|  |  |  |  |
| --- | --- | --- | --- |
| **Minería de datos**  **“Proyecto final”** | | | |
| **PF** | **#Rubén Abundiz Galván**  **#Angélica Atonal Morales**  **#Erick Guevara Pérez**  **#Edgar Alan Zarate Pérez** | **8A** | **E5** |

**Resumen**

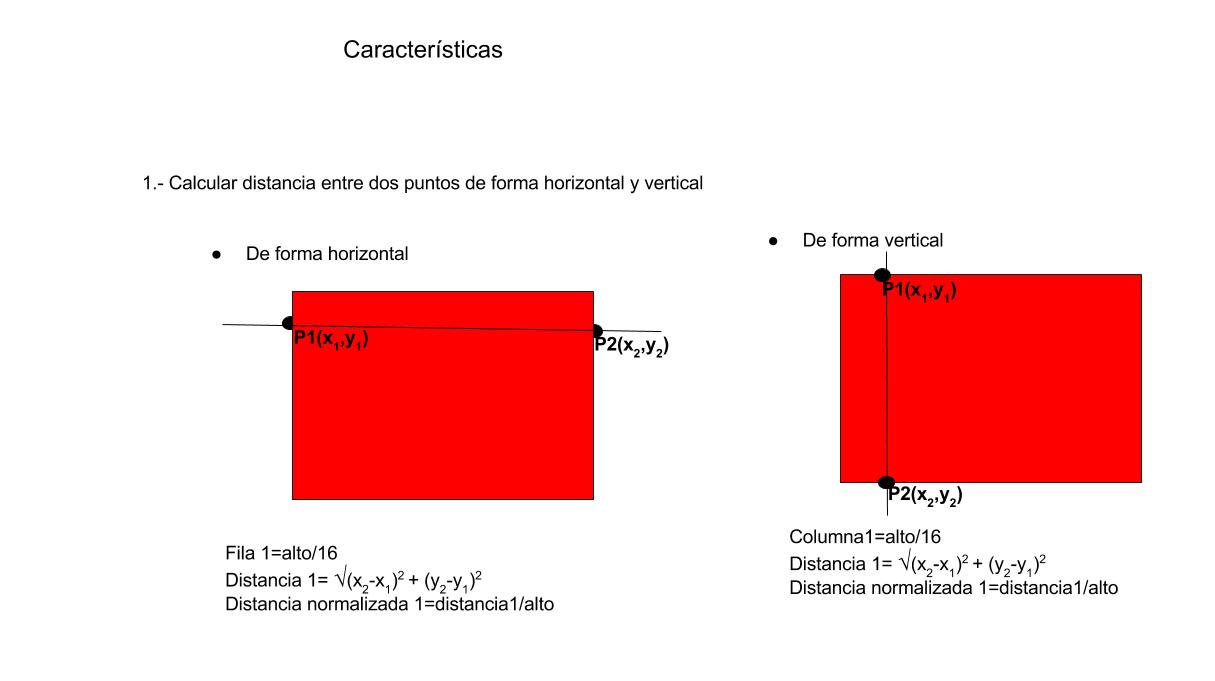
En el presente reporte explica el proceso de reconocimiento de imágenes, en el cual se analizaron las características programadas para saber que tipo de figura geométrica es la que se dibujó.

**Introducción**

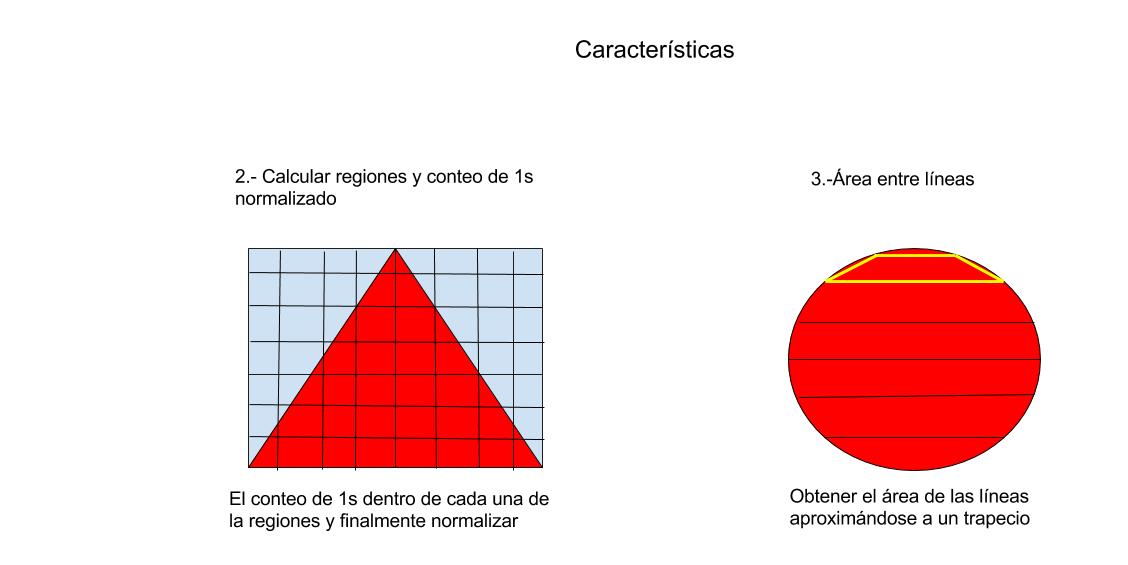
El programa reconocedor de imágenes que trata de emular la capacidad del ojo humano para reconocer imágenes. Es un software que permite el reconocimiento óptico de los caracteres contenidos en una imagen, de forma que estos se vuelven comprensibles o reconocibles para una computadora.

Las características para poder reconocer una imagen son las siguientes:

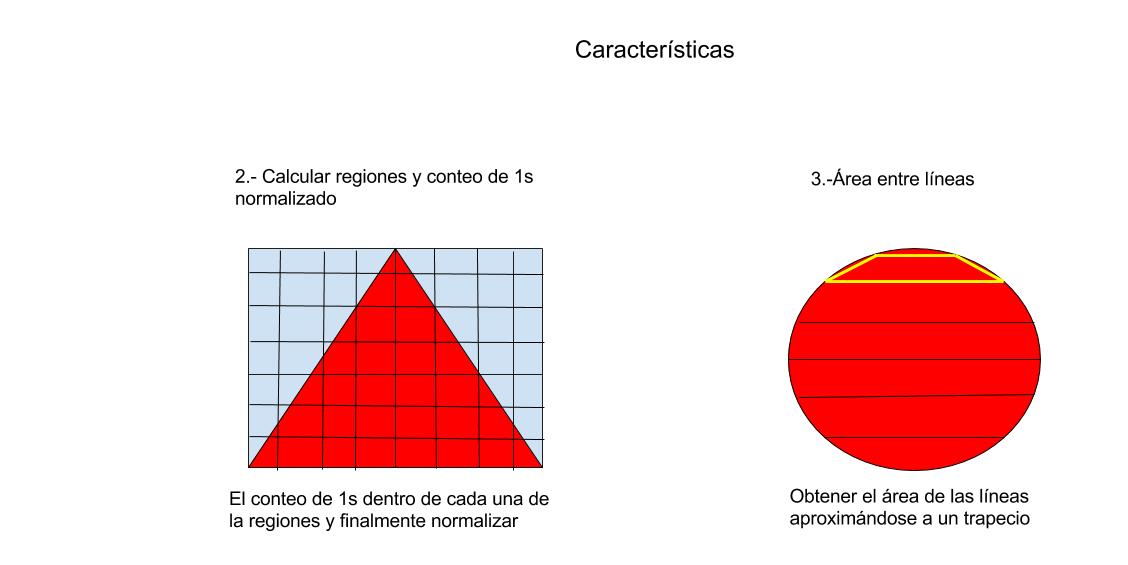
* Primer característica



* Segunda característica



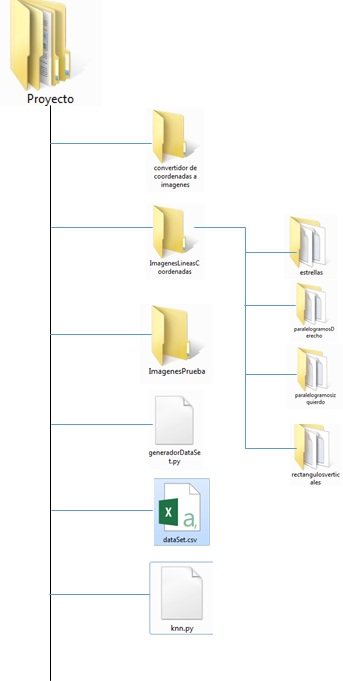
* Tercera característica



**Requerimientos**

* Equipo de computo
* Sistema Operativo Windows 7, 8, 8.1 o 10
* Python Anaconda versión 3.6
* Dibujar una imagen

**Estructura de archivos**



**KNN**

**Resumen**

En este documento se explica un ejemplo de cómo funciona el método KNN en el cual se tendrá que clasificar un elemento de prueba para conocer a que clase pertenece.

**Introducción**

El algoritmo KNN es un clasificador supervisado, normalmente empleado en Inteligencia Artificial para detectar patrones y, evidentemente, clasificarlos de acuerdo a determinada norma. La idea básica sobre la que se fundamenta este paradigma es que un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus K vecinos más cercanos.

Inicialmente el algoritmo requiere ser entrenado con patrones bien conocidos, por ejemplo, imaginemos que tenemos una muestra de 100 imágenes 40 de ellas tienen un automóvil, las otras no, entrenamos dicho algoritmo para que reconozca cuales imágenes tienen un auto y cuáles no.

Posteriormente se selecciona un umbral de certeza para que, el algoritmo pueda disertar en una imagen nueva que no pertenezca a la muestra original y decida si hay o no un auto, con una probabilidad igual o mayor a nuestro umbral (pero nunca menor).

**Funcionamiento del programa**

