

**UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE INGENIERÍA CIVIL**



**RECONOCIMIENTO DE CARACTERES EN CAMPOS  
DE FORMULARIOS USANDO REDES NEURONALES**

**INFORME DE SUFICIENCIA**

**Para optar el Título Profesional de:**

**INGENIERO CIVIL**

**CARLOS ALBERTO DE LA CRUZ LÁZARO**

**Lima- Perú**

**2012**

## *DEDICATORIA*

*A mi madre y a mi padre que siempre me apoyaron en mi educación.*

*A mis hermanos José y Ramón que me enseñaron nociones y disciplina de programación.*

	<b>Pág.</b>
<b>RESUMEN .....</b>	<b>3</b>
<b>LISTA DE FIGURAS .....</b>	<b>5</b>
<b>LISTA DE TABLAS.....</b>	<b>7</b>
<b>LISTA DE SIGLAS .....</b>	<b>8</b>
<b>LISTA DE SÍMBOLOS .....</b>	<b>8</b>
<b>LISTA DE PROGRAMAS.....</b>	<b>9</b>
<b>INTRODUCCIÓN.....</b>	<b>10</b>
<b>CAPÍTULO I.- GENERALIDADES .....</b>	<b>11</b>
1.1. INTRODUCCIÓN .....	11
1.2. PLANTEAMIENTO DEL PROBLEMA.....	12
1.3. OBJETIVO DEL ESTUDIO .....	12
1.4. ETAPAS DEL SISTEMA ICR DESARROLLADO.....	12
1.5. METODOLOGÍA DE TRABAJO .....	15
<b>CAPÍTULO II.- MARCO TEÓRICO.....</b>	<b>16</b>
2.1. INTELIGENCIA ARTIFICIAL Y REDES NEURONALES.....	16
2.2. ÁREAS DE ESTUDIO EN LA INTELIGENCIA ARTIFICIAL .....	16
2.3. REDES NEURONALES ARTIFICIALES .....	17
2.3.1. Usos.....	18
2.3.2. Estructura de una Red Neuronal Artificial .....	18
2.3.3. Tipos de Redes Neuronales .....	22
2.3.4. Algoritmo de aprendizaje.....	24
2.3.5. Algoritmo de retropropagación (backpropagation).....	24
2.4. REDES NEURONALES EN MATLAB .....	24
2.4.1. Redes “Feedforward”.....	25
2.4.2. Funciones de entrenamiento .....	26
2.4.3. Diseño de la Red Neuronal.....	27

<b>CAPÍTULO III.- APLICACIÓN DE REDES NEURONALES A CATASTRO .....</b>	<b>31</b>
3.1. ANTECEDENTES .....	31
3.2. DESCRIPCIÓN DEL ALGORITMO .....	32
3.3. DESARROLLO DEL PROGRAMA EN MATLAB .....	33
3.3.1. Resultado del reconocimiento de datos en una ficha catastral .....	53
3.3.2. Apuntes Complementarios .....	55
3.4. RESUMEN DEL PROCEDIMIENTO.....	55
3.5. COMPARACIÓN CON EL PROGRAMA SMARTZONE. ....	56
<b>CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>59</b>
4.1. CONCLUSIONES.....	59
4.2. RECOMENDACIONES .....	60
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>ANEXOS .....</b>	<b>65</b>
Anexo A: Código del programa “CARLOSICR” en MATLAB .....	67
Anexo B: Mini Guía del MATLAB .....	81
Anexo C: Ejemplo de reconocimiento de patrones del MATLAB.....	97
Anexo D: Glosario.....	113
Anexo E: Fichas Catastrales Individuales escaneadas .....	115

## RESUMEN

El estudio tiene como antecedente el Levantamiento Catastral del Centro Poblado Santa Cruz, distrito de Végueta, provincia de Huaura, departamento de Lima, que se realizó en el año 2011 y que abarca 300 registros obtenidos de un trabajo de llenado de fichas catastrales de un total 19 manzanas, según se indica en el "EXPEDIENTE TÉCNICO DEL CATASTRO URBANO DEL CENTRO POBLADO "SANTA CRUZ" VEGUETA – HUAURA - LIMA CON APLICACIONES GIS".

El problema fundamental es transferir la información obtenida en el levantamiento catastral de las fichas catastrales a una base de datos digital, porque ocasiona un gasto de recursos, tiempo y mano de obra adicional. Para esto, la solución que se plantea es desarrollar un programa en MATLAB para reconocer el texto escrito en las fichas catastrales a partir de una imagen escaneada de la misma, para luego convertirla en un archivo de texto.

Finalmente, se desarrolla un programa llamado "CARLOSICR" con los programas "framework": "Neural Network Toolbox" e "Imaging Processing Toolbox" del MATLAB 2011b, los cuales son un conjunto de funciones implementadas para crear aplicaciones de MATLAB.

El programa reconoce texto escrito a mano en fichas de catastro urbano escaneadas y realiza las siguientes funciones:

Primero, hace la representación binaria de la imagen de texto (en dos colores blanco y negro), también limpia las manchas (conocidas como "ruido" en el contexto de las redes neuronales) del texto escaneado.

Segundo, obtiene imágenes individuales del texto escrito a mano de todos los campos de registro de la ficha catastral.

Tercero, segmenta y recorta cada imagen de texto de cada campo de registro de la ficha catastral para obtener individuales de letras y números.

Cuarto, crea y entrena la red neuronal artificial para clasificar los caracteres.

Quinto, crea un archivo de salida en extensión csv.

El tipo de red que se utiliza para darle solución al problema del reconocimiento óptico de caracteres es una red neuronal tipo multicapa denominada Backpropagation (con retroalimentación).

El aporte es dejar un programa que reconozca texto escrito en las fichas catastrales y lo convierta a texto ASCII, el cual es reconocido por cualquier procesador de textos como el Bloc de notas, Microsoft Word o Microsoft Excel de Windows, automatizando la etapa de transferencia de datos.

## LISTA DE FIGURAS

	Pág.
<b>Figura N° 1.01.-</b> Diagrama de flujo del reconocimiento de caracteres. ....	13
<b>Figura N° 1.02.-</b> Diagrama de flujo del reconocimiento de texto. ....	14
<b>Figura N° 2.01.-</b> Estructura de una neurona biológica. ....	19
<b>Figura N° 2.02.-</b> Estructura de una neurona artificial. ....	20
<b>Figura N° 2.03.-</b> Estructura de una Red Neuronal Artificial. ....	21
<b>Figura N° 2.04.-</b> Funciones de Activación: Función de grado y tangente hiperbólica (De izquierda a derecha). ....	22
<b>Figura N° 2.05.-</b> Estructura de la red neuronal utilizada. ....	23
<b>Figura N° 2.06.-</b> Modelo de red neuronal para el reconocimiento de caracteres. Fuente: MATLAB. ....	27
<b>Figura N° 3.01.-</b> Configuración de la memoria RAM máxima a usar por el MATLAB. ....	33
<b>Figura N° 3.02.-</b> Cambiar de coma a punto el separador decimal en MATLAB. ....	34
<b>Figura N° 3.03.-</b> Ubicación de la carpeta de trabajo. ....	35
<b>Figura N° 3.04.-</b> Letra D escrita a mano escaneada y guardada como D.jpg ....	35
<b>Figura N° 3.05.-</b> Imagen en formato binario. ....	38
<b>Figura N° 3.06.-</b> Imagen en formato binario y filtrado de imperfecciones. ....	38
<b>Figura N° 3.07.-</b> Ficha catastral escaneada como imagen binaria. ....	39
<b>Figura N° 3.08.-</b> Obtención de los campos de datos en la Ficha Catastral en imagen binaria. ....	40
<b>Figura N° 3.09.-</b> Izq.: Acercamiento al ubigeo. Der.: Acercamiento a DPTO ....	41
<b>Figura N° 3.10.-</b> Uso de la herramienta "data cursor" para tomar dos esquinas. ....	41
<b>Figura N° 3.11.-</b> A la izquierda la imagen original, a la derecha la imagen recortada y limpia. ....	42
<b>Figura N° 3.12.-</b> A la izquierda la imagen original, a la derecha su matriz de características o vector patrón. ....	43
<b>Figura N° 3.13.-</b> Formas de escribir los caracteres de entrenamiento de la red. ....	44
<b>Figura N° 3.14.-</b> Primera hoja de los caracteres de entrenamiento de la red. ....	45
<b>Figura N° 3.15.-</b> Segunda hoja de los caracteres de entrenamiento de la red. ....	46
<b>Figura N° 3.16.-</b> Ejemplo de caracteres escritos reconocidos. ....	46

<b>Figura N° 3.17.-</b> Entrenamiento de la red con el “Neural Network Training Toolbox” del MATLAB.....	51
<b>Figura N° 3.18.-</b> Salida de carácter en formato “txt” leído por el bloc de notas del Sistema Operativo Windows.....	52
<b>Figura N° 3.19.-</b> Salida de texto en formato “txt” leído por el bloc de notas del Sistema Operativo Windows.....	52
<b>Figura N° 3.20.-</b> Letras en imágenes recortadas. ....	53
<b>Figura N° 3.21.-</b> Letras en MATLAB.....	54
<b>Figura N° 3.22.-</b> Letras en archivo de texto de formato “csv”.....	54
<b>Figura N° 3.23.-</b> Reconocimiento ICR exitoso de “SmartZone” de un campo de la ficha catastral. ....	57
<b>Figura N° 3.24.-</b> Salida de texto de los campos reconocido por “SmartZone”. ...	58
<b>Figura N° B.01.-</b> Ejemplo de “subplot” para mostrar una imagen y su vector ....	88
<b>Figura N° B.02.-</b> Se muestra una imagen usando la función “imshow”.....	89
<b>Figura N° B.03.-</b> Acceso al menú “help” del “Image Processing Toolbox” desde el menú “start” .....	90
<b>Figura N° B.04.-</b> Barra de herramientas del modo “debug” del MATLAB .....	93
<b>Figura N° C.01.-</b> Modelo de red neuronal artificial.....	104
<b>Figura N° C.02.-</b> Gráfica de Nivel de ruido vs Porcentaje de error de reconocimiento del ejemplo “prprob”.....	109
<b>Figura N° C.03.-</b> Nivel de ruido vs Porcentaje de error de reconocimiento. ....	110
<b>Figura N° E.01.-</b> Ficha individual escaneada. Cara frontal. ....	116
<b>Figura N° E.02.-</b> Ficha individual escaneada. Cara posterior. ....	117

## LISTA DE TABLAS

	Pág.
<b>Tabla N° 3.01.-</b> Vectores “targets” de cada patrón de carácter para el entrenamiento de la red.....	48
<b>Tabla N° 3.02.-</b> Datos de la ficha catastral en una hoja de Excel.....	54

## LISTA DE SIGLAS

- API** : Application Programming Interface.  
(Interfaz de Programación para Aplicaciones).
- ASCII** : American Standard Code for Information Interchange.  
(Código Estándar Americano para el Intercambio de Información).
- CAPTCHA** : Completely Automated Public Turing test to tell Computers and Humans Apart.  
(Prueba de "Turing" pública y automática para diferenciar máquinas y humanos).
- IA** : Inteligencia Artificial.
- ICR** : Intelligent Character Recognition.  
(Reconocimiento Inteligente de Caracteres).
- OCR** : Optical Character Recognition.  
(Reconocimiento Óptico de Caracteres).
- OMR** : Optical Mark Recognition.  
(Reconocimiento de Marcas Ópticas).
- RGB** : Red Green Blue.  
(Rojo Verde Azul).
- RNA** : Red Neuronal Artificial.  
(Red Neuronal).
- SDK** : Software Development Kit.  
(Equipo de Desarrollo de Software).
- SIG** : Sistema de Información Geográfica.
- SNCP** : Sistema Nacional Integrado de Información Catastral Predial.

## LISTA DE SÍMBOLOS

- $X_1, X_2, \dots, X_n$  : Variables asociadas a los pesos sinápticos.
- $W_{ij}$  : Pesos sinápticos.
- $f$  : Función de activación.

## LISTA DE PROGRAMAS

- SMARTZONE ICR/OCR, de la empresa ACCUSOFT PEGASUS.  
Procesa campos de registro en formularios tanto si ellos fueron digitados (OCR) o escritos a mano (ICR).
- A2iA FIELDREADER, de la empresa A2iA S.A.  
Captura datos escritos a mano e impresos en los documentos estructurados, como por ejemplo formularios.
- CARLOSICR, de Carlos De la cruz.  
Reconoce texto escrito en fichas catastrales y lo transfiere a un archivo de texto. Se ejecuta desde el MATLAB 2011.

## INTRODUCCIÓN

El trabajo de transferir texto de las fichas catastrales a un archivo de texto, es una tarea larga y monótona, por tanto, es necesario pensar en una forma de optimizar el tiempo, así que al revisar ejemplos ya creados de redes neuronales del MATLAB (un ejemplo de reconocimiento de caracteres puede ser apreciado en el Anexo C) se encontró que este programa tiene herramientas para crear aplicaciones para este tipo de tareas, capaces de reconocer caracteres a partir de imágenes de letras escritas a mano.

El Capítulo I abarca el panorama general de la tecnología de reconocimiento de texto y ubica el contexto del tema dentro de los sistemas OCR e ICR.

EL Capítulo II constituye todo el fundamento teórico de las redes neuronales.

El Capítulo III contiene su aplicación al catastro urbano, la metodología utilizada y el desarrollo del programa en MATLAB utilizando redes neuronales.

El Capítulo IV comprende las conclusiones y recomendaciones finales, producto del desarrollo teórico y algorítmico del tema abordado.

# CAPÍTULO I.- GENERALIDADES

## 1.1. INTRODUCCIÓN

Los sistemas que, a partir de un texto escrito o impreso en papel o similar, crean un fichero de texto en un soporte de almacenamiento informático, se denominan sistemas OCR. Un sistema OCR cuenta con la característica de poder "aprender", mediante una red neuronal o correlación no lineal, patrones de caracteres binarios (letras y números en blanco y negro del alfabeto) que representen las posibles variaciones de tamaño y forma de los diferentes caracteres impresos que pueden aparecer en los documentos, los que se pueden "reconocer y convertir" de texto escrito en papel a texto almacenado en un fichero ASCII.

Avanzados OCR, los constituyen los ICR (Intelligent Character Recognition) y los OMR (Optical Mark Recognition). Los ICR reconocen diferentes formas y estilos de caracteres escritos a mano para ser aprendidos por una computadora, y los OMR (Optical Mark Recognition) reconocen marcas impresas.

En 1993 se inventó el procesamiento de formularios automático [20], en el cual generalmente se combina la tecnología de redes neuronales con los OCR, para formar ICR que actualicen automáticamente su base de datos para aprender reconocer nuevas formas y estilos de caracteres escritos. Para el presente trabajo no se va usar OMR porque no se va a reconocer ninguna marca impresa como aspa, "check", guión y barra inclinada.

Algunos de los programas ICR existentes en el mercado internacional son SmartZone, A2iA FieldReader. El programa SmartZone usa OCR cuando los caracteres fueron digitados y cuando fueron escritos a mano usa ICR

con redes neuronales. Este programa se usará para comparar los resultados con el programa desarrollado en MATLAB en el presente trabajo.

## **1.2. PLANTEAMIENTO DEL PROBLEMA**

El problema fundamental es transferir la información escrita en las fichas catastrales a una base de datos de archivo de texto.

## **1.3. OBJETIVO DEL ESTUDIO**

- Reducir el tiempo de procesamiento de transferencia de datos de fichas catastrales a formatos digitales de texto.
- Utilizar redes neuronales artificiales para reconocer caracteres escritos a mano sin desarrollar la parte matemática de las redes, porque estas se crean con funciones del programa "framework" del MATLAB llamado "Neural Network Toolbox".

## **1.4. ETAPAS DEL SISTEMA ICR DESARROLLADO**

Las etapas del programa "CARLOSICR" desarrollado en MATLAB se muestran en un diagrama de flujo. El programa se ha subdividido en dos versiones independientes: ICR1 e ICR2, como se muestra en el anexo A.

El diagrama de flujo del algoritmo del programa "ICR1" desarrollado en MATLAB en su versión para reconocer letras y números individuales se muestra en la figura N° 1.01.

El diagrama de flujo del algoritmo del programa "ICR2" desarrollado en MATLAB en su versión para reconocer palabras y texto se muestra en la figura N° 1.02.

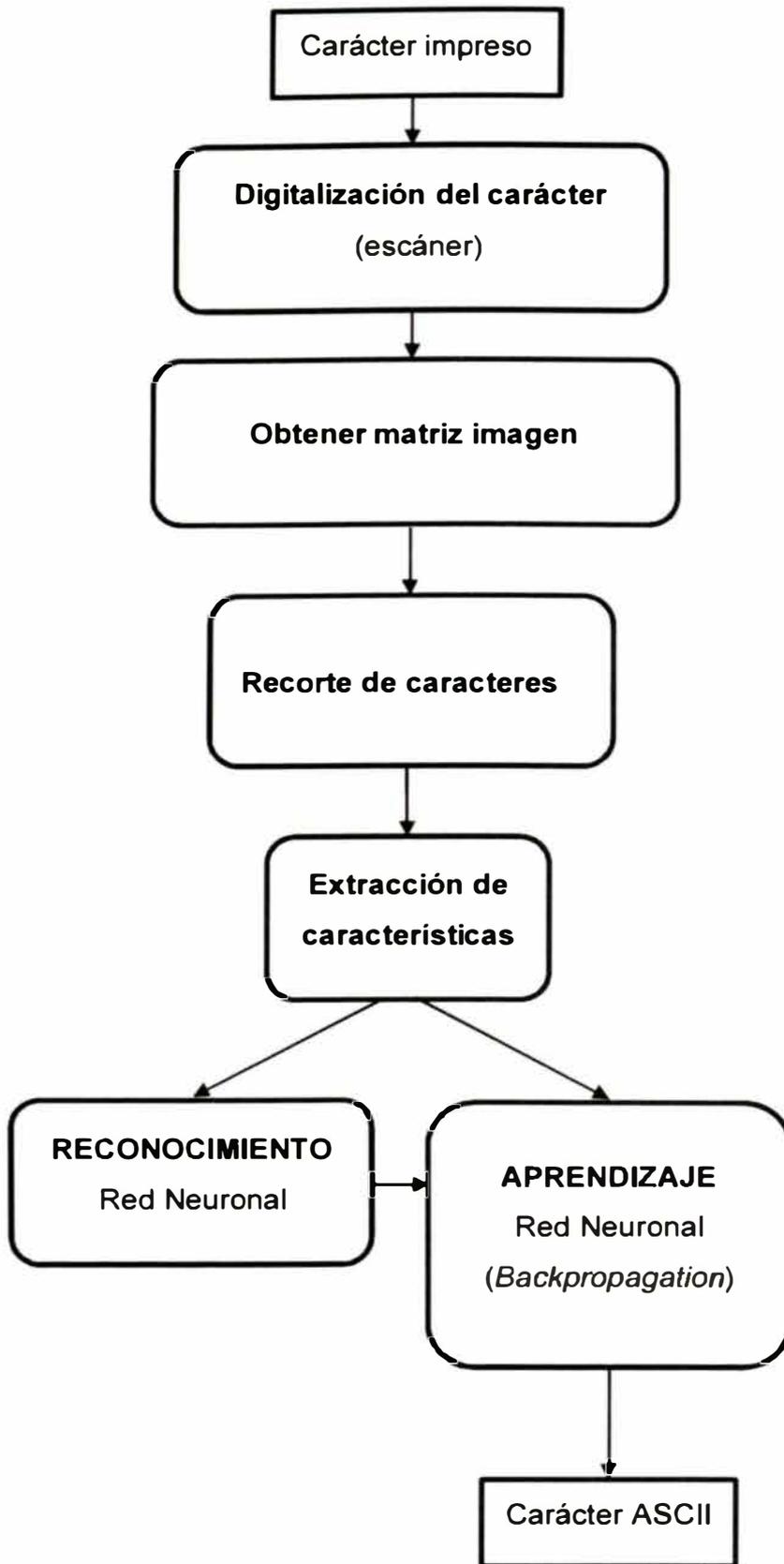


Figura N° 1.01.- Diagrama de flujo del reconocimiento de caracteres.

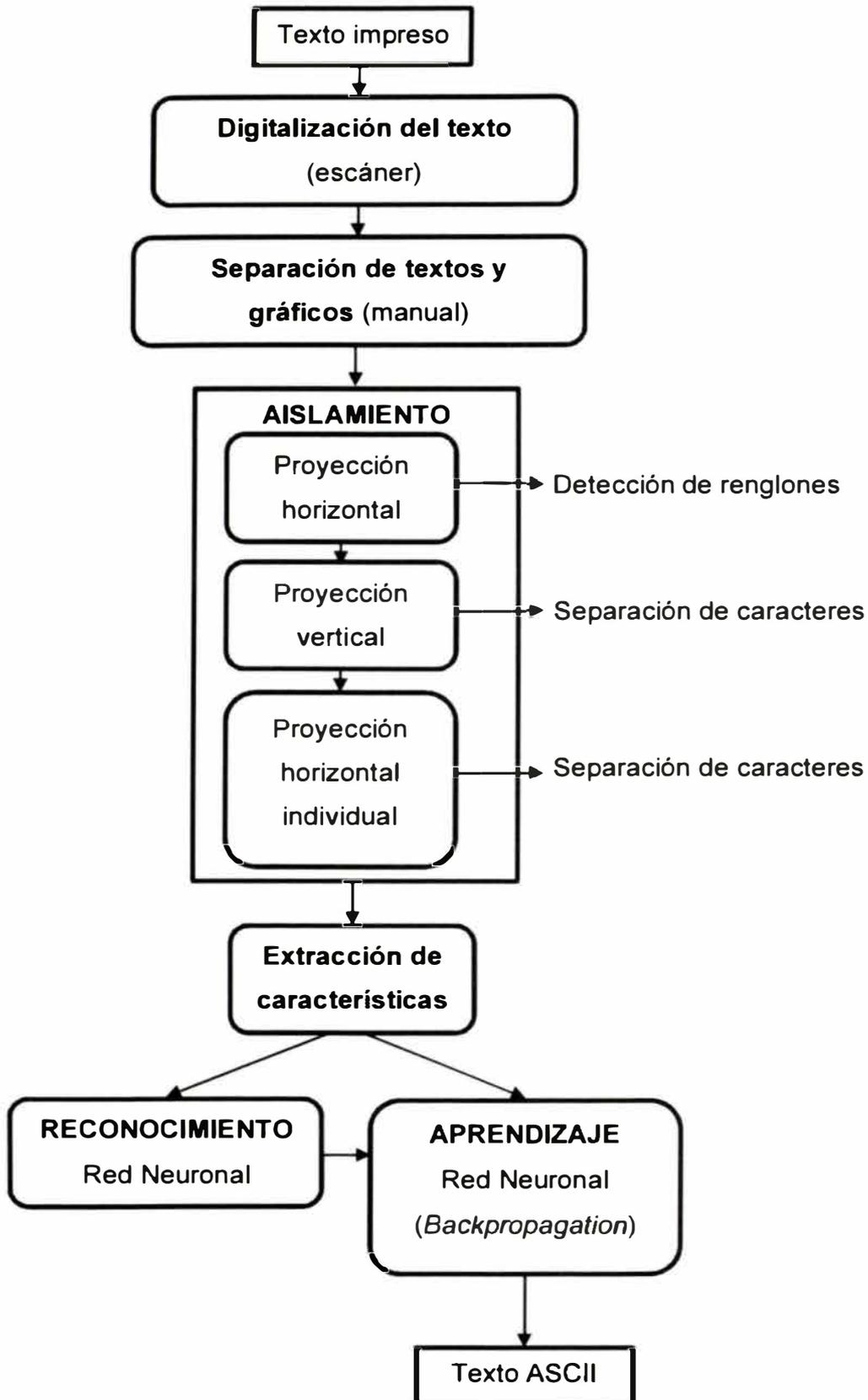


Figura N° 1.02.- Diagrama de flujo del reconocimiento de texto.

## 1.5. METODOLOGÍA DE TRABAJO

Se realiza un escaneo de dos (02) fichas catastrales de registro representativas, de la cual se elige la más nítida. En la ficha, la parte textual de los registros será reconocida como texto en imagen por la herramienta de procesamiento de imágenes del MATLAB y por medio de un algoritmo de redes neuronales se le asocia su respectiva letra o número del alfabeto castellano; luego otro algoritmo lo escribe como un carácter ASCII en una salida de archivo de texto simple y finalmente para ordenar el texto en filas y columnas se transforma a un formato de archivo delimitado por comas (extensión csv) que es leído por una hoja de cálculo de Excel.

# CAPÍTULO II.- MARCO TEÓRICO

## 2.1. INTELIGENCIA ARTIFICIAL Y REDES NEURONALES

La Inteligencia Artificial es una rama de las ciencias de la computación que estudia los fundamentos teóricos y prácticos del diseño de sistemas de computación, que exhiben características inteligentes del ser humano como resolución de problemas, comprensión de lenguajes, aprendizaje y razonamiento lógico.

Otra definición más completa es el ámbito de estudio que se enfoca en la explicación y emulación de la conducta inteligente en función de procesos computacionales basados en la experiencia y el conocimiento continuo del ambiente. Según Nebendah (1988) y Delgado (1998). [4] [16]

## 2.2. ÁREAS DE ESTUDIO EN LA INTELIGENCIA ARTIFICIAL

Las principales son: Teoría de Juegos, Visión Artificial, Robótica, Comprensión del Lenguaje, Traducción Automática, Programación Automática, Sistemas Expertos, Redes Neuronales y Algoritmos Genéticos.

Las Redes Neuronales son de dos tipos: artificiales y biológicas.

Las **Redes Neuronales Biológicas** se componen de neuronas biológicas reales que están conectadas o relacionadas funcionalmente en un sistema nervioso. Tiene aplicaciones en el ámbito de la neurociencia.

Las **Redes Neuronales Artificiales** se componen de neuronas artificiales que están interconectadas y procesan información en forma paralela y en serie para formar una red neuronal artificial. Las RNA están en la base de diversas aplicaciones de la IA, por eso se encuentran en la

categoría de técnicas básicas dentro la IA. Sin embargo, el usuario final no necesita conocerla sino los profesionales que se dedican a su aplicación y generación de aplicaciones comerciales. Para el desarrollo del presente trabajo, se ha creado RNA empleando el programa de redes neuronales del MATLAB llamado "Neural Network Toolbox".

### 2.3. REDES NEURONALES ARTIFICIALES

Las Redes Neuronales Artificiales son sistemas de computación que permiten la resolución de problemas que no pueden ser descritos fácilmente mediante un proceso de algoritmo tradicional, como por ejemplo, el reconocimiento de formas. Con las redes se expresa la solución de un problema, no como una secuencia de pasos, sino como la evolución de un sistema inspirado en el funcionamiento del cerebro y dotado de cierta "inteligencia". Tal sistema no es sino la combinación de una gran cantidad de elementos simples de procesos (nodos o neuronas) interconectados que, operando de forma masivamente paralela, consiguen resolver el problema.

Una definición es:

*"Una red neuronal artificial es un procesador distribuido en paralelo de forma masiva que tiene una tendencia natural para almacenar conocimiento de forma experimental y lo hace disponible para su uso." [7]*

Las redes neuronales artificiales son conocidas también como modelos conexionistas. Las conexiones sirven para transmitir las salidas de unos nodos a las entradas de otros. El funcionamiento de un nodo es similar al de las neuronas biológicas presentes en el cerebro. Suele aceptarse que la información memorizada en el cerebro está relacionada con los valores sinápticos de las conexiones entre las neuronas. De igual forma, se dice que las redes neuronales tienen la capacidad de "aprender" mediante el ajuste de las conexiones entre nodos. Estas conexiones tienen un valor numérico asociado denominado peso, que puede ser positivo (conexiones de excitación) o negativo (conexiones de inhibición).

### 2.3.1. Usos

Según el nivel de dificultad del uso, van desde lo más simple a lo complejo:

El uso más simple, es para ajuste simple de datos a una función de ajuste.

El uso medio, en reconocimiento de letras y números usando OCR. Tareas de clasificación de patrones.

El uso más difícil, en reconocimiento de rostros, ojos, huellas digitales en los sistemas de seguridad de personal en una empresa. También en reconocimiento de texto tipo CAPTCHA.

Otros usos son en:

- Transporte y comunicaciones: En optimización de rutas y optimización de recursos.
- Electricidad: Predicción de consumo de energía eléctrica.
- Estructuras: Reconocimiento de patrones de fallas.

### 2.3.2. Estructura de una Red Neuronal Artificial

Existen diferentes tipos de estructura de redes neuronales, según el tipo de problema a solucionar. Para comprender la estructura de una neurona artificial, antes se describirá brevemente, la estructura de una **neurona biológica**.

La célula nerviosa es la *neurona*, y es la unidad funcional del sistema nervioso. La estructura de una neurona biológica se muestra en la figura N° 2.01.

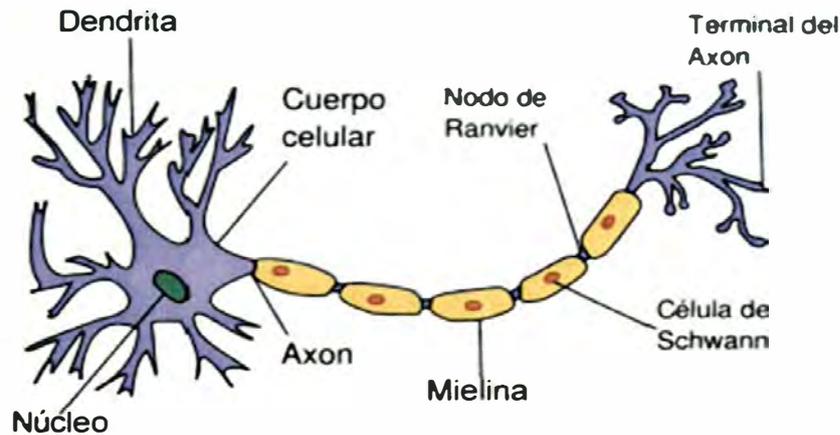


Figura N° 2.01.- Estructura de una neurona biológica. [2]

Las dendritas y el cuerpo celular o soma reciben la sinapsis de otras neuronas, del medio interno o del medio externo y las llevan al axón. El segmento inicial del axón integra las señales nerviosas que reciben las dendritas y soma, luego el axón “decide” las señales o potenciales de acción que ha de propagar hacia sus terminales. Los terminales de los axones hacen contacto con otras neuronas en uniones llamadas sinapsis o unión de neuronas. Entonces, cuando las señales llegan encuentran la unión y se transmiten a otra neurona, transfiriéndoles la información que ella portaba.

Las dendritas y soma son la zona receptora.

El axón es la zona conductora o propagadora. La conducción es interna.

El terminal del axón es la zona transmisora. La transmisión es de una neurona a otra neurona. [2]

Las **redes neuronales artificiales** están formadas por un conjunto de unidades de procesamiento interconectadas llamadas *neuronas*, entonces, la unidad fundamental para la RNA es una *neurona artificial*, la cual se muestra en la figura N° 2.02.

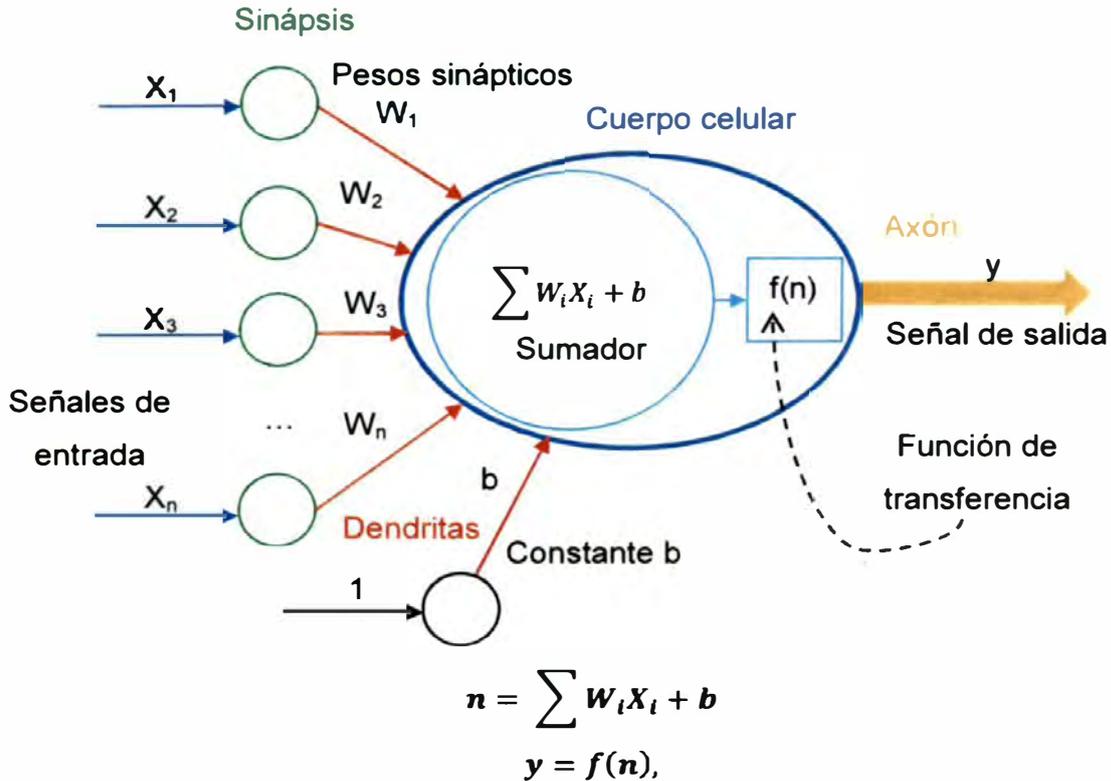
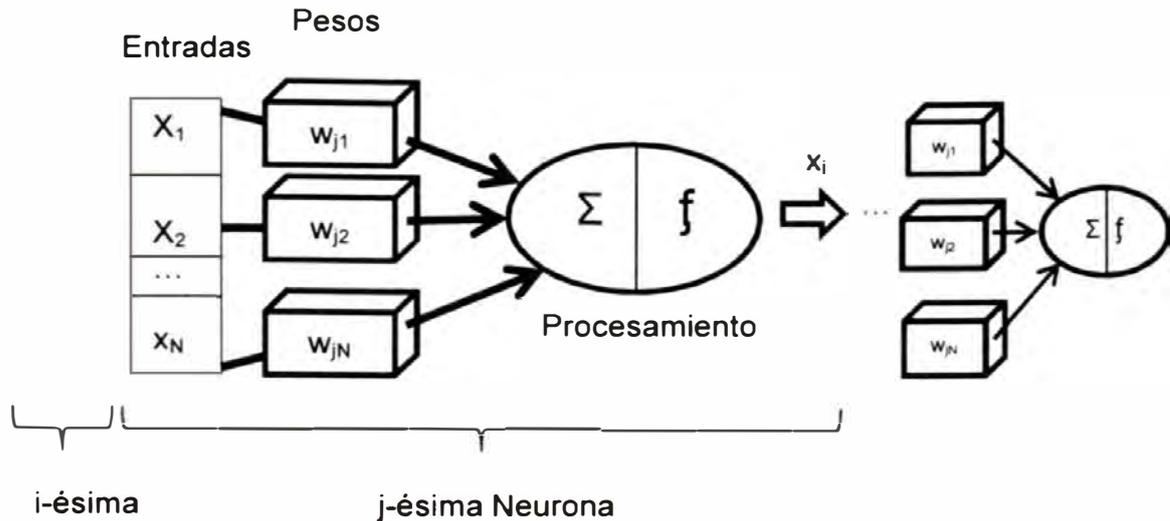


Figura N° 2.02.- Estructura de una neurona artificial.

En la neurona artificial los nodos (dendritas) reciben las entradas  $X_1, X_2, \dots, X_n$  de otros nodos por medio de la conexión entre nodos (sinapsis). Ahora para identificar la intensidad de las conexiones entre nodos se le asocia un valor numérico denominado peso, que puede ser positivo (conexiones de excitación) o negativo (conexiones de inhibición). Suele aceptarse que la información memorizada en el cerebro está relacionada con los valores sinápticos de las conexiones entre las neuronas. De igual forma, se dice que las RNA tienen la capacidad de “aprender” mediante el ajuste de las conexiones entre nodos. Estas conexiones tienen un valor numérico asociado denominado peso. Entonces con el ajuste de los pesos, mediante un entrenamiento de la neurona con datos de entrada y salida deseada, se mejora la capacidad de aprender de la neurona artificial.

El modelo de neurona artificial más conocido es de McCulloch y Pitts, por ser el primer modelo inicial propuesto sobre neuronas artificiales, el cual se muestra en la figura N° 2.03. [5]



**Figura N° 2.03.-** Estructura de una Red Neuronal Artificial.

Cada neurona recibe como entrada un conjunto de señales discretas o continuas, representadas por las variables  $x_1, x_2, \dots, x_n$ , las cuales están asociadas a los pesos sinápticos, que son representados por las variables  $w_{ij}$ , y son ellos en los que se guarda la mayor parte del conocimiento que la red neuronal tiene sobre la tarea en cuestión, además determinan el nivel de influencia de la neurona  $j$  para la neurona  $i$ .

Existen dos etapas de procesamiento para cada neurona: "suma y activación".

- En la primera etapa, las señales de entrada  $x_j$  y los pesos  $w_{ij}$  son combinados mediante la siguiente sumatoria.

$$y_i = \sum_{j=1}^N w_{ij}x_j$$

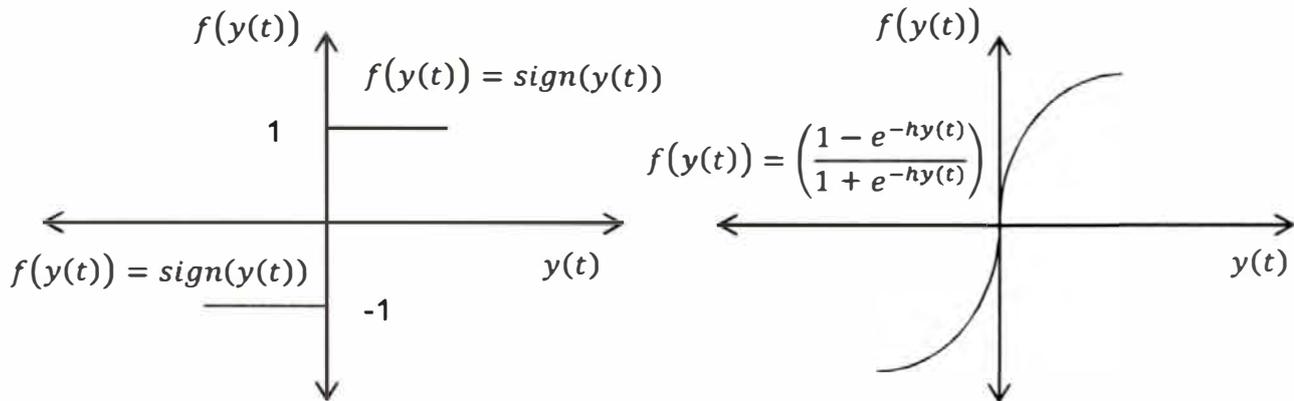
Donde  $y_i$  representa el estado interno de la *i-ésima* neurona.

- En la segunda etapa, la salida de la neurona es generada a través de la aplicación de una función de activación.

$$x_i = f(y_i)$$

Donde la salida de la neurona es representada por  $x_i$  y  $f$  corresponde a la función de activación aplicada al estado interno de la neurona, que tiene

como objetivo limitar el nivel de activación entre  $[-1, 1]$  ó  $[0, 1]$  en el caso de  $x_i$  sea un valor continuo y si  $x_i$  es discreto entonces el nivel de activación puede ser:  $\{-1, 1\}$  ó  $\{0, 1\}$ . Ver figura N° 2.04.



**Figura N° 2.04.-** Funciones de Activación: Función de grado y tangente hiperbólica (De izquierda a derecha).

Existen varios tipos de funciones de activación, la figura N° 2.04 muestra dos de las funciones de activación más usadas: La función de grado y la tangente hiperbólica. Como se vio en la figura N° 2.03, la salida de una neurona puede ser la entrada de otra. Generalmente, una red neuronal se forma por muchas neuronas de algunos acoplados. [5]

### 2.3.3. Tipos de Redes Neuronales

Existen muchos tipos de redes neuronales, porque cada uno resuelve un problema diferente de entrada y salida de datos. (Para una descripción de cada tipo, ver [6]).

Según su conexión las redes se clasifican en “feedforward” y “feedback”. Las redes “feedforward” (conexiones hacia adelante) más importantes son:

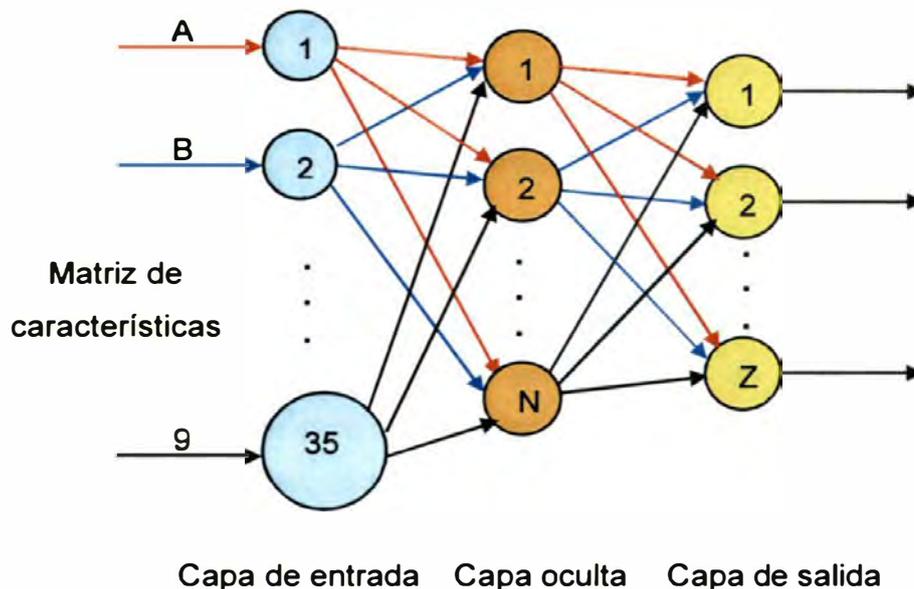
- Perceptrón.
- Perceptrón multicapa.
- Adaline.

- Backpropagation.
- Mapas autoorganizados.
- Radiales.

Otros tipos de redes son:

- Aprendizaje asociado.
- Redes competitivas
- Redes recurrentes

Un modelo utilizado en una gran variedad de aplicaciones, entre las que se incluyen las de reconocimiento de formas, es la red multicapa con conexiones unidireccionales hacia delante (feedforward). En la figura N° 2.05 se muestra la red de este tipo utilizada en el sistema OCR para el reconocimiento de caracteres, donde cada círculo representa una neurona o nodo y apiladas en columna forman una capa.



**Figura N° 2.05.- Estructura de la red neuronal utilizada.**

El tipo de red que se utilizará para darle solución al problema del reconocimiento óptico de caracteres es una red neuronal tipo multicapa denominada *backpropagation* (con retroalimentación). [17]

### **2.3.4. Algoritmo de aprendizaje**

Los algoritmos en Aprendizaje Automático pueden ser clasificados en dos categorías: supervisados y no supervisados. Se usa el algoritmo en aprendizaje supervisado para construir "modelos" que generalmente predicen ciertos valores esperados. Para ello, los algoritmos supervisados requieren que se especifiquen los valores de entrada/salida (input/output o input/target) con un conjunto en pares de vectores con entradas reales de la forma, conocido como conjunto de entrenamiento o conjunto de ejemplos. Para nuestro caso se usa un algoritmo supervisado y el "input" son vectores binarios de 35 bits que representan números y patrones de caracteres del alfabeto y el "target" son vectores unitarios que tienen la unidad (bit) en el número de elemento igual al número de orden de la letra en el alfabeto. (Ver "2.4. Redes Neuronales en MATLAB" de este capítulo).

Los algoritmos de aprendizaje generalmente calculan los parámetros de una función (pesos) que permiten aproximar los valores de salida en el conjunto de entrenamiento.

### **2.3.5. Algoritmo de retropropagación (backpropagation)**

La propagación hacia atrás de errores o retropropagación es un algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales. El algoritmo consiste en minimizar un error (comúnmente cuadrático) por medio de una gradiente descendiente, por lo que la parte esencial del algoritmo es cálculo de las derivadas parciales de dicho error con respecto a los parámetros de la red neuronal.

Los algoritmos supervisados necesitan que se especifiquen valores de entrada y salida deseada u objetivo, para determinar la calidad de la aproximación del modelo obtenido por el algoritmo.

## **2.4. REDES NEURONALES EN MATLAB**

En internet existe gran número de "API" y "frameworks" que implementan las RNA. Uno de estos "frameworks" es la herramienta de redes

neuronales del MATLAB (Neural Network Toolbox), que ofrece implementaciones genéricas de funciones que crean redes neuronales, así como, de implementaciones específicas que generan redes tales como perceptrón, backpropagation, radiales, redes recurrentes y mapas autoorganizados. A continuación se lista las funciones más conocidas según las redes que producen:

1. Perceptrón: perceptron (perceptron).
2. Backpropagation: feedforwardnet (feedforward neural network), fitnet (function fitting neural network), patternnet (pattern recognition network).  
En versiones anteriores (como MATLAB 2009) se usaba la función newff (crea una red "feedforward backpropagation"), sin embargo a partir de MATLAB 2010b ya está en desuso y en su lugar se usa la función feedforwardnet().
3. Radiales: newrb (design radial basis network).
4. Redes recurrentes: layrecnet (layer recurrent neural network).
5. Mapas autoorganizados: selforgmap (self-organizing map).

Para ver una lista completa de las funciones ver "new networks functions" en el buscador del menú ayuda de MATLAB. [13]

La forma más fácil de crear una red neuronal en MATLAB es usar una de las funciones de creación de redes neuronales de la barra de herramientas de redes neuronales (Neural Network Toolbox), como la función *feedforwardnet()* que crea redes "feedforward" a usar para el presente caso.

#### **2.4.1. Redes "Feedforward"**

Las redes "feedforward" tienen conexiones unidireccionales de la capa de entrada a las capas de salida. Son usadas comúnmente para predicción, reconocimiento de patrones y ajuste de funciones no lineales. Soportan redes "backpropagation" ("feedforward backpropagation"), perceptrón y "cascade-forward backpropagation". Las redes "feedforward" consisten de una serie de capas. La primera capa tiene una

conexión desde la entrada de la red. Cada capa subsecuente tiene conexión desde la anterior capa. La capa final produce la salida de la red.

La función `feedforwardnet()` toma como entrada un vector de  $1 \times N$  con  $N$  capas o neuronas ocultas y entrena las neuronas con una función "backpropagation" y devuelve una red neuronal "feedforward" con  $N+1$  capas.

En general pueden ser usadas con cualquier clase de entrada para mapear salidas. Una red "feedforward" con una capa oculta y suficientes neuronas en las capas ocultas, pueden ajustar cualquier clase de problema de mapeo entrada-salida finita.

Versiones especializadas de la red "feedforward" incluyen redes de ajuste (`fitnet`) y reconocimiento de patrones (`patternnet`). Una variación en la red "feedforward" es la cascada hacia adelante de la red (`cascaforwardnet`) la cual tiene conexiones adicionales desde la entrada a cada capa y desde cada capa a todas las capas siguientes. [13]

#### 2.4.2. Funciones de entrenamiento

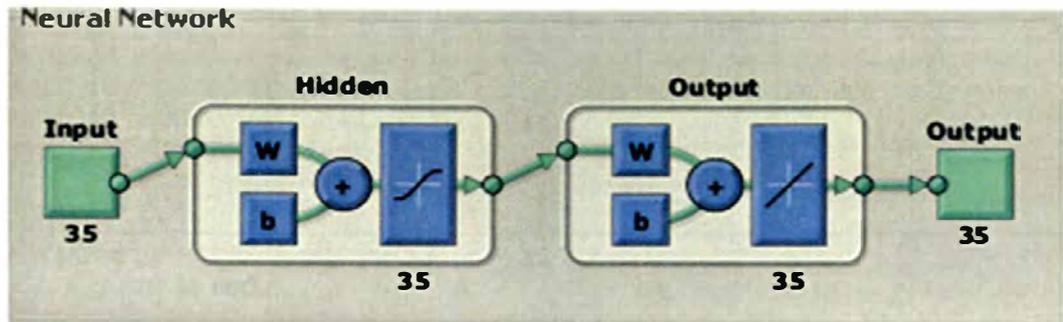
La función de entrenamiento más rápida es generalmente "`trainlm`", y es la función de entrenamiento por defecto para la función "feedforwardnet". El método "quasi-Newton" con función de entrenamiento "`trainbfg`" también es rápido. Sin embargo ambas funciones tienden a ser menos eficientes en redes largas (con miles de pesos), ya que requieren más memoria y más tiempo de cálculo. También "`trainlm`" tiene mejor rendimiento en problemas de ajuste de funciones (regresión no lineal) más que en problemas de reconocimiento de patrones. Todas las funciones de entrenamiento usan algoritmos de retropropagación (`backpropagation`).

Cuando hay entrenamiento de redes largas, y redes de reconocimiento de patrones, las funciones de entrenamiento "`trainscg`" y "`trainrp`" son buenas opciones. Sus requerimientos de memoria son relativamente

pequeños y sin embargo son muchos más rápidos que los algoritmos estándares de gradiente descendente. [14]

### 2.4.3. Diseño de la Red Neuronal

El modelo de red neuronal artificial usado se muestra en la figura N° 2.06.



$$output = f((input).w + b)$$

**Figura N° 2.06.-** Modelo de red neuronal para el reconocimiento de caracteres. Fuente: MATLAB

El flujo de trabajo para el proceso general de diseño de la red neuronal tiene siete pasos principales:

1. Colectar los datos.
2. Crear la red.
3. Configurar la red.
4. Inicializar los pesos (weights) y las constantes (biases).
5. Entrenar la red.
6. Validar la red (análisis post-entrenamiento).
7. Usar la red.

Ejemplo: Escribir en un nuevo archivo script (extensión .m) de MATLAB el siguiente código enmarcado y resaltado con un fondo de color:

1. Colectar los datos (con ruido).

En esta parte los datos se crean con ruido, para que el entrenamiento sea con ruido y la red reconozca mejor los patrones.

```
% Se crean los datos sin ruido.
[alphabet,targets] = prprob;
% Se crean los datos con ruido.
setdemorandstream(pi);
numNoise = 60; % 60 veces, 60 pasadas.
alphabet2 =
min(max(repmat(alphabet,1,numNoise)+randn(35,35*numNoise)*0.
2,0),1); %0.2 es el máximo nivel de ruido (noiseLevel) para
error cero.
targets2 = repmat(targets,1,numNoise);
```

## 2. Crear la red.

```
% Para reconocer este problema usaremos una configuración de
red neuronal feedforward para reconocimiento de patrones
con 35 neuronas ocultas.
net = feedforwardnet(35, 'trainscg'); % Crea la red neuronal
con 35 neuronas ocultas y función de entrenamiento
'trainscg'.
```

## 3. Configurar la red.

```
net = configure(net,alphabet2,targets2); % Obtiene los
parámetros de configuración.
net.trainParam.epochs = 1000; % Periodos.
net.trainParam.goal = 0; % Error de iteración.
```

## 4. Inicializar los pesos

```
net = init(net); % Inicia la configuración.
net.iw{1,1} % configure los pesos.
net.b{1} % configura las constantes (biases).
```

## 5. Entrenar la red.

```
net = train(net,alphabet2,targets2); % Entrena la red con
ruido.
```

## 6. Validar la red.

```
y = net(alphabet); % obtiene la salida de la red para la
entrada "alphabet".
errores = gsubtract(targets,y); % Resta "target" menos "y".
performance = perform(net,y,targets) % Obtiene un error
global.
```

## 7. Usar la red.

```
yd = net(letraD); % Se ingresa un vector patrón D.
```

A continuación se explica cada paso:

Primero se colectan los datos ("alphabet" y "targets") para el entrenamiento de la red neuronal usando una función prprob(), la cual

está creada en un “script” llamado `prprob.m`, incluido en los archivos del MATLAB (Ver el anexo C).

La llamada de la función es:

```
[alphabet,targets] = prprob;
```

La función `prprob()` tiene dos salidas:

La primera salida es “alphabet”, que está formado por vectores columnas de patrones de caracteres como los siguientes:

El vector patrón de A es [0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1] de tamaño 35x1.

El vector patrón de B es [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0] de tamaño 35x1.

El vector patrón de C es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0] de tamaño 35x1.

El vector patrón de D es [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0] de tamaño 35x1.

...

El vector patrón de Z es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 0] de tamaño 35x1.

El vector patrón de 9 es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0] de tamaño 35x1.

Los vectores patrones cuando se plotean con la función `plotchar()` del MATLAB muestran las letras y números a reconocer. (Ver figura N° 3.12).

La segunda salida es “targets”, que es una matriz identidad de 26x26 formada por vectores columna unitarios de 26x1, y cada vector se asocia a cada letra de la siguiente manera:

El vector patrón A su vector columna “target” es [1, 0, ..., 0] de tamaño 26x1.

El vector patrón B su vector columna "target" es [0, 1, ..., 0] de tamaño 26x1.

...

El vector patrón Z su vector columna "target" es [0, 0, ..., 1] de tamaño 26x1.

Los vectores unitarios "target" tienen la unidad en el número de elemento igual al número de orden de la letra en el alfabeto. La "A" está en la primera posición del alfabeto, le corresponde la unidad en la primera posición, la "B" le corresponde la unidad en la segunda posición, la "C" la unidad en la tercera posición, ..., el "Y" la unidad en la vigésima quinta posición y el "Z" la unidad en la vigésima sexta posición.

Segundo, tercero y cuarto se crea la red neuronal con la función `feedforwardnet()` con una configuración de 35 neuronas ocultas, función de entrenamiento 'trainscg', pesos y constante de 1 y se guarda la red en la variable "net" como un objeto. Son 35 para que sea una neurona por cada letra y número. El código se muestra a continuación:

```
net = feedforwardnet(35, 'trainscg');
```

Quinto la función `train()` entrena la red con ruido y guarda el resultado en la variable "net". Esta función está presente a partir de MATLAB 2011, no corre en MATLAB 2009. El código se muestra a continuación:

```
net = train(net, alphabet2, targets2);
```

En la figura 3.17 se muestra la interfaz del entrenamiento con la función `train()`.

Sexto en la validación se calcula un error del reconocimiento.

Séptimo se usa la red con el mejor entrenamiento y menor error.

# CAPÍTULO III.- APLICACIÓN DE REDES NEURONALES A CATASTRO

## 3.1. ANTECEDENTES

Luego de realizar el Catastro Urbano del Centro Poblado Santa Cruz había 300 registros que procesar a una hoja de cálculo, por lo que se vio necesario optimizar el tiempo empleado en registrar cada predio usando un BlackBerry para registrar y procesar al mismo tiempo estando en el trabajo de llenado de fichas, pero el único inconveniente fue que no se iba a disponer de un registro de fichas impresas con los datos de los propietarios de cada predio y con sus firmas a puño y letra. Así que se optó por usar otro método que a partir de las fichas impresas y llenadas a mano, se procesara más rápido la información a una hoja de cálculo digital. Entonces al tener conocimiento que las redes neuronales son una técnica que puede emplearse en el reconocimiento de caracteres de texto impreso o escrito a mano en imágenes escaneadas, se eligió ésta alternativa como solución al problema fundamental. También porque se dispone de una herramienta de redes neuronales en MATLAB, que ya está implementada e incluso presenta un ejemplo desarrollado de reconocimiento de caracteres (llamado "character recognition" en MATLAB). Así mismo por tener apoyo instructivo del asesor; solo había que hacer código en MATLAB, buscar información para aplicar y adaptarlo al reconocimiento de caracteres en campos de formularios. Sin embargo, hubo una dificultad, que fue aprender a procesar imágenes usando un programa llamado "Image Processing Toolbox" en MATLAB [1] [3] [12], lo cual tomó regular tiempo, y fue estudiado con el fin de procesar los escaneados de imagen a matriz de características, siendo este último la entrada de la red neuronal artificial. El programa de procesamiento de imágenes consta básicamente de funciones con una aplicación

específica, como por ejemplo pasar de imagen a matriz, recortar, eliminar pixeles. Este procesamiento de imágenes se explica en el punto 3.3 “desarrollo del programa en MATLAB” y también se adjunta una referencia en el Anexo B.

Luego de procesar la información, la base de datos de la hoja de cálculo se transfiere a la base de datos de un Sistema de Información Geográfica (SIG), para generar el catastro vinculado con la cartografía, como establecen las normas del Sistema Nacional Integrado de Información Catastral Predial (SNCP). Cabe mencionar que el catastro urbano fue realizado en el “EXPEDIENTE TÉCNICO DEL CATASTRO URBANO DEL CENTRO POBLADO “SANTA CRUZ” VEGUETA – HUAURA - LIMA CON APLICACIONES GIS”.

### **3.2. DESCRIPCIÓN DEL ALGORITMO**

El algoritmo se explica con pseudocódigo:

- I. Leer la imagen.
- II. Segmentación. Es el recorte excedente de área sobrante que rodea al carácter. Comprende los siguientes pasos:
  - II.1 Recortar la imagen.
  - II.2 Separar cada línea.
  - II.3 El texto como párrafo es separado en líneas.
  - II.4 Separar en palabras.
  - II.5 Las palabras son separadas por partes.
  - II.6 Separar en letras.
  - II.7 Cada palabra es separada en caracteres.
- III. Creación de la red neuronal.
- IV. Entrenar la red neuronal.
- V. Simulación de la red neuronal.
- VI. Convertir a texto.
- VII. Interpretación de los resultados.

### 3.3. DESARROLLO DEL PROGRAMA EN MATLAB

Debido al poder de la herramienta del MATLAB 2011b y la existencia de unas funciones (en la herramienta de redes neuronales y de procesamiento de imágenes) que facilitan el trabajo con las redes neuronales y el procesamiento de imágenes, se ha considerado implementar el reconocimiento de caracteres con las mismas. Para ello, se revisa primero el uso de las funciones en los anexos B, C y D. Luego ver el código completo que se encuentra en el anexo A. Finalmente revisar el desarrollo del programa.

A continuación se dan instrucciones del desarrollo del programa en MATLAB, elaborado por fuente propia:

#### 1. Configuración inicial.

##### 1.1. Configuración de la memoria máxima a usar por el MATLAB.

Para soportar el entrenamiento de 35 neuronas en MATLAB se utiliza como mínimo 200 MB de memoria RAM para objetos JAVA (valor obtenido experimentando con diversos posibles valores).

La configuración requerida se encuentra en la siguiente ruta, estando en MATLAB:

Menu File > Preferences > General > Java Head Memory > Configurar 200 MB como mínimo.

En el presente caso se fijó en 196 MB. Ver figura N° 3.01.

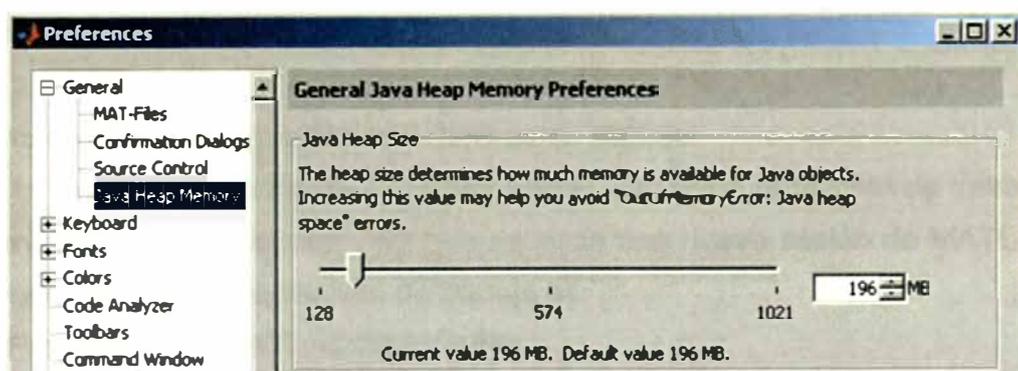


Figura N° 3.01.- Configuración de la memoria RAM máxima a usar por el MATLAB.

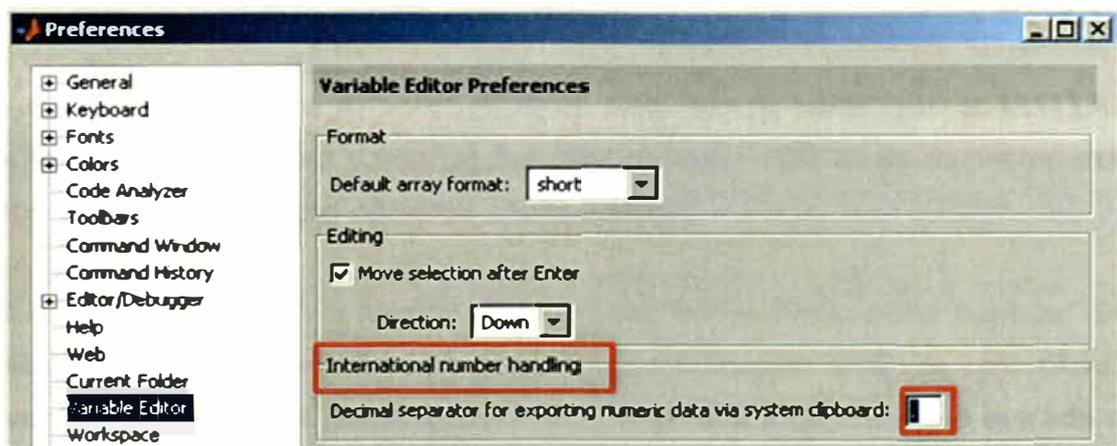
Si se configura menor memoria máxima, se mostraría un mensaje de error como el siguiente “OutOfMemoryError: Java head space” al inicio de la ventana de comandos del MATLAB.

## 1.2. Configurar el punto como separador decimal.

La configuración se encuentra en la siguiente ruta:

Menu File > Preferences > Variable Editor > International number handling > En “decimal separator for exporting...” escribir un punto (.).

En la figura N° 3.02, se muestra la configuración.



**Figura N° 3.02.-** Cambiar de coma a punto el separador decimal en MATLAB.

Finalmente para que los cambios tengan efecto en MATLAB presionar el botón ok.

## 2. Definir la carpeta de trabajo.

La carpeta de trabajo ubica los archivos de trabajo.

En la barra de herramientas “Current Folder” se define la carpeta de trabajo. También se la define cada vez que se inicia una nueva sesión de MATLAB, ya que por defecto la carpeta de trabajo es:

C:\Program Files\MATLAB\R2011b\bin.

En la figura N° 3.03 se muestra un ejemplo de la carpeta de trabajo definida.



**Figura N° 3.03.-** Ubicación de la carpeta de trabajo.

Esto es importante, puesto que si tienen abiertos dos scripts con el mismo nombre, esta carpeta será la única que los diferenciará para ejecutar el correcto script.

3. **Crear un nuevo archivo script.** Los “scripts” tienen extensión “.m” en MATLAB. Los pasos para crear un “script” son:

Menu File > New > Script.

También se crea al presionar “ctrl + N”

Asignarle un nombre al script (ICR en este caso). Notar que el MATLAB diferencia mayúsculas y minúsculas. Por ejemplo, ICR no es lo mismo que icr. También no acepta números como prefijos.

4. **Leer la imagen,** con la función `imread()` del “Image Processing Toolbox” del MATLAB. La función `imread()` convierte una imagen RGB de rojo (Red), verde (Green) y azul (Blue) a una matriz indexada y recibe como entrada el nombre y extensión de un archivo de imagen. También su ruta local en el disco duro si no se ha configurado la carpeta de trabajo.

Los formatos de entrada de la imagen pueden estar en extensión JPG, PNG, TIF y más; lo importante es que sean imágenes de colores RGB. Ver figura N° 3.04.

No usar el formato TIFF de “Microsoft Paint” porque la función `imread()` no lo soporta.



**Figura N° 3.04.-** Letra D escrita a mano escaneada y guardada como D.jpg

Por ejemplo para leer una letra D escaneada llamada D.jpg se escribe así:

```
imread('D.jpg');
```

y el resultado se guarda en la variable imageRGB. Así:

```
imageRGB = imread('D.jpg');
```

Si no se configuró la capeta de trabajo, la dirección sería:

```
imageRGB = imread('D:\OCR\D.jpg');
```

Suponiendo que la imagen está en la unidad D y dentro de la carpeta OCR. También la imagen puede ubicarse escribiendo una ruta local, usando un punto. Así:

```
imageRGB = imread('\.D.jpg');
```

El punto busca el archivo en la misma carpeta donde se guardó el script.

Otra manera es ingresar el nombre de la imagen por teclado para evitar estar cambiando el nombre de la misma desde el código. Esta es la opción elegida. Para realizar esto se utiliza la función input(), que pide dos argumentos de entrada: el texto a mostrar en pantalla y el tipo de dato que se va a ingresar. El tipo de dato es "s" de string o cadena pues va a ser texto ingresado por teclado. Luego la salida se guarda en la variable imagen, la cual es la entrada de la función imread().

El input se escribe así:

```
imagen=input('Ingresa el nombre de archivo con su  
extensión:\n','s'); % Ingreso imagen por teclado.  
imageRGB = imread(imagen); % Lee imagen como RGB.
```

La variable imageRGB guarda la lectura de imread() como una matriz indexada con números entre 0 y 255, lista para aplicarle funciones de procesamiento de imágenes.

imageRGB =

Columns 1 through 79

```
255 255 239 255 244 255 251 ... 241 236 245 255 232 255 255
255 253 254 218 255 250 253 ... 237 253 251 255 230 247 233
253 255 246 251 245 243 237 ... 248 255 245 240 255 240 255
255 250 255 255 241 247 254 ... 255 243 247 237 253 245 241
242 232 255 255 253 255 252 ... 255 251 233 255 255 235 254
228 254 251 248 4 11 0 37 ... 251 233 255 255 235 254 255
...
255 250 255 255 241 247 254 ... 241 236 245 255 232 255 255
```

5. **Convertir la matriz de imagen RGB a imagen escala de grises**, con la función `rgb2gray()` del "Image Processing Toolbox" del MATLAB

```
imagegray = rgb2gray(imageRGB);
```

6. **Convertir de imagen escala de grises a imagen binaria.**

La función `graythresh()` define el umbral (`threshold`) de imagen global utilizando el método de Otsu, como sigue:

```
threshold = graythresh(imagegray);
```

Luego la función `im2bw()` binariza la imagen con umbral guardado en la variable "threshold" y el operador binario (`~`) invierte colores. Binarizar la imagen es cambiar los valores de la matriz a ceros (0) y unos (1) e invertir colores es intercambiar valores, donde era cero ahora es uno y viceversa.

Luego la salida se guarda en la variable "imagebinary".

```
imagebinary = ~im2bw(imagegray, threshold);
```

La figura N° 3.05 muestra la imagen convertida a formato binario.



**Figura N° 3.05.-** Imagen en formato binario.

### 7. Filtrar basuritas (ruido) de las imágenes.

La función `bwareaopen()`, del "Image Processing Toolbox" del MATLAB, elimina objetos pequeños de la imagen.

Por ejemplo, en la imagen binaria guardada en la variable "imagebinary" se van a eliminar objetos que contienen menos de 30 píxeles.

```
imagebinary = bwareaopen(imagebinary,30);
```

Esto es importante cuando se llega a la etapa de reconocimiento de la letra, a fin que la letra quede lo más nítida posible. Ver figura N° 3.06.



**Figura N° 3.06.-** Imagen en formato binario y filtrado de imperfecciones.

### 8. Obtener campos de los formularios

En la ficha individual escaneada (ver figura N° 3.07), se procede a identificar los campos a reconocer (ver figura N° 3.08). Luego dos puntos, el primer punto la esquina izquierda superior del campo y el segundo punto la esquina derecha inferior del mismo campo (ver figura N° 3.09 y 3.10). Finalmente con la función `imcrop()` del "Image Processing Toolbox" se recorta el campo basándose en los dos puntos esquinas anteriores. La misma operación se realiza para los siguientes campos. Para ver el uso de la función `imcrop()`

revisar en el anexo B la "Mini Guía del MATLAB".

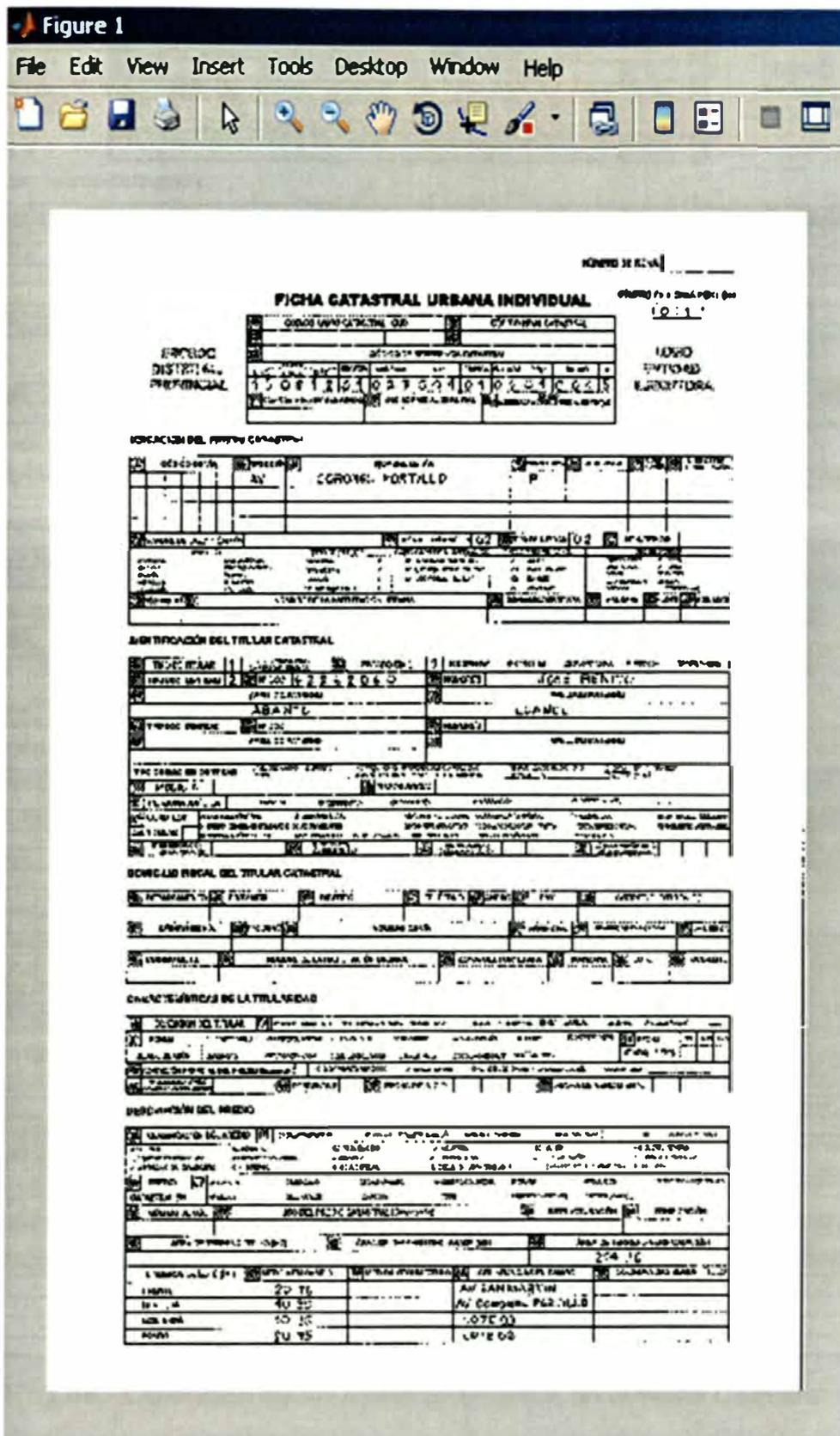


Figura N° 3.07.- Ficha catastral escaneada como imagen binaria.

NÚMERO DE FICHA

NÚMERO DE FICHAS POR LOTE

### FICHA CATASTRAL URBANA INDIVIDUAL

**ESCUDO  
DISTRITAL,  
PROVINCIAL**

CÓDIGO ÚNICO CATASTRAL - CUC				CÓDIGO HOJA CATASTRAL			
CÓDIGO DE REFERENCIA CATASTRAL							
TRAMO	SECTOR	MANZANA	LOTE	EDIFICIO	ENTRADA	PISO	UNIDAD DC
1	5	0	8	1	2	0	1
CÓDIGO CONTRIBUYENTE DE RENTAS				CÓDIGO PREDIAL DE RENTAS		ACUMULADA A CÓDIGO PREDIAL DE RENTAS	
1				0		2	

**LOGO  
ENTIDAD  
EJECUTORA**

**UBICACIÓN DEL PREDIO CATASTRAL**

CÓDIGO DE VÍA	TIPO DE VÍA	NOMBRE DE VÍA	TIPO DE PASEO	Nº MUNICIPAL	CÓDIGO NÚMERO	Nº DE CENSO DE NUMERACIÓN
	AV	CORONEL PORTILLO	P			

**IDENTIFICACIÓN DEL TITULAR CATASTRAL**

TIPO DE TITULAR	ESTADO CIVIL	Nº DOC.	NOMBRES
1	1	142262060	JOSE BENITO
APELIDO PATERNO		APELIDO MATERNO	
ABANTO		LLANOL	

**DOMICILIO FISCAL DEL TITULAR CATASTRAL**

DEPARTAMENTO	PROVINCIA	DISTRITO	TÉLFONO	ANEXO	FAX	CORREO ELECTRÓNICO

**CARACTERÍSTICAS DE LA TITULARIDAD**

CONDICIÓN DEL TITULAR	FORMA	DE ADQUISICIÓN	CONDICIÓN ESPECIAL DEL PREDIO
101	1	1	1

**DESCRIPCIÓN DEL PREDIO**

CLASIFICACIÓN DEL PREDIO	ÁREA DE TERRENO TÍTULO DAD	ÁREA DE TERRENO DECLARADA (M2)	ÁREA DE TERRENO VERIFICADA (M2)
101			214.76

UNIDADES DE LOTE (M2)	MEDIDA EN CAMPO	MEDIDA SEGÚN TÍTULO	COLONIAS EN CAMPO	COLONIAS SEGÚN TÍTULO
FRENTE	20.76		AV. SAN MARTÍN	
DERECHA	10.35		AV. CORONEL PORTILLO	
ESQUERDA	10.35		LOTE 03	
FONDO	20.75		LOTE 02	

Figura N° 3.08.- Obtención de los campos de datos en la Ficha Catastral en imagen binaria.

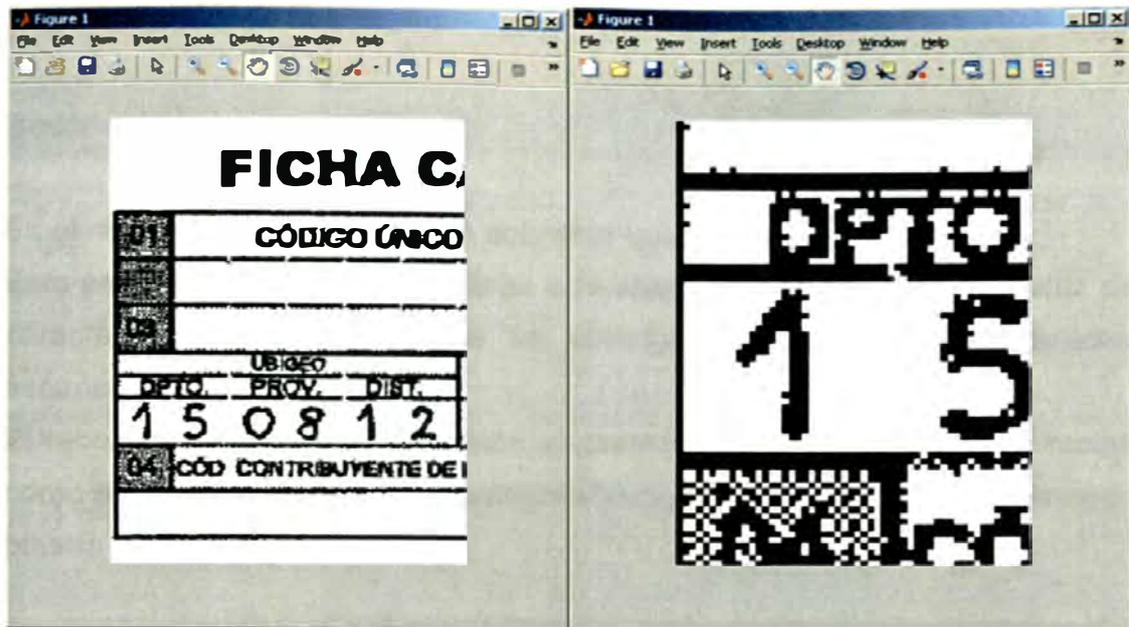


Figura N° 3.09.- Izq.: Acercamiento al ubigeo. Der.: Acercamiento a DPTO

Cada número es obtenido por separado porque cada campo de carácter en la ficha original está separado por una línea vertical que no debe entrar en el recorte del carácter porque provoca confusión. En la figura N° 3.09 en el campo DPTO, entre el 1 y el 5 se observa que no presenta la mencionada línea, pero entre los caracteres 2 y 7 del campo MANZANA en la figura N° 3.08, si se aprecia dicha línea.

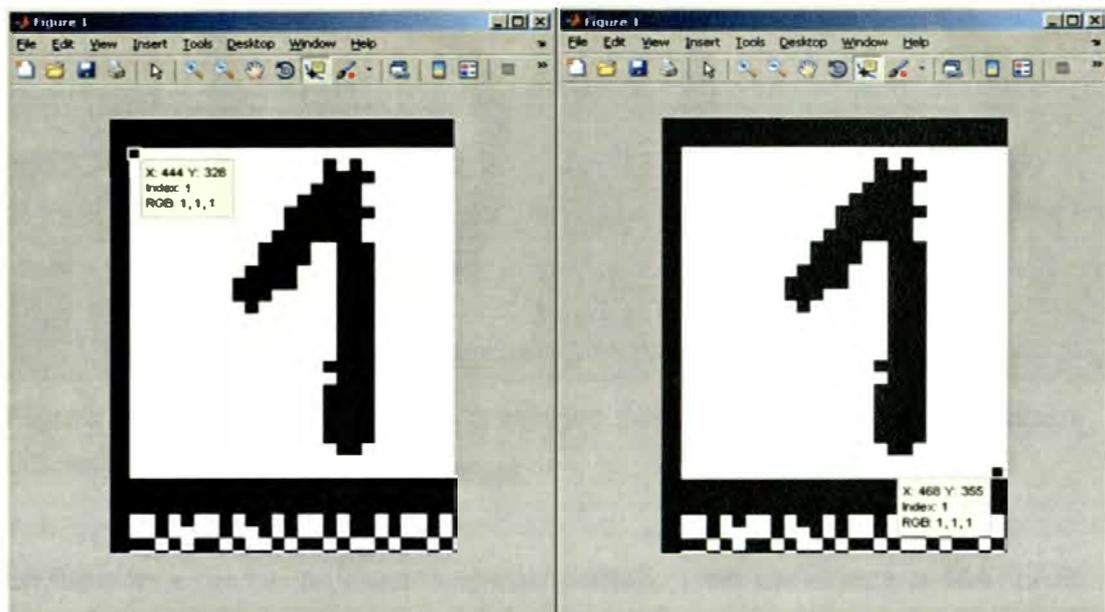


Figura N° 3.10.- Uso de la herramienta "data cursor" para tomar dos esquinas.

## 9. Segmentación de la imagen.

### 9.1. Recorte de la imagen.

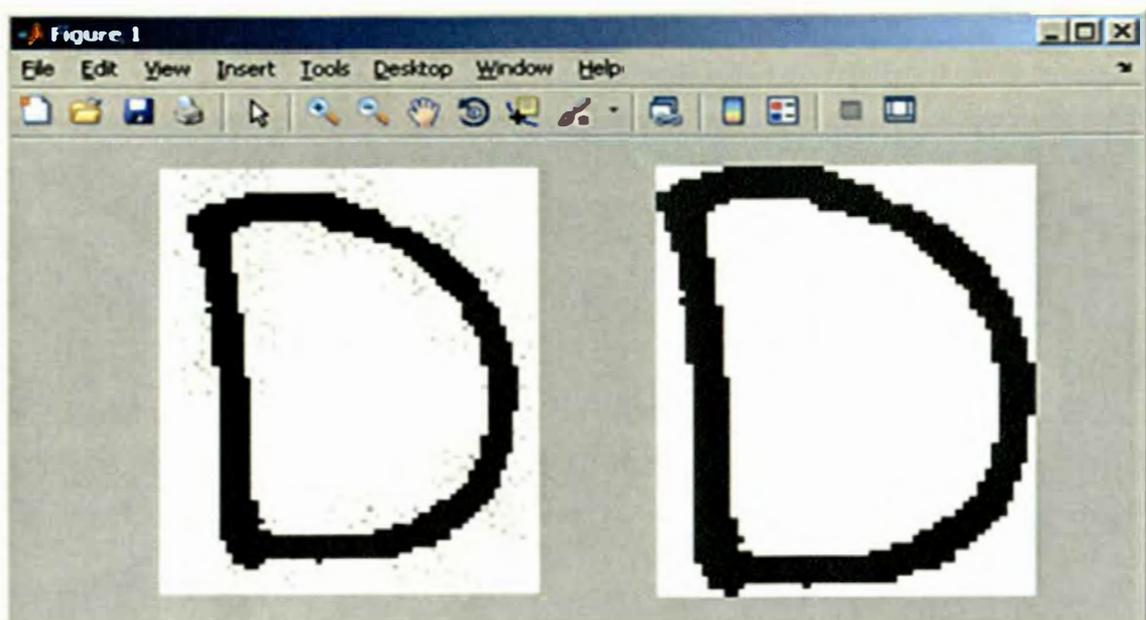
Es el recorte excedente de área sobrante que rodea al carácter.

Esto es importante cuando se llega a la etapa de obtención de la matriz de características, para que ésta se obtenga lo más similar al carácter escaneado.

El recorte se realiza con la función `ajustarAlTamanoDeLaImagen()` y recibe como argumento de entrada la variable "imagebinary" que guarda la imagen binaria limpia. Así:

```
imagenRecortada = ajustarAlTamanoDeLaImagen(imagebinary);
```

Una comparación entre la imagen original y la imagen recortada limpia de la letra D se muestra en la figura N° 3.11.



**Figura N° 3.11.-** A la izquierda la imagen original, a la derecha la imagen recortada y limpia.

La función `ajustarAlTamanoDeLaImagen()` no pertenece al MATLAB.

## 10. Obtener la matriz de características.

Con la función `obtenerMatrizDeCaracteristicas()` se obtiene un vector que representa con una matriz la letra D a partir de la imagen recortada.

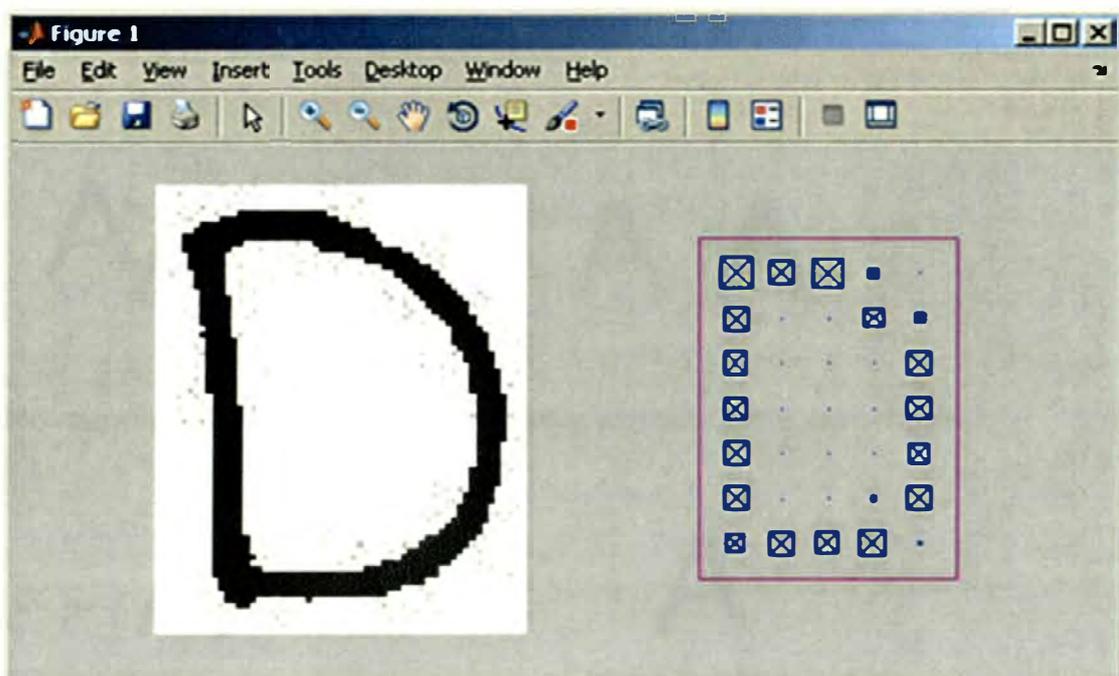
```
vectorLetra =
obtenerMatrizDeCaracteristicas(imagenRecortada);
```

La función `obtenerMatrizDeCaracteristicas()` no pertenece al MATLAB.

La matriz de características es un vector columna de tamaño 35 del patrón del carácter. El vector columna del patrón D es:

```
D = [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1
0 0 0 1 1 1 1 1 0]';
```

Con la función `plotchar()` se muestra el vector anterior como una letra D dibujada por elementos cuadrados con X, tal como lo muestra la figura N° 3.12. Se eligió tamaño 35 porque la función `plotchar()` plotea vectores de 7 filas por 5 columnas.



**Figura N° 3.12.-** A la izquierda la imagen original, a la derecha su matriz de características o vector patrón.

Para una lista completa de los vectores patrones de caracteres, ver el archivo "prprob.m" en el Anexo C.

## 11. Crear y entrenar la Red Neuronal Artificial.

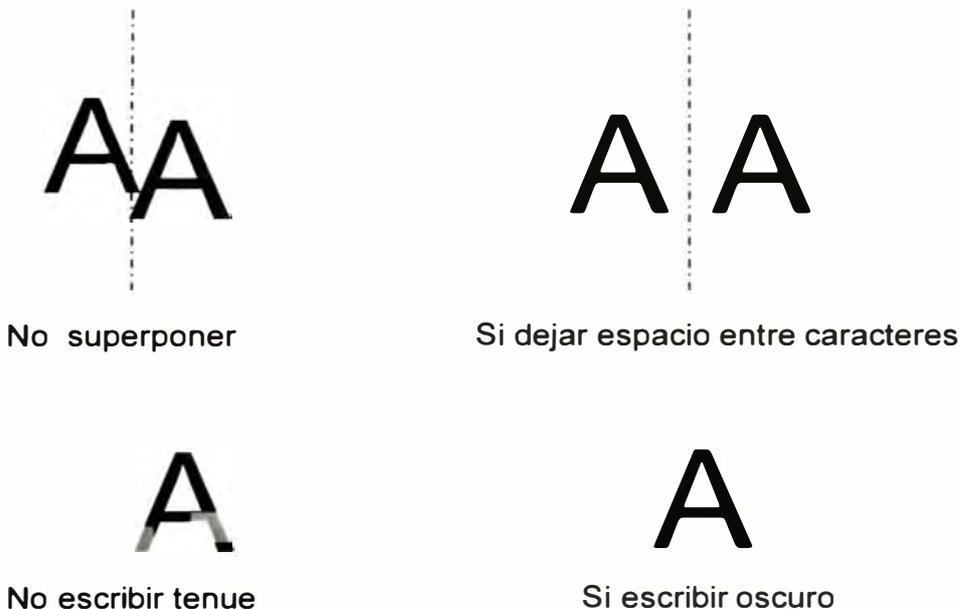
### 11.1. Se crea y entrena con la función crearRedNeuronal.

```
net = crearRedNeuronal();
```

La función crearRedNeuronal() no pertenece al MATLAB (su contenido completo se encuentra en el archivo crearRedNeuronal.m del anexo A) y crea la red neuronal artificial de la siguiente manera:

**Primero** se obtiene las entradas (alphabet) y salidas deseadas (targets).

La entrada "alphabet" es obtenida de un conjunto de caracteres escritos a mano (ver figuras N° 3.14 y 3.15) con un lapicero negro de tinta liquida en una hoja en blanco sobre una hoja cuadrículada para guiar la horizontalidad. Los caracteres tienen el tamaño de una cuadrícula y siempre deben tener un espacio horizontal que los separe, de tal manera que para el mismo renglón una línea vertical no debe cortar dos o más caracteres. En la figura N° 3.13 se muestra un ejemplo de lo expresado.



**Figura N° 3.13.-** Formas de escribir los caracteres de entrenamiento de la red.



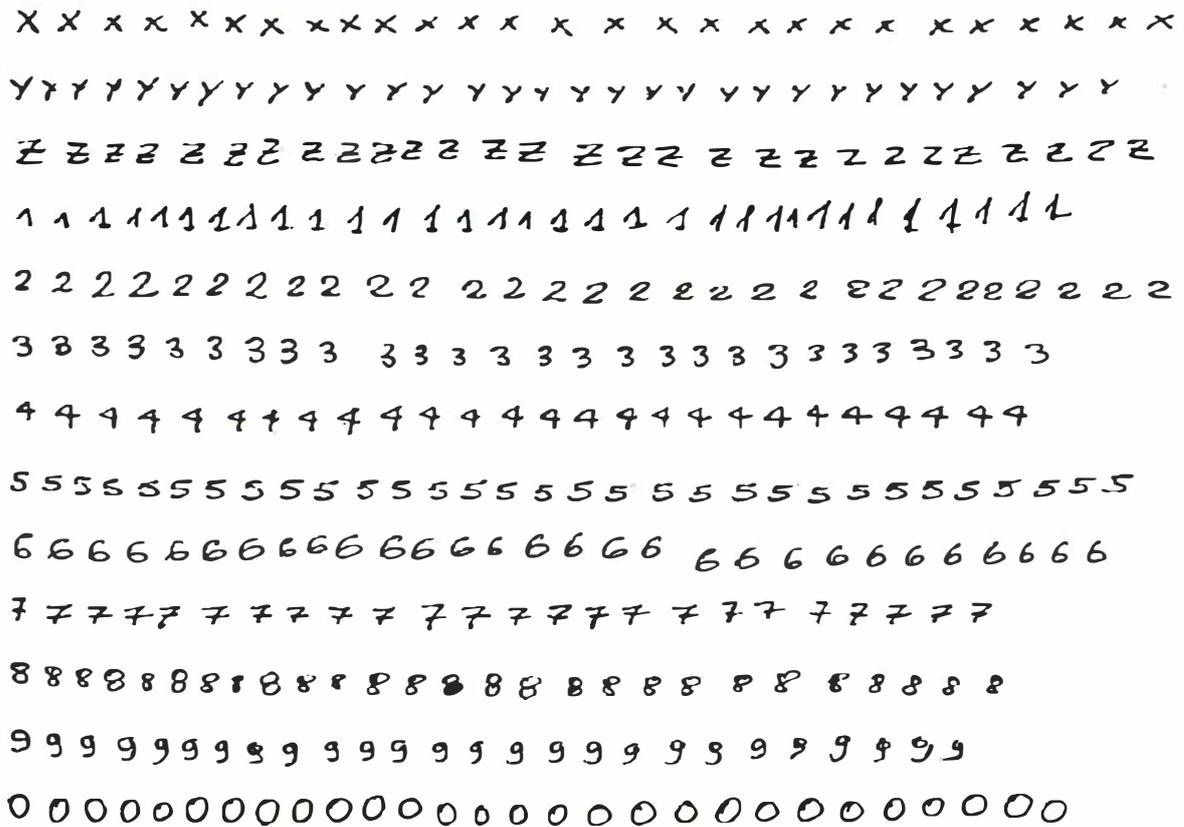


Figura N° 3.15.- Segunda hoja de los caracteres de entrenamiento de la red.

La entrada "alphabet" está formada por vectores columna de patrones de caracteres escritos a mano, algunos de los cuales se plotean y muestran en la figura N° 3.16. Estos fueron obtenidos con el "ICR2".

La entrada "targets" está formada de vectores unitarios, como se muestra en la tabla N° 3.01, los cuales fueron obtenidos de una hoja de cálculo.

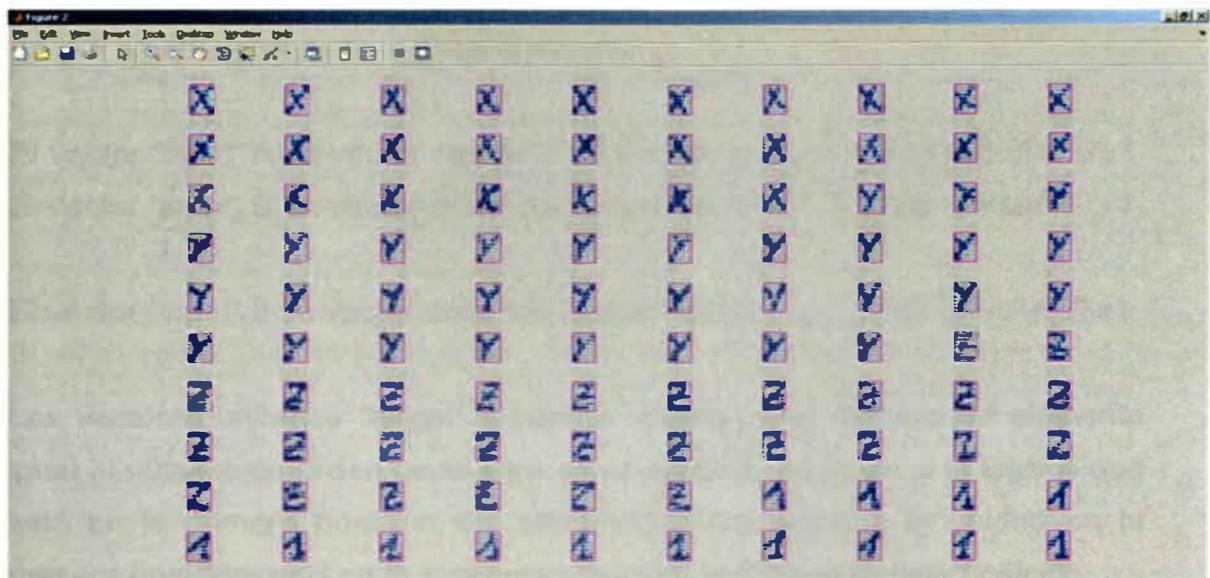


Figura N° 3.16.- Ejemplo de caracteres escritos reconocidos.

Adicionalmente, "alphabet" contiene vectores columna de patrones de caracteres contenidos en el archivo "script" "prprob.m", y estos se muestran a continuación:

El vector patrón de A es [0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1] de tamaño 35x1.

El vector patrón de B es [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0] de tamaño 35x1.

El vector patrón de C es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0] de tamaño 35x1.

El vector patrón de D es [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0] de tamaño 35x1.

...

El vector patrón de 8 es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0] de tamaño 35x1.

El vector patrón de 9 es [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0] de tamaño 35x1.

El vector patrón de 0 es [0 0 1 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0] de tamaño 35x1.

Los vectores patrones cuando se plotean con la función plotchar() muestran las letras y números a reconocer. (Ver figura N° 3.12). Entonces los vectores de "prprob" más los vectores de los caracteres escritos a mano conforman la entrada "alphabet" para el entrenamiento de la red neuronal.

La salida deseada "targets", guarda vectores columna unitarios que juntos forman una matriz identidad de 37x37 (ver tabla N° 3.01), y cada vector se asocia a cada letra de la siguiente manera:

El vector "input" A su vector columna "target" es [1, 0, ..., 0] de tamaño 37x1.

El vector "input" B su vector columna "target" es [0, 1, ..., 0] de tamaño 37x1.

...

El vector "input" 9 su vector columna "target" es [0, 0, ..., 1] de tamaño 37x1.

Los vectores unitarios "target" tienen la unidad en el número de elemento igual al número de orden de la letra en el alfabeto, es decir, a la letra A que está en la primera posición del alfabeto, le corresponde la unidad en la primera posición; la B en la segunda posición, la C en la tercera posición, ..., el 8 en la trigésima cuarta posición y el 9 en la trigésima quinta posición.

**Tabla Nº 3.01.- Vectores columna "targets" de cada patrón de carácter para el entrenamiento de la red.**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6	7	8	9	0	.			
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fuente: Elaboración propia.

Se crean manualmente las variables “alphabet” y “targets” en MATLAB. Los vectores “alphabet” y “target” se asignan a las variables “alphabet” y “targets” respectivamente. Luego estas variables se guardan manualmente en un archivo “datosFicha.mat”, el cual las devuelve con el comando “load”, como se muestra a continuación:

```
load datosFicha; %devuelve "alphabet" y "targets"
```

**Segundo**, se crean 5 caracteres con ruido (distorsión del patrón original) por cada carácter de “alphabet” y se adicionan a este y la suma se guarda en la variable “alphabet2”. Estos caracteres con ruido mejoran el reconocimiento de caracteres (por el resultado obtenido en la figura N° C.02 del anexo C) y se obtienen con las funciones repmat(), randn() y un número de ruidos (numNoise) de 5. El siguiente código lo muestra:

```
numNoise = 5;  
alphabet2=min(max(repmat(alphabet,1,numNoise)+randn(  
35,35*numNoise)*0.2,0),1);
```

0.2 es el nivel de ruido y su valor varía la capacidad de reconocimiento de una red entrenada. La obtención del valor 0.2 se explica el anexo C, en “cálculo del nivel de ruido”.

También se multiplican los “targets” un número de veces igual al número de ruido (numNoise) y se guarda el resultado en “targets2”. Así:

```
targets2 = repmat(targets,1,numNoise);
```

**Tercero**, la función feedforwardnet() crea la red neuronal con 35 neuronas ocultas y función de entrenamiento ‘trainscg’ y se guarda en la variable “net”. Son 35 neuronas para que sea una por cada letra y número. El código se muestra a continuación:

```
net = feedforwardnet(35, 'trainscg');
```

**Cuarto**, se entrena la red con ruido, donde "alphabet2" y "targets2" deben tener el mismo número de columnas. El entrenamiento se realiza con la función train(), número de periodos de 1000 y error de iteración de cero, luego el resultado se guarda en la variable "net" (esta función está presente en MATLAB 2011, no corre en MATLAB 2009). El código se muestra a continuación:

```
net.trainParam.epochs = 1000; % Periodos.  
net.trainParam.goal = 0; % Error de iteración.  
net = train(net,alphabet2,targets2);
```

Luego se obtiene la salida "y" evaluando la entrada "alphabet" en la red "net" para obtener el error global (un solo valor) y un vector error entre "y" y "targets".

En la figura 3.17 se muestra la interfaz del entrenamiento con la función train(). A continuación se muestra el código integrado dentro de la función crearRedNeuronal(). Así:

```
function net = crearRedNeuronal()  
  
load datosFicha; %devuelve "alphabet" y "targets".  
  
setdemorandstream(pi); % Para obtener siempre el mismo  
conjunto de valores aleatorios en cada ejecución. Es opcional.  
numNoise = 5;  
alphabet2 =  
min(max(repmat(alphabet,1,numNoise)+randn(35,35*numNoise)*0.2,  
0),1); %0.2 es el máximo nivel de ruido (noiseLevel) para  
error cero. Nivel de ruido esta entre 0 y 1.  
targets2 = repmat(targets,1,numNoise);  
  
% Para reconocer este problema usaremos una configuración de  
red neuronal feedforward para reconocimiento de patrones con  
35 neuronas ocultas.  
net = feedforwardnet(35, 'trainscg'); % Crea la red neuronal  
con 35 neuronas ocultas.  
net = configure(net,alphabet2,targets2); % Obtiene los  
parámetros de configuración.  
net.trainParam.epochs = 1000; % Periodos.  
net.trainParam.goal = 0; % Error de iteración.  
net = train(net,alphabet2,targets2); % Entrena la red con  
ruido.  
  
y = net(alphabet); % obtiene la salida de la red para  
"alphabet".
```

```
errores = gsubtract(targets,y); Resta "target" menos "y".
performance = perform(net,y,targets) % Obtiene un error
global.
```

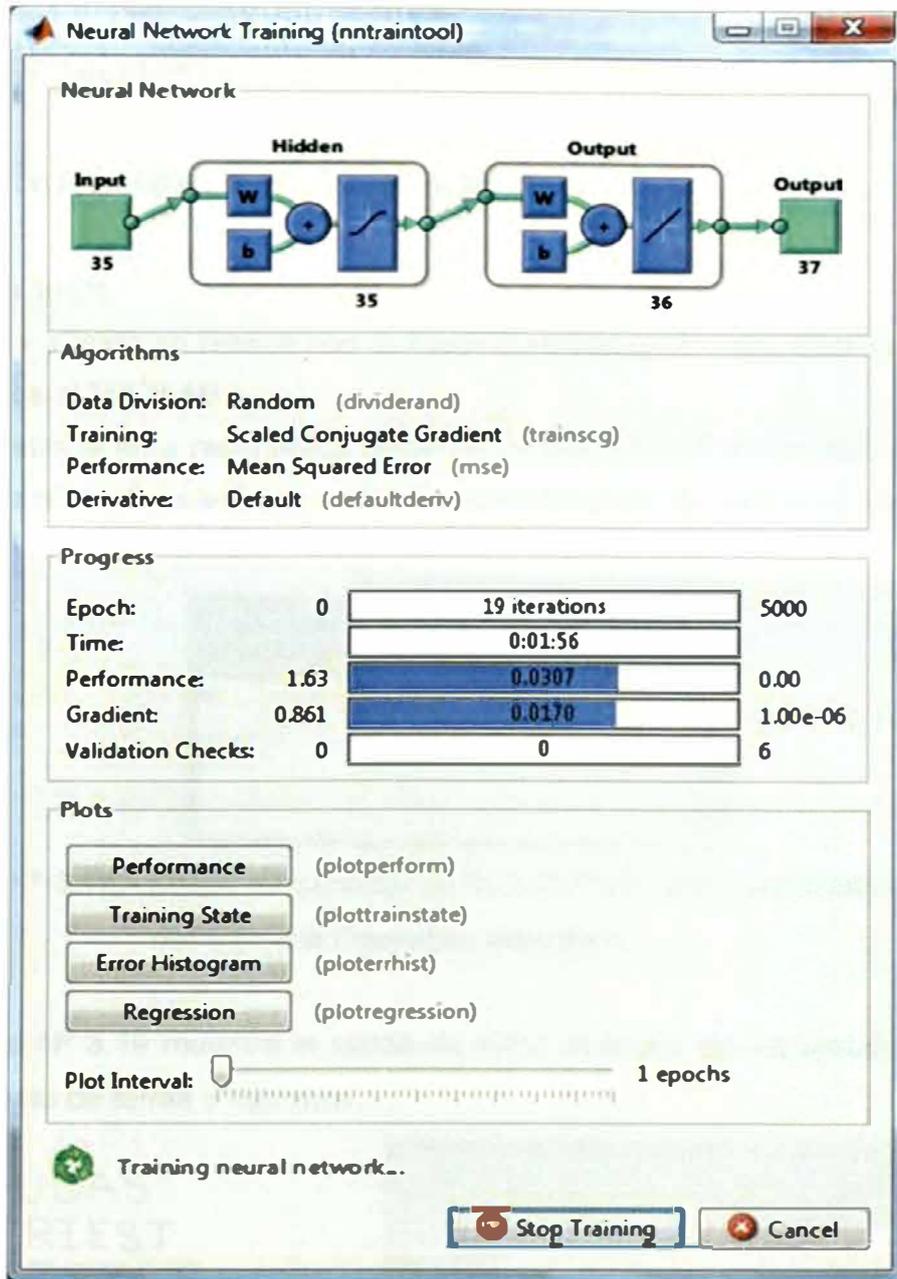


Figura Nº 3.17.- Entrenamiento de la red con el "Neural Network Training Toolbox" del MATLAB.

Luego se guarda manualmente el objeto de la red neuronal llamado "net" devuelto por la función crearRedNeuronal() en un archivo "RedEntrenada.mat" para usarlo en las siguientes fichas.

## 12. Reconocer el carácter.

Con la función LeerLetra() se reconoce el carácter utilizando la red entrenada.

```
letra = leerLetra(vectorLetra, net);
```

La función leerLetra() no pertenece al MATLAB.

## 13. Salida a texto.

La salida a texto se realiza con la función salidaLetraATxt(). Esta función no pertenece al MATLAB.

Se muestra la letra reconocida como un carácter ASCII en un bloc de notas. La figura N° 3.18 muestra la salida de ICR1 después de reconocer la letra D



Figura N° 3.18.- Salida de carácter en formato "txt" leído por el bloc de notas del Sistema Operativo Windows.

La figura N° 3.19 muestra la salida de ICR2 después de reconocer un texto compuesto de letras y números.

JUDAS  
PRIEST  
775758  
HOLA  
DIEGO  
12312  
367945

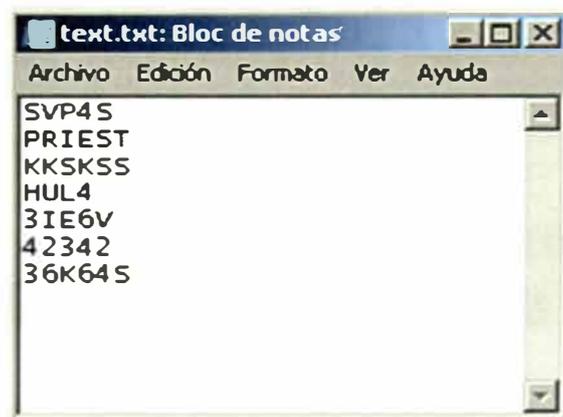


Figura N° 3.19.- Salida de texto en formato "txt" leído por el bloc de notas del Sistema Operativo Windows.

Se aprecia que todavía falta entrenar la red para reconocer más casos de salida de texto.

#### 14. Prueba y ejecución del programa.

Escribir el nombre del script principal en la consola de la ventana de comandos del MATLAB (en este caso fue ICR1 e ICR2). De esta forma:  
 >> ICR1 o ICR2.

#### 3.3.1. Resultado del reconocimiento de datos en una ficha catastral

En la figura N° 3.20 se muestran las imágenes de los campos de datos a reconocer obtenidos de la ficha catastral de la figura N° E.01. Cabe mencionar que el entrenamiento de la red para reconocer caracteres en una ficha catastral se realizó con estas letras y números mas no con los caracteres de la figura N° 3.14 y 3.15, por los resultados obtenidos en la figura N° 3.19.

150812 01 027 001 01 01001 5  
 AV CORONEL PORTILLO P  
 02 02  
 1 1  
 2 42262060 JOSE BENITO  
 ABANTO LLANOL  
 01 01  
 07 214.76  
 20.76 AV. SAN MARTIN  
 10.35 AV. CORONEL PORTILLO  
 10.35 LOTE 03  
 20.75 LOTE 02

Figura N° 3.20.- Letras en imágenes recortadas.

Los caracteres reconocidos en MATLAB, los caracteres reconocidos en un archivo de texto delimitado por comas "csv" y en una tabla de Excel se muestran en las figuras N° 3.20, 3.21 y tabla N° 3.02, respectivamente.

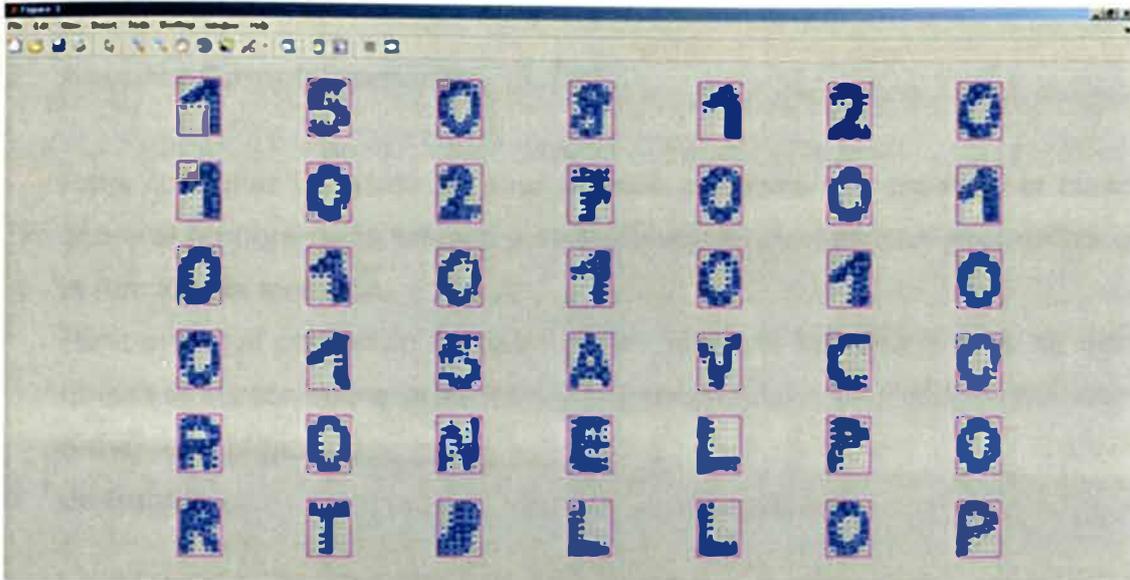


Figura N° 3.21.- Letras en MATLAB.

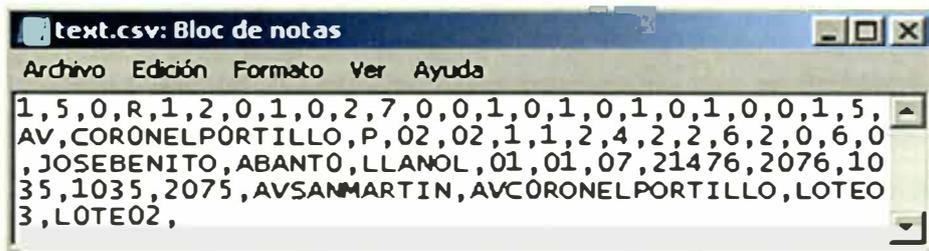


Figura N° 3.22.- Letras en archivo de texto de formato "csv".

Tabla N° 3.02.- Datos de la ficha catastral en una hoja de Excel.

Num	Dpto	Prov	Dist	Sector	MZ	Lote	Edif	Entra	Piso	Und	DC	TipoVia
1	15	OR	12	01	027	001	01	01	01	001	5	AV

NombVia	TipPue	TipEdif	TipInt	TipTitu	EstCiv	TipDoctd	DNI
CORONELPORTILLO	P	02	02	01	01	02	42262060

Nombres	ApPate	ApMate	Num	ClasPred	PredEn	AreaLoteVerif(m2)
JOSEBENITO	ABANTO	LLANOL	1	1	7	21476.00

FrenteCampo	DerCampo	IzqCampo	FondoCampo	ColindFrentecampo
2076.00	1035.00	1035.00	2075.00	AVSANMARTIN

ColindDerCampo	ColindIzqcampo	ColindFondoCampo
AVCORONELPORTILLO	LOTEO3	LOTEO2

Fuente: *Elaboración propia.*

### 3.3.2. Apuntes Complementarios

Para consultar la ayuda de una función presionar F1 estando el cursor sobre el nombre de la función y se desplegará una ventana informativa de la función en mención.

Para entrar al contenido de una función desde la llamada a ésta, se debe ubicar el cursor sobre la función y presionar Ctrl + D. Previamente debe haberse configurado la carpeta de trabajo en la ubicación de los archivos de trabajo.

### 3.4. RESUMEN DEL PROCEDIMIENTO

Las fichas catastrales se rellenan usando un lapicero azul o negro de "tinta líquida". Luego se escanean a colores en formato "jpg" o "tif" sin inclinación y dobleces. Evitar borrones, marca de corrector y lápiz.

Se escribe diferentes juegos de caracteres o abecedario (todas las letras mayúsculas, los dígitos del "0" al "9" y signos de puntuación) con diferentes tamaños en una hoja en blanco para escanearlo y sirva de entrada para el entrenamiento. Luego el programa procede al aislamiento de cada uno de los caracteres que forman el texto en los campos y posteriormente a la extracción de un conjunto de características capaces de definir cada carácter aislado. Una vez obtenido el vector de características de cada uno de los caracteres incluidos en el documento, se activa la etapa de reconocimiento, que en este caso consiste en una red neuronal capaz de generar como salida el código ASCII de 8 bits del carácter cuyo vector de características recibe como entrada.

Después se somete la red neuronal a un proceso de aprendizaje, en el cual se utilizan los diferentes juegos de caracteres escritos. Durante este proceso se ajustan los parámetros internos de la red de forma reiterativa

hasta que sea capaz de generar como salida el código ASCII del carácter cuya imagen se presenta a su entrada, para todas las formas y tamaños considerados para tal carácter. Después el programa separa la salida por comas y lo guarda en un archivo delimitado por comas (extensión "csv") que es leído como columnas en una hoja de Excel. Finalmente con una macro se ordenan los datos en una tabla de Excel. Al final se guarda la red entrenada en un archivo (extensión .m) de MATLAB, de tal manera que el aprendizaje se realice solo una vez y sirva para futuras fichas catastrales.

Para las siguientes fichas escaneadas solo será necesario ejecutar el programa usando la red entrenada y guardada anteriormente, de esta manera se ahorra tiempo de procesamiento, pues cada ejecución es un procesamiento de datos de una ficha catastral en una hoja de Excel.

### **3.5. COMPARACIÓN CON EL PROGRAMA SMARTZONE.**

Para reconocer exitosamente los textos escaneados se usa el programa SMARTZONE, que usa la tecnología de redes neuronales para hacer reconocimiento OCR e ICR. Pertenece a la empresa ACCUSOFT. Se usó la versión de prueba llamada "FormAssist", el cual está especializado en el reconocimiento de texto en fichas o formularios de registro escritas a mano. También existen versiones gratuitas sin compilar en modo de prueba (solo hasta que se las compre) descargando la plataforma SDK desde la web del programa.

El procedimiento de uso es:

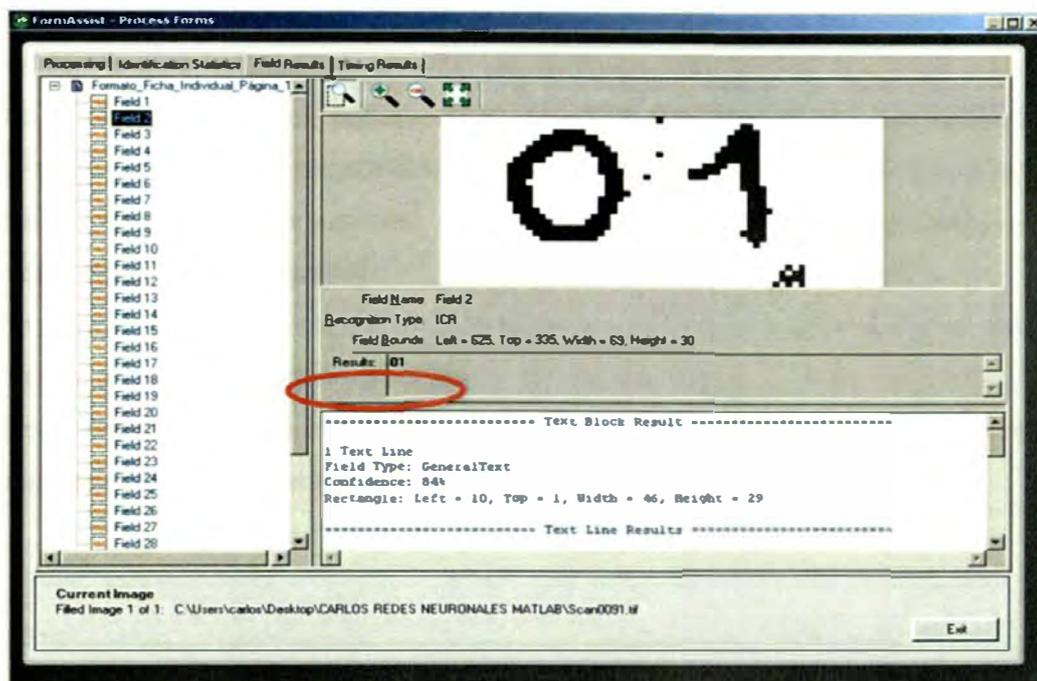
1. Escanear dos fichas en formato monocromo. La primera sin rellenar y la segunda rellena de datos. El escáner debe estar configurado para dar su salida en blanco y negro. La primera ficha sin rellenar servirá como ficha patrón, el programa compara esta ficha patrón, que teóricamente está perfectamente alineada y escalada, con la segunda ficha para ajustar la verticalidad y escala de la ficha segunda.
2. Ejecutar "FormAssit" y elegir "Create a new form template library", luego ok.

3. Menu File > Add Form to form set... > Elegir la ficha patrón.
4. Reconocer y resaltar los campos del formulario donde hay texto a reconocer. Menu Tools > ICR Field o con la herramienta ICR. 

5. Procesar. Menu tools > Process form o el botón . Luego elegir la imagen de la segunda ficha escaneada completa con datos para reconocer.

Al procesar se mejora la imagen, se identifica el formulario y alinea la verticalidad. Luego el programa reconoce el texto encasillado en los campos identificados y señalados anteriormente.

6. En la sección "Field Result", se muestra los resultados en forma independiente (ver figura N° 3.23), los cuales pueden ser copiados manualmente a un archivo de texto. Ver la figura N° 3.24.



**Figura N° 3.23.-** Reconocimiento ICR exitoso de "SmartZone" de un campo de la ficha catastral.

El resultado se muestra en "Results". Luego manualmente se copia y pega desde "Results" a un archivo de texto.

Los datos a reconocer se mostraron en la figura N° 3.08.

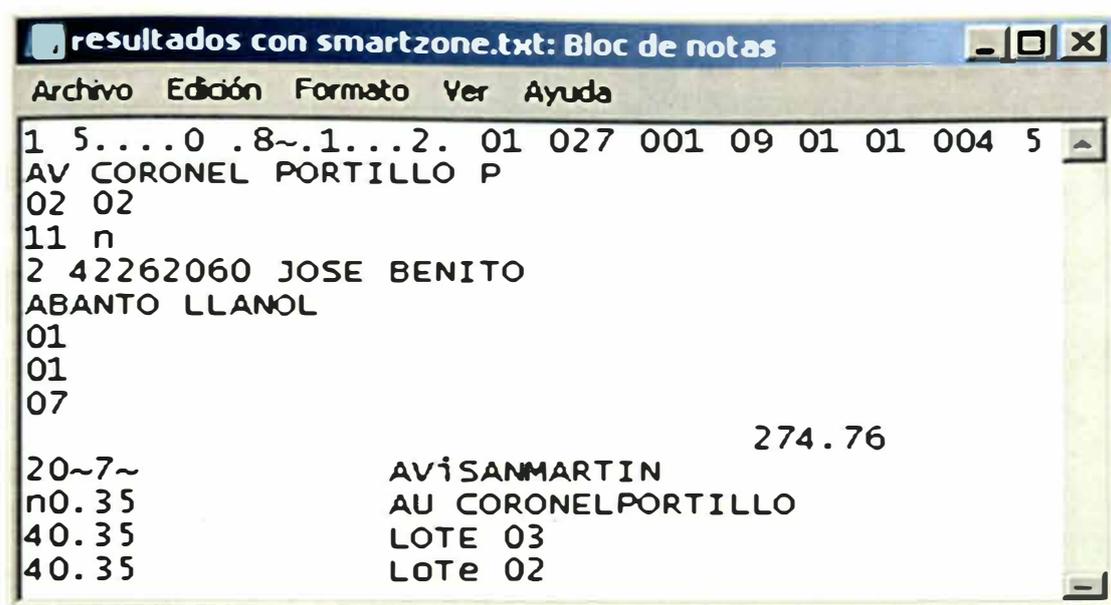


Figura Nº 3.24.- Salida de texto de los campos reconocido por "SmartZone".

Se observa que la salida reconoce un 80% los caracteres del texto escrito a mano. Falta mejorar el reconocimiento ICR del programa SMARTZONE, puesto que son redes neuronales que reconocen diferentes estilos y formas de texto pero de forma general, no especifica a la caligrafía del texto de los campos de la ficha catastral.

El programa debe ser capaz de entrenar su ICR con entrada personalizada de texto escrito, posiblemente, esta limitación de no poder ingresar texto personalizado se debe a que el programa es una versión de prueba mas no una versión completa con todas sus funcionalidades.

# CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

## 4.1. CONCLUSIONES

- Para lograr un buen resultado, se debe entrenar la red neuronal con un mínimo de 30 patrones de entrada de cada letra y número con diferentes tamaños y formas. Además para evitar mutilaciones y huecos de los caracteres en el formato binario, se debe escribir con lapicero tinta líquida de la misma caligrafía de la persona que escribió los caracteres. También para mejorar el entrenamiento con la función `train()` es necesaria la inclusión de ruido mediante funciones estocásticas de MATLAB.
- Para evitar confusión de la red al clasificar patrones de caracteres en el entrenamiento, es necesario limpiar manchas y puntos no deseados (ruidos) usando la función `bwareopen()` además recortar los campos de caracteres con la función `imcrop()`.
- Para tener como resultado un texto en minúsculas, se usa la función `lower()` sobre la variable "palabra" en la función `salidaAtxt()`.
- Se debe resolver la ambigüedad entre el cero y la letra "O".
- Se debe resolver reconocer espacios entre palabras con la función `imcrop()`.
- Se puede usar también la función `pattemnet()` para crear redes neuronales en el reconocimiento de patrones.
- Los parámetros usados en la red son 35 neuronas ocultas, 5 patrones con ruido de cada carácter, función de entrenamiento "trainscg", periodos de 1000, error de 0.0001, porque los resultados obtenidos son buenos. Sin embargo debido a las limitaciones del procesador de mi laptop, no se ha experimentado con periodo de 5000, error de cero, función de entrenamiento "trainrp" [14], inicializar pesos con  $iw\{1,1\}$  y constante  $b\{1\}$ . [15]. No usar

demasiadas neuronas porque se sobrecarga la capacidad de generalización (predecir nuevas formas de caracteres) de la red.

- De los procedimientos de uso del MATLAB se determina que la imagen a reconocer debe estar en formato monocromo, el ángulo de inclinación de la imagen debe ser cero, el número de neuronas ocultas máximo es una por cada carácter diferente. Además se deben considerar más factores para el reconocimiento como: la profundidad del entrenamiento (número de patrones de entrada) y el número de entrenamientos de la Red Neuronal.
- Se puede escribir datos en una ficha digital usando una tableta personal, pero el texto escrito no sirve para formar una base de datos de archivo de texto.
- En los procedimientos de llenado de las fichas, la escritura de las letras a reconocer es tipo arial, mayúscula, con trazos lentos, firmes, rectos. En mayúsculas porque es más legible y para evitar los trazos curvos malos producto de una mala caligrafía.

#### 4.2. RECOMENDACIONES

- Se recomienda extender el programa CARLOSICR con códigos que produzcan coordenadas de corte de imagen a la función `imcrop()` para tener totalmente automatizado el proceso de reconocimiento de caracteres.
- Se recomienda comparar resultados con otros programas como "neuron" (neuron software), el cual ha sido usado en el programa RECAM. [5]
- No pegarse a los bordes de los campos de datos para no tener problemas en la etapa de segmentación de texto en el desarrollo del programa.
- Escribir con lapicero negro de tinta líquida para evitar huecos en las letras de imagen binaria que reduzcan el reconocimiento de los textos escaneados.
- Se recomienda escanear en formato de archivo de imagen TIF no en formato JPG, puesto que este último deteriora la calidad original de la imagen por ser un formato comprimido que ocupa menos Kilobytes de espacio.

- Usar una PC con memoria RAM mínima de 4 GB, porque el entrenamiento de la red con 35 neuronas ocultas usando la función de entrenamiento “trainlm” lo necesita.
- Para el reconocimiento de texto en las fichas de catastro se recomienda utilizar el programa SmartZone (usa la tecnología de redes neuronales), porque es un “software” entrenado para múltiples casos de reconocimiento de caracteres escritos a mano.

## BIBLIOGRAFÍA

- [1] BARRAGÁN GUERRERO, Diego Orlando. **“Reconocimiento de Caracteres Ópticos (OCR) usando MATLAB”**. Artículo, Matpic, Loja - Ecuador, 2011.  
<http://www.matpic.com/esp/MATLAB/ocr.html>
- [2] CALA. **“Red Neuronal”**. Epistemowikia. Revista Hiperenciclopédica de Divulgación del Saber. Segunda Época, Año VI. Vol. 5. Núm. 4. Octubre 2011.
- [3] CUEVAS JIMENEZ, Erik V., ZALDIVAR NAVARRO, Daniel. **“Visión por Computador utilizando MATLAB y el Toolbox de Procesamiento Digital de Imágenes”**. Artículo, UTPL Madrid. 2006.
- [4] DELGADO, Alberto. **“Inteligencia Artificial y Minirobots”**. EDOE Ediciones. Segunda Edición. Santa Fé de Bogotá D.C. - Colombia, Julio 1998.
- [5] FLORES QUISPE, Roxana. **“Reconocimiento de caracteres manuscritos aislados en campos de formularios utilizando redes neuronales”**. Tesis de maestría, FIEE / 2006 / M-328, N° 27, Facultad de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Ingeniería, Lima-Perú. 2006.
- [6] GOPAL, Sucharita. **“Unit 188 - Artificial Neural Networks for Spatial Data Analysis”**. Department of Geography and Centre for Remote Sensing. Boston University. Boston. 1998.  
<http://www.ncgia.ucsb.edu/giscc/units/u188/u188.html>
- [7] HAYKIN, S. **“Neural Networks: A Comprehensive Foundation”**. NY: Macmillan, p. 2. 1994.  
<http://www.ncgia.ucsb.edu/giscc/units/u188/u188.html>
- [8] HUDSON BEALE, Mark. T. HAGAN Martin, B. DEMUTH Howard. **“Neuronal Network ToolBox User’s Guide”**, MathWorks, R2011b, September 2011.
- [9] MATHWORKS. **“Neural Network Toolbox Character Recognition”**. Menu Help > Product Help > Contents > Neural Network ToolBox > Demos > Pattern Recognition and Classification > Character Recognition > appcr1.m. MATLAB 2011b version 7.13.0.564, 2011.
- [10] MATHWORKS. **“Neural Network Toolbox Character Recognition”**. Menu Help > Product Help > Contents > Neural Network ToolBox > Demos > Pattern Recognition and Classification > Character Recognition > appcr1.m.

MATLAB 2009.

- [11] MATHWORKS. “**Character Recognition: constant and linear models**”. “News Reader” digital de MathWorks, MATLAB 2011b version 7.13.0.564, 2011.
- [12] MATHWORKS. “**Imaging Processing Toolbox Getting Started**”. Menu Help > Product Help > Contents > Imaging Processing Toolbox > Getting Started > Example1 - Reading and writing Images - MATLAB 2011b version 7.13.0.564, 2011.
- [13] MATHWORKS. “**Neural Network Toolbox User's Guide**”. Menu Help > Product Help > Contents > Neural Network Toolbox > User's Guide > Functions > New Networks Functions. MATLAB 2011b version 7.13.0.564, 2011.
- [14] MATHWORKS. “**Neural Network Toolbox User's Guide**”. Menu Help > Product Help > Contents > Neural Network Toolbox > User's Guide > Multilayer Networks and Backpropagation Training > Train the Network > Training Algorithms. MATLAB 2011b version 7.13.0.564, 2011.
- [15] MATHWORKS. “**Neural Network Toolbox User's Guide**”. Menu Help > Product Help > Contents > Neural Network Toolbox > User's Guide > Functions > Network Initialization Functions > Init. MATLAB 2011b version 7.13.0.564, 2011.
- [16] NEBENDA, Dieter. “**Sistemas Expertos. Ingeniería y Comunicación**”. Editores Marcombo. Barcelona 1988.
- [17] ORDÓÑEZ L., Juan Pablo. “**Reconocimiento Óptico de Caracteres (OCR) con Redes Neuronales**”. Paper, estado del arte, Loja - Ecuador, 2009.  
<http://jpordonez.wordpress.com/2009/06/14/reconocimiento-optico-de-caracteres-ocr-con-redes-neuronales/>
- [18] ROMERO Luis Alonso. “**Reconocimiento de caracteres escritos usando redes neuronales**”. Applets de redes neuronales, Catedrático de Ciencia de la computación e inteligencia Artificial, Facultad de Ciencias, Salamanca, España, 2006.  
<http://avellano.usal.es/~lalonso/RNA/AppletORT.html>
- [19] TELLO G. Edwin. “**Detección de caracteres mediante Redes Neuronales con entrada de datos con Excel**”. Informe de suficiencia, 786-TGE, Facultad de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Ingeniería, Lima-Perú. 2005.

[20] WIKIPEDIA. “ICR”. Fundación Wikipedia. 2011.

# ANEXOS

# ANEXO A

## Anexo A: Código del programa “CARLOSICR” en MATLAB

### Reconociendo Caracteres

#### ICR1.m

```
% Neural ICR (Intelligent Character Recognition).  
% Modificado y adaptado por Carlos de la Cruz de dos fuentes:  
% OCR MATLAB de MatPic: http://www.matpic.com/esp/MATLAB/ocr.html  
% OCR Redes Neuronales de Ordoñez:  
http://jpordonez.wordpress.com/2009/06/14/reconocimiento-optico-de-caracteres-ocr-con-redes-neuronales/  
  
clear % Limpia la ventana de comandos.  
clc % Limpia el workspace.  
imagen=input('Ingresa el nombre de archivo con su  
extensión:\n','s'); % Ingreso imagen por teclado.  
imageRGB = imread(imagen); % Lee imagen como RGB.  
imagegray = rgb2gray(imageRGB); % Convierte imagen RGB a escala de  
grises.  
threshold = graythresh(imagegray); % Umbral (threshold) de la  
imagen global utilizando el método de Otsu.  
imagebinary = ~im2bw(imagegray,threshold); % Binariza la imagen e  
invierte colores.  
imagebinary = bwareaopen(imagebinary,30); % Filtra basuritas  
(ruido) de las imagenes. Remueve todos los objetos que contienen  
menos de 30 pixeles.  
imagenRecortada = ajustarAlTamanoDeLaImagen(imagebinary); %  
Recorta espacios blancos o vacios.  
imagenRecortada = ~imagenRecortada; % Cambia a letras negras. Con  
operador binario NOT (~) letras son negras (ceros) y fondo es  
blanco (unos).  
vectorLetra = obtenerMatrizDeCaracteristicas(imagenRecortada); %  
Redimensiona a una matriz de 5 x 7 representado como vector.  
net = crearRedNeuronal(); % Crea y entrena la red.  
subplot(1,2,1), imshow(imagen); % Muestra la imagen original.  
subplot(1,2,2), plotchar(vectorLetra); % Muestra su vector letra  
de la imagen.  
letra = leerLetra(vectorLetra, net); % Convierte de vector a  
texto. Solo reconoce mayúsculas!.  
salidaLetraAtxt(letra); % Guarda en la letra en un archivo  
de texto y lo abre.
```

## Reconociendo Palabras

### ICR2.m

```
% Neural ICR (Intelligent Character Recognition).
% Modificado y adaptado por Carlos de la Cruz de dos fuentes:
% OCR MATLAB de MatPic: http://www.matpic.com/esp/MATLAB/ocr.html
% OCR Redes Neuronales de Ordoñez:
http://jpordonez.wordpress.com/2009/06/14/reconocimiento-optico-
de-caracteres-ocr-con-redes-neuronales/

clear % Limpia la ventana de comandos.
clc % Limpia el workspace.
imagen=input('Ingresa el nombre de archivo con su
extensión:\n','s'); % Ingreso imagen por teclado.
imageRGB = imread(imagen); % Lee imagen como RGB.
imagegray = rgb2gray(imageRGB); % Convierte imagen RGB a escala de
grises.
threshold = graythresh(imagegray); % Umbral (threshold) de la
imagen global utilizando el método de Otsu.
imagebinary = ~im2bw(imagegray,threshold); % Binariza la imagen e
invierte colores.
imagebinary = bwareaopen(imagebinary,30); % Filtra basuritas
(ruido) de las imagenes. Remueve todos los objetos que contienen
menos de 30 pixeles.
net = crearRedNeuronal(); % Crea y entrena la red.
imshow(imagen); % Muestra la imagen original.
salidaAtxt(net, imagebinary); % Segmenta, convierte a texto y
crea la salida en txt.
```

### Códigos de las funciones utilizadas por ICR1 e ICR2.

Las funciones a continuación son comunes a ambos. Notar que las funciones están asociadas a un script del MATLAB (extensión .m) con el mismo nombre.

### crearRedNeuronal.m

```
function net = crearRedNeuronal()

load datosFicha; %devuelve "alphabet" y "targets"

setdemorandstream(pi); % Para tener siempre los mismos valores
aleatorios con randn(). Es opcional.

numNoise = 5; % 5 veces, 5 pasadas.
alphabet2 =
min(max(repmat(alphabet,1,numNoise)+randn(35,35*numNoise)*0.2,0),1
); %0.2 es el máximo nivel de ruido (noiseLevel) para error cero.
Nivel de ruido esta entre 0 y 1.
targets2 = repmat(targets,1,numNoise);
```

```
% Para reconocer este problema usaremos una configuración de red
neuronal feedforward para reconocimiento de patrones con 35
neuronas ocultas.
net = feedforwardnet(35, 'trainscg'); % Crea la red neuronal con
35 neuronas ocultas.
net = configure(net,alphabet2,target2); % Obtiene los parámetros
de configuración.
net.trainParam.epochs = 1000; % Periodos.
net.trainParam.goal = 0; % Error de iteración.
net = train(net,alphabet2,target2); % Entrena la red con ruido.

y = net(alphabet); % obtiene la salida de la red para "alphabet".
errores = gsubtract(target,y); Resta "target" menos "y".
performance = perform(net,y,target) % Obtiene un error global.
```

### Prprob.m

```
function [alphabet,target] = prprob()
%PRPROB (Pattern Recognition PROBLEM)
%
% [ALHABET,TARGETS] = PRPROB()
% Devuelve:
%   ALPHABET - matriz 35x35 de elementos mapas de bit 5x7 para
%             cada letra y numero.
%   TARGETS - vectores objetivos de 35x35.

% Modificado por Carlos de la Cruz.
% Agregue los patrones de los números del 0 al 9.

letterA = [0 0 1 0 0 ...
           0 1 0 1 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterB = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ]';

letterC = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterD = [1 1 1 1 0 ...
```

```

        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 1 1 1 0 ]';

letterE = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

letterF = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ]';

letterG = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 1 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterH = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterI = [0 1 1 1 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 1 1 1 0 ]';

letterJ = [1 1 1 1 1 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           1 0 1 0 0 ...
           0 1 0 0 0 ]';

letterK = [1 0 0 0 1 ...
           1 0 0 1 0 ...
    
```

```
1 0 1 0 0 ...
1 1 0 0 0 ...
1 0 1 0 0 ...
1 0 0 1 0 ...
1 0 0 0 1 ]';

letterL = [1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 1 1 1 1 ]';

letterM = [1 0 0 0 1 ...
1 1 0 1 1 ...
1 0 1 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ]';

letterN = [1 0 0 0 1 ...
1 1 0 0 1 ...
1 1 0 0 1 ...
1 0 1 0 1 ...
1 0 0 1 1 ...
1 0 0 1 1 ...
1 0 0 0 1 ]';

letterO = [0 1 1 1 0 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
0 1 1 1 0 ]';

letterP = [1 1 1 1 0 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 1 1 1 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 0 0 0 0 ]';

letterQ = [0 1 1 1 0 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 1 0 1 ...
1 0 0 1 0 ...
0 1 1 0 1 ]';

letterR = [1 1 1 1 0 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
```

```
1 1 1 1 0 ...
1 0 1 0 0 ...
1 0 0 1 0 ...
1 0 0 0 1 ]';

letterS = [0 1 1 1 0 ...
1 0 0 0 1 ...
0 1 0 0 0 ...
0 0 1 0 0 ...
0 0 0 1 0 ...
1 0 0 0 1 ...
0 1 1 1 0 ]';

letterT = [1 1 1 1 1 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ]';

letterU = [1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
0 1 1 1 0 ]';

letterV = [1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
0 1 0 1 0 ...
0 0 1 0 0 ]';

letterW = [1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 0 0 1 ...
1 0 1 0 1 ...
1 1 0 1 1 ...
1 0 0 0 1 ]';

letterX = [1 0 0 0 1 ...
1 0 0 0 1 ...
0 1 0 1 0 ...
0 0 1 0 0 ...
0 1 0 1 0 ...
1 0 0 0 1 ...
1 0 0 0 1 ]';

letterY = [1 0 0 0 1 ...
1 0 0 0 1 ...
0 1 0 1 0 ...
0 0 1 0 0 ...
```

```
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ]';

letterZ = [1 1 1 1 1 ...
0 0 0 0 1 ...
0 0 0 1 0 ...
0 0 1 0 0 ...
0 1 0 0 0 ...
1 0 0 0 0 ...
1 1 1 1 1 ]';

number1 = [0 0 1 0 0 ...
0 1 1 0 0 ...
1 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ...
0 0 1 0 0 ]';

number2 = [0 1 1 1 0 ...
1 0 0 0 1 ...
0 0 0 0 1 ...
0 0 0 1 0 ...
0 0 1 0 0 ...
0 1 0 0 0 ...
1 1 1 1 1 ]';

number3 = [0 1 1 1 0 ...
1 0 0 0 1 ...
0 0 0 0 1 ...
0 0 1 1 1 ...
0 0 0 0 1 ...
1 0 0 0 1 ...
0 1 1 1 0 ]';

number4 = [0 0 0 1 0 ...
0 0 1 1 0 ...
0 1 0 1 0 ...
1 0 0 1 0 ...
1 1 1 1 1 ...
0 0 0 1 0 ...
0 0 0 1 0 ]';

number5 = [1 1 1 1 1 ...
1 0 0 0 0 ...
1 0 0 0 0 ...
1 1 1 1 0 ...
0 0 0 0 1 ...
0 0 0 0 1 ...
1 1 1 1 0 ]';

number6 = [0 0 1 1 1 ...
0 1 0 0 0 ...
1 0 0 0 0 ...
1 1 1 1 0 ...
1 0 0 0 1 ...
```

```

        1 0 0 0 1 ...
        0 1 1 1 0 ]';

number7 = [1 1 1 1 1 ...
           0 0 0 0 1 ...
           0 0 0 1 0 ...
           0 0 0 1 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ]';

number8 = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

number9 = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 1 1 1 ...
           0 0 0 0 1 ...
           0 0 0 1 0 ...
           1 1 1 0 0 ]';

number0 = [0 0 1 0 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 0 1 0 ...
           0 0 1 0 0 ]';

alphabet =
[letterA, letterB, letterC, letterD, letterE, letterF, letterG, letterH, .
..
letterI, letterJ, letterK, letterL, letterM, letterN, letterO, letterP, . .
.
letterQ, letterR, letterS, letterT, letterU, letterV, letterW, letterX, . .
.
letterY, letterZ, number1, number2, number3, number4, number5, number6, . .
.
        number7, number8, number9, number0]; % alphabet tiene 36
columnas. Es una matriz de 35 x 36

targets = eye(35)+ones(35,1); % Crea matriz identidad de 35x36
    
```

## ajustarAlTamanoDeLaImagen.m

```
function imagenRecortada = ajustarAlTamanoDeLaImagen(imagen)
% imagen debe tener letras negras (ceros) en fondo blanco (unos)
% El algoritmo FUNCIONA CON LETRAS NEGRAS SOLAMENTE!!!!

% Cambiando a letras negras
imagen = ~imagen; % Con operador binario NOT (~) letras son negras
(ceros) y fondo es blanco (unos)

% Encontrar los límites de la imagen
[y2temp x2temp] = size(imagen);
x1=1;
y1=1;
x2=x2temp;
y2=y2temp;

% Encontrar a la izquierda espacios en blanco
cntB=1;
while (sum(imagen(:,cntB))==y2temp) %mientras toda la línea este
vacía
    x1=x1+1;
    cntB=cntB+1;
end

% Encontrar arriba espacios en blanco
cntB=1;
while (sum(imagen(cntB,:))==x2temp) %mientras toda la línea este
vacía
    y1=y1+1;
    cntB=cntB+1;
end

% Encontrar a la derecha espacios en blanco
cntB=x2temp;
while (sum(imagen(:,cntB))==y2temp) %mientras toda la línea este
vacía
    x2=x2-1;
    cntB=cntB-1;
end

% Encontrar abajo espacios en blanco
cntB=y2temp;
while (sum(imagen(cntB,:))==x2temp) %mientras toda la línea este
vacía
    y2=y2-1;
    cntB=cntB-1;
end

% Recortar la imagen hasta los bordes encontrados
imagenRecortada=imcrop(imagen,[x1,y1,(x2-x1),(y2-y1)]); % imcrop
recorta la imagen
imagenRecortada = ~imagenRecortada; %Vuelve a letras blancas con
fondo negro como al inicio
```

**obtenerMatrizDeCaracteristicas.m**

```
function letra = obtenerMatrizDeCaracteristicas(imagen)
% Esta función toma una imagen binaria y la cambia a esta para una
matriz de 5 x 7 que es representado como un simple vector.

imagen_7050=imresize(imagen,[70,50]); % Redimensiona la imagen
for cnt=1:7
    for cnt2=1:5
        %%Se encuentra el porcentaje de cada caja (100 pixeles por
        100 pixeles) de la imagen que está vacío ya que negro
        tiene un valor de cero por lo tanto no sumara a área
        ocupada
        porcentajeVacio=sum(imagen_7050((cnt*10-
        9:cnt*10),(cnt2*10-9:cnt2*10)));
        %%Se asigna el porcentaje a una posición del vector que
        representa la letra.
        letra((cnt-1)*5+cnt2)=sum(porcentajeVacio);
    end
end
letra=((100-letra)/100); %% Se dividen todos los complementos de
los valores de la matriz para cien para tener valores entre 0 y 1
que representan el área ocupada.
letra=letra'; %% transpuesta de una matriz
```

**Código de las funciones de ICR2****salidaAtxt.m**

```
function salidaAtxt(net, img)

palabra = [];
resto=img;
fid = fopen('text.txt', 'wt'); % Abre text.txt como archivo para
escritura (write)
while 1
    [lineaUnoImg resto] = obtenerLineas(resto); % Separa texto en
línea uno y demás
    lineaUnoTexto = obtenerLetras(lineaUnoImg, net); % Las letras
ingresadas a obtenerLetras deben ser blancas con fondo negro
    palabra = [palabra, lineaUnoTexto]; % Concatena caracteres
    fprintf(fid,'%s\n',palabra); % Escribe 'palabra' en archivo de
texto (mayúsculas)
    palabra=[]; % Limpia la variable 'palabra'
    if isempty(resto) % Si variable 'resto' se quedó vacía en
función 'obtenerLineas' entonces termina el while
        break
    end
end
fclose(fid); % Cierra la escritura en 'text.txt'
winopen('text.txt'); % Abre el archivo'text.txt'
```

## obtenerLineas.m

```
function [fl re]=obtenerLineas(im_texto)
% Divide texto en líneas
% im_texto->input image; fl->first line; re->remain line
im_texto=ajustarAlTamanoDeLaImagen(im_texto);
num_filas=size(im_texto,1); % numero filas contadas en pixeles
for s=1:num_filas
    if sum(im_texto(s,:))>0 % suma componentes de las filas
        nm=im_texto(1:s-1, :); % Primera línea de la matriz
        rm=im_texto(s:end, :); % Matriz de línea restante
        fl=ajustarAlTamanoDeLaImagen(nm);
        re=ajustarAlTamanoDeLaImagen(rm);
        break
    else
        fl=im_texto; % Solo una línea.
        re=[];
    end
end
end
```

## obtenerLetras.m

```
function palabra = obtenerLetras(imgn, net2)
%*-*-*-*Calculando componentes conectados*-*-*-*
% Etiqueta y cuenta componentes conectados
% bwlabel pone índices a los objetos de la imagen binaria.
%imgn = ~imgn; % Cambia a letras blancas con fondo negro
[L Ne] = bwlabel(imgn); % L matriz tamaño igual a imgn con
elementos los índices. Ne es número total de índices en L.
palabra = [];
for n=1:Ne
    [r,c] = find(L==n); % Encuentra índice
    n1=imgn(min(r):max(r),min(c):max(c)); % Extract letter
    imagenRecortada = ajustarAlTamanoDeLaImagen(n1); %
Recorta espacios blancos o vacíos. FUNCIONA CON LETRAS NEGRAS
SOLAMENTE.
    imagenRecortada = ~imagenRecortada; %Cambia a letras
negras con fondo blanco
    vLetra = obtenerMatrizDeCaracteristicas(imagenRecortada);
% Redimensiona a una matriz de 5 x 7 representado como vector
    letra = leerLetra(vLetra, net2); % De vector a texto
    palabra = [palabra, letra]; % Concatena los caracteres
end
%*-*-*-*FIN Calculando componentes conectados*-*-*-*
```

## leerLetra.m

```
function letra=leerLetra(vLetra, net2 )
% Busca en la red neuronal, el vector target asociado al vector
vLetra ingresado y devuelve un carácter tipo cadena conteniendo la
letra.

% SOLO RECONOCE MAYUSCULAS!

yn = net2(vLetra); % La red neuronal devuelve el vector target
asociado al vector vLetra
yyn = compet(yn); % "Compet" devuelve un vector igual tamaño con
uno en la posición del elemento de máximo valor absoluto en el
vector y el resto ceros.
answer = find(yyn == 1); % "find" devuelve la posición del
elemento 1 en el vector "yyn"

if answer==1
    letra='A';
elseif answer==2
    letra='B';
elseif answer==3
    letra='C';
elseif answer==4
    letra='D';
elseif answer==5
    letra='E';
elseif answer==6
    letra='F';
elseif answer==7
    letra='G';
elseif answer==8
    letra='H';
elseif answer==9
    letra='I';
elseif answer==10
    letra='J';
elseif answer==11
    letra='K';
elseif answer==12
    letra='L';
elseif answer==13
    letra='M';
elseif answer==14
    letra='N';
elseif answer==15
    letra='O';
elseif answer==16
    letra='P';
elseif answer==17
    letra='Q';
elseif answer==18
    letra='R';
elseif answer==19
    letra='S';
elseif answer==20
    letra='T';
```

```
elseif answer==21
    letra='U';
elseif answer==22
    letra='V';
elseif answer==23
    letra='W';
elseif answer==24
    letra='X';
elseif answer==25
    letra='Y';
elseif answer==26
    letra='Z';
    *-*-*-*-*
elseif answer==27
    letra='1';
elseif answer==28
    letra='2';
elseif answer==29
    letra='3';
elseif answer==30
    letra='4';
elseif answer==31
    letra='5';
elseif answer==32
    letra='6';
elseif answer==33
    letra='7';
elseif answer==34
    letra='8';
elseif answer==35
    letra='9';
else
    letra='0';
end
```

# ANEXO B

## Anexo B: Mini Guía del MATLAB

### Nociones básicas

#### Matrices

> A= [1 2 3; 5 3 8; 8 3 8]      Crea una matriz con tres vectores fila.

A =

```
1  2  3
5  3  8
8  3  8
```

>> A(1,:)      Muestra la fila 1.

ans =

```
1  2  3
```

>> A(:,1)      Muestra la columna 1.

ans =

```
1
5
8
```

>>

**Cambiar valores de la primera columna en una matriz existente.**

A =

```
1  2  4
```

```
4 5 6
7 2 5
```

>> A(:,1)=[3;4;3]      Cambia la columna 1 por el vector columna [3;4;3].

A =

```
3 2 4
4 5 6
3 2 5
```

>> B=A(:,1:2)      Crea una matriz B las columnas 1 hasta 2 de la matriz A.

B =

```
1 2
5 3
8 3
```

Unir dos matrices

>> A=[1;2;3];

>> B=[3;4;5];

>> C=[A B]      Concatena dos vectores columna para crear matriz C.

C =

```
1 3
2 4
3 5
```

>> A=(1:1:3)      Crea un vector fila de 3 elementos. Va de 1 a 3, aumentando de 1 en 1 (el termino central).

A =

```
1 2 3
```

>> A=ones(1,3)      Crea un vector fila de 3 elementos unos, indicando sus dimensiones.

A =

1 1 1

>> B=zeros(1,3)      Crea un vector fila de 3 elementos nulos o ceros, indicando sus dimensiones

B =

0 0 0

>>

**clear** : Limpia la ventana de comandos.

**clc** : Limpia el "workspace".

**clf** : Cierra todas las ventana de ploteo de figuras (figure).

### Funciones utilizadas del MATLAB

#### randn()

randn(3,5); Crea matriz de 3 x 5 con elementos números reales aletatorios.

#### repmat()

B = repmat(eye(2),3,4) % crea una matriz de 3 x 4 de elementos eye(2). eye(2) es una matriz identidad de 2 x 2.

	1	2	3	4	
B =	1	0	1	0	}
	0	1	0	1	}
eye(2)	1	0	1	0	}
	0	1	0	1	}
	1	0	1	0	}
	0	1	0	1	}

} 1  
} 2  
} 3

2 x 3 = 6

2 x 4 = 8

B es una matriz de 6 x 8 elementos.

#### cell()

Crea un arreglo de celdas. Donde se almacena imágenes, matrices, vectores, listas y más. Para el trabajo se utilizó un arreglo de objetos, como imágenes.

Ejemplo 01: Crea un cell de 5 elementos en cada columna.

```
>> x={1 2 3 4 5}
      [1] [2] [3] [4] [5]
```

Ejemplo 02: Para crear el arreglo se usa "cell".

```
>> h=cell(2,2)
```

h =

```
 [] []
 [] []
```

Ejemplo 03: Crear una variable A de 2x2 y asignar un valor a cada elemento.

Para ver la variable se usa el nombre de la variable que es "A" y para asignarle valores también usa el nombre de la variable indicando el número de fila y columna (encerrado entre llaves) donde se realizará la asignación.

```
>> A{1,1}='A.jpg'
```

A =

```
'A.jpg'
```

```
>> A{1,2}=5
```

A =

```
'A.jpg' [5]
```

```
>> A{2,1}=[1 2 3]
```

A =

```
'A.jpg'    [5]
[1x3 double]  []
```

```
>> k=input('Ingresa el nombre de archivo con su extensión:\n','s');
```

Ingresa el nombre de archivo con su extensión:

D.jpg

```
>> A{2,2}=k
```

A =

```
'A.jpg'    [ 5 ]
[1x3 double]  'D.jpg'
```

```
>> A{2,2}
```

ans =

D.jpg

Ejemplo 04: Concatenar dos objetos cell

```
>> A=cell(1);
```

```
>> B=cell(1);
```

```
>> A=[A,B]    Concatena A y B y lo guarda en A.
```

A =

```
[] []
```

**struct()**

**struct simple:**

```
bola.masa = 10;
```

```
bola.posicion = [0, 0, 100];  
bola.velocidad = [0, 0, 0];  
>> bola
```

```
bola =
```

```
    masa: 10  
  posicion: [0 0 100]  
  velocidad: [0 0 0]
```

```
>> bola.posicion
```

```
ans =
```

```
    0    0  100
```

```
>> bola.posicion = ball.posicion + [0, 10, 0]
```

```
bola =
```

```
    masa: 10  
  posicion: [0 10 100]  
  velocidad: [0 0 0]
```

### **struct de struct:**

pelotas es un struct de 4 elementos struct, cada elemento en una columna, entonces, para entrar al primer elemento se usa: `pelotas{1,1}`, al segundo es `pelotas{1,2}` y así sucesivamente. Para entrar a los elementos del primer elemento struct de pelotas, es así:

```
pelotas(1,1).bola.posicion;  
pelotas(1,1).bola.posicion(1);      Si "posición" tiene más de un elemento.
```

### **size()**

Determina el tamaño de una variable.

```
>> A = [ 1 2 6 7 9; 3 4 2 9 6; 4 2 7 5 8]
```

A =

```
 1  2  6  7  9
 3  4  2  9  6
 4  2  7  5  8
```

```
>> size(A)
```

ans =

```
 3  5
```

```
>> size(A,1)
```

ans =

```
 3
```

```
>> size(A,2)
```

ans =

```
 5
```

size(A); % Devuelve el tamaño de la matriz.

size(A,1); % Devuelve el número de filas de la matriz.

size(A,2); % Devuelve el número de columnas de la matriz.

### plotchar()

plotchar(vectorD);



### subplot()

En un subplot(x,y,n) sus parámetros indican lo siguiente:

“x” el número de filas que se divide la ventana.

“y” el número de columnas que se divide la ventana

‘n’ el número de elemento a mostrar de la ventana dividida. Se cuenta de izquierda a derecha. Su valor máximo es “x\*y”.

Para mostrar una imagen y un “plotchar” de letra D. Ver figura N° B.01.

```
subplot(1,2,1); imshow(imagenD); % Imagen D
subplot(1,2,2); plotchar(vectorD); % Vector D
```

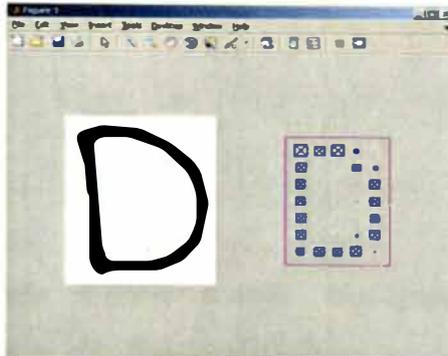


Figura N° B.01.- Ejemplo de “subplot” para mostrar una imagen y su vector

### **whos()**

Muestra información de la foto (orden de la matriz, tipo de dato de los valores).

### **Funciones de la herramienta de procesamiento de imágenes**

El “*Image Processing Toolbox*” realiza procesamiento de imágenes. Estas son las funciones utilizadas.

### **imread():**

Ejemplo 01: Con ruta relativa

```
imagenEnMatriz = imread('./imagenes/A.bmp');
```

Ejemplo 02:

Directamente escribiendo ruta de trabajo y nombre de la imagen conjuntamente como se muestra

```
I = imread('C:\Users\carlos\Desktop\Neuronas\reconocimiento
de caracteres\fotos\A.jpg'); %Lee la foto pero como matriz
de pixeles y la guarda en I.
```

Ejemplo 03:

Otra forma es configurar la carpeta de trabajo primero desde "current folder" y luego leer la foto sin anteponerle la ruta.

Mi carpeta de trabajo es C:\Users\carlos\Desktop\Neuronas\reconocimiento de caracteres\fotos

```
I = imread('A.jpg'); %Lee la foto pero como matriz de  
píxeles y la guarda en I.
```

### imshow()

Muestra imágenes desde como matriz indexada. Es muy usada en procesamiento de imágenes. Ver figura N° B.02.

```
>> imshow(imagenA);
```

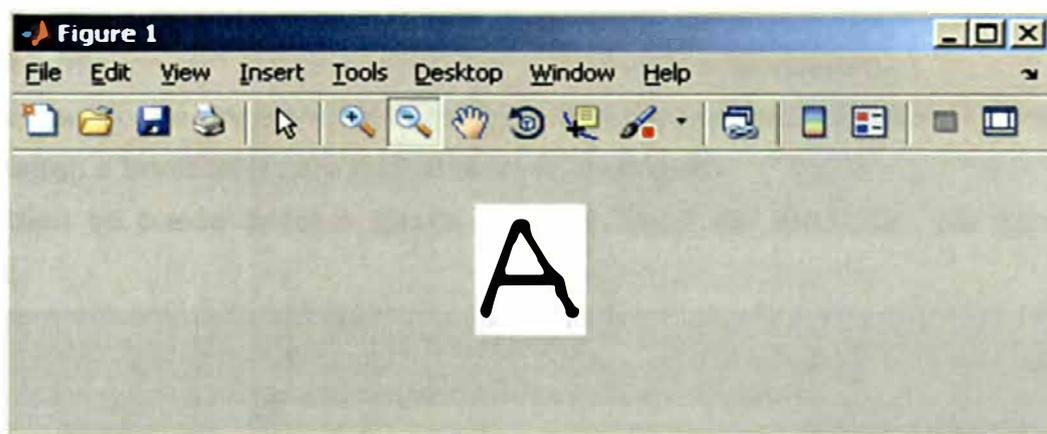


Figura N° B.02.- Se muestra una imagen usando la función "imshow"



La herramienta "Data cursor" se utiliza para ver la posición x, y del cursor en unidades píxeles. El origen (0,0) se encuentra siempre en la esquina izquierda superior de todas las imágenes.

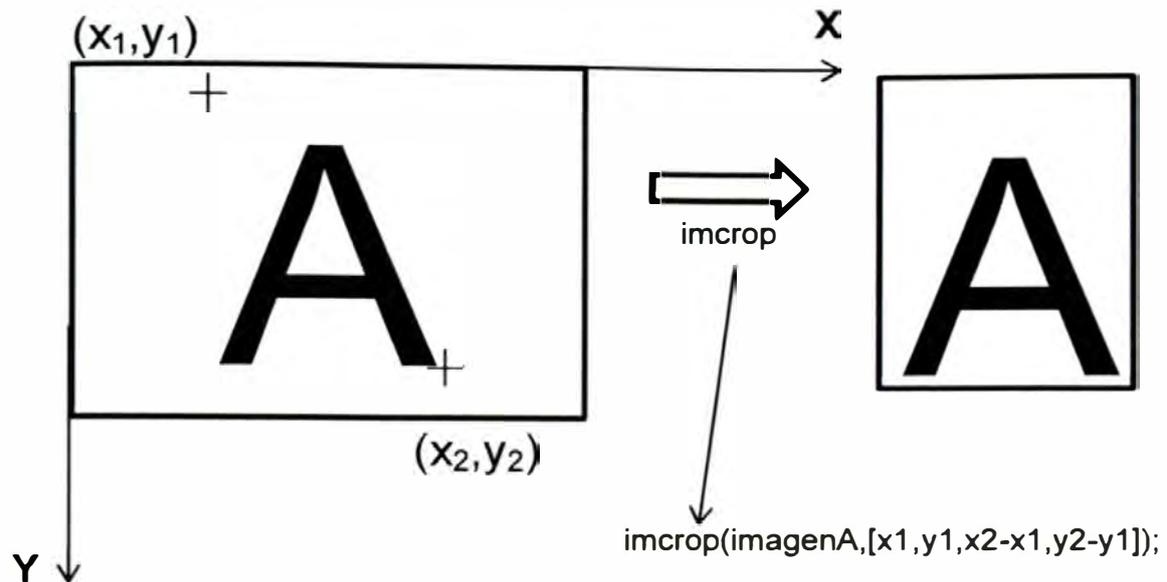


La herramienta "pan" permite mover la imagen. Es utilizada cuando se ha hecho un zoom de la imagen y se quiere ver las partes que no alcanza en el recuadro de visualización.

### imcrop()

En la imagen anterior "imagenA":

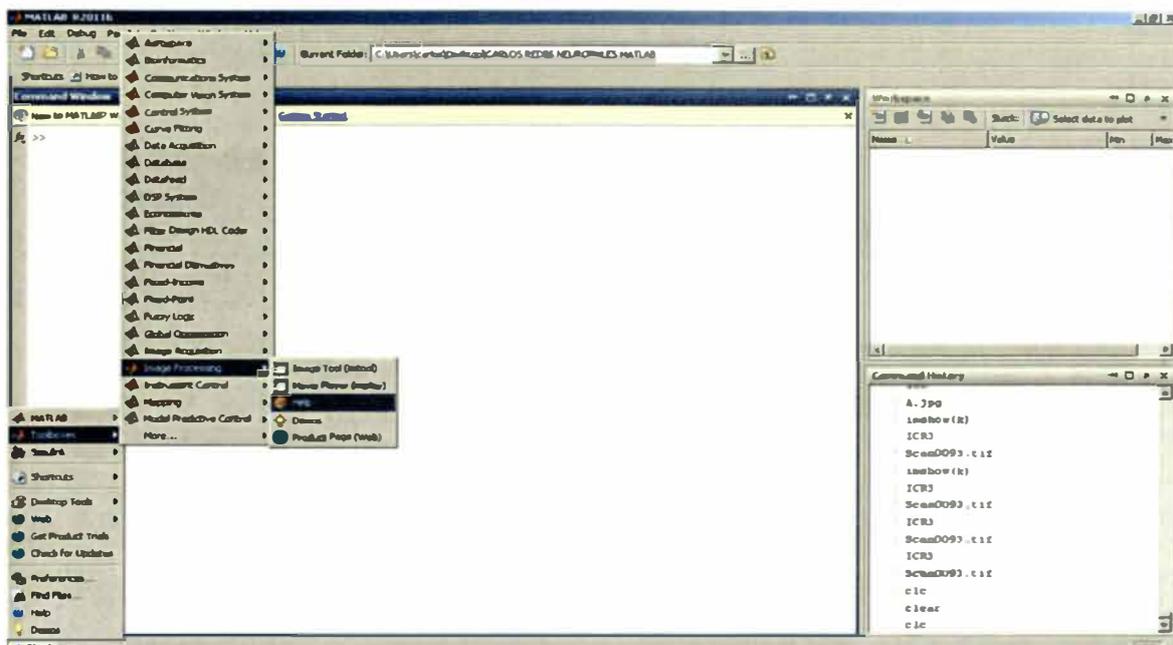
```
imcrop(imagenA,[x1,y1,x2-x1,y2-y1]); Recorta una imagen definiendo sus bordes
```



La información de esta función se encuentra en la siguiente ayuda del MATLAB: menu help > Product help > Image Processing Toolbox > Functions > Spatial Transformation and Image Registration > Spatial Transformation.

Donde se puede encontrar otras funciones como la función imrotate() para rotar la imagen e imresize() para redimensionar la imagen.

También se puede acceder desde el menú "start" del MATLAB. Ver figura N° B.03.



**Figura N° B.03.-** Acceso al menú "help" del "Image Processing Toolbox" desde el menú "start"

Para referencia adicional de las funciones de procesamiento de imágenes en

MATLAB consultar en internet el artículo "Visión por Computador utilizando MATLAB y el Toolbox de Procesamiento Digital de Imágenes" de CUEVAS JIMENEZ, Erik V. y ZALDIVAR NAVARRO, Daniel. UTPL Madrid. 2006.

### **bwlabel()**

Determina los componentes de una imagen. Ejemplo: Sea "imagenA" la imagen de la letra A, así:



Es una A, sin embargo la función bwlabel() determina dos componentes porque la letra no está bien cerrada. Las dos componentes son figuras como sigue:



```
[L Ne] = bwlabel(imagenA);
```

```
% L = imagenA
```

```
% Ne = 2 Número de componentes.
```

### **find()**

```
>> X = [3 2 0; -5 0 7; 0 0 1]
```

```
[r,c,v] = find(X) find devuelve los índices 11 21 12 23 y 33 de los  
elementos no cero de la matriz X. También los valores.
```

```
X =
```

```
3 2 0  
-5 0 7  
0 0 1
```

Los valores no cero (de arriba abajo) de la matriz X son:  
3, -5, 2, 7 y 1 de índices 11 21 12 23 33 respectivamente.

```
r =
```

Son los índices fila de 11 21 12 23 33

```
1  
2  
1  
2  
3
```

c = Son los índices columna de 11 21 12 23 33

1  
1  
2  
3  
3

v = Son los valores no cero de la matriz X.

3  
-5  
2  
7  
1

### **edge()**

Detecta contorno de las imágenes.

Solo funciona para matrices bidimensionales (<280x272 uint8>)

No funciona para matrices tridimensionales (<82x72x3 uint8>).

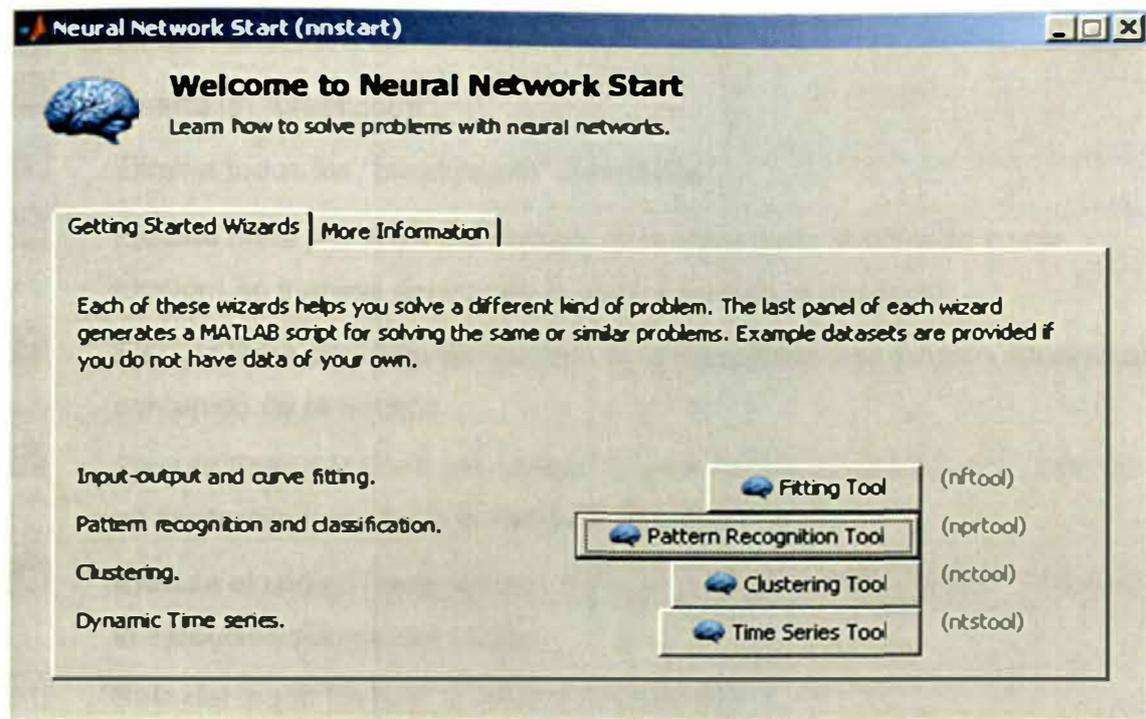
### **Funciones de la herramienta de redes neuronales**

feedforwardnet(). (Ver Redes Neuronales en MATLAB del capítulo II).

train(). (Ver Redes Neuronales en MATLAB del capítulo II).

**nnstart:** Significa "Neural Network Start" y es un asistente del MATLAB para crear script de redes neuronales. Se ejecuta escribiendo "nnstart" en la ventana de comandos del MATLAB, así:

```
>> nnstart
```



### Uso del modo “debug” del MATLAB

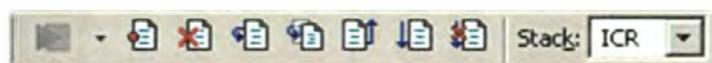
Con este modo se programa más rápido. En realidad se depuran errores más fácilmente al ver paso a paso el funcionamiento del código en el “script”. Depurar es identificar y corregir un error.

Elementos:

“Breakpoints”, son puntos de pausa de la ejecución del “script”, si no se coloca uno, el código del script se ejecutará normalmente, es decir, hasta el final. Gráficamente es un círculo rojo (●). Ejemplo: Se ha colocado un “breakpoint” en la línea número 20 y cuando se ejecuta el script, la ejecución del programa se detiene en este punto.

20 ● → cont = 0;

En la figura N° B.04 se muestra la barra de herramientas del modo “debug”.



**Figura N° B.04.-** Barra de herramientas del modo “debug” del MATLAB

-  Guarda y ejecuta un “script”.
-  Inserta un “breakpoint”.
-  Elimina todos los “breakpoints” insertados.
-  Ejecuta línea por línea del código. Si la línea tiene la llamada a una función, no ingresa dentro; se lo salta y ejecuta la siguiente.
-  Ejecuta línea por línea del código. Si la línea tiene una función, ingresa al contenido de la función.
-  Ejecuta línea por línea del código. Si está dentro de una función, sale de su contenido y vuelve a la llamada de la función.
-  Ejecuta el código hasta el final, o hasta el próximo “breakpoint”. Reanuda la ejecución normal del código.
-  Sale del modo “debug” y detiene la ejecución.

## **Configuración y Uso del MATLAB**

Configuración:

Restaurar la distribución de ventanas iniciales del MATLAB. Esto es necesario cuando se cierra una o varias ventanas del MATLAB por error. Se ejecuta así:

Menu Desktop > Desktop Layout > Default

Uso:

Mostrar los comandos escritos anteriormente de tal manera que no es necesario volver a escribir el mismo comando, basta con presionar flecha arriba varias veces hasta buscar el comando deseado. Esta búsqueda de comandos recorre el historial de comandos escritos desde la instalación del MATLAB. Se ejecuta así: Flecha arriba y flecha abajo del teclado

Workspace:



-  Crear nueva variable (ctrl + N).
-  Edita el contenido (doble clic).
-  Importa una variable existente. Es igual al comando “load”.
-  Guardar variables del “workspace” (ctrl + S).
-  Elimina variables del “workspace” (botón suprimir o “DEL”).

Para guardar:

Seleccionar la variable a guardar, clic derecho sobre la selección y “save as...”. Para guardar variables no consecutivas puedes dejar presionado “ctrl” mientras seleccionas.

Si se requiere guardar todas las variables se usa el botón 

Guardar las variables tienen mucho uso porque se las puede usar posteriormente en cualquier script usando el comando “load”. Ejemplo.

Guardo la variable  $x=[1\ 2\ 3\ 4\ 5]$  en un archivo datos.mat

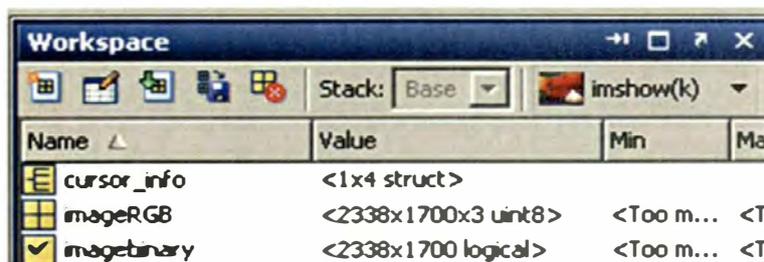
```
>> load datos.mat
```

```
>> load datos
```

También se puede obtener datos desde un archivo txt, así:

```
>> load datos.txt
```

En los archivos .mat se pueden guardar matrices, imágenes, redes neuronales, estructuras y más.



# ANEXO C

## Anexo C: Ejemplo de reconocimiento de patrones del MATLAB

El script original se llama `appcr1.m` y se encuentra en la ruta:

`C:\Program Files\MATLAB\R2011b\toolbox\nnet\indemos\appcr1.m`

A continuación se presenta el ejemplo comentado y traducido. Además explicado en más detalle en algunas partes.

### Reconocimiento de carácter

Esta demostración ilustra como entrenar una red neuronal para realizar un reconocimiento de carácter simple.

### Contenido

Definiendo el problema.

Creando la primera red neuronal.

Entrenando la primera red neuronal.

Entrenando la segunda red neuronal.

Probando ambas redes neuronales.

### Definiendo el problema

El script `prprob.m` define una matriz `X` con 26 columnas, una para cada letra del alfabeto. Cada columna tiene 35 valores que pueden ser 1 ó 0. Cada columna de 35 valores define un "bitmap" de 5x7 de una letra.

La matriz `T` es una matriz identidad de 26x26 que relaciona los 26 vectores de entrada a las 26 clases.

```
[X,T] = prprob; % Devuelve X los alphabet y T los
targets
```

La función `prprob()` en el archivo script "prprob" es:

```
function [alphabet,targets] = prprob()
%PRPROB Definición del problema de reconocimiento de carácter.
%
%   [ALHABET,TARGETS] = PRPROB()
%   Devuelve:
%   ALPHABET - Matriz mapa de "bit" de 35x26 para cada letra.
%   TARGETS - Vectores objetivo de 26x26.
```

```

letterA = [0 0 1 0 0 ...
           0 1 0 1 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterB = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ]';

letterC = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterD = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ]';

letterE = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

letterF = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ]';

letterG = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 1 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterH = [1 0 0 0 1 ...

```

```

        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 1 1 1 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ]';

letterI = [0 1 1 1 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 1 1 1 0 ]';

letterJ = [1 1 1 1 1 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           1 0 1 0 0 ...
           0 1 0 0 0 ]';

letterK = [1 0 0 0 1 ...
           1 0 0 1 0 ...
           1 0 1 0 0 ...
           1 1 0 0 0 ...
           1 0 1 0 0 ...
           1 0 0 1 0 ...
           1 0 0 0 1 ]';

letterL = [1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

letterM = [1 0 0 0 1 ...
           1 1 0 1 1 ...
           1 0 1 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterN = [1 0 0 0 1 ...
           1 1 0 0 1 ...
           1 1 0 0 1 ...
           1 0 1 0 1 ...
           1 0 0 1 1 ...
           1 0 0 1 1 ...
           1 0 0 0 1 ]';

letterO = [0 1 1 1 0 ...
           1 0 0 0 1 ...
    
```

```

        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        1 0 0 0 1 ...
        0 1 1 1 0 ]';

letterP = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ]';

letterQ = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 1 0 1 ...
           1 0 0 1 0 ...
           0 1 1 0 1 ]';

letterR = [1 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 0 ...
           1 0 1 0 0 ...
           1 0 0 1 0 ...
           1 0 0 0 1 ]';

letterS = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           0 1 0 0 0 ...
           0 0 1 0 0 ...
           0 0 0 1 0 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterT = [1 1 1 1 1 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ]';

letterU = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

letterV = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
    
```

```

        1 0 0 0 1 ...
        1 0 0 0 1 ...
        0 1 0 1 0 ...
        0 0 1 0 0 ]';

letterW = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 1 0 1 ...
           1 1 0 1 1 ...
           1 0 0 0 1 ]';

letterX = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 0 1 0 ...
           0 0 1 0 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

letterY = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 0 1 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ...
           0 0 1 0 0 ]';

letterZ = [1 1 1 1 1 ...
           0 0 0 0 1 ...
           0 0 0 1 0 ...
           0 0 1 0 0 ...
           0 1 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

alphabet =
[letterA, letterB, letterC, letterD, letterE, letterF, letterG, letterH, .
..
letterI, letterJ, letterK, letterL, letterM, letterN, letterO, letterP, ..
.
letterQ, letterR, letterS, letterT, letterU, letterV, letterW, letterX, ..
.
    letterY, letterZ];

targets = eye(26);

```

El patrón de la letra A es:

```

letterA = [0 0 1 0 0 ...
           0 1 0 1 0 ...
           0 1 0 1 0 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...

```

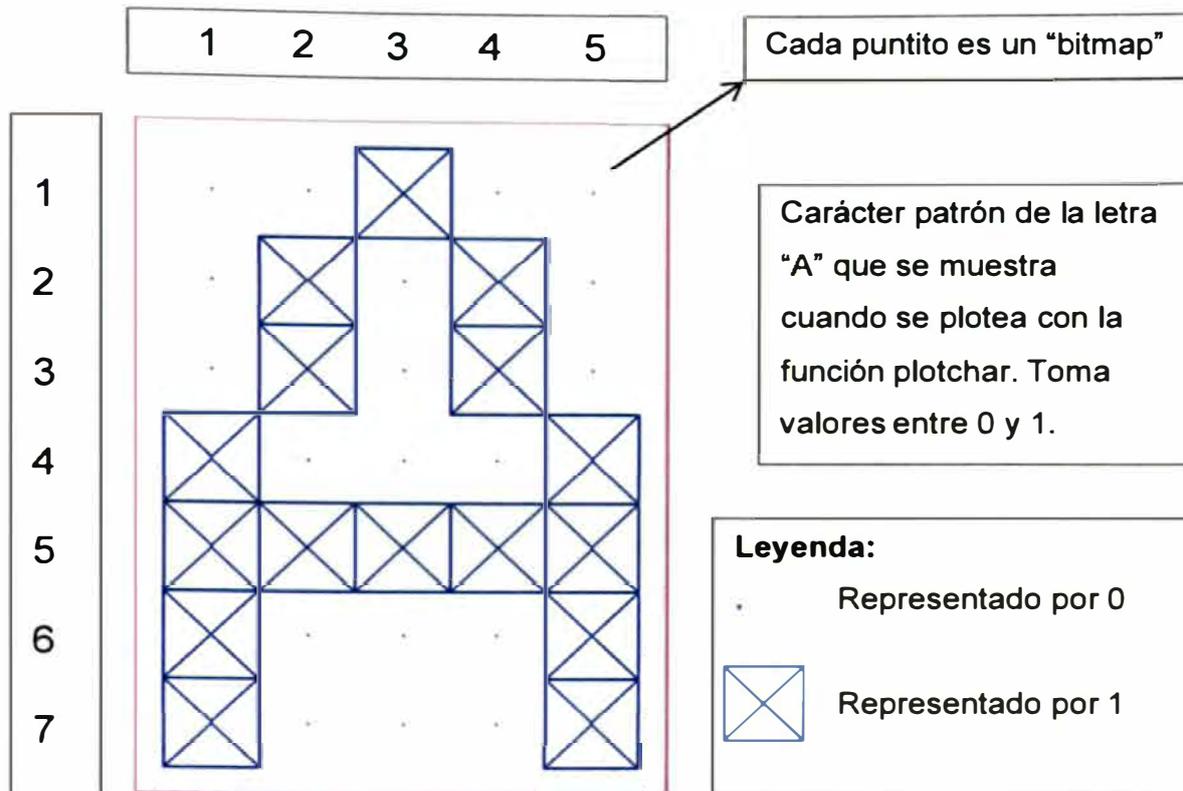
```
1 0 0 0 1 ...  
1 0 0 0 1 ]';
```

Desplegando por filas los números de la matriz letterA luego lo ordeno en columna y sería la primera columna de la matriz X, que al plotear muestra la letra A.

```
X = [ 0  
      0  
      1  
      0  
      0  
      0  
      1  
      0  
      1  
      0  
      0  
      1  
      0  
      1  
      0  
      0  
      0  
      1  
      1  
      1  
      1  
      1  
      1  
      1  
      1  
      1  
      0  
      0  
      0  
      1  
      1  
      0  
      0  
      0  
      1 ]
```

Aquí A, la primera letra, es ploteadada como un mapa de bit.

```
figure  
plotchar(X(:,1))
```



### Creando la Primera Red Neuronal

Para resolver este problema se usa una configuración de red neuronal para reconocer 25 neuronas ocultas.

Desde que la red neuronal es inicializada con pesos aleatorios, los resultados después del entrenamiento varían enormemente cada vez que el ejemplo de prueba es ejecutado. Para evitar esta aleatoriedad, la semilla aleatoria es configurada para reproducir los mismos resultados cada vez. Esto no es necesario para todas las aplicaciones.

```
setdemorandstream(pi); %define semilla y aleatoriedad
cte.

net1 = feedforwardnet(25); %patron reconoc. Con 25
neuronas.
view(net1)% muestra el diagrama de abajo.
```

En la figura N° C.01 se muestra el modelo de red neuronal artificial utilizada.

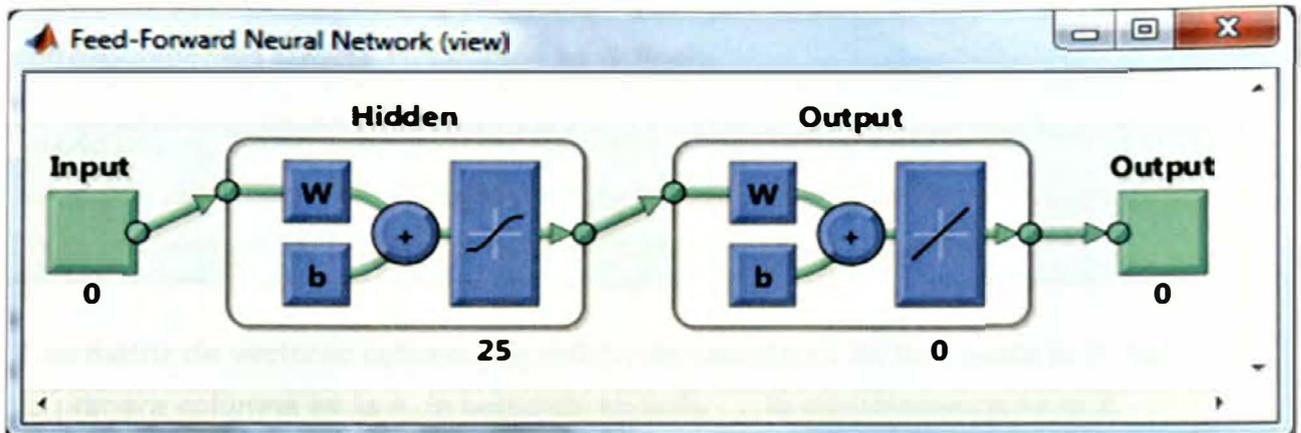


Figura N° C.01.- Modelo de red neuronal artificial

### Entrenando la Primera Red Neuronal

La función de entrenamiento divide la data en configuraciones de entrenamiento, validación y prueba. La configuración de entrenamiento es usada para actualizar la red. La configuración de validación es usada para detener la red antes de sobreajustar los datos de entrenamiento, además preserva buena generalización. La configuración de prueba actúa como una medida completamente independiente de lo bien que la red puede esperar para hacer nuevos ejemplos.

El entrenamiento se detiene cuando la red ya no es posible que mejore en las sesiones de entrenamiento o validación.

```
net1.divideFcn = '';  
net1 = train(net1,X,T); %divide data en entrenamiento,  
validación y test
```

### Entrenando la Segunda Red Neuronal

Sería conveniente que la red no solo reconozca formas de letras perfectamente, sino también versiones con ruido de las letras. Por tanto se tratará de entrenar una segunda red de datos con ruido y se comparará su habilidad para generalizar con la primera red.

Aquí son creadas 30 copias con ruido de cada letra  $X_n$ . Los valores están limitados por el mínimo y máximo que se sitúan entre 0 y 1. Los correspondientes targets  $T_n$  también se definen.

```
numNoise = 30;
Xn = min(max repmat(X,1,numNoise)+randn(35,26*numNoise)*0.2,0),1);
Tn = repmat(T,1,numNoise); %Tn tiene 780 columnas
```

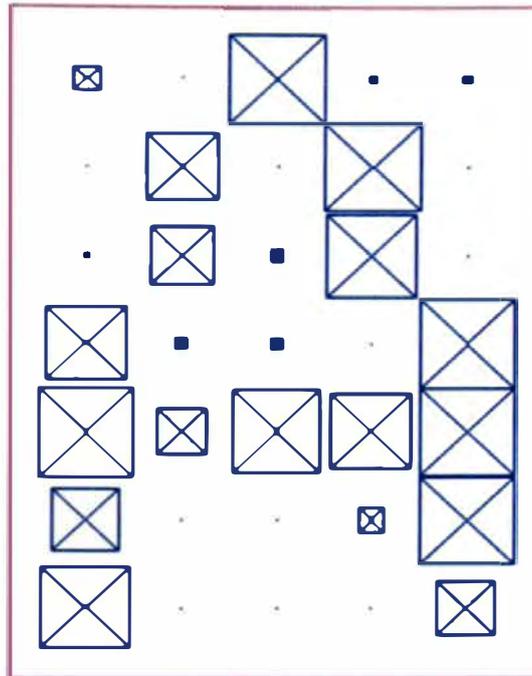
X es matriz de vectores columna de patrón de caracteres de la A hasta la Z. Así: La primera columna es la A, la segunda es la B, ..., la vigésimosexta es la Z.

X =	0 1 0 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1	26x26
	0 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1	
	1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1	
	0 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1	
	0 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1	
	0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0	
	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	
	1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	
	0 1 1 1 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1	
	0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0	
	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0	
	0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0	
	1 0 1 1 1	
	0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0	
	1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0	
	0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0	
	0 1 0 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 1 1	
	0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0	
	1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 0 0 0	
	1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0	
	1 0 1 0 1	
	1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0	
	1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0	
	1 1 0 1 0 0 1 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0 0 0	
	1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1	

T es una matriz identidad de 26 filas por 26 columnas. Así:

1 0
0 1 0





Aquí la segunda red es creada y entrenada.

```
net2 = feedforwardnet(25); %red segunda creada
net2 = train(net2,Xn,Tn); %red segunda entrenada
```

A la letra "A" corresponde el target  $[1;0]$  de 26 elementos.

A la letra "B" corresponde el target  $[0;1;0]$  de 26 elementos.

...

A la letra "Z" corresponde el target  $[0;1]$  de 26 elementos.

Entonces a todos los patrones de A con ruido que son  $X_n(:,1)$ ,  $X_n(:,27)$ ,  $X_n(:,53)$ , ...,  $X_n(:,780)$ , les corresponde los vectores  $T_n(:,1)$ ,  $T_n(:,27)$ ,  $T_n(:,53)$ ,  $T_n(:,79)$ ... ,  $T_n(:,780)$ , respectivamente. Todos estos vectores son iguales al target  $[1;0]$  de 26 elementos.

## Probando Ambas Redes Neuronales

```
noiseLevels = 0:.05:1;
numLevels = length(noiseLevels);
percError1 = zeros(1,numLevels);
percError2 = zeros(1,numLevels);
for i = 1:numLevels
    Xtest=min(max(repmat(X,1,numNoise)+randn(35,26*numNoise)*noiseLevels(i),0),1);
    Y1 = net1(Xtest);
    percError1(i) = sum(sum(abs(Tn-compet(Y1))))/(26*numNoise*2);
    Y2 = net2(Xtest);
    percError2(i) = sum(sum(abs(Tn-compet(Y2))))/(26*numNoise*2);
end

figure
plot(noiseLevels,percError1*100,'--
',noiseLevels,percError2*100);
title('Percentage of Recognition Errors');
xlabel('Noise Level');
ylabel('Errors');
legend('Network 1','Network 2','Location','NorthWest')
```

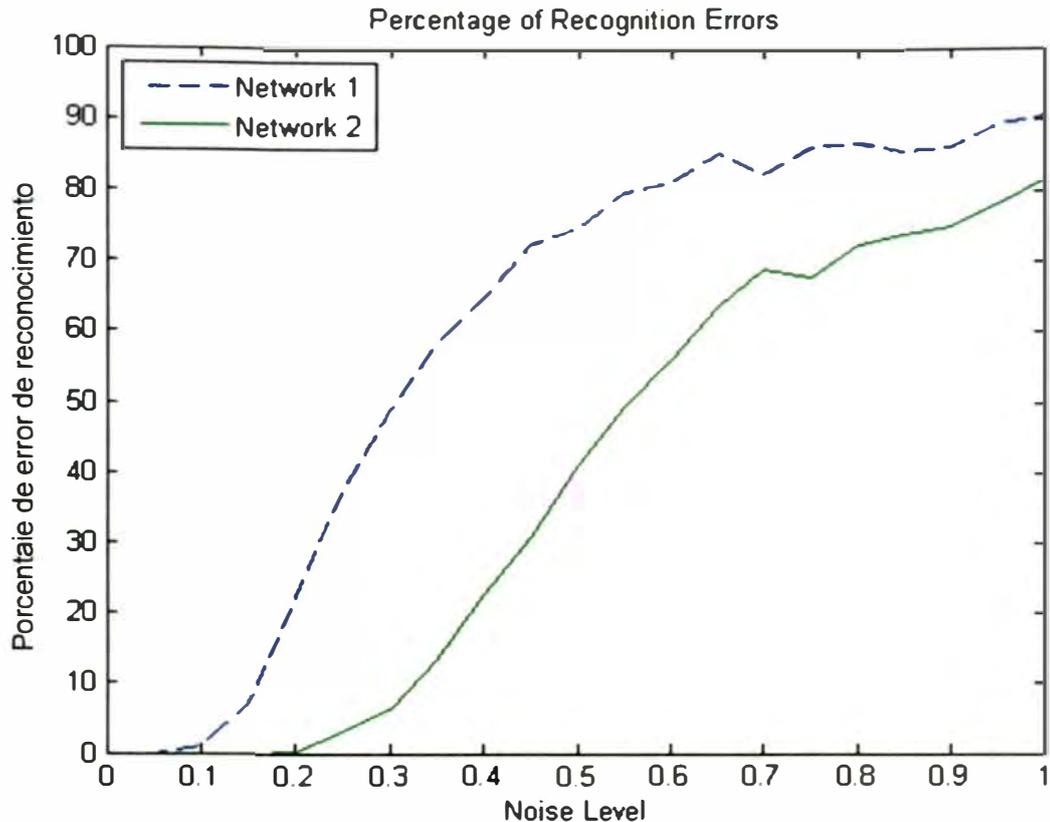


Figura N° C.02.- Gráfica de Nivel de ruido vs Porcentaje de error de reconocimiento del ejemplo "prprob".

De la figura N° C.02 se observa que la Red 1, entrenada sin ruido, tiene más errores debido al ruido de las letras ingresadas, en cambio la Red 2 fue entrenada con ruido. **Entonces se concluye que una red entrenada con ruido tiene menor error para reconocer caracteres, además 0.2 es el máximo nivel de ruido (numNoise) que proporciona un porcentaje de error de reconocimiento cero para una red entrenada con ruido.**

Luego el "noiseLevel" usado es 0.2. Esto justifica el 0.2 en la siguiente expresión:

```
alphabet2=min(max repmat(alphabet,1,numNoise)+randn(35,26*nu  
mNoise)*0.2,0),1)
```

La expresión anterior crea matemáticamente patrones distorsionados de caracteres para ser usados en el entrenamiento de la segunda red neuronal artificial.

## Cálculo del nivel de ruido

Para crear los patrones con ruido se debe calcular el nivel de ruido (Noise Level) en la siguiente expresión:

```
min(max(repmat(alphabet,1,numNoise)+randn(35,26*numNoise)*noiseLevel,0),1)
```

Para calcular el valor se observa el nivel de ruido (noiseLevel) máximo para un porcentaje de error cero de reconocimiento en la curva de la red entrenada con ruido de la gráfica nivel de ruido vs Porcentaje de reconocimiento. Esta gráfica se obtiene al ejecutar el "script" appcr1.m cambiando previamente el número de neuronas, número de ruidos (numNoise) y una función de entrenamiento. Entonces, probando para 35 neuronas ocultas, número de ruido de 60 y función de entrenamiento "trainscg" en los parámetros de entrada, se observa un nivel de ruido máximo (noiseLevel) de 0.22, tal como lo muestra la figura N° C.03.

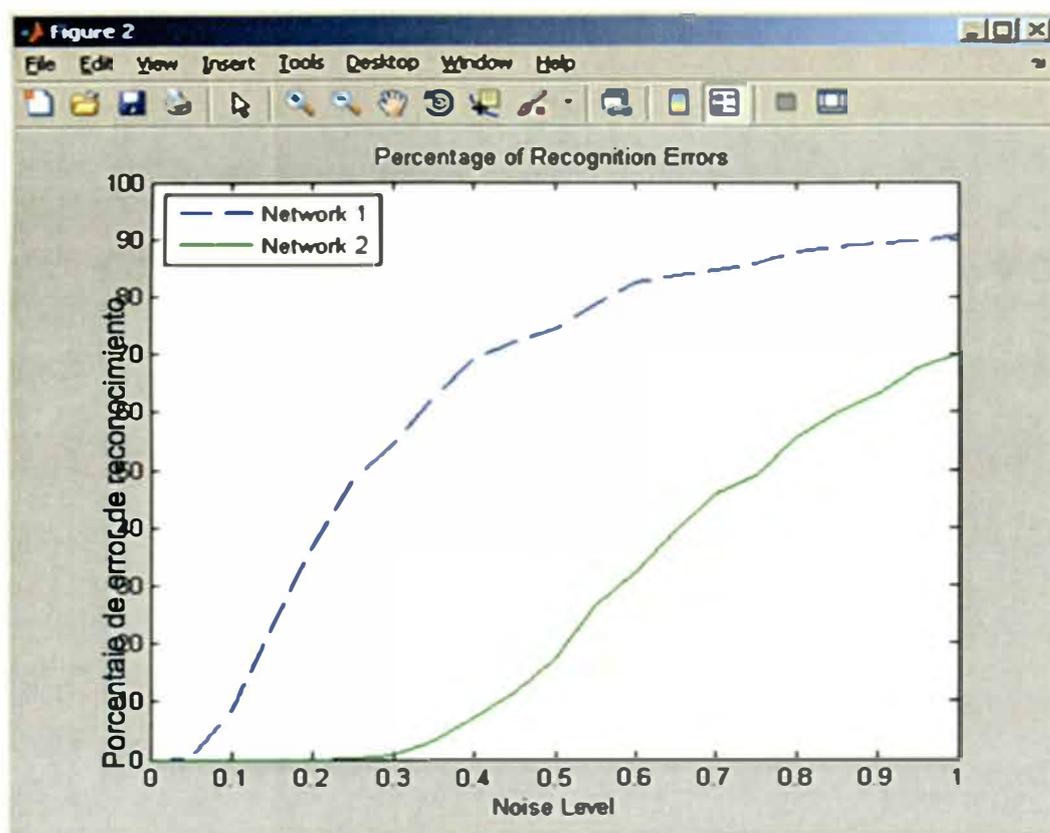


Figura N° C.03.- Nivel de ruido vs Porcentaje de error de reconocimiento.

Fuente: MATLAB.

Significa que los patrones de caracteres binarios se les puede distorsionar adicionando ruido máximo de 0.22 en el entrenamiento. Más ruido la red tendría dificultad de reconocer los patrones. Entonces para asegurarme elijo un nivel de ruido de 0.20. Finalmente, observar de la figura N° C.03 que el nivel de ruido esta entre 0 y 1.

# ANEXO D

## Anexo D: Glosario

**Neurona:** Es la unidad mínima de procesamiento de información de la Red Neuronal Artificial, que tiene la capacidad de aprender y almacenar datos de cualquier tipo, trabajando en forma interconectada con otras neuronas. Aprende mediante un entrenamiento basado en experiencias de prueba y error supervisadas.

**Peso:** Llamado peso sináptico, es la memoria de información de la neurona.

**Procesamiento de formularios:** Es un proceso mediante el cual se captura la información introducida en los campos de datos y se convierte en un formato digital.

**Ruido:** Es la distorsión del patrón original con puntos, pixelados, manchas, impurezas y pérdida de calidad, que deterioran el reconocimiento de letra o número en la imagen escaneada. En el anexo C se muestra un patrón con ruido de la letra A.

**Segmentación:** Es el recorte excedente de área sobrante que rodea al carácter.

**Script:** Es un archivo de texto que guarda sentencias de código en MATLAB. Las sentencias son la mínima expresión de código funcional que ejecuta comandos de MATLAB. Están separadas por el punto y coma.

**Software Framework:** Un conjunto reutilizable de bibliotecas o clases para un sistema de software (o subsistema).

### Palabras clave en el buscador google:

reconocimiento caracteres redes neuronales matlab

separar letras matlab

ocr with neural network

redes neuronales matlab

matlab ocr

icr

# ANEXO E

## **Anexo E: Fichas Catastrales Individuales escaneadas**

La ficha catastral usada para el reconocimiento con el programa se ha vuelto a rellenar nuevamente con un lapicero azul de tinta líquida, debido a la pérdida de nitidez de los caracteres de los campos la ficha catastral original cuando se transformaba la imagen escaneada a imagen binaria. La ficha para el reconocimiento se escanea en formato RGB, como archivo de imagen TIFF para convertir con el programa a formato monocromo o binario. Formato monocromo es una imagen de dos colores: blanco y negro. El reconocimiento de letras es de datos binarios en blanco y negro. La ficha individual escaneada para el reconocimiento se muestra en la figura N° E.01 y figura N° E.02.

Adicionalmente se presentan 5 fichas catastrales originales del Levantamiento Catastral del Centro Poblado Santa Cruz de la manzana 14 de los lotes 01 al 05. Se puede apreciar la falta de nitidez, borrones y aclarado de los caracteres, los cuales provocan mutilación de los caracteres por las zonas claras del lapicero en la imagen escaneada en formato binario. Estas no fueron utilizadas para el reconocimiento.

NÚMERO DE FICHA

**FICHA CATASTRAL URBANA INDIVIDUAL**

NÚMERO DE FICHAS POR LOTE

ESCUDO  
DISTRITAL,  
PROVINCIAL

LOGO  
ENTIDAD  
EJECUTORA

01 CÓDIGO ÚNICO CATASTRAL - CUC				02 CÓDIGO HOJA CATASTRAL							
03 CÓDIGO DE REFERENCIA CATASTRAL											
URBANO	PSV	DIST.	SECTOR	MANZANA	LOTE	EDIFICA	ENTRADA	PISO	UNIDAD	DC	
1	5	0	8	1	2	0	1	0	1	0	
04 Cód. CONTRIBUYENTE DE RENTAS				05 Código PREDIAL DE RENTAS				06 Código ACUMULADA A Código PREDIAL DE RENTAS			

**UBICACIÓN DEL PREDIO CATASTRAL**

07	CÓDIGO DE VÍA	08	TIPO DE VÍA	09	NOMBRE DE VÍA	10	TIPO DE PUERTA	11	N° MUNICIPAL	12	COND. PALMER	13	N° DE CERTIF. DE NUMERACIÓN		
			AV		CORONEL PORTILLO		P								
14 NOMBRE DE LA EDIFICACIÓN				15	TIPO DE EDIFICACIÓN	02	16	TIPO DE INTERIOR	02	17	N° INTERIOR				
TIPO DE VÍA				TIPO DE PUERTA		CONDICIÓN DE NUMERACIÓN		TIPO DE EDIFICACIÓN		TIPO DE INTERIOR					
18 Código HL				19 NOMBRE DE LA HABILITACIÓN URBANA				20 ZONA/SECTOR/ETAPA		21	MANZANA	22	LOTE	23	SUB-LOTE

**IDENTIFICACIÓN DEL TITULAR CATASTRAL**

24	TIPO DE TITULAR	1	1 = PERSONA NATURAL 2 = PERSONA JURÍDICA	25	ESTADO CIVIL	1	01 SOLTERO (A)	02 CASADO (A)	03 DIVORCIADO (A)	04 VIUDO (A)	05 CONVIVIENTE		
26	TIPO DOC. IDENTIDAD	2	27 N° DOC.	4 2 2 6 2 0 6 0	28	NOMBRES						JOSE BENITO	
29 APELLIDO PATERNO				30 APELLIDO MATERNO				LLANOL					
26	TIPO DOC. IDENTIDAD	27	N° DOC.	28 NOMBRES		APELLIDO MATERNO							
TIPO DE DOC. DE IDENTIDAD				31 N° DE R.U.C.		32 RAZÓN SOCIAL							
33 PERSONA JURÍDICA				34 COND. ESP.		35 N° DE RESOLUCIÓN DE EXONERACIÓN							

**DOMICILIO FISCAL DEL TITULAR CATASTRAL**

39	DEPARTAMENTO	40	PROVINCIA	41	DISTRITO	42	TELÉFONO	43	ANEXO	44	FAX	45	CORREO ELECTRÓNICO
07	CÓDIGO DE VÍA	08	TIPO DE VÍA	09	NOMBRE DE VÍA	11	N° MUNICIPAL	14	NOMBRE DE EDIFICACIÓN	17	N° INTERIOR		
18	CÓDIGO DE HU	19	NOMBRE DE LA HABILITACIÓN URBANA	20	ZONA/SECTOR/ETAPA	21	MANZANA	22	LOTE	23	SUB-LOTE		

**CARACTERÍSTICAS DE LA TITULARIDAD**

46	CONDICIÓN DEL TITULAR	01	01 PROPIETARIO ÚNICO	02 SUCESIÓN INTERSTADIA	03 POSSESION	04 SOCIEDAD CONYUGAL	05 CO-TITULARIDAD	06 LITIGIO	07 OTROS (especificar)			
47	FORMA DE ADQUISICIÓN	01 COMPRA VENTA	02 ANTICIPA LEGÍTIMA	03 TESTAMENTO	04 DONACIÓN	05 ADQUISICIÓN	06 FURTO	07 EXPROPIACIÓN	08 FECHA	09 DÍA	10 MES	11 AÑO
49	CONDICIÓN ESPECIAL DEL PREDIO	01 MONUMENTO HISTÓRICO	02 PREDIO PÉDREGO	03 SISTEMA DE AYUDA DE AERONAVEGACIÓN	04 OTROS (especificar)							

**DESCRIPCIÓN DEL PREDIO**

36	CLASIFICACIÓN DEL PREDIO	01	01 OTRA HABITACIÓN	02 TIENDA - DEPÓSITO - ALMACÉN	03 PREDIO EN ESPEDIO	04 OTROS (especificar)	05 TERRENO EN OBRAS					
37	PREDIO CATASTRAL EN	01 BALSA	02 MERCADO	03 GRUPO FAMILIAR	04 CENTRO COMERCIAL	05 CUESTA	06 CALLEJÓN	07 PREDIO DESPRENDIDO				
38	CÓDIGO DE USO	01	USO DEL PREDIO CATASTRAL (Descripción)				02	ESTRUCTURACIÓN	03	ZONIFICACIÓN		
39	ÁREA DE TERRENO TÍTULO (M <sup>2</sup> )	40	ÁREA DE TERRENO DECLARADA (M <sup>2</sup> )	41	ÁREA DE TERRENO VERIFICADA (M <sup>2</sup> )	214.76						
42	LINDEROS DE LOTE (ML)	43	MEDIDA EN CAMPO	44	MEDIDA SEGÚN TÍTULO	45	COLINDANCIAS EN CAMPO	46	COLINDANCIAS SEGÚN TÍTULO			
	FRENTE		20.76				AV. SAN MARTIN					
	DERECHA		10.35				AV. CORONEL PORTILLO					
	IZQUIERDA		10.35				LOTE 03					
	FONDO		20.75				LOTE 02					

Figura N° E.01.- Ficha individual escaneada. Cara frontal.

**SERVICIOS BÁSICOS**

87	LUZ	1	88	AGUA	1	89	TELÉF.	70	DESAGÜE	1	90	N° SUM. LUZ	2664919	91	N° CONTRATO DE ANUA	92	N° TELÉFONO
----	-----	---	----	------	---	----	--------	----	---------	---	----	-------------	---------	----	---------------------	----	-------------

**CONSTRUCCIONES**

93	N° PIDO BOTANICO MEZZANINE	94		95	96										97	98	99			
		FECHA DE CONSTRUCCIÓN			MEP	ECB	ECO	CATEGORIAS												
		MEB	AÑO		ESTRUCTURA					ACABADOS					ÁREA CONSTRUÍDA (M <sup>2</sup> )		UCA			
			03	04	03														43.68	

**OBRAS COMPLEMENTARIAS / OTRAS INSTALACIONES**

100	CÓDIGO	DESCRIPCIÓN	97			98			99			101	102	103	104	105	106	
			MEP	ECB	ECO	LARGO	ANCHO	ALTO	PRODUCTO TOTAL	UNIDAD DE MEDIDA	UCA							

**DOCUMENTOS**

107	TIPO DE DOCUMENTO	108	N° DE DOCUMENTO	109			110	ÁREA AUTORIZADA	111	TIPO DE DOCUMENTO	112	N° DE DOCUMENTO	113				
				DA	MEB	AÑO							DA	MEB	AÑO	ÁREA AUTORIZADA	

**INSCRIPCIÓN DEL PREDIO CATASTRAL EN EL REGISTRO DE PREDIOS**

104	TIPO DE PARTIDA REGISTRAL	105	NÚMERO	106	FOJAS	107	ASIENTO	108	FECHA DE INSCRIPCIÓN DEL PREDIO	109	DECLARATORIA DE FÁBRICA	110	AS INSC DE FÁBRICA	111	FECHA DE INSCRIPCIÓN DE FÁBRICA
-----	---------------------------	-----	--------	-----	-------	-----	---------	-----	---------------------------------	-----	-------------------------	-----	--------------------	-----	---------------------------------

**EVALUACIÓN DEL PREDIO CATASTRAL**

114	EVALUACIÓN DEL PREDIO CATASTRAL	115	ÁREA DE TERRENO INVADIDA (M <sup>2</sup> )
	PREDIO CATASTRAL CASO		EN LOTE COLIGANTE
	PREDIO CATASTRAL RECONSTRUÍDO		EN ÁREA PÚBLICA
	PREDIO CATASTRAL SUPLENIDO		EN ZONA DE RESERVA
	PREDIO CATASTRAL COMPUESTO		EN ÁREA AFERENTE

**INFORMACIÓN COMPLEMENTARIA**

116	CONDICIÓN DE DECLARANTE	117	IDENTIFICACIÓN DE LOS LITIGANTES	118	ESTADO DE LLENADO DE LA FICHA
	1 = TITULAR CATASTRAL, DE REPRESENTANTE LEGAL, DE ARRENDATARIO, O FAMILIAR		1 = FICHA COMPLETA		1 = FICHA COMPLETA
	2 = VECINO		2 = FICHA INCOMPLETA		2 = FICHA INCOMPLETA
	3 = OTRO (especificar)		3 = FICHA COMPLETA EN OFICINA		3 = FICHA COMPLETA EN OFICINA
			4 = FICHA COMPLETA EN CONTROL EXTERNO		4 = FICHA COMPLETA EN CONTROL EXTERNO
119	MANTENIMIENTO	120	N° DE HABITANTES	121	N° DE FAMILIAS

**OBSERVACIONES**

VIVIENDA CON ADOBE EN MAL ESTADO

DECLARO BAJO JURAMENTO QUE LOS DATOS COMBINADOS EN LA DECLARACIÓN SON VERDADEROS  
LA FICHA CATASTRAL CERTIFICA LA EXISTENCIA Y CARACTERÍSTICAS DEL PREDIO. ESTA FICHA NO GENERA DERECHOS DE PROPIEDAD, NI RESUELVE LAS OBLIGACIONES MUNICIPALES

120	FIRMA DEL DECLARANTE	121	FIRMA DEL SUPERVISOR	122	FIRMA DEL TÉCNICO CATASTRAL	123	V°B° DEL VERIFICADOR CATASTRAL
	DNI: 42262060		DNI:		DNI: 41427571		DNI:
	NOMBRES: JOSE BENITO		NOMBRES:		NOMBRES: CARLOS DE LA CRUZ LAZARO		NOMBRES:
	APELLIDOS: ABANTO LLANOL		APELLIDOS:		APELLIDOS:		APELLIDOS:
	FECHA: 22/11/2011		FECHA:		FECHA: 22/11/2011		FECHA:

Figura N° E.02.- Ficha individual escaneada. Cara posterior.





NÚMERO DE FICHA

**FICHA CATASTRAL URBANA INDIVIDUAL**

NÚMERO DE FICHAS POR LOTE

ESCUDO  
DISTRITAL,  
PROVINCIAL

01	CÓDIGO ÚNICO CATASTRAL - CUC	02	CÓDIGO HOJA CATASTRAL
03 CÓDIGO DE REFERENCIA CATASTRAL			
04	CÓD. CONTRIBUYENTE DE RENTAS	05	CÓDIGO PREDIAL DE RENTAS
06 ACUMULADA (CÓDIGO PREDIAL DE RENTAS)			

USO	SECTOR	MANZANA	LOTE	EDIFICA	ENTRADA	PISO	UNIDAD	DC
15	08	12	01	01	04	00	2	01

LOGO  
ENTIDAD  
EJECUTORA

**UBICACIÓN DEL PREDIO CATASTRAL**

07	CÓDIGO DE VÍA	08	TIPO DE VÍA	09	NOMBRE DE VÍA	10	TIPO DE PUEBLO	11	N° MUNICIPAL	12	CANTÓN	13	N° DE CERTIF. (V. CLASIFICACIÓN)
			Av.		DE LA UNIÓN		D						

14	NOMBRE DE LA EDIFICACIÓN		15	TIPO DE EDIFICACIÓN	16	TIPO DE EDIFICACIÓN	17	N° INTERIOR			
18	CÓDIGO HU	NOMBRE DE LA HABILITACIÓN URBANA		19	ZONA/SECTOR/ETAPA	20	MANZANA	21	LOTE	22	SUB-LOTE

**IDENTIFICACIÓN DEL TITULAR CATASTRAL**

24	TIPO DE TITULAR	25	ESTADO CIVIL	26	NOMBRES	27	APELLIDO MATERNO
28	TIPO DOC. IDENTIDAD	29	APELLIDO PATERNO	30	NOMBRES	31	APELLIDO MATERNO
32	TIPO DE DOC. DE IDENTIDAD	33	RAZÓN SOCIAL	34	PERSONA JURÍDICA	35	COND. ESP.
36	N° DE RESOLUCIÓN DE ENONERACIÓN	37	N° DE BOLSA DE PENSIONES	38	FECHA DE INICIO DE LA ENONERACIÓN	39	FECHA DE VENCIMIENTO DE LA ENONERACIÓN

**DOMICILIO FISCAL DEL TITULAR CATASTRAL**

39	DEPARTAMENTO	40	PROVINCIA	41	DISTRITO	42	TÉLEFONO	43	ANEXO	44	FAX	45	CORREO ELECTRÓNICO
07	CÓDIGO DE VÍA	08	TIPO DE VÍA	09	NOMBRE DE VÍA	10	N° MUNICIPAL	11	NOMBRE DE EDIFICACIÓN	12	N° INTERIOR	18	CODIGO DE HU
19	NOMBRE DE LA HABILITACIÓN URBANA	20	ZONA/SECTOR/ETAPA	21	MANZANA	22	LOTE	23	SUB-LOTE				

**CARACTERÍSTICAS DE LA TITULARIDAD**

46	CONDICIÓN DEL TITULAR	47	FORMA	48	FECHA DE ADQUISICIÓN	49	CONDICIÓN ESPECIAL DEL PREDIO	50	FECHA DE INICIO	51	FECHA DE VENCIMIENTO
----	-----------------------	----	-------	----	----------------------	----	-------------------------------	----	-----------------	----	----------------------

**DESCRIPCIÓN DEL PREDIO**

52	CLASIFICACIÓN DEL PREDIO	53	LIBRO DEL PREDIO CATASTRAL	54	ESTRUCTURACIÓN	55	ZONIFICACIÓN
56	ÁREA DE TERRENO TÍTULO (M <sup>2</sup> )	57	ÁREA DE TERRENO DECLARADA (M <sup>2</sup> )	58	ÁREA DE TERRENO VERIFICADA (M <sup>2</sup> )	59	LARGOS DE LOTE (M)
60	FRONTE	61	DERECHA	62	IZQUIERDA	63	FONDO

**SERVICIOS BÁSICOS**

67	LUZ	68	AGUA	69	TELEF.	70	DESAGÜE	71	Nº SUM. LUZ	72	Nº CONSUMO DE AGUA	73	Nº TELEFONO
----	-----	----	------	----	--------	----	---------	----	-------------	----	--------------------	----	-------------

**CONSTRUCCIONES**

74	Nº PISO Sótano Mezzanine	75 FECHA DE CONSTRUCCIÓN			76	MEP	77	ECS	78	ECC	CATEGORÍAS								ÁREA CONSTRUIDA (M <sup>2</sup> )		86	
		MES	AÑO	ESTRUCTURA							ACABADOS				85	86	87	UCA				
				79							80	81	82	83					84	85		86
							03	04		04												

**OBRAS COMPLEMENTARIAS / OTRAS INSTALACIONES**

89	CÓDIGO	81	DESCRIPCIÓN	75 FECHA COMET.			76	MEP	77	ECS	78	ECC	DIMENSIONES VERIFICADAS			95	PRODUCTO	96	UNIDAD	98	
				82	83	84							92	93	94						95

**DOCUMENTOS**

97	TIPO DE DOCUMENTO	99	Nº DE OTORGAMIENTO	98 FECHA			100	ÁREA AUTORIZADA	97	TIPO DE DOCUMENTO	99	Nº DE DOCUMENTO	98 FECHA			100	ÁREA AUTORIZADA
				01	02	03							04	05	06		

**INSCRIPCIÓN DEL PREDIO CATASTRAL EN EL REGISTRO DE PREDIOS**

104	TIPO DE PARTIDA REGISTRAL	106	NÚMERO	105	FOLIOS	107	ASIENTO	108	FECHA DE INSCRIPCIÓN DEL PREDIO	109	DECLARATORIA DE FABRICA	110	AS INIC. DE FABRICA	111	FECHA DE INSCRIPCIÓN DE FABRICA
-----	---------------------------	-----	--------	-----	--------	-----	---------	-----	---------------------------------	-----	-------------------------	-----	---------------------	-----	---------------------------------

**EVALUACIÓN DEL PREDIO CATASTRAL**

112	EVALUACIÓN DEL PREDIO CATASTRAL				113	ÁREA DE TERRENO INVADIDA (M <sup>2</sup> )			
	PREDIO CATASTRAL OBRADO		PREDIO CATASTRAL RECONSTRUIDO			EN LOTE COLINDANTE		EN ÁREA PÚBLICA	
	PREDIO CATASTRAL SUPLENIDO		PREDIO CATASTRAL CORRIDO			EN JARDÍN DE ABLAYMENTO		EN ÁREA ESTABLECIDA	

**INFORMACIÓN COMPLEMENTARIA**

114	CONDICIÓN DE DECLARANTE	115 IDENTIFICACIÓN DE LOS LITIGANTES		116	ESTADO DE LLENADO DE LA FICHA		
		79	Nº DOCUMENTO	APELLIDOS Y NOMBRES DE LOS LITIGANTES	DÓNDE EL DECLARANTE	1 = FICHA COMPLETA	2 = FICHA INCOMPLETA
						3 = FICHA INCOMPLETA	4 = FICHA INCOMPLETA
118	MANTENIMIENTO	117	Nº DE HABITANTES	119	Nº DE FAMILIAS		

**OBSERVACIONES**

ABANDONADO.			
DECLARO BAJO JURAMENTO QUE LOS DATOS CONSIGNADOS EN LA DECLARACIÓN SON VERDADEROS. LA FICHA CATASTRAL CERTIFICA LA EXISTENCIA Y CARACTERÍSTICAS DEL PREDIO. ESTA FICHA NO OTORGA DERECHO DE PROPIEDAD, NI ESTABLECE LAS OBLIGACIONES FISCALES.			
120	FIRMA DEL DECLARANTE	121	FIRMA DEL SUPERVISOR
122	FIRMA DEL TÉCNICO CATASTRAL	123	VºBº DEL VERIFICADOR CATASTRAL
DNI:		DNI:	
NOMBRES:		NOMBRES:	
APELLIDOS:		APELLIDOS:	
FECHA:		FECHA:	

NÚMERO DE FICHA

**FICHA CATASTRAL URBANA INDIVIDUAL**

NÚMERO DE FICHAS POR LOTE

ESCUDO  
DISTRITAL,  
PROVINCIAL

LOGO  
ENTIDAD  
EJECUTORA

CÓDIGO ÚNICO CATASTRAL - CUC										CÓDIGO HOJA CATASTRAL									
CÓDIGO DE REFERENCIA CATASTRAL																			
URBANO		SECTOR		MANZANA		LOTE		EDIFICA		ENTRADA		PISO		UNIDAD		DC			
15081201		01		014		003		01		01		01		001		001			
CÓD. CONTRIBUYENTE DE RENTAS					CÓDIGO PREDIAL DE RENTAS					UNIDAD ACUMULADORA CÓDIGO PREDIAL DE RENTAS									

**UBICACIÓN DEL PREDIO CATASTRAL**

CÓDIGO DE VÍA		TIPO DE VÍA		NOMBRE DE VÍA				TIPO DE PUERTA		N° MUNICIPAL		COND. NUMER.		N° DE CERTIF. DE NUMERACIÓN	
		A0-		LOS ANGELES				P							
NOMBRE DE LA EDIFICACIÓN				TIPO DE EDIFICACIÓN				TIPO DE INTERIOR				N° INTERIOR			
TIPO DE VÍA		TIPO DE PUERTA		CONDICIÓN DE NUMERACIÓN		TIPO DE EDIFICACIÓN		TIPO DE INTERIOR		TIPO DE INTERIOR		TIPO DE INTERIOR		TIPO DE INTERIOR	
AV. AVENIDA DE CALLE JARDIN PARQUE AL. ALAMBRADO		C/AL. CARRETERA PASEO PROLONGACION PASEO AL. VILLAGE CALLE CHIRINO		PRINCIPAL SECUNDARIA GARAJE ESTACIONAMIENTO		P S E E		01. GENERADO POR NUMC. 02. AUT. GEN. POR EL TIT. CAT. 03. GEN. POR EL TEC. CAT.		01. BLOQUE 02. CASA / CHALET 03. EDIFICIO 04. PASELLÓN		01. TERCIAL 02. TERCIAL 03. TERCIAL 04. TERCIAL 05. TERCIAL 06. TERCIAL 07. TERCIAL 08. TERCIAL 09. TERCIAL 10. TERCIAL		01. TERCIAL 02. TERCIAL 03. TERCIAL 04. TERCIAL 05. TERCIAL 06. TERCIAL 07. TERCIAL 08. TERCIAL 09. TERCIAL 10. TERCIAL	
CÓDIGO HU		NOMBRE DE LA HABILITACIÓN URBANA						ZONA/SECTOR/ETAPA		MANZANA		LOTE		SUB-LOTE	

**IDENTIFICACIÓN DEL TITULAR CATASTRAL**

TIPO DE TITULAR		ESTADO CIVIL		NOMBRES		APELLIDO MATERNO	
1 = PERSONA NATURAL 2 = PERSONA JURÍDICA		01 SOLTERO (A) 02 CASADO (A) 03 DIVORCIADO (A) 04 VIUDO (A) 05 CONVIVIENTE					
TIPO DOC. IDENTIDAD		N° DOC.		NOMBRES		APELLIDO MATERNO	
27							
TIPO DOC. IDENTIDAD		N° DOC.		NOMBRES		APELLIDO MATERNO	
27							
TIPO DE DOC. DE IDENTIDAD		RAZÓN SOCIAL		FECHA DE INICIO DE LA EXHIBICIÓN		FECHA DE VENCIMIENTO DE LA EXHIBICIÓN	
01 NO PRESENTA DOCUMENTO 02 DNI		01 GOBIERNO CENTRAL 02 GOBIERNO LOCAL 03 CLERP. GENERAL DE NOMBRES 04 UNIVERSIDADES 05 ORGANIZACIÓN POLITICA		03 CARNET DE IDENTIDAD DE POLICIA NACIONAL 04 CARNET DE IDENTIDAD DE FUERZAS ARMADAS 05 PARTIDA DE NACIMIENTO 06 PASAPORTE		07 CARNET DE EXTRANJERIA 08 OTROS (especificar)	
N° DE R.U.C.		PERSONA JURÍDICA		COND. ESP.		DEL TITULAR	
		01 EMPRESA 02 COOPERATIVA 03 ASOCIACIÓN 04 FUNDACIÓN 05 OTROS (especificar)		01 GOBIERNO CENTRAL 02 GOBIERNO LOCAL 03 CLERP. GENERAL DE NOMBRES 04 UNIVERSIDADES 05 ORGANIZACIÓN POLITICA		06 BENEFICENCIA PÚBLICA 07 COMUNIDAD CAMP. / NATIVA 08 ORGANISMOS INTERNI. 09 GOBIERNO EXTRANJERO	
N° DE RESOLUCIÓN DE EXHIBICIÓN		N° DE BOLETA DE FENOMENA		FECHA DE INICIO DE LA EXHIBICIÓN		FECHA DE VENCIMIENTO DE LA EXHIBICIÓN	

**DOMICILIO FISCAL DEL TITULAR CATASTRAL**

DEPARTAMENTO		PROVINCIA		DISTRITO		TELÉFONO		ANEXO		FAX		CORREO ELECTRÓNICO			
CÓDIGO DE VÍA		TIPO DE VÍA		NOMBRE DE VÍA				N° MUNICIPAL		NOMBRE DE EDIFICACIÓN		N° INTERIOR			
CÓDIGO DE HU		NOMBRE DE LA HABILITACIÓN URBANA						ZONA/SECTOR/ETAPA		MANZANA		LOTE		SUB-LOTE	

**CARACTERÍSTICAS DE LA TITULARIDAD**

CONDICIÓN DEL TITULAR		FORMA		CONDICIÓN ESPECIAL DEL PREDIO (Especificación)		N° DE RESOLUCIÓN DE EXHIBICIÓN DEL PREDIO	
01 PROPIETARIO ÚNICO 02 SUCECIÓN INTERVISTA 03 POSESIÓN 04 SOCIEDAD CONYUGAL 05 CO-TITULARIDAD 06 LITIGIO 07 OTROS (especificar)		01 COMPRA VENTA 02 ANTIPO LEGÍTIMA 03 TESTAMENTO 04 DONACIÓN 05 ADICIÓN 06 FUSIÓN 07 SUEPRIOPIACIÓN 08 FUSIÓN 09 SUEPRIOPIACIÓN		01 MONUMENTO HISTÓRICO 02 PREDIO RÍPTICO 03 SISTEMA DE AYUDA DE ACCESIBILIDAD 04 OTROS (especificar)		01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	
FECHA DE ADQUISICIÓN		FECHA DE ADQUISICIÓN		FECHA DE INICIO		FECHA DE VENCIMIENTO	

**DESCRIPCIÓN DEL PREDIO**

CLASIFICACIÓN DEL PREDIO		PREDIO		CÓDIGO DE USO		USO DEL PREDIO CATASTRAL (Distinguido)		ESTRUCTURACIÓN		ZONIFICACIÓN					
01 CASA PARTICIÓN 02 TIENDA - DEPÓSITO - ALMACÉN 03 PREDIO EN ESPIDO 04 OTROS (especificar)		01 CALLE 02 CALLE 03 CALLE 04 CALLE 05 CALLE 06 CALLE 07 CALLE 08 CALLE 09 CALLE 10 CALLE 11 CALLE 12 CALLE 13 CALLE 14 CALLE 15 CALLE 16 CALLE 17 CALLE 18 CALLE 19 CALLE 20 CALLE 21 CALLE 22 CALLE 23 CALLE 24 CALLE 25 CALLE 26 CALLE 27 CALLE 28 CALLE 29 CALLE 30 CALLE 31 CALLE 32 CALLE 33 CALLE 34 CALLE 35 CALLE 36 CALLE 37 CALLE 38 CALLE 39 CALLE 40 CALLE 41 CALLE 42 CALLE 43 CALLE 44 CALLE 45 CALLE 46 CALLE 47 CALLE 48 CALLE 49 CALLE 50 CALLE 51 CALLE 52 CALLE 53 CALLE 54 CALLE 55 CALLE 56 CALLE 57 CALLE 58 CALLE 59 CALLE 60 CALLE 61 CALLE 62 CALLE 63 CALLE 64 CALLE 65 CALLE 66 CALLE 67 CALLE 68 CALLE 69 CALLE 70 CALLE 71 CALLE 72 CALLE 73 CALLE 74 CALLE 75 CALLE 76 CALLE 77 CALLE 78 CALLE 79 CALLE 80 CALLE 81 CALLE 82 CALLE 83 CALLE 84 CALLE 85 CALLE 86 CALLE 87 CALLE 88 CALLE 89 CALLE 90 CALLE 91 CALLE 92 CALLE 93 CALLE 94 CALLE 95 CALLE 96 CALLE 97 CALLE 98 CALLE 99 CALLE 100 CALLE		01 CALLE 02 CALLE 03 CALLE 04 CALLE 05 CALLE 06 CALLE 07 CALLE 08 CALLE 09 CALLE 10 CALLE 11 CALLE 12 CALLE 13 CALLE 14 CALLE 15 CALLE 16 CALLE 17 CALLE 18 CALLE 19 CALLE 20 CALLE 21 CALLE 22 CALLE 23 CALLE 24 CALLE 25 CALLE 26 CALLE 27 CALLE 28 CALLE 29 CALLE 30 CALLE 31 CALLE 32 CALLE 33 CALLE 34 CALLE 35 CALLE 36 CALLE 37 CALLE 38 CALLE 39 CALLE 40 CALLE 41 CALLE 42 CALLE 43 CALLE 44 CALLE 45 CALLE 46 CALLE 47 CALLE 48 CALLE 49 CALLE 50 CALLE 51 CALLE 52 CALLE 53 CALLE 54 CALLE 55 CALLE 56 CALLE 57 CALLE 58 CALLE 59 CALLE 60 CALLE 61 CALLE 62 CALLE 63 CALLE 64 CALLE 65 CALLE 66 CALLE 67 CALLE 68 CALLE 69 CALLE 70 CALLE 71 CALLE 72 CALLE 73 CALLE 74 CALLE 75 CALLE 76 CALLE 77 CALLE 78 CALLE 79 CALLE 80 CALLE 81 CALLE 82 CALLE 83 CALLE 84 CALLE 85 CALLE 86 CALLE 87 CALLE 88 CALLE 89 CALLE 90 CALLE 91 CALLE 92 CALLE 93 CALLE 94 CALLE 95 CALLE 96 CALLE 97 CALLE 98 CALLE 99 CALLE 100 CALLE		01 CALLE 02 CALLE 03 CALLE 04 CALLE 05 CALLE 06 CALLE 07 CALLE 08 CALLE 09 CALLE 10 CALLE 11 CALLE 12 CALLE 13 CALLE 14 CALLE 15 CALLE 16 CALLE 17 CALLE 18 CALLE 19 CALLE 20 CALLE 21 CALLE 22 CALLE 23 CALLE 24 CALLE 25 CALLE 26 CALLE 27 CALLE 28 CALLE 29 CALLE 30 CALLE 31 CALLE 32 CALLE 33 CALLE 34 CALLE 35 CALLE 36 CALLE 37 CALLE 38 CALLE 39 CALLE 40 CALLE 41 CALLE 42 CALLE 43 CALLE 44 CALLE 45 CALLE 46 CALLE 47 CALLE 48 CALLE 49 CALLE 50 CALLE 51 CALLE 52 CALLE 53 CALLE 54 CALLE 55 CALLE 56 CALLE 57 CALLE 58 CALLE 59 CALLE 60 CALLE 61 CALLE 62 CALLE 63 CALLE 64 CALLE 65 CALLE 66 CALLE 67 CALLE 68 CALLE 69 CALLE 70 CALLE 71 CALLE 72 CALLE 73 CALLE 74 CALLE 75 CALLE 76 CALLE 77 CALLE 78 CALLE 79 CALLE 80 CALLE 81 CALLE 82 CALLE 83 CALLE 84 CALLE 85 CALLE 86 CALLE 87 CALLE 88 CALLE 89 CALLE 90 CALLE 91 CALLE 92 CALLE 93 CALLE 94 CALLE 95 CALLE 96 CALLE 97 CALLE 98 CALLE 99 CALLE 100 CALLE		01 CALLE 02 CALLE 03 CALLE 04 CALLE 05 CALLE 06 CALLE 07 CALLE 08 CALLE 09 CALLE 10 CALLE 11 CALLE 12 CALLE 13 CALLE 14 CALLE 15 CALLE 16 CALLE 17 CALLE 18 CALLE 19 CALLE 20 CALLE 21 CALLE 22 CALLE 23 CALLE 24 CALLE 25 CALLE 26 CALLE 27 CALLE 28 CALLE 29 CALLE 30 CALLE 31 CALLE 32 CALLE 33 CALLE 34 CALLE 35 CALLE 36 CALLE 37 CALLE 38 CALLE 39 CALLE 40 CALLE 41 CALLE 42 CALLE 43 CALLE 44 CALLE 45 CALLE 46 CALLE 47 CALLE 48 CALLE 49 CALLE 50 CALLE 51 CALLE 52 CALLE 53 CALLE 54 CALLE 55 CALLE 56 CALLE 57 CALLE 58 CALLE 59 CALLE 60 CALLE 61 CALLE 62 CALLE 63 CALLE 64 CALLE 65 CALLE 66 CALLE 67 CALLE 68 CALLE 69 CALLE 70 CALLE 71 CALLE 72 CALLE 73 CALLE 74 CALLE 75 CALLE 76 CALLE 77 CALLE 78 CALLE 79 CALLE 80 CALLE 81 CALLE 82 CALLE 83 CALLE 84 CALLE 85 CALLE 86 CALLE 87 CALLE 88 CALLE 89 CALLE 90 CALLE 91 CALLE 92 CALLE 93 CALLE 94 CALLE 95 CALLE 96 CALLE 97 CALLE 98 CALLE 99 CALLE 100 CALLE							
ÁREA DE TERRENO TÍTULO (M2)		ÁREA DE TERRENO DECLARADA (M2)		ÁREA DE TERRENO VERIFICADA (M2)		LINDEROS DE LOTE (MIL)		MEDIDA EN CAMPO		MEDIDA SEGÚN TÍTULO		COUNDANCIAS EN CAMPO		COUNDANCIAS SEGÚN TÍTULO	
				203 29											
FRENTE		DERECHA		IZQUIERDA		FONDO		10.20		14.43		14.93		10.20	

SERVICIOS BÁSICOS

62	LUZ	65	AGUA	68	TELEF.	70	DESAGÜE	72	N° SUM. LUZ	74	N° CONTRATO DE AGUA	76	N° TELEFONO
----	-----	----	------	----	--------	----	---------	----	-------------	----	---------------------	----	-------------

CONSTRUCCIONES

N° PISO BÓTANO MEZZANINE	FECHA DE CONSTRUCCIÓN		MEP	ECS	ECC	CATEGORÍAS										ÁREA CONSTRUIDA (M <sup>2</sup> )	
	MES	AÑO				ESTRUCTURA					ACABADOS					TERMINADA	EN CONSTRUCCIÓN

OBRAS COMPLEMENTARIAS / OTRAS INSTALACIONES

CÓDIGO	DESCRIPCIÓN	FECHA CONST.			ECC	DIMENSIONES VERIFICADAS			PRODUCTO TOTAL	UNIDAD DE MEDIDA
		MEP	ECS	ECC		LARGO	ANCHO	ALTO		

DOCUMENTOS

TIPO DE DOCUMENTO	N° DE DOCUMENTO	FECHA			ÁREA AUTORIZADA	TIPO DE DOCUMENTO	N° DE DOCUMENTO	FECHA			ÁREA AUTORIZADA
		DÍA	MES	AÑO				DÍA	MES	AÑO	

INSCRIPCIÓN DEL PREDIO CATASTRAL EN EL REGISTRO DE PREDIOS

104	TIPO DE PARTIDA REGISTRAL	105	NÚMERO	106	FOJAS	107	ASIENTO	108	FECHA DE INSCRIPCIÓN DEL PREDIO	109	DECLARATORIA DE FÁBRICA	110	AS. FÁB. DE FÁBRICA	111	FECHA DE INSCRIPCIÓN DE FÁBRICA
-----	---------------------------	-----	--------	-----	-------	-----	---------	-----	---------------------------------	-----	-------------------------	-----	---------------------	-----	---------------------------------

EVALUACIÓN DEL PREDIO CATASTRAL

112	EVALUACIÓN DEL PREDIO CATASTRAL				ÁREA DE TERRENO INVAJADA (M <sup>2</sup> )			
	PREDIO CATASTRAL DIBO	PREDIO CATASTRAL RECONSTRUIDO	EN LOTE COLINDANTE	EN ÁREA PÚBLICA				
	PREDIO CATASTRAL RECONSTRUIDO	PREDIO CATASTRAL CONFORME	EN JARDÍN DE ABLAMIENTO	EN ÁREA ESTABLE				

INFORMACIÓN COMPLEMENTARIA

114	CONDICIÓN DE DECLARANTE	115	IDENTIFICACIÓN DE LOS LITIGANTES	116	ESTADO DE LLENADO DE LA FICHA
	1 = TITULAR CATASTRAL, 2 = REPRESENTANTE LEGAL, 3 = ARRENDATARIO, 4 = FAMILIAR, 5 = VECINO, 6 = OTRO (Especificar)		1 = FICHA COMPLETA, 2 = FICHA INCOMPLETA, 3 = COMPLETADA EN ORIGEN, 4 = COMPLETADA EN CONTROL EXTERIOR		1 = POR SER PREDIO CATASTRAL NUEVO, 2 = POR VARIACIÓN EN LA CONSTRUCCIÓN, 3 = POR CAMBIO DEL TITULAR CATASTRAL, 4 = POR CAMBIO DE DBO

OBSERVACIONES

NO SE ENCUENTRA PROPICIAFABO APARTE DE LA DECLARACIÓN

120	FIRMA DEL DECLARANTE	121	FIRMA DEL SUPERVISOR	122	FIRMA DEL TÉCNICO CATASTRAL	123	V°B° DEL VERIFICADOR CATASTRAL
DNI:		DNI:		DNI: 09402610	DNI:		
NOMBRES:		NOMBRES:		NOMBRES: KARIN JANET	NOMBRES:		
APELLIDOS:		APELLIDOS:		APELLIDOS: ANIBERTO TELLO	APELLIDOS:		
FECHA:		FECHA:		FECHA: 22/11/2011	FECHA:		

NÚMERO DE FICHA

**FICHA CATASTRAL URBANA INDIVIDUAL**

NÚMERO DE FICHAS POR LOTE

ESCUDO  
DISTRITAL,  
PROVINCIAL

LOGO  
ENTIDAD  
EJECUTORA

CÓDIGO ÚNICO CATASTRAL - CUC				CÓDIGO HOJA CATASTRAL			
CÓDIGO DE REFERENCIA CATASTRAL							
DPTO.	PROV.	DIST.	SECTOR	MANZANA	LOTE	EDIFICA	ENTRADA
1	5	0	8	1	2	0	1
0	1	2	0	1	0	1	0
CÓD. CONTRIBUYENTE DE RENTAS		CÓDIGO PREDIAL DE RENTAS		UNIDAD ACUMULADA A CÓDIGO PREDIAL DE RENTAS			
06		05		09			

**UBICACIÓN DEL PREDIO CATASTRAL**

CÓDIGO DE VÍA		TIPO DE VÍA		NOMBRE DE VÍA		TIPO DE PUERTA		N° MUNICIPAL		COND. NÚMERO		N° DE CERTIF. DE NUMERACIÓN			
		A0		LOS ANGELES		P									
NOMBRE DE LA EDIFICACIÓN				TIPO DE EDIFICACIÓN		TIPO DE INTERIOR		N° INTERIOR							
TIPO DE VÍA				TIPO DE PUERTA		CONDICIÓN DE NUMERACIÓN		TIPO DE EDIFICACIÓN		TIPO DE INTERIOR					
DA CALLE				PRINCIPAL		01 GENERADO POR MANO		01 BLOCK		DEPARTAMENTO					
DA AVDA				SECUNDARIA		02 ALT. GEN. POR EL SIT. CAT.		02 CASA / CHALET		CASA / CHALET					
DA AVDA				GARAGE		03 GEN. POR EL TERC. CAT.		03 EDIFICIO		DIGNA					
DA AVDA				ESTACIONAMIENTO				04 PABELLÓN		SERVICIOS					
DA AVDA										DEPOSITO					
DA AVDA										19 OTRAS (especificar)					
CÓDIGO HU		NOMBRE DE LA HABILITACIÓN URBANA				ZONA/SECTOR/ETAPA		MANZANA		LOTE		SUB-LOTE			
18						20		21		22		23			

**IDENTIFICACIÓN DEL TITULAR CATASTRAL**

TIPO DE TITULAR		1 = PERSONA NATURAL 2 = PERSONA JURÍDICA		ESTADO CIVIL		01 SOLTERO (A)		02 CASADO (A)		03 DIVORCIADO (A)		04 VIUDO (A)		05 CONVIVIENTE	
TIPO DOC. IDENTIDAD		27 N° DOC.		NOMBRES		228		APELLIDO PATERNO		229		APELLIDO MATERNO		230	
TIPO DOC. IDENTIDAD		22 N° DOC.		NOMBRES		228		APELLIDO PATERNO		229		APELLIDO MATERNO		230	
TIPO DE DOC. DE IDENTIDAD		01 NO PRESENTA DOCUMENTO		02 CARNET DE IDENTIDAD DE POLICIA NACIONAL		03 PARTIDA DE NACIMIENTO		04 CARNET DE EXTRANJERIA		05 OTROS (especificar)					
N° DE R.U.C.		RAZÓN SOCIAL													
PERSONA JURÍDICA		01 EMPRESA		02 COOPERATIVA		03 ASOCIACIÓN		04 FUNDACIÓN		05 OTROS (especificar)					
COND. ESP.		01 GOBIERNO CENTRAL		02 GOBIERNO LOCAL		03 GOBIERNO FEDERAL		04 SERVICIO PÚBLICO		05 ENTIDADES RELIGIOSAS		06 ENTIDADES RELIGIOSAS		07 GOBIERNO EXTRANJERO	
DEL TITULAR		07 GOBIERNO CENTRAL		08 GOBIERNO LOCAL		09 GOBIERNO FEDERAL		10 SERVICIO PÚBLICO		11 ENTIDADES RELIGIOSAS		12 ENTIDADES RELIGIOSAS		13 GOBIERNO EXTRANJERO	
N° DE RESOLUCIÓN DE EXEMERACIÓN		N° DE BOLETA DE FUNDACIÓN		FECHA DE NAC. DE LA EXEMERACIÓN		FECHA DE VENCIMIENTO DE LA EXEMERACIÓN									
36		37		38		39									

**DOMICILIO FISCAL DEL TITULAR CATASTRAL**

DEPARTAMENTO		PROVINCIA		DISTRITO		TELÉFONO		ANEXO		FAX		CORREO ELECTRÓNICO	
CÓDIGO DE VÍA		TIPO DE VÍA		NOMBRE DE VÍA		N° MUNICIPAL		NOMBRE DE EDIFICACIÓN		N° INTERIOR			
CÓDIGO DE HU		NOMBRE DE LA HABILITACIÓN URBANA				ZONA/SECTOR/ETAPA		MANZANA		LOTE		SUB-LOTE	
18						20		21		22		23	

**CARACTERÍSTICAS DE LA TITULARIDAD**

CONDICIÓN DEL TITULAR		01 PROPIETARIO ÚNICO		02 SUCESIÓN INTERVENCIA		03 POSEEDOR		04 SOCIEDAD CONJUNTA		05 CO-TITULARIDAD		06 LITIGIO		07 OTROS (especificar)	
FORMA		01 COMPRA VENTA		02 ANTICIPA LEGÍTIMA		03 TESTAMENTO		04 DONACIÓN		05 ADSCRIPCIÓN		06 FUSIÓN		07 ESPROPRIACIÓN	
DE ADQUISICIÓN		08 FUSIÓN		09 FUSIÓN		10 FUSIÓN		11 FUSIÓN		12 FUSIÓN		13 FUSIÓN		14 FUSIÓN	
CONDICIÓN ESPECIAL DEL PREDIO (Especificar)		01 MONUMENTO HISTÓRICO		02 PREDIO FORTIFICADO		03 SISTEMA DE AJUDA DE AERODIVISIÓN		04 OTROS (especificar)							
N° DE RESOLUCIÓN DE EXEMERACIÓN DEL PREDIO		PORCENTAJE		FECHA DE INICIO		FECHA DE VENCIMIENTO									
36		37		38		39									

**DESCRIPCIÓN DEL PREDIO**

CLASIFICACIÓN DEL PREDIO		01 CASA HABITACION		02 TERRENO - DEPÓSITO - ALMACÉN		03 PREDIO EN ESPICIO		04 OTROS (especificar)		05 TERRENO EN COSECHA	
A.1 OLIVICULTIVO		A.2 HORTICULTIVO		A.3 CINE, TEATRO		A.4 INDUSTRIA		A.5 TALLER		A.6 ESTABLE / TEMPLO	
A.7 CENTRO DE SERVICIOS		A.8 SERVIDO DE COMIDA		A.9 PARQUE		A.10 CEMENTERIO		A.11 BUS ESTACIÓN		A.12 SERVIDO FINANCIERO	
A.13 TEMPLO DE TRANSPORTE		A.14 MERCADO		A.15 CLUB SOCIAL		A.16 CLUB DE ESPORTES		A.17 PLAZA DE ESTACIONAMIENTO		A.18 OTROS	
PREDIO CATASTRAL EN		01 GRUPO		02 ESPACIO		03 CAMPO PEBAL		04 CENTRO COMERCIAL		05 OLIVAR	
06 SOLAR		07 CORRALÓN		08 ALDEA		09 AJE		10 PREDIO EN ESPICIO		11 OTROS (especificar)	
CÓDIGO DE USO		USO DEL PREDIO CATASTRAL (Especificar)				ESTRUCTURACIÓN		ZONIFICACIÓN			
30						31		32			
ÁREA DE TERRENO TÍTULO (M <sup>2</sup> )		ÁREA DE TERRENO DECLARADA (M <sup>2</sup> )		ÁREA DE TERRENO VERIFICADA (M <sup>2</sup> )							
				20.3 29							
LINDEROS DE LOTE (M.)		MEDIDA EN CAMPO		MEDIDA SEGÚN TÍTULO		COORDENADAS EN CAMPO		COORDENADAS SEGÚN TÍTULO			
FRENTE		10.20		10.20		20.3 29		20.3 29			
DERECHA		19.93		19.93		20.3 29		20.3 29			
IZQUIERDA		19.93		19.93		20.3 29		20.3 29			
FONDO		10.20		10.20		20.3 29		20.3 29			

**SERVICIOS BÁSICOS**

74	LUZ	75	AGUA	76	TELEF.	77	DESAGÜE	78	N° SUM. LUZ	79	N° CONTRATO DE AGUA	80	N° TELEFONO
----	-----	----	------	----	--------	----	---------	----	-------------	----	---------------------	----	-------------

**CONSTRUCCIONES**

74	N° PISO SÓTANO MEZZANINE	76			77	78	CATEGORÍAS										84						
		FECHA DE CONSTRUCCIÓN		MEP			ECS	ECC	ESTRUCTURA					ACABADOS					ÁREA CONSTRUIDA (M <sup>2</sup> )				
		MEZ	AÑO				79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	
				02	03	03																	

MEP: MATERIAL ESTRU. PREDOMINANTE    ECS: ESTADO DE CONSERVACIÓN    ECC: ESTADO DE LA CONSTRUCCIÓN    LCA: UBICACIÓN DE CONSTRUCCIÓN ANTERIOR/ALTERNATIVA

01 CONCRETO    01 BUENO    01 TERMINADO    01 EN RETIRO MUNICIPAL    01 ALTURA NO REGULAMENTARIA  
 02 LADRILLO    02 BUENO    02 EN CONSTRUCCIÓN    02 EN JARDÍN DE AISLAMIENTO    02 EN PARQUE  
 03 ADOSADO/CHACA, MACISA    03 BIELLADO    03 INCONCLUSA    03 EN VÍA PÚBLICA    03 EN BIEN DOMIN.  
 04 BARRIL    04 MALO    04 EN FERRIS    04 EN LOTTE COLONARIANTE    04 EN BIEN DOMIN.

**OBRAS COMPLEMENTARIAS / OTRAS INSTALACIONES**

CÓDIGO	DESCRIPCIÓN	76			77	78	DIMENSIONES VERIFICADAS			PRODUCTO	UNIDAD DE MEDIDA	LCA
		MEZ	AÑO	MEP			ECS	ECC	LARGO			

**DOCUMENTOS**

74	TIPO DE DOCUMENTO	N° DE DOCUMENTO	76			77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	
			DA	MEZ	AÑO																			ÁREA AUTORIZADA

TIPO DE DOCUMENTOS

01 COMPROMISO DE OBRAS    04 ÚLTIMA DECLARACIÓN JURADA AUTORIZADA    07 DOCUMENTO DE TERMINACIÓN    08 COMPROMISO CATASTRAL    09 COMPROMISO NOTARIAL  
 02 LICENCIA DE CONSTRUCCIÓN    05 RESOLUCIÓN DE EXPROPIACIÓN    08 DOCUMENTO DE INICIACIÓN DE CONSTRUCCIÓN    10 PLAN REGULACIONAL CATASTRAL    11 PLAN REGULACIONAL MUNICIPAL  
 03 RESOLUCIÓN DE EMERGENCIA    06 PROMESA    09 PLAN DE ORDENAMIENTO DE PARTIDAS DE ORDENACIÓN    12 PLAN REGULACIONAL CATASTRAL    13 PLAN REGULACIONAL MUNICIPAL

REGISTRO NOTARIAL DE LA ESCRITURA PÚBLICA    NOMBRE DE LA NOTARIA    KARDER    FECHA DE ESCRITURA PÚBLICA

**INSCRIPCIÓN DEL PREDIO CATASTRAL EN EL REGISTRO DE PREDIOS**

104	TIPO DE PARTIDA REGISTRAL	105	NÚMERO	106	FOJAS	107	ASIENTO	108	FECHA DE INSCRIPCIÓN DEL PREDIO	109	DECLARATORIA DE FABRICA	110	AS INC. DE FABRICA	111	FECHA DE INSCRIPCIÓN DE FABRICA
-----	---------------------------	-----	--------	-----	-------	-----	---------	-----	---------------------------------	-----	-------------------------	-----	--------------------	-----	---------------------------------

**EVALUACIÓN DEL PREDIO CATASTRAL**

112	EVALUACIÓN DEL PREDIO CATASTRAL		113	ÁREA DE TERRENO INVADIDA (M <sup>2</sup> )	
	PREDIO CATASTRAL ORDEN		PREDIO CATASTRAL BOMBEVALLADO	EN LOTTE COLONARITE	EN ÁREA PÚBLICA
	PREDIO CATASTRAL SUPRALINDO		PREDIO CATASTRAL COMPOST	EN JARDÍN DE AISLAMIENTO	EN ÁREA ESTABLE

ORDEN: NO DECLARA    SUPRALINDA: DECLARA MENOS    BOMBEVALLADA: DECLARA MÁS    CONFORME: DECLARA BIEN    ESTOS DATOS SE OBTENDRAN DEL CRUCE DE INFORMACIÓN DE CAMPO Y GABINETE.

**INFORMACIÓN COMPLEMENTARIA**

114	CONDICIÓN DE DECLARANTE	01 TITULAR CATASTRAL	02 REPRESENTANTE LEGAL	03 ARRENDATARIO	04 FAMILIAR	05 VECINO	06 OTRO (Especificar)
115	IDENTIFICACIÓN DE LOS LITIGANTES			ESTADO DE LLENADO DE LA FICHA			
TO	N° DOCUMENTO	APELLIDOS Y NOMBRES DE LOS LITIGANTES		CÓDIGO DEL CONTRAHENTE	1 = FICHA COMPLETA	2 = COMPLETADA EN OFICINA	
					2 = FICHA INCOMPLETA	3 = COMPLETADA EN CONTROL EXTERIOR	
116	MANTENIMIENTO			1 = POR SER PREDIO CATASTRAL NUEVO	2 = POR VARIACIÓN EN LA CONSTRUCCIÓN	3 = POR CAMBIO DEL TITULAR CATASTRAL	4 = POR CAMBIO DE USO

**OBSERVACIONES**

cerco de Ladrillo.

DECLARO BAJO JURAMENTO QUE LOS DATOS CONSIGNADOS EN LA DECLARACIÓN SON VERDADEROS  
LA FICHA CATASTRAL CERTIFICA LA EXISTENCIA Y CARACTERÍSTICAS DEL PREDIO. ESTA FICHA NO GENERA DERECHOS DE PROPIEDAD, NI REHUSAR LAS OBLIGACIONES MUNICIPALES.

120	FIRMA DEL DECLARANTE	121	FIRMA DEL SUPERVISOR	122	FIRMA DEL TÉCNICO CATASTRAL	123	V°S° DEL VERIFICADOR CATASTRAL
DNI:		DNI:		DNI: 09902610			
NOMBRES:		NOMBRES:		NOMBRES: KARIN JANTI			
APELLIDOS:		APELLIDOS:		APELLIDOS: ARBUSTO TELLO			
FECHA:		FECHA:		FECHA: 22/11/2011			