

REPOSITORIO ACADÉMICO UPC

Sistema de reconocimiento de caracteres numéricos para actualización de base de datos

Item Type	info:eu-repo/semantics/bachelorThesis
Authors	Espejo Pérez, Caterina María
Citation	[1] E. Pérez and C. María, "Sistema de reconocimiento de caracteres numéricos para actualización de base de datos," Universidad Peruana de Ciencias Aplicadas (UPC), 2017.
Publisher	Universidad Peruana de Ciencias Aplicadas (UPC)
Rights	info:eu-repo/semantics/openAccess
Download date	11/11/2023 10:09:05
Item License	http://creativecommons.org/licenses/by-nc-nd/4.0/
Link to Item	http://hdl.handle.net/10757/622402



UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA ELECTRÓNICA

**Sistema de reconocimiento de caracteres numéricos
para actualización de base de datos**

TESIS Para optar el Título de: INGENIERA ELECTRÓNICA

AUTOR:

Espejo Pérez, Caterina María

ASESOR:

Salas Arriarán, Sergio

LIMA - PERÚ 2017

TABLA DE CONTENIDO

RESUMEN	10
INTRODUCCIÓN	11
CAPÍTULO 1. ASPECTOS INTRODUCTORIOS	13
1.1. Presentación y Justificación del Problema	13
1.2. Estado del Arte	14
1.3. Ventajas y Desventajas	20
1.4. Aplicaciones	25
1.5. Objetivos	28
1.5.1. Objetivo General	28
1.5.2. Objetivos específicos	28
1.6. Viabilidad	29
1.6.1. Viabilidad técnica	29
1.6.2. Viabilidad económica	29
1.6.3. Viabilidad operativa	30
1.6.4. Viabilidad social	30
1.6.5. Viabilidad ambiental	31
1.7. Breve descripción del proyecto	31
1.8. Comentarios finales	36
CAPÍTULO 2. MARCO TEÓRICO	37
2.1. Iluminación blanca	37
2.2. Cámara digital	37
2.3. Función Signo	44
2.4. Teoría de Imágenes	44
2.5. Proyecciones	48
2.6. Filtro Gaussiano	49
2.7. Interpolación Bicúbica	50
2.8. Distancia euclíadiana	51
2.9. Funciones de Matlab	52
2.9.1. Función bwmorph	52

2.9.2. Función bwtraceboundary	52
2.10. Descriptores de Fourier	54
2.11. Redes Neuronales	55
2.11.1. Características de las Redes Neuronales	57
2.11.2. Perceptrón de Múltiple Capa (MLP)	58
2.11.3. Backpropagation.....	61
2.12. Porcentaje de reconocimientos	67
2.13. Interfaz visual Guide	68
2.14. Microsoft Access	69
2.14.1. Tablas	70
2.15. Comentarios finales	71
CAPÍTULO 3. DESCRIPCIÓN DEL HARDWARE DEL PROYECTO	72
3.1. Comentarios finales	79
CAPÍTULO 4. DESCRIPCIÓN DEL SOFTWARE DEL PROYECTO	80
4.1. Captura de la imagen	83
4.2. Binarización de la imagen	86
4.3. Segmentación de la imagen	90
4.4. Rotación del segmento	92
4.5. Captura del caracter	96
4.6. Algoritmos para reconocimiento	99
4.6.1. Cuadrantes	99
4.6.2. Variaciones	101
4.6.3. Distancias	102
4.6.4. Descriptores de Fourier	105
4.7. Entrenamiento de la red neuronal.....	107
4.8. Reconocimiento del caracter	110
4.9. Interfaz visual	112
4.10. Diagramas de bloques del software del proyecto	119
4.11. Comentarios finales	123
CAPÍTULO 5. PRUEBAS, RESULTADOS Y VALIDACIÓN	124
5.1. Comentarios finales	137
CONCLUSIONES	139
RECOMENDACIONES PARA TRABAJOS FUTUROS	141

ANEXO	142
Manual de operación de la interfaz visual	142
BIBLIOGRAFÍA	149

LISTA DE FIGURAS

Figura 1.2.1 Ejemplo de formulario completado.....	16
Figura 1.2.2 Personalización del documento en el software	17
Figura 1.2.3 Proceso del software ABBYY FlexiCapture	17
Figura 1.2.4 Ejemplo del proceso del componente A2ia TextReader	18
Figura 1.2.5 Ejemplo de formulario a procesar por OcuLens	19
Figura 1.2.6 Ejemplo de formulario completado con números	19
Figura 1.4.1 Formato de evaluación	25
Figura 1.4.2 Trazo de números para el acta de escrutinio	26
Figura 1.4.3 Ejemplo de un acta de escrutinio llenada.....	26
Figura 1.4.4 Trazo de números y letras para el padrón general de hogares	27
Figura 1.4.5 Ejemplo de padrón general de hogares	27
Figura 1.7.1 Diagrama pictórico de la fase de entrenamiento	33
Figura 1.7.2 Diagrama pictórico de la fase de reconocimiento	34
Figura 1.7.3 Diagrama de flujo de la interfaz visual	35
Figura 2.2.1 Elementos que componen un sensor	38
Figura 2.2.2 Estructura CMOS	39
Figura 2.2.3 Representación de la longitud focal	40
Figura 2.2.4 Imágenes obtenidas con la longitud focal respectiva	40
Figura 2.2.5 Ejemplo de una imagen en distintas resoluciones.....	41
Figura 2.2.6 Cambios de enfoque.....	43
Figura 2.3.1 Función signo	44
Figura 2.4.1 Esquema del cubo de color RGB	47
Figura 2.6.1 Representación de una máscara de filtro 3x3.....	49
Figura 2.7.1 Píxeles considerados para la interpolación bicúbica.....	51
Figura 2.9.2.1 Direcciones de búsqueda.....	53
Figura 2.11.1 Neurona artificial	56
Figura 2.11.2.1 Perceptrón	58
Figura 2.11.2.2 Estructura del MLP	60
Figura 2.11.3.1 Estructura del MLP del proyecto	62

Figura 2.11.3.2 Función de activación sigmoide.....	63
Figura 3.1 Cámara web.....	72
Figura 3.2 Plano del eje vertical	76
Figura 3.3 Plano del eje horizontal	76
Figura 3.4 Foto del recinto	77
Figura 3.5 Foto del recinto visto desde arriba	77
Figura 3.6 Ejemplos de formularios	78
Figura 3.7 Diagrama de flujo del hardware	78
Figura 3.8 Pasos para la instalación del hardware.....	79
Figura 4.1 Diagrama pictórico de la fase de entrenamiento	81
Figura 4.2 Diagrama pictórico de la fase de reconocimiento	82
Figura 4.1.1 Visualización de la cámara.....	84
Figura 4.1.2 Imagen obtenida en escala de grises	84
Figura 4.1.3 Imagen recortada.....	85
Figura 4.1.4 Sección a analizar de la ficha óptica	85
Figura 4.1.5 Diagrama de flujo de la captura de la imagen.....	86
Figura 4.2.1 Histograma de la imagen obtenida	88
Figura 4.2.2 Histograma de la imagen binarizada	88
Figura 4.2.3 Imagen binarizada	89
Figura 4.2.4 Diagrama de flujo de la binarización de la imagen.....	90
Figura 4.3.1 Proyección vertical de la imagen binarizada.....	91
Figura 4.3.2 Imagen del segmento elegido.....	91
Figura 4.3.3 Diagrama de flujo de la segmentación de la imagen.....	92
Figura 4.4.1 Imagen del segmento inclinado.....	92
Figura 4.4.2 Imagen enderezada con interpolación bicúbica	93
Figura 4.4.3 Imagen binarizada del segmento enderezado.....	93
Figura 4.4.4 Imagen del segmento enderezado	94
Figura 4.4.5 Diagrama de flujo de la rotación del segmento.....	95
Figura 4.5.1 Proyección horizontal del segmento	96
Figura 4.5.2 Resultados de cada paso para la captura del carácter.....	97
Figura 4.5.3 Diagrama de flujo de la captura del carácter.....	98
Figura 4.6.1.1 Direcciones de búsqueda en cada cuadrante	100
Figura 4.6.1.2 Diagrama de flujo del algoritmo de las variaciones en los cuadrantes .	101

Figura 4.6.2.1 Diagrama de flujo del algoritmo de las variaciones en filas y columnas	102
Figura 4.6.3.1 Posiciones de la matriz caracter con respecto a su centro.....	103
Figura 4.6.3.2 Direcciones del algoritmo de las distancias	104
Figura 4.6.3.3 Diagrama de flujo del algoritmo de las distancias	104
Figura 4.6.4.1 Valores de todos los descriptores de Fourier	106
Figura 4.6.4.2 Valores de los 10 primeros descriptores de Fourier.....	106
Figura 4.6.4.3 Diagrama de flujo del algoritmo de los descriptores de Fourier.....	107
Figura 4.7.1 Documento Excel	108
Figura 4.7.2 Entrenamiento de la red neuronal	109
Figura 4.7.3 Diagrama de flujo del entrenamiento de la red neuronal	110
Figura 4.8.1 Diagrama de flujo del reconocimiento del caracter	112
Figura 4.9.1. Ejemplos de formularios llenados	116
Figura 4.9.2 Plataforma visual Guide de Matlab.....	116
Figura 4.9.3 Base de datos de MS Access	117
Figura 4.9.4 Diagrama de flujo de la interfaz visual	118
Figura 4.10.1 Diagrama de bloques de la captura de la imagen	119
Figura 4.10.2 Diagrama de bloques de la captura del segmento	120
Figura 4.10.3 Diagrama de bloques para el entrenamiento	121
Figura 4.10.4 Diagrama de bloques para el reconocimiento	122
Figura 5.1 Mensaje de un verdadero reconocimiento.....	124
Figura 5.2 Mensaje de un falso reconocimiento.....	125
Figura 5.3 Verdaderos reconocimientos por caracter	126
Figura 5.4 Ejemplos del n�mero cero.....	128
Figura 5.5 Diferentes formas de aprendizaje del n�mero uno.....	128
Figura 5.6 Ejemplos del inconveniente del n�mero uno al escalar a 32x32	129
Figura 5.7 Porcentajes de reconocimiento del n�mero uno.....	129
Figura 5.8 Ejemplos del n�mero dos	130
Figura 5.9 Porcentajes de reconocimiento del n�mero dos	130
Figura 5.10 Ejemplos del n�mero tres	131
Figura 5.11 Porcentajes de reconocimiento del n�mero tres.....	131
Figura 5.12 Ejemplos del n�mero cuatro.....	131
Figura 5.13 Porcentajes de reconocimiento del n�mero cuatro.....	132

Figura 5.14 Ejemplos del número cinco	132
Figura 5.15 Porcentajes de reconocimiento del número cinco	132
Figura 5.16 Ejemplos del número seis	133
Figura 5.17 Porcentajes de reconocimiento del número seis.....	133
Figura 5.18 Ejemplos del número siete	134
Figura 5.19 Porcentajes de reconocimiento del número siete	134
Figura 5.20 Ejemplos de aprendizaje del número ocho.....	134
Figura 5.21 Porcentajes de reconocimiento del número ocho.....	135
Figura 5.22 Ejemplos del número nueve	136
Figura 5.23 Porcentajes de reconocimiento del número nueve	136
Figura A1.1 Primer paso a seguir	142
Figura A1.2 Segundo paso a seguir	143
Figura A1.3 Tercer paso a seguir	143
Figura A1.4 Cuarto paso a seguir	144
Figura A1.5 Quinto paso a seguir.....	145
Figura A1.6 Sexto paso a seguir.....	145
Figura A1.7 Séptimo paso a seguir.....	146
Figura A1.8 Octavo paso a seguir	147
Figura A1.9 Base de datos actualizada.....	147
Figura A1.10 Noveno paso a seguir	148

LISTA DE TABLAS

Tabla 1.3.1 Cuadro comparativo de ventajas y desventajas	23
Tabla 1.6.2.1 Informe económico inicial.....	30
Tabla 2.11.2.1 Número de funciones linealmente separables	59
Tabla 3.1 Especificaciones de la cámara web	73
Tabla 4.9.1 Descripción de las tablas de la base de datos	115
Tabla 5.1 Verdaderos reconocimientos por carácter	126
Tabla 5.2 Carácter a reconocer y carácter reconocido.....	127
Tabla 5.3 Informe económico final	137

RESUMEN

El presente proyecto consiste en implementar un sistema de reconocimiento de caracteres numéricos escritos a mano para la actualización de una base de datos. El funcionamiento del sistema se basa en tomar una foto con una cámara web a un formulario de datos llenados manualmente, reconocerlos y guardarlos en la base de datos, en sus casillas correspondientes. El sistema tiene la capacidad de reconocer números escritos a mano gracias al entrenamiento de redes neuronales. Para un correcto entrenamiento, se utilizaron 185 ejemplares de cada número escritos por diferentes personas (un total de 1850 caracteres). El producto final está implementado en Matlab y se relaciona con las bases de datos de Microsoft Access, donde se guardan los números reconocidos.

INTRODUCCIÓN

El sistema de reconocimiento de caracteres extrae de una imagen los caracteres que componen un texto para almacenarlos en un formato de texto. En una imagen, los caracteres se describen indicando cada uno de los píxeles que los forman pero al convertirlos a un formato de texto son descritos por un solo valor.

Actualmente, el reconocimiento preciso en textos mecanografiados se considera un problema resuelto. El reconocimiento de la impresión manual (aquella que proviene de la caligrafía humana) sigue siendo una fuente de intensa investigación. Existen tres métodos para transcribir los datos escritos a una computadora y son los siguientes: el ICR, el OCR y el OMR. El ICR y OCR son sistemas de reconocimiento usados con imágenes y el OMR es una tecnología de captura de datos.

El Reconocimiento Inteligente de Carácteres (ICR por sus siglas en inglés) convierte imágenes de caracteres en letra manuscrita (no cursiva) a caracteres descifrados por una computadora. Estas imágenes son obtenidas por un escáner. El Reconocimiento Óptico de Carácteres (OCR por sus siglas en inglés) es semejante a la tecnología ICR pero con la desventaja de que solo puede convertir imágenes de caracteres en letra a máquina. El Reconocimiento Óptico de Marcas (OMR por sus siglas en inglés) detecta la ausencia o presencia de una marca, pero no su forma. Los formularios contienen pequeños círculos (burbujas) que deben ser completados. Este sistema no reconoce caracteres en letra manuscrita o a máquina.

El problema existente es el tiempo que se utiliza para transcribir los datos llenados en el formulario a la computadora. Este trabajo lo realiza un digitador, el cual puede demorarse convenientemente el tiempo que deseé porque le pagan por hora. Además, hay casos en que se le paga por destajo y el digitador al intentar transcribir la mayor cantidad de formularios que pueda en un determinado tiempo, le lleva a posibles errores, especialmente en el caso de los números.

Las tecnologías ICR y OCR pueden ser altamente precisas si se consideran las siguientes condiciones: el sistema está configurado apropiadamente, los formularios

están diseñados correctamente y los caracteres obtenidos son detallados y sin errores. De acuerdo a esto, se trata de resolver el problema pero bajo condiciones ideales.

En este proyecto se tiene como finalidad agilizar el llenado de la base de datos y evitar errores humanos que pueden suceder al transcribir los datos a la computadora. El empleado solo tiene que poner el formulario en el soporte para tomarle una foto, mediante algoritmos se trasladarían directamente los datos del formulario a la computadora y con ello se completa el llenado de la base de datos.

La aplicación básica del proyecto es adquirir los datos de un formulario y llevarlos a la base de datos de la computadora. Esto resulta de mucha utilidad para los bancos, aseguradoras, hospitalares, colegios y universidades. También podría ampliarse para encuestas, actas de escrutinio e incluso el censo.

Una aplicación indirecta del proyecto es beneficiar a las personas invidentes. Los escáneres de la actualidad solo capturan el texto con la letra a máquina y el programa que utilizan en sus computadoras (JAWS) se los lee, pero esto no ocurre cuando el texto está escrito manualmente. El proyecto tiene la capacidad de convertir el texto con letra manuscrita a formato de texto en la computadora para luego poder ser leído por el programa JAWS.

CAPÍTULO 1. ASPECTOS INTRODUCTORIOS

En el presente capítulo se explicarán detalladamente los aspectos introductorios del proyecto. Estos aspectos son los siguientes: presentación y justificación del problema, estado del arte, ventajas y desventajas, aplicaciones, objetivos, viabilidad y una breve descripción del proyecto.

1.1. Presentación y Justificación del Problema

El proyecto se elaboró debido al problema existente del tiempo utilizado para transcribir los datos llenados en un formulario a la computadora. Este es un trabajo desarrollado por un digitador, quien puede retrasar convenientemente la transcripción de la información considerando que le pagan por hora. Además, en algunos casos se le paga por destajo y el digitador puede cometer errores al intentar transcribir la mayor cantidad de formularios, especialmente si son números. En el Perú, un digitador puede ganar entre 850 y 2000 soles mensuales en tiempo completo, dependiendo de las tareas a realizar, y 400 soles a tiempo parcial¹. El presente proyecto no excede de 750 soles.

Actualmente, se cuenta con las tecnologías ICR y OCR, las cuales resultan ser altamente precisas pero bajo las siguientes condiciones: el sistema está configurado apropiadamente, los formularios han sido completados correctamente y los caracteres obtenidos son detallados y sin errores. De acuerdo a esto, se trata de resolver el problema bajo condiciones ideales.

En el presente proyecto se tiene como finalidad agilizar el llenado de la base de datos y evitar errores humanos que pueden suceder al transcribir los datos a la computadora. El empleado solo debe colocar el formulario en el soporte para tomarle una foto, mediante algoritmos se trasladarían directamente los datos del formulario a la computadora y con ello se completa el llenado de la base de datos. Por ejemplo, si se tiene un formulario

¹ Cfr. Empleos Bumeran Perú 2016; Computrabajo 2016.

con 26 casillas, al sistema le tomaría 2.60 segundos para reconocer los caracteres. También, si se quisiese actualizar los datos de un formulario de evaluación (examen de admisión de institutos o universidades), el proyecto debería realizarlo en no más de 1 segundo.

1.2. Estado del Arte

Phokharatkul y Kimpan en la publicación “Handwritten Thai character recognition using Fourier descriptors and genetic neural networks” de la revista Computational Intelligence, emplean los descriptores de Fourier para reconocer los caracteres escritos en thai. Además, se clasifican los caracteres por los bordes internos y externos, y sus curvaturas. Se utilizan las redes neuronales genéticas para el reconocimiento. Se obtiene un índice de reconocimiento de 99.12%, de malos reconocimientos de 0.23% y un 0.65% de rechazos.

Igualmente, Chung y Wong en la publicación “Handwritten character recognition by Fourier descriptors and neural network” en la IEEE Conference on Speech and Image Technologies for Computing and Telecommunications, utilizan los descriptores de Fourier, pero se complementan con la técnica de transición de bordes para el reconocimiento de caracteres escritos a mano. Este reconocimiento se realiza a través de una red neuronal. Se obtiene como resultado el 96% de precisión para los números escritos a mano, del 0 al 9.

También, Rajput y Mali en la publicación “Fourier descriptor based isolated Marathi handwritten numeral recognition” de la revista International Journal of Computer Applications, proponen los descriptores de Fourier para el reconocimiento de números aislados escritos en marathi. Se utilizan tres clasificadores diferentes para reconocer los números de prueba y son los siguientes: vecindad más cercana (nearest neighborhood NN), vecindad más cercana de K (K-nearest neighborhood KNN) y máquina de vectores de soporte (Support Vector Machine SVM). Se clasifican con 64 descriptores con una base de datos de 13000 muestras de números empleando el método de validación cruzada cinco veces para el cálculo de resultados. Se obtienen los porcentajes de reconocimiento de 97,05%, 97,04% y 97,85% para cada método, respectivamente.

Además, Singh et al. en la publicación “Hand written character recognition using twelve directional feature input and neural network” de la revista International Journal of Computer Applications, emplean las gradientes para extraer las propiedades de los caracteres. Se utiliza una máscara de Sobel para obtener las gradientes de cada píxel. A partir de estos resultados, se agrupan las gradientes en 12 direcciones. Como el carácter está compuesto por una matriz de 32x32, se tendrían 1024 valores de dirección para ser entrenados por la red neuronal. Se obtiene 97% de rendimiento. Cabe resaltar que los caracteres analizados son tres en hindi, uno en inglés y un carácter especial.

Asimismo, Hallale y Salunke en la publicación “Offline handwritten digit recognition using neural network” de la revista International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, realizan el pre-procesamiento para la obtención de características de los números escritos a mano. También se estandariza la imagen a una matriz de 32x32 y se ingresan estos 1024 valores de los píxeles a una red neuronal. Se entrena 25 ejemplos de cada número y se realizan las pruebas con 50 imágenes. El promedio de verdaderos reconocimientos de todos los dígitos es de 91.2%.

También, Ahmad et al. en la publicación “Urdu nastaleeq optical character recognition” de la revista Proceedings of World Academy of Science, Engineering and Technology, reconocen los caracteres en urdu nastaleeq en texto impreso. Para ello se basan en el nivel de complejidad del texto y la segmentación de los caracteres de acuerdo a esto. Las redes neuronales son empleadas para el reconocimiento. Se probó un promedio de 93.4% de aciertos.

Asimismo, Zand et al. en la publicación “Recognition-based segmentation in Persian character recognition” de la revista Proceedings of World Academy of Science, Engineering and Technology, reconocen los caracteres escritos en persa. Se utiliza el método de segmentación de palabras y la técnica de segmentación basado en el reconocimiento. Se dividen las palabras respecto a las propiedades de su estructura y de acuerdo a ello se reconocen los caracteres.

En el mercado, existen sistemas que reconocen los caracteres. La mayoría solo tiene la capacidad de reconocer los caracteres escritos a máquina. Por ejemplo, se puede

observar esta característica en los escáneres. Sin embargo, existen varias tecnologías que reconocen caracteres escritos a mano.

Por ejemplo, ABBYY FlexiCapture es un software de captura de datos y procesamiento de formularios en papel y archivos PDF. Se encuentra diseñado para transformar documentos de cualquier estructura y complejidad en datos listos para el trabajo. Consiste de tecnologías de reconocimiento, clasificación automática de documentos y de una arquitectura altamente escalable y personalizable. Puede reconocer 192 idiomas. El producto estándar cuesta 169 dólares y el corporativo es de 249 dólares².

El funcionamiento del software ABBYY FlexiCapture consiste en escanear un lote de formularios completados (Figura 1.2.1) utilizando un escáner de alta velocidad, que escanee al menos 10 páginas por minuto. Se puede personalizar el documento en el software para que coincida con el formulario a analizar (Figura 1.2.2). La mayoría de los datos se reconocen automáticamente; pero los que no, se transmiten a un operador humano. Después de la verificación, los datos se guardan en una base de datos. Se requiere de sólo un operador para todo el proceso debido a que todas las etapas, excepto la verificación, se encuentran automatizadas³. Este proceso se observa en la Figura 1.2.3.

Miss
First Name: Jennifer
Last Name: Pitt
Middle Initial: A
Birth Date: 01-06-1966
Nationality: British
SSN: 719706442
Country: USA
ZIP code: 94633

Figura 1.2.1 Ejemplo de formulario completado⁴

² Cfr. ABBYY 2017.

³ Cfr. ABBYY 2017.

⁴ Cfr. ABBYY 2017.

The screenshot shows a software interface for document personalization. At the top, there's a section titled 'PERSONAL DETAILS' with fields for 'Main' (Addressing dropdown set to 'Mr'), 'SSN' (empty field), and 'Name' (First Name, Initial, Last Name). Below that is a 'Nationality' field and 'City of Birth' and 'Birth Date' fields. A horizontal line separates this from the 'ADDRESS/CONTACT DETAILS' section, which contains an 'Address' field with 'Address type' and 'ZIP' fields.

Figura 1.2.2 Personalización del documento en el software⁵

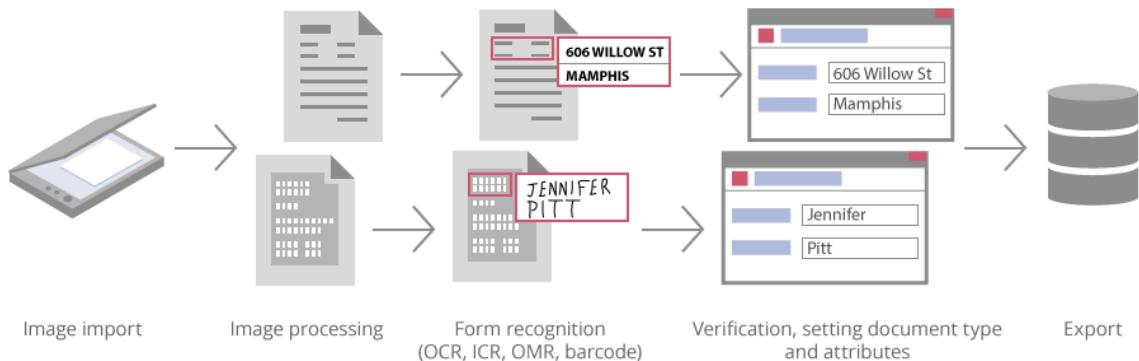


Figura 1.2.3 Proceso del software ABBYY FlexiCapture⁶

Además, A2ia TextReader es un componente de reconocimiento dedicado a transcripción completa de documentos impresos y manuscritos. Se obtiene como resultado datos transformados en formatos editables. Se basa en tecnología de redes

⁵ Cfr. ABBYY 2017.

⁶ Cfr. ABBYY 2017.

neuronales⁷. En la Figura 1.2.4 se muestra un ejemplo del proceso del componente A2ia TextReader.

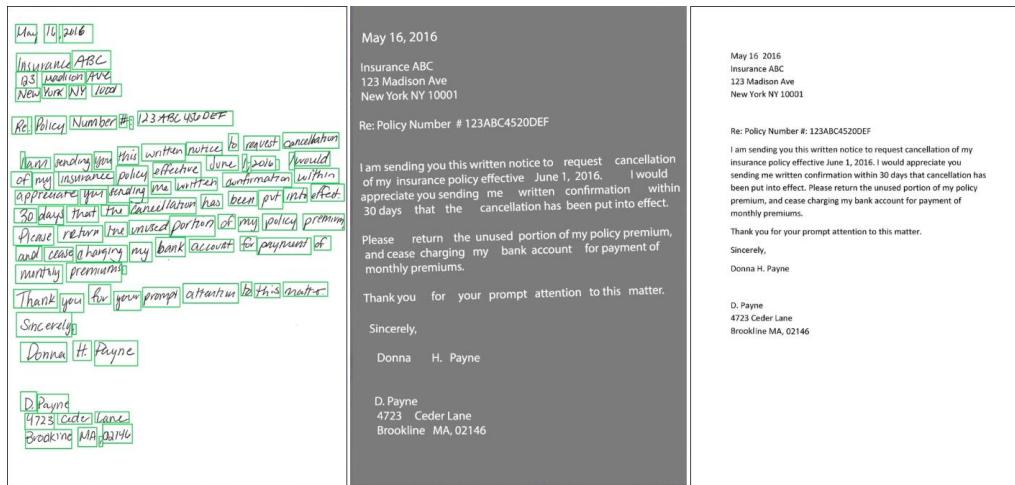


Figura 1.2.4 Ejemplo del proceso del componente A2ia TextReader⁸

También, OcuLens es un conjunto de componentes de desarrollo de software que permite crear un software de captura de datos personalizado. El cual cuenta con la capacidad de procesar formularios individuales llenos en un corto período de tiempo. Puede escanear de 30 a 40 formularios por minuto⁹.

El proceso de OcuLens consta de escanear formularios de papel. Luego, reconoce los datos valiosos de las imágenes. Por ejemplo, casillas de verificación, códigos de barras, caracteres impresos en máquina o caracteres escritos a mano. Además, el usuario debe de asegurarse en la computadora de reconocimiento que procesa los formularios adecuados. De ser necesario, debe validar e interpretar los datos que se han reconocido. Por último, se exportan los datos validados y correctos a las bases de datos para su posterior procesamiento¹⁰.

⁷ Cfr. A2iA TextReader 2017.

⁸ Cfr. A2iA TextReader 2017.

⁹ Cfr. Active Record Software Systems 2017.

¹⁰ Cfr. Active Record Software Systems 2017.

En las aplicaciones reales, se encuentran casillas de verificación, códigos de barra, letras y números impresos (legibles en 100%) y números escritos manualmente (reconocibles con un 98% de precisión). Una solución frecuente es registrar por el método tradicional sólo las partes que el software puede reconocer con dificultad, como las escritas de forma manual. Debido a que se ingresarían los datos más rápido, incluso si la computadora procesa las demás partes¹¹. En las Figuras 1.2.5 y 1.2.6 se muestran unos ejemplos de formularios completados para ser procesados por el software OcuLens.

OcuLens - Demo Form - Work Report		053632
		49 07:10 1 07:58
		18 09:10 2 09:47
	6.	63 09:51 3 10:11

Figura 1.2.5 Ejemplo de formulario a procesar por OcuLens¹²

Employee ID:	1007	Date year:	2009	month:	0
Morn.	<input checked="" type="checkbox"/>	Aftern.	<input type="checkbox"/>	Night	<input type="checkbox"/>
			Working hours	start (hh:mm):	06:07
				finish	- 11

Figura 1.2.6 Ejemplo de formulario completado con números¹³

¹¹ Cfr. Active Record Software Systems 2017.

¹² Cfr. Active Record Software Systems 2017.

¹³ Cfr. Active Record Software Systems 2017.

1.3. Ventajas y Desventajas

Una desventaja de Singh et al. en la publicación “Hand written character recognition using twelve directional feature input and neural network” es que los algoritmos utilizados pueden resultar computacionalmente costosos. En el presente proyecto, el entrenamiento de la red neuronal brinda los pesos necesarios para posteriormente realizar el reconocimiento. Por lo tanto, no es necesario entrenar cada vez que se deseé reconocer un carácter y por ello el costo computacional es bajo. Además, los algoritmos utilizados para obtener la información del carácter no son computacionalmente costosos.

En algunos casos, como en Ahmad et al. en la publicación “Urdu nastaleeq optical character recognition”, el reconocimiento es utilizado para texto impreso. Esto se considera una desventaja debido a que el sistema está limitado. En el proyecto se completa la información de los formularios a mano y se han entrenado las redes neuronales con ejemplares diferentes de cada carácter escritos por diversas personas. Con ello, se puede realizar el reconocimiento de caracteres numéricos escritos a mano.

Además, en los casos de texto escrito se observa la desventaja del proceso de identificar los caracteres en la palabra debido a que están entrelazados. Estos casos se encuentran en Ahmad et al. en la publicación “Urdu nastaleeq optical character recognition” y en Zand et al. en la publicación “Recognition-based segmentation in Persian character recognition”. En el presente proyecto se usan casillas para evitar este inconveniente. Cabe resaltar que el sistema reconoce caracteres numéricos, los cuales no se escriben entrelazados entre ellos. También, de haberse utilizado sólo líneas, las personas podrían escribir sobre ella o con números de diferentes tamaños, lo cual complicaría la captura del segmento del formulario. Al utilizar las casillas para completar los datos solicitados se evita este problema, porque el segmento tiene un solo tamaño, a pesar de que los caracteres escritos dentro pueden ser de distinto tamaño.

También, en Zand et al. en la publicación “Recognition-based segmentation in Persian character recognition” se encuentra la desventaja de que si un carácter no se ha reconocido, los demás no se reconocerían correctamente. Al utilizar casillas en el proyecto para completar los datos de los formularios, cada reconocimiento es independiente entre los caracteres. Por lo tanto, no se tiene este problema. Si se llega a

reconocer correctamente o no un carácter, no perjudica el reconocimiento de los siguientes caracteres.

En Hallale y Salunke en la publicación “Offline handwritten digit recognition using neural network”, aunque se realice el pre-procesamiento para mejorar la imagen a analizar se debe considerar que se ingresan 1024 valores para entrenar la red neuronal, lo cual es una desventaja. Esto se debe a que la imagen del número está compuesta por valores de píxel cero (fondo) que no se deberían ingresar a la red por no brindar ninguna información esencial del carácter. Por lo tanto, se estaría entrenando con data sin valor, lo que no es deseable. Además, se entranan 25 ejemplos de cada número, por lo que el porcentaje obtenido no resulta fidedigno al ser una cantidad muy pequeña para entrenar la red neuronal.

En algunos casos, el porcentaje de precisión es de 99.12% o 96%. Estos porcentajes se obtienen en Phokharatkul y Kimpan en la publicación “Handwritten Thai character recognition using Fourier descriptors and genetic neural networks” y en Chung y Wong en la publicación “Handwritten character recognition by Fourier descriptors and neural network”, respectivamente. Sin embargo, en el presente proyecto se logró un resultado de verdaderos reconocimientos de 91.30% (porcentaje sobre el total de caracteres) y falsos rechazos en un 0.20%. En el Capítulo 5 se encuentra mayor información sobre los resultados obtenidos. Cabe resaltar que el proyecto implementa un sistema que aplica sus algoritmos para reconocer formularios y fichas ópticas. En los artículos se explican los sistemas pero no se tiene evidencia de que se aplique en la realidad.

En Chung y Wong en la publicación “Handwritten character recognition by Fourier descriptors and neural network”, para obtener el 96% de precisión se entrenaron las redes neuronales con un total de 80 caracteres, los cuales son los números del 0 al 9 con diferentes tamaños y estilos de escritura. Además, se indica que el tiempo de reconocimiento es de 0.18 segundos por carácter. Por lo tanto, se habría entrenado la red solo con 8 ejemplares de cada número, lo que significa que el porcentaje obtenido no es completamente confiable debido a que son muy pocos ejemplos para que las redes neuronales hayan entrenado correctamente. Con respecto al tiempo de reconocimiento, si se tiene como ejemplo un formulario con 26 casillas, al sistema le tomaría 4.68 segundos para reconocer los datos. En cambio, el proyecto lo realizaría en 2.60

segundos debido a que tiene un tiempo promedio de reconocimiento de 0.10 segundos por carácter.

En Phokharatkul y Kimpan en la publicación “Handwritten Thai character recognition using Fourier descriptors and genetic neural networks” al obtener el 99.12% se tiene 0.65% de rechazos. Además, se entrenó a la red neuronal con 1200 ejemplos de palabras escritas en thai por 60 personas. El proyecto tiene un porcentaje de falsos rechazos de 0.2%. En este artículo se identifican los caracteres escritos en thai, por lo que no se podría realizar una comparación por completo con el proyecto, debido a que no se están evaluando los mismos caracteres. Sin embargo, se hace referencia por los métodos utilizados para poder obtener información de caracteres.

Con respecto a los productos que existen en el mercado, se observa que tienen la capacidad de reconocer los números escritos a mano sin muchos inconvenientes¹⁴. En el presente proyecto se podría aumentar la cantidad de ejemplares de cada número para que se efectúe un mejor entrenamiento de las redes neuronales. Por lo tanto, se podría mejorar el porcentaje de verdaderos reconocimientos.

Además, debido a que estas tecnologías procesan archivos escaneados; por ejemplo, en formato PDF. Se tiene la opción de emplear escáneres para el procesamiento de los formularios, los cuales pueden escanear al menos 10 páginas por minuto¹⁵ o de 30 a 40 formularios por minuto¹⁶. Como una recomendación para trabajos futuros del proyecto, se considera modificar el recinto para colocar un grupo de formularios. Además, se capturaría automáticamente la información de cada uno de ellos.

En la Tabla 1.3.1 se muestran las distintas ventajas y desventajas del presente proyecto en comparación a las diferentes propuestas de los artículos y sistemas especificados en el estado del arte.

¹⁴ Cfr. ABBYY 2017; A2iA TextReader 2017.

¹⁵ Cfr. ABBYY 2017.

¹⁶ Cfr. Active Record Software Systems 2017.

Tabla 1.3.1 Cuadro comparativo de ventajas y desventajas

	Proyecto	Estado del Arte
Ventajas	<ul style="list-style-type: none"> • El entrenamiento brinda los pesos para el reconocimiento, el costo computacional es bajo. Los algoritmos no son computacionalmente costosos. • Se reconocen los caracteres numéricos escritos a mano, no solo a máquina. • Se usan casillas. Los números no se escriben entrelazados. • Cada reconocimiento es independiente entre los caracteres. • Se implementa un sistema para reconocer formularios y fichas ópticas. • Tiempo de reconocimiento de 0.10 segundos por carácter. • No excede de 750 soles. 	<ul style="list-style-type: none"> • En algunos casos, el porcentaje de precisión es de 99.12% o 96%. • Para obtener el 99.12%, se entrenó a la red neuronal con 1200 ejemplos de palabras escritas en thai por 60 personas. • Los productos del mercado reconocen los números escritos a mano sin muchos inconvenientes. • Se usan escáneres para el procesamiento de los formularios, los cuales escanean un promedio de 25 páginas por minuto.
Desventajas	<ul style="list-style-type: none"> • Verdaderos reconocimientos de 91.30% (porcentaje sobre el total de caracteres). • Para el entrenamiento, se utilizaron 185 ejemplares de cada número escritos por diferentes personas (total 1850). • El recinto procesa un formulario a la vez, no un grupo. 	<ul style="list-style-type: none"> • Los algoritmos utilizados pueden resultar computacionalmente costosos. • El reconocimiento es para texto impreso. El sistema está limitado. • En el caso de texto escrito se debe identificar los caracteres en la palabra porque están entrelazados. • Si un carácter no se ha reconocido,

	<p>los demás no se reconocerían correctamente.</p> <ul style="list-style-type: none"> • Se ingresan 1024 valores para entrenar la red neuronal, la mayoría de píxel cero (fondo). • Al entrenar 25 ejemplos de cada número, el porcentaje obtenido no resulta fiable. • En los artículos no hay información de que se apliquen los algoritmos en la realidad. • Para obtener el 96%, se entrenó la red neuronal con 80 caracteres; es decir, con 8 ejemplares de cada número. Este porcentaje no es confiable. • Tiempo de reconocimiento de 0.18 segundos por carácter. • El usuario debe revisar que se reconocieron correctamente los caracteres porque el software puede tener dificultad con los datos escritos de forma manual. Se prefiere que el usuario los ingrese. • El producto estándar cuesta 169 dólares y el corporativo 249 dólares.
--	--

1.4. Aplicaciones

El presente proyecto cuenta con la aplicación básica de adquirir la información de un formulario y actualizar la base de datos de la computadora. Esta aplicación puede ser empleada en diversas instituciones en las que se requiera completar un formulario. Por ejemplo en los bancos, porque cuentan con un formato diferente para cada trámite a realizar por el cliente. Este caso también se da en las aseguradoras, hospitales, colegios y universidades. Además, la aplicación podría ampliarse para encuestas, actas de escrutinio e incluso el censo.

En la Figura 1.4.1 se muestra un formato de evaluación que es mayormente utilizado en los exámenes de admisión para los institutos o universidades. Estos deben ser llenados por los postulantes a dichas instituciones.

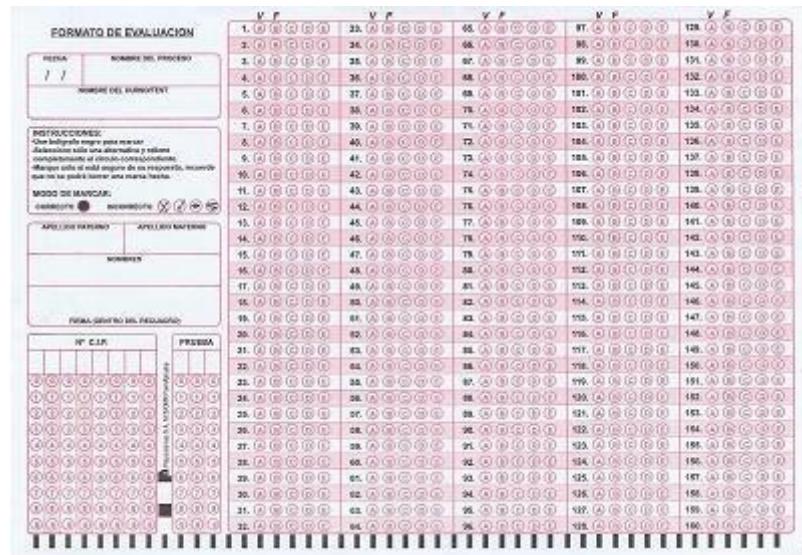


Figura 1.4.1 Formato de evaluación

En las Figuras 1.4.2 y 1.4.3 se muestra el trazo recomendado de los números para llenar un acta de escrutinio y un ejemplo de esta completa¹⁷. Esta es una de las tareas de los miembros de mesa en un proceso electoral.

0 1 2 3 4 5 6 7 8 9

Figura 1.4.2 Trazo de números para el acta de escrutinio¹⁸

C ACTA DE ESCRUTINIO	
Siendo las 4:10 p.m., del 05 de octubre de 2014, se inicia el ACTO DE ESCRUTINIO.	
ORGANIZACIONES POLÍTICAS	
1	ORGANIZACIÓN POLITICA LOCAL DISTRITAL EL DEPORTE ES SALUD
2	ORGANIZACIÓN POLITICA LOCAL PROVINCIAL JUNTOS EN ARMONIA
3	ORGANIZACION POLITICA LOCAL PROVINCIAL UN MUNDO MEJOR
4	PARTIDO POLITICO AMANECER DE NUEVO
5	ALIANZA ELECTORAL CIUDADANOS DECIDIDOS Y VALIENTES
6	MOVIMIENTO HOJAS AL VIENTO
7	MOVIMIENTO SEMBRANDO LA AMISTAD
VOTOS EN BLANCO	
VOTOS NULOS	
VOTOS IMPUGNADOS	
TOTAL DE VOTOS EMITIDOS ➔	
	180 180

Figura 1.4.3 Ejemplo de un acta de escrutinio llenada¹⁹

¹⁷ Cfr. ONPE 2014: 41-42

¹⁸ Cfr. ONPE 2014: 41.

¹⁹ Cfr. ONPE 2014: 42.

En las Figuras 1.4.4 y 1.4.5 se muestra el trazo recomendado de los números y letras para llenar el padrón general de hogares y un ejemplo de esta²⁰. Estos padrones son utilizados para el censo el cual generalmente es a nivel nacional y deben ser llenados por los voluntarios o empadronadores.



Figura 1.4.4 Trazo de números y letras para el padrón general de hogares²¹

Figura 1.4.5 Ejemplo de padrón general de hogares²²

El proyecto cuenta con la aplicación indirecta de beneficiar a las personas invidentes. Actualmente, los escáneres solo capturan el texto con la letra a máquina y las personas emplean un programa en sus computadoras para que se los lea. Por ejemplo, el programa JAWS (acrónimo de Job Access With Speech)²³. Sin embargo, esto no se

²⁰ Cfr. SISFOH 2016.

²¹ Cfr. SISFOH 2016.

²² Cfr. SISFOH 2016.

²³ Cfr. Freedom Scientific 2016.

puede realizar cuando el texto está escrito manualmente. El proyecto tiene la capacidad de convertir el texto con letra manuscrita a formato de texto en la computadora y con ello pueda ser leído por el programa JAWS.

1.5. Objetivos

A continuación, se presentan el objetivo general y los objetivos específicos del presente proyecto.

1.5.1. Objetivo General

Desarrollar un sistema computacional que permita reconocer caracteres numéricos escritos a mano con un porcentaje de verdaderos reconocimientos mayor al 90% y de falsos rechazos menor al 1%, con la finalidad de almacenar automáticamente los caracteres en una base de datos.

1.5.2. Objetivos específicos

- Desarrollar una estructura de hardware que permita la adquisición de las imágenes y un adecuado reconocimiento.
- Diseñar un formulario para llenar caracteres numéricos escritos a mano el cual facilite su segmentación.
- Investigar técnicas algorítmicas que permitan una adecuada segmentación en base a la iluminación definida por el hardware.
- Analizar e implementar algoritmos de procesamiento de imágenes y redes neuronales.
- Alcanzar una velocidad computacional de un tiempo no mayor a 3 segundos por formulario.
- Validar el algoritmo aplicando las técnicas de verdaderos reconocimientos, falsos reconocimientos y falsos rechazos.

1.6. Viabilidad

A continuación, se presentan los diferentes aspectos de la viabilidad del presente proyecto. Estos aspectos son los siguientes: viabilidad técnica, económica, operativa, social y ambiental.

1.6.1. Viabilidad técnica

En el aspecto técnico, el proyecto requiere conocimientos de procesamiento de imágenes y de inteligencia artificial, los cuales se han conseguido durante la carrera de Ingeniería Electrónica en la universidad. Estos son importantes debido a que se necesita de una correcta captura de la imagen para que se proceda a entrenar la red neuronal y con ello obtener buenos resultados. Además, debido a que se trabaja con una computadora, se necesita un software de programación en el cual se aplican los conocimientos de programación adquiridos. El software utilizado es Matlab.

Con respecto a la infraestructura construida y utilizada en el proyecto, esta se consiguió en la ciudad de Lima-Perú. No hubo necesidad de importar los componentes del proyecto debido a que ninguno resultó difícil de conseguir. Entre ellos se encuentran la cámara web, las lámparas y los focos para la iluminación.

1.6.2. Viabilidad económica

Se considera que el presente proyecto es viable económicamente debido a que sus componentes rodean los S/. 500. En la Tabla 1.6.2.1 se observa mayor detalle del informe económico inicial en que se basa el proyecto. Cabe resaltar que a este informe se tendría que aumentar el tiempo que se le invierta al proyecto para diseñar los algoritmos, aplicarlos y validar sus resultados. Además, se debería valorar los conocimientos que sean empleados.

Tabla 1.6.2.1 Informe económico inicial

Componentes del proyecto	Precio en soles (S.)
Cámara web	340
Soporte para cámara web	100
Soporte para formularios	20
Lámpara (Cantidad 2)	34
Foco (Cantidad 2)	32
TOTAL	526

1.6.3. Viabilidad operativa

Como se explicó anteriormente, el proyecto cuenta con la aplicación básica de adquirir la información de un formulario y actualizar automáticamente una base de datos a partir de los caracteres numéricos escritos a mano. Se considera que es viable operativamente debido a que las instituciones que necesiten de esta aplicación se encontrarían beneficiadas con el proyecto. Además, actualmente las instituciones cuentan con computadoras y el software utilizado se instala con facilidad.

Cabe resaltar que la estructura es parte del sistema propuesto. Por lo tanto, se garantiza que el procedimiento va a ser el apropiado debido a que se encuentran juntos la estructura con el software. Si el software estuviese aislado, el sistema no funcionaría correctamente porque con otro hardware no se lograría la iluminación adecuada.

1.6.4. Viabilidad social

Para un correcto desempeño del proyecto, se debe trabajar con la iluminación del que está compuesto. Esto podría dañar la vista de la persona si es que se trabaja en un lugar que no se encuentre iluminado, ya sea por luz natural o artificial. Por lo tanto, se

recomienda que el lugar de trabajo cuente con iluminación, además de la del proyecto, para que no se ponga en riesgo la salud visual de la persona.

También, como se puede llegar a trabajar frente a la computadora por varias horas, se podría ocasionar cansancio visual. Para evitar esto, se recomienda descansar la vista mirando a otro lado, que no sea la pantalla de la computadora, cada cierto tiempo.

1.6.5. Viabilidad ambiental

El proyecto consume energía eléctrica para su funcionamiento. Se utilizaron focos ahorradores porque tienen un tiempo de vida mayor a los focos incandescentes. Debido a esto, no se considera que el proyecto contamine el medio ambiente.

Se diseñaron unos formularios para ser llenados por diferentes personas y a partir de ellos obtener una base de datos de los caracteres numéricos escritos a mano. No se considera que sea un peligro para el medio ambiente ya que la cantidad de hojas generadas no fue de gran magnitud. Por ejemplo, no se sobrepasó de 50 hojas.

1.7. Breve descripción del proyecto

El proyecto consiste en implementar un sistema de reconocimiento de caracteres numéricos escritos a mano para la actualización automática de una base de datos. El funcionamiento del sistema se basa en tomar una foto con una cámara web a un formulario de datos llenados manualmente, reconocerlos y guardarlos en la base de datos, en sus casillas correspondientes. Para ello, se deben obtener los segmentos del formulario que se deseen analizar, esto se realiza con la segmentación de la imagen capturada y procesada. Luego, se capturan los caracteres que componen cada segmento.

Para poder reconocer los caracteres numéricos de los formularios, primero se debe tener una gran cantidad de ejemplares de cada número escritos por diferentes personas. Con esto se entrenaría correctamente la red neuronal porque estaría aprendiendo diferentes formas y estilos para escribir un mismo número. Se utilizaron 185 ejemplares de cada número, siendo un total de 1850 caracteres.

Para el reconocimiento de los caracteres, se realiza el mismo procedimiento de segmentación de los formularios y captura de los caracteres de cada segmento. Con los resultados obtenidos del entrenamiento, se procede a identificar el carácter. Después de reconocer todos los caracteres de cada segmento, se guardan los valores identificados en una base de datos. El producto final está implementado en Matlab y se relaciona con las bases de datos de Microsoft Access, donde se guardan los números reconocidos.

En la Figura 1.7.1 se muestra el diagrama pictórico de la fase de entrenamiento del proyecto y en la Figura 1.7.2, el diagrama pictórico de la fase de reconocimiento. En la Figura 1.7.3 se muestra el diagrama de flujo de los pasos a seguir en la interfaz visual del proyecto.

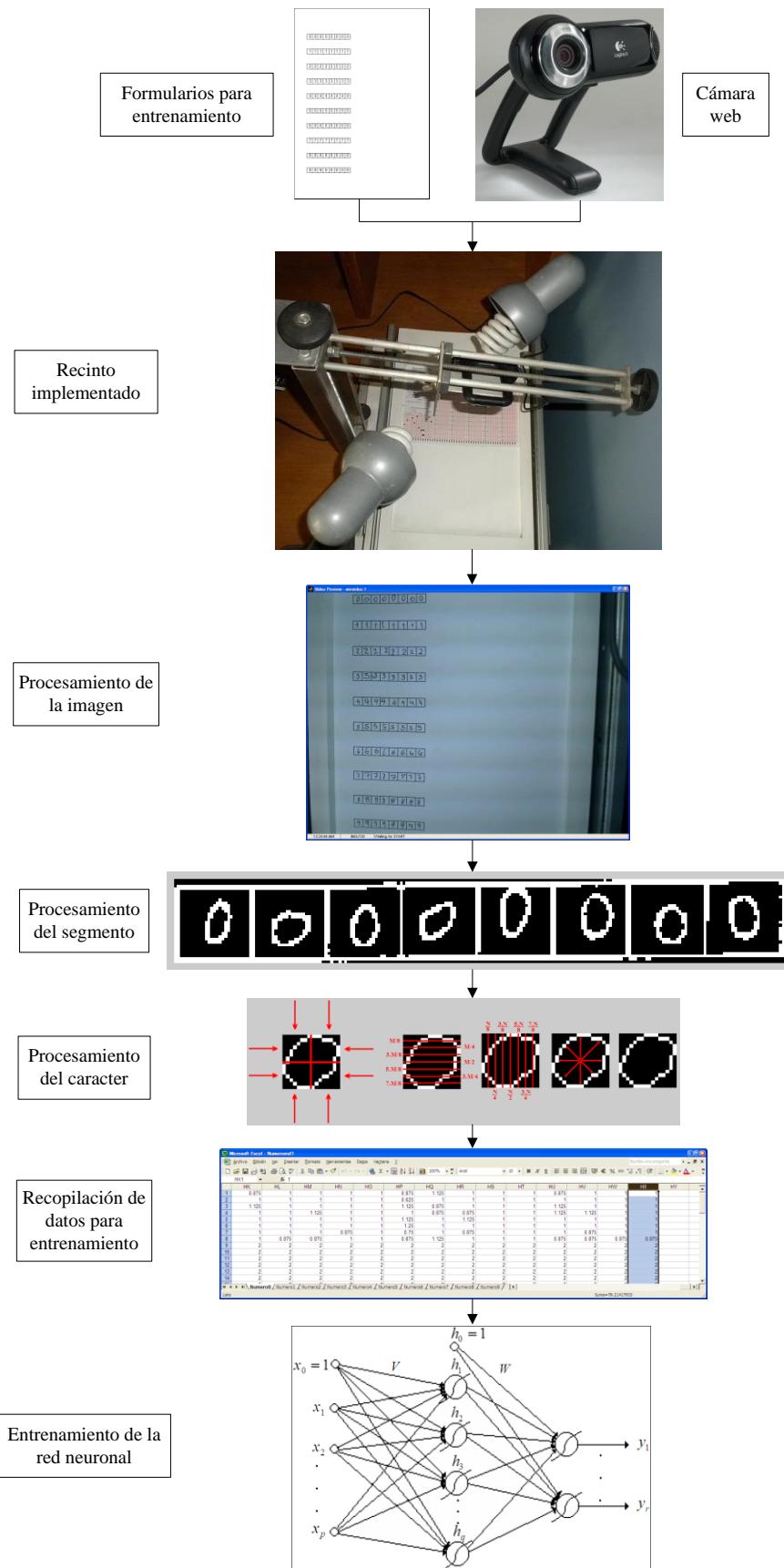


Figura 1.7.1 Diagrama pictórico de la fase de entrenamiento

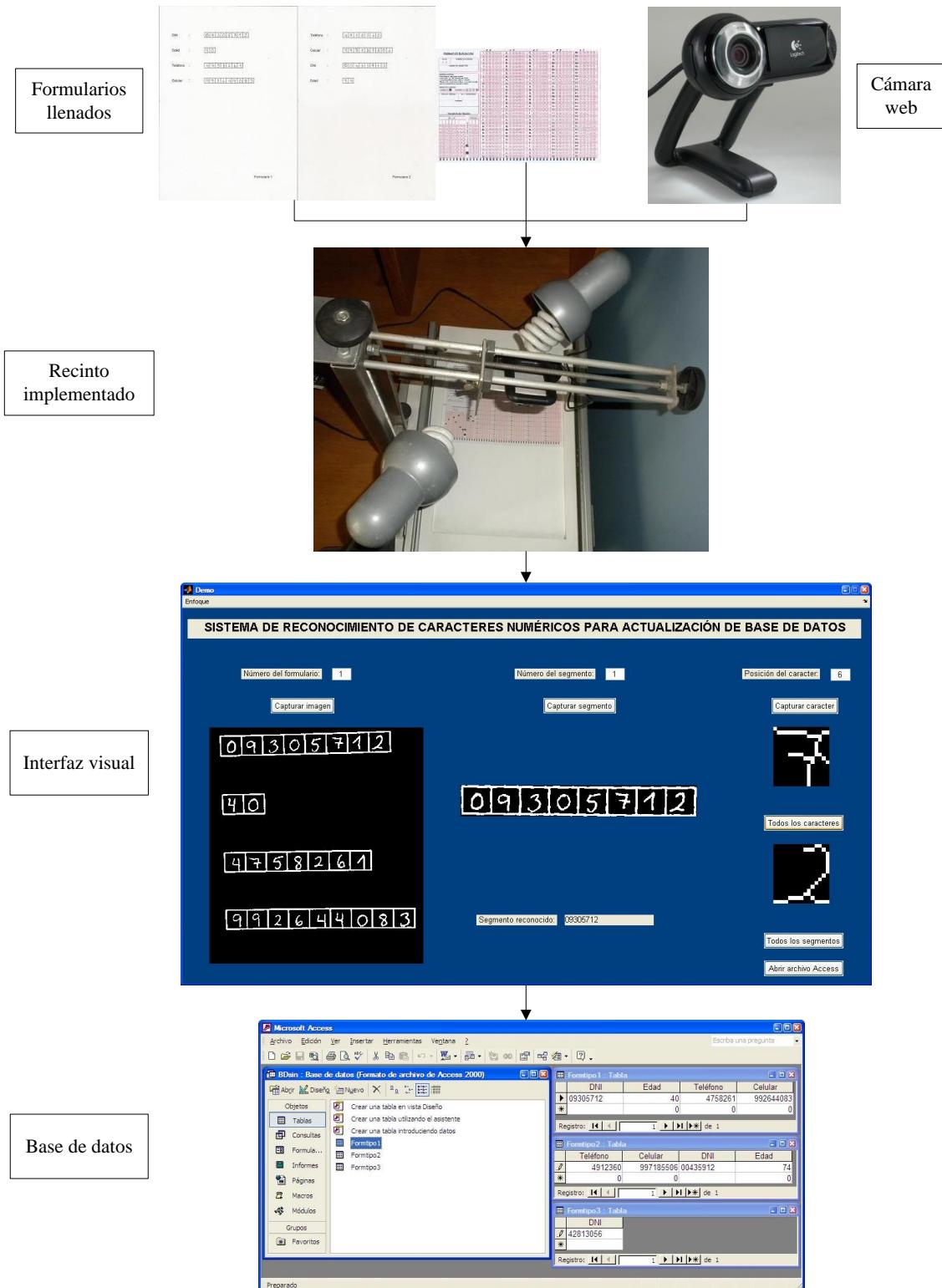


Figura 1.7.2 Diagrama pictórico de la fase de reconocimiento

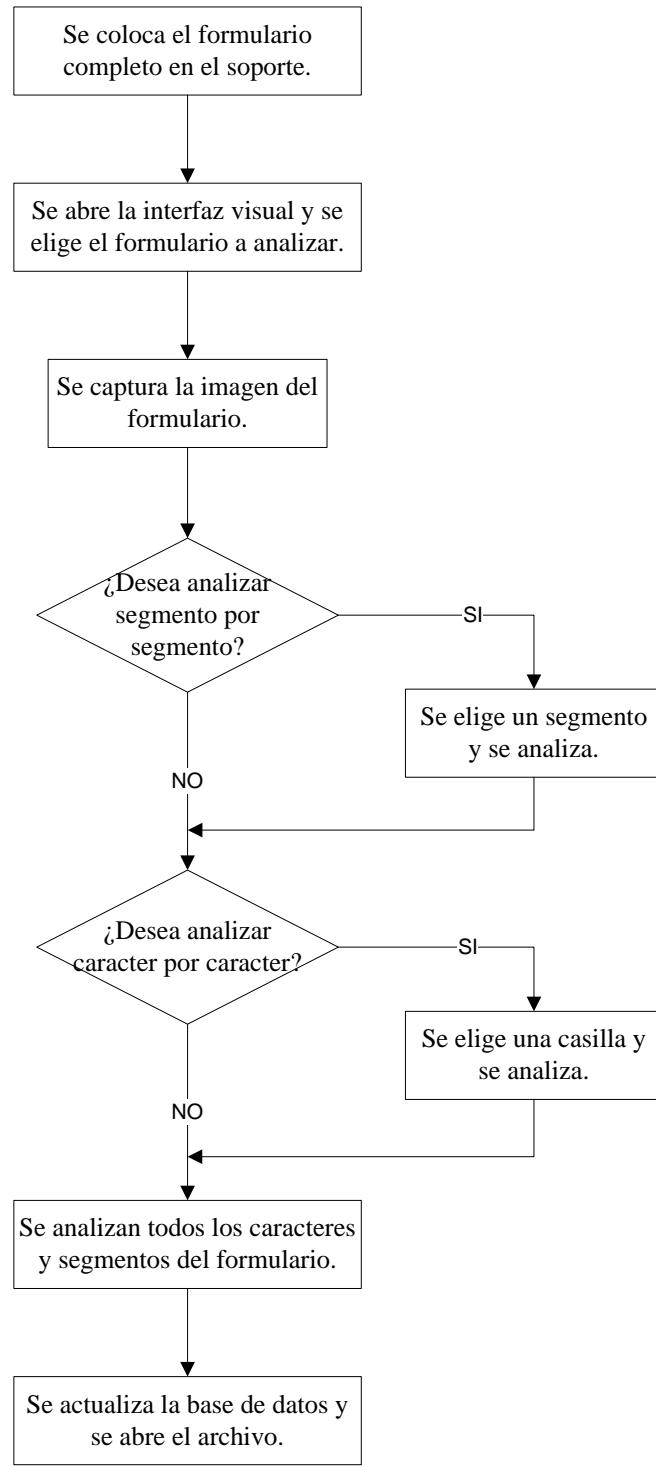


Figura 1.7.3 Diagrama de flujo de la interfaz visual

1.8. Comentarios finales

En el capítulo 1 se explicaron los aspectos introductorios del proyecto. La presentación y justificación del problema existente al cual se plantea una solución. Se describió el estado del arte realizado para una buena investigación previa al proyecto. Se incluyeron las ventajas y desventajas del proyecto con respecto a los demás sistemas investigados. Además, se identificaron las aplicaciones y las instituciones que se verían beneficiadas con el sistema propuesto. Se fundamentaron los objetivos y se identificaron los diferentes aspectos de la viabilidad del proyecto (técnica, económica, operativa, social y ambiental). Por último, se desarrolló una breve descripción del sistema diseñado. En el siguiente capítulo se verá el marco teórico necesario para la realización del proyecto.

CAPÍTULO 2. MARCO TEÓRICO

En el presente capítulo se explicará el marco teórico utilizado en el proyecto. Se describirán las teorías, principios y fundamentos científicos con los que se desarrolló el proyecto.

2.1. Iluminación blanca

La luz blanca ofrece mejores rendimientos al ojo humano, tanto en su gama de temperaturas de color como en su índice de reproducción cromática. Esta ofrece un mejor contraste de colores y ayuda en el reconocimiento de objetos²⁴.

Algunas ventajas de la luz blanca es que resalta los detalles y los colores oscuros. Utilizando menos watts brinda mayor luminosidad con lo que genera menos consumo. También, tiene una vida útil de 5 mil a 7 mil horas aproximadamente. Una ventaja importante es que su intensidad de luz es constante, no se pierde al pasar de las horas. Además, contamina menos ya que no contiene mercurio o gas²⁵.

2.2. Cámara digital

Las características más relevantes del sensor de una cámara digital son la cantidad de píxeles que es capaz de tomar y el tamaño de ellos. El píxel es la unidad mínima con un único color e intensidad en una imagen. Al ser más pequeño cada píxel y contar con mayor cantidad de ellos, mayor sería la resolución. Lo que significa que se tendría una

²⁴ Cfr. Ramos 2017.

²⁵ Cfr. Casa Fija 2017.

mejor calidad de la imagen. En el mercado, esto se indica como la cantidad de megapíxeles (Mp) que la cámara es capaz de tomar²⁶.

El sensor de una cámara digital está formado por millones de celdas fotosensibles, cada una de ellas de tamaño microscópico. Al indicar los megapíxeles de una cámara, se refiere a los millones de píxeles (celdas) que forman parte del sensor de la cámara. Por ejemplo una cámara de 12Mp tiene 12 millones de celdas fotosensibles. Al tomar una foto, cada píxel del sensor recoge información de la luz que le llega. Cada celda incluye un fotodiodo (elemento fotosensible) el cual recibe fotones que generan electrones al interactuar con los átomos del material. Por lo tanto, convierte esta luz en electricidad. Además, incluye la electrónica necesaria para que cada elemento funcione de forma independiente y con ello se pueda leer la información de cada píxel al tomar una foto. En la mayoría de los sensores actuales, cada celda incluye también un pequeño lente individual para concentrar la luz en la superficie sensible²⁷. En la Figura 2.2.1 se observan los elementos que componen un sensor.

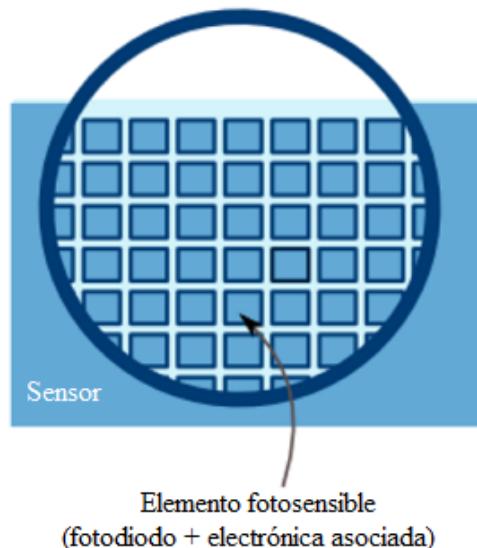


Figura 2.2.1 Elementos que componen un sensor²⁸

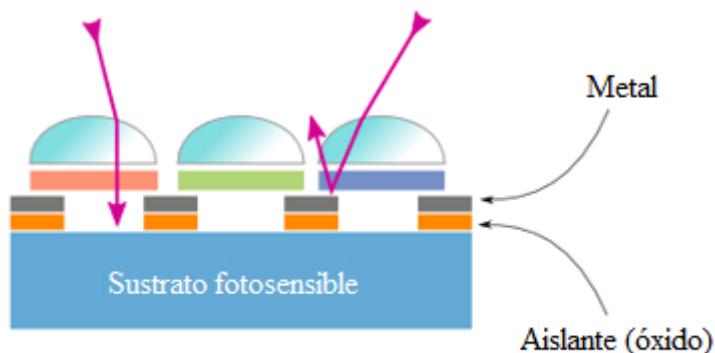
²⁶ Cfr. ElegirElectronica.com 2017.

²⁷ Cfr. Quecamarareflex.com 2015a.

²⁸ Cfr. Quecamarareflex.com 2015a.

Las celdas sólo pueden detectar la intensidad de la luz mas no el color. Los sensores incluyen filtros que dividen la luz en tres componentes, los cuales son el rojo, verde y azul. En la mayoría de los sensores se utiliza un filtro de Bayer, por lo que unas celdas reciben sólo la luz correspondiente a su componente rojo, otras sólo el componente azul y otras sólo el componente verde²⁹.

En la actualidad, la mayoría de sensores están basados en tecnología CMOS (siglas en inglés de Complementary Metal-Oxide-Semiconductor). Sobre el material fotosensible se añade la circuitería, compuesta por capas de zonas aislantes (óxidos) y metal. Esta fabricación consiste en expandir la circuitería hacia arriba, sobre el sustrato de silicio y sobre esta estructura electrónica se colocan el filtro de Bayer y los lentes³⁰. En la Figura 2.2.2 se observa la estructura CMOS.



CMOS: Complementary Metal-Oxide-Semiconductor

Figura 2.2.2 Estructura CMOS³¹

A continuación, se presentan algunas propiedades de una cámara digital y de las imágenes que se obtendrían con ella, con sus correspondientes definiciones:

- **Longitud focal:** Es la distancia que hay entre el sensor y el lente, cuando se enfoca a un sujeto al infinito. Esto sucede debido a que para poder enfocar al infinito el

²⁹ Cfr. Quecamarareflex.com 2015a.

³⁰ Cfr. Quecamarareflex.com 2015a.

³¹ Cfr. Quecamarareflex.com 2015a.

lente se desplaza lo más lejos posible del sensor³². En la Figura 2.2.3 se muestra una representación de la longitud focal. En la Figura 2.2.4 se observan diferentes imágenes que se han obtenido con la longitud focal indicada.

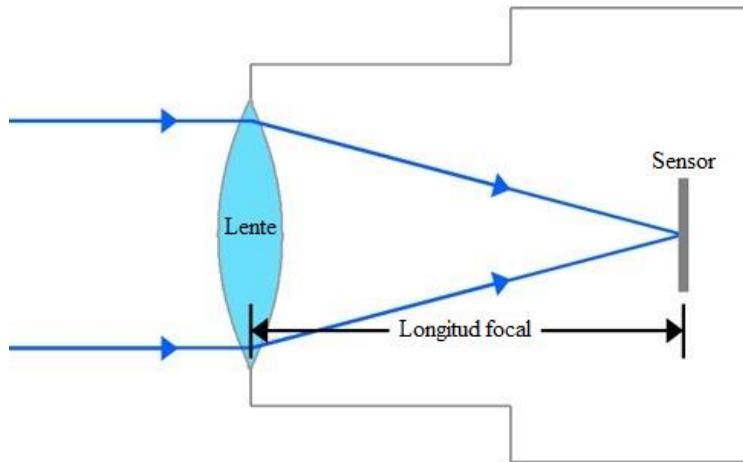


Figura 2.2.3 Representación de la longitud focal³³

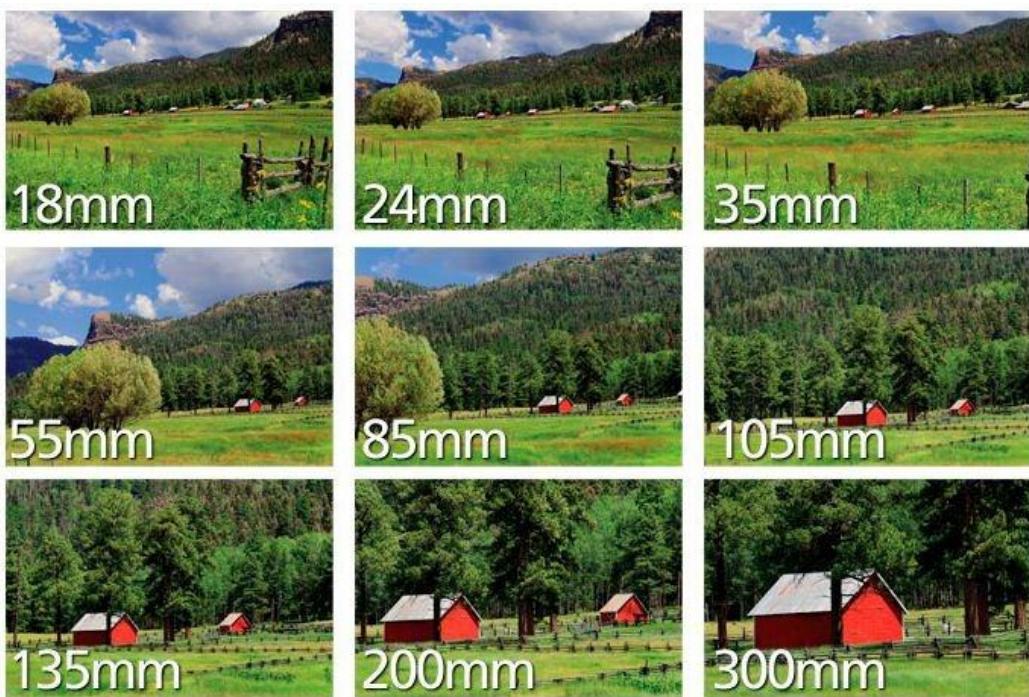


Figura 2.2.4 Imágenes obtenidas con la longitud focal respectiva³⁴

³² Cfr. Fotografía para principiantes 2012.

³³ Cfr. Fotografía para principiantes 2012.

- **Ángulo de visión:** Es la magnitud que determina la parte de la escena que es captada por el sensor. De acuerdo al tipo de lente que se utiliza es que varía el tamaño de la porción captada. El ángulo de visión de una cámara es función del tamaño de la superficie que registra la imagen y de la longitud focal³⁵.
- **Resolución óptica:** Indica la cantidad de detalle que se puede observar en una imagen. Se utiliza para describir qué tan nítida es una imagen. A mayor resolución, mejor se podrá ver el detalle de la imagen obtenida, se contará con una imagen de mejor calidad. La resolución de la imagen se define con dos números enteros, donde el primero es la cantidad de columnas de píxeles de la imagen y el segundo es la cantidad de filas. Además, describe el número total de píxeles en la imagen, el cual se expresa como el múltiplo correspondiente a millón (mega). Este se calcula al multiplicar la cantidad de columnas en una imagen por la cantidad de filas³⁶. En la Figura 2.2.5 se observa un ejemplo de una misma imagen en diferentes resoluciones.

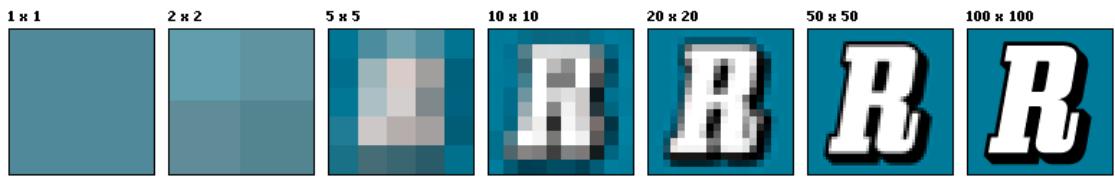


Figura 2.2.5 Ejemplo de una imagen en distintas resoluciones³⁷

- **Captura de imagen (4:3 SD o 16:9 W):** La relación de aspecto indica la proporción de anchura y altura de una imagen. Esta relación se expresa separando la proporción de la anchura con la altura por dos puntos. El formato horizontal 4:3 es la proporción estándar o definición estándar (SD por sus siglas en inglés) para cámaras digitales. El formato panorámico 16:9 es relacionado a la televisión y video en alta

³⁴ Cfr. Fotografía para principiantes 2012.

³⁵ Cfr. Wikipedia 2017a.

³⁶ Cfr. Wikipedia 2017b.

³⁷ Cfr. Wikipedia 2017b.

definición (HD por sus siglas en inglés)³⁸. Se cuenta con la opción de realizar la captura de la imagen en estos formatos.

- **Captura de video (4:3 SD o 16:9 W):** Es el caso similar con respecto a la relación de aspecto de la captura de imagen, pero aplicado a la captura de video.
- **Cuadros por segundo:** También conocido como fotogramas por segundo o FPS (siglas en inglés de frames per second). Se define como la velocidad a la cual un dispositivo muestra imágenes consecutivas (cuadros)³⁹.
- **Zoom óptico:** El acercamiento de la imagen se realiza por medio de la posición relativa de los lentes que conforman el objetivo. Por lo tanto, la imagen que llega al sensor sería efectivamente aumentada, manteniendo la resolución de la fotografía, no existen pérdidas⁴⁰.
- **Brillo:** También conocido como luminosidad o claridad. Es una propiedad de los colores que conforman una imagen. El brillo indica el aspecto luminoso del color analizado. Por ejemplo, si un color es muy oscuro, entonces su brillo es muy débil⁴¹.
- **Contraste:** Una imagen que cuenta con un buen contraste significa que se pueden distinguir muy bien sus bordes y las líneas que lo conforman. Las separaciones entre sus partes son muy nítidas y diferenciadas⁴².
- **Enfoque:** Se considera que un elemento de la escena se encuentra enfocado cuando los puntos de la imagen que corresponden a ese elemento son muy pequeños. Al observar una imagen, los elementos enfocados aparecen con mucho contraste. Por el

³⁸ Cfr. El paisaje perfecto 2013.

³⁹ Cfr. Wikipedia 2017c.

⁴⁰ Cfr. ElegirElectronica.com 2017.

⁴¹ Cfr. Wikipedia 2017d.

⁴² Cfr. Quecamarareflex.com 2015b.

contrario, los elementos que no están enfocados se muestran más borrosos y hasta pueden resultar irreconocibles⁴³.

El proceso de enfoque implica el movimiento de la lente con respecto al plano del sensor. En el caso de una lente simple, cuando se aleja la lente del sensor, se enfocan objetos más cercanos. Pero si se acerca la lente al sensor, se enfocan los objetos que se encuentran más lejos⁴⁴. En la Figura 2.2.6 se observan los cambios de enfoque con relación a la distancia de la lente al sensor.

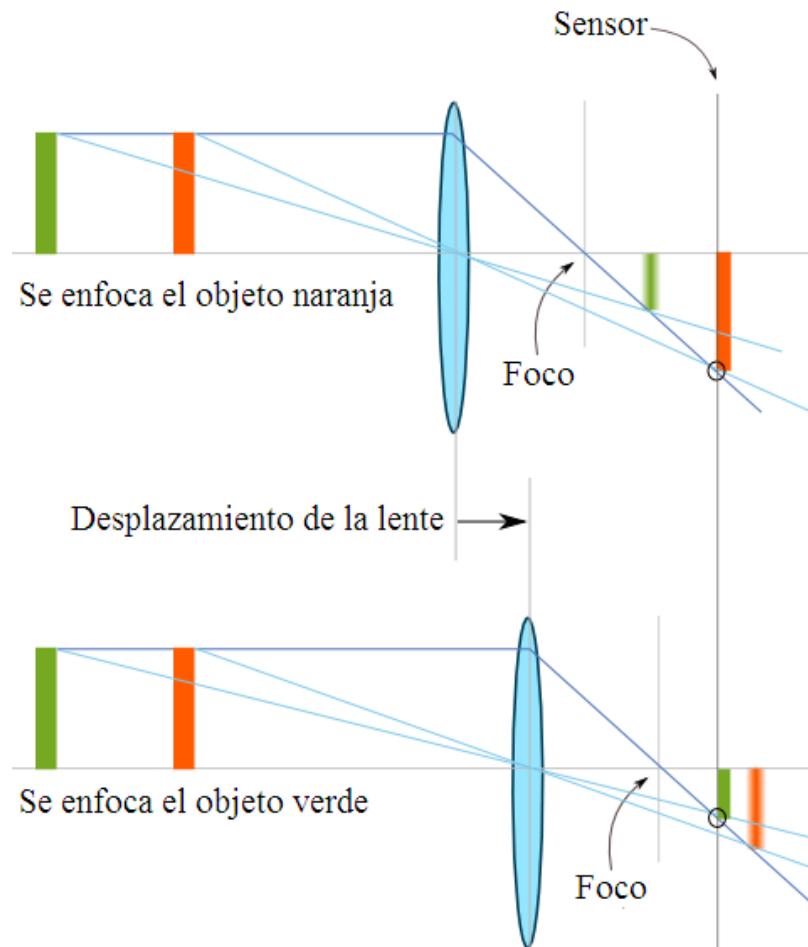


Figura 2.2.6 Cambios de enfoque⁴⁵

⁴³ Cfr. Quecamarareflex.com 2015b.

⁴⁴ Cfr. Quecamarareflex.com 2015b.

⁴⁵ Cfr. Quecamarareflex.com 2015b.

2.3. Función Signo

La función signo es una función definida a trozos o función por partes. Se requiere de varias fórmulas para poder representarla, cada una de las cuales establece el comportamiento de la función en un cierto fragmento. Esta función obtiene el signo de cualquier número real que tenga como entrada. La función signo se representa con la expresión (2.3.1) y se observa en la Figura 2.3.1.

$$\operatorname{sgn}(x_{SG}) = \begin{cases} 1 & \text{si } x_{SG} > 0 \\ 0 & \text{si } x_{SG} = 0 \\ -1 & \text{si } x_{SG} < 0 \end{cases} \quad (2.3.1)$$

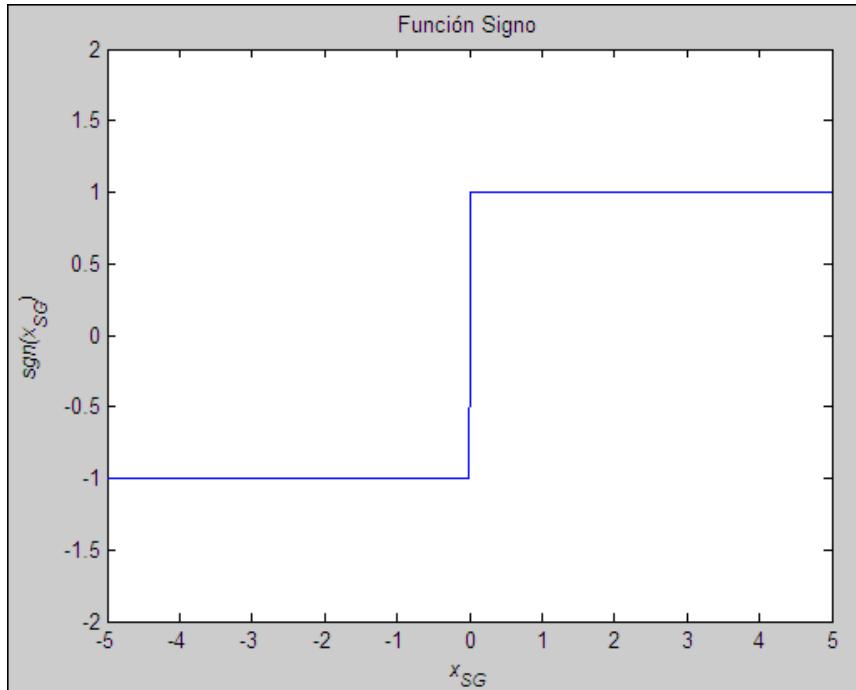


Figura 2.3.1 Función signo

2.4. Teoría de Imágenes

Una imagen se puede representar por una función bidimensional; por ejemplo $f_{ID}(x_{ID}, y_{ID})$, donde x_{ID} e y_{ID} son coordenadas espaciales, y la amplitud de f_{ID} se denomina intensidad o nivel de gris en ese par de coordenadas. La imagen es una imagen digital cuando x_{ID} , y_{ID} , y los valores de intensidad de f_{ID} son cantidades discretas. El campo del procesamiento digital de imágenes se refiere a su procesamiento por medio de una

computadora. Cabe resaltar que una imagen digital está compuesta por un número finito de elementos, cada uno integrado por una ubicación y un valor en específico. Estos se denominan elementos de imagen, pels o píxeles, siendo esta última la expresión más utilizada⁴⁶. Por lo tanto, el objetivo del procesamiento de imágenes es generar imágenes digitales a partir de datos detectados. Para ello se requiere realizar los siguientes dos procesos: muestreo y cuantificación⁴⁷.

Una imagen puede ser continua con respecto a sus coordenadas y en su amplitud. Para que se transforme a formato digital, se deben obtener muestras de la función en sus coordenadas y amplitud. La digitalización de los valores de coordenadas se denomina muestreo y la de los valores de amplitud es denominada cuantificación. Para muestrear una función, se toman muestras igualmente espaciadas a lo largo de una línea horizontal. El conjunto de estas ubicaciones discretas brinda como resultado la función muestreada. Sin embargo, los valores de las muestras todavía abarcan (verticalmente) un rango continuo de valores de intensidad. Para formar una función digital, estos valores también deben ser convertidos a cantidades discretas; es decir, deben ser cuantificados. Los niveles de intensidad continua se cuantifican asignando uno de los valores divididos a cada muestra. Esta asignación se realiza dependiendo de la proximidad vertical de una muestra a una marca vertical⁴⁸.

Por ejemplo, sea $f_{IC}(s_{IC}, t_{IC})$ una función de imagen continua de dos variables continuas, s_{IC} y t_{IC} . Esta función se transforma en una imagen digital por muestreo y cuantificación. Se considera que la imagen continua se muestrea en una matriz 2-D, $f_{ID}(x_{ID}, y_{ID})$, la cual está compuesta por M filas y N columnas, donde (x_{ID}, y_{ID}) son coordenadas discretas. Para una mejor notación, se utilizan valores enteros para estas coordenadas: $x_{ID} = 0, 1, 2, \dots, M-1$ e $y_{ID} = 0, 1, 2, \dots, N-1$. Así; por ejemplo, el valor de la imagen digital en el origen es $f_{ID}(0, 0)$ y el siguiente valor de coordenadas a lo largo de la primera fila es $f_{ID}(0, 1)$. En general, el valor de la imagen en cualquier coordenada

⁴⁶ Cfr. Gonzalez y Woods 2008: 1-2.

⁴⁷ Cfr. Gonzalez y Woods 2008: 52.

⁴⁸ Cfr. Gonzalez y Woods 2008: 52-53.

(x_{ID}, y_{ID}) se denota por $f_{ID}(x_{ID}, y_{ID})$, donde x_{ID} e y_{ID} son enteros. La sección del plano real abarcada por las coordenadas de una imagen se denomina dominio espacial, con x_{ID} e y_{ID} siendo referidas como coordenadas espaciales⁴⁹.

Las matrices numéricas se utilizan para el procesamiento y el desarrollo de algoritmos. En forma de ecuación, una matriz numérica $M \times N$ se representa como la expresión (2.4.1)⁵⁰.

$$f_{ID}(x_{ID}, y_{ID}) = \begin{bmatrix} f_{ID}(0,0) & f_{ID}(0,1) & \cdots & f_{ID}(0, N-1) \\ f_{ID}(1,0) & f_{ID}(1,1) & \cdots & f_{ID}(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_{ID}(M-1,0) & f_{ID}(M-1,1) & \cdots & f_{ID}(M-1, N-1) \end{bmatrix} \dots \dots \dots \quad (2.4.1)$$

Donde:

x_{ID}	= coordenada espacial (fila)
y_{ID}	= coordenada espacial (columna)
$f_{ID}()$	= intensidad o nivel de gris
M	= número total de filas
N	= número total de columnas

Ambos lados de esta ecuación son formas equivalentes de expresar en modo cuantitativo una imagen digital. El lado derecho es una matriz de números reales donde cada elemento se denomina píxel⁵¹.

En el modelo RGB, cada color aparece en sus componentes primarios de rojo (R), verde (G) y azul (B). Este modelo se basa en un sistema de coordenadas cartesianas. En la Figura 2.4.1 se muestra el cubo que es el subespacio de color de interés. En este cubo, los valores primarios RGB están en tres esquinas y los colores secundarios cian, magenta y amarillo están en otros tres vértices. Además, el color negro está en el origen (0, 0, 0) y el blanco está en la esquina más alejada del origen (1, 1, 1). La escala de grises se extiende desde el negro al blanco, a lo largo de la línea que los une, debido a que sus valores RGB son iguales. Los diferentes colores de este modelo son puntos

⁴⁹ Cfr. Gonzalez y Woods 2008: 55.

⁵⁰ Cfr. Gonzalez y Woods 2008: 56.

⁵¹ Cfr. Gonzalez y Woods 2008: 56.

sobre o dentro del cubo y se definen por vectores que se extienden desde el origen. Por comodidad, todos los valores de color se han normalizado por lo que el cubo mostrado es unitario. Esto significa que todos los valores de R, G y B se encuentran en el rango $[0,1]$ ⁵².

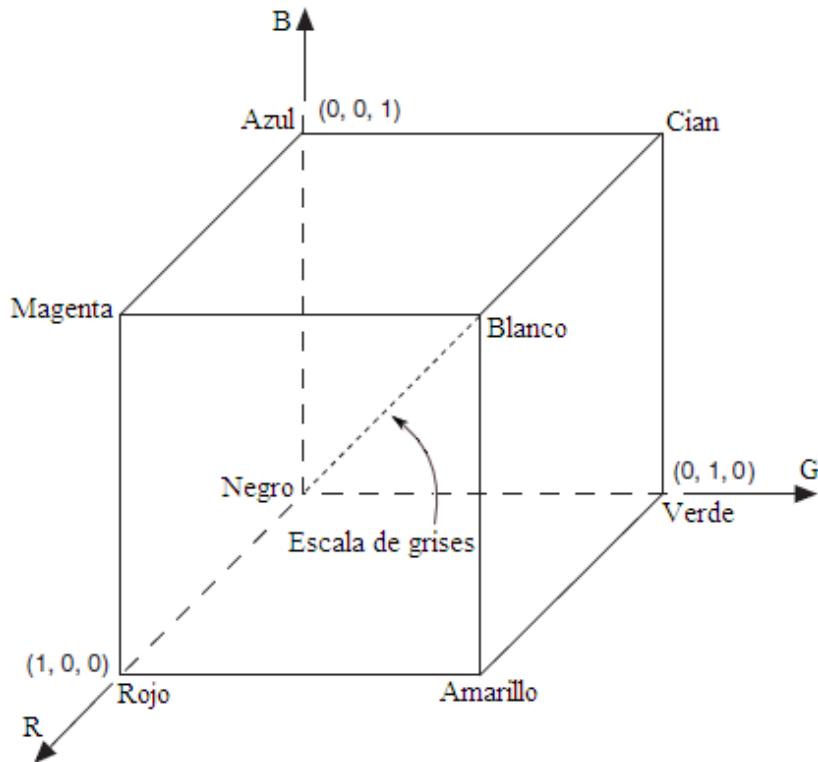


Figura 2.4.1 Esquema del cubo de color RGB⁵³

Las imágenes que son representadas por el modelo de color RGB están compuestas por tres componentes de imágenes, una para cada color primario. Al introducirse en un monitor RGB, estas tres imágenes se combinan en la pantalla para generar una imagen compuesta de color⁵⁴. Las imágenes en escala de grises tienen la particularidad de que estos tres componentes de imágenes de las que están compuestas cuentan con los

⁵² Cfr. Gonzalez y Woods 2008: 402.

⁵³ Cfr. Gonzalez y Woods 2008: 402.

⁵⁴ Cfr. Gonzalez y Woods 2008: 403.

2.7. Interpolación Bicúbica

La interpolación es el proceso matemático que consiste en calcular nuevos datos a partir de los disponibles, pretendiendo que se comparta la misma forma. Es el proceso de usar datos conocidos para estimar valores en ubicaciones desconocidas⁵⁸. En imágenes digitales, significa que se crean nuevos píxeles a partir de los existentes. Por lo tanto, la imagen resultante representaría lo mismo que la imagen original, sólo que con más píxeles. La interpolación se aplica en diversos procesos relacionados con las imágenes digitales. Por ejemplo, la rotación de una imagen, deformación o aumentar su resolución.

En la interpolación bicúbica se considera la vecindad de los 4x4 píxeles conocidos más cercanos (un total de 16 píxeles). Se tienen los cuatro píxeles que están más próximos a las esquinas en diagonal y también sus puntos más cercanos. Por estar localizados a distintas distancias del píxel de valor desconocido, en el cálculo se da mayor peso a los píxeles más cercanos. El valor de intensidad asignado al punto (x_B, y_B) se obtiene utilizando la ecuación (2.7.1), donde los dieciséis coeficientes se determinan a partir de las dieciséis ecuaciones en dieciséis incógnitas que se pueden escribir usando los dieciséis vecinos más cercanos del punto (x_B, y_B) ⁵⁹. En la Figura 2.7.1 se muestran los píxeles a considerar para la interpolación bicúbica, representados con el color rojo.

$$v_B(x_B, y_B) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x_B^i y_B^j \quad \dots \quad (2.7.1)$$

Donde:	x_B	= coordenada espacial (fila)
	y_B	= coordenada espacial (columna)
	a	= coeficiente
	$v_B()$	= intensidad

⁵⁸ Cfr. Gonzalez y Woods 2008: 65.

⁵⁹ Cfr. Gonzalez y Woods 2008: 66.

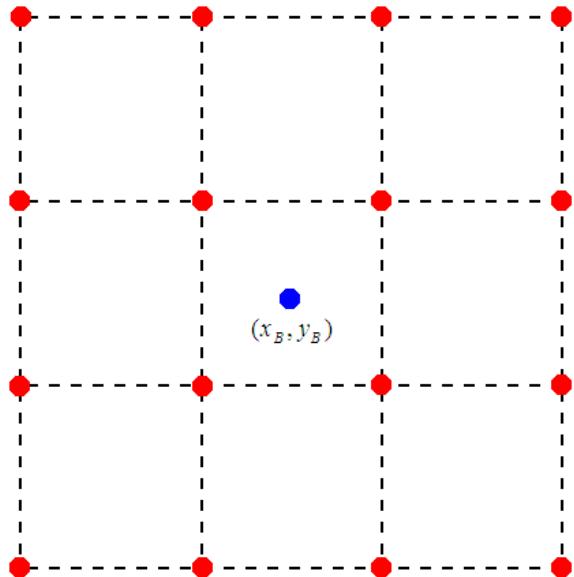


Figura 2.7.1 Píxeles considerados para la interpolación bicúbica

Por lo general, la interpolación bicúbica hace un mejor trabajo de preservar el detalle fino⁶⁰. Se obtiene como resultado imágenes más nítidas que con otros métodos de interpolación. Sin embargo, es un buen compromiso entre tiempo de procesado y calidad de resultado.

2.8. Distancia euclíadiana

La distancia euclíadiana entre dos píxeles pertenecientes a una imagen, representada por una matriz, se calcula con la expresión (2.8.1), donde un píxel tiene la posición (x_{E_1}, y_{E_1}) y el otro se encuentra en la posición (x_{E_2}, y_{E_2}) .

$$d_E = \sqrt{(x_{E_1} - x_{E_2})^2 + (y_{E_1} - y_{E_2})^2} \quad \dots \dots \dots \quad (2.8.1)$$

En el presente proyecto se calcularán las distancias euclidianas desde el centro de la matriz al primer píxel blanco que se encuentre en diversas direcciones definidas. La

⁶⁰ Cfr. Gonzalez y Woods 2008: 66.

distancia euclíadiana se obtiene con la expresión (2.8.2), donde el centro de la matriz es el píxel en la posición (i, j) y el píxel blanco encontrado tiene la posición (x_E, y_E) .

$$d_E = \sqrt{(x_E - i)^2 + (y_E - j)^2} \quad \dots \dots \dots (2.8.2)$$

2.9. Funciones de Matlab

En esta sección se explicarán las funciones de Matlab utilizadas en el presente proyecto para la etapa del procesamiento de los caracteres.

2.9.1. Función bwmorph

La función `bwmorph` está definida como operaciones morfológicas específicas utilizadas en imágenes binarias. Se tiene la opción de establecer cuántas veces se desea aplicar la operación en la imagen ingresada. Esta cantidad de veces puede ser indicada como '`Inf`', en cuyo caso la operación elegida se repite hasta que la imagen ya no cambie más⁶¹.

La operación utilizada en el proyecto es '`thin`'. Al elegir '`Inf`' en la cantidad de veces para aplicar la operación, esta adelgaza los objetos a solo líneas. Se encarga de eliminar los píxeles para que un objeto sin agujeros se contraiga a un trazo mínimamente conectado. Sin embargo, en el caso de un objeto con agujeros, se encoge a un anillo conectado a medio camino entre cada agujero y el límite exterior⁶².

2.9.2. Función bwtraceboundary

La función `bwtraceboundary` está definida como el trazo del contorno de un objeto en una imagen binaria. Se debe tener en cuenta que los píxeles distintos de cero forman

⁶¹ Cfr. The Mathworks Inc. 2016.

⁶² Cfr. The Mathworks Inc. 2016.

parte de un objeto y los píxeles de valor cero pertenecen al fondo. Se debe ingresar como vector las coordenadas espaciales (fila y columna) del píxel que se encuentra en el contorno del objeto, desde el cual se desea que empiece el trazado. Además, se debe indicar la dirección inicial de búsqueda para el siguiente píxel que pertenezca al objeto y se encuentre conectado al píxel inicial. Se ingresan valores como 'N' que indica el norte o 'NE' para el noreste⁶³. En la Figura 2.9.2.1 se muestran las diferentes direcciones de búsqueda que se pueden ingresar a la función.

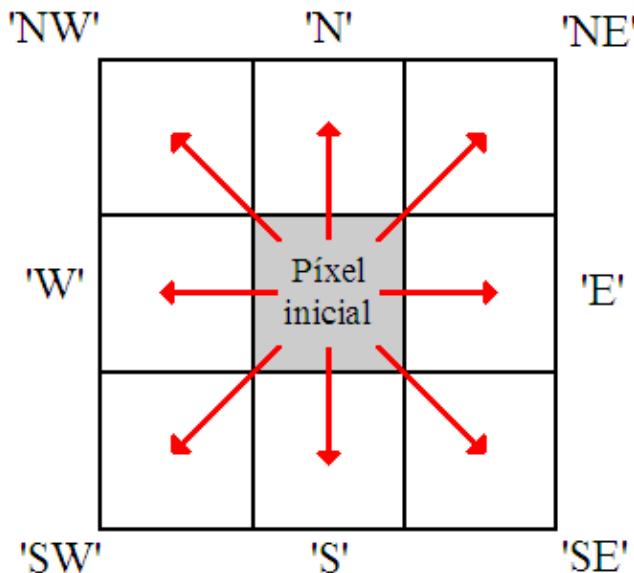


Figura 2.9.2.1 Direcciones de búsqueda⁶⁴

Además, se tiene la opción de elegir la conectividad a ser utilizada para trazar el contorno de la imagen. Por ejemplo, se puede indicar que se desea la vecindad de cuatro píxeles conectados. Por lo tanto, las direcciones de búsqueda estarían limitadas a 'N' (norte), 'E' (este), 'S' (sur) y 'W' (oeste). En cambio, si se indicase la vecindad de ocho píxeles conectados, se generaría la búsqueda en las direcciones indicadas en la figura anterior. También se puede ingresar a la función el número máximo de píxeles en el contorno que se desea extraer y la dirección en que se prefiera trazar el contorno. La función identifica todos los píxeles en el contorno cuando se le ingresa el valor de 'Inf' y

⁶³ Cfr. The Mathworks Inc. 2016.

⁶⁴ Cfr. The Mathworks Inc. 2016.

La cantidad de puntos en el contorno se representa con L y k tiene que cumplir la condición (2.10.3)⁶⁸.

$$0 \leq k \leq L - 1 \quad \dots \quad (2.10.3)$$

Posteriormente, se elimina el primer valor porque es el valor DC. Para obtener la característica de los descriptores de Fourier de ser insensibles a la traslación y a la rotación se aplica la expresión (2.10.4)⁶⁹.

$$r_D(k) = \sqrt{|a_D(k)|^2 + |b_D(k)|^2} \quad \dots \quad (2.10.4)$$

Además, para que los descriptores de Fourier sean insensibles al cambio de escala se aplica la expresión (2.10.5)⁷⁰.

$$S(k) = r_D(k) / r_D(1) \quad \dots \quad (2.10.5)$$

El vector $S(k)$ es simétrico a partir de la posición $(L-1)/2$ y el primer valor siempre es igual a 1; por lo tanto, se recorta el vector desde la posición 2 hasta la posición $(L-1)/2$. Los primeros valores de los descriptores de Fourier indican la forma general del carácter y los últimos indican los pequeños detalles.

2.11. Redes Neuronales

La neurona artificial estándar es un elemento de procesamiento que calcula una salida multiplicando su vector de entradas por un vector de pesos y este resultado es aplicado a una función de activación. En la Figura 2.11.1 se observa la representación de una neurona artificial⁷¹.

⁶⁸ Cfr. Phokharatkul y Kimpan 2002: 281; Chung y Wong 1997: 392.

⁶⁹ Cfr. Phokharatkul y Kimpan 2002: 282; Chung y Wong 1997: 392.

⁷⁰ Cfr. Phokharatkul y Kimpan 2002: 282; Chung y Wong 1997: 392.

⁷¹ Cfr. Arnáez 2007.

un correcto comportamiento de la red neuronal ante datos de entrada nuevos al completar su entrenamiento.

2.11.1. Características de las Redes Neuronales

La red neuronal es una red interconectada que se parece al cerebro humano. Una de sus características más importantes es su capacidad de aprender. En el caso que se presente un conjunto de entrenamiento donde se conocen valores de entrada y salida, se podría crear un modelo que ayude a clasificar nuevos datos. Los resultados que se logran mediante su uso son alentadores, especialmente en algunos campos como el reconocimiento de patrones. El backpropagation es el algoritmo más popular que se utiliza⁷⁴.

Las redes neuronales se utilizan cuando las reglas o criterios para encontrar una solución no son claros. Por esta razón, a veces se les conoce como caja negra, porque pueden resolver un problema pero es complicado de explicar cómo se resolvió. Algunas ventajas de las redes neuronales son que pueden adaptarse a nuevos escenarios, son tolerantes a fallos y pueden manejar datos con ruido. La mayor desventaja podría ser el tiempo que utiliza para su entrenamiento. Además, se requiere de una gran cantidad de muestras para que el entrenamiento sea eficiente. Por último, es complejo de explicar lo que está sucediendo dentro de una red neuronal⁷⁵.

Las redes neuronales cuentan con las siguientes características más resaltantes⁷⁶:

- Son dispositivos no-lineales.
- Pueden aprender un mapeo.
- Son adaptables.

⁷⁴ Cfr. Cilimkovic.

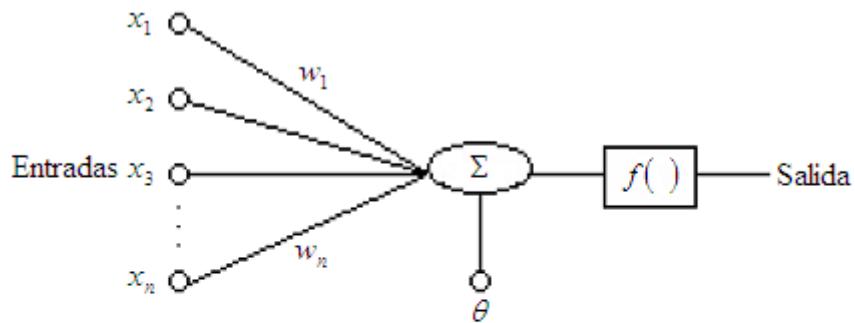
⁷⁵ Cfr. Cilimkovic.

⁷⁶ Cfr. Arnáez 2007.

- Las respuestas están basadas en evidencia.
- Usan contexto, es decir, a más información, la respuesta es más veloz y mejor.
- Son tolerantes a fallas, o sea, que la falla es degradada.
- Su diseño y análisis es uniforme.
- Tienen analogía neuro-biológica.
- Su procesamiento es masivamente paralelo, distribuido y realizado con operaciones individuales simples.
- Aprenden por sí mismas.
- Tienen capacidad de generalizar.

2.11.2. Perceptrón de Múltiple Capa (MLP)

La capa más simple de una sola neurona es el perceptrón y está representado en la Figura 2.11.2.1⁷⁷.



Donde: $y_{(t)} = \Phi_{(v)} = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases}$

$$v = \sum_{i=1}^n x_i w_i$$

Figura 2.11.2.1 Perceptrón⁷⁸

⁷⁷ Cfr. Arnáez 2007.

⁷⁸ Cfr. Arnáez 2007.

El perceptrón puede representar solamente las funciones que sean linealmente separables, para lo cual su funcionamiento queda reducido notablemente conforme se va incrementando el número de bits de las entradas. Esto se observa en la Tabla 2.11.2.1⁷⁹.

Tabla 2.11.2.1 Número de funciones linealmente separables⁸⁰

Número de bits	Cantidad de funciones	Funciones Linealmente Separables	Porcentaje de trabajo de Perceptrón (%)
2	16	14	87
3	256	104	40
4	65536	1882	3
5	10E+09	94572	0.00002

Los perceptrones de múltiple capa hacen posible resolver problemas no-linealmente separables, con unidades lineales agrupadas en múltiples capas. La configuración más utilizada del MLP es la red neuronal de dos capas, donde la primera capa tiene unidades no-lineales con funciones de activación sigmoides y la segunda con funciones de activación lineales. Es necesario que las entradas y las salidas sean escaladas dentro del rango de trabajo de las funciones de activación de las unidades no-lineales. El algoritmo de entrenamiento es el Backpropagation. En la Figura 2.11.2.2 se muestra esta estructura del MLP⁸¹.

⁷⁹ Cfr. Arnáez 2007.

⁸⁰ Cfr. Arnáez 2007.

⁸¹ Cfr. Arnáez 2007.

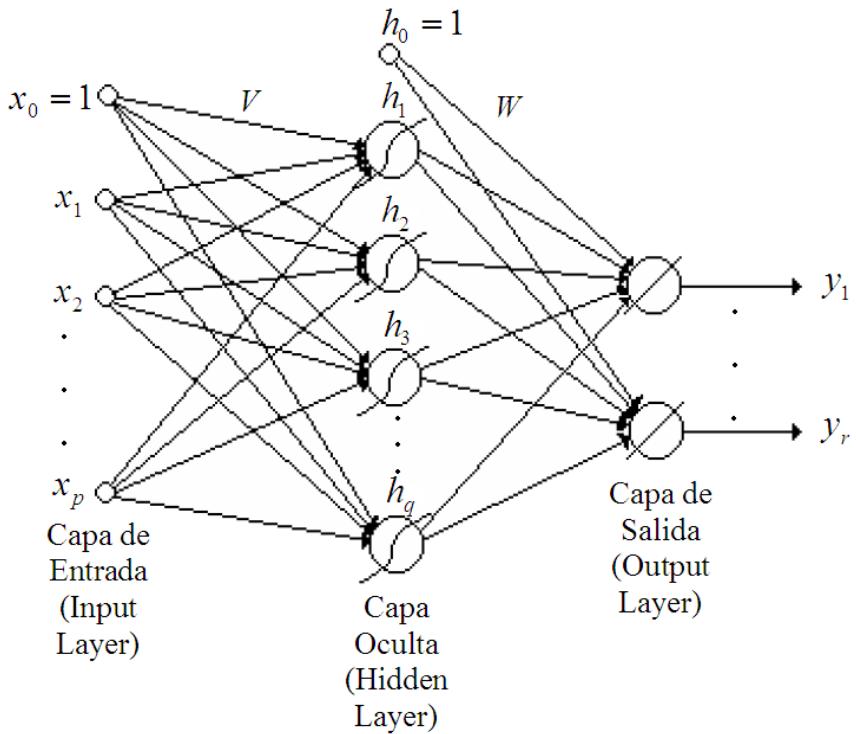


Figura 2.11.2.2 Estructura del MLP⁸²

El criterio para seleccionar la cantidad de unidades en la capa oculta es comenzar por un número similar al mayor entre la cantidad de entradas y de salidas. Luego se entrenará y se buscará un comportamiento adecuado. Posteriormente se prepara una red con una unidad más y se vuelve a entrenar. Si el compromiso entre la cantidad de neuronas o unidades de la capa oculta y la cantidad de ciclos o etapas de entrenamiento no es adecuada, se vuelve a entrenar. Cuando el entrenamiento no es sustancialmente diferente, se deja la cantidad de neuronas anterior y se puede decir que ha llegado a obtener el mejor número de unidades ocultas⁸³.

⁸² Cfr. Arnáez 2007.

⁸³ Cfr. Arnáez 2007.

2.11.3. Backpropagation

La idea principal en este algoritmo es que los errores de las unidades de la capa oculta se determinen mediante la propagación posterior (backpropagation) de los errores de las unidades de la capa de salida. El backpropagation también se puede considerar para funciones de activación no lineal y redes de múltiple capa. Cabe resaltar que si se utilizan funciones de activación lineal, una red de múltiple capa no es más potente que una red de una sola capa⁸⁴.

Una red neuronal que se alimenta mediante su avance se compone de una estructura en capas. Cada capa cuenta con unidades que reciben su entrada de las unidades que se encuentran en una capa directamente debajo y envían su salida a las unidades de la capa que se localizan directamente encima. En una misma capa no existen conexiones. La activación de una unidad oculta es una función de las entradas ponderadas más un sesgo (bias). La salida de las unidades ocultas se distribuye en la siguiente capa oculta hasta la última capa de unidades ocultas, de las cuales sus salidas se introducen en la capa de salida⁸⁵.

En el presente trabajo se utilizará la configuración del MLP de dos capas. En esta configuración, la primera capa tiene unidades no-lineales con funciones de activación sigmoides, al igual que la segunda capa. En la Figura 2.11.3.1 se observa la estructura del MLP utilizado en el proyecto.

⁸⁴ Cfr. Kröse y Van der Smagt 1996: 33.

⁸⁵ Cfr. Kröse y Van der Smagt 1996: 33.

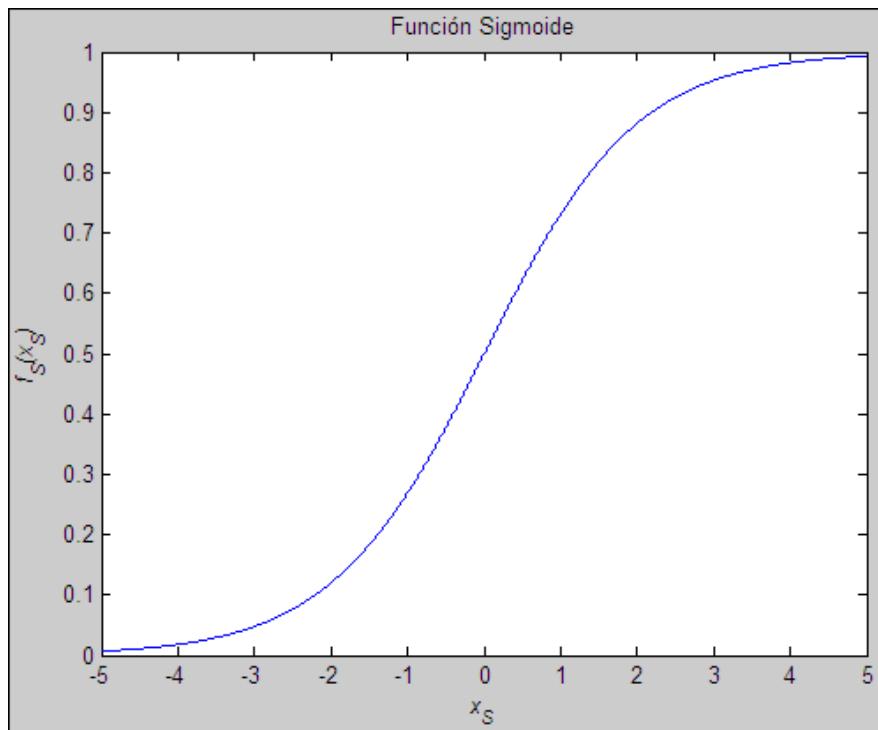


Figura 2.11.3.2 Función de activación sigmoide

$$y = \Phi_{(y^-)} \dots \quad (2.11.3.2)$$

$$y^- = Wh^+ \dots \quad (2.11.3.3)$$

$$h^+ = \Phi_{(h^-)} \dots \quad (2.11.3.4)$$

$$h^- = Vx \dots \quad (2.11.3.5)$$

- Donde:
- x = vector de entradas
 - h^- = punto antes de entrar a la capa oculta h
 - h^+ = punto después de la salida de la capa oculta h
 - $\Phi()$ = función de activación sigmoide
 - V = vector de pesos de la capa oculta
 - W = vector de pesos de la capa de salida
 - y^- = punto antes de entrar a la capa de salida y
 - y = vector de salida

Además, la función para realizar la actualización de los parámetros del algoritmo de Backpropagation se encuentra expresada en (2.11.3.6)⁸⁷.

$$p_{(i+1)} = p_{(i)} + \eta \frac{\partial J}{\partial p} \quad \dots \dots \dots \quad (2.11.3.6)$$

Donde: p = parámetro a actualizar
 η = factor de aprendizaje
 J = función de costos

El parámetro a actualizar es p , η es el factor de aprendizaje y J es la función de costos. La función de costos utilizada es $J=E$, la cual es el error cuadrático medio. En las ecuaciones (2.11.3.7) y (2.11.3.8) se representa la función de costos, donde y_d es la salida deseada⁸⁸.

$$E = \frac{1}{2} e^2 \quad \dots \dots \dots \quad (2.11.3.7)$$

$$e = y_d - y \quad \dots \dots \dots \quad (2.11.3.8)$$

Donde: E = error cuadrático medio
 e = error
 y_d = salida deseada
 y = salida actual

Como primer paso, se debe aplicar el algoritmo de Backpropagation para el peso W . Por lo tanto, la expresión generada a partir de la función de actualización sería la (2.11.3.9)⁸⁹.

$$W_{(i+1)} = W_{(i)} + \eta \frac{\partial J}{\partial W} \quad \dots \dots \dots \quad (2.11.3.9)$$

⁸⁷ Cfr. Arnáez 2007.

⁸⁸ Cfr. Arnáez 2007.

⁸⁹ Cfr. Arnáez 2007.

La derivada parcial se puede descomponer por la regla de la cadena en la expresión (2.11.3.10)⁹⁰.

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial E} \frac{\partial E}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial y^-} \frac{\partial y^-}{\partial W} \dots \quad (2.11.3.10)$$

Además, es de conocimiento el resultado de cada una de las derivadas parciales. Estas se muestran en las ecuaciones de (2.11.3.11) a (2.11.3.15)⁹¹.

$$\frac{\partial J}{\partial E} = 1 \dots \quad (2.11.3.11)$$

$$\frac{\partial E}{\partial e} = \frac{1}{2} \cdot 2e = e \dots \quad (2.11.3.12)$$

$$\frac{\partial e}{\partial y} = -1 \dots \quad (2.11.3.13)$$

$$\frac{\partial y}{\partial y^-} = y(1-y) \dots \quad (2.11.3.14)$$

$$\frac{\partial y^-}{\partial W} = h^+ = \Phi_{(Vx)} \dots \quad (2.11.3.15)$$

Por lo tanto, al reemplazar estas ecuaciones en la expresión (2.11.3.10) se obtendría como resultado la expresión (2.11.3.16)⁹².

$$\frac{\partial J}{\partial W} = -e \cdot y \cdot [1-y] \cdot \Phi_{(Vx)} \dots \quad (2.11.3.16)$$

De esta manera, la actualización del peso W sería representada por (2.11.3.17)⁹³.

⁹⁰ Cfr. Arnáez 2007.

⁹¹ Cfr. Arnáez 2007.

⁹² Cfr. Arnáez 2007.

botones y controles deslizantes. Además, se pueden crear aplicaciones personalizadas propias para que otras personas las utilicen¹⁰¹.

En la interfaz visual GUIDE (entorno de desarrollo de GUI) se pueden diseñar interfaces para aplicaciones personalizadas. Este interfaz cuenta con un editor de diseño para generar gráficamente la interfaz de usuario. El código Matlab para construir la interfaz es generado automáticamente por GUIDE, el cual se puede modificar para programar el detalle y procedimiento de la aplicación¹⁰².

2.14. Microsoft Access

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados para su posterior uso. Por ejemplo, en casa se suele tener un catálogo de libros, este se puede guardar en una base de datos para buscar luego un libro con facilidad. También, se puede almacenar en una base de datos las recetas de cocina y con ello poder buscar la que se necesite. La mayoría de las bases de datos se encuentran en formato digital, por lo que sus aplicaciones más frecuentes son para la gestión de empresas o instituciones públicas.

Microsoft Access es un sistema gestor de bases de datos. Es una interfaz atractiva y fácil de usar. Como primer paso, se debe escribir en el cuadro de texto “Aregar tablas” y se eligen las tablas adecuadas para definir los campos, relaciones y reglas entre ellas, de ser requerido¹⁰³. El programa Microsoft Access tiene la capacidad de agregar nuevos datos a una base de datos, modificar datos existentes en la base de datos, eliminar información, organizar y ver los datos de diferentes formas, y compartir los datos con otras personas mediante informes¹⁰⁴.

¹⁰¹ Cfr. The Mathworks Inc. 2017.

¹⁰² Cfr. The Mathworks Inc. 2017.

¹⁰³ Cfr. Microsoft Office 2017a.

¹⁰⁴ Cfr. Microsoft Office 2017b.

2.14.1. Tablas

Una base de datos computarizada es un contenedor de objetos y puede contener más de una tabla. Una tabla tiene una apariencia semejante a la de una hoja de cálculo, considerando que los datos se almacenan en filas y columnas. La principal diferencia en almacenar los datos es la forma en la que se organizan estos. Una propiedad de una base de datos es su flexibilidad, por lo que los datos se deben organizar en tablas para evitar redundancias¹⁰⁵.

Una tabla está compuesta por registros (filas) en los cuales se almacena la información. Cada registro está formado por uno o varios campos (columnas). Estos últimos se deben designar con un determinado tipo de datos. Por ejemplo, texto, fecha, hora o número. El tipo de datos de un campo indica la clase de datos almacenados en él, como grandes cantidades de texto o archivos adjuntos¹⁰⁶.

En una base de datos de Access, las propiedades de tabla son atributos que afectan su apariencia o su comportamiento. Estas se establecen en la hoja de propiedades de la tabla en la vista Diseño. Un escenario similar se observa con las propiedades de campo y se pueden establecer algunas de ellas en Vista de hoja de datos. También se puede realizar en la vista Diseño por medio del panel Propiedades del campo¹⁰⁷.

Para crear una tabla, se puede crear una base de datos nueva, insertar una tabla en una base de datos existente o importar o establecer un vínculo a una tabla de otro origen de datos. Al crear una nueva base de datos en blanco, se inserta automáticamente una nueva tabla vacía. Posteriormente, se ingresan los datos en la tabla para empezar a definir los campos¹⁰⁸.

¹⁰⁵ Cfr. Microsoft Office 2017b.

¹⁰⁶ Cfr. Microsoft Office 2017b.

¹⁰⁷ Cfr. Microsoft Office 2017b.

¹⁰⁸ Cfr. Microsoft Office 2017b.

2.15. Comentarios finales

En el capítulo 2 se explicó el marco teórico utilizado en el proyecto. Se describieron todas las teorías, principios y fundamentos científicos que se aplicaron para desarrollar el sistema propuesto. Este capítulo es la parte de teoría que explica cómo opera cada procesamiento y en los demás capítulos se debe evidenciar cómo se aplica. Todas las técnicas que se describen en el marco teórico se explicarán en los demás capítulos del libro. En el siguiente capítulo se describirá detalladamente el hardware del proyecto.

CAPÍTULO 3. DESCRIPCIÓN DEL HARDWARE DEL PROYECTO

En el presente capítulo se explicará el hardware del proyecto. Se describirá la arquitectura y funcionamiento general del hardware con los que se desarrolló el proyecto.

En el proyecto se utiliza una cámara web para tomar las fotos a los formularios llenados. La cámara web es una Logitech QuickCam Pro 9000¹⁰⁹. En la Figura 3.1 se muestra la cámara web utilizada. En la Tabla 3.1 se observan sus especificaciones y dimensiones.



Figura 3.1 Cámara web¹¹⁰

¹⁰⁹ Cfr. Logitech 2016.

¹¹⁰ Cfr. Logitech 2016.

Tabla 3.1 Especificaciones de la cámara web¹¹¹

Especificaciones:	
Tipo de Conexión	USB con cable
Tipo de USB	Alta velocidad USB 2.0, UVC
USB VID_PID	VID_046D&PID_0990
Micrófono	Incorporado, Cancelación de ruido
Tipo de Sensor y Lente	Vidrio, CMOS
Tipo de Enfoque	Automático
Campo de Visión (Field of View FOV)	75° Diagonal FOV
Longitud Focal	2 mm
Resolución Óptica (Verdadero)	2 MP (1600x1200)
Captura de Imagen (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (JPG - Verdadero) 3 MP (2048x1536), 4 MP (2304x1728), 5 MP (2592x1944), 8 MP (3264x2448) (JPG - Software Mejorado)
Captura de Imagen (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (JPG - Verdadero)
Captura de Video (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (WMV - Verdadero)

¹¹¹ Cfr. Logitech 2016.

Captura de Video (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (WMV - Verdadero)			
Cuadros por segundo (max)	30 fps @ 800x600, 15 fps @ HD 960x720 (Límite de Hardware)			
Efectos de Video (VFX)	No disponible			
Luz Derecha	Luz Derecha 2			
Botones	Foto			
Luces Indicadoras (LED)	Actividad/Encendido			
Tono de Privacidad	Software (NOTA: Silencia el video pero no el audio)			
Tamaño del Clip (max)	6.35 mm o 0.25 pulgadas al infinito (No desmontable)			
Longitud del Cable	6 pies o 2 metros			
Dimensiones:				
Componente	Anchura	Profundidad / Longitud	Altura	Peso (con base y cable)
Cámara web sin base	90 mm (3.54 pulgadas)	32 mm (1.26 pulgadas)	40 mm (1.57 pulgadas)	147.42 g (5.2 onzas)
Cámara web con base	90 mm (3.54 pulgadas)	120 mm (4.72 pulgadas)	40 mm (1.57 pulgadas)	147.42 g (5.2 onzas)

Se pueden utilizar diversas cámaras web que cumplan con las especificaciones necesarias en el proyecto. De acuerdo a la Tabla 3.1, las características de la cámara web que son necesarias para un correcto desenvolvimiento del sistema son las correspondientes al campo de visión, longitud focal y la resolución óptica. La cámara debe contar con la capacidad de capturar una imagen de 960x720 de resolución. Además, se debería tener la opción de modificar manualmente las propiedades de la cámara que se utilizarían. Por ejemplo, el enfoque, el zoom o el brillo.

Se diseñó y construyó un soporte para la cámara web y los formularios. El soporte para la cámara web es de acero inoxidable y está compuesto por dos tornillos sínfín, uno en vertical para poder subir y bajar la cámara y otro en horizontal para moverla a la derecha o izquierda. En las Figuras 3.2 y 3.3 se observa el plano original realizado para el soporte de la cámara. Sin embargo, en el eje horizontal se ha colocado un contrapeso cuya distancia es variable y con ello se evita que se incline el soporte. Además, la base para la cámara tiene la opción de modificar su inclinación y su altura. El soporte para los formularios es una base de madera de 25x45 cm con una hendidura de 21x30 cm para que se pueda colocar una hoja A4. El eje vertical del soporte se encuentra a una distancia de 15.5 cm de la parte superior del soporte de los formularios. La base de la cámara se coloca a 12 cm del eje vertical, a una altura de 28 cm del soporte de los formularios y con un ángulo de 45°. El recinto en total tiene las dimensiones de 55x53x43 cm. Se colocan dos lámparas en esquinas contrarias entre sí en el soporte con focos ahorradores de 23 watts de potencia Cool Day Light para iluminar el formulario. Con estas lámparas se evita la diferencia de iluminación si se trabaja de día o de noche. En las Figuras 3.4 y 3.5 se muestran unas fotos del recinto.

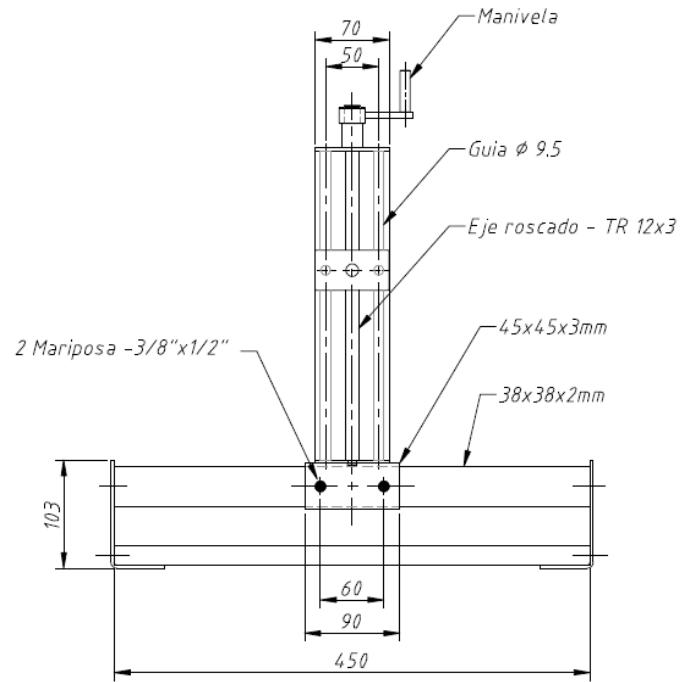


Figura 3.2 Plano del eje vertical

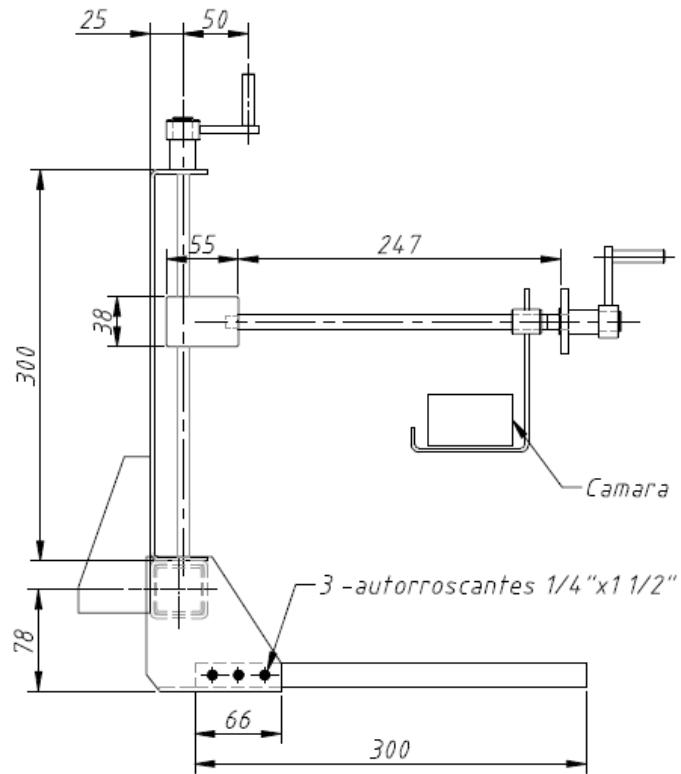


Figura 3.3 Plano del eje horizontal



Figura 3.4 Foto del recinto



Figura 3.5 Foto del recinto visto desde arriba

Además, se diseñaron dos tipos de formularios como ejemplo para la actualización de una base de datos. Estos formularios cuentan con casillas para que se escriban los diferentes caracteres, uno en cada casilla. Por lo tanto, se evita que los caracteres se toquen entre ellos. En la Figura 3.6 se observan los ejemplos de formularios utilizados.

En la Figura 3.7 se observa el diagrama de flujo del hardware del proyecto. En la Figura 3.8 se muestran los pasos a seguir para lograr una correcta instalación del hardware del

proyecto, que permita la adquisición de las imágenes y posteriormente un adecuado reconocimiento.

DNI : <input type="text" value="12345678"/>	Teléfono : <input type="text" value="1234567890"/>
Edad : <input type="text" value="20"/>	Celular : <input type="text" value="1234567890"/>
Teléfono : <input type="text" value="1234567890"/>	DNI : <input type="text" value="12345678"/>
Celular : <input type="text" value="1234567890"/>	Edad : <input type="text" value="20"/>

Formulario 1

Formulario 2

Figura 3.6 Ejemplos de formularios

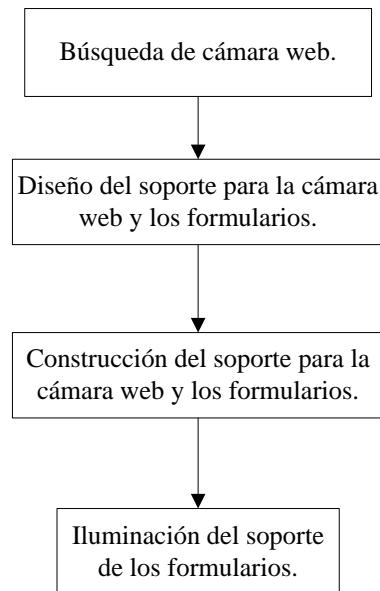
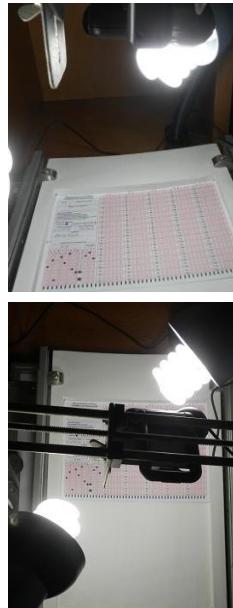


Figura 3.7 Diagrama de flujo del hardware

1.- Se coloca la cámara web en su soporte.



3.- Se ilumina el soporte de formularios.



2.- Se coloca el formulario en su soporte.



4.- Se conecta la cámara web a la computadora.



Figura 3.8 Pasos para la instalación del hardware

3.1. Comentarios finales

En el capítulo 3 se explicó el hardware del proyecto. Se describió la arquitectura y funcionamiento general del hardware con los que se desarrolló el proyecto. Un componente principal es la cámara web, pero se pueden utilizar diversas cámaras que cumplan con las especificaciones requeridas. Se diseñaron y construyeron los soportes para la cámara y los formularios. Además, se instalaron lámparas para trabajar con una buena iluminación. Por último, se diseñaron dos tipos de formularios. En el siguiente capítulo se describirá detalladamente el software del proyecto.

CAPÍTULO 4. DESCRIPCIÓN DEL SOFTWARE DEL PROYECTO

En el presente capítulo se explicará el software del proyecto. Se describirá cada etapa del software indicando sus características y funcionamiento con los que se desarrolló el proyecto. En la Figura 4.1 se muestra el diagrama pictórico de la fase de entrenamiento del proyecto y en la Figura 4.2, el diagrama pictórico de la fase de reconocimiento.

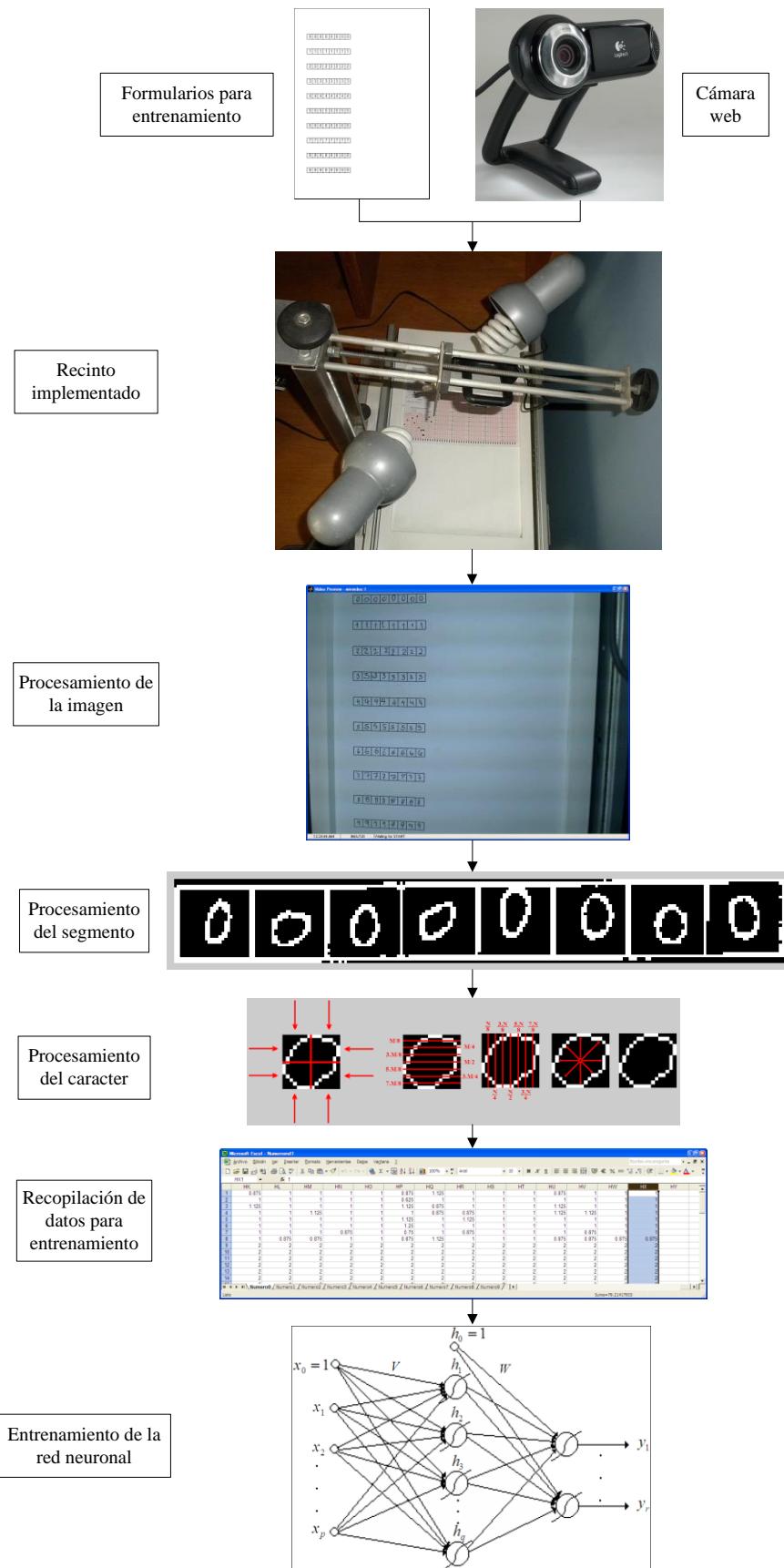


Figura 4.1 Diagrama pictórico de la fase de entrenamiento

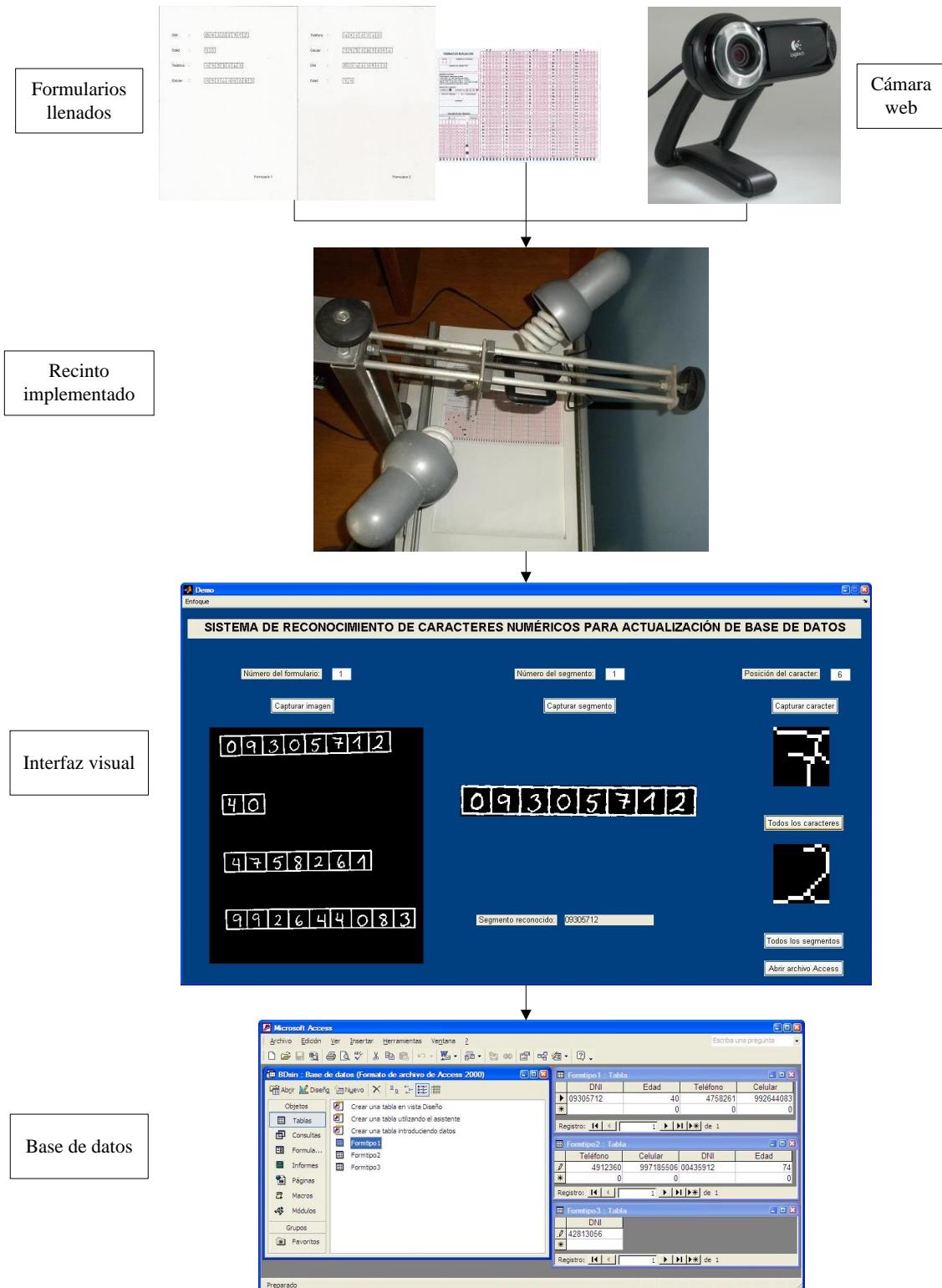


Figura 4.2 Diagrama pictórico de la fase de reconocimiento

4.1. Captura de la imagen

Se utiliza una cámara web, la cual funciona como un sensor que captura las imágenes en forma de matrices. Cada píxel es representado en una fila y columna de esta matriz. El color de cada píxel es definido por su valor.

Se debe instalar la cámara a la computadora a través del driver. No se necesitará ninguna configuración especial en Matlab para la cámara, a menos que los drivers no sean compatibles. Se adquieren las imágenes desde la cámara web a través del toolbox de Matlab. Se utiliza la expresión `vid=videoinput('winvideo')` para crear un objeto que controle la cámara¹¹².

Para obtener las propiedades que contiene la cámara se utilizan las expresiones `src=getselectedsource(vid)` y `get(src)`. Por ejemplo, algunas propiedades son el brillo, el contraste o el enfoque. Con ello, se puede manipular los valores que están por defecto.

En el programa, se configuran las propiedades de la cámara web para tomar la foto del formulario. Se vuelven manuales todas las propiedades automáticas y se eligen sus valores como por ejemplo el enfoque de 90, el zoom de 50 o el brillo de 120.

La expresión `preview(vid)` sirve para visualizar un flujo de imágenes obtenidas de la cámara. En la Figura 4.1.1 se observa el resultado de utilizar esta expresión con las propiedades configuradas. Con la expresión `I=getsnapshot(vid)` se obtiene una imagen como una matriz.

¹¹² Cfr. UPM 2017.

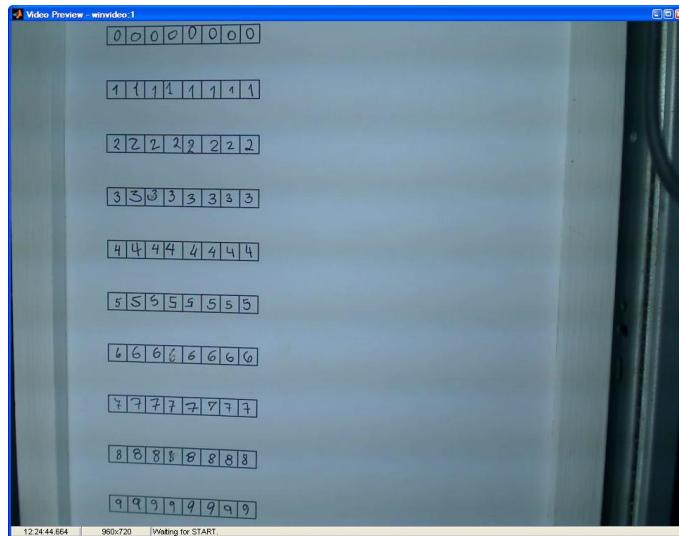


Figura 4.1.1 Visualización de la cámara

Se captura la foto del formulario con una resolución de 960x720 para que cada carácter tenga mayor cantidad de píxeles y se puedan capturar con mayor precisión. Esta matriz de la imagen se cambia a escala de grises para poder trabajar con ella (Figura 4.1.2). Por último, se corta la matriz para obtener solo el sector donde se encuentran los segmentos a analizar (Figura 4.1.3). En el caso de la ficha óptica (formato de evaluación) solo se analizaría la porción donde se encuentra escrito el DNI (Documento Nacional de Identidad). Esta sección se encuentra resaltada con un rectángulo rojo en la Figura 4.1.4.

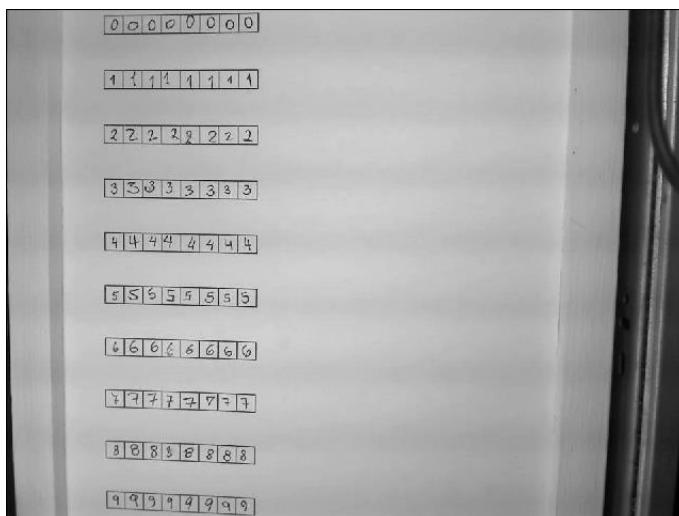


Figura 4.1.2 Imagen obtenida en escala de grises

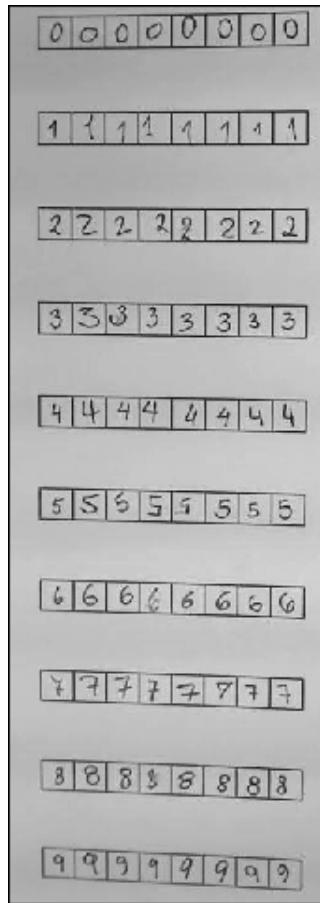


Figura 4.1.3 Imagen recortada

FORMATO DE EVALUACIÓN	
NOMBRE:	NÚMERO DEL PROYECTO:
DNI/NOMBRE DEL CARGO/ESTADO:	
ANTICOCHEO: Alta fotografía negra para imprimir - impresión sobre una alternativa y colores - impresión en color - impresión en blanco y negro Algunas veces se solicita impresión de los respondentes, necesitando que no se produzca una estampa falsa.	
MODO DE MARCAR: CÁRAMELA <input checked="" type="checkbox"/> INGRIETE <input type="checkbox"/>	
APELLIDO PATERNO: EPEJO APELLIDO MATRERO: PÉREZ NOMBRE: CATERINA	
FIRMA/OTRO DATO DEL REQUERIDOO	
PRESA: 42813056 A.D.C.	

Figura 4.1.4 Sección a analizar de la ficha óptica

En la Figura 4.1.5 se muestra el diagrama de flujo de la captura de la imagen.

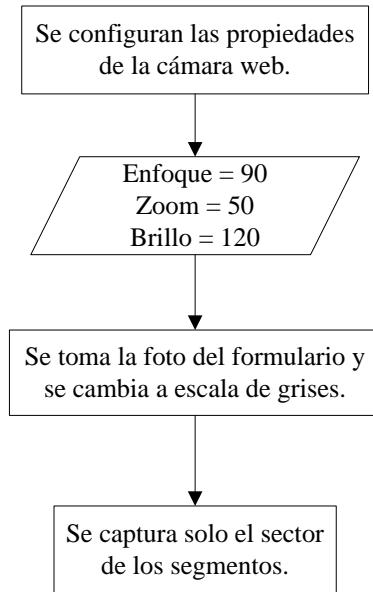


Figura 4.1.5 Diagrama de flujo de la captura de la imagen

4.2. Binarización de la imagen

Para el proceso de binarización, primero se calcula el histograma de la matriz de la imagen obtenida y se guarda en un vector columna, donde la posición indica la tonalidad y el valor indica el número de píxeles de esa tonalidad (ver Capítulo 2.4). Seguidamente, se calcula la probabilidad de las tonalidades dividiendo cada valor del vector entre la cantidad total de píxeles. Este vector sirve para calcular la media, que es la suma de las tonalidades multiplicadas por su respectiva probabilidad, y la varianza, que es el resultado de elevar al cuadrado la diferencia de un vector de tonalidades y la media, y multiplicarlo por el vector de probabilidad. Por lo tanto, con estos dos valores se calcula un valor umbral ideal para poder identificar los píxeles oscuros de la imagen, que en este caso es el resultado de restarle a la media 1.5 veces la raíz de la varianza. Se podría haber restado dos veces la raíz de la varianza, pero no se capturarían bien los caracteres escritos a lapicero debido a que no son tan oscuros como la tinta con que está hecho el formulario. En el caso de la ficha óptica, se le resta 0.25 veces la raíz de la varianza.

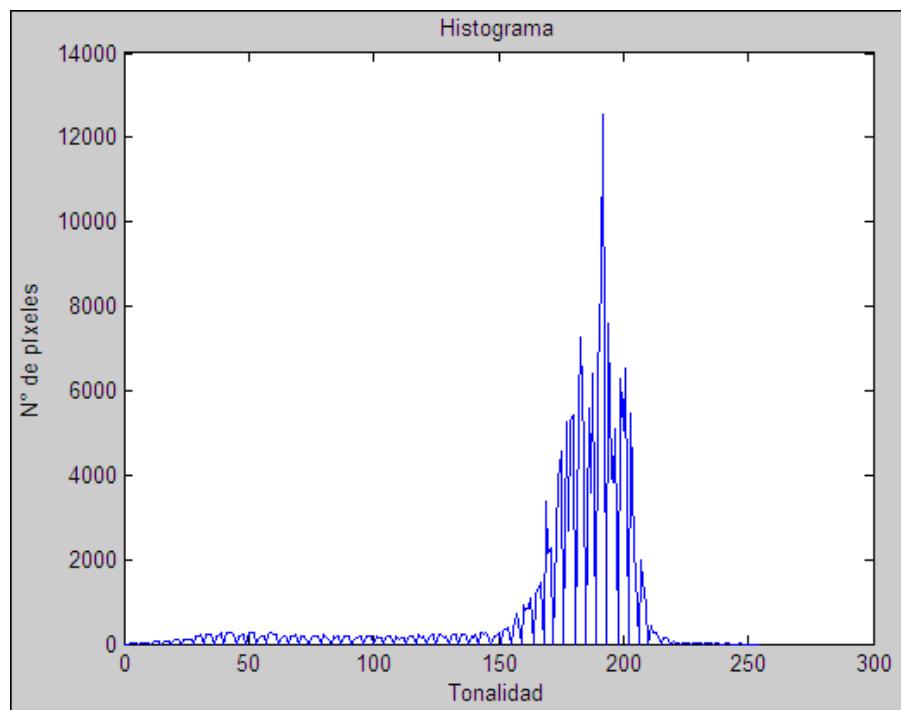


Figura 4.2.1 Histograma de la imagen obtenida

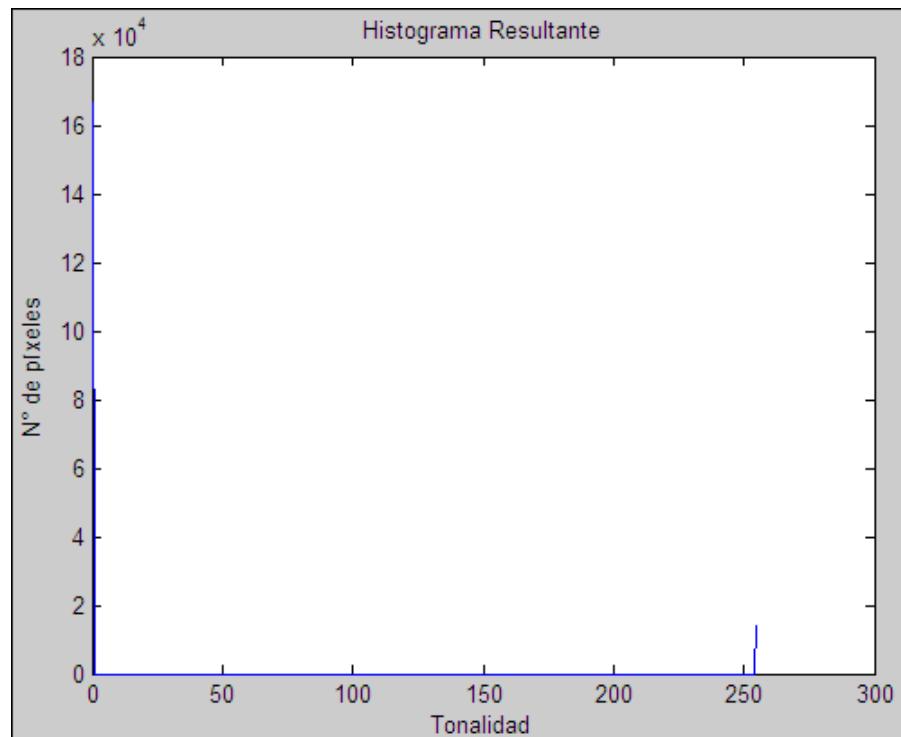


Figura 4.2.2 Histograma de la imagen binarizada

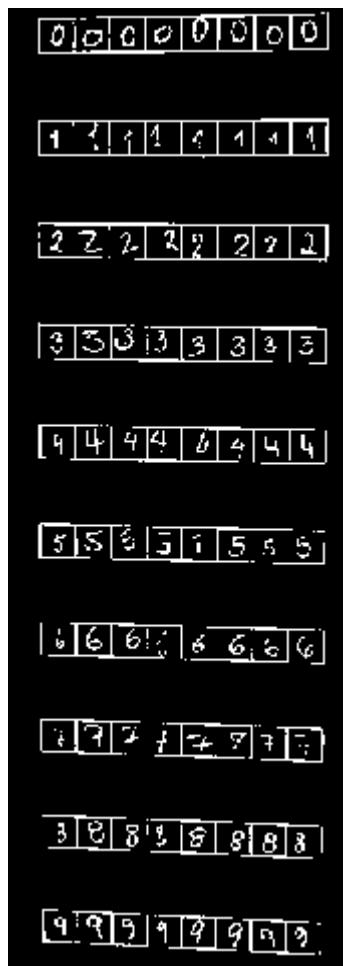


Figura 4.2.3 Imagen binarizada

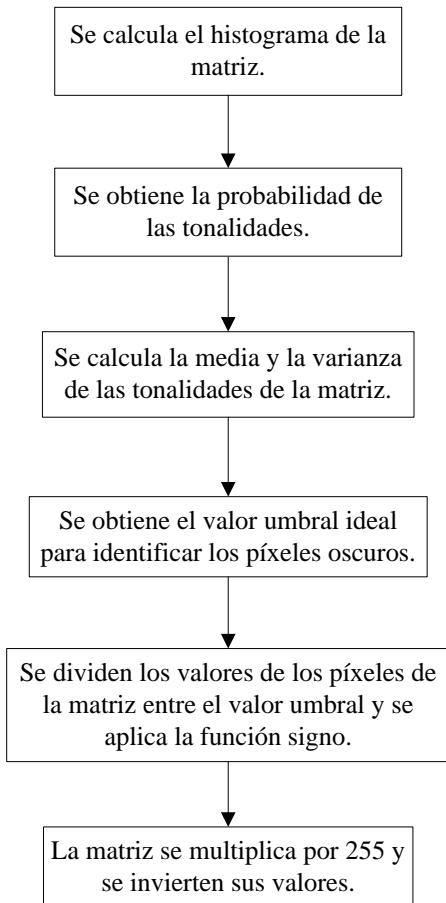


Figura 4.2.4 Diagrama de flujo de la binarización de la imagen

4.3. Segmentación de la imagen

Después se procede con la segmentación, se calcula la proyección vertical de la imagen binarizada para obtener las filas en que se encuentran los segmentos (ver Capítulo 2.5 expresión (2.5.1)). El resultado de la proyección vertical se muestra en la Figura 4.3.1.

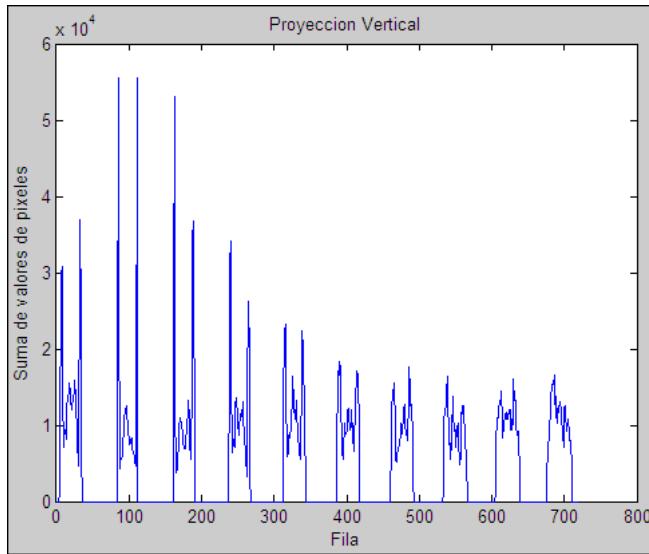


Figura 4.3.1 Proyección vertical de la imagen binarizada

En el vector se obtienen valores iguales a 0, que indican que no hay un segmento en esas filas pues el fondo es negro; o valores múltiplos de 255, que indican que hay varios píxeles blancos en esas filas. Se buscan las posiciones en que el valor de la proyección vertical cambia de 0 a uno diferente de 0 o viceversa y se guardan en un nuevo vector llamado segmento. Si el vector de la proyección vertical comienza con un número diferente a 0, entonces también se guarda la posición uno. Por lo tanto, en el vector segmento se van a encontrar los valores de las filas iniciales y finales de cada segmento de la imagen, ingresados de par en par. Dependiendo del número de segmento elegido, se recuperarán los valores de la fila inicial y final. La fila inicial se calcula con la expresión (4.3.2). La fila final se calcula con la expresión (4.3.3). Se obtendrá una matriz del segmento elegido que es una porción de la matriz binarizada, la cual se muestra en la Figura 4.3.2.

$$\text{Fila inicial} = 2 \cdot (\text{Número de segmento}) - 1 \quad \dots \quad (4.3.2)$$

$$\text{Fila final} = 2 \cdot (\text{Número de segmento}) \quad \dots \quad (4.3.3)$$



Figura 4.3.2 Imagen del segmento elegido

En la Figura 4.3.3 se muestra el diagrama de flujo de la segmentación de la imagen.

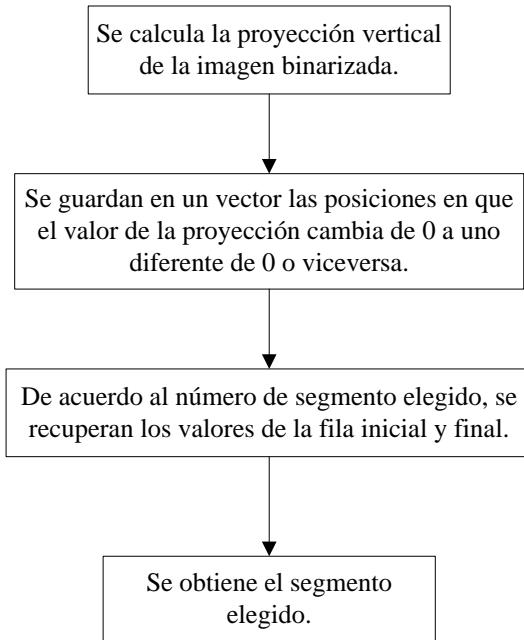


Figura 4.3.3 Diagrama de flujo de la segmentación de la imagen

4.4. Rotación del segmento

Además, puede darse el caso de que por el ángulo de la cámara con respecto a algunos segmentos del formulario estos no se encuentren derechos, lo que conllevaría a tener que girar la matriz del segmento. Para ello se procede a capturar los píxeles con valor que se encuentren en la primera y última columna del segmento. Se tiene la matriz del segmento inclinado con las posiciones obtenidas, la cual se muestra en la Figura 4.4.1.



Figura 4.4.1 Imagen del segmento inclinado

Posteriormente, se captura la columna del primer píxel blanco en la primera fila y si este valor es mayor a la mitad del número total de columnas, entonces se captura la fila del primer píxel blanco en la primera columna; de lo contrario, se captura la fila del primer

Para obtener solo el segmento, se procede a capturar los píxeles con valor que se encuentren en su primera fila, última fila, primera columna y última columna. Se utiliza este proceso porque genera un menor costo computacional comparado con el de las proyecciones verticales y horizontales, debido a que no recorre toda la matriz del segmento para obtener las posiciones deseadas. Por último, se corta la matriz a los valores de las filas y columnas obtenidos como resultado. Esta imagen se muestra en la Figura 4.4.4.



Figura 4.4.4 Imagen del segmento enderezado

En la Figura 4.4.5 se muestra el diagrama de flujo de la rotación del segmento.

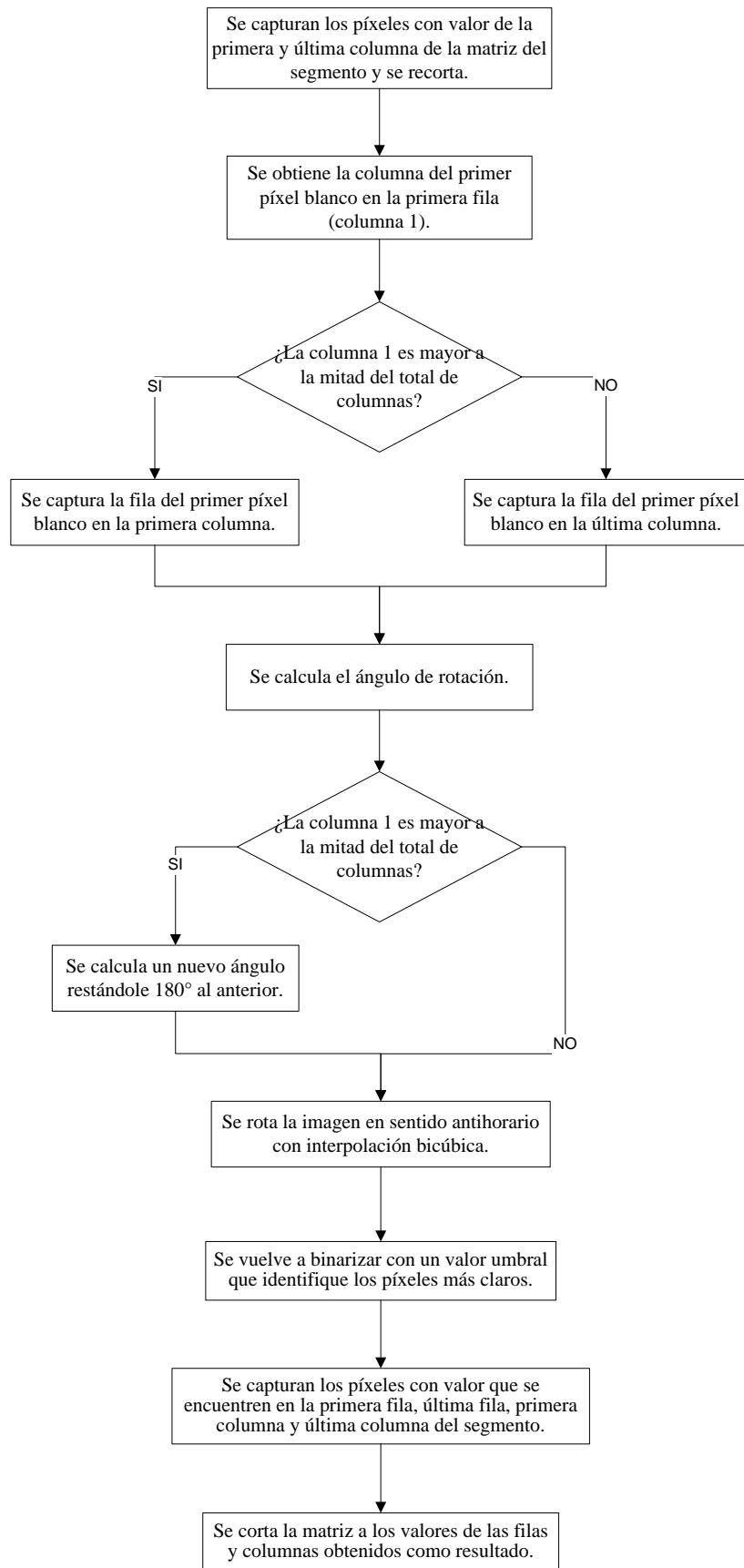


Figura 4.4.5 Diagrama de flujo de la rotación del segmento

4.5. Captura del carácter

Para capturar el carácter, se calcula la proyección horizontal para obtener las columnas en que se encuentran todos los caracteres del segmento (ver Capítulo 2.5 expresión (2.5.2)). El resultado de la proyección horizontal se muestra en la Figura 4.5.1.

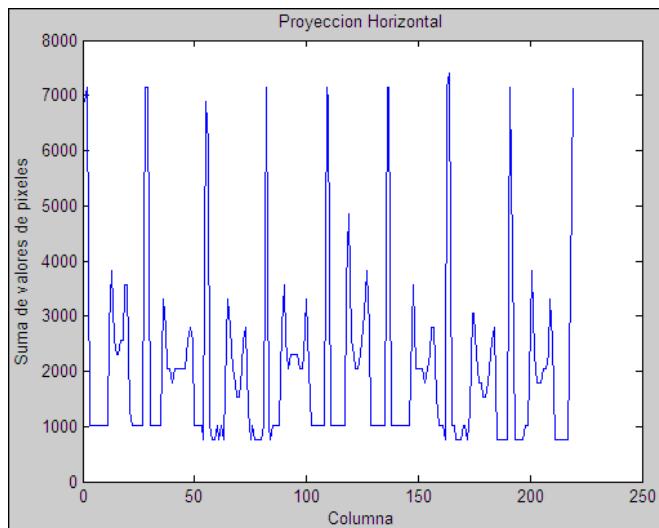


Figura 4.5.1 Proyección horizontal del segmento

Se eligen las columnas en las que el valor de la proyección horizontal sea mayor al número de filas del segmento disminuido en 6, multiplicado por el valor de un pixel blanco (255). Mediante este proceso se obtienen los valores en que se encuentran las divisiones del segmento y se guardan en un vector para que posteriormente se pueda capturar el carácter de cada casilla.

Al elegir una casilla, se restan algunos píxeles de los bordes de la matriz del carácter para que no se capture la casilla en sí. Luego, se procede a buscar la primera fila, última fila, primera columna y última columna de los píxeles blancos del carácter para eliminar el borde negro. Al finalizar, se tiene solo la información del carácter en una matriz (matriz carácter).

Se necesita una matriz de tamaño estándar para todos los caracteres, por lo tanto se cambia la escala de la matriz a una matriz de 32x32 utilizando la interpolación bicúbica (ver Capítulo 2.7). Después, se vuelve a binarizar la matriz pues la interpolación crea nuevos valores de píxeles. Para ello se realiza el proceso de binarización y se calcula el

valor umbral para poder capturar los píxeles más claros de la matriz, que en este caso es el resultado de sumarle a la media la mitad de la raíz de la varianza.

Después, se adelgaza el carácter a solo líneas (proceso de ‘thinning’). Esto se realiza con la función bwmorph del Matlab¹¹³, utilizando la operación ‘thin’ (ver Capítulo 2.9.1). Se eliminan píxeles de tal forma que un objeto se reduce a un mínimo de trazos conectados. Nuevamente se procede a eliminar el borde negro como resultado de este paso, de acuerdo a la explicación anterior para obtener la matriz carácter.

Al disminuir la cantidad de píxeles que representan al carácter, se cambia la escala de la matriz a una matriz de 16x16 utilizando la interpolación bicúbica. Para ello se sigue un proceso similar al explicado anteriormente para escalar a 32x32 píxeles. Luego, se aplica el filtro gaussiano a la matriz con una máscara 3x3 y desviación estándar 0.8 (ver Capítulo 2.6). Por último, se adelgaza nuevamente el carácter y se elimina el borde negro para obtener solo la matriz con información del carácter.

En la Figura 4.5.2 se muestra la matriz carácter después de cada paso utilizado para que se realice un mejor análisis. En la Figura 4.5.3 se muestra el diagrama de flujo de la captura del carácter.

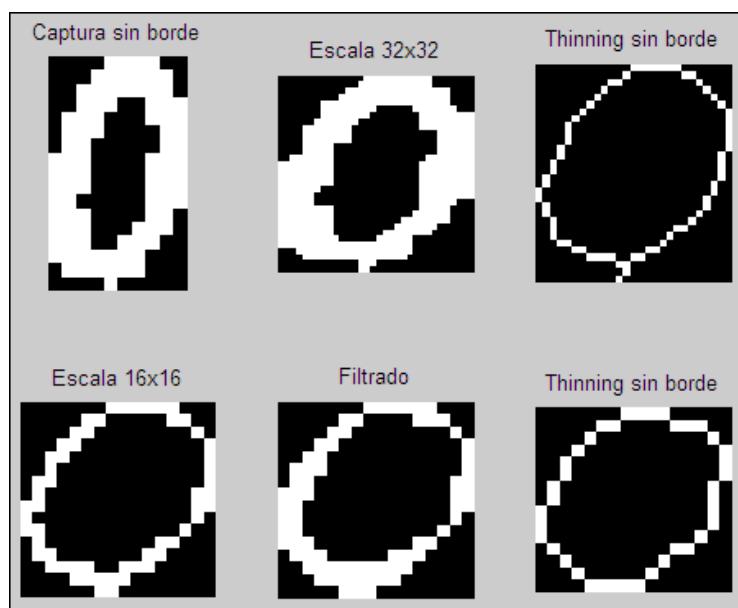


Figura 4.5.2 Resultados de cada paso para la captura del carácter

¹¹³ Cfr. The Mathworks Inc. 2016.

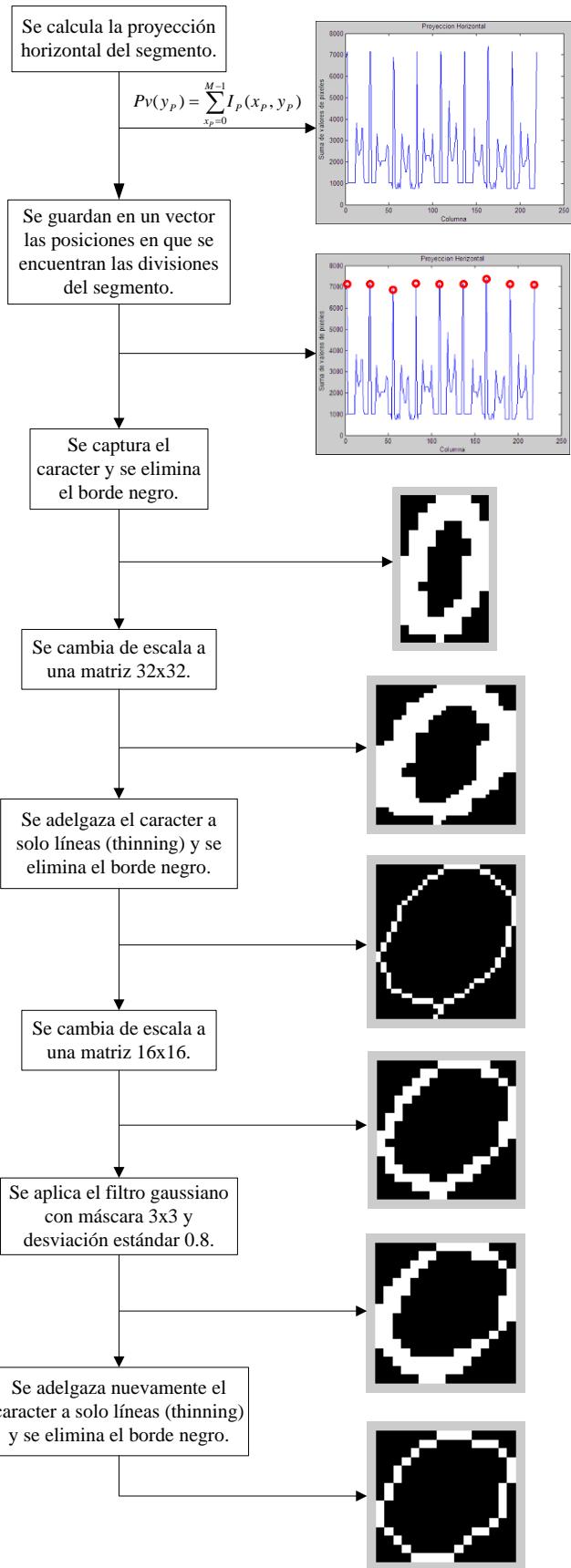


Figura 4.5.3 Diagrama de flujo de la captura del carácter

4.6. Algoritmos para reconocimiento

Se diseñaron diversos algoritmos para determinar los ideales para un mejor entrenamiento y reconocimiento de los caracteres escritos a mano. A partir de los resultados obtenidos de cada uno de ellos, se eligieron cuatro algoritmos que trabajen en conjunto para lograr este fin y son los siguientes:

1. Cuadrantes¹¹⁴.
2. Variaciones.
3. Distancias.
4. Descriptores de Fourier.

4.6.1. Cuadrantes

El primer algoritmo que se diseñó fue el de calcular el número de variaciones de píxeles negros a blancos en los cuadrantes de la matriz carácter. Se agregan píxeles de valor 0 en el borde de la matriz, pero solo un píxel de ancho. Después, se divide la matriz en cuatro cuadrantes iguales. El tamaño de los cuadrantes dependerá de cada carácter debido a que no todos los caracteres tienen el mismo tamaño. En un cuadrante se comienza a contar las variaciones de píxel negro (0) a blanco (255) en cada una de las filas, el resultado se divide entre el total de filas de ese cuadrante menos 1 (debido a que se le agregó una fila de píxeles negros). Un procedimiento parecido se realiza en cada una de las columnas del cuadrante y se obtiene otro resultado. La dirección en que se van a buscar las variaciones en los cuadrantes se observa en la Figura 4.6.1.1. De cada cuadrante se obtienen 2 resultados, lo que daría un total de 8 valores por carácter. Estos valores se guardan en un vector llamado VN.

En la Figura 4.6.1.2 se muestra el diagrama de flujo del algoritmo de las variaciones en los cuadrantes.

¹¹⁴ Cfr. Chung y Wong 1997: 392-393.

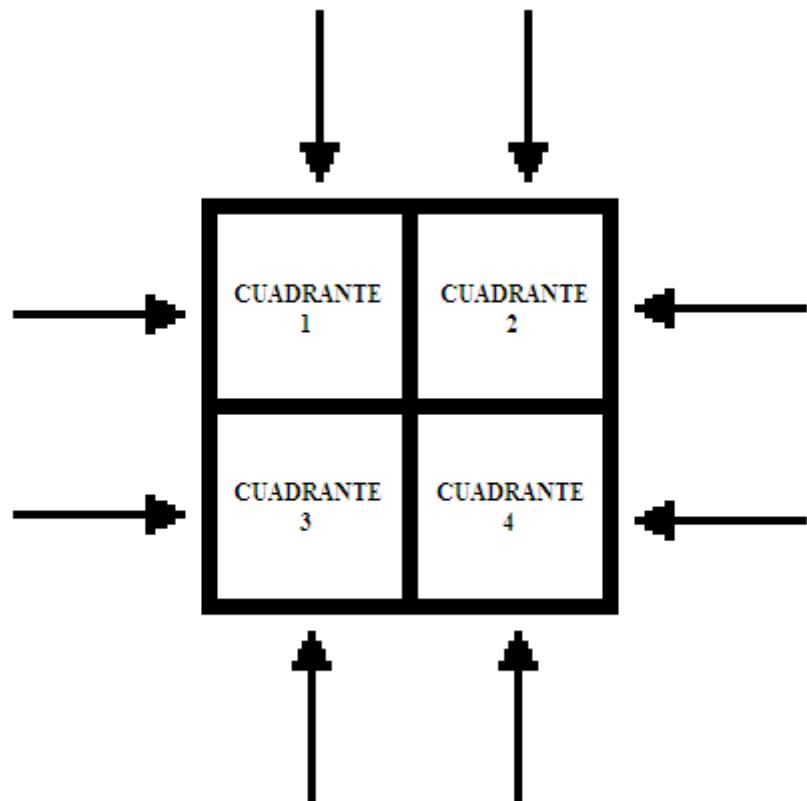


Figura 4.6.1.1 Direcciones de búsqueda en cada cuadrante¹¹⁵

¹¹⁵ Cfr. Chung y Wong 1997: 393.

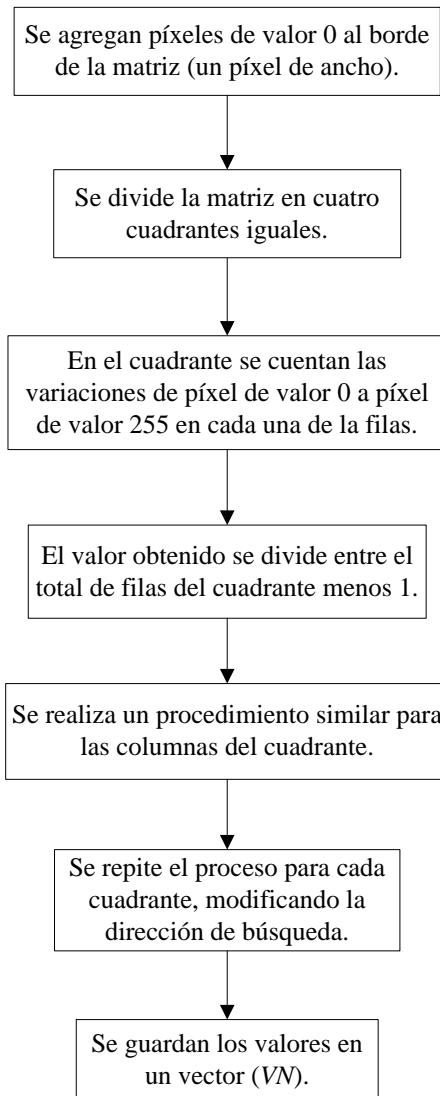


Figura 4.6.1.2 Diagrama de flujo del algoritmo de las variaciones en los cuadrantes

4.6.2. Variaciones

El segundo algoritmo que se diseñó fue el de calcular el número de variaciones de píxeles negros a blancos en determinadas filas y columnas. Se eligen las filas $M/8$, $M/4$, $3.M/8$, $M/2$, $5.M/8$, $3.M/4$ y $7.M/8$, donde M es el número total de filas de la matriz carácter. En cada una de ellas se obtienen las posiciones en que inician y finalizan los píxeles blancos (255). Se guardan estas posiciones en un vector y se calcula su tamaño para dividirlo entre 2. Con esto se obtiene la cantidad de variaciones por fila. Para cada fila se calcula un valor, entonces se tendrá un vector de 7 elementos. Se obtiene otro vector al realizar un procedimiento semejante para las columnas $N/8$, $N/4$, $3.N/8$, $N/2$,

$5.N/8$, $3.N/4$ y $7.N/8$, donde N es el número total de columnas de la matriz carácter. Por último, se concatenan los dos vectores para tener uno de 14 elementos. Estos 14 valores se concatenan al vector VN .

En la Figura 4.6.2.1 se muestra el diagrama de flujo del algoritmo de las variaciones en filas y columnas.

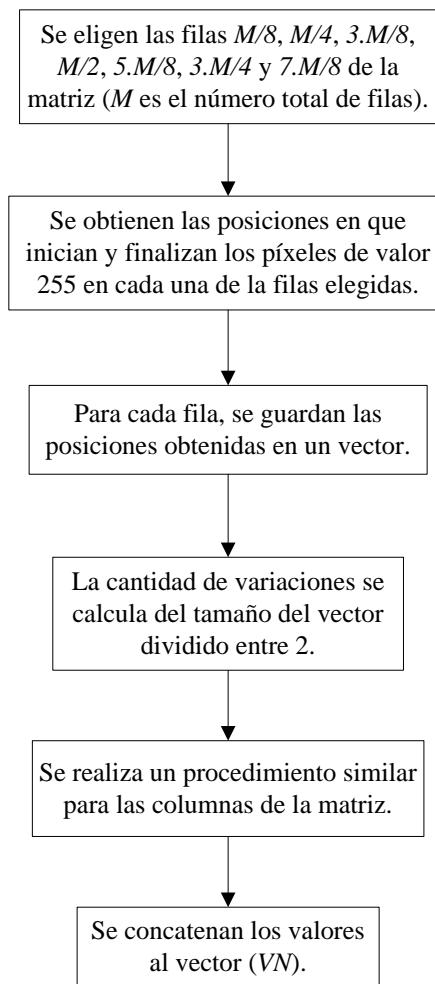


Figura 4.6.2.1 Diagrama de flujo del algoritmo de las variaciones en filas y columnas

4.6.3. Distancias

El tercer algoritmo que se diseñó fue el de calcular las distancias euclidianas desde el centro de la matriz carácter al primer píxel blanco encontrado en diversas direcciones definidas (ver Capítulo 2.8 expresión (2.8.2)), donde el centro de la matriz carácter es el píxel en la posición (i,j) y el píxel blanco encontrado tiene la posición (x_E,y_E) .

En la Figura 4.6.3.1 se muestran las posiciones de la matriz carácter con respecto a su centro, donde M es el número total de filas y N es el número total de columnas de la matriz. La posición central dependerá de cada carácter debido a que no todos los caracteres tienen el mismo tamaño. Las direcciones en que se van a buscar los píxeles blancos (255) se observan en la Figura 4.6.3.2. En el caso de que no se encuentre un píxel de valor 255, se colocará como distancia un valor igual a 0. De cada dirección se obtiene una distancia, lo que daría un total de 8 valores por carácter. Estos valores se concatenan al vector VN .

En la Figura 4.6.3.3 se muestra el diagrama de flujo del algoritmo de las distancias.

(1,1)				(1,j)				(1,N)
	\dots			\vdots			\dots	
		(i-2,j-2)		(i-2,j)		(i-2,j+2)		
			(i-1,j-1)	(i-1,j)	(i-1,j+1)			
(i,1)	\dots	(i,j-2)	(i,j-1)	(i,j)	(i,j+1)	(i,j+2)	\dots	(i,N)
			(i+1,j-1)	(i+1,j)	(i+1,j+1)			
		(i+2,j-2)		(i+2,j)		(i+2,j+2)		
	\dots			\vdots			\dots	
(M,1)				(M,j)				(M,N)

Figura 4.6.3.1 Posiciones de la matriz carácter con respecto a su centro

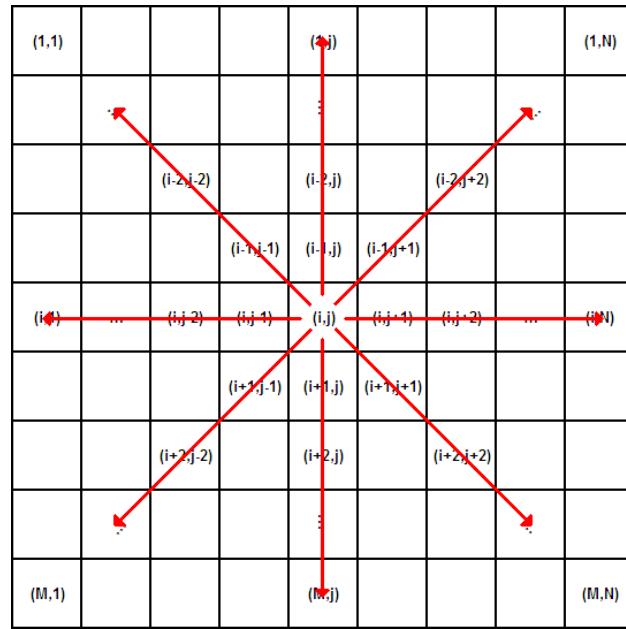


Figura 4.6.3.2 Direcciones del algoritmo de las distancias

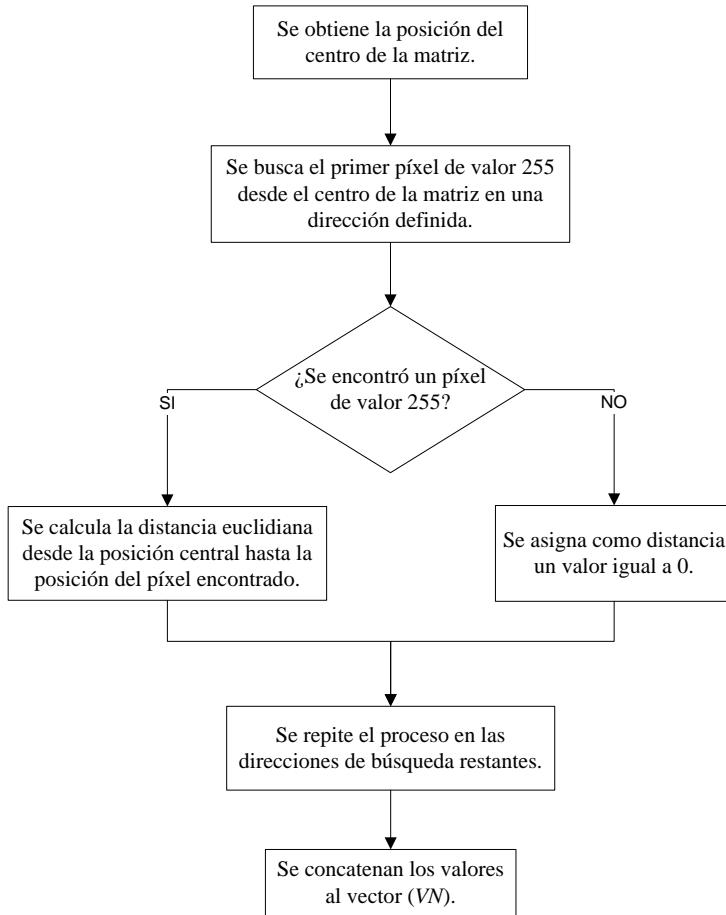


Figura 4.6.3.3 Diagrama de flujo del algoritmo de las distancias

4.6.4. Descriptores de Fourier

El cuarto algoritmo que se diseñó fue el de los descriptores de Fourier. Se obtienen los valores de x_f e y_c de los píxeles del contorno del carácter. Esto se realiza con la función `bwtraceboundary` del Matlab¹¹⁶ que brinda como resultado una matriz con la primera columna de los valores de x_f y la segunda columna de los valores de y_c (ver Capítulo 2.9.2). Se elimina el último elemento pues tiene el mismo valor que el primero (tiene que llegar al primer píxel donde comenzó a capturar los datos).

Se calcula la Transformada Discreta de Fourier de los x_f y de los y_c , de acuerdo a lo explicado en el Capítulo 2.10. Se elimina el primer valor por ser el valor DC. Posteriormente, se aplican las expresiones para obtener las características de los descriptores de Fourier de ser insensibles a la traslación, rotación y al cambio de escala (ver Capítulo 2.10).

El vector resultante es simétrico a partir de la posición $(L-1)/2$ (L es la cantidad de puntos en el contorno) y el primer valor siempre es igual a 1; por lo tanto, se recorta el vector desde la posición 2 hasta la posición $(L-1)/2$. Los primeros valores de los descriptores de Fourier indican la forma general del carácter y los últimos indican los pequeños detalles, por ello se eligen solo los 10 primeros descriptores. Estos 10 valores se concatenan al vector VN .

En las Figuras 4.6.4.1 y 4.6.4.2 se observa el vector de los descriptores recortado y sus 10 primeros valores. Los últimos valores no son significativos para el entrenamiento de las redes neuronales. En la Figura 4.6.4.3 se muestra el diagrama de flujo del algoritmo de los descriptores de Fourier.

¹¹⁶ Cfr. The Mathworks Inc. 2016.

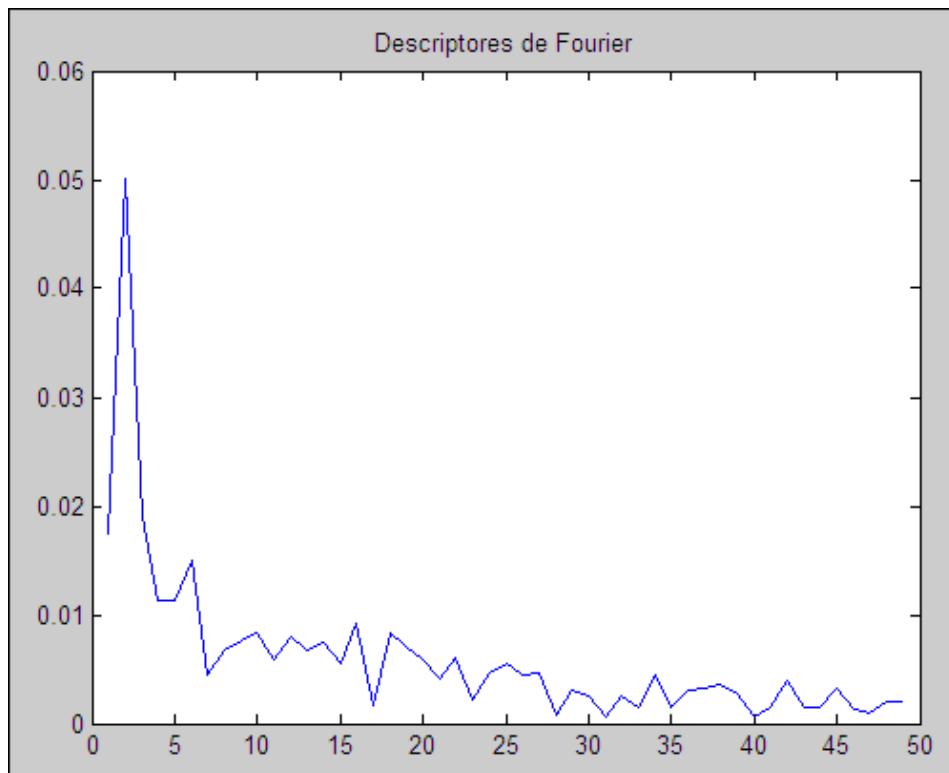


Figura 4.6.4.1 Valores de todos los descriptores de Fourier

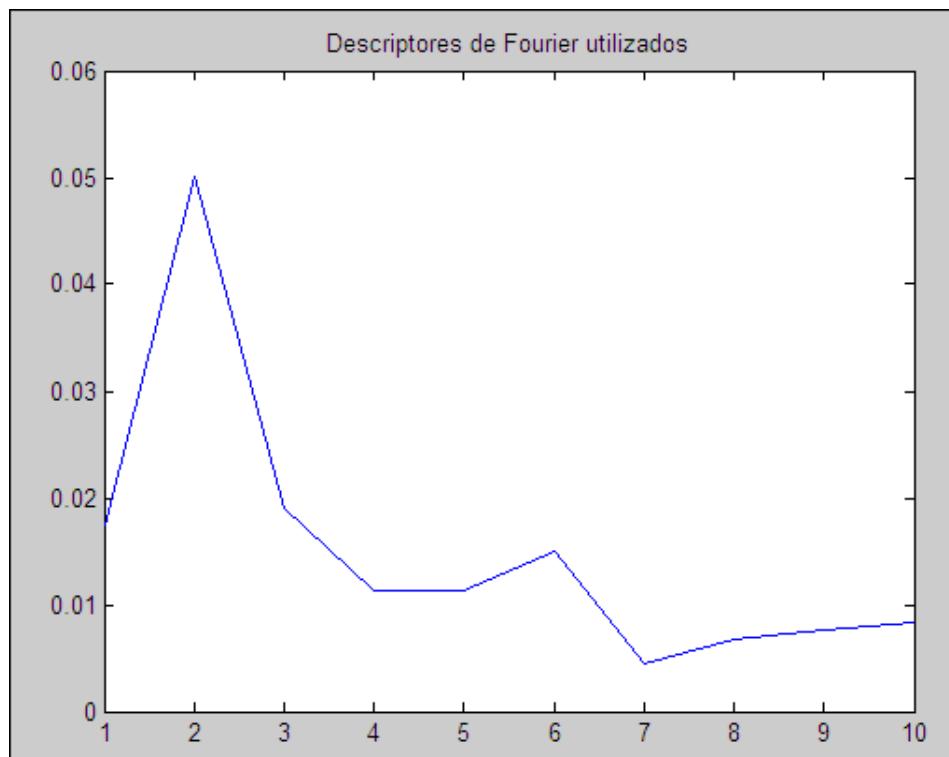


Figura 4.6.4.2 Valores de los 10 primeros descriptores de Fourier

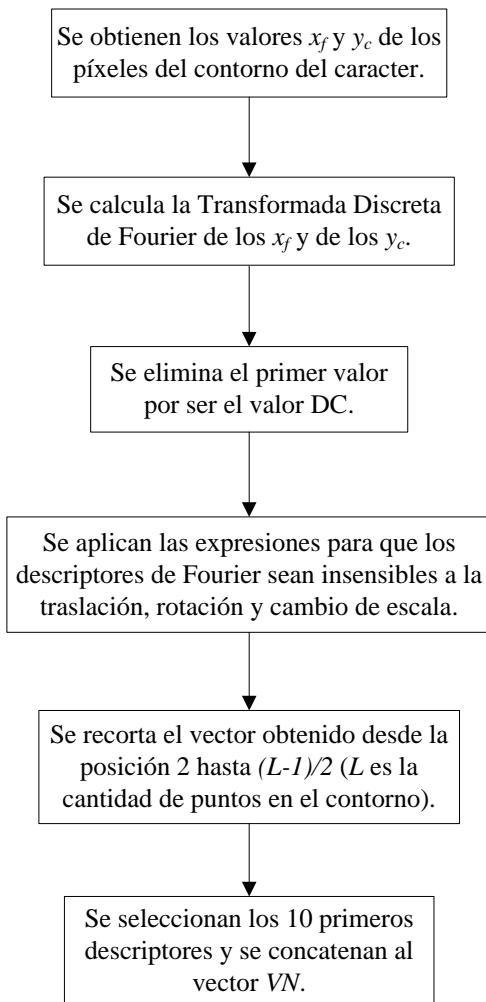


Figura 4.6.4.3 Diagrama de flujo del algoritmo de los descriptores de Fourier

4.7. Entrenamiento de la red neuronal

El vector VN representa a un carácter y utiliza cuatro algoritmos para ello. Se le aplica la transpuesta y se guarda en una hoja Excel. Cada uno de los caracteres del segmento se guarda en una columna distinta de una hoja Excel y cada uno de los segmentos se guarda en distintas hojas del mismo documento Excel. Este procedimiento se realiza para guardar una gran cantidad de números escritos por diferentes personas y con ello poder entrenar la red neuronal. En la Figura 4.7.1 se muestra el archivo Excel donde se almacenan los vectores VN de cada carácter escrito.

Figura 4.7.1 Documento Excel

La red neuronal se diseñó con una capa de entrada, una capa oculta y una capa de salida. En esta configuración, la capa oculta tiene unidades no-lineales con funciones de activación sigmoides, al igual que la capa de salida (ver Capítulo 2.11). La capa de entrada cuenta con 40 entradas, las cuales serían los valores de los que está compuesto el vector *VN*. Estos 40 valores son los obtenidos con los cuatro algoritmos y son los siguientes: los 8 de los cuadrantes (2 por cada cuadrante), las 14 variaciones (7 de las filas y 7 de las columnas), las 8 distancias euclidianas, y los 10 primeros descriptores de Fourier. La capa oculta está conformada por 70 neuronas. Este valor se obtuvo después de entrenar varias veces la red neuronal con diferentes cantidades de neuronas en la capa oculta, hasta encontrar la cantidad con la que se consiguió el error deseado y posteriormente los resultados que cumplan el objetivo del proyecto. Por último, la capa de salida se compone de 4 neuronas porque el máximo valor a representar en binario es el número 9.

Para el entrenamiento de las redes neuronales, se recuperan los valores guardados en las hojas del documento Excel y se guardan en una matriz que será la entrada para las neuronas. Se guardan de forma aleatoria para que las neuronas no aprendan un orden específico. Las salidas representarán cada número pero en binario, por ejemplo el 9 será representado por 1001, y se guardarán en una matriz salida en el mismo orden que la matriz entrada. El entrenamiento se realizó con 70 neuronas en la capa oculta y 185 ejemplares diferentes de cada número, que es un total de 1850 caracteres. Se finalizó

con las iteraciones al obtener un error cuadrático medio de 4.9994×10^{-7} . Este error alcanzado en el entrenamiento se observa en la Figura 4.7.2. Un error menor no permitiría una buena generalización, los caracteres a reconocer tendrían que ser casi idénticos a los de entrenamiento. Con los cuatro algoritmos trabajando en conjunto se obtuvieron buenos resultados pues se está entrenando a la red neuronal con gran información del carácter.

En la Figura 4.7.3 se muestra el diagrama de flujo del entrenamiento de la red neuronal.

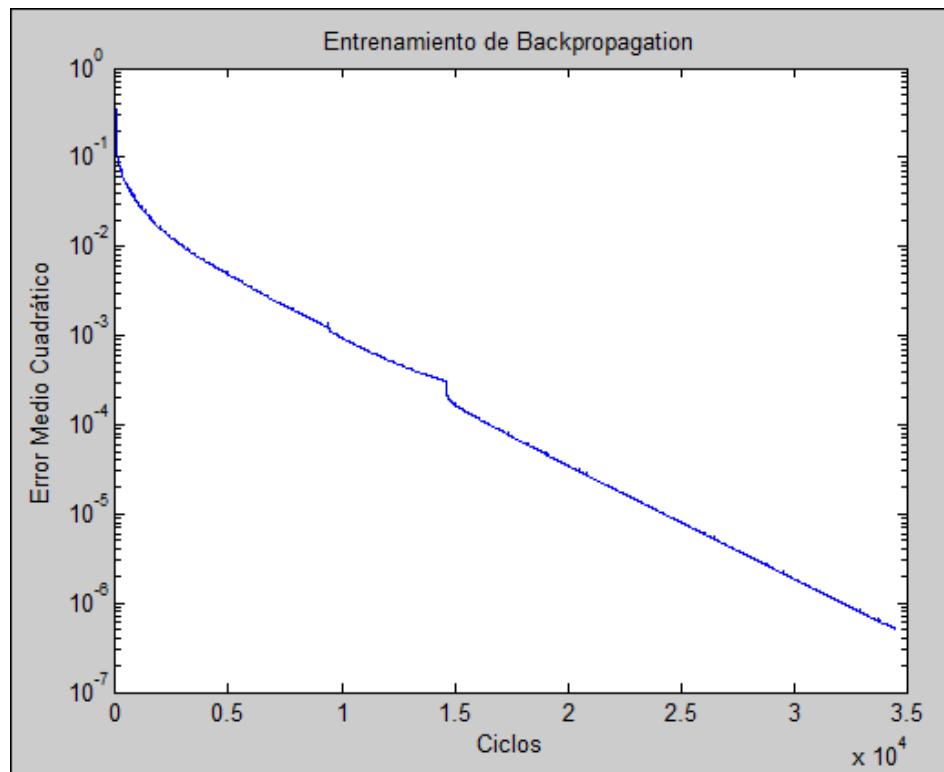


Figura 4.7.2 Entrenamiento de la red neuronal

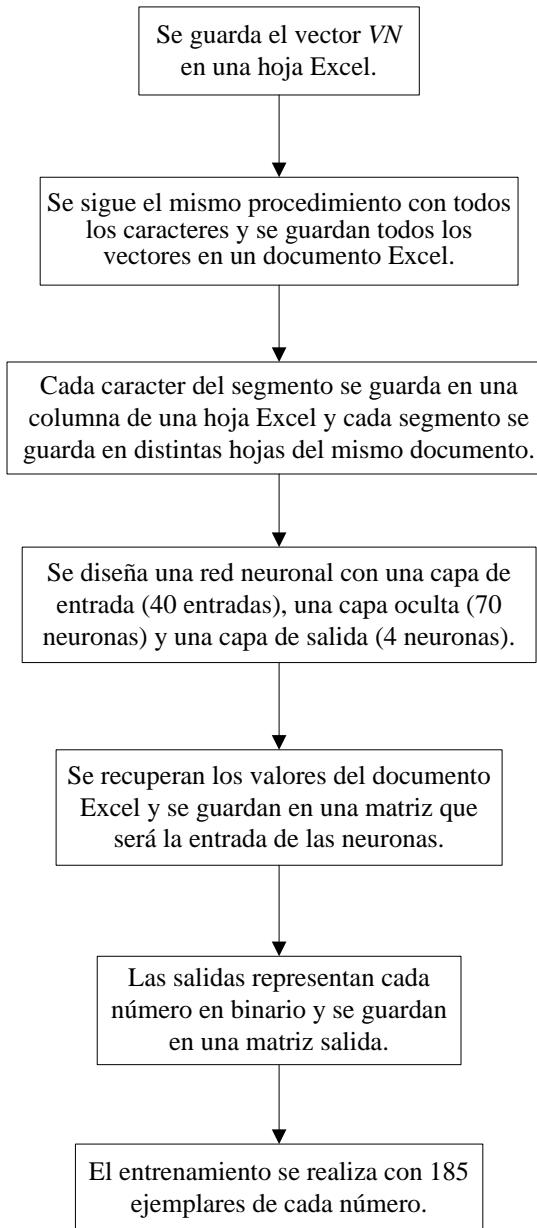


Figura 4.7.3 Diagrama de flujo del entrenamiento de la red neuronal

4.8. Reconocimiento del carácter

En el proceso de reconocimiento, se modifica el zoom de la cámara web a 120. Esto se debe a que solo se desea analizar la información relevante de los formularios, donde se encuentran las casillas llenadas. No se desea analizar el título de las casillas a llenar. Para el caso de la ficha óptica (formato de evaluación), la base de la cámara se debe colocar a 5.8 cm del eje vertical. Una distancia de 6.2 cm a la izquierda con respecto a su posición original (ver Capítulo 3).

Al tomarle foto al formulario llenado, se procede con todos los pasos explicados en este capítulo hasta obtener el vector VN que representa a un carácter. Se realiza el reconocimiento del vector VN con los pesos obtenidos del entrenamiento aplicando la función de activación sigmoide. Debido a que el diseño de la red neuronal cuenta con una capa oculta y una capa de salida, ambas con la función sigmoide, primero se aplica esta función al vector VN con el bias agregado (cuyo valor es igual a 1) multiplicado por los pesos correspondientes a la capa oculta. Después, a este resultado con el bias agregado, multiplicado por los pesos de la capa de salida se vuelve a aplicar la función sigmoide (ver Capítulo 2.11 ecuaciones de (2.11.3.2) a (2.11.3.5)). Por lo tanto, se obtiene un vector de salida el cual representa al número reconocido pero en binario. Por último, se modifica este vector del número en binario a su equivalente en base diez, obteniendo con ello solo un valor por carácter.

Luego, se van guardando los caracteres que se reconocen de un segmento en un vector y cuando se termina de reconocer todos los caracteres del segmento, se guarda este vector como un dato en un nuevo vector. Este último vector tiene cada segmento del formulario como un dato, para luego guardar sus valores a las casillas correspondientes en la base de datos.

Cabe resaltar que antes de tomar la foto, se debe saber con cual formulario se está trabajando. Esto es debido a que cada tipo de formulario tiene una base de datos diferente y pueden ingresarse los datos de manera incorrecta. Por ejemplo, en diferente orden o pueden faltar datos por llenar.

En la Figura 4.8.1 se muestra el diagrama de flujo del reconocimiento del carácter.

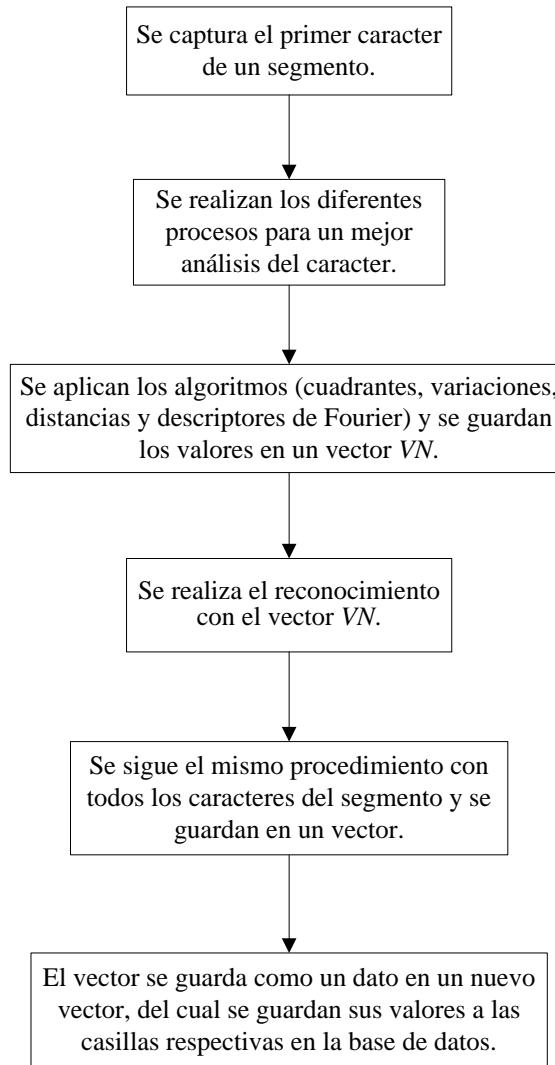


Figura 4.8.1 Diagrama de flujo del reconocimiento del carácter

4.9. Interfaz visual

La interfaz visual del sistema de reconocimiento se implementó en la plataforma visual Guide de Matlab, la cual está relacionada con las bases de datos de Microsoft Access. Se obtienen imágenes del formulario y del segmento. Tiene la opción de mostrar la imagen y reconocer solo un carácter del segmento o todos sus caracteres. Además, se ingresan los datos del formulario a la base de datos con la opción de analizar todos los segmentos. Por último, se puede abrir el archivo de la base de datos que se está utilizando.

La ventana de la interfaz visual está conformada por ciertos títulos, casillas y botones, los cuales se detallan a continuación:

- Número del formulario: Este título está compuesto por una casilla cuyo valor se puede modificar para indicar el formulario que se va a analizar y con ello ingresar correctamente su información a la base de datos. Debido a que el sistema se implementó para reconocer tres tipos de formularios (dos formularios diseñados y una ficha óptica), en la casilla solo se deben ingresar los valores del 1 al 3.
- Capturar imagen: Este botón realiza los procesos de capturar la foto del formulario, cambiar la matriz a escala de grises y recortarla para solo obtener el sector donde se encuentran los segmentos a analizar. Además, se efectúa la binarización de la imagen. Estos procesos se especificaron en las secciones anteriores. Cabe resaltar que en el caso de que se indique un valor mayor a 3 en la casilla del número del formulario, al presionar este botón aparecerá un mensaje de error y no se realizará ningún proceso debido a que no existe ese número de formulario.
- Número del segmento: En este título, el usuario tiene la opción de elegir el segmento que desea analizar al ingresar su valor en la casilla. Los posibles valores que se pueden indicar en esta casilla dependen de la cantidad de segmentos que componen el formulario analizado. El formulario 3 solo contiene un segmento; por lo tanto, no se considera el valor que se indique en esta casilla.
- Capturar segmento: Este botón se encarga de capturar el número de segmento elegido. En el caso de que el segmento se encuentre inclinado, el sistema procedería a enderezarlo. Anteriormente se explicaron con más detalle estos procesos. Si en la casilla del número de segmento, se indica un valor mayor al total de segmentos del formulario, entonces al presionar este botón se desplegará un mensaje de error y no se efectuará ningún proceso porque no existe ese número de segmento. El botón se encuentra deshabilitado en el caso de estar analizando el formulario número 3 (ficha óptica).
- Posición del carácter: Este título está compuesto por una casilla cuyo valor se puede modificar para indicar la posición del carácter, que se encuentra en el segmento elegido, que se desea a analizar. Los posibles valores que se pueden indicar en esta casilla dependen de la cantidad de caracteres que componen el segmento analizado.

- Capturar carácter: Este botón realiza los procesos para capturar el carácter cuya posición ha sido indicada. Además, se aplican los cuatro algoritmos diseñados para proceder con el reconocimiento del carácter seleccionado. Estos procesos se especificaron en las secciones anteriores. Por último, se desplegará un mensaje que indique el reconocimiento obtenido, al cual se presionará “OK” para continuar en la interfaz. Cabe resaltar que en el caso de que se indique un valor mayor al total de caracteres en el segmento, al presionar este botón aparecerá un mensaje de error y no se realizará ningún proceso debido a que no existe esa posición.
- Todos los caracteres: Este botón se encarga de capturar, procesar y reconocer todos los caracteres de los que está compuesto el segmento seleccionado en la casilla “Número del segmento”. Al finalizar el proceso de cada carácter, aparecerá un mensaje con el reconocimiento obtenido el cual se cerrará automáticamente hasta completar todos los caracteres del segmento. Además, se irán agregando los reconocimientos en la línea de “Segmento reconocido”.
- Todos los segmentos: Este botón realiza los procesos de capturar, procesar y reconocer todos los caracteres de los que están compuestos todos los segmentos del formulario indicado en la casilla “Número del formulario”. Al igual que el botón “Todos los caracteres”, al finalizar el proceso de cada carácter se desplegará un mensaje con el reconocimiento obtenido, el cual se cerrará automáticamente hasta completar todos los caracteres de todos los segmentos. Además, se irán agregando los reconocimientos en la línea de “Segmento reconocido” para cada uno de los segmentos. Por último, se ingresan los reconocimientos obtenidos del formulario a la base de datos. En el caso de estar analizando el formulario número 3 (ficha óptica), el botón se encuentra deshabilitado debido a que solo cuenta con un segmento.
- Segmento reconocido: Este título cuenta con una casilla no modificable manualmente, donde se irán agregando los reconocimientos obtenidos de cada carácter de un segmento. Esto sucede luego de presionar el botón “Todos los caracteres” o “Todos los segmentos”. Para este último caso, al finalizar el proceso de reconocimiento de un segmento, los datos de la casilla se borran para proceder en agregar los reconocimientos de los caracteres del siguiente segmento a analizar.

- Abrir archivo Access: Este botón abre el archivo Access en el cual se ha estado ingresando la información procesada del formulario analizado. Por lo tanto, se puede comprobar la actualización de la base de datos.

En la base de datos se tienen diferentes tablas que representan a un tipo de formulario, pues tienen distintos campos u otra cantidad de ellos. En la Tabla 4.9.1 se observan los tipos de datos utilizados para cada campo dependiendo del formulario a analizar. Con respecto al DNI, se considera como un texto debido a que puede comenzar con el número cero y este valor debe ser almacenado. La cantidad de valores a ingresar para cada campo se encuentra limitado en el formulario a completar.

Tabla 4.9.1 Descripción de las tablas de la base de datos

Formulario	DNI	Edad	Teléfono	Celular
1	Texto	Número	Número	Número
Formulario 2	Teléfono	Celular	DNI	Edad
	Número	Número	Texto	Número
Ficha óptica	DNI			
	Texto			

En la Figura 4.9.1 se muestran unos ejemplos de los formularios utilizados para la actualización de la base de datos. En la Figura 4.9.2 se muestra la interfaz visual con un formulario de ejemplo y en la Figura 4.9.3 se muestra la base de datos que ha sido actualizada con los datos ingresados del formulario. En la Figura 4.9.4 se muestra el diagrama de flujo de los pasos a seguir en la interfaz visual.

DNI : <input type="text" value="09305712"/>	Teléfono : <input type="text" value="4912360"/>
Edad : <input type="text" value="40"/>	Celular : <input type="text" value="997185506"/>
Teléfono : <input type="text" value="4758261"/>	DNI : <input type="text" value="00435912"/>
Celular : <input type="text" value="992644083"/>	Edad : <input type="text" value="74"/>

Formulario 1 Formulario 2

Figura 4.9.1. Ejemplos de formularios llenados

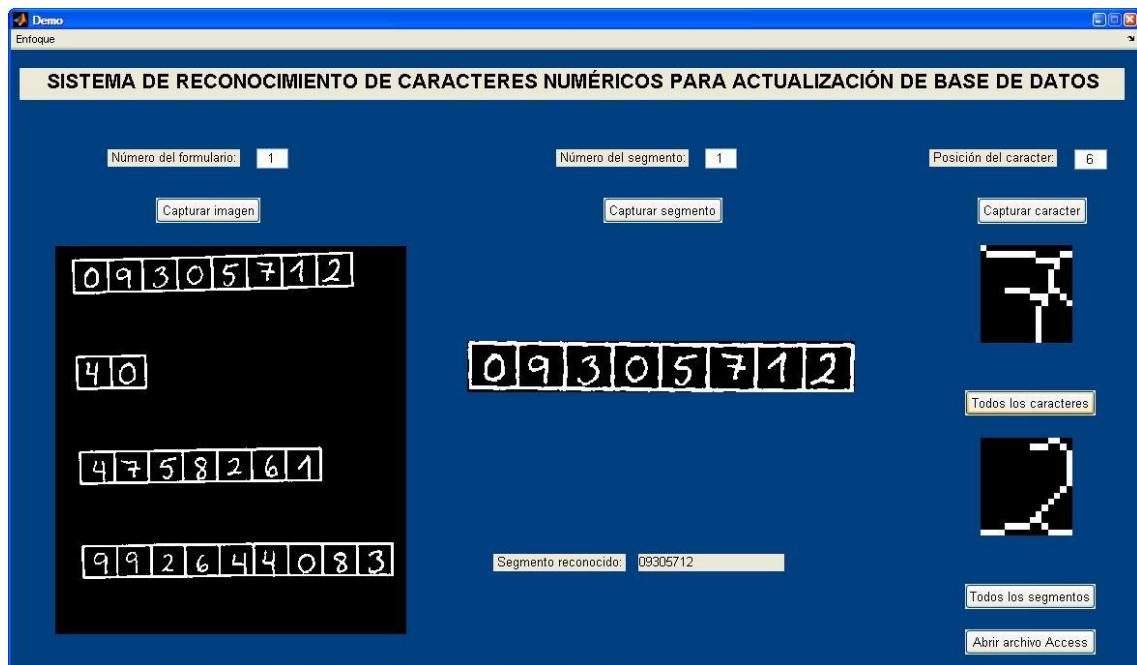


Figura 4.9.2 Plataforma visual Guide de Matlab

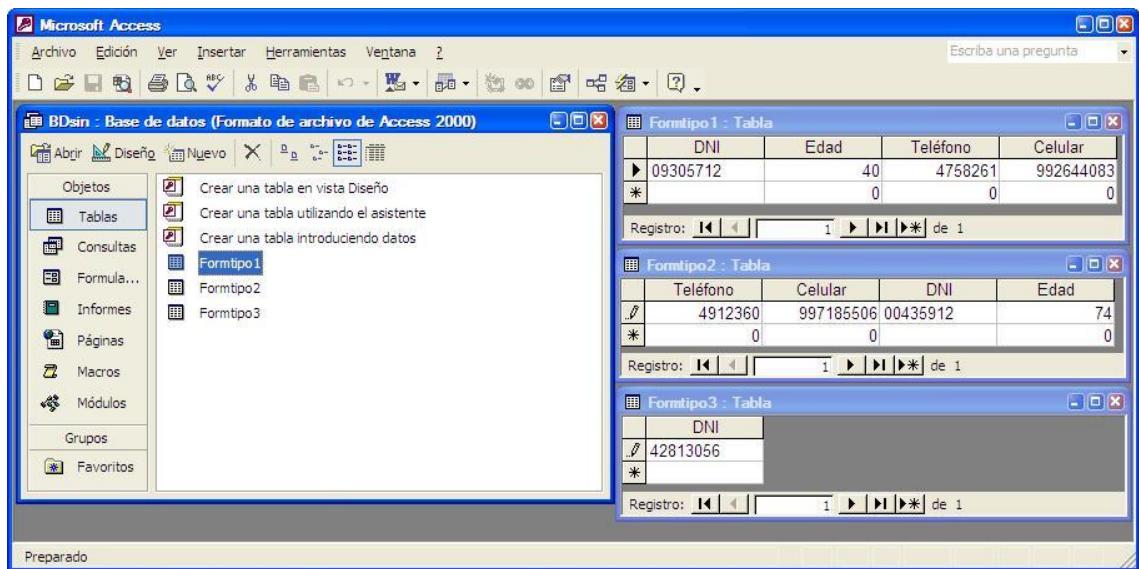


Figura 4.9.3 Base de datos de MS Access

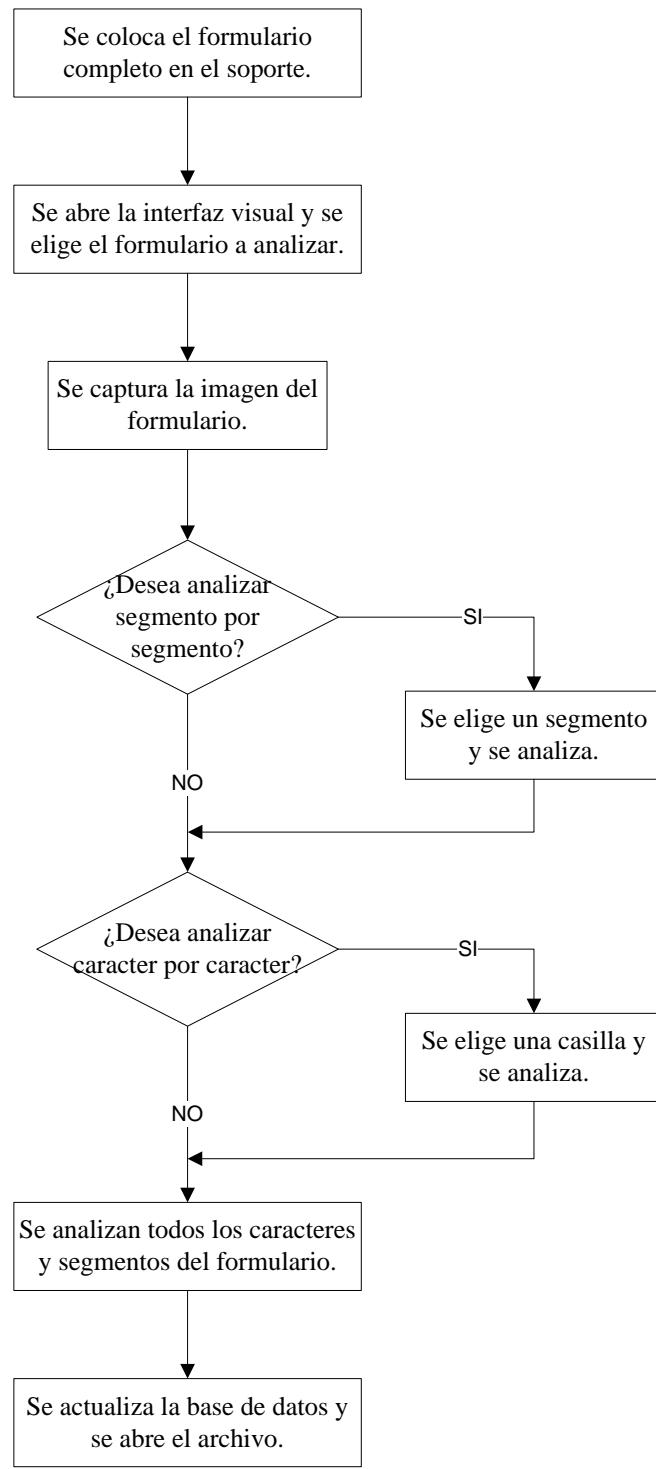


Figura 4.9.4 Diagrama de flujo de la interfaz visual

4.10. Diagramas de bloques del software del proyecto

En la Figura 4.10.1 se muestra el diagrama de bloques desde la captura de la imagen hasta la binarización.

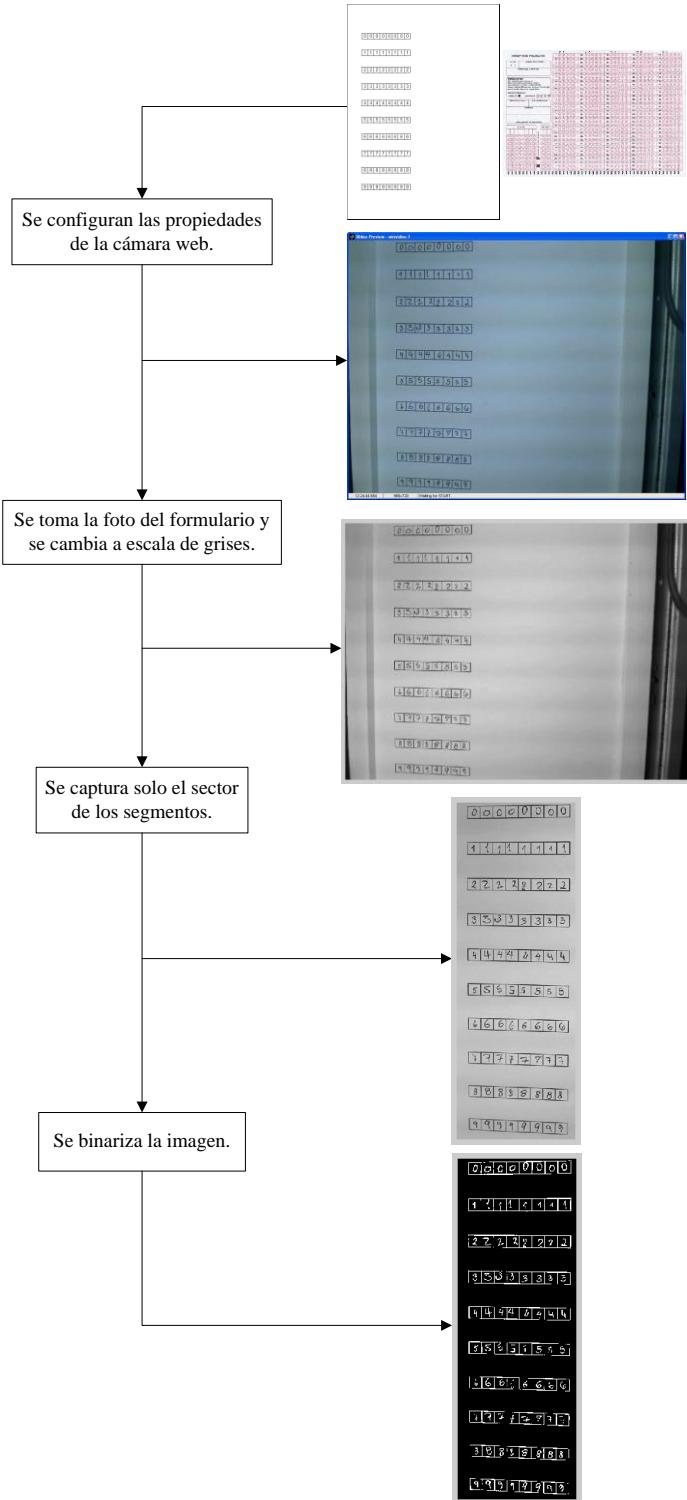


Figura 4.10.1 Diagrama de bloques de la captura de la imagen

En la Figura 4.10.2 se muestra el diagrama de bloques desde la segmentación de la imagen hasta la captura del segmento. Si se analiza el formato de evaluación, no se realizaría la segmentación porque sólo se enfoca en una porción del formato.

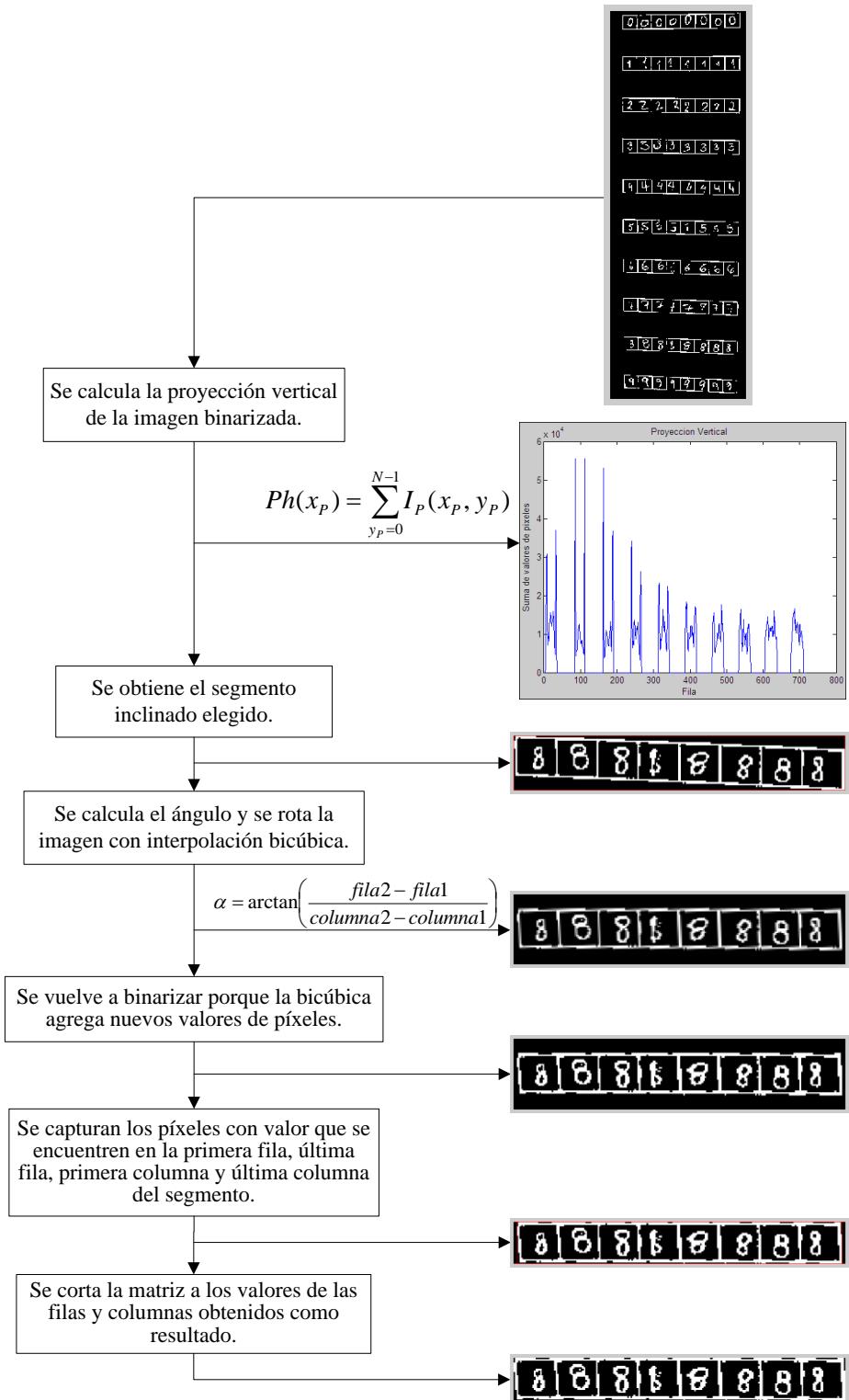


Figura 4.10.2 Diagrama de bloques de la captura del segmento

En la Figura 4.10.3 se muestra el diagrama de bloques de los algoritmos diseñados (cuadrantes, variaciones, distancias y descriptores de Fourier) para el entrenamiento del carácter.

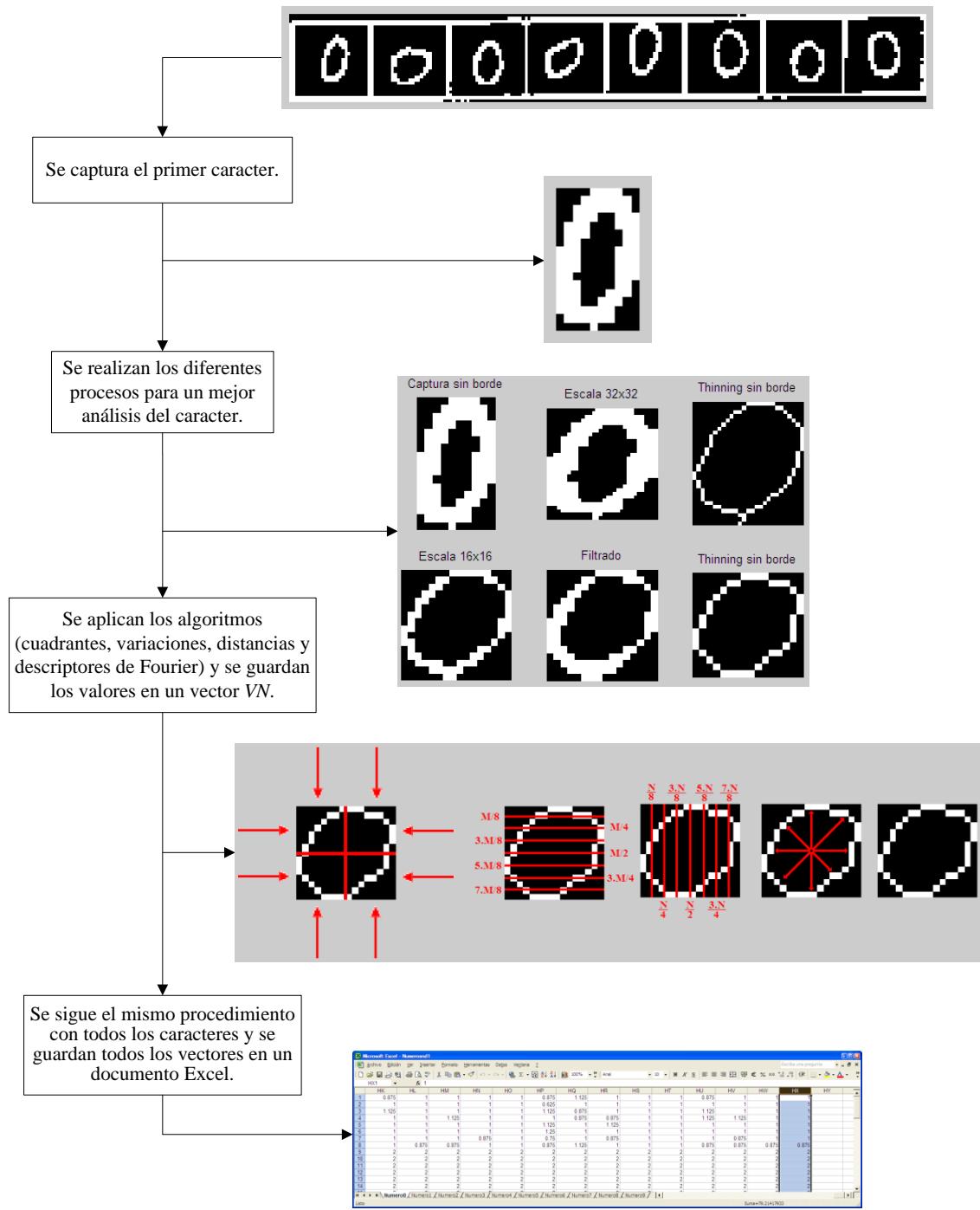


Figura 4.10.3 Diagrama de bloques para el entrenamiento

En la Figura 4.10.4 se muestra el diagrama de bloques del reconocimiento de un carácter y la base de datos que es actualizada a partir de los datos del formulario.

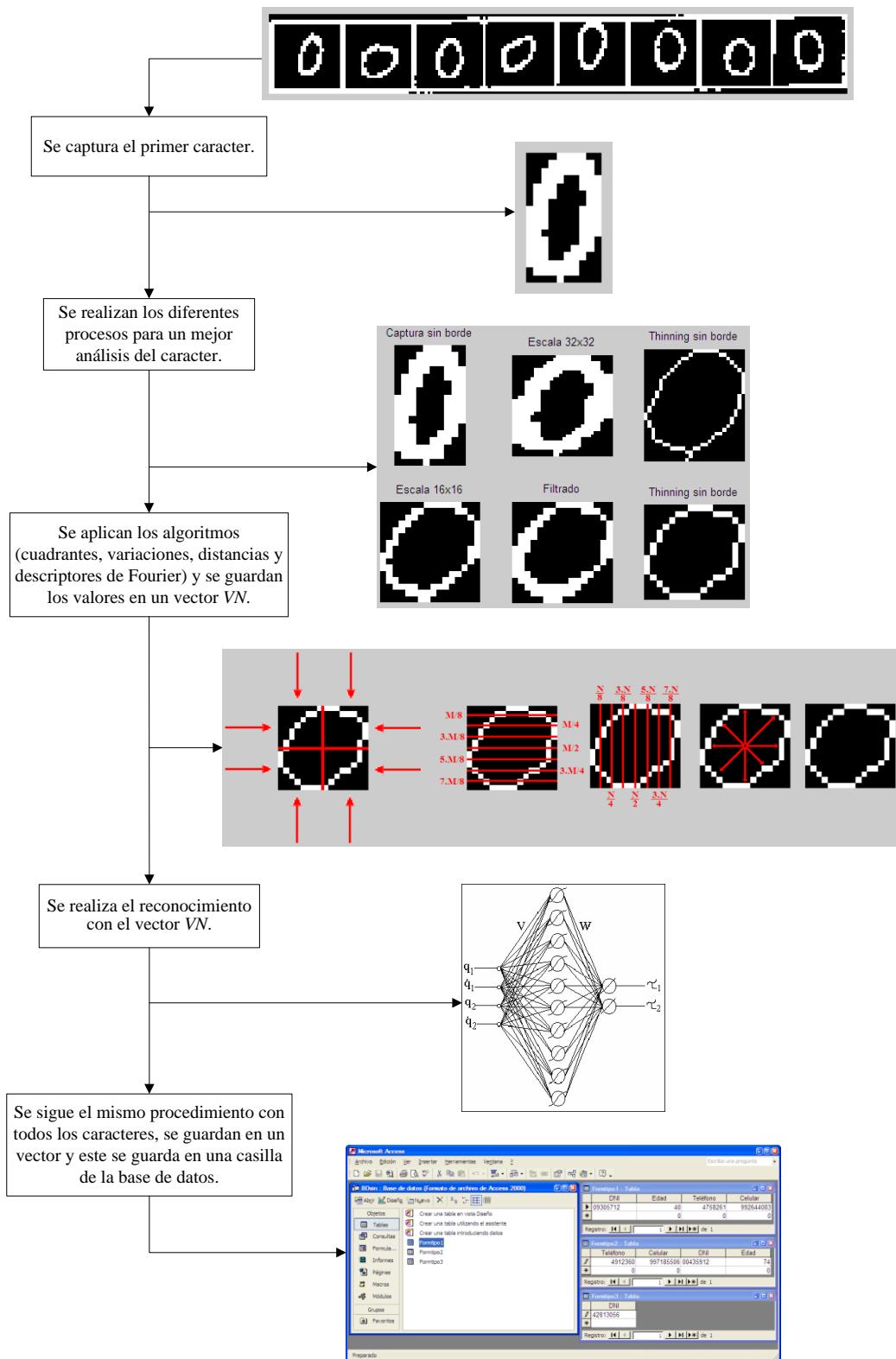


Figura 4.10.4 Diagrama de bloques para el reconocimiento

4.11. Comentarios finales

En el capítulo 4 se explicó el software del proyecto. Se describió cada etapa del software indicando sus características y funcionamiento con los que se desarrolló el sistema. El proceso consiste en capturar la imagen del formulario para proceder a su binarización y segmentación. Después, se capturan cada uno de los caracteres que componen a un segmento del formulario y se aplican los algoritmos para su posterior identificación. Además, se explicaron a detalle los algoritmos diseñados para un correcto reconocimiento de los caracteres numéricos escritos a mano. Este reconocimiento se realiza en base al entrenamiento de las redes neuronales. Finalmente, se actualiza la base de datos con los valores identificados. En el siguiente capítulo se describirán las pruebas realizadas, sus resultados y la validación del proyecto.

CAPÍTULO 5. PRUEBAS, RESULTADOS Y VALIDACIÓN

En las Figuras 5.1 y 5.2 se pueden observar ejemplos de los mensajes al identificar los caracteres del formulario, en los casos de que sean correctos o incorrectos, respectivamente. En el primer caso, se solicita identificar el carácter que se encuentra en el primer cuadrado del primer segmento. El sistema realiza un verdadero reconocimiento al informar que el carácter es el número cero. En el segundo caso, se solicita identificar el carácter que se encuentra en el séptimo cuadrado del cuarto segmento. El sistema realiza un falso reconocimiento al informar que el carácter es el número cuatro cuando realmente es un cero. En este ejemplo se observa que el problema estuvo en la captura del carácter al eliminarse el borde y con ello no se obtuvo un cero completamente cerrado. Por último, en caso de no reconocerse el carácter se observaría un mensaje de error.

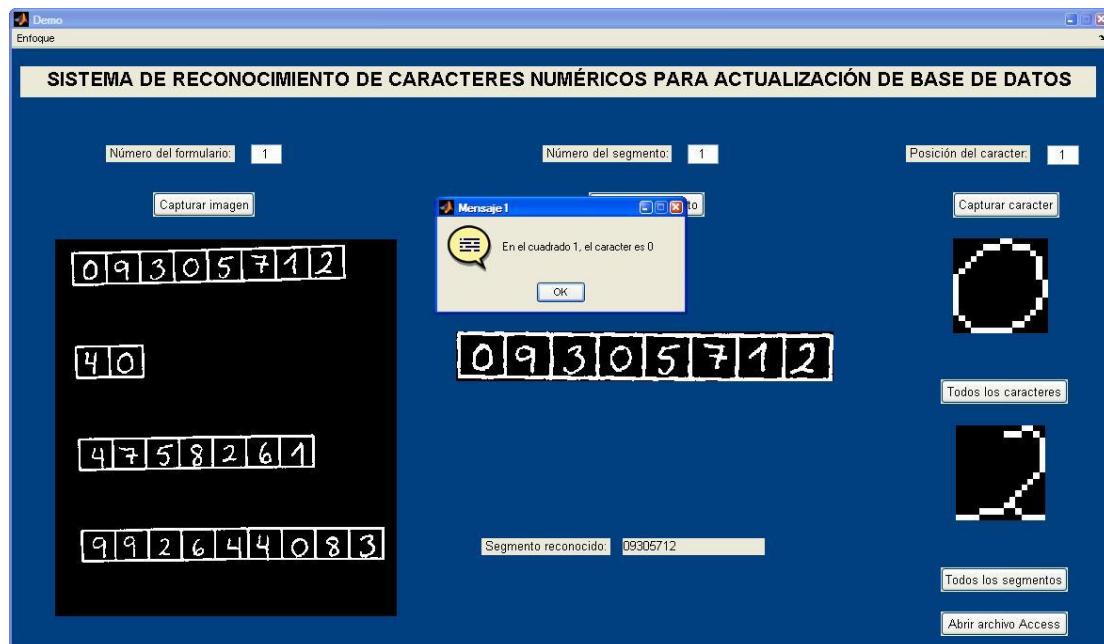


Figura 5.1 Mensaje de un verdadero reconocimiento

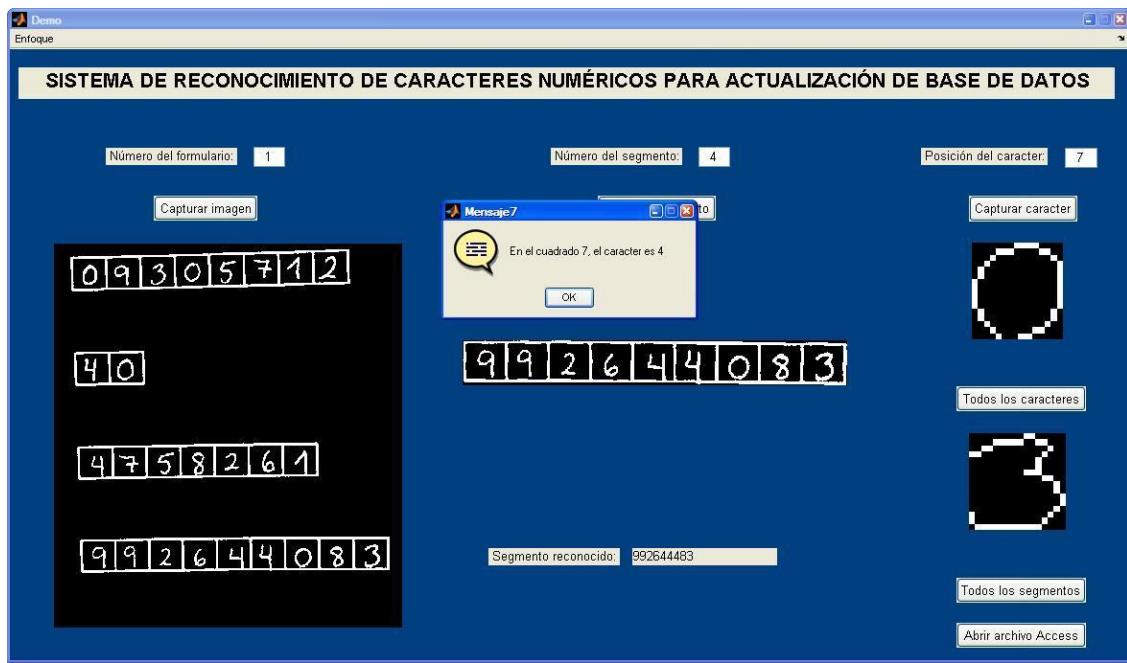


Figura 5.2 Mensaje de un falso reconocimiento

Gracias a los algoritmos diseñados, se logró identificar 913 verdaderos reconocimientos de un total de 1000 caracteres, brindando un resultado de 91.30% (porcentaje sobre el total de caracteres), con el entrenamiento de 185 ejemplares de cada carácter escritos por diferentes personas (un total de 1850 caracteres). No se reconocieron 2 caracteres por lo que se obtienen falsos rechazos en un 0.20%, el cual es un buen resultado (ver Capítulo 2.12). En la Tabla 5.1 y Figura 5.3 se muestran los porcentajes de verdaderos reconocimientos por cada carácter. A partir de esto, se puede analizar los caracteres que tienen un alto porcentaje de verdaderos reconocimientos y los que no, con los cuales se tienen problemas.

Tabla 5.1 Verdaderos reconocimientos por caracter

Caracter	Verdaderos Reconocimientos
0	100.00%
1	85.00%
2	87.74%
3	84.26%
4	91.09%
5	93.27%
6	97.87%
7	92.38%
8	87.10%
9	95.12%

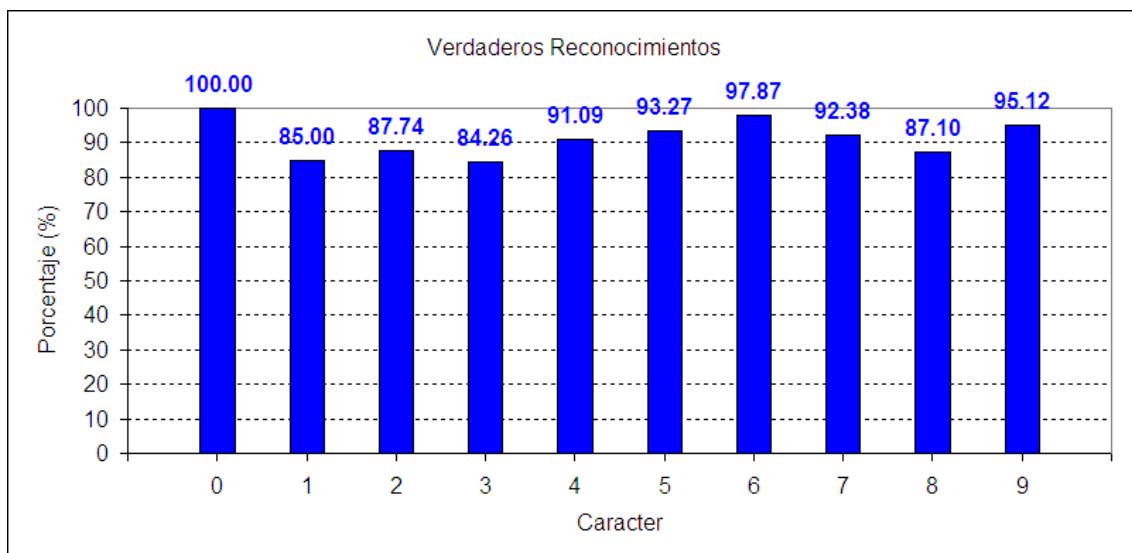


Figura 5.3 Verdaderos reconocimientos por caracter

El análisis se realiza por separado para encontrar los caracteres que presentan más dificultades en su reconocimiento. Debido a ello, se puede identificar el nivel de eficiencia de la red neuronal considerando la información ingresada con los algoritmos propuestos. En la Tabla 5.2 se muestran las cantidades de caracteres reconocidos de acuerdo al carácter que se ingresó a las redes neuronales para reconocerlo.

Tabla 5.2 Carácter a reconocer y carácter reconocido

		Carácter a reconocer									
		0	1	2	3	4	5	6	7	8	9
Carácter reconocido	0	107	3	4	-	2	-	-	-	5	1
	1	-	85	1	3	1	4	1	3	1	-
	2	-	1	93	-	-	1	1	-	2	-
	3	-	6	6	91	-	-	-	3	1	-
	4	-	1	-	-	92	1	-	-	1	-
	5	-	4	-	1	5	97	-	1	-	-
	6	-	-	1	-	1	-	92	1	-	-
	7	-	-	1	13	-	1	-	97	-	-
	8	-	-	-	-	-	-	-	-	81	2
	9	-	-	-	-	-	-	-	-	1	78
No se reconoció		-	-	-	-	-	-	-	-	1	1
Total		107	100	106	108	101	104	94	105	93	82

Con respecto al número cero, se obtuvo como resultado 107 verdaderos reconocimientos de un total de 107 diferentes ceros. Esto significa un porcentaje de 100%. Este resultado se debe a que la forma de escribir el número cero es muy diferente a la de los demás números, por lo que existe una muy baja probabilidad de equivocación. En la Figura 5.4 se muestran algunos ejemplos de números cero escritos por diferentes personas, donde se observa lo explicado anteriormente.

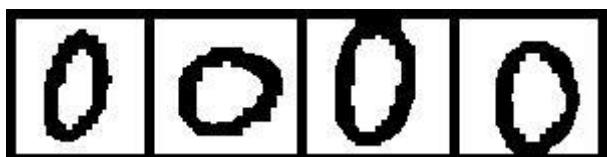


Figura 5.4 Ejemplos del número cero

Cabe resaltar que este porcentaje de verdaderos reconocimientos podría verse perjudicado cuando se escribe muy cerca a la casilla. Esto se puede observar en el caso de procesar un número cero como el de la tercera o cuarta casilla de la Figura 5.4. Debido a ello, en el proceso de eliminar los bordes no se obtendría un cero completamente cerrado. Por lo tanto, el reconocimiento podría fallar. Aunque, a pesar de ello, los algoritmos implementados se encuentran en la capacidad de obtener un correcto reconocimiento.

El porcentaje de verdaderos reconocimientos del número uno es de los más bajos. Se reconocieron correctamente 85 números de 100 diferentes números uno. Esto puede deberse a que se entrenaron las redes neuronales con cuatro formas diferentes de escribir el número uno. Por ello, se puede haber dificultado el entrenamiento. En la Figura 5.5 se observan estas cuatro formas.



Figura 5.5 Diferentes formas de aprendizaje del número uno

Además, en el caso de procesar un número uno como el de la tercera casilla de la Figura 5.5, se tienen algunos inconvenientes en la captura del carácter al realizar el cambio de escala a 32x32. Se ha observado que esto sucede, en especial, cuando el número se encuentra ligeramente inclinado. En estos casos, al escalar se generan unas pequeñas líneas horizontales en los extremos del número, con lo que se parece más al número cinco y en otros casos al número tres. Por lo tanto, no se logra obtener un correcto reconocimiento. En la Figura 5.6 se muestran algunos ejemplos de este inconveniente. En la Figura 5.7 se observa una gráfica con el porcentaje de reconocimientos del número uno.

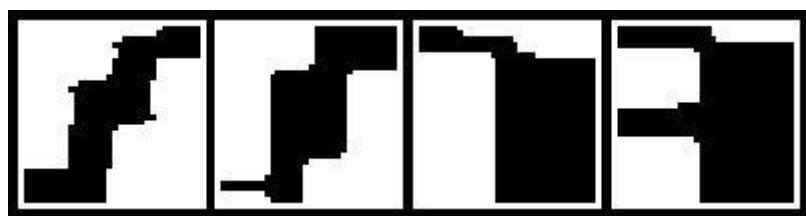


Figura 5.6 Ejemplos del inconveniente del número uno al escalar a 32x32

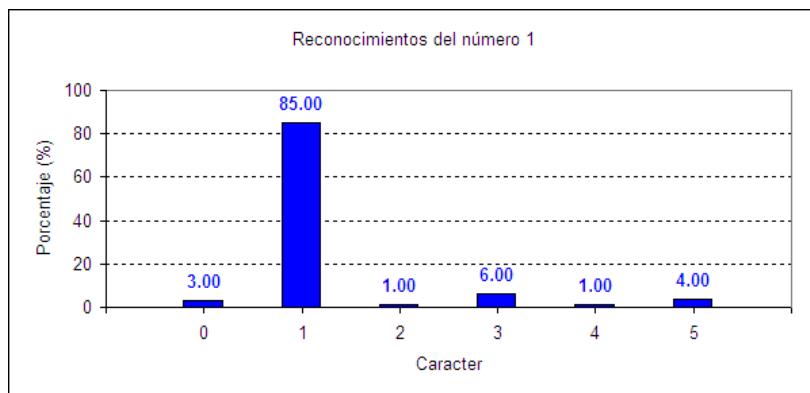


Figura 5.7 Porcentajes de reconocimiento del número uno

Con respecto al número dos, se obtuvo como resultado 93 verdaderos reconocimientos de un total de 106 diferentes números dos. Esto significa un porcentaje de 87.74%. En el proceso de eliminar los bordes, se observa una situación similar al número cero debido a que no se obtendría la parte inferior del número. Por lo tanto, lo reconocería como un siete. En la Figura 5.8 se muestran algunos ejemplos de números dos escritos por diferentes personas. En la cuarta casilla se observa lo explicado anteriormente.



Figura 5.8 Ejemplos del número dos

Además, en algunos casos el sistema reconoce incorrectamente al número dos como un tres. Esto puede deberse a que la forma de escribir el número dos es como la del tres, pero sin terminar, como escribir un tres hasta un poco más de la mitad. Las redes neuronales pueden interpretarlo como ello, por esa razón lo considerarían un tres. En la Figura 5.9 se muestra una gráfica con el porcentaje de reconocimientos del número dos.

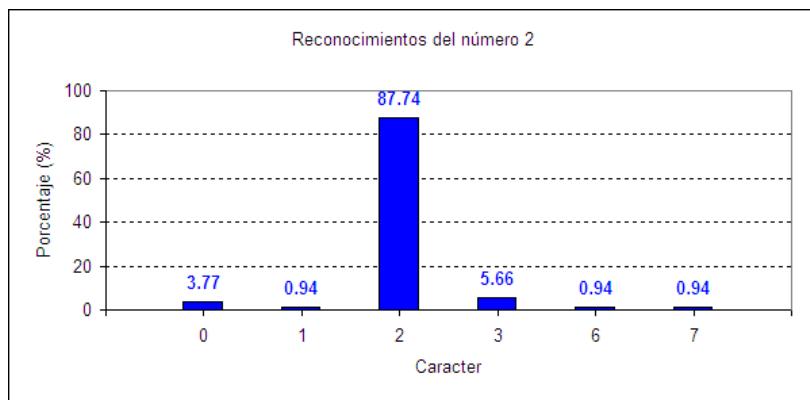


Figura 5.9 Porcentajes de reconocimiento del número dos

El porcentaje de verdaderos reconocimientos del número tres es el más bajo. Se reconocieron correctamente 91 números de 108 diferentes números tres. Se obtiene un porcentaje de 84.26%. En ciertos casos, se reconoce erróneamente como un número siete. Esto se debe a que el número siete con una línea transversal en el medio tiene una forma muy parecida a la del número tres. Además, existen falsos reconocimientos con el número uno de acuerdo a lo explicado anteriormente. En la Figura 5.10 se muestran algunos ejemplos del número tres y en la Figura 5.11 se observa una gráfica con el porcentaje de reconocimientos del número tres.



Figura 5.10 Ejemplos del número tres

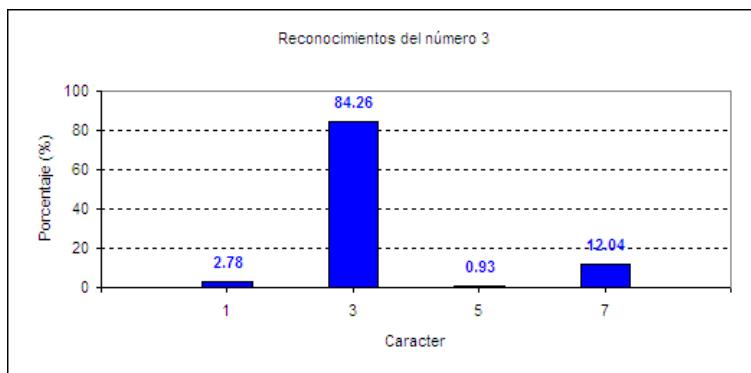


Figura 5.11 Porcentajes de reconocimiento del número tres

Con respecto al número cuatro, se obtuvo como resultado 92 verdaderos reconocimientos de un total de 101 diferentes números cuatro. Esto significa un porcentaje de 91.09%. No se entrenaron los números cuatro de la siguiente forma: 4, porque lo reconocería incorrectamente con el número nueve. En la Figura 5.12 se observan ejemplos del número cuatro.

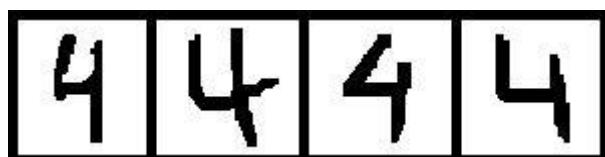


Figura 5.12 Ejemplos del número cuatro

Además, en algunos casos el sistema reconoce incorrectamente al número cuatro como un cinco. Esto puede deberse a que la forma de escribir el número cuatro es un poco parecida a la del cinco. Como escribir un cinco, pero sin el par de líneas horizontales a los extremos y con un pequeño trazo vertical al lado derecho. Las redes neuronales pueden interpretarlo como un cinco, por esa razón puede ocurrir el error en el

reconocimiento. En la Figura 5.13 se observa una gráfica con el porcentaje de reconocimientos del número cuatro.

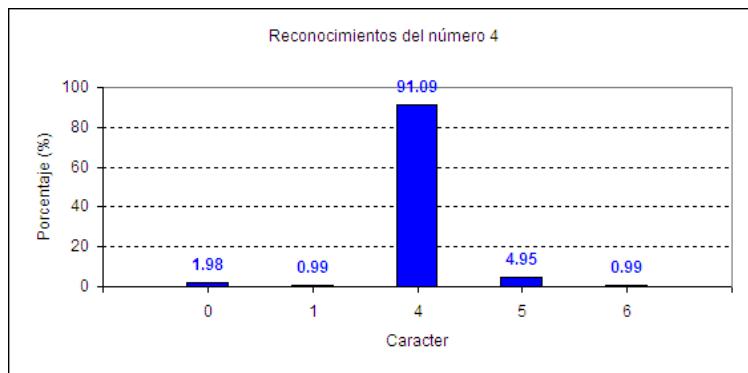


Figura 5.13 Porcentajes de reconocimiento del número cuatro

El porcentaje de verdaderos reconocimientos del número cinco es 93.27%. Se reconocieron correctamente 97 números de 104 diferentes cincos. Existen falsos reconocimientos con el número uno de acuerdo a lo explicado anteriormente. En la Figura 5.14 se muestran algunos ejemplos del número cinco y en la Figura 5.15 se observa una gráfica con el porcentaje de reconocimientos del número.

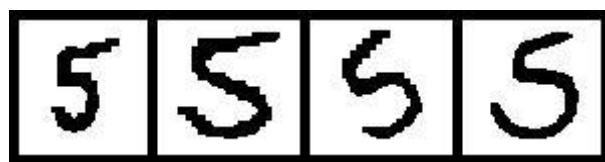


Figura 5.14 Ejemplos del número cinco

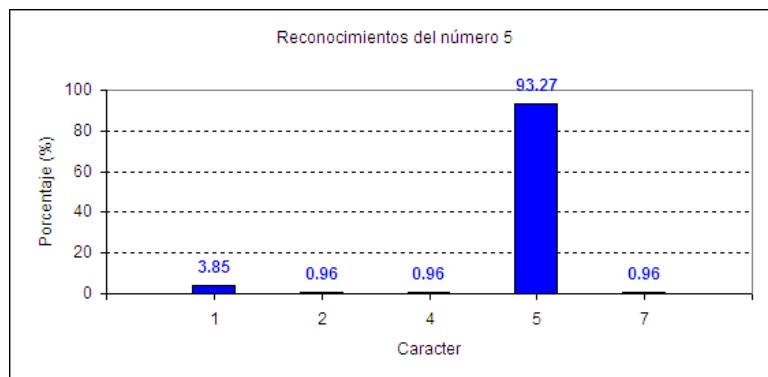


Figura 5.15 Porcentajes de reconocimiento del número cinco

Con respecto al número seis, se obtuvo como resultado 92 verdaderos reconocimientos de un total de 94 diferentes números dos. Esto significa un porcentaje de 97.87%. Este porcentaje se encuentra entre uno de los más altos. En la Figura 5.16 se observan algunos ejemplos del número seis y en la Figura 5.17 se observa una gráfica con el porcentaje de reconocimientos del número.



Figura 5.16 Ejemplos del número seis

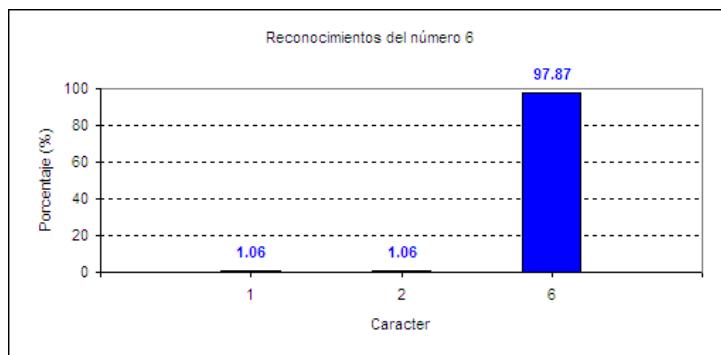


Figura 5.17 Porcentajes de reconocimiento del número seis

Cabe resaltar que este porcentaje de verdaderos reconocimientos podría verse afectado cuando se escribe muy cerca a la casilla, como en el caso de la tercera casilla de la Figura 5.16. Debido a ello, en el proceso de eliminar los bordes no se obtendría un número seis completamente cerrado porque se encuentra muy cerca al borde inferior. Por lo tanto, el reconocimiento podría fallar, identificándolo posiblemente como el número cinco. Aunque, a pesar de ello, los algoritmos implementados se encuentran en la capacidad de obtener un correcto reconocimiento.

El porcentaje de verdaderos reconocimientos del número siete es 92.38%. Se reconocieron correctamente 97 números de 105 diferentes números siete. En algunos casos, cuando el siete no se escribe con una línea transversal en el medio, el sistema lo puede reconocer como el número uno. Además, en ciertos casos, se reconoce

erróneamente como un número tres según lo explicado anteriormente. En la Figura 5.18 se muestran ejemplos del número siete. En la cuarta casilla se observa un ejemplo de un siete escrito sin la línea transversal. En la Figura 5.19 se muestra una gráfica con el porcentaje de reconocimientos del número.

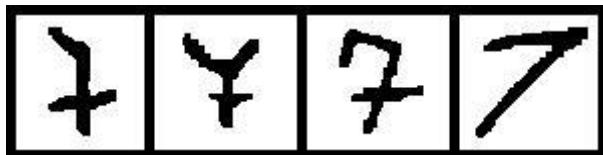


Figura 5.18 Ejemplos del número siete

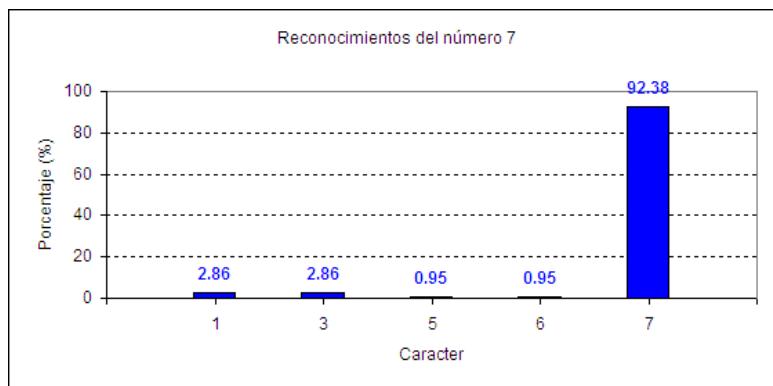


Figura 5.19 Porcentajes de reconocimiento del número siete

Con respecto al número ocho, se obtuvo como resultado 81 verdaderos reconocimientos de un total de 93 diferentes números ocho. Esto significa un porcentaje de 87.10%. Este porcentaje se encuentra entre uno de los más bajos. Puede deberse a que la forma de escribir el número ocho es muy diferente entre las personas. En la Figura 5.20 se muestran algunos ejemplos de números ocho escritos por diferentes personas utilizados para el entrenamiento de las redes neuronales, donde se observa lo explicado anteriormente.

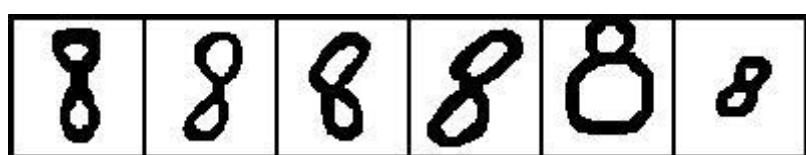


Figura 5.20 Ejemplos de aprendizaje del número ocho

Además, en el caso de procesar un número ocho escrito como el de la quinta casilla de la Figura 5.20, es posible que el sistema lo reconozca como un número cero. Esto podría deberse a que el número ocho es escrito con un círculo inferior de mayor tamaño en comparación con el círculo superior. Por lo tanto, tiende a parecerse más al número cero que a un ocho. En la Figura 5.21 se muestra una gráfica con el porcentaje de reconocimientos del número ocho, donde la X significa que no se reconoció el carácter.

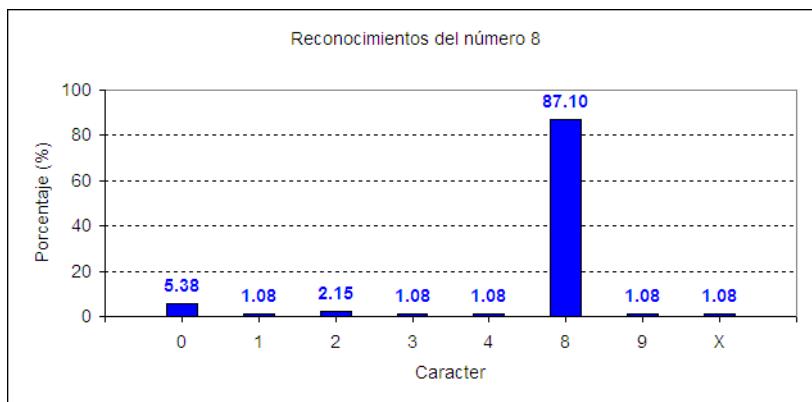


Figura 5.21 Porcentajes de reconocimiento del número ocho

Cabe resaltar que este porcentaje de verdaderos reconocimientos podría verse aún más perjudicado cuando se escribe muy cerca a la casilla. Debido a ello, en el proceso de eliminar los bordes no se obtendría un ocho completamente cerrado. Sin importar que no se complete la parte superior del número o la inferior. Por lo tanto, el reconocimiento podría fallar.

El porcentaje de verdaderos reconocimientos del número nueve es 95.12%. Se reconocieron correctamente 78 números de 82 diferentes nueves. Este porcentaje se encuentra entre uno de los más altos. En algunos casos, el sistema se equivocó en reconocer el número nueve por el número ocho. Esto sucede porque algunas personas terminan de escribir el número nueve con una línea curva, no con una línea recta. Por lo tanto, las redes neuronales pueden interpretarlo como un número ocho que no se encuentra completamente cerrado. En la Figura 5.22 se observan algunos ejemplos del número nueve. En la tercera y cuarta casilla se muestra la forma de escribir el nueve explicada anteriormente. En la Figura 5.23 se muestra una gráfica con el porcentaje de

reconocimientos del número nueve, donde la X significa que no se reconoció el carácter.



Figura 5.22 Ejemplos del número nueve

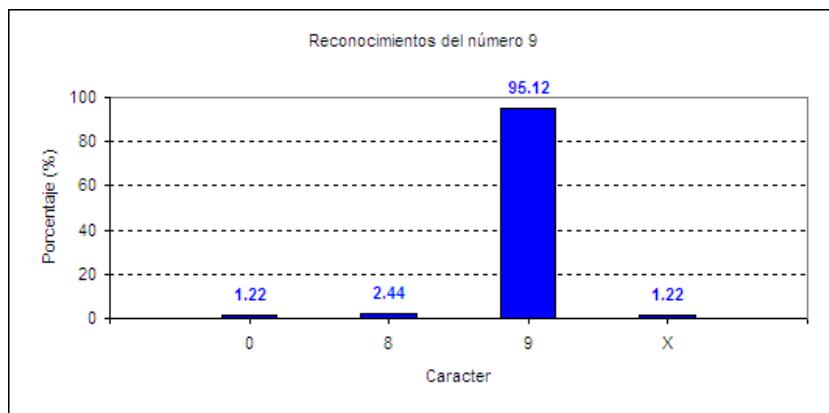


Figura 5.23 Porcentajes de reconocimiento del número nueve

Además, este porcentaje de verdaderos reconocimientos podría verse afectado cuando se escribe muy cerca a la casilla. Debido a ello, en el proceso de eliminar los bordes no se obtendría un nueve completamente cerrado. Esto sucede cuando se escribe muy cerca de la línea superior de la casilla, por lo que el número sería reconocido como un cuatro. Esto podría suceder el caso de procesar un número nueve como el de la segunda casilla de la Figura 5.22.

Se producen algunos errores en la fase de reconocimiento. Esto se debe a que al trabajar con redes neuronales se tiene un error que permite una buena generalización. También podrían deberse al proceso utilizado para la captura de cada carácter, lo que conllevaría que el error se origine en la fase de captura y no en la de reconocimiento.

Por otro lado, se calculó que el proyecto diseñado cuenta con un tiempo promedio de reconocimiento de 0.10 segundos por carácter. Los formularios diseñados son

conformados por 26 casillas, por lo que al sistema le tomaría 2.60 segundos para reconocer los caracteres. En el caso de la ficha óptica (formato de evaluación), por solo analizarse un segmento, el proyecto actualizaría la base de datos en no más de 1 segundo.

Por último, en la Tabla 5.3 se muestra el informe económico final del proyecto. Para realizar este informe se consideró el tiempo que se invirtió al proyecto para diseñar los algoritmos, aplicarlos y validar sus resultados. Además, se valoraron los conocimientos empleados. Esto se agregó a la lista de componentes del proyecto.

Tabla 5.3 Informe económico final

Proyecto	Precio en soles (S/.)
Cámara web	340
Soporte para cámara web	100
Soporte para formularios	20
Lámpara (Cantidad 2)	34
Foco (Cantidad 2)	32
Desarrollo del software	224
TOTAL	750

5.1. Comentarios finales

En el capítulo 5 se explicaron las pruebas realizadas y los resultados obtenidos del proyecto. El sistema propuesto logra un porcentaje de 91.30% (porcentaje sobre el total de caracteres), porque se identificaron 913 verdaderos reconocimientos de un total de 1000 caracteres numéricos escritos a mano. Además, se realiza un análisis por separado para encontrar los caracteres que presentan más dificultades en su reconocimiento. Por lo tanto, se detallan los porcentajes obtenidos por cada carácter. En los siguientes

capítulos se definirán las conclusiones y las recomendaciones para trabajos futuros con respecto al proyecto diseñado.

CONCLUSIONES

- Se debe tener una buena iluminación en el recinto y un buen enfoque de la cámara web para una mejor captura de la imagen. De no contar con ello se vería perjudicada la correcta captura de los caracteres. Esto significa que no importa que tan buenos sean los resultados con los algoritmos utilizados, si el inicio del proceso es erróneo los resultados serán erróneos también. Por ello se colocan las dos lámparas en el soporte de formularios, para que no exista diferencia de iluminación si se trabaja de día o de noche.
- De preferencia, se deben escribir con un trazo continuo los caracteres en el formulario. Con ello se evitarían posibles errores al capturar el contorno, debido a que se podría obtener sin que esté completamente cerrado o solo una parte del carácter. Aunque en el proceso de captura del carácter, el filtro gaussiano sirve para ayudar a que no existan espacios en el trazo del carácter. Esto resulta muy útil para los descriptores de Fourier.
- Se pueden utilizar diversas cámaras web que cumplan con las especificaciones necesarias en el proyecto. Esto se considera una gran ventaja debido a que no se está limitando el sistema a un solo modelo de cámara. Solo se debería tener la opción de modificar manualmente las propiedades de la cámara que se utilizarían y que se pueda obtener una foto con la resolución requerida.
- Para una correcta binarización de la imagen, se calcula un valor umbral ideal que capture adecuadamente los caracteres del formulario llenado. Este valor depende de si se está analizando los formularios o la ficha óptica (formato de evaluación). Al obtener este valor a partir del histograma de la imagen capturada, se tiene la particularidad de que puede adaptarse de acuerdo a cada foto capturada de los formularios.
- Para ejecutar la segmentación de la imagen, en el proceso de binarización se invierten los valores de los píxeles. Esto se realiza para que los caracteres y las casillas cuenten con información debido a que tienen píxeles con valor (255). Por lo tanto, el fondo de los formularios contendría valores de píxeles igual a 0. Con ello,

al realizar la proyección vertical de la imagen binarizada se logran obtener las posiciones de los segmentos del formulario. En el caso de la ficha óptica no se ejecutaría la segmentación porque sólo se enfoca en una porción de la ficha.

- Los descriptores de Fourier son insensibles a la traslación, rotación y cambio de escala. Además, los primeros descriptores indican la forma general del carácter, por ello solo se utilizan los 10 primeros para el entrenamiento de las redes neuronales. Estos descriptores de Fourier son de gran utilidad cuando los caracteres están escritos de forma muy inclinada en las casillas. En estos casos, los demás algoritmos podrían tener algunas dificultades porque obtendrían información que no sea tan similar a otros ejemplos del mismo número. Los descriptores sirven de apoyo y se complementan con los algoritmos.
- El entrenamiento de las redes neuronales se finalizó al obtener un error cuadrático medio de 4.9994×10^{-7} . Un error menor no permitiría una buena generalización, los caracteres a reconocer tendrían que ser casi idénticos a los de entrenamiento. La generalización es una medida de un correcto comportamiento de la red neuronal ante nuevos datos de entrada al completar su entrenamiento.
- Cabe resaltar que el problema en la identificación de caracteres no se debe en gran medida a la red neuronal, pues se obtuvo un buen resultado de verdaderos reconocimientos (91.30%), sino a la extracción de información de los caracteres con los algoritmos propuestos. También, al proceso de captura del carácter. Se debe considerar elaborar otros algoritmos y añadirlos a los propuestos para obtener más información de los caracteres y con ello más precisión de los detalles de cada uno.
- El entrenamiento de la red neuronal brinda los pesos necesarios para posteriormente realizar el reconocimiento. Por lo tanto, no es necesario entrenar cada vez que se desee reconocer un carácter y por ello el costo computacional es bajo. Además, los algoritmos diseñados para obtener la información del carácter no son computacionalmente costosos. Debido a esto, el tiempo de reconocimiento también resulta bajo, obteniendo 0.10 segundos por carácter.

RECOMENDACIONES PARA TRABAJOS FUTUROS

Se podría modificar la base de datos para que solo se pueda actualizar por medio del sistema de reconocimiento. Con ello, se anularía la modificación manual de la base de datos y se evitaría la manipulación de la información. Esto sería de gran importancia para las actas de escrutinio y así evitar posibles fraudes.

Una recomendación para obtener más información del carácter podría ser utilizar los métodos que se basan en las propiedades del carácter. Uno de ellos es analizar el lazo del carácter, en el caso de los números serían el 0, 6, 8 y 9. Además, se utiliza el concepto del depósito de agua, como si se derramara agua en los lados del carácter (arriba, abajo, derecha e izquierda). Por último, se analiza la concavidad de los perfiles del carácter¹¹⁷.

Además, al obtener más información relevante de acuerdo a los algoritmos adicionados, se podría ampliar el reconocimiento a letras escritas a mano, no sólo números. Por lo tanto, se adicionaría el reconocimiento de formularios que incluyan nombres, direcciones o correos electrónicos. Para ello también se tendría que aumentar la cantidad de ejemplares de cada número y letra, para que se efectúe un mejor entrenamiento de las redes neuronales.

Por otro lado, se podría modificar el recinto para colocar un grupo de formularios y se capture automáticamente la información de cada uno de ellos. Para ello, se conectaría a una tarjeta (por ejemplo Arduino) para que pase a otro formulario cuando se haya actualizado la base de datos. Además, se programaría para que detenga el proceso en caso no se obtenga una captura de la información, para que no se actualice la base de datos.

¹¹⁷ Cfr. Chanda, Pal y Ramos Terrades 2009.

ANEXO

Manual de operación de la interfaz visual

Se deben seguir los siguientes pasos para lograr una correcta actualización de la base de datos. En este trabajo se tiene tres tipos de formularios para ser completados e ingresar la información a sus respectivos campos. La base de la cámara se debe colocar a 12 cm del eje vertical en el caso de los formularios 1 y 2, y a 5.8 cm para el formulario 3.

1. Colocar la cámara web en su soporte y el formulario completo en su soporte. En el caso del formulario tipo exámenes de admisión (número 3), se debe colocar en la parte superior del soporte. Encender las lámparas y conectar la cámara web a la computadora. En la Figura A1.1 se observan estos pasos.



Figura A1.1 Primer paso a seguir

2. Abrir la interfaz visual y llenar la casilla del número de formulario a analizar. En la Figura A1.2 se observa este paso.

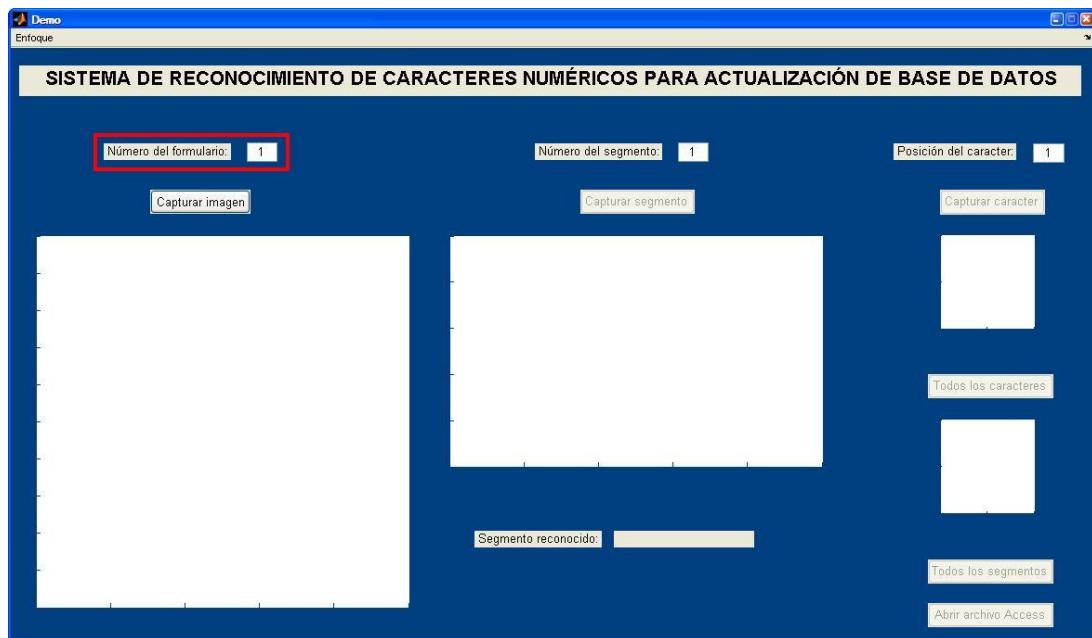


Figura A1.2 Segundo paso a seguir

3. Presionar el botón “Capturar imagen”. En la Figura A1.3 se observa este paso.

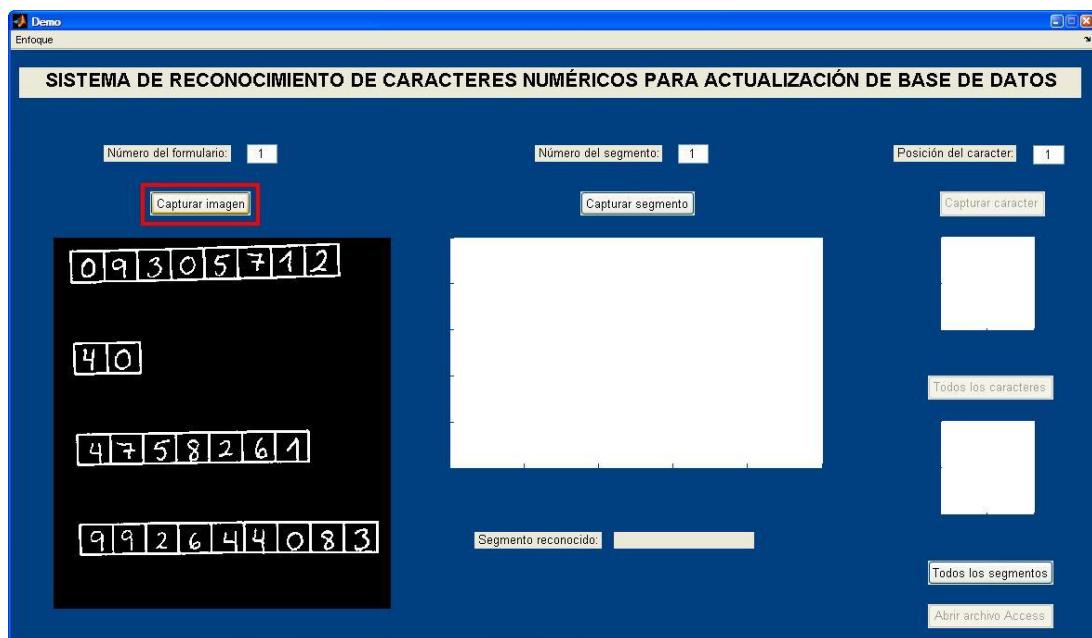


Figura A1.3 Tercer paso a seguir

4. Para analizar segmento por segmento, debe elegir un número de segmento llenando la casilla y presionar el botón “Capturar segmento”. Esta opción no está habilitada para el formulario 3 por ser sólo un segmento. En la Figura A1.4 se observa este paso.

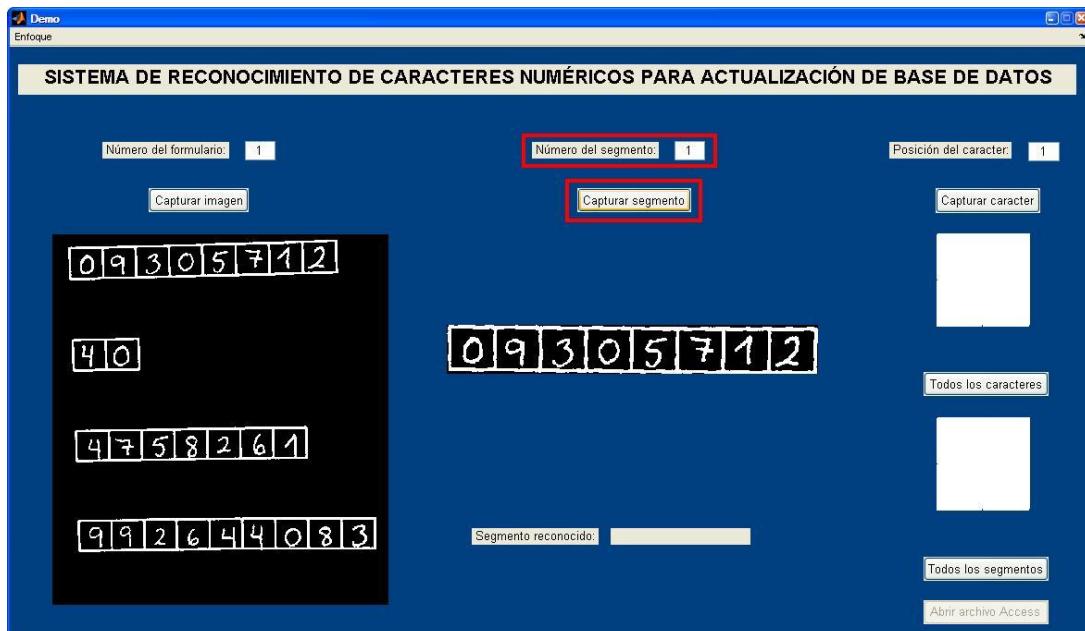


Figura A1.4 Cuarto paso a seguir

5. Para analizar carácter por carácter, debe elegir una posición de los caracteres del segmento elegido (llenando la casilla) y presionar el botón “Capturar carácter”. Aparecerá un mensaje que indique el reconocimiento obtenido. Luego presionar “OK” en la ventana del mensaje para continuar. En la Figura A1.5 se observa este paso.
6. Para analizar todos los caracteres del segmento, debe presionar el botón “Todos los caracteres”. Aparecerán los mensajes con el reconocimiento obtenido, pero se cerrarán automáticamente. Además, se irán agregando los reconocimientos en la línea de “Segmento reconocido”. En la Figura A1.6 se observa este paso.

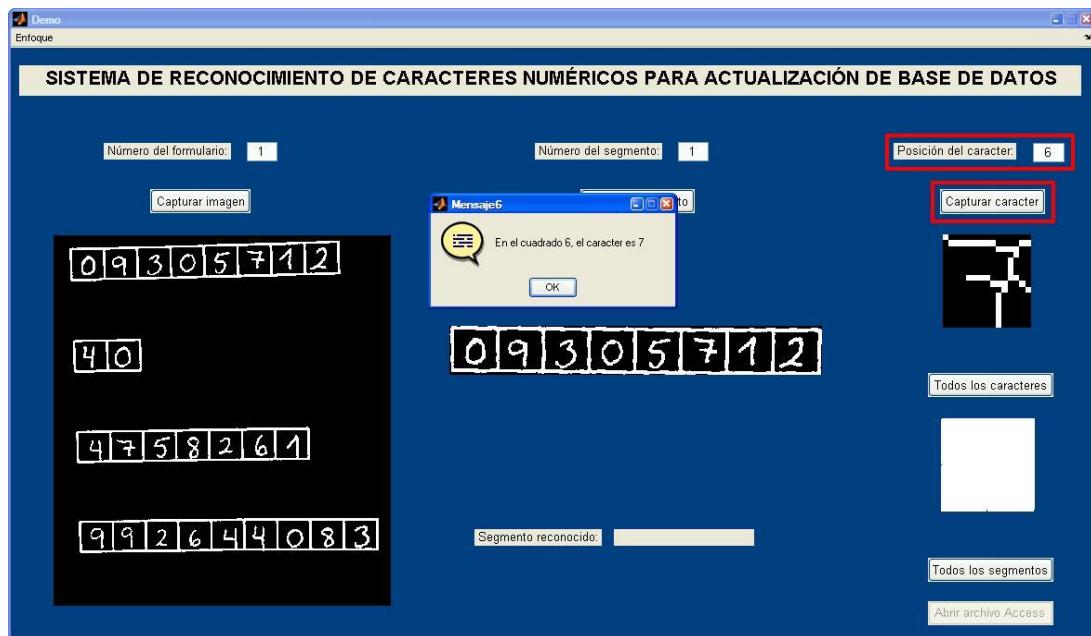


Figura A1.5 Quinto paso a seguir

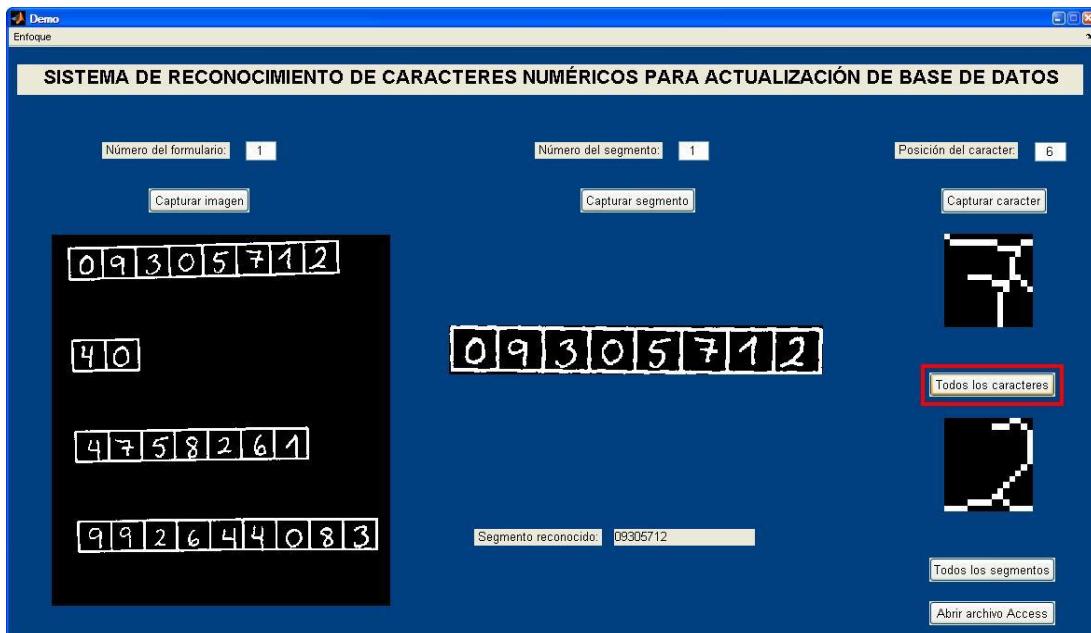


Figura A1.6 Sexto paso a seguir

7. Para analizar todos los segmentos del formulario, debe presionar el botón “Todos los segmentos”. Esta opción no está habilitada para el formulario 3 por ser sólo un segmento. En la Figura A1.7 se observa este paso.

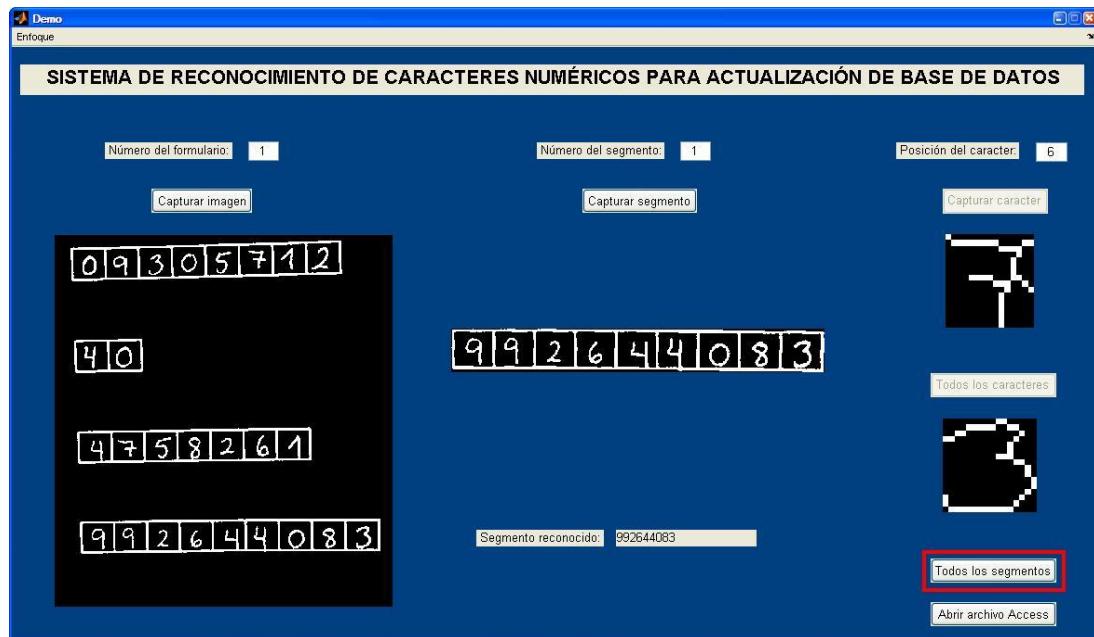


Figura A1.7 Séptimo paso a seguir

8. Presionar el botón “Abrir archivo Access” para comprobar la actualización de la base de datos. En la Figura A1.8 se observa este paso. En la Figura A1.9 se muestra la base de datos actualizada.
9. Cerrar la ventana de la interfaz visual al finalizar el proceso. En la Figura A1.10 se observa este paso.

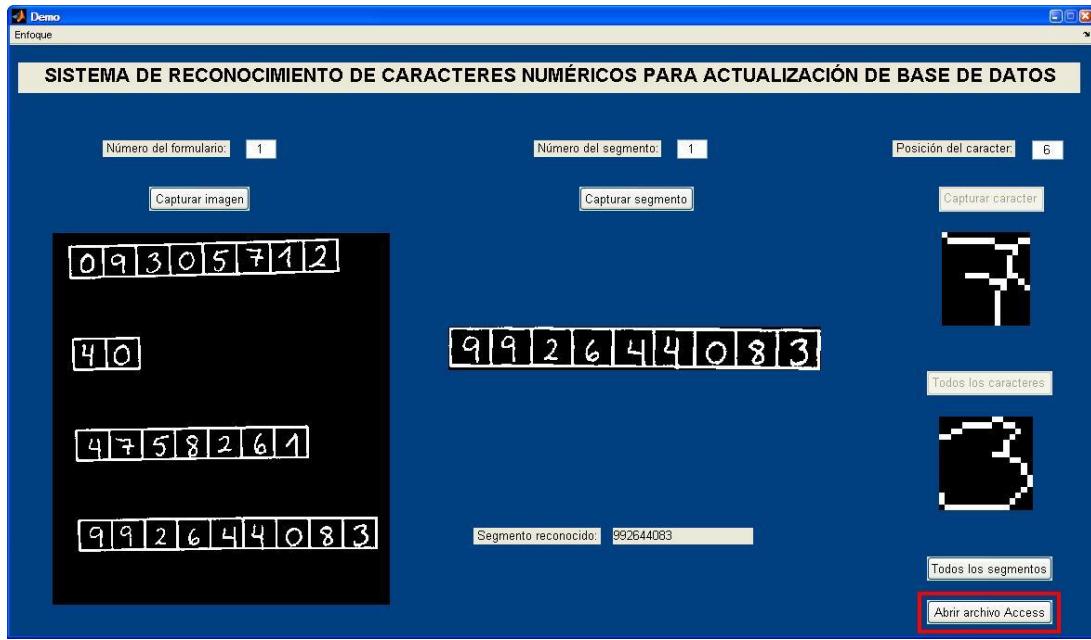


Figura A1.8 Octavo paso a seguir

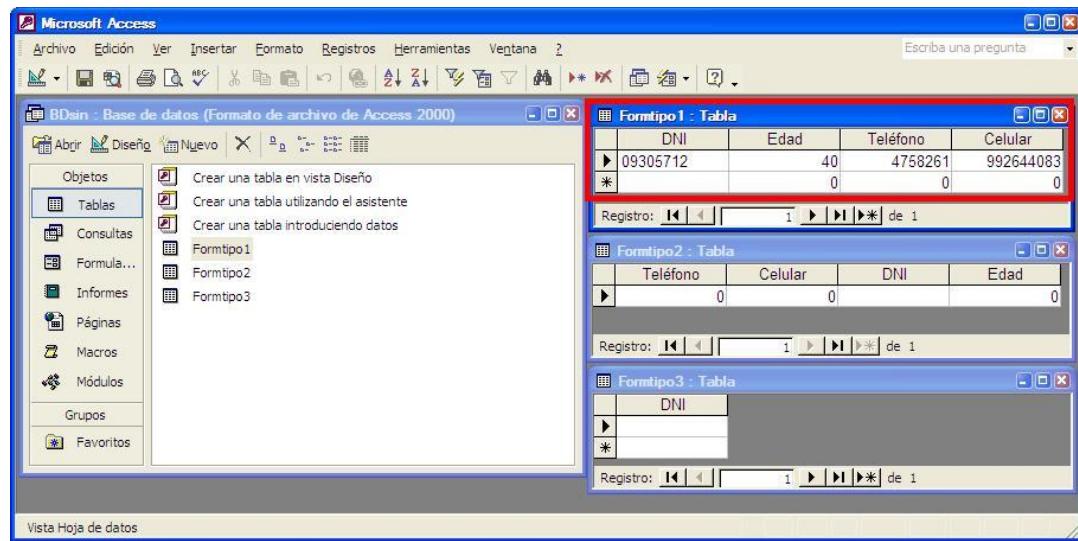


Figura A1.9 Base de datos actualizada

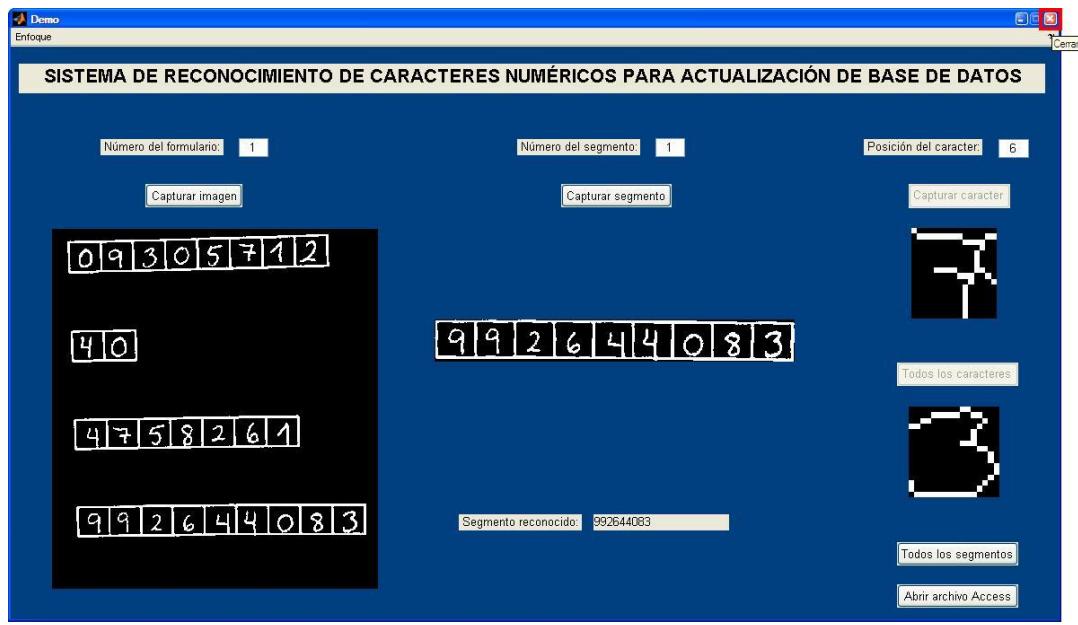


Figura A1.10 Noveno paso a seguir

BIBLIOGRAFÍA

- [1] EMPLEOS BUMERAN PERÚ (2016) (<http://www.bumeran.com.pe/empleos-busqueda-digitador.html>) Sitio web de Bumeran; contiene ofertas de trabajo en el Perú (consulta: 24 de octubre).
- [2] COMPUTRABAJO (2016) (<http://www.computrabajo.com.pe/trabajo-de-digitador>) Sitio web de CompuTrabajo; contiene ofertas de trabajo en el Perú (consulta: 24 de octubre).
- [3] PHOKHARATKUL, Pisit y KIMPAN, Chom (2002) Handwritten Thai character recognition using Fourier descriptors and genetic neural networks, pp. 270-293. En: Computational Intelligence, vol. 18, No. 3.
- [4] CHUNG, Yuk Ying y WONG, Man to (1997) Handwritten character recognition by Fourier descriptors and neural network, pp. 391-394. En: IEEE Conf. on Speech and Image Technologies for Computing and Telecommunications.
- [5] RAJPUT, Ganapatsingh y MALI, Shankar (2010) Fourier descriptor based isolated Marathi handwritten numeral recognition, pp. 9-13. En: International Journal of Computer Applications, vol. 3, No. 4.
- [6] SINGH, Dayashankar; SINGH, Sanjay y DUTTA, Maitreyee (2010) Hand written character recognition using twelve directional feature input and neural network, pp. 82-85. En: International Journal of Computer Applications, vol. 1, No. 3.
- [7] HALLALE, Sumedha y SALUNKE, Geeta (2013) Offline handwritten digit recognition using neural network, pp. 4373-4377. En: International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, No. 9.

- [8] AHMAD, Zaheer y otros (2007) Urdu nastaleeq optical character recognition, pp. 249-252. En: Proceedings of World Academy of Science, Engineering and Technology, vol. 26.
- [9] ZAND, Mohsen; NAGHSH, Ahmadreza y AMIRHASSAN, S. (2008) Recognition-based segmentation in Persian character recognition, pp. 183-187. En: Proceedings of World Academy of Science, Engineering and Technology, vol. 28.
- [10] ABBYY (2017) (<https://www.abbyy.com/en-apac/products/>) Sitio web de ABBYY, contiene productos de software de captura de datos y procesamiento de formularios en papel y archivos PDF (consulta: 11 de mayo).
- [11] A2IA TEXTREADER (2017) (<http://www.a2ia.com/es/a2ia-textreader>) Sitio web de A2iA TextReader, componente de reconocimiento dedicado a transcripción completa de documentos impresos y manuscritos (consulta: 11 de mayo).
- [12] ACTIVE RECORD SOFTWARE SYSTEMS (2017) (<http://activerecord.com/>) Sitio web de Active Record Software Systems, Ltd., contiene información del conjunto de componentes de desarrollo de software OcuLens (consulta: 11 de mayo).
- [13] OFICINA NACIONAL DE PROCESOS ELECTORALES (ONPE) (2014) Cartilla de instrucción para miembros de mesa – Elecciones Municipales 2014. Lima.
- [14] SISTEMA DE FOCALIZACIÓN DE HOGARES (SISFOH) (2016) (<http://www.sisfoh.gob.pe.empadronamiento.shtml?x=409>) Sitio web de SISFOH; contiene información del empadronamiento de hogares y la ficha socioeconómica única (consulta: 13 de agosto).

- [15] FREEDOM SCIENTIFIC (2016) (<http://www.freedomscientific.com/JAWSHQ/JAWSHeadquarters01>) Sitio web de Freedom Scientific; contiene información del programa JAWS (consulta: 13 de agosto).
- [16] RAMOS, Miguel Ángel (2017) La luz blanca en la visión humana (consulta: 28 de enero). (<http://www.lucescei.com/estudios-y-eficiencia/iluminacion-y-salud/la-luz-blanca-en-la-vision-humana/>)
- [17] CASA FIJA (2017) Luz blanca vs Luz amarilla (consulta: 28 de enero). (<http://casafija.com/luz-blanca-vs-luz-amarilla/>)
- [18] ELEGIRELECTRONICA.COM (2017) Características de una cámara digital hogareña (consulta: 06 de mayo). (http://elegirelectronica.com/camara_digital.html)
- [19] QUECAMARAREFLEX.COM (2015a) Cómo funciona el sensor de una cámara digital (consulta: 25 de octubre de 2017). (<http://quecamarareflex.com/como-funciona-el-sensor-de-una-camara-digital/>)
_____ (2015b) Cómo funciona el enfoque automático en las cámaras (consulta: 06 de mayo de 2017). (<http://quecamarareflex.com/como-funciona-el-enfoque-automatico-en-las-camaras/>)
- [20] FOTOGRAFÍA PARA PRINCIPIANTES (2012) Longitud Focal, distancia de enfoque y Profundidad del campo (Día 8) (consulta: 25 de octubre de 2017). (<http://fotografiaparaprincipiantes.blogspot.pe/2012/06/longitud-focal-distancia-de-enfoque-y.html>)

- [21] WIKIPEDIA (2017a) Ángulo de visión (consulta: 06 de mayo).
(https://es.wikipedia.org/wiki/Ángulo_de_visión)
_____ (2017b) Resolución de imagen (consulta: 25 de octubre).
(https://es.wikipedia.org/wiki/Resolución_de_imagen)
_____ (2017c) Fotogramas por segundo (consulta: 25 de octubre).
(https://es.wikipedia.org/wiki/Fotogramas_por_segundo)
_____ (2017d) Luminosidad (color) (consulta: 07 de mayo).
([https://es.wikipedia.org/wiki/Luminosidad_\(color\)](https://es.wikipedia.org/wiki/Luminosidad_(color)))
- [22] EL PAISAJE PERFECTO (2013) Relación de aspecto en fotografía (consulta: 25 de octubre de 2017).
(<http://www.elpaisajeperfecto.com/2013/08/relacion-de-aspecto-en-fotografia.html>)
- [23] GONZALEZ, Rafael y WOODS, Richard (2008) Digital Image Processing. Tercera edición. New Jersey: Pearson Prentice Hall.
- [24] THE MATHWORKS INC. (2016) Symbolic Math Toolbox: User's Guide. Mathworks Inc.
- [25] ARNÁEZ, Enrique (2007) Robótica e Inteligencia Artificial. Material de clase. Lima: UPC.
- [26] CILIMKOVIC, Mirza. Neural networks and back propagation algorithm. Instituto de Tecnología Blanchardstown, Dublin, Irlanda.
- [27] KRÖSE, Ben y VAN DER SMAGT, Patrick (1996) An introduction to Neural Networks. Octava edición. Amsterdam: Universidad de Amsterdam.
- [28] THE MATHWORKS INC. (2017) GUI de MATLAB (consulta: 05 de mayo).
(<https://es.mathworks.com/discovery/matlab-gui.html>)

- [29] MICROSOFT OFFICE (2017a) ACCESS (consulta: 05 de mayo).
(<https://products.office.com/es/access>)
_____ (2017b) Conceptos básicos sobre base de datos (consulta: 29 de agosto).
(<https://support.office.com/es-es/article/Conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>)
- [30] LOGITECH (2016) (http://support.logitech.com/en_us/product/quickcam-pro-9000) Sitio web de Logitech; contiene las características de la cámara web QuickCam Pro 9000 y su manual de usuario (consulta: 31 de julio).
- [31] UNIVERSIDAD POLITÉCNICA DE MADRID (UPM) (2017) Adquisición de imágenes y entornos gráficos (consulta: 29 de enero).
(http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/PracticasVA/prac2VA_AdquisicionGUI.pdf)
- [32] CHANDA, Sukalpa; PAL, Umapada y RAMOS TERRADES, Oriol. (2009) Word-wise thai and roman script identification. En: ACM Transactions on Asian Language Information Processing, vol. 8, No. 3, artículo 11, 21 páginas.