|  |  |  |
| --- | --- | --- |
| T 3 | MARIA DEL ROCIO DIAZ FLORES  ARTURO ROJAS PORTILLA  MINERÍA DE DATOS  REPORTE | 8 C |

**INTRODUCCIÓN**

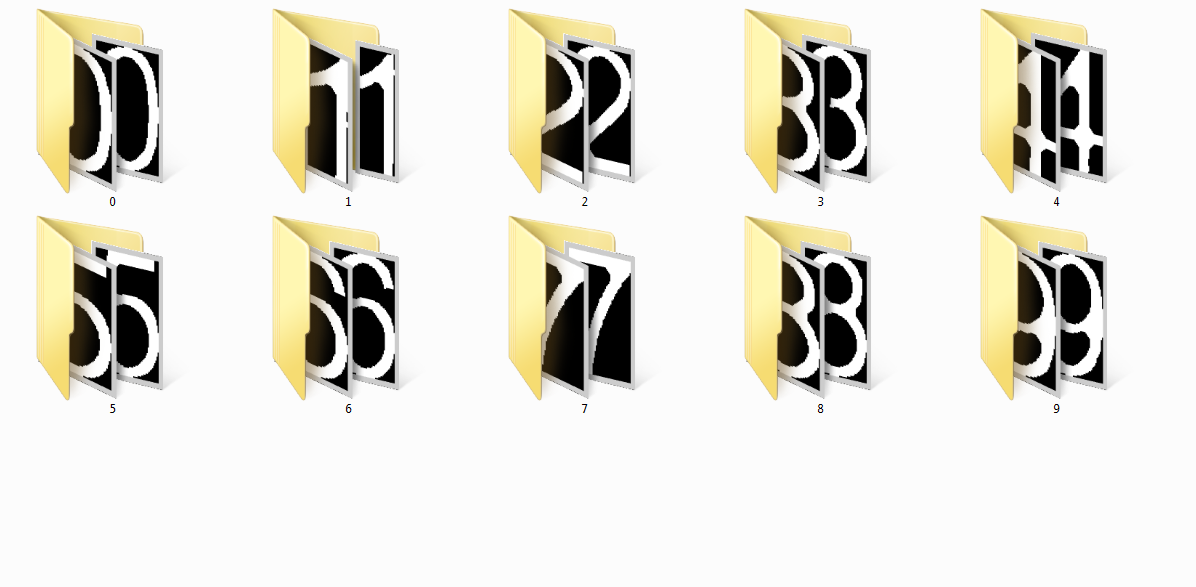
Un sistema OCR es un tipo de software que se encarga de reconocimiento óptico de caracteres. Se encarga de extraer de una imagen los caracteres de un texto y los guarda en un formato que pueda editarse como texto.

**REQUERIMIENTOS**

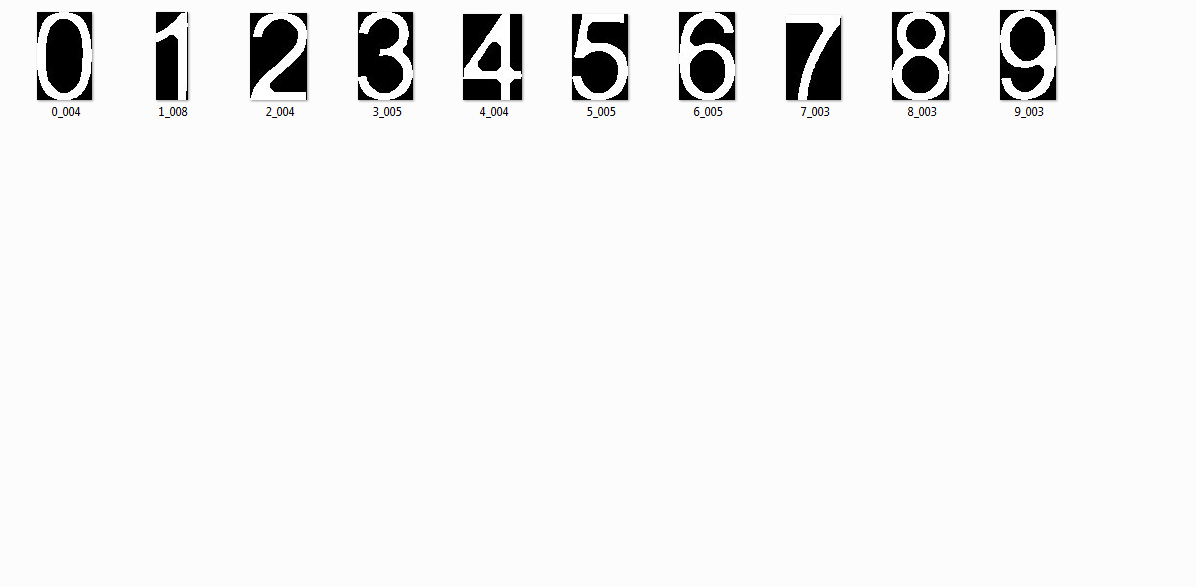
* Anaconda 2.4.1 - Spyder
* Python 3.5.1 - Excel

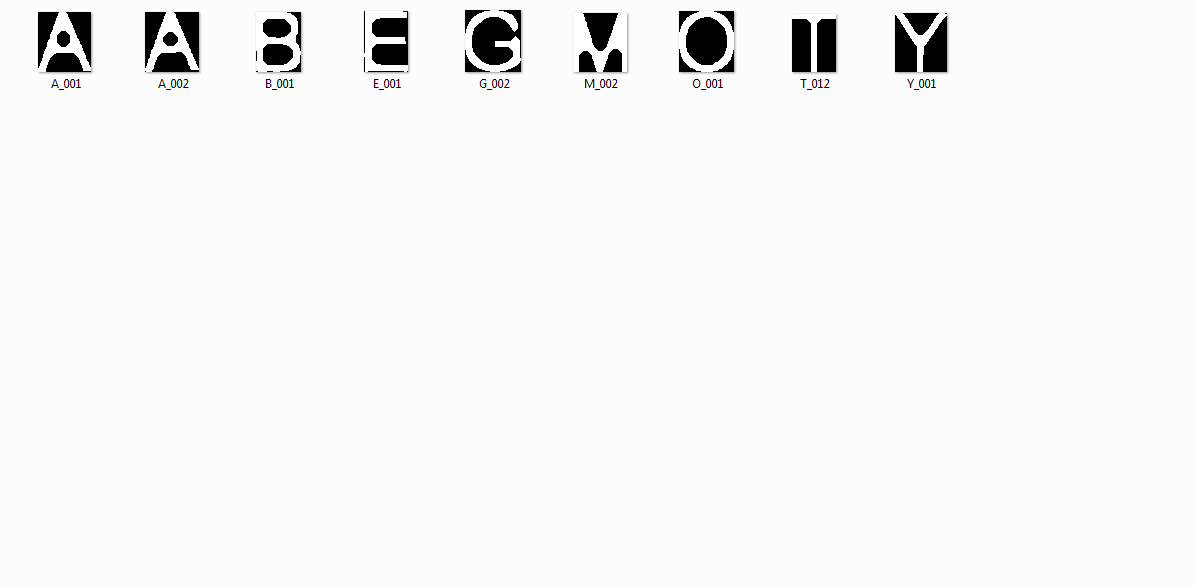
**CONJUNTO DE IMÁGENES**

El conjunto de imágenes está conformado por 10 carpétas con numeros de 0 al 9 (imágen 1), cada carpeta contiene un aproximado de 238 imágenes binarias con un total aproximado de 2,380 imágenes y 26 carpetas de la A a la Z (imágen 2), cada carpeta con un aproximado de 460 imágenes.

  
**Imagen 1. Carpetas numeradas del 0 al 9, contienen imágenes binarias**.  
**Imagen 2. Carpetasde la A a la z.**

Cada imagen mide aproximadamente 53x85 píxeles, son aproximadamente 300 imágenes por clase, las clases a utilizar son las antes vistas; del 0 al 9 y de la A a la Z en total son 36 clases.



  
**Imagen 3. Imágenes binarias contenidas en las carpetas.**

**CREAR DATASET**

Los siguientes pasos son para la creación del DataSet:

1: Leer carpetas con las imágenes contenidas.

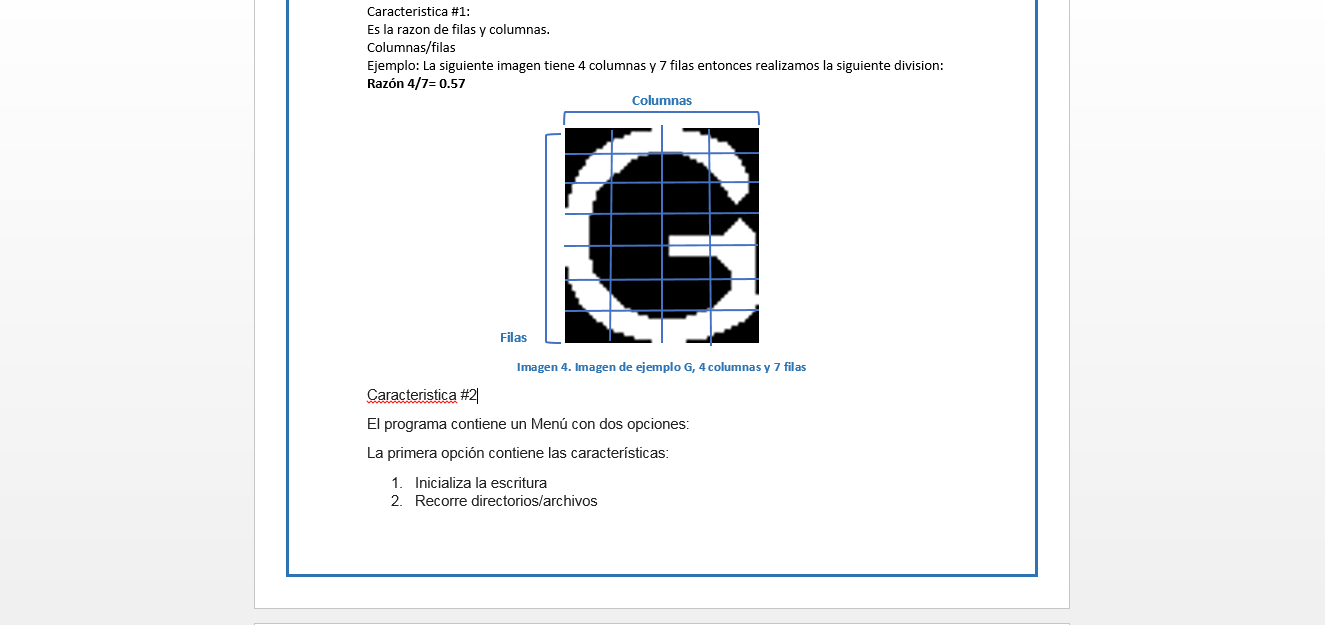
2: Por cada carpeta se lee cada imagen.

3: Se obtienen las características de cada imagen y se escriben en un documento CSV.

**CARACTERÍSTICAS DE LAS IMÁGENES**

**Caracteristica #1:**

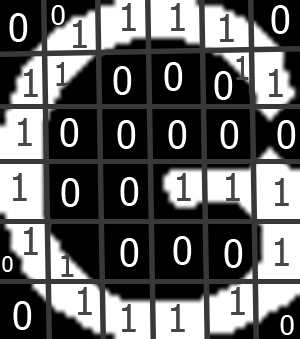
Es la razon de filas y columnas.  
Columnas/filas   
Ejemplo: La siguiente imagen tiene 4 columnas y 7 filas entonces realizamos la siguiente division:  
**Razón 4/7= 0.57**



**Filas**

**Imagen 4. Imagen de ejemplo G, 4 columnas y 7 filas**

**Característica #2:**Área de la imagen   
Se calcula el número de unos que hay en la imagen y se divide entre la razón  
Ejemplo: La imagen tiene 6 filas y 6 columnas (imagen 5), por lo tanto el área de la imagen es 6\*6 y su área es 36. **A=6\*6 A=36**  
Ahora se recorre la imagen y se van contabilizando los 1 que existen en la imagen y se divide entre el área de la imagen. **Número de unos/área de la imagen.**

  
**Imagen 5. Imagen de ejemplo G, 6 columnas y 6 filas, identificando 0’s y 1’s.**

**Características #3, 4, 5, 6, 7, 8**

Recorremos el vector contando los unos y al final se divide entre el tamaño del vector

Razón = 1’s /tamaño del vector

Vector # #1 #2 #3

(col/4) (col/2) (col/4)



#4   
  
 (col/4)

#5

(col/2)

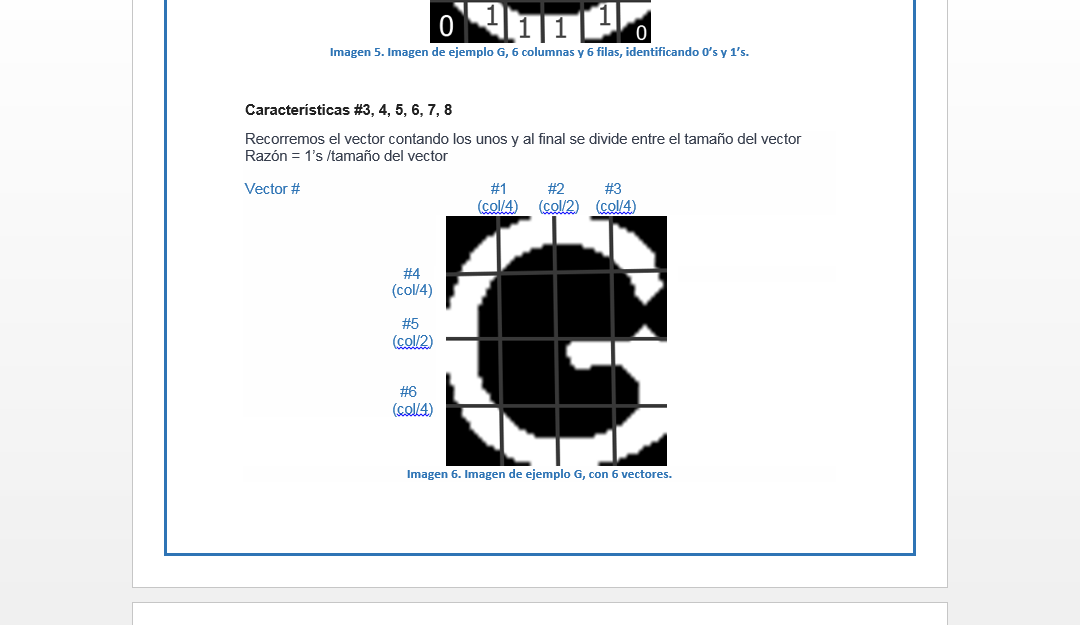
#6

(col/4)  
**Imagen 6. Imagen de ejemplo G, con 6 vectores.**

**Características # 9, 10, 11,12, 13, 14**

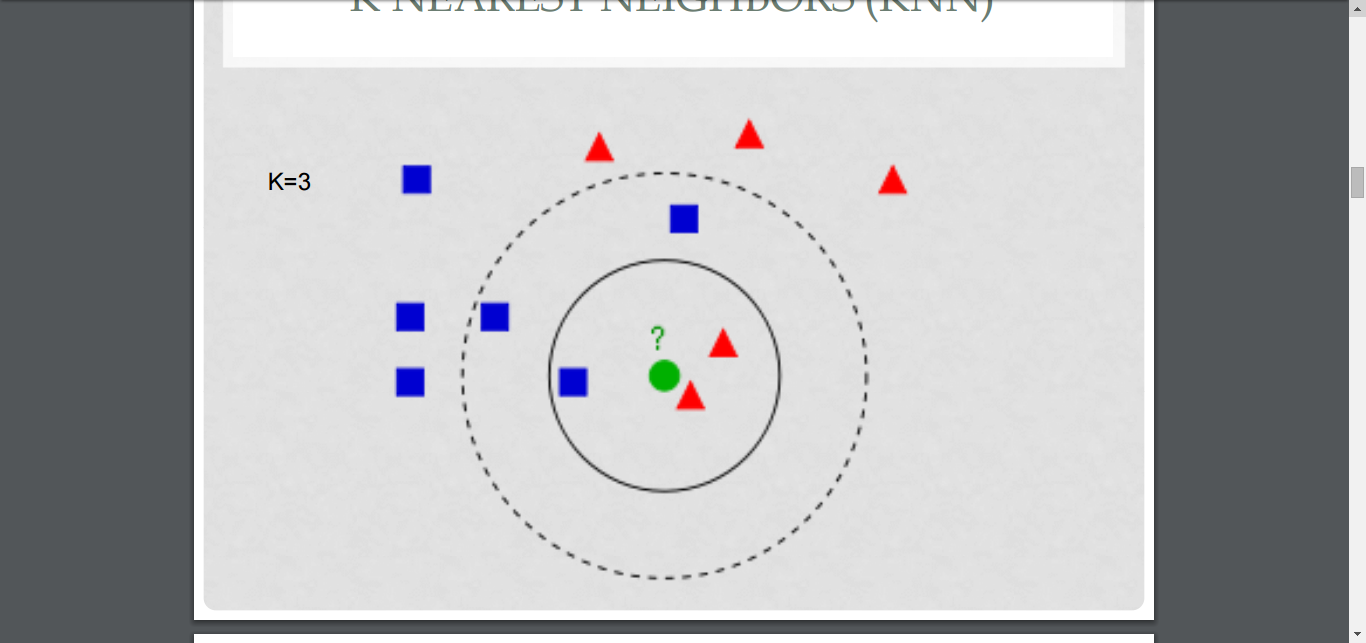
Se cuentan los cambios de 0 a 1 y viceversa en los vectores (imagen 7).

La característica es el número de cambios o cortes que hay en la imagen.

  
 **Imagen 7. Imagen de ejemplo G, con 6 vectores.**

**FORMA DE CLASIFICAR**

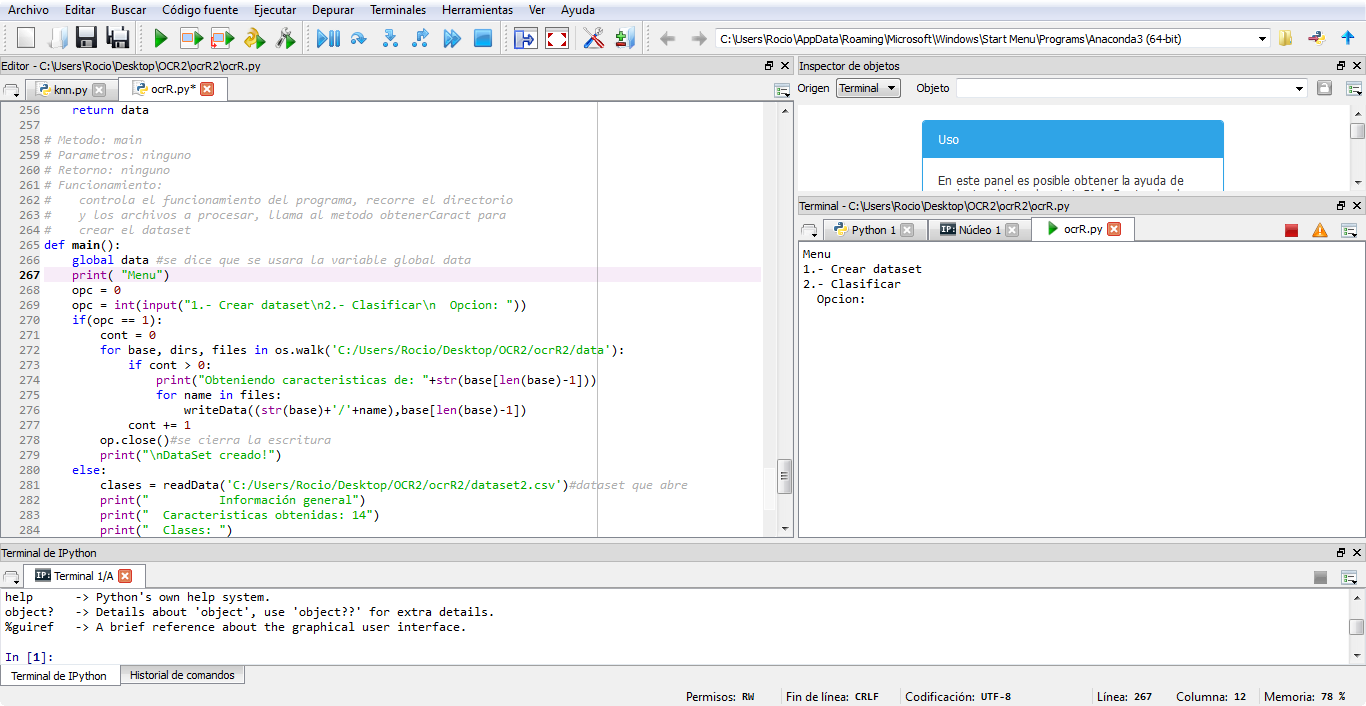
Ya se ha generado el DataSet con las características y necesitamos un método de clasificación. Se utiliza el método KNN que es un método de clasificación y hace la clasificación dependiendo de las características obtenidas de cualquier instancia.   
Se debe ingresar una nueva instancia con características propias y tenemos que elegir un numero k de instancias a comparar. Este número k, debe ser impar ya que si se compara con un número par de instancias jamás podremos decir a que grupo de características pertenece (en la regla de 2 de 3). Una vez comparada la nueva instancia con el conjunto de características que se encuentran en el DataSet, se determinara los “vecinos cercanos” con ello sabremos a que clase pertenece.



El ejemplo que se desea clasificar es el círculo verde. Para k = 3 este es clasificado con la clase triángulo, ya que hay solo un cuadrado y 2 triángulos, dentro del circulo que los contiene. Si k = 5 este es clasificado con la clase cuadrado, ya que hay 2 triángulos y 3 cuadrados, dentro del circulo externo.

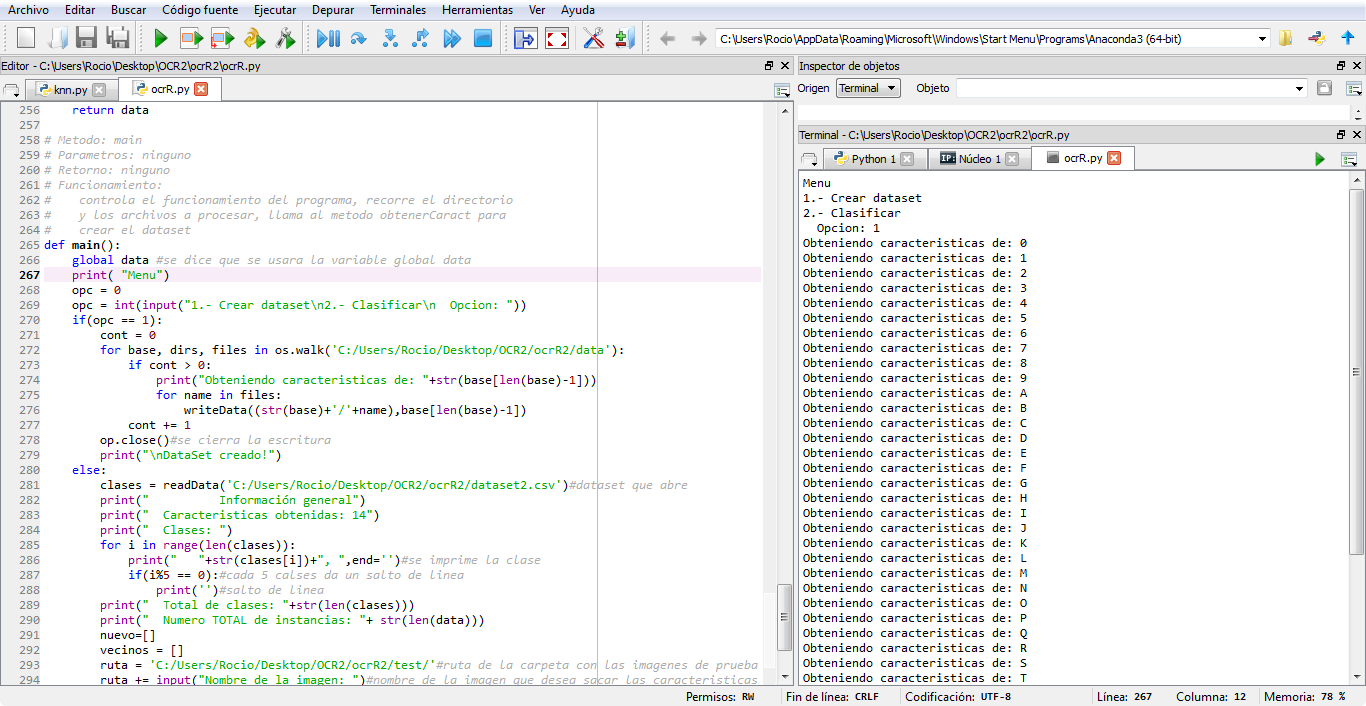
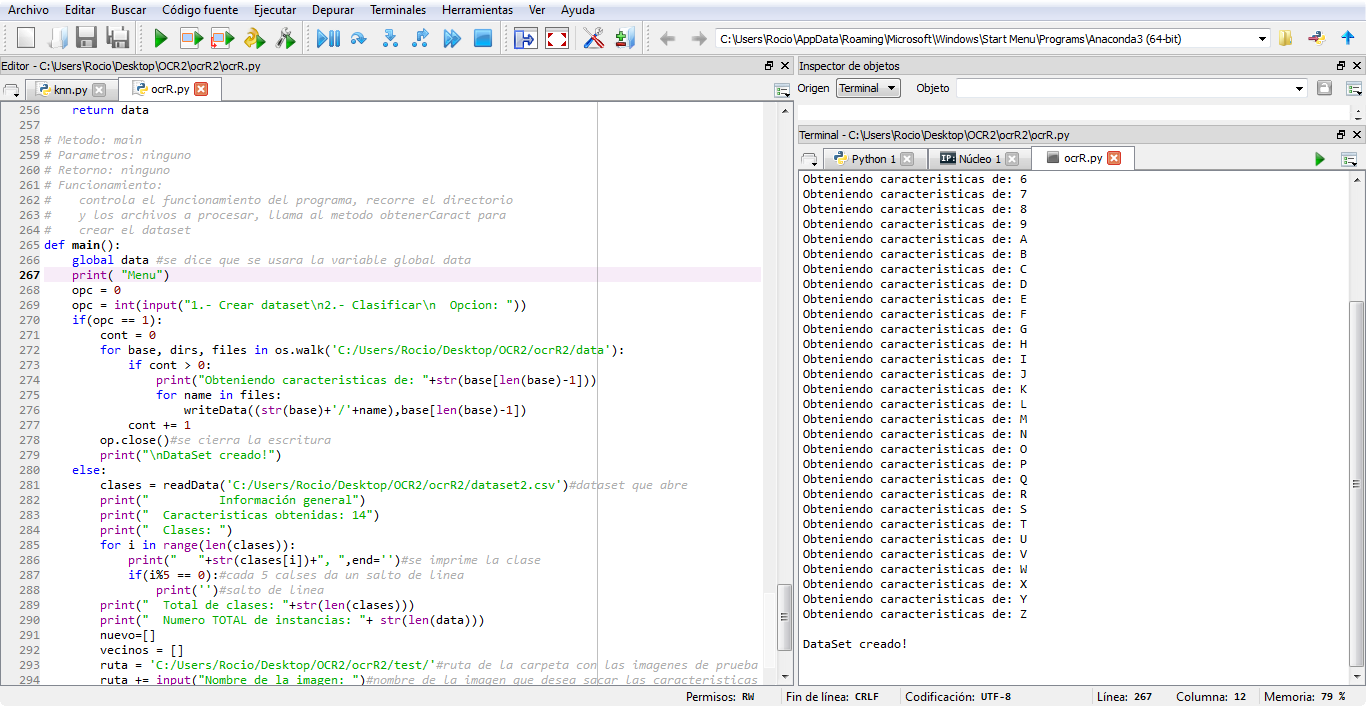
**Imagen 7. Ejemplo de KNN.**  
**FUNCIONAMIENTO DEL PROGRAMA (corrida)**

Menú principal del programa con dos opciones a elegir.



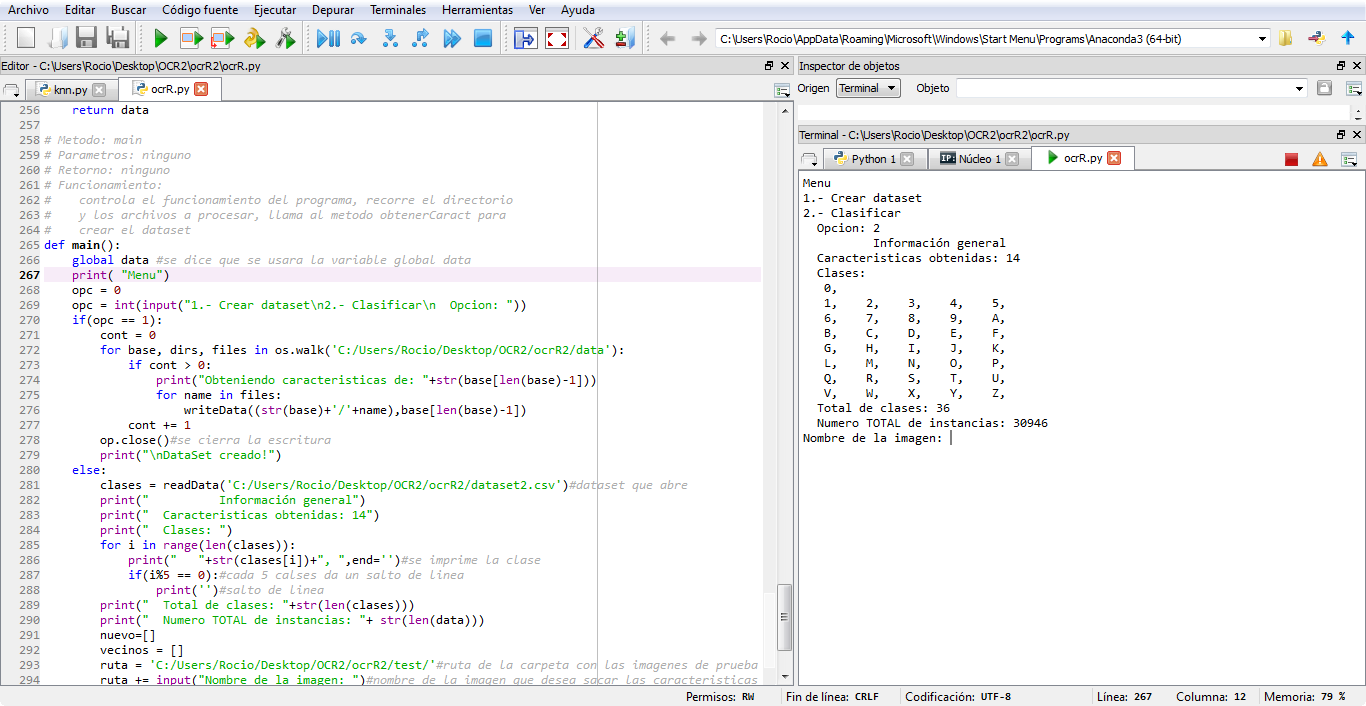
**Imagen 8. Menú del programa.**

Para poder elegir la opcion uno, ya debemos de contar con el conjunto de imágenes para poder generar nuestro DataSet. Despues de haber elegido la primera opción nos aparecera un recuadro que nos indica qué caracteristicas se estan obteniendo (imagen 8). Una vez creado nos aparece un mesaje indicando que ya se ha creado el DataSet (imagen9).

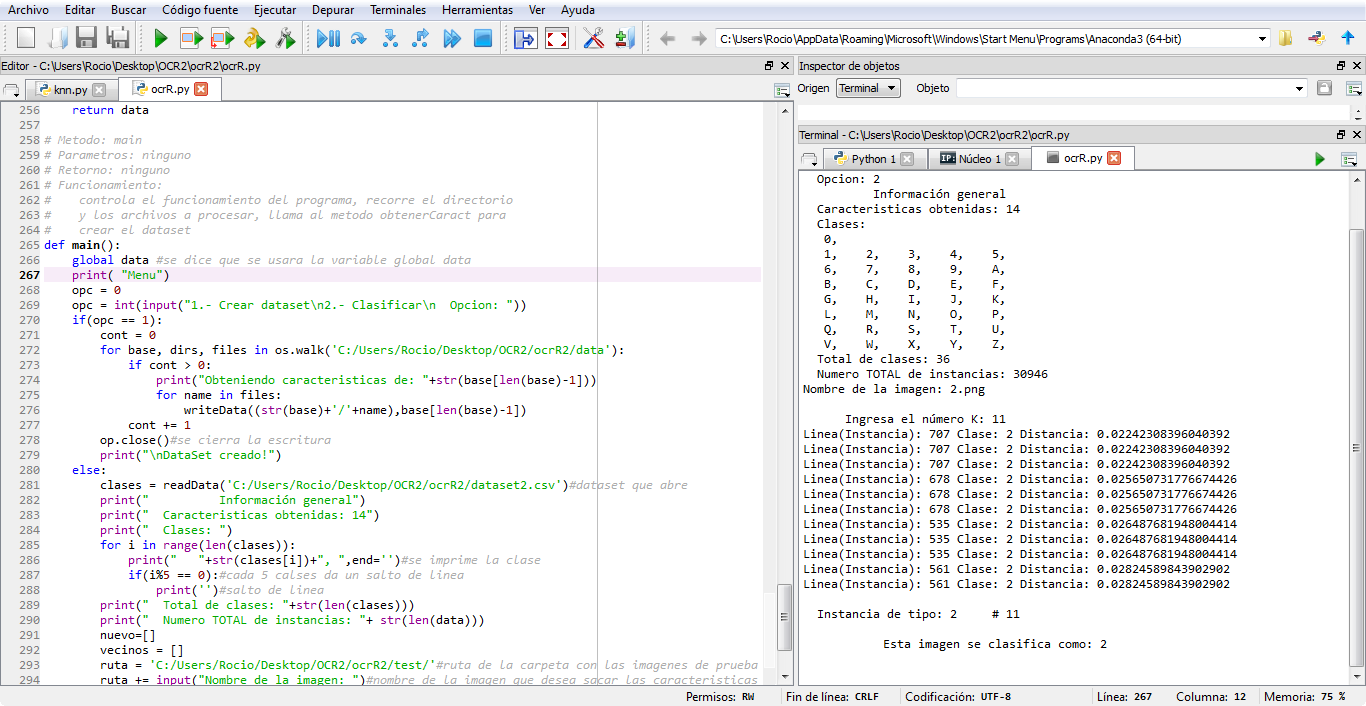
**Imagen 8. Características que se están obteniendo. Imagen 9. El DataSEt se ha creado**

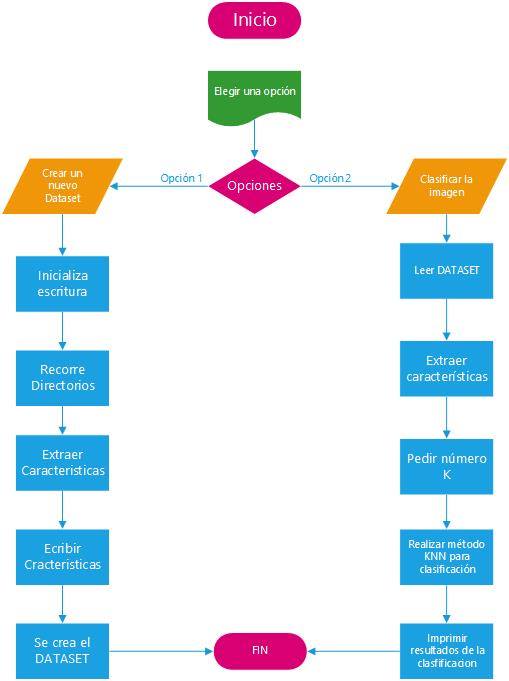
Para poder elegir la opción 2, ya debemos de contar con el DataSet que se generó en la opción 1. Al elegir la opción 2 nos aparecerá la información general de nuestro Dataset.

  
 **Imagen 10. Información del DataSet**

Después debemos ingresar el nombre de una imagen binaria para determinar sus características y a continuación se pedirá el número k de vecinos para comparar las características de la imagen con las del conjunto de imágenes que se encuentran en el DataSet.

Por último se mostrará la información de los k vecinos más cercanos. Se imprime el número de instancias y la votación. Al final se mostrara un mensaje determinando a que pertenece la imagen (imagen 11).

  
**Imagen 11. Nombre de imagen, numero de K y clasificación.**

  
**Imagen 12. Diagrama OCR.**

El programa se encuentra en la siguiente dirección de GitHub

<https://github.com/Rousdays/OCR>