



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y  
COMUNICACION DE DATOS**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERA ELECTRONICA EN REDES Y COMUNICACIÓN  
DE DATOS**

**AUTOR: VINTIMILLA SARMIENTO, MARIA GABRIELA**

**TEMA: “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN  
QUE TRADUZCA EL ABECEDARIO Y LOS NÚMEROS DEL UNO AL  
DIEZ DEL LENGUAJE DE SEÑAS A TEXTO PARA AYUDA DE  
DISCAPACITADOS AUDITIVOS MEDIANTE DISPOSITIVOS MÓVILES  
ANDROID.”**

**DIRECTOR: ING. ALULEMA, DARWIN**

**CODIRECTOR: ING. SAENZ, FABIAN**

**SANGOLQUÍ, JUNIO 2014**

## CERTIFICADO

Se Certifica que el trabajo titulado “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN QUE TRADUZCA EL ABECEDARIO Y LOS NÚMEROS DEL UNO AL DIEZ DEL LENGUAJE DE SEÑAS A TEXTO PARA AYUDA DE DISCAPACITADOS AUDITIVOS MEDIANTE DISPOSITIVOS MÓVILES ANDROID.”, realizado por María Gabriela Vintimilla Sarmiento, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad De Las Fuerzas Armadas – ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a María Gabriela Vintimilla Sarmiento que lo entregue al Dr. Nikolai Espinosa en su calidad de Director de Carrera.

Sangolquí, Junio de 2014

---

Ing. Darwin Alulema

DIRECTOR

---

Ing. Fabián Saenz.

CODIRECTOR

**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE****INGENIERÍA ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS****DECLARACIÓN DE RESPONSABILIDAD****MARÍA GABRIELA VINTIMILLA SARMIENTO**

DECLARO QUE:

El proyecto de grado denominado **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN QUE TRADUZCA EL ABECEDARIO Y LOS NÚMEROS DEL UNO AL DIEZ DEL LENGUAJE DE SEÑAS A TEXTO PARA AYUDA DE DISCAPACITADOS AUDITIVOS MEDIANTE DISPOSITIVOS MÓVILES ANDROID”** ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros. Consecuentemente este trabajo es de mi autoría. En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, Junio 25 de 2014

---

María Gabriela Vintimilla Sarmiento

1721781548

**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE****INGENIERÍA ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS****AUTORIZACIÓN**

Yo, MARIA GABRIELA VINTIMILLA SARMIENTO

Autorizo a la Universidad de las Fuerzas Armadas ESPE, la publicación en la biblioteca virtual de la Institución del trabajo **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN QUE TRADUZCA EL ABECEDARIO Y LOS NÚMEROS DEL UNO AL DIEZ DEL LENGUAJE DE SEÑAS A TEXTO PARA AYUDA DE DISCAPACITADOS AUDITIVOS MEDIANTE DISPOSITIVOS MÓVILES ANDROID”**, cuyo contenido, ideas y criterios con de mi exclusiva responsabilidad y autoría.

Sangolquí, Junio 25 de 2014

---

María Gabriela Vintimilla Sarmiento

1721781548

## **DEDICATORIA**

A mis padres y mis hermanos por su gran apoyo en las buenas y las malas y por ser el motor que me inspira a superarme cada día.

## **AGRADECIMIENTO**

A Dios, a mis Padres y a mis hermanos por apoyarme en todas mis metas, a mis grandes amigas Pao y Gaby por hacer de cada momento en la universidad un recuerdo inolvidable, a Franklin por su ayuda y a Hugo por su cariño, comprensión y ternura.

A mi director de tesis Ing. Darwin Alulema y al Ing. Fabian Saenz por sus enseñanzas y su paciencia, gracias a todos.

*Gabriela Vintimilla*

## INDICE DE CONTENIDOS

	Pág.
CAPÍTULO I.....	1
DESCRIPCION DEL PROYECTO .....	1
1.1    Introducción.....	1
1.1.1    Antecedentes.....	1
1.1.2    Justificación e Importancia.....	9
1.1.3    Alcance.....	10
1.1.4    Objetivos.....	10
1.2    Estudio de las necesidades de las personas con discapacidad auditiva.....	11
1.2.1    Tipos de discapacidad.....	12
1.2.2    Derechos humanos de las personas con discapacidades distintas.....	18
1.3    Lenguaje de señas.....	19
1.3.1    Declaración Universal de los Derechos Lingüísticos.....	19
1.3.2    Origen de las lenguas de señas.....	22
1.3.3    El Abecedario de acuerdo al Lenguaje de Señas.....	24
CAPÍTULO II.....	26
ESTUDIO DE LA ARQUITECTURA DE ANDROID.....	26
2.1    Características.....	26
2.1.1    Características de Software.....	27
2.2    Arquitectura de Android.....	28
2.2.1    Diagrama de la Arquitectura de Android.....	29
2.3    Sensores disponibles en Android.....	36
2.4    Descripción de las herramientas utilizadas.....	37
2.4.1    Java.....	37
2.4.2    PHP.....	43
2.4.3    MySQL.....	46
2.4.4    Protocolo HTTPS.....	48
2.4.5    Redes Neuronales.....	50
CAPÍTULO III:.....	55
DISEÑO DE LA APLICACIÓN .....	55
3.1    Diseño del algoritmo.....	55
3.1.1    Planteamiento teórico.....	55
3.1.2    Descripción de actividades.....	55

3.1.3	Tratamiento de la imagen.....	57
3.2	Redes neuronales artificiales.....	63
3.2.1	Modelo de una neurona.....	64
3.2.2	Arquitecturas.....	66
3.2.3	Aprendizaje.....	67
3.2.4	Entrenamiento de la red neuronal.....	70
3.3	Generalización.....	73
3.4	Grafo del sistema.....	79
3.4.1	Diagrama de Arquitectura.....	79
3.4.2	Diagrama de Grafos.....	82
3.4.3	Diagrama de Secuencia.....	83
3.4.4	Diagrama de Clases.....	84
3.4.5	Diseño gráfico de la aplicación.....	85
3.4.6	Diagrama de casos de uso.....	86
3.5	Funcionamiento.....	88
3.5.1	Entrenamiento de la Red.....	88
3.5.2	Reconocimiento de la Red.....	90
3.6	Descripción de los módulos.....	93
CAPÍTULO IV.....		97
PRUEBAS DE USO.....		97
4.1	Entrenamiento del Sistema.....	97
4.1.1	Mejor desempeño de back-propagation.....	97
4.2	Pruebas del sistema.....	100
4.2.1	Tabla de resultados de pruebas en dispositivos móviles.....	101
CAPÍTULO V.....		102
CONCLUSIONES Y RECOMENDACIONES.....		102
5.1	Conclusiones.....	102
5.2	Recomendaciones.....	103
BIBLIOGRAFÍA.....		105
ANEXOS.....		<b>¡Error! Marcador no definido.</b>
CODIGO JAVA:.....		<b>¡Error! Marcador no definido.</b>
CODIGO ANDROID:.....		<b>¡Error! Marcador no definido.</b>
CODIGO PHP (APRENDE.PHP).....		<b>¡Error! Marcador no definido.</b>
CODIGO PHP (LIBDATA.PHP).....		<b>¡Error! Marcador no definido.</b>

**CODIGO PHP (RECONOCER.PHP) .....¡Error! Marcador no definido.**  
**CODIGO PHP (RECONOCER\_PESOS.PHP) .....¡Error! Marcador no definido.**  
**CODIGO PHP (TALKTOSIGN.PHP) .....¡Error! Marcador no definido.**

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. B-Touch de Zhenwei You.....	2
Figura 2. Owasys 22C.....	3
Figura 3. Una versión hipotética de una video llamada usada por la ASL.....	4
Figura 4. T-Loop de Vodafone.....	5
Figura 5. FaceTime.....	6
Figura 6. Imagen de subtítulos en un iPhone.....	6
Figura 7. Mensajes con iMessage.....	7
Figura 8. Icono de Melovibe.....	7
Figura 9. Deaf - Hearing. Aplicación Android.....	8
Figura 10. Magic Glove.....	9
Figura 11. Estructura del oído.....	16
Figura 12. Diccionario Oficial de Lengua de Señas Ecuatoriana y Guías para Padres y Docentes de Personas Sordas y la Guía para Intérpretes de Lengua de Señas Ecuatoriana.....	21
Figura 13. Abecedario de acuerdo al lenguaje de las señas.....	25
Figura 14. Números del uno al diez de acuerdo al lenguaje de las señas.....	25
Figura 15. Estructura de la Arquitectura de Android.....	29

Figura 16. Arquitectura del sistema.....	29
Figura 17. Arquitectura del Sistema Linux Kernel.....	30
Figura 18. Arquitectura del Sistema Librerías.....	30
Figura 19. Arquitectura del Sistema Android Runtime.....	32
Figura 20. Arquitectura del Sistema Android Application Framework.....	33
Figura 21. Arquitectura del Sistema Android Aplicaciones.....	35
Figura 22. Estructura de un perceptrón multinivel.....	51
Figura 23. Modelo de una neurona.....	53
Figura 24. Imagen es escala de grises.....	56
Figura 25. Binarización de la imagen.....	56
Figura 26. Imagen reducida a 40x40 pixeles.....	56
Figura 27. Red neuronal con $n=1600$ , $q$ capas ocultas y $r=8$ .....	60
Figura 28. Unidad Neuronal.....	61
Figura 29. Modelo no lineal de una neurona.....	64
Figura 30. Transformación producida por la presencia del umbral.....	66
Figura 31. Topologías de redes Neuronales.....	67
Figura 32. Diagrama que representa el aprendizaje supervisado.....	69
Figura 33. Diagrama que representa el aprendizaje no supervisado.....	69

Figura 34. Datos ajustados apropiadamente (generalización buena).....	75
Figura 35. Datos sobre ajustados (generalización pobre).....	75
Figura 36. Ilustración de la regla de detención temprana basada en la validación cruzada.....	77
Figura 37. Diagrama de Arquitectura Talk to sign.....	80
Figura 38. Diagrama de Grafos Talk to sign.....	82
Figura 39. Diagrama de Secuencia Talk to Sign, en modo aprender.....	83
Figura 40. Diagrama de Secuencia Talk to Sign, Genera Pesos.....	83
Figura 41. Diagrama de Secuencia Talk to Sign, en modo reconocer.....	84
Figura 42. Diagrama de Clases.....	84
Figura 43. Diseño gráfico de la aplicación.....	85
Figura 44 Diagrama General.....	86
Figura 45. Diagrama nivel siguiente (Modo Aprender).....	86
Figura 46. Diagrama nivel siguiente (Modo Genera Pesos).....	86
Figura 47. Diagrama nivel siguiente (Modo Reconocer).....	87
Figura 48. Modo Aprender.....	88
Figura 49. Letra a aprender.....	88
Figura 50. Botón Captura imagen.....	88
Figura 51. Captura de la imagen.....	89

Figura 52. Imagen tratada en interfaz de aplicación.....	89
Figura 53. Modo Genera Pesos.....	90
Figura 54. Modo Reconocer.....	90
Figura 55. Letra a reconocer.....	90
Figura 56. Botón Captura imagen.....	91
Figura 57. Captura de la imagen.....	91
Figura 58. Imagen tratada en interfaz de aplicación.....	92
Figura 59. Respuesta desplegada en la interfaz de la aplicación.....	92
Figura 60. Función de activación anti simétrica.....	98
Figura 61. Función de activación no simétrica.....	99

**INDICE DE TABLAS**

	Pág.
Tabla 1. Características de Android.....	27
Tabla 2. Letras y números del alfabeto en binario.....	60
Tabla 3. Tipos de función de activación.....	72
Tabla 4. Tabla de resultados de pruebas letra A.....	101

**INDICE DE ECUACIONES**

	Pág.
Ecuación 1. Modelo de una neurona.....	53
Ecuación 2. Método de la luminancia.....	57
Ecuación 3. Salida de cada neurona.....	61
Ecuación 4. Función de activación sigmoide.....	61
Ecuación 5. Función Sigmoidal.....	72
Ecuación 6. Función Gaussiana.....	72

## GLOSARIO

IOS	I Operative System
RAM	Random-access memory
SDK	software development kit
APK	Application PacKage File
J2ME	Java 2 Micro Edition
JRE	Java Runtime Environment
PHP	Hypertext Preprocesso
MYSQL	Structured Query Language
HTTPS	Hypertext Transfer Protocol Secure
URL	uniform resource locator
TCP/IP	Transmission Control Protocol
MIME	Multipurpose Internet Mail Extension
MLP	MultiLayer Perceptrons
SLP	SimpleLayer Perceptrons
LMS	Least Mean Square
RGB	Red, Green, Blue

## RESUMEN

Este trabajo se basa en las redes neuronales y la forma de encontrar la arquitectura más óptima, que sea capaz de aprender y devolver una respuesta correcta; para llegar al objetivo de reconocer letras y números. Se realizó dos procedimientos, el tratamiento de la imagen y el entrenamiento de la red. El tratamiento de la imagen consiste en convertirla a escala de grises, luego a una imagen de blancos y negros, y por ultimo reducir la imagen capturada a una matriz de 40x40 pixeles teniendo en cuenta que el color de fondo debe ser un color oscuro sin brillo, de esta manera se obtiene un arreglo de 1600 bits. Para el entrenamiento de la red se utilizaron dos modos, el modo aprender y el modo genera pesos, para aprender primero se realiza el tratamiento de la imagen y el resultado se envía al servidor de base de datos ubicado en la nube, en donde será almacenado, el modo genera pesos, como su nombre lo dice, genera los pesos de la letra según las que se encuentren almacenadas en la base de datos. Para el modo reconocer, primero se realiza el tratamiento de la imagen, el resultado se envía al servidor de inteligencia Artificial ubicado en la nube, el servidor obtiene los 1600 bits y realiza el procesamiento de la red mediante el algoritmo de back-propagation, este toma los valores de pesos almacenados en la base de datos realiza la comparación con los valore de la entrada hasta obtener una respuesta, ésta respuesta se enviará al dispositivo móvil desplegándose en la pantalla. Los servidores se encuentran en la nube debido a que los recursos del dispositivo móvil para este procesamiento son limitados.

**Palabras clave:** Redes neuronales, Back-Propagation, cloud, Android, inteligencia Artificial.

## ABSTRACT

This work is based on neural networks and how to find the optimal architecture that is able to learn and return a correct answer; to reach the goal of recognizing letters and numbers. Two methods of image processing and network training was performed. The image processing is to convert it to grayscale, then to a black and white image, and finally reduce the captured image to a matrix of 40x40 pixels taking into account that the background color should be a dark color dull in this way an array of 1600 bits is obtained. For network training two modes are used, how to learn and how generates pesos, to learn first image processing and the result is sent to the server database located in the cloud is done, where it will be stored the mode generates pesos, as its name says, it generates the weights of the letter as the ones stored in the database. To recognize how, first the image processing is performed, the result is sent to the server located Artificial intelligence in the cloud, the server gets the 1600 bits and performs the processing of the network using the back- propagation algorithm, this takes the values of weights stored in the database performs compared to input values for a response, this response will be sent to the mobile device being displayed on the screen. The servers are in the cloud because mobile resources for this processing are limited.

**Keywords:** Neural Networks, Back- Propagation, cloud, Adroid, Artificial Intelligence.

## CAPÍTULO I

### DESCRIPCION DEL PROYECTO

#### 1.1 Introducción.

##### 1.1.1 Antecedentes.

Las tecnologías móviles se consideran el sector de mayor auge dentro del ámbito tecnológico actual. Es por ello que un gran número de novedades tecnológicas se centran en la telefonía móvil.

El móvil, hoy en día, ya no se considera un mero dispositivo para realizar llamadas, sino que posee múltiples usos cotidianos como puede ser el envío de mensajes, conexión a internet, reproducción de música, etc. En cierto modo se ha convertido en una especie de ordenador personal. Aunque en la actualidad, se desarrollan nuevas tecnologías para la integración de las personas con discapacidad para así alcanzar una sociedad "plural" y "sin brechas digitales".

Según avanzan las tecnologías móviles y éstos dispositivos comienzan a funcionar como sistemas cada vez más complejos, el esfuerzo para que los discapacitados puedan acceder a estas tecnologías es cada vez mayor. Para ello, los fabricantes de móviles utilizan, por ejemplo, tecnologías de "reconocimiento automático del habla" y "conversión de texto a voz", siendo posible, entre otras cosas, facilitar la operación con el dispositivo a invidentes y su uso como sintetizadores de voz para personas con discapacidad auditiva. Con el tiempo, estas tecnologías han ido mejorando y se ha

conseguido una mayor interacción en el uso del dispositivo móvil y los discapacitados, como por ejemplo la consulta de la agenda. No obstante, uno de los mayores retos en la actualidad para la integración de las personas ciegas, supone desarrollar un método de lectura de la pantalla móvil similar al de las pantallas táctiles actuales.

### **Dispositivos para discapacidad Visual:**

Recientemente se ha presentado un móvil, el B-Touch de Zhenwei You, cuya pantalla es táctil-Braille e incluye también sistemas de reconocimiento de voz, lector de textos y escáner de objetos.



Figura 1. B-Touch de Zhenwei You.

En el 2004, Telefónica Móviles, ONCE y Owasys lanzaron un teléfono móvil para personas ciegas, el Owasys 22C, éste dispositivo comunica al usuario en forma de locución los mensajes recibidos, las llamadas perdidas, así como la persona que llama. También se puede conocer por voz el estado del teléfono (cobertura, batería, hora y fecha, etc.) o el contenido de un mensaje de texto antes de enviarlo.



Figura 2. Owasy 22C.

En esta línea existe una gran variedad de móviles "easy to use", diseñados para ser utilizados de forma muy sencilla y con funcionalidades básicas, para realizar llamadas y enviar mensajes.

Además, los fabricantes de móviles ponen a disposición de personas con discapacidad visual dispositivos móviles convencionales que incorporan nuevas funcionalidades para facilitar el manejo del móvil, como pueden ser: elección individual de un tono diferente por cada persona, teclados más espaciosos, teclas con relieve, ajuste del tamaño y claridad de la pantalla, e indicador de batería sonoro.

Actualmente, la marca de móvil más popular entre este tipo de usuarios es Nokia por la variedad de aplicaciones que incorpora. Apple ha realizado también desarrollos para acercar el uso del iPhone a las personas invidentes, como la tecnología VoiceOver, que "lee" lo que hay en pantalla en cada momento.

### Dispositivos para discapacidad Auditiva:

Las personas sordas o con discapacidad auditiva, aunque están algo más apartadas de las funcionalidades básicas de un teléfono, por lo que también son objeto para el desarrollo de nuevos proyectos, que intentan ofrecer mayores oportunidades, como MobileASL. Se trata de un software desarrollado por estudiantes y profesores de la Universidad de Washington que facilita la comunicación a través de signos con el móvil.



Figura 3. Una versión hipotética de una video llamada usada por la ASL.

El T-Loop de Vodafone, que es un lazo inductivo que permite a personas con discapacidad auditiva mantener conversaciones sin interferencias.

Vodafone España anunció la disponibilidad del lazo inductivo T-Loop con el teléfono móvil Vodafone 710. De tamaño reducido, este accesorio hace compatible el audífono y el teléfono móvil para que las personas con discapacidad auditiva puedan mantener una conversación sin interferencias.

El lazo inductivo T-Loop cuenta con funcionalidad manos libres, botón de encendido y apagado, indicador luminoso, micrófono incorporado y control de volumen desde el accesorio. Su tamaño facilita el manejo y permite controlar algunas funciones y utilizar el móvil con total normalidad. El teléfono Vodafone 710, es un dispositivo 3G que permite a los usuarios conectarse a Internet a gran velocidad sin necesidad de tener una conexión ADSL o WIFI. Este teléfono de última generación cuenta además con otras funcionalidades como el reproductor MP3, videollamada, cámara de 1,3 Megapixels o bluetooth.



Figura 4. T-Loop de Vodafone.

Apple por otra parte propone varias maneras de comunicarse con las características de iOS como video llamadas facetime, mensajes de texto ilimitados y tecnologías de apoyo, tales como subtítulos y el audio mono ayudará a disfrutar de su contenido.

FaceTime permiten comunicarse en más de un sentido. Atrapa a todos los gestos y la expresión facial. Gracias a sus servicios de video de alta calidad y velocidad de cuadros rápida, FaceTime es ideal para las personas que se comunican con lenguaje de señas.



Figura 5. FaceTime.

Los subtítulos Permite ver películas, programas de televisión y podcasts con subtítulos. Las leyendas aparecen en la pantalla en letras blancas fáciles de leer sobre un fondo negro.



Figura 6. Imagen de subtítulos en un iPhone.

iMessage permite iniciar una conversación - o mantenerlo en marcha - sin necesidad de decir o escuchar una palabra. Envía mensajes ilimitados a cualquier persona en un iPhone, iPad, iPod touch o Mac.



Figura 7. Mensajes con iMessage.

Sin embargo Android no se queda atrás, con Orange lanzan una aplicación específica para personas con discapacidad auditiva, Melovibe, que mejora la experiencia con su terminal móvil al permitir la asociación de una vibración personalizada (o melodía vibratoria) a cada uno de los contactos de su agenda.

Melovibe se convierte así en una alternativa real a la llamada tradicional para las personas con discapacidad auditiva y, en general, todos aquellos usuarios que necesitan utilizar a menudo su teléfono en modo de vibración.



Figura 8. Icono de Melovibe.

Otra aplicación de Android que se puede encontrar en GooglePlay es Deaf – Hearing Chat, la cual permite la comunicación entre sordos y oyentes, que convierte el habla en texto y da nuevas posibilidades para una comunicación de pleno valor, está compuesto de dos dispositivos Android, conectados por Bluetooth.

Uno de los dispositivos (teléfono o tableta), llamado Device-D (DeafPerson), lo tiene la persona sorda. El segundo dispositivo (teléfono o tableta), llamado Device-H (HearingPerson), lo tiene la persona oyente.



Figura 9. Deaf - Hearing. Aplicación Android.

Existen otras aplicaciones para Android como Magic Glove, que consiste en un guante con sensores que permite traducir los gestos del lenguaje de señas de la mano en comandos y texto para controlar el dispositivo móvil. Magic Glove es un guante provisto de sensores de movimiento y un controlador que es capaz de convertir gestos de la lengua de signos. De modo que Magic Glove podrá funcionar utilizando la cámara frontal del móvil.



Figura 10. Magic Glove

### 1.1.2 Justificación e Importancia.

De acuerdo a la constitución de la República del Ecuador, en el Título II, artículo 11, numeral 2, “Todas las personas son iguales y gozarán de los mismos derechos, deberes y oportunidades”, es por eso que es una prioridad incluir a éstas personas al desarrollo tecnológico.

El ámbito económico ha sido un factor importante para que éstas personas con discapacidad sean discriminadas por la sociedad, pero gracias a la creciente conciencia social se ha logrado incluir a personas con discapacidad auditiva en trabajos dignos, creando espacios y tecnología para ellos, ofreciéndoles así una vida digna, y ocupando el lugar en la sociedad que se merecen.

Al ser el lenguaje de señas desconocido para la mayoría de la gente, este proyecto se enfocó en ayuda social e inclusión a la tecnología, ofreciéndoles así una aplicación para dispositivos móviles Android, para uso cotidiano, de ésta manera, las personas

con discapacidad auditiva podrán comunicarse con personas que no conocen su lenguaje.

### **1.1.3 Alcance.**

Se realizó el diseño e implementación de una aplicación para un dispositivo móvil Android que permite la interrelación de los sordos con el resto de la gente.

La aplicación recibe datos de señas del abecedario y los números del uno al diez mediante la cámara del teléfono, luego convierte las señas en texto.

Se realizaron las pruebas pertinentes tanto en simuladores como físicas de manera que se garantiza la calidad de servicio brindada por la aplicación.

### **1.1.4 Objetivos.**

#### **1.1.4.1 General.**

- Desarrollar e implementar una aplicación que traduzca el abecedario y los números del uno al diez del lenguaje de señas a texto para ayuda de discapacitados auditivos mediante dispositivos móviles Android.

#### **1.1.4.2 Específicos.**

- Determinar las necesidades para una persona con discapacidad auditiva.
- Investigar el lenguaje de programación para Android.
- Diseñar los módulos que integran el software.
- Desarrollar e implementar la aplicación sobre dispositivos móviles comerciales.

### **1.2 Estudio de las necesidades de las personas con discapacidad auditiva.**

Las principales necesidades de las personas con discapacidad son:

- Las personas con alguna discapacidad tienen limitaciones en un entorno familiar o médico-asistencial.
- Dificultad el aprendizaje y la adquisición de conocimientos
- Carencia de independencia y autonomía en muchos aspectos, y por ende, la incapacidad de formar una familia, a quién cuidar, educar y proteger.
- Es importante resaltar y compartir a los padres que viven con algún hijo con discapacidad la urgencia de verlos como iguales y estar orgullosos de sus logros y objetivos alcanzados.
- Obtener un trabajo e ingresos monetarios propios para lograr la independencia y autorrealización de la persona. Puesto que una seguridad financiera, desencadenará confianza y seguridad en otros aspectos de su vida y plenitud como ser humano.

- Las características comunicativas de las personas sordas y el uso del canal visual como fuente importante de recepción de información, provoca que las personas sordas encuentren barreras e inadaptaciones del medio en su vida

Según la Organización Mundial de la Salud (OMS), Discapacidad es un término general que abarca las deficiencias, las limitaciones de la actividad y las restricciones de la participación.

- Una deficiencia es toda pérdida o anormalidad de una estructura o función psicológica, fisiológica o anatómica.
- Una discapacidad es toda restricción o ausencia (debida a una deficiencia) de la capacidad de realizar una actividad en la forma o dentro del margen que se considera normal para un ser humano.
- Una minusvalía es una situación desventajosa para un individuo determinado consecuencia de una deficiencia o una discapacidad, que limita o impide el desempeño de un rol que es normal en su caso (en función de su edad, sexo o factores sociales y culturales).

Por consiguiente, la discapacidad es un fenómeno complejo que refleja una interacción entre las características del organismo humano y las características de la sociedad en la que vive.

### **1.2.1 Tipos de discapacidad.**

En general la población percibe la discapacidad como una condición permanente. Sin embargo, hay discapacidades temporales y discapacidades permanentes. Por

ejemplo, fracturarse el brazo o perder grados de visión por la exposición de los ojos a algún agente químico nocivo, pueden ocasionar discapacidad temporal, y por otra parte, también hay niveles de discapacidad: leve, moderada o severa.

De manera general se puede clasificar a la discapacidad en los siguientes tipos:

- Discapacidad Psíquica (Mental, Intelectual, Cognitiva), que es una disminución en las habilidades cognitivas e intelectuales del individuo.
- Discapacidad Física (motora, motriz).- Quienes la padecen ven afectadas sus habilidades motrices, por ejemplo la Parálisis Cerebral.
- Discapacidad Sensorial.- Aquí se encuentran aquellas discapacidades relacionadas con la disminución de uno o varios sentidos. Dentro de esta clasificación tenemos:
  - Discapacidad Auditiva
  - Discapacidad Visual
  - Discapacidad Multisensorial

#### **1.2.1.1 Discapacidad Psíquica.**

La Discapacidad Psíquica, intelectual. Es un término utilizado cuando una persona no tiene la capacidad de aprender a niveles esperados y funcionar normalmente en la vida cotidiana.

La causa puede ser una lesión, enfermedad o un problema en el cerebro. Algunas de las causas más frecuentes de discapacidad intelectual, como el síndrome de Down, el síndrome alcohólico fetal, el síndrome X frágil, afecciones genéticas, defectos

congénitos e infecciones, ocurren antes del nacimiento. Otras causas ocurren durante el parto o poco después del nacimiento. En otros casos, las causas de discapacidad intelectual no se presentan sino hasta cuando el niño es mayor, tales como lesiones graves de la cabeza, accidentes cerebro-vasculares o ciertas infecciones.

#### **1.2.1.2 Discapacidad Física.**

La visión que se tiene de las personas con discapacidad física ha variado con el paso de los años, antiguamente las personas con discapacidad física eran relegadas a su hogar, sin posibilidades de socializar y mucho menos a opciones laborales, esta condición día a día va mejorando, y los gobiernos van tomando mayor conciencia de implementar políticas tendientes a que las ciudades sean lugares más amables hacia la persona que vive con una discapacidad.

Existen diversas causas por las cuales se presenta la discapacidad física; factores congénitos, hereditarios, cromosómicos, por accidentes o enfermedades degenerativas, neuromusculares, infecciosas o metabólicas entre muchas.

Tipos de Discapacidad Física:

- Lesión medular
- Esclerosis múltiple
- Parálisis cerebral
- Mal de Parkinson
- Espina bífida
- Distrofia muscular

- Acondroplasia
- Albinismo

### 1.2.1.3 Discapacidad Sensorial.

#### 1.2.1.3.1 Discapacidad Auditiva.

El término "discapacidad auditiva" es un término genérico usado para describir todas las pérdidas de audición.

Normalmente, las ondas sonoras atraviesan el oído externo hasta llegar al tímpano, o membrana timpánica, que inicia su vibración y pone en movimiento la cadena de huesecillos (martillo, yunque y estribo). Estos huesecillos transfieren la energía al oído interno, en donde los fluidos que se encuentran en su interior entran en movimiento, provocando que las células ciliadas transformen estas vibraciones en impulsos eléctricos, que se transmiten a través de las fibras nerviosas auditivas al cerebro.

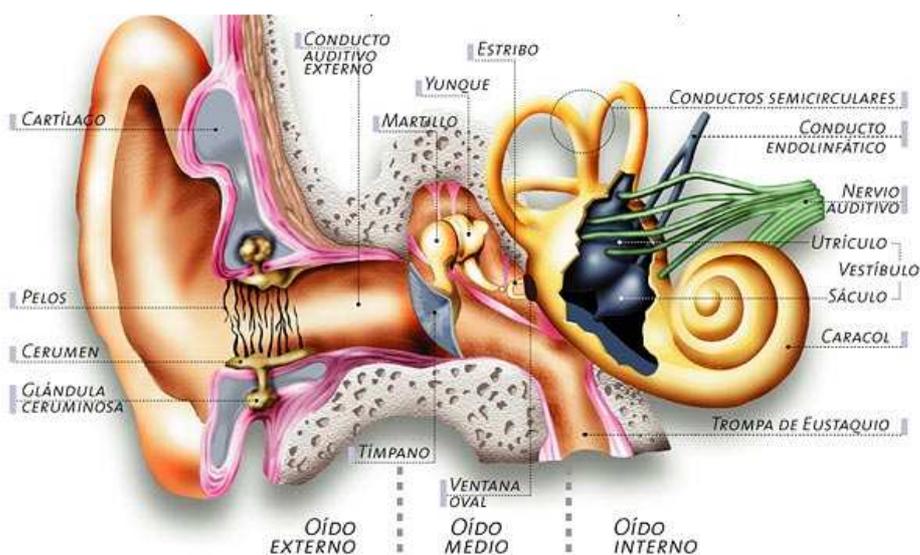


Figura 11. Estructura del oído.

Se considera las deficiencias auditivas como aquellas alteraciones cuantitativas en una correcta percepción de la audición, es por esto que existen varios tipos de pérdida de audición:

- Hipoacusia: Pérdida parcial. disminución de la capacidad auditiva que permite adquirir el lenguaje oral por la vía auditiva.
- Sordera (cofosis): pérdida total de la audición y el lenguaje se adquiere por la vía visual.

#### **1.2.1.3.2 Discapacidad Visual.**

La gran mayoría de los niños con problemas de visión reciben su educación en centros de régimen ordinario donde, con el apoyo adecuado, pueden obtener una buena escolaridad. Sin embargo, en algunos casos las dificultades de estos niños no se reconocen o no son comprendidas.

#### **1.2.1.3.3 Discapacidad Multisensorial.**

Los niños y jóvenes que tienen una combinación de dificultades visuales y auditivas pueden ser denominados "**discapacitados multisensoriales**". Con frecuencia también se les llama sordo-ciegos.

En el Ecuador existen 1'653.000 personas con discapacidad registradas en el Consejo Nacional de Discapacidades (CONADIS), en torno a las cuales se han implementado algunas Políticas Sociales, se han reconocido derechos y elaborado

determinadas normativas. Si bien estas conquistas pretenden mejorar el nivel de vida de las personas con discapacidad, en términos de integración social y acceso a beneficios, su cumplimiento y ejecución es relativa.

La discapacidad, además ha sido tratada por diversos enfoques disciplinarios como problema de una persona o individuo con limitaciones físicas y/o mentales, prescindiendo de las relaciones sociales que se tejen a su alrededor. Así desde la ciencia médica o desde la visión ética/religiosa, la discapacidad es asumida por un lado, como un todo homogéneo y por otro lado referida únicamente a las personas con limitaciones físicas y/o mentales.

Si bien la discapacidad afecta a todos los miembros del grupo familiar, particularmente las madres que se encargan del cuidado y rehabilitación de las personas especiales, en este caso de niños. Esta situación constituye para las madres situaciones y experiencias pocas veces reconocidas y significando para muchas madres, impactos emocionales, desvalorización y procesos de exclusión.

Desde el sentido común, en términos de derechos, se asume social y éticamente, como única prioridad la rehabilitación e integración, de las personas con limitaciones físicas y/o mentales.

Sin embargo, las necesidades no solo hacen referencia a la "insuficiencia de ingresos", a las "privaciones materiales", sino también a las restricciones de oportunidades en tanto posibilidad de ser y hacer, la homogeneización de las diferencias sociales, la falta de políticas integrales y las limitaciones de la normatividad en torno a la discapacidad.

En el contexto ecuatoriano, la discapacidad entendida como un campo en el que convergen una multiplicidad de sujetos, intereses, conocimientos, necesidades, se

presenta como un espacio de ambigüedades y de conflictos tanto para las personas con limitaciones como para sus familiares.

### **1.2.2 Derechos humanos de las personas con discapacidades distintas.**

Los derechos humanos son universales, políticos, civiles, económicos, sociales y culturales, pertenecen a todos los seres humanos, incluyendo a las personas con distintas discapacidades. Las personas discapacitadas deben gozar de sus derechos humanos y libertades fundamentales en términos iguales con otros en la sociedad, sin discriminación de ningún tipo.

Los derechos humanos para las personas discapacitadas incluyen los siguientes derechos indivisibles, interdependientes e interrelacionados.

- El derecho a la no distinción, exclusión, restricción o preferencia basada en su condición de discapacidad basada en el efecto de deteriorar el disfrute de los derechos humanos y libertades fundamentales.
- El derecho a la igualdad de oportunidades.
- El derecho a una completa igualdad y protección ante la Ley.
- El derecho a un alto estándar de salud para un tratamiento médico, psicológico y funcional, de igual manera a una rehabilitación médica y social y otros servicios necesarios para el máximo desarrollo de las capacidades, habilidades y auto-confianza
- El derecho a trabajar, de acuerdo con sus capacidades, a recibir salarios que igualitarios que contribuyan a un estándar de vida adecuado.
- El derecho a ser tratado con dignidad y respeto.

### **1.3 Lenguaje de señas.**

#### **1.3.1 Declaración Universal de los Derechos Lingüísticos.**

“El ejercicio de derechos lingüísticos individuales solo puede ser efectivo si se respetan los derechos colectivos de todas las comunidades y de los grupos lingüísticos que forman parte de un país”.

El lenguaje de las señas es un lenguaje natural de expresión gesto espacial y de percepción visual, también puede ser táctil para algunas personas con sordoceguera, gracias a la lengua de señas las personas con deficiencias auditivas pueden establecer un canal de comunicación con su entorno social, el cual puede estar conformado por otras personas con las mismas deficiencias o por cualquier persona que conozca la lengua de señas empleada.

Para entender las necesidades de las personas con discapacidades auditivas hay que tener en claro la diferencia entre lenguaje y lengua.

#### **Lenguaje:**

El lenguaje es la capacidad inherente de los seres humanos para comunicarse por medio de signos (usualmente secuencias sonoras, pero también gestos y señas, así como signos gráficos) y expresarlas mediante un sistema de signos llamado lengua.

#### **Lengua:**

Hablar de lengua es hablar de idioma, es decir, es un conjunto de símbolos de convención a la que una comunidad le otorga un significado común, cada sociedad

estructura un sistema de símbolos particular relacionado con su forma propia de ver el mundo.

Así la lengua de las señas es una lengua natural de expresión y percepción visual gracias a la cual las personas sordas pueden establecer un canal de comunicación con su entorno, ya sea conformado por otros individuos sordos o por cualquier persona que conozca la lengua de señas empleada.

Por otro lado, la lengua de señas constituye la lengua primera o lengua nativa para los niños sordos la misma que les asegura el desarrollo de la capacidad humana del lenguaje, la comunicación y el desarrollo intelectual.

La lengua de señas se utiliza casi exclusivamente entre personas sordas, su origen es tan antiguo como el de la lengua oral (o incluso más antigua), además ha sido y sigue siendo empleada por comunidades oyentes.

Así como las lenguas orales no son universales. A la lengua de señas le ocurre lo mismo, cada país tiene su propia lengua de señas, Por ello existe de la Lengua de Señas Ecuatoriana (LSEC); y hasta como les sucede a las lenguas orales, la lengua de señas de un mismo país también tiene dialectos, es decir, variaciones lingüísticas debido las diferencias culturales, de identidad y geografía.

En el año 1982, la entonces Sociedad de Sordos Adultos “Fray Luis Ponce de Leon” y actualmente Asociación de Personas Sordas de Pichincha firma un acuerdo con la Fundación Interamericana y se establece un convenio de trabajo mutuo entre estas dos organizaciones. A este acuerdo se le llamó Proyecto “Mano a Mano”. Dentro de los objetivos de este proyecto estaba la creación de una guía básica o diccionario de lengua de señas ecuatoriana que recogiera las señas básicas utilizadas por la Comunidad Sorda Ecuatoriana. Después de un arduo trabajo en el que también

participaron las personas sordas de Guayaquil, finalmente en el año 1987 se publicó el libro-diccionario “Lenguaje de Señas: Guía Básica Sobre una Comunicación Especial Tomo I”.

En Quito, el 18 de octubre de 2012, la Federación Nacional de Personas Sordas del Ecuador (Fenasec) pone a disposición de las comunidades sorda y oyente del país, el primer Diccionario Oficial de Lengua de Señas Ecuatoriana, las Guías para Padres y Docentes de Personas Sordas y la Guía para Intérpretes de Lengua de Señas Ecuatoriana.



Figura 12. Diccionario Oficial de Lengua de Señas Ecuatoriana y Guías para Padres y Docentes de Personas Sordas y la Guía para Intérpretes de Lengua de Señas Ecuatoriana.

La producción del diccionario, que está en circulación a escala nacional desde el 15 de octubre del 2012, cuenta con el aval del Ministerio de Educación y fue impulsado por la Vicepresidencia de la República de Ecuador, con el financiamiento de la producción.

El diccionario, ilustrado y a color consta de dos tomos que recogen alrededor de 4.000 señas, ilustración, descripción y un anexo de 300 señas que incluye nombres de países, continentes, provincias y ciudades del Ecuador, así como de las asociaciones locales, números, entre otros temas. Este documento además tiene un DVD que presenta toda la información del diccionario en formato digital e interactivo.

La elaboración de este diccionario es una iniciativa de la comunidad sorda ecuatoriana para romper las barreras de comunicación. El Diccionario constituye una herramienta para fortalecer el derecho de la persona sorda para usar su lengua natural.

El diccionario y las guías estarán a disposición de forma gratuita en Fenasec para los centros educativos y las asociaciones de personas sordas, padres, docentes e intérpretes en las distintas ciudades del país.

### **1.3.2 Origen de las lenguas de señas.**

Aun cuando hoy en día las lenguas de señas se utilizan casi exclusivamente entre personas con sordera, su origen es tan antiguo como el de las lenguas orales o incluso más en la historia de la Humanidad, y también han sido y siguen siendo empleadas por comunidades de oyentes. De hecho, los indios americanos de la región de las Grandes Llanuras de América del Norte, usaban una lengua de señas para hacerse entender entre etnias que hablaban lenguas muy diferentes con fonologías extremadamente diversas, el sistema estuvo en uso hasta mucho después de la conquista europea.

Otro caso, también amerindio, se dio en la isla de Manhattan, donde vivía una tribu única en la que un gran número de sus integrantes eran sordos, debido a la herencia de

desarrollo de un gen dominante, y que se comunicaban con una lengua gestual, un caso similar se desarrolló en la isla de Martha's Vineyard al sur del estado de Massachusetts, donde debido al gran número de sordos se empleó una lengua de señas que era de uso general también entre oyentes, hasta principios del siglo XX.

Pese a esto, no existen referencias documentales sobre estas lenguas antes del siglo XVII, los datos que se poseen tratan, sobre todo, de sistemas y métodos educativos para personas sordas.

En el año 1620 Juan de Pablo Bonet publica su Reducción de las letras y Arte para enseñar a hablar los Mudos, considerado como el primer tratado moderno de Fonética y Logopedia, en el que se proponía un método de enseñanza oral de los sordos mediante el uso de señas alfabéticas configuradas unimanualmente, divulgando así en toda Europa, y después en todo el mundo, el alfabeto manual, útil para mejorar la comunicación de los sordos y mudos.

En 1817 Gallaudet fundó la primera escuela de la nación para las personas sordas, en Hartford, Connecticut, y Clerc se convirtió en el primer maestro sordo de lengua de señas de los Estados Unidos. Pronto las escuelas para las personas sordas empezaron a aparecer en varios estados. Entre ellos la Escuela de Nueva York que abrió sus puertas en 1818. En 1820 otra escuela se abrió en Pennsylvania, y un total de veintidós escuelas se habían establecido a lo largo de los Estados Unidos por el año 1863.

### 1.3.3 El Abecedario de acuerdo al Lenguaje de Señas.

En el país no existe un sistema estandarizado de lengua de señas. Estas difieren entre regiones e inclusive ciudades, por lo que los sordos se adaptan cuando conocen a personas que no se comunican en su mismo 'idioma'.

La lengua materna de los sordos es la de señas, cuya composición gramatical es diferente a la construcción de oraciones del español. Pero, pese a que existen modelos internacionales, Ecuador no tiene uno que incluya los modismos y frases características del país.

La lengua de señas es similar en un 80% a nivel nacional, pero el restante 20% puede provocar ligeros problemas de comunicación. Por ejemplo, existen tres maneras diferentes de decir arroz y todas son válidas.

En casi todo el mundo se usan un grupo de señas para representar las letras del alfabeto con el que se escribe la lengua oral del país. Es esto lo que se denomina alfabeto manual o alfabeto dactilológico. En el caso de los países de habla hispana, donde se usa el alfabeto latino, las personas sordas usan un mismo alfabeto manual, común para todos los países (con algunas variaciones de índole menor en la forma de algunas letras). En el Reino Unido se usa un alfabeto manual distinto, en implica la utilización de ambas manos. En los países que usan alfabetos distintos al latino tal como el hebreo, árabe, amhárico, entre otros, existen diferentes formas de representación entre las personas sordas. En este mismo orden de ideas, un sistema diferente se aplica en los países donde se usan sistemas de escritura no alfabéticos como es el caso de Japón y China.

A continuación se muestra la representación manual de los números del uno al diez y las letras del abecedario utilizado como recurso de comunicación para deletrear algunas palabras o como instrumento en la educación de personas sordas en nuestro país.

Las señas de la figura 13 muestran la letra del alfabeto correspondiente tal como se ve cuando las ejecuta una persona frente a nosotros, con la mano derecha.

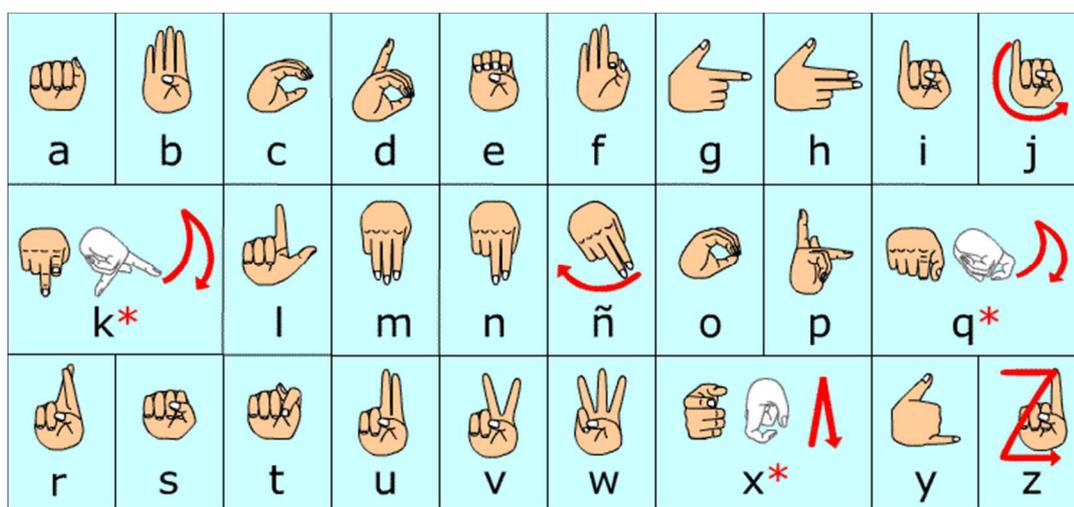


Figura 13. Abecedario de acuerdo al lenguaje de las señas.



Figura 14. Números del uno al diez de acuerdo al lenguaje de las señas.

## CAPÍTULO II

### ESTUDIO DE LA ARQUITECTURA DE ANDROID

#### 2.1 Características.

Las características principales de Android son las siguientes:

- Android es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas.
- Es un software de código libre, lo que significa que no solo pueden mejorarlo los desarrolladores de Google, sino que también se nutre de las aportaciones de desarrolladores externos.
- El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.
- No es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.
- Existen 700.000 de aplicaciones gratuitas y disponibles para la tienda de aplicaciones oficial de Android Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung.

### 2.1.1 Características de Software.

Las características principales del software se detallan a continuación en la siguiente tabla:

Tabla 1. Características de Android.

Característica	Descripción
Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D.
Almacenamiento	Posee una base de datos liviana SQLite, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+ y WiMAX.
Mensajería	Soporta SMS y MMS, mensajería de texto y Android Cloud to Device Messaging Framework (C2DM).
Navegador WEB	El navegador web incluido en Android está basado en el motor de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
Soporte para Streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.

<b>Característica</b>	<b>Descripción</b>
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse usando el plugin de Herramientas de Desarrollo de Android.
Google Play	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-Táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel.
Bluetooth	Fue agregado en la versión para el envío de archivos y la exploración del directorio telefónico.
Video Llamada	Android soporta videollamada a través de Google Talk desde su versión HoneyComb.
Multitarea	Multitarea real de aplicaciones quiere decir que las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
Tethering	Permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2).

## 2.2 Arquitectura de Android.

La arquitectura de Android está formada por varias capas como se muestra en la figura 15, estas capas facilitan al desarrollador la creación de aplicaciones. Además, ésta distribución permite acceder a las capas más bajas mediante el uso de librerías, para que una aplicación haga uso de los componentes de hardware de los teléfonos.

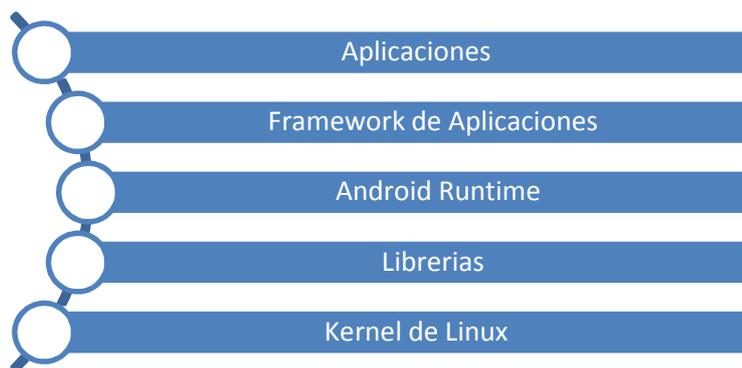


Figura 15. Estructura de la Arquitectura de Android.

Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como pila.

### 2.2.1 Diagrama de la Arquitectura de Android.

La arquitectura de Android está diseñada como se indica en la figura 16.

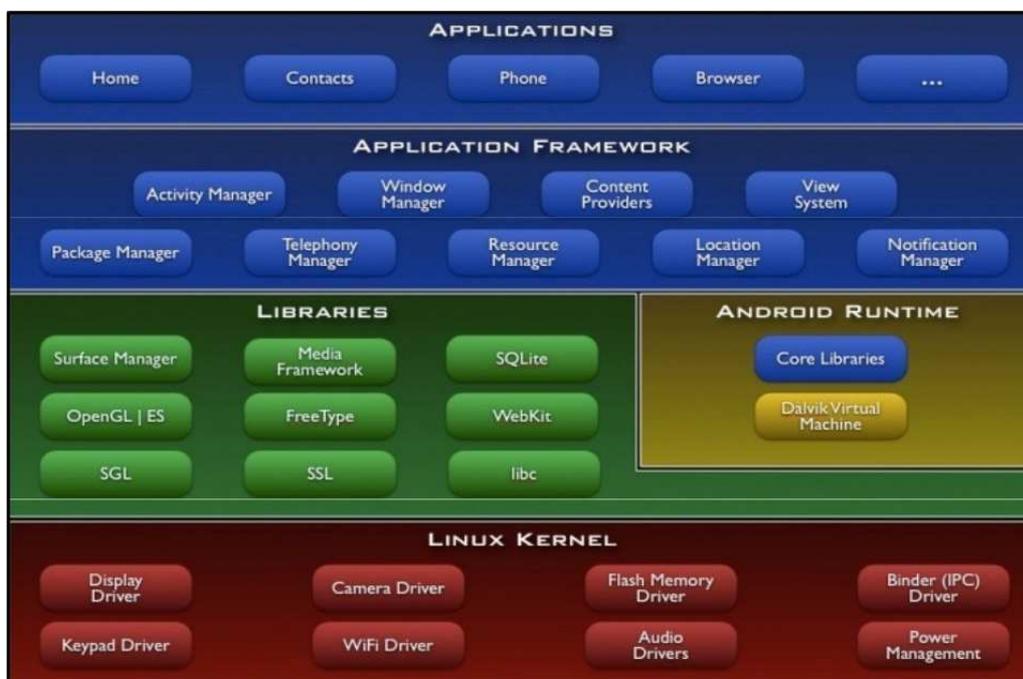


Figura 16. Arquitectura del sistema.

### 2.2.1.1 Linux Kernel.



Figura 17. Arquitectura del Sistema Linux Kernel.

Es el núcleo del sistema operativo. Android está basado en el kernel de Linux versión 2.6, similar al que puede incluir cualquier distribución de Linux, como Ubuntu, solo que adaptado a las características del hardware en el que se ejecutará Android, es decir, para dispositivos móviles. El núcleo actúa como una capa de abstracción entre el hardware y el software.

El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores. Para cada elemento de hardware del teléfono existe un controlador (o driver) dentro del kernel que permite utilizarlo desde el software. El kernel también se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, etc.) y del sistema operativo en sí: procesos, elementos de comunicación (networking), etc.

### 2.2.1.2 Librerías.



Figura 18. Arquitectura del Sistema Librerías.

La siguiente capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura de hardware específica del teléfono. Éstas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma más eficiente.

Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- Librería lib: Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- Librería Surface Manager: Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- OpenGL/SL y SGL: Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/SL maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, SGL proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.

- Librería Media Libraries: Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.).
- FreeType: Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- Librería SSL: Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- Librería SQLite: Creación y gestión de bases de datos relacionales.
- Librería WebKit: Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

### 2.2.1.3 Android Runtime.



Figura 19. Arquitectura del Sistema Android Runtime.

El entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí se encuentran las librerías con funcionalidades habituales de Java así como otras específicas de Android.

El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las

aplicaciones se compilan una única vez y de ésta forma estarán listas para distribuirse con la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación

Entre las características más importantes tenemos:

- Core Libraries: Formado por un subconjunto de Apache Harmony. Utiliza el Kernel de Linux para la ejecución de aplicaciones (1 aplicación = 1 proceso = 1 DVM)
- Máquina virtual Dalvik: Permite ejecutar aplicaciones programadas en Java.

#### 2.2.1.4 Application Framework.



Figura 20. Arquitectura del Sistema Android Application Framework.

Esta capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik. Siguiendo el diagrama se encuentra:

- Activity Manager. Se encarga de administrar la pila de actividades de las aplicaciones así como su ciclo de vida.

- **Windows Manager.** Se encarga de organizar lo que se mostrará en pantalla. Básicamente crea las superficies en la pantalla que posteriormente pasarán a ser ocupadas por las actividades.
- **Content Provider.** Esta librería crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
- **Views.** Son los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.
- **Notification Manager.** Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado, ésta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.
- **Package Manager.** Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes.
- **Telephony Manager.** Esta librería permite realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
- **Resource Manager.** Esta librería permite gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o layouts.
- **Location Manager.** Esta librería permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.

- **Sensor Manager.** Esta librería permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
- **Cámara:** Esta librería permite hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.
- **Multimedia.** Esta librería permite reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

#### 2.2.1.5 Aplicaciones.



Figura 21. Arquitectura del Sistema Android Aplicaciones.

En ésta capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.

En ésta capa se encuentra la aplicación principal del sistema: Inicio (Home) o lanzador (launcher), ya que es la que permite ejecutar otras aplicaciones mediante una lista y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso widgets, que son también aplicaciones de esta capa.

También, en ésta capa se ubicarán las aplicaciones ‘preinstaladas’ y las desarrolladas por el desarrollador, aplicaciones escritas en lenguaje JAVA y aplicaciones incluidas por defecto:

- Cliente email
- Gestor SMS
- Navegador
- Contactos
- Android Market...

### **2.3 Sensores disponibles en Android.**

Android tiene los siguientes sensores disponibles:

- Sensor acelerómetro: medir aceleraciones por gravedad y cambios de movimiento.
- Sensor gravimétrico: medir la aceleración debida a la gravedad.
- Sensor giroscópico: detectar giros.
- Sensor de iluminación: ajustar iluminación pantalla.
- Sensor de aceleración lineal: medir aceleraciones descontando la gravedad.
- Sensor de campo magnético: brújula, detectar campos magnéticos.
- Sensor de orientación: indicar dirección a la que apunta el dispositivo.
- Sensor de presión: altímetro, barómetro.
- Sensor de proximidad: si hay un objeto a menos de 5 cm (al hablar por teléfono).
- Sensor de vector de rotación: detectar giros.

- Sensor de temperatura: evitar sobrecalentamientos.

## **2.4 Descripción de las herramientas utilizadas.**

### **2.4.1 Java.**

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

Java le permite jugar, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como, por ejemplo, cursos en línea, servicios bancarios en línea y mapas interactivos. Si no dispone de Java, muchas aplicaciones y sitios web no funcionarán. Por defecto, Java le notificará inmediatamente que hay nuevas actualizaciones listas para instalarse.

#### **2.4.1.1 Diseño y desarrollo.**

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android, al contrario de otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

#### **2.4.1.2 Aplicaciones.**

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o SDK provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

#### **2.4.1.3 Entornos de funcionamiento.**

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática.

#### **2.4.1.4 En dispositivos móviles y sistemas empotrados**

Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition), una versión del entorno de ejecución Java reducido y altamente optimizado, se ha producido toda una revolución en lo que a la extensión de Java se refiere.

Es posible encontrar microprocesadores diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard), teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de TV y otros pequeños electrodomésticos.

El modelo de desarrollo de estas aplicaciones es muy semejante a las applets de los navegadores salvo que en este caso se denominan MIDlets.

#### **2.4.1.4.1 En el navegador web.**

Desde la primera versión de Java existe la posibilidad de desarrollar pequeñas aplicaciones (Applets) en Java que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Estas mini-aplicaciones se ejecutan en una JVM que el navegador tiene configurada como extensión (plug-in) en un contexto de seguridad restringido configurable para impedir la ejecución local de código potencialmente malicioso.

El éxito de este tipo de aplicaciones no fue realmente el esperado debido a diversos factores, siendo quizás el más importante la lentitud y el reducido ancho de banda de las comunicaciones en aquel entonces que limitaba el tamaño de las applets que se incrustaban en el navegador. La aparición posterior de otras alternativas (aplicaciones web dinámicas de servidor) dejó un reducido ámbito de uso para esta tecnología, quedando hoy relegada fundamentalmente a componentes específicos para la intermediación desde una aplicación web dinámica de servidor con dispositivos ubicados en la máquina cliente donde se ejecuta el navegador.

Las applets Java no son las únicas tecnologías (aunque sí las primeras) de componentes complejos incrustados en el navegador. Otras tecnologías similares pueden ser: ActiveX de Microsoft, Flash, Java Web Start, etc.

#### **2.4.1.4.2 En sistemas de servidor.**

En la parte del servidor, Java es más popular que nunca, desde la aparición de la especificación de Servlets y JSP (Java Server Pages).

Hasta entonces, las aplicaciones web dinámicas de servidor que existían se basaban fundamentalmente en componentes CGI y lenguajes interpretados. Ambos tenían diversos inconvenientes (fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica).

#### **2.4.1.4.3 En aplicaciones de escritorio.**

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier PC.

En las primeras versiones de la plataforma Java existían importantes limitaciones en las APIs de desarrollo gráfico (AWT). Desde la aparición de la biblioteca Swing la situación mejoró substancialmente y posteriormente con la aparición de bibliotecas como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.

#### **2.4.1.4.4 Plataformas soportadas.**

Una versión del entorno de ejecución Java JRE (Java Runtime Environment) está disponible en la mayoría de equipos de escritorio. Sin embargo, Microsoft no lo ha incluido por defecto en sus sistemas operativos. En el caso de Apple, éste incluye una versión propia del JRE en su sistema operativo, el Mac OS. También es un producto

que por defecto aparece en la mayoría de las distribuciones de GNU/Linux. Debido a incompatibilidades entre distintas versiones del JRE, muchas aplicaciones prefieren instalar su propia copia del JRE antes que confiar su suerte a la aplicación instalada por defecto. Los desarrolladores de applets de Java o bien deben insistir a los usuarios en la actualización del JRE, o bien desarrollar bajo una versión antigua de Java y verificar el correcto funcionamiento en las versiones posteriores.

#### **2.4.1.5 Industria relacionada.**

Sun Microsystem, como creador del lenguaje de programación Java y de la plataforma JDK, mantiene fuertes políticas para mantener una especificación del lenguaje<sup>5</sup> así como de la máquina virtual<sup>6</sup> a través del JCP. Es debido a este esfuerzo que se mantiene un estándar de facto.

Son innumerables las compañías que desarrollan aplicaciones para Java y/o están volcadas con esta tecnología:

- La industria de la telefonía móvil está fuertemente influenciada por la tecnología Java.
- Los entornos de desarrollo Netbeans y Eclipse ha tomado un lugar importante entre la comunidad de desarrolladores Java.
- La fundación Apache tiene también una presencia importante en el desarrollo de bibliotecas y componentes de servidor basados en Java.
- IBM, BEA, IONA, Oracle,... son empresas con grandes intereses y productos creados en y para Java.

### 2.4.2 PHP.

PHP es un lenguaje de programación originalmente diseñado para el desarrollo de portales dinámicos. PHP es la sucesión de un producto anterior: PHP/FI (Personal Home Page / Forms Interpreter). Este fue creado en 1995 por Rasmus Lerdof, como un conjunto de scripts Perl para realizar tracking. Con el tiempo y al necesitar de mayor funcionalidad, se escribió una implementación en C que podía interactuar con base de datos, y permitió el desarrollo de aplicaciones dinámicas simples para web.

PHP/FI incluyó las mismas funcionalidades que PHP posee en la actualidad, con variables similares a las utilizadas en Perl, interpretación automática de variables y sintaxis embebida de HTML.

La versión 2.0 de PHP/FI se oficializó en 1997, cuando con alrededor de 50000 dominios ya lo utilizaba, aunque se mantuvo una largo tiempo en pruebas y en estado beta.

Sin embargo, en 1997 también se oficializó la salida de PHP 3.0 a su fase de pruebas abierta al público, siendo esta la primera versión PHP que trabajaba como la actual. Esta fue creada por Andi Gutmans y Zeev Suraski, tras una refactorización completa del código tras encontrar varios problemas dentro de PHP/FI 2.0. Con la ayuda del creador original, Rasmus Lerdof, se anunció que a PHP 3.0 como el sucesor de PHP/FI y el trabajo en este se detuvo casi completamente.

Una de las principales ventajas de PHP 3.0 fue su capacidad de extenderse, junto a su sólida estructura en su relación con diferentes bases de datos, protocolos y APIs (Application Programming Interface). Esto permitió a muchos desarrolladores el unirse a este trabajo conjunto y añadir nuevos módulos y funcionalidades, lo que

contribuyó a su gran éxito. Junto a esto, introdujo la sintaxis orientada a objetos y muchas nuevas innovaciones que lo convirtieron en un lenguaje poderoso y consistente.

También, se cambió el nombre de PHP para remover toda implicación con la persona, volviéndose esta un acrónimo recursivo: “PHP Hypertext Preprocessor”.

La evolución de PHP se realizó a partir de la versión PHP 3.0, que tras varios meses de prueba se oficializó en 1998. Ese mismo año, los creadores Andi Gutmans y Zeev Suraski se embarcaron en el proyecto de reescribir el núcleo de PHP, mejorando el rendimiento del mismo y las capacidades de la modulación de la base PHP. Con estas mejoras, se introdujo la versión 4.0 de PHP en 1999, junto con un amplio rango de nuevas funcionalidades, siendo finalmente presentado al mercado en mayo de 2000. Entre sus principales mejoras, permitió la interacción con un mayor rango de servidores Web, sesiones HTTP, seguridades y constructores de lenguaje.

La versión 5 de PHP, la cual se encuentra actualmente en el mercado, se publicó en 2004, tras una refactorización del núcleo de PHP para manejar nuevas funcionalidades basándose en un nuevo modelo de objetos.

Esta evolución permite a PHP ser uno de los lenguajes más confiables al momento del desarrollo Web. Esto debido a que presenta diferentes ventajas frente a sus competidores. Entre estas cabe destacar que es rápido, estable seguro, es fácil de entender y utilizar, y además es gratis.

Entre sus principales ventajas, cabe recalcar las siguientes:

- El código es interpretado directamente por el servidor, por lo que el usuario no necesita de ningún browser o tecnología especial para poder ver y utilizar las páginas PHP.
- Es simple de aprender y utilizar, especialmente cuando se posee conocimientos en C, lo que llevó a que sea especialmente utilizado en el desarrollo masivo del Internet.
- Es multiplataforma, y corre sobre casi todo sistema operativo, incluyendo Unix, Windows y Mac.
- No utiliza una gran cantidad de recursos, por lo que es rápido y no afecta el rendimiento de procesos concurrentes.
- Utiliza sistemas de extensiones modulares con gran cantidad de librerías para gráficos, XML, encriptación, etc. Además permite a los desarrolladores el programar sus propias extensiones y compilarlas en el ejecutable.
- Puede correr con servidores de interfaz como Apache, IIS, Roxen y THTTPD. En cuanto a base de datos, puede interactuar con casi todas (de manera nativa o mediante el uso de ODBC), incluyendo MySQL, MS SQL, Oracle y muchas otras.
- Posee gran cantidad de información y scripts gracias a la gran comunidad Open Source que trabaja sobre él.

Gracias a estas ventajas, la utilización de PHP como lenguaje de programación para desarrollo web fue creciendo paulatinamente en sus primeros años, considerándose de los más útiles y una de las razones para el rápido desarrollo del Internet. Sin embargo, en los últimos años, PHP ha ido decreciendo en su mercado debido a la aparición de nuevas tecnologías y necesidades. Sin embargo, PHP es

todavía uno de los lenguajes de programación más utilizados, solicitados y aceptados dentro de la comunidad de desarrolladores, poseyendo un 10.156% de popularidad para Septiembre de 2009.

En cuanto a un nivel de penetración dentro del Internet, en Octubre de 2008, PHP se encontraba dentro de una gran cantidad de sitios, con un 32.84% de sitios, con una ligera tendencia a la baja.

Esta tendencia a la baja ha provocado un nuevo enfoque a promover el trabajo conjunto de este lenguaje con tecnologías de rápido manejo, en proyectos como LAMP. Adicionalmente, se planeó una ofensiva de mercado, conjuntamente con la promoción del nuevo PHP 6.0, que conllevará una nueva refactorización del código y muchas nuevas funcionalidades.

PHP se maneja como un software libre bajo la licencia PHP, que no es compatible con la GNU General Public License, debido a las restricciones del término PHP.

### **2.4.3 MySQL.**

MySQL es el sistema motor de base de datos “Open Source” más popular del mundo, y uno de los motores de bases de datos más utilizados en el desarrollo de portales en Internet.

Desarrollado, mantenido y distribuido por SUN Microsystems, MYSQL es un sistema de manejo de base de datos, una colección de datos estructurados, que permite el almacenamiento y manejo de los mismos dentro de un sistema organizado. MySQL es un RDBMS (Relational DataBase Management System), un sistema de manejo de

una base de datos relacional, es decir, de una base de datos que guarda los datos en tablas separadas y no en un solo contenedor de data.

La separación de los datos de esta manera añade flexibilidad, capacidades de manejo y velocidad de transacciones. Permite el manejo de los datos mediante SQL (Structured Query Language), el lenguaje estándar de acceso a bases de datos, definido por el estándar SQL ANSI/ISO. El lenguaje SQL fue creado en 1986 y desde entonces se encuentra en constante evolución, siendo la versión actual la de 2003 (SQL:2003).

MySQL es un software Open Source, lo que significa que es posible para cualquier desarrollador el modificar el software a su conveniencia. Es posible para cualquier persona el descargar el software de MySQL y utilizarlo libre de costo. En cuestiones de licenciamiento, MySQL utiliza la licencia GPL (GNU General Public License), la cual define todo lo que se puede o no hacer con el código en aplicaciones comerciales. Para éstas, MySQL provee de un producto licenciado por sus propios estatutos, el cual cuenta con características extras no presentes en las versiones gratuitas.

MySQL promueve las principales características de su sistema de base de datos. Estas características son su velocidad, su confiabilidad y su facilidad de uso. Fue desarrollado originalmente para el manejo de grandes bases de datos más rápidamente que los sistemas tradicionales en su momento, y se ha utilizado en ambientes con gran número de transacciones por algunos años. Sin embargo, y a pesar de encontrarse en constante crecimiento, ha dejado paso a sistemas más robustos, seguros, y, obviamente, caros.

A pesar de todo, MySQL ofrece hoy en día un grande y útil set de funciones para los desarrolladores, un buen nivel de conectividad, velocidad y seguridad, lo que lo hace muy útil para el desarrollo en Internet. Estas características le permiten trabajar

en sistemas cliente – servidor de manera eficiente, puesto que consiste en un servidor multi-proceso que soporta diferentes programas de cliente, herramientas administrativas y la conectividad con un amplio rango de APIs.

#### **2.4.4 Protocolo HTTPS.**

Hypertext Transfer Protocol Secure, más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

##### **2.4.4.1 Características técnicas.**

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar. El puerto estándar para este protocolo es el 443.

#### **2.4.4.2 Diferencias con HTTP.**

En el protocolo HTTP las URLs comienzan con "http://" y utilizan por defecto el puerto 80, las URLs de HTTPS comienzan con "https://" y utilizan el puerto 443 por defecto.

HTTP es inseguro y está sujeto a ataques man-in-the-middle que pueden permitir al atacante obtener acceso a cuentas de un sitio web e información confidencial. HTTPS está diseñado para resistir esos ataques y ser más seguro.

#### **2.4.4.3 Capas de red.**

HTTP opera en la capa más alta del modelo OSI, la capa de aplicación; pero el protocolo de seguridad opera en una subcapa más baja, cifrando un mensaje HTTP previo a la transmisión y descifrando un mensaje una vez recibido. Estrictamente hablando, HTTPS no es un protocolo separado, pero refiere el uso del HTTP ordinario sobre una Capa de Conexión Segura SSL o una conexión con (TLS).

#### **2.4.4.4 Configuración del servidor.**

Para preparar un servidor web que acepte conexiones HTTPS, el administrador debe crear un certificado de clave pública para el servidor web. Este certificado debe estar firmado por una autoridad de certificación para que el navegador web lo acepte. La autoridad certifica que el titular del certificado es quien dice ser. Los navegadores web generalmente son distribuidos con los certificados raíz firmados por la mayoría

de las autoridades de certificación por lo que estos pueden verificar certificados firmados por ellos.

#### **2.4.4.5 Usar un control de acceso.**

El sistema puede también ser usado para la de clientes con el objetivo de limitar el acceso a un servidor web a usuarios autorizados. Para hacer esto el administrador del sitio típicamente crea un certificado para cada usuario, un certificado que es guardado dentro de su navegador. Normalmente, este contiene el nombre y la dirección de correo del usuario autorizado y es revisado automáticamente en cada reconexión para verificar la identidad del usuario, potencialmente sin que cada vez tenga que ingresar una contraseña.

### **2.4.5 Redes Neuronales.**

#### **2.4.5.1 Introducción.**

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.

Se parece al cerebro humano en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso de aprendizaje.

- Los pesos sinápticos o fuerza con que están interconectadas las neuronas se utilizan para almacenar la información.

#### 2.4.5.2 Perceptrón multinivel.

Dentro de las redes neuronales, las más utilizadas son las redes con múltiples capas que funcionan hacia adelante. Esta red está compuesta por un conjunto de nodos de entrada que componen la capa de entrada, un conjunto de una o más capas ocultas de neuronas y una capa de neuronas de salida.

La señal de entrada se propaga hacia adelante, desde la capa de entrada por la oculta hasta la salida; este tipo de configuración se conoce como MLP o “MultiLayer Perceptrons”.

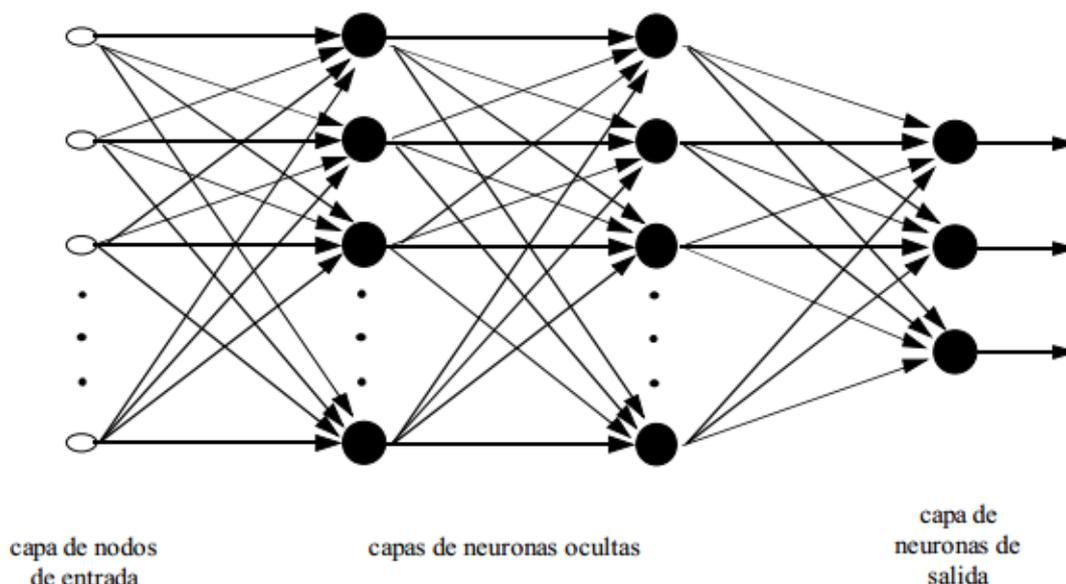


Figura 22. Estructura de un perceptrón multinivel.

El hecho de que este tipo de red se aplique para resolver con éxito multitud de problemas se debe a la utilización del algoritmo de aprendizaje que actualmente está

más extendido, el algoritmo o regla back propagation, el cuál es una generalización de la regla LMS “Least Mean Square”, por lo tanto también se basa en la corrección del error.

El proceso back propagation consiste en dos pasadas a través de las diferentes capas de la red, una pasada hacia adelante y una pasada hacia atrás. En la pasada hacia adelante, se aplica en la capa de entrada un patrón o vector de entrada, este propaga su efecto a través de las diferentes capas y como consecuencia produce un vector de salida. Durante este proceso, los pesos sinápticos de la red son fijos y no se modifican. Durante la pasada hacia atrás, en cambio, los pesos si se modifican de acuerdo con la regla de corrección del error.

La señal de salida real se compara con la señal deseada y como resultado se obtiene una señal de error, que se propaga en dirección contraria a través de la red modificando los pesos, de forma que, al volver a pasar el vector de entrada hacia adelante, la respuesta obtenida se asemeje más a la salida deseada.

Se puede decir que un perceptrón multicapa tiene tres características principales:

- El modelo de cada neurona (figura 23) incluye una función no lineal. En este caso, a diferencia del perceptrón donde es la función escalón, y debido a la necesidad de que sea una función continua y derivable, es la función sigmoide, donde:
  - $u_k$  es la suma total de la actividad interna en la neurona
  - $k$  (la señal de entrada) e
  - $y_k$  la salida que se produce en la neurona.

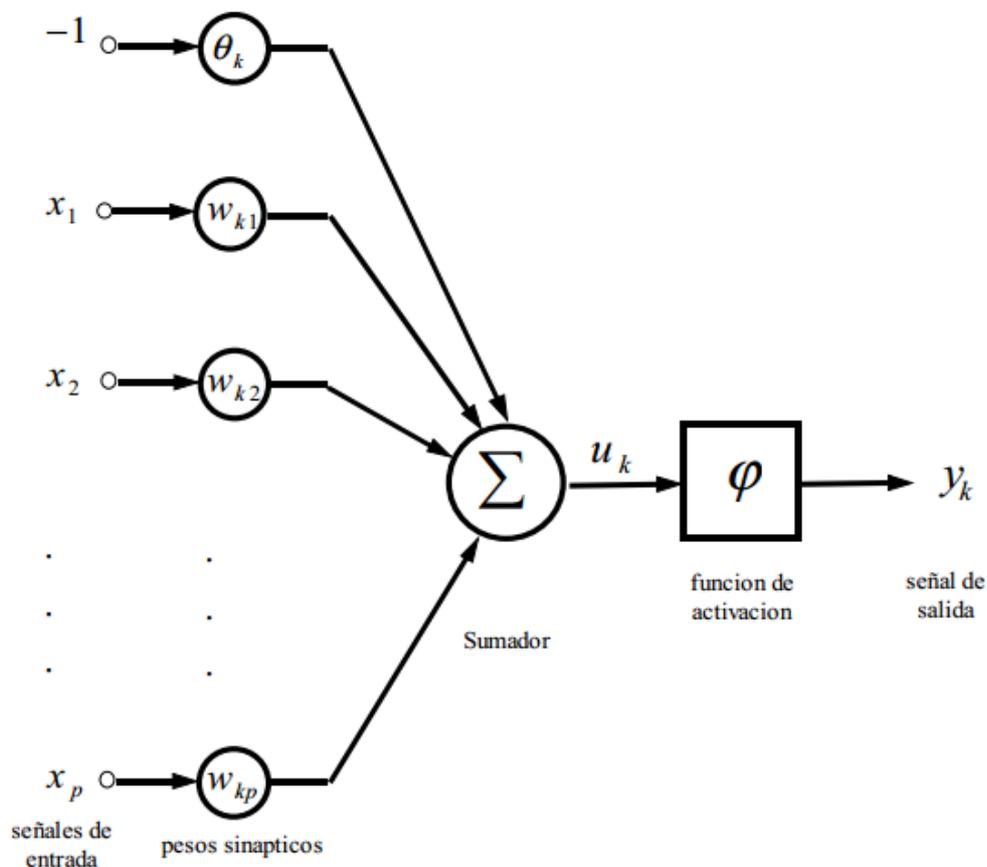


Figura 23. Modelo de una neurona.

$$y_k = \frac{1}{1 + \exp(-u_k)}$$

Ecuación 1

- La red contiene una o más capas ocultas de neuronas que no forman parte ni de la entrada ni de la salida. Estas neuronas ocultas capacitan a la red para aprender progresivamente cualquier correspondencia entre la entrada y la salida y almacenar internamente esta información.
- La red posee un gran número de conexiones, estas vienen determinadas por los pesos de la red. Un cambio en la conexión entre las neuronas equivale a un cambio en los pesos.

La combinación de estas características, hace que la habilidad de esta red para aprender a partir del entrenamiento sea más eficiente.

Este comportamiento hace que sea difícil conocer la respuesta de la red. Esto se debe a dos motivos, el comportamiento no lineal de las neuronas, las cuales están muy interconectadas, (lo que hace difícil un análisis teórico de la red) y la existencia de neuronas ocultas, que impide poder “ver” cómo se produce el aprendizaje y determinar cuáles son las características que mejorarían el aprendizaje.

El desarrollo del algoritmo back propagation proporciona un método eficiente para entrenar este tipo de redes. Aunque no es capaz de resolver todos los problemas, se ha demostrado como el mejor de todos, ya que tiene la capacidad de auto adaptar los pesos de las neuronas intermedias para aprender la relación que existe entre el conjunto de vectores o patrones de entrada y su correspondiente salida, y poder aplicar esa relación después del entrenamiento a nuevos vectores de entrada imperfectos o con ruido. Esta capacidad se conoce como generalización. La red debe encontrar una representación interna que le permita generar las salidas deseadas durante la etapa de entrenamiento, y posteriormente durante el funcionamiento ser capaz de generar salidas para entradas que no le fueron mostradas durante el aprendizaje pero que se asemejan a alguna de las que si le fueron mostradas.

## **CAPÍTULO III:**

### **DISEÑO DE LA APLICACIÓN**

#### **3.1 Diseño del algoritmo.**

##### **3.1.1 Planteamiento teórico.**

El trabajo se divide en tres bloques:

- El primero, la fase del tratamiento de la imagen.
- El segundo el entrenamiento de la red.
- El tercero, el reconocimiento mediante la red neuronal.

##### **3.1.2 Descripción de actividades.**

- Obtención de escala de grises: con este proceso se obtiene una imagen en una escala de grises comprendida en el intervalo  $[0,255]$ .



Figura 24. Imagen es escala de grises.

- Binarización: proceso tras el cual se obtiene una imagen binaria (negro 1 y blanco 0).

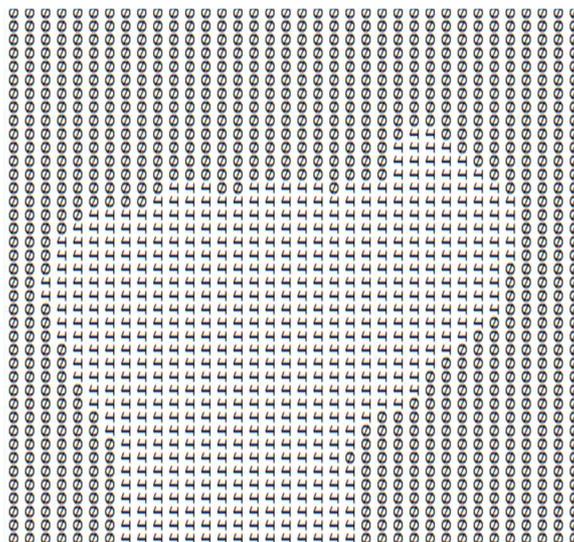


Figura 25. Binarización de la imagen.

- Reducción: con el objetivo de optimizar la entrada para la red neuronal se reduce su tamaño a 40x40 píxeles.



Figura 26. Imagen reducida a 40x40 píxeles.

- Obtención de características: obtención de un array de 0 y 1 que contiene los píxeles de la imagen tratada. Este array servirá como entrada a la red neuronal.
- Diseño red neuronal.
- Entrenamiento de la Red.
- Fase de pruebas.

### 3.1.3 Tratamiento de la imagen.

Partiendo de una imagen de captada por el dispositivo, el primer paso es transformar esta imagen en escala de grises. Para ello se utiliza el método de la luminancia transformando cada una de las componentes RGB a una sola mediante la ecuación siguiente:

$$Y = 0.299 * R + 0.587 * G + 0.144 * B$$

Ecuación 2.

Eso se realiza para quitar la información del color. Acto seguido se procede a realizar la binarización para obtener una imagen monocromática y así eliminar la información no útil. Para ello se ha seguido la siguiente función:

$$pixel[i] = \begin{cases} Max & \text{if } pixel[1] > threshold \\ 0 & \text{otherside} \end{cases}$$

*Pixel[i] da el valor de gris del pixel número i, siendo  $i \in [0, tamaño\_imagen]$*

Threshold=Umbral=40

El umbral tiene un valor de 40 en base a una serie de pruebas realizadas, con este valor se puede obtener un valor más exacto de negro y blanco.

```

for (int y = 0; y < height; y++){
    for (int x = 0; x < width; x++){
        pixelColor = imagen.getPixel(x, y);
        R = (int)((Color.red(pixelColor) ) * 0.3 );
        G = (int)((Color.green(pixelColor)) * 0.59);
        B = (int)((Color.blue(pixelColor)) * 0.11);
        valornuevo= R + G + B;
        if(valornuevo>40){
            valornuevo=0;
        }
        else
        {
            valornuevo=255;
            valorGenerado= valorGenerado + 1;
        }
        imagen_bn.setPixel(x, y, Color.rgb(valornuevo, valornuevo,
valornuevo));
    }
}

```

El objetivo es reducir el tamaño de la imagen a 40x40 píxeles, y crear un array de 1600 posiciones. Se recorre desde el primer al último píxel de la imagen rellenando el array de la siguiente forma:

$$array[i] = \begin{cases} 1 & pixel[i] == 255 \\ 0 & otherwise \end{cases}$$

*Pixel[i]* da el valor de gris del píxel número *i*, siendo  $i \in [0, 1600]$

Si no se reduce el tamaño de la imagen, se tendría un array intratable para la red neuronal. Finalmente se obtiene un array que se evaluará para determinar de qué letra se trata.

```

imagen_20=redimensionarImagenMaximo(imagen_bn, 40, 40);
for (int x = 0; x <40; x++){
    for (int y=0 ; y<40 ; y++){
        pixelColor = imagen_20.getPixel(x, y);
        if(pixelColor== -1)
        {
            pixelColor_t=0;
        }
        if(pixelColor==1)
        {
            pixelColor_t=0;
        }
        if(pixelColor== -16777216)
        {
            pixelColor_t=1;
        }
        if(pixelColor==16777216)
        {
            pixelColor_t=1;
        }
        datas_valor=datas_valor + Integer.toString(pixelColor_t);
    }
}

```

Una vez terminado el tratamiento de imágenes se procede al diseño de la red neuronal así como sus conjuntos de entrenamiento y de pruebas.

En el diseño de la red neuronal el primer nivel se encuentre la capa de entrada que contiene 1600 entradas, cada una de ellas tomará un valor de 0 o 1 según el array de clasificación construido anteriormente, a continuación las capas ocultas. Por último la capa de salida contará con 8 unidades devolviendo un valor en cada una de ellas entre 0 y 1. Este vector de 8 componentes hace referencia al valor binario de la letra o número.

```

iv_foto.setImageBitmap(imagen_20);
System.out.println("Prueba:"+datas_valor);
letra_aprende=listLetras.getSelectedItem().toString();
List<NameValuePair> nameValuePairs;
nameValuePairs = new ArrayList<NameValuePair>();
nameValuePairs.add(new BasicNameValuePair("data1" , datas_valor));
nameValuePairs.add(new BasicNameValuePair("modo" , modo_sistema));
nameValuePairs.add(new BasicNameValuePair("letra" ,letra_aprende));
DoPOST mDoPOST = new DoPOST(MainActivity.this, "");
mDoPOST.execute(nameValuePairs);

```

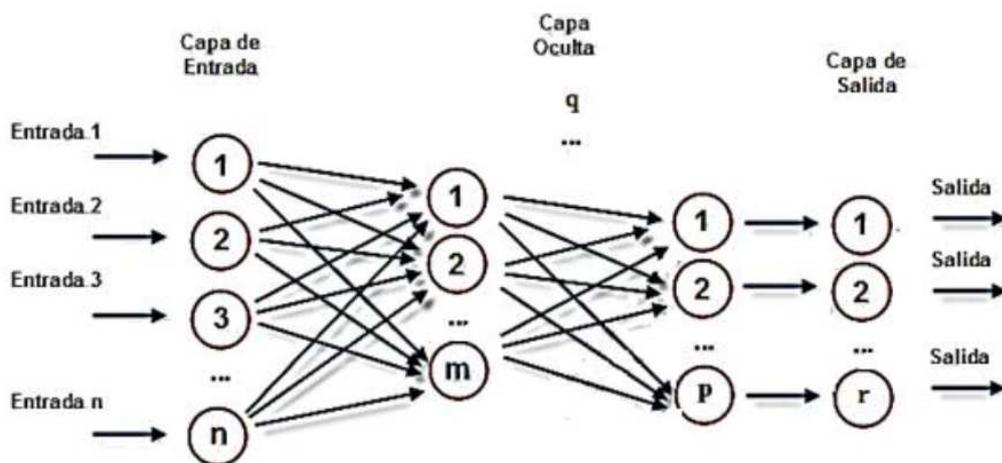


Figura 27. Red neuronal con  $n=1600$ ,  $q$  capas ocultas y  $r=8$

Tabla 2. Letras y números del alfabeto en binario.

LETRA BINARIO, MAYÚSCULAS	NUMEROS BINARIO, 1-10
A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
L	01001100
M	01001101
N	01001110
O	01001111
P	01010000
R	01010010
S	01010011
T	01010100
U	01010101
V	01010110
W	01010111
X	01011000
Y	01011001

La salida de cada una de las unidades se calcula de la siguiente manera:

$$a_i = g\left(\sum_{j=0}^n w_{ji} a_j\right)$$

Ecuación 3.

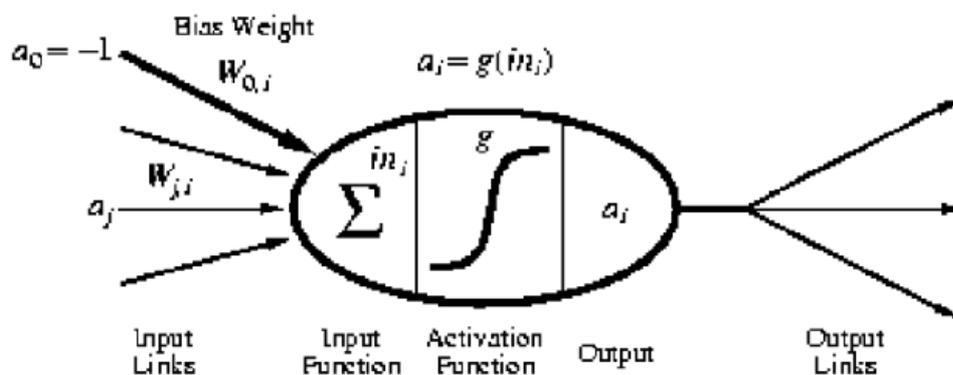


Figura 28. Unidad Neuronal

Donde  $g$  es una función de activación, el sumatorio  $\sum_{j=0}^n w_{ji} a_j$  (notado  $i_ni$ ) se hace sobre todas las unidades  $j$  que envían su salida a la unidad  $i$ . Excepto para  $j=0$ , que se considera una entrada ficticia  $a_0=-1$  y un peso  $w_{0i}$  denominado umbral.

En cuanto al modo de entrenamiento de las unidades se ha optado por la retropropagación usando como función de activación sigmoide. Por lo tanto:

$$g(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

Ecuación 4.

Esta función determinará en función de las entradas si el valor es 0 o 1 de la siguiente forma:

$$salida = \begin{cases} 1 & \sigma(x) > umbral \\ 0 & otherwise \end{cases}$$

Suponiendo una red neuronal con  $n$  unidades en la capa de entrada,  $m$  en la de salida y  $L$  capas en total. La capa 1 es la de entrada y la capa  $L$  la de salida. Cada unidad de una capa  $l$  está conectada con todas las unidades de la capa  $l+1$ . El peso entre la unidad  $i$  y la unidad  $j$  es denotado por  $\omega_{ij}$ .

Dado un ejemplo  $(\vec{x}, \vec{y})$  perteneciente a  $D$ :

- Si  $i$  es una unidad de la capa de entrada, notaremos por  $x_i$  la componente de  $\vec{x}$  correspondiente a dicha unidad.
- Si  $k$  es una unidad de la capa de salida, notaremos por  $y_k$  la componente de  $\vec{y}$  correspondiente a dicha unidad.

El algoritmo de retropropagación se desarrolla teniendo en cuenta un conjunto de entrenamiento  $D$ , con ejemplos de la forma  $(\vec{x}, \vec{y})$  con un factor de aprendizaje  $\eta$  y la función  $\sigma$ .

1. Inicializar los pesos de la red con valores cercanos a 0, positivos o negativos.
2. Repetir hasta que se satisfaga el criterio de parada:
  - 2.1 Para cada ejemplo  $(\vec{x}, \vec{y})$  perteneciente a  $D$  hacer:
    - 2.1.1 Calcular la salida  $a_i$  de cada unidad  $i$ , propagando valores hacia delante.
    - 2.1.2 Calcular los errores  $\Delta_i$  de cada unidad  $i$  y actualizar los pesos  $\omega_{ij}$ , propagando valores hacia atrás.

3. Devolver la red.

El punto 2.1.1 se desarrolla en:

- 3.1 Para cada nodo  $i$  de la capa de entrada hacer  $a_i = x_i$
- 3.2 Para  $l$  desde 2 hasta  $L$  hacer

3.2.1 Para cada nodo  $i$  de la capa  $l$  hacer  $in_i = \sum_j w_{ji} a_j = \sigma(in_i)$  (donde en el sumatorio anterior hay un sumando por cada unidad  $j$  de la capa  $l-1$ ).

El punto 2.1.2 se desarrolla:

3.3 Para cada unidad  $i$  en la capa de salida hacer  $\Delta_i = \sigma(in_i)(y_i - a_i)$

3.4 Para  $l$  desde  $L-1$  hasta 1 (decrementando  $l$ ) hacer

3.4.1 Para cada nodo  $j$  en la capa  $l$  hacer

i.  $\Delta_j = \sigma'(in_j) \sum_i w_{ji} \Delta_i$  (donde el sumatorio anterior tiene un sumando por cada unidad  $i$  de la capa  $l+1$ ).

ii. Para cada nodo  $i$  en la capa  $l+1$  hacer  $w_{ji} = w_{ji} + n a_j \Delta_i$

Los conjuntos de entrenamiento y de prueba son totalmente configurables por el usuario. Para el primero habrá que tener en cuenta que se dispone de 1600 entradas y para ese patrón se le asocia la posición (en notación binaria) de la letra del abecedario. Naturalmente el conjunto de prueba tiene que ser mucho menor que el de entrenamiento para que haya una buena precisión.

## 3.2 Redes neuronales artificiales.

Como lo implica su nombre, una red neuronal artificial consiste en una red de neuronas artificiales interconectadas. El concepto se basa vagamente en el funcionamiento el cerebro de un animal. Un cerebro consiste en un sistema de células interconectadas, las cuales son, aparentemente, responsables de los pensamientos, la

memoria y la conciencia. En una red neuronal artificial, se imitan estas habilidades por software.

A continuación se presenta el modelo de una neurona, las arquitecturas de red neuronal y los distintos métodos de aprendizaje

### 3.2.1 Modelo de una neurona

Una neurona es una unidad de procesamiento de información que es fundamental para la red neuronal [Hilera y Martínez, 1995]. El diagrama en bloques de la figura 26 muestra el modelo de una neurona, el cual forma las bases para diseñar redes neuronales artificiales.

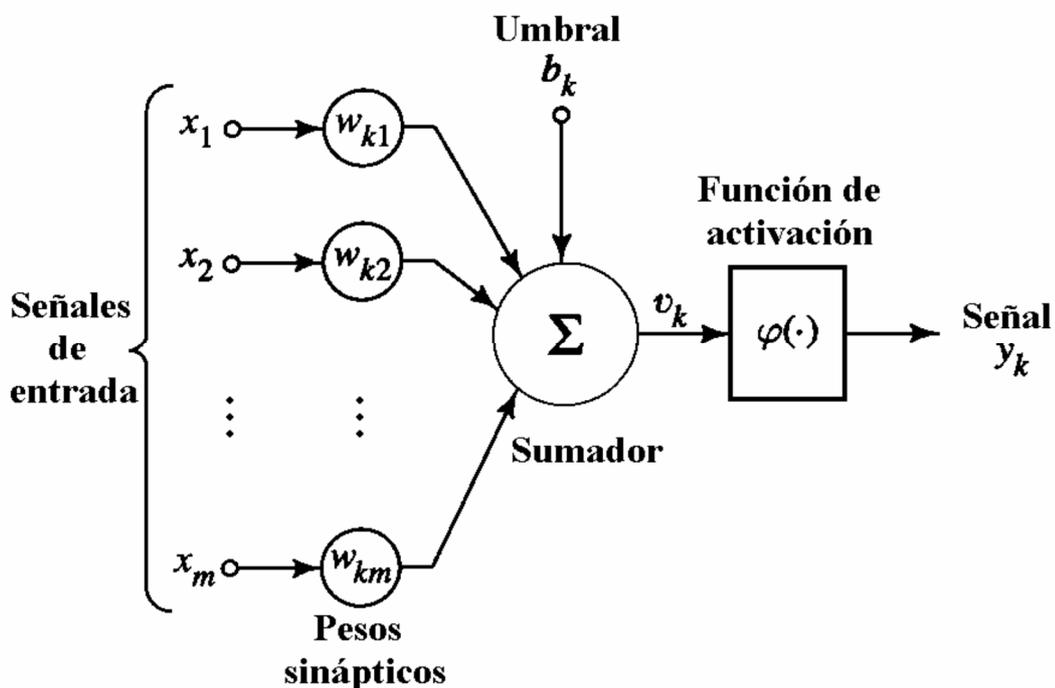


Figura 29. Modelo no lineal de una neurona.

Aquí se pueden identificar tres elementos básicos del modelo neuronal:

- a. Un conjunto de sinapsis o conexiones, cada una caracterizada por un peso propio.
- b. Un sumador para sumar las señales de entrada, ponderadas con los pesos respectivos de las sinapsis de la neurona. Las operaciones descritas aquí constituyen un combinador lineal.
- c. Una función de activación para limitar la amplitud de la salida de la neurona. Generalmente, el rango de amplitud de salida de una neurona se normaliza a un intervalo cerrado unitario  $[0, 1]$  o alternativamente  $[-1, 1]$ . Usualmente la función de activación es no lineal, tal como una función escalón, sigmoideal o gaussiana.

El modelo neuronal de la figura 29 también incluye un umbral aplicado externamente, denotado por  $b_k$ . El umbral  $b_k$  tiene el efecto de subir o bajar la entrada neta de la función de activación, dependiendo de si es positivo o negativo, respectivamente.

En particular, dependiendo de si el umbral  $b_k$  es positivo o negativo, la relación entre el campo local inducido  $v_k$  de la neurona  $k$  y la salida del combinador lineal  $u_k$  se modifica de la manera ilustrada en la figura 30 de aquí el nombre de “campo local inducido”. Se puede notar como resultado de esta transformación, el gráfico de  $v_k$  versus  $u_k$  ya no pasa por el origen.

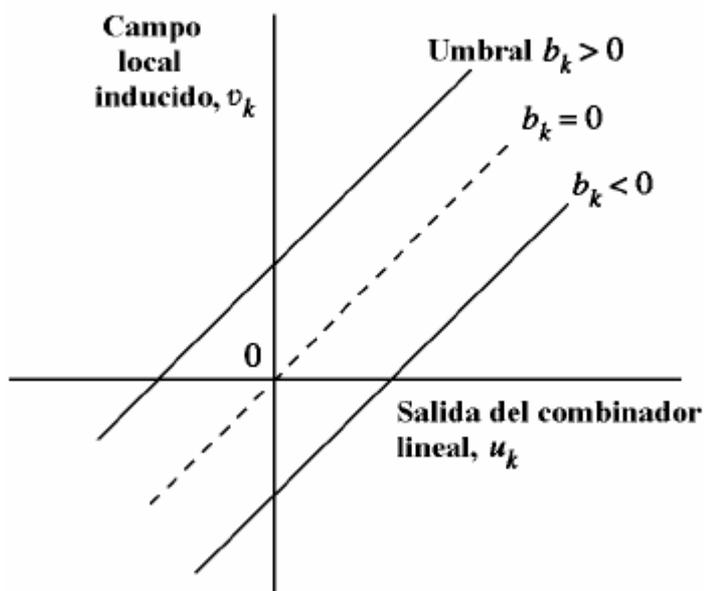


Figura 30. Transformación producida por la presencia del umbral.

### 3.2.2 Arquitecturas.

El modo en que se conecta una red neuronal se denomina topología o arquitectura. Las topologías de red más usadas son las redes con conexión hacia delante o redes feedforward.

Estas redes son una generalización de los perceptrones multicapa (MLPs), que constituyen la categoría de topologías más utilizada en computación neuronal.

El Perceptrón de una capa (SLP), una de las topologías de red más antiguas, consiste en una capa de unidades computacionales; las unidades de entrada no desarrollan ningún cálculo (figura 31a). La capa de entrada se conecta completamente con la capa de salida, es decir, cada unidad de entrada está conectada a cada unidad de salida.

El Perceptrón multicapa (MLP, figura 28b) tiene capas adicionales, llamadas capas ocultas. Aunque se dice que la mayoría de los algoritmos de entrenamiento se aplican a los MLPs, en realidad pueden entrenar a cualquier red que tenga una topología feedforward (figura 31c). Esta generalización a partir de los MLPs es muy útil para las redes neuronales evolutivas.

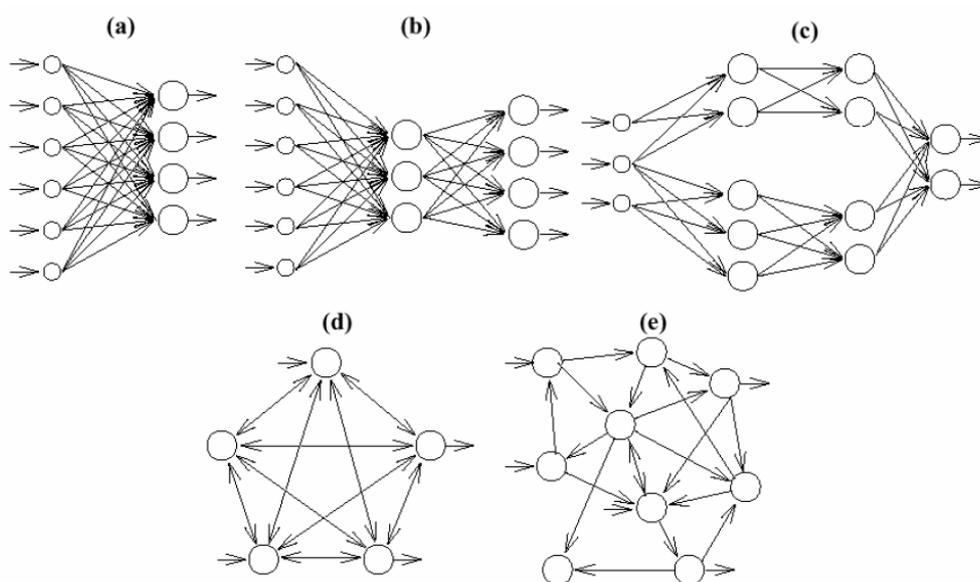


Figura 31. Topologías de redes Neuronales.

- a) Un perceptrón de una capa (SLP) conectado completamente
- b) Un perceptrón multicapa (MLP) conectado completamente
- c) Un MLP modular
- d) Una red recurrente conectada completamente
- e) Una red recurrente conectada parcialmente.

### 3.2.3 Aprendizaje.

Existen dos fases en toda aplicación de las redes neuronales: la fase de entrenamiento (o aprendizaje) y la fase de prueba.

- Fase de entrenamiento: Se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos (parámetros de diseño) que definen el modelo neuronal.
- Fase de prueba o funcionamiento directo: Una vez entrenado este modelo, se procesan los patrones de prueba que constituyen la entrada habitual de la red, analizándose de esta manera las prestaciones definitivas de la red.

Normalmente, los pesos óptimos se obtienen optimizando (minimizando) alguna función de energía. Por ejemplo, un criterio muy utilizado en el llamado entrenamiento supervisado, es minimizar el error cuadrático medio entre el valor de salida y el valor real esperado.

El aprendizaje en una red neuronal se lleva a cabo típicamente utilizando ejemplos. El entrenamiento se logra al ajustar los pesos de las conexiones iterativamente de manera tal que las redes entrenadas puedan desarrollar ciertas tareas. El aprendizaje en redes neuronales se puede dividir a grandes rasgos en supervisado, no supervisado y por refuerzo.

- El aprendizaje supervisado se basa en la comparación directa entre la salida actual de una red neuronal y la salida deseada correcta, también conocida como la salida o dato objetivo. Se formula a menudo como la minimización de una función de error tal como el error cuadrático medio total entre la salida actual y la salida deseada sumada sobre todos los datos disponibles.
- El aprendizaje por refuerzo es un caso especial del aprendizaje supervisado donde la salida deseada exacta es desconocida. Se basa sólo en la información de si la salida actual es o no correcta.

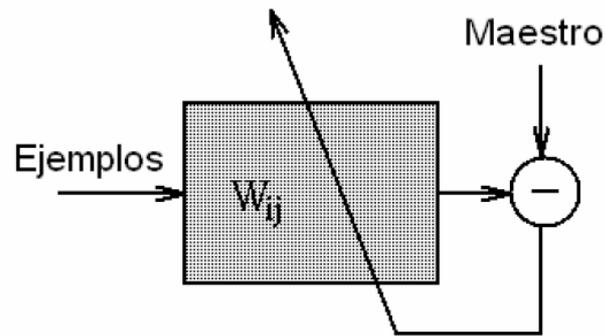


Figura 32. Diagrama que representa el aprendizaje supervisado.

- El aprendizaje no supervisado está basado solamente en las correlaciones entre los datos de entrada. No hay información disponible sobre la “salida correcta” para el aprendizaje.

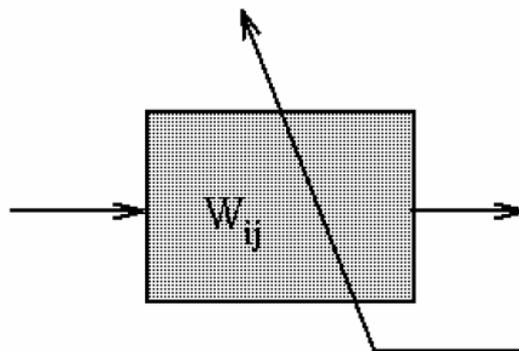


Figura 33. Diagrama que representa el aprendizaje no supervisado.

Las aplicaciones del mundo real deben acometer dos tipos diferentes de requisitos en el procesamiento. En un caso, se requiere la prueba en tiempo real pero el entrenamiento se puede realizar “fuera de línea”. En otras ocasiones, se requieren ambos procesos, el de prueba y el de entrenamiento, en tiempo real. Estos dos requisitos implican velocidades de proceso muy diferentes, que afectan a los algoritmos y hardware usados.

### **3.2.4 Entrenamiento de la red neuronal.**

Para entrenar una red neuronal y aproximarla a una función deseada, se utiliza un algoritmo de aprendizaje.

El algoritmo de aprendizaje ajusta automáticamente los pesos de una red neuronal de manera que pueda mejorar su habilidad para entregar una salida deseada a partir de una entrada dada.

Un algoritmo de aprendizaje entrena una red al examinar repetidamente cómo responde la red a sus datos de entrenamiento y determinando cómo debería ajustar sus pesos para mejorar la salida.

Uno de los algoritmos de entrenamiento se denomina “back-propagation” (retropropagación), utilizado para entrenar a las redes feedforward multicapa, una clase importante de redes neuronales. Las redes feedforward multicapa consisten en un conjunto de unidades sensoriales (nodos fuente) que constituyen la capa de entrada, una o más capas ocultas de nodos computacionales, y una capa de salida.

Las señales de entrada se propagan a través de la red en dirección hacia adelante, capa por capa. Estas redes neuronales son llamadas comúnmente “perceptrones multicapa” (MLPs), y representan una generalización del Perceptrón de una capa.

#### **3.2.4.1 Algoritmo back-propagation.**

En la aplicación del algoritmo back-propagation, se distinguen dos pasadas distintas de cómputo. La primera pasada se refiere como la pasada hacia adelante, y la segunda como la pasada hacia atrás.

En la pasada hacia delante los pesos permanecen inalterados por toda la red, y las señales funcionales de la red se computan neurona por neurona.

Así, la fase hacia delante comienza en la primera neurona oculta al presentarle el vector de entrada, y termina en la capa de salida al computar la señal de error para cada neurona de esta capa.

La pasada hacia atrás, por el otro lado, comienza en la capa de salida al pasar las señales de error hacia la izquierda a través de la red, capa por capa, y computando recursivamente. Este proceso recursivo permite que los pesos experimenten cambios. Esta computación recursiva continúa, capa por capa, al propagar los cambios de todos los pesos de la red.

#### **3.2.4.1.1 El Cómputo de Doble Pasada**

En la aplicación del algoritmo back-propagation, se distinguen dos pasadas distintas de cómputo. La primera pasada se refiere como la pasada hacia delante, y la segunda como la pasada hacia atrás.

En la pasada hacia delante los pesos permanecen inalterados por toda la red, y las señales funcionales de la red se computan neurona por neurona.

La fase hacia delante comienza en la primera neurona oculta al presentarle el vector de entrada, y termina en la capa de salida al computar la señal de error para cada neurona de esta capa.

La pasada hacia atrás, por el otro lado, comienza en la capa de salida al pasar las señales de error hacia la izquierda a través de la red, capa por capa, y computando recursivamente el  $d$  (gradiente local) para cada neurona. Este proceso recursivo permite que los pesos experimenten cambios.

### 3.2.4.1.2 Función de Activación.

En redes computacionales, la Función de Activación de un nodo define la salida de un nodo dada una entrada o un conjunto de entradas.

Se puede decir que un circuito estándar de computador se comporta como una red digital de funciones de activación al activarse como "ON" (1) u "OFF" (0), dependiendo de la entrada. Esto es similar al funcionamiento de un Perceptrón en una Red neuronal artificial.

El valor de red, expresado por la función de base,  $u(w, x)$ , se transforma mediante una función de activación no lineal. Las funciones de activación más comunes son la función sigmoideal y gaussiana:

Tabla 3. Tipos de función de activación.

Tipo de función	Cálculo de la salida
Funcion Sigmoideal	$f(u_i) = \frac{1}{1 + e^{-\frac{u_i}{\sigma}}}$ <p style="text-align: center;">Ecuación 5</p>
Funcion Gaussiana	$f(u_i) = ce^{-\frac{u_i}{\sigma}}$ <p style="text-align: center;">Ecuación 6</p>

### 3.2.4.1.3 Tipos de entrenamiento.

El entrenamiento consiste en ir ajustando los pesos gradualmente hasta que el vector de salida resultante coincida con el vector de salida deseado.

El entrenamiento supervisado parte de un vector de entrada del cual se conoce su vector de salida deseada o al menos una aproximación a él. Al par de vectores representando los valores de entrada y salida deseada se le denomina par de entrenamiento.

Este proceso consiste en aplicar el vector de entrada a la red. La diferencia o cambio existente entre el vector de salida y el vector de salida deseada se reduce a través de diversos algoritmos existentes. Se continúa probando diversos vectores de entrada ajustando pesos, hasta que la diferencia con la salida deseada es mínima.

En el entrenamiento no supervisado se desconoce la salida, únicamente se proporciona un vector de entrada. Lo que se busca es generar después de varios vectores de entrada, salidas que sean consistentes. Es decir, que los pesos se vayan ajustando poco a poco a través del reconocimiento de patrones, regularidades, propiedades estáticas, etc.

Así, las entradas similares producirán el mismo tipo de salida. Otra forma de explicar esto es, que este proceso extrae propiedades estadísticas del conjunto de entrenamiento.

### **3.3 Generalización**

En el aprendizaje back-propagation, normalmente se comienza con una muestra de entrenamiento y se usa el algoritmo back-propagation para computar los pesos de un Perceptrón multicapa al cargar tantos ejemplos de entrenamiento como sea posible dentro de la red. Entonces se espera que la red así diseñada generalice.

Se dice que una red neuronal generaliza bien cuando el mapeo entrada-salida computado por la red es correcto (o casi) para datos de testeo nunca utilizados en la creación o entrenamiento de la red.

El proceso de aprendizaje (entrenamiento de la red neuronal) puede ser visto como un problema de “ajuste de curva”. La red en sí misma puede considerarse simplemente como un mapeo entrada-salida no lineal. Tal punto de vista permite ver a la generalización no como una propiedad mística de las redes neuronales sino simplemente como el efecto de una buena interpolación no-lineal de los datos de entrada.

La red desarrolla una interpolación útil principalmente porque los perceptrones multicapa con funciones de activación continuas conducen a funciones de salida que también son continuas.

Las figuras 34 y 35 ilustran cómo puede ocurrir la generalización en una red hipotética. El mapeo entrada-salida no lineal representado por la curva mostrada en esta figura es computada por la red como resultado de aprender los puntos etiquetados como “datos de entrenamiento”.

El punto sobre la curva marcado como “generalización” es visto entonces como el resultado de la interpolación desarrollada por la red. Una red neuronal que se diseña para generalizar bien producirá un mapeo entrada-salida correcto aun cuando la entrada es ligeramente distinta de los ejemplos usados para entrenar a la red, como se ilustra en la figura. Sin embargo, cuando una red neuronal aprende demasiados ejemplos de entrada-salida, la red terminará memorizando el set de entrenamiento.

Puede hacerlo al encontrar una característica (debido al ruido, por ejemplo) que está presente en los datos de entrenamiento pero no es cierta para la función subyacente

que debe modelizarse. A este fenómeno se lo conoce como sobreajuste o sobreentrenamiento.

Cuando la red está sobreentrenada, pierde su habilidad para generalizar entre patrones de entrada-salida similares.

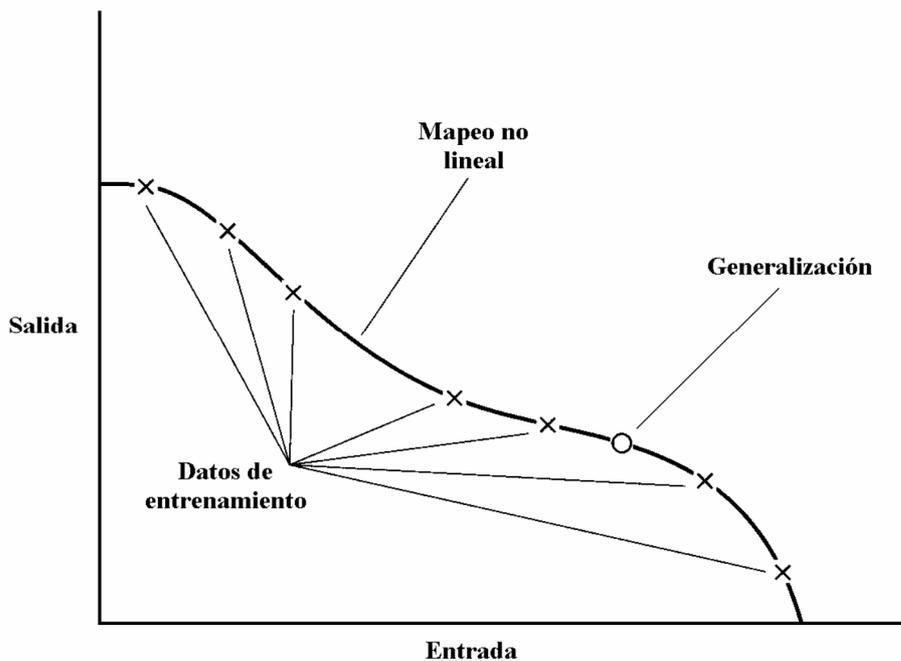


Figura 34. Datos ajustados apropiadamente (generalización buena).

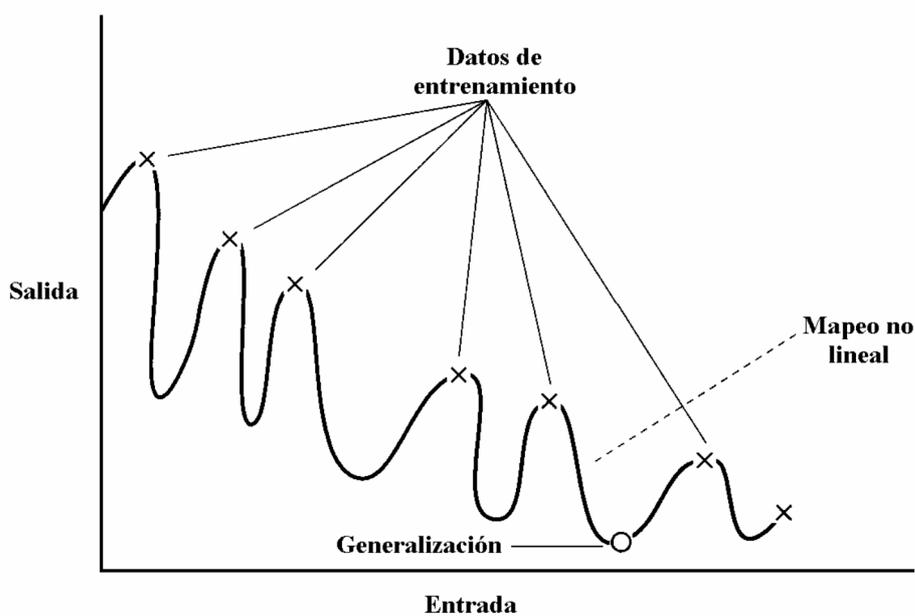


Figura 35. Datos sobre ajustados (generalización pobre).

### **3.3.1.1 Validación Cruzada.**

La esencia del aprendizaje back-propagation es codificar un mapeo entrada-salida (representado por un conjunto de ejemplos etiquetados) en los pesos y umbrales de un Perceptrón multicapa. Lo que se desea es que la red se entrene bien de forma tal que aprenda lo suficiente acerca del pasado para generalizar en el futuro. Desde esta perspectiva el proceso de aprendizaje debe elegir la parametrización de la red más acorde a este conjunto de ejemplos.

El set de entrenamiento se particiona aleatoriamente en dos subsets disjuntos:

- El subset de estimación, usado para seleccionar el modelo.
- El subset de validación, usado para evaluar o validar el modelo.

### **3.3.1.2 Entrenamiento con Detención Temprana (Early Stopping).**

Comúnmente, un Perceptrón multicapa entrenado con el algoritmo de back-propagation aprende en etapas, moviéndose desde la realización de funciones de mapeo bastante simples a más complejas a medida que progresa la sesión de entrenamiento. Esto se ejemplifica por el hecho de que en una situación típica el error cuadrático medio disminuye con el incremento del número de epochs durante el entrenamiento: comienza con un valor grande, decrece rápidamente, y luego continúa disminuyendo lentamente mientras la red hace su camino hacia un mínimo local sobre la superficie de error. Teniendo como meta a una buena generalización.

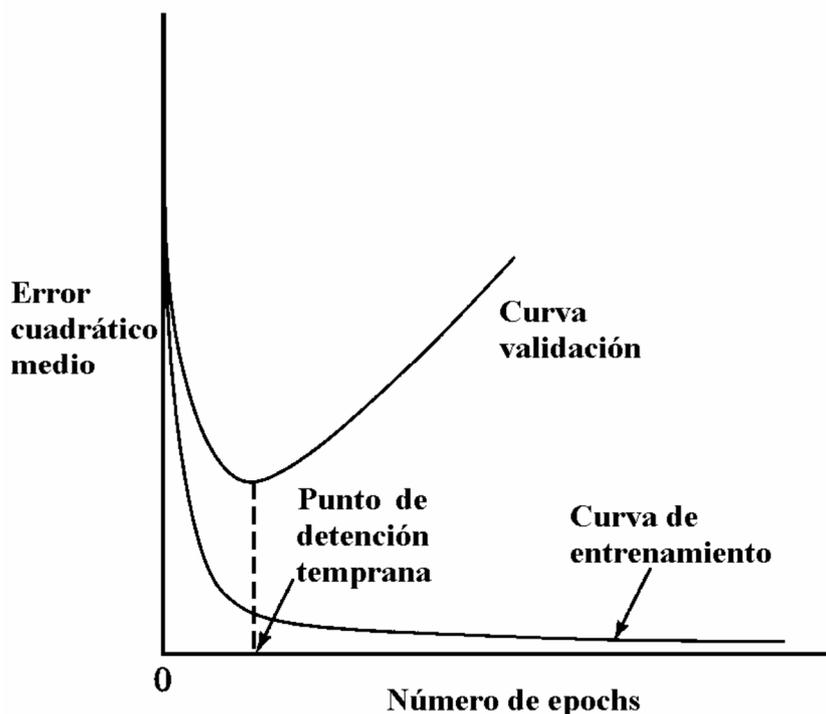


Figura 36. Ilustración de la regla de detención temprana basada en la validación cruzada.

Una teoría estadística del fenómeno del sobreajuste se presenta en, donde se identifican dos modos de comportamiento dependiendo del tamaño del set de entrenamiento:

- a. **Modo no-asintótico:** Para este modo de comportamiento, el método de entrenamiento con detención temprana mejora la performance de generalización de la red por sobre el entrenamiento exhaustivo (es decir, cuando se utiliza el set de ejemplos completo para el entrenamiento y la sesión no se detiene).
- b. **Modo asintótico:** Para este modo de comportamiento, la mejora en la performance de generalización producida por el uso del método de entrenamiento con detención temprana por sobre el entrenamiento exhaustivo

es pequeña. En otras palabras, el entrenamiento exhaustivo es satisfactorio cuando el tamaño del set de entrenamiento es grande comparado con el número de parámetros de la red.

### 3.3.1.3 Variantes a la Validación Cruzada

El enfoque a la validación cruzada descrito se denomina método *hold-out*. Existen otras variantes de validación cruzada que encuentran sus propios usos prácticos, particularmente cuando hay escasez de ejemplos etiquetados (entrada y salida deseada). El modelo se entrena sobre todos los subsets excepto por uno, y el error de validación se mide al evaluarlo sobre el subset dejado de lado.

### 3.3.1.4 Poda de la Red (Pruning)

Para resolver problemas del mundo real con redes neuronales usualmente se requiere el uso de redes altamente estructuradas de bastante tamaño. Una red neuronal con tamaño mínimo es menos probable que aprenda el ruido en los datos de entrenamiento, y puede así generalizar mejor sobre datos nuevos. Se puede lograr este objetivo de diseño con una de las siguientes formas:

- a. **Crecimiento de la red:** en cuyo caso comenzamos con un Perceptrón multicapa pequeño (pequeño para cumplir la tarea en cuestión) y luego agregamos una nueva neurona o una nueva capa de neuronas ocultas solo cuando no seamos capaces de cubrir la especificación de diseño.

- b. **Poda de la red (pruning):** en cuyo caso se comienza con un Perceptrón multicapa grande con una performance adecuada para el problema en cuestión, y entonces lo podamos al debilitar o eliminar ciertos pesos en una manera selectiva y ordenada.

Al diseñar un Perceptrón multicapa por el método que sea, se construye un *modelo* no-lineal del fenómeno físico responsable de la generación de ejemplos de entrada-salida usados para entrenar la red.

### **3.4 Grafo del sistema.**

#### **3.4.1 Diagrama de Arquitectura.**

El sistema está realizado bajo tres capas, los cuales son:

- Capa 1, Usuario o dispositivo móvil.
- Capa 2, Servidor de inteligencia artificial.
- Capa 3, Servidor de base de datos.

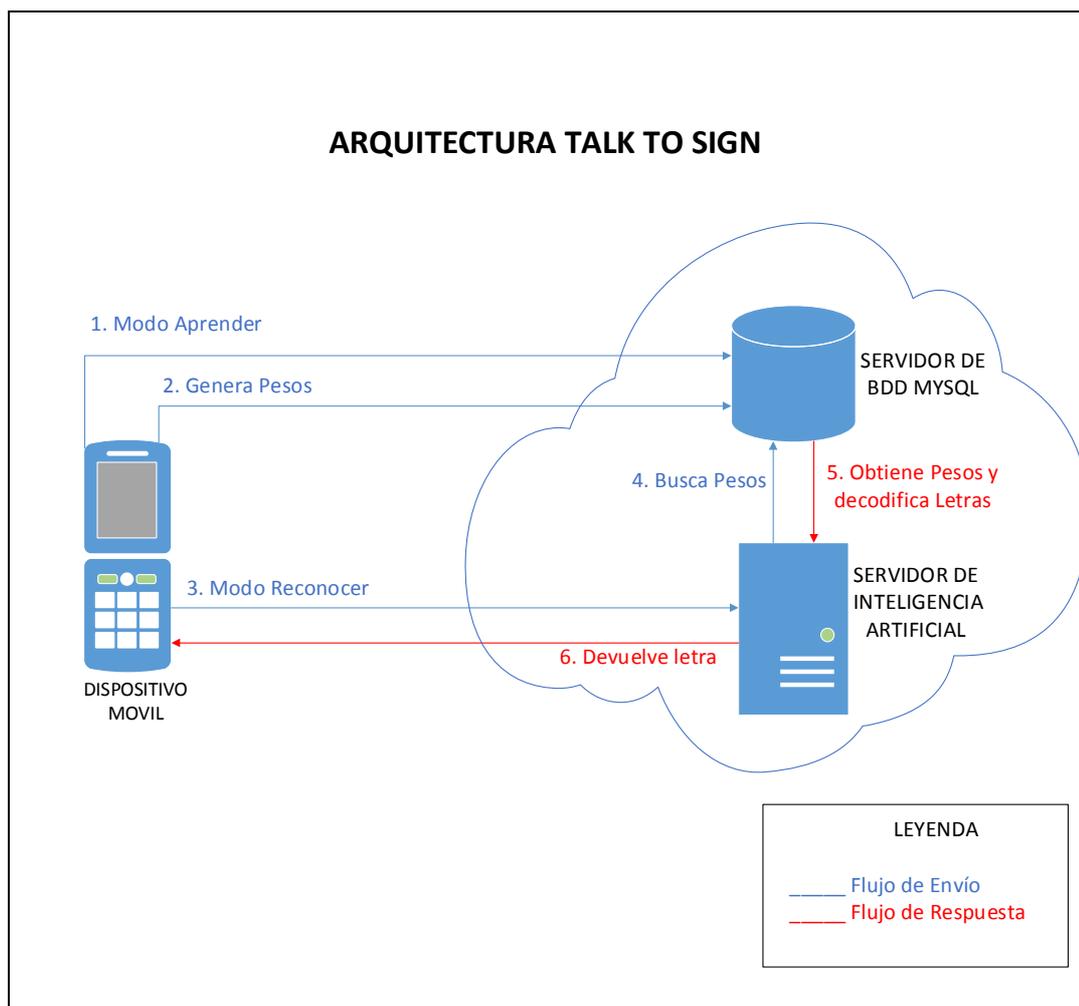


Figura 37. Diagrama de Arquitectura Talk to sign.

La capa 1, está formada por un dispositivo móvil con sistema operativo Android de la versión 4 en adelante.

La capa 2, está formada por un servidor de inteligencia artificial, el cual se encuentra la nube, en un servidor de aplicaciones (proveedor godaddy), y se conecta al siguiente web service: <https://www.googes.com/talk>. Las características que tiene el servidor web son las siguientes:

- IP Address 72.167.131.115
- Operating System: LINUX
- PHP Version: 5

- Data Center: United States
- SSL Certificate: Installed

La capa 3, está formada por el servidor de base de datos, el cual se encuentra en la nube, en el servidor de aplicaciones godaddy, para acceder al mismo ingresamos con la siguiente URL:

[https://p3nlmysqladm001.secureserver.net/dgrid50/37/index.php?lang=es-utf-8&convcharset=iso-8859-1&collation\\_connection=utf8\\_unicode\\_ci&token=8b4a5eddbd721290940cc107947e6bfa](https://p3nlmysqladm001.secureserver.net/dgrid50/37/index.php?lang=es-utf-8&convcharset=iso-8859-1&collation_connection=utf8_unicode_ci&token=8b4a5eddbd721290940cc107947e6bfa)

Características de base de datos:

- My SQL v5.0
- Está compuesto por las siguientes tablas:
  - Tabla talk\_letras: se almacenan los patrones de letras previamente tomados con el dispositivo móvil.
  - Tabla talk\_pesos: se almacena los pesos del aprendizaje del algoritmo back-propagation.

### 3.4.2 Diagrama de Grafos.

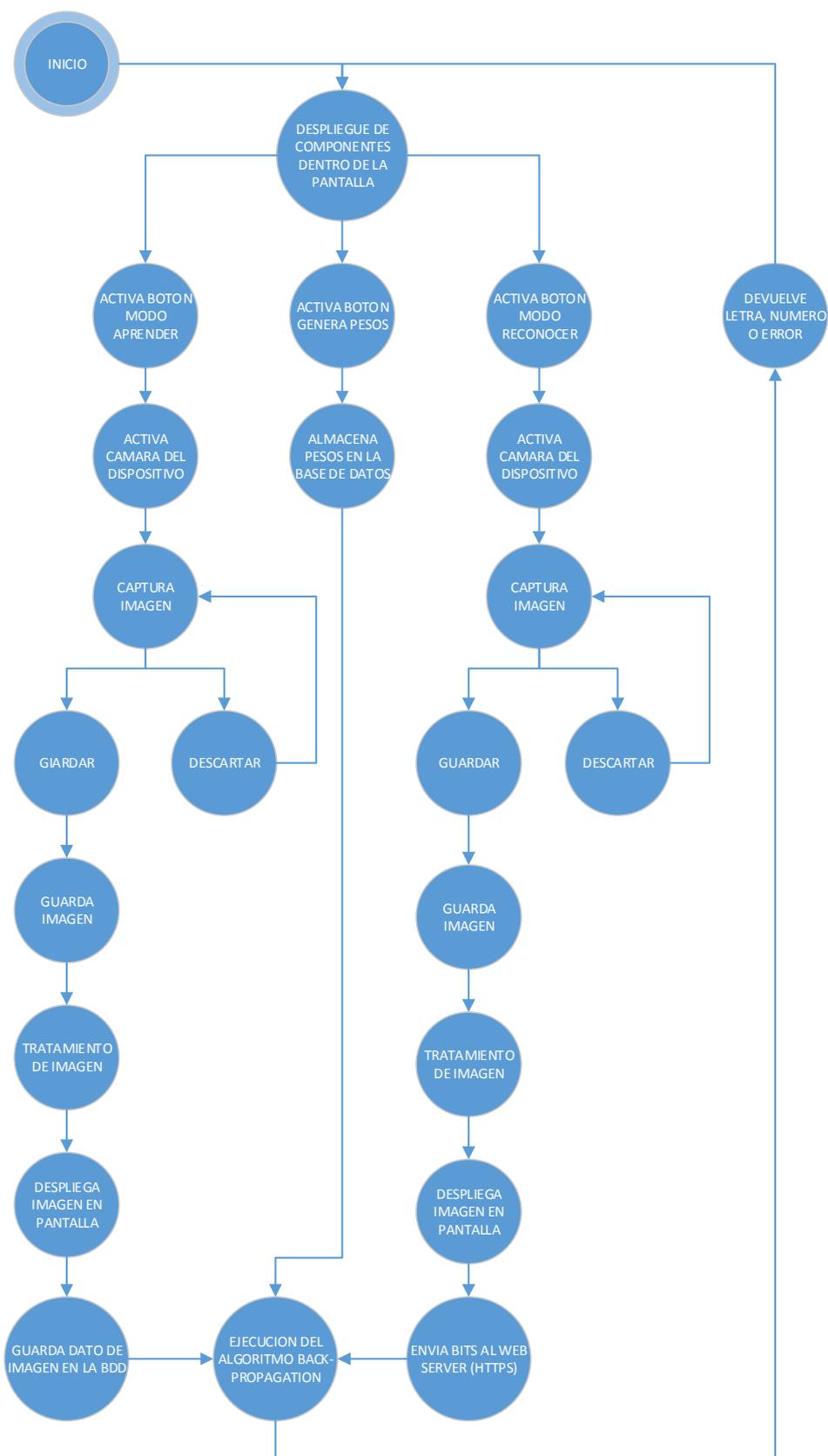


Figura 38. Diagrama de Grafos Talk to sign.

### 3.4.3 Diagrama de Secuencia.

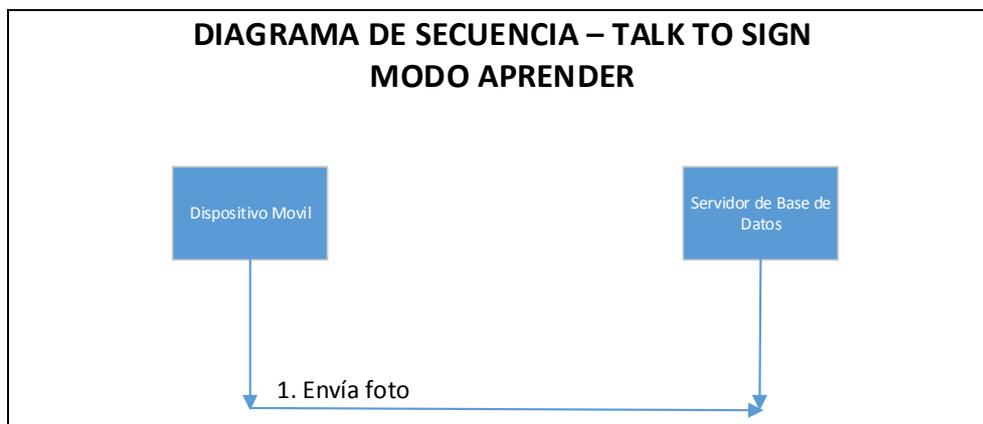


Figura 39. Diagrama de Secuencia Talk to Sign, en modo aprender

El dispositivo móvil capta la imagen, hace el tratamiento de la imagen y envía los bits por el protocolo https a la base de datos para ser almacenados.



Figura 40. Diagrama de Secuencia Talk to Sign, Genera Pesos

El dispositivo móvil toma las muestras de las imágenes de la letra capturadas y genera los pesos respectivos para almacenarlos en la base de datos.

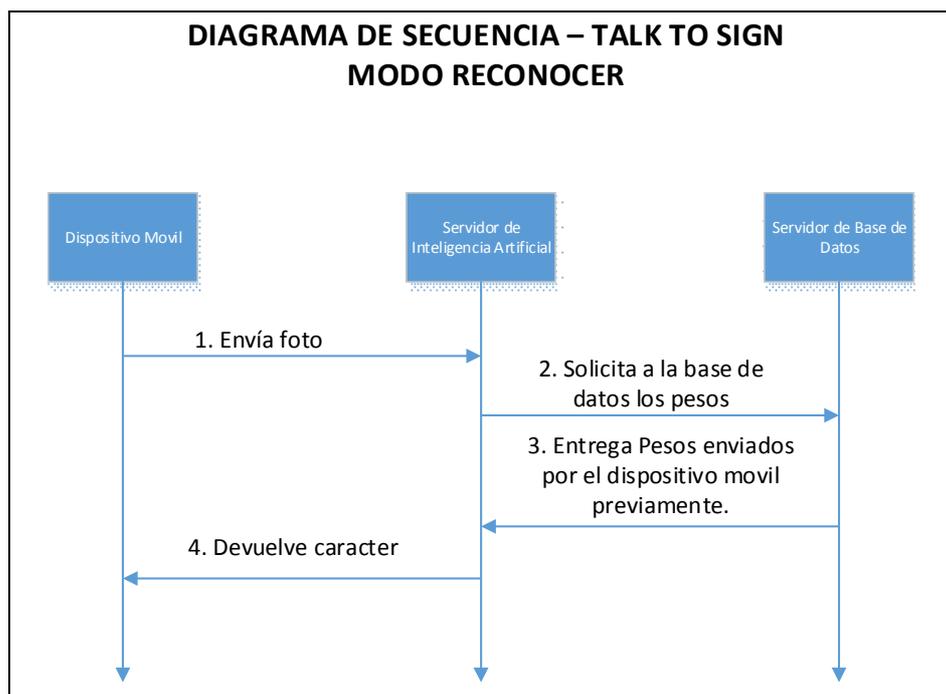


Figura 41. Diagrama de Secuencia Talk to Sign, en modo reconocer.

Dispositivo móvil: este capta la imagen, hace el tratamiento de la imagen, envía los 1600 bits por el protocolo https al servidor de inteligencia artificial el cual recibe los bits, extrae de la base de datos los pesos enviados previamente por el dispositivo móvil y hace la comparación con la trama que llego ejecutando el algoritmo back propagation devolviendo el resultado de la trama al dispositivo móvil.

### 3.4.4 Diagrama de Clases

Los diagramas de clases describen la estructura estática de un sistema.

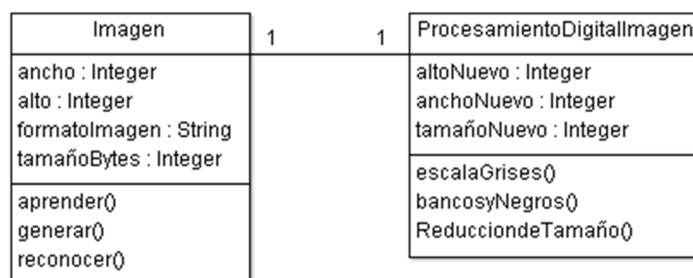


Figura 42. Diagrama de Clases

### 3.4.5 Diseño gráfico de la aplicación.

La aplicación tiene un modo gráfico, el cual se indica en la figura 39. En este se puede observar:

- Nombre de la aplicación.
- Botones de selección:
  - Modo aprender.
  - Letras y números a aprender.
  - Modo Genera Pesos.
  - Modo reconocer.
- Botón que activa la cámara del dispositivo.

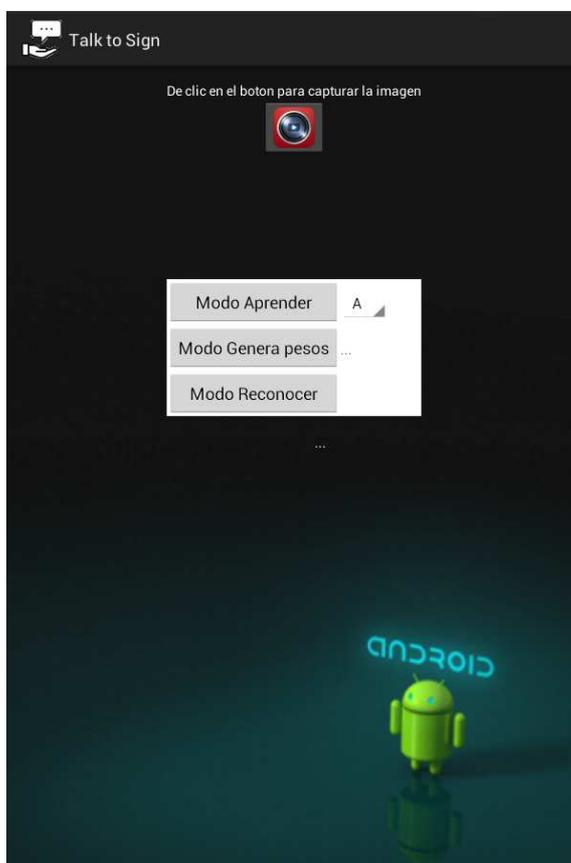


Figura 43. Diseño gráfico de la aplicación.

### 3.4.6 Diagrama de casos de uso.

#### 3.4.6.1 Sistema (Nivel 0)

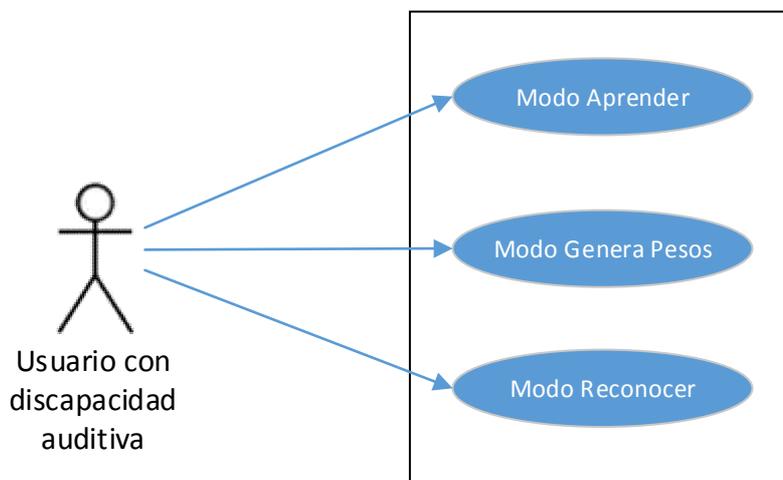


Figura 44 Diagrama General.

#### 3.4.6.2 Casos de uso (Niveles siguientes y a detalle)

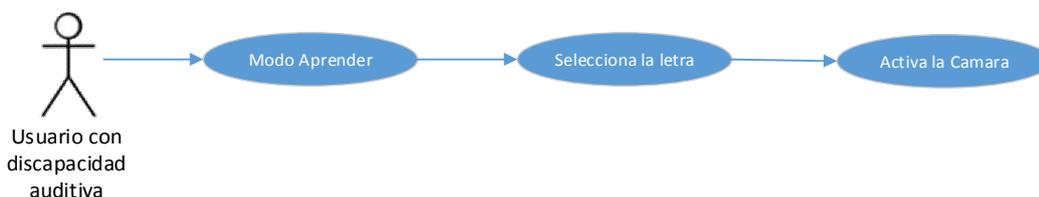


Figura 45. Diagrama nivel siguiente (Modo Aprender).

**Objetivo:** El usuario con discapacidad auditiva podrá ingresar al modo aprender y seleccionar la letra si desea que el sistema aprenda más patrones de la letra seleccionada. Cada patrón de letra aprendida se almacenara en la base de datos, por lo que es necesario contar con servicio de internet.

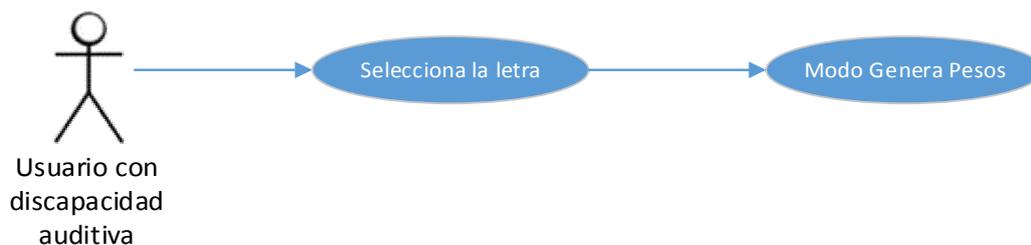


Figura 46. Diagrama nivel siguiente (Modo Genera Pesos).

**Objetivo:** El usuario con discapacidad auditiva podrá seleccionar una letra y generar los pesos, los pesos se generaran de acuerdo al número de muestras que se encuentren en la base de datos. Los pesos generados serán almacenados en la base de datos, por lo que es necesario contar con servicio de internet.

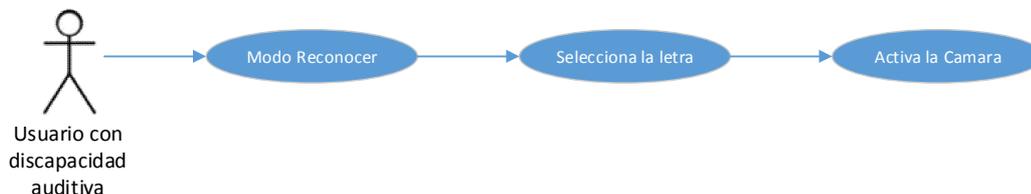


Figura 47. Diagrama nivel siguiente (Modo Reconocer).

**Objetivo:** El usuario con discapacidad auditiva podrá seleccionar una letra que desee reconocer. Por ejemplo, si el usuario desea reconocer la letra A, seleccionara la letra A, y activara la cámara, capturará la letra A correspondiente al lenguaje de las señas y la aplicación si reconoce este patrón mostrara como resultado la A, en caso de no reconocer el patrón, mostrará un mensaje que no conoce ese patrón. De igual manera funciona para cada una de las letras del abecedario y los números del 1 al 10.

### 3.5 Funcionamiento.

#### 3.5.1 Entrenamiento de la Red.

Para el entrenamiento de la Red se debe aprender las letras y generar los pesos de las mismas, por lo que se realizan los siguientes pasos:

- Seleccionar el Modo Aprender de la interfaz gráfica.

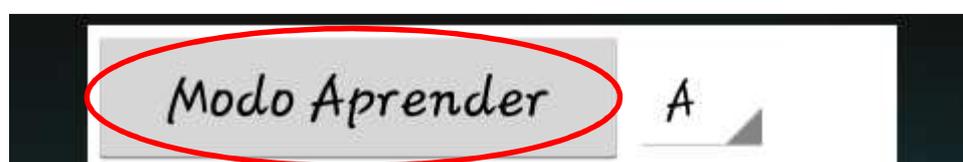


Figura 48. Modo Aprender

- Seleccionar la letra o número que se va a aprender.



Figura 49. Letra a aprender.

- Seleccionar el botón para capturar la imagen.

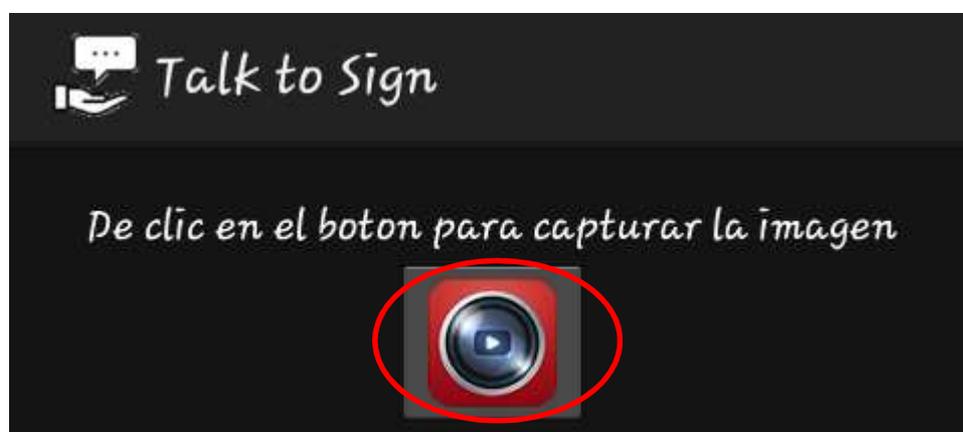


Figura 50. Botón Captura imagen.

- Tomar la fotografía. Una vez tomada la fotografía presionar el botón guardar para enviar los datos a la base de datos o Descartar para volver a tomar otra fotografía.



Figura 51. Captura de la imagen.

- La imagen tratada se mostrará en la interfaz de la aplicación.

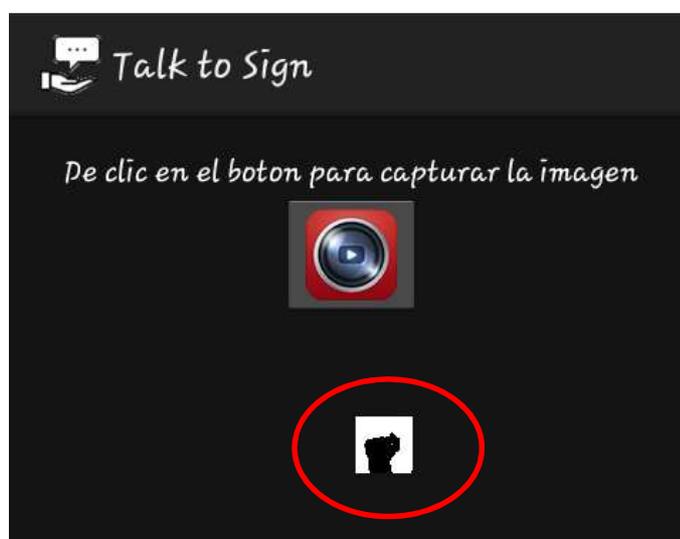


Figura 52. Imagen tratada en interfaz de aplicación.

- Seleccionar el Modo Genera Pesos, al seleccionar este modo se generaran los pesos de la letra seleccionada en base a las muestras almacenadas en la base de datos.

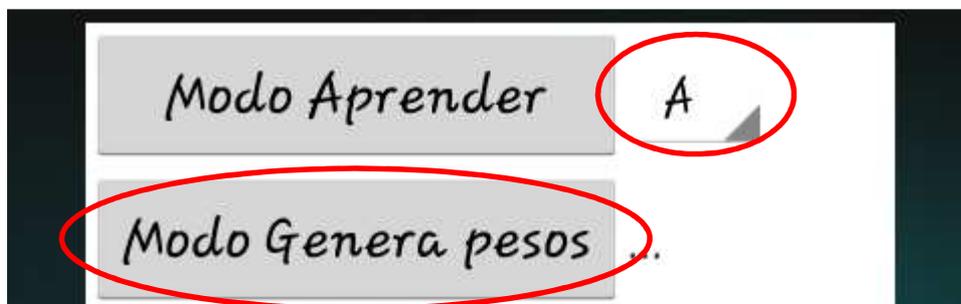


Figura 53. Modo Genera Pesos.

### 3.5.2 Reconocimiento de la Red.

Para el reconocimiento de la Red se deben realizar los siguientes pasos:

- Seleccionar el Modo Reconocer de la interfaz gráfica.

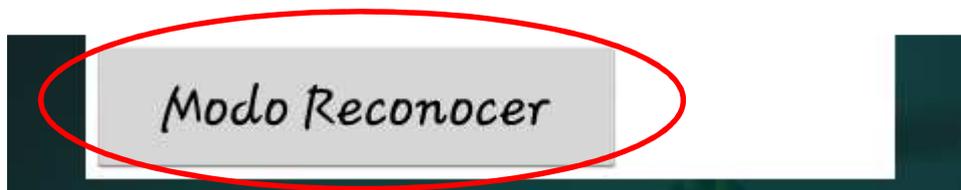


Figura 54. Modo Reconocer.

- Seleccionar la letra o número que desea reconocer.



Figura 55. Letra a reconocer.

- Seleccionar el botón para capturar la imagen.

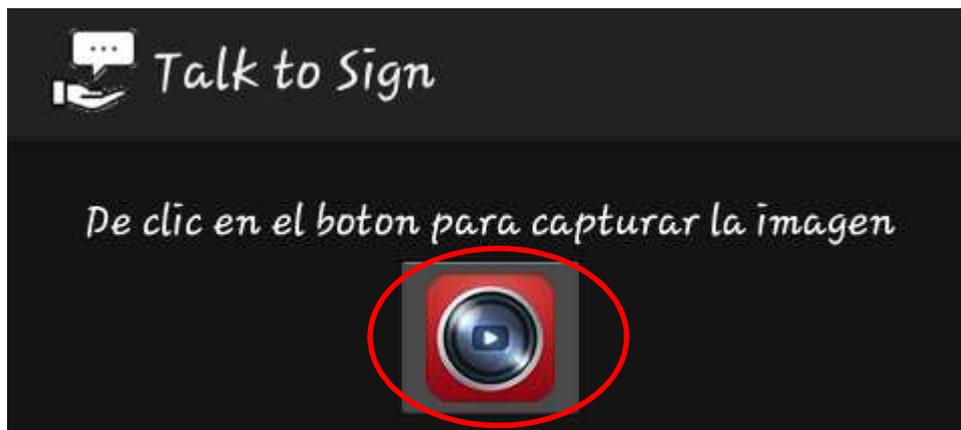


Figura 56. Botón Captura imagen.

- Tomar la fotografía. Una vez tomada la fotografía presionar el botón guardar para enviar los datos a la base de datos o Descartar para volver a tomar otra fotografía.



Figura 57. Captura de la imagen.

- La imagen tratada se mostrará en la interfaz de la aplicación.

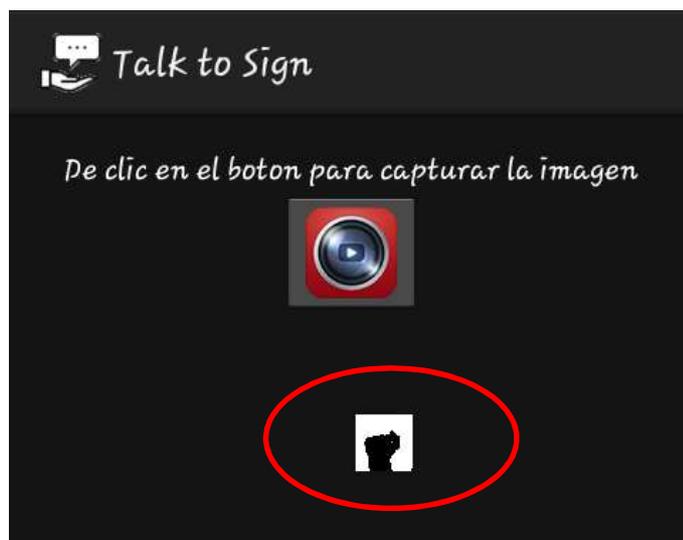


Figura 58. Imagen tratada en interfaz de aplicación.

- La respuesta se mostrara en la interfaz de la aplicación.

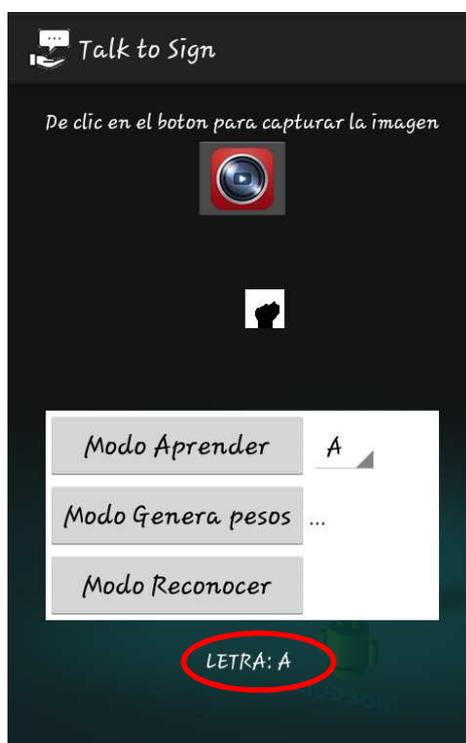


Figura 59. Respuesta desplegada en la interfaz de la aplicación.

### 3.6 Descripción de los módulos.

- **Función:** redimensionarImagenMaximo

**Archivo:** MainActivity.java.

**Entrada:** Parámetros de la imagen capturada (ancho y alto).

**Salida:** Matriz

**Descripción:** Realiza el redimensionamiento de la imagen que ingresa, devolviendo una matriz que servirá como entrada para el servidor de inteligencia artificial.

- **Función:** onCreate

**Archivo:** MainActivity.java.

**Entrada:** Botón

**Salida:** Redirección a cada uno de los procesos indicados en cada botón.

**Descripción:** Se crean todas las listas, botones, e imágenes definidas en la parte grafica de Android. Se define la acción de cada botón para una respectiva función.

- **Función:** onActivityResult

**Archivo:** MainActivity.java.

**Entrada:** Imagen

**Salida:** Imagen después del tratamiento.

**Descripción:** Se realiza el tratamiento de la imagen, convirtiéndola a escala de grises, realizando la binarización y la redimensión de la imagen a una nueva de 40x40 píxeles

- **Función: DoPOST**

**Archivo:** MainActivity.java.

**Entrada:** Letra y código.

**Salida:** Conexión hacia el servidor.

**Descripción:** Realiza la conexión https hacia el servidor de inteligencia artificial y base de datos.

- **Función: onPostExecute**

**Archivo:** MainActivity.java.

**Entrada:** Bits.

**Salida:** Letra.

**Descripción:** Devuelve la letra que obtiene del servidor de inteligencia artificial y la despliega en la pantalla.

- **Función: Aprender.**

**Archivo:** aprender.php.

**Entrada:** variable Modo, letra.

**Salida:** Guarda la trama para ser procesada.

**Descripción:** Guarda las tramas captadas por el dispositivo móvil en la base de datos para ser tomadas como muestras de aprendizaje.

- **Función: Libdata.**

**Archivo:** libdata.php.

**Entrada:** Petición.

**Salida:** Conexión.

**Descripción:** Establece la conexión hacia la base de datos, especifica clave y usuario para el ingreso a la base.

- **Función: Reconocer.**

**Archivo:** reconocer.php.

**Entrada:** Petición.

**Salida:** Devuelve la letra o número.

**Descripción:** Muestras de aprendizaje, número de entradas, numero de capas y los 8 bits de salida (letra o número). Aquí se encuentra desarrollado el Algoritmo back-propation de inteligencia artificial, el cual es capaz de reconocer las letras del lenguaje de señas. Toma los pesos aprendidos de las letras y hace la comparación con el patrón que llega del dispositivo móvil.

- **Función: Reconocer Pesos.**

**Archivo:** reconocer\_pesos.php.

**Entrada:** Letra.

**Salida:** Pesos de la letra.

**Descripción:** Devuelve los pesos de una letra, generado en base al número de letras almacenadas en la base de datos.

- **Función: Talk to Sign.**

**Archivo:** talktosign.php.

**Entrada:** variable Modo.

**Salida:** Activa el modo de funcionamiento de acuerdo a la variable.

**Descripción:** Activa en el dispositivo móvil los botones de “Aprender”, “Genera Pesos” o “Reconocer”, con el botón aprender se envía el patrón de la letra al

servidor de base de datos. Y con el botón de reconocer se envía el arreglo de 1600 bits al web server por protocolo http.

## CAPÍTULO IV

### PRUEBAS DE USO

#### 4.1 Entrenamiento del Sistema.

##### 4.1.1 Mejor desempeño de back-propagation.

A menudo se dice que el diseño de una red neuronal usando el algoritmo back-propagation es más un arte que una ciencia en el sentido en que el ajuste de muchos de los factores involucrados en el diseño es el resultado de la experiencia personal. Si bien hay algo de verdad en esto, existen métodos que mejoran significativamente la performance del algoritmo back-propagation como se describe a continuación:

- a. Actualización secuencial versus batch: El modo secuencial del algoritmo back-propagation es más rápido computacionalmente que el modo batch. Esto es especialmente cierto cuando el set de entrenamiento es grande y altamente redundante.
- b. Maximizar el contenido de información: Como regla general, cada ejemplo de entrenamiento presentado al algoritmo back-propagation se debe elegir basándose en que su contenido de información sea lo más grande posible para la tarea dada.

Dos modos de lograr este objetivo son:

- El uso de un ejemplo que resulte en el mayor error de entrenamiento.
- El uso de un ejemplo que es radicalmente diferente de aquellos previamente usados.

En las tareas de clasificación de patrones usando el aprendizaje back-propagation secuencial, una técnica sencilla que se utiliza comúnmente es randomizar el orden en el cual se presentan los ejemplos al Perceptrón multicapa de un epoch al siguiente. Idealmente, la randomización asegura que los ejemplos sucesivos en un epoch presentado a la red raramente pertenecen a la misma clase.

- c. La función de activación: Un Perceptrón multicapa entrenado con el algoritmo back-propagation puede, en general, aprender más rápido cuando la función de activación sigmoideal construida dentro del modelo de neurona es anti simétrica, comparado a cuando ésta es no-simétrica.

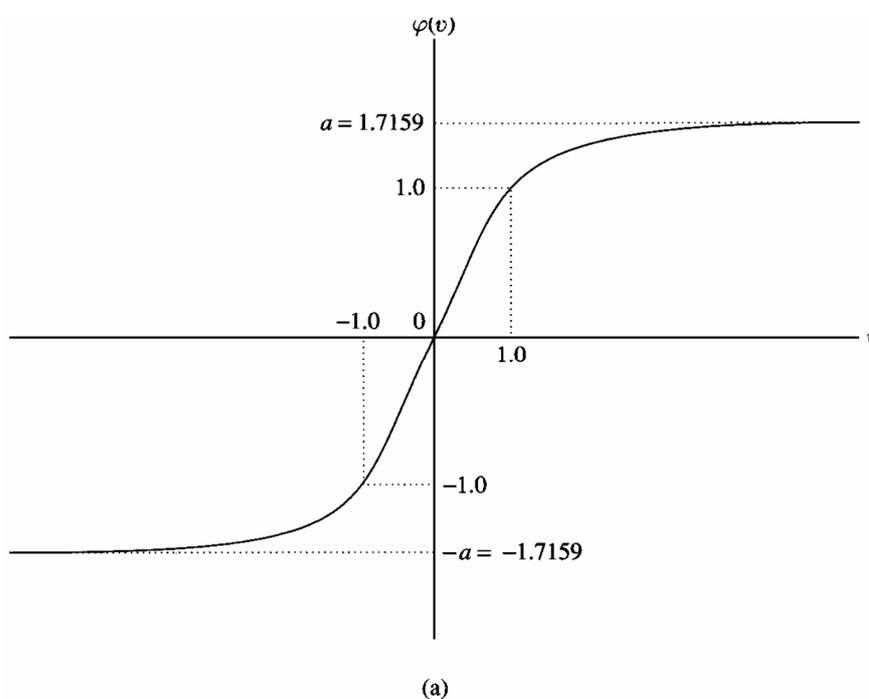


Figura 60. Función de activación anti simétrica.

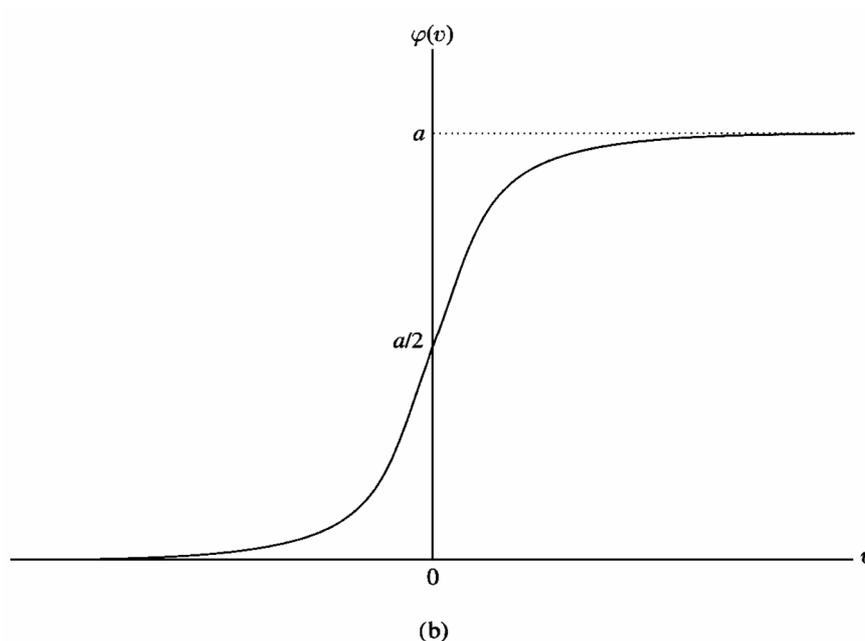


Figura 61. Función de activación no simétrica.

- d. Valores objetivo: Es importante que los valores objetivo (las respuestas deseadas) sean elegidos dentro del rango de la función de activación sigmoide. De lo contrario el algoritmo back-propagation tiende a llevar a infinito a los parámetros libres de la red, y por lo tanto el proceso de aprendizaje se hace más lento al ser dirigido por neuronas ocultas en saturación.
- e. Normalizar las entradas: Cada variable de entrada debería ser preprocesada de manera tal que su valor medio, promediado sobre el set de entrenamiento completo, esté cercano a cero
- f. Inicialización: Cuando a los pesos se les asignan valores iniciales grandes, es altamente probable que las neuronas de la red entren en saturación. Si esto ocurre, los gradientes locales en el algoritmo back-propagation asumen valores

pequeños, lo cual a su vez causará que el proceso de aprendizaje sea más lento. Sin embargo, si a los pesos se les asignan valores iniciales chicos, el algoritmo back-propagation puede operar en un área muy aplanada alrededor del origen de la superficie de error; esto es particularmente cierto en el caso de las funciones de activación anti simétricas como la función tangente hiperbólica.

- g. Aprender desde pistas: El proceso de aprendizaje desde ejemplos puede ser generalizado para incluir aprendizaje desde pistas, que puede lograrse al permitir la inclusión de información a priori que puede tener de la función en el proceso de aprendizaje.
- h. Tasa de aprendizaje: Todas las neuronas en un Perceptrón multicapa idealmente aprenderían a la misma velocidad. Las neuronas con muchas entradas deberían tener una tasa de aprendizaje menor que las neuronas con pocas entradas de manera de mantener un tiempo de aprendizaje similar para todas las neuronas de la red.

## **4.2 Pruebas del sistema.**

Para las pruebas del sistema se ha utilizado la función de activación. En el aprendizaje back-propagation, normalmente se comienza con una muestra de entrenamiento y se usa el algoritmo de back-propagation para computar los pesos de un perceptrón multicapa al cargar tantos ejemplos de entrenamiento como sea posible dentro de la red, en general, se aprendizaje es más rápido.

#### 4.2.1 Tabla de resultados de pruebas en dispositivos móviles.

La tabla de resultados se realizó con 20 muestras de la letra A, las 10 primeras muestras son muestras correctas de la letra, mientras que las 10 siguientes son muestras erróneas.

La trama de aprendizaje se refiere a cuantas muestras de la letra A están almacenadas en la base de datos, en este caso se almacenó 4 muestras. La trama de error se refiere a cuantas muestras de error están almacenadas en la base de datos, en este caso se almacenó 1 muestra.

En el resultado se indicara 1 como resultado esperado y 0 como resultado incorrecto.

Tabla 4. Tabla de resultados de pruebas letra A

No. muestras	Letra	Tramas de Aprendizaje	Trama de error	Resultado
1	A	4	1	1
2	A	4	1	1
3	A	4	1	1
4	A	4	1	1
5	A	4	1	1
6	A	4	1	1
7	A	4	1	1
8	A	4	1	1
9	A	4	1	1
10	A	1	1	0
11	A	4	1	1
12	A	4	1	1
13	A	4	1	1
14	A	4	1	1
15	A	4	1	1
16	A	4	1	1
17	A	4	1	1
18	A	4	1	1
19	A	4	1	1
20	A	4	1	1

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones.

- La aplicación únicamente es capaz de reconocer símbolos a partir de imágenes previamente aprendidas, en caso de no reconocer la imagen tomada la aplicación no devolverá ninguna letra, por lo que puede ser la base para una aplicación que reconozca palabras completas.
- Desde el punto de vista de objetivos se consiguió una aplicación que sea capaz realizar un tratamiento digital de la imagen con un formato definido y mediante una red neuronal con 3 capas identificar la letra o número que la persona está haciendo.
- No todo el conjunto de símbolos del lenguaje de signos internacional ha podido ser tratado como se planteó originalmente, debido a que los símbolos de las letras como como la j, k, ñ, q, x, z que necesitan de movimiento por lo que se podría en un futuro adaptar el sistema a ser capaz de identificar movimiento.
- La variación en la iluminación y la variación del color de fondo pueden provocar imposibilidad para reconocer ciertas imágenes porque el entrenamiento no le aporta suficiente información a la red. Esto se debe a que al momento de realizar el tratamiento de la imagen el brillo o un color claro de fondo se van a mostrar como "negro" y la aplicación va a devolver error.
- La aplicación al reconocer patrones similares (como la a, e, s, t) devuelve error o la primera letra que encuentre, en este caso la a, esto se debe a que la variación

entre las letras es mínima, para solventar esto se requiere de más aprendizaje y entrenamiento para cada una de las letras.

- Si las constantes (tasa de aprendizaje, inercia y capas) que representan al parámetro de aprendizaje no se encuentran bien ajustadas, pueden surgir varias dificultades y efectos no deseados, como la convergencia lenta o la inestabilidad del algoritmo que produce oscilaciones en el aprendizaje.
- Al momento de realizar el análisis para el número de capas se debe tener en cuenta que a mayor número de capas la aplicación va a tardar más en responder. Por lo que se decidió implementar con 3 capas.
- Se decidió realizar una aplicación de tres niveles para mayor velocidad en la respuesta, dos de los niveles se encuentran en la nube, el de inteligencia artificial y el de Base de datos, si estos se implementaban en el dispositivo móvil su tiempo de procesamiento aumentaría, y por ende su tiempo de respuesta.
- La exactitud de la respuesta no depende de la resolución de la cámara del dispositivo móvil, ya que la imagen capturada sea de la resolución que sea pasa por un tratamiento de la imagen para convertirla en bits de unos y ceros.

## **5.2 Recomendaciones.**

- Se podría realizar una segunda versión en la cual la aplicación reconozca palabras completas en tiempo real captando por video.
- Al momento de devolver la respuesta, la aplicación podría reconocer la letra que está ingresando a pesar de no ser la seleccionada.

- La clave para mejorar la habilidad del sistema es cambiar la forma en que se codifican las redes neuronales. La implementación actual requiere se especifiquen de antemano el número máximo de nodos y capas ocultas. Las implementaciones futuras de este sistema deberían explorar la codificación indirecta de las topologías sin especificar los nodos y capas ocultas.
- Sería importante repetir los experimentos presentados aquí para entradas diferentes como objetos, de esta manera también aumentaría el procesamiento y la base de datos.

## BIBLIOGRAFÍA

B-Touch. (2013). *Cell phone with Braille touchscreen is an iPhone for the blind*. Recuperado el 12 de enero de 2014, de [http://www.dvce.com/archives/2009/06/b\\_touch\\_cell\\_ph.php](http://www.dvce.com/archives/2009/06/b_touch_cell_ph.php)

Owasys 22C. (2003). *Terminal dirigido a personas ciegas o con dificultades visuales*. Recuperado el 12 de enero de 2014, de [http://www.adndesign.es/ADN\\_pd\\_Owtma\\_22C.html](http://www.adndesign.es/ADN_pd_Owtma_22C.html)

MobileASL. (Febrero de 2014). *Dispositivo para discapacidad auditiva*. Recuperado el 12 de enero de 2014, de <http://mobileasl.cs.washington.edu/>

T-Loop de Vodafone. (s.f.). *Dispositivo para discapacidad auditiva*. Recuperado el 12 de enero de 2014 <http://www.vodafone.es/conocenos/es/sala-de-prensa/notas-de-prensa/vodafone-espana-anuncia-la-disponibilidad-de-t/>

Apple. (2014). *Su visión de lo que puede hacer está a punto de cambiar*. Recuperado el 12 de enero de 2014 <http://www.apple.com/accessibility/ios/#hearing>

Melovibe. (2014). *Dispositivo para discapacidad auditiva*. Recuperado el 12 de enero de 2014 [https://play.google.com/store/apps/details?id=com.orange.melovibe&hl=es\\_419](https://play.google.com/store/apps/details?id=com.orange.melovibe&hl=es_419)

Deaf – Hearing Chat. (2014). *Dispositivo para discapacidad auditiva*. Recuperado el 12 de enero de 2014 [https://play.google.com/store/apps/details?id=g.example.android.BluetoothChat&hl=es\\_419](https://play.google.com/store/apps/details?id=g.example.android.BluetoothChat&hl=es_419)

Magic Glove. (2012). *Dispositivo para discapacidad auditiva*. Recuperado el 12 de enero de 2014 <http://tecnotrenes.blogspot.com/2012/01/magic-glove-sistema-de-dictado-para.html>

Atendiendo Necesidades. (2012). *Discapacidad: tipos y características*. Recuperado el 20 de enero de 2014 <http://atendiendonecesidades.blogspot.com/2012/11/distintos-tipos-de-discapacidad-y-sus-caracteristicas.html>

PDHRE. (s.f.). *Derechos humanos de las personas con discapacidades distintas*. (s.f.). Recuperado el 22 de enero de 2014 <http://www.pdhre.org/rights/disabled-sp.html>

Andes. (2012). *Vicepresidencia del Ecuador impulsa el primer Diccionario Oficial de Lengua de Señas Ecuatoriana*. Recuperado el 23 de enero de 2014 <http://www.andes.info.ec/es/sociedad/7820.html>

Sordos Ecuador. (s.f.). *Lenguaje de señas*. Recuperado el 23 de enero de 2014 <http://www.sordosecuador.com/lengua-de-senas/>

La hora. (2 de septiembre de 2012). *Lengua de señas ecuatoriana*. Recuperado el 25 de enero de 2014 de

[http://www.lahora.com.ec/index.php/noticias/show/1101386618/-1/Lista\\_Lengua\\_de\\_Se%C3%B1as\\_ecuatoriana\\_.html#.UglHpDI99sk](http://www.lahora.com.ec/index.php/noticias/show/1101386618/-1/Lista_Lengua_de_Se%C3%B1as_ecuatoriana_.html#.UglHpDI99sk)

Samsung. (s.f.). *Características de Android*. Recuperado el 25 de enero de 2014 de <http://www.samsung.com/es/article/android-2-2-os-explained>

Android. (4 de julio de 2011). *Arquitectura de Android*. Recuperado el 25 de enero de 2014 de <http://androideity.com/2011/07/04/arquitectura-de-android/>

Android. (s.f.). *Arquitectura de Android*. Recuperado el 25 de enero de 2014 de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Android. (4 de julio de 2011). *Sensores disponibles en Android*. Recuperado el 25 de enero de 2014 de <http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/36-unidad-5-entradas-en-android-teclado-pantalla-tactil-y-sensores/154-los-sensores>

Java. (s.f.). *Que es Java*. Recuperado el 30 de enero de [http://www.java.com/es/download/whatis\\_java.jsp](http://www.java.com/es/download/whatis_java.jsp)

Android. (2014). *Android Características*. Recuperado el 30 de enero de 2014 de [http://es.wikipedia.org/wiki/Android#Dise.C3.B1o\\_y\\_desarrollo](http://es.wikipedia.org/wiki/Android#Dise.C3.B1o_y_desarrollo)

Pazmiño, A. (2010). *Construcción e implantación del sistema de administración de cartera de proveedores, clientes y productos en el área de la conectividad y redes de comunicaciones*. Recuperado el 2 de febrero de 2014 de <http://repositorio.puce.edu.ec/handle/22000/3767>

Fiszelew, A. (2002). *Generación automática de redes neuronales con ajuste de parámetros basado en algoritmos genéticos*. Recuperado el 10 de febrero de 2014 de

Sangolquí, Junio 25 de 2014

**ELABORADO POR:**

---

Srta. María Gabriela Vintimilla Sarmiento

**DIRECTOR DE LA CARRERA DE ELECTRÓNICA, REDES Y  
COMUNICACIÓN DE DATOS:**

---

Dr. Nikolai Espinosa.