ESPECIFICACIÓN Y DISEÑO DE UN TRADUCTOR DE TEXTO A BRAILLE.

JHOLBERT GEOVANNY LADINO VEGA
YOLMAN SAUL LUNA ROPERO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2012

ESPECIFICACIÓN Y DISEÑO DE UN TRADUCTOR DE TEXTO A BRAILLE.

JHOLBERT GEOVANNY LADINO VEGA
YOLMAN SAUL LUNA ROPERO

Trabajo de grado presentado como requisito para optar al título De Ingeniero Electrónico

Director
DR. RICARDO LLAMOSA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2012

Durante estos años, siempre han estado presentes apoyando y buscando mi bienestar, sin pedirles nada me han dado todo y han logrado hacerme sonreír muchas veces, es por eso que hoy quiero agradecer a esas personas que me han dado felicidad, a mi familia y en especial:

A *Dios* el centro de mi vida, ese papá, ese amigo fiel que en tierras lejanas, me acompaño, guio mis pasos y me escucho, siempre atento a ayudarme.

A mis papás, *Víctor Ladino* y *Esperanza Vega* por el amor, apoyo, fortaleza y confianza que siempre han depositado en mí. Espero no dejar de escuchar sus consejos porque me han llevado a ser feliz.

A mis hermanos, *Edwin* y *Lorena* que siempre han creído en mí, sus palabras de afecto siempre fueron motivo para seguir, siempre estaré atento a ayudarlos.

A *Laura Valentina*, *Juliana*, Isabela.

A mis Abuelitos por su especial cariño.

A mis amigos, por su compañía y colaboración.

JHOLBERT LADINO

AGRADEZCO

Especialmente a mis padres Saúl Luna y Berenice Ropero por su apoyo incondicional, por sus consejos y en especial por sus enseñanzas que he recibido, a mis hermanos por estar siempre a mi lado.

Quiero agradecer a mi tía Ofelia Luna por su motivación y confianza depositada en mí.
Finalmente agradezco a mis amigos por su motivación y colaboración en el desarrollo de este trabajo de investigación.

YOLMAN SAUL LUNA ROPERO

AGRADECIMIENTOS

Al Profesor y Director Ricardo LLamosa por confiar en nosotros y permitir el día de hoy presentar los resultados a nuestra propuesta.

A OCTOPLUS.

A la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

A la Universidad Industrial de Santander.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

RESUMEN

TÍTULO: DISEÑO Y ESPECIFICACIÓN DE UN TRADUCTOR DE TEXTO A BRAILLE.*

AUTORES: JHOLBERT GEOVANNY LADINO VEGA**
YOLMAN SAUL LUNA ROPERO**

PALABRAS CLAVE: Segmentación, Binarización, Corregir Inclinación, Braille, Procesamiento digital de Imágenes, OCR, Traductor.

DESCRIPCIÓN:

La manera en que el procesamiento digital de imágenes permite que un diseñador interactué con herramientas computacionales para la creación de un algoritmo que de solución a un problema actual de la comunidad invidente, el acceso a la informaci.

En este trabajo se presenta una descripción del diseño de un algoritmo que traduce un texto en prosa a braille. El algoritmo es planteado basándonos en la teoría del tratamiento digital de imágenes, para ello se analizaron diferentes técnicas y en el documento se indicaran los respectivos resultados a cada prueba. Se analizo además las opciones de captura de imágenes para obtener el mejor dispositivo de entrada y el diseño de un dispositivo que permitiera dicha salida o visualización al actor.

Se realiza una simulación de este algoritmo que mediante una serie de pruebas usando imágenes de texto tomados con una cámara web de alta definición (HD), indican que es posible una futura implementación de este proceso. El código da como salida una cadena de caracteres los cuales son enviados vía USB. Para este proceso se hace necesario el uso de un microcontrolador (Modulo JM60), el cual captura cada caracter respuesta y realiza su conversión para la respectiva visualización en diodos Led (1 lógico (5v) alto relieve, 0 lógico (0v) bajo relieve).

* Trabajo de Grado

** Facultad de Ingenierías Físico – Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director. Ricardo Llamosa.

ABSTRACT

TITLE: SPECIFICATION AND DESING OF A TEXT TO BRAILLE TRANSLATOR.[*]

AUTHORS: JHOLBERT GEOVANNY LADINO VEGA [**]
YOLMAN SAUL LUNA ROPERO[**]

KEY WORDS: Segmentation, Binarization, Correct inclination, Braille, Digital image processing,

OCR, Translator.

DESCRIPTION:

The digital image processing allows the designer to interact with computational tools, for development of an algorithm. An idea is proposed in this paper, is to take the first step to solve the current problem in the blind community, access to information.

This document shows a description of the algorithm design which translates a prose text into Braille. The algorithm is proposed based on the theory of digital image processing. For this, we analyzed different techniques and in the document will indicate the respective results to each test. It also analyzed the capture options to obtain the best input device and the design of a device that would allow the output or display to the actor.

We performed a simulation of this algorithm that by means of a series of tests using images of text taken with a high definition webcam, it shows that is possible a future implementation of this process. The code gives as output a succession of characters which are sent by USB. For this process, it is necessary the use of a microcontroller (Module JM60), that captures every character response and carries out its conversion for the respective LEDs display (logic 1 (5V) high relief, logic 0 (0V), low relief).

[*] Degree Project

[**] Physical – Mechanics Engineering Faculty. Electrical, Electronic and Telecommunications engineering School. Director. LLAMOSA Ricardo.

1. INTRODUCCIÓN

1.1 Motivación

Un interés propio en las aplicaciones y funciones del procesamiento digital de imágenes, es la principal motivación para la realización de este proyecto. Se busca dar solución a un problema de acceso a la información que tiene la comunidad invidente. Además de un constante aprendizaje con el análisis de cada una de las etapas que componen el proceso. Se aspira también, que este diseño sirva de motivación para que la comunidad académica se interese en el diseño de software y dispositivos que faciliten la vida a esta comunidad.

Se presentan en este documento las pruebas realizadas al algoritmo y se darán las especificaciones del mismo, además, determinar las entrada y salida que serían los encargados de tener interacción con los invidentes y/o personas con baja visión.

La mayor parte de análisis de esta propuesta se centra en el análisis y creación del software encargado del procesamiento y tratamiento de la imagen que tiene la información de texto.

La creación de este documento se inicia con el estado del arte, ítem fundamental para la visión global, funcionamiento y posibles fallos con este tipo de propuestas. El objetivo es obtener un texto en formato digital, y con ayuda del reconocimiento de caracteres ópticos (OCR), traducir a lenguaje Braille que es el sistema de lecto-escritura del invidente.

1.2 Objetivos

Objetivos Generales

Diseñar un sistema que permita traducir un texto reconocido mediante OCR al sistema Braille.

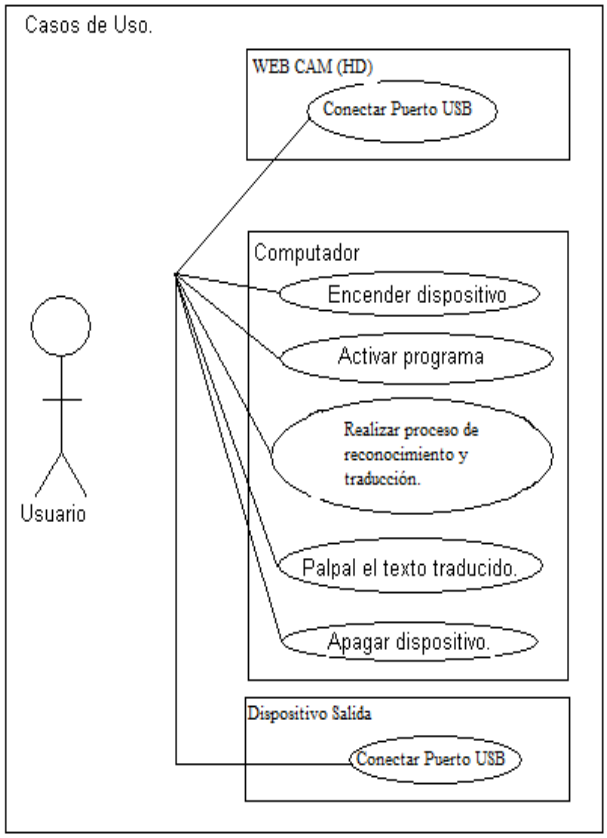
Objetivos Específicos

- Capturar, reconocer y digitalizar una imagen cuyo contenido domine un texto en prosa e idioma español.
- Procesar la información de la imagen (texto en prosa) mediante un software que permita el reconocimiento óptico de caracteres.
- Diseñar un algoritmo que permita convertir (traducir) el texto extraído del documento a lenguaje Braille, basados en patrones de reconocimiento.
- Generar especificaciones técnicas de funcionamiento del algoritmo; en base a los resultados obtenidos. Y determinar los equipos de entrada y salida adecuados para este diseño.

1.3 Planteamiento.

A continuación se modela el sistema basándose en lenguaje visual (UML). Con el cual se pretende representar gráficamente las reglas de estructura y comportamiento del proyecto, además de obtener más ágilmente las especificaciones ante los posibles cambios.

Figura 1 Casos de uso.

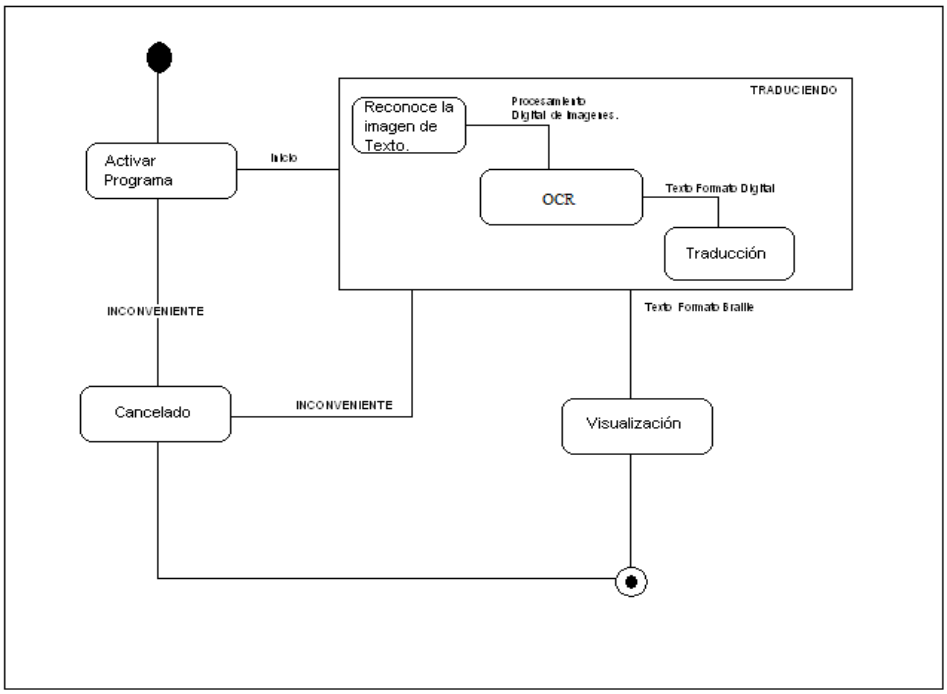


Fuente: Autores.

En la Figura 1 y hasta la Figura 4, se indica la interacción que tiene el usuario con la serie de eventos ocurridos durante el proceso, también se especifica que el actor en esta investigación serán las personas con limitación visual o de baja visión.

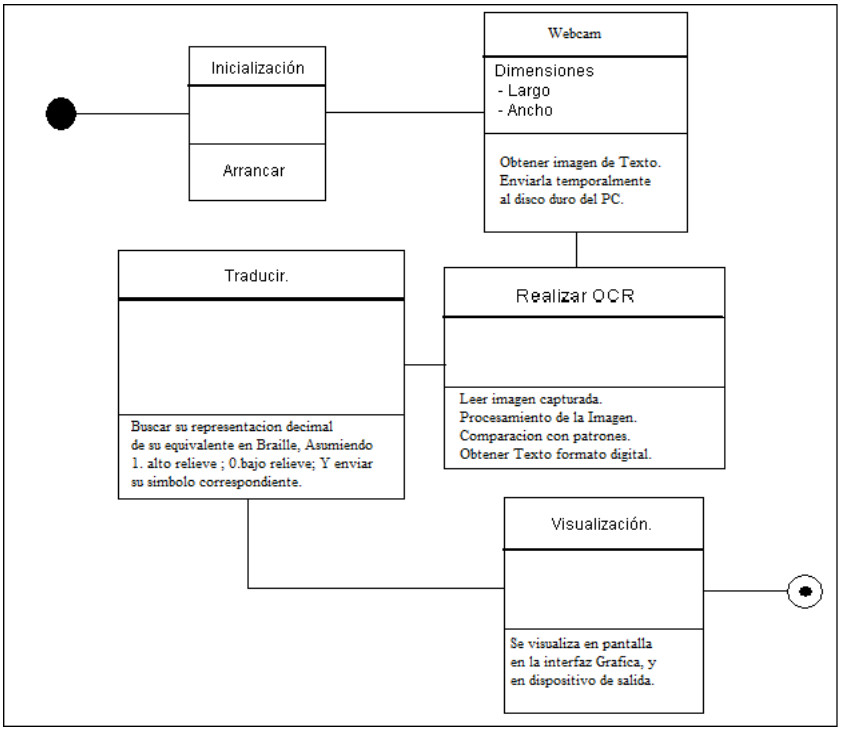
Un diagrama de secuencia es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos. Con los diagramas se comprende básicamente el proceso que se debe realizar para obtener el objetivo principal, además, se aclaran las diferentes etapas, opciones y tareas que se deben cumplir para continuar el proceso. En la Figura 4, observamos la interacción del conjunto de objetos a través del tiempo, se hace énfasis en el orden y el momento en que se envían los mensajes a los objetos.

Figura 2 Diagrama de estados.



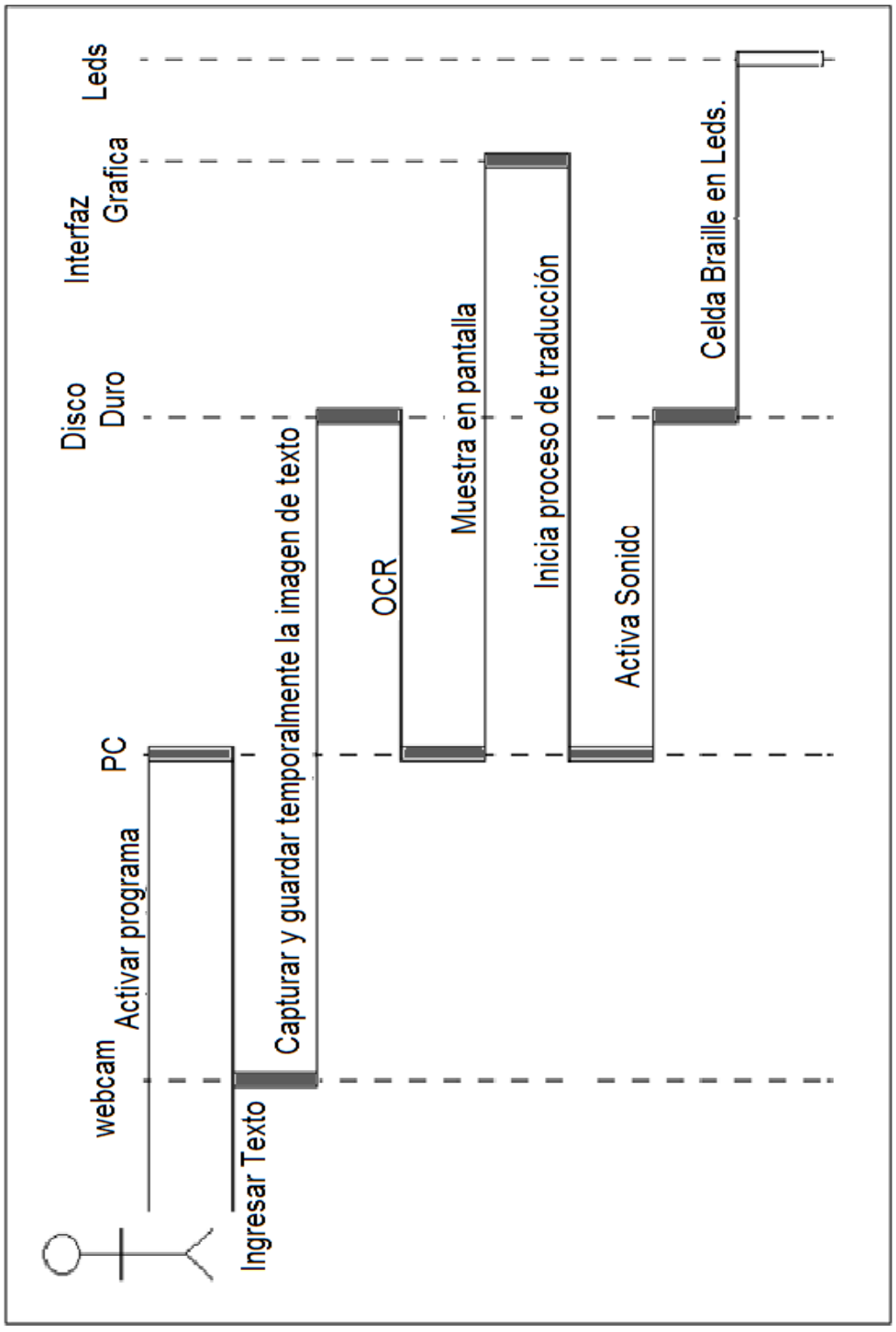
Fuente: Autores.

Figura 3 Diagrama de clases.



Fuente: Autores.

Figura 4 Diagrama de secuencias.



Fuente: Autores.

1.4 Estructura del documento

Este documento está estructurado en capítulos que cuentan y evidencian los pasos, resultados, y métodos utilizados para el desarrollo de esta investigación.

Introducción.

En este capítulo se tratan los porqués de la realización del proyecto, los objetivos propuestos para su desarrollo y su estructura.

Estado del arte.

En él se indican las investigaciones realizadas por otros autores en el campo del reconocimiento óptico de caracteres. También explica algunas de las técnicas utilizadas y aplicaciones de sistemas ya implementados (Ver Figura 5).

Descripción del sistema software.

Este capítulo describe las etapas necesarias para crear el algoritmo de reconocimiento de caracteres. En cada etapa se expresa las especificaciones necesarias de entrada y se describe la respuesta adecuada para cada una de ellas.

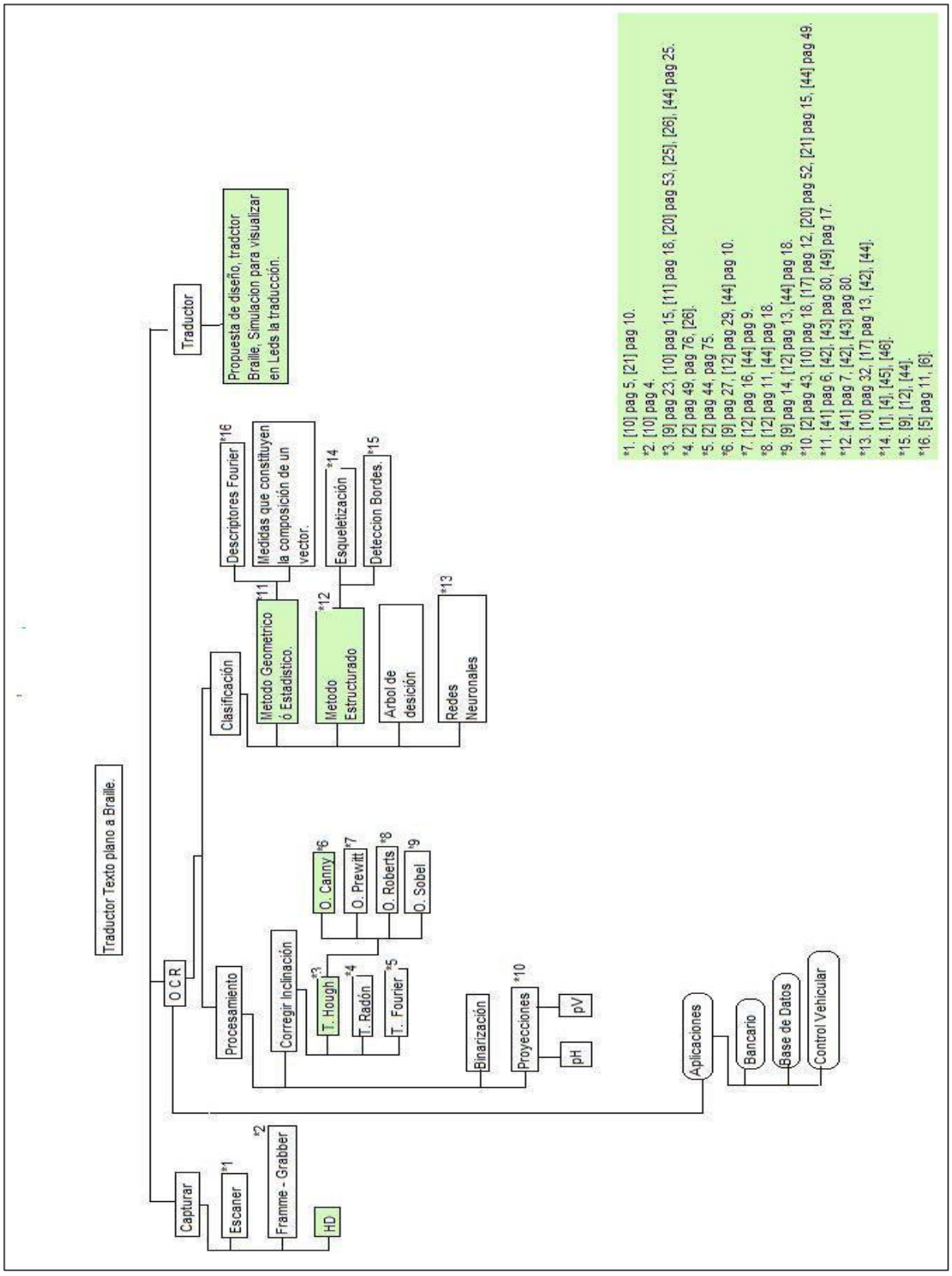
Pruebas del sistema software.

Se indican los resultados obtenidos para cada herramienta analizada con el propósito de elegir la que mejor desempeño muestre en cada una de las etapas descritas en el capítulo de descripción de sistema software.

Descripción del sistema hardware.

Este capítulo describe las pruebas realizadas a los dispositivos encargados de captura de imágenes para elegir y especificar un dispositivo apropiado para la aplicación. Se describe el proceso que realiza el computador encargado de procesar el algoritmo. Y el diseño de una propuesta para el dispositivo de salida. La simulación que se realiza es sencilla.

Figura 5 Diagrama de estado del arte.



Fuente: Autores.

Conclusiones.

En este capítulo se presentara una serie de conclusiones a partir de la observación de los resultados.

Líneas de trabajos futuros.

Con base en los resultados obtenidos se plantean ideas de desarrollo que permiten mejoras a este proceso de traducción, además de posibles aplicaciones para el proceso OCR.

Bibliografía.

Son las referencias bibliográficas que se han consultado.

Anexos.

Además de la información descrita en los capítulos anteriores, se tienen documentos de anexos, como información adicional para la solución y desarrollo del proyecto.

2. ESTADO DEL ARTE

Al hacer un análisis del comportamiento del cuerpo humano, este se describe como un sistema, en el cual se hace necesario un sensor que capture datos y un procesador que los analice y distribuya. El cuerpo humano utiliza los cinco sentidos como sensores para este objetivo, siendo el más importante el sentido de la visión, con él cual se distinguen las diferentes tareas o procesos que se quieren realizar, esta información es enviada al cerebro y en cuestión de segundos se tiene una decisión sobre las partes del cuerpo que se activan para ejercer la función elegida.

¿Cómo hacer cuando no contamos con este potente sensor (sentido)? ¿Cuando no podemos ver las opciones? En este documento se propone una solución a este problema, indicando que es posible imitar esta carencia utilizando una cámara que se utiliza como sensor para realizar el trabajo desarrollado por el sentido de la visión (capturar imágenes), y un computador que sería el encargado de indicar el proceso que se debe realizar en esta aplicación. La actividad que realiza el computador se programa con instrucciones de código que permita la comunicación entre los dispositivos. Gracias al avance tecnológico, hoy día todas las personas tienen facilidad de acceso a un computador, el precio y el tamaño de estos han cambiado considerablemente desde el primer sistema creado.

En 1929 se obtuvo en Alemania una patente de OCR la cual consistía en una maquina mecánica, que utilizando plantillas detectaba una imagen y se enviaba una luz sobre ella cuando era detectada, a partir de este momento comenzaron los estudios que permiten hoy día hacer del tratamiento digital de imágenes, parte fundamental de la automatización de procesos. ¿Pero porque se dice que son de automatización? Estos sistemas facilitan de manera general la solución de un proceso, su rápida respuesta y eficiencia permiten que sea más eficaz que el

resultado obtenido por la capacidad de varios empleados, sin embargo aún es necesario de un monitoreo constante de un operario especializado en el tema, que corrija y este pendiente de la información obtenida por dichos sistemas. Mediante estos sistemas se pueden llevar a cabo diversas tareas, tales como el almacenamiento de documentos escaneados en formato textual, la lectura de formularios de forma automática, la reproducción de una canción a partir de su partitura escaneada, y ahora están muy de la mano de la parte de seguridad, ya sea por reconocimiento de huella o iris, se están realizando multitud de proyectos, entre otros [2,19].

A pesar de las ventajas que hacen popular e interesantes a los sistemas de reconocimiento de símbolos ó caracteres y del rango de trabajo se deben tener algunas consideraciones cuando se va a diseñar un sistema con esta herramienta de procesamiento. En el caso del reconocimiento de caracteres, por lo general, se limita al uso de ciertos estilos de letra previamente definidos. Lo mismo ocurre con los formularios de lectura automática, donde se requiere que las casillas que el usuario marque se rellenen de una forma y color determinado, evitando tachones que generarían una mala respuesta. Para el reconocimiento de matrículas y de los anteriores casos nombrados es importante que la imagen obtenida sea de gran calidad donde se distingan claramente las letras, símbolos y espacios.

Varias aplicaciones se ven mejoradas gracias al reconocimiento óptico de caracteres. El objetivo general de la sociedad actual es digitalizar toda la información existente para poder analizarla y transmitirla, y con ello crear bases de datos con información disponibles de interés del usuario o interesados.

Desde 1929, existen estudios los cuales han generado la creación de códigos para este proceso, códigos costosos y a los que pocas personas tienen acceso. Hoy día existen programas en línea tanto gratuitos como pre-pagados que realizan

este proceso, los gratuitos no garantizan un buen resultado, los pre-pagados aún son costosos.

Este trabajo tiene como propósito documentar y realizar una aplicación de OCR creando un código que desarrolle el proceso descrito y que sea la base de futuras mejoras para este tipo de aplicaciones.

El reconocimiento de símbolos y caracteres es un trabajo difícil, no sólo por el estilo de letra propio de cada escritor, sino también por la amplia variedad de combinaciones de los caracteres que debe soportar el sistema reconocedor.

OCR “Reconocimiento óptico de caracteres” sin embargo es un proceso tradicional para el reconocimiento de símbolos, basados en plantillas y con limitaciones, ya sea en tipo de letra o tamaño. Con el constante avance tecnológico también se cuenta ahora con sistemas de reconocimiento automático de caracteres más recientes, son capaces de adaptar su procesador de reconocimiento para poder aprender nuevos símbolos y notaciones a partir del reconocimiento tradicional OCR, actualmente la utilización de redes neuronales que pretende ser como un cerebro capaz de comprender con entrenamiento el método de reconocimiento y corrección de palabras haciendo de estas las más eficientes actualmente. Sin embargo el OCR es un concepto del que se puede aprender mucho y con el que se pueden diseñar varias propuestas interesantes.

El proyecto se analiza utilizando técnicas que permitirán el reconocimiento de texto en prosa e idioma español. Adicionalmente se deja claro que el sistema de reconocimiento beneficia la vida académica y cultura del usuario con baja o nula visión, si se realiza una implementación del sistema. Por tal se deben tener consideraciones importantes a la hora de la captura de la imagen (materia prima). Las técnicas utilizadas para obtener un buen OCR son: captura de la imagen, binarización, segmentación y clasificación. Como etapa adicional su representación visual.

2.1 Investigación de otros autores.

➢ Historia OCR.

Para tener un poco más de conocimiento acerca del reconocimiento óptico de caracteres se reseña a través del tiempo, cómo ha llegado a ser lo que es ahora. Todo comienza en 1929 con una maquina mecánica que basada el reconocimiento de una imagen mediante plantillas, cuando la imagen de la plantilla coincidía con el caracter que analizaba, una luz señalaba dicho elemento. Las primeras aplicaciones prácticas de los sistemas OCR fueron de un desarrollo muy limitado hasta la década de los 50, y su aplicación se fue haciendo de uso más generalizado a partir de los años 60.

En 1950, David Shepard decide que es posible construir una máquina para realizar el proceso en problema de una agencia de seguridad de las fuerzas armadas de EEUU, convertir mensajes impresos en lenguajes para almacenarlos en un computador, y, con la ayuda del cocinero de Harvey, un amigo, construyeron Gismo. Este suceso fue divulgado en los periódicos Washington post, Daily News y New York Times, en el año 1953, después de que su patente fuera concedida. En este momento, Shepard fundó Intelligent Machines Research Corporation (IMR), comenzando a fabricar el primero de varios sistemas del OCR usados para operaciones comerciales.

El primer sistema comercial fue instalado en Readers Digest en 1955, que, muchos años más tarde, fue donado al museo Smithsonian, donde fue puesto en exhibición. El segundo sistema fue vendido a los Standard Oil Company de California para leer impresiones en tarjetas de crédito para propósitos de facturación, además se vendieron muchos más sistemas a compañías petroleras. Otros sistemas vendidos por el IMR durante los últimos años 50 incluyeron un escáner de páginas para la fuerza aérea de los Estados Unidos para la lectura y

transmisión de mensajes escritos a máquina. IBM y otras empresas fueron licenciadas más adelante sobre las patentes del OCR de Shepard [17,42].

El servicio postal de Estados Unidos ha estado utilizando las máquinas de OCR para clasificar el correo desde que 1965, basados en la tecnología ideada sobre todo por el inventor prolífico Jacob Rabinow. El primer uso del OCR en Europa sucedió en la oficina de Gran Bretaña. En 1965 se comenzó a planear un sistema de actividades bancarias completo, Nacional Giro, usando la tecnología del OCR, ideó un proceso que revolucionó los sistemas del pago de cuentas en el Reino Unido. El correo postal de Canadá ha estado utilizando sistemas OCR desde 1971. Los sistemas OCR leen el nombre y la dirección del destinatario, e imprimen un código de barras en el sobre basados en el código postal del mismo. Después, las cartas necesitan solamente ser clasificadas por los compaginadores, menos costosos que necesitan leer solamente el código de barras. Para evitar interferencia con el campo de dirección escrita a mano, que se puede situar en cualquier parte de la carta, se usa una tinta especial leída bajo una ultravioleta. Esta tinta parece anaranjada en condiciones normales de la iluminación. Los sobres marcados con el código de barras que son leídos por la máquina pueden ser posteriormente procesados [2, 17, 42].

- Aplicaciones OCR.

Las aplicaciones OCR en los últimos años han tenido un crecimiento aplicado a lo largo de todo espectro de industrias, OCR permite que los documentos escaneados dejen de ser un archivo de imagen, y convertirlos en un documento digital, donde se pueden hacer búsquedas y correcciones. Con la ayuda de OCR, ya no es necesario volver a escribir manualmente los documentos que se necesitan en formato digital para su inscripción en bases de datos electrónicas.

Tales como uso bancario, uso bases de datos, entre otras [48], sistema de control vehicular [49].

2.2 Software disponible.

Actualmente existen programas OCR disponibles, que tiene la finalidad de reconocer caracteres o respuestas, por ejemplo, el procesamiento de formularios y obtener textos en formato digital. A continuación se indicaran algunos de los más importantes programas comerciales en la industria.

- ExperVision, fue fundada en 1987, para diseñar, desarrollar y comercializar el reconocimiento óptico de caracteres (OCR) para los sistemas de lenguajes universales. Su sistema proporciona una alternativa eficaz para el teclado manual. Están comprometidos a proporcionar OCR como parte integrante de la digitalización de documentos más potente y soluciones de gestión, para ayudar a alcanzar la máxima productividad. Cuentan con un sistema de reconocimiento de caracteres propio que es significativamente más rápido y preciso que cualquier otra tecnología [2] [1].

- ABBYY FineReader es un programa para el reconocimiento óptico de caracteres (OCR) aplicación desarrollada por ABBYY. FineReader fue diseñado como una aplicación de nivel profesional para convertir imágenes escaneadas, fotografías de documentos y PDF, en archivos Microsoft Word , Microsoft Excel , Microsoft PowerPoint , RTF , HTML, PDF, CSV y archivos de texto que permiten ser editables, es decir obteniendo texto digital [2][2] .

- OmniPage fue uno de los primeros programas OCR en ejecutarse en ordenadores personales. Convierte las imágenes, tales como las de documentos escaneados y archivos en formato PDF en textos digitales compatibles con Microsoft Word, Excel, Adobe Acrobat. El software se comercializa en dos versiones, una dirigida a profesionales que incluye una

[1] http://www.expervision.com/
[2] http://latam.abbyy.com/

serie de características adicionales, incluyendo software de voz avanzada para la producción de audio-libros, y una versión básica más orientada para uso personal. Trabaja con las diferentes versiones de Windows y Mac OS X sistemas operativos. El software es compatible con los interfaces estándar como TWAIN y WIA , de tal modo que cubre prácticamente todos los escáneres y dispositivos multi-función en el mercado [2][3].

- Readiris es un software de reconocimiento óptico de caracteres para Microsoft Windows y Mac OS. Es producido por la empresa belga de Sistemas integrados de reconocimiento de imágenes IRIS Group SA. Actualmente se encuentra en la versión 12. Adicionalmente el producto ofrece la posibilidad de digitalizar a mano notas impresas [2][4].

2.3 Técnicas utilizadas

En este apartado se detallará las herramientas importantes que se utilizaran en el desarrollo del proyecto, sin embargo en la Figura 6, se encuentran todas las herramientas que serán utilizadas.

- Captura de imagen digital.

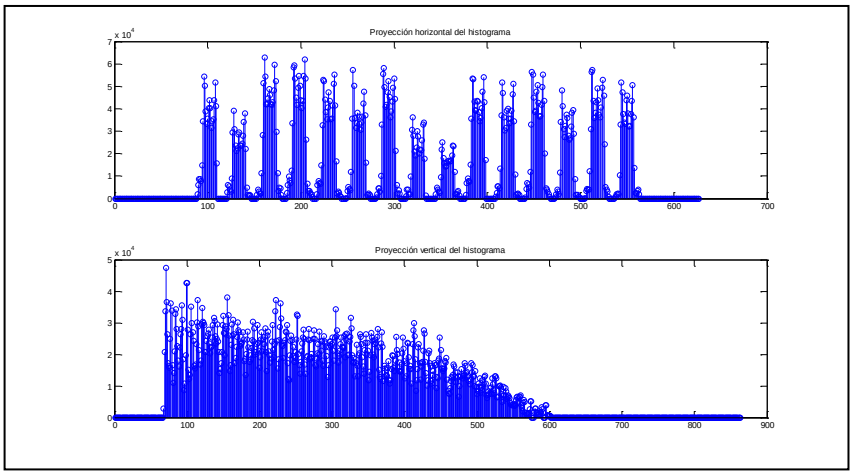
Lo primero y esencial para poder llevar a cabo un análisis o proceso digital de una imagen, es obtener la representación gráfica en un computador, la cual se someterá a través de herramientas especializadas, para realizar las diferentes mejoras y cambios a la imagen para obtener un resultado esperado. Existen en la actualidad tres métodos que permiten realizar la captura. Escáner, Framme – Grabber, Webcam [10, 21].

[3] ftp://ftp.scansoft.com/nuance/datasheets/ds_op17_standard_es.pdf
[4] http://www.irislink.com/c3-1684-58/Readiris-12-for-Windows.aspx

➢ Proyecciones horizontal y vertical.

Las proyecciones Vertical y Horizontal se usan para detectar los símbolos presentes. La proyección Horizontal es el resultado de la suma de pixeles blancos en cada una de las filas que conforma la imagen, la proyección Vertical por su lado es la suma de pixeles blancos en cada columna que conforma la matriz de la imagen. Estas proyecciones construyen un grafico que es la representación de los vectores de las dos proyecciones con el valor total de color blanco por fila o columna según sea el caso.

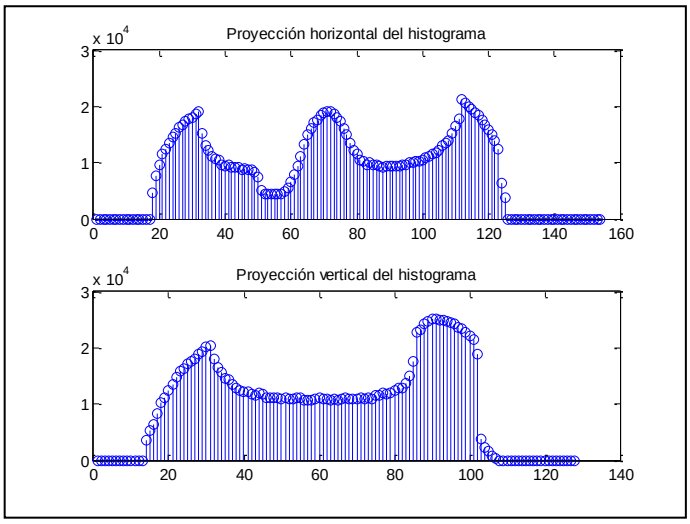
Figura 6 Proyecciones X e Y de un texto, Segmento superior proyección horizontal, Segmento inferior vertical.



Fuente: Autores.

En la Figura 6, se puede ver un ejemplo gráfico de las proyecciones que se deben saber utilizar e interpretar (*imagen original de proyecto, realizada a una imagen del proceso para comprobar teoría*). Cuando se hace el análisis de proyecciones a un párrafo en la imagen, la proyección importante es la horizontal, que indicara el número de líneas que conforman el texto, si se realiza este análisis a una línea, la importante es la proyección vertical, que indica el número de caracteres ó símbolos que conforman la línea. Pero si se realiza a un caracter, ambas son importantes porque ambas definen la forma y composición del mismo.

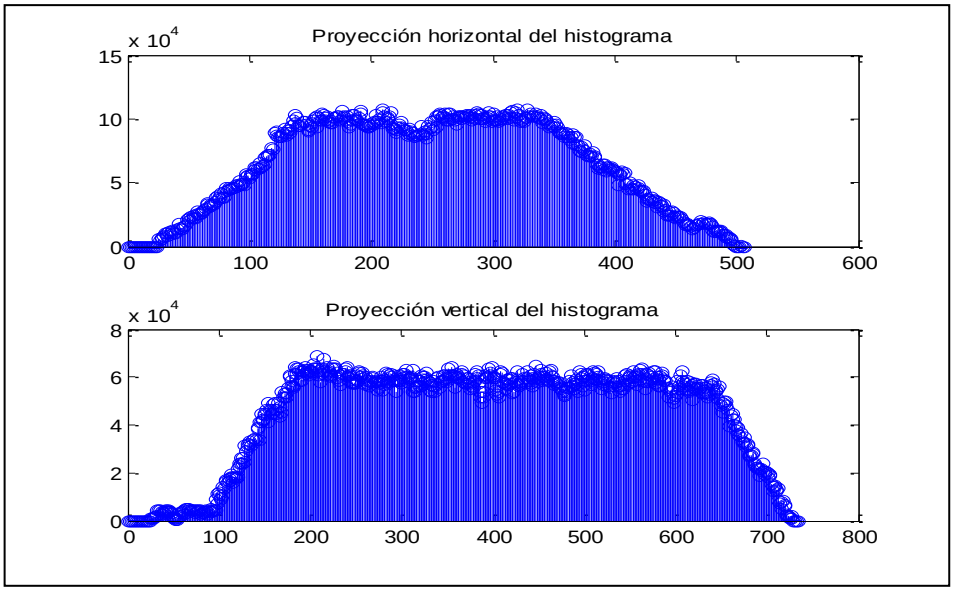
Figura 7 Proyección X y Y del caracter a.



Fuente: Autores.

Cuando la imagen presenta rotación el proceso de proyecciones indica un vector con valor de blanco en todas las filas y columnas y no es posible realizar el análisis de segmentación.

Figura 8 Proyección X e Y de un texto con 5º de rotación.

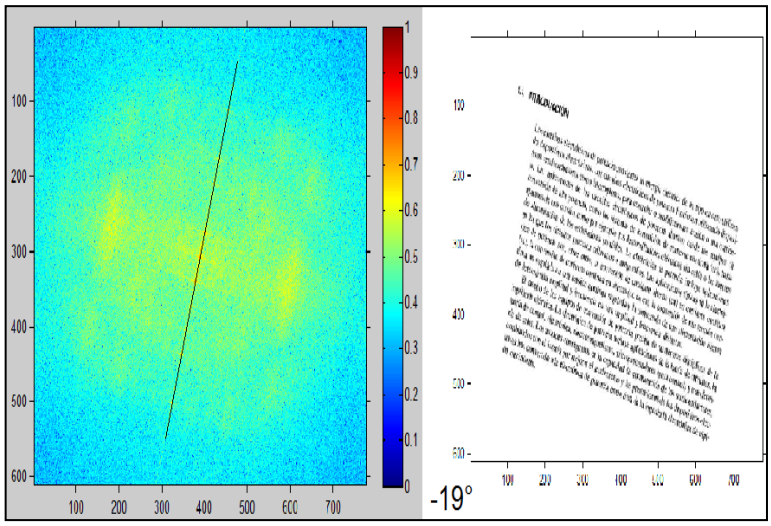


Fuente: Autores.

➢ Transformada discreta de Fourier.

En el caso de que exista un ángulo de rotación ocasionado porque el usuario ubica mal el documento para la captura, se obtendrán malos resultados como se observa en la Figura 8, donde la rotación genera resultados impredecibles. Para solucionar este problema se puede recurrir a herramientas que sean capaces de detectar la rotación para corregirla porque es importante para el reconocimiento y clasificación. Con ayuda de la transformada de Fourier es posible encontrar una gráfica donde una línea central indica la rotación del documento. En la Figura 9, se pueden ver los resultados obtenidos al aplicarla sobre un ejemplo.

Figura 9 Transformada de Fourier de un texto rotado 5º.



Fuente: Autores.

Se puede ver que en la transformada de la imagen de texto inclinado, se obtiene una línea que indica la inclinación del texto en la Figura 9. Sin embargo esta transformada no indica el ángulo de rotación pero ayuda a detectar la línea de inclinación necesaria para encontrar el ángulo de rotación. La transformada discreta de *Fourier* (DFT) para una imagen M x N y su inversa son definidas en (1).

$$F\ u,v\ = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{(-ju\frac{2\pi}{M}x-jv\frac{2\pi}{N}y)}$$

$$j = \sqrt{-1}$$

$$f\ x,y\ = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{(ju\frac{2\pi}{M}x+jv\frac{2\pi}{N}y)} \quad (1)$$

La función f(x,y) representa el píxel de la imagen en la coordenada (x,y) y toma valor real positivo. Por otra parte, F(u,v) es una función que representa la transformada de *Fourier* discreta, mediante números complejos. Estos se representan mediante su módulo y la fase en lugar de hacerlo mediante la parte real e imaginaria.

El módulo y la fase de un número complejo vienen definido según las expresiones indicadas en (2).

$$Modulo\ F\ = \sqrt{real(F)^2 + imaginaria(F)^2}$$

$$Fase\ F\ = \text{atan}(\frac{imaginaria(F)}{real(F)}) \quad (2)$$

El módulo indica cuántas componentes de una cierta frecuencia están presentes, mientras que la fase indica la localización de las componentes frecuenciales.

La DFT suele representarse mediante el módulo, centrando el origen de coordenadas (0,0) en el centro de la imagen, de modo que las componentes de baja frecuencia queden centradas y las altas frecuencias en los extremos de la imagen. Para implementar dicha transformada se utilizará la transformada rápida de *Fourier* (FFT), un algoritmo que obtiene el mismo resultado que la DFT con sólo n·log(n) operaciones aritméticas (en lugar de n^2 operaciones mediante el uso de la transformada directa), lo que se traduce en una reducción del tiempo de procesamiento [2].

➢ Componentes conexas

Una componente conexa no es más que un conjunto de píxeles que se han agrupado a partir de cierta característica que los identifica dentro de una imagen. Esta agrupación permite entre otras cosas, separar partes de una imagen, encontrar relaciones entre los elementos de una misma componente o incluso relaciones entre distintas componentes, así como aplicar ciertas operaciones y filtros a porciones de una imagen.

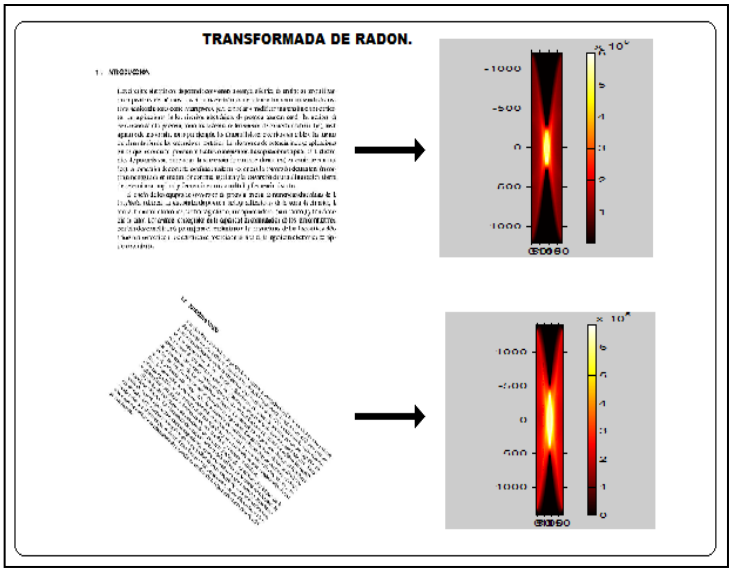
El problema de localización mejor resuelto hasta ahora es el de encontrar los bloques de caracteres (párrafos) en una hoja de texto. Esto puede conseguirse realizando un proceso de "diezmado que favorezca el color del texto". Aquí, la imagen inicial es de dos niveles: el blanco y el negro [10].

➢ Transformada Radón

Al trabajar con imágenes, el objetivo de tenerlas digitalmente es obtener información de ellas para un proceso determinado con el cual se quiere trabajar, la mayoría de veces el reconocimiento de las características que posee una imagen se puede desarrollar a partir de transformadas que permitan la detección de líneas, la transformada de radón es una de estas que se encarga de detectar líneas rectas.

La Transformada Radón realiza las proyecciones de una imagen en función de un eje imaginario que gira alrededor de la imagen, estas proyecciones describen una serie de ángulos, que indican la inclinación de la línea analizada, en la Figura 10, se indica un ejemplo grafico de la transformada de Radón para una imagen con inclinación y una imagen sin inclinación. La transformada de la imagen rotada es más intensa, debido a la rotación de las líneas presentes.

Figura 10 Transformada Radón para imagen de texto con 0° y 45° respectivamente. (Original).



Fuente: Autores.

La transformada Radón se define mediante la expresión (3):

$$R_\theta\ x' = \int_{-\infty}^{\infty} f\ x'cos\theta - y'sin\theta,\ x'sin\theta + y'cos\theta\ \ dy'$$

$$donde\ \begin{matrix} x' \\ y' \end{matrix} = \begin{matrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{matrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (3)$$

x = líneas de la imagen.

y = columnas de la imagen.

θ = ángulo de inclinación.

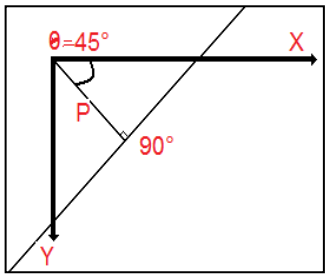
Como se indicó en el anterior párrafo la función de esta transformada permite encontrar el ángulo de rotación de la imagen. Una vez determinado el ángulo es necesario realizar una corrección, se deben realizar las proyecciones Horizontales girando en cada iteración la imagen o eje el número de grado adecuado, con cierto

valor de resolución, para así buscar el máximo valor en dicha proyección. Una vez determinado el ángulo de inclinación también se puede usar la función “imrotate” para su rotación [2, 26].

➢ Transformada Hough.

La transformada de Hough realiza la misma operación que la transformada de Radón, reconoce líneas rectas, sin embargo también permite encontrar y reconocer formas de líneas circulares. Hough consiste en que para cada punto que se desea averiguar si es parte de una línea se aplica una operación, con lo que se averiguan las posibles líneas de las que puede ser parte del punto. Hough interpreta la ecuación $p = xcos\varphi + ysin\varphi$ como una ecuación con parámetros (*x,y*) y variables (*p*, φ). Todas las parejas (*p*, φ) que cumplen la ecuación $p=x\cos\varphi+y\sin\varphi$ (4), para coordenadas dadas *x* y *y* describen líneas rectas que pasan por el punto (*x,y*).

Figura 11 Representación de cómo funciona el algoritmo Hough.

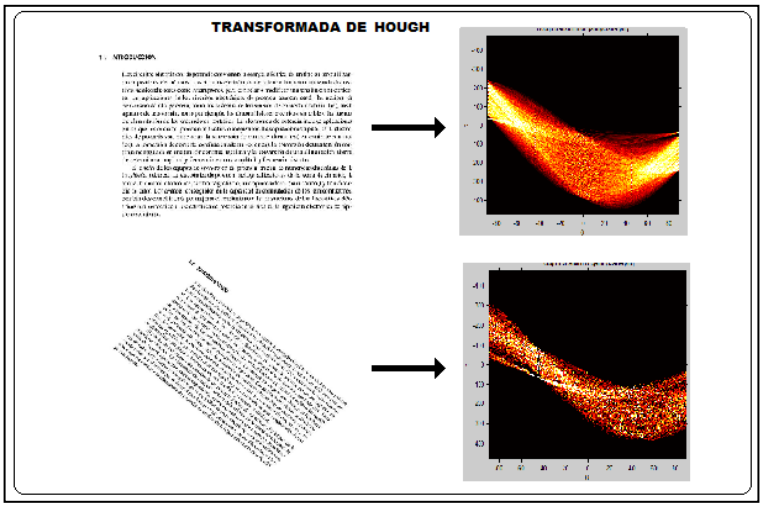


Fuente: Autores.

Usando el método de Radón contamos todos los puntos (*x,y*) contenidos en la región $R_{p,\varphi,\delta}$, es decir (*x,y*) ϵ $R_{p,\varphi,\delta}$. A partir del método de Hough acumulamos todas las líneas (*p*, φ) que pasan por el punto (*x,y*) en una celda y que satisfacen aproximadamente la ecuación de la línea, es decir otra vez (*x,y*) ϵ $R_{p,\varphi,\delta}$.

Para la representación de línea, usamos la función descrita en (4), donde p es la distancia desde el origen a la línea a lo largo de un vector perpendicular a la línea, y φ es el ángulo entre el eje X, y este vector. Al usarla se crea una matriz de espacio donde sus filas y columnas corresponden a los valores de p y φ respectivamente. Se calcula p para cada φ, los máximos valores determinan el ángulo de la línea.

Figura 12 Transformada Hough para imagen de texto con 0° y 45° respectivamente. (Original)



Fuente: Autores.

Con los conceptos anteriormente explicados se busca dar al lector una idea más completa de la forma en cómo funcionan algunas de las herramientas importantes usadas en este proyecto, esta investigación busca de manera general plantear una solución que sea de ayuda para la comunidad invidente. Es un diseño de un software que realice la traducción mediante herramientas de procesamiento digital de imágenes y las especificaciones de dispositivos de entrada y salida para el proceso.

El reconocimiento de caracteres tiene gran aceptación y es un proceso importante en la industria y ya se mencionó las diferentes opciones donde se ha utilizado y el software que realiza esta operación. [20, 25, 26].

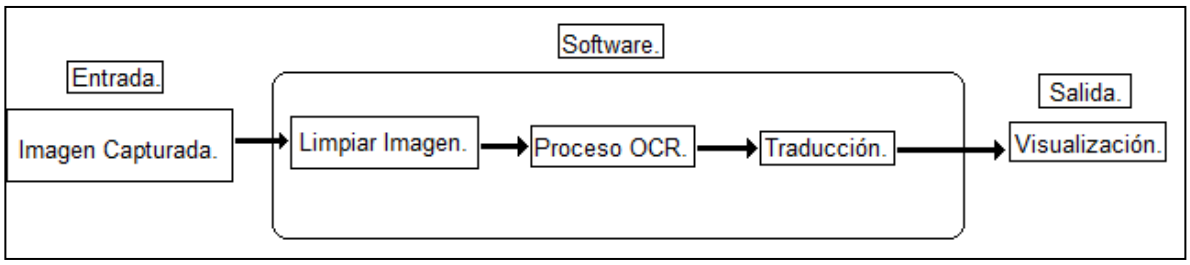
3. DESCRIPCION DEL SISTEMA SOFTWARE

En este capítulo se documenta y explica, las etapas en las que se divide el proceso de software, el cual es el encargado de:

- Reconocer la imagen capturada y realizar un análisis de pre-procesamiento para obtener una imagen limpia (sin inclinación ni ruido).
- Usar técnicas computacionales necesarias para obtener un algoritmo que realice OCR (Reconocimiento Óptico de Caracteres).
- Traducir la cadena de caracteres que tiene como salida el proceso de OCR a lenguaje Braille.

Este capítulo se explica a manera de bloques y con diagramas de flujo el diseño realizado para este proceso, indicando así las especificaciones necesarias en cada etapa para el funcionamiento del algoritmo.

Figura 13 Diagrama de bloques pasos para código.



Fuente: Autores.

3.1 Pre-procesamiento.

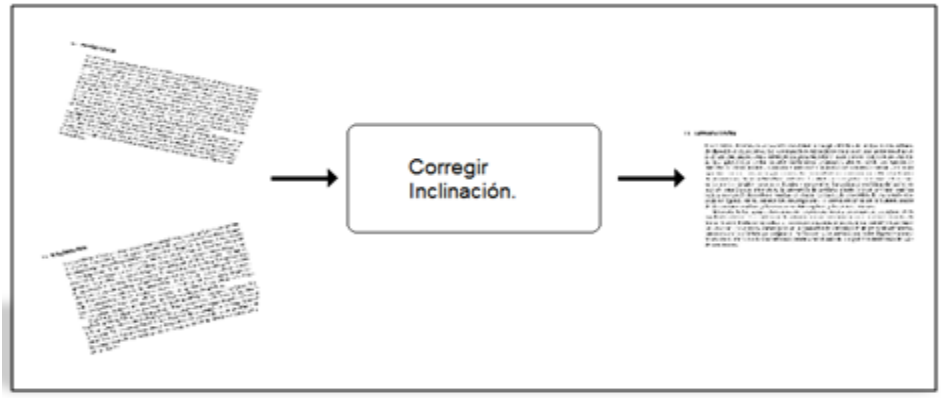
Etapa inicial del proceso. Se debe tener una imagen limpia, para ello se realiza un análisis de la imagen que permita obtener el grado de inclinación del texto y su respectiva corrección, en el apartado anterior se indicaron algunos métodos y propuestas para realizar este proceso, sin embargo necesita de algunas funciones previas que garanticen el correcto arreglo, como la etapa de binarización y

localización de bordes para la detección de ángulos y el uso de transformaciones especiales para la corrección de inclinación.

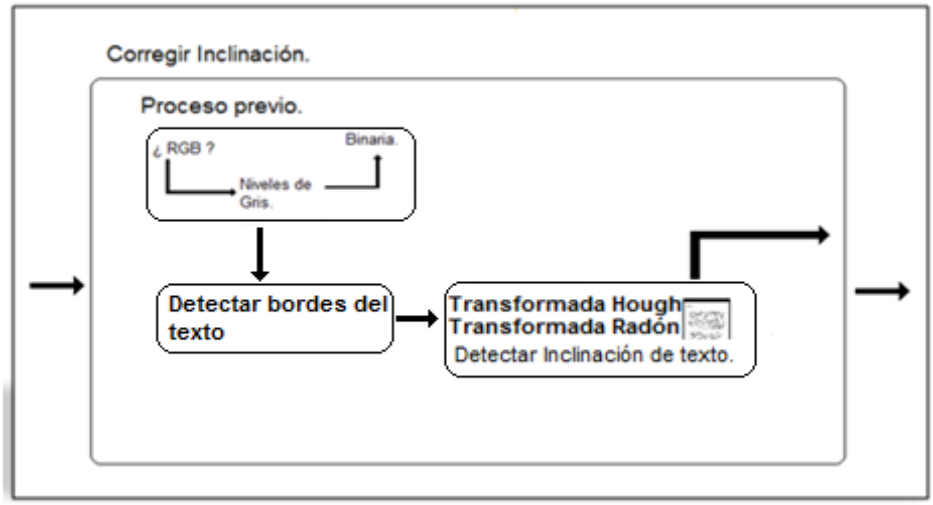
➢ Corregir inclinación.

La cámara web, está configurada para capturar una imagen en formato RGB, es una imagen a color que se debe convertir a nivel de grises para facilitar cálculos y posteriormente obtener la imagen en su mínima expresión sin pérdida de datos, proceso que se conoce como binarización, este proceso es necesario para utilizar ya sea la transformada de Hough o la transformada de radón para detectar la inclinación.

Figura 14 (a) Diagrama ilustrativo para corregir inclinación, (b) Proceso interno corregir inclinación.



(a)

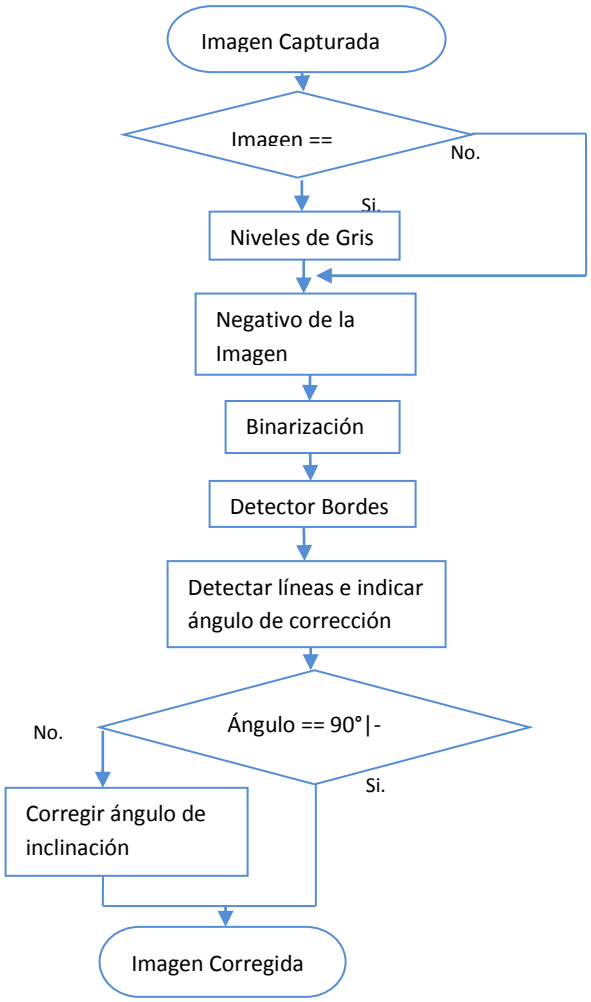


(b)

Fuente: Autores.

El grafico en forma de bloques de lo que se debe realizar en esta etapa del proceso se indica en la Figura 14, ya sea con la transformada de Hough o Radón lo esencial es detectar el ángulo de inclinación del texto. En la Figura 15, se traducen los pasos a diagrama de flujo para obtener una imagen limpia, es decir apta para entrar al proceso de OCR. Como especificaciones para esta etapa del proceso se indica, que la imagen de entrada es la imagen capturada por la webcam y la imagen de salida debe ser una imagen sin inclinación.

Figura 15 Diagrama de Flujo 1 llamado, Corregir Inclinación.

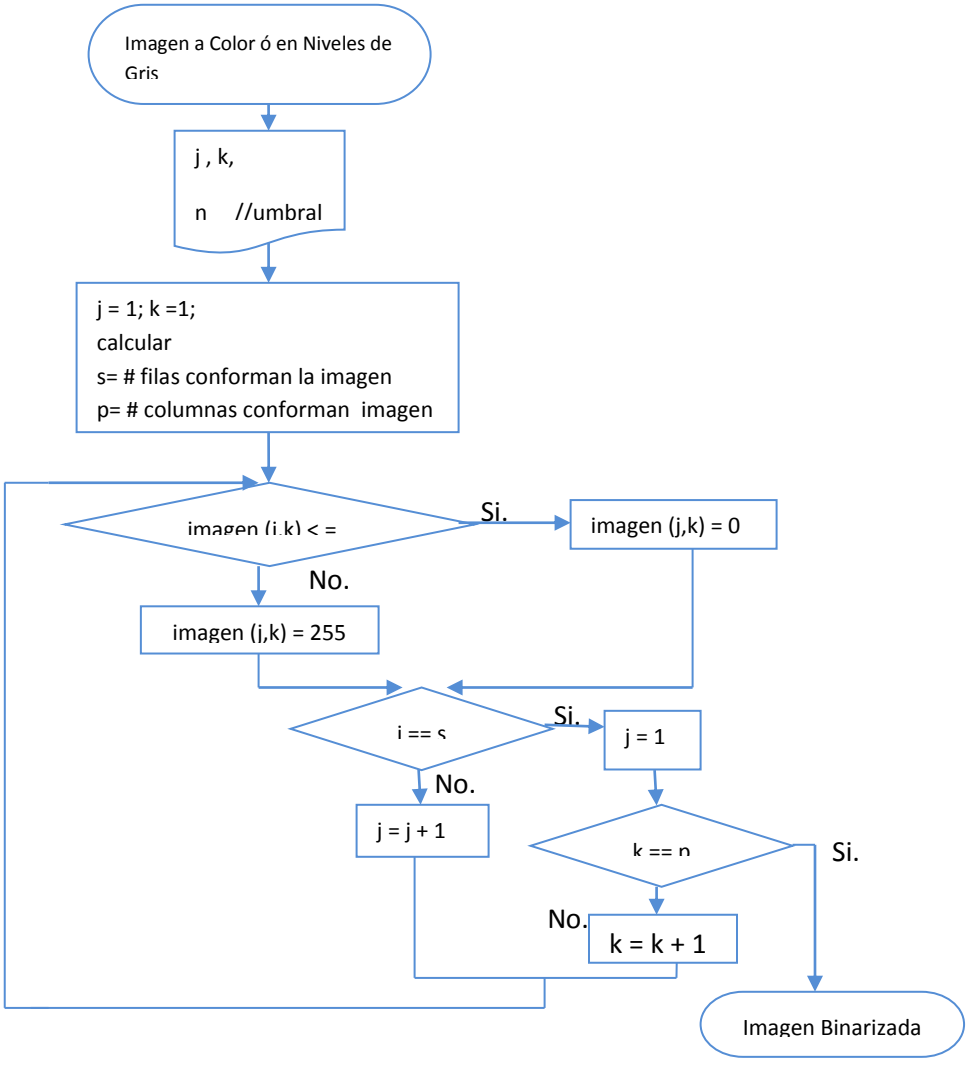


Fuente: Autores.

➢ Binarización.

En el diagrama de flujo de la Figura 15, se indica que para realizar la corrección de la inclinación es necesario un proceso de binarización, busca una representación mínima de la imagen sin perder datos de proceso. En otras palabras cuando se tiene una imagen a color o en nivel de gris, se debe convertir a una imagen binivel (blanco y negro).

Figura 16 Diagrama de Flujo 2 llamado, Binarización.



Fuente: Autores.

En la figura 16, se muestra en diagrama de flujo este proceso que no es más que hacer una comparación pixel a pixel de la imagen con un umbral de decisión. Si es menor o igual al umbral convierte a color negro si es mayor será blanco.

3.2 Segmentación.

La segmentación de imágenes puede ser definida como la partición de una imagen en regiones. Una región (objeto), es un grupo de píxeles conectados que tiene propiedades similares. Las regiones son importantes para la interpretación de las imágenes, corresponden a objetos en la escena. Una imagen puede contener varios objetos y además cada objeto puede contener varias regiones, que corresponden a partes del mismo.

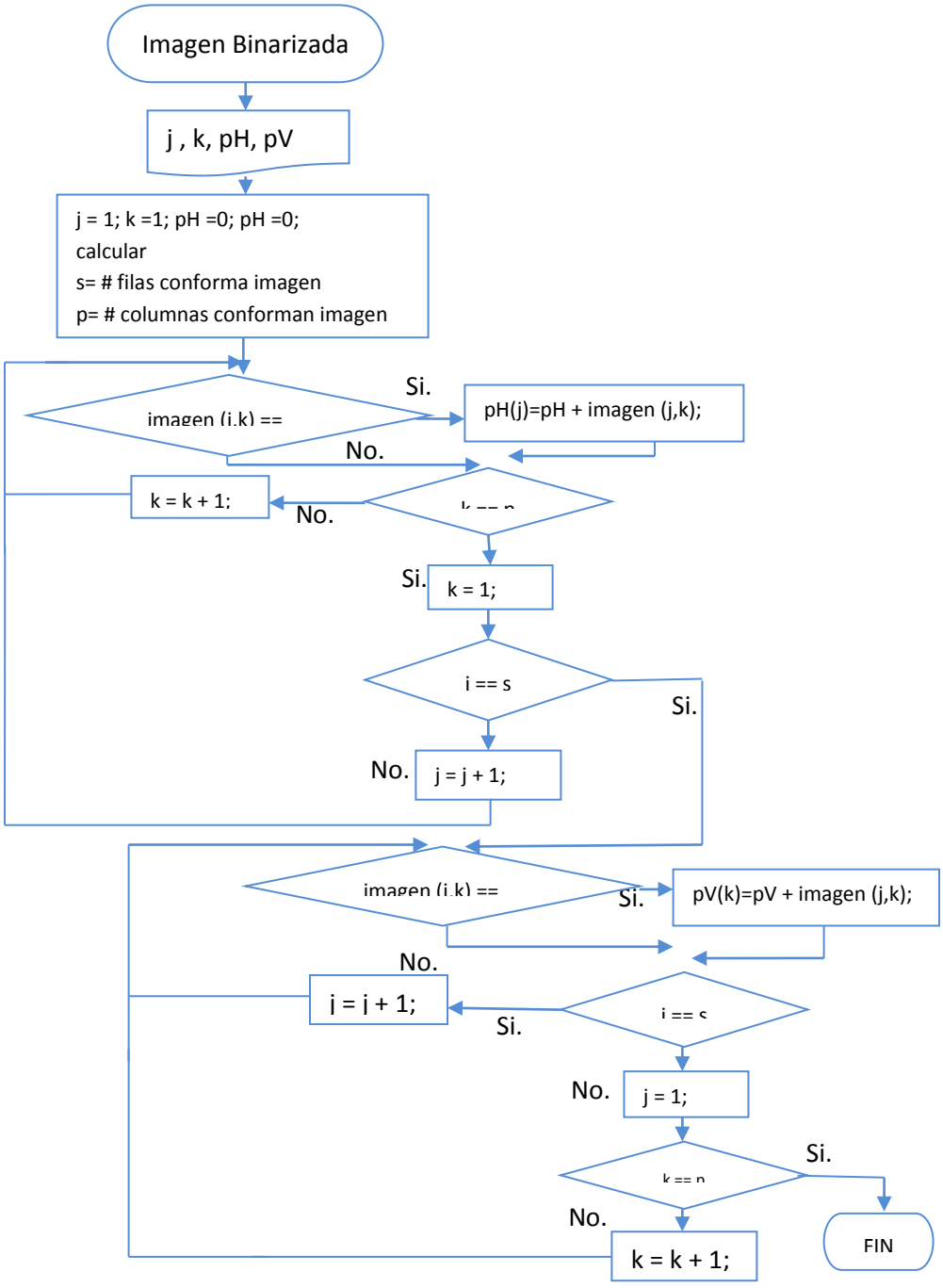
Esta etapa constituye el bloque fundamental de nuestro sistema de reconocimiento. Su función principal consiste en analizar la imagen y extraer los símbolos o caracteres de manera adecuada para el buen funcionamiento de los bloques posteriores.

Para la extracción de caracteres se utiliza la herramienta de las proyecciones ya explicadas (ver estado del arte). Con la interpretación de su resultado se obtiene la ubicación de las componentes que conforman la imagen.

En un primer análisis de segmentación es importante la proyección horizontal, ya que nos indica la cantidad de líneas que conforman el texto, la proyección vertical proveniente de este primer análisis no tiene sentido ya que la combinación de letras de arriba abajo, no permite un reconocimiento claro de la ubicación de estos caracteres.

En la Figura 17, se representa en diagrama de flujo del proceso de proyecciones. Donde pH hace referencia a la proyección horizontal y pV a la proyección vertical, al realizar este proceso se obtienen los dos vectores para la imagen pero no son igual de importante según el nivel de segmentación.

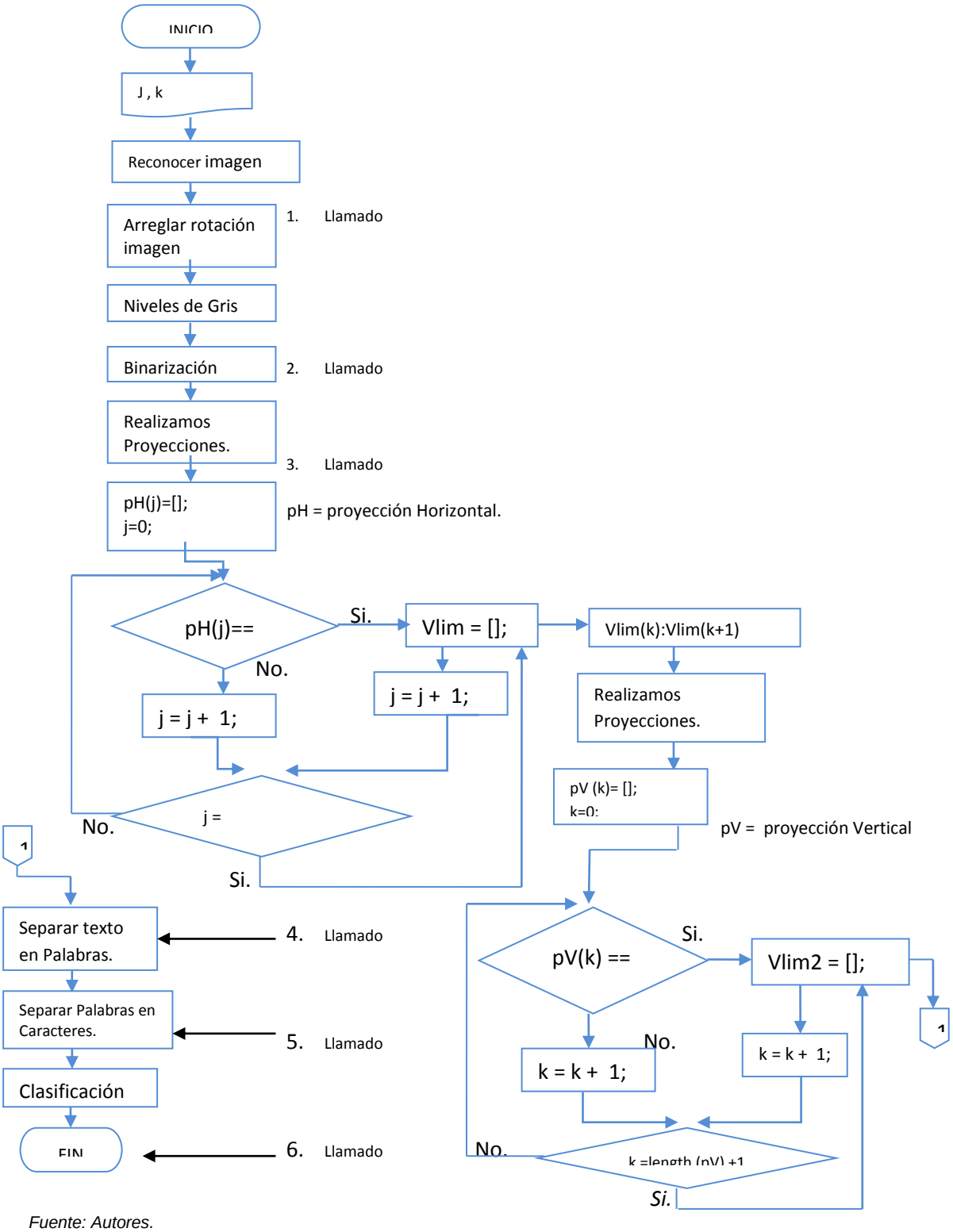
Figura 17 Diagrama de Flujo 3 llamado, proyecciones.



Fuente: Autores.

Una segunda etapa del proceso de segmentación consiste en analizar las líneas encontradas en el primer análisis y separar esta línea de texto en palabras.

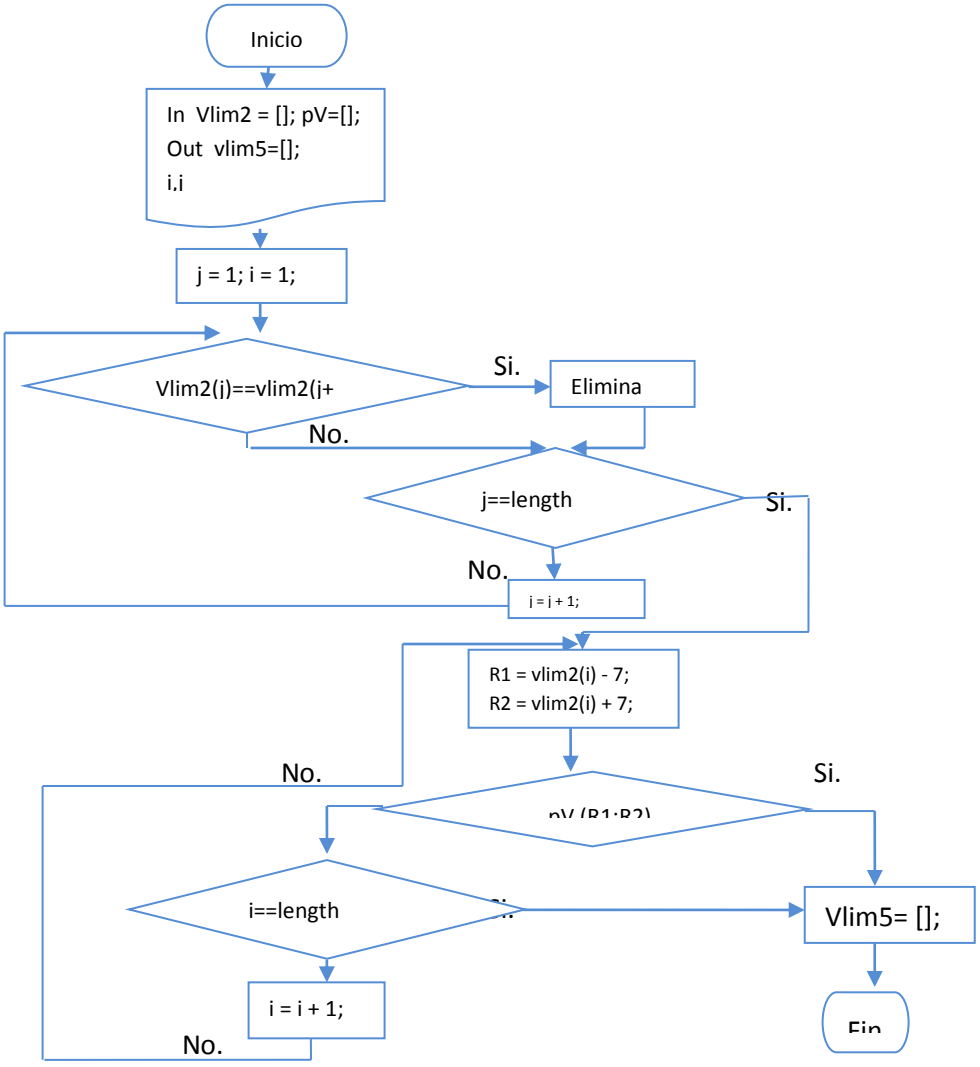
Figura 18 Diagrama de flujo para el proceso de OCR.



Fuente: Autores.

El diagrama de la Figura 18, indica el proceso general de OCR. En él se observa además las dos primeras etapas de segmentación, que tiene como salida un vector de datos en *"1"*. En el diagrama vlim es el vector de límites que indican las líneas de texto y vlim2 es el límite que indica la separación de caracteres. Sin embargo al revisar el propósito del sistema se requiere una nueva etapa de segmentación para obtener definidos los límites de las palabras que conforman las líneas de texto encontradas.

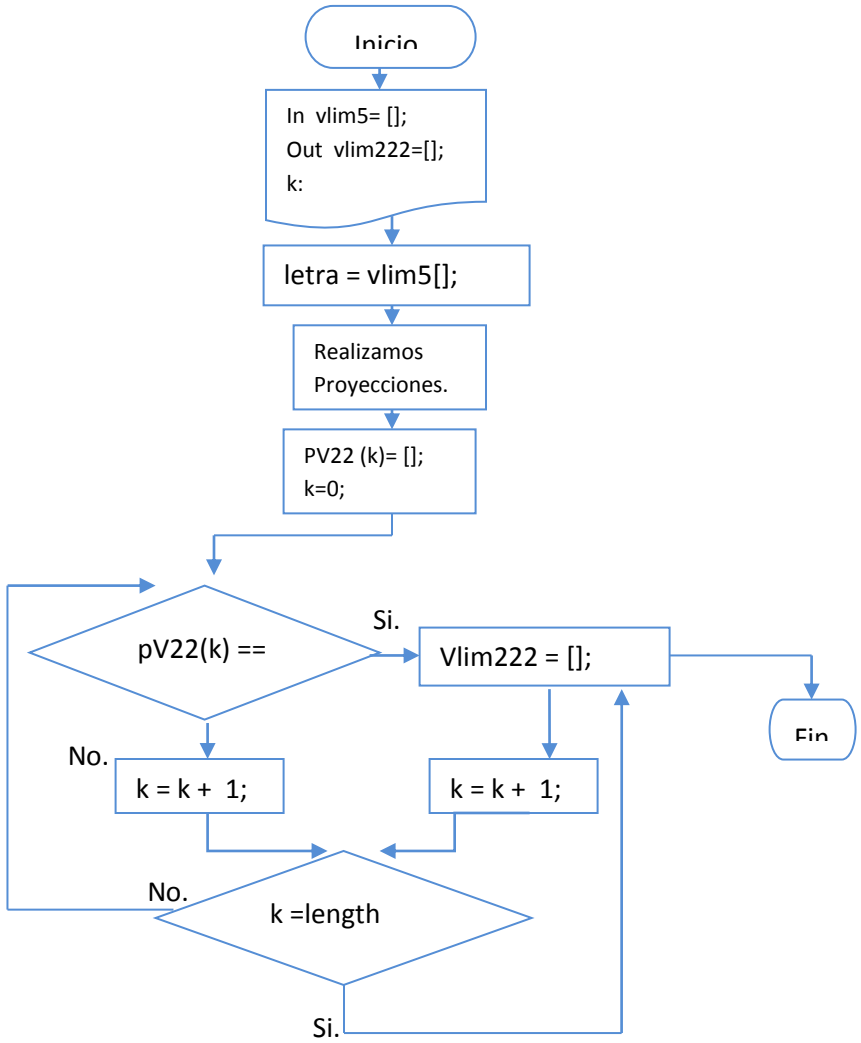
Figura 19 Diagrama de Flujo 4 llamado, Separar texto en palabras.



Fuente: Autores.

Con el vector de límites vlim2 y la proyección vertical de la línea de texto analizada, se buscan los nuevos límites que son encontrados cuando hay un largo segmento en la proyección vertical con valor cero, proceso indicado en la Figura 19.

Figura 20 Diagrama de Flujo llamado 5, Separar palabras en caracteres.



Fuente: Autores.

Finalmente la etapa de segmentación termina con la separación de la palabra en caracteres, proceso señalado en el diagrama de flujo de la Figura 20. En este vector final encontramos los límites superior e inferior del caracter reconocido.

Para llegar a este resultado es importante la interpretación adecuada de las proyecciones según el nivel de segmentación.

Primer nivel……Proyección horizontal.

Segundo nivel……...Proyección vertical.

Tercer nivel……Proyección vertical.

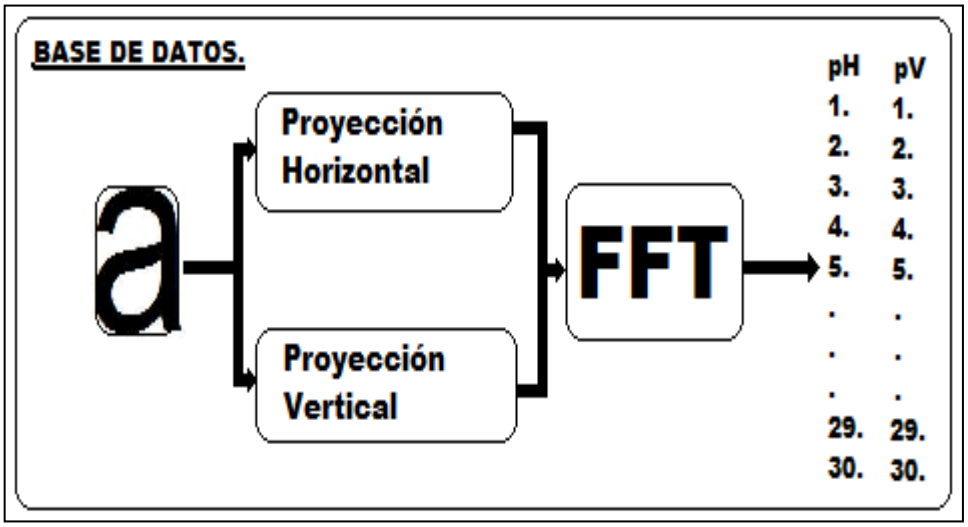
3.3 Clasificación.

Una vez tenemos el texto separado en caracteres procedemos a la clasificación. Etapa final para un sistema de reconocimiento OCR. La clasificación se realiza con la comparación a una base de datos.

➢ Base de datos.

Para el desarrollo de la base de datos, se busca una característica que sea única en cada símbolo, utilizamos la transformada de Fourier y tomamos las primeras 30 componentes que nos determinan la información necesaria de la imagen, es una forma más fácil de clasificación, que el hacer una comparación pixel a pixel de la imagen con algún patrón.

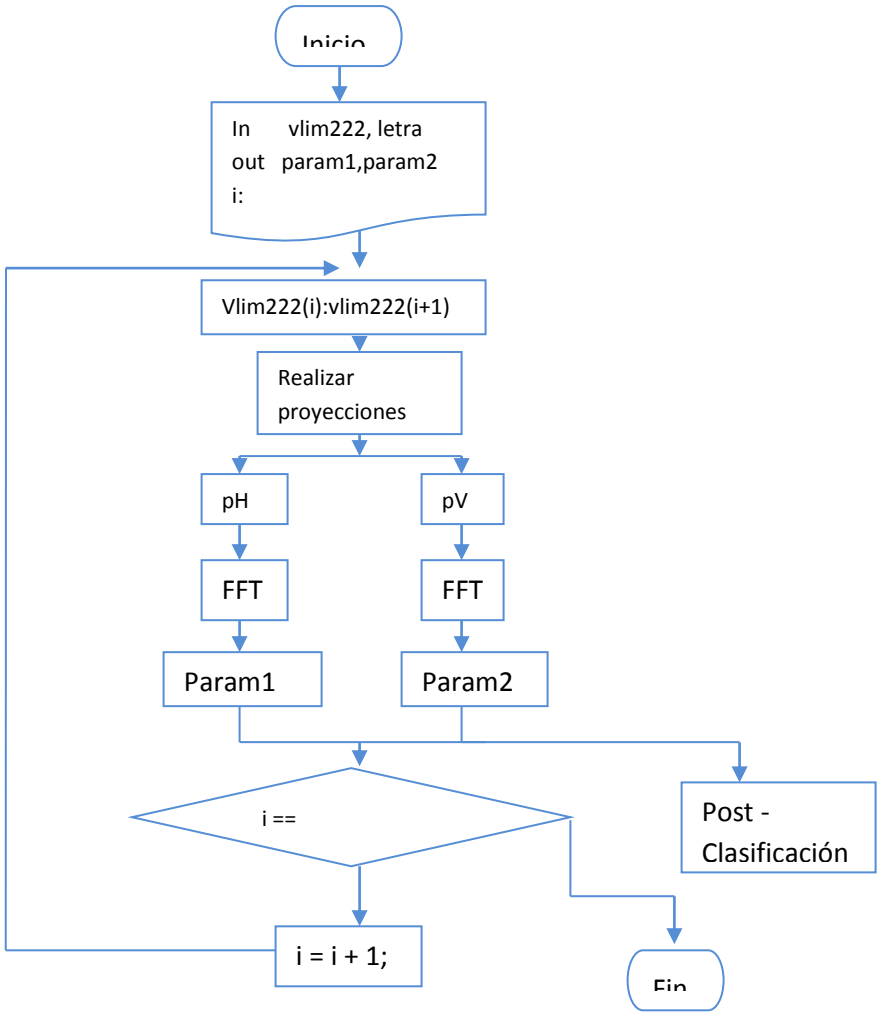
Figura 21 Diagrama de bloques para creación de una base de datos para esta aplicación.



Fuente: Autores.

Mediante la transformada de Fourier se puede caracterizar la morfología de la imagen del caracter analizado. Se utilizarán los 30 primeros componentes para acelerar el proceso de ejecución y caracterizar la imagen mediante una aproximación de la forma. El diagrama de bloques se encuentra en la Figura 21. Y su respectivo diagrama de flujo en la figura 22. Este proceso se utiliza para la creación de la base de datos para esta aplicación y para obtener los patrones de reconocimiento para la clasificación en el proceso de OCR.

Figura 22 Diagrama de flujo para creación base de datos y patrones de imagen.



Fuente: Autores.

Para la extracción de características se debe normalizar la imagen de forma que se puedan comparar de forma objetiva todos los elementos bajo el mismo contexto. Esta normalización se realiza en dos etapas primero la eliminación del espacio sobrante alrededor del símbolo y la segunda un escalado a un tamaño de 42 x 24 píxeles.

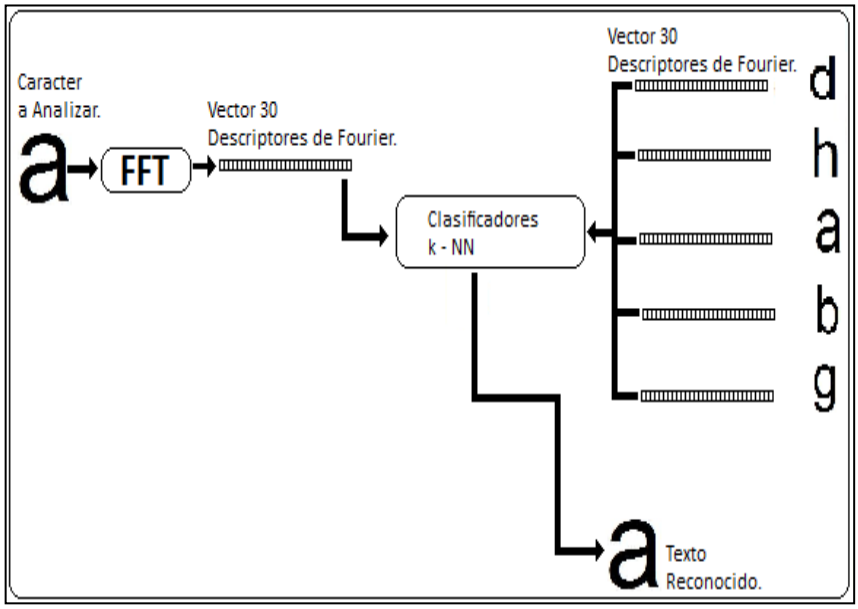
- Clasificación.

Un algoritmo de clasificación por distancia mínima es utilizado para esta etapa del proceso, el k-NN (k-Nearest Neighbours) el cual realiza una comparación entre las distancias de los vectores del caracter analizado y la base de datos, elige la clase más frecuente entre los "k" vecinos más cercanos. Se utiliza la distancia Euclídea que se define mediante la expresión:

$$distancia = \sum_{i=1}^{n} (X_{mi} - X_{ti})^2 \qquad (5)$$

Donde X_{mi} corresponde con la muestra de la base de datos y X_{ti} a la muestra a evaluar.

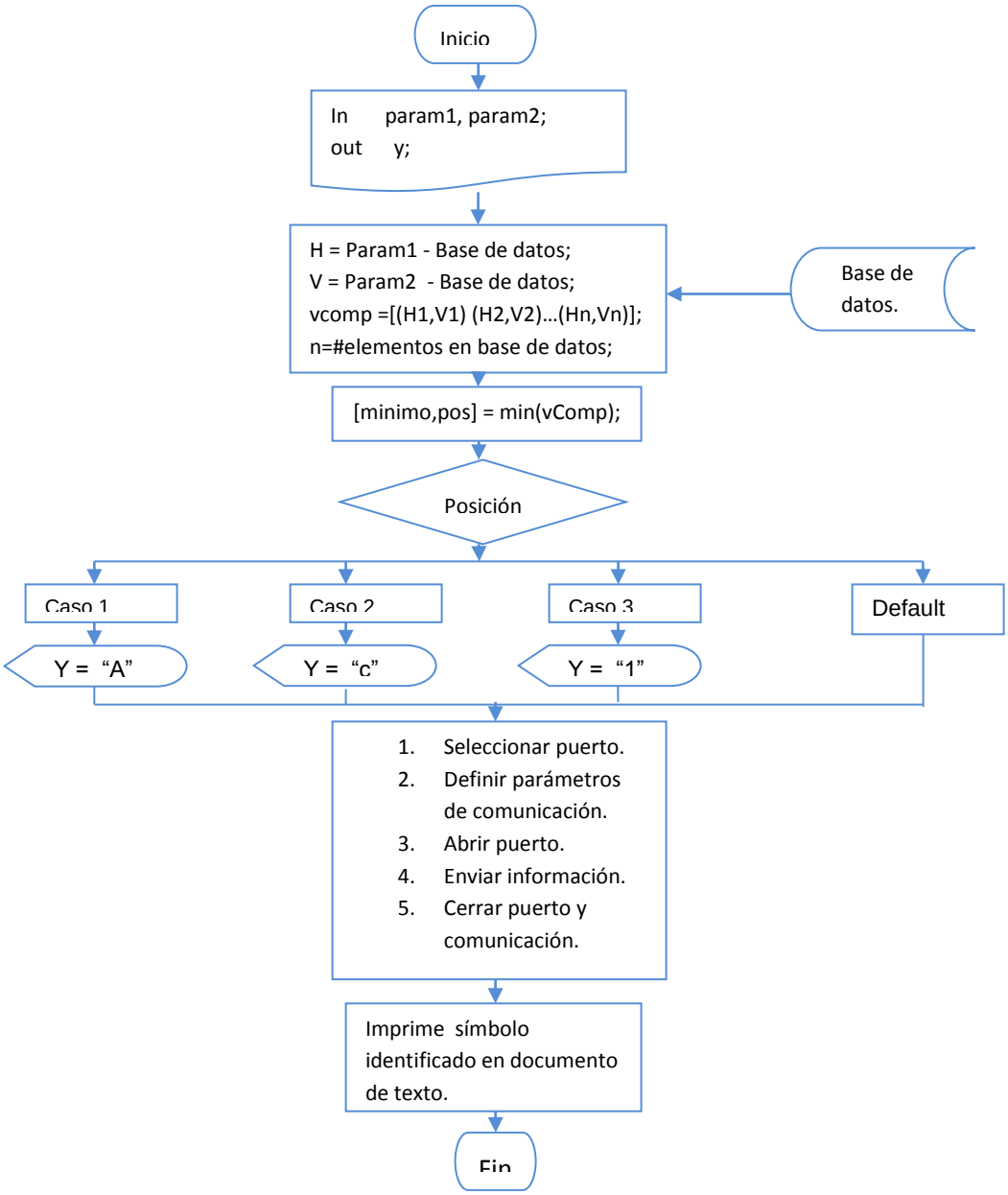
Figura 23 Proceso de clasificación explicado en diagrama de bloques.



Fuente: Autores.

El diagrama de bloques indicado en la Figura 23, enseña el proceso con los pasos a seguir para encontrar el caracter realizando una comparación. En la Figura 24 encontramos el llamado 6, que es el diagrama de flujo correspondiente al proceso de clasificación y envió de datos.

Figura 24 Diagrama de flujo para clasificación y envió de datos.



Fuente: Autores.

El proceso realizado describe una comparación del caracter analizado con cada uno de los componentes de la base de datos, y la menor distancia encontrada en este proceso selecciona al caracter reconocido, y continua con el proceso descrito en este diagrama.

4. PRUEBAS DEL SISTEMA SOFTWARE

4.1 Introducción

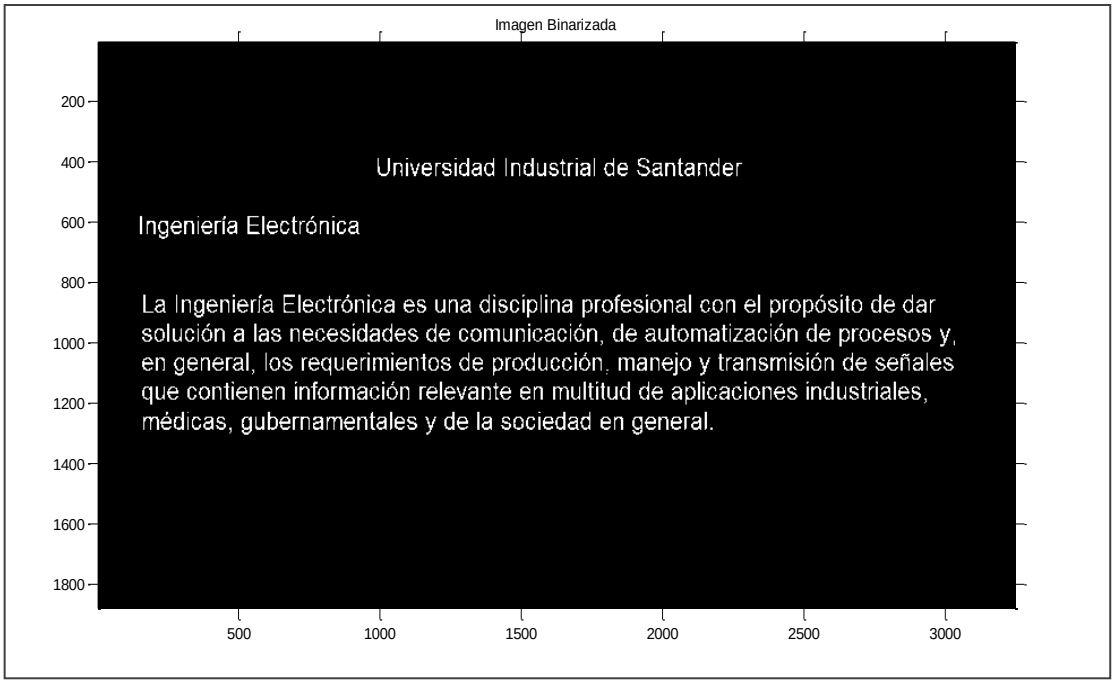
En este capítulo se evidencia los resultados obtenidos en el proceso de diseño, se analizan las gráficas de las pruebas realizadas a cada una de las etapas, binarización, corrección de inclinación, operadores, transformadas y segmentación. Las pruebas del sistema son realizadas con base a la herramienta matemática Matlab.

4.2 Prueba etapa binarización.

Para la etapa de binarización se realiza un barrido pixel a pixel donde se efectúa una comparación con un umbral en busca de obtener una imagen en dos niveles. En la Figura 25, se deja evidencia del resultado de binarización para una de las imágenes de prueba. Es importante el análisis de la imagen y del proceso de binarización, para elegir el umbral adecuado que elimine el ruido presente en la imagen, en esta aplicación son puntos o líneas que no pertenecen al texto. Al elegir un umbral inadecuado se obtendrán resultados errados, la imagen resultante de un umbral mal elegido se indica en la Figura 26, donde se observa que se pierde información de los caracteres.

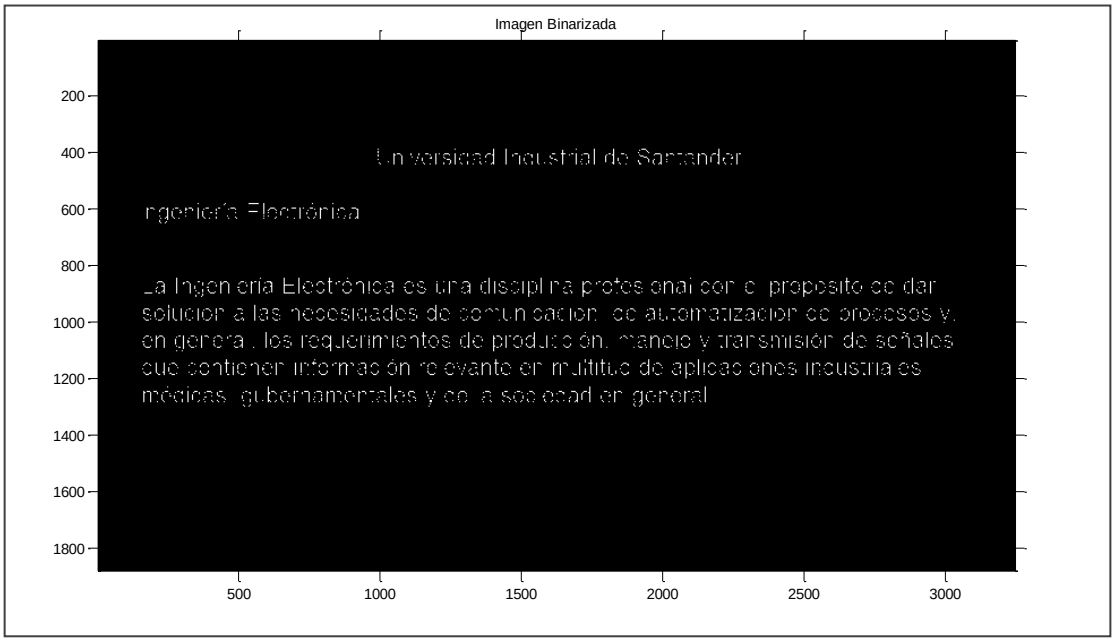
El proceso de binarización se utiliza en dos etapas que son: primero para corregir la inclinación y luego en el proceso principal para tener una imagen en dos niveles. Es un proceso sencillo y lento por qué se debe recorrer toda la imagen en la comparación, pero es una tarea indispensable en el desarrollo del proceso.

Figura 25 Resultado prueba de proceso binarización con umbral adecuado.



Fuente: Autores.

Figura 26 Resultado prueba de proceso binarización con umbral mal seleccionado.



Fuente: Autores.

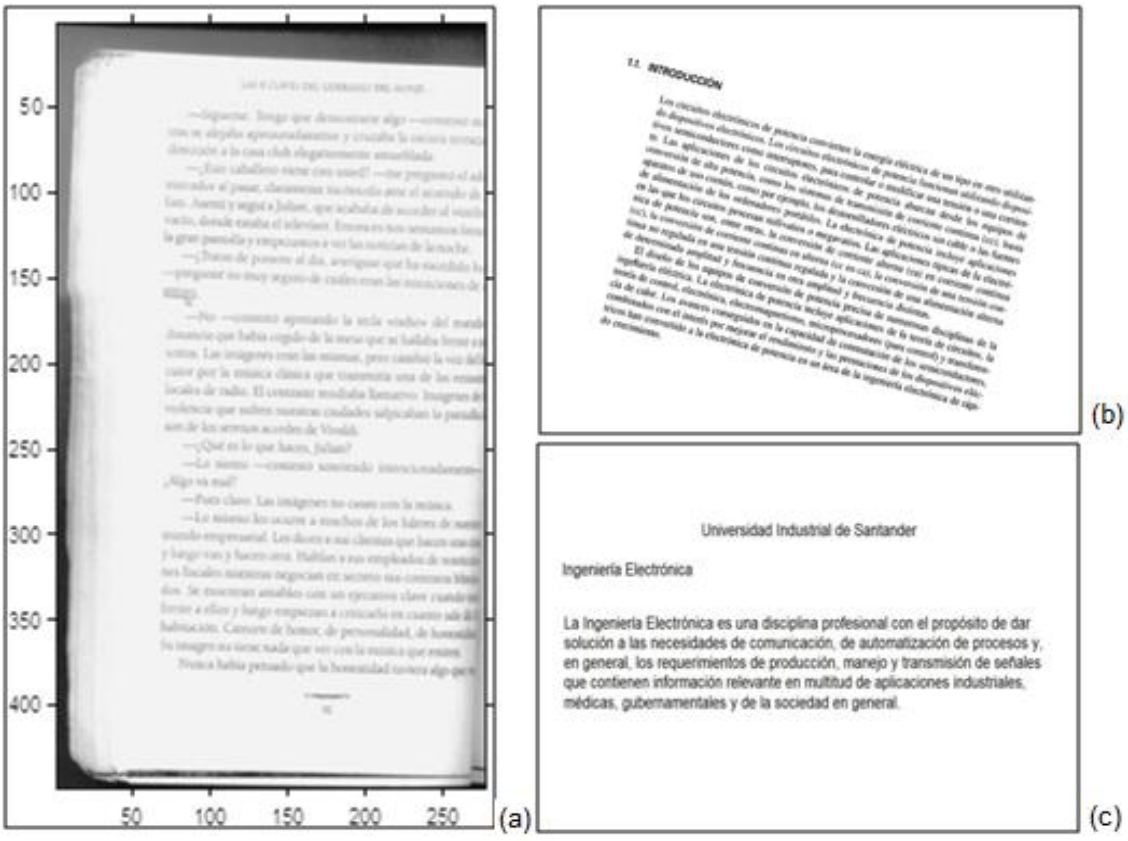
4.3 Pruebas para etapa de corregir inclinación.

Para corregir la inclinación de una imagen existen varios métodos de solucionarlo. Para encontrar el método adecuado se realizaron una serie de pruebas que determinaron la manera más efectiva de realizar el proceso para esta aplicación.

Inicialmente se deben detectar los bordes de la imagen, en especial los bordes laterales, ya sea porque el texto los presenta o porque estos se obtienen al detectar los bordes de las letras que conforman el texto, a continuación se indican las pruebas realizadas a los detectores de borde más efectivos en teoría.

Las imágenes de la figura 27 tienen una inclinación de 2.5°, 15° y 0° respectivamente.

Figura 27 Imágenes originales con inclinaciones para prueba con detectores de borde (a) 2.5°, (b) 15° y (c) 0°.



Fuente: Autores.

➢ Operado ‘prewitt’

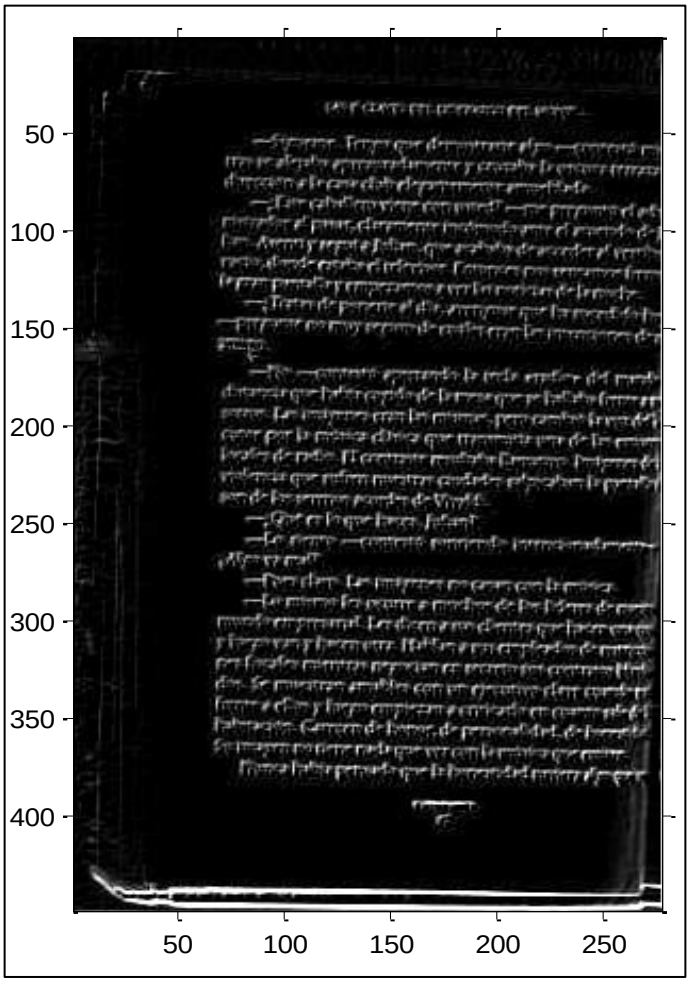
Resalta los bordes horizontales y verticales como resultado de aplicar el gradiente para obtener las matrices que constituirán las máscaras. Se aplico este operador a las tres imágenes indicadas en la Figura 28, utilizando la herramienta “edge”.

➢ II= edge(image1,'prewitt');

Donde:

Image1 = imagen analizada
II = imagen respuesta con bordes detectados

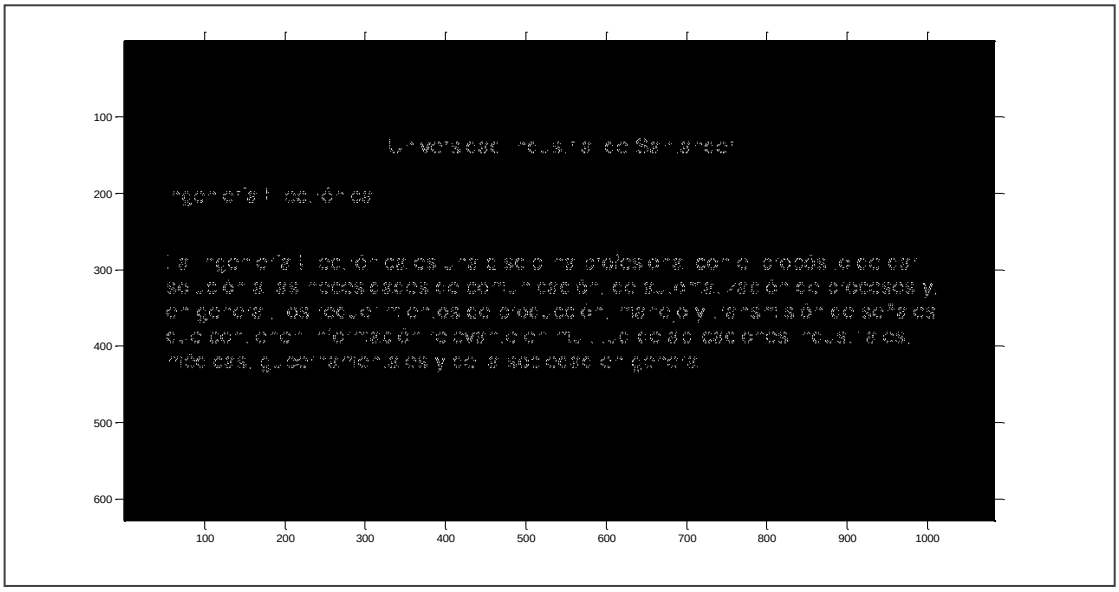
Figura 28 Resultado de detección de bordes con operador ‘Prewitt’ para imagen con 2.5° de inclinación.



Fuente: Autores.

En la Figura 28, se observa como el operador de “prewitt” resalta las letras que conforman el documento, pero no tiene en cuenta las líneas que conforman el límite de la hoja del texto. A pesar que los bordes se resaltan no son adecuados para esta aplicación. El proceso indico que el ángulo era de 0° un dato erróneo ya que la imagen original presenta un error de 2.5°. Con esta sola prueba se concluye que este operador no es adecuado sin embargo se indican los resultados para las otras dos imágenes presentadas en este documento en la Figura 27.

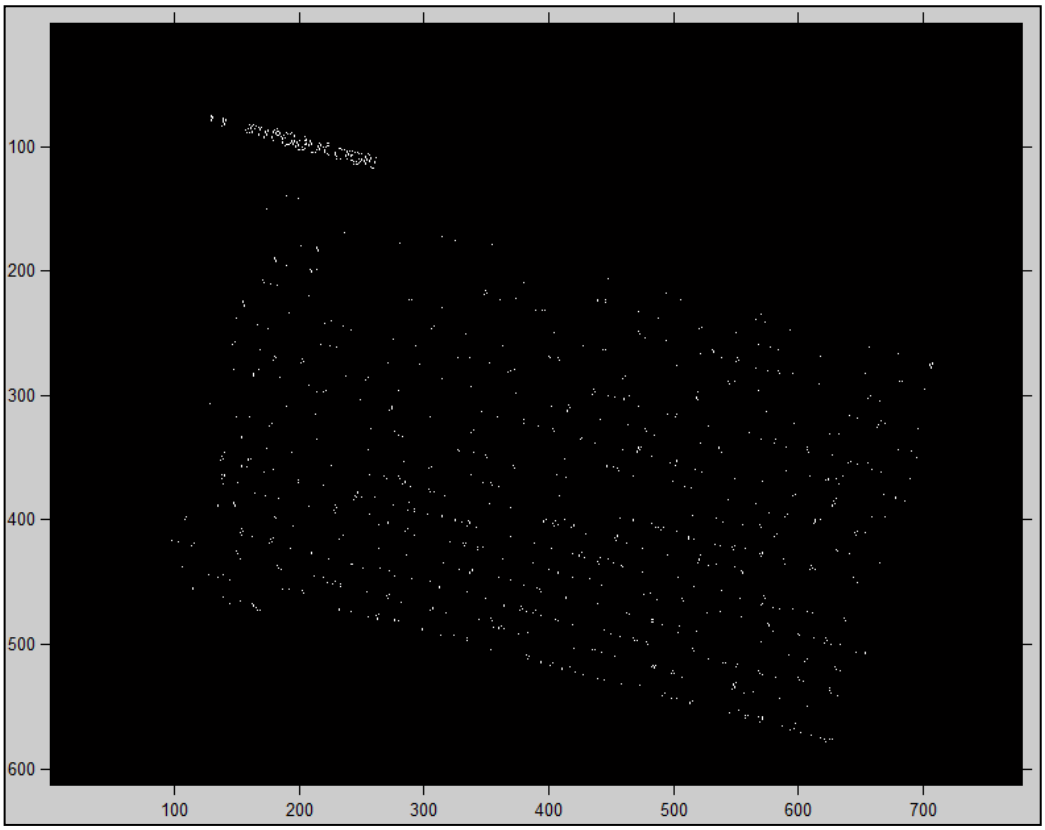
Figura 29 Resultado de detección de bordes con operador ‘Prewitt’ para imagen sin inclinación.



Fuente: Autores.

Estas imágenes son de mayor tamaño sin borde lateral de texto. ‘Prewitt’ detecta alguno de los bordes de las letras que conforman el texto, estos segmentos detectados crean una línea punteada que es reconocida por una transformada encargada de detectar líneas para encontrar ángulo de inclinación. En los dos casos se detecta el ángulo de inclinación. En la Figura 29, el ángulo de inclinación es 0°, resultado correcto, no necesitaba corrección, pero en la Figura 30, el ángulo reconocido fue de 75.05°, es decir un error de 0.05°. Error de resolución tenido en cuenta en el algoritmo.

Figura 30 Resultado de detección de bordes con operador 'Prewitt' para imagen con 15° de inclinación.



Fuente: Autores.

El proceso de detección de borde, utilizando el operador "prewitt" genera buenos resultados, sin embargo para esta aplicación como se indica en la figura 28, el resultado no fue acertado y por eso no se utiliza este operador, pero se deja evidencia de sus resultados.

- Operador 'Sobel'

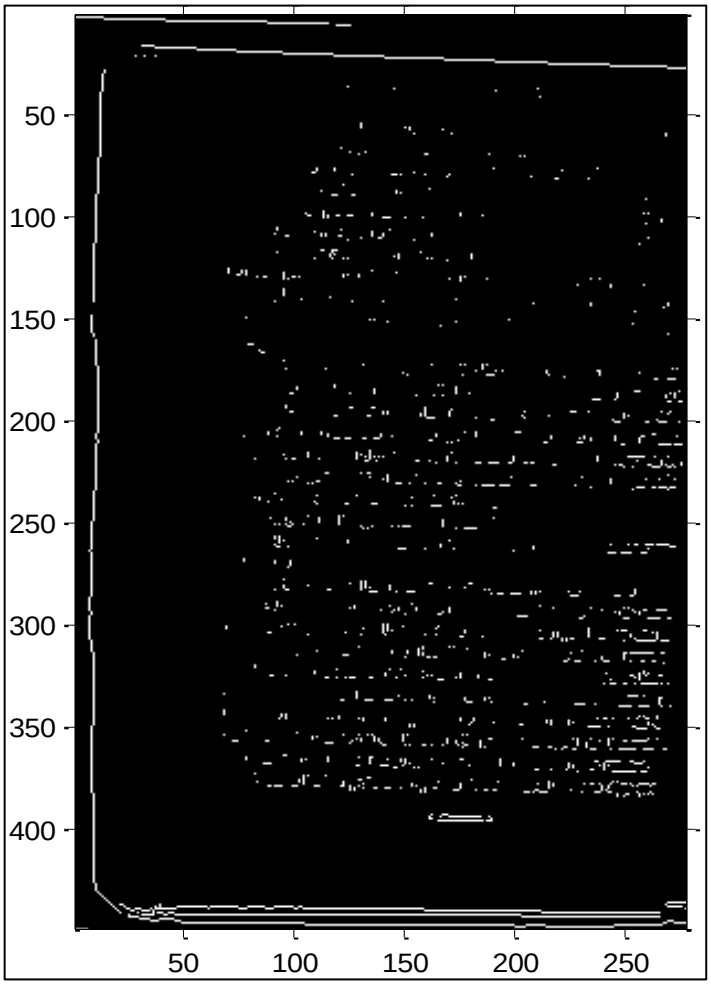
Es un operador diferencial discreto que calcula una aproximación al gradiente de la función de intensidad de una imagen utilizada para detectar borde. Para analizar este operador se uso la herramienta "edge".

```
image=edge(image1,'sobel');
```

Donde:

Imagen1= imagen analizada
image = imagen respuesta con bordes detectados

Figura 31 Resultado de detección de borde con operador 'sobel' para imagen con 2.5° de inclinación.

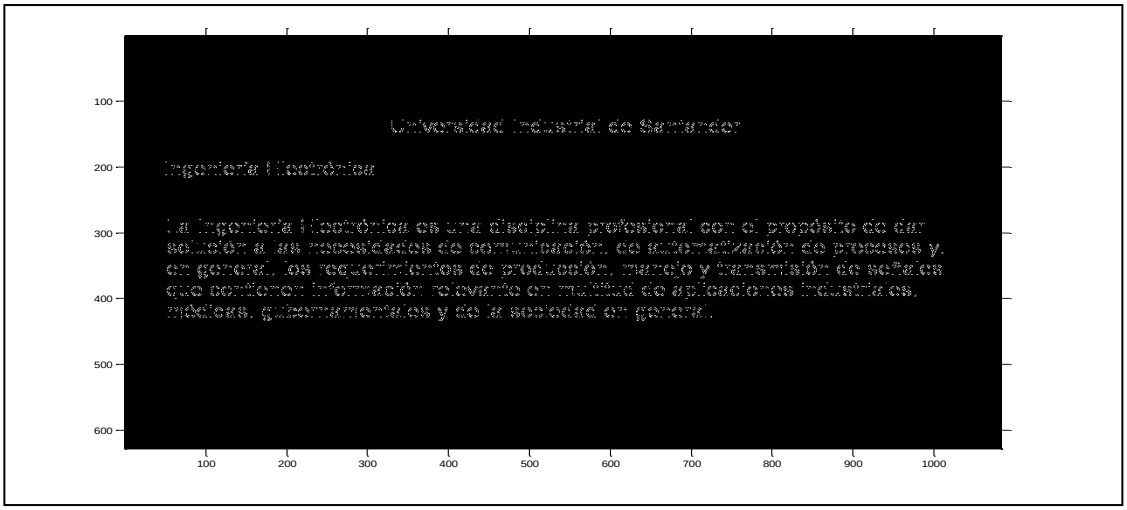


Fuente: Autores.

Se observa claramente la línea lateral la cual define el contorno de la hoja de texto, al utilizar la transformada para detectar ángulo de inclinación del documento, se encontró un ángulo de 2.505° que corresponde con el ángulo de inclinación del texto. Las pruebas a la segunda imagen (Figura 32), presentada en el análisis indican un ángulo de 90°, resultado acertado. En las figuras (31 y 32), la respuesta de este operador no se detecta a simple vista el texto pero los puntos que se

observan crean una línea punteada que es detectada por la transformada encargada de reconocer líneas.

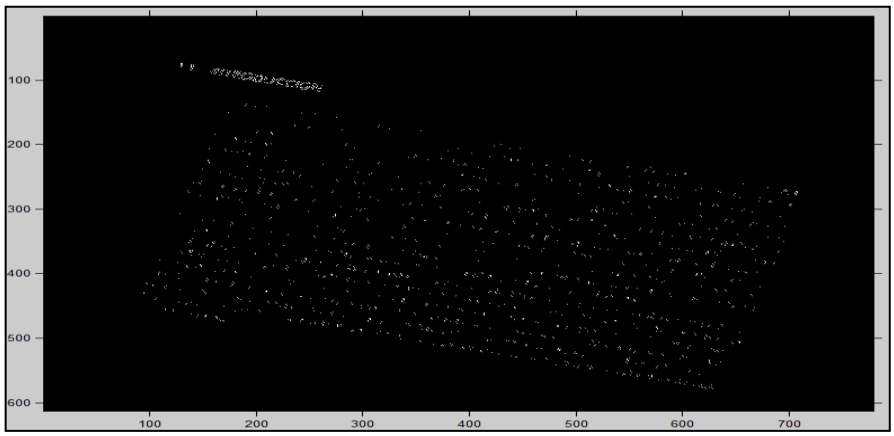
Figura 32 Resultado de detección de borde con operador 'sobel' para imagen sin inclinación.



Fuente: Autores.

Para la tercera imagen analizada con pruebas en este documento, se encontró un ángulo de inclinación del texto de 75.05 °. Las imágenes resultado del operador se indican en las Figuras 31,32 y 33. Los tres resultados son adecuados. Es opcional para el desarrollo de esta aplicación ya que las pruebas realizadas arrojaron buenos resultados.

Figura 33 Resultado de detección de borde con operador 'sobel' para imagen con 15° de inclinación.



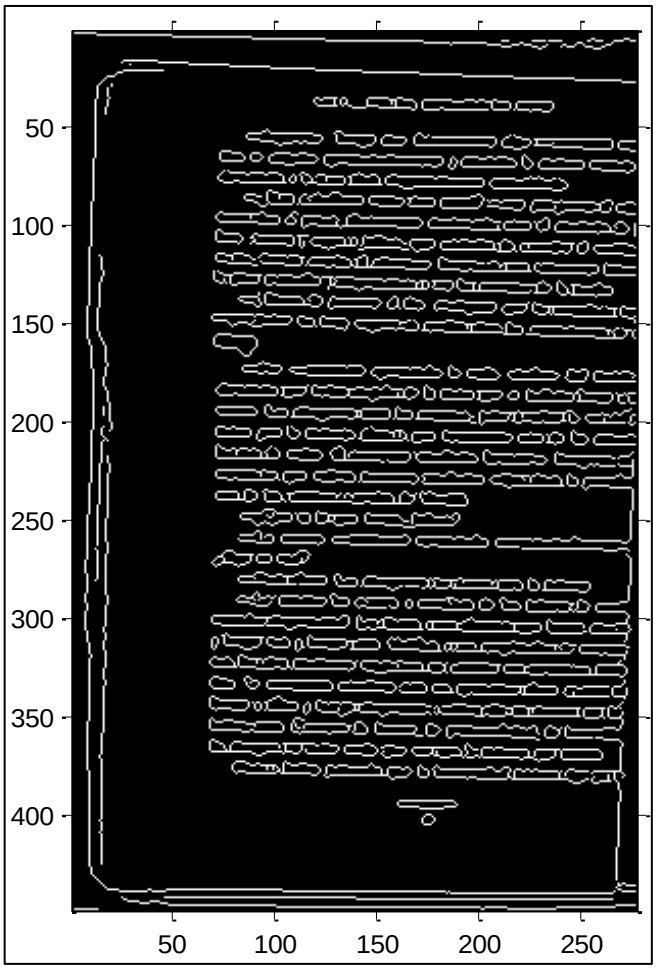
Fuente: Autores.

➢ Operador ‘Canny’

En la Figura 34, se grafica el resultado del operador ‘Canny’ el cual obtiene una descripción de las líneas y formas que encuentra en el documento. Para la prueba realizada a la primera imagen el ángulo detectado fue de 2.5°.

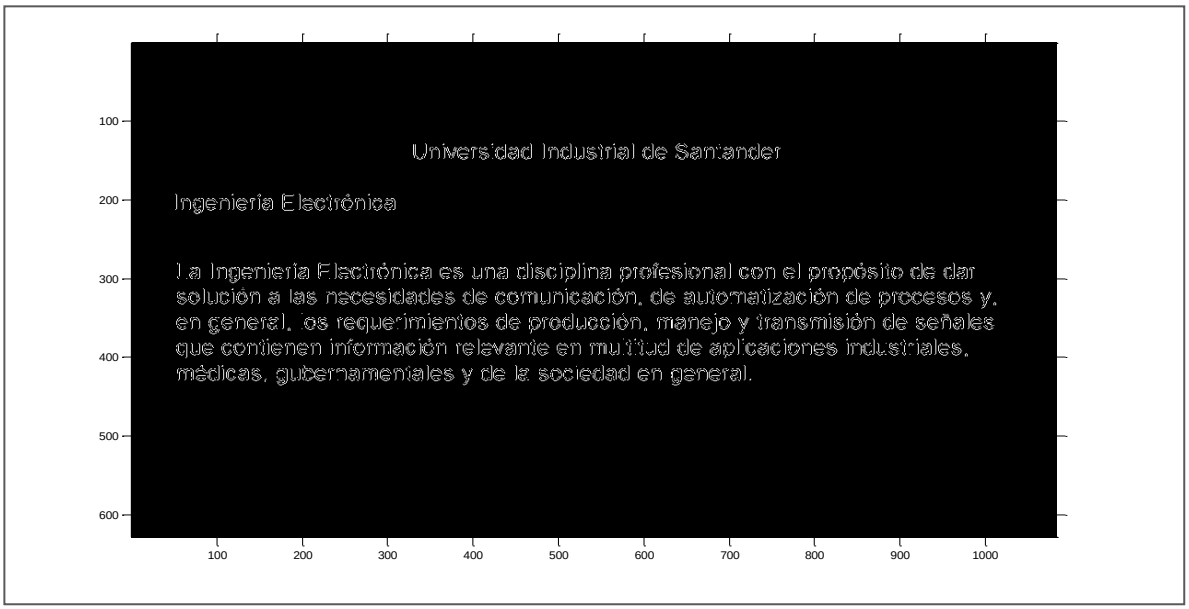
En las otras pruebas realizadas con este operador el resultado fue correcto, se indican gráficamente en la Figura 35, para la imagen sin inclinación y Figura 36, para la imagen inclinada 75°. Los resultados son distintos porque el tamaño de la letra en las dos imágenes de texto es diferente.

Figura 34 Detector de bordes Operador ‘Canny’.



Fuente: Autores.

Figura 35 Detector de bordes Operador “Canny”.



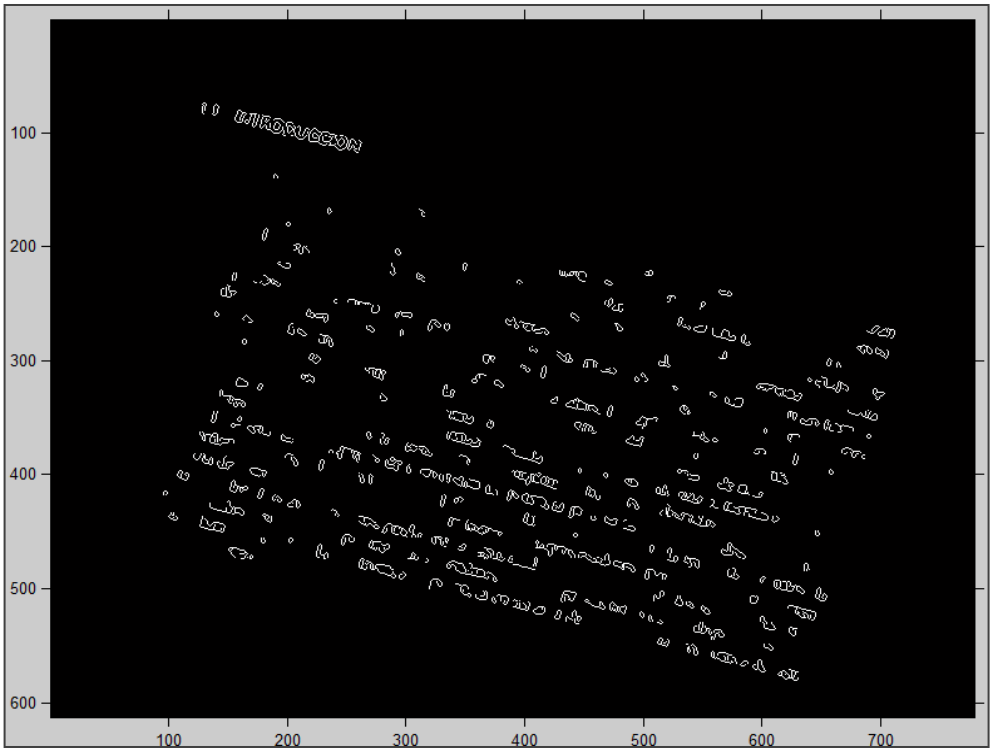
Fuente: Autores.

El operador ‘canny’ presento buenos resultados en las pruebas realizadas, detecto al igual que ‘sobel’ la inclinación en el texto con bordes en los extremos y sin ellos, para elegir el mejor entre los dos, según teoría de ***‘canny’*** un operador se considera óptimo si cumple con:

- Buena detección- marcar el mayor número real en los bordes de la imagen como sea posible.
- Buena localización- los bordes de marca deben estar lo más cerca posible del borde de la imagen real.
- Respuesta mínima - El borde de una imagen sólo debe ser marcado una vez.

Observando las seis imágenes de respuesta de estos operadores y la teoría de borde óptimo se concluye que ‘canny’ presenta una mejor grafica que ‘sobel’ a pesar que en ninguno de los dos casos la detección total de bordes es exacta. Figura 36 en ‘canny’ y Figuras 31, 32 y 33 en ‘sobel’.

Figura 36 Detector de bordes Operador “Canny”.



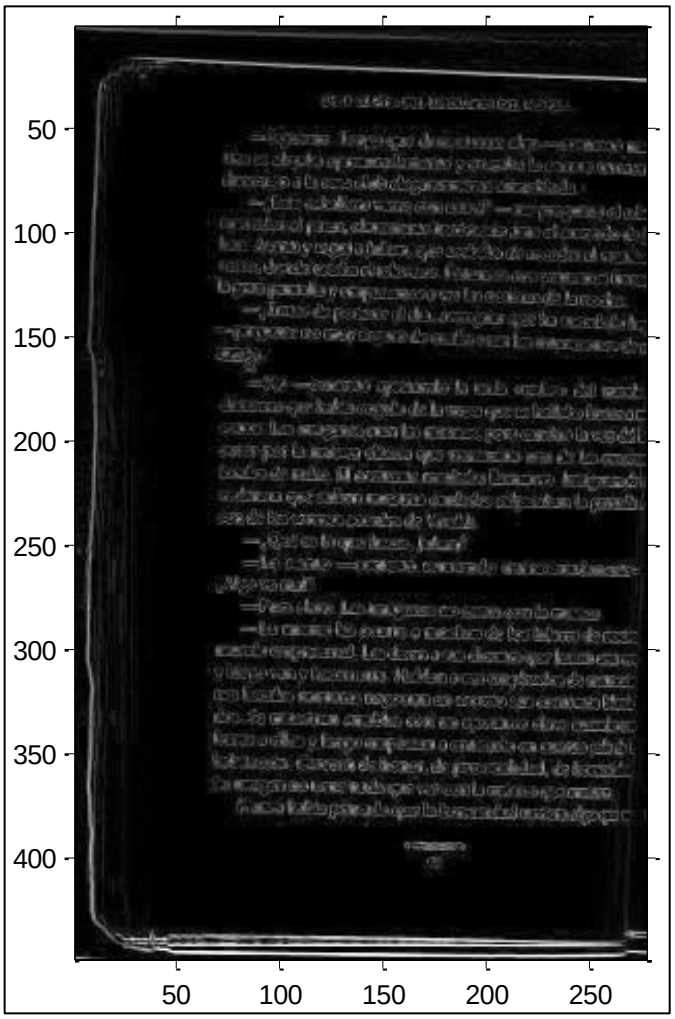
Fuente: Autores.

➢ Operador ‘Roberts’

Es el operador que mejor detección de bordes presenta según la teoría de borde optimo de **‘canny’**. En las tres figuras se observa el borde de la totalidad de las letras detectadas, sin embargo en la prueba para la imagen con 2.5° de inclinación cuando se utiliza este operador el ángulo no fue reconocido, el resultado fue de 0° de inclinación resultado erróneo. La grafica del descriptor de “Roberts” para este primer proceso se muestra en la Figura 37.

Lo esencial es ayudar con los operadores de borde a la transformada usada para detectar el ángulo de inclinación, eliminando algunas características en las letras y resaltando sus límites, y así reconocer más fácil las líneas que se forman. Este operador nos muestra prácticamente la misma imagen siendo excelente en su reconocimiento de bordes pero no adecuado para esta aplicación.

Figura 37 Detector de bordes Operador “Roberts”.



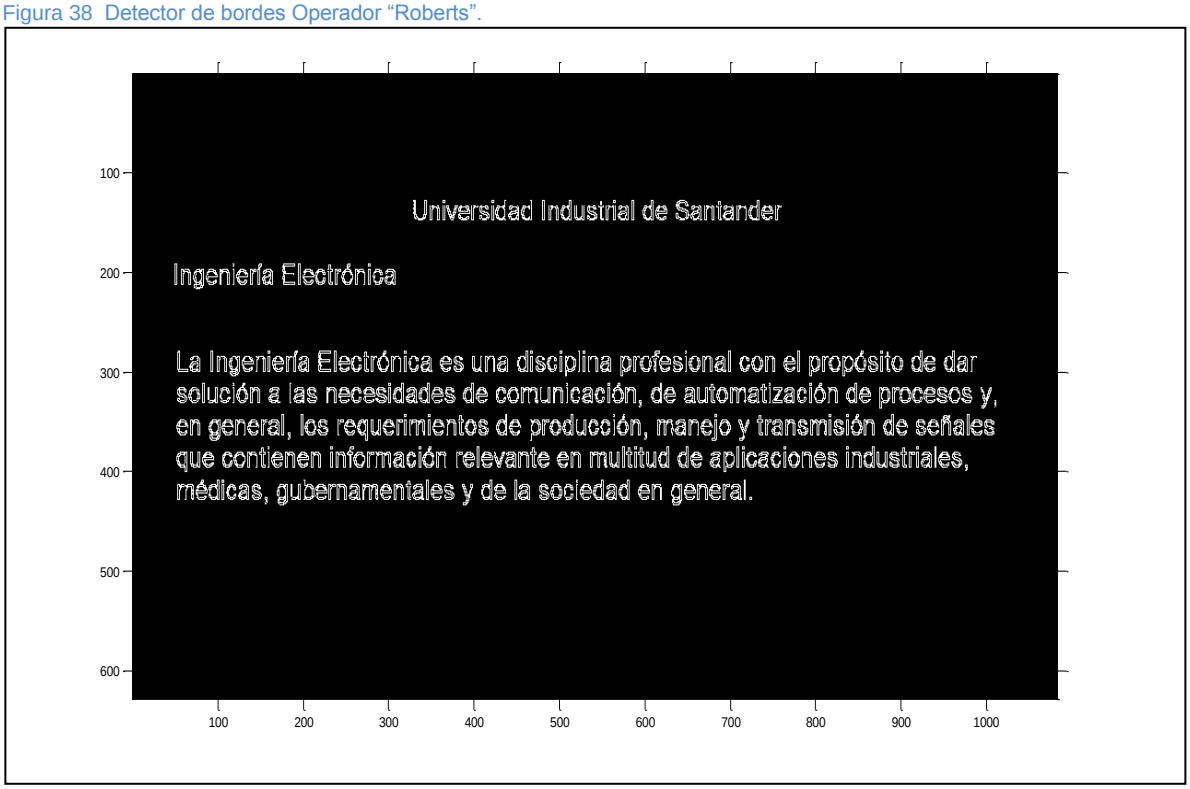
Fuente: Autores.

Los bordes de las letras iniciales de cada línea crean una línea que fue detectada en los otros dos casos por la transformada encargada de encontrar el ángulo de inclinación del documento, en los dos casos fue un reconocimiento acertado y los ángulos encontrados. Las graficas evidencia del proceso de detección de borde con operador “Roberts” se muestran en la Figura 38 y Figura 39.

Los operadores analizados encontraron bordes en todos los casos, sin embargo para una siguiente etapa es necesario que el borde sea reconocido, ‘Prewitt’ y ‘Roberts’ presentaron problemas con imágenes de texto pequeños por eso no son

utilizados en este proceso, entre ‘sobel’ y ‘canny’ que presentaron buenos resultados en todas las pruebas, se encontró en las imágenes que ‘canny’ presenta mejor reconocimiento de borde con respecto a ‘sobel’ para esta aplicación y fue elegido para encontrar los bordes de la imagen.

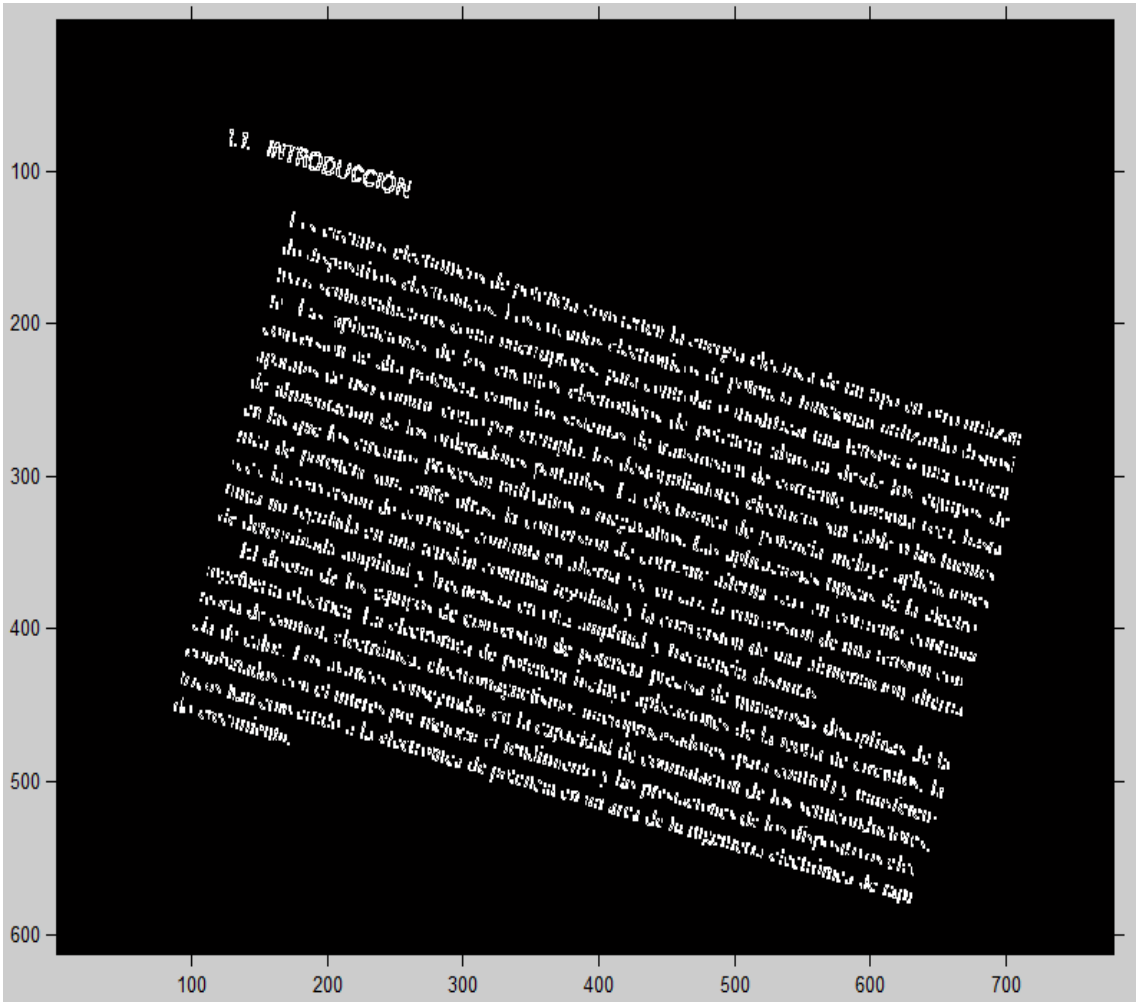
Figura 38 Detector de bordes Operador “Roberts”.



Fuente: Autores.

El proceso de detección de bordes es sencillo pero necesario para corregir la inclinación del texto, sin esta corrección no es posible continuar porque el resultado será erróneo. Se realizaron las pruebas con estos operadores para encontrar al más acertado. Existen otras operaciones que se pueden utilizar para ayudar a la transformada a detectar la línea de inclinación del texto, la transformada de Fourier indica esta inclinación.

Figura 39 Detector de bordes Operador "Roberts".



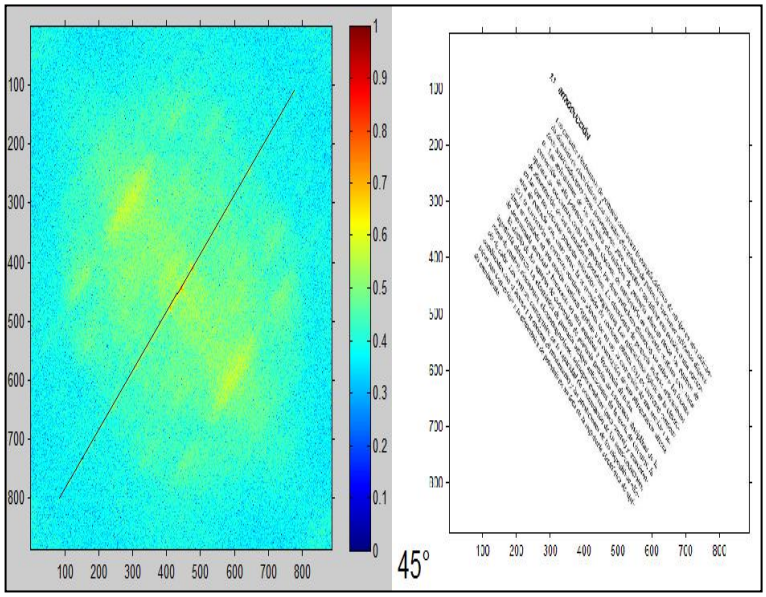
Fuente: Autores.

- Transformada de Fourier.

En el estado del arte se definió la función y utilidad de esta transformada, a continuación se indicaran los resultados obtenidos para dicho proceso. La transformada de Fourier encuentra la inclinación del texto realizando el trabajo del operador de borde, en busca de la mejor respuesta para detectar el ángulo se presentan algunas pruebas realizadas con esta herramienta.

En la Figura 40, se observa la imagen obtenida trabajando con la transformada de Fourier, crea una línea central que tiene la misma rotación del documento. En la primera prueba el resultado fue acertado. El ángulo de inclinación del documento es de 45°

Figura 40 Pruebas para corregir inclinación utilizando transformada de Fourier para detectar corrimiento.

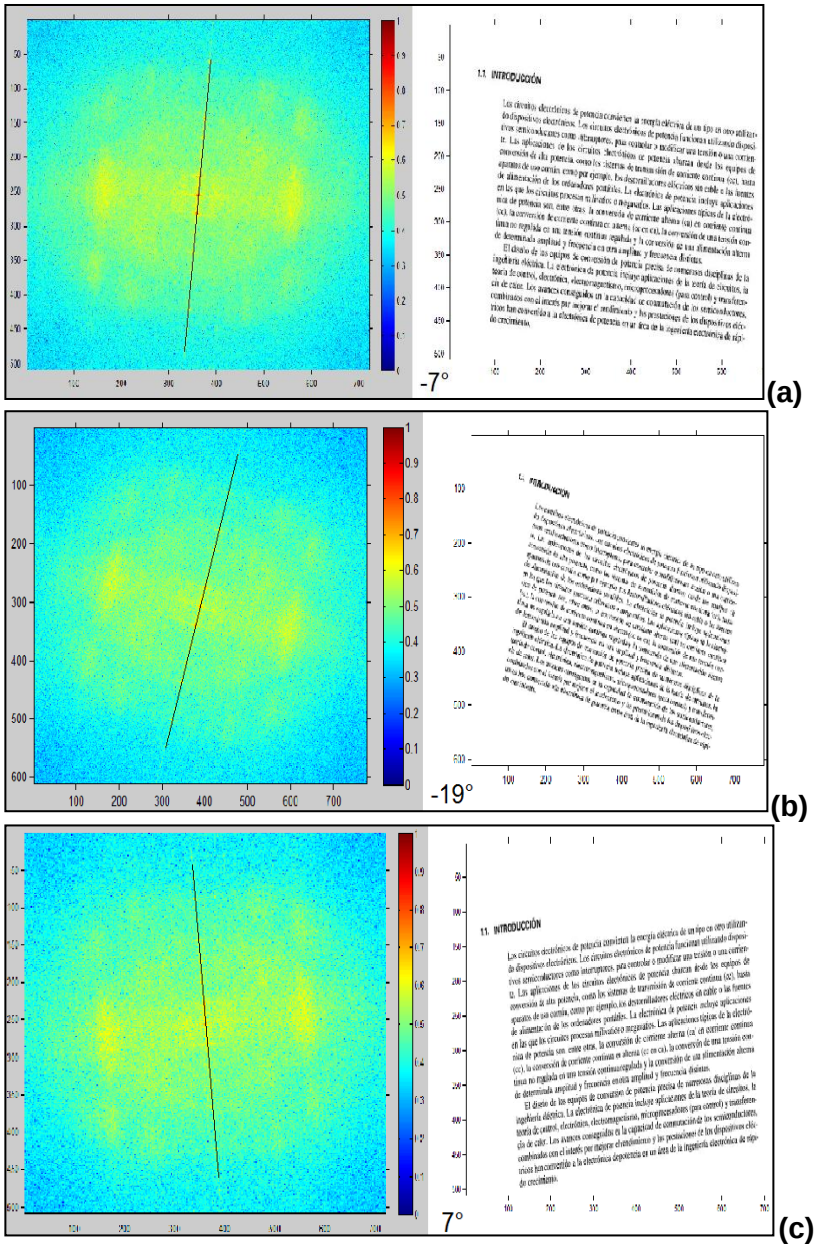


Fuente: Autores.

Los resultados graficados en la Figura 41 reconocen la inclinación del texto pero con una variación entre 2° y 4°, como se ha indicado este resultado no es adecuado para el proceso ya que la resolución o error permitido es entre más o menos 0.05°.

Una rotación de 2° es aun un resultado erróneo ya que la operación de segmentación no detectaría en forma correcta las líneas presentes en el texto, sin embargo si el resultado de la transformada de Fourier fuera constante este error seria corregido en el algoritmo, pero este proceso es variable y para esta aplicación que requiere un resultado exacto.

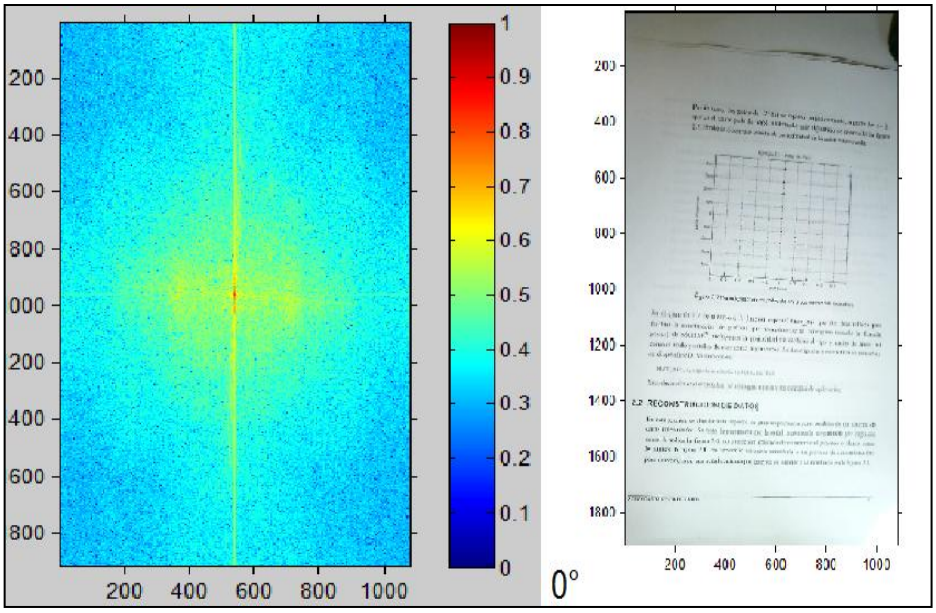
Figura 41 Pruebas para corregir inclinación utilizando transformada de Fourier para detectar corrimiento.



Fuente: Autores.

Una nueva prueba ratifica que esta herramienta no es constante. Cuando la inclinación es mínima, no es reconocida con esta herramienta. La grafica respuesta a este análisis se indica en la figura 42, que esta imagen tiene una inclinación real de 8.5°.

Figura 42 Prueba errada para corregir inclinación utilizando transformada de Fourier.



Fuente: Autores.

4.4 Transformada de Radón.

Radón como ya se indico es una importante transformada que describe líneas reconociendo el borde del documento (ver estado del arte), es otra manera además de Hough que permite determinar este ángulo de inclinación, para esto se hace uso de la herramienta computacional “radón” la cual calcula su transformada que es la proyección de la intensidad de la imagen a lo largo de una línea radial orientada a un ángulo especifico.

- [R,xp] = radon(imagen,tetha);

Donde:

R = proyección de la imagen analizada ‘tetha’ grados rotada.

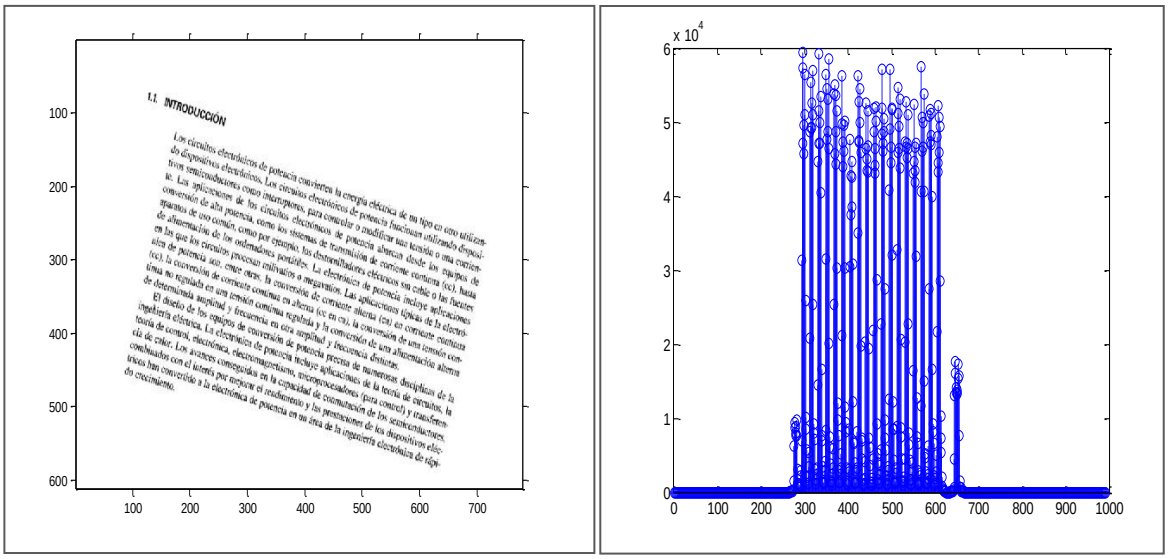
imagen = imagen a analizar.

tetha = ángulo de la imagen.

La proyección de la intensidad que realiza radón permite ver el número de líneas presentes en el texto pero es necesario conocer el ángulo previamente o realizar un proceso iterativo para conocer en el barrido de ángulos la mejor proyección. Radón presenta una variación de 0.4° con el resultado que presenta Hough.

Las graficas de respuesta a este proceso se indican para evidencia del proceso. En la Figura 43, muestra la transformada de radón para el texto, se indica el ángulo de inclinación del documento y se observa las líneas que conforman el texto.

Figura 43 Prueba para corrección de inclinación con transformada de Radón para detectar corrimiento.

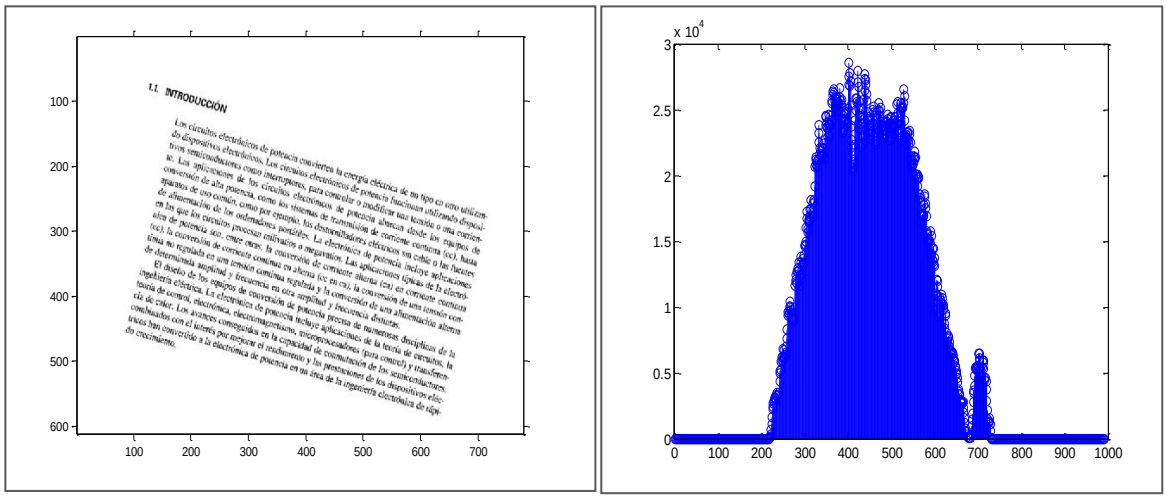


Fuente: Autores.

Cuando el proceso se realiza para un ángulo diferente al de la inclinación del documento el grafico del vector resultante que se grafica en la figura 44, no indica ningún valor real o interesante para el proceso, en el proceso iterativo este tipo de resultados no se tienen en cuenta.

En la figura 45, se indica otra prueba realizada con la herramienta “radón” para el ángulo indicado se muestran las líneas que conforman el texto.

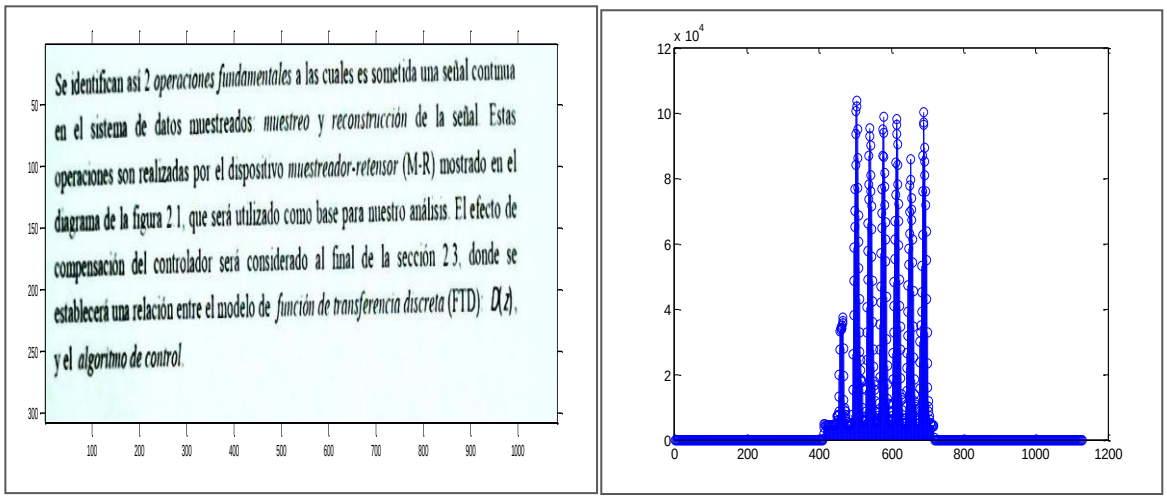
Figura 44 Prueba para corrección de inclinación con transformada de Radón para detectar corrimiento.



Fuente: Autores.

El análisis a estos resultados muestra una variación de 0.2° más que Hough en la respuesta del proceso. Por conocimiento y mejor respuesta se elige a Hough para detectar líneas para esta aplicación de proceso.

Figura 45 Prueba para corrección de inclinación con transformada de Radón para detectar corrimiento.

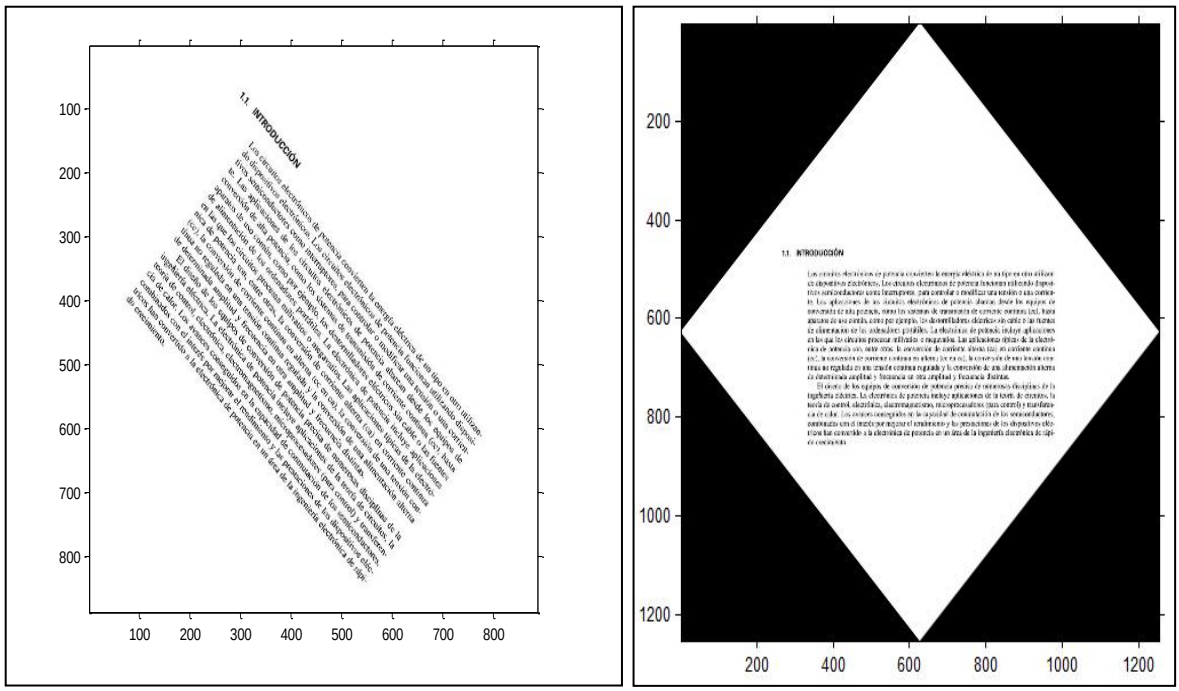


Fuente: Autores.

4.5 Pruebas hough – canny.

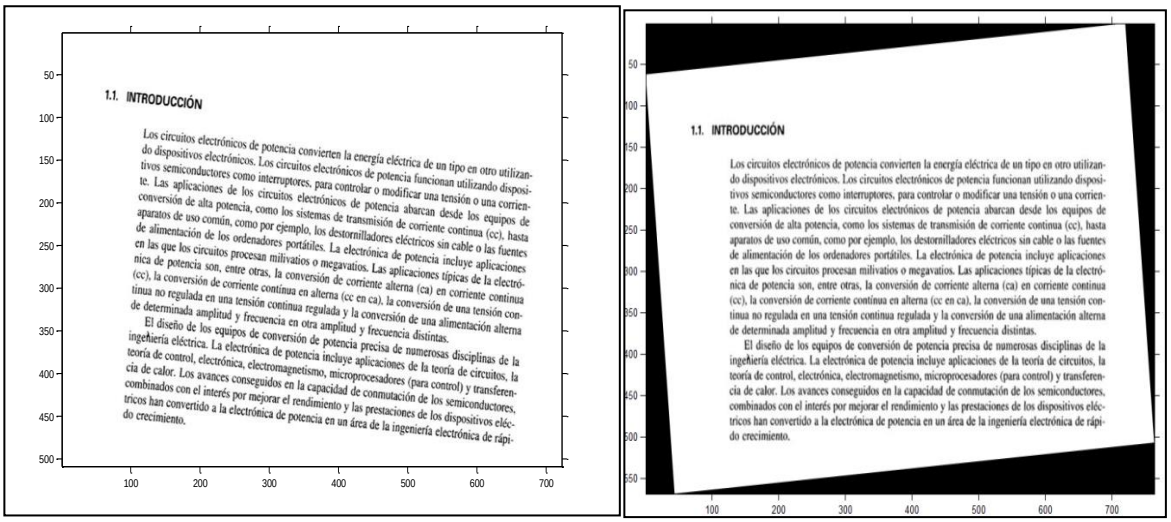
En las pruebas con los operadores de borde y la transformada de Fourier se indico el porqué 'Canny' fue la mejor opción en esta aplicación, para el proceso descrito se utilizo la transformada de Hough que detectaba la línea descrita por los operadores de borde y indicaba el ángulo de inclinación. En estas pruebas se obtuvieron buenos resultados y en el ítem anterior se observo el proceso con la transformada de radón, al analizar las figuras respuesta de las pruebas realizadas se concluye que Hough es la herramienta adecuada para corregir la inclinación del documento. Se presentan ahora las pruebas realizadas con estas dos herramientas que muestra la respuesta final es decir la corrección de la imagen analizada.

Figura 46 Pruebas con herramientas Hough y "Canny", para corrección adecuada de inclinación.



Fuente: Autores.

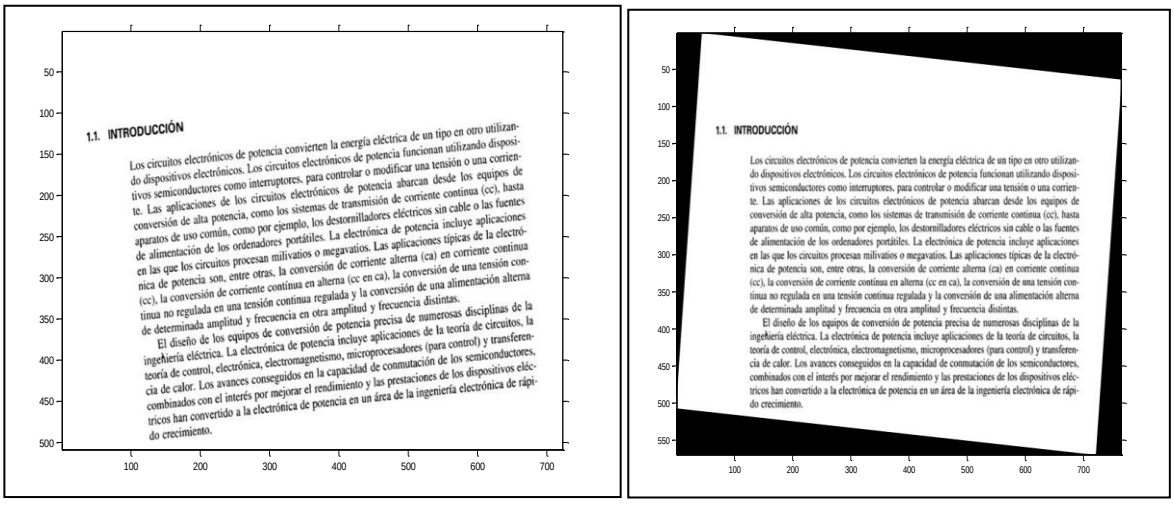
Figura 47 Pruebas con herramientas Hough y “Canny”, para corrección adecuada de inclinación.



Fuente: Autores.

En la Figura 46 y la Figura 47, se muestran los resultados a dos pruebas realizadas utilizando el operador ‘canny’ y la transformada de Hough para detectar el ángulo de inclinación, en las dos pruebas el resultado fue exacta. El color negro que se observa en las imágenes es la rotación que se realiza para corregir la imagen.

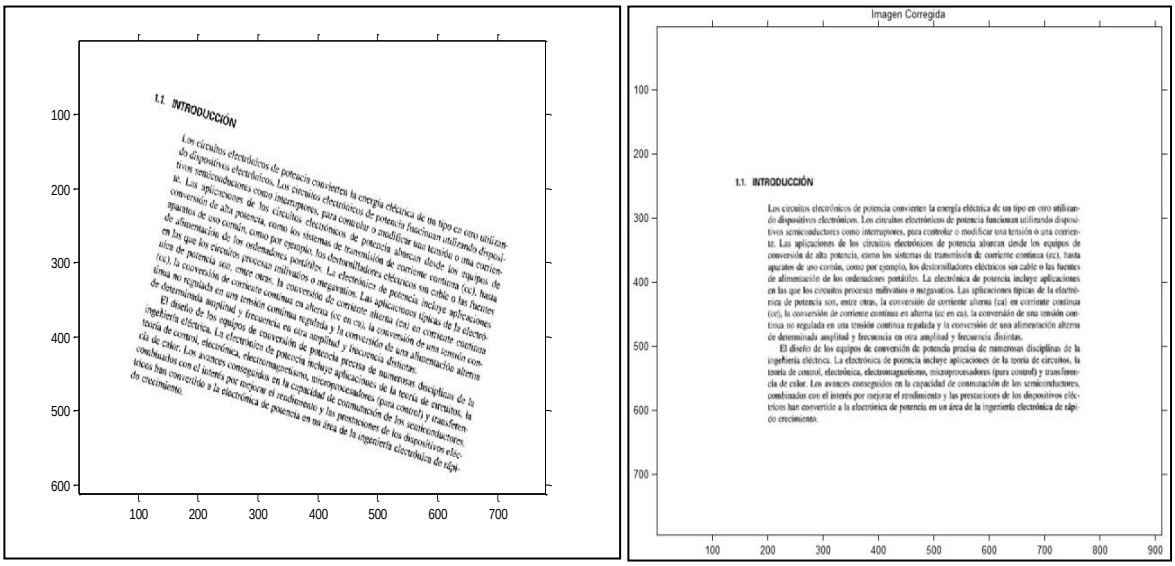
Figura 48 Pruebas con herramienta Hough y “Canny”, para corrección adecuada de inclinación.



Fuente: Autores.

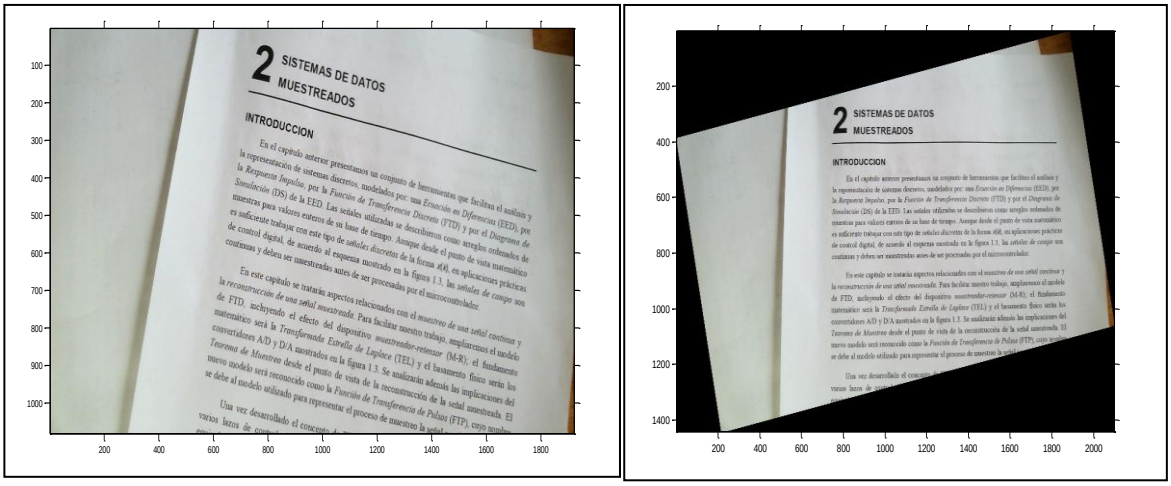
La imagen debe tener un fondo claro que en el proceso de binarizacion se puede remplazar adecuadamente. En la Figura 49, se indica la forma adecuada de salida para el proceso de corrección de inclinación.

Figura 49 Pruebas con herramienta Hough y “Canny”, con fondo correcto.



Fuente: Autores.

Figura 50 Una Pruebas más con herramienta Hough y “Canny”, con corrección adecuada de Inclinación.



Fuente: Autores.

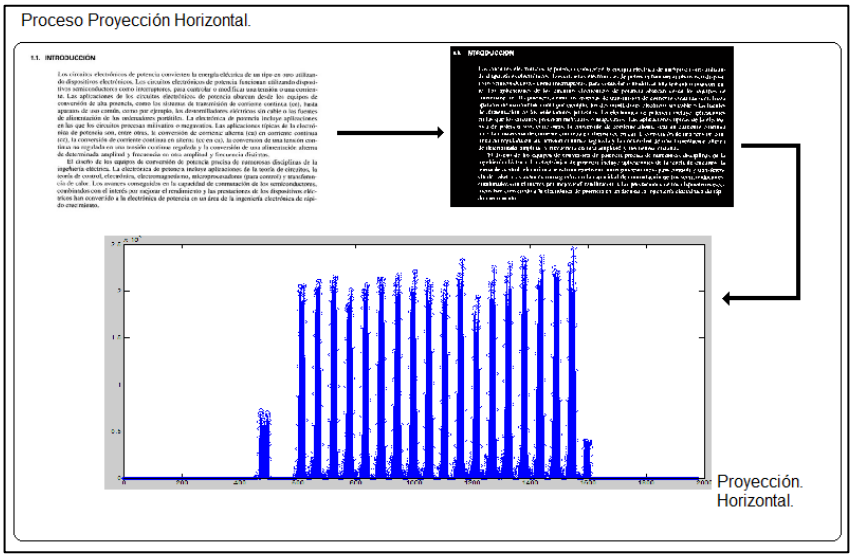
En todas las pruebas realizadas se obtuvo una corrección adecuada del documento, para la corrección de inclinación se usa el método de proyecciones y con ayuda de las herramientas computacionales se hizo posible utilizar este método para la corrección. Las transformaciones especiales se utilizan para realizar iteraciones en un margen del ángulo detectado y de más o menos la resolución utilizada para encontrar la mejor imagen de salida para este proceso.

4.6 Resultados pruebas etapas de segmentación

➢ 1er nivel de segmentación.

Corresponde a la descomposición del texto en líneas. En los diagramas de flujo de la Figura 17 y la Figura 18, se indica el proceso que se debe realizar para obtener esta descomposición. Una vez se tiene la imagen binarizada se comienza el barrido sumando los pixeles en color blanco que se encuentran a lo largo de las filas y columnas de la imagen para tener las proyecciones vertical y horizontal para su respectivo análisis.

Figura 51 Utilizando proyecciones para segmentar la imagen, Proyección horizontal.



Fuente: Autores.

El Diagrama de la Figura 51, presenta un grafico con los bloques más relevantes de la primera etapa de segmentación. Se presentan 3 imágenes que son la imagen capturada, la imagen binarizada y la grafica de la proyección horizontal. Se aplica el proceso de la Figura 16, a la imagen capturada y así obtener la imagen en dos niveles de color.

El resultado del proceso de proyección son dos vectores que indican la intensidad de color en cada una de las filas y columnas de la imagen. Al analizar estos vectores se procede a realizar una serie de iteraciones en busca de valores “0”, que indican los limites de corte para el proceso de separación de líneas.

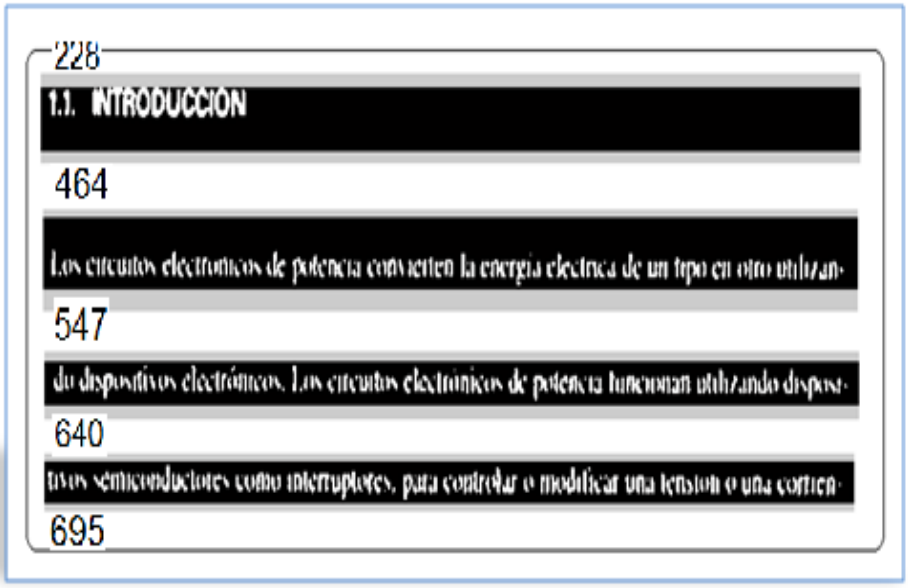
Tabla 1 Vector de límites de corte en la proyección horizontal.

vlim <1x22 double>

	1	2	3	4	5	6
1	228	464	547	640	695	748
2						
3						

Fuente: Autores.

Figura 52 Segmentación de imagen para vector indicado en la tabla 1.

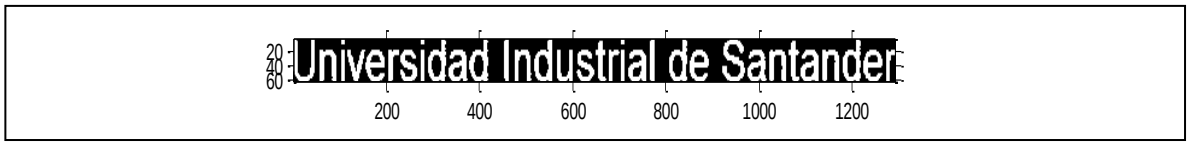


Fuente: Autores.

Este proceso iterativo tiene como respuesta un vector de límites, un ejemplo de este proceso se indica en la tabla 1. Los valores de vector indican el corte que se debe realizar a la imagen para tener cada una de las imágenes de las líneas de texto que son la información de entrada para las siguientes etapas de segmentación. La interpretación del proceso de corte se visualiza en la Figura 52.

Un ejemplo más detallado de lo que ocurre en la primera etapa de segmentación se presentada en la Figuras 53, 54 7 55 donde se muestran las nuevas imágenes extraídas de la imagen de texto inicial que fue ya graficado en la Figura 28(c).

Figura 53 Imagen resultado de primera etapa de segmentación. Titulo del texto.



Fuente: Autores

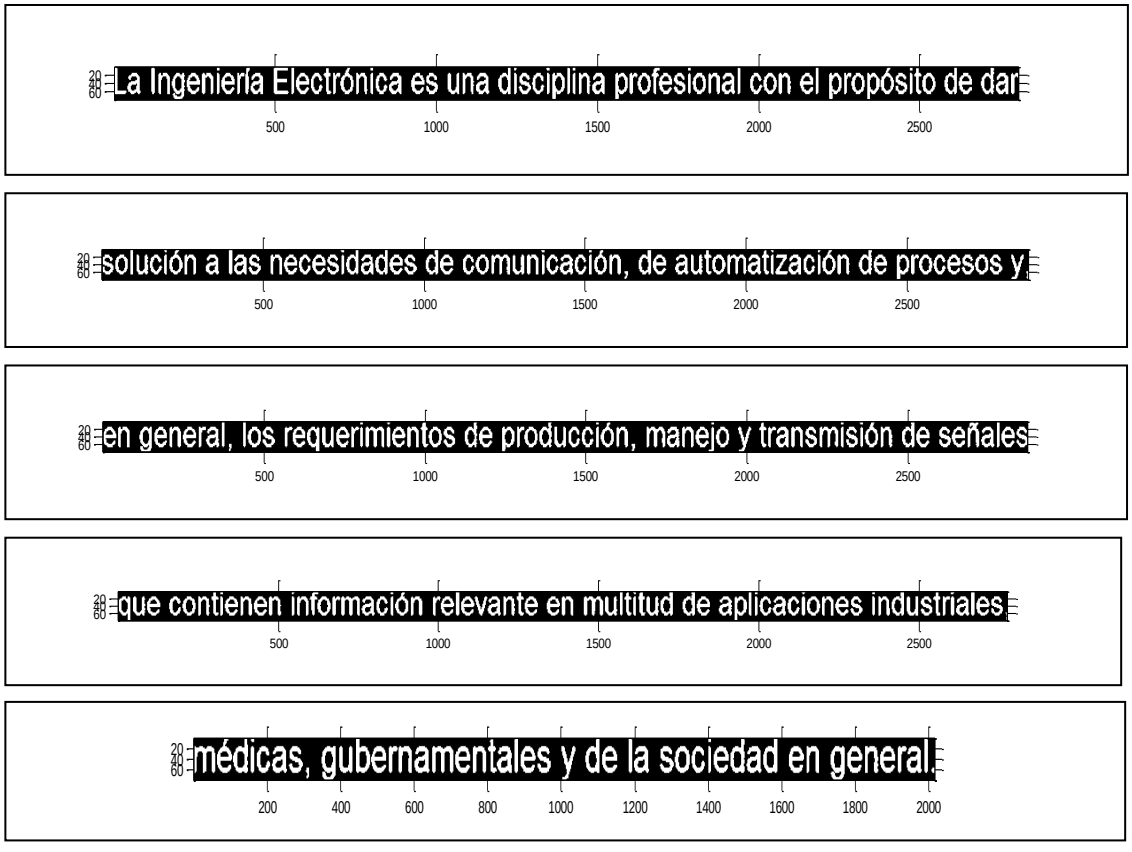
En la Figura 53, se muestra el titulo del documento que se esta analizando, esta imagen ya se ha sometido a una etapa de recorte de fondo, esto se realiza con el fin de eliminar partes de imagen sin información

Figura 54 Imagen resultado de primera etapa de segmentación. Subtitulo del texto.

Fuente: Autores

En la Figura 54, se observa el subtitulo del documento analizado. Y en la Figura 55 el párrafo de texto que está separado por líneas. El ejemplo muestra visualmente al texto separado en líneas. El proceso realizado para cualquier imagen de texto es el mismo.

Figura 55 Imágenes de las líneas de párrafo de texto extraídas de la imagen de la Figura 28c.

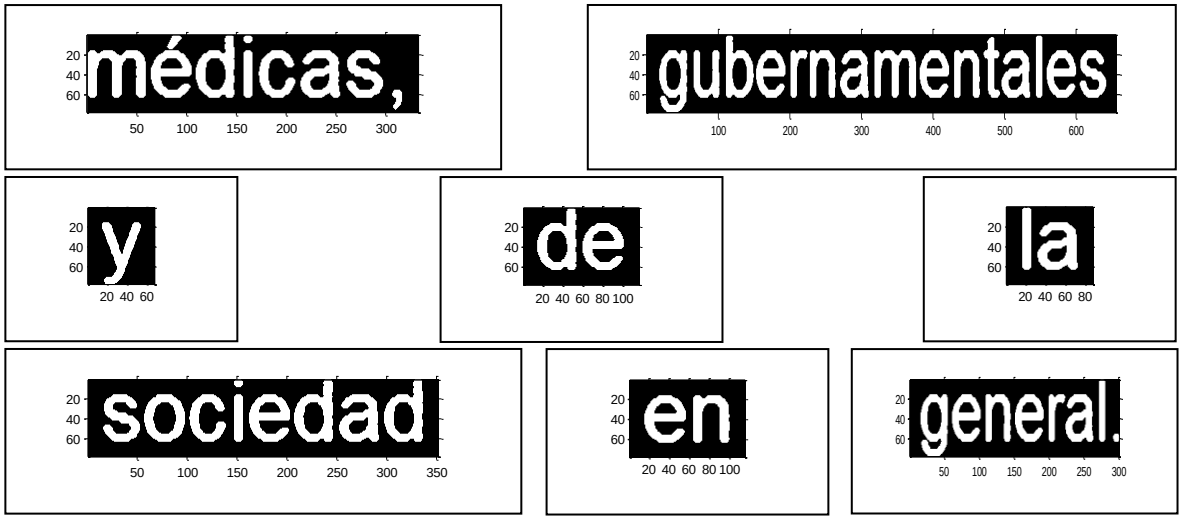


Fuente: Autores

➢ *2° nivel de segmentación*

El segundo nivel de segmentación, realiza iteraciones como se indica en el diagrama de flujo de la Figura 20 y busca separar cada una de las líneas de texto que fueron extraídas en la primera etapa de segmentación en palabras. Se presenta a continuación los resultados obtenidos para la imagen de la Figura 55e.

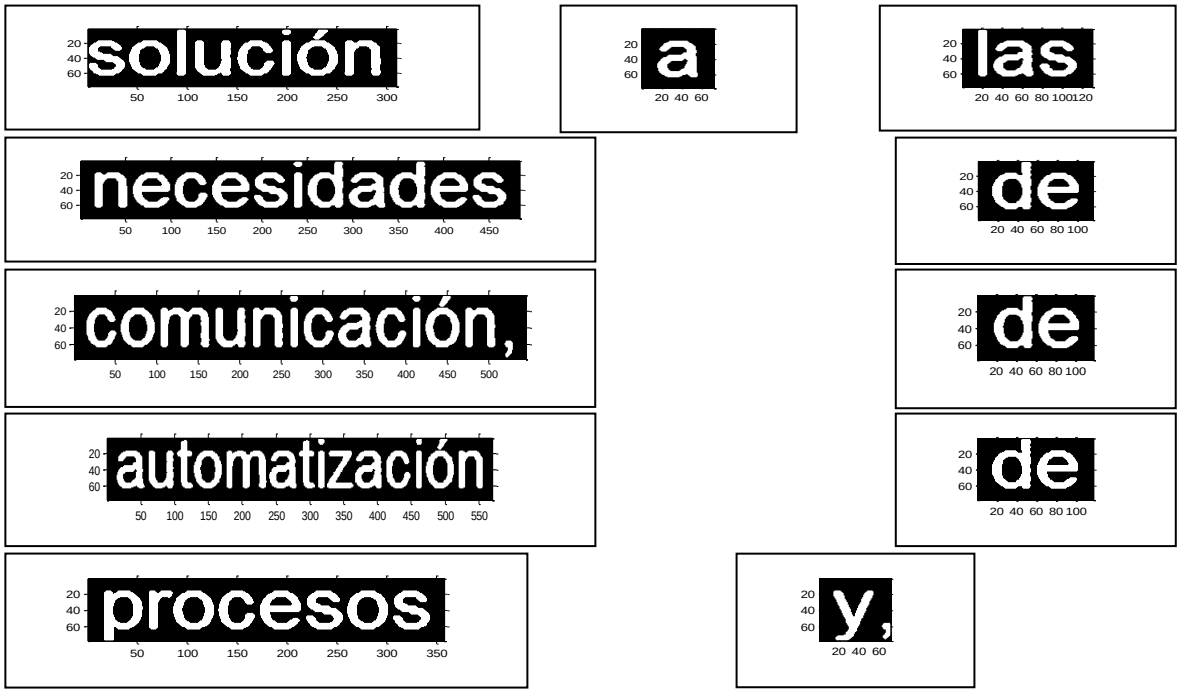
Figura 56 Imágenes de las palabras resultantes de la segunda etapa de segmentación para imagen de la Figura 55e.



Fuente: Autores

El algoritmo realiza la proyección vertical a cada una de las imágenes de línea de texto en busca ceros para obtener el límite de corte.

Figura 57 Imágenes de las palabras resultantes de la segunda etapa de segmentación para imagen de la Figura 55b.

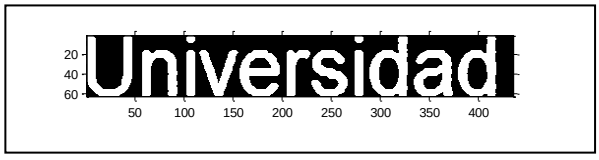


Fuente: Autores

➢ 3er nivel de segmentación.

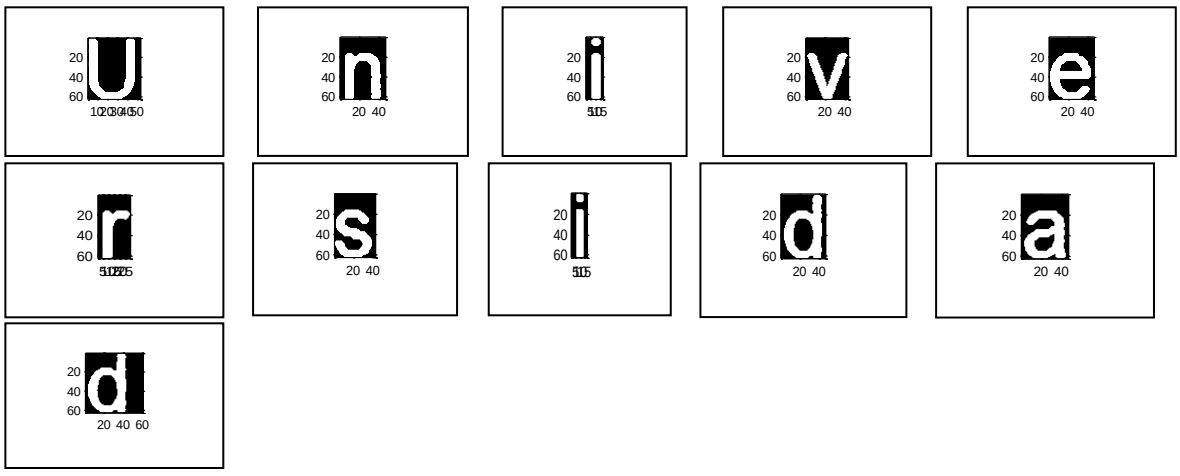
La etapa final de este proceso consiste en analizar las palabras que se generan en el segundo nivel de segmentación, realizar una proyección vertical y analizar los cortes en cero. Finalmente tenemos las imágenes de cada uno de los caracteres que forman la palabra, que son las imágenes de entrada para el siguiente proceso de reconocimiento y clasificación. Cada etapa tiene un propósito que debe cumplir, al no realizar adecuadamente su función interrumpe el reconocimiento y traducción del mismo. A continuación en la Figura 58 se indica el resultado que se obtiene al analizar la palabra "Universidad".

Figura 58 Palabra analizada para tercera etapa de segmentación.



Fuente: Autores

Figura 59 Imágenes resultado de la tercera etapa de segmentación para imagen de la Figura 58.

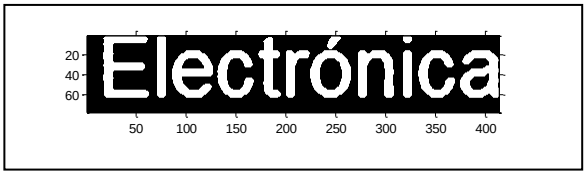


Fuente: Autores

El mismo proceso se realiza para cada una de las palabras que conforman el texto. Se ve sencillo y fácil, pero requiere de conocimiento y análisis en cada una

de las etapas, a pesar que en la mayoría de los cortes el proceso fue limpio, se presentaron casos especiales donde las letras por su forma unían los caracteres que estaban a su alrededor.

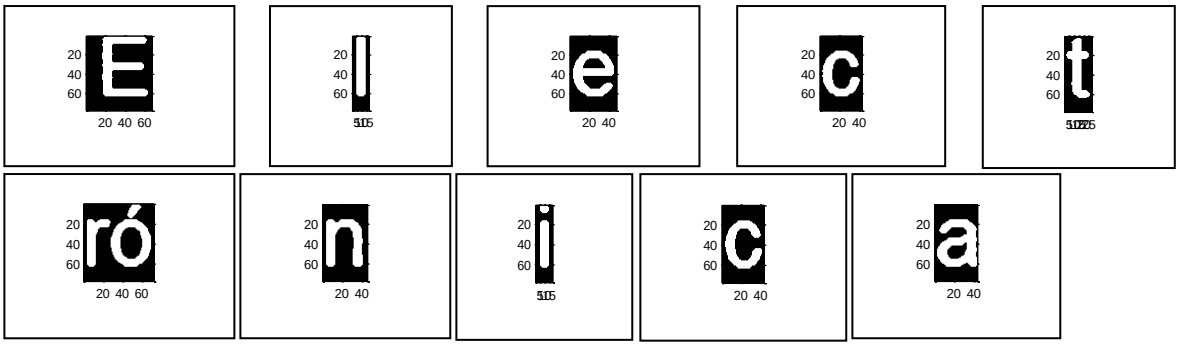
Figura 60 Palabra analizada para segundo ejemplo en la tercera etapa de segmentación.



Fuente: Autores

La Figura 60, grafica la palabra “Electrónica” la forma de composición de las letras “r” y “ó” no crea ningún espacio entre ellas siendo reconocidas como un solo símbolo o caracter, proceso que se indica en la Figura 61.

Figura 61 Imágenes resultado de la tercera etapa de segmentación para imagen de la Figura 60.



Fuente: Autores

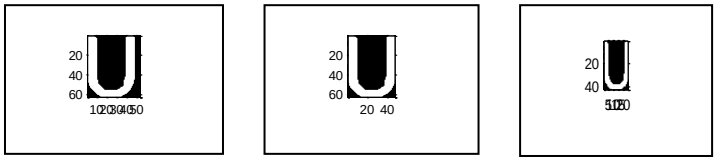
Durante el desarrollo y análisis se encontraron caracteres unidos relacionados con las letras “r” y “j”, en el código correspondiente a la etapa de clasificación se tienen en cuenta estos casos especiales.

4.7 CLASIFICACIÓN.

Para la clasificación del dato en la imagen es necesario realizar un recorte que elimina el exceso de fondo en la imagen y posteriormente normalizar la imagen a

un tamaño de 42 x 24 pixeles. Este proceso se muestra en la Figura 62.

Figura 62 Ejemplo de normalización de imagen para caracter 'u'.



Fuente: Autores

Con el caracter normalizado se realizan las proyecciones a esta imagen y obtenemos un vector de datos para la proyección horizontal y uno para la proyección vertical, se realiza la transformada de Fourier a estos datos y se guarda esta información en 30 valores para cada vector.

Figura 63 Vector de 30 valores de la transformada de Fourier para la proyección horizontal del caracter 'u'.

param1 = [14.4706 0.2442 0.1640 0.2994 0.2808 0.3113 0.4714 0.2697

0.2849 0.3582 0.2038 0.0729 0.2103 0.0875 0.1173 0.1176

0.1173 0.0875 0.2103 0.0729 0.2038 0.3582 0.2849 0.2697

0.4714 0.3113 0.2808 0.2994 0.1640 0.2442];

Fuente: Autores

Figura 64 Vector de 30 valores de la transformada de Fourier para la proyección vertical del caracter 'u'.

param2 = [10.2105 1.7016 4.9629 4.8348 0.2276 2.2792 1.1295 0.1346

0.6061 0.6624 0.7240 1.4065 0.4449 0.7013 0.4903 0.0526

0.4903 0.7013 0.4449 1.4065 0.7240 0.6624 0.6061 0.1346

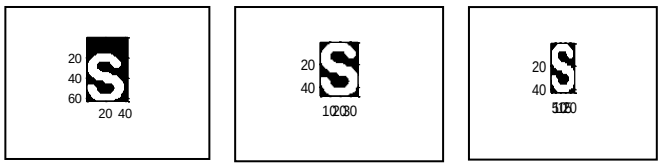
1.1295 2.2792 0.2276 4.8348 4.9629 1.7016];

Fuente: Autores

Se indican dos ejemplos que muestran los resultados para la primera etapa de clasificacion, el caracter "U" y el caracter "s" procesos graficados en la figuras 45 y 46 respectivamente.

El proceso que se visualiza en este documento es el mismo que se realiza para cada uno de los caracteres separados en el proceso de segmentacion, los casos especiales son previamente identificados para anexar su representacion a la base de datos y obtener un buen resultado. El proceso con los casos especiales es el mismo.

Figura 65 Ejemplo de normalización de imagen para caracter 's'.



Fuente: Autores

Figura 66 Vector de 30 valores de la transformada de Fourier para la proyección horizontal del caracter 's'.

param1 = [18.4762 0.3151 3.0015 0.4446 0.6346 0.7559 0.3885 0.3439
0.5855 0.2011 0.0952 0.4574 0.3245 0.3525 0.3339 0
0.3339 0.3525 0.3245 0.4574 0.0952 0.2011 0.5855 0.3439
0.3885 0.7559 0.6346 0.4446 3.0015 0.3151];

Fuente: Autores

Figura 67 Vector de 30 valores de la transformada de Fourier para la proyección vertical del caracter 's'.

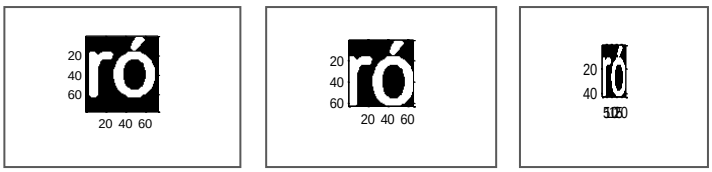
param2 = [17.1212 5.3392 4.5444 0.5113 0.8566 0.3674 0.6708 0.3706
0.2682 0.1524 0.0802 0.3699 0.5421 0.0146 0.3402 0.4545
0.3402 0.0146 0.5421 0.3699 0.0802 0.1524 0.2682 0.3706
0.6708 0.3674 0.8566 0.5113 4.5444 5.3392];

Fuente: Autores

Se debe obtener los vectores de característica que represente este nuevo símbolo que será formado por los dos caracteres, se debe analizar todo el texto y tener en la base de datos todas a las posibles uniones que se presenten para logran una

buena traducción. Estos análisis cambian dependiendo el tipo de letra y el tamaño de la misma, así como la calidad de la imagen capturada.

Figura 68 Ejemplo de normalización de imagen para carácter de caso especial 'ró'.



Fuente: Autores

Figura 69 Vector de 30 valores de la transformada de Fourier para la proyección horizontal del caracter 'ró'.

param1 = [14.2500 4.1402 2.8317 1.3897 1.3884 0.1803 0.1809 0.8644
0.5572 0.1903 0.2598 0.3047 0.0691 0.3687 0.3178 0.0500
0.3178 0.3687 0.0691 0.3047 0.2598 0.1903 0.5572 0.8644
0.1809 0.1803 1.3884 1.3897 2.8317 4.1402];

Fuente: Autores

Figura 70 Vector de 30 valores de la transformada de Fourier para la proyección vertical del caracter 'ró'.

param2 = [13.2500 1.4844 3.3601 4.8668 1.3931 0.4375 2.2933 0.3749
0.5972 0.5404 0.5962 0.6862 0.8957 1.0326 0.2030 0.0625
0.2030 1.0326 0.8957 0.6862 0.5962 0.5404 0.5972 0.3749
2.2933 0.4375 1.3931 4.8668 3.3601 1.4844];

Fuente: Autores

5. ESPECIFICACIONES DE FUNCIONAMIENTO DEL ALGORITMO

5.1 Algoritmo.

A continuación se indicaran las especificaciones necesarias para el funcionamiento del algoritmo, para ello se debe cumplir con las entradas y salidas especificadas en cada etapa del proceso.

- Inclinación.

```
Function Imagen_inicial = C_Inclinacion(Imagen_nueva)
```

Entrada

- Imagen_inicial= imagen capturada por webcam.

Salida

- Imagen_nueva = imagen corregida, es decir sin inclinación.

A la imagen capturada por webcam se debe realizar el proceso descrito en la figura 16, una vez se tiene el ángulo de inclinación se debe compensar el ángulo faltante para tener 90°. Las pruebas realizadas fueron desarrolladas con el armazón diseñado en el anexo 5, como dato importante se indica el uso de una bombilla de luz blanca en el interior del armazón.

El umbral escogido en base a prueba y error aplicado a la base de datos de imágenes incluidas en el cd adjunto a este documento, arrojo como resultado un valor de umbral menor o igual a 40 para el proceso de binarizacion realizado en la etapa de corrección de inclinación.

Para validar esta etapa de proceso se realizaron pruebas a imágenes ideales (imágenes limpias) que se indican en la tabla, que se presenta a continuación con los resultados promedio de 10 pruebas, con ella se valida el funcionamiento del

proceso de corrección de inclinación, la resolución del proceso es de 0.05° el cual es un valor aceptable para el proceso. El umbral para el análisis con las imágenes ideales fue de un valor igual a 170.

Tabla 2 Resultados promedio de pruebas realizadas para encontrar umbral adecuado de binarización.

Formato	Imagen analizada	Angulo de inclinación real,	Angulo de inclinación detectado,	Error
	Imágenes ideales			
JPG	1234	89	89,3	0%
	Dibujo1	85	85,45	1%
	Dibujo2	85	84,9	0%
	Dibujo3	75	73,15	3%
	Dibujo4	75	75	0%
	Dibujo1234	89	89,2	0%
	luna1	2	2,2	9%
	luna2	0,1	0,05	0%
	luna3	0,1	0,15	0%
	luna5	0,001	0,05	0%
	luna7	2	2,1	5%
	luna8	1,5	1,45	3%
	luna9	78	78,5	1%
	luna10	4	4,1	2%
	luna11	85	85,2	0%
	luna12	85	84,6	0%
	luna15	2	2	0%
	luna16	0,1	0,1	0%
	luna17	0,1	0,05	0%
	luna19	0,1	0,05	0%
	luna90	85	85,5	1%
	pruebas21	90	90	0%
	pruebas22	90	90	0%
	pruebas23	90	90	0%
PNG	prueba	90	88,5	2%

Fuente: Autores

Para los umbrales escogidos a prueba y error, fueron satisfactorios en cada una de las pruebas realizadas.

- Binarización.

```
Function  y = binarización(x,n)
```

Entrada

- x = imagen a color ó en nivel de grises.
- n = umbral.

Salida

- y = imagen en su mínima expresión sin perder datos (imagen binarizada).

Se realiza un barrido pixel a pixel pasando por todas las líneas y columnas que conforman el texto. Se compara el valor del pixel con un valor umbral n, y se clasifica en dos si es mayor que el umbral va a 255 si es menor va a cero. La imagen de salida presenta dos niveles de gris, blanco y negro.

Esta etapa de binarizacion es más estricta que la realiza para corregir la inclinación, nuevamente analizando las imágenes a prueba y error se obtuvo el umbral escogido para el proceso, al cual se le realizan algunas operaciones morfológicas para obtener la imagen más limpia para continuar con la etapa de segmentación.

Para imágenes ideales el umbral elegido fue de 149, no hay necesidad de aplicar mas operaciones porque la imagen respuesta es apta para el proceso de segmentación.

Para imágenes reales es decir capturadas a través de la webcam y con el armazón, el umbral elegido para el proceso se define como el mínimo nivel de gris presente en la imagen más un valor fijo de 44. Analizando el proceso se hizo

necesario aplicar un fuerte proceso de erosión para limpiar los pixeles de ruido presentes en la imagen, y un proceso de dilatación suave, para darle forma a los caracteres.

- Proyecciones.

```
Function  [pH,pV] = proyecciones(x)
```

Entrada

- x = Negativo de una imagen Binarizada.

Salida

- pH = Proyección Horizontal.
- pV = Proyección Vertical.

Las proyecciones son parte esencial de nuestro proceso. A la imagen de entrada se realiza un barrido pixel a pixel en cada fila o columna según el caso y se suma los pixeles en blanco, al final tendremos un valor de blanco en cada fila de la imagen que da a la proyección horizontal, y un valor de blanco de cada columna de la imagen para la proyección vertical.

- Separar texto en palabras.

```
function vlim5 = palabras1(vlim2,pV2)
```

Entrada

- vlim2 = Vector que tiene los primeros limites de corte para la proyección vertical.
- pV2 = proyección vertical de la línea analizada

Salida

- vlim5 = Vector con valores donde corta las palabras que conforman la línea analizada.

Se debe realizar una serie de iteraciones para lograr tener el texto por palabras, según el tamaño de la imagen se debe analizar la cantidad de pixeles entre palabras para lograr una separación apropiada.

- Separar palabras en caracteres.

```
function [vlim222]= letras1(letra,retesl)
```

Entrada

- letra = Cada una de las palabra encontradas anteriormente.
- Retesl = Distancia en pixeles de la palabra analizada.

Salida

- Vlim222 = Vector con cada uno de los valores limites para cada caracter.

Cada una de las palabras reconocidas en el anterior proceso y la distancia en pixeles que la forman son los parámetros de entrada para este algoritmo, se realiza una proyección a esta palabra, para encontrar nuevos cortes por cero, y obtener así el vector de salida con cada uno de los caracteres. Cuando solo hay una letra se usa la comparación con la distancia para su adecuado reconocimiento.

- Patrones del caracter.

```
function [param1 param2]= preclasif(vlim222,letra)
```

Entrada

- Letra = imagen de la palabra.
- Vlim222 = límite superior e inferior de caracter a analizar.

Salida

- param1 = Vector salida 30 componentes de Fourier de la proyección Horizontal.
- param2 = Vector salida 30 componentes de Fourier de la proyección Vertical.

Esta función analiza el vector de letra analizada, recorta el espacio sobrante alrededor del caracter, realiza una proyección de esta nueva imagen y realiza la transformada de Fourier a cada componente de salida de la proyección, esta transformada nos entrega un vector de 30 componentes que describen la forma de la imagen analizada.

- Clasificación.

```
function y = funClasif12(x,y)
```

Entrada

- y = Vector con 30 componentes de Fourier de la proyección vertical aplicada a la imagen del caracter.
- x = Vector con 30 componentes de Fourier de la proyección horizontal aplicada a la imagen del caracter.

Salida

- y = Caracter en formato digital.

Compara los datos de entrada, es decir los componentes de la proyección y los compara por k – NN vecinos por medio de la distancia euclídea, al que resulte menor distancia le asigna el dato o caracter correspondiente de salida, además cuando tiene detectado el dato, envía un caracter ASCII, mediante formato USB, que es leído por el modulo MJ60, y visualizado en Leds, para indicar que se está realizando la traducción.

5.2 Salida.

La salida del algoritmo transmite información que representa la traducción al puerto USB. Para interpretar esta información se propone una línea Braille como salida del sistema. Una línea Braille comprende un número determinado de celdas braille, controles de mando, puerto USB y fuente de alimentación en determinados casos. Los controles de mando teniendo en cuenta las líneas braille en el mercado pueden ser: “línea siguiente” ó “línea anterior”, esto para que el usuario interactué con el sistema.

6. DESCRIPCION HARDWARE

En este capítulo se documenta y explica, las etapas en las que se divide el proceso de hardware, el cual es el encargado de:

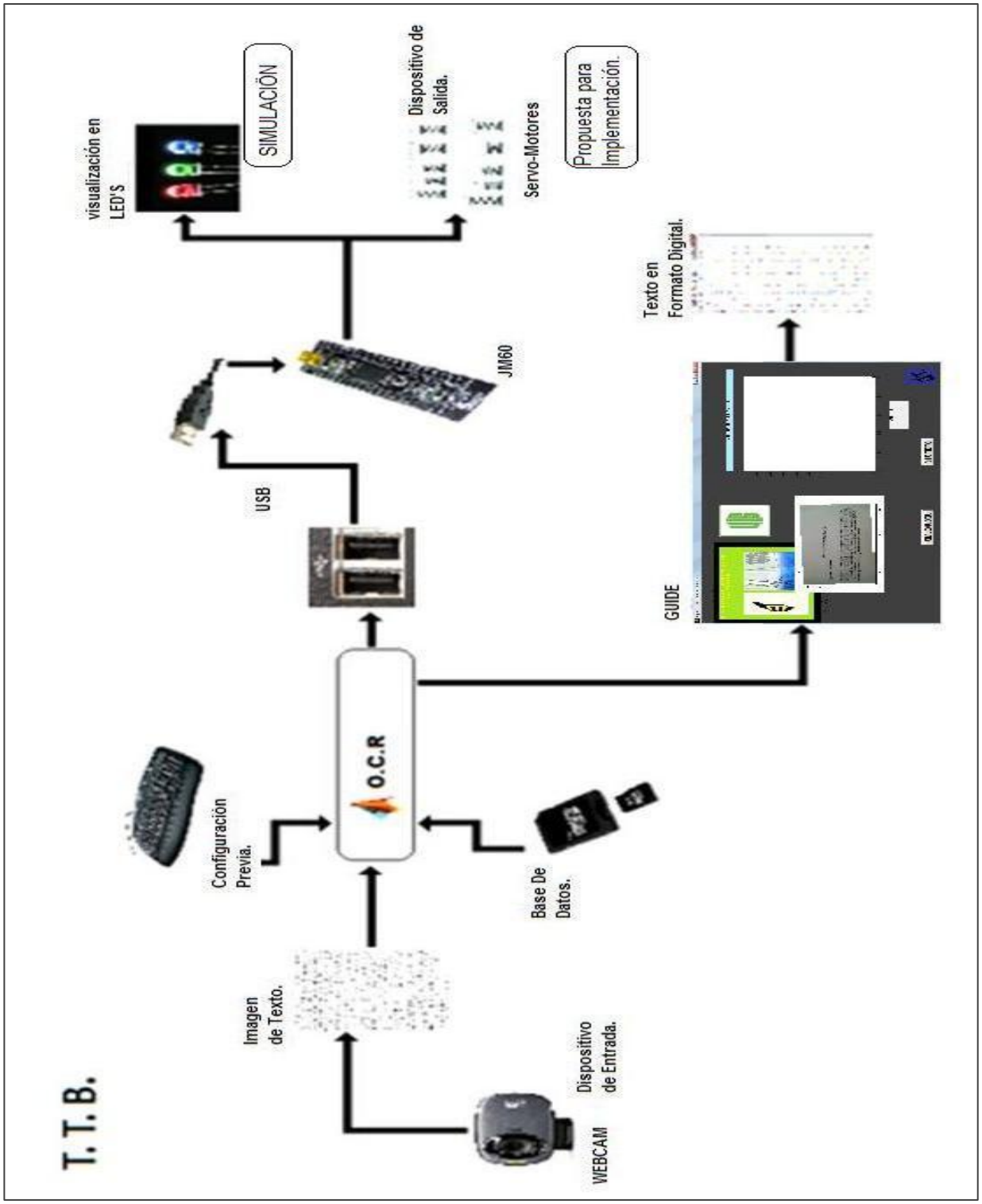
- Capturar automáticamente la imagen de texto seleccionada por el usuario.
- Procesar el software encargado de traducir un texto y presentar su visualización en una interfaz grafica.
- Visualizar el resultado de una simulación del software en Led's que indican la traducción del texto.
- Presentar una propuesta de hardware para futura implementación.

6.1 Funcionalidad

Se busca que el procedimiento desarrollado sea capaz de analizar texto impreso en papel. Sin embargo para este proceso se debe analizar si el texto es manuscrito o texto impreso. Si es manuscrita la recomendación es una consistencia en la forma de las letras y el tamaño de ellas, si el texto es impreso es importante el tamaño y tipo de letra. Este análisis se debe realizar para obtener una gran base de datos que permita un reconocimiento global de textos. El proceso de simulación hardware presentado en este documento es sencillo ya que el proyecto presentado es un diseño, la simulación se realiza para presentar una prueba visual del proceso, lo que implica que cuente con ciertas limitaciones. Se deja evidencia de todos los pasos y de las recomendaciones para una implementación de la propuesta.

A continuación en la Figura 71, se enseña una imagen de bloques que describe el funcionamiento del sistema diseñado.

Figura 71 Diagrama de bloques del sistema.



Fuente: Autores.

6.2 Captura de Imágenes.

➢ Captura de imágenes con Escáner

La mayoría de aplicaciones OCR utiliza un escáner para captura de imágenes. Para nuestra aplicación es importante que la captura sea automática y no fue posible una comunicación automática entre el lenguaje programador y el dispositivo.

➢ Dispositivos webcam

Las webcam o cámaras web son dispositivos para capturar además de transmitir imágenes y video en tiempo real conectadas mediante formato USB al computador, por lo general son utilizadas en chats, videoconferencias o videos cortos. Este dispositivo permite una captura automática que facilita el procesamiento de la imagen.

1. HP MediaSmart Webcam

Matlab referencia la cámara y permite capturar video en las dimensiones descritas en la Tabla 1. El tamaño de la imagen es importante porque se quiere una imagen bien definida donde se observe claramente las letras y símbolos que conforman la imagen. Cuando se realizan las operaciones morfológicas del código de software sobre una imagen pequeña esta operación une las letras y no permite una buena identificación, por eso se elige el mayor tamaño.

Figura 72 Dispositivo de entrada (WEBCAM interna del computador HP).

Fuente: http://www.computadoras-portatiles.com

En la imagen de la Figura 72 se indica gráficamente el dispositivo webcam utilizado. Para el análisis de todas las webcam analizadas en el anexo 4 se explica el proceso necesario para obtener las características técnicas de captura para cada webcam.

Tabla 3 Formatos posibles de captura de video para cámara web del dispositivo HP.

Tamaño	Megapixeles
'RGB24_160x120'	0.019 Mp
'RGB24_176x144'	0.025 Mp
'RGB24_320x240'	0.076 Mp
'RGB24_352x288'	0.1 Mp
'RGB24_640x480'	0.31 Mp

Fuente: Autores.

2. Webcam Logitech C120

El proceso de captura es el mismo que se explico con la anterior cámara web. Se indica gráficamente en la Figura 73 una imagen del dispositivo. La Logitech C120 permite una captura del mismo tamaño máximo de imagen que la explicada en el ítem 1. En diferencia con la anterior cámara analizada esta tiene como ventajas que regula la luz automáticamente, enfoca todo lo que se mueve dentro de su alcance, además de su buena nitidez que garantiza una buena captura.

Figura 73 Dispositivo de entrada (WEBCAM Logitech C120).

Fuente: http://www.logitech.com/es-roam/home

Tabla 4 Formatos posibles de captura de video para cámara web C120.

Tamaño	Megapixeles
'RGB24_160x120'	0.019 Mp
'RGB24_160x90'	0.014 Mp
'RGB24_176x144'	0.025 Mp
'RGB24_320x180'	0.057 Mp
'RGB24_320x200'	0.064 Mp
'RGB24_320x240'	0.077 Mp
'RGB24_352x288'	0.1 Mp
'RGB24_640x360'	0.23 Mp
'RGB24_640x400'	0.25 Mp
'RGB24_640x480'	0.31 Mp

Fuente: Autores

En la tabla 2 se indican los formatos de video que son posibles para capturar video con este dispositivo.

3. Webcam Logitech C615

Como se viene indicando el tamaño de la imagen es un factor importante en la escogencia de la cámara web y en el resultado del proceso aplicado. La Logitech C615 es una cámara web HD (Alta Definición) que tiene una buena resolución en las imágenes que toma, además realiza enfoques automáticos.

Figura 74 Dispositivo de entrada (WEBCAM Logitech C615).

Fuente: http://www.logitech.com/es-roam/home

Tabla 5 Formatos posibles de captura de video para cámara web C615

Tamaño	Megapixeles
'RGB24_1024x576'	0.59 Mp
'RGB24_1280x720'	0.92 Mp
'RGB24_1600x896'	1.43 Mp
'RGB24_160x120'	0.019 Mp
'RGB24_176x144'	0.025 Mp
'RGB24_1920x1080'	2.07 Mp
'RGB24_320x240'	0.076 Mp
'RGB24_352x288'	0.101 Mp
'RGB24_432x240'	0.136 Mp
'RGB24_640x360'	0.23 Mp
'RGB24_640x480'	0.307 Mp
'RGB24_800x448'	0.286 Mp
'RGB24_800x600'	0.48 Mp
'RGB24_864x480'	0.41 Mp
'RGB24_960x720'	0.69 Mp

Fuente: Autores

- *Pruebas realizadas con cámaras WEB.*

Al realizar la prueba con la cámara web se observa que cumple el objetivo de capturar automáticamente la imagen e inmediatamente comenzar con el proceso de reconocimiento. El problema ahora es encontrar la cámara web adecuada para la aplicación.

Cámara web interna computador HP

Las primeras pruebas fueron realizadas con la cámara web interna del equipo de trabajo. En la figura 75 se indica la gráfica de respuesta obtenida al tomar una captura al video que tenía por resolución 0.025 Mp (Ver Tabla 2). Es una imagen pequeña y las letras no se distinguen esta distorsionada y con problemas de luminosidad, se decide realizar una prueba con el formato de resolución mayor, según tabla 2 el formato más grande permitido por esta cámara web es el de 0.31 Mp y su resultado se muestra gráficamente en la Figura 76.

Figura 75 Imagen capturada de video con resolución 0.025 Mp.

Fuente: Autores

La imagen no es visible aunque se distinguen las letras del documento, presenta problemas de enfoque, la mayoría de letras se ven distorsionadas y el fondo de la imagen no es constante por problemas de luminosidad en el entorno.

Figura 76 Imagen capturada de video con resolución 0.31 Mp.

Fuente: Autores

Cámara web Logitech C120

Con las pruebas realizadas con la cámara web HP el problema presente fue las letras distorsionadas se realizaron pruebas con la cámara web C120 que enfoca la imagen automáticamente y regula la luz de su entorno para presentar una imagen nítida. La respuesta de prueba realizada con esta cámara se indica en la figura 77. La imagen respuesta es más clara donde se observan claramente las letras. Una vez solucionado el problema se realizo el proceso con esta imagen, pero al ser sometida al algoritmo de proceso se presento unión de letras y no se reconocían

bien los espacios. Se concluye con esta prueba que la imagen es de un tamaño ‘pequeño’ para la aplicación.

Figura 77 Imagen capturada de video con resolución 0.31 Mp.

Fuente: Autores

Cámara web Logitech C615

En busca de una imagen de mayor tamaño se elige la cámara web de alta definición C615 que realiza video full HD, al observar la tabla 4 esta cámara permite video con una resolución de 2 Mp, lo cual permite una fotografía formato RGB de 1920 x 1080 que es una imagen ‘grande’ donde claramente se observan las letras y espacios necesitados en esta aplicación. No se aprecia el tamaño real de las imágenes pero se aclara que según las pruebas realizadas la Logitech C615 es la cámara web óptima para el proceso.

- Requisitos básicos webcam elegida.

Sistema operativo: Windows xp, Windows vista o Windows 7(32 o 64 bits).

Procesador: A 1 GHz o mayor.

Memoria RAM: 512 de RAM (mínimo).

Disco duro: con espacio libre de 200 MB.

Puerto: USB 1.1 disponible (se recomienda USB 2.0)

Figura 78 Imagen capturada de video con resolución 2 Mp

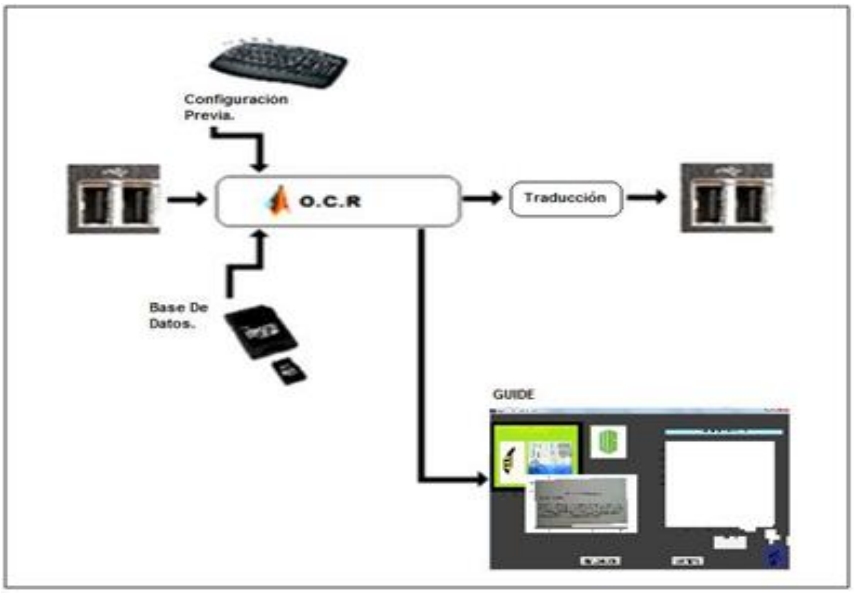
Fuente: Autores

- Requisitos del sistema

Para el desarrollo del proceso es necesario un computador encargado de realizar las operaciones especificadas en el algoritmo, a continuacion se indican las especificaciones del computador con el cual se realizo el analisis presentado y en la Figura 79, se indica el proceso realizado por el computador.

Fabricante : Hewlett – Packard
Modelo : HP Pavilion dv4 notebook PC
Procesador: AMD Athlon(tm) X2 Dual-Core QL-65 2.10 GHz
Memoria RAM: 2,00 GB
Tipo de sistema: Sistema operativo de 32 bits
Edicion windows: Windows Vista
Programa: Matlab, Logitech, Codewarrior 10.0, Octoplus Terminal.
Sonido: Altavoces y auriculares dobles, vienen con el equipo (IDT High Definition Audio).

Figura 79 Proceso realizado por el computador.



Fuente: Autores

- Configuracion Previa.

El diseño de software OCR, necesita de una configuración antes de realizar el proceso. Para capturar la imagen es necesaria una camara web que es conectada al computador por medio de un puerto USB, la salida de datos del software OCR es una cadena de caracteres que son traducidos y enviados a otro puerto USB para su respectiva visualizacion. Es necesario indicar el puerto que sera usado por cada dispositivo para obtener los resultados esperados.

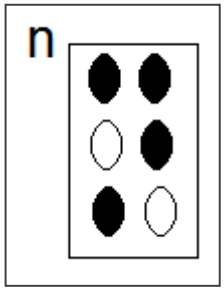
- Base de datos.

La base de datos son dos vectores de 30 valores para cada uno de los caracteres analizados, se creo una base de datos para la simulacion que se presenta en este documento (ver codigo en anexo3). Para una implementacion de este proceso se debe reaizar una base de datos mayor teniendo en cuenta el tamaño y los diferentes tipos de letra para cada caracter con el interes de que el reconocimiento sea para todo tipo de letra.

➢ Traduccion.

El proceso de traducción para cada uno de los caracteres extraidos de la imagen tienen el mismo modo de operación. En la Figura 80, se indica una imagen que simula la plantilla braille para el caracter ‘n’.

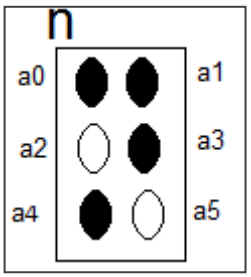
Figura 80 Plantilla Braille para caracter ‘n’



Fuente: Autores

Los puntos negros en la imagen indican el alto relieve (1 logico) y los puntos en blanco indican el bajo relieve (0 logico). El proceso utilizado en este documento es el siguiente, en la figura 81, se indica la asignacion de valores para la representacion.

Figura 81 Valores seleccionados para plantilla Braille.



Fuente: Autores

La representacion binaria de un ‘caracter’ se realiza como se indica a continuacion.

‘caracter’ = [a0 a1 a2 a3 a4 a5];

Tabla 6 Valores asignados para traducción de caracteres.

a0	32
a1	16
a2	8
a3	4
a4	2
a5	1

Fuente: Autores

Ahora para el caracter del ejemplo ‘n’.

‘n’ binario = 110110;

Buscamos su representacion decimal, se remplaza el valor indicado en la tabla 4 cuando el valor es de ‘1’.

‘n’ decimal = 32 + 16 + 0 + 4 + 2+ 0;

‘n’ decimal = 54

El caracter ascii correspondiente a este valor es el simbolo ‘6’ que es valor que enviamos via USB al puerto elegido para la visualizacion. En el anexo 1 del documento se indica una tabla de valores para los caracteres utilizados en esta simulación.

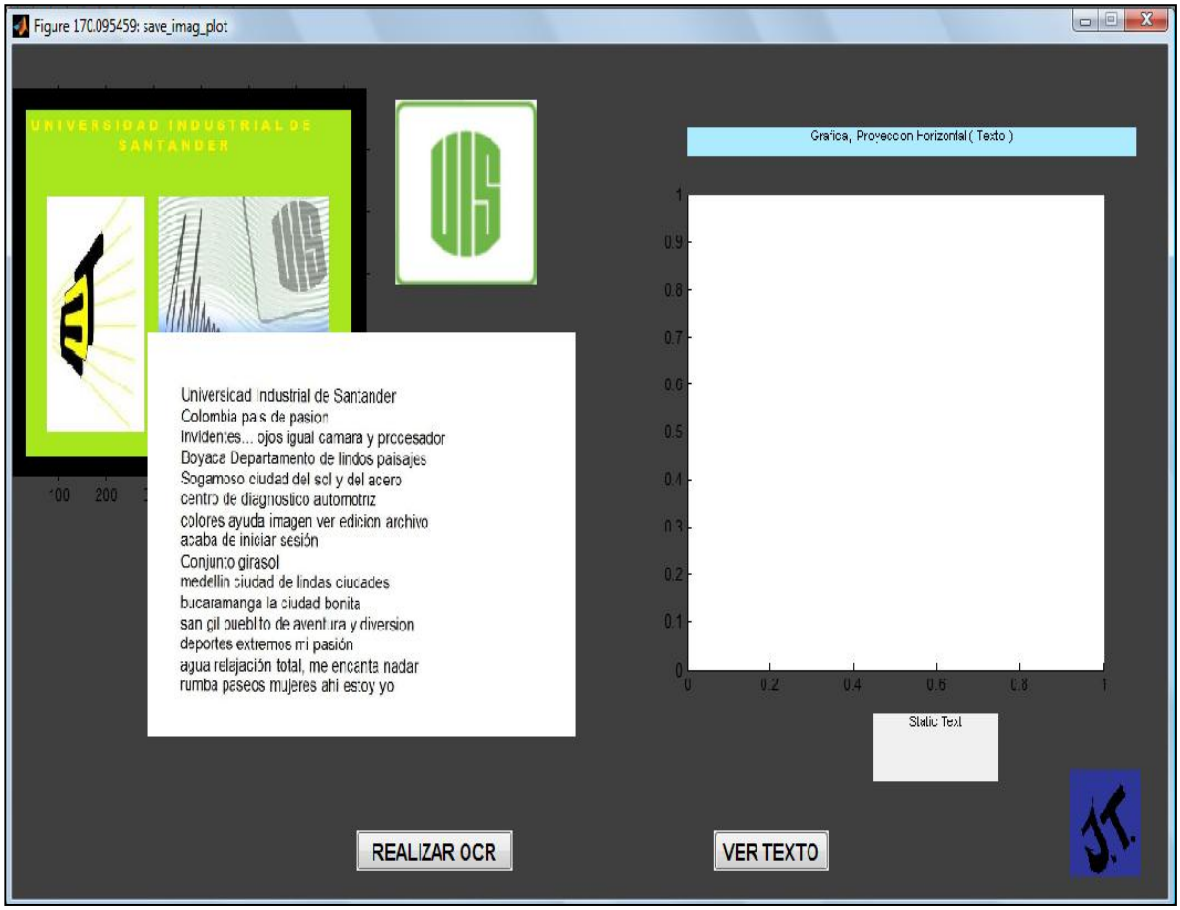
- Interfaz grafica (GUIDE).

En la figura 82 se presenta un diseño sencillo, simplemente se busca expresar los resultados de una manera elegante. Se utilizan 2 ‘push button’ con ellos se busca iniciar el proceso de traducción con el que dice ‘Realizar OCR’, y con el otro cuando se ha terminado todo el proceso visualizar el texto en formato digital ‘Ver texto’. Se busco también usar un ‘Static text’ para indicar el momento cuando el

texto está listo, es decir cuando en la pantalla ‘Static text’ aparezca un ‘Listo’ se puede hacer uso del ‘push button’ que dice ‘Ver texto’. Solo 2 ‘axes’ son importantes en este diseño, los demás son parte del fondo de la GUI, los dos importantes son uno donde se indica la imagen a tratar, y el otro donde se muestra la proyección horizontal de la imagen, y se puede observar en ella el número de líneas que el texto tiene.

(Para la creación de la Guide o interfaz grafica se obtuvo información del ‘Manual de interfaz grafica de usuario en matlab parte1, Diego Barragan’)

Figura 82 Interfaz grafica del proceso.

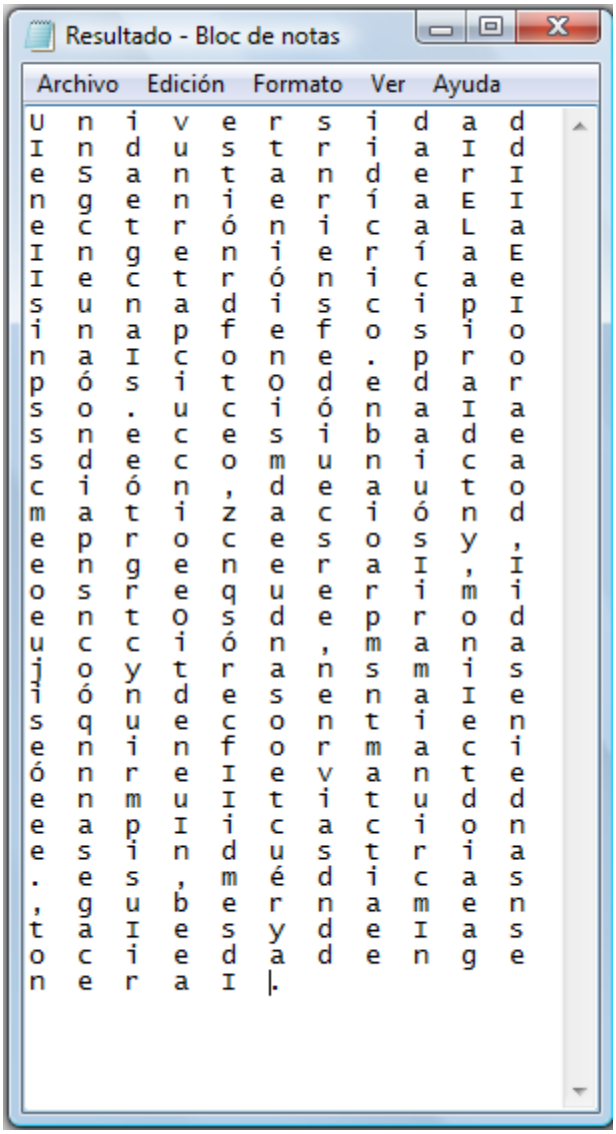


Fuente: Autores

➢ Visualizacion texto en formato digital

En la Figura 82 se observa la interfaz grafica del proceso, una vez terminado el proceso y al hacer ‘clik’ sobre el push button ‘VER TEXTO’ se abre una ventana de block de notas donde se encuentra el texto en formato digital. En la figura 83 se observa un ejemplo de este proceso donde se ven las letras y simbolos reconocidos, en esta ventana es posible editar el texto si es necesario.

Figura 83 Texto en formato digital reconocido por el proceso de OCR.

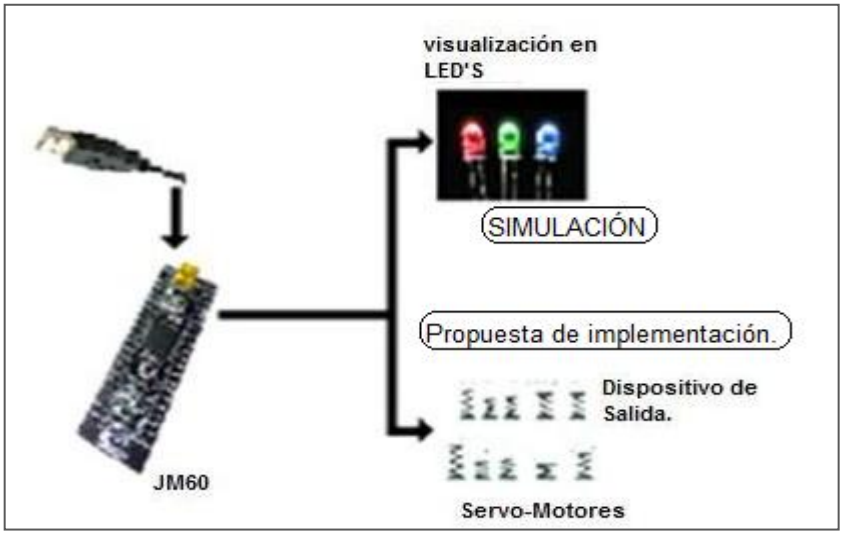


Fuente: Autores

➢ Visualizacion en Leds.

En la Figura 83, se enseña graficamente el proceso que realiza el modulo JM60, el dato que esta en el puerto USB se captura y este valor se visualiza en seis led's que simulan la plantilla Braille. Se propone el uso de seis servomotores que realizaran la visualizacion de la traduccion en alto y bajo relieve como funciona este proceso realmente, en item adelantes se indicaran los circuitos que realizaran este proceso.

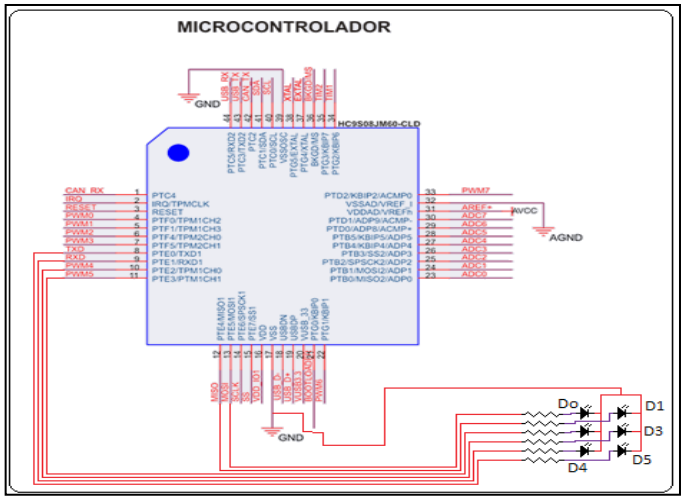
.Figura 84 Proceso modulo JM60.



Fuente: Autores

Para la visualización del proceso, se hace una conexión de seis diodos Led en el puerto E del microcontrolador en los pines menos significativos para simular la plantilla que tiene el caracter braille. Se quiere observar en los Led's la simulación de la traducción siendo el alto relieve un valor de 5V que hace que el Led se encienda y el bajo relieve un valor de 0V. En la figura 84, se muestra el circuito realizado para la simulación, con la que se quiere mostrar visualmente el resultado.

Figura 85 Conexión modulo JM60 para simulación.



Fuente: Autores

7. CONCLUSIONES

La recopilación de información y herramientas de procesamiento digital de imágenes permite de manera libre una interacción con ellas en busca del resultado adecuado para lograr el reconocimiento de caracteres. Para ellos se dividió en etapas importantes este proceso, siendo una de las primeras en importancia la corrección de inclinación, las pruebas indicadas, dan como mejor opción a "Hough" y "Canny" entre las herramientas analizadas, sin embargo queda evidencia en este documento de las demás opciones, para aplicaciones diferentes en trabajos futuros relacionados con corrección de ángulo.

La etapa central del proceso hace referencia al uso de las proyecciones y a su adecuada interpretación, dependiendo el nivel de segmentación. Es importante indicar que para una etapa avanzada de segmentación, el tamaño de la letra se vuelve una variable importante ya que se presenta la unión de letras si no se escoge el umbral correspondiente al tamaño de letra analizado. Además cuando se realizan cambios significativos de letra es importante una calibración previa al código, para revisar y reconocer el espacio entre palabras.

La clasificación y simulación del código fue una etapa interesante, la primera por su complejidad, ya que se debe realizar el proceso para varios patrones del mismo caracter, y que en ocasiones se perdían partes del símbolo que debían ser tenidos en cuenta. Para la segunda se utilizan los mismos caracteres ASCII, buscando su clasificación en hexadecimal que se acomodaba al número en decimal que representaba dicho símbolo, (Anexo1).

En un sistema OCR, la calidad de la imagen juega un papel decisivo, ya que si no se captura la información necesaria, es de gran complejidad realizar un buen

reconocimiento, y se requieren muchos recursos computacionales en el proceso de los datos. Se comprueba con esta investigación la importancia y funcionalidad de las cámaras web de alta definición, que capturan gran cantidad de información en un instante.

8. TRABAJO FUTURO

Con el desarrollo de esta investigación se llego además a la conclusión, que sería posible una serie de mejoras para este proceso y otras aplicaciones de OCR.

8.1 Bases de Datos.

Una de las tareas de mayor importancia y mayor dedicación en la investigación realizada fue dedicada a la base de datos. Esta fue realizada para varios patrones del mismo caracter para cada uno de los caracteres del alfabeto, además de símbolos presentes. Una mejora para este proceso es que esta base de datos se pueda crear automáticamente con solo seleccionar con el “mouse” los caracteres necesarios para el reconocimiento, esto es posible con ayuda de los descriptores de Fourier y de la función ”ginput” para selección de caracteres, o mejor aun utilizando la ayuda del centroide para reconocer los símbolos presentes, (Anexo2).

8.2 Implementación.

Si bien en este libro se presenta una propuesta, lo ideal sería ver la implementación total del proceso, acá se presenta una salida caracter a caracter que es visualizada en leds, pero configurando el microcontrolador, usando la memoria, podríamos crear una plantilla donde se podrían representar más caracteres.

8.3 Reconocimiento de placas en CDA (Centro de Diagnostico Automotriz).

Si bien actualmente es importante para los automotores la realización de un examen, que permita su movilización, es cierto que estos centros en ocasiones realizan las pruebas con otros vehículos de años superiores o en mejor estado, para dar satisfacción al cliente. El uso de cámaras web, ubicadas en cada una de

las etapas de las que consta el examen, donde el reconocimiento de la placa sea la activación del examen, tendría un mejor manejo y control en la realización de dicho examen.

8.4 Reconocimiento de Placas en Parqueaderos.

El cobro de tiempo de parqueadero en algunos casos es más de lo estipulado, un programa que reconozca la placa y hora de entrada así como de salida, donde el valor estipulado por ley ya sea por hora o minutos este indicado y sea fijo, permita el cobro adecuado y correcto del servicio.

9. BIBLIOGRAFÍA

[1] Carranza F. “Esqueletización”, Escuela Académico profesional de Informática. Universidad Nacional de Trujillo. Esqueletizacion.pdf.

[2] Carretero D. “Sistema de Reconocimiento de Partituras Musicales”, Escuela Politécnica superior. Universidad Carlos III de Madrid.

[3] Díaz J., Patiño E. “Diseño e Implementación de un clasificador de Imágenes Faciales para ser Utilizado en un Sistema de Reconocimiento de Rostros en Tiempo Real”, Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Universidad Industrial de Santander.

[4] Gamino A., Ortiz A., Salgado J., Trujillo V. “Algoritmo Heurístico Morfológico para Extracción de Esqueletos en Imágenes en Tonos de Gris”, Algoritmo_Heuristico.pdf

[5] Giménez E. “Aplicación de Descriptores de Fourier y Redes Neuronales Artificiales Para el Reconocimiento de Formas”. Universitat Jaume-I. FD_ANN_E67.pdf

[6] González E., Felíu V.,Oliver A., Sánchez L.. “Descriptores de Fourier para identificación y posicionamiento de objetos en entorno 3D”, Departamento de Informática, E.T.S.I Industriales, Escuela Superior de informática, E.T.S.I Industriales. Universidad de Matanzas(cuba),Universidad de Castilla(España).

[7] González, R.C. y R.E. Woods. Digital Image Processing. Prentice Hall, 2001.

[8] González, R.C. y R.E. Woods. Digital Image Processing using Matlab. Prentice Hall, 2004.

[9] González S. “Segmentación de secuencias de Imágenes Estereoscópicas mediante Competición de Regiones para el Modelado 3D de Músculos Artificiales”, Ingeniería de Telecomunicación. Universidad Politécnica de Cartagena.

[10] Grupo procesado de Imagen y Realidad Virtual. “Introducción y Resumen Teórico”. http://webs.uvigo.es/gpi-rv/ficheros/pub/reports/cap1.pdf

[11] Grupo procesado de Imagen y Realidad virtual. “O.C.R. para Matrículas de Automóviles”. http://webs.uvigo.es/gpi-rv/ficheros/pub/reports/cap2.pdf

[12] Introducción a la Bioingeniería. “Segmentación de imágenes” web.mac.com/ci3m/Site_3/Material_Adicional_files/CAP-5.PDF

[13] Jain, A.K. Foundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs, 1989.

[14] “La transformada de Fourier”, Dpto. Teoría de la Señal y Comunicaciones. Escuela Técnica Superior de Ing. Telecomunicación. Universidad de Vigo.

[15] Lopez O., Rodriguez M. “Sistema de lectura de texto e identificación de imágenes para personas no videntes”, Facultad de Ingeniería Eléctrica y Electrónica. Escuela Politécnica Nacional.

[16] Mathworks. MATLAB and Simulink for Technical Computing. www.mathworks.com/

[17] Ordóñez J. “Reconocimiento óptico de Caracteres con Redes Neuronales”. Universidad Técnica Particular de Loja. Estado-del-arte.pdf.

[18] Pajares, G. DE LA CRUZ, J. Visión por Computador. Rama. 2001.

[19] Pajares G. “Aplicaciones Industriales de la Visión por Computador”, Departamento de Arquitectura de Computadores y Automática. Universidad Complutense de Madrid.

[20] Pastor M. “Aportaciones al Reconocimiento Automático de Texto Manuscrito”, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.

[21] Pérez J. “Segmentación y búsqueda de patrones en partituras musicales”, Escuela Superior de Ciencias Experimentales y Tecnología. Universidad Rey Juan Carlos.

[22] Schalkoff, R.J. Digital Image Processing and Computer Vision. Wiley, New York. 1999.

[23] Sonka, M., HLAVAC, V., BOYLE, R. Image Processing, Analysis and Machine Vision. Brooks/Cole-Thomsom Publish. 2000.

[24] Umbaugh, S.E. Computer Imaging. Digital Image Analysis and Processing. CRC Press. 2005.

[25] Urrea J., Ospina E. "Implementación de la Transformada de Hough para la detección de líneas para un sistema de visión de bajo nivel", Departamento Matemáticas, Departamento de Ingeniería de Sistemas y Computación. Universidad Tecnológica de Pereira.

[26] Voss K., Suesse H., W.Ortmann. "Radon, Hough, Acumulación y el método SDR", Departamento de Matemáticas y Computación. Universidad Friedrich Schiller, Jena / Alemania. 2004

[27] Freescale, MCF51JM128 ColdFire® Integrated Microcontroller Reference Manual.pdf

[28] Freescale, Microcontrolador, MCF51JM128.pdf

[29] Analog Device, Motor and Power control Solutions, Motor and Power control Solutions.pdf

[30] Analog Device, Power Management, Power Management.pdf

[31] ILSI, Cristal de Quartz, ILCX10 series.pdf

[32] ST Microelectronics, Transistor Mosfet canal N, STD12NF06L.pdf

[33] ST Microelectronics, Transistor Mosfet canal P, STD12PF06.pdf

[34] ST Microelectronics, Diodo, STPSC1006.pdf

[35] Texas Instrument, Amplificador Operacional, TL084.pdf

[36] Analog Device, Full/Low Speed 5kV USB Digital Isolator, ADuM4160.pdf

[37] Texas Instrument, Optocouplers, 4N37.pdf

[38] Texas Instruments, Active Clamp current mode PWM controller, LM5025.pdf

[39] ST Microelectronics, Transistor MOSFET, STS2DNF30L.pdf

[40] FAIRCHILD Semiconductor, Transistor PN2222.

[41] Universidad de Valencia. “El Reconocimiento de Formas”. http://www.uv.es/hmr/Tesis/PDF/Cap1.pdf

[42] Ordóñez J. “Reconocimiento óptico de Caracteres con Redes Neuronales”. Universidad Técnica Particular de Loja. Estado-del-arte.pdf.

[43] Universidad Técnica del Norte. Facultad de Ingenierías en Técnicas Aplicadas “Algoritmos de Reconocimiento Capítulo 2”. http://www.utn.edu.ec

[44] Calderón F. “Operaciones morfológicas”. Operaciones Morfologicas.pdf

[45] Guajardo I, Caballos F, García J. “Operaciones sobre imágenes binarias representadas por árboles binarios basados en interpolación” Escuela Técnica Superior Ingeniería Informática. Universidad de Sevilla.

[46] Regajo D, Parra C. “Reconocimiento Automático de Matriculas”. Universidad Carlos III de Madrid.

[47] Gil J. “Reconocimiento de Patrones”. Centro de Aplicaciones de Tecnologías de Avanzada. CENATAV

[48] Cvision. Smarter Document Capture.

[49] Andrade G, López J. “Sistema de control vehicular utilizando reconocimiento óptico de caracteres”. Facultad de ingeniería Eléctrica y computación. Escuela superior Politécnica.
http://www.dspace.espol.edu.ec/bitstream/123456789/1458/1/2973.pdf

ANEXO 1

REPRESENTACION BRAILLE

En este capítulo se indican los caracteres analizados, su representación binaria, su representación decimal, y su representación hexadecimal, con estos datos, se encontró que la mejor forma de representar dicho caracter en formato braille, era enviar su equivalente en hexadecimal.

	A	Minuscula		Mayuscula		b	B	minuscula		mayusc	
		"100000"		"010001" "100000"				"101000"		"010001" "101000"	
		Decimal	32	Decimal	17 32			Decimal	40	Decimal	17 40
		Repre.	' '	Repre.	'^Q' ' '			Repre.	'('	Repre.	'^Q' '('

c	C	Minuscula		Mayuscula		d	D	minuscula		mayusc	
		"110000"		"010001" "110000"				"110100"		"010001" "110100"	
		Decimal	48	Decimal	17 48			Decimal	52	Decimal	17 52
		Repre.	'0'	Repre.	'^Q' '0'			Repre.	'4'	Repre.	'^Q' '4'

e	E	Minuscula		Mayuscula		f	F	minuscula		mayusc	
		"100100"		"010001" "100100"				"111000"		"010001" "111000"	
		Decimal	36	Decimal	17 36			Decimal	56	Decimal	17 56
		Repre.	' $'	Repre.	'^Q' '$ '			Repre.	'8'	Repre.	'^Q' '8'

g	G	Minuscula		Mayuscula		h	H	minuscula		mayusc	
		"111100"		"010001" "111100"				"101100"		"010001" "101100"	
		Decimal	60	Decimal	17 60			Decimal	44	Decimal	17 44
		Repre.	'< '	Repre.	'^Q' '<'			Repre.	','	Repre.	'^Q' ','

i	I	Minuscula		Mayuscula		j	J	minuscula		mayusc	
		"011000"		"010001" "011000"				"011100"		"010001" "011100"	
		Decimal	24	Decimal	17 24			Decimal	28	Decimal	17 28
		Repre.	' ^X'	Repre.	'^Q' '^X'			Repre.	'^\'	Repre.	'^Q' '^\'

k	K	Minuscula		Mayuscula		l	L	minuscula		mayusc	
		"100010"		"010001"				"101010"		"010001"	
				"100010"						"101010"	
		Decimal	34	Decimal	17			Decimal	42	Decimal	17
					34						42
		Repre.	' " '	Repre.	'ΛQ'			Repre.	'*'	Repre.	'ΛQ'
					' " '						'*'

m	M	Minuscula		Mayuscula		n	N	minuscula		mayusc	
		"110010"		"010001"				"110110"		"010001"	
				"110010"						"110110"	
		Decimal	50	Decimal	17			Decimal	54	Decimal	17
					50						54
		Repre.	'2'	Repre.	'ΛQ'			Repre.	'6'	Repre.	'ΛQ'
					'2'						'6'

o	O	Minuscula		Mayuscula		p	P	minuscula		mayusc	
		"100110"		"010001"				"111010"		"010001"	
				"100110"						"111010"	
		Decimal	38	Decimal	17			Decimal	58	Decimal	17
					38						58
		Repre.	'&'	Repre.	'ΛQ'			Repre.	':'	Repre.	'ΛQ'
					'&'						':'

q	Q	Minuscula		Mayuscula		r	R	minuscula		mayusc	
		"111110"		"010001"				"101110"		"010001"	
				"111110"						"101110"	
		Decimal	62	Decimal	17			Decimal	46	Decimal	17
					62						46
		Repre.	'>'	Repre.	'ΛQ'			Repre.	'.'	Repre.	'ΛQ'
					'>'						'.'

s	S	Minuscula		Mayuscula		t	T	minuscula		mayusc	
		"011010"		"010001"				"011110"		"010001"	
				"011010"						"011110"	
		Decimal	26	Decimal	17			Decimal	30	Decimal	17
					26						30
		Repre.	'ΛZ'	Repre.	'ΛQ'			Repre.	'^^'	Repre.	'ΛQ'
					'ΛZ'						'^^'

u	U	Minuscula		Mayuscula		v	V	minuscula		mayusc	
		"100011"		"010001"				"101011"		"010001"	
				"100011"						"101011"	
		Decimal	35	Decimal	17			Decimal	43	Decimal	17
					35						43
		Repre.	'#'	Repre.	'ΛQ'			Repre.	'+'	Repre.	'ΛQ'
					'#'						'+'

w	W	Minuscula		Mayuscula		x	X	minuscula		mayusc	
		"011101"		"010001"				"110011"		"010001"	
				"011101"						"110011"	
		Decimal	29	Decimal	17			Decimal	51	Decimal	17
					29						51
		Repre.	'Λ]'	Repre.	'ΛQ'			Repre.	'3'	Repre.	'ΛQ'
					'Λ]'						'3'

y	Y	Minuscula		Mayuscula		z	Z	minuscula		mayusc	
		"110111"		"010001" "110111"				"100111"		"010001" "100111"	
		Decimal	55	Decimal	17 55			Decimal	39	Decimal	17 39
		Repre.	'7'	Repre.	'^Q' '7'			Repre.	'^g'	Repre.	'^Q' '^g'

0	1	Cero		Uno		2	3	Dos		Tres	
		"010111" "011100"		"010111" "100000"				"010111" "101000"		"010111" "110000"	
		Decimal	23 28	Decimal	23 32			Decimal	23 40	Decimal	23 48
		Repre.	'^w' '^\'	Repre.	'^w' ' '			Repre.	'^w' '('	Repre.	'^w' '0'

4	5	Cuatro		Cinco		6	7	Seis		Siete	
		"010111" "110100"		"010111" "100100"				"010111" "111000"		"010111" "111100"	
		Decimal	23 52	Decimal	23 36			Decimal	23 56	Decimal	23 60
		Repre.	'^w' '4'	Repre.	'^w' '$'			Repre.	'^w' '8'	Repre.	'^w' '<'

8	9	Ocho		Nueve		.	,	Punto		Coma	
		"010111" "101100"		"010111" "011000"				"000010"		"001000"	
		Decimal	23 44	Decimal	23 24			Decimal	2	Decimal	8
		Repre.	'^w' ','	Repre.	'^w' '$'			Repre.	'^B'	Repre.	'^H'

á	é	a - tilde		e - tilde		í	ó	i - tilde		o - tilde	
		"101111"		"011011"				"010010"		"010011"	
		Decimal	47	Decimal	27			Decimal	18	Decimal	19
		Repre.	'/'	Repre.	'^['			Repre.	'^R'	Repre.	'^S'

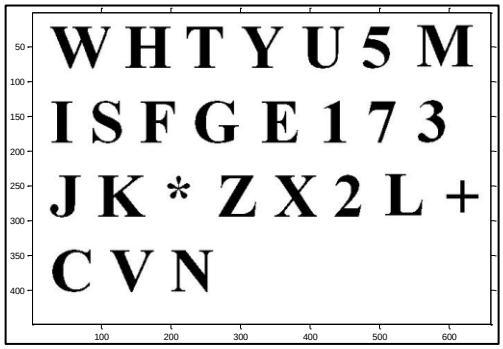
ú	Numero	u - tilde		NUMERO	
		"011111"		"010111"	
		Decimal	31	Decimal	23
		Repre.	'^_'	Repre.	'^w'

ANEXO 2

INFORMACION ADICIONAL

Para la detección y reconocimiento de símbolos, existen otras posibilidades para encontrar su posición, y a partir de esta una nueva forma de reconocimiento.

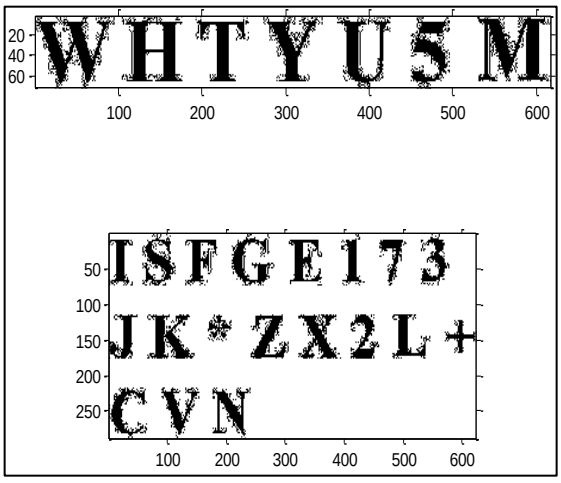
Gráfica 1 Imagen para encontrar letras.



Fuente: Autores

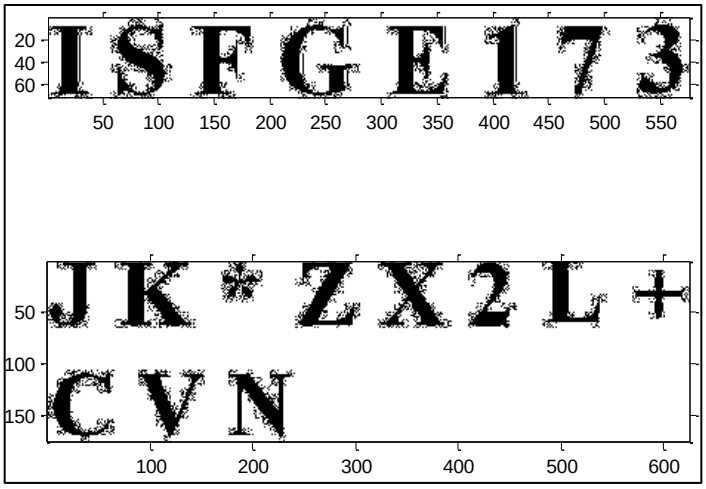
Básicamente el programa analiza la imagen en busca de donde la suma de cada pixel de una fila de cero, es decir donde se encuentran los espacios en blanco entre línea de letras y otra línea de letras. Y ahí corta y dejamos la nueva imagen

Gráfica 2 Imagen resultante primera iteración.



Fuente: Autores

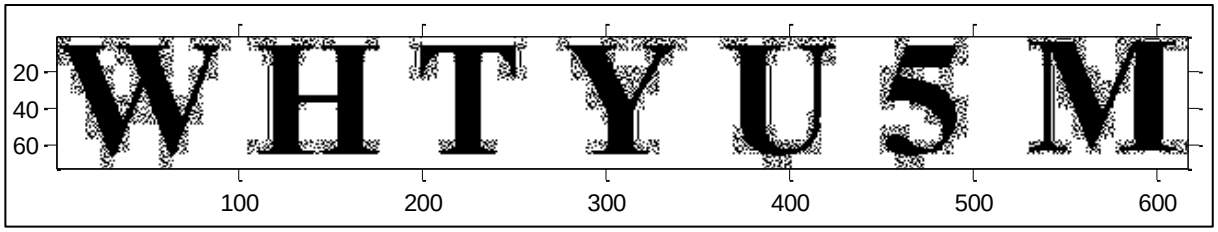
Gráfica 3 Imagen resultante 1 segunda iteración.



Fuente: Autores

Se realizan 4 iteraciones para esta imagen ya que tenemos 4 líneas. El programa se detiene cuando la imagen resultante es cero.

Gráfica 4 Figura resultante de la primera iteración.

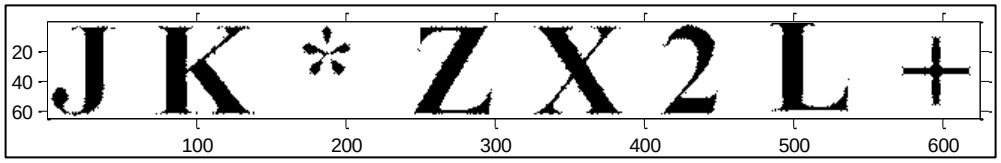


Fuente: Autores

Sin embargo observamos algo que no gusta en la imagen resultante, al observar la Grafica 4 se distingue una serie de manchas que aparecen en el contorno de las letras, si bien este programa va a formar de uno general, donde ya se ha procesado la imagen para corregir la inclinación y donde por la reducción de tamaño de tamaño realizada se pueda perder información es básico y fundamental que el proceso de esta sección sea limpio, es decir las letras salgan después del corte limpias sin las manchas observadas en la Figura. Para ello se hace uso de la operación de dilatación. “imdilate” en el toolbox de MATLAB.

- el1=strel('disk',1);
- I1=imdilate(Imagen,el1);

Gráfica 5 Figura resultante luego de la limpieza.



Fuente: Autores

- Centroide de Símbolos.

El Centroide es el centro geométrico de un elemento, en esta ocasión se hace necesario para determinar la ubicación del centro de cada letra, no es importante como tal el centro de la letra, sino que haga contacto con ella, lo anterior porque el siguiente proceso necesita la ubicación de dichas letras que conforman la línea para identificarlas y extraer letra por letra.

Para la obtención del Centroide se divide en cinco procesos:

1. Binarización de la imagen.
2. Redimensionamiento de la imagen Binarizada.

Cuando se realiza este proceso con una imagen recién adquirida ya sea de escáner o cámara, el tamaño de esta es “grande” y es necesaria cambiarlo porque el costo computacional es proporcional y hace que el proceso se vuelva extremadamente lento, como ya se indicó en ítem pasados donde era necesario hacer un cambio de tamaño para realizar la corrección de la inclinación, en este proceso no es necesario volver a cambiar el tamaño, sin embargo se hace énfasis en que es parte del proceso de encontrar un Centroide de una imagen para otras posibles aplicaciones.

3. Segmentación.

Se procede a la separación de las piezas que conforman la imagen, para de esta manera obtener la imagen por separado y comenzar el proceso de identificación, sin embargo este método resulta algo obsoleto debido a que existe una toolbox con la cual es posible realizar dicho proceso sin necesidad de realizar esta segmentación, para esta aplicación resulta más apropiado el método de la toolbox, sin embargo como se ha mencionado cuando se trabaja con imágenes cada proceso requiere un análisis especial y un tratamiento diferente.

4. Calculo de Centroide.

Para esto es necesario la ayuda de las proyecciones tanto horizontales como verticales con las cuales con las cuales se localizarían los puntos máximos y mínimos de cada proyección, siento el centroide de la imagen la intersección de ellos. Sin embargo para nuestro análisis hacemos uso de la siguiente función.

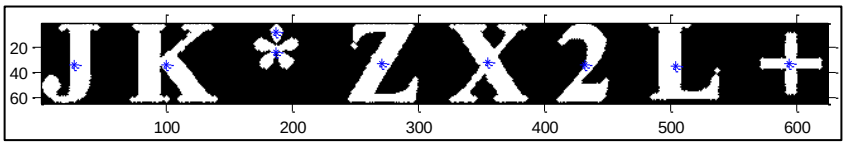
```
s  = regionprops(L, 'centroid');
```

Regionprops es una toolbox de MATLAB que se utiliza para describir regiones dependiendo de ayudas básicas como el área, numero de Euler, mínimo rectángulo que envuelve a la imagen entre otras. Para nuestra aplicación se utiliza la función centroid, que con ayuda de regionprops calcula la posición del centroide de la región.

5. Etiquetado

Dependiendo de la aplicación se etiqueta la ubicación del centroide en la Figura o simplemente se guarda el valor de este resultado para futuros usos.

Gráfica 6 Diagrama de letras y símbolos con ubicación de Centroide.



Fuente: Autores

Tabla 7 Ubicación de Centroide.

centroids <9x2 double>

	1	2	3
1	26.9282	33.6987	
2	100.3537	34.1203	
3	186.7328	23.7874	
4	187.0700	8.2900	
5	270.8031	32.9225	
6	355.0821	31.8615	
7	432.1028	33.6141	
8	503.9883	34.8601	
9	594.4674	32.5000	
10			

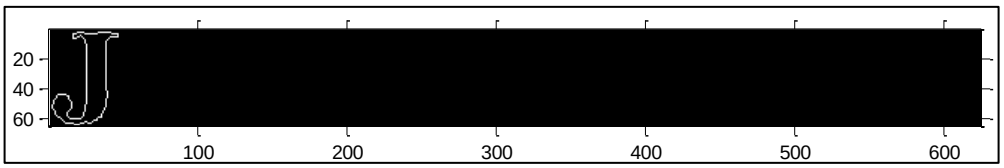
Fuente: Autores

En la Grafica 6 representa la imagen de salida, donde se distingue en azul la selección de los centroides, se representa gráficamente la etiqueta para la visualización de lo que se hace en el proceso, sin embargo en el 3 símbolo hizo falta mayor profundidad en el proceso de dilatación para reconstruir el símbolo, hay algún espacio en negro que separa al símbolo por eso se representan dos centroides, importante resaltar esto para mejorar el programa en una futura aplicación, para nuestro análisis es correcto, no se hace el reconocimiento de símbolos solo letras y números pero se debe analizar.

- Descriptores de Fourier

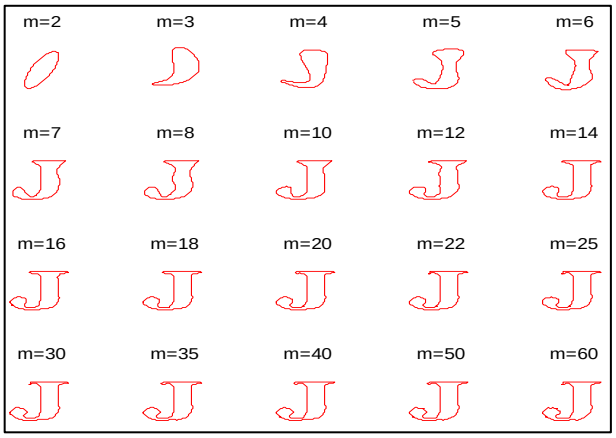
En ítem anteriores, se hablo de los descriptores de Fourier acá indicamos como funcionan, en una serie de graficas, según el numero de descriptores usados, la forma de la letra cambia, al observar la letra desde 30 descriptores hasta 60 descriptores no hay gran diferencia.

Gráfica 7 Letra J separada del texto,



Fuente: Autores

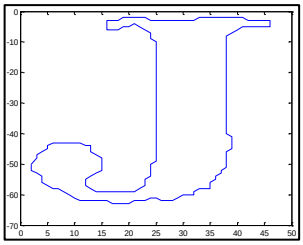
Gráfica 8 Descriptores de Fourier de letra J.



Fuente: Autores

En el proceso principal para la creación de base de datos y para la clasificación de caracteres se utilizaron los primeros 30 componentes, se observa en la Grafica 8 que desde el valor 10 se puede reconocer la forma del caracter.

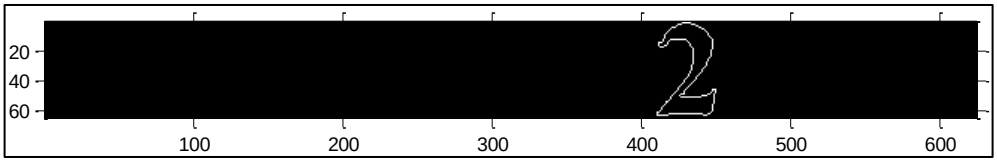
Gráfica 9 Forma reconocida por Descriptores de Fourier.



Fuente: Autores

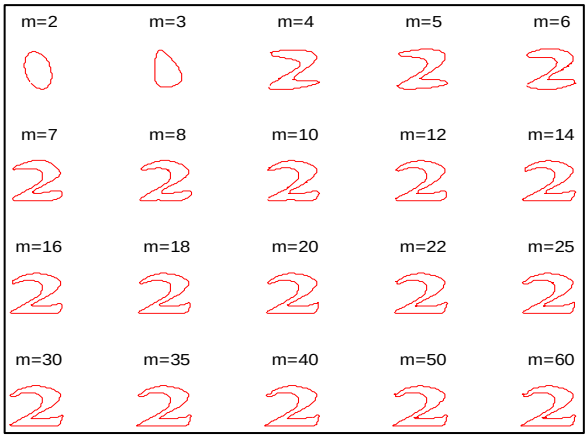
Otro ejemplo grafico del proceso se indica en la Gráfica 10.

Gráfica 10 Caracter 2 separado del texto.



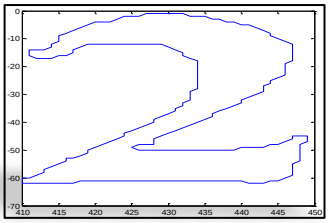
Fuente: Autores

Gráfica 11 Descriptores de Fourier del número 2.



Fuente: Autores

Gráfica 12 Forma reconocida por Descriptores de Fourier.



Fuente: Autores

Como se indica en los dos ejemplos de manera automática se logra detectar cada uno de los caracteres presentes en una imagen de texto, encontrar su ubicación mediante la ubicación del centroide, y reconocerlos para crear una base de datos para las aplicaciones deseadas. Las pruebas acá indicadas fueron realizadas para caracteres con tamaño de letras grandes, es necesario realizar el proceso adecuado para letras de texto que normalmente se usa.

- Esqueletización.

La esqueletización busca de manera óptima encontrar un patrón único continuo que contenga la menor cantidad datos posibles, pero que siga aun conteniendo un rastro del objeto original. Obtener un esqueleto, es un proceso iterativo y puede que demande mucho tiempo de procesamiento ya que generalmente todos los algoritmos, requieren muchos recorridos por toda la imagen. Existen a su vez algoritmos que provienen de estructuras matemáticas complejas y otros de cálculos lógicos; pero los cuales finalmente producen tener una nueva imagen con menos datos.

El esqueleto obtenido debe tener las siguientes tres propiedades.

· Tan delgado como sea posible
· Conectado
· Centrado

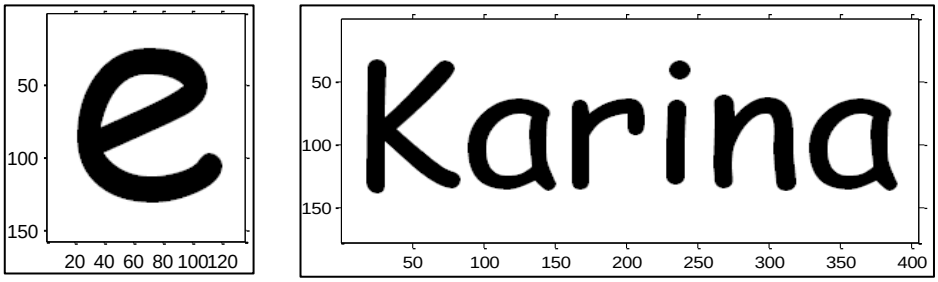
Cuando estas características son alcanzadas el algoritmo debe detenerse. Esta etapa es importante para el reconocimiento del caracter, el cual será comparado con patrones previamente guardados. Se realiza este proceso porque es más fácil la comparación cuando se tienen menos datos que analizar pensando también en el tiempo de ejecución.

La importancia del procesamiento digital de imágenes realizado previamente a esta etapa radica en que encontrar la información que necesitan los algoritmos de Reconocimiento de Patrones (RP) para que este sea robusto y funcione de una manera adecuada no es una tarea fácil. Un conflicto al que se enfrentan algunos algoritmos RP de una imagen como las redes neuronales, las memorias asociativas y los clasificadores, es saber qué información de ésta es necesaria para que el sistema sea adecuado al clasificar dichos patrones.

El algoritmo de adelgazamiento o también llamado esqueleto, proporciona información mínima de una imagen original. El algoritmo, que se propone, para encontrar esqueletos en tonos de gris se basa en el esqueleto morfológico para imágenes binarias.

1. El primer paso es obtener la imagen en escala de grises la cual será la entrada del algoritmo.

Gráfica 13 Imágenes en Escala de Grises.



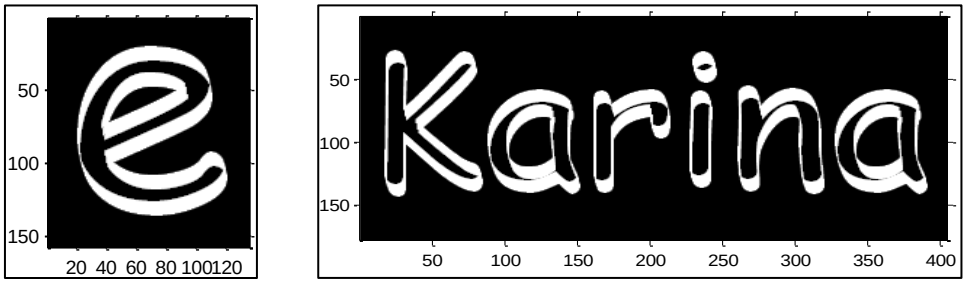
Fuente: Autores

2. Encontramos el gradiente morfológico de la imagen.

El gradiente morfológico se basa en las operaciones básicas de la morfología matemática, como lo son la dilatación y la erosión; al hacer la diferencia de la dilatación con la erosión en ese orden, se obtiene la extracción del contorno enfatizado de la imagen.

$$G = (A \oplus B) - (A \ominus B)$$

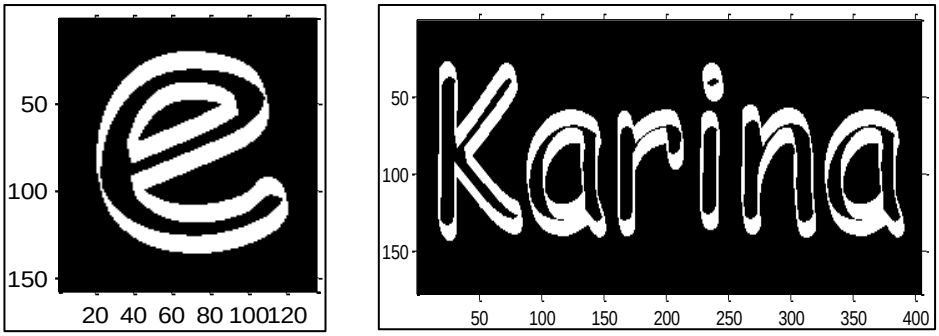
Gráfica 14 Gradiente morfológicos de imagen.



Fuente: Autores

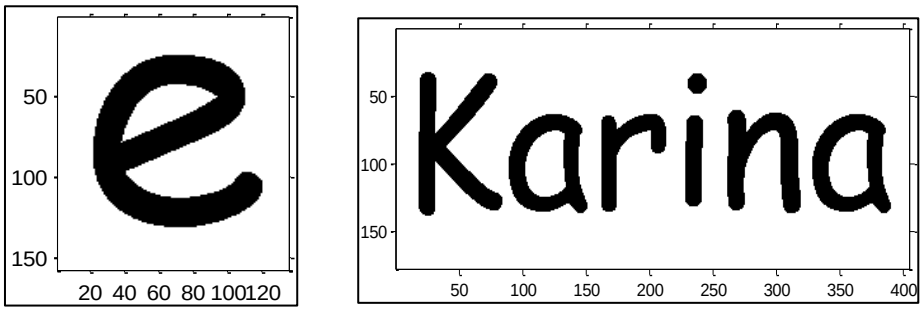
3. Realizamos la binarización de las imágenes, es decir, la imagen resultante del proceso anterior es decir la imagen G, y de la imagen de entrada se utilizan dos umbrales diferentes, de pruebas realizadas se encontró que el umbral de la imagen inicial debe ser mayor.

Gráfica 15 Binarización del gradiente de la imagen.



Fuente: Autores

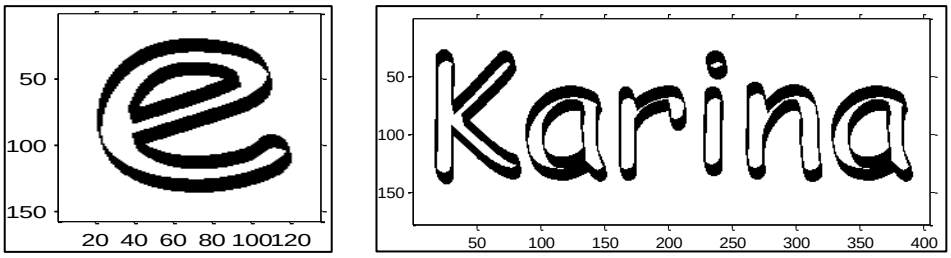
Gráfica 16 Binarización Imagen Original.



Fuente: Autores

Obtenemos el negativo de la imagen resultante de la binarización de G

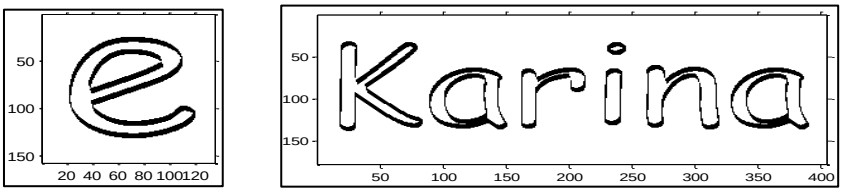
Gráfica 17 Negativo de la imagen Binarizada de G



Fuente: Autores

4. **H = Not (Binarización G) | (Binarización inicial)**

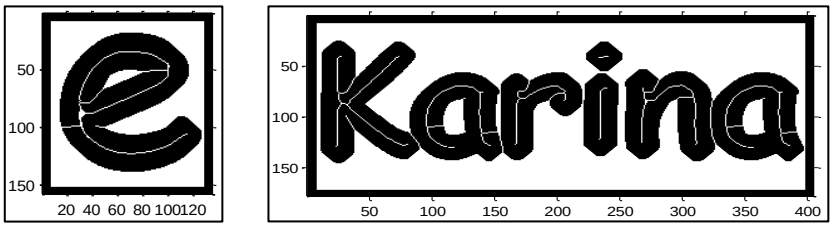
Gráfica 18 Operación para encontrar bordes de Imagen.



Fuente: Autores

5. Con ayuda de la herramienta “bwmorph” realizamos operaciones morfológicas como “thin” la cual realizara adelgazamiento repetidas veces hasta que esta deje de cambiar.

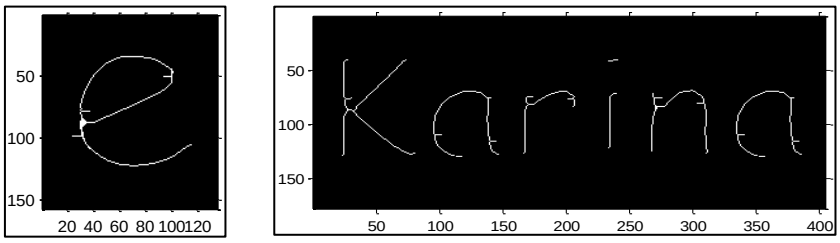
Gráfica 19 Diagrama de adelgazamiento Imagen



Fuente: Autores

6. Se realiza la diferencia entre la imagen resultante de la operación anterior y la de la binarización de la imagen inicial, el resultado es el esqueleto de la imagen.

Gráfica 20 Esqueleto de la Imagen



Fuente: Autores

La finalidad de esta etapa del proceso es obtener el esqueleto de la imagen, que se indica en la Figura anterior, es decir con forma definida, que su grosor no sea tan extenso. En la Figura número 75 se observa que se pierde un poco de

información en las letras a, sin embargo no es mucha la perdida. En la Figura 75 donde se observa la letra e, se distingue claramente el esqueleto y es claro que corresponde a una letra con toda la información necesaria, esta es la importante ya que el reconocimiento de patrones para este documento, se realiza letra a letra y es importante para la aplicación sugerida de trabajo futuro.

ANEXO 3

A continuación se indican los códigos desarrollados, para la simulación en prueba de los resultados de la investigación realizada.

1. OCR
2. Corregir Inclinación
3. Binarización
4. Proyecciones
5. Recortar Líneas en Palabras
6. Recortar Palabras en Caracteres
7. Preclasificación
8. Clasificación
9. Modulo JM60

1. OCR (principal)

```
function param = OCR(p)
%--------------//----------------------//------------------//---------//
% Programa principal OCR.
%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                 ///////
%--------------//----------------------//------------------//---------//

%p=imread('C:\Users\JT_LAD\Documents\Proyecto de Grado\Traductor_ Texto-
Braille\pruebas22.jpg');
p=imread('C:\Users\JT_LAD\Pictures\manual2_JT.jpg');
%--------------//----------------------//------------------//---------//
```

```
% PREPROCESAMIENTO.
%--------------//----------------------//------------------//---------//
p = C_Inclinacion(p);      %---\\\ Enderazamos la imagen.
p=imresize(p,3);           %---\\\ Cambio de tamaño
p = rgb2gray(p);           %---\\\ Convertimos la imagen a niveles de gris.
%figure,imshow(p),title('Imagen Niveles de Gris')
%pause(2)
f = length(p(:,1));        %---\\\ Número de filas de la matriz imagen.
p = 255 - p;       %---\\\ Inversión de niveles para facilidad de análisis.
w=(1/8.3)*[1 1 1;1 1 1;1 1 1];
p=imfilter(p,w);           %---\\\ Filtro.
%--------------//----------------------//------------------//---------//
% Binarización.
%--------------//----------------------//------------------//---------//
%Consideramos que existe en una partitura no manuscrita gran
% cantidad de blanco y negro, por lo que será suficiente con considerar un
% umbral intermedio obtenido como la posición del mínimo del histograma de
% la partitura.
p = umbraliza(p,170);
%figure,imshow(p),title('Imagen Binarizada')
%pause(2)
se=strel('disk',1);
y2=imdilate(p,se);
%imshow(y2)
%--------------//----------------------//------------------//---------//
% SEGMENTACIÓN.
%--------------//----------------------//------------------//---------//
% Separar texto en lineas
%--------------//----------------------//------------------//---------//
[pH,pV] = proyHist(y2);        %---\\\ Proyección horizontal de histograma.
%--------------//----------------------//------------------//---------//
%búsqueda de caracterización de ceros en la proyeccion horizontal para se
%parar el texto en lineas.
%--------------//----------------------//------------------//---------//
c1 = min(pH);
% Vector de límites.
vlim = [];
for j = 1:1:length(pH)
if pH(j) == 0
if c1 == min(pH)
lim1 = j;
c1 = 1;
end
else
if c1 == 1
vlim = [vlim round(((j-1)+ lim1 )/2)];
c1 = 0;
end
end
end
% Agregamos el último límite.
%vlim = [vlim round(( lim1 + f )/2)];
red1=length(vlim);
vlim(red1+1)=length(pH);
%--------------//----------------------//------------------//---------//
% SEGMENTACIÓN.
%--------------//----------------------//------------------//---------//
% Separar lineas en palabras
%--------------//----------------------//------------------//---------//
for rst = 1:1:length(vlim)-1
```

```
% lineas
a=(y2(vlim(rst):vlim(rst+1),:));
fid = fopen('Resultado.txt','a');
fprintf(fid,'\n');
fclose(fid);
% a=(y2(745:923,:));
%--------------//----------------------//-----------------//---------//
% Recortar al borde de las letras
%--------------//----------------------//-----------------//---------//
[f7 c7]=find(a);
lmaxc=max(c7);lminc=min(c7);
lmaxf=max(f7);lminf=min(f7);
imgn=a(lminf:lmaxf,lminc:lmaxc);
%figure,imshow(imgn)
%pause(1.2)
[pH2,pV2] = proyHist(imgn);
%stem(pV2)
f1 = length(imgn(1,:));
c12 = min(pV2);
% Vector de límites.
vlim2 = [];
vlim2(1)=1;
for j = 2:1:length(pV2)
if pV2(j) == 0
if c12 == min(pV2)
lim12 = j;
c12 = 1;
end
else
if c12 == 1
vlim2 = [vlim2 round(((j-1)+lim12)/2)];
c12 = 0;
end
end
end
% Agregamos el último límite.
%vlim2 = [vlim2 round((lim12+f1)/2)];
red=length(vlim2);
vlim2(red+1)=length(pV2);
%--------------//----------------------//-----------------//---------//

%--------------//----------------------//-----------------//---------//
vlim5 = palabras3(vlim2,pV2);
for ww=1:1:length(vlim5)-1
letra=imgn(:,vlim5(ww):vlim5(ww+1));
%se=strel('disk',5);
%se1=strel('square',3);
%letr=imdilate(letra,se1);
%letra=imerode(letr,se);
%figure,imshow(letra)
%pause(1.2)
retesl= vlim5(ww+1) - vlim5(ww);
%--------------//----------------------//-----------------//---------//

%--------------//----------------------//-----------------//---------//
[vlim222]= letras1(letra,retesl);
%--------------//----------------------//-----------------//---------//

%--------------//----------------------//-----------------//---------//
[param1 param2]= preclasif(vlim222,letra);
```

```
end
end
%for ggf=1:1:length(vlim5)-1
%figure,imshow(imgn(:,vlim5(ggf):vlim5(ggf+1)))
%end
end
```

2. Corregir Inclinación.

```
function Imagen_NEW = C_Inclinacion(imagen_INI)

 %imagen_INI=imread('C:\Users\JT_LAD\Pictures\Dibujo4.jpg');

 %imshow(imagen_INI)
 %imagen_INI=imresize(imagen_INI,3);

%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                 ///////
%--------------//----------------------//------------------//---------//
% Esta etapa el codigo corrige las inclinaciones INICIALES que se      //
% presentan en el texto.                                               //
% -------------//----------------------//------------------//---------//

% INICIO
% -------------//----------------------//------------------//---------//
[m,n,numComp] = size(imagen_INI);%---\\\
if numComp == 3                  %---\\\ Lo esencial es tener la imagen
   imagen = rgb2gray(imagen_INI);%---\\\ en niveles de gris, para ello si
else                             %---\\\ esta en otro formato debemos
    imagen = Imagen_INI;         %---\\\ convertirla al formato necesitado.
end                              %---\\\

% -------------//----------------------//------------------//---------//
imagen = 255-imagen; %---\\\  Para realizar la transformada de Houng se
%figure,imshow(imagen)%---\\\  hace necesario invertir los niveles de gris.

% -------------//----------------------//------------------//---------//

%[m,pos] = min(imhist(imagen))%---\\\ Trabaja con dos niveles de gris
imagen = umbraliza(imagen,180); %---\\\ facilita los calculos e iteraciones


% -------------//----------------------//------------------//---------//

imagen=edge(imagen,'canny',0.5);
%figure,imshow(imagen)
resolucion = 0.05;                                %---/// Realizamos la
[H,T] = hough(imagen,'ThetaResolution',resolucion);%---/// transformada de
                                  %---/// Hough con un resolucion de 0.5°

M_maximos = max(H);               %  El valor máximo de la transformada
[maximo, columna] = max(M_maximos);%  de Hough nos indica el angulo de
                                  %  inclinación de la linea que
```

```
angulo = (T(columna));              %  rodea el texto o borde del documento.
angulo                              %
% -------------//----------------------//------------------//---------//
% DECISIONES
% -------------//----------------------//------------------//---------//
% Si el ángulo obtenido es de + ó - 90°, es por que la imagen no posee
% ningún tipo de inclinación.
% -------------//----------------------//------------------//---------//
if (angulo==-90 || angulo==90)
Imagen_NEW = imagen_INI;
%figure,imshow(Imagen_NEW)
return
% -------------//----------------------//------------------//---------//
else
% -------------//----------------------//------------------//---------//
% Si el ángulo es negativo entonces tendremos que hacer las correcciones
% aplicándole un ángulo positivo y viceversa.
% -------------//----------------------//------------------//---------//
if angulo<0
    inclinacion = -(90+angulo);
else
    inclinacion = 90-angulo;
end
% -------------//----------------------//------------------//---------//
% PROYECCIONES
% -------------//----------------------//------------------//---------//
% Ahora para afinar más en la correccion de la inclinación iremos haciendo
% pequeñas rotaciones en torno a la aproximación que nos ha proporcionado
% la transformada de Hough.
% El proceso sería rotar la imagen un ángulo hasta obtener un valor máximo
% en la proyección del histograma.
% -------------//----------------------//------------------//---------//

iteraciones = 15;
anguloMedio = (inclinacion*pi)/180;
anguloSup = anguloMedio+resolucion/2;
anguloInf = anguloMedio-resolucion/2;

for k = 1:iteraciones
% -------------//----------------------//------------------//---------//
%TRANSFORMACION ESPECIAL
% -------------//----------------------//------------------//---------//
transformacion = maketform('affine',[cos(anguloSup) sin(anguloSup) 0 ;...
    -sin(anguloSup) cos(anguloSup) 0 ; 0 0 1]);
res = imtransform(imagen,transformacion,'FillValues',0);
%figure,imshow(res)
[pH,pV] = proyHist(res);
maxProySup = max(pH);

transformacion = maketform('affine',[cos(anguloMedio) sin(anguloMedio) 0 ;...
    -sin(anguloMedio) cos(anguloMedio) 0 ; 0 0 1]);
res = imtransform(imagen,transformacion,'FillValues',0);
[pH,pV] = proyHist(res);
maxProyMed = max(pH);

transformacion = maketform('affine',[cos(anguloInf) sin(anguloInf) 0 ;...
    -sin(anguloInf) cos(anguloInf) 0 ; 0 0 1]);
res = imtransform(imagen,transformacion,'FillValues',0);
[pH,pV] = proyHist(res);
maxProyInf = max(pH);
```

```
if (maxProyInf>maxProyMed  || maxProyInf > maxProySup)
   anguloSup = anguloMedio;
    anguloMedio = (anguloSup+anguloInf)/2;
else
    anguloInf = anguloMedio;
    anguloMedio = (anguloSup+anguloInf)/2;
end
end
% -------------//----------------------//------------------//---------//
% Finalmente obtenemos la imagen corregida.
% -------------//----------------------//------------------//---------//
transformacion = maketform('affine',[cos(anguloMedio) sin(anguloMedio) 0 ;...
    -sin(anguloMedio) cos(anguloMedio) 0 ; 0 0 1]);
% -------------//----------------------//------------------//---------//
% Se realiza la correccion a la imagen inicial...!!
% -------------//----------------------//------------------//---------//
Imagen_NEW= imtransform(imagen_INI,transformacion,'FillValues',255);
%ee=strel('disk',1);
%imCorregida1=imdilate(imCorregida,ee);
%imCorregida2=imerode(imCorregida,ee);
%w=(1/16)*[1 2 1;2 4 2;1 2 1];
%refed=imfilter(imCorregida,w);
%h=medfilt2(imCorregida,[3,3]);
%figure,imshow(Imagen_NEW),title('Imagen Corregida')
return
end
```

3. Binarización.

```
function y = umbraliza(x,n)
%-------------//----------------------//------------------//---------//
% Binarización de una imagen en niveles de gris.
%-------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 //////
% \\\\  Yolman Saul Luna Ropero...                                 //////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                 //////
%-------------//----------------------//------------------//---------//
filas = length(x(:,1));
columnas = length(x(1,:));

for j = 1:1:filas
    for k = 1:1:columnas
        if x(j,k) < n
            y(j,k) = 0;
        else
            y(j,k) = 255;
        end
    end
end
y = uint8(y);
%-------------//----------------------//------------------//---------//
```

4. Proyecciones.

```
function [pH,pV] = proyHist(x)
%--------------//----------------------//------------------//---------//
% Función para el cálculo de las proyecciones de histograma.
%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                  ///////
%--------------//----------------------//------------------//---------//
% Esta etapa el codigo corrige las inclinaciones INICIALES que se      //
% presentan en el texto.                                               //
% -------------//----------------------//------------------//---------//
% PRUEBAS
% -------------//----------------------//------------------//---------//
 %x=imread('C:\Users\JT_LAD\Pictures\Dibujo1234.jpg');
 %x=imrotate(x,-90);
 %p = rgb2gray(x);
% Número de filas de la matriz imagen.
%f = length(p(:,1));
% Inversión de niveles para mayor facilidad de análisis.
%p = 255 - p;
% Binarización. Consideramos que existe en una partitura no manuscrita gran
% cantidad de blanco y negro, por lo que será suficiente con considerar un
% umbral intermedio obtenido como la posición del mínimo del histograma de
% la partitura.
%[m,pos] = min(imhist(p));
%p = umbraliza(p,255-pos);
%p = double(p);
% -------------//----------------------//------------------//---------//
% INICIO
% -------------//----------------------//------------------//---------//
filas = length(x(:,1));
columnas = length(x(1,:));
% Proyecciones horizontal y vertical del histograma.
pH = zeros(1,filas);
pV = zeros(1,columnas);
for i = 1:1:filas
    for j = 1:1:columnas
        pH(i) = pH(i) + double(x(i,j));
        pV(j) = pV(j) + double(x(i,j));
    end
end
% -------------//----------------------//------------------//---------//
% Representación de los resultados.
% -------------//----------------------//------------------//---------//
%figure
%imshow(x), title('Imagen')
%figure
%subplot(2,1,1), stem(pH), title('Proyección horizontal del histograma');
%subplot(2,1,2); stem(pV), title('Proyección vertical del histograma');
%figure,stem(pH)
```

5. Recortar líneas en Palabras.

```
function vlim5 = palabras3(vlim2,pV2)
%--------------//----------------------//------------------//---------//
% Funcion realizan varias iteraciones para separar texto en palabas.
%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                 ///////
%--------------//----------------------//------------------//---------//
for j=1:1:length(vlim2)-1
if vlim2(j) == vlim2(j+1)
for k=j:1:length(vlim2)-1
vlim2(k)=vlim2(k+1);
end
end
end
for i=2:1:length(vlim2)-1
R=vlim2(i)-10:vlim2(i)+10;
for l=R(1):1:R(13)
if pV2(l)== 0
for k= l+1:1:l+3
if pV2(k) == 255
vlim2(i)= vlim2(i+1);
elseif pV2(k) == 510
vlim2(i) = vlim2 (i+1);
else
vlim2(i)= vlim2(i);
end
end
end
end
end
for j=1:1:length(vlim2)-1
if vlim2(j) == vlim2(j+1)
for k=j:1:length(vlim2)-1
vlim2(k)=vlim2(k+1);
end
end
end
for i=1:1:length(vlim2)-1
if vlim2(i)== max(vlim2)
break
end
end
vlim3=ones(1,i);
for j=1:1:i
vlim3(j)=vlim2(j);
end
for i=1:1:length(vlim3)-1
restasm(i)=vlim3(i+1)-vlim3(i);
end
vlim41=[];
der=1;
for ii=1:1:length(restasm)
if restasm(ii) <= 29
```

```
vlim41(der)= vlim3(ii);
der=der+1;
else
vlim41(der)= vlim3(ii);
der=der+1;
rete=round(restasm(ii)/2);
vlim41(der)=vlim3(ii)+rete;
der=der+1;
end
end

vlim5=[];
vlim5(1)=1;
ds=2;
for i=2:1:length(vlim41)-1
re=pV2(vlim41(i)-10:vlim41(i)+10);
if max(re)== 0
vlim5(ds)=vlim41(i);
ds=ds+1;
end
end
vlim5(ds)=length(pV2);
%--------------//----------------------//-----------------//---------//
```

6. Recortar Palabras en Caracteres.

```
function [vlim222]= letras1(letra,retesl)
%--------------//----------------------//-----------------//---------//
% Funcion realizan varias iteraciones para separar texto en palabas.
%--------------//----------------------//-----------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                  ///////
% \\\\  Yolman Saul Luna Ropero...                                  ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                  ///////
%--------------//----------------------//-----------------//---------//
vlim222=[];
if retesl < 40
vlim222(1)= 1;
vlim222(2)= retesl;
else
[pH22,pV22] = proyHist(letra);
%%stem(pV22);
c121 = min(pV22);
% Vector de límites.
vlim222 = [];
vlim222(1)=1;
for j = 2:1:retesl+1
if pV22(j) == 0
if c121 == min(pV22)
lim121 = j;
c121 = 1;
end
else
if c121 == 1
vlim222 = [vlim222 round(((j-1)+lim121)/2)];
c121 = 0;
end
```

```
end
end
% Agregamos el último límite.
fek=length(vlim222);
vlim222(fek+1)= length(pV22);
vlim222 = corre(retesl,pV22,vlim222);
%for ggf=1:1:length(vlim222)-1
%figure,imshow(letra(:,vlim222(ggf):vlim222(ggf+1)))
%end
end
%--------------//----------------------//------------------//---------//
```

7. Preclasificación.

```
function [param1 param2]= preclasif(vlim222,letra)
%--------------//----------------------//------------------//---------//
% Funcion obtiene los descriptores de Fourier de cada carácter analizado.
%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                  ///////
%--------------//----------------------//------------------//---------//
for ss =1:1:length(vlim222)-1
ass=letra(:,vlim222(ss):vlim222(ss+1));
[ax bx]=size(ass);
if bx ~= 1
[f7 c7]=find(ass);
lmaxc=max(c7);lminc=min(c7);
lmaxf=max(f7);lminf=min(f7);
imgn=ass(lminf:lmaxf,lminc:lmaxc);
%figure,imshow(imgn)
[a77 b77]=size(imgn);
img_r = imgn(1:a77/42:end,1:b77/24:end);
%figure,imshow(img_r)
[pH77,pV77] = proyHist(img_r);
sim1=pH77;
simb1 = sim1/max(abs(sim1));
par1 = abs(fft(simb1,30));
sim2=pV77;
simb2 = sim2/max(abs(sim2));
par2 = abs(fft(simb2,30));
param1=[];
param2=[];
param1 = [param1; par1];
param2 = [param2; par2];
funClasif12real(param2,param1);
end
end
%--------------//----------------------//------------------//---------//
```

8. Clasificación.

```
function y = funClasif12real(x,y)
%--------------//----------------------//------------------//---------//
% Funcion tiene la base de datos, realiza la comparación con los caracteres
% encontrados, y devuelve un caracter o simbolo al archivo.text, asi como
% envia la traducción del dato encontrado al puerto 10 del pc.
%--------------//----------------------//------------------//---------//
% \\\\\\Diseño y Especificacion de un Traductor de Texto a brille.///////
% \\\\  Jholbert G. Ladino Vega...                                 ///////
% \\\\  Yolman Saul Luna Ropero...                                 ///////
%\\\\\ UNIVERSIDAD INDUSTRIAL DE SANTANDER - 2012                 ///////
%--------------//----------------------//------------------//---------//
%--------------//----------------------//------------------//---------//
% BASE DE DATOS.
%--------------//----------------------//------------------//---------//
% A.
letraAV=[17.0000,6.7429,2.9500,0.3133,0.3568,1.4287,0.1895,0.3393,0.1372,0.1401,0
.2405,0.1438,0.2013,0.2039,0.1386,0.1818,0.1386,0.2039,0.2013,0.1438,0.2405,0.140
1,0.1372,0.3393,0.1895,1.4287,0.3568,0.3133,2.9500,6.7429];
letraAH=[14.3158,2.0710,2.4673,2.1552,1.4302,0.6963,0.3045,0.6705,0.9141,0.6765,0
.8272,0.3165,0.2607,0.3250,0.6971,0.7368,0.6971,0.3250,0.2607,0.3165,0.8272,0.676
5,0.9141,0.6705,0.3045,0.6963,1.4302,2.1552,2.4673,2.0710];
% a.
letraaV=[15.7692,3.8872,5.2259,1.3086,1.0850,0.7138,0.6631,0.6389,0.3022,0.4317,0
.3635,0.1206,0.0862,0.1604,0.3045,0.1795,0.3045,0.1604,0.0862,0.1206,0.3635,0.431
7,0.3022,0.6389,0.6631,0.7138,1.0850,1.3086,5.2259,3.8872];
letraaH=[17.8696,0.6202,3.1677,0.6403,0.6369,0.6133,0.3099,0.1222,0.5000,0.3315,0
.1304,0.4386,0.2226,0.1837,0.3792,0.2174,0.3792,0.1837,0.2226,0.4386,0.1304,0.331
5,0.5000,0.1222,0.3099,0.6133,0.6369,0.6403,3.1677,0.6202];
% B.
letraBV=[14.2619,1.6770,5.5244,2.8566,1.2587,1.1461,0.2980,0.6637,0.7893,0.5834,0
.7099,0.3284,0.1073,0.3838,0.4499,0.5952,0.4499,0.3838,0.1073,0.3284,0.7099,0.583
4,0.7893,0.6637,0.2980,1.1461,1.2587,2.8566,5.5244,1.6770];
letraBH=[19.1818,1.5106,3.4703,1.9650,0.7141,0.4166,0.8343,0.9235,0.8035,0.3833,0
.2405,0.3280,0.5205,0.5115,0.1688,0.0000,0.1688,0.5115,0.5205,0.3280,0.2405,0.383
3,0.8035,0.9235,0.8343,0.4166,0.7141,1.9650,3.4703,1.5106];
% b.
letrabV=[11.7381,2.2143,5.3930,2.3089,1.2797,0.7226,0.5819,0.9581,0.6584,0.6540,0
.6313,0.5073,0.3918,0.1543,0.5399,0.7857,0.5399,0.1543,0.3918,0.5073,0.6313,0.654
0,0.6584,0.9581,0.5819,0.7226,1.2797,2.3089,5.3930,2.2143];
letrabH=[14.9524,4.2319,2.0252,0.9193,1.0135,0.2381,0.3912,0.4027,0.2973,0.6029,0
.2076,0.4347,0.2586,0.6341,0.2826,0.3810,0.2826,0.6341,0.2586,0.4347,0.2076,0.602
9,0.2973,0.4027,0.3912,0.2381,1.0135,0.9193,2.0252,4.2319];
% C.
letraCV=[11.8387,1.8797,4.1729,2.2573,0.7402,1.2296,0.4620,0.0454,0.2952,0.2503,0
.2115,0.2844,0.3503,0.1955,0.1349,0.0323,0.1349,0.1955,0.3503,0.2844,0.2115,0.250
3,0.2952,0.0454,0.4620,1.2296,0.7402,2.2573,4.1729,1.8797];
letraCH=[12.7778,4.2390,1.2586,0.5596,1.0097,0.8941,0.2320,0.1129,0.2914,0.3749,0
.2003,0.2834,0.3405,0.2414,0.2215,0.2222,0.2215,0.2414,0.3405,0.2834,0.2003,0.374
9,0.2914,0.1129,0.2320,0.8941,1.0097,0.5596,1.2586,4.2390];
% c.
letracV=[13.6111,3.1058,5.1022,0.8006,0.8155,0.3481,0.5179,0.5432,0.3317,0.3256,0
.0735,0.2786,0.4142,0.1936,0.1530,0.0556,0.1530,0.1936,0.4142,0.2786,0.0735,0.325
6,0.3317,0.5432,0.5179,0.3481,0.8155,0.8006,5.1022,3.1058];
```

```
letracH=[18.1667,3.6741,1.5046,1.2349,1.2541,0.0962,0.2975,0.0862,0.6970,0.1656,0
.0962,0.1270,0.3699,0.2525,0.3579,0.1667,0.3579,0.2525,0.3699,0.1270,0.0962,0.165
6,0.6970,0.0862,0.2975,0.0962,1.2541,1.2349,1.5046,3.6741];
% D.
letraDV=[10.9048,0.9406,4.0334,3.8279,1.2834,1.6059,1.1511,0.3949,0.3238,0.6190,0
.7335,0.7685,1.0503,0.5181,0.2755,0.0476,0.2755,0.5181,1.0503,0.7685,0.7335,0.619
0,0.3238,0.3949,1.1511,1.6059,1.2834,3.8279,4.0334,0.9406];
letraDH=[15.3500,3.0042,2.2448,1.5040,0.8281,0.2784,0.3352,0.3887,0.3920,0.3050,0
.2179,0.4225,0.5076,0.4387,0.2721,0.0500,0.2721,0.4387,0.5076,0.4225,0.2179,0.305
0,0.3920,0.3887,0.3352,0.2784,0.8281,1.5040,2.2448,3.0042];
% d.
letradV=[12.5854,2.4413,5.5658,2.4828,0.9945,0.3786,0.6732,1.2437,0.9677,0.2939,0
.3755,0.4943,0.7418,0.6287,0.4136,0.3902,0.4136,0.6287,0.7418,0.4943,0.3755,0.293
9,0.9677,1.2437,0.6732,0.3786,0.9945,2.4828,5.5658,2.4413];
letradH=[16.1905,3.9272,2.3368,1.1389,1.0365,0.2651,0.3563,0.5504,0.5561,0.3401,0
.2897,0.4827,0.1430,0.4551,0.2368,0.0952,0.2368,0.4551,0.1430,0.4827,0.2897,0.340
1,0.5561,0.5504,0.3563,0.2651,1.0365,1.1389,2.3368,3.9272];
% E.
letraEV=[11.5000,3.2088,3.8792,2.4223,1.5046,1.2892,0.5898,0.1576,0.3533,0.5032,0
.5714,0.5960,0.4050,0.2550,0.2067,0.0238,0.2067,0.2550,0.4050,0.5960,0.5714,0.503
2,0.3533,0.1576,0.5898,1.2892,1.5046,2.4223,3.8792,3.2088];
letraEH=[14.1667,2.4658,5.2804,3.6957,0.8309,0.2205,0.3985,1.4173,1.2747,0.4286,0
.0833,0.2941,1.0236,0.9346,0.3002,0.0833,0.3002,0.9346,1.0236,0.2941,0.0833,0.428
6,1.2747,1.4173,0.3985,0.2205,0.8309,3.6957,5.2804,2.4658];
% e.
letraeV=[16.2571,4.3372,4.5384,0.8753,0.5075,0.3024,0.3140,0.5835,0.4960,0.1524,0
.0571,0.2912,0.2802,0.2077,0.2821,0.0286,0.2821,0.2077,0.2802,0.2912,0.0571,0.152
4,0.4960,0.5835,0.3140,0.3024,0.5075,0.8753,4.5384,4.3372];
letraeH=[16.9583,2.9466,4.1957,1.0333,0.8750,0.6250,1.2254,0.8820,0.2909,0.3358,0
.5417,0.7567,0.2397,0.1020,0.4590,0.6250,0.4590,0.1020,0.2397,0.7567,0.5417,0.335
8,0.2909,0.8820,1.2254,0.6250,0.8750,1.0333,4.1957,2.9466];
% F.
letraFV=[10.0476,3.7531,4.1649,2.6176,1.5744,0.6978,0.3110,0.7427,1.0014,0.7402,0
.5254,0.4105,0.3856,0.4346,0.4895,0.5238,0.4895,0.4346,0.3856,0.4105,0.5254,0.740
2,1.0014,0.7427,0.3110,0.6978,1.5744,2.6176,4.1649,3.7531];
letraFH=[14.9583,2.4619,5.1475,3.6429,0.8930,0.1250,0.3919,1.4620,1.4216,0.3330,0
.0417,0.3396,0.9803,0.9880,0.2753,0.1250,0.2753,0.9880,0.9803,0.3396,0.0417,0.333
0,1.4216,1.4620,0.3919,0.1250,0.8930,3.6429,5.1475,2.4619];
% f.
letrafV=[10.3659,6.6947,3.2706,1.8300,0.4238,1.2982,1.2398,0.4036,0.2404,0.9296,0
.7198,0.1750,0.5152,0.6267,0.6068,0.0244,0.6068,0.6267,0.5152,0.1750,0.7198,0.929
6,0.2404,0.4036,1.2398,1.2982,0.4238,1.8300,3.2706,6.6947];
letrafH=[14.9545,2.0580,2.2299,2.9978,0.9802,1.2273,0.8999,0.1374,0.3574,0.5445,0
.3550,0.9366,0.6014,0.4485,0.1794,0.1364,0.1794,0.4485,0.6014,0.9366,0.3550,0.544
5,0.3574,0.1374,0.8999,1.2273,0.9802,2.9978,2.2299,2.0580];
% G.
letraGV=[14.4333,1.9631,4.9492,2.8632,0.2112,1.6371,0.9061,0.2766,0.1342,0.5427,0
.1764,0.5068,0.3712,0.2097,0.0227,0.1000,0.0227,0.2097,0.3712,0.5068,0.1764,0.542
7,0.1342,0.2766,0.9061,1.6371,0.2112,2.8632,4.9492,1.9631];
letraGH=[16.3333,2.5025,3.3413,2.1598,0.4074,0.7349,0.4631,0.5457,0.7156,0.4145,0
.1667,0.1135,0.5760,0.4486,0.2729,0.2222,0.2729,0.4486,0.5760,0.1135,0.1667,0.414
5,0.7156,0.5457,0.4631,0.7349,0.4074,2.1598,3.3413,2.5025];
%g
letragV=[15.4737,2.1361,5.9554,2.0960,1.0289,0.5476,0.5333,1.0286,0.6468,0.3934,0
.5711,0.4975,0.4798,0.2674,0.5661,0.4211,0.5661,0.2674,0.4798,0.4975,0.5711,0.393
4,0.6468,1.0286,0.5333,0.5476,1.0289,2.0960,5.9554,2.1361];
letragH=[19.9048,2.3954,0.2294,0.9853,1.4023,1.1438,0.4711,0.2702,0.2253,0.2890,0
.3333,0.1939,0.1852,0.0924,0.3306,0.4762,0.3306,0.0924,0.1852,0.1939,0.3333,0.289
0,0.2253,0.2702,0.4711,1.1438,1.4023,0.9853,0.2294,2.3954];
% H.
```

```
letraHV=[10.1905,1.9149,4.6123,5.3511,0.7115,2.4527,1.4797,0.1920,0.5219,0.7443,0
.6667,1.6006,0.5460,0.8093,0.5200,0.0476,0.5200,0.8093,0.5460,1.6006,0.6667,0.744
3,0.5219,0.1920,1.4797,2.4527,0.7115,5.3511,4.6123,1.9149];
letraHH=[14.4783,3.6832,3.0204,1.8571,0.9943,0.1895,0.9824,0.8748,0.6559,0.5955,0
.1992,0.4561,0.6110,0.5141,0.5003,0.0435,0.5003,0.5141,0.6110,0.4561,0.1992,0.595
5,0.6559,0.8748,0.9824,0.1895,0.9943,1.8571,3.0204,3.6832];
% h.
letrahV=[11.7143,1.6815,6.6025,2.8955,1.3740,0.1237,0.6677,1.9083,0.5141,0.8772,0
.1798,0.6716,1.0396,0.5276,0.7508,0.1429,0.7508,0.5276,1.0396,0.6716,0.1798,0.877
2,0.5141,1.9083,0.6677,0.1237,1.3740,2.8955,6.6025,1.6815];
letrahH=[15.3043,3.7433,2.4448,1.1437,0.9358,0.3559,0.0577,0.5610,0.3606,0.2384,0
.1568,0.4261,0.1086,0.4189,0.3020,0.1739,0.3020,0.4189,0.1086,0.4261,0.1568,0.238
4,0.3606,0.5610,0.0577,0.3559,0.9358,1.1437,2.4448,3.7433];
% I.
letraIV=[21.3333,7.2772,4.3258,1.5905,0.4561,1.5878,1.4825,0.5184,0.5248,1.0756,0
.8693,0.1576,0.5218,0.8776,0.7244,0.0476,0.7244,0.8776,0.5218,0.1576,0.8693,1.075
6,0.5248,0.5184,1.4825,1.5878,0.4561,1.5905,4.3258,7.2772];
letraIH=[29.2273,0.7665,0.7482,0.7187,0.6791,0.6315,0.5778,0.5207,0.4627,0.4066,0
.3550,0.3102,0.2742,0.2481,0.2325,0.2273,0.2325,0.2481,0.2742,0.3102,0.3550,0.406
6,0.4627,0.5207,0.5778,0.6315,0.6791,0.7187,0.7482,0.7665];
% i.
letraiV=[20.9474,7.4887,4.0966,1.1441,0.5198,1.2762,1.3059,0.5368,0.4834,0.9488,0
.7368,0.2314,0.3717,0.8708,0.7871,0,0.7871,0.8708,0.3717,0.2314,0.7368,0.9488,0.4
834,0.5368,1.3059,1.2762,0.5198,1.1441,4.0966,7.4887];
letraiH=[24.5455,4.7029,3.3926,2.8905,2.3570,1.1923,0.2950,0.4011,0.0653,0.4729,0
.8332,0.7832,0.4012,0.4369,0.4598,0.1818,0.4598,0.4369,0.4012,0.7832,0.8332,0.472
9,0.0653,0.4011,0.2950,1.1923,2.3570,2.8905,3.3926,4.7029];
% J.
letraJV=[9.8974,4.4157,5.2025,2.0919,0.5723,0.6542,1.1993,1.2740,0.4097,0.2159,0.
7055,0.8537,0.5249,0.0523,0.5013,0.6154,0.5013,0.0523,0.5249,0.8537,0.7055,0.2159
,0.4097,1.2740,1.1993,0.6542,0.5723,2.0919,5.2025,4.4157];
letraJH=[11.0526,0.0767,0.1505,0.2186,0.2787,0.3287,0.3670,0.3927,0.4058,0.4068,0
.3974,0.3800,0.3584,0.3373,0.3216,0.3158,0.3216,0.3373,0.3584,0.3800,0.3974,0.406
8,0.4058,0.3927,0.3670,0.3287,0.2787,0.2186,0.1505,0.0767];
%j
letrajV=[14.3514,8.7067,1.1697,2.8125,0.8678,1.3110,0.9187,0.9048,0.8992,0.5771,0
.8229,0.2497,0.7979,0.1951,0.7213,0.1351,0.7213,0.1951,0.7979,0.2497,0.8229,0.577
1,0.8992,0.9048,0.9187,1.3110,0.8678,2.8125,1.1697,8.7067];
letrajH=[16.6190,2.8386,1.7688,1.7071,2.1411,1.8001,0.8291,0.4663,0.7199,0.2731,0
.6190,1.2575,1.2407,0.6375,0.3565,0.6190,0.3565,0.6375,1.2407,1.2575,0.6190,0.273
1,0.7199,0.4663,0.8291,1.8001,2.1411,1.7071,1.7688,2.8386];
% K.
letraKV=[10.7857,2.7916,4.2303,2.7247,1.9504,1.0960,0.5010,0.1539,0.4795,0.4542,0
.6789,0.4384,0.2665,0.2102,0.1010,0.1190,0.1010,0.2102,0.2665,0.4384,0.6789,0.454
2,0.4795,0.1539,0.5010,1.0960,1.9504,2.7247,4.2303,2.7916];
letraKH=[25.5385,0.8949,1.2523,0.8935,0.4333,0.5808,0.7866,0.7096,0.1467,0.6439,0
.5044,0.2015,0.2233,0.1835,0.1978,0,0.1978,0.1835,0.2233,0.2015,0.5044,0.6439,0.1
467,0.7096,0.7866,0.5808,0.4333,0.8935,1.2523,0.8949];
% k.
letrakV=[9.8571,2.8051,4.7356,3.0409,1.8525,0.8810,0.2142,0.7548,1.0348,1.1121,0.
9645,0.4754,0.0955,0.4592,0.7938,0.9048,0.7938,0.4592,0.0955,0.4754,0.9645,1.1121
,1.0348,0.7548,0.2142,0.8810,1.8525,3.0409,4.7356,2.8051];
letrakH=[18.6667,4.1128,2.1014,1.3243,1.1012,0.2309,0.6224,0.4239,0.2301,0.5593,0
.1333,0.5454,0.2816,0.2968,0.3812,0,0.3812,0.2968,0.2816,0.5454,0.1333,0.5593,0.2
301,0.4239,0.6224,0.2309,1.1012,1.3243,2.1014,4.1128];
% L.
letraLV=[8.0238,4.7223,4.4200,2.9156,1.1975,0.0714,0.7665,1.2416,1.0643,0.5149,0.
0238,0.5032,0.8339,0.7888,0.4382,0.0714,0.4382,0.7888,0.8339,0.5032,0.0238,0.5149
,1.0643,1.2416,0.7665,0.0714,1.1975,2.9156,4.4200,4.7223];
```

```
letraLH=[7.4167,0.0995,0.1336,0.1679,0.1941,0.2083,0.2091,0.1968,0.1735,0.1429,0.
1102,0.0830,0.0700,0.0724,0.0799,0.0833,0.0799,0.0724,0.0700,0.0830,0.1102,0.1429
,0.1735,0.1968,0.2091,0.2083,0.1941,0.1679,0.1336,0.0995];
% l.
letralV=[20.8095,7.4880,4.0419,1.2037,0.6703,1.4469,1.1535,0.5022,0.4522,0.8381,0
.8354,0.2985,0.3911,0.8591,0.7870,0,0.7870,0.8591,0.3911,0.2985,0.8354,0.8381,0.4
522,0.5022,1.1535,1.4469,0.6703,1.2037,4.0419,7.4880];
letralH=[28.8182,1.0590,0.8237,0.7872,0.9257,0.9448,0.7697,0.5285,0.4364,0.4812,0
.4810,0.4046,0.3058,0.2521,0.2596,0.2727,0.2596,0.2521,0.3058,0.4046,0.4810,0.481
2,0.4364,0.5285,0.7697,0.9448,0.9257,0.7872,0.8237,1.0590];
% M.
letraMV=[13.1190,0.7393,3.4145,4.1355,2.9971,0.8931,1.9994,1.4500,0.8308,0.5292,0
.5837,0.6640,1.1431,1.1621,0.5219,0.7857,0.5219,1.1621,1.1431,0.6640,0.5837,0.529
2,0.8308,1.4500,1.9994,0.8931,2.9971,4.1355,3.4145,0.7393];
letraMH=[26.6667,1.3653,1.0044,0.6952,0.4800,0.3055,0.6664,0.5477,0.4187,0.5427,0
.3712,0.2563,0.3948,0.3150,0.3984,0,0.3984,0.3150,0.3948,0.2563,0.3712,0.5427,0.4
187,0.5477,0.6664,0.3055,0.4800,0.6952,1.0044,1.3653];
% m.
letramV=[13.2619,1.1454,1.0422,7.5099,0.1927,0.4185,2.4241,0.1710,0.0994,1.4717,0
.1448,0.1921,2.0311,0.0057,0.0895,0.2143,0.0895,0.0057,2.0311,0.1921,0.1448,1.471
7,0.0994,0.1710,2.4241,0.4185,0.1927,7.5099,1.0422,1.1454];
letramH=[18.3478,2.3553,1.6077,1.0208,0.6970,0.3283,0.1676,0.1539,0.1008,0.2207,0
.1568,0.0445,0.1241,0.2371,0.2831,0.2609,0.2831,0.2371,0.1241,0.0445,0.1568,0.220
7,0.1008,0.1539,0.1676,0.3283,0.6970,1.0208,1.6077,2.3553];
% N.
letraNV=[12.0952,0.7315,4.5951,5.0681,0.6038,2.2263,1.4540,0.2196,0.6018,0.6582,0
.7265,1.5220,0.5071,0.7574,0.4993,0.0476,0.4993,0.7574,0.5071,1.5220,0.7265,0.658
2,0.6018,0.2196,1.4540,2.2263,0.6038,5.0681,4.5951,0.7315];
letraNH=[25.0667,2.0345,1.3198,0.5871,0.8389,0.3464,0.6453,0.3791,0.5224,0.5053,0
.6766,0.2136,0.3848,0.5336,0.5975,0,0.5975,0.5336,0.3848,0.2136,0.6766,0.5053,0.5
224,0.3791,0.6453,0.3464,0.8389,0.5871,1.3198,2.0345];
% n.
letranV=[13.4286,0.7244,7.3828,3.0667,1.4340,0.0952,0.6606,2.1143,0.3590,0.9273,0
.1429,0.6882,1.0641,0.4534,0.7776,0.0952,0.7776,0.4534,1.0641,0.6882,0.1429,0.927
3,0.3590,2.1143,0.6606,0.0952,1.4340,3.0667,7.3828,0.7244];
letranH=[18.5652,2.2338,1.5874,0.9815,0.3324,0.1895,0.0415,0.0665,0.0981,0.1707,0
.3043,0.2338,0.1512,0.0991,0.1340,0.2174,0.1340,0.0991,0.1512,0.2338,0.3043,0.170
7,0.0981,0.0665,0.0415,0.1895,0.3324,0.9815,1.5874,2.2338];
% O.
letraOV=[13.5357,1.3077,4.6678,3.2618,0.1384,1.6839,1.1775,0.2300,0.1918,0.1228,0
.0357,0.2137,0.4375,0.1073,0.2045,0.1071,0.2045,0.1073,0.4375,0.2137,0.0357,0.122
8,0.1918,0.2300,1.1775,1.6839,0.1384,3.2618,4.6678,1.3077];
letraOH=[15.0588,2.3979,1.6113,1.0621,0.7701,0.4815,0.4144,0.1114,0.0421,0.4200,0
.4118,0.2563,0.4049,0.4311,0.3420,0.2353,0.3420,0.4311,0.4049,0.2563,0.4118,0.420
0,0.0421,0.1114,0.4144,0.4815,0.7701,1.0621,1.6113,2.3979];
% o.
letraoV=[14.2222,2.3272,5.5528,1.3500,1.1517,0.5576,0.1802,0.6482,0.6906,0.4479,0
.1944,0.1066,0.4859,0.2037,0.2297,0.3889,0.2297,0.2037,0.4859,0.1066,0.1944,0.447
9,0.6906,0.6482,0.1802,0.5576,1.1517,1.3500,5.5528,2.3272];
letraoH=[19.8889,1.2929,0.9749,1.0948,0.9998,0.2887,0.1313,0.1892,0.2361,0.2585,0
.2003,0.1389,0.3163,0.1624,0.0817,0,0.0817,0.1624,0.3163,0.1389,0.2003,0.2585,0.2
361,0.1892,0.1313,0.2887,0.9998,1.0948,0.9749,1.2929];
% P.
letraPV=[11.7143,2.9344,4.8784,2.8782,0.9600,0.4292,0.6489,1.1552,1.0961,0.4847,0
.1237,0.4322,0.7436,0.7869,0.3966,0.0476,0.3966,0.7869,0.7436,0.4322,0.1237,0.484
7,1.0961,1.1552,0.6489,0.4292,0.9600,2.8782,4.8784,2.9344];
letraPH=[19.2727,0.9125,3.4138,3.0943,0.7226,0.1639,0.8277,0.7915,0.6189,0.6570,0
.0455,0.2760,0.9152,0.5730,0.3066,0.0909,0.3066,0.5730,0.9152,0.2760,0.0455,0.657
0,0.6189,0.7915,0.8277,0.1639,0.7226,3.0943,3.4138,0.9125];
% p.
```

```
letrapV=[12.3810,2.3993,5.4197,2.6033,0.9881,0.4062,0.5467,1.2576,0.9905,0.4521,0
.2807,0.2532,0.8610,0.8556,0.2862,0,0.2862,0.8556,0.8610,0.2532,0.2807,0.4521,0.9
905,1.2576,0.5467,0.4062,0.9881,2.6033,5.4197,2.3993];
letrapH=[19.7727,2.2427,0.1581,1.1304,1.2725,1.0275,0.5830,0.2168,0.3263,0.5503,0
.3608,0.4144,0.3230,0.0882,0.3035,0.2273,0.3035,0.0882,0.3230,0.4144,0.3608,0.550
3,0.3263,0.2168,0.5830,1.0275,1.2725,1.1304,0.1581,2.2427];
% Q.
letraQV=[11.6571,1.7616,4.1128,2.6414,0.7440,1.8419,0.5707,0.3149,0.1873,0.1922,0
.0000,0.3663,0.2196,0.1611,0.0530,0.1143,0.0530,0.1611,0.2196,0.3663,0.0000,0.192
2,0.1873,0.3149,0.5707,1.8419,0.7440,2.6414,4.1128,1.7616];
letraQH=[14.2222,1.5587,1.2569,0.4144,0.6914,0.4194,0.6271,0.3452,0.2026,0.2741,0
.2422,0.1196,0.2754,0.3407,0.2829,0,0.2829,0.3407,0.2754,0.1196,0.2422,0.2741,0.2
026,0.3452,0.6271,0.4194,0.6914,0.4144,1.2569,1.5587];
% q.
letraqV=[12.1667,2.3752,5.4249,2.4870,1.1196,0.5818,0.4098,1.1421,0.8975,0.5141,0
.3963,0.4134,0.6120,0.4381,0.4242,0.5000,0.4242,0.4381,0.6120,0.4134,0.3963,0.514
1,0.8975,1.1421,0.4098,0.5818,1.1196,2.4870,5.4249,2.3752];
letraqH=[20.2857,2.4881,0.3583,1.0217,1.3580,1.2150,0.6595,0.2304,0.4572,0.3253,0
.2474,0.3138,0.1855,0.0677,0.2649,0.2857,0.2649,0.0677,0.1855,0.3138,0.2474,0.325
3,0.4572,0.2304,0.6595,1.2150,1.3580,1.0217,0.3583,2.4881];
% R.
letraRV=[12.2381,1.0755,5.0655,2.7379,1.5421,1.6539,0.9884,0.3214,0.3214,0.4283,0
.7785,0.8930,0.6405,0.6096,0.3285,0.0952,0.3285,0.6096,0.6405,0.8930,0.7785,0.428
3,0.3214,0.3214,0.9884,1.6539,1.5421,2.7379,5.0655,1.0755];
letraRH=[18.2727,1.7042,3.8735,1.7174,0.9936,0.1203,0.6694,0.9606,0.7169,0.5193,0
.2083,0.5953,0.6117,0.6253,0.2897,0.0909,0.2897,0.6253,0.6117,0.5953,0.2083,0.519
3,0.7169,0.9606,0.6694,0.1203,0.9936,1.7174,3.8735,1.7042];
% r.
letrarV=[11.5610,6.6899,4.2570,0.4783,1.0801,1.3521,0.5528,0.5380,0.7235,0.4011,0
.1936,0.5240,0.4106,0.1727,0.2868,0.4878,0.2868,0.1727,0.4106,0.5240,0.1936,0.401
1,0.7235,0.5380,0.5528,1.3521,1.0801,0.4783,4.2570,6.6899];
letrarH=[15.3750,3.1105,2.4670,1.5412,1.1556,0.2917,0.5807,0.5724,0.3794,0.2632,0
.2602,0.3298,0.2560,0.1072,0.4028,0.5417,0.4028,0.1072,0.2560,0.3298,0.2602,0.263
2,0.3794,0.5724,0.5807,0.2917,1.1556,1.5412,2.4670,3.1105];
% S.
letraSV=[15.6667,4.1068,4.7054,1.3793,1.4623,0.8647,0.1583,0.0092,0.3758,0.1911,0
.4485,0.1079,0.0520,0.1733,0.1090,0.3333,0.1090,0.1733,0.0520,0.1079,0.4485,0.191
1,0.3758,0.0092,0.1583,0.8647,1.4623,1.3793,4.7054,4.1068];
letraSH=[16.8947,0.7650,3.0422,0.6802,1.2122,1.0876,0.3508,0.0820,0.3592,0.2037,0
.2735,0.2745,0.2207,0.3978,0.3319,0.0526,0.3319,0.3978,0.2207,0.2745,0.2735,0.203
7,0.3592,0.0820,0.3508,1.0876,1.2122,0.6802,3.0422,0.7650];
% s.
letrasV=[17.1212,5.3587,4.5432,0.5791,0.8928,0.4576,0.6634,0.4599,0.2643,0.2140,0
.0606,0.4063,0.5032,0.0248,0.3634,0.4545,0.3634,0.0248,0.5032,0.4063,0.0606,0.214
0,0.2643,0.4599,0.6634,0.4576,0.8928,0.5791,4.5432,5.3587];
letrasH=[19.4000,0.1846,3.0969,0.3730,0.6076,0.7937,0.4492,0.3568,0.5120,0.2467,0
.1000,0.4405,0.4102,0.2993,0.4392,0,0.4392,0.2993,0.4102,0.4405,0.1000,0.2467,0.5
120,0.3568,0.4492,0.7937,0.6076,0.3730,3.0969,0.1846];
% T.
letraTV=[7.3095,4.7710,2.7322,3.0556,1.7980,0.8810,0.4378,0.5215,0.6935,0.8240,0.
8292,0.4129,0.2497,0.3037,0.5493,0.5476,0.5493,0.3037,0.2497,0.4129,0.8292,0.8240
,0.6935,0.5215,0.4378,0.8810,1.7980,3.0556,2.7322,4.7710];
letraTH=[10.6667,4.1873,3.5121,2.4668,1.2415,0.1250,0.6667,0.9644,0.8359,0.4469,0
.0417,0.4283,0.6667,0.6659,0.4147,0,0.4147,0.6659,0.6667,0.4283,0.0417,0.4469,0.8
359,0.9644,0.6667,0.1250,1.2415,2.4668,3.5121,4.1873];
% t.
letratV=[12.1951,8.0170,2.2994,0.1372,1.3501,1.0573,0.2371,0.7680,0.6606,0.2131,0
.5610,0.3753,0.2256,0.4509,0.3746,0.0488,0.3746,0.4509,0.2256,0.3753,0.5610,0.213
1,0.6606,0.7680,0.2371,1.0573,1.3501,0.1372,2.2994,8.0170];
```

```
letratH=[15.6522,2.7120,2.2411,1.2635,1.1169,0.4518,0.1159,0.2364,0.3948,0.3296,0
.0000,0.2719,0.2847,0.1186,0.1717,0.2609,0.1717,0.1186,0.2847,0.2719,0.0000,0.329
6,0.3948,0.2364,0.1159,0.4518,1.1169,1.2635,2.2411,2.7120];
% U.
letraUV=[10.6316,2.0009,4.9612,4.9761,0.5079,1.9742,0.9073,0.5158,0.8312,0.9785,0
.3451,1.1702,0.5725,0.7781,0.3726,0.4737,0.3726,0.7781,0.5725,1.1702,0.3451,0.978
5,0.8312,0.5158,0.9073,1.9742,0.5079,4.9761,4.9612,2.0009];
letraUH=[14.3889,0.5039,0.2759,0.2118,0.3101,0.3469,0.3093,0.2241,0.2114,0.2692,0
.2778,0.2814,0.3093,0.2877,0.2129,0.1667,0.2129,0.2877,0.3093,0.2814,0.2778,0.269
2,0.2114,0.2241,0.3093,0.3469,0.3101,0.2118,0.2759,0.5039];
% u.
letrauV=[13.5476,1.0065,7.2551,3.2056,1.1375,0.1326,0.5357,1.9312,0.5213,0.6271,0
.1326,0.4827,0.9300,0.4102,0.5110,0.1190,0.5110,0.4102,0.9300,0.4827,0.1326,0.627
1,0.5213,1.9312,0.5357,0.1326,1.1375,3.2056,7.2551,1.0065];
letrauH=[15.7826,0.3893,0.5012,0.2475,0.2851,0.2609,0.2298,0.2218,0.2229,0.1027,0
.0000,0.1279,0.1379,0.1315,0.1835,0.1304,0.1835,0.1315,0.1379,0.1279,0.0000,0.102
7,0.2229,0.2218,0.2298,0.2609,0.2851,0.2475,0.5012,0.3893];
% v.
letravV=[16.8800,6.2678,4.0159,1.0768,0.1156,0.7808,0.0740,0.3787,0.1448,0.0616,0
.2227,0.1302,0.3992,0.0914,0.2460,0.2400,0.2460,0.0914,0.3992,0.1302,0.2227,0.061
6,0.1448,0.3787,0.0740,0.7808,0.1156,1.0768,4.0159,6.2678];
letravH=[25.2308,1.1993,1.4633,1.1517,0.7683,1.1538,0.8604,0.8157,0.7515,0.6830,0
.6296,0.7842,0.4735,0.4587,0.7535,0.9231,0.7535,0.4587,0.4735,0.7842,0.6296,0.683
0,0.7515,0.8157,0.8604,1.1538,0.7683,1.1517,1.4633,1.1993];
% V.
letraVV=[17.7895,5.8167,4.9247,0.6398,0.2490,1.0566,0.1932,0.5128,0.2940,0.0241,0
.1393,0.1215,0.1120,0.0868,0.1987,0.1053,0.1987,0.0868,0.1120,0.1215,0.1393,0.024
1,0.2940,0.5128,0.1932,1.0566,0.2490,0.6398,4.9247,5.8167];
letraVH=[25.1000,0.8697,0.2986,0.1854,0.5068,0.1000,0.8774,0.8544,0.6171,0.4854,0
.1000,0.4978,0.5479,0.0766,0.4617,0.1000,0.4617,0.0766,0.5479,0.4978,0.1000,0.485
4,0.6171,0.8544,0.8774,0.1000,0.5068,0.1854,0.2986,0.8697];
% W.
letraWV=[16.2308,4.6210,4.0630,1.7411,0.3749,3.1409,0.7018,0.1274,0.2354,0.1070,0
.7778,0.6263,0.2434,0.2713,0.3632,0.2308,0.3632,0.2713,0.2434,0.6263,0.7778,0.107
0,0.2354,0.1274,0.7018,3.1409,0.3749,1.7411,4.0630,4.6210];
letraWH=[24.9231,1.1537,0.5714,0.0182,0.8654,0.6296,0.4925,0.2502,0.5652,0.3259,0
.1332,0.5119,0.4925,0.2203,0.4632,0.1538,0.4632,0.2203,0.4925,0.5119,0.1332,0.325
9,0.5652,0.2502,0.4925,0.6296,0.8654,0.0182,0.5714,1.1537];
% w.
letrawV=[16.8400,5.3348,3.4777,2.0464,0.5987,2.8434,1.3167,0.8810,0.2595,0.4692,0
.1058,0.5408,0.7413,0.2051,0.0660,0.2000,0.0660,0.2051,0.7413,0.5408,0.1058,0.469
2,0.2595,0.8810,1.3167,2.8434,0.5987,2.0464,3.4777,5.3348];
letrawH=[26.0833,0.7420,0.5649,0.9019,0.5370,0.1443,0.2081,0.8634,0.0094,0.0774,0
.3632,0.2192,0.1107,0.1546,0.4823,0.2500,0.4823,0.1546,0.1107,0.2192,0.3632,0.077
4,0.0094,0.8634,0.2081,0.1443,0.5370,0.9019,0.5649,0.7420];
% X.
letraXV=[17.5217,6.7050,4.0541,1.0162,0.3782,0.8984,0.3122,0.3581,0.3334,0.1981,0
.2645,0.2259,0.2438,0.1042,0.0865,0.0435,0.0865,0.1042,0.2438,0.2259,0.2645,0.198
1,0.3334,0.3581,0.3122,0.8984,0.3782,1.0162,4.0541,6.7050];
letraXH=[23.1667,2.3352,2.1981,1.1329,0.3222,0.7407,0.4845,0.3068,0.2125,0.2383,0
.5069,0.3275,0.2528,0.2165,0.7719,0.1667,0.7719,0.2165,0.2528,0.3275,0.5069,0.238
3,0.2125,0.3068,0.4845,0.7407,0.3222,1.1329,2.1981,2.3352];
% x.
letraxV=[16.8065,6.0484,4.0244,0.9698,0.4430,1.0078,0.3652,0.1621,0.1800,0.3244,0
.2581,0.2122,0.2474,0.1407,0.1523,0.1613,0.1523,0.1407,0.2474,0.2122,0.2581,0.324
4,0.1800,0.1621,0.3652,1.0078,0.4430,0.9698,4.0244,6.0484];
letraxH=[23.3125,1.9024,1.7842,0.8235,0.9934,0.4961,0.7170,0.7147,0.2941,0.4348,0
.3802,0.1616,0.1580,0.2258,0.3014,0.5625,0.3014,0.2258,0.1580,0.1616,0.3802,0.434
8,0.2941,0.7147,0.7170,0.4961,0.9934,0.8235,1.7842,1.9024];
% Y.
```

```
letraYV=[10.5385,5.1830,0.4826,2.0951,1.7662,0.5921,0.6322,0.0518,0.3361,0.5366,0
.6934,0.5025,0.5041,0.4223,0.2194,0.2308,0.2194,0.4223,0.5041,0.5025,0.6934,0.536
6,0.3361,0.0518,0.6322,0.5921,1.7662,2.0951,0.4826,5.1830];
letraYH=[22.8000,4.1154,1.6592,0.4932,1.0209,0.1000,0.6538,0.5984,0.2380,0.2945,0
.1732,0.1703,0.0507,0.1269,0.2608,0.2000,0.2608,0.1269,0.0507,0.1703,0.1732,0.294
5,0.2380,0.5984,0.6538,0.1000,1.0209,0.4932,1.6592,4.1154];
% y.
letrayV=[14.5217,6.1092,2.9973,0.4808,0.6733,0.9876,0.4082,0.3659,0.2592,0.1550,0
.2301,0.1185,0.1887,0.0686,0.0902,0,0.0902,0.0686,0.1887,0.1185,0.2301,0.1550,0.2
592,0.3659,0.4082,0.9876,0.6733,0.4808,2.9973,6.1092];
letrayH=[26.2000,1.6838,1.3017,0.9172,0.6611,0.4359,0.4236,0.4038,0.2044,0.6624,1
.4933,0.3919,0.0236,0.0906,0.2166,0.2000,0.2166,0.0906,0.0236,0.3919,1.4933,0.662
4,0.2044,0.4038,0.4236,0.4359,0.6611,0.9172,1.3017,1.6838];
% Z.
letraZV=[19.2727,7.3892,3.0427,0.5021,0.2050,0.6315,0.5499,0.2753,0.2554,0.3082,0
.0455,0.3077,0.2301,0.2184,0.1516,0.0909,0.1516,0.2184,0.2301,0.3077,0.0455,0.308
2,0.2554,0.2753,0.5499,0.6315,0.2050,0.5021,3.0427,7.3892];
letraZH=[11.0000,3.8616,2.9806,1.9878,1.1268,0.1102,0.6332,0.8055,0.7389,0.4697,0
.1909,0.3344,0.5370,0.4942,0.4582,0.0833,0.4582,0.4942,0.5370,0.3344,0.1909,0.469
7,0.7389,0.8055,0.6332,0.1102,1.1268,1.9878,2.9806,3.8616];
% z.
letrazV=[19.6957,7.3468,2.7244,0.8940,0.2845,0.4348,0.6843,0.5856,0.3043,0.0190,0
.2609,0.3698,0.2583,0.1232,0.2410,0.3043,0.2410,0.1232,0.2583,0.3698,0.2609,0.019
0,0.3043,0.5856,0.6843,0.4348,0.2845,0.8940,2.7244,7.3468];
letrazH=[11.5000,3.9229,3.3713,2.2203,1.0080,0.0417,0.6884,1.0104,0.9005,0.4902,0
.0722,0.4475,0.5575,0.6131,0.5320,0.0833,0.5320,0.6131,0.5575,0.4475,0.0722,0.490
2,0.9005,1.0104,0.6884,0.0417,1.0080,2.2203,3.3713,3.9229];
%NUMEROS
% 1.
Num1V=[1.22891566265060,1.18412596155967,1.07962788162711,0.988130049882155,0.965
966255345384,0.994030666618473,1.01489238527108,1.00259256387769,0.97368490751540
1,0.956415073455549,0.957888154713299,0.963576394707829,0.961011229186750,0.95154
0100520833,0.942915837253968,0.939759036144578,0.942915837253968,0.95154010052083
3,0.961011229186750,0.963576394707829,0.957888154713299,0.956415073455549,0.97368
4907515401,1.00259256387769,1.01489238527108,0.994030666618473,0.965966255345384,
0.988130049882155,1.07962788162711,1.18412596155967];
Num1H=[10.5000000000000,3.98510968499922,2.55841895612745,1.72047740058897,1.4376
6196822715,0.750000000000000,0.750000000000000,0.700992272546999,0.40827386240656
3,0.406149620291133,0.750000000000000,0.695532777136725,0.750000000000000,0.28503
4714172260,0.182846830206558,0,0.182846830206558,0.285034714172260,0.750000000000
000,0.695532777136725,0.750000000000000,0.406149620291133,0.408273862406563,0.700
992272546999,0.750000000000000,0.750000000000000,1.43766196822715,1.7204774005889
7,2.55841895612745,3.98510968499922];
% 2.
Num2V=[5.71428571428571,0.382659935445714,1.32770701876977,0.832189210197366,0.75
6908660344450,0.868966075756888,0.793120278302475,1.02064521516301,0.235431362296
613,1.22112021841375,0.377964473009227,0.943320896354241,0.835159168296860,0.4208
35414250386,0.793261362425738,0.142857142857142,0.793261362425738,0.4208354142503
86,0.835159168296860,0.943320896354241,0.377964473009227,1.22112021841375,0.23543
1362296613,1.02064521516301,0.793120278302475,0.868966075756888,0.756908660344450
,0.832189210197366,1.32770701876977,0.382659935445714];
Num2H=[1,1,1.00000000000000,1,1,1,1.00000000000000,1.00000000000000,1.00000000000
000,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.000000
00000000,1.00000000000000,1.00000000000000,1.00000000000000,1,1,1,1.0000000000000
0,1];
% 3.
Num3V=[6.13333333333333,0.785800722070119,2.05281227004846,0.279173340527241,1.43
632493144014,0.592546294487706,0.553465790064941,0.717875131400024,0.916083534591
411,0.815036482717850,0.466666666666667,0.710363010872822,0.134610785868874,0.752
994925706424,0.682602120663791,0.266666666666667,0.682602120663791,0.752994925706
```

```
424,0.134610785868874,0.710363010872822,0.466666666666667,0.815036482717850,0.916
083534591411,0.717875131400024,0.553465790064941,0.592546294487706,1.436324931440
14,0.279173340527241,2.05281227004846,0.785800722070119];
Num3H=[4.50000000000000,4.28444087199787,3.68156988181598,2.81302665480166,1.8526
8298442253,1.00000000000000,0.500000000000000,0.481626630191664,0.476650977844342
,0.294755897948401,6.20633538311818e-
17,0.296375309958991,0.500000000000000,0.569002529951373,0.535175494447114,0.5000
00000000000,0.535175494447114,0.569002529951373,0.500000000000000,0.2963753099589
91,6.20633538311818e-
17,0.294755897948401,0.476650977844342,0.481626630191664,0.500000000000000,1.0000
0000000000,1.85268298442253,2.81302665480166,3.68156988181598,4.28444087199787];
% 4.
Num4V=[1,1,1.00000000000000,1,1,1,1.00000000000000,1.00000000000000,1.00000000000
000,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.000000
00000000,1.00000000000000,1.00000000000000,1.00000000000000,1,1,1,1.0000000000000
0,1];
Num4H=[1.50000000000000,0.374955402013947,0.218456135200220,0.115075208389028,0.1
77101410832430,0.128769688409428,0.0340660758046434,0.0236836273945433,0.08683303
26915121,0.126863828998258,0.107142857142857,0.126430707509107,0.124181863137632,
0.0696606374583908,0.0627156774493966,0.0714285714285714,0.0627156774493966,0.069
6606374583908,0.124181863137632,0.126430707509107,0.107142857142857,0.12686382899
8258,0.0868330326915121,0.0236836273945433,0.0340660758046434,0.128769688409428,0
.177101410832430,0.115075208389028,0.218456135200220,0.374955402013947];
% 5.
Num5V=[9.13333333333333,1.37813463966502,2.54321112093930,1.74372083797356,0.2844
16779322706,1.41106736590112,0.245660715769269,1.66227044805850,0.762027246970066
,0.738988704841590,1.09138036958299,1.78818700752241,1.03154992955851,0.890686494
748927,0.528571332683166,0.0666666666666673,0.528571332683166,0.890686494748927,1
.03154992955851,1.78818700752241,1.09138036958299,0.738988704841590,0.76202724697
0066,1.66227044805850,0.245660715769269,1.41106736590112,0.284416779322706,1.7437
2083797356,2.54321112093930,1.37813463966502];
Num5H=[4,3.79003766483841,3.20553181258663,2.37498978412272,1.49667615846563,0.86
6025403784439,0.809016994374947,0.973088595296687,0.972985998611842,0.78065583025
6033,0.500000000000000,0.306297945806990,0.309016994374947,0.308049757675851,0.19
4485273686708,0,0.194485273686708,0.308049757675851,0.309016994374947,0.306297945
806990,0.500000000000000,0.780655830256033,0.972985998611842,0.973088595296687,0.
809016994374947,0.866025403784439,1.49667615846563,2.37498978412272,3.20553181258
663,3.79003766483841];
% 6.
Num6V=[2.87500000000000,2.81022071181827,2.62232382181396,2.33025199289484,1.9644
9821582040,1.56624551076771,1.18786146922866,0.894364308580659,0.746890809267807,
0.733945263360698,0.760345316287277,0.745752732882469,0.657826063577540,0.4966134
26925827,0.285164255503292,0.125000000000000,0.285164255503292,0.496613426925827,
0.657826063577540,0.745752732882469,0.760345316287277,0.733945263360698,0.7468908
09267807,0.894364308580659,1.18786146922866,1.56624551076771,1.96449821582040,2.3
3025199289484,2.62232382181396,2.81022071181827];
Num6H=[3.50000000000000,3.29076133211800,2.70767001525198,1.87684375639992,0.9976
87613552592,0.500000000000000,0.809016994374947,1.08294266575672,1.08495785721958
,0.852911199400401,0.500000000000000,0.218023639140459,0.309016994374948,0.447873
565302382,0.495992905520407,0.500000000000000,0.495992905520407,0.447873565302382
,0.309016994374948,0.218023639140459,0.500000000000000,0.852911199400401,1.084957
85721958,1.08294266575672,0.809016994374947,0.500000000000000,0.997687613552592,1
.87684375639992,2.70767001525198,3.29076133211800];
% 7.
Num7V=[8.09090909090909,1.52577500828993,0.780374865791634,1.45057662621550,1.703
54746244947,1.52932762193279,0.890366187429036,0.678239795457180,0.74151236282547
9,1.64742504219739,1.28243054360599,0.928570687526111,0.506788742122975,0.6761626
27429435,0.702011065485508,1.18181818181818,0.702011065485508,0.676162627429435,0
.506788742122975,0.928570687526111,1.28243054360599,1.64742504219739,0.7415123628
```

```
25479,0.678239795457180,0.890366187429036,1.52932762193279,1.70354746244947,1.450
57662621550,0.780374865791634,1.52577500828993];
Num7H=[1,1,1.00000000000000,1,1,1,1.00000000000000,1.00000000000000,1.00000000000
000,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.00000000000000,1,1,1,1.000000
00000000,1.00000000000000,1.00000000000000,1.00000000000000,1,1,1,1.0000000000000
0,1];
% 8.
Num8V=[4.61111111111111,0.215220619064634,2.52884316744998,1.29717206646734,1.366
10459187804,0.580017028272808,1.81840573174390,1.08614056439494,1.09927288665254,
1.27029166976453,0.364302140239000,0.769195972989528,0.649457971303250,1.01237784
805603,0.426080000778144,1.27777777777778,0.426080000778144,1.01237784805603,0.64
9457971303250,0.769195972989528,0.364302140239000,1.27029166976453,1.099272886652
54,1.08614056439494,1.81840573174390,0.580017028272808,1.36610459187804,1.2971720
6646734,2.52884316744998,0.215220619064634];
Num8H=[4,3.79003766483841,3.20553181258663,2.37498978412272,1.49667615846563,0.86
6025403784439,0.809016994374947,0.973088595296687,0.972985998611842,0.78065583025
6033,0.500000000000000,0.306297945806990,0.309016994374947,0.308049757675851,0.19
4485273686708,0,0.194485273686708,0.308049757675851,0.309016994374947,0.306297945
806990,0.500000000000000,0.780655830256033,0.972985998611842,0.973088595296687,0.
809016994374947,0.866025403784439,1.49667615846563,2.37498978412272,3.20553181258
663,3.79003766483841];
% 9.
Num9V=[7.04761904761905,1.10485499980428,1.66678825223832,0.548843625817959,1.717
31560449509,2.40794226281344,0.663895320329088,0.608523124021618,0.30589509573629
2,1.53947149647467,1.57431192903015,1.09473816239600,0.408598524485871,0.75720220
8208288,1.19520095956222,1.04761904761905,1.19520095956222,0.757202208208288,0.40
8598524485871,1.09473816239600,1.57431192903015,1.53947149647467,0.30589509573629
2,0.608523124021618,0.663895320329088,2.40794226281344,1.71731560449509,0.5488436
25817959,1.66678825223832,1.10485499980428];
Num9H=[3,2.95629520146761,2.82709091528520,2.61803398874990,2.33826121271772,2.00
000000000000,1.61803398874990,1.20905692653531,0.790943073464693,0.38196601125010
5,0,0.338261212717716,0.618033988749895,0.827090915285202,0.956295201467611,1,0.9
56295201467611,0.827090915285202,0.618033988749895,0.338261212717716,0,0.38196601
1250105,0.790943073464693,1.20905692653531,1.61803398874990,2.00000000000000,2.33
826121271772,2.61803398874990,2.82709091528520,2.95629520146761];
% 0.
Num0V=[1.97058823529412,1.87806811866368,1.64461734360316,1.38069642207188,1.1889
9632773683,1.07141209412276,0.959706492549959,0.844934213793437,0.800174764953946
,0.845449381402516,0.885777667433859,0.846654258998335,0.751746378579657,0.693951
906490867,0.712515162953098,0.735294117647059,0.712515162953098,0.693951906490867
,0.751746378579657,0.846654258998335,0.885777667433859,0.845449381402516,0.800174
764953946,0.844934213793437,0.959706492549959,1.07141209412276,1.18899632773683,1
.38069642207188,1.64461734360316,1.87806811866368];
Num0H=[3,2.95629520146761,2.82709091528520,2.61803398874990,2.33826121271772,2.00
000000000000,1.61803398874990,1.20905692653531,0.790943073464693,0.38196601125010
5,0,0.338261212717716,0.618033988749895,0.827090915285202,0.956295201467611,1,0.9
56295201467611,0.827090915285202,0.618033988749895,0.338261212717716,0,0.38196601
1250105,0.790943073464693,1.20905692653531,1.61803398874990,2.00000000000000,2.33
826121271772,2.61803398874990,2.82709091528520,2.95629520146761];
% uj.
letraujV=[10.1026,1.9381,1.7512,5.2546,2.6259,1.8506,1.0817,0.5051,0.3097,0.4849,
0.8072,1.0860,1.3967,0.1911,0.1430,0.0513,0.1430,0.1911,1.3967,1.0860,0.8072,0.48
49,0.3097,0.5051,1.0817,1.8506,2.6259,5.2546,1.7512,1.9381];
letraujH=[14.9474,3.8184,2.8644,1.0629,0.6670,1.3346,0.9278,0.3156,0.6126,0.6556,
0.3451,0.3726,0.5674,0.4926,0.4042,0.4211,0.4042,0.4926,0.5674,0.3726,0.3451,0.65
56,0.6126,0.3156,0.9278,1.3346,0.6670,1.0629,2.8644,3.8184];
% re.
letrareV=[13.0952,1.9327,3.2286,4.8691,1.0921,0.9316,1.7165,0.5450,0.4878,0.5331,
0.5599,0.7417,0.6972,0.9192,0.0178,0.0476,0.0178,0.9192,0.6972,0.7417,0.5599,0.53
31,0.4878,0.5450,1.7165,0.9316,1.0921,4.8691,3.2286,1.9327];
```

```
letrareH=[20.1000,1.3952,4.2982,0.2067,0.6765,0.6245,1.0515,0.6197,0.1760,0.1216,
0.4583,0.6719,0.1637,0.2014,0.2535,0.6000,0.2535,0.2014,0.1637,0.6719,0.4583,0.12
16,0.1760,0.6197,1.0515,0.6245,0.6765,0.2067,4.2982,1.3952];
% oj.
letraojV=[9.4872,1.6857,2.7413,3.7400,2.1041,2.1398,0.8919,0.2031,0.1222,0.4498,0
.6426,0.6712,0.8007,0.7073,0.2151,0.0513,0.2151,0.7073,0.8007,0.6712,0.6426,0.449
8,0.1222,0.2031,0.8919,2.1398,2.1041,3.7400,2.7413,1.6857];
letraojH=[16.3529,3.9806,3.5066,1.4499,1.0706,0.9132,0.8609,0.6635,0.6787,0.9584,
0.3857,0.4076,0.4880,0.5867,0.4707,0.1176,0.4707,0.5867,0.4880,0.4076,0.3857,0.95
84,0.6787,0.6635,0.8609,0.9132,1.0706,1.4499,3.5066,3.9806];
% ra.
letraraV=[14.1667,1.8758,3.1823,5.5824,1.0242,0.6830,1.9453,0.6828,0.6898,0.9342,
0.2271,0.5585,1.4014,0.7253,0.2661,0.0238,0.2661,0.7253,1.4014,0.5585,0.2271,0.93
42,0.6898,0.6828,1.9453,0.6830,1.0242,5.5824,3.1823,1.8758];
letraraH=[19.2273,1.0538,2.8865,0.3840,0.4450,0.3936,0.6061,0.0355,0.2101,0.2346,
0.0787,0.3955,0.2498,0.1517,0.1212,0.1364,0.1212,0.1517,0.2498,0.3955,0.0787,0.23
46,0.2101,0.0355,0.6061,0.3936,0.4450,0.3840,2.8865,1.0538];
% ro.
letraroV=[12.2381,1.0236,2.4661,5.5909,0.8779,0.5719,2.1690,0.4196,0.6827,0.3394,
0.7027,0.6075,0.8051,1.0482,0.2652,0.0952,0.2652,1.0482,0.8051,0.6075,0.7027,0.33
94,0.6827,0.4196,2.1690,0.5719,0.8779,5.5909,2.4661,1.0236];
letraroH=[18.6000,2.2319,1.6725,0.8224,0.6483,0.4500,0.0477,0.1313,0.2162,0.1055,
0.0866,0.1313,0.1739,0.2378,0.0561,0,0.0561,0.2378,0.1739,0.1313,0.0866,0.1055,0.
2162,0.1313,0.0477,0.4500,0.6483,0.8224,1.6725,2.2319];
% rt.
letrartV=[10.8571,1.4897,6.4662,0.5590,2.8770,0.0476,0.3220,0.3477,1.4463,0.1208,
1.2963,0.1935,0.1169,0.0330,1.0988,0.0952,1.0988,0.0330,0.1169,0.1935,1.2963,0.12
08,1.4463,0.3477,0.3220,0.0476,2.8770,0.5590,6.4662,1.4897];
letrartH=[13.6250,3.5657,3.2398,1.4222,1.2036,0.4805,0.3659,0.3352,0.6552,0.1765,
0.1909,0.3486,0.6399,0.2755,0.3130,0.2917,0.3130,0.2755,0.6399,0.3486,0.1909,0.17
65,0.6552,0.3352,0.3659,0.4805,1.2036,1.4222,3.2398,3.5657];
% aj.
letraajV=[10.8462,2.1998,2.5749,3.9156,2.4560,1.8623,0.7057,0.3899,0.1172,0.1718,
0.6451,0.9840,0.9406,0.3851,0.1225,0.1282,0.1225,0.3851,0.9406,0.9840,0.6451,0.17
18,0.1172,0.3899,0.7057,1.8623,2.4560,3.9156,2.5749,2.1998];
letraajH=[15.8500,4.2156,2.8180,2.4950,1.3016,0.8718,0.9813,0.2841,0.2617,1.0429,
0.4359,0.3997,0.4685,0.6734,0.5969,0.2500,0.5969,0.6734,0.4685,0.3997,0.4359,1.04
29,0.2617,0.2841,0.9813,0.8718,1.3016,2.4950,2.8180,4.2156];
% ,.
letracomaV=[16.1000,7.6133,1.5181,0.5966,1.0763,0.6016,0.6178,0.3034,0.5768,0.371
3,0.3473,0.2761,0.0843,0.4242,0.5237,0,0.5237,0.4242,0.0843,0.2761,0.3473,0.3713,
0.5768,0.3034,0.6178,0.6016,1.0763,0.5966,1.5181,7.6133];
letracomaH=[23.2727,3.2783,0.8555,0.3870,0.5500,0.2531,0.2748,0.1342,0.3208,0.267
0,0.2273,0.2568,0.1843,0.2934,0.2470,0.1818,0.2470,0.2934,0.1843,0.2568,0.2273,0.
2670,0.3208,0.1342,0.2748,0.2531,0.5500,0.3870,0.8555,3.2783];
% ..
letrapuntoV=[20.0270,7.9096,3.6259,0.3918,0.8708,1.0174,0.8315,0.3379,0.5052,0.63
77,0.3832,0.0885,0.3087,0.5438,0.5846,0.2973,0.5846,0.5438,0.3087,0.0885,0.3832,0
.6377,0.5052,0.3379,0.8315,1.0174,0.8708,0.3918,3.6259,7.9096];
letrapuntoH=[28.0909,1.6914,1.1568,0.6502,0.5858,0.6364,0.4847,0.2136,0.0374,0.08
35,0.0000,0.1362,0.2240,0.2192,0.1474,0.0909,0.1474,0.2192,0.2240,0.1362,0.0000,0
.0835,0.0374,0.2136,0.4847,0.6364,0.5858,0.6502,1.1568,1.6914];
% d*.
letraddV=[9.7317,1.9633,4.5798,3.3089,1.1869,1.3499,0.5479,0.3936,0.8498,0.5959,0
.7329,0.7153,0.1848,0.3006,0.6034,0.5122,0.6034,0.3006,0.1848,0.7153,0.7329,0.595
9,0.8498,0.3936,0.5479,1.3499,1.1869,3.3089,4.5798,1.9633];
letraddH=[13.1000,3.3961,2.0855,1.4613,1.4547,0.3000,0.8202,0.1969,0.3265,0.2113,
0.5000,0.2422,0.3837,0.0638,0.0554,0.3000,0.0554,0.0638,0.3837,0.2422,0.5000,0.21
13,0.3265,0.1969,0.8202,0.3000,1.4547,1.4613,2.0855,3.3961];
% r*.
```

```
letrarrV=[9.7143,5.9487,4.7001,1.5916,0.4223,1.2925,1.3898,0.5712,0.3483,0.8383,0
.8238,0.1955,0.5327,0.7905,0.5741,0.1905,0.5741,0.7905,0.5327,0.1955,0.8238,0.838
3,0.3483,0.5712,1.3898,1.2925,0.4223,1.5916,4.7001,5.9487];
letrarrH=[13.7826,3.0566,2.3598,1.6439,1.1348,0.7118,0.2877,0.1771,0.3226,0.3461,
0.3135,0.2528,0.1905,0.2148,0.2785,0.3043,0.2785,0.2148,0.1905,0.2528,0.3135,0.34
61,0.3226,0.1771,0.2877,0.7118,1.1348,1.6439,2.3598,3.0566];
% o*.
letraooV=[15.0000,2.6107,5.8419,1.3239,1.2940,0.5628,0.0770,0.7345,0.4344,0.4526,
0.1714,0.2849,0.3755,0.1856,0.2295,0.3714,0.2295,0.1856,0.3755,0.2849,0.1714,0.45
26,0.4344,0.7345,0.0770,0.5628,1.2940,1.3239,5.8419,2.6107];
letraooH=[19.0526,1.4539,1.3291,1.2477,1.0640,0.1053,0.2134,0.3406,0.1438,0.2527,
0.2105,0.3497,0.3182,0.1811,0.2046,0.5263,0.2046,0.1811,0.3182,0.3497,0.2105,0.25
27,0.1438,0.3406,0.2134,0.1053,1.0640,1.2477,1.3291,1.4539];
% v*.
letravvV=[18.1053,5.8755,4.8796,0.6154,0.2090,1.1898,0.1048,0.1875,0.2815,0.0288,
0.2294,0.2361,0.3025,0.1023,0.1737,0.2105,0.1737,0.1023,0.3025,0.2361,0.2294,0.02
88,0.2815,0.1875,0.1048,1.1898,0.2090,0.6154,4.8796,5.8755];
letravvH=[25.6000,0.7380,0.7828,0.5090,1.1463,0.3606,0.6520,0.5481,0.5756,0.6090,
0.2646,0.6545,0.2120,0.2156,0.3493,0.2000,0.3493,0.2156,0.2120,0.6545,0.2646,0.60
90,0.5756,0.5481,0.6520,0.3606,1.1463,0.5090,0.7828,0.7380];
% t*.
letrattV=[12.5610,8.0390,1.9612,0.2305,1.4502,0.7248,0.6137,0.9238,0.3529,0.6091,
0.5349,0.0712,0.4950,0.4312,0.2366,0.4146,0.2366,0.4312,0.4950,0.0712,0.5349,0.60
91,0.3529,0.9238,0.6137,0.7248,1.4502,0.2305,1.9612,8.0390];
letrattH=[16.2609,2.7974,2.1908,1.0156,1.3111,0.5014,0.3006,0.2126,0.5331,0.3118,
0.1568,0.2560,0.4331,0.1611,0.1593,0.1739,0.1593,0.1611,0.4331,0.2560,0.1568,0.31
18,0.5331,0.2126,0.3006,0.5014,1.3111,1.0156,2.1908,2.7974];
% y*.
letrayyV=[15.0769,6.2470,2.8753,0.4226,0.4260,0.5000,0.1691,0.5457,0.0534,0.0576,
0.2692,0.0507,0.0737,0.0508,0.3700,0.0769,0.3700,0.0508,0.0737,0.0507,0.2692,0.05
76,0.0534,0.5457,0.1691,0.5000,0.4260,0.4226,2.8753,6.2470];
letrayyH=[24.1538,2.6808,2.0950,0.9979,0.6350,1.0406,0.3196,0.8818,0.4591,0.5894,
0.4679,0.8731,0.5946,0.1146,0.6319,0.4615,0.6319,0.1146,0.5946,0.8731,0.4679,0.58
94,0.4591,0.8818,0.3196,1.0406,0.6350,0.9979,2.0950,2.6808];
% h*.
letrahhV=[9.5238,2.3479,4.6314,4.5339,0.7228,1.2963,0.4636,0.3872,1.2447,0.9556,0
.5040,0.8275,0.3990,0.5682,0.7749,0.1429,0.7749,0.5682,0.3990,0.8275,0.5040,0.955
6,1.2447,0.3872,0.4636,1.2963,0.7228,4.5339,4.6314,2.3479];
letrahhH=[13.0909,3.4045,2.6361,1.2819,1.3948,0.3182,0.5455,0.1617,0.5661,0.0575,
0.3608,0.0888,0.2837,0.3033,0.4477,0.1818,0.4477,0.3033,0.2837,0.0888,0.3608,0.05
75,0.5661,0.1617,0.5455,0.3182,1.3948,1.2819,2.6361,3.4045];
% nj*.
letranjV=[10.5676,1.5047,1.9456,5.4001,2.4257,1.7074,1.2594,0.6695,0.2913,0.5325,
0.7611,1.1057,1.3676,0.2594,0.0987,0.1351,0.0987,0.2594,1.3676,1.1057,0.7611,0.53
25,0.2913,0.6695,1.2594,1.7074,2.4257,5.4001,1.9456,1.5047];
letranjH=[16.1053,4.2824,3.9331,1.6994,1.0768,1.0644,0.4914,0.1515,0.5490,0.8412,
0.4178,0.1592,0.4237,0.3254,0.4082,0.1053,0.4082,0.3254,0.4237,0.1592,0.4178,0.84
12,0.5490,0.1515,0.4914,1.0644,1.0768,1.6994,3.9331,4.2824];
% l*.
letrallV=[21.1429,7.1021,4.3475,1.6650,0.4705,1.4846,1.5886,0.5819,0.5240,1.1542,
0.8571,0.2095,0.5410,0.9241,0.7465,0,0.7465,0.9241,0.5410,0.2095,0.8571,1.1542,0.
5240,0.5819,1.5886,1.4846,0.4705,1.6650,4.3475,7.1021];
letrallH=[28.7273,1.2670,1.2501,1.2221,1.1837,1.1355,1.0784,1.0138,0.9432,0.8685,
0.7925,0.7185,0.6510,0.5956,0.5585,0.5455,0.5585,0.5956,0.6510,0.7185,0.7925,0.86
85,0.9432,1.0138,1.0784,1.1355,1.1837,1.2221,1.2501,1.2670];
% b*.
letrabbV=[12.6829,2.3956,5.4933,2.4526,1.0525,0.4232,0.5000,1.2447,0.9537,0.4909,
0.1707,0.3376,0.7536,0.8221,0.2816,0.0976,0.2816,0.8221,0.7536,0.3376,0.1707,0.49
09,0.9537,1.2447,0.5000,0.4232,1.0525,2.4526,5.4933,2.3956];
```

```
letrabbH=[16.3333,3.9346,2.2304,0.9568,1.1894,0.2974,0.2675,0.6563,0.4594,0.2984,
0.1717,0.5737,0.1610,0.4732,0.3241,0.1429,0.3241,0.4732,0.1610,0.5737,0.1717,0.29
84,0.4594,0.6563,0.2675,0.2974,1.1894,0.9568,2.2304,3.9346];
% ia*.
letraiaV=[12.4146,1.2168,3.4395,4.4157,1.4713,1.8231,0.3869,0.6216,0.6416,0.4365,
0.4634,0.5124,0.3041,0.1858,0.3986,0.7561,0.3986,0.1858,0.3041,0.5124,0.4634,0.43
65,0.6416,0.6216,0.3869,1.8231,1.4713,4.4157,3.4395,1.2168];
letraiaH=[15.4762,3.9631,3.5151,2.2595,0.8347,0.0952,0.2101,0.7152,0.3995,0.1213,
0.4762,0.7816,0.1545,0.3355,0.4722,0.3333,0.4722,0.3355,0.1545,0.7816,0.4762,0.12
13,0.3995,0.7152,0.2101,0.0952,0.8347,2.2595,3.5151,3.9631];
% ró*.
letrarorV=[13.2500,1.4844,3.3601,4.8668,1.3931,0.4375,2.2933,0.3749,0.5972,0.5404
,0.5962,0.6862,0.8957,1.0326,0.2030,0.0625,0.2030,1.0326,0.8957,0.6862,0.5962,0.5
404,0.5972,0.3749,2.2933,0.4375,1.3931,4.8668,3.3601,1.4844];
letrarorH=[14.2500,4.1402,2.8317,1.3897,1.3884,0.1803,0.1809,0.8644,0.5572,0.1903
,0.2598,0.3047,0.0691,0.3687,0.3178,0.0500,0.3178,0.3687,0.0691,0.3047,0.2598,0.1
903,0.5572,0.8644,0.1809,0.1803,1.3884,1.3897,2.8317,4.1402];
% ó*.
letraoooV=[15.8929,3.9411,4.5067,1.9596,1.3821,0.4475,0.2093,0.7083,0.5648,0.6097
,0.2789,0.1419,0.2894,0.1208,0.2343,0.3929,0.2343,0.1208,0.2894,0.1419,0.2789,0.6
097,0.5648,0.7083,0.2093,0.4475,1.3821,1.9596,4.5067,3.9411];
letraoooH=[15.3684,3.3482,2.0735,1.2069,1.1275,0.4497,0.2209,0.5569,0.4385,0.2744
,0.2294,0.5685,0.2064,0.5034,0.5276,0.1053,0.5276,0.5034,0.2064,0.5685,0.2294,0.2
744,0.4385,0.5569,0.2209,0.4497,1.1275,1.2069,2.0735,3.3482];
% fe*.
letrafeV=[10.9024,3.1736,2.9007,2.2994,2.2246,1.7413,0.4852,0.7529,0.6800,0.1087,
0.1267,0.3470,0.3554,0.6533,0.5786,0.8049,0.5786,0.6533,0.3554,0.3470,0.1267,0.10
87,0.6800,0.7529,0.4852,1.7413,2.2246,2.2994,2.9007,3.1736];
letrafeH=[17.2105,3.5477,3.4092,2.9957,1.2921,0.4111,0.1557,0.7665,0.5348,0.5723,
0.3287,0.7382,0.4972,0.1781,0.6539,0.3684,0.6539,0.1781,0.4972,0.7382,0.3287,0.57
23,0.5348,0.7665,0.1557,0.4111,1.2921,2.9957,3.4092,3.5477];
% fo*.
letrafoV=[10.2683,2.6042,3.0698,2.6199,2.2254,2.0402,0.2814,0.7892,0.7386,0.2657,
0.0645,0.1860,0.5139,0.7141,0.6360,0.7561,0.6360,0.7141,0.5139,0.1860,0.0645,0.26
57,0.7386,0.7892,0.2814,2.0402,2.2254,2.6199,3.0698,2.6042];
letrafoH=[15.3684,3.4096,2.6415,1.6506,1.3872,0.3795,0.2689,0.5389,0.3312,0.3508,
0.1053,0.6397,0.4140,0.3053,0.2778,0.1053,0.2778,0.3053,0.4140,0.6397,0.1053,0.35
08,0.3312,0.5389,0.2689,0.3795,1.3872,1.6506,2.6415,3.4096];
% eet*.
letraeetV=[17.7407,5.7389,3.6752,1.4973,0.9580,0.1481,0.2893,0.5414,0.4961,0.2275
,0.1481,0.1328,0.2893,0.2569,0.1616,0.0370,0.1616,0.2569,0.2893,0.1328,0.1481,0.2
275,0.4961,0.5414,0.2893,0.1481,0.9580,1.4973,3.6752,5.7389];
letraeetH=[14.1667,3.9153,2.6513,2.9862,0.9196,0.8750,0.1378,0.7685,0.7769,0.6740
,0.3975,0.4961,0.3463,0.2862,0.7514,0.5000,0.7514,0.2862,0.3463,0.4961,0.3975,0.6
740,0.7769,0.7685,0.1378,0.8750,0.9196,2.9862,2.6513,3.9153];
% ro*.
letrarotV=[11.4524,1.7493,2.0838,4.9827,1.1607,1.1557,2.3641,1.1180,0.3475,0.5515
,0.2076,0.5144,0.4600,0.7602,0.9757,0.7381,0.9757,0.7602,0.4600,0.5144,0.2076,0.5
515,0.3475,1.1180,2.3641,1.1557,1.1607,4.9827,2.0838,1.7493];
letrarotH=[11.4524,1.7493,2.0838,4.9827,1.1607,1.1557,2.3641,1.1180,0.3475,0.5515
,0.2076,0.5144,0.4600,0.7602,0.9757,0.7381,0.9757,0.7602,0.4600,0.5144,0.2076,0.5
515,0.3475,1.1180,2.3641,1.1557,1.1607,4.9827,2.0838,1.7493];

% CÁLCULO DE LOS PARÁMETROS PARA COMPARACIÓN.
% Vector suma de diferencias para la obtención del mínimo.
AV=sum(abs(x-letraAV));AH=sum(abs(y-letraAH));aV=sum(abs(x-
letraaV));aH=sum(abs(y-letraaH));
BV=sum(abs(x-letraBV));BH=sum(abs(y-letraBH));bV=sum(abs(x-
letrabV));bH=sum(abs(y-letrabH));
```

```
CV=sum(abs(x-letraCV));CH=sum(abs(y-letraCH));cV=sum(abs(x-
letracV));cH=sum(abs(y-letracH));
DV=sum(abs(x-letraDV));DH=sum(abs(y-letraDH));dV=sum(abs(x-
letradV));dH=sum(abs(y-letradH));
EV=sum(abs(x-letraEV));EH=sum(abs(y-letraEH));eV=sum(abs(x-
letraeV));eH=sum(abs(y-letraeH));
FV=sum(abs(x-letraFV));FH=sum(abs(y-letraFH));fV=sum(abs(x-
letrafV));fH=sum(abs(y-letrafH));
GV=sum(abs(x-letraGV));GH=sum(abs(y-letraGH));gV=sum(abs(x-
letragV));gH=sum(abs(y-letragH));
HV=sum(abs(x-letraHV));HH=sum(abs(y-letraHH));hV=sum(abs(x-
letrahV));hH=sum(abs(y-letrahH));
IV=sum(abs(x-letraIV));IH=sum(abs(y-letraIH));iV=sum(abs(x-
letraiV));iH=sum(abs(y-letraiH));
JV=sum(abs(x-letraJV));JH=sum(abs(y-letraJH));jV=sum(abs(x-
letrajV));jH=sum(abs(y-letrajH));
KV=sum(abs(x-letraKV));KH=sum(abs(y-letraKH));kV=sum(abs(x-
letrakV));kH=sum(abs(y-letrakH));
LV=sum(abs(x-letraLV));LH=sum(abs(y-letraLH));lV=sum(abs(x-
letralV));lH=sum(abs(y-letralH));
MV=sum(abs(x-letraMV));MH=sum(abs(y-letraMH));mV=sum(abs(x-
letramV));mH=sum(abs(y-letramH));
NV=sum(abs(x-letraNV));NH=sum(abs(y-letraNH));nV=sum(abs(x-
letranV));nH=sum(abs(y-letranH));
OV=sum(abs(x-letraOV));OH=sum(abs(y-letraOH));oV=sum(abs(x-
letraoV));oH=sum(abs(y-letraoH));
PV=sum(abs(x-letraPV));PH=sum(abs(y-letraPH));pV=sum(abs(x-
letrapV));pH=sum(abs(y-letrapH));
QV=sum(abs(x-letraQV));QH=sum(abs(y-letraQH));qV=sum(abs(x-
letraqV));qH=sum(abs(y-letraqH));
RV=sum(abs(x-letraRV));RH=sum(abs(y-letraRH));rV=sum(abs(x-
letrarV));rH=sum(abs(y-letrarH));
SV=sum(abs(x-letraSV));SH=sum(abs(y-letraSH));sV=sum(abs(x-
letrasV));sH=sum(abs(y-letrasH));
TV=sum(abs(x-letraTV));TH=sum(abs(y-letraTH));tV=sum(abs(x-
letratV));tH=sum(abs(y-letratH));
UV=sum(abs(x-letraUV));UH=sum(abs(y-letraUH));uV=sum(abs(x-
letrauV));uH=sum(abs(y-letrauH));
VV=sum(abs(x-letraVV));VH=sum(abs(y-letraVH));vV=sum(abs(x-
letravV));vH=sum(abs(y-letravH));
WV=sum(abs(x-letraWV));WH=sum(abs(y-letraWH));wV=sum(abs(x-
letrawV));wH=sum(abs(y-letrawH));
XV=sum(abs(x-letraXV));XH=sum(abs(y-letraXH));xV=sum(abs(x-
letraxV));xH=sum(abs(y-letraxH));
YV=sum(abs(x-letraYV));YH=sum(abs(y-letraYH));yV=sum(abs(x-
letrayV));yH=sum(abs(y-letrayH));
ZV=sum(abs(x-letraZV));ZH=sum(abs(y-letraZH));zV=sum(abs(x-
letrazV));zH=sum(abs(y-letrazH));
ver1=sum(abs(x-Num1V));hor1=sum(abs(y-Num1H));
ver2=sum(abs(x-Num2V));hor2=sum(abs(y-Num2H));
ver3=sum(abs(x-Num3V));hor3=sum(abs(y-Num3H));
ver4=sum(abs(x-Num4V));hor4=sum(abs(y-Num4H));
ver5=sum(abs(x-Num5V));hor5=sum(abs(y-Num5H));
ver6=sum(abs(x-Num6V));hor6=sum(abs(y-Num6H));
ver7=sum(abs(x-Num7V));hor7=sum(abs(y-Num7H));
ver8=sum(abs(x-Num8V));hor8=sum(abs(y-Num8H));
ver9=sum(abs(x-Num9V));hor9=sum(abs(y-Num9H));
ver0=sum(abs(x-Num0V));hor0=sum(abs(y-Num0H));
ujV=sum(abs(x-letraujV));ujH=sum(abs(y-letraujH));
reV=sum(abs(x-letrareV));reH=sum(abs(y-letrareH));
```

```
ojV=sum(abs(x-letraojV));ojH=sum(abs(y-letraojH));
raV=sum(abs(x-letraraV));raH=sum(abs(y-letraraH));
roV=sum(abs(x-letraroV));roH=sum(abs(y-letraroH));
rtV=sum(abs(x-letrartV));rtH=sum(abs(y-letrartH));
ajV=sum(abs(x-letraajV));ajH=sum(abs(y-letraajH));
lcomaV=sum(abs(x-letracomaV));lcomaH=sum(abs(y-letracomaH));
lpuntoV=sum(abs(x-letrapuntoV));lpuntoH=sum(abs(y-letrapuntoH));
ddV=sum(abs(x-letraddV));ddH=sum(abs(y-letraddH));
rrV=sum(abs(x-letrarrV));rrH=sum(abs(y-letrarrH));
ooV=sum(abs(x-letraooV));ooH=sum(abs(y-letraooH));
vvV=sum(abs(x-letravvV));vvH=sum(abs(y-letravvH));
ttV=sum(abs(x-letrattV));ttH=sum(abs(y-letrattH));
yyV=sum(abs(x-letrayyV));yyH=sum(abs(y-letrayyH));
hhV=sum(abs(x-letrahhV));hhH=sum(abs(y-letrahhH));
njV=sum(abs(x-letranjV));njH=sum(abs(y-letranjH));
llV=sum(abs(x-letrallV));llH=sum(abs(y-letrallH));
bbV=sum(abs(x-letrabbV));bbH=sum(abs(y-letrabbH));
iaV=sum(abs(x-letraiaV));iaH=sum(abs(y-letraiaH));
rorV=sum(abs(x-letrarorV));rorH=sum(abs(y-letrarorH));
oooV=sum(abs(x-letraoooV));oooH=sum(abs(y-letraoooH));
eeV=sum(abs(x-letrafeV));eeH=sum(abs(y-letrafeH));
foV=sum(abs(x-letrafoV));foH=sum(abs(y-letrafoH));
eetV=sum(abs(x-letraeetV));eetH=sum(abs(y-letraeetH));
rotV=sum(abs(x-letrarotV));rotH=sum(abs(y-letrarotH));

vComp = [sum([AV AH]) sum([aV aH]) sum([BV BH]) sum([bV bH]) sum([CV CH]) sum([cV
cH]) sum([DV DH]) sum([dV dH]) sum([EV EH]) sum([eV eH]) sum([FV FH]) sum([fV
fH])...
    sum([GV GH]) sum([gV gH]) sum([HV HH]) sum([hV hH]) sum([IV IH]) sum([iV iH])
sum([JV JH]) sum([jV jH]) sum([KV KH]) sum([kV kH]) sum([LV LH]) sum([lV lH])...
    sum([MV MH]) sum([mV mH]) sum([NV NH]) sum([nV nH]) sum([OV OH]) sum([oV oH])
sum([PV PH]) sum([pV pH]) sum([QV QH]) sum([qV qH]) sum([RV RH]) sum([rV rH])...
    sum([SV SH]) sum([sV sH]) sum([TV TH]) sum([tV tH]) sum([UV UH]) sum([uV uH])
sum([VV VH]) sum([vV vH]) sum([WV WH]) sum([wV wH]) sum([XV XH]) sum([xV xH])...
    sum([YV YH]) sum([yV yH]) sum([ZV ZH]) sum([zV zH]) sum([ver1 hor1])
sum([ver2 hor2]) sum([ver3 hor3]) sum([ver4 hor4]) sum([ver5 hor5])...
    sum([ver6 hor6]) sum([ver7 hor7]) sum([ver8 hor8]) sum([ver9 hor9]) sum([ver0
hor0]) sum([ujV ujH]) sum([reV reH]) sum([ojV ojH]) sum([raV raH]) sum([roV
roH])...
    sum([rtV rtH]) sum([ajV ajH]) sum([lcomaV lcomaH]) sum([lpuntoV lpuntoH])
sum([ddV ddH]) sum([rrV rrH]) sum([ooV ooH]) sum([vvV vvH]) sum([ttV ttH])
sum([yyV yyH])...
    sum([hhV hhH]) sum([njV njH]) sum([llV llH]) sum([bbV bbH]) sum([iaV iaH])
sum([rorV rorH]) sum([oooV oooH]) sum([eeV eeH]) sum([foV foH]) sum([eetV eetH])
sum([rotV rotH])];

% Buscamos el mínimo para la decisión del símbolo.
[minimo,pos] = min(vComp);

% SELECCIÓN
    switch pos
        case 1
            y = ' A ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
```

```
fopen(puerto);
h=' ';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 2
            y = ' a ';
           % x1 = [1 0 0 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=' ';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 3
            y = ' B ';
            %x1 = [0 1 0 0 0 1 1 0 1 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='(';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 4
            y = ' b ';
            %x1 = [1 0 1 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
```

```
fopen(puerto);
h='(';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 5
            y = ' C ';
            %x1 = [0 1 0 0 0 1 1 1 0 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='0';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 6
            y = ' c ';
            %x1 = [1 1 0 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='0';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 7
            y = ' D ';
            %x1 = [0 1 0 0 0 1 1 1 0 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='4';
h1='Q';
```

```
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 8
            y = ' d ';
            %x1 = [1 1 0 1 0 0]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='4';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 9
            y = ' E ';
            %x1 = [0 1 0 0 0 1 1 0 0 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='$';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 10
            y = ' e ';
           % x1 = [1 0 0 1 0 0]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='$';
fwrite(puerto,h,'uint8');
```

```
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 11
            y = ' F ';
            %x1 = [0 1 0 0 0 1 1 1 1 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='8';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 12
            y = ' f ';
            %x1 = [1 1 1 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='8';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 13
            y = ' G ';
            %x1 = [0 1 0 0 0 1 1 1 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='<';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
```

```
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 14
            y = ' g ';
            %x1 = [1 1 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='<';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 15
            y = ' H ';
            %x1 = [0 1 0 0 0 1 1 0 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=',';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 16
            y = ' h ';
            %x1 = [1 0 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=',';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
```

```
clear puerto;
disp('STOP');
        case 17
            y = ' I ';
            %x1 = [0 1 0 0 0 1 0 1 1 0 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='X';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 18
            y = ' i ';
            %x1 = [0 1 1 0 0 0]
        clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='X';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 19
            y = ' J ';
            %x1 = [0 1 0 0 0 1 0 1 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
```

```
clear puerto;
disp('STOP');
        case 20
            y = ' j ';
            %x1 = [0 1 1 1 0 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 21
            y = ' K ';
            %x1 = [0 1 0 0 0 1 1 0 0 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='"';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 22
            y = ' k ';
            %x1 = [1 0 0 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='"';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 23
```

```
            y = ' L ';
            %x1 = [0 1 0 0 0 1 1 0 1 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='*';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 24
            y = ' l ';
            %x1 = [1 0 1 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='*';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 25
            y = ' M ';
            %x1 = [0 1 0 0 0 1 1 1 0 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='2';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 26
```

```
            y = ' m ';
            %x1 = [1 1 0 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='2';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 27
            y = ' N ';
            %x1 = [0 1 0 0 0 1 1 1 0 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='6';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 28
            y = ' n ';
            %x1 = [1 1 0 1 1 0]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='6';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 29
            y = ' O ';
            %x1 = [0 1 0 0 0 1 1 0 0 1 1 0];
clc; disp('BEGIN');
```

```
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 30
            y = ' o ';
            %x1 = [1 0 0 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 31
            y = ' P ';
            %x1 = [0 1 0 0 0 1 1 1 1 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=':';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 32
            y = ' p ';
            %x1 = [1 1 1 0 1 0];
clc; disp('BEGIN');
```

```
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=':';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 33
            y = ' Q ';
            %x1 = [0 1 0 0 0 1 1 1 1 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='>';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 34
            y = ' q ';
            %x1 = [1 1 1 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='>';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 35
            y = ' R ';
            %x1 = [0 1 0 0 0 1 1 0 1 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
```

```
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='.';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 36
            y = ' r ';
            %x1 = [1 0 1 1 1 0]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='.';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 37
            y = ' S ';
            %x1 = [0 1 0 0 0 1 0 1 1 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='Z';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 38
            y = ' s ';
            %x1 = [0 1 1 0 1 0]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
```

```
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='Z';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 39
            y = ' T ';
            %x1 = [0 1 0 0 0 1 0 1 1 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='^';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 40
            y = ' t ';
            %x1 = [0 1 1 1 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='^';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 41
            y = ' U ';
            %x1 = [0 1 0 0 0 1 1 0 0 0 1 1]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
```

```
fopen(puerto);
h='#';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 42
            y = ' u ';
            %x1 = [1 0 0 0 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='#';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 43
            y = ' V ';
            %x1 = [0 1 0 0 0 1 1 0 1 0 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='+';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 44
            y = ' v ';
            %x1 = [1 0 1 0 1 1]
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
```

```
fopen(puerto);
h='+';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 45
            y = ' W ';
            %x1 = [0 1 0 0 0 1 0 1 1 1 0 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=']';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 46
            y = ' w ';
            %x1 = [0 1 1 1 0 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=']';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 47
            y = ' X ';
            %x1 = [0 1 0 0 0 1 1 1 0 0 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='3';
h1='Q';
```

```
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 48
            y = ' x ';
            %x1 = [1 1 0 0 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='3';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 49
            y = ' Y ';
            %x1 = [0 1 0 0 0 1 1 1 0 1 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='7';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 50
            y = ' y ';
            %x1 = [1 1 0 1 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='7';
fwrite(puerto,h,'uint8');
```

```
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 51
            y = ' Z ';
            %x1 = [0 1 0 0 0 1 1 0 0 1 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='g';
h1='Q';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 52
            y = ' z ';
            %x1 = [1 0 0 1 1 1];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='g';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 53
            y = ' 1 ';
            %x1 = [0 1 0 1 1 1 1 0 0 0 0 0];
        case 54
            y = ' 2 ';
            %x1 = [0 1 0 1 1 1 1 0 1 0 0 0];
        case 55
            y = ' 3 ';
            %x1 = [0 1 0 1 1 1 1 1 0 0 0 0];
        case 56
            y = ' 4 ';
            %x1 = [0 1 0 1 1 1 1 1 0 1 0 0];
        case 57
            y = ' 5 ';
            %x1 = [0 1 0 1 1 1 1 0 0 1 0 0];
        case 58
```

```
            y = ' 6 ';
            %x1 = [0 1 0 1 1 1 1 1 1 0 0 0];
        case 59
            y = ' 7 ';
            %x1 = [0 1 0 1 1 1 1 1 1 1 0 0];
        case 60
            y = ' 8 ';
            %x1 = [0 1 0 1 1 1 1 0 1 1 0 0];
        case 61
            y = ' 9 ';
            %x1 = [0 1 0 1 1 1 0 1 1 0 0 0];
        case 62
            y = ' 0 ';
            %x1 = [0 1 0 1 1 1 0 1 1 1 0 0];
        case 63
            y = ' u  j ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
h1='#';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 64
            y = ' r  e ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='$';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 65
            y = ' o  j ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
```

```
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
h1='&';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 66
            y = ' r  a ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=' ';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 67
            y = ' r  o ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 68
            y = ' r  t ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
```

```
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='^';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 69
            y = ' a  j ';
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
h1=' ';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 70
            y = ' , ';
            %x1 = [0 0 1 0 0 0];
            clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='H';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 71
            y = ' . ';
            %x1 = [0 0 0 0 1 0];
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
```

```
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='B';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 72
            y = ' d ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='4';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 73
            y = ' r ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='.';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 74
            y = ' o ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
```

```
disp('STOP');
        case 75
            y = ' v ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='+';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 76
            y = ' t ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='^';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 77
            y = ' y ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='7';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 78
            y = ' h ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
```

```
set(puerto,'FlowControl','none');
fopen(puerto);
h=',';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 79
            y = ' n  j ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='\';
h1='6';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 80
            y = ' l ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='*';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 81
            y = ' b ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='(';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
```

```
delete(puerto);
clear puerto;
disp('STOP');
        case 82
            y = ' í  a ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h=' ';
h1='R';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 83
            y = ' r  ó ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='S';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 84
            y = ' ó ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='S';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
```

```
        case 85
            y = ' f  e ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='$';
h1='8';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 86
            y = ' f  o ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
h1='8';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 87
            y = ' é ';
clc; disp('BEGIN');
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='[';
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
        case 88
            y = ' r  o ';
clc; disp('BEGIN');
```

```
puerto=serial('COM10');
set(puerto,'BaudRate',9600);
set(puerto,'DataBits',8);
set(puerto,'Parity','none');
set(puerto,'StopBits',1);
set(puerto,'FlowControl','none');
fopen(puerto);
h='&';
h1='.';
fwrite(puerto,h1,'uint8');
pause(2.7)
fwrite(puerto,h,'uint8');
pause(2.7)
fclose(puerto);
delete(puerto);
clear puerto;
disp('STOP');
    end
    fid = fopen('Resultado.txt','a');
    fprintf(fid,'%c',y);
    fclose(fid);
```

9. Modulo JM60

```
#include "EFmJM60_StackConfig.h"
/*****************************************************************/
/*---------- Espacio para declaracion de constantes  ------------*/
/*****************************************************************/

/*****************************************************************/
/*------- Espacio para declaracion de variables globales  -------*/
/*****************************************************************/
UINT16 dato_usb;
/*****************************************************************/
/*****************************************************************/
/*-------------------- Espacio para funciones  ------------------*/
/*****************************************************************/
#include "usb.h"                          //Incluye librerias USB
#include "usb_cdc.h"                     //Incluye librerias modo CDC

/*------------------------------- PRINTF --------------------------------*/
extern void TERMIO_Init(void){cdc_init();}            //Inicializar Terminal
extern void TERMIO_PutChar(UINT8 ch){cdc_write(ch);}  //Terminal para Imprimir
/*****************************************************************/
/*****************************************************************/
/*------------------- Espacio de codigo principal ------------------*/
/*****************************************************************/
void main(void) {
SOPT1 = 0x20;                                    //Deshabilita el modulo COP
(WDT)
usb_init();                                    //incializa puerto USB

cdc_init();                                              //inicializa modo CDC
```

```
PTEDD=(0xFF);                        // Puerto D como salida
PTED=(0x00);

enable_interrupt(INT_GLOBAL);    //Habilita interrupcion

 for(;;){                                                  //bucle infinito
  cdc_process();                 // libreria CDC
  delay_ms(100);
  if(cdc_kbhit()!=0){            //Pregunta si llego dato por USB
    dato_usb=cdc_getch();        //Captura dato que llego
    cdc_write(dato_usb);         //Reenvia dato
    output_e(dato_usb);         //muestra en ledS PUERTO E, dato que llego
      }
    }
 //delay_ms(1500);
 //PTDD_PTDD0=0;
  }
```

ANEXO 4

CONFIGURACIÓN PREVIA DE LA CAMARA WEB

Para saber la posición de la cámara web adecuada se debe realizar una configuración:

- inicialmente buscar la información de los adaptadores disponibles.

```
 imaqhwinfo

ans =

    InstalledAdaptors: {'coreco'  'winvideo'}//Adaptadores

         MATLABVersion: '7.8 (R2009a)'

           ToolboxName: 'Image Acquisition Toolbox'

        ToolboxVersion: '3.3 (R2009a)'
```

El adaptador puede ser utilizado por varios dispositivos, para obtener la información del adaptador se puede utilizar la función anterior y se adiciona un argumento. En *'DeviceIDS' se observa la cantidad de cámaras disponibles.*

```
>>imaqhwinfo('winvideo')

AdaptorDllName:'C:\ProgramFiles\MATLAB\R2009a\toolbox\imaq\imaqadptors\win32\mwwinvideoimaq.dll'

  AdaptorDllVersion: '3.3 (R2009a)'

         AdaptorName: 'winvideo'

           DeviceIDs: {[1] [2]}     // Encontramos las  webcams disponibles en el ordenador.

          DeviceInfo: [1x1 struct]
```

En esta prueba se observan dos dispositivos, el análisis de esta prueba indica que se tiene disponible la cámara web interna del computador y la que se conecta en

formato USB. Al solicitar nuevamente información pero indicando el 'Device = 1' se observan los datos técnicos de la cámara.

```
>>infor=imaqhwinfo('winvideo',1)

        DefaultFormat: 'RGB24_640x480'

   DeviceFileSupported: 0

           DeviceName: 'Logitech HD Webcam C615'

             DeviceID: 1

     ObjectConstructor: 'videoinput('winvideo', 1)'

      SupportedFormats: {1x30 cell}

>> infor.SupportedFormats

'I420_1024x576'  'I420_1280x720'  'I420_1600x896'  'I420_160x120'  'I420_176x144'  'I420_1920x1080'
'I420_320x240'  'I420_352x288'  'I420_432x240'  'I420_640x360'  'I420_640x480'  'I420_800x448'
'I420_800x600'  'I420_864x480'  'I420_960x720'  'RGB24_1024x576'  'RGB24_1280x720'  'RGB24_1600x896'
'RGB24_160x120'  'RGB24_176x144'  'RGB24_1920x1080'  'RGB24_320x240'  'RGB24_352x288'
'RGB24_432x240'  'RGB24_640x360'  'RGB24_640x480'  'RGB24_800x448'  'RGB24_800x600'
'RGB24_864x480'  'RGB24_960x720'
```

- Una vez se conoce la ubicación de la cámara y los formatos permitidos, se habilita el puerto de video de matlab con la siguiente función;

```
obj = videoinput(adaptorname,deviceID,format)
```

Para tener el máximo de rendimiento de la webcam se selecciona el formato de mayor tamaño en pixeles 'RGB24_1920x1080' y la función "*videoinput"* se ejecuta de la siguiente forma:

```
videow=videoinput('winvideo',1,'RGB24_1920x1080');
```

La captura de la imagen se realiza con la función "*getnapshot"* donde se adquiere una imagen instantánea que puede ser hasta de 8 mega píxeles.

```
imagenw=getsnapshot(videow);
```

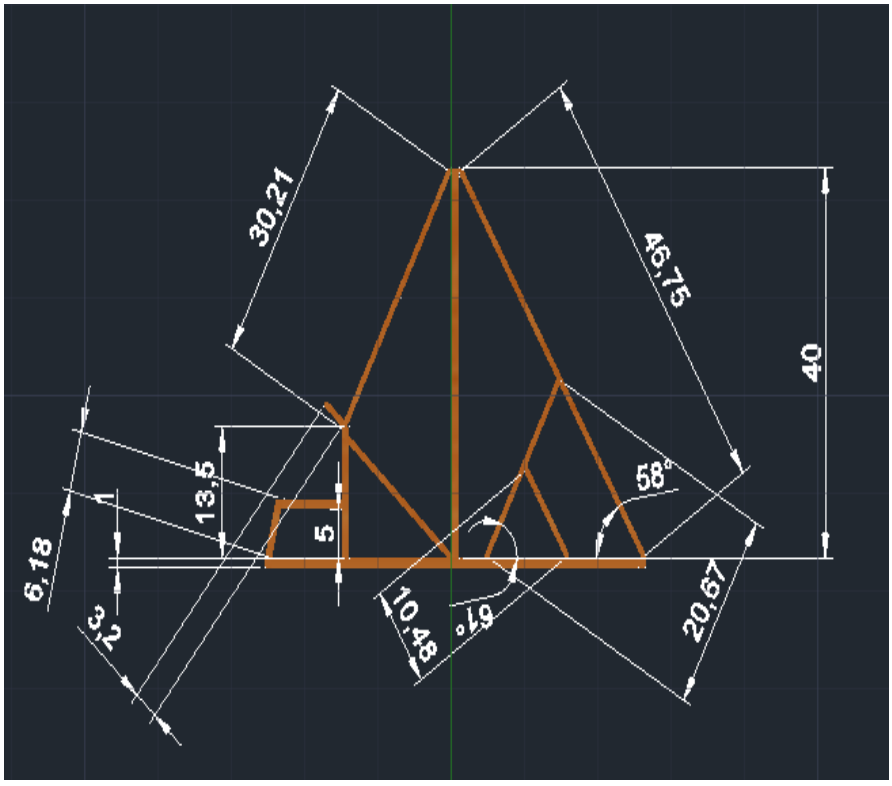
ANEXO 5

DESCRIPCIÓN DEL ARMAZÓN.

1. *DISEÑO DEL ARMAZÓN.*

El armazón es el soporte donde se ubicara la camara web y la hoja del documento que se quiere traducir, tambien se ubican los led's que en este proceso hacen parte de la simulacion del codigo. En una implementacion de este proyecto donde van los led's se ubicarian los elementos que hacen juego con los servomotores para la visualizacion de alto relieve del proceso.

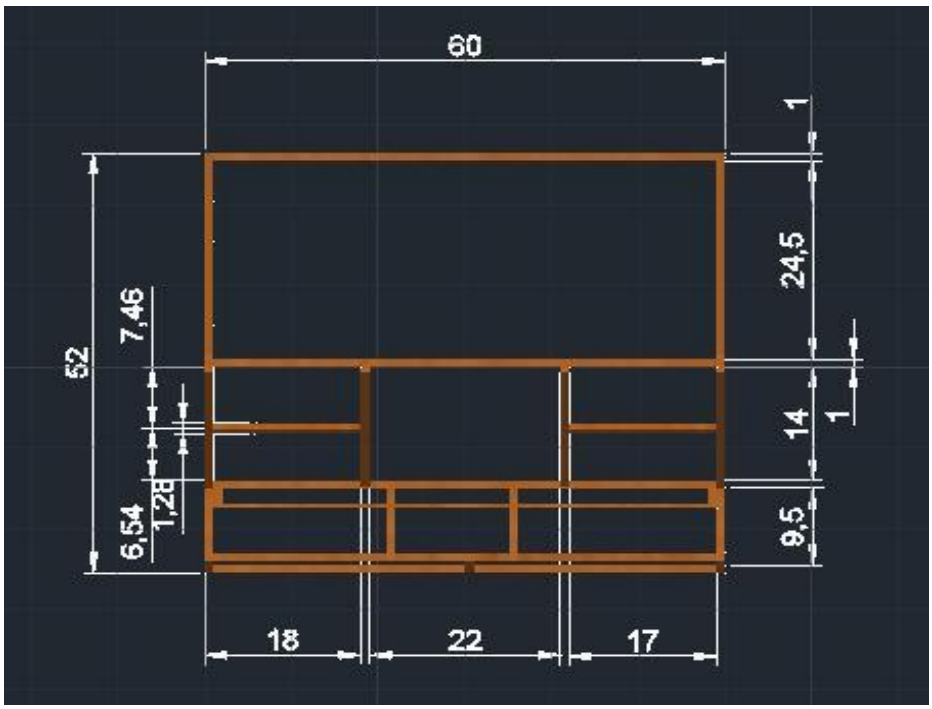
.Gráfica 1 Vista Derecha del diseño del armazón.



Fuente: Autores

En la Gráfica 1 se observa la vista derecha del armazón, donde se indican las dimensiones necesitadas para la caja. En la Gráfica 2 se indica la vista superior. La ubicación del soporte de longitud ‘20.67’ con una inclinación de 61° en la Gráfica 1 es donde estará ubicada la cámara web que es paralela al soporte de longitud ‘30.21’ donde va ubicada la hoja de texto que se quiere analizar. La Gráfica 2 indica las dimensiones faltantes que se visualizan en la vista superior.

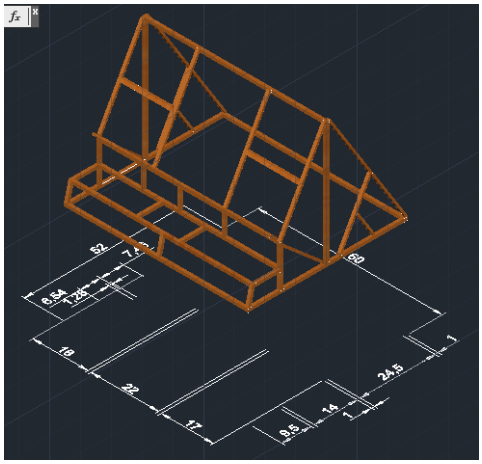
.Gráfica 2 Vista superior del diseño del armazón.



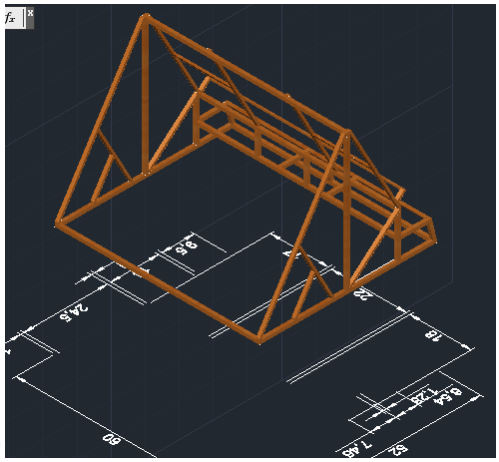
Fuente: Autores

Las vistas ‘Isométrica SE’ e ‘Isométrica NO’ muestran el diseño final del armazón realizado en ‘Autocad’ que es un programa de diseño asistido por computadora en dos y tres dimensiones. El armazón es el soporte que da la forma de la caja diseñada para esta aplicación. Es un diseño sencillo, en esta etapa de desarrollo e concluye que hace falta la integración de las facultades y carreras de la Universidad Industrial de Santander. Las recomendaciones de un diseñador industrial permitirían un diseño más elegante y ergonómico.

.Gráfica 3 (a) Vista Isométrica SE (b) Vista Isométrica NO.



(a)

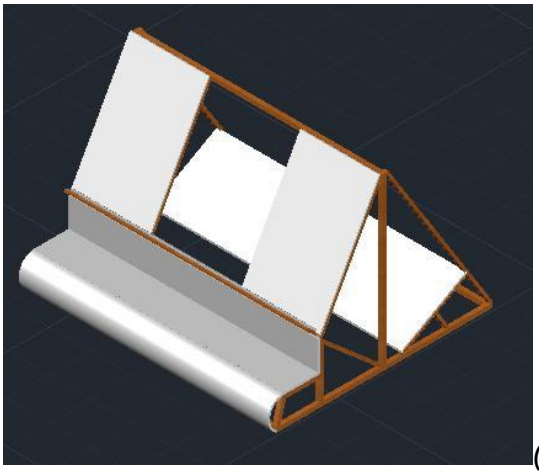


(b)

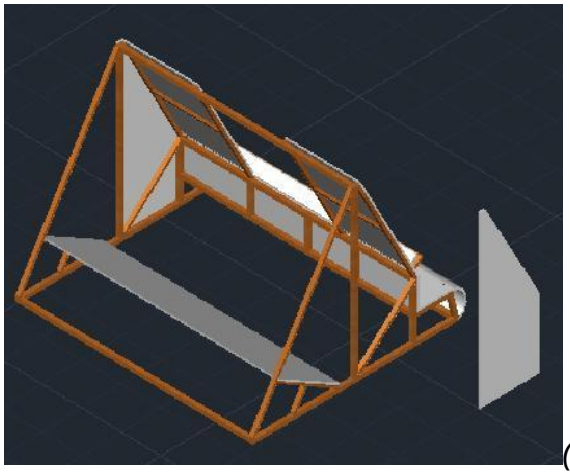
Fuente: Autores

Una vez diseñado el soporte, se comenzó a ubicar las tablas que son la carcasa exterior de la caja, en la Gráfica 4 se observa este proceso, cuando se ubican todas las tablas se procede a indicar el tipo de material para tener una visión más real de la caja. La parte en color azul indicada en la Gráfica 5 es un vidrio antirreflejo donde se ubica la hoja de texto.

.Gráfica 4 Proceso de Ubicación de carcasa.



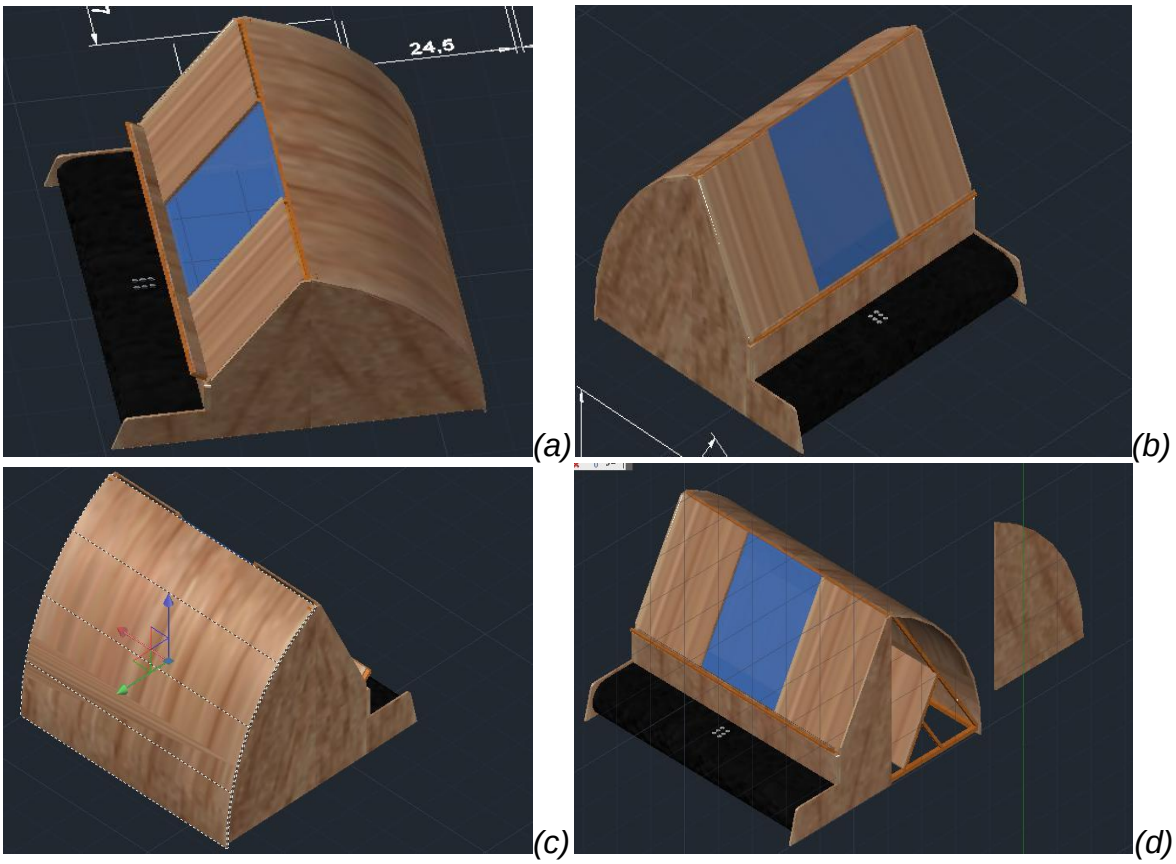
(a)



(b)

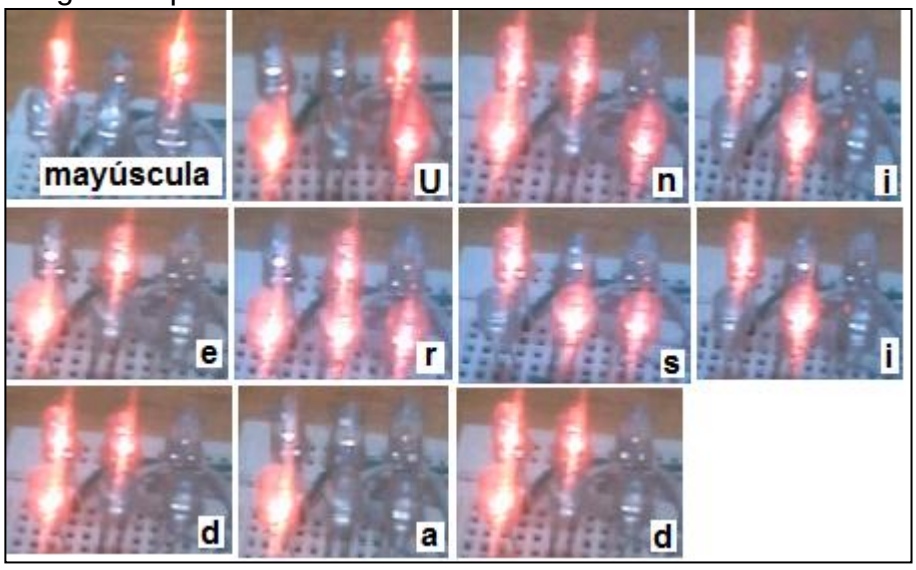
Fuente: Autores

.Gráfica 5 Carcasa para simulación del proceso



Fuente: Autores

Imagen del proceso de simulación.

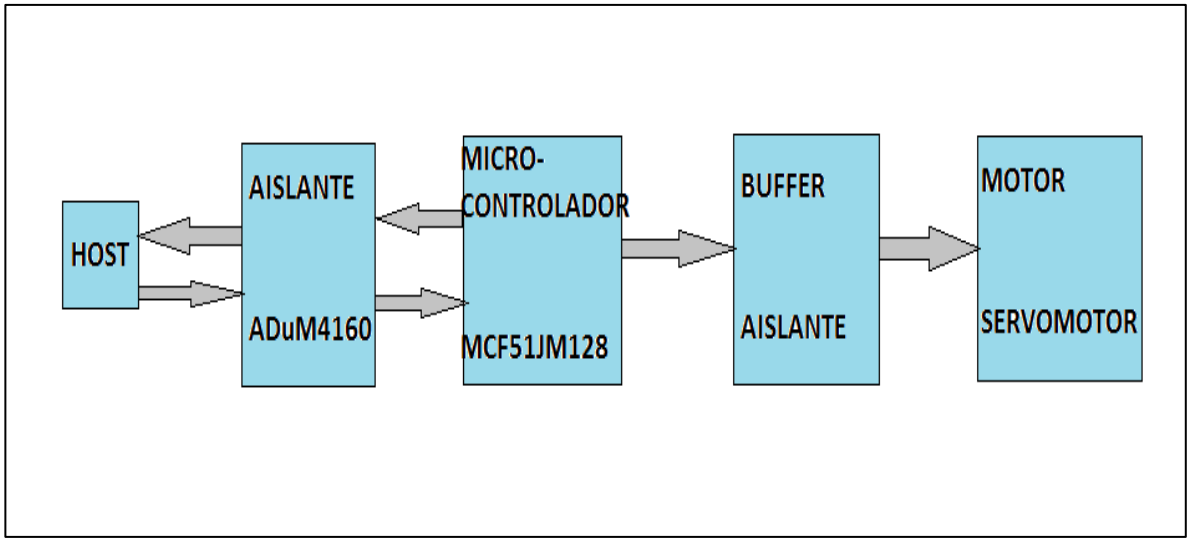


ANEXO 6

Diseño dispositivo de salida con servomotores.

Este ítem se trata temas relacionados con la salida del traductor, el diseño de la salida en forma general para un arreglo de seis servomotores, el diseño consta de una comunicación serial USB, una etapa de aislamiento, una etapa de procesamiento y una etapa de potencia. La etapa de aislamiento se propone entre el PC y el microcontrolador, ya que el dispositivo en general va a ser manipulado por el usuario.

Figura 86 Diagrama de bloques general de un sistema de control en lazo directo.



Fuente: Autores

- Microcontroladores series MCF51JM128

El micro-controladores de la serie MCF51JM128 tiene un sistema-on –chip(SoCs) basado en coldfire core V1 y cuenta con:

- El procesador puede operar a una frecuencia máxima de 50.33 MHz(los periféricos operan a la mitad de la frecuencia)

- Integrando las tecnologías más importantes en las aplicaciones industriales tales como USB on-the-go, controlador de red de área y aceleración criptográfica.
- El micro-controlador es basado en los diseños del micro-controlador series MC9S08JM60.

Comunicación USB

El USB es un bus de cable que permite el intercambio de datos entre un ordenador y una amplia gama de periféricos accesibles al mismo tiempo. Los periféricos conectados comparten el ancho de banda a través de un protocolo USB host-scheduled, token-based. El bus periférico permite ser adjuntado, configurado, utilizado y separado, mientras que el host y otros periféricos se encuentran en operación.

El software USB proporciona una visión uniforme del sistema a todas las aplicaciones del software, ocultando los detalles de implementación que hacen la aplicación de este software más portable. Gestiona la dinámica de conexión y desconexión de periféricos.

Sólo hay un host en cualquier sistema USB. La interfaz USB a la computadora, host del sistema que se conoce como el controlador de host. Puede haber varios dispositivos USB en cualquier sistema, tales como joysticks, parlantes, impresoras, dispositivos USB, etc. Estos dispositivos presentan una interfaz USB estándar en términos de comprensión, respuesta y capacidad estándar.

El host inicia con operaciones a periféricos especiales mientras el dispositivo responde a las operaciones de control. El dispositivo envía y recibe datos desde y hacia el host, utilizando un protocolo estándar USB 2.0, los periféricos funciona a velocidad nominal y a baja velocidad 12 MHz y 1.5 MHz respectivamente.

USB (Bus Serial Universal) es la conexión estándar de periféricos y dispositivos portables más popular, tales como cámaras digitales, dispositivos de audio, memorias flash, aplicaciones especiales. La conexión ON-THE-GO (OTG) complementa a la conexión USB estándar extendiendo las especificaciones USB a aplicaciones Peer-to-Peer. Usando USB OTG en tecnología electrónica de consumo a periféricos y dispositivos se pueden comunicar entre si por ejemplo una cámara digital se puede conectar directamente a una impresora.

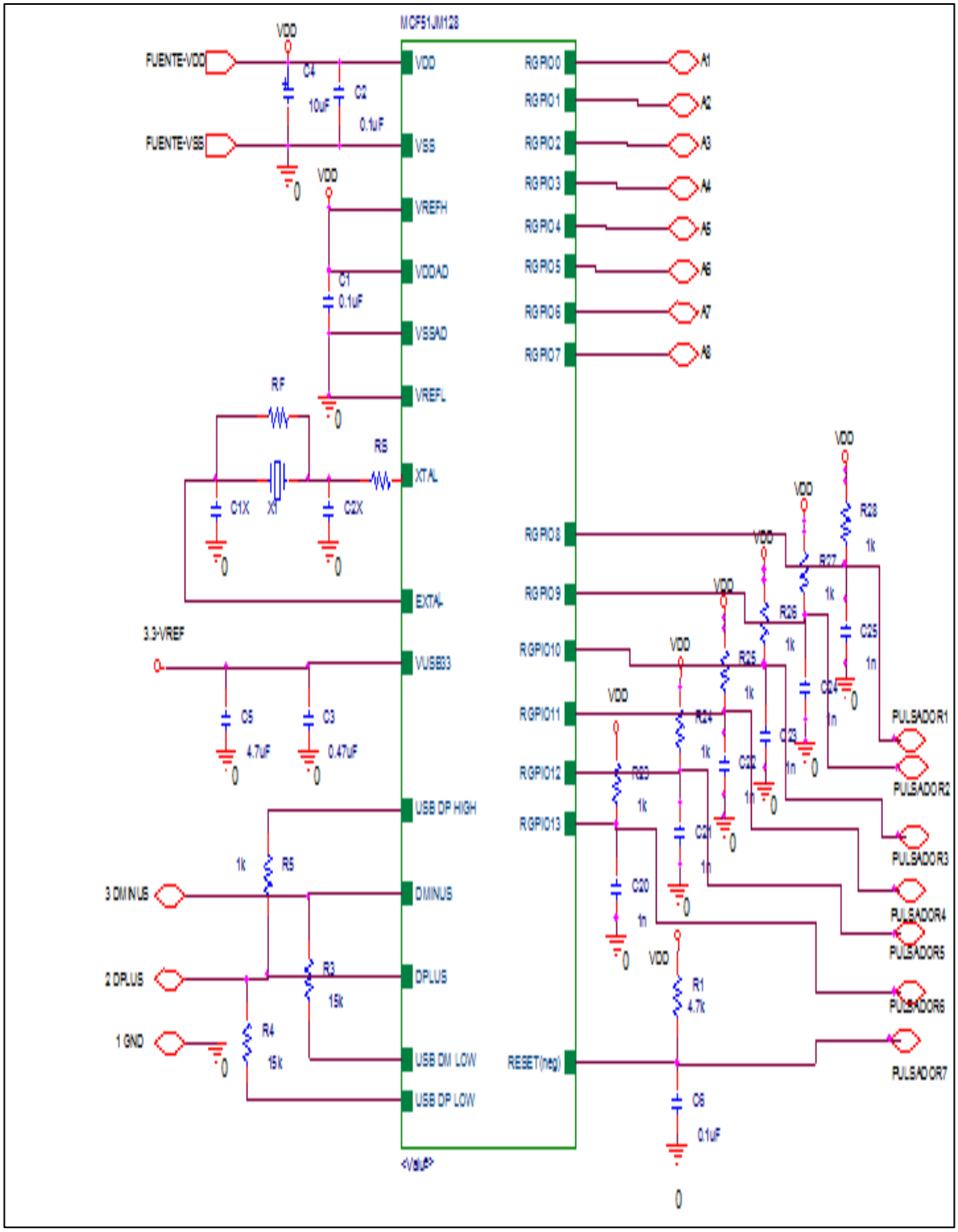
El micro-controlador MCF51JM128 cuenta con 16 bits RGPIO de propósito general puede ser utilizados como canales de entrada, canales de salida de comparadores o PWM.

En este caso el MCF52HM128 se programa con un algoritmo de control que tiene como entradas el puerto USB conectado en la forma OTG y pulsadores normalmente abiertos como se muestra en la Figura 87.

El algoritmo de control recibe datos en formato ASCCI provenientes del software, el algoritmo interpreta el código y trasmite la señal de control a la etapa de buffer del sistema de control general. Los interruptores funcionan como el tablero de instrumentos del sistema de control, el cual le da la información necesaria al algoritmo de control para realizar sus rutinas especificas.

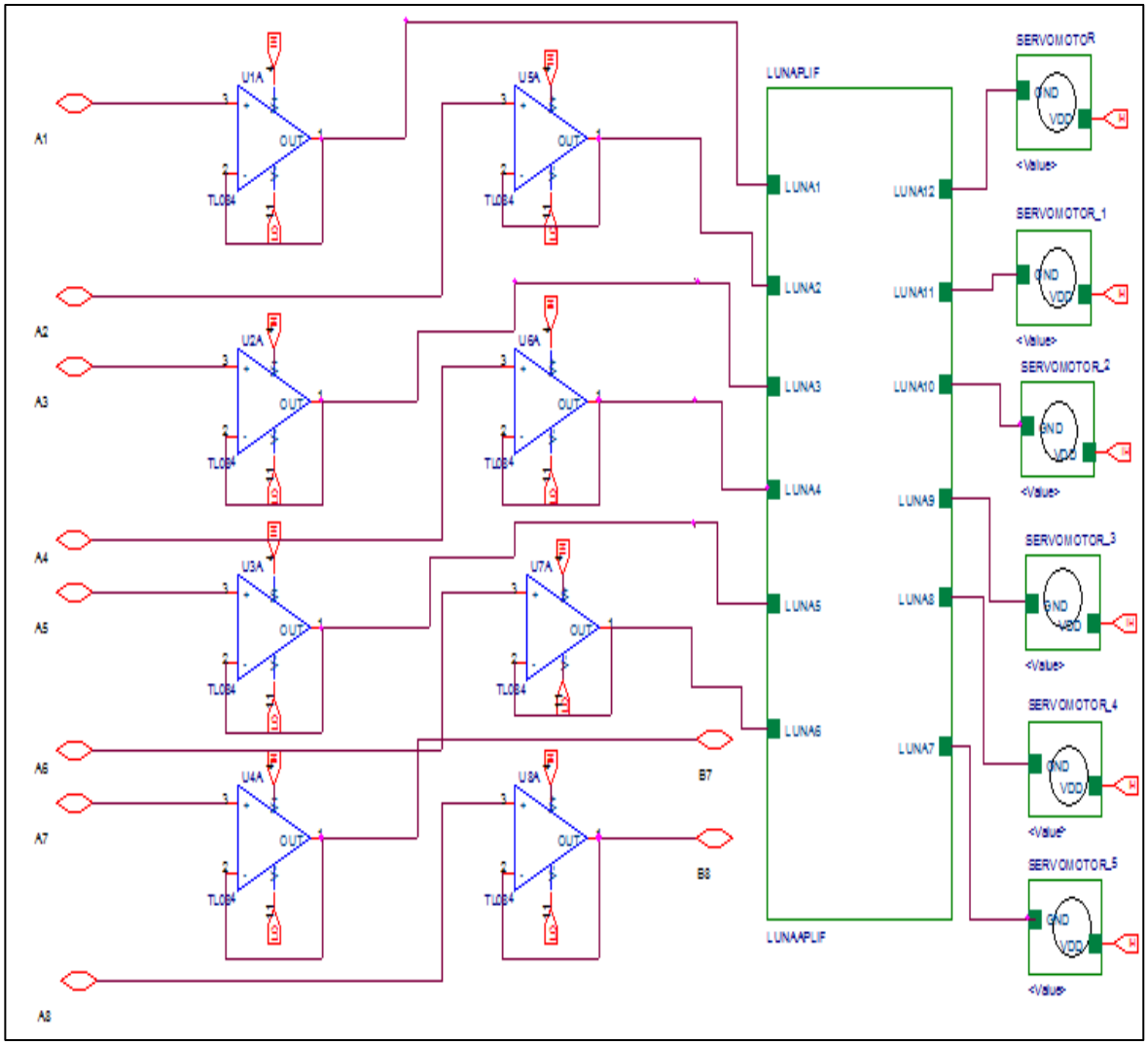
Es necesario contar con una frecuencia externa mínima de 12 MHz para que los periféricos USB funcionen correctamente, esta frecuencia se obtiene de un cristal de QUARZT como se muestra en la figura 87, la alimentación de micro-procesador es de 5 Volts DC y una tensión de referencia de 3.3 Volts DC.

Figura 87 Conexiones Básicas para microcontrolador.



Fuente: Autores

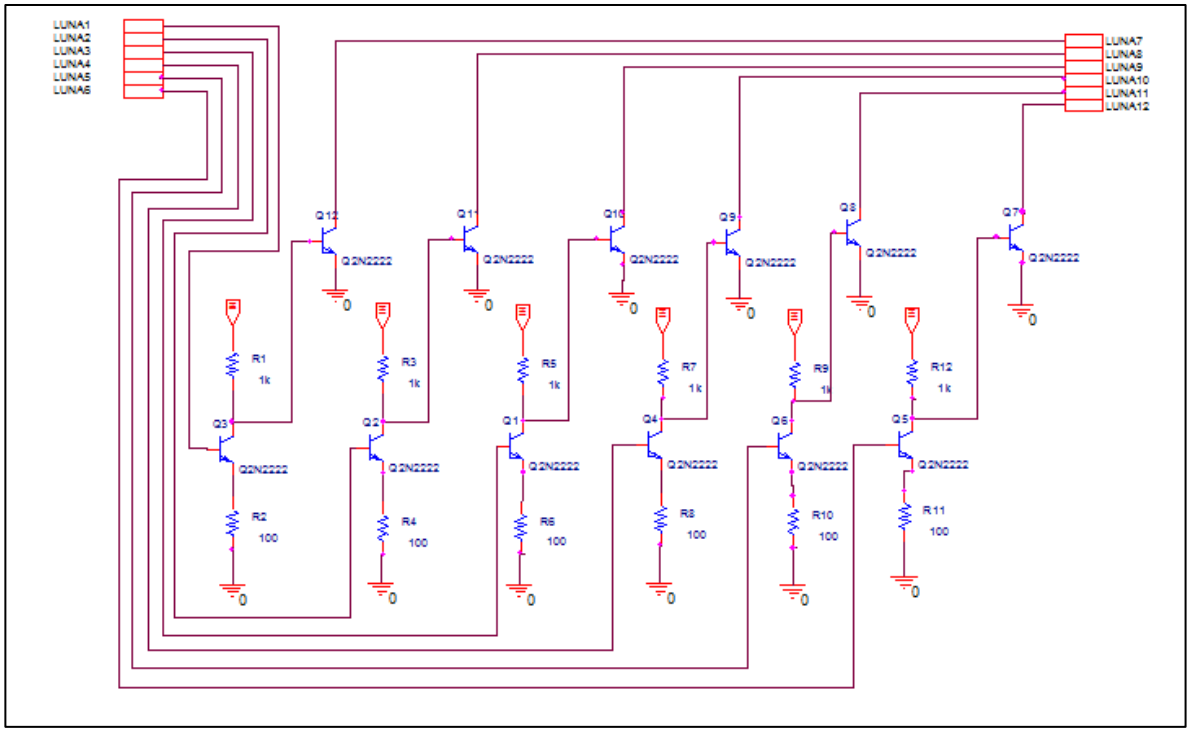
Figura 88 Buffer y Aplicación del lazo directo de control



Fuente: Autores

El diagrama de la Figura 88, tiene como entrada a la señal de control proveniente del micro-controlador, el buffer conformado por el amplificador operacional TL084 funciona como acople de alta impedancia con el fin de proteger las salidas del micro-controlador de altas corrientes, la salida de los operacionales llega a una etapa de amplificación y potencia de la señal, conformada por un arreglo de transistores 2N2222 como se muestra en la figura 89, finalmente la etapa de potencia se comunica con unos servomotores HS-55.

Figura 89 Amplificación y Potencia.

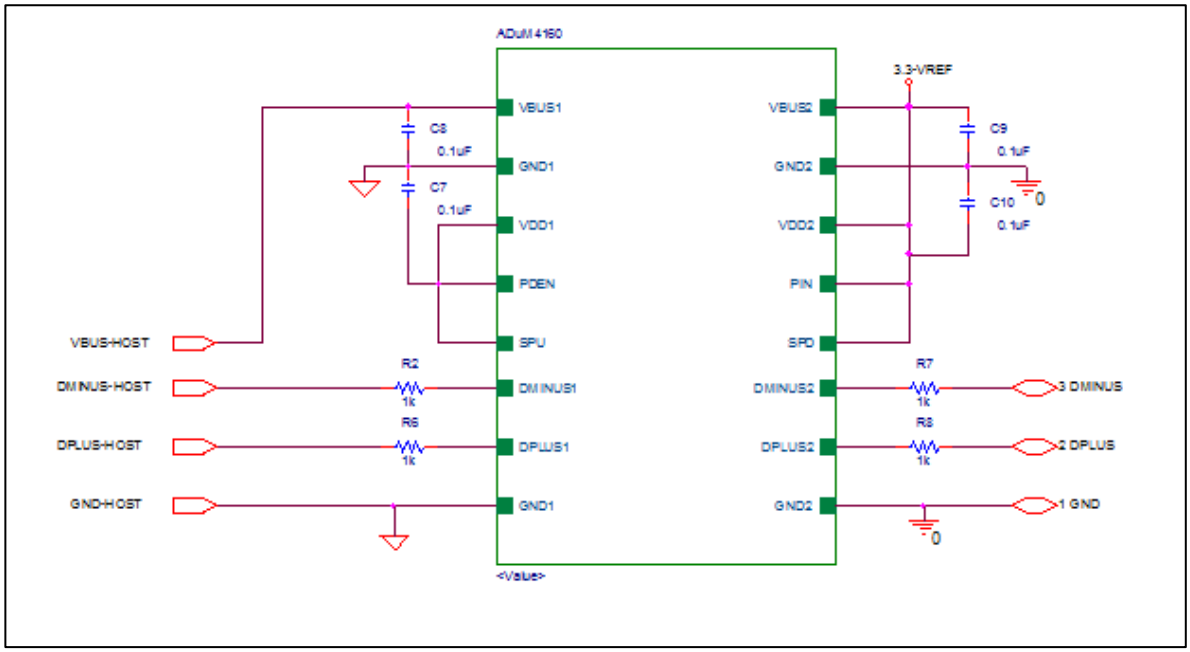


Fuente: Autores

La operación de los servomotores está entre una tensión de 4.8 a 6.0 Volts y una corriente de 150 a 180 mili-Amperes a operación nominal, el arreglo de transistores soporta la demanda de tensión y corriente de los servomotores, en la Figura 90, se observa que las líneas de control de los servomotores es independiente para cada servomotor, es decir por cada servomotor hay una línea de control, teniendo en cuenta que los servomotores son de características similares, la etapa de amplificación y potencia de la señal, tendrá las características de eléctricas(tensión, corriente) muy similares. Teniendo en cuenta las características anteriores la etapa de amplificación y potencia la conforma un par de transistores de baja potencia conectado en cascada como lo muestra la Figura 89, al primer transistor recibe la señal proveniente del amplificador operacional y la amplifica e invierte dicha señal, el segundo transistor tiene la función de restaurar la señal de la forma original a demás actúa como switch del

servomotor. Por último los servomotores son el corazón de un arreglo mecánico donde su objetivo será mostrar la traducción en alto relieve, para que el usuario pueda leer.

Figura 90 Conexiones recomendadas para el ADuM4160



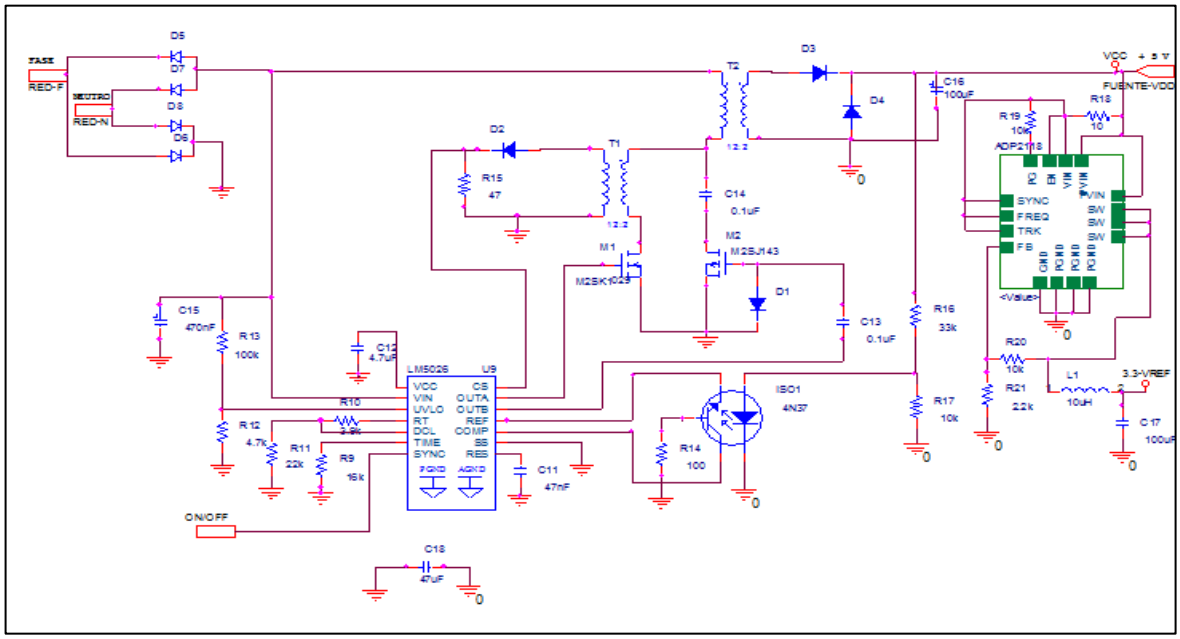
Fuente: Autores

El ADuM4160 (Figura 90) es un aislador de puerto USB, con base en tecnología Analog Devices, iCoupler tecnology. La combinación de CMOS de alta velocidad y transformadores con tecnología monolítica con núcleo de aire, estos componentes de aislamiento proporcionan características excepcionales de rendimiento y se integran fácilmente con una baja y alta velocidad USB, compatibles con los dispositivos periféricos estándares.

Muchos microcontroladores implementan USB, de modo que sólo se presentan los canales DPLUS y DMINUS a terminales externos. Esto es deseable en muchos casos, ya que minimiza los componentes externos y simplifica el diseño, sin embargo, esto presenta retos particulares cuando es necesario un aislamiento. Líneas de USB de forma automática debe cambiar entre impulsando activamente

DPLUS / DMINUS, la recepción de datos y permitiendo que las resistencias externas para establecer el estado de inactividad del bus. El ADuM4160 proporciona los mecanismos para detectar la dirección del flujo de datos y control sobre el estado de los buffers de salida. Dirección de datos se determina sobre una base paquete por paquete.

Figura 91 Propuesta, fuente alimentación de 5 y 3.3 V a 2 A.



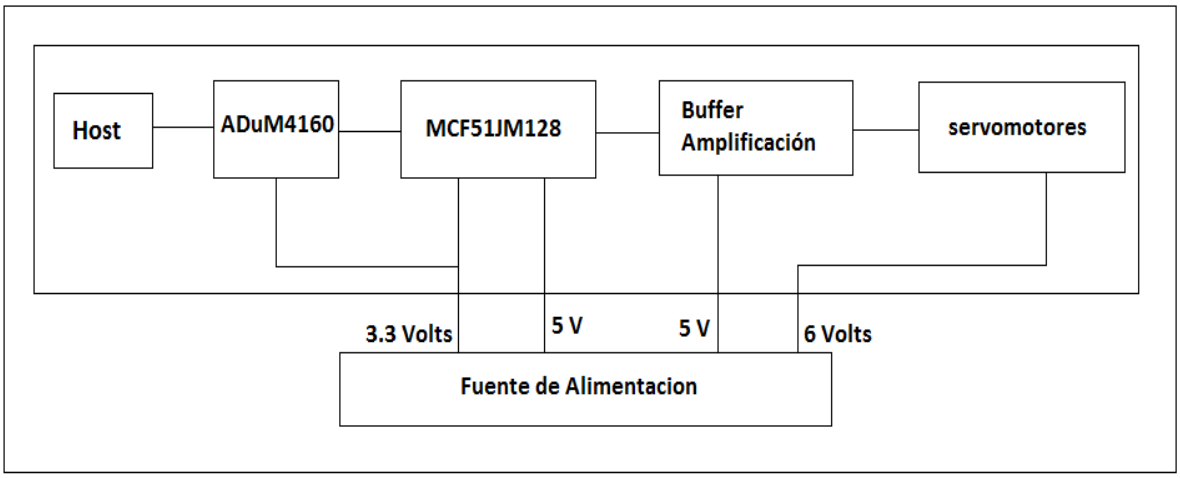
Fuente: Autores

Especificaciones del dispositivo de salida

- Puerto USB OTG
- Fuente de Alimentación.

En la figura 92, se ilustra los valores de tensión a condiciones funcionales de operación del dispositivo, la fuente de alimentación cuenta con una entrada una tensión de 120 Volts AC y tensiones de salida de sus reguladores de 3.3, 5 y 6 Volts DC, con una corriente de 1.5 Amperes como se muestra en la tabla 6.

Figura 92 Diagrama alimentación del dispositivo de salida.



Fuente: Autores

Los valores relacionados en la tabla 8, son tomados de las hojas de datos de los dispositivos electrónicos seleccionados, los datos de tensión son los recomendados por el fabricante y los datos de corriente son los valores máximos soportados por cada dispositivo según su fabricante.

Tabla 8 Características importantes de dispositivos utilizados para diseño.

Dispositivo	Tensión	Corriente
ADuM4160	3.3 Vref DC	6 mA
MCF51JM128	3.3 Vref – 5 Volts DC	120 mA
Buffer	5 Volts DC	6*(2.8 mA)
Servomotores	6 Volts DC	6*(180 mA)
Fuente alimentación	**120 V AC. - 3.3,5,6 V DC**	**1.5 Amperes**

Fuente: Autores