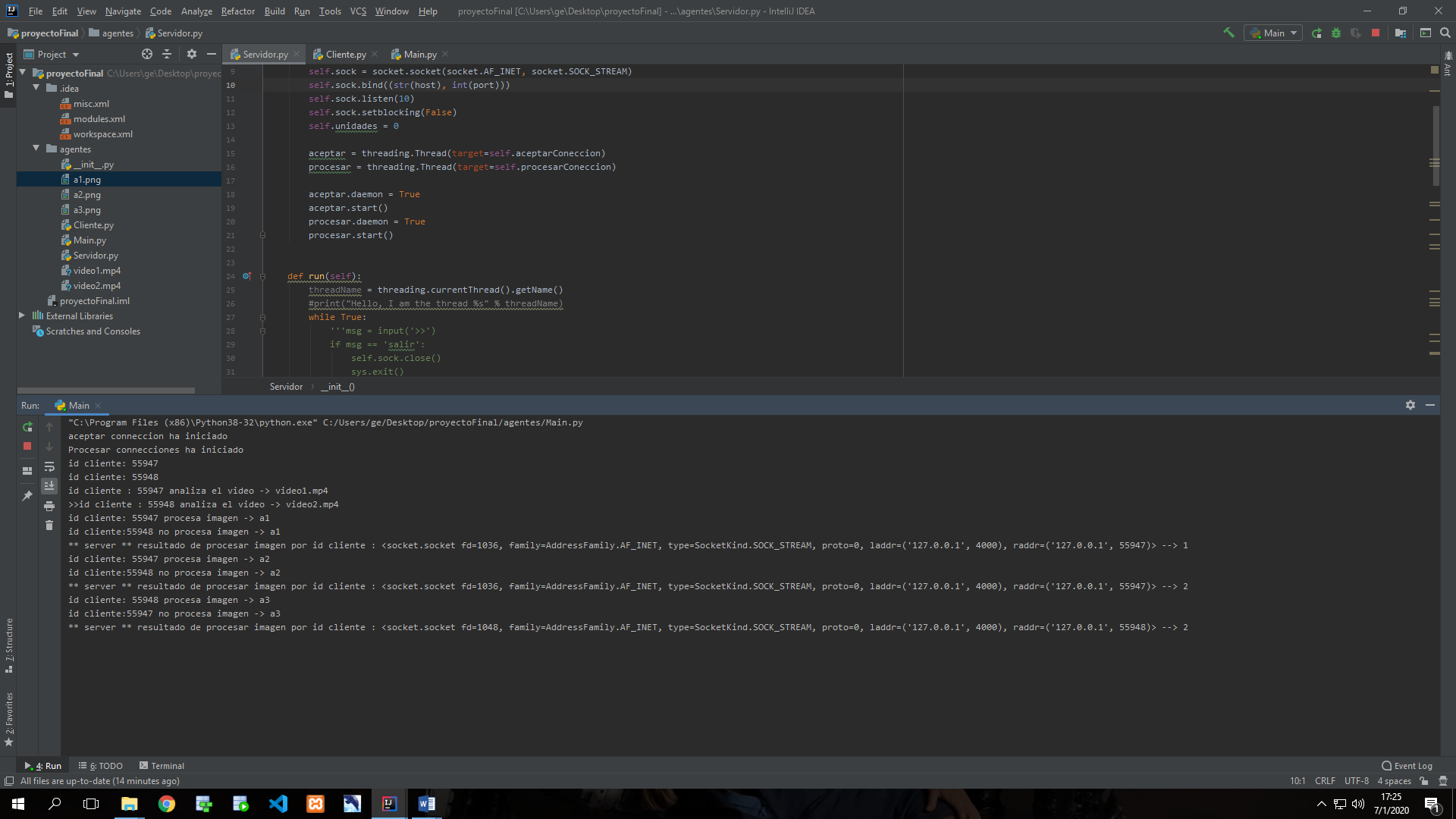
**Requisitos: (windows)**

* Python, con el módulo PIP, en su última versión (actualizar: >> pip install --upgrade pip )
* Numpy: >> pip install numpy
* Matplotlib: >> pip install matplotlib
* Opencv: >> pip install opencv-contrib-python
* Pytesseract: >> pip install pytesseract

Ese es el programa: “tesseract-ocr-w32-setup-v5.0.0-alpha.20191030”

* Instalar el software poner en las variables de entorno la dirección de donde se instaló: C:\Program Files (x86)\Tesseract-OCR

(no pude adjuntar a este proyecto el programa porque pesa 41 MB, y Moodle solo acepta 20MB) Intellij Idea (IDE)

Los resultados deberían salir de esta manera:

En las hileras grandes están los resultados parciales de cada imagen, que también se escriben en el archivo

Una hilera grande por cada imagen distinta a las demás, si no sale de esta manera volver a compilar todo con el martillito, porfavor.