



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“RECONOCIMIENTO DE CARACTERES EN IMÁGENES
DIGITALIZADAS Y SINTETIZACIÓN DE VOZ USANDO UN
DISPOSITIVO MÓVIL CON SO ANDROID”**

INFORME DE PROYECTO DE GRADUACIÓN

Previa a la obtención del título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS MULTIMEDIA**

PRESENTADA POR:

**CARPIO OCHOA SAÚL STEVEN
LAVAYEN IÑAMAGUA JUAN FRANCISCO**

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

Agradecemos:

A Dios por darnos una familia que con su cariño y apoyo han sido parte fundamental de cada uno de los éxitos de nuestra vida.

A nuestro Director el Ph.D. Xavier Ochoa, por su colaboración y soporte incondicional durante todo el proyecto.

A nuestros amigos y compañeros que de alguna u otra forma contribuyeron a que este proyecto se lleve a cabo.

DEDICATORIA

A mis queridos padres por haberme enseñado lo más importante de la vida, los valores de la honestidad y la persistencia, que me han guiado para cumplir todos mis objetivos.

Saúl Carpio Ochoa

A Dios por haberme permitido cumplir con este período de formación académica que culmina con la presentación y sustentación de este proyecto.

A mis padres y hermanos, por su amor incondicional, apoyo constante y motivación fundamental proporcionados a lo largo de mi formación personal y estudiantil.

Juan Lavayen Iñamagua

TRIBUNAL DE SUSTENTACIÓN

Ph.D. Sixto García A.

PRESIDENTE DEL TRIBUNAL

Ph.D. Xavier Ochoa Ch.

**DIRECTOR DEL PROYECTO
DE GRADUACIÓN**

MSc. Marisol Villacrés F.

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

Carpio Ochoa Saúl Steven

Lavayen Ñamagua Juan Francisco

RESUMEN

El amplio campo de estudio que constituye la aplicación de la tecnología actual y el desarrollo de otras nuevas, en el aspecto de solucionar o al menos facilitar el normal desenvolvimiento de algunas actividades para las personas con ciertas discapacidades sensoriales; en el presente trabajo, nos enfocamos a aquellas con discapacidades del tipo visual.

Estas personas, generalmente, y dependiendo del nivel de afectación de su sentido de la vista, necesitan diferentes herramientas que les permitan captar información visual del entorno que los rodea, información presente en carteles, señales de tránsito, anuncios en la vía pública, etc.

La solución propuesta en el presente proyecto como alternativa de respuesta a lo antes expuesto, consiste en la construcción de una aplicación para dispositivos móviles Android que permita al usuario tomar fotografías y extraer, dependiendo de las condiciones de la captura, mismas que serán detalladas más adelante; el texto contenido en la imagen a través de la ejecución de un algoritmo OCR adaptado para el contexto local, haciendo uso de herramientas y metodologías de procesamiento de imágenes con el fin de optimizar este proceso.

Una vez realizado este proceso y obtenido el texto, se somete a una corrección semántica con el fin de aumentar precisión del resultado al eliminar posibles errores en las palabras reconocidas por la aplicación; Una

vez corregido, dicho texto será devuelto al usuario en forma de voz y presentará la opción de repetir la reproducción el número de veces que considere necesario. Como función adicional, se incluye la funcionalidad de obtener el posicionamiento geográfico y retornarlo, así mismo, en forma de voz.

Para el desarrollo de esta aplicación se han integrado componentes existentes en el API nativo de Android, para manejar funcionalidades del dispositivo móvil como son el manejo de interfaz, la captura de la imagen a través de la cámara, la reproducción de voz y la obtención de ubicación vía GPS; y para las secciones como el procesamiento de la imagen y la detección OCR recurrimos a librerías especializadas de código abierto como son Leptonica y Tesseract OCR respectivamente. Dicha combinación da como resultado una aplicación que ha presentado resultados exitosos basándonos en las pruebas realizadas, manteniendo un desempeño eficiente conforme las condiciones de captura de la imagen se acercan a los valores óptimos. En el presente proyecto también se analiza la aceptación de la aplicación en temas de usabilidad, midiéndose con la experiencia de usuarios con discapacidad visual.

ÍNDICE GENERAL

RESUMEN	V
ÍNDICE GENERAL	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	X
INTRODUCCIÓN	XI
CAPÍTULO 1	1
1 JUSTIFICACIÓN Y ANÁLISIS DEL PROBLEMA	1
1.1 Antecedentes	1
1.2 Descripción del problema	3
1.3 Propuesta de solución	5
1.4 Justificación	7
CAPÍTULO 2	9
2 REVISIÓN DEL ESTADO DEL ARTE	9
2.1 Algoritmos de OCR	9
2.1.1 Pre procesamiento de la imagen	10

2.1.2 Detección de texto en la imagen	11
2.1.3 Reconocimiento de caracteres	12
2.2 Sintetización de texto	12
2.2.1 Corrección semántica	13
2.2.2 Conversión de texto en voz	13
2.3 Ubicación Geográfica	15
2.3.1 Reconocimiento de ubicación con GPS	16
CAPÍTULO 3	23
3 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN	23
3.1 Análisis y diseño de la aplicación	23
3.1.1 Diseño de la interfaz	27
3.2 Diseño de módulos	30
3.2.1 Módulo de captura y procesamiento de imagen	31
3.2.2 Módulo de reconocimiento de texto	33
3.2.3 Módulo de corrección semántica	34
3.2.4 Módulo de síntesis de texto a voz	35

3.2.5 Módulo de localización geográfica	35
3.3 Alcance del diseño	36
3.4 Diseño de pruebas	36
CAPÍTULO 4	39
4 DESARROLLO Y ANÁLISIS DE RESULTADOS	39
4.1 Implementación	39
4.2 Pruebas	45
4.2.1 Pruebas funcionales	46
4.2.2 Pruebas de usabilidad	49
4.3 Análisis de resultados	51
CONCLUSIONES	53
RECOMENDACIONES	55
ANEXOS	56
BIBLIOGRAFÍA	60

ÍNDICE DE FIGURAS

Figura 3.1.	Diagrama de Casos de Uso	25
Figura 3.2	Interfaz de Usuario de la Aplicación	28
Figura 3.3	Icono de Leer Imagen	29
Figura 3.4	Icono de Repetir Lectura	29
Figura 3.5	Icono de Localización	29
Figura 3.6	Icono de Salir	30
Figura 3.7	Diagrama de interacción de módulos	31
Figura 4.1	Binarización de imagen con Leptonica	42

ÍNDICE DE TABLAS

Tabla I	Resultados de pruebas con problemas de captura	46
Tabla II	Resultados de pruebas con parámetros medidos	47
Tabla III	Resultados de pruebas con métricas de precisión	48
Tabla IV	Resultados de pruebas de usabilidad	50

INTRODUCCIÓN

Con el desarrollo tecnológico y la constante innovación en dispositivos móviles, se busca ofrecer nuevas herramientas útiles para personas con discapacidades de todo tipo; en el presente proyecto apuntamos a crear una aplicación para personas con discapacidad visual.

Generalmente, y dependiendo del nivel de afectación de su sentido de la vista, necesitan diferentes herramientas que les permita identificar información visual del entorno que los rodea, como letreros, señales y otros

Con la finalidad de suplir estas necesidades, existen sistemas de adquisición y procesamiento digital de imágenes, con los cuales se obtiene diversas herramientas para el reconocimiento de caracteres; estos sistemas, llamados OCR son, básicamente las que permiten a través de un dispositivo electrónico, obtener los caracteres contenidos en las imágenes capturadas.

Un OCR se basa en algoritmos que convierten las letras representadas en una imagen digital en caracteres ASCII, para poder posteriormente ser leído y editado por algún procesador de textos común y corriente.

Como parte complementaria a los OCR se hace uso de sistemas de sintetización de voz, que mediante un conjunto de algoritmos producen los sonidos de los vocablos del idioma español; de esta manera completamos el proceso detección y lectura de textos en imágenes.

Se plantea hacer uso de todas estas librerías, aplicaciones y otros desarrollos para crear una aplicación que permita a los usuarios con discapacidad visual parcial, obtener lectura de textos capturados por medio de un dispositivo móvil de una manera sencilla y efectiva.

CAPÍTULO 1

1. JUSTIFICACIÓN Y ANÁLISIS DEL PROBLEMA

En este capítulo se revisan antecedentes del problema que se desea solucionar, de las soluciones existentes y la justificación de este trabajo que consiste en la implementación de una aplicación para ayudar a la lectura de textos a las personas con discapacidad visual parcial.

1.1. Antecedentes

En Ecuador según el Censo 2010 existe alrededor de 185.600 personas con discapacidad visual, de ellos un 8% tienen acceso a internet y usan teléfonos móviles [1].

Para este segmento de la población, la interacción juega un papel preponderante, dando facilidades en la usabilidad y experiencia del usuario, para promover la accesibilidad en el

ámbito de los ordenadores e internet, existen programas desarrollados específicamente para facilitarles el uso del ordenador en función de su discapacidad.

Entre ellos podemos destacar:

- **Hal para Windows en español:** Permite a los usuarios con discapacidad visual escuchar todo lo que ocurre o está en su ordenador [2].
- **Jaws para Windows:** Un efectivo lector de pantallas en cuanto a funcionamiento y compatibilidad. Es una aplicación que permite a las personas con discapacidad visual procesar textos y caracteres de los programas que se ejecuten en cualquier versión de Windows [3].
- **Open Book:** Software que provee todo lo necesario para usar el ordenador personal como una máquina de lectura, con características de ayuda para baja visión. Su funcionamiento básico consiste en que al escanear un texto impreso, éste automáticamente se lee en voz alta. Acepta documentos en caracteres impresos de cualquier tipo y en contraste [4].

- **Windows-Eyes:** Es un avanzado programa lector de pantalla. Tiene la capacidad de leer todo tipo de páginas web y documentos en formato PDF. Compatibilidad con Windows 9X/Me. Adaptable a cualquier tipo de sintetizador externo [5].

Con el paso de los años los teléfonos móviles y la tecnología desarrollada para ellos, ofrece una mayor oferta de opciones de inclusión para personas con discapacidades de todo tipo. Existen aplicaciones con reconocimiento automático de texto 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 déficit del habla.

1.2. Descripción del problema

La discapacidad visual usualmente se relaciona a la ausencia total del sentido de la vista, pero en realidad existe un rango de discapacidad visual en el cual se encuentran ubicados niños, jóvenes y adultos con baja calidad de visión o que presentan patologías relacionadas con el tema: miopía, astigmatismo, etc.; esta condición les impide beneficiarse de la estimulación normal

y les conlleva tener que vivir con las implicaciones propias de su discapacidad, tanto en su vida laboral y personal.

Estas personas no pueden movilizarse y realizar actividades cotidianas con normalidad a causa de su condición; actividades tales como comprar artículos en el supermercado, leer letreros o diferenciar portadas de libros; la capacidad de realizar estas y otras tareas forman parte de las actividades diarias que requieren hacer las personas con discapacidad visual.

Muchos de ellos requieren de la ayuda de otras personas que las asistan en sus actividades cotidianas, reduciendo la capacidad de ser una persona autosuficiente, en ciertos casos se respaldan en objetos como bastones y otros más tecnológicos como sensores de proximidad, para poder movilizarse.

Otra de las implicaciones de la discapacidad presente en este conjunto de individuos es la limitación que presentan para el uso de herramientas tecnológicas, es decir, no pueden hacer uso pleno de la tecnología debido a las limitantes sensoriales a las que se ven sometidos. Esto hace surgir la necesidad de crear interfaces de usuario adaptadas a dichas limitantes.

Con la evolución de la tecnología móvil la brecha que separa a este grupo de los demás, es cada vez más corta, sistemas operativos como Android y iOS ya cuentan con sus adaptaciones para discapacidades visuales y abren un mundo de aplicaciones para su beneficio utilizando medios de retroalimentación alternativos para mejorar la accesibilidad del usuario.

1.3. Propuesta de solución

La solución que se propone consiste en una aplicación de interacción sencilla que permita la captura de texto en imágenes y su posterior lectura al usuario, con el fin de proporcionar una herramienta de ayuda a personas con discapacidad visual parcial; la misma que debe cumplir con ciertos parámetros de adaptabilidad y accesibilidad para constituir una herramienta útil para dichas personas.

La función central de la aplicación consiste en el proceso de adquisición de texto a partir de una imagen digitalizada, para lo cual empleamos un algoritmo OCR previamente implementado por Google para ser utilizado en aplicaciones móviles. Si bien el OCR es el núcleo de la aplicación, un aspecto importante de la

misma constituye la interfaz, debido a que el usuario objetivo son personas con discapacidades visuales parciales, la hemos simplificado al máximo posible, dividiendo la pantalla táctil del dispositivo en cuatro partes, las cuales serán los accesos a las funcionalidades respectivas.

El módulo de sintetización de voz será el encargado de llevar la información obtenida al usuario en forma de voz, habiendo realizado previamente el análisis del texto correspondiente.

La aplicación contará con una interfaz de pantalla dividida en secciones y retroalimentación auditiva para su fácil uso, y presentará las siguientes funciones definidas:

- Captura de imagen, lectura y salida a voz de texto (bajo determinadas condiciones de distancia/iluminación)
- Repetición de lectura del último texto capturado.
- Recuperación y salida a voz de la ubicación del dispositivo a través del GPS.
- Salir de la aplicación

La aplicación será desarrollada para dispositivos móviles que cumplan con las siguientes características:

- Pantalla táctil

- Cámara
- S.O. Android (versión 2.2 o superior)
- GPS

1.4. Justificación

Hemos identificado y analizado ciertas aplicaciones que tienen como finalidad facilitar la movilidad de las personas con discapacidad visual; éstas son, en su mayoría, aplicaciones comerciales desarrolladas por empresas móviles para proporcionar esta ayuda a sus usuarios.

Estas aplicaciones, tales como: “EasyWalk”, “Walkietalkie” e “Intersection explorer” permiten, utilizando el sistema GPS del dispositivo, localizar una ubicación y obtener información para llegar a la misma; dicha información es transmitida al usuario a través de un sistema sintetizador de voz.

En cuanto a aplicaciones que hacen uso de algoritmos OCR, tenemos varios ejemplos de ellas, orientadas a diversas funciones; así es el caso de “Cam Scanner”, “Mobile Doc scanner”, entre otras, cuyo propósito es el de capturar imágenes de documentos impresos para luego digitalizarlas y transformar dichas imágenes a texto editable.

Los Lectores de códigos de barra y los denominados códigos QR, muy utilizados en la actualidad, también hacen uso de ciertos principios de procesamiento digital de imágenes, en este caso con tecnología OMR (Optical mark recognition) y podemos mencionar algunos de ellos tales como “Barcode Scanner”, “QR Scanner”, etc.

Hemos basado nuestro análisis de las aplicaciones antes mencionadas en las versiones respectivas de las mismas para dispositivos móviles que hacen uso de S.O. Android versión 2.2 en adelante.

Si bien hemos identificado diversas aplicaciones que hacen uso de tecnologías que hemos considerado para el presente proyecto, todas ellas están orientadas a diferentes usos, muy pocas de ellas reúnen todas las características que son requeridas para el problema presentado en el presente documento y adaptadas a un contexto local.

Es por esto que se justifica el desarrollo de una aplicación que reúna dichas características y se oriente específicamente al objetivo que proponemos, facilitar el desenvolvimiento de algunas actividades para las personas con discapacidad visual parcial.

CAPÍTULO 2

2. REVISIÓN DEL ESTADO DEL ARTE

En el presente capítulo revisaremos los diferentes elementos relacionados a las características del presente proyecto tales como: algoritmos, librerías, plataformas y aplicaciones existentes que cumplen funciones similares, y tienen por objetivo ayudar a las personas con discapacidad visual a realizar distintas actividades.

Entre las descripciones realizadas a continuación, buscamos explicar cada uno de los componentes esenciales de los que hará uso la solución planteada.

2.1. Algoritmos de OCR

El reconocimiento de caracteres, o conocido como OCR (Optical Character Recognition) por su acrónimo en inglés, encierra el proceso de reconocer texto alfanumérico de un lenguaje específico en una imagen; mediante cálculos

estadísticos, detección de patrones, transformado y comparaciones entre sí, se debe tener en cuenta que la exactitud de la distinción de los caracteres o símbolos depende del entrenamiento y no es 100% efectiva [6].

Los algoritmos de OCR pueden ser diferenciados según el modelo matemático que utiliza para el reconocimiento de patrones siendo los más conocidos: Redes Neuronales, método lógico, método evolutivo, método probabilista, método geométrico, entre otros; además existen softwares libres como: Cuneiformes, OCRopus, Tesseract, Ocrad y algunos propietarios como: ExperVision, FineReader, Microsoft Office Document Imagine entre otros [11].

2.1.1. Pre procesamiento de la imagen

El pre procesamiento es un paso de preparación o adecuación de la imagen que tiene como objetivo eliminar ruido o imperfección que no pertenezca al texto, para el caso de OCR, la normalización de la imagen también puede implicar un proceso de binarización.

El ruido que puede aparecer en una imagen digital, puede ser provocado por problemas de iluminación, defectos técnicos en la adquisición o el proceso de

binarizado de la imagen, se utilizan diversos algoritmos [6]:

- **Etiquetado:** para la división de la imagen en regiones de componentes conectadas.
- **Erosión / expansión:** para la eliminación de pequeños grupos de píxeles.
- **Umbralizado de histograma:** para eliminar/seleccionar los objetos más brillantes o más oscuros de la imagen.

2.1.2. Detección de texto en la imagen

Para la detección de texto, en primera instancia se debe determinar la ubicación del mismo dentro de la imagen, considerando características tales como: la disposición espacial, la cantidad de líneas, espacios en blanco entre otros; con el objetivo de mantener la fidelidad del texto en la imagen original. Este proceso debe considerar un conjunto de restricciones, con el fin de facilitar la operación de aislamiento [7]:

- Los renglones que forman el texto no deberán encontrarse sobrepuestos o inclinados, además

deberán estar separados al menos por una línea en blanco.

- Los caracteres no deben estar sobrepuestos o unidos.

2.1.3. Reconocimiento de caracteres

El primer paso para el reconocimiento de caracteres consiste en el aislamiento de cada caracter dentro de la línea o renglon, se genera una ventana rectangular en cada caracter separándolos por los espacios entre ellos, cuando el espacio en blanco es mayor, se detecta un espacio en blanco y se interpreta como una separación de palabras.

Dependiendo del modelo matemático utilizado para el reconocimiento se procede a extraer las características de la imagen del caracter para ser analizada y obtener su equivalente en texto [7].

2.2. Sintetización de texto

El proceso de sintetización de texto abarca las diferentes operaciones realizadas una vez efectuado el reconocimiento de caracteres, cuando el texto obtenido es procesado y corregido para ser devuelto por la aplicación como una respuesta auditiva.

La corrección semántica se realiza para intentar eliminar posibles errores presentados en la fase previa, de esta forma se obtiene un texto más depurado, que más adelante será convertido a voz y reproducido por el dispositivo.

2.2.1. Corrección semántica

El enfoque de las herramientas encargadas de la corrección semántica de texto, se centra en la utilización de un diccionario. La metodología habitual empieza buscando una palabra en el mismo, cuando es encontrada se pasa a la siguiente, caso contrario se selecciona la primera de entre una lista de palabras ordenadas de acuerdo a la distancia con la que fue objeto de análisis, es decir buscando la máxima proximidad a la palabra correcta.[8]

2.2.2. Conversión de texto en voz

Las aplicaciones que hacen uso de un conversor de texto a voz (CTV) dependen en alto grado de la adecuada corrección del texto que reciben como entrada. Si el texto se encuentra mal estructurado (mal escrito, mal puntuado, etc), la lectura que se obtiene del mismo será

incorrecta, lo cual puede ocasionar que el usuario, quien no ve el texto que la aplicación recibe, se sienta defraudado por esta “mala lectura”. [9]

La conversión del texto obtenido a voz se realiza mediante la generación, por medios automáticos, de una voz artificial que genera idéntico sonido al producido por una persona al leer un texto en voz alta. En otras palabras, los conversores texto a voz son sistemas que permiten la conversión de textos en voz sintética. Este proceso consta de tres fases determinadas [10]:

- **Procesamiento del texto:** Consiste en la Normalización (expansión de abreviaciones, conversión de números y fechas en texto), silabificación y conversión de los grafemas en fonemas.
- **Generación de la prosodia:** Se genera la información prosódica para producir la voz, teniendo en cuenta la entonación, duración y energía de los fonemas.
- **Generación de la voz sintética:** Se produce la voz considerando la información obtenida en las dos fases previas.

2.3. Ubicación Geográfica

Los servicios de Ubicación LBS, por sus siglas en inglés; permiten a los usuarios de telefonía móvil tener una nueva área de servicios según su posición geográfica; así mismo a los desarrolladores de software que brindan nuevas aplicaciones con información de las condiciones del tráfico, rutas, búsquedas de sitios entre otros [12].

Los servicios de Ubicación pueden ser clasificados en 3 categorías: [13].

- **Seguridad pública:** la ubicación del usuario del móvil puede ser desempeñar un papel preponderante en caso de alguna emergencia.
- **Servicios al cliente:** encierra todo el conjunto de servicios que se ofrecen al usuario del teléfono, entre estos existen:
 - Navegación
 - Ubicación de publicidad
 - Ubicación por búsquedas
 - Buscador de amigos y familia
 - Servicios empresariales:
 - Ubicación de juegos móviles, entre otros.

- **Servicios empresariales:** permite a las empresas hacer seguimiento de sus activos, optimización de entrega de productos, y demás servicios relacionados al transporte.

En mayo del 2011 Google anuncio la apertura del API de Google Places; un servicio que devuelve datos de lugares utilizando requerimientos HTTP, dando como respuesta especificaciones de ubicación como latitud/longitud coordenadas.

Los cuatro tipos de solicitudes están disponibles con el API de Google Place son [12]:

- Búsqueda de Lugares
- Detalles de Lugares
- Check-ins en Lugares
- Reportes de Lugares

El API de Google Places tiene una cantidad limitada de consultas; solo permite 1000 requerimientos diarios por usuario con su API Key, en caso de requerir más solicitudes el servicio debe ser pagado.

La ubicación del dispositivo puede ser obtenida por la red proveedora de servicios telefónicos o por Satélites (GPS).

2.3.1. Reconocimiento de ubicación con GPS

La Red Provedora de ubicación de Android determina la ubicación del usuario usando la antena de telefonía móvil y las señales Wi-Fi del dispositivo, proporcionando información sobre la ubicación de una manera que funciona en interiores y al aire libre, responde más rápido y utiliza menos energía de la batería[12].

El Sistema de Posicionamiento Global (GPS) utiliza un grupo de 24 satélites que orbitan la Tierra. GPS encuentra la posición del usuario mediante el cálculo de las diferencias en los tiempos de las señales de diferentes satélites, que se toma para llegar al receptor; las señales GPS son decodificadas, por lo que el dispositivo debe tener un receptor GPS.

GPS Asistido (A-GPS) es una tecnología para los teléfonos que integran la red móvil con el GPS para dar una mejor precisión de 5 a 10 metros. Tiene un mejor tiempo de respuesta, con mayor cobertura y útil en interiores con menos consumo de energía de la batería y requiere un menor número de satélites.

2.4. Componentes existentes

Con respecto a librerías o desarrollos de OCR en tecnología móvil podemos mencionar entre los más destacados:

- **ABBY Mobile OCR engine:** es un kit de desarrollo de software que permite integrar tecnología OCR en aplicaciones móviles tanto para Android e IOs.
- **Asprise Java OCR:** ofrece un API para utilizar en aplicaciones java, con funcionalidades de extracción de texto de documentos escaneados e imágenes.
- **Tesseract-ocr:** Es una librería open source, está considerado como uno de los motores OCR libres con mayor precisión disponible actualmente.

A continuación presentamos la descripción de las herramientas consideradas para la implementación del proyecto:

Java

Java es un lenguaje de programación orientado a objetos diseñado y desarrollado por Sun Microsystems, y actualmente propiedad de Oracle Corporation. Se lo ha seleccionado para este proyecto debido a que es el principal lenguaje que se usa para crear aplicaciones que corren en dispositivos con SO Android.

La principal ventaja del uso de Java es que, al emplear el SDK de Android la programación se vuelve mucho más sencilla, y al tratarse de la plataforma más usada para el desarrollo de aplicaciones móviles, existen numerosas herramientas gratuitas y de pago con múltiples propósitos que se pueden integrar para un fin determinado.

Android API

Android es el sistema operativo para dispositivos móviles desarrollado por la compañía Google, y ofrece a su gran cantidad de desarrolladores a nivel mundial, un API que permite utilizar en una aplicación todas las funcionalidades que brinda el sistema operativo dependiendo de la versión del mismo y del tipo de dispositivo en el que se ejecute.

Existen diferentes versiones del API, siendo estos variantes continuas en el tiempo lanzadas al público periódicamente por Google, generalmente junto con cada nueva versión del sistema operativo Android, ya que en cada nuevo “release” se incluyen nuevas características que ofrecen los dispositivos o se mejoran algunas ya existentes; así, las herramientas para los desarrolladores de esta plataforma se mantienen siempre

actualizadas y permiten explotar al máximo la funcionalidad y rendimiento del smartphone.

Hemos considerado el nivel 16 del API de Android, liberado en julio del 2012 como primera actualización del API para la versión del SO Android “JellyBean” 4.1. Dentro del API, podemos definir las herramientas específicas más importantes dentro de cada módulo, las mismas se detallan a continuación:

- **Camera**

La sección del API encargada del manejo de la cámara es fundamental dentro de la aplicación, aquí encontramos los procedimientos necesarios que permiten llevar a cabo el proceso de captura de la imagen que es el primer paso para el reconocimiento de texto.

- **Predictive Text**

Predictivetext es el nombre con el que se conoce al componente que permite realizar la corrección ortográfica a través del diccionario del SO. Esta herramienta se usa en el modo de escritura predictivo que es el más empleado entre los usuarios de dispositivos Android.

- **Text to Speech**

Text to Speech es una herramienta de Android que permite convertir cadenas de texto a voz para luego

reproducirlo a través de la salida de audio del dispositivo. Este componente es el responsable de recibir la cadena de texto extraído de la imagen y convertirlo a una señal auditiva enviada al usuario como respuesta final.

Además, esta herramienta, al ser parte del API nativo del SO Android, se utiliza frecuentemente en el desarrollo de aplicaciones que buscan adaptar la accesibilidad del dispositivo para usuarios con discapacidad visual.

- **Location manager**

Existen varias herramientas en el API de Android que se emplean para obtener la localización geográfica del dispositivo a través del GPS, Location Manager es una de ellas; esta clase provee funciones bastante sencillas de integrar en una aplicación y hace uso de los datos proporcionados por Google Maps para devolver información precisa en tiempo real.

- **Leptonica**

Leptonica es una librería de open source que se emplea para el procesamiento de imágenes en distintas plataformas, el API proporcionado por esta librería permite realizar el procesamiento de una imagen determinada. Dicho procesamiento consiste en la

binarización y filtrado de la imagen entre otros procesos específicos con diferentes fines dependiendo del caso.

- **Tesseract OCR**

Tesseract es un motor OCR originalmente creado por la compañía Hewlett Packard como software propietario a mediados de los años 80; posteriormente liberado en el 2005 y siendo actualmente desarrollado por Google, es en la actualidad el motor OCR de código abierto más potente que existe.

Al utilizar Tesseract conjuntamente con el API de Android, y al ser ambas características desarrolladas por Google se obtiene la ventaja de una fácil adaptabilidad a la plataforma y óptimo uso de las herramientas disponibles.

Otra de las principales ventajas del uso de Tesseract es la facilidad para el manejo del idioma, teniendo la posibilidad de configurar ciertos parámetros de la herramienta con el fin de optimizar la precisión del reconocimiento de cada caracter.

CAPÍTULO 3

3. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

En este capítulo se muestra el análisis realizado para el desarrollo de la solución propuesta; las herramientas, configuraciones y otros factores importantes que se deben considerar.

Describiremos la estructura y funcionamiento de cada módulo y la interacción entre ellos, terminando con el diseño de pruebas con el que se evaluará la aplicación.

3.1. Análisis y diseño de la aplicación

La solución que se propone consiste en una aplicación de interacción sencilla que permita la captura de texto en imágenes y su posterior lectura al usuario, con el fin de proporcionar una herramienta de ayuda a personas con discapacidad visual parcial; la misma que debe cumplir con ciertos parámetros de adaptabilidad y accesibilidad para constituir una herramienta útil

para dichas personas al momento de realizar actividades específicas bajo condiciones determinadas.

Esta aplicación será desarrollada para dispositivos con SO Android, dado que estos constituyen la herramienta ideal para este propósito gracias a su portabilidad y popularidad entre el público en general, así como la flexibilidad que presenta para el desarrollo gracias a los diferentes componentes y librerías preexistentes que se pueden identificar en esta plataforma.

El diseño de la interfaz principal de esta aplicación constituye un elemento de importancia dentro de la misma debido a las características del usuario final, lo cual exige una especial atención a la adaptación de la interfaz común para dispositivos táctiles.

La aplicación presentará las siguientes funciones definidas, como se muestra en la Figura 3.1 y se describe a continuación:

- Captura de imagen, lectura y salida a voz de texto (bajo determinadas condiciones de distancia/iluminación)
- Repetición de lectura del último texto capturado.
- Recuperación y salida a voz de la ubicación del dispositivo a través del GPS.
- Salir de la aplicación

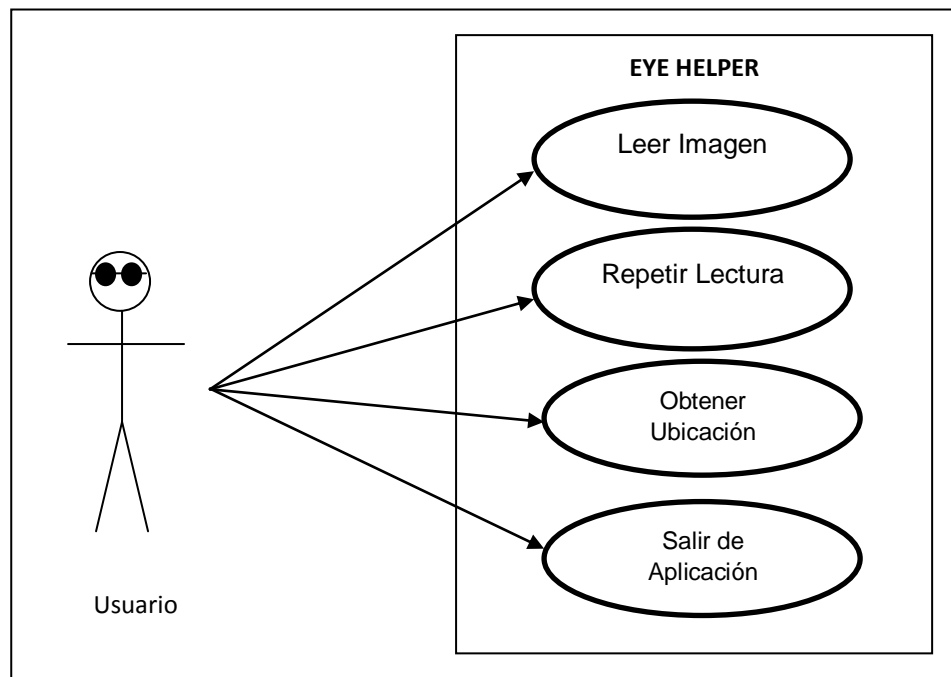


Figura 3.1. Diagrama de Casos de Uso

La aplicación será desarrollada para dispositivos móviles con las siguientes características:

- Pantalla táctil
- Cámara
- S.O. Android (versión 2.2 o superior)
- GPS

Los parámetros definidos para el correcto funcionamiento de la aplicación constituyen un factor crítico en el alcance del mismo, puesto que constituyen el marco de eficacia establecido; dichos parámetros consisten en:

- **Iluminación:** Natural o artificial, en caso de que la cámara del dispositivo no cuente con flash, la iluminación del entorno deberá ser la adecuada para que en la imagen capturada pueda distinguirse claramente el texto.
- **Fondo y Contraste:** El Fondo de la imagen en donde se encuentre el texto no debe contaminar el mismo o confundirse con él, es decir, se requiere un fondo plano, libre de otra simbología o ilustración que pueda generar ruido, y con el contraste adecuado que permita identificar fácilmente el texto a leer.
- **Color:** El color está relacionado directamente con el contraste de la imagen, y del tono e intensidad del mismo depende la capacidad de la aplicación para ejercer correctamente el proceso de binarización, siendo así necesaria una combinación en la que el texto destaque de su fondo mediante el color de ambos.
- **Fuente:** El tipo de fuente es un factor determinante, ya que los símbolos reconocidos mediante el algoritmo de OCR son limitados y se encuentran supeditados a los factores antes mencionados, es así

que el mejor desempeño de la aplicación se obtiene al usar fuentes de tipo imprenta, no manuscritos, y empleando símbolos del alfabeto latino para representar los fonemas.

3.1.1. Diseño de la interfaz

La interfaz de la aplicación consiste en una pantalla seccionada en 4 cuadrantes de igual proporción, cada una de estas secciones será ilustrada como un botón con su respectivo icono (diferente para cada una), además de un campo de texto donde se muestra el texto leído como se puede observar en la Figura 3.2.

Estos botones serán los accesos a las respectivas funcionalidades de la aplicación, como se explica a continuación:

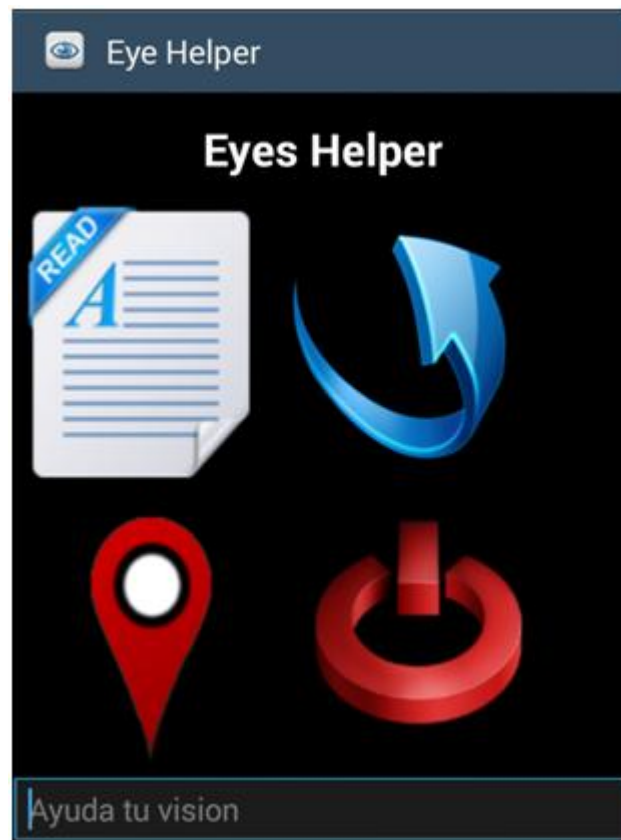


Figura 3.2 Interfaz de Usuario de la Aplicación

- **Leer Imagen.-** Se activará la cámara del dispositivo para tomar una foto del texto deseado, cuando la foto está lista, automáticamente se realiza la lectura y síntesis a voz del texto obtenido.



Figura 3.3Icono de Leer Imagen

- **Repetir lectura.-** Se repite la lectura, es decir, se sintetiza a voz el último texto extraído de una imagen



Figura 3.4Icono de Repetir Lectura

- **Obtener localización.-** Se recupera la localización geográfica actual a través del GPS del dispositivo y se sintetiza a voz dicha información



Figura 3.5Icono de Localización

- **Salir.-** Cierra la aplicación



Figura 3.6Icono de Salir

3.2. Diseño de módulos

A continuación se describen las características de los distintos módulos de los que hace uso la aplicación, pudiendo trabajar independientemente cada uno de ellos o en un funcionamiento grupal dependiendo de la funcionalidad requerida; estos representan uno a uno los pasos del flujo funcional de la aplicación, ya que en cada uno de ellos se realiza un proceso cuyo resultado podrá alimentar al siguiente, estableciendo la ruta hasta el resultado final del conjunto.

La estructura modular de la aplicación permite la mejor comprensión de su funcionamiento y el rol que cumple cada módulo dentro de la misma, como se puede observar en la Figura 3.7

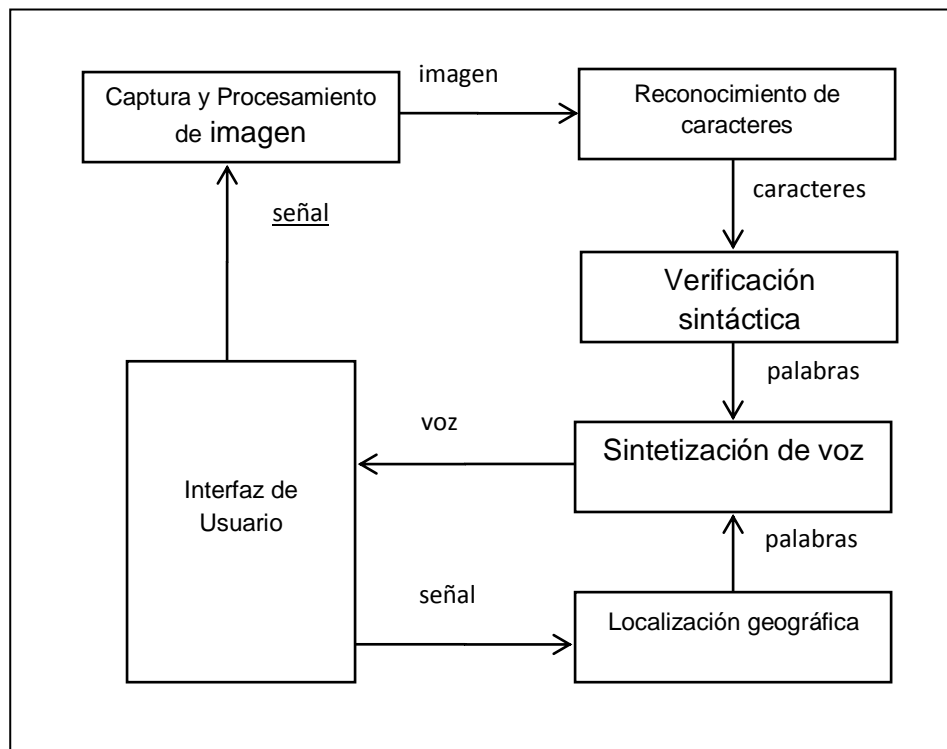


Figura 3.7 Diagrama de interacción de módulos

3.2.1. Módulo de captura y procesamiento de imagen

Este módulo hace uso de la cámara del dispositivo a través del manejo de las funciones correspondientes presentes en el API nativo del sistema Android. La imagen es capturada y guardada en la memoria del teléfono para su posterior uso.

La forma en que se captura la imagen dentro de la aplicación es igual a la acción de tomar una foto con la aplicación de cámara por defecto instalada en el dispositivo, teniendo así la posibilidad de emplear las

herramientas prestadas dependiendo del tipo, modelo y marca de dispositivo empleado, entre éstas están: flash, enfoque, tipo de captura, etc.

Una vez que se ha capturado con éxito la imagen, esta se almacena en la memoria del teléfono mientras la aplicación haga uso de ella.

Antes de aplicar el reconocimiento de caracteres, se necesita procesar la imagen con el fin de prepararla para el correcto desarrollo del siguiente proceso; en este módulo se hace uso de “Leptonica”, una librería desarrollada en java que posee potentes funcionalidades para el procesamiento de imágenes.

La imagen capturada se transforma en un mapa de bits, se somete a un proceso de binarización realizada mediante un método global Otsu, luego a los procesos de segmentación y normalización previas al proceso de reconocimiento de caracteres. Este conjunto de procesos constituyen la parte central de la aplicación y serán analizados en la sección de implementación del proyecto en el siguiente capítulo del presente documento.

Una vez realizados estos procesamientos, la imagen es devuelta lista para ser trabajada por siguiente módulo.

3.2.2. Módulo de reconocimiento de texto

Este módulo hace uso de un algoritmo OCR para obtener los caracteres existentes en una imagen sometida al correspondiente pre procesamiento.

Para el reconocimiento de caracteres dentro de la imagen se utilizan las funciones provistas por el API para android de la librería “Tesseract OCR” que constituye una importante herramienta ya que permite fácil acceso a las posibilidades que presenta dicha librería. Tesseract OCR es una librería open source multiplataforma y es, hoy por hoy, la más potente herramienta OCR de código abierto, actualmente mantenida por Google, presenta múltiples y continuas mejoras sobre otras opciones OCR; como por ejemplo la continua incorporación de nuevos lenguajes lo que aumenta su flexibilidad y adaptabilidad a los propósitos requeridos.

La librería además emplea archivos de parametrización de lenguaje, esto es por la necesidad de adaptabilidad a la simbología de cada lenguaje para establecer coincidencias de dichos símbolos con los presentados en

la imagen; en el presente proyecto hemos establecido la configuración de Tesseract OCR para el idioma español.

3.2.3. Módulo de corrección semántica

Si bien algunas características de corrección son utilizadas en el módulo OCR, es preciso realizar una corrección post-reconocimiento, donde el texto obtenido es sometido a un filtro utilizando una característica del sistema android que hace uso del diccionario para buscar similitudes más próximas para los vocablos donde se sospecha de posibles errores en la detección del texto exacto. Así, se garantiza la mayor fidelidad de la semántica obtenida a través de la aplicación.

La herramienta, denominada “predictivetext” es un elemento común en la escritura no sólo de smartphones Android sino en los teléfonos celulares en general, y funciona de manera sencilla estableciendo el grado de similitud de palabras escritas con las existentes en una base de datos o diccionario de un idioma en particular o varios de estos y reemplazando la palabra procesada por la de mayor similitud, en caso de ser necesario.

3.2.4. Módulo de síntesis de texto a voz

El proceso de síntesis de texto a voz, se simplifica utilizando una de las características provistas por el API nativo de Android; estas funcionalidades se obtienen de la clase denominada "Text to Speech", la cual permite recibir el texto y transformarlo en una señal de audio que es reproducida por el dispositivo.

El uso de esta característica busca facilitar el desarrollo de este módulo, y gracias a la flexibilidad que presenta al permitir el manejo de pronunciación en varios idiomas en su configuración regional. En los smartphones Android podemos encontrar la síntesis de voz en características nativas como por ejemplo los comandos de voz y las opciones de accesibilidad presentes dependiendo de la versión del sistema operativo y la gama del dispositivo.

3.2.5. Módulo de localización geográfica

Para realizar el proceso de localización geográfica se emplea el GPS del dispositivo y se obtiene la ubicación vía internet utilizando la herramienta Google maps.

En caso de que el GPS se encuentre deshabilitado, este módulo lanzará un alerta al usuario indicando que es imposible obtener la ubicación.

Esta característica se desarrolló utilizando el API nativo de Android y no ameritó el uso de ninguna librería externa adicional.

3.3. Alcance del diseño

La aplicación tiene como propósito funcionar como una herramienta de soporte para personas con discapacidades visuales parciales, brindando una ayuda en las actividades diarias de estas personas mediante el uso de dispositivos inteligentes.

Su funcionamiento está delimitado por una serie de factores tanto del entorno como del uso del dispositivo, estos factores pueden ser: la cantidad de luz, posición de la cámara y nitidez de la imagen.

3.4. Diseño de pruebas

Las pruebas forman parte importante del desarrollo de cualquier aplicación, aseguran que el producto de software cumple con sus objetivos predefinidos.

Permite la estimación de efectividad, para el usuario, la efectividad significa qué tan confiable es la aplicación y sus resultados que tan precisos pueden ser.

Para la evaluación de la aplicación evaluaremos su efectividad, bajo los siguientes parámetros:

- **Precisión de lectura (PL):** Este valor en un rango entre cero y uno representa el porcentaje de palabras leídas correctamente en la imagen capturada (Ecuación 4.1).

$$PL = \frac{\# \text{ palabras leídas correctamente}}{\# \text{ palabras en la imagen}}$$

- **Precisión de corrección (PC):** representado por el porcentaje de palabras corregidas correctamente de las palabras leídas equivocadas (Ecuación 4.2).

$$PL = \frac{\# \text{ palabras corregidas correctamente}}{\# \text{ palabras leídas fallidas}}$$

- **Velocidad de lectura (VL):** Este valor será medido en la cantidad de palabras por segundo que lee la aplicación, midiendo el tiempo desde que se captura la imagen hasta que es leída (Ecuación 4.3).

$$VL = \frac{\# \text{ palabras leídas}}{\text{tiempo (seg)}}$$

Para determinar un valor total de precisión para la aplicación se establecerá un porcentaje de precisión de las palabras que fueron correctamente emitidas por la aplicación.

En la medición de estas pruebas utilizaremos un conjunto de muestras de diferentes textos en condiciones controladas de tamaño, color, iluminación y fuentes de texto; con el fin de comparar el texto leído y el texto capturado por la cámara del dispositivo móvil.

CAPÍTULO 4

4. DESARROLLO Y ANÁLISIS DE RESULTADOS

En este capítulo se realiza la descripción del proceso de implementación así como de las herramientas empleadas y las posteriores pruebas funcionales y de usabilidad en base a las cuales se obtendrán las conclusiones generales del proyecto.

4.1. Implementación

Para la implementación de la aplicación, podemos identificar las diferentes herramientas (lenguaje de programación, APIs, librerías adicionales, etc.) usadas según el módulo correspondiente como detallaremos más adelante.

A continuación listamos las herramientas:

- Java
- Android API

- Camera
- Predictive Text
- Text to Speech
- Location Manager
- Leptonica
- Tesseract OCR

Para iniciar el proceso de implementación, el primer paso consistió en la construcción de la interfaz propuesta en el diseño, utilizando el IDE Eclipse, debido a su sencillez y facilidad presentada para el manejo de java y el API de Android; una vez construida la pantalla central (y única), pasamos al desarrollo de los diferentes módulos de la aplicación.

Comenzando por el módulo de captura de imagen, se realizaron pruebas de captura a través de la cámara de un dispositivo Samsung GT-I8190 mismo que empleamos en la mayoría de las pruebas de desarrollo iniciales; al principio existieron inconvenientes en el manejo de la orientación de la imagen cuando esta se realizaba en una posición no-vertical, pero fueron resuelto mediante una conversión de la imagen inmediatamente después de que esta es obtenida.

Una vez la imagen es capturada, se procedió a desarrollo del módulo de procesamiento, aquí utilizamos las funciones de

leptonica para ejecutar el proceso de binarización y eliminación de ruido como se muestra en la Figura 4.1; el uso de las funcionalidades presentes en la mencionada librería permitió concluir el desarrollo de este módulo.

La imagen capturada en formato JPG por la cámara es transformada en un mapa de bits en escala de grises con una profundidad de 8 bits. Mediante la experimentación de la variación de dicha profundidad, definimos que con el valor señalado se mantiene una calidad aceptable en el mapeo a escala de gris aun cuando el contraste entre el contenido y el fondo no es demasiado elevado.

La Binarización se realiza mediante un método de umbralización global Otsu estándar, el cual permite hallar el umbral de binarización de manera adecuada, siendo un método adaptativo según los píxeles de la imagen como conjunto ya que emplea métodos estadísticos basándose en los valores de los píxeles de cada segmento definido. Cabe señalar, que al emplear este método, inicialmente se incluyó una variante del mismo utilizando una constante de “fracción de score máximo”, misma que consiste en definir un valor utilizado para determinar el rango sobre el cual se busca el mínimo en el histograma de la imagen; se realizaron pruebas experimentales

de este parámetro con valores de 0.1 y 0.2, obteniendo buenos resultados en cuanto a la nitidez y reducción de ruido sin pérdida de información utilizando la primera medida, por lo cual mantuvimos dicho parámetro con un valor fijo de 0.1. Adicionalmente, para la segmentación de la imagen se emplea la parametrización estándar de Leptonica con un tamaño de 32 x 32 píxeles por segmento.

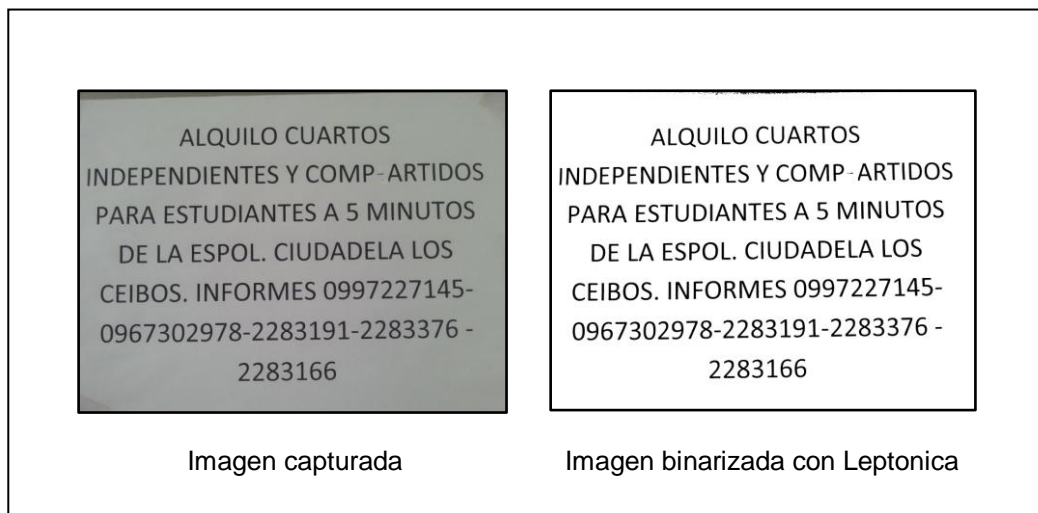


Figura 4.1 Binarización de imagen con Leptonica

Tesseract hace uso de la imagen procesada previamente para ejecutar sobre ella el algoritmo OCR; luego, para el desarrollo de dicho módulo, se presentaron problemas iniciales de configuración regional de la librería puesto que esta hace uso de archivos especiales dependiendo del idioma del texto a

reconocer, siendo el inglés el lenguaje por defecto, debido a esto, se debió incluir estos archivos adicionales en el proyecto e incluirlos al momento de la instalación.

Luego de construir un módulo de reconocimiento de texto escrito en español, definimos el alcance del mismo, estableciendo mediante ensayos y correcciones, las condiciones ideales en que el módulo permitía obtener exactamente el texto presente en la imagen.

El siguiente paso fue el desarrollo del proceso de corrección semántica, este representó cierto nivel de dificultad ya que consistía en un proceso inicialmente desconocido, dado que la idea del uso de Predictive Text surgió más adelante, mientras tanto, en el principio del desarrollo, se pensó en realizar consultas directas al diccionario del sistema operativo y realizar las correcciones mediante funciones creadas por los desarrolladores; esta idea se descartó posteriormente puesto que se observó una baja eficiencia del algoritmo y se adoptó el uso de Predictive Text, apreciando una notable mejoría el desempeño del módulo.

La sección correspondiente a la síntesis de texto a voz fue la menos compleja dentro de la implementación, la herramienta Text to Speech permitía a través del uso de una simple función,

recibir el texto como cadena de caracteres, retornar la señal auditiva y reproducirla a través del dispositivo; esto representó la finalización de la implementación del flujo funcional de la aplicación, consistente en capturar una imagen, leer su texto y devolverlo en forma de voz. El siguiente paso fue realizar la integración de los módulos presentes hasta el momento, acoplarlos con la interfaz diseñada y realizar las pruebas correspondientes.

La implementación concluyó con la inclusión del módulo de geolocalización como un complemento extra, para lo cual empleamos la clase de Android Location Manager. Empleando las funciones proporcionadas por dicha herramienta, conseguimos obtener con precisión los datos de ubicación del GPS del dispositivo y mediante un proceso realizado gracias a funciones de la misma clase, transformar las coordenadas en datos de localización relativos a un lugar existente en el mapa; esto se devuelve al usuario utilizando el mismo módulo de síntesis a voz antes mencionado.

De esta forma, una vez acoplados los componentes descritos, se realizaron diferentes pruebas preliminares, se realizaron correcciones de errores menores en el funcionamiento de la aplicación y manejo de errores de la aplicación como por

ejemplo la localización que devolvía la aplicación cuando el GPS del dispositivo se encontraba apagado o la función de repetir el último texto leído; y con esto concluyó el proceso de implementación.

4.2. Pruebas

Para la aplicación desarrollada se realizaron pruebas de funcionalidad y usabilidad, con el fin de determinar la validez de la misma y según el diseño descrito en el Diseño de pruebas.

Un aspecto importante que fue mencionado anteriormente son las condiciones en las que se captura la imagen y los problemas que se podrían encontrar, entre estos destacan: enfoque (ver Anexo A), contraste (ver Anexo B) y orientación (ver Anexo C); a continuación en la Tabla I se muestra un conjunto de pruebas para establecer cuál de estos problemas es el más influyente en la efectividad de la aplicación.

# Muestra	Palabras visibles	Palabras leídas	Precisión
Enfoque			0,52
1	6	3	0,50
2	30	23	0,77
3	5	2	0,40

4	15	8	0,53
5	15	6	0,40
Contraste			0,33
1	23	5	0,22
2	6	3	0,50
3	2	0	0,00
4	20	9	0,45
5	12	6	0,50
Orientación			0,85
1	2	2	1,00
2	4	3	0,75
3	23	20	0,87
4	7	6	0,86
5	12	9	0,75

Tabla I Resultados de pruebas con problemas de captura

En la Tabla I podemos observar, que de los problemas que influyen menos en la efectividad de la aplicación esta la orientación y que el caso contrario el problema más influyente es el mal contraste en la imagen capturada, bajando el porcentaje de precisión promedio a un 33%; cabe mencionar que no se estableció un nivel para el problema en la imagen, es decir se considera que en todas las imágenes capturadas bajo un mismo problema la afección es la misma.

4.2.1. Pruebas funcionales

Para determinar la efectividad de la aplicación se establecieron unas medidas de precisión en el capítulo

anterior, con las cuales calculábamos el porcentaje de precisión de palabras leídas (PL) y palabras corregidas (PC).

Bajo el uso de estas métricas de control, se realizaron pruebas en las que se registraban los siguientes datos:

- Palabras visibles en la imagen
- Palabras leídas correctamente por la aplicación
- Palabras leídas erróneamente
- Palabras corregidas correctamente
- Tiempo de respuesta (en segundos)

Luego de realizar el conjunto de prueba bajo diferentes condiciones de luz, diferencia de fuentes de texto y contraste de texto en imagen, se obtuvo la Tabla II.

# Muestra	Palabras visibles	Palabras leídas	Palabras con error	Palabras corregidas	Tiempo respuesta (seg)
1	4	4	0	0	2.80
2	5	4	1	1	3.01
3	3	3	0	0	1.99
4	5	3	2	1	3.05
5	7	6	1	0	2.57
6	9	9	0	0	3.21

7	8	6	2	2	2.49
8	8	7	1	0	2.63
9	9	8	1	1	2.89
10	6	6	0	0	2.01

Tabla II Resultados de pruebas con parámetros medidos

# Muestra	Precisión de Lectura	de	Precisión de Corrección	Precisión final
1	1.0		1.0	1.0
2	0.8		1.0	1.0
3	1.0		1.0	1.0
4	0.6		0.5	0.80
5	0.86		0.0	0.86
6	1.0		0.0	1.0
7	0.75		1.0	1.0
8	0.88		0.0	0.88
9	0.89		1.0	1.00
10	1.0		1.0	1.00

Tabla III Resultados de pruebas con métricas de precisión.

La Tabla III muestra una precisión media por muestra, que nos permite tener una idea de cuan efectiva fue la aplicación en su desempeño en cada prueba, los cálculos

de precisión para cada muestra fueron descritos en el capítulo anterior en el Diseño de pruebas.

En promedio la precisión final de la aplicación en el conjunto de muestras fue de 95% lo que nos permite decir que la aplicación es muy precisa en su funcionalidad, cabe recalcar que esta precisión también está sujeta a las condiciones antes mencionadas que pueden afectar la captura de la imagen.

4.2.2. Pruebas de usabilidad

En las pruebas de usabilidad se realizaron a 5 personas de entre 17 y 30 años de edad con discapacidad visual entre un 65% y 82% que debían mediante una serie de preguntas, calificar cuán fácil fue utilizar la aplicación para “leer” textos cortos puestos en paredes y puertas.

Para obtener estos resultados se realizaron las siguientes preguntas, las cuales debían ser valoradas entre 1 y 10, siendo 1 muy difícil y 10 muy fácil:

- **P1:** ¿Cuál fue el nivel de dificultad para acceder a la aplicación?
- **P2:** ¿Son identificables las opciones que muestra la aplicación?

- **P3:** ¿Cuál fue el nivel de dificultad para capturar la imagen?
- **P4:** ¿Es la retroalimentación auditiva entendible?

Luego de realizar estas preguntas a los usuarios los resultados de los usuarios fueron los que se describen en la Tabla IV

Usuario	P1	P2	P3	P4
1	8	10	7	10
2	8	10	6	9
3	10	7	8	10
4	7	7	6	10
5	9	8	8	9

Tabla IV Resultados de pruebas de usabilidad.

La Tabla IV muestra los valores calificados por los usuarios de la aplicación, en el que promediando los resultados obtenidos obtenemos un valor de 8,35 de calificación de usabilidad en un rango de 1 a 10.

Este valor nos permite decir que la aplicación fue fácil de usar para personas con discapacidades visuales parciales.

4.3. Análisis de resultados

Los resultados obtenidos nos muestran que en el mejor de los casos obtuvimos un 100% de precisión total y un 80% en el peor de los casos y en promedio una precisión total de 95%.

La precisión de corrección está estrechamente ligada a la precisión de lectura debido a que una lectura exacta nos provee automáticamente una precisión de corrección del 100%.

La tendencia nos muestra según los datos en la Tabla I que mientras más palabras se muestran en la imagen la cantidad de palabras con errores leídas se incrementan, y por tanto la precisión de corrección disminuye por la cantidad de palabras que se corrigen erróneamente.

En promedio la precisión de lectura de toda la muestras es de 88% y la precisión de corrección es de un 75%, lo que implica que nuestra aplicación es más precisa al momento de leer la imagen, que al momento de corregir las palabras mal leídas.

El tiempo de respuesta promedio obtenido en el muestreo fue de 0,46 segundos por palabra, siendo el valor más alto 0,70 segundos y el más bajo 0,31 segundos.

CONCLUSIONES

1. Los resultados obtenidos en las pruebas realizadas indican que la precisión de la aplicación en el reconocimiento de caracteres se incrementa proporcionalmente según las condiciones ambientales, de iluminación y estructurales de la imagen capturada se acercan a las condiciones establecidas en el alcance del presente proyecto, esto es, el nivel de eficacia de la aplicación es el máximo cuando se trata de una imagen clara, con una tipo de fuente estándar y con alto contraste entre el texto y el fondo.
2. La síntesis a voz de los dispositivos en los que se probó la aplicación presenta una precisión del 100% en la lectura, es decir, el texto reconocido por el OCR fue exactamente el texto que se devolvió como respuesta al usuario.
3. El volumen del texto procesado es el principal factor que afecta el tiempo de respuesta, teniendo así que en un dispositivo, se obtiene una respuesta casi inmediata en textos cortos (de 1 a 7 palabras

aproximadamente) y aumentando progresivamente con cada palabra adicional, ya que se incrementa el procesamiento realizado por el motor OCR.

4. El bajo contraste en una imagen que se desea procesar para la obtención del texto que contiene, es más influyente que otros problemas como el desenfoque de la imagen o la inclinación del texto en la misma.

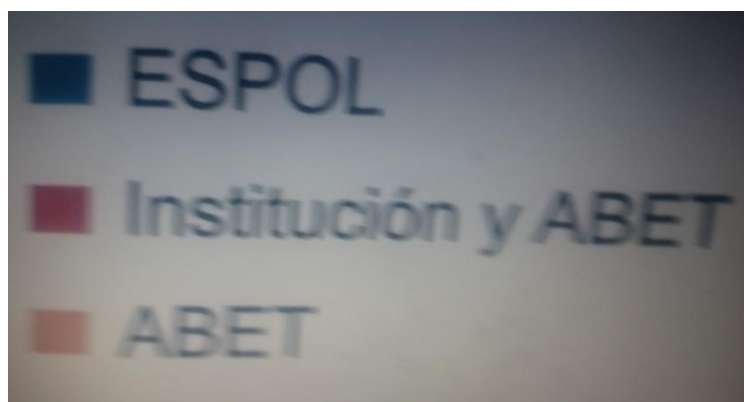
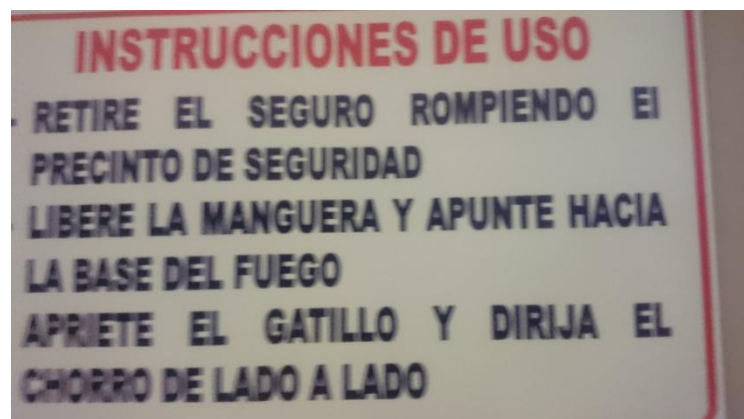
RECOMENDACIONES

1. La experiencia de usuario y accesibilidad podrían ser mejorados por medio del uso de comandos de voz y guías auditivas en las funcionalidades de la aplicación.
2. La utilización de otros recursos de hardware, de los actuales dispositivos móviles como el giroscopio y acelerómetro, podrían corregir problemas de orientación y alineación del texto en la imagen capturada, dando mejores resultados en la lectura de la aplicación.
3. El uso de texto predictivo, podría ser reemplazado por un algoritmo más eficiente, que no solo busque coincidencias sino que vaya aprendiendo con su uso, basándose en inteligencia artificial o alguna otra metodología.
4. El desarrollo de una automatización de la captura de imagen, con detección de texto; permitiría una mejora significativa en el proceso que más complicaciones tuvo en la usabilidad de la aplicación.

ANEXOS

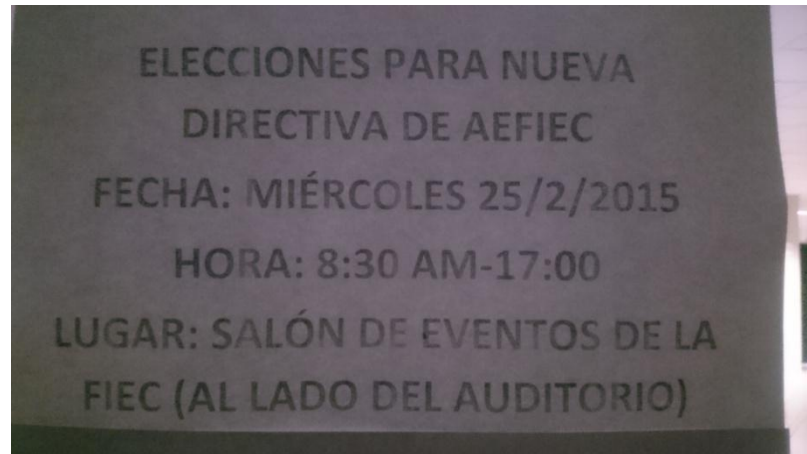
ANEXO A

MUESTRA DE IMÁGENES CON PROBLEMA DE ENFOQUE



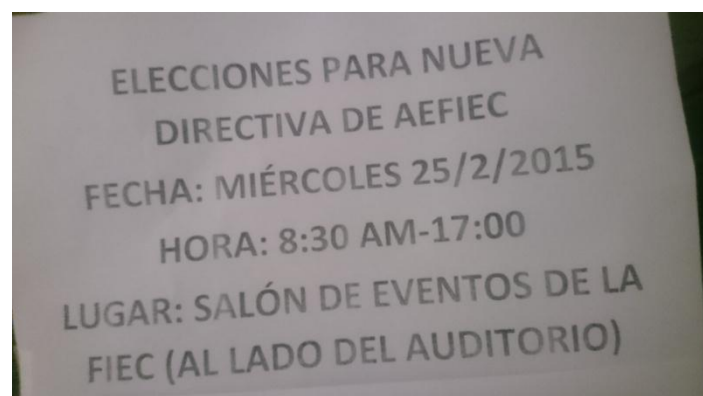
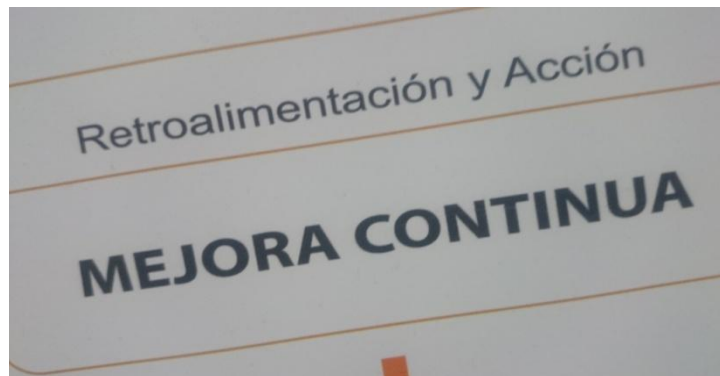
ANEXO B

MUESTRA DE IMÁGENES CON PROBLEMA DE CONTRASTE



ANEXO C

MUESTRA DE IMÁGENES CON PROBLEMA DE ORIENTACIÓN



BIBLIOGRAFÍA

- [1] INEC Instituto Nacional de Estadísticas y Censo, Censo de población y vivienda, 2010.
- [2] Synapse Adaptive, DOLPHIN HAL Lector de Pantalla Web Oficial, http://www.synapseadaptive.com/dolphin/hal_spanish_win.htm, consultado el 10 julio 2014.
- [3] Freedom Scientific, Jaws Web Oficial, <http://www.freedomscientific.com/Products/Blindness/Jaws>, consultado el 10 julio 2014.
- [4] Freedom Scientific, OpenBook Web Oficial, <http://www.freedomscientific.com/Products/LowVision/OpenBook>, consultado el 10 julio 2014.
- [5] GW Micro, Window-Eyes Web Oficial, <http://www.gwmicro.com/Window-Eyes>, consultado el 10 julio 2014.
- [6] Sánchez, C., Sardonís, V., Reconocimiento Óptico de Caracteres (OCR), Universidad Carlos III – Madrid, <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf> consultado el 20 julio 2014
- [7] González, J. R. H., Villaverde, J. P. R., & de Mesa, J. A. G. (1996). Sistema de reconocimiento óptico de caracteres (OCR) con redes neuronales. Departamento de Ciencias de la Computación, Universidad de Alcalá de Henares.
- [8] Plüss, B., Pompino, L., Tratamiento Automático de Reglas Ortográficas

para la Detección y Corrección de Errores,

<http://www.open.ac.uk/blogs/brianpluss/wp-content/uploads/2011/07/corrector-EST-paper.pdf>

[9] Armenta, A., Escalada, G., Garrido, J. M., & Rodríguez, M. Á. (2003). Desarrollo de un corrector ortográfico para aplicaciones de conversión texto-voz. *Procesamiento del lenguaje natural*, 31, 65-72.

[10] Agüero, P. D. (2012). Síntesis de voz aplicada a la traducción voz a voz (Doctoral dissertation, Tesis Doctoral. Universidad Politécnica de Cataluña. <http://hdl.handle.net/10803/97035>).

[11] Gutierrez, R., Frydson, M., Vintimilla B., Aplicación de Visión por computador para el Reconocimiento Automático de Placas Vehiculares utilizando OCR's Convencionales, https://www.dspace.espol.edu.ec/bitstream/123456789/19074/1/Paper_Gutierrez_Frydson_Vintimilla.pdf

[12] Singhal, M., & Shukla, A. (2012). Implementation of Location based Services in Android using GPS and Web Services. *IJCSI International Journal of Computer Science Issues*, 9(1), 237-242.

[13] IAMAI-Indicus, Location Based Services on Mobile in India, http://www.iamai.in/Upload/policy/LBS_Draft_Indicus.pdf