



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Diseño e Implementación de un Sistema Móvil Tributario.

TRABAJO DE TITULACIÓN

AUTORES: Castillo Guzmán, David Fernando

Sánchez Portocarrero, Pedro Iván

DIRECTORA: Ludeña González, Patricia Jeanneth, Mgtr.

LOJA – ECUADOR

2018

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magister.

Patricia Jeanneth Ludeña González.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: “**Diseño e Implementación de un Sistema Móvil Tributario**”, realizado por **Castillo Guzmán David Fernando - Sánchez Portocarrero Pedro Iván**, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, agosto de 2018

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Nosotros, **Castillo Guzmán David Fernando** y **Sánchez Portocarrero Pedro Iván**, declaramos ser autores del presente trabajo de titulación: “Diseño e Implementación de un Sistema Móvil Tributario”, de la Titulación de Electrónica y Telecomunicaciones, siendo **Patricia Jeanneth Ludeña González** directora del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Adicionalmente, declaramos conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f.

Castillo Guzmán David Fernando

1104277049

f.

Sánchez Portocarrero Pedro Iván

1105904567

DEDICATORIA

A Dios por haberme permitido culminar esta etapa de mi vida, que sin su presencia no podría haber logrado llegar hasta aquí.

A mis Padres Fernando Castillo y Patricia Guzmán, que, sin su sacrificio, sus consejos y sobre todo su amor y apoyo incondicional no habría podido culminar mis estudios, ellos han sido y serán siempre mi modelo a seguir.

A mis hermanos Xavier, Daniel y Gabriela por siempre estar ahí cuando los necesito y por siempre preocuparse de mi a su manera.

A Romina Romero, por ser una de las personas más importantes, por ser la persona que me dio el regalo más hermoso del mundo, nuestra preciosa hija.

A mi hija Arianna Castillo, que su sola presencia me hace querer mejorar en todos los aspectos de mi vida para poder brindarle un futuro mejor.

A mi compañero de trabajo Pedro Sánchez, por la confianza, apoyo, esfuerzo, dedicación y sobre todo paciencia a la hora de trabajar en este proyecto.

A toda mi familia y amigos en general que sin su apoyo y consejos no podría haber superado algunos de los obstáculos presentes a lo largo de mi vida universitaria.

David Castillo

DEDICATORIA

Este trabajo lo dedico con amor y cariño al regalo más grande Diosito me dio; mi amada familia, ellos son mi inspiración, motivación y apoyo en cada decisión y proyecto que realizo.

A mis padres Ivan Sánchez y Mirian Portocarrero, siempre han estado a mi lado dando lo mejor de sí para forjarme y convertirme en la persona que soy en la actualidad, todos mis logros personales y profesionales van dedicados a ellos, sin su apoyo nada de eso hubiera sido posible, gracias mamita y papito.

A mi abuelito Benigno Armijos, que, desde pequeño fue mi ejemplo a seguir y a pesar de no estar unidos por la sangre, tenemos un vínculo mucho más fuerte que se basa en el amor, confianza y cariño.

A mis hermanos Marbely, Deyberth y Dariana, son las personas que llenan de orgullo mi vida, los quiero ñañitos.

A mis angelitos Renato Paredes y mi Tío Clotario Ordoñez, sé que ellos me cuidan y me bendicen desde el cielo, estarían muy orgullosos de mi logro.

A mi novia que es la persona que le da un plus de felicidad a mi vida, es mi complemento, mi inspiración, es quien me motiva a ser mejor a dar lo mejor de mi cada día, es quien me hace sentir vivo.

A todos mis buenos amigos, con ellos he compartido una infinidad de anécdotas, gracias por su apoyo y por estar tanto en mis alegrías y tristezas.

Pedro Sánchez

AGRADECIMIENTO

Principalmente a Dios, por ser nuestra guía y fortaleza, su infinito amor ha permitido que a pesar de todas las dificultades que se han presentado no desmayemos y nos encontremos en el lugar en el que estamos, gracias Dios por bendecirnos cada día y darnos la oportunidad de ver lo hermosa y valiosa que es la vida, gracias Dios por permitirnos tener y disfrutar de una linda familia que siempre nos ha apoyado y ha sido la principal motivación en cada meta planteada.

A nuestros padres, por todo el amor y apoyo en cada decisión tomada, por el esfuerzo y sacrificio que hacen todos los días para que podamos alcanzar el éxito.

A nuestros amigos con los cuales hemos compartido momentos y experiencias inolvidables a lo largo de toda nuestra vida universitaria.

A mi amigo y compañero con el cual realice este proyecto, sin su apoyo, paciencia y dedicación no hubiera podido culminar exitosamente este trabajo.

A nuestra directora de tesis, Mgtr. Patricia Ludeña que, con su paciencia, ayuda y experiencia nos ha guiado para poder culminar con éxitos este trabajo.

¡Nuevamente gracias a Dios, su voluntad he hecho posible todo esto!

¡GRACIAS TOTALES 🙏!

David y Pedro

INDICE DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN.....	II
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	III
DEDICATORIA	IV
AGRADECIMIENTO	VI
INDICE DE CONTENIDOS.....	VII
INDICE DE FIGURAS.....	XI
INDICE DE TABLAS.....	XIII
RESUMEN.....	1
ABSTRACT	2
INTRODUCCIÓN.....	3
CAPÍTULO I.....	5
ESTADO DEL ARTE.....	5
1.1 PLANTEAMIENTO DEL PROBLEMA	6
1.2 OBJETIVOS	6
1.2.1 Objetivo General	6
1.2.2 Objetivos Específicos.....	6
1.3 JUSTIFICACIÓN E IMPORTANCIA DEL PROYECTO.....	7
1.4 ALCANCE DEL PROYECTO.....	7
1.5 TRABAJOS EXISTENTES Y HERRAMIENTAS SIMILARES	8
1.6 MARCO TEÓRICO.....	12
1.6.1 SRI y Tributación	12
1.6.2 Dispositivos Móviles.....	15
1.6.2.1 Sistemas Operativos Móviles	15
1.6.2.1.1IOS.....	15
1.6.2.1.2Android.....	16
1.6.2.1.3Arquitectura Android.....	16
1.6.2.1.4Aplicaciones móviles	18

1.6.2.1.4.1	Ciclo de vida de una aplicación	18
1.6.2.1.4.2	Ciclo de vida de una actividad	18
1.7	PROCESAMIENTO DIGITAL DE IMÁGENES.....	20
1.7.1	Adquisición	20
1.7.1.1	Imagen digital.....	21
1.7.2	Pre-procesado	21
1.7.3	Segmentación.....	22
1.7.4	Representación y descripción	22
1.7.5	Reconocimiento e interpretación.....	22
1.8	OPERACIONES GENERALES EN IMÁGENES DIGITALES	22
1.8.1	Umbralización	23
1.8.2	Binarización	23
1.8.3	Transformaciones morfológicas	24
1.8.3.1	Elemento Estructural	24
1.8.3.2	Operaciones Morfológicas.....	24
1.8.4	Filtrado espacial.....	25
1.9	RECONOCIMIENTO ÓPTICO DE CARACTERES “OCR”	26
1.9.1	Motores OCR.....	29
1.9.1.1	Tesseract OCR.....	29
1.9.1.2	Google Mobile Vision.....	30
1.10	BASE DE DATOS	31
1.11	CÓDIGOS QR (QUICK RESPONSE).....	33
CAPÍTULO II.....		34
DISEÑO DE SOLUCIÓN Y SELECCIÓN DE HERRAMIENTAS.....		34
2.1	Metodología	35
2.2	Diseño de Solución	35
2.2.1	Tipos de comprobantes	36
2.3	Estructura de la aplicación móvil	36

2.3.1	Módulo de autenticación y registro de usuarios	37
2.3.2	Módulos de adquisición de Datos	38
2.3.2.1	Ingreso manual.....	38
2.3.2.2	Ingreso mediante código QR.....	38
2.3.2.3	Ingreso mediante OCR.....	39
2.4	Almacenamiento de la información.....	40
2.5	Módulo de visualización y generación de resumen de gastos	40
2.6	Selección de herramientas.....	40
2.6.1	Selección de entorno de desarrollo.....	41
2.6.2	Selección de motor OCR	42
2.7	Análisis de Requisitos	47
2.7.1	Requisitos no funcionales	47
2.7.2	Requisitos funcionales	47
2.7.3	Criterios de aceptación	48
2.7.4	Plantilla combinada.....	49
2.7.5	Identificación de actores.	50
2.7.6	Modelos de caso de Uso	50
2.7.6.1	Especificaciones de casos de Uso	51
2.7.7	Diagrama de navegación	54
2.7.8	Descripción del mapa de navegación	55
CAPÍTULO III.....		57
IMPLEMENTACIÓN.....		57
3.1	Módulo de autenticación y registro de usuarios.....	58
3.2	Módulo de adquisición de datos	59
3.3	Método de adquisición de datos manual	60
3.4	Método de adquisición de datos por QR	61
3.5	Método de adquisición de datos por OCR.....	62
3.6	Pre-Procesado	65

3.7	Almacenamiento de la información.....	67
3.8	Módulo de visualización y generación de resumen de gastos deducibles	69
CAPÍTULO IV		71
PRUEBAS Y VALIDACIÓN.....		71
4.1	Pruebas de Usabilidad	72
4.2	Resultados obtenidos del test de usabilidad.....	72
4.3	Validación de módulos.	74
4.3.1	Adquisición de datos manual	75
4.4	Adquisición de datos mediante escaneo de código QR.....	76
4.5	Adquisición de datos mediante escaneo de código QR.....	78
4.6	Manual de usuario.....	79
CONCLUSIONES		80
RECOMENDACIONES.....		81
BIBLIOGRAFÍA.....		82
ANEXOS.....		87
ANEXO A.....		88
ANEXO B.....		93
ANEXO C		95
ANEXO D		96

INDICE DE FIGURAS

CAPÍTULO I

Figura 1. 1 Arquitectura Android	16
Figura 1. 2 Ciclo de vida de un aplicativo Android.....	19
Figura 1. 3 Esquema de reconocimiento básico.....	20
Figura 1. 4 Histograma ideal	23
Figura 1. 5 Máscara centrada en (x, y) con sus puntos respectivos	25
Figura 1. 6 Coeficientes de la máscara	26
Figura 1. 7 Esquema básico de reconocimiento óptico de caracteres	28
Figura 1. 8 Segmentación y normalización, desde arriba hacia abajo: imagen original, eliminación de ruido y detección, normalización y tamaño de caracteres.....	28
Figura 1. 9 Arquitectura Tesseract	30
Figura 1. 10 Tipos de segmentación de texto en Google Mobile Vision	31
Figura 1. 11 Servicios que ofrece Firebase	32
Figura 1. 12 Código QR	33

CAPÍTULO II

Figura 2. 1 Facturas físicas: a) Manual, b) Impresa	36
Figura 2. 2 Estructura del aplicativo móvil	37
Figura 2. 3 Factura impresa con código QR.....	39
Figura 2. 4 Procesado de imágenes para la selección de motor OCR a) Tesseract OCR , b) Google Mobile Vision	43
Figura 2. 5 Condiciones analizadas para la selección de motor OCR (Facturas); 1, Angulo clara cerca; 2, Angulo obscura cerca; 3, Obscura cerca; 4, Clara lejos.....	45
Figura 2. 6 Modelos de caso de uso	51
Figura 2. 7 Mapa de navegación.....	55

CAPÍTULO III

Figura 3. 1 Diagrama de funcionamiento módulo de autenticación y registro de usuarios... 58	58
Figura 3. 2 Diagrama de funcionamiento módulo de autenticación y registro de usuarios... 59	59

Figura 3. 3	Interface Login, Registro y recuperación de contraseña olvidada.....	59
Figura 3. 4	Obtención de instancia, Android Studio IDE	60
Figura 3. 5	Diagrama de funcionamiento método de adquisición de datos manual	60
Figura 3. 6	Formulario para ingreso de comprobantes de forma manual.....	61
Figura 3. 7	Esquema de funcionamiento módulo de adquisición de datos por QR	61
Figura 3. 8	Ejemplo decodificador QR y data Matrix.....	62
Figura 3. 9	Interfaz módulo de adquisición por código QR	62
Figura 3. 10	Esquema de funcionamiento módulo de adquisición de datos por QR	63
Figura 3. 11	Segmentación previa a pre-procesado	64
Figura 3. 12	Corrección de perspectiva etapa Pre-procesado.....	65
Figura 3. 13	Filtrado de la imagen etapa Pre-procesado.....	66
Figura 3. 14	Fragmento de código TextRecognizer.....	66
Figura 3. 15	Interfaz gráfica adquisición mediante OCR	67
Figura 3. 16	Estructura de información dentro de la base de datos.....	67
Figura 3. 17	Reglas Firebase Database.....	68
Figura 3. 18	Estructura de la base de datos en diagrama de árbol.....	68
Figura 3. 19	Esquema de funcionamiento módulo de visualización y generación de anexo	69
Figura 3. 20	Interfaz gráfica módulo de visualización y generación de anexo	70

CAPÍTULO IV

Figura 4. 1	Test de usabilidad de aplicación	73
Figura 4. 2	Comprobación de existencia de información en a base de datos	74
Figura 4. 3	Validación campo a campo adquisición manual de información	75
Figura 4. 4	Validación código QR.....	76
Figura 4. 5	Validación reconocimiento óptico de caracteres.....	77
Figura 4. 6	Validación Módulo de visualización y generación de anexo	78

INDICE DE TABLAS

Tabla 1. 1 Características de herramientas móviles similares	11
Tabla 1. 2 Tabla de rubros deducibles para declaración al IVA	13
Tabla 1. 3 Formato de gastos deducibles de impuestos aceptados por el SRI	14
Tabla 1. 4 Estados de una actividad.....	19
Tabla 1. 5 Ventajas y desventajas de uso Firebase.....	32

CAPÍTULO II

Tabla 2. 1 Comparativa Android Studio vs Eclipse.	41
Tabla 2. 2 Comparativa Google Vision - Tesseract OCR.....	42
Tabla 2. 3 Condiciones establecidas para selección de motor OCR.....	42
Tabla 2. 4 Porcentajes de caracteres errados Google Mobile Vision OCR	44
Tabla 2. 5 Porcentajes de caracteres errados Tesseract OCR	44
Tabla 2. 6 Resultados Obtenidos por motores OCR	45
Tabla 2. 7 Resultados por motores OCR	46
Tabla 2. 8 Características de equipos utilizados en las pruebas.....	46
Tabla 2. 9 Requisitos no funcionales	47
Tabla 2. 10 Requisitos funcionales	48
Tabla 2. 11 Criterios de aceptación	48
Tabla 2. 12 Plantilla combinada de la aplicación	49
Tabla 2. 13 Actores del sistema	50
Tabla 2. 14 Caso de uso de registro de nuevos usuarios.	51
Tabla 2. 15 Caso de uso login y cierre de sesión de usuarios.	52
Tabla 2. 16 Caso de uso ingreso manual de comprobantes.	52
Tabla 2. 17 Caso de uso ingreso por QR de comprobantes	53
Tabla 2. 18 Caso de uso ingreso por OCR de comprobantes.....	53
Tabla 2. 19 Caso de uso de Visualización de estado de cuenta y generación de anexo.	54

RESUMEN

El objetivo de la tecnología es facilitar al usuario realizar una tarea específica haciendo uso de distintas herramientas que día a día van evolucionando. En la actualidad, los dispositivos móviles son herramientas que no únicamente pueden realizar llamadas o enviar SMS, estos equipos se han convertido en parte de nuestra vida cotidiana ya que pueden llevar a cabo una gran cantidad de tareas como el procesamiento de datos y gestión de información mediante la integración de múltiples tecnologías. Respecto a lo mencionado este proyecto tiene como finalidad describir todos los fundamentos teóricos y prácticos para el desarrollo de un aplicativo móvil en Android que permita extraer datos tributarios de facturas haciendo uso de la cámara del móvil. La aplicación final integra tecnologías como QR (Quick Response), Bases de datos y digitalización de texto mediante el uso de algoritmos de OCR (Reconocimiento Óptico de Caracteres) los cuales fueron probados y validados en base a las necesidades del proyecto, por último, se genera un anexo en formato Microsoft Excel que puede ser adjuntado a la plataforma del SRI (Servicio de Rentas Internas).

PALABRAS CLAVE: Tributación, OCR, QR, Android, Dispositivo móvil.

ABSTRACT

The objective of the technology is to facilitate the work to perform a specific task making use of the tools that are evolving day by day. Currently, mobile devices are tools that can be sent by phone or send SMS, these computers have become part of our daily lives and can carry out a large number of tasks such as data processing and information management by multiple technologies. With regard to this topic has all the theoretical and practical foundations for the development of a mobile device based on Android that can be extracted extract the data from the applications of use of the mobile camera. The final application will integrate several technologies such as QR (Rapid Response), Databases and text digitization through the use of OCR algorithms (Optical Character Recognition) which were tested and validated based on the project's needs, finally, an annex is generated in format Microsoft Excel that can be attached to the platform of the SRI (Internal Revenue Service).

KEYWORDS: Taxation, OCR, QR, Android, Mobile device.

INTRODUCCIÓN

A medida que la sociedad avanza, el uso de distintos dispositivos tecnológicos ha facilitado la vida cotidiana de las personas. Desde la creación del primer dispositivo móvil en el año 1983 hasta la fecha, estos equipos, han revolucionado la industria de las telecomunicaciones, actualmente permite al usuario disponer de conexiones a internet de altas velocidades, cámaras de última generación y alta capacidad de procesamiento de datos en tiempo real (Aranaz Tudela, 2009), todos estos avances van de la mano del uso de sistemas operativos (SO) que gracias al uso de aplicaciones permiten efectuar distintas tareas facilitando o gestionando actividades a desarrollar.

Con el nacimiento de Android, la industria de los teléfonos móviles cambió notablemente ya que este Sistema Operativo fue desarrollado netamente para smartphones con limitaciones en hardware, lo que permitió el desarrollo de muchas aplicaciones.

En 1997 nace el SRI como respuesta a la elevada evasión tributaria. Actualmente la misión es asegurar la suficiencia recaudatoria destinada al fomento de la cohesión social. Para esto se ha creado múltiples impuestos que abarcan muchos campos.

El impuesto a la Renta es un valor que deben cancelar obligatoriamente y de forma anual, todas las personas que perciben un sueldo superior \$ 11.290 (monto establecido para el 2018). Para ello el SRI da la facilidad de presentar un informe de gastos personales para que a través de éste deducir el valor a pagar año a año, los requisitos para que este anexo tenga validez es tener los formularios donde se corrobore la información declarada.

En este documento habla acerca del desarrollo de un proyecto que busca facilitar el proceso actual de declaración de impuesto a la renta, así evitar el gasto innecesario de tiempo y dinero. Para esto se usó varias tecnologías dentro de un aplicativo móvil el cual se encarga de realizar todo el proceso, que va desde la adquisición y recolección de información hasta la generación de un archivo final, proceso el cual, siendo llevado de manera manual, es largo y en el caso de exceder las fechas límites para realizar la declaración de impuesto a la renta, el contribuyente estará sujeto a multas e intereses por mora (Servicio de Rentas Internas del Ecuador (SRI), n.d.).

La aplicación propuesta integra múltiples tecnologías, estas tecnologías son OCR (Optical Character Recognition), códigos QR (Quick Response), el uso de bases de datos (Firebase), funcionando todo en el sistema operativo Android, el cual es el sistema con mayor acogida en la actualidad.

El aplicativo consta de varios módulos de adquisición de datos (OCR, QR, MANUAL), toda la información es almacenada dentro de un base de datos en tiempo real (Firebase DataBase). La aplicación cuenta también con un apartado que permite visualizar los montos por rubros actuales, el número de facturas ingresadas con opción a generar un anexo en formato Microsoft Excel compatible con la plataforma web del SRI.

CAPÍTULO I

ESTADO DEL ARTE

1.1 Planteamiento del problema

En la actualidad, para que el proceso de declaración de impuesto a la renta se lo lleve de forma óptima, requiere cierto grado de conocimientos en el área de contabilidad. En personas con escasos conocimientos en el tema, este proceso puede tener cierto nivel de dificultad, ésta es una de las razones por la cual, la mayor parte del tiempo se recurre a contratar los servicios de terceras personas (Contador), estos se encargan de realizar toda la recolección de datos hasta la generación del anexo que se debe adjuntar anualmente a la plataforma del SRI.

El Servicio de Rentas Internas “SRI” sanciona a todos los contribuyentes que sobrepasan las fechas límite para realizar su declaración de impuesto a la renta, por ende, nace la necesidad de implementar un sistema rápido y preciso que se encargue de detectar y almacenar información referente a facturas sin requerir un excesivo tiempo para hacerlo, una de las soluciones más viables es el uso de un smartphone.

El objetivo de este proyecto es sustituir este trabajo manual y de cierta manera optimizarlo para que cualquier persona con escasos conocimientos y tiempo, pueda declarar haciendo uso de su smartphone.

Para llevar a cabo lo mencionado, se fusionará distintos tipos de tecnologías dentro de un aplicativo móvil desarrollado en ANDROID STUDIO, el aplicativo permitirá la adquisición y almacenamiento de información de facturas.

1.2 Objetivos

1.2.1 Objetivo General

- Implementar múltiples tecnologías para el desarrollo de una aplicación móvil en sistema Android enfocada a gestión tributaria.

1.2.2 Objetivos Específicos

- Estudiar la línea base sobre tributación y herramientas móviles similares.
- Diseñar e implementar módulos de métodos de ingreso de datos tributarios de facturas (Generación, adquisición, lectura y digitalización de texto para obtención de datos a través de cámara móvil).
- Diseñar e implementar modelos de cálculo y generación de anexos de impuesto a la renta.
- Optimizar y validar el aplicativo.

1.3 Justificación e importancia del Proyecto

La finalidad de este proyecto es desarrollar un sistema capaz de adquirir información de facturas usando “Reconocimiento óptico de Caracteres” y a la vez complementarlo uniendo varios tipos de tecnologías dentro de un mismo aplicativo para hacerlo más completo y eficiente.

El sistema permite adjuntar los datos e información de las facturas haciendo uso de la cámara del móvil, con una simple captura de una foto o a su vez ingresar dicha información directamente mediante teclado, una vez almacenada toda la información se podrá visualizar los montos actuales mediante gráficos porcentuales.

Actualmente en el país no existe un sistema o aplicación capaz de realizar el proceso que va desde la adquisición de información tributaria de las facturas hasta la generación del resumen de gastos deducibles de impuestos, este sistema ha sido desarrollado como un apoyo para que todos los contribuyentes con corta disponibilidad de tiempo y conocimientos, puedan generar su anexo y adjuntarlo a la plataforma web del SRI, de esta manera se puede evitar gastos “innecesarios”, ahorrar tiempo y dinero evitando multas y sanciones.

1.4 Alcance del proyecto

El proyecto busca automatizar y agilizar al proceso con el cual se realiza la declaración de impuesto a la renta, evitando un gasto de tiempo excesivo al momento de hacerlo, para esto se ha desarrollado una aplicación para la plataforma Android, cuyos requerimientos mínimos del equipo son: Android 4.0.3+, 1+GB de Memoria RAM y Cámara de 5+ Megapíxeles. Dicha aplicación consta de la integración de varias tecnologías como: Bases de datos, Códigos QR, y Reconocimiento óptico de caracteres para digitalización de texto a través de imagen.

Cabe recalcar que la aplicación está dirigida a personas que sobrepasen la base imponible por el SRI y quieran disminuir el pago de este impuesto, por ello ésta permitirá:

- Registrar nuevos usuarios y posterior ingreso con sus credenciales (usuario y clave).
- Tener un menú amigable con el usuario para fácil uso de los distintos métodos de ingreso de datos.
- Almacenar información facturas; sea de forma manual, usando código QR (método de ingreso de datos propuesto del que se hablará adelante) o digitalizando texto a través una foto de una factura.
- Visualizar montos por rubros, y número de facturas existentes hasta la fecha.
- Generar resumen de gastos deducibles de impuesto en formato Microsoft Excel.

La información necesaria la ingresará el usuario haciendo uso de tres métodos de ingreso: manual, OCR, QR. Esta información será almacenada en una base de datos donde se gestionará todos los datos que el usuario brinde, finalmente se podrá generar el anexo en el formato aceptado por la plataforma del SRI.

1.5 Trabajos existentes y herramientas similares

Se realizó una búsqueda de artículos, tecnologías y aplicaciones relacionadas con el tema propuesto, con el objetivo de analizar todas las alternativas que existen hasta la fecha. A continuación, se presentan los trabajos con mayor similitud al objetivo de este proyecto.

El reconocimiento óptico de caracteres tiene un amplio número de aplicaciones uno de los usos más comunes es, usarlo para reconocimiento automático de placas vehiculares como se habla en (Do et al., 2016) y (Ramiah, Liong, & Jayabalan, 2015) ambos trabajos hablan sobre el desarrollo un programa en Android capaz de capturar una imagen desde el dispositivo móvil y haciendo uso de bibliotecas OpenCV, motor Tesseract y redes neuronales puede extraer la información de la placa del vehículo y posteriormente almacenar dicha información en una base de datos.

(Baharlou, Hemayat, Saberhari, & Yaghoobi, 2015) presenta un nuevo algoritmo de reconocimiento de matrículas altamente optimizado para reducir el margen de error, su método consta en encontrar la ubicación de la placa, teniendo en cuenta los criterios de "Sensibilidad al ángulo" para los caracteres, este proceso es independiente de la ubicación de la placa y su porcentaje de efectividad varía dependiendo de la cantidad de componentes en la imagen, hablan también sobre el uso de filtros para reducir el margen de ruido para finalmente obtener una precisión de 95.33%.

(Vuong & Do, 2014) describe el diseño e implementación de un lector de tarjetas con nombre multilinguaje en la plataforma Android, la imagen cargada desde el teléfono se rectificará haciendo uso de diferentes técnicas de procesamiento de imagen antes de extraer sus caracteres mediante el uso del motor Tesseract OCR, gracias a Google (API) haciendo uso de plantillas se pudo buscar coincidencias de patrones y extraer la información de nombre y número móvil.

Otro trabajo similar es (Kongtalin, Minsakorn, Yodchaloemkul, Boontarak, & Phongsuphap, 2014) el cual trata de la extracción de texto de documentos médicos y al igual que todos los trabajos referenciados en este documento, usan Tesseract OCR Engine para extraer texto contenido en la imagen, los creadores tomaron en cuenta distintos aspectos como: cámara

del dispositivo, separación entre letras, tipo de tipografía y el tamaño del bloque de texto, todos estos factores fluyen en el resultado según los autores.

Tanto (Vuong & Do, 2014), (Magalhaes, Ribeiro, Alves, & Guevara, 2017) y (Jiang, Gonnot, Yi, & Saniie, 2017) son proyectos basados en el motor Tesseract OCR, cuyo objetivo es utilizar una cámara montada en un dispositivo con un sistema Android para identificar correctamente texto, objetos como botiquines o portadas de libros respectivamente y luego proporcionar al usuario información relevante al respecto, cabe recalcar que varias de estas investigaciones están dirigidas a personas de edad avanzada o con discapacidad visual ya que tienen la capacidad de convertir todo el texto en voz, estos documentos analizan el flujo de diseño y los respectivos resultados experimentales de sus proyectos.

El trabajo de (Heliński, Kmiecik, & Parkoła, 2012) realiza un análisis comparativo de la precisión de dos motores OCR; TesseractOCR y FineReader10 Corporate Edition, ambos motores son soluciones de alto nivel y de código abierto. La comparación se basa en analizar una serie de documentos históricos. Estas dos herramientas se ajustaron para mejorar la tasa de precisión de reconocimiento de texto impreso, las herramientas utilizadas para el cálculo del mismo fueron proporcionadas por el proyecto IMPACT, éste es mencionado en dicho trabajo.

(Bhaskar, Lavassar, & Green, 2010) presenta un algoritmo de reconocimiento de caracteres aplicado a la extracción de información de tarjetas de crédito, las fotos en diferentes condiciones son cargadas desde un teléfono Android, previo al procesamiento de la imagen pasa por MATLAB para ser mejorada y luego ser procesada haciendo uso de TesseractOCR. Dicha implementación tiene un éxito en distintas condiciones en las que se capturo la imagen. El algoritmo de MATLAB en DROID demostró ser muy lento, por ende, fue simplificado para su posterior implementación en el teléfono.

En base a las búsquedas realizadas se encontró un sistema denominado "PhotoOCR" (Bissacco, Cummins, Netzer, & Neven, 2013) cuyo objetivo es la extracción de texto confiable de imágenes con una entrada similar al reconocimiento de voz, los autores aseguran que su sistema tiene un mejor rendimiento en esta área y es capaz de reconocer textos en una variedad de condiciones desafiantes de procesamiento de imágenes, ya que cuenta con aprendizaje automático, este ha mejorado sustancialmente la clasificación de caracteres aislados, este sistema también tiene un tiempo promedio de procesamiento de 600ms.

A continuación, se muestra un recuento de algunos sistemas funcionales implementados actualmente bajo la plataforma Android, estos comparten algunos objetivos similares a los de este trabajo, todas estas están disponibles en la Google Play Store.

Exaccta Tax

- Desarrollador: Exaccta Soluciones S.L.
- Puntaje: 3.3
- Descargas: 5 mil

Aplicación dirigida a autónomos y asesorías, únicamente se carga fotos de facturas o tickets y automáticamente se contabilizan solos en función del perfil tributario, todas las fotos de los comprobantes pueden ser utilizadas como justificantes tributarios ya que esta aplicación está homologada y certificada por la Agencia Estatal de Administración Tributaria Española.

Controlar Gastos

- Desarrollador: CASorin
- Puntaje: 4.6
- Descargas: 1 millón

Esta aplicación permite administrar gastos y llevar una contabilidad básica del día a día de manera simple y sencilla. Toda la información puedes protegerla haciendo uso de contraseña o pin de seguridad.

Renta 2017 Ecuador

- Desarrollador: Rommel Vicente Torres
- Puntaje: 5
- Descargas: +50

Permite gestionar facturas, asignar valores deducibles de impuesto del sistema de rentas internas SRI.

SRI Móvil

- Desarrollador: SRI Ecuador
- Puntaje: 4.3
- Descargas: 100 mil

Brinda a los usuarios acceso a noticias, información de agencias, consultas tributarias públicas y redes sociales de la institución desde un smartphone, su interfaz amigable al usuario. Los servicios que ofrece son: Estado Tributario, Valor de Matrícula, Deudas, Validez de Documentos Físicos, Impuesto a la Renta Causado, seguimiento de trámites, etc.

Expense Manager

- Desarrollador: Bishinews
- Puntaje: 4.3
- Descargas: 5 millones

Gestor de gastos personales, seguimiento de ingresos o egresos diario, semanal o anual. Respaldo de información en la nube o la memoria interna del dispositivo.

Smart Receipts

- Desarrollador: Smart Receipts LLC
- Puntaje: 4.4
- Descargas: 500 mil

Escanea tus recibos y genera un resumen de gastos, tiene opción a generar informes en PDF, CSV y ZIP en formato personalizable tanto para control de finanzas como de empleador. Toda la información esta segura y solo el usuario tiene acceso a la misma. Permite extraer información de recibos mediante OCR (reconocimiento óptico de caracteres) y finalmente permite visualizar la información con gráficos de gastos por categorías.

Receipts Bank

- Desarrollador: Receipts Bank
- Puntaje: 4.2
- Descargas: 100 mil

Permite el escaneo de recibos para gestión, automatización y seguimiento de gastos. Integra tecnología OCR para extracción de datos de facturas y recibos, información respaldada y asegurada mediante almacenamiento en la nube.

La tabla 1.1 presenta un resumen de todos los sistemas implementados, con sus principales características, funcionalidad y plataformas en las cual trabajan.

Tabla 1. 1 Características de herramientas móviles similares

APLICACIÓN	EXACCTA EXPENS	CONTROLADOR DE GASTOS	RENTA 2017 ECUADOR	SRI MOVIL	EXPENSE MANAGER	SMART RECEIPTS	RECEIPT BANK
							
S.O.	ANDROID	ANDROID – IOS	ANDROID	ANDROID- IOS	ANDROID	ANDROID - IOS	ANDROID

TIPO DE APLICACIÓN	CONTROLA DOR DE GASTOS + PERMITE DECLARAR EL IVA SOLO EN ESPAÑA	PERMITE UNA CONTABILIDAD BÁSICA, CONTRALA INGRESOS Y EGRESOS	PERMITE REALIZAR LA DECLARACIÓN DEL IVA	VISUALIZA EN TIEMPO REAL LA DECLARACIÓN DEL IVA	CONTROLADOR DE GASTOS, CONTABILIDAD BÁSICA	ESCÁNER DE RECIBOS, CONTROLADOR DE GASTOS Y GENERA INFORMES DE GASTOS	ESCÁNER DE RECIBOS
CONEXIÓN A INTERNET NECESARIA	Si	No	No	Si	No	Si	Si
ÚLTIMA ACTUALIZACIÓN	27/5/2018	11/5/2018	17/2/2017	1/2/2018	26/11/2017	3/6/2018	13/6/2018
VERSIÓN DISPONIBLE	2.15.404	1.2.9	1.00	1.8.5	3.4.7	4.12.0.446	3.1
MÉTODOS DE INGRESO	MANUAL IMAGEN	MANUAL	MANUAL	NINGUNO	MANUAL	MANUAL IMAGEN	MANUAL IMAGEN
COSTO	PAGA	GRATUITA	GRATUITA	GRATUITA	GRATUITA	GRATUITA Y PAGA	PAGA
USUARIO FINAL	EMPRESARIAL	PERSONAL	PERSONAL	PERSONAL EMPRESARIAL	PERSONAL	PERSONAL	PERSONAL
TIPO DE PROCESAMIENTO	EN EL SMARTPHONE	EN EL SMARTPHONE	EN EL SMARTPHONE	NINGUNO	EN EL SMARTPHONE	EN LA NUBE	EN LA NUBE

Fuente: Autores

Elaboración: Autores.

1.6 Marco Teórico

El presente apartado muestra la base teórica y términos frecuentemente utilizados referente al desarrollo del proyecto. Lógicamente al ser este un proyecto en el que se está usando imágenes digitales se explica todo lo que engloba respecto al tema. También se habla brevemente acerca de ¿por qué? y ¿para qué? el pago de impuesto a la renta y que personas u organizaciones tienen la obligación de hacerlo.

1.6.1 SRI y Tributación

El servicio de rentas internas o SRI es el organismo técnico y autónomo del estado ecuatoriano que desde su nacimiento en 1997 tiene el principal objetivo de luchar contra la evasión y las malas prácticas tributarias generadas por la inexistencia de una cultura tributaria en el país.

La función primordial del SRI es la recaudación de impuestos y según lo estipula la ley son las únicas contribuciones obligatorias, tanto para personas naturales como sociedades, en general los impuestos son el precio de vivir en una sociedad civilizada.

A partir del año 2008, el SRI reformo la ley de equidad tributaria para dar la posibilidad de que las personas naturales pueden deducir sus gastos personales para el pago del impuesto a la renta, los deducibles no consideran el impuesto al valor agregado (IVA) ni el impuesto al consumo (ICE). Para la aplicación de la deducción de gastos personales se incorporan como dependientes a los padres, a los hijos del cónyuge o pareja en unión de hecho y a los hijos mayores de edad que dependan económicamente del contribuyente. Es obligatorio para poder deducir los gastos los comprobantes de venta (facturas).

Es necesario que el contribuyente presente un documento donde anexe la proyección de gastos personales, donde se adjuntará la información de gastos en vivienda, educación, arte y cultura, vestimenta, alimentación y salud.

En la tabla 1.2 se muestran los rubros y los montos máximos que se pueden declarar en el anexo.

Tabla 1. 2 Tabla de rubros deducibles para declaración al IVA

RUBRO	CONTINENTE	GALAPAGOS
	LIMITE AÑO 2018	LIMITE AÑO 2018
VIVIENDA	\$3 662,75	\$6 604,22
EDUCACION, ARTE Y CULTURA	\$3 662,75	\$6 604,22
ALIMENTACION	\$3 662,75	\$6 604,22
VESTIMENTA	\$3 662,75	\$6 604,22
SALUD	\$14 651,00	\$26 416,88

Fuente: (Servicio de Rentas Internas del Ecuador, 2018)

Elaboración: Servicio de Rentas Internas del Ecuador, SRI.

Actualmente se puede deducir los siguientes rubros:

Vivienda: son deducibles el arriendo de un único inmueble usado para la vivienda, los intereses de préstamos hipotecarios otorgados por instituciones autorizadas, gastos destinados a la ampliación, remodelación, restauración, adquisición o construcción de una sola vivienda al igual que los impuestos prediales de un único bien inmueble.

Educación: en este rubro se puede declarar como deducibles las matrículas, pensiones en todos los niveles del sistema educativo, así como los cursos, seminarios de formación profesional debidamente aprobados por el Ministerio de Educación o del trabajo.

Además de que se considera como deducible los implementos como útiles, uniformes, textos escolares y material didáctico utilizados en la educación, así como los servicios prestados por los centros de cuidado infantil

Salud: se puede deducir los honorarios médicos a profesionales con título profesional avalado por el Centro Nacional de Educación Superior (Conesup), al igual que los servicios de salud prestados por clínicas, hospitales, laboratorios clínicos y farmacias autorizadas por el Ministerio de Salud Pública.

Los medicamentos e insumos médicos como lentes y prótesis, además de la medicina prepagada y prima de seguros médicos. Todos estos valores se pueden declarar como deducibles.

Alimentación: las compras de alimentos para consumo humano y las pensiones alimenticias son declarables como deducibles.

Vestimenta: son deducibles todos aquellos gastos realizados por cualquier tipo de prenda de vestir.

En resumen, para que un contribuyente pueda deducir su impuesto a la renta debe primero tener los comprobantes y llenar su anexo que debe cumplir un cierto formato.

La tabla 1.3 muestra el formato en el cual se debe ingresar la información de los comprobantes, la asignación se la realiza de la siguiente manera: en la columna A el RUC del proveedor, en la columna B el número de comprobantes de dicho proveedor, en la columna C se encontrará el valor de cuanto vamos a deducir de dicho proveedor y finalmente en la columna D a que rubro vamos a asignar a dicho valor de los 5 posibles.

Tabla 1. 3 Formato de gastos deducibles de impuestos aceptados por el SRI

RUC PROVEEDOR	CANTIDAD DE COMPROBANTES	BASE IMPONIBLE	TIPO DE GASTO
1103359624001	1	\$7.75	ALIMENTACION
1102828165001	1	\$5.50	ALIMENTACION
0691728137001	1	\$7.74	ALIMENTACION
1103910301001	5	\$25.80	ALIMENTACION
1104437817001	1	\$28.00	ALIMENTACION
1102061080001	1	\$5.50	ALIMENTACION
1102650072001	1	\$9.50	ALIMENTACION
1102654793001	1	\$8.25	ALIMENTACION
1104787468001	2	\$26.79	ALIMENTACION
1103205983001	1	\$14.91	ALIMENTACION
1103761027001	2	\$26.85	ALIMENTACION

1104375710001	1	\$5.25	ALIMENTACION
1103757504001	2	\$21.65	ALIMENTACION
1104758469001	1	\$19.20	ALIMENTACION
1101749404001	2	\$8.18	ALIMENTACION

Fuente: Autores

Elaboración: Autores.

1.6.2 Dispositivos Móviles

Los dispositivos móviles son equipos compactos de tamaño reducido capaces de procesar datos e información, con conexiones a redes móviles o redes de datos, etc. Han sido diseñados para cumplir una o varias funciones en específico, de acuerdo con esta definición, existe una gran cantidad de dispositivos móviles, desde reproductores de audio, GPS entre otros como se menciona en (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, n.d.), en este proyecto nos centramos en el uso del teléfono móvil (smartphone).

Los Smartphones o teléfonos inteligentes, son equipos que al igual que los teléfonos móviles “convencionales” permiten realizar llamadas o enviar mensajes de texto, pero a diferencia de éstos, cuentan con características similares a un computador. Los teléfonos inteligentes tienen la capacidad de ejecutar o permitir a instalación de programas o aplicaciones, estas pueden ser desarrolladas por el fabricante o por terceros.

1.6.2.1 Sistemas Operativos Móviles

Los sistemas operativos móviles, son una capa de software básico que permite la interacción entre el usuario y el hardware del dispositivo, las principales funciones del Sistema Operativo son administrar los recursos del equipo para que puedan ser utilizados de manera eficiente, coordinar el hardware y organizar archivos y directorios. En la actualidad Android e IOS son los sistemas operativos móviles con mayor acogida en el mercado. El sistema operativo que se utilizó para el desarrollo del proyecto fue el Android.

1.6.2.1.1 IOS

Este es un sistema operativo móvil desarrollado por Apple en sus inicios fue creado para el iPhone, pero con el tiempo fue adaptado y modificado para que trabaje en diferentes dispositivos de la marca (IPad e IPod). Este sistema está basado en el concepto de manipulación directa, el usuario puede interactuar directamente con la pantalla del dispositivo por medio de gestos multitáctiles como toques, pellizcos y deslices. Algunas de las características de este sistema operativo son: creación de carpetas, centro de notificaciones, multitarea entre otros (gcfaprendelibre.org, 2016).

1.6.2.1.2 *Android*

Android es un sistema operativo de código abierto adquirido por Google y Open Handset Alliance, el objetivo de este es satisfacer las diferentes necesidades de los operadores y fabricantes. Está basada en Linux, este usa dos tipos de máquinas virtuales: Davilk o Android Runtime, que son variantes de Java Virtual Machine (Pungacho Sánchez, 2017).

Las principales características de Android son:

- Crear aplicaciones haciendo uso de distintas herramientas y bibliotecas.
- Puede ser ampliado para incorporar nuevas tecnologías ya que es de código abierto.
- Ha sido diseñado para optimizar los recursos y el hardware del equipo móvil.
- Está en constante evolución y desarrollo.

1.6.2.1.3 *Arquitectura Android*

Previo al desarrollo de una aplicación móvil en Android, es importante saber la estructura de sistema operativo Android.

Android es una pila de software de varios niveles o capas creada para una variedad amplia de dispositivos. Cada capa o nivel puede ocupar elementos de la capa inferior para realizar alguna función. En el diagrama de la Figura 1.1 se muestran los componentes principales de la plataforma Android.

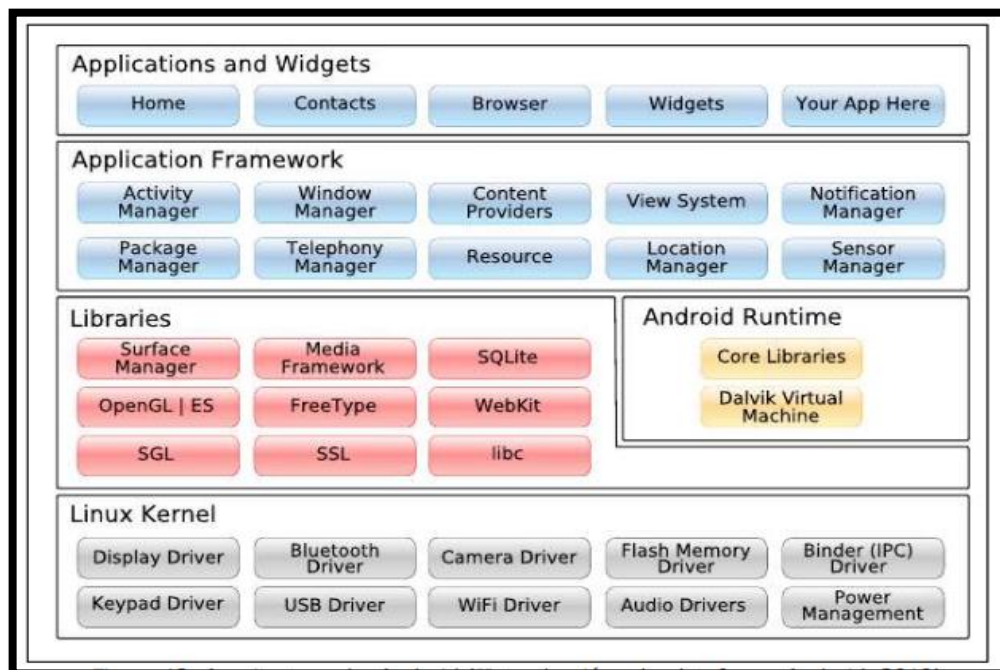


Figura 1. 1 Arquitectura Android

Fuente: Tomado de (Ciencias Empresariales & Andrea Ramírez Norambuena Christopher Edgardo Arevalo Gonzales, 2010)

Elaboración: Ramírez, Andrea; Arevalo, Christopher.

Aplicaciones: todas las aplicaciones están escritas usando lenguaje de programación java, estas pueden estar formadas por: Services, Activity, Intent, Broadcast y Content Providers.

Framework de Aplicaciones: el desarrollador tiene acceso a los diferentes APIs usados por las aplicaciones base. Su arquitectura está diseñada para simplificar el reuso de componentes, cualquier usuario puede publicar sus aplicaciones y que hagan uso de este sujeto a reglas de seguridad del framework (sites.google.com, n.d.).

Una capa de servicios disponibles para aplicaciones incluye:

- **Activity Manager:** maneja y administra el ciclo de vida de las aplicaciones y provee un comportamiento común en la navegación (Ciencias Empresariales & Andrea Ramírez Norambuena Christopher Edgardo Arevalo Gonzales, 2010).
- **Windows Manager:** organiza o que se va a presentar en pantalla.
- **Content Provider:** se encarga de encapsular los datos que se compartirán con otras aplicaciones para tener control sobre el acceso a la información.
- **Views:** éstos son elementos como botones, listas, cuadros de texto, etc. Estos ayudan a construir una interfaz de usuario.
- **Package Manager:** esta biblioteca permite obtener la información de los paquetes instalados en el dispositivo, además permite gestionar la instalación de paquetes adicionales.
- **Location Manager:** determina la posición geográfica del equipo haciendo uso de GPS o de redes disponibles.
- **Sensor Manager:** permite el uso de elementos de hardware como el giroscopio, acelerómetro, etc.
- **Cámara:** esta librería permite el uso de las cámaras del equipo ya sea para filmar o tomar una fotografía.
- **Multimedia:** esta permite la visualización y reproducción de contenido multimedia en el equipo.

Bibliotecas: el sistema operativo incluye varias bibliotecas C/C++ usadas por diferentes componentes. Estas se muestran al desarrollador a través del framework de aplicaciones, algunos ejemplos de estas son: Bibliotecas de gráficos, 3D, bibliotecas C standard (sites.google.com, n.d.).

Runtime de Android: incluye bibliotecas que dan la mayor parte de las funciones disponibles en el lenguaje de programación Java. Cada aplicación corre su propio proceso. La máquina

virtual se basa en registros y corre clases compiladas por java que han son transformadas en formato .dex por la herramienta “dx” (Ciencias Empresariales & Andrea Ramírez Norambuena Christopher Edgardo Arevalo Gonzales, 2010).

Núcleo – Linux: Android depende de la versión 2.6 de Linux para el uso de servicios del sistema como: gestión de memoria, seguridad, gestión de procesos, drivers, etc. También actúa como de conceptualización entre el hardware y software (Developers Android, n.d.).

1.6.2.1.4 *Aplicaciones móviles*

El desarrollo móvil está en constante evolución, existe una gran cantidad de aplicaciones móviles, estas sirven para entretener, informar, organizar y comunicar al usuario. Las aplicaciones están diseñadas para usar un determinado hardware como: cámara, pantalla, altavoces, procesador etc. La “limitación” que estas poseen son las que el equipo tiene, ya que estas se desarrollan en base a pre-requisitos que el equipo debe cumplir.

1.6.2.1.4.1 *Ciclo de vida de una aplicación*

Este proceso se crea cuando una parte del algoritmo debe ser ejecutado y continua vivo hasta no ser requerido por el sistema. Una característica importante es que el tiempo de vida de un proceso, no está controlado directamente por una aplicación, quien determina el tiempo de vida es el sistema operativo, basándose en el conocimiento que tiene sobre las partes de la aplicación, estas tareas son tareas netamente del sistema operativo.

1.6.2.1.4.2 *Ciclo de vida de una actividad*

Como ya se comentó, Android trabaja con una pila de actividades, cada vez que una nueva actividad se crea, se posiciona en lo más alto y la actividad anterior permanece debajo de ella. La tabla 1.4 presenta todos los estados de una actividad.

Tabla 1. 4 Estados de una actividad.

Estado	Descripción
onCreate	Es ejecutado cuando se crea por primera vez una actividad (inicio de la aplicación).
onRestart	Se ejecuta cuando se reinicia la aplicación.
onStart	Se ejecuta cuando la aplicación está visible al usuario.
onResume	Se ejecuta mientras el usuario está interactuando con la aplicación.
onPause	Se ejecuta cuando se va a interactuar con una actividad anterior.
onStop	Se ejecuta cuando la actividad deja de estar visible para el usuario.
onDestroy	Se ejecuta previa a la destrucción de la actividad.

Fuente: Autores

Elaboración: Autores.

La figura 1.2 presenta la relación que existe entre los estados a la hora de implementar un aplicativo.

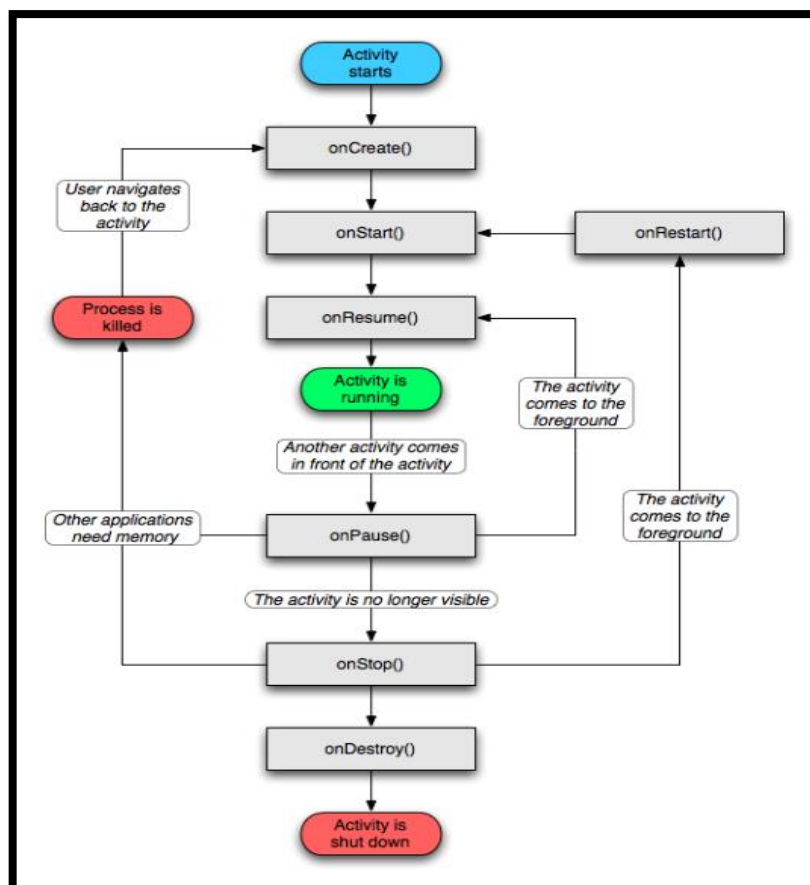


Figura 1. 2 Ciclo de vida de un aplicativo Android

Fuente: Tomado de (Daniela et al., 2012)

Elaboración: Aramayo, Daniela; Gascuña, Alberto; Sanchez, Eduardo.

1.7 Procesamiento digital de imágenes

El procesamiento de imágenes digitales hace referencia a la manipulación de forma digital de imágenes. Uno de los principales objetivos de este, es mejorar el aspecto y calidad de una imagen haciendo uso de un conjunto de técnicas para facilitar la búsqueda de información (Ocampo Carrion, 2011).

Las fases del procesamiento digital de imágenes por lo general están divididas en 6 áreas estas son: adquisición, pre-procesado, segmentado, descripción y finalmente reconocimiento e interpretación. La figura 1.10 presenta un esquema básico de reconocimiento de caracteres.

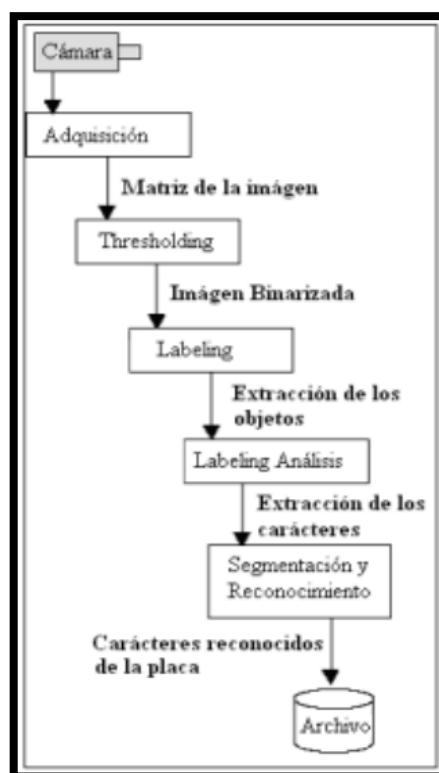


Figura 1. 3 Esquema de reconocimiento básico

Fuente: Tomado de (Alonso & Torres, 2014)

Elaboración: Rolón, Ricardo; Domínguez, Vicente.

1.7.1 Adquisición

Este proceso se encarga de la obtención digital de la imagen con la cual se va a trabajar, esta de la puede capturar haciendo uso de un equipo que disponga una cámara, estos equipos son capaces de convertir todo lo capturado en el medio físico y digitalizarlo dividiendo la imagen en regiones conocidas como píxeles.

1.7.1.1 Imagen digital

Las imágenes digitales son funciones bidimensionales $I = f(x, y)$, en donde x y y son coordenadas espaciales en un plano, los valores que toma x y y son proporcionales a la intensidad o brillo I . en resumen una imagen digital puede representarse por una matriz en la cual sus elementos toman el nombre de pixel, este es la parte más pequeña de una imagen digital (León Aucapiña, 2017).

La unidad con la que comúnmente se refiere al tamaño de una imagen se denomina megapixel (un millón de pixeles) ésta define el tamaño máximo a la que un dispositivo puede crear una imagen digital. La manera de representar una imagen es por medio de un matriz, en donde las posiciones de los pixeles están asociadas a los valores m, n de los elementos de dicha matriz, donde m es el número de columnas y n es el número de filas en la imagen (Báez López, 2006).

En el procesamiento de imágenes digitales se trabaja con cuatro tipos de imágenes (Sanchez Tirado, 2006) estas pueden ser:

Imágenes indexadas: reduce los colores a un máximo de 256, de esta manera se elimina información y disminuye el tamaño del archivo.

Imágenes RGB (Rojo-Verde-Azul): reproduce los colores en la pantalla utilizando tres canales de 8 bits con un total de 24 bits de color por pixel, soporta tres tipos de formatos BPM, PNG y JPG.

Imágenes en escalado de grises: utiliza diferentes tonos de gris, en imágenes de 8 bits, puede tener hasta 256 tonalidades diferentes, los valores de los bits están comprendidos entre 0 y 255 (negro y blanco).

Imágenes binarias: cuentan con una profundidad de color de 1 bit, representa los pixeles de la imagen usando uno de dos valores de color, blanco o negro.

1.7.2 Pre-procesado

Consiste en aplicar un conjunto de técnicas que permiten mejorar las condiciones visuales de la imagen, así puede adecuarse para su posterior análisis. esta fase incluye técnicas como: realce de contraste, eliminaciones de áreas innecesarias, realce de detalles y eliminación de ruido (Garcia Santillán, 2008).

1.7.3 Segmentación

Esta etapa consiste en aplicar técnicas de agrupación para excluir zonas que no son de interés dentro de la imagen. Es decir, dividir la imagen en grupos de píxeles o regiones, de esta forma se obtiene una nueva imagen únicamente con los objetos que interesen.

Los algoritmos de segmentado se basan en el uso de dos propiedades básicas del escalado de grises que son: similitud y discontinuidad (Palomino & Roman Concha, 2014).

Discontinuidad: divide la imagen basándose en cambios de los niveles de gris.

Similitud: divide la imagen realizando una búsqueda de zonas con valores similares, las técnicas usadas se basan en: crecimiento de la región, división de región y umbralización.

1.7.4 Representación y descripción

El grupo resultante de píxeles segmentados que constituyen el contorno de una determinada región, es representado y descrito para su posterior procesado.

Existen dos formas de representar una región: la primera en términos de su contorno y la segunda en términos de píxeles que conforman dicha región, en ambos casos se necesita convertir adecuadamente la información (López Airam, 2007).

Luego de describir el área de representación seleccionada. Se elige una representación externa cuando el objeto de interés tiene características como esquinas e inflexiones y una representación interna cuando el objeto principal se centra en propiedades como la textura y el color (López Airam, 2007).

1.7.5 Reconocimiento e interpretación

La última etapa del procesamiento digital de imagen es en la que se asigna una etiqueta a un objeto en base a la información brindada por sus descriptores. Se procede a interpretar la escena resultante con todos los cambios generados por las etapas anteriores.

(Álvarez Durán, 2014) comenta que la interpretación implica asignar significado a un grupo de caracteres reconocidos.

1.8 Operaciones Generales en imágenes digitales

Varias operaciones pueden ser realizadas en imágenes digitales, a continuación, se presentan las más relevantes en el ámbito del procesamiento de imágenes.

1.8.1 Umbralización

La umbralización o “thresholding”, es el proceso en el cual se busca el umbral óptimo que permita diferenciar dentro de la misma imagen un objeto de otro (objetos de fondo y objetos del primer plano). Las diferentes técnicas de umbralización buscan obtener el valor de umbral que permita binarizar la imagen separando correctamente el fondo y el objeto.

Muchas técnicas de umbralización se basan en la información estadística que brinda el histograma, este sirve de gran ayuda en imágenes donde los objetos tienen una textura o superficie homogénea con un fondo más o menos uniforme. El inconveniente de la umbralización es encontrar el valor adecuado del umbral (T) que permitan una separación adecuada entre el fondo y el objeto (Cattaneo, Larcher, Ruggeri, Herrera, & Bionni, 2011). La figura 1.11 muestra un histograma de niveles de gris en cada pixel.

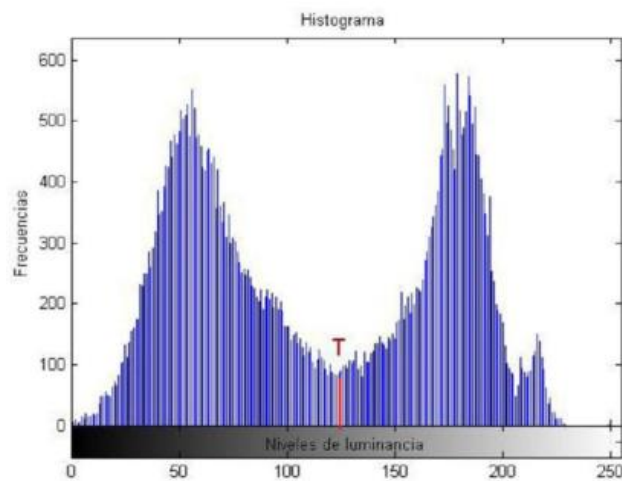


Figura 1. 4 Histograma ideal

Fuente: tomado de (Cattaneo et al., 2011)

Elaboración: Cattaneo; Larcher; Ruggeri, Herrera.

1.8.2 Binarización

La mayoría de algoritmos de reconocimiento parten de imágenes binarias, por ende, es conveniente convertir dicha imagen. Esto permite disminuir el tamaño de los datos que serán procesados.

La binarización es una variante del proceso de umbralización. Para binarizar se requiere un valor umbral, este es un valor de intensidad que consiste, en dejar en uno a todos los pixeles mayores o iguales al valor umbral y en cero a todos los menores (Ramírez-Manzanares, 2014).

El valor umbral puede ser modificado manualmente o puede ser seleccionado de forma automática dando como resultado una imagen binaria (compuesta de unos y ceros), de esta forma se logra encontrar de forma eficiente y rápida el objeto de interés (León Aucapiña, 2017).

1.8.3 Transformaciones morfológicas

Esta técnica tiene como objetivo extraer estructuras geométricas. Las operaciones utilizadas para las transformaciones morfológicas tienen de entrada una imagen y la salida es una imagen modificada, a continuación, se describen los elementos requeridos para el procesado morfológico (García Santillán, 2008).

1.8.3.1 Elemento Estructural

La utilización de un elemento estructural, es un conjunto de puntos que conforman una imagen, se determina la estructura sobre la cual se aplica la operación morfológica.

Este elemento puede tener cualquier forma o tamaño y si es elegido de acuerdo con la morfología que se va a interrelacionar en función de las formas que se desee extraer (Ocampo Carrion, 2011).

1.8.3.2 Operaciones Morfológicas

Estas brindan información sobre la estructura o forma de una imagen, de esta manera se puede extraer desde la imagen componentes útiles así como las características importantes de los objetos (León Aucapiña, 2017). Estas características son: Erosión, Dilatación, Apertura y Cierre.

- **Erosión:** se reemplaza el pixel con el menor valor para objetos luminosos y con el de mayor valor para objetos oscuros. Esta operación produce un efecto de contracción en los objetos (oscuros o luminosos) según sea el caso.
- **Dilatación:** esta operación es contraria a la erosión, se reemplaza con el pixel de menor valor para objetos oscuros y con el pixel de mayor valor para objetos luminosos
- **Apertura:** este se caracteriza por pulir los bordes de un hueco por medio de operaciones, elimina pequeños objetos para disminuir ruido. Primero realiza una erosión y luego una dilatación.
- **Cierre:** este método es aplicable cuando se requiere llenar espacios y conectar objetos que se encuentran próximos entre sí. Primero realiza una dilatación y luego una erosión.

1.8.4 Filtrado espacial

Esta es una operación usada para atenuar o resaltar detalles dentro de una imagen, se basa en los valores de los píxeles que están dentro un área de interés. Opera sobre un grupo de píxeles que bordean a un píxel central, por lo que los píxeles vecinos proporcionan información importante sobre el brillo en el área que se está tratando (Jara Oyarzo, 2006).

- **Filtrado especial paso bajos**

Este tipo de filtrado se emplea para hacer que una imagen aparezca algo borrosa y también para minimizar el ruido, el objetivo es suavizar los contrastes espaciales presentes en la imagen. El filtro tiene por defecto permitir el paso de componentes de baja frecuencia espacial (León Aucapiña, 2017).

- **Filtrado Promedio**

El proceso de filtrado paso bajos consta de reemplazar el valor de cada píxel por el promedio de los niveles de gris de los píxeles vecinos definidos por la máscara. Según la ecuación [1] dada una imagen $f(x, y)$ de tamaño $n \times m$, $g(x, y)$ es el valor del nivel de gris suavizado de la imagen en un punto (x, y) , este se obtiene promediando los valores de nivel de grises de los puntos comprendidos en cierta vecindad de (x, y) (González & Wintz, 1996).

$$g(x, y) = \frac{1}{M} \sum_{(n,m) \in S} f(n, m) \quad [1]$$

El conjunto de coordenadas de los puntos vecinos a (x, y) es S , incluyendo el propio (x, y) y el número de puntos de vecindad es M . Para comprender esta parte vamos a analizar las figuras 1.12 y 1.13 tomadas de (González & Wintz, 1996), en donde se desea cambiar el valor en (x, y) , en otras palabras asignar el valor promedio a $f(x, y)$.

		⋮		
	$(x - 1, y - 1)$	$(x, y - 1)$	$(x + 1, y - 1)$	
...	$(x - 1, y)$	(x, y)	$(x + 1, y)$...
	$(x - 1, y + 1)$	$(x, y + 1)$	$(x + 1, y + 1)$	
		⋮		

Figura 1. 5 Máscara centrada en (x, y) con sus puntos respectivos
Fuente: tomado de (González & Wintz, 1996)
Elaboración: González, R & Wintz, P.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figura 1. 6 Coeficientes de la máscara
Fuente: tomado de (González & Wintz, 1996)
Elaboración: González, R & Wintz, P.

La operación se realiza centrando la máscara en el punto (x, y) y multiplicando cada punto debajo de la máscara por el coeficiente correspondiente, dando una suma como resultado.

$$g(x, y) = \frac{1}{9} \{f(x - 1, y - 1) + f(x, y - 1) + f(x + 1, y - 1) + f(x - 1, y) + f(x, y) + f(x + 1, y) + f(x - 1, y + 1) + f(x, y + 1) + f(x + 1, y + 1)\}$$

Al hacer un promedio de vecinos aparece una difuminación de los bordes y otros detalles de contraste.

1.9 Reconocimiento óptico de caracteres “OCR”

Si se desea procesar información en un computador desde algún documento físico, existen dos opciones: la primera consiste en introducir toda esa información de manera manual haciendo uso del teclado del equipo, siendo esta una opción larga y tediosa. La segunda opción es automatizar este proceso haciendo uso de un sistema de reconocimiento óptico de caracteres OCR, el cual haciendo uso de software y hardware adecuado puede reducir de forma considerable en tiempo de entrada de los datos (Javier & Fernández, 2008).

Como menciona (Javier & Fernández, 2008) esta tecnología engloba varias técnicas basadas en estadísticas, transformadas, comparaciones y forma de caracteres, todas estas técnicas se complementan entre sí, de esta forma permite identificar de manera automática los distintos tipos de caracteres alfanuméricos que existen en un determinado alfabeto, esto se analizó detalladamente en el apartado “1.7 Procesamiento de Imágenes Digitales”, dichas técnicas pueden disminuir el margen de error y la mal interpretación de algún carácter, ya que la precisión al momento de digitalizar disminuye si el conjunto de símbolos es mayor.

El reconocimiento óptico de caracteres permite extraer automáticamente información (caracteres de un determinado alfabeto) partiendo de una imagen, para poder realizar este proceso se emplean varios algoritmos informáticos automatizados que permiten digitalizar el

texto expuesto en una imagen, el resultado ideal de esta conversión, debería ser el mismo del texto en la imagen de entrada (Singh, 2013).

Los pasos para convertir texto contenido en una imagen a texto procesable son los siguientes; en primer lugar, se escanea el archivo haciendo uso de una cámara, luego de esto se ingresa el mapa de datos (imagen binarizada) a un motor OCR el cual permitirá extraer todo el texto procesable.

La coincidencia e interpretación de caracteres está en relación directa con la calidad de la imagen a procesar.

Desde el nacimiento de algoritmos OCR, muchos servicios han adecuado este proceso para aumentar su rendimiento.

Algunos ejemplos de uso de algoritmos de reconocimiento son: reconocimiento de matrículas, reconocimiento de texto manuscrito, indexación de bases de datos, etc.

La figura 1.14 presenta un esquema básico de trabajo de algoritmos de reconocimiento óptico de caracteres, estos algoritmos se basan en 4 etapas: binarización o adecuación de la imagen, fragmentación o segmentación, adelgazamiento de componentes o extracción de características y finalmente reconocimiento o comparación de patrones.

- **Binarización o adecuación (Pre-procesado):** la mayoría de algoritmos de reconocimiento óptico de caracteres parten de una imagen binaria, de esta forma es conveniente pasar una imagen a escala de grises o a blanco y negro (filtrado) para que se conserven las propiedades esenciales de la imagen y no perder información. Ya se habló de este proceso en el apartado 1.7, con el que se obtiene claramente los contornos de los caracteres y símbolos de la imagen.
- **Segmentación o fragmentado de la imagen (Selección de zona de interés):** esta es una de las mayores dificultades del reconocimiento, una vez procesada la imagen se fragmentará en diferentes componentes conexas (parte en que los pixeles son adyacentes entre sí). Esto permite descomponer un texto en diferentes partes significativas y finalmente se pueda proceder a reconocer (Javier & Fernández, 2008). La figura 1.15 presenta el proceso de segmentación normalizado.
- **Adelgazamiento de componentes o extracción de características (Representación digital):** una vez normalizada y aislados los componentes conexos, se aplica un proceso de adelgazamiento para cada uno. Consiste en realizar un barrido de pixeles excesivos (puntos de los contornos) conservando su tipología. Toda esta información está representada en una matriz bidimensional de valores binarios.

- Reconocimiento o comparación de patrones (Distinción de carácter contenido en la imagen):** esta es la etapa final, se comparan los caracteres obtenidos con los teóricos (patrones de un alfabeto). El éxito del reconocimiento óptico de caracteres se basa en el buen funcionamiento de esta etapa (existencia de un amplio alfabeto). Existen diferentes formas en las cuales se lleva a cabo la comparación, el método más común es el de Proyección (conteo de pixeles horizontales y verticales por cada carácter, luego los compara con un alfabeto hasta encontrar la coincidencia), existen otros métodos como algoritmo KNN, arboles de decisión, redes neuronales, métodos geométricos o estocásticos, etc. (Javier & Fernández, 2008)

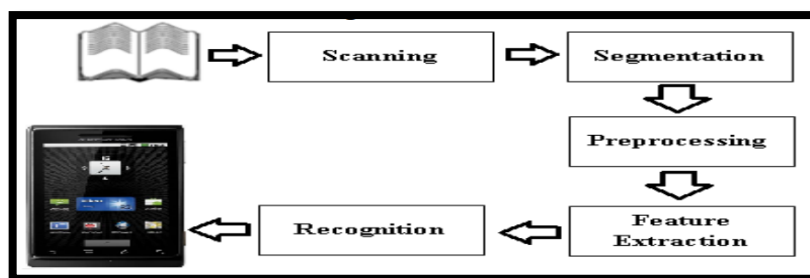


Figura 1. 7 Esquema básico de reconocimiento óptico de caracteres

Fuente: tomado de (Ravina, Supriya, & Nilam, 2013)

Elaboración: Ravina, M; Supriya, I; Nilam D.

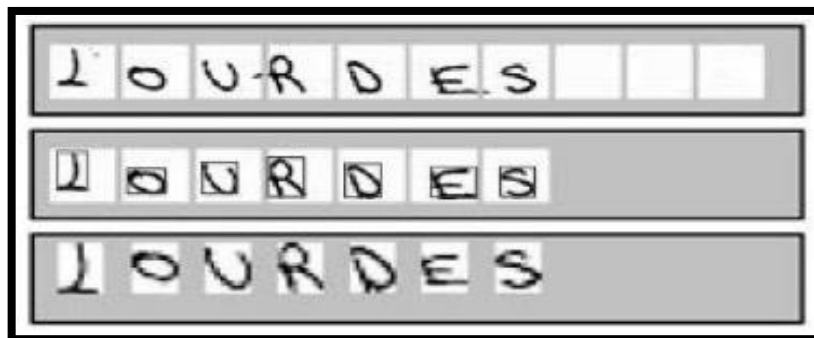


Figura 1. 8 Segmentación y normalización, desde arriba hacia abajo: imagen original,

eliminación de ruido y detección, normalización y tamaño de caracteres

Fuente: tomado de (Ravina et al., 2013)

Elaboración: Sanchez, J.; Sandoñis, V.

- Problemas con el reconocimiento óptico de caracteres**

Partiendo de una imagen en perfectas condiciones, es decir, una imagen que cuente con dos niveles de gris, el reconocimiento de caracteres se realiza básicamente comparando cada carácter con los de una plantilla o patrón que contienen todas las posibles coincidencias. Pero

las imágenes reales no son perfectas, por ende, el reconocimiento óptico puede presentar varios inconvenientes como:

- El dispositivo encargado de capturar la imagen, puede introducir niveles de grises o ruido que no pertenece a la imagen original.
- El espaciamiento entre caracteres, al no ser siempre el mismo puede provocar problemas de reconocimiento.
- La conexión de dos o más caracteres por pixeles comunes puede causar algún tipo de error.
- La tipografía, perspectiva, condiciones lumínicas afectan de gran manera al reconocimiento.

1.9.1 Motores OCR

En la actualidad existen varios algoritmos o motores de reconocimiento óptico de caracteres muchos de ellos son de código abierto, a continuación, se describirá los más utilizados.

1.9.1.1 Tesseract OCR

Es un motor de código abierto que se desarrolló inicialmente entre 1984 – 1994 por HP, desde 2006 es desarrollada por Google.

Actualmente es capaz de reconocer múltiples idiomas de forma inmediata y tiene un soporte Unicode (UTF-8), la librería permite varios formatos de salida como: texto plano, HTML, PDF, TSV.

Se debe tener en cuenta que en la mayor parte de casos para mejorar el reconocimiento se debe mejorar la calidad. Esta librería también puede ser modificada para reconocer idiomas adicionales (Smith, 2017).

La última versión estable de esta librería es la 3.05.01 fue lanzada el primero de junio de 2017.

- **Arquitectura Tesseract**

Tesseract funciona con una tecnología de análisis de diseño de página desarrollada de forma independiente. De esta forma Tesseract al igual que los otros motores acepta imágenes binarias y puede manejar texto tradicional en blanco sobre negro y viceversa.

Los bordes del componente se almacenan, se realiza la anidación de contornos que reúne los contornos para formar un Blob. Tales Blobs están organizados en líneas de texto. Las líneas de texto se analizan para el tono fijo y el texto proporcional. Luego, las líneas se dividen en palabras según el espaciado de caracteres. El tono fijo se divide en celdas de caracteres y el

texto proporcional se divide en palabras por espacios definidos y espacios difusos, finalmente resuelve dichos espacios para ubicar el texto (Ravina et al., 2013).

La arquitectura Tesseract se la puede visualizar en la figura 1.16.

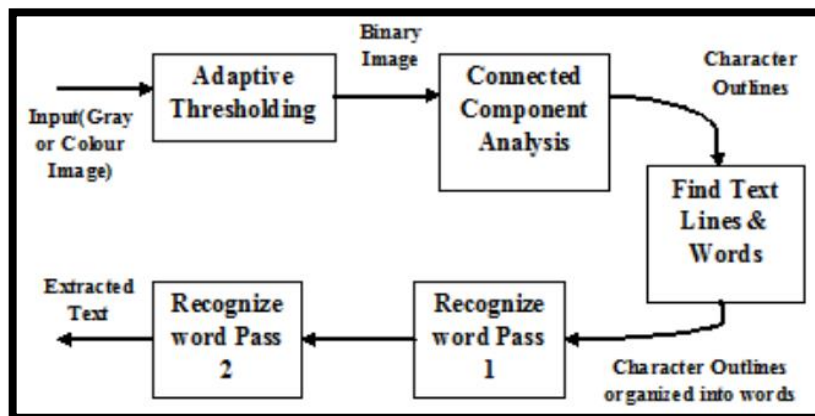


Figura 1. 9 Arquitectura Tesseract
Fuente: tomado de (Ravina et al., 2013)
Elaboración: Ravina, M; Supriya, I; Nilam D.

1.9.1.2 Google Mobile Vision

Mobile visión es una librería desarrollada por Google que es capaz de realizar reconocimiento óptico de caracteres en tiempo real de forma eficiente y correcta, sin requerir excesivo tiempo de computo, esta puede reconocer múltiples idiomas y sus márgenes de error son bajos.

Ésta cuenta con AI (inteligencia artificial) y Machine Learning (aprendizaje automático). Esta API permite encontrar objetos en fotos y videos, incluye detectores, que ubican y describen objetos visuales, está impulsada por eventos que rastrean la posición de dichos objetos.

Actualmente, Mobile Vision API da al usuario la posibilidad de detectar rostros, realizar lecturas de código de barras y códigos QR (Quick Response) y realizar OCR, todo esto se puede implementar de forma grupal o por separado (Google Developers, 2018).

Estructuración del texto

La API Mobile Vision permite al usuario segmentar el texto a reconocer de tres formas: bloques, líneas, y palabras.

- **Bloque:** conjunto contiguo de líneas de texto, como una columna o un párrafo.
- **Línea:** conjunto contiguo de palabras en sobre el mismo eje vertical.

- **Palabra:** conjunto contiguo de caracteres alfanuméricos en el mismo eje vertical.

La figura 1.17 resalta los tipos de segmentación de texto que realiza Google Mobile Vision, recuadro de color celeste, es un bloque de texto, recuadros resaltados de color azul, son líneas de texto y finalmente los de color azul oscuro son palabras.

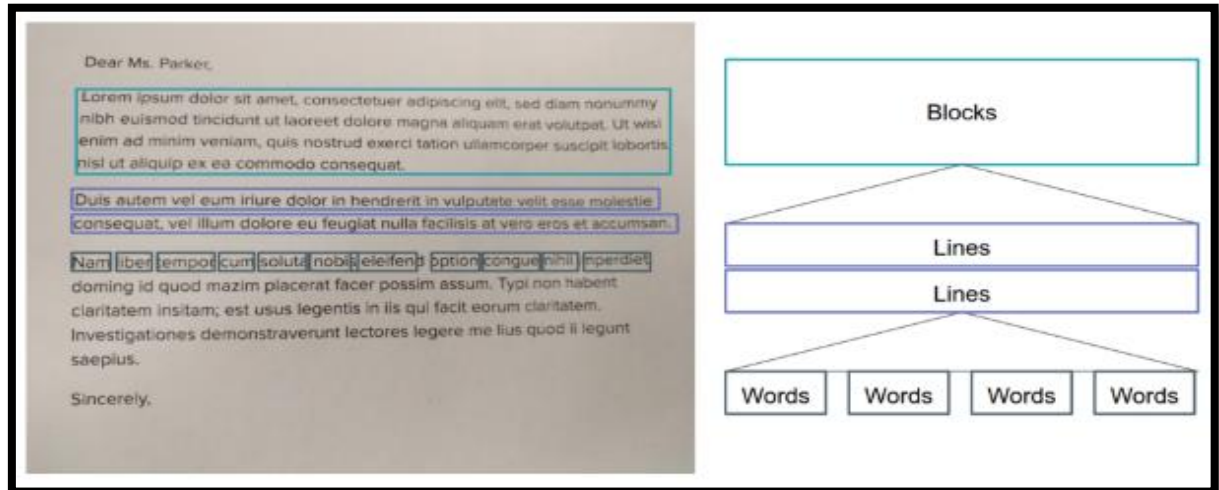


Figura 1. 10 Tipos de segmentación de texto en Google Mobile Vision

Fuente: tomado de (Google Developers, 2018)

Elaboración: Google Developers.

1.10 Base de Datos

De forma sencilla una base de datos es un contenedor digital, este permite almacenar gran cantidad de información de manera ordenada. Toda base de datos está estructurada de dos partes: la información almacenada y el programa que ese encarga de gestionar y procesar dichos datos. Estos programas se los conoce con el nombre de SGDB (software de gestión de base de datos), permiten comunicar la base de datos con las aplicaciones y usuarios que acceden a ella.

En la actualidad existes múltiples servicios de base de datos como Firebase, Firestore, MySQL, SQLite, etc.

- **Firebase**

Firebase es un grupo de herramientas orientadas a facilitar la creación de aplicaciones de alta calidad, esta es una nueva plataforma de desarrollo móvil en la nube de Google. Está disponible para Android, IOS y web. Esta herramienta brinda servicios como (figura 1.18): alojamiento en tiempo real (Firebase Database), autenticación (Firebase Authentication), mensajería en la nube (Cloud Messaging), Almacenamiento (Storage), Hosting, configuración

remota (Remote Config), laboratorio de prueba (Test Lab) y reporte de errores (Crash Reporting).

También permite la visualización de la actividad de los usuarios de forma estadística. En la tabla 1.5, se detallan algunas ventajas y desventajas de usar Firebase.

Tabla 1. 5 Ventajas y desventajas de uso Firebase.

Ventajas	Desventajas
Fácil implementación	Construir índices manualmente
No necesita configuraciones en el servidor, sin scripts PHP	En caso de ser necesario, registro de eventos manual
El modelo de reglas de seguridad que permite aplicar privilegios de lectura/escritura y validación de datos en todo el árbol dentro de la base de datos	Las reglas de validación de datos no admiten objetos complejos (se necesita validar los nodos secundarios individuales por separado)
Modo free soporta hasta 50 conexiones simultaneas	
Soporte integrado para servicios de autenticación como Facebook, Google, y Twitter	
Actualizaciones en tiempo real	

Fuente: Autores

Elaboración: Autores.

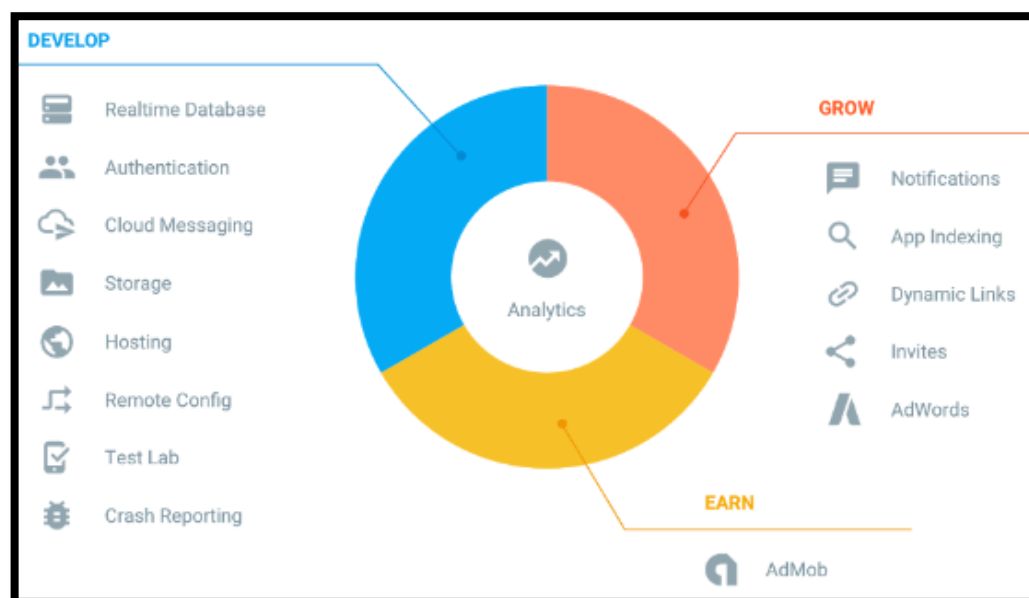


Figura 1. 11 Servicios que ofrece Firebase

Fuente: tomado de (Google Developers, 2018)

Elaboración: Google Developers.

1.11 Códigos QR (Quick Response)

Un código QR a diferencia que un código de barras, es un código bidimensional que puede almacenar caracteres información como texto, SMS, correo electrónico, URL, imagen, audio y algunos otros formatos.

Se puede extraer la información almacenada en los códigos con varias librerías, pero la comúnmente utilizada es la Biblioteca de Google Vision. La capacidad de almacenamiento de un código QR varía dependiendo del tipo información que contenga, en caso de ser solo caracteres numéricos soporta máximo 7089 caracteres, alfanuméricos máximo 4296 caracteres, binarios máximo 2953 bytes.



Figura 1. 12 Código QR

Fuente: Autores

Elaboración: Autores.

CAPÍTULO II

DISEÑO DE SOLUCIÓN Y SELECCIÓN DE HERRAMIENTAS

Este capítulo trata diferentes puntos como: metodología usada para el desarrollo del aplicativo móvil, diseño de solución planteada y selección de herramientas.

Se detallan temas como requisitos funcionales y no funcionales, identificación del actor principal, generación de casos de uso para el usuario y finalmente se presenta un mapa de navegación detallado del funcionamiento del aplicativo. Para el apartado de selección de herramientas, se establecieron criterios como usabilidad, desarrollo previo y adaptabilidad entre módulos, se realizó un análisis comparativo para posterior elección de acuerdo a nuestros requerimientos.

2.1 Metodología

El desarrollo de software es una tarea compleja, esta viene de la mano con una metodología que permita el avance del mismo, algunos autores definen una colección de procedimientos, herramientas, técnicas y documentos auxiliares mediante los cuales un proyecto puede ser organizado y desarrollado exitosamente.

Existen dos tipos de metodologías aplicables: las metodologías tradicionales y las metodologías ágiles. Ambos tipos están orientados a estructurar, planificar y controlar el proceso de creación de algún tipo de software.

Como se menciona en (Gasca Mantilla, Camargo Ariza, & Medina Delgado, 2013) y (Avison & Fitzgerald, 2003) la mayoría de metodologías existentes constan de cinco etapas: análisis, diseño, desarrollo, prueba y entrega.

Este proyecto fue desarrollado siguiendo la **metodología de trabajo incremental**, la cual divide todos los procesos realizados en bloques o módulos funcionales que posteriormente serán analizadas por el usuario. Cada avance contempla la ejecución de pruebas sobre cada módulo, esto permite garantizar el correcto funcionamiento del sistema (Lamar, 1999).

2.2 Diseño de Solución

El objetivo de este proyecto es automatizar la forma actual de realizar la declaración de impuesto a la renta, haciendo uso de un aplicativo móvil. De esta forma se pretende que la intervención del usuario sea la mínima, por ende, no se requerirá invertir un excesivo tiempo en este proceso.

En base a la problemática expuesta en el primer capítulo se planteó desarrollar una aplicación que permita al usuario, extraer los datos tributarios de facturas, almacenar toda la información dentro de una base de datos y finalmente el usuario tendrá opción a generar un anexo de gastos deducibles de impuesto a la renta en un formato aceptado por el servicio de rentas

internas. Lógicamente para hacer uso de todos estos servicios, se debe tener acceso a Internet.

2.2.1 Tipos de comprobantes

Se pretende abarcar todos los comprobantes existentes a excepción de las facturas electrónicas, estas se adjuntan directamente a la plataforma web del Servicio de Rentas Internas. Trabaja con facturas físicas manuales y físicas impresas.

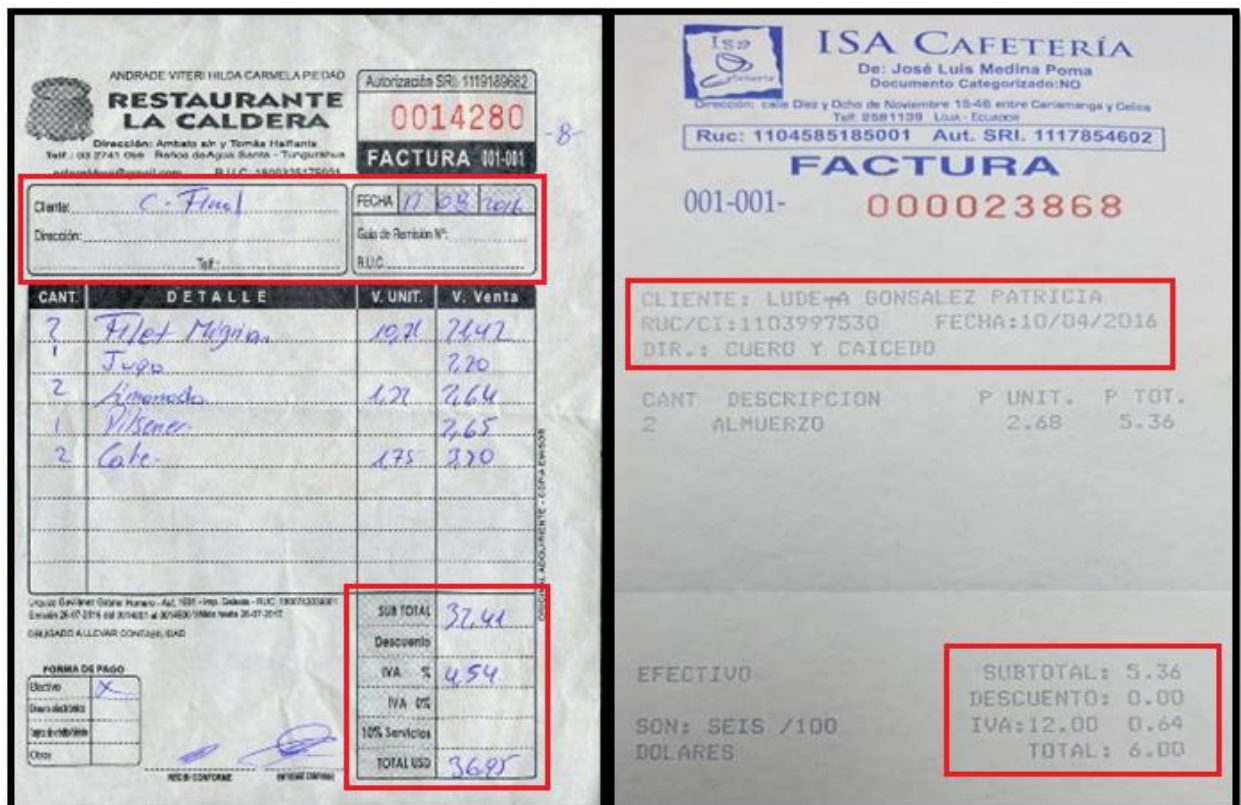


Figura 2. 1 Facturas físicas: a) Manual, b) Impresa

Fuente: Autores

Elaboración: Autores.

Todos los comprobantes contienen el mismo tipo de información (Servicio de Rentas Internas del Ecuador, 2018), los datos que se considera importantes son los que se va a extraer y se encuentran resaltados a continuación.

- Nombre del cliente
- Nombre del proveedor
- CI/RUC proveedor
- CI/Ruc cliente
- Número de factura
- Fecha
- Dirección
- Subtotal
- Tota
- IVA 12%
- IVA 0%
- Descuento

2.3 Estructura de la aplicación móvil

En base a la metodología, se optó por dividir el aplicativo en tres diferentes módulos: módulo de autenticación y registro, módulo de adquisición de datos y módulo de visualización (generación de anexo), de esta manera se puede desarrollar parte por parte, hacer pruebas de funcionamiento y finalmente integrar todo en una sola aplicación. La figura 2.2 muestra un esquema de cómo se estructuró la aplicación.

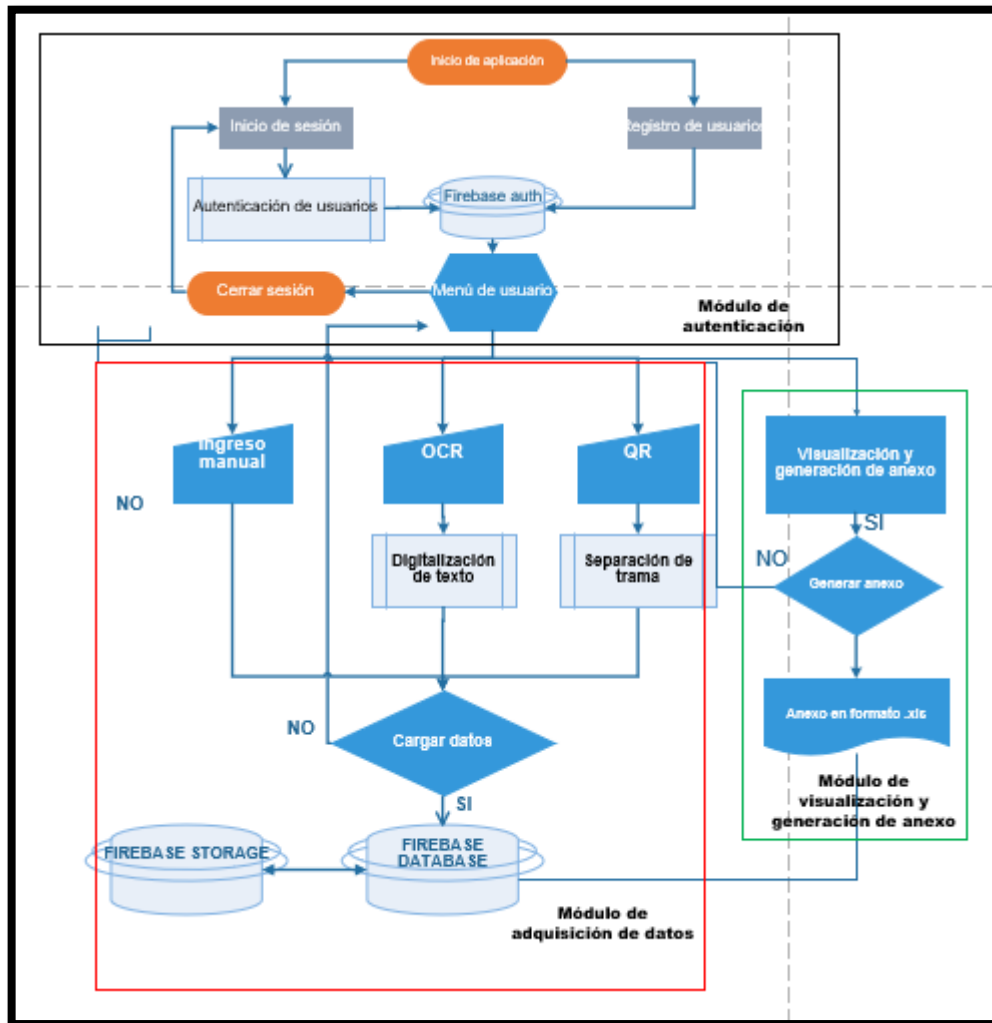


Figura 2. 2 Estructura del aplicativo móvil

Fuente: Autores

Elaboración: Autores.

2.3.1 Módulo de autenticación y registro de usuarios

Como se pretende que cada usuario almacene información y únicamente este tenga acceso a la misma, se optó por la creación de un apartado que se encargue de realizar el proceso de autenticación y registro de usuarios.

En este, el usuario debe ingresar sus datos y credenciales, serán validadas, si toda la información es correcta, se permitirá el acceso a todos los servicios que se ofrece dentro de la aplicación. Dicho modulo también permite el restablecer una contraseña en caso del que el usuario la olvidara.

2.3.2 Módulos de adquisición de Datos

Para abarcar con todos los tipos de comprobantes existentes, se planteó crear un método de ingreso por cada uno, dando como resultado tres formas en las cuales el usuario puede adjuntar información a la base de datos. Puede hacerlo de forma manual, escaneando un código QR (se explicará esta propuesta más adelante) y finalmente puede extraer información de usando Reconocimiento óptico de caracteres.

La comunicación existente entre la base de datos y cada uno de los módulos es bidireccional ósea de lectura y escritura. Lectura para verificar la existencia de algún comprobante y escritura para guardarlo en el caso de que no exista.

2.3.2.1 Ingreso manual

Se optó por este método para abarcar con las facturas y comprobantes físicos de escritura manual, como el que se presenta en la figura 2.1 (Izquierda). Este consta de un formulario en el que el usuario debe ingresar información como:

- Nombre
- Número de comprobante
- Fecha
- Cedula del comprador
- Nombre del proveedor
- RUC del Proveedor
- Subtotal
- IVA
- Rubros (Alimentación, Salud, Educación, Vestimenta, Vivienda).


Previo al envió a la base de datos, toda la información adjuntada será validará campo a campo para evitar que se guarden datos erróneos.

2.3.2.2 Ingreso mediante código QR

Este es un método de ingreso en el que se pretende que el proveedor implemente a su sistema de facturación, la factura impresa contendrá un código QR con toda la información importante de la misma (nombre, número de comprobante, fecha, cédula del comprador, nombre del

proveedor, RUC del proveedor, Subtotal, IVA, y los rubros) y tendrá cierta estructura (código de identificación e información ordenada) esto permitirá verificar si el código es correcto y adjuntar a información de forma automática a la base de datos.

El usuario únicamente debe dirigirse a esta opción dentro de la aplicación escanear dicho código y todos los datos se guardarán automáticamente. La figura 2.3 presenta un ejemplo de cómo estaría compuesta la factura



ITAJU
S.R.L.
INSTITUTO TÉCNICO
CASA 19872
AV. DIGNIDAD PARALELO N° 100
ZONA NORTE
TELÉFONO: 2222111
LA PAZ-BOLIVIA

NIT: 123456022
N° FACTURA: 529
N° AUTORIZACIÓN: 23041440

ORIGINAL
Escribenza de adultos y otros tipos de Escribenza

FACTURA

Lugar y fecha: La Paz, 15 de marzo de 2015 NIT/CI: 4313912
Señor(es): Gabriela Rosales
Alumno(a): José Barríos Rosales

CÓDIGO	DESCRIPCIÓN	SUBTOTAL
2000	Matrícula	100.00
1000	Mensualidad marzo-2015	500.00

Son: Seiscientos 00/100 Bolivianos. **TOTAL Bs** 600.00

Código de Control: BD-54-F1-4E-77

Fecha Límite de Emisión: 30/08/2015

ESTA FACTURA CONTRIBUYE AL DESARROLLO DEL PAÍS. EL USO ILÍCITO DE ÉSTA SERÁ SANCIONADO DE ACUERDO A LEY
Ley N°483: "El proveedor debe brindar atención al destinatario, con respeto, calidad y cordialidad a los usuarios y usuarias"

Figura 2. 3 Factura impresa con código QR

Fuente: Autores

Elaboración: Autores

2.3.2.3 Ingreso mediante OCR

Este método de ingreso implementa un algoritmo de reconocimiento óptico de caracteres, de esta forma se puede extraer información impresa en la factura como: Nombre, CI del Cliente, Fecha, Numero de factura, Subtotal y Total. El usuario únicamente debe realizar una captura clara del comprobante y la aplicación se encargará de llenar el formulario con los datos obtenidos, al final el usuario puede validar toda la información resultante, editar en caso de

ser necesario y realizar el desglose del subtotal por rubrica, todo previo al envío de información a la base de datos. Este método es únicamente aplicable a facturas impresas figura 2.1 B.

2.4 Almacenamiento de la información

Como se pudo analizar, todos los módulos propuestos en este proyecto están en comunicación directa con una base de datos. Para esto se realizó una búsqueda de herramientas que nos permitan estructurar la información y a su vez sea de fácil implementación. Estas búsquedas dieron como resultado la elección de la plataforma Firebase, esta ofrece múltiples servicios de los cuales se pretende utilizar cuatro de ellos: Firebase Authentication, Firebase Database, Firebase Storage y para seguimiento de comportamiento de la aplicación Crash Reporting. En cuanto a seguridad y acceso a la información, Firebase trabaja haciendo uso de reglas, y según el administrador lo requiera puede abrir o restringir el acceso a cierta información.

Firebase también nos va a permitir visualizar estadísticas de uso de la aplicación, reporte de bugs y problemas en la misma, esta es una razón más por la cual es una buena propuesta para implementar.

Dentro de Firebase Database, se creará un directorio en la nube con cada ID obtenida mediante el registro de usuario, en el cual se ira almacenando toda la información de cada módulo (módulo de registro y módulo de adquisición de datos).

2.5 Módulo de visualización y generación de resumen de gastos

Este apartado posee comunicación en tiempo real con la base de datos (solo lectura), cada cambio dentro de la base de datos y se reflejará de inmediato en este módulo. Como su nombre lo indica, este módulo permitirá la visualización de información almacenada en la base de datos (número de facturas existentes, montos por rubrica e incluso gráficos estadísticos con estos valores).

Aquí el usuario tiene la opción a generar el anexo con el resumen de gastos deducibles de impuesto, para esto se buscó herramientas que permitan generar un archivo Microsoft Excel, esta búsqueda dio como resultado el uso de la librería JEXCEL API.

2.6 Selección de herramientas

Se realizó un análisis para determinar las herramientas y librerías con las cuales se va a desarrollar la implementación del proyecto.

2.6.1 Selección de entorno de desarrollo

Para el desarrollo de aplicaciones móviles para el S.O. Android existen múltiples entornos de desarrollo como Android Studio, Eclipse y IntelliJ IDEA.

Este aplicativo móvil se lo desarrolló en Android Studio, esta plataforma tiene múltiples ventajas en relación a otros entornos de desarrollo.

Android Studio

Es un IDE de desarrollo oficial para el sistema operativo Android, ofrece características como:

- **Soporte:** con la intención de que este IDE esté presente al desarrollar cualquier tipo de aplicación donde se encuentre este SO. (Android Wear, Android Auto).
- **Herramientas Lint:** Detecta problemas de rendimiento, usabilidad y compatibilidad de versiones.
- **ProGuard:** Para optimizar y reducir el código del proyecto al exportar a APK.
- **Alertas en tiempo Real:** indicativos de errores de redacción o de compatibilidad

Los requerimientos para que este funcione son el de contar con al menos de 2 GB de memoria RAM y 400 MB de espacio libre, con la única obligación de contar con el Java Development kit en la versión 7 o superior con esto se asegura el funcionamiento de este IDE. La tabla 2.1 muestra una comparación rápida con su competencia directa Eclipse.

Tabla 2. 1 Comparativa Android Studio vs Eclipse.

Características	Android Studio	Eclipse
Sistema de construcción	Gradle	ANT
Construcción y gestión de proyectos basado en Maven.	Si	No
Construir variantes y generación de múltiples APK	Si	No
Refactorización y completado avanzado de código Android	Si	No
Firma APK y gestión de almacén de claves	Si	Si
Soporte para NDK (Native Development Kit: herramientas para implementar código nativo escrito en C y C++)	Si	Si
Soporte para Google Cloud Platform (Google Vision)	Si	No
Vista en tiempo real de renderizado de layouts	Si	No
Nuevos módulos en proyecto	Si	No
Editor de navegación	Si	No
Visualización de recursos desde editor de código	Si	No

Fuente: Autores

Elaboración: Autores.

Android Studio es el IDE (entorno de desarrollo) más actual y optimizado que se puede utilizar ofreciendo muchas funciones que facilitan el desarrollo de aplicaciones móviles en este sistema operativo.

2.6.2 Selección de motor OCR

Como se ha revisado en el capítulo anterior la búsqueda realizada, mostró que los dos motores para reconocimiento óptico de caracteres más utilizados son Google Visión y Tesseract OCR, estos poseen características similares como se visualiza en la tabla 2.2

Previo a la elección del motor se realizó una serie de pruebas, para analizar el comportamiento y funcionamiento del motor OCR, se tomaron en cuenta parámetros como tiempos de computo (tiempo de procesamiento desde la inserción de la imagen hasta la obtención del texto) y el nivel de precisión a la de la interpretación de los caracteres.

Tabla 2. 2 Comparativa Google Vision - Tesseract OCR

GOOGLE VISION	TESSERACT OCR
Multilinguaje, reconocimiento en tiempo real	Multilinguaje
Tres tipos de segmentación (bloques, líneas y palabras)	Tres tipos de segmentación (bloques, líneas y palabras)
Reconocimiento en tiempo real	Reconocimiento en tiempo real
Multiplataforma	Multiplataforma
Google	Hp
Uso de Machine Learnig (Aprendizaje automático) e inteligencia artificial (AI) para mejorar el procesamiento y reconocimiento de caracteres.	

Fuente: Autores

Elaboración: Autores.

El tipo de pruebas que se realizo fue en cuanto a extracción de texto en diferentes escenarios o condiciones, se detallan en la tabla 2.3 y pueden ser visualizadas en la figura 2.4.

Tabla 2. 3 Condiciones establecidas para selección de motor OCR

Condición	Descripción
Clara cercana	Distancia, 15 cm del objetivo, con buenas condiciones luminosas.
Clara lejana	Dilatancia mayor a 15 cm del objetivo, con buenas condiciones luminosas.

Obscura cercana	Dilatancia, 15 cm del objetivo, con condiciones luminosas deficientes.
Obscura lejana	Distancia mayor a 15 cm del objetivo con condiciones luminosas deficientes.
Clara cercana con inclinación	Distancia, 15 cm del objetivo, con buenas condiciones luminosas y con distinta perspectiva (ángulo de inclinación).
Clara lejana con inclinación	Distancia mayor a 15 cm del objetivo, con buenas condiciones luminosas y con distinta perspectiva (ángulo de inclinación).
Obscura cercana con inclinación	Distancia, 15 cm del objetivo, con condiciones luminosas deficientes y con distinta perspectiva (ángulo de inclinación).
Obscura lejana con inclinación	Distancia mayor a 15 cm del objetivo, con condiciones luminosas deficientes y con distinta perspectiva (ángulo de inclinación).

Fuente: Autores

Elaboración: Autores.

En un inicio se realizaron pruebas en imágenes que únicamente contenían bloques de texto con distintas tipografías y tamaños de letra (figura 2.5), los resultados de estas pruebas se pueden visualizar en las tablas 2.4 y 2.5.

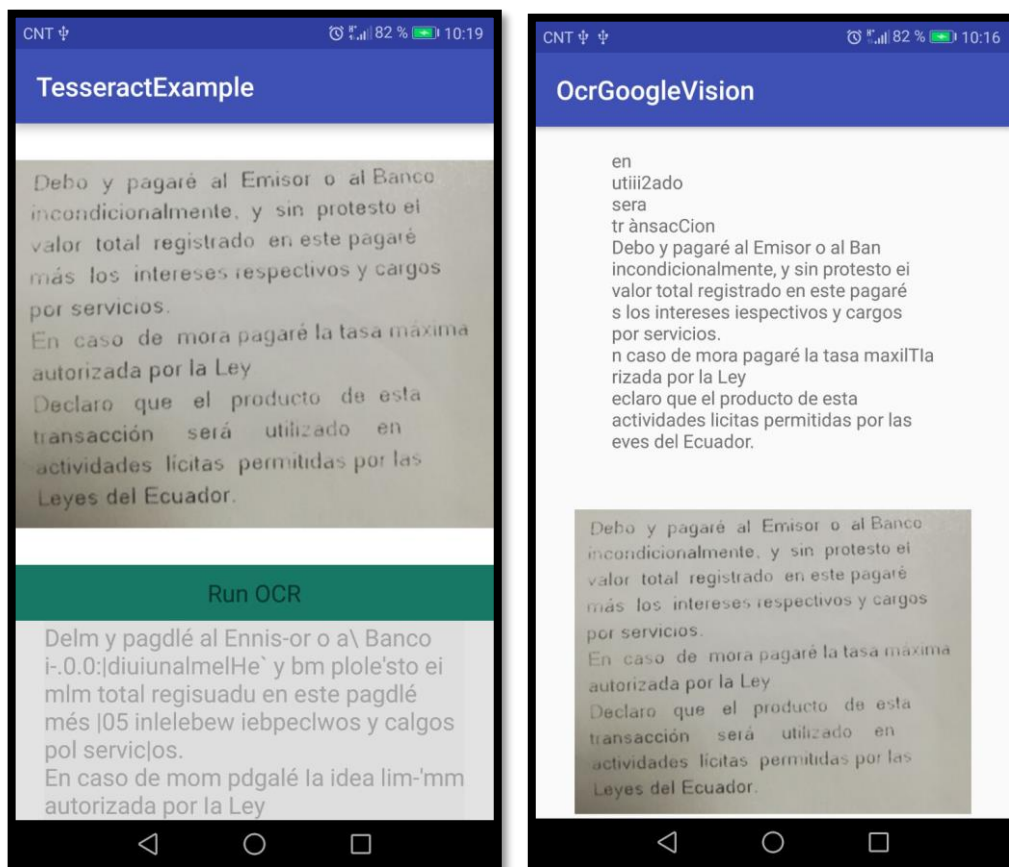


Figura 2. 4 Procesado de imágenes para la selección de motor OCR a) Tesseract OCR , b) Google Mobile Vision

Fuente: Autores

Elaboración: Autores.

El número total de muestras analizadas fue de 96, 2 imágenes por tipografía en todas las condiciones vistas en la tabla 2.4 y 2.5.

La métrica es el porcentaje de error [2], este es la correlación que existente entre el texto original y el texto digitalizado por los motores seleccionados.

$$\% \text{ Error} = \frac{\text{Caracteres}_{\text{totales}} - \text{Caracteres}_{\text{acertados}}}{\text{Caracteres}_{\text{totales}}} \quad [2]$$

Tabla 2. 4 Porcentajes de caracteres errados Google Mobile Vision OCR

GOOGLE MOBILE VISION						
	Tipografía 1	Tipografía 2	Tipografía 3	Tipografía 4	Tipografía 5	Tipografía 6
Clara cerca	1,09%	0,73%	0,56%	0,72%	2,40%	7,09%
Clara lejos	1,09%	0,24%	1,12%	0,24%	1,80%	9,45%
Ángulo claro cerca	3,20%	41,84%	1,12%	20,92%	1,79%	11,02%
Ángulo claro lejos	3,80%	19,70%	0,60%	62,28%	0,60%	54,33%
Obscuro cerca	6,52%	1,45%	0,00%	14,40%	5,98%	23,02%
Obscuro lejos	2,71%	0,72%	0,00%	16,78%	1,79%	44,88%
Ángulo obscuro cerca	3,80%	2,18%	0%	24,57%	2,94%	37,79%
Ángulo obscuro lejos	3,26%	2,18%	1,12%	24,57%	2,94%	37,79%

Fuente: Autores

Elaboración: Autores.

Tabla 2. 5 Porcentajes de caracteres errados Tesseract OCR

TESSERACT						
	Tipografía 1	Tipografía 2	Tipografía 3	Tipografía 4	Tipografía 5	Tipografía 6
Clara cerca	82,60%	7,31%	0,56%	0,72%	76,04%	92,12%
Clara lejos	1,08%	5,83%	1,12%	47,93%	42,51%	37,00%
Ángulo claro cerca	35,65%	4,98%	78,86%	16,02%	12,43%	91,15%
Ángulo claro lejos	55,43%	19,70%	0,56%	62,28%	0,59%	54,33%
Obscuro cerca	31,52%	3,16%	77,52%	14,35%	5,98%	90,55%
Obscuro lejos	55,43%	24,57%	20,22%	0,97%	10,17%	70,86%
Ángulo obscuro cerca	46,73%	3,16%	4,46%	26,76%	32,93%	99,21%
Ángulo obscuro lejos	57,02%	20,35%	23,30%	1,22%	16,32%	75,90%

Fuente: Autores

Elaboración: Autores.

Al analizar los resultados, en casi todos los escenarios, el motor con menor margen de caracteres errados es Google Mobile Vision.

Pero en cualquiera de los dos casos los márgenes de error son elevados, razón por la cual se cree conveniente utilizar alguna técnica de mejoramiento de imagen (Pre-procesado) antes de ingresar la imagen a motor. Como el módulo OCR propuesto está enfocado a extraer información de facturas se procedió a realizar las mismas pruebas en facturas, se analizó un total de 96 muestras con y sin Pre-procesado.

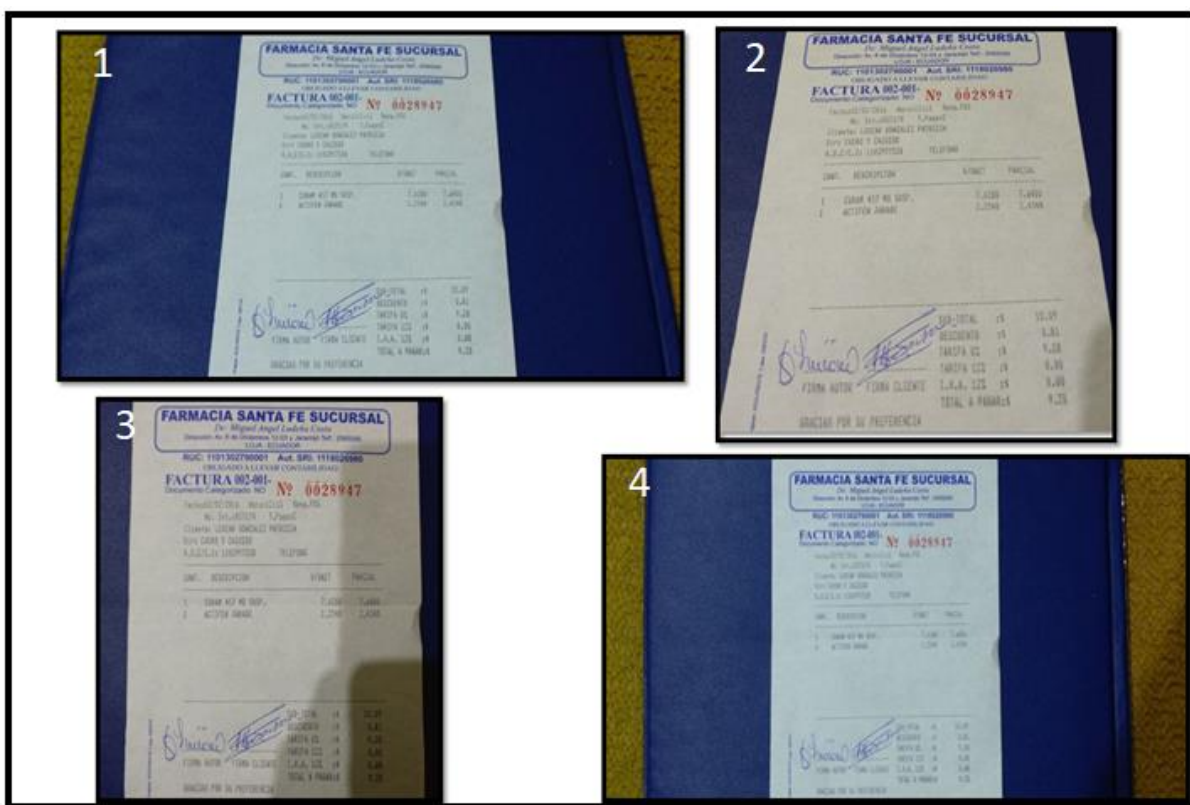


Figura 2. 5 Condiciones analizadas para la selección de motor OCR (Facturas); 1, Angulo clara cerca; 2, Angulo obscura cerca; 3, Obscura cerca; 4, Clara lejos

Fuente: Autores

Elaboración: Autores.

En la tabla 2.6 se presentan los resultados de la comparativa realizada entre las dos librerías al aplicar un pre-procesado (mejoramiento de la imagen) en las facturas.

Tabla 2. 6 Resultados Obtenidos por motores OCR

EVENTO	%ERROR GOOGLE VISION	%ERROR GOOGLE VISION PRE- PROCESADO	%ERROR TESSERACT	%ERROR TESSERACT PRE- PROCESADO
Clara cercana	26,21%	3,54%	85,37%	76,44%
Clara lejana	71,53%	48,92%	63,39%	60,81%
Clara claro en ángulo	6,51%	4,86%	71,52%	66,05%

Clara lejana en ángulo	73,51%	67,04%	67,80%	63,20%
Oscura cercana	9,57%	4,78%	53,93%	77,85%
Oscura lejana	25,86%	22,79%	58,55%	56,57%
Oscura cercana en ángulo	9,10%	5,05%	87,61%	52,41%
Oscura lejana en ángulo	7,69%	6,61%	85,77%	81,73%

Fuente: Autores

Elaboración: Autores.

Para evaluar los tiempos de procesamiento, se realizaron pruebas en tres dispositivos destinos: un teléfono gama media, gama alta y el emulador nativo de Android Studio sus características se pueden visualizar en la tabla 2.8, se debe tener en cuenta que estos tiempos varían dependiendo del procesador del equipo. Los resultados de este análisis se presentan en la tabla 2.7, se puede observar que Google Mobile Vision es hasta 5 veces más rápido que Tesseract.

Tabla 2. 7 Resultados por motores OCR

Tiempo de procesamiento	Google Mobile Vision	Tesseract OCR
Gama alta (OnePlus 3T)	759 ms	6444 ms
Gama Media (Huawei P9 Lite)	1567ms	8345 ms
SDK Android Emulator	2147 ms	5290 ms

Fuente: Autores

Elaboración: Autores.

Tabla 2. 8 Características de equipos utilizados en las pruebas

Equipo	RAM	ROM	Procesador
Gama alta (OnePlus 3T)	6 GB	64 GB	8 núcleos, 2.4 GHz
Gama Media (Huawei P9 Lite)	2 GB	16 GB	8 núcleos, 1.7GHz
SDK Android Emulator	2 GB	8 GB	En relación al ordenador

Fuente: Autores

Elaboración: Autores.

Con estos resultados se realizó una tabla de ponderación para cuantificar que tan buenos son estos motores OCR, el resultado del mismo se lo puede visualizar en el Anexo B, con estos resultados el motor OCR que mejor se adecúa a nuestras necesidades es el de Google Vision este tiene una mayor precisión y es de fácil implementación.

2.7 Análisis de Requisitos

Se establecieron algunos requisitos para el correcto desarrollo del aplicativo móvil, dentro de estos se encuentran los requisitos funcionales y no funcionales estos se establecen al iniciar del diseño.

2.7.1 Requisitos no funcionales

Van relacionados con la arquitectura del sistema a desarrollar y con las características como el lenguaje de programación en el que se va a trabajar, el tiempo de respuesta, la disponibilidad, etc.

La tabla 2.9 muestra los requisitos no funcionales que se tomaron en cuenta para el desarrollo del mismo.

Tabla 2. 9 Requisitos no funcionales

ATRIBUTO	DETALLE
Plataforma:	Android 4.0.1+
Tipo de aplicación:	Aplicación Móvil
Lenguaje de programación:	Java y XML
Gestor de base de datos:	Firebase
Herramienta de desarrollo:	Android Studio
Tiempo de respuesta:	Menor a 7 segundos para cualquier operación excepto extracción por OCR ahí se da un margen de 10 segundos de respuesta.
Disponibilidad:	Todo el tiempo (conexión a internet)
Metáfora de interfaz:	Orientada a formularios

Fuente: Autores

Elaboración: Autores.

2.7.2 Requisitos funcionales

Los requisitos funcionales son servicios que obligatoriamente debe tener el sistema a desarrollar, los servicios que requiere el sistema propuesto, estos se muestran en la tabla 2.10. Se ha catalogado por el tipo de interacción que tendrán con el usuario, oculta si la participación del usuario es nula y evidente si la participación del usuario es necesaria.

Tabla 2. 10 Requisitos funcionales

Ref	Función	Categoría
R1	Registrar nuevo usuario	Evidente
R2	Login usuario	Oculto
R3	Registrar comprobante manualmente	Evidente
R4	Registrar comprobante QR	Evidente
R5	Registrar comprobante OCR	Evidente
R6	Envío de información a base de datos	Oculto
R7	Visualización de estado de declaración	Oculto
R8	Generación de anexo en formato .xls	Oculto
R9	Cerrar sesión	Evidente

Fuente: Autores

Elaboración: Autores.

2.7.3 Criterios de aceptación

Son aspectos mínimos que debe cumplir cada uno de los requisitos funcionales para que se lo considere como funcional, pero si se sobrepasa estos mínimos no importaría porque se obtendría mayor satisfacción al usuario final.

En la tabla 2.11 se puede observar los criterios de aceptación para el aplicativo desarrollado.

Tabla 2. 11 Criterios de aceptación

Requisito funcional	Criterio de aceptación
Registrar nuevo usuario	Debe permitir el registro de un nuevo usuario a la base de datos, este previo al registro ingresa sus datos personales y credenciales como usuario y contraseña que son las que le permitirán su posterior acceso a la aplicación.
Login	Debe permitir al usuario ingresar sus credenciales, realizar un autenticado, si es exitoso permite el acceso al sistema y si no, da opción a restablecer la contraseña o registrarse si este no lo ha hecho.
Registrar comprobante manualmente	Debe permitir ingresar la información de comprobantes en forma de formularios y revisar cada uno de los campos para evitar que se suba información errónea
Registrar comprobante QR	Debe utilizarse la cámara y un decodificador de códigos QR integrado en la aplicación para decodificar el código, además de notificar al usuario que su código es invalido o si la factura se guardó exitosamente.
Registrar comprobante OCR	Implementar un motor QR para la digitalización de texto de comprobantes y a través del mismo llenar un formulario.

	Dar al usuario la posibilidad de corregir en caso de que la digitalización no sea exitosa
Envío de información a base de datos	Los datos recopilados por los métodos de adquisición de datos de comprobantes deben permitir subir esta información a una base de datos directamente desde cada módulo.
Visualización de estado de declaración	El usuario en todo momento puede revisar los montos actuales por rubro.
Generación de anexo en formato .xls	El anexo generado estará en formato .xls formato aceptado por la plataforma del SRI (Servicio de Rentas Internas).
Cerrar sesión	En cualquier momento el usuario puede salir de su cuenta. Sus datos no se verán comprometidos

Fuente: Autores

Elaboración: Autores.

2.7.4 Plantilla combinada

La plantilla combinada muestra los atributos del sistema con los requisitos funcionales y a su vez se especifica algunas características. En la tabla 2.12 se puede observar la plantilla combinada del sistema.

Tabla 2. 12 Plantilla combinada de la aplicación

REF.	FUNCIÓN	CATEGORÍA	ATRIBUTO	DETALLE Y RESTRICCIÓN	CATEGORÍA
R1	Registrar usuario	Evidente	Tiempo de respuesta	3 segundos	Obligatorio
			Tolerancia a fallos	Entregar aviso si un dato ingresado es erróneo	
			Metáfora de interfaz	Basada en formulario	
R2	Login	Oculto	Tiempo de respuesta	1.5 segundos aproximadamente	Obligatorio
R3	Registrar comprobante manualmente	Evidente	Tiempo de respuesta	3 segundos	Obligatorio
			Tolerancia a fallos	Entregar aviso si un dato ingresado es erróneo	
			Metáfora de interfaz	Basada en formulario	
R4	Registrar comprobante QR	Evidente	Tiempo de respuesta	3 segundos	Obligatorio
			Tolerancia a fallos	Entregar aviso si QR es correcto	
			Metáfora de interfaz	Basada uso de cámara del dispositivo	
R5		Evidente	Tiempo de respuesta	Hasta 10 segundos	Obligatorio

	Registrar comprobante OCR		Tolerancia a fallos	Entregar aviso si la información es errónea	
			Metáfora de interfaz	Basada en el uso de un selector	
R6	Envío de información a base de datos	Oculto	Tiempo de respuesta	3 segundos	Obligatorio
R7	Visualización de estado de declaración	Oculto	Tiempo de respuesta	3 segundos	Obligatorio
			Metáfora de interfaz	Basada en gráfico pastel	
R8	Generación de anexo en formato .xls	Oculto	Tiempo de respuesta	3 segundos	Obligatorio
R9	Cerrar sesión	Oculto	Tiempo de respuesta	3 segundos	Obligatorio

Fuente: Autores

Elaboración: Autores.

2.7.5 Identificación de actores.

Como el aplicativo se lo desarrolla para ayudar a personas que necesiten realizar su declaración de impuesto a la renta se ha establecido como actor principal a un usuario final único que reemplace a todas estas personas, se lo describe en la tabla 2.13.

Tabla 2. 13 Actores del sistema

Actor	Usuario final
Descripción	Persona que necesite la aplicación para realizar su anexo de impuesto a la renta con los comprobantes no electrónicos.
Características:	El usuario podrá tener uso ilimitado de la aplicación contenida en el dispositivo móvil. Accediendo a todos los servicios ofrecidos por la app, tales como ingresar sus comprobantes con cualquiera de los tres métodos de adquisición, revisar el estado de cuenta y poder generar el anexo correspondiente al año fiscal en curso.

Fuente: Autores

Elaboración: Autores.

2.7.6 Modelos de caso de Uso

Una vez identificado el actor principal del sistema se debe especificar en qué caso el “Usuario Final” podrá hacer uso del sistema. Así en la figura 2.6 se muestran los que se consideró más importantes.

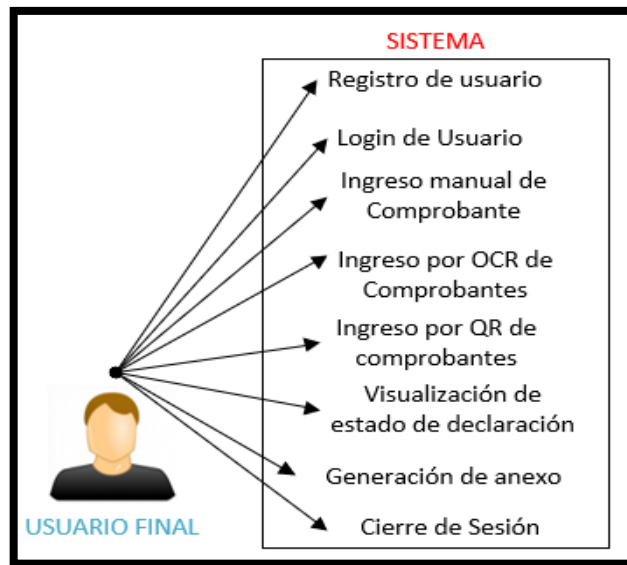


Figura 2. 6 Modelos de caso de uso
Fuente: Autores
Elaboración: Autores.

2.7.6.1 Especificaciones de casos de Uso

Para especificar cada uno de los el usuario debe contar con conexión a Internet cuando vaya a realizar cada una de las siguientes actividades estas se detallan a continuación.

En la tabla 2.14 se detalla el registro de un nuevo usuario al sistema.

Tabla 2. 14 Caso de uso de registro de nuevos usuarios.

Caso de Uso		Registro de usuario.
ID	1	
Descripción	El usuario ingresa sus datos y accede a la base de datos (Firebase Database) de la aplicación.	
Actores del sistema	Usuario.	
Actores secundarios	No tiene.	
Precondiciones	Ninguna.	
Flujos principales	<p>El usuario ingresa sus datos personales tales como email, contraseña y cedula de identidad.</p> <p>El sistema verifica que los datos ingresados son correctos.</p> <p>Si los datos son correctos el sistema envía estos datos en forma de credenciales donde los firma.</p> <p>Si los datos son erróneos el sistema envía una alerta para que el usuario sea notificado.</p>	
Post-condiciones	Ingreso a la aplicación.	

Fuente: Autores
Elaboración: Autores.

En la tabla 2.15 se puede observar el caso en el que el usuario va a ingresar a la interfaz principal para eso debe hacer uso de las credenciales que ingresó en el registro.

Tabla 2. 15 Caso de uso login y cierre de sesión de usuarios.

Caso de Uso	Login de usuario y cierre de sesión.
ID	2 y 8
Descripción	El usuario ingresa sus datos y accede al menú principal de la aplicación (Login). El usuario una vez culminada la actividad presiona el botón para salir del sistema (Cierre de sesión).
Actores del sistema	Usuario.
Actores secundarios	No tiene.
Precondiciones	Ninguna.
Flujos principales	Login: <ul style="list-style-type: none"> • El usuario ingresa sus datos requeridos. • El sistema verifica que los datos ingresados se encuentren autenticados. • Si los datos son correctos el sistema da acceso a la pantalla principal. • Si los datos son erróneos el sistema envía una alerta para que el usuario sea notificado y corrija. • Si el Usuario olvido sus credenciales puede pedir se le envíen al correo. Cierre de sesión: <ul style="list-style-type: none"> • El usuario debe tener una sesión activa dentro de la aplicación. • Al presionar el botón de cierre de sesión el usuario debe confirmar su salida.
Post-condiciones	Ingreso a la aplicación y haberse registrado previamente

Fuente: Autores

Elaboración: Autores.

En la tabla 2.16 se detalla el ingreso de comprobantes por el método manual haciendo uso un formulario.

Tabla 2. 16 Caso de uso ingreso manual de comprobantes.

Caso de Uso	Ingreso manual de comprobantes
ID	3
Descripción	El usuario ingresa sus datos de sus comprobantes mediante el uso de un formulario.
Actores del sistema	Usuario.
Actores secundarios	No tiene.
Precondiciones	Ninguna.

Flujos principales	El usuario selecciona del menú principal el apartado de ingreso manual. El sistema arroja el formulario correspondiente al ingreso manual. Si los datos son correctos el sistema da acceso a poder cargar los a la base de datos Si los datos son erróneos el sistema envía una alerta para que el usuario sea corrija.
Post-condiciones	Ingreso a la aplicación y contar con comprobantes

Fuente: Autores

Elaboración: Autores.

En la tabla 2.17 se detalla el ingreso de comprobantes por el método QR haciendo de un decodificador QR implementado en la misma aplicación.

Tabla 2. 17 Caso de uso ingreso por QR de comprobantes

Caso de Uso	Ingreso por QR de comprobantes
ID	4
Descripción	El usuario ingresa sus datos de sus comprobantes mediante de un decodificador de códigos QR implementado en la aplicación.
Actores del sistema	Usuario.
Actores secundarios	No tiene.
Precondiciones	Ninguna.
Flujos principales	El usuario selecciona del menú principal el apartado de ingreso por QR. El sistema lanza el decodificador. Si código QR es correctos el sistema cargar los a la base de datos. Si los datos son erróneos el sistema envía una alerta para que el usuario escanee un código correcto.
Post-condiciones	Ingreso a la aplicación y contar con códigos QE correctos

Fuente: Autores

Elaboración: Autores.

En la tabla 2.18 se detalla el ingreso de comprobantes por el método OCR que es el método más importante, este se encarga de la digitalización de la información presente en los comprobantes y se lo estructura dentro de un formulario.

Tabla 2. 18 Caso de uso ingreso por OCR de comprobantes.

Caso de Uso	Ingreso por OCR de comprobantes
ID	5
Descripción	El usuario ingresa sus datos de sus comprobantes mediante el uso de la cámara o una fotografía de la galería.
Actores del sistema	Usuario.
Actores secundarios	No tiene.
Precondiciones	Ninguna.
Flujos principales	El usuario selecciona del menú principal el apartado de ingreso por OCR.

	El sistema ejecuta para que el usuario seleccione si desea procesar una imagen que ya esté en la galería o una foto nueva.
Post-condiciones	Ingreso a la aplicación.

Fuente: Autores
Elaboración: Autores.

En la tabla 2.19 se detalla el caso en el que un usuario quiere visualizar la información dentro de la aplicación, es desde aquí que se visualiza información de cada rubro y se puede generar el anexo en formato aceptado por el SRI.

Tabla 2. 19 Caso de uso de Visualización de estado de cuenta y generación de anexo.

Caso de Uso	Visualización de estado de cuenta y generación de anexo.
ID	6 y 7
Descripción	El usuario puede observar como esta su declaración por rubros a través de cuadros porcentuales
Actores del sistema	Usuario.
Actores secundarios	No tiene.
Precondiciones	Ninguna.
Flujos principales	El usuario selecciona del menú principal el apartado de Mi perfil El sistema ejecuta el apartado correspondiente trayendo los datos de la Firebase y representando en un diagrama de pastel. El usuario puede extraer de los datos actuales el anexo correspondiente en el formato aceptado por el SRI
Post-condiciones	Ingreso a la aplicación.

Fuente: Autores
Elaboración: Autores.

2.7.7 Diagrama de navegación

Una vez que hemos definidos las características de la aplicación y donde va a participar el usuario debemos tener un mapa de navegación es decir a donde nos llevará cada botón del aplicativo, como se puede observar en la figura 2.7 esta es la secuencia que sigue la aplicación.

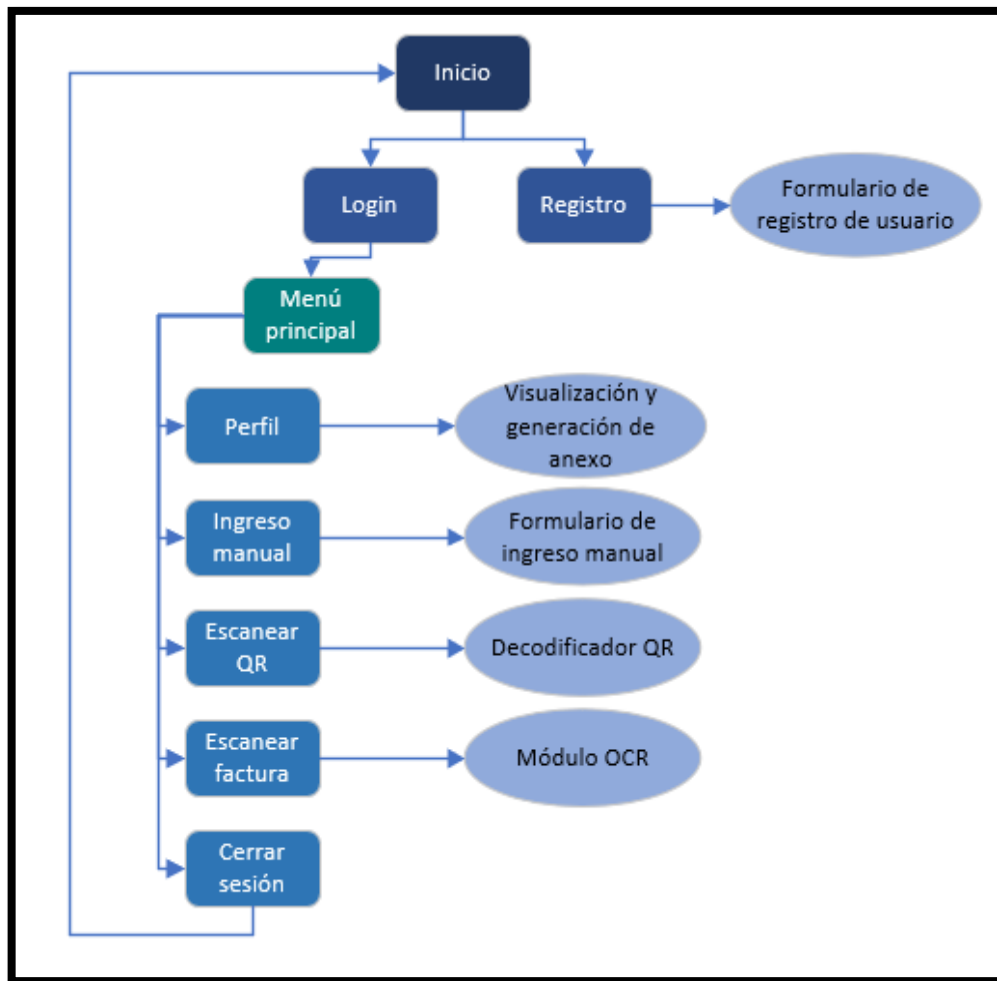


Figura 2. 7 Mapa de navegación

Fuente: Autores

Elaboración: Autores.

2.7.8 Descripción del mapa de navegación

Login: parte del aplicativo en la que usuario debe ingresar sus credenciales para ser verificado y darle acceso a los demás servicios con los que cuenta la misma.

Registro: el usuario ingresa sus datos para ser validados y obtener un espacio dentro de la base de datos.

Menú principal: se despliegan todos los servicios ofrecidos.

Perfil: parte del aplicativo que muestra los montos guardados en los diferentes rubros haciendo uso de gráficos estadísticos además brinda la posibilidad de generar el anexo con las facturas declaradas hasta el momento.

Ingreso Manual: apartado donde se ingresa la información de los comprobantes, haciendo uso de formularios. Una vez que esté lleno correctamente se lo puede guardar dentro de la base de datos.

Escáner QR: esta parte del aplicativo lleva integrado un decodificador QR que ayuda a extraer la información de los comprobantes de manera más rápida, una vez decodificado el código QR, es validado y se guarda automáticamente la información a la base de datos.

Escanear Facturas: permite el ingreso al módulo OCR creado para digitalización y extracción del texto de los comprobantes, una vez realizado este proceso se despliega un formulario para validación y corrección de la información extraída de la imagen luego se puede adjuntar la información a la base de datos.

Cerrar Sesión: finaliza la sesión de un usuario y cierra la conexión con la base de datos.

CAPÍTULO III
IMPLEMENTACIÓN

En este capítulo se detalla el proceso de implementación e integración en la aplicación de cada uno de los módulos expuestos en el capítulo anterior.

3.1 Módulo de autenticación y registro de usuarios

El módulo se implementó haciendo uso de dos servicios que ofrece Firebase, estos son Firebase Database y Firebase Authentication. Estos están enfocados a la creación y acceso de los usuarios. El diagrama de la figura 3.1 presenta un esquema básico de funcionamiento de este módulo y en figura 3.2 se muestra su interface.

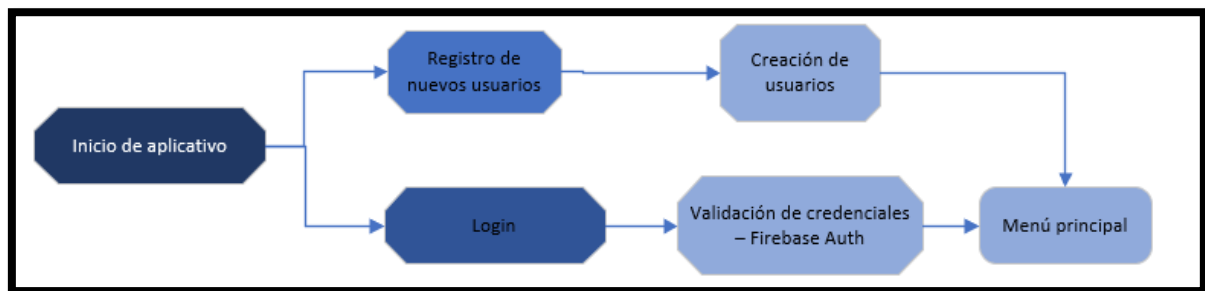


Figura 3. 1 Diagrama de funcionamiento módulo de autenticación y registro de usuarios

Fuente: Autores

Elaboración: Autores.

El registro y acceso de usuarios se lo realiza con haciendo uso de los métodos “createUserWithEmailAndPassword”, “signInWithEmailAndPassword” respectivamente, una vez que el usuario complete el formulario de registro ingresando todos los datos requeridos, se crea un nuevo usuario con las credenciales de correo y contraseña, Firebase Authentication se encarga de devolver un ID de usuario, al mismo tiempo crea un directorio con este ID en Firebase Database con toda la información registrada (Nombre, Correo, y cedula de identidad).

En caso de que sea un inicio de sesión, el usuario ingresa sus credenciales Firebase Authentication devuelve un token de autenticación en el caso de este ser verdadero se establece la comunicación y se da acceso a los demás servicios caso contrario se niega la comunicación y se devuelve un objeto de error, la figura 3.2 resume el proceso de autenticación y registro.

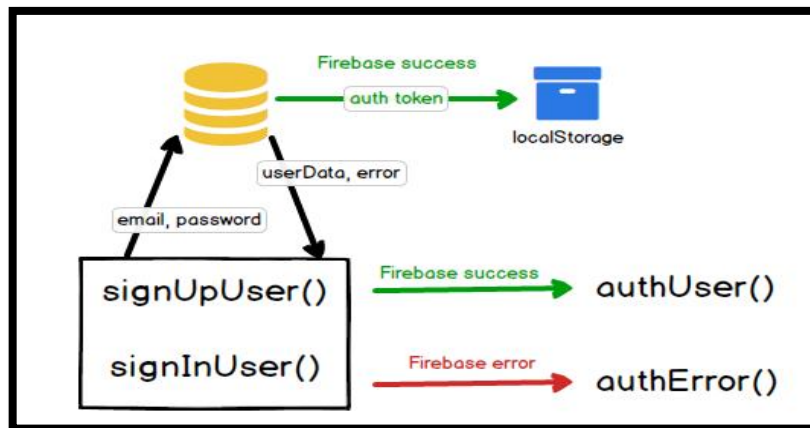


Figura 3. 2 Diagrama de funcionamiento módulo de autenticación y registro de usuarios
Fuente: Autores
Elaboración: Autores.

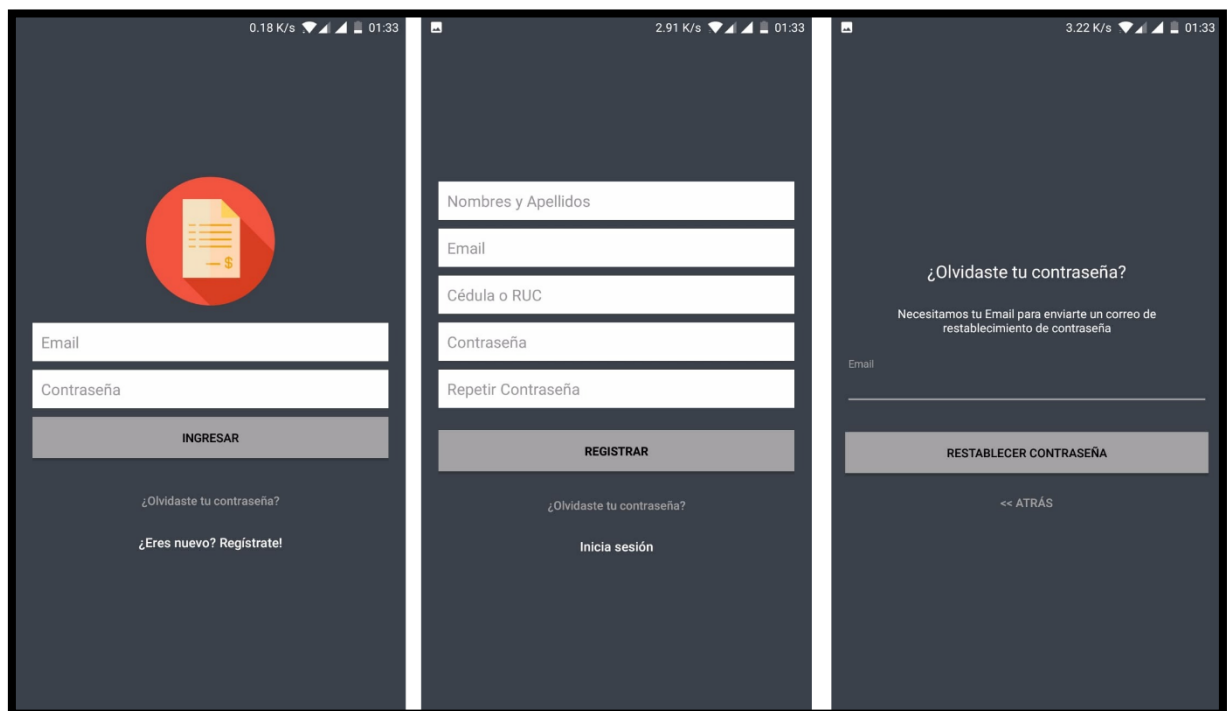


Figura 3. 3 Interface Login, Registro y recuperación de contraseña olvidada
Fuente: Autores
Elaboración: Autores.

3.2 Módulo de adquisición de datos

Como se explicó en la fase de diseño, este módulo está constituido por tres métodos de ingreso de datos (Manual, QR y OCR), estos se encuentran en comunicación directa con la base de datos, obteniendo primero una instancia (figura 3.4) "DatabaseReference" y una

referencia “getInstance().getReference()” para posterior lectura y escritura. La lectura y escritura de datos se realiza en formato JSON. Los tres métodos de ingreso previo a la escritura en la base de datos, realizan una consulta con el escuchador “.addValueEventListener” para verificar si es que la información que se desea enviar ya ha sido almacenada previamente, si no existe información se puede adjuntar sin inconvenientes la información de la nueva factura a la base de datos, si es el caso, se notificara al usuario para que revise la información o pueda actualizarla.

```
private DatabaseReference mDatabase;  
// ...  
mDatabase = FirebaseDatabase.getInstance().getReference();
```

Figura 3. 4 Obtención de instancia, Android Studio IDE
Fuente: Autores
Elaboración: Autores.

3.3 Método de adquisición de datos manual

Consta de un formulario el cual recoge toda la información y la envía a la base de datos. El diagrama básico de funcionamiento de este método de adquisición se presenta en la figura 3.5. y si interface gráfica se muestra en la figura 3.6

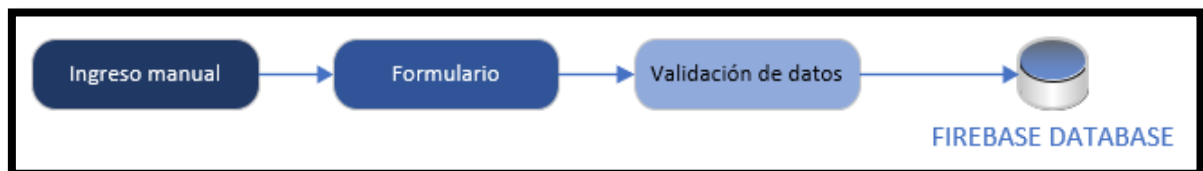


Figura 3. 5 Diagrama de funcionamiento método de adquisición de datos manual
Fuente: Autores
Elaboración: Autores.

Haciendo uso del método “child” se obtiene la ubicación del directorio en la base de datos y con “setValue” se puede enviar toda la información. El código también realiza una validación campo a campo previa al envío.

3.99 K/s 13:08

Nombre: Pedro Ivan Sanchez Portoc

Factura N°: _____

Fecha dd/mm/aa: _____

CI/RUC Cliente: 1105904567

Comercial: _____

CI/RUC Comercial: _____

Subtotal: _____

Salud: _____

Vestimenta: _____

Educación: _____

Alimentación: _____

Vivienda: _____

Varios: _____

Total \$: 0.00

CARGAR DATOS

Figura 3. 6 Formulario para ingreso de comprobantes de forma manual
Fuente: Autores
Elaboración: Autores.

3.4 Método de adquisición de datos por QR

En este método de adquisición, se desarrolló un decodificador QR haciendo uso de la librería Google Mobile Vision y el uso de la clase “BarcodeDetector”. Es esquema básico de funcionamiento se lo presenta en a figura 3.7.

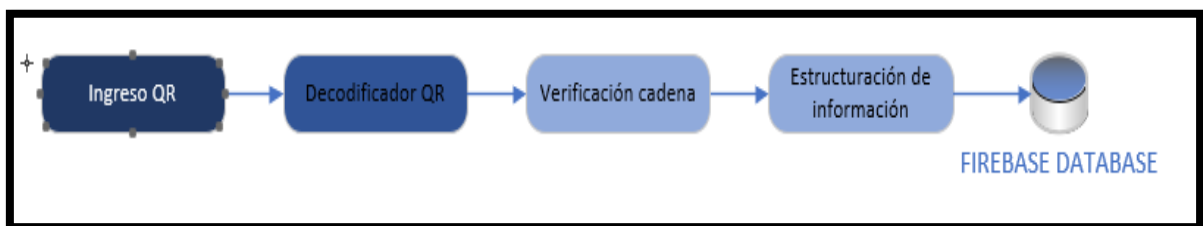


Figura 3. 7 Esquema de funcionamiento módulo de adquisición de datos por QR
Fuente: Autores
Elaboración: Autores.

Una vez decodificado y estructurado el código se usa la función “split” para separar la información y asignar a los campos respectivos, finalmente se envía todo a la base de datos.

```
BarcodeDetector detector = new BarcodeDetector.Builder(context)
    .setBarcodeFormats(Barcode.DATA_MATRIX | Barcode.QR_CODE)
    .build();
```

Figura 3. 8 Ejemplo decodificador QR y data Matrix

Fuente: Autores

Elaboración: Autores.

La figura 3.9 presenta la interfaz gráfica de este módulo de adquisición.

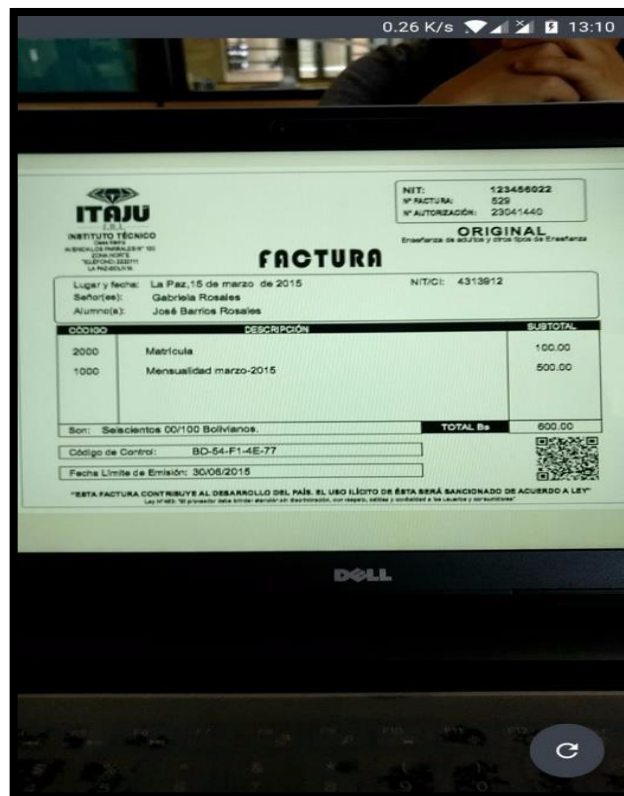


Figura 3. 9 Interfaz módulo de adquisición por código QR

Fuente: Autores

Elaboración: Autores.

3.5 Método de adquisición de datos por OCR

Este es el apartado más complejo del desarrollo de la aplicación. La figura 3.10 presenta un esquema básico de funcionamiento.

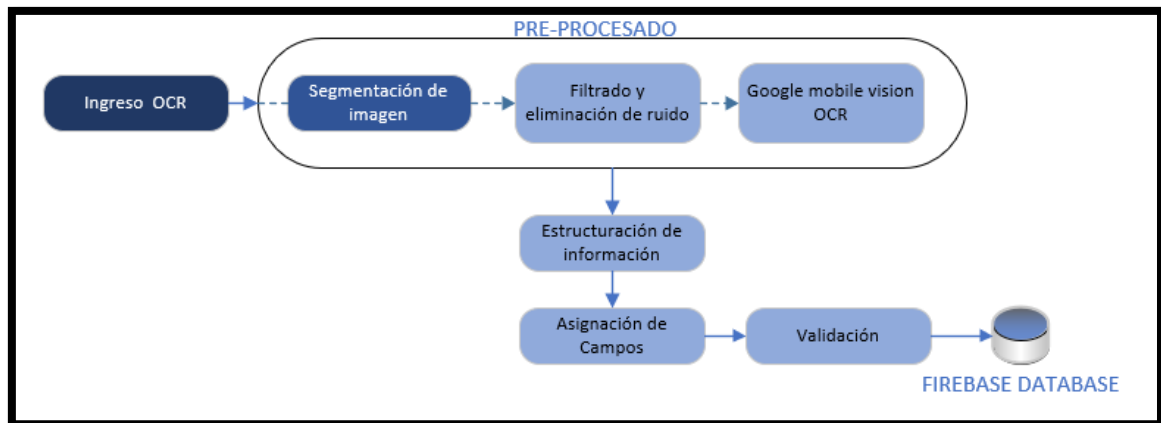


Figura 3. 10 Esquema de funcionamiento módulo de adquisición de datos por QR

Fuente: Autores

Elaboración: Autores.

Se estructuró de la siguiente manera: primero se seleccionó las facturas en las que se va a trabajar. El tipo de facturas son las impresas con el sistema VISUAL FACT, este sistema está presente en la mayoría de comercios y su estructura (tipografía, tamaño de letra, espaciado, etc.) es la misma en cualquiera de sus versiones.

Se decidió segmentar la factura en dos partes: encabezado y resumen de gastos como se presenta en la figura 3.11, de esta forma, lo que se hace es, reducir el área de mapeo y eliminar zonas que no contengan información requerida.

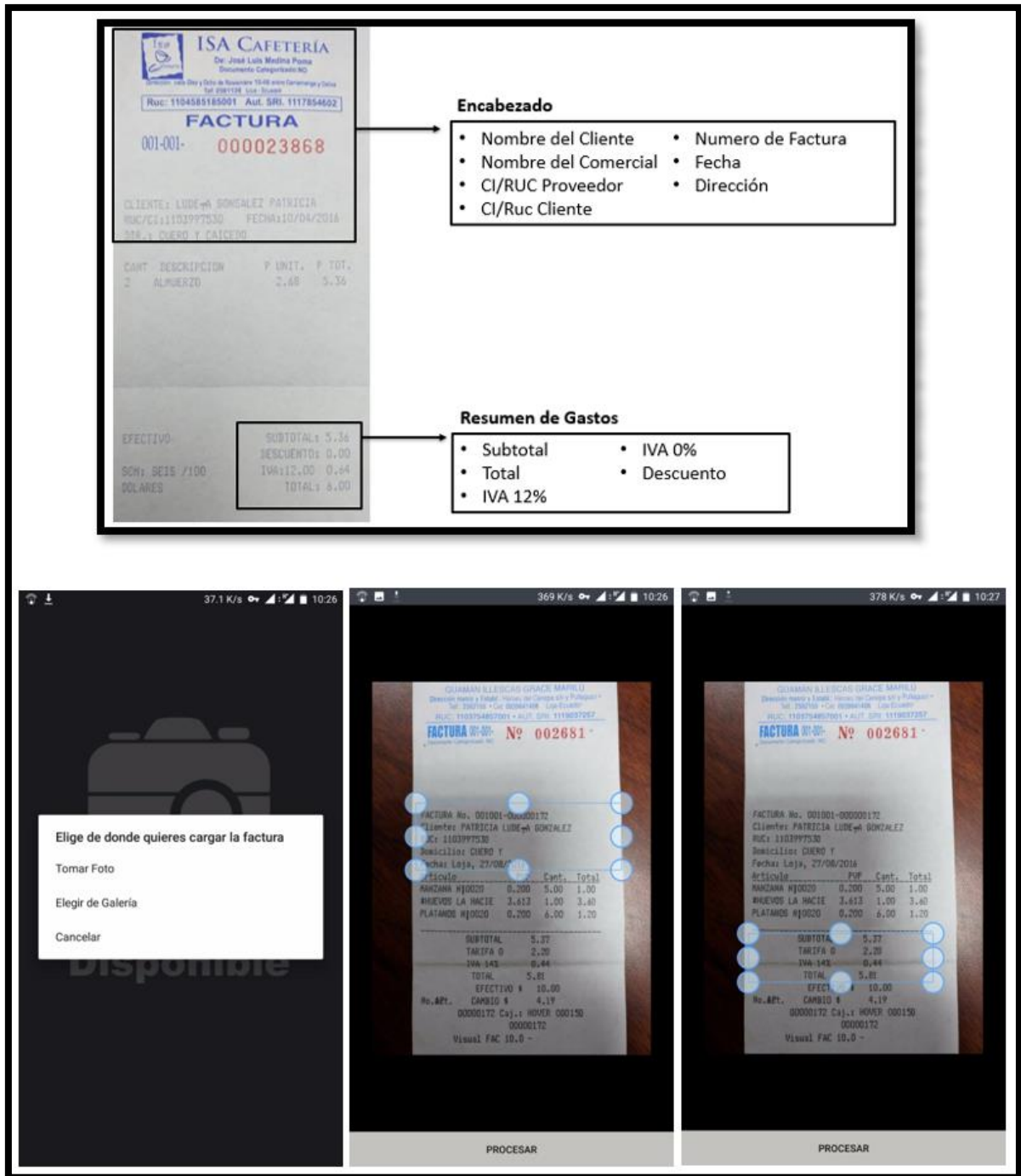


Figura 3. 11 Segmentación previa a pre-procesado
Fuente: Autores
Elaboración: Autores.

En base al análisis de rendimiento de motores OCR, se optó por realizar un pre-procesado para mejorar la calidad de la imagen antes de realizar el reconocimiento con Google Mobile Vision.

3.6 Pre-Procesado

Este apartado habla sobre el tratamiento hacia la imagen para mejorar la calidad de la misma, de esta forma incrementar en grado de precisión. El pre-procesado consiste en seleccionar el área de mapeo (eliminar áreas innecesarias en la imagen), aplicación de filtrado y eliminación de ruido (Capítulo I) y en caso de ser requerido corrección de perspectiva (enderezar la imagen si esta tiene un ángulo de inclinación). Al final se obtiene un mapa de datos (imagen binaria) esta se ingresa directamente al motor OCR, esto se facilitó haciendo uso de la librería scanLibrary. La figura 3.12 muestra la selección de área de mapeo y corrección de perspectiva.



Figura 3. 12 Corrección de perspectiva etapa Pre-procesado
Fuente: Autores
Elaboración: Autores.

Aplicando algunas técnicas vistas en el apartado 1.6.2.3 se mejora notablemente la condición de la imagen a procesar figura 3.13.

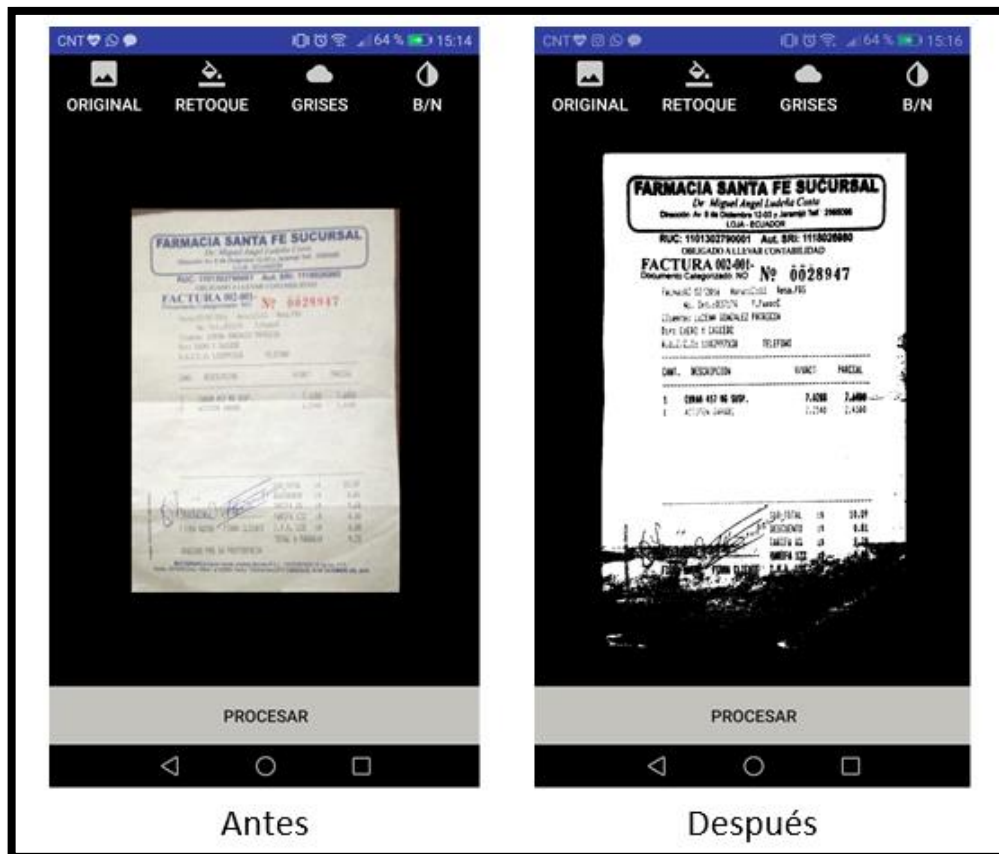


Figura 3. 13 Filtrado de la imagen etapa Pre-procesado

Fuente: Autores

Elaboración: Autores.

“TextRecognizer” este objeto detector procesa imágenes y determina el texto que aparece dentro de la misma, “TextBlock” permite realizar una segmentación por bloques de texto, esto se lo puede ver en la figura 3.14.

Una vez estructurada la cadena se realiza búsqueda de la información palabra por palabra, luego se asigna de información a los campos correspondientes, el usuario verifica la información que la información sea correcta y puede cargar la factura a la base de datos, figura 3.15.

```

Frame frame = new Frame.Builder().setBitmap(bitmap).build();
SparseArray<TextBlock> items = textRecognizer.detect(frame);
StringBuilder stringBuilder = new StringBuilder();

for (int i = 0; i < items.size(); i++) {
    TextBlock textBlock = items.valueAt(i);
    stringBuilder.append(textBlock.getValue());
    stringBuilder.append("\n");
}

```

Figura 3. 14 Fragmento de código TextRecognizer

Fuente: Autores

Elaboración: Autores.

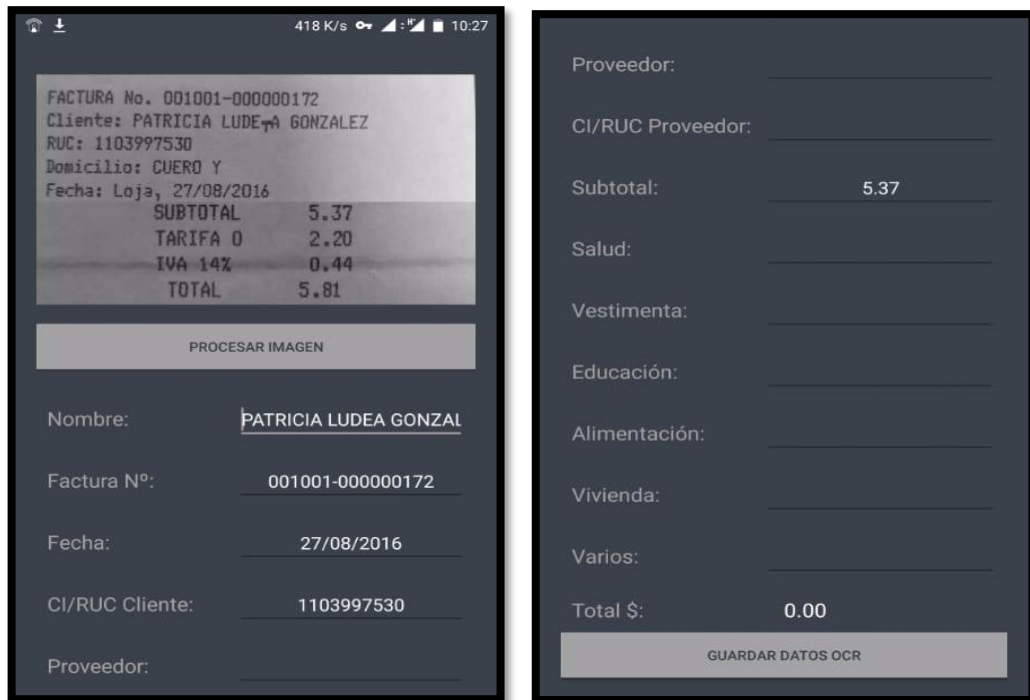


Figura 3. 15 Interfaz gráfica adquisición mediante OCR

Fuente: Autores

Elaboración: Autores.

3.7 Almacenamiento de la información

Todos los módulos de la aplicación se encuentran conectados con la base de datos. El esquema de estructura de la base de datos se presenta en la figura 3.16.

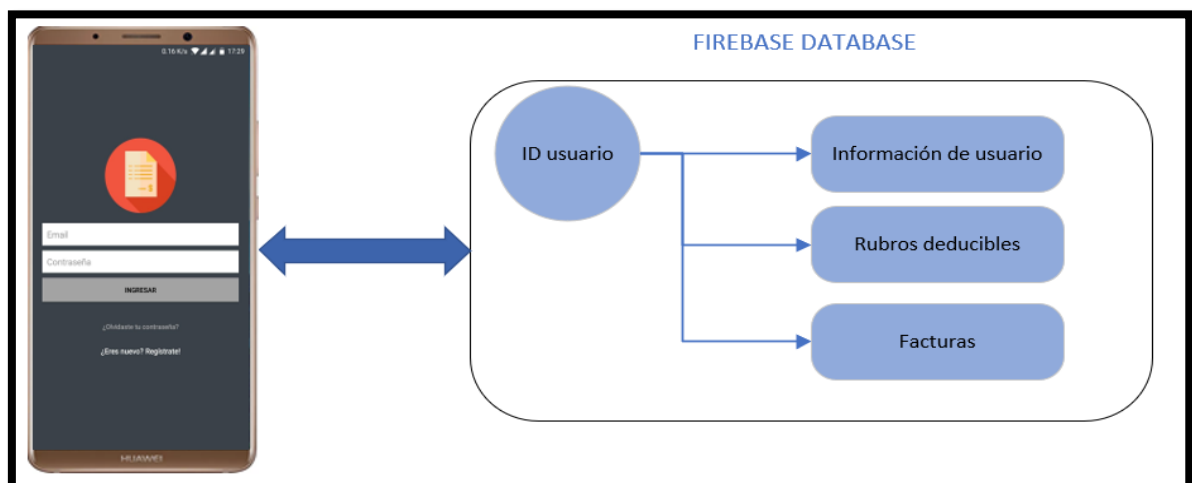


Figura 3. 16 Estructura de información dentro de la base de datos

Fuente: Autores

Elaboración: Autores.

Como ya se habló, la base de datos está estructurada haciendo uso del servicio Firebase Database, este permite la visualización de la información cargada en la misma en forma de árbol, dicho árbol puede tener hasta 32 niveles de profundidad.

Cada vez que registra un nuevo usuario se asigna un ramal o directorio dentro de la base de datos con el ID de usuario, este va a permitir que únicamente ese usuario tengo acceso a la información contenida dentro del directorio. Haciendo uso de las reglas de seguridad de Firebase Database se determina quién tiene acceso de lectura o escritura a la base de datos, cómo se estructuran los datos y qué índices existen (figura 3.17).

```

1  {
2    "rules": {
3      "Usuarios": {
4        "Suid": {
5          ".read": "$Suid === auth.uid",
6          ".write": "$Suid === auth.uid"
7        }
8      }
9    }
10  }

```

Figura 3. 17 Reglas Firebase Database

Fuente: Autores

Elaboración: Autores.

Dentro del directorio de usuario la información esta estructurada de esta forma: existe un directorio que contiene información general de usuario y otro con información de proveedor (las facturas estas separadas por proveedor y número de factura) como se puede ver en la figura 3.18 derecha.

La figura 3.18 muestra la estructura del árbol dentro de la base de datos.

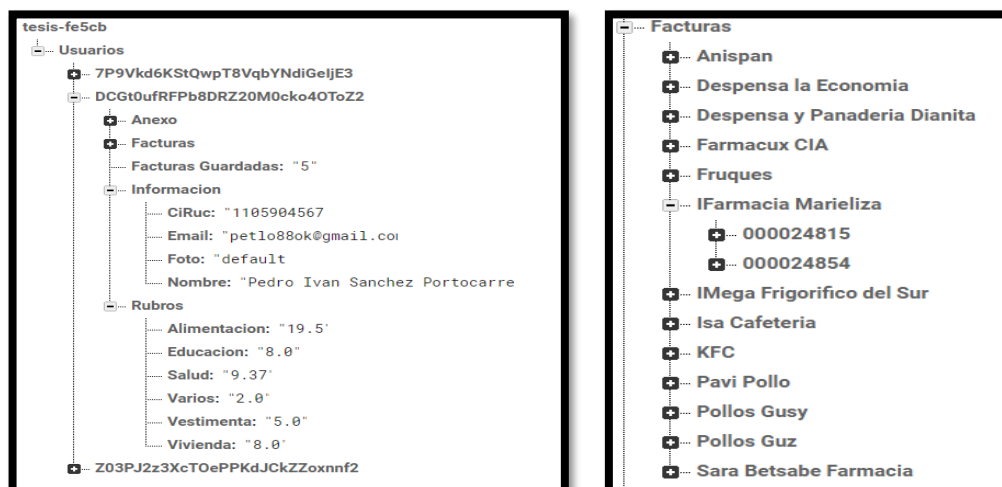


Figura 3. 18 Estructura de la base de datos en diagrama de árbol

Fuente: Autores

Elaboración: Autores.

Como se mencionó en la fase de diseño se implementaron cuatro servicios que ofrece Firebase estos son: Autenticación, Base de datos, Reporte de errores y Almacenamiento. Los dos últimos servicios son adicionales (no objeto del proyecto) y visibles únicamente al administrador. Para su implementación se agregaron las dependencias “com.google.firebase:firebase-crash:16.0.1” y “com.google.firebase:firebase-storage:16.0.1”.

3.8 Módulo de visualización y generación de resumen de gastos deducibles

Este presenta gráficos estadísticos de los montos actuales por rubros, permite visualizar el actual por rubro y el número de facturas almacenadas en a base de datos.

El módulo tiene una comunicación con la base de datos de solo lectura, detecta cada cambio en la misma en tiempo real. El esquema básico de funcionamiento de este módulo se presenta en la figura 3.19. La creación de los gráficos estadísticos se la realizo haciendo uso de la librería MPAndroidChart y a asignación de montos por proveedor al documento “xlm” mediante de JEXCEL API.

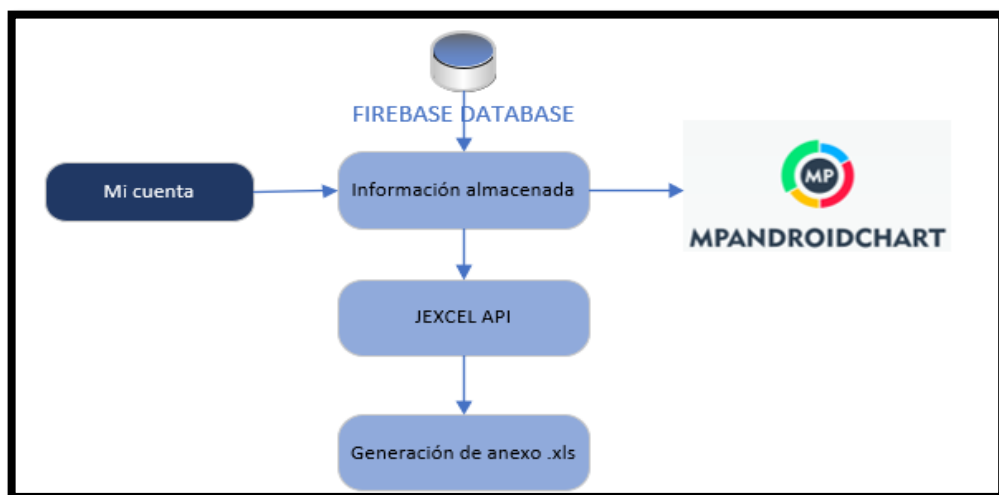


Figura 3. 19 Esquema de funcionamiento módulo de visualización y generación de anexo

Fuente: Autores

Elaboración: Autores.

La figura 3.20 muestra la interfaz de visualización de los montos actuales almacenados y la posibilidad de generar el resumen de gastos deducibles.



Figura 3. 20 Interfaz gráfica módulo de visualización y generación de anexo
Fuente: Autores
Elaboración: Autores.

CAPÍTULO IV
PRUEBAS Y VALIDACIÓN

El capítulo habla acerca de resultados obtenidos del uso aplicación, se realizaron pruebas de usabilidad y se validó los métodos de ingreso de comprobantes

4.1 Pruebas de Usabilidad

Las pruebas de usabilidad tienen la finalidad de medir el grado de dificultad de interacción entre el usuario y la aplicación.

Existen dos formas de medir la usabilidad, estas son:

Expertos: Un equipo de expertos evalúa cada uno de los aspectos del aplicativo, desde la parte visual a la parte funcional, estos equipos se encargan de hacer fallar el aplicativo y así evitar cualquier tipo de riesgo. Existen muchas empresas o personal independiente encargado de realizar esta actividad, pero sus servicios son costosos.

Encuestas: Es esta es una de las opciones más usadas consiste en diseñar una encuesta y que los usuarios finales la prueben.

Las pruebas que se realizaron fueron usando encuestas, se seleccionó un grupo de 10 personas, se les entrego el aplicativo final y se les solicito que cumplan las actividades.

Acciones a medir

Las acciones que se les pidió a los usuarios calificar fueron las siguientes.

- Cree un usuario nuevo
- Inicie sesión con el usuario antes creado
- Cargue una factura usando el método de ingreso manual
- Cargue una factura usando el método QR
- Cargue una factura usando el método OCR
- Visualice su estado actual de declaración
- Genere el anexo
- Cierre su sesión

El formato de la encuesta utilizada se la puede visualizar en el anexo.

4.2 Resultados obtenidos del test de usabilidad

Para obtener mejores resultados las personas que realizaron el test de usabilidad deben ser ajenas al equipo de desarrollo.

En la figura, podemos ver a algunos usuarios realizando las encuestas. Se realizaron un total de diez encuestas, cabe mencionar que cinco de estas personas estaban en edades comprendidas de 20 a 35 años y los otros 5 restantes en edades superiores a las antes mencionadas, los resultados obtenidos se detallan en la tabla.



Figura 4. 1 Test de usabilidad de aplicación

Fuente: Autores

Elaboración: Autores.

Los resultados obtenidos de test realizado se muestran en la tabla 4.1

Tabla 4. 1 Resultados de test de usabilidad

Actividad a medir	Fácil	Normal	Difícil
Cree un usuario nuevo	100%	0%	0%
Inicie sesión con el usuario antes creado	100%	0%	0%
Cargue una factura usando el método de ingreso manual	100%	0%	0%
Cargue una factura usando el método QR	100%	0%	0%
Cargue una factura usando el método OCR	70%	30%	0%

Visualice su estado actual de declaración	100%	0%	0%
Genere el anexo	100%	0%	0%
Cierre su sesión	100%	0%	0%

Fuente: Autores

Elaboración: Autores.

Con esto se puede evidenciar que la aplicación es intuitiva de fácil manejo, en campo de adquisición mediante reconocimiento óptico de caracteres, 3 de los 10 encuestados comentaron que este método tomaba alrededor del tiempo que toma cargar una factura usando el método de ingreso manual.

4.3 Validación de módulos.

Como se lo expuso en el capítulo 2, la metodología utilizada fue la de desarrollo incremental que exigió la validación de cada uno de los módulos, ahora se detalla el tipo de validación usada por cada módulo. En todos los módulos de adquisición de información, se consulta previo al envío de la información que la base de datos no cuente con la misma información ya almacenada, si está ya se encuentra almacenada da al usuario la opción de poder editar la misma, figura 4.2.



Figura 4. 2 Comprobación de existencia de información en a base de datos

Fuente: Autores

Elaboración: Autores.

4.3.1 Adquisición de datos manual

Previo al envío de la información a la base de datos, se comprueba que todos los campos no se encuentren vacíos, luego verifica de la misma manera campo a campo que la información que ingreso el usuario sea la requerida, es decir en los campos en los que se tenga que ingresar número de cedula, nombres, montos, etc. se ingresan los caracteres correspondientes sean números o letras.

En la figura 4.3, podemos ver un ingreso erróneo y un ingreso correcto, el sistema solo permitirá subir los datos correctos, por el contrario, notificara al usuario que datos están errados.

Campo	Estado Erróneo (Izquierda)	Estado Correcto (Derecha)
Nombre:	Pedro Ivan Sanchez Portoc	Pedro <u>Ivan</u> Sanchez Portoc
Factura N°:	[Vacío] (Campo incompleto)	00001
Fecha dd/mm/aa:	23/7/2018	23/7/2018
CI/RUC Cliente:	1105904567	1105904567
Proveedor:	Crealvid	Crealvid
CI/RUC Proveedor:	1102390133	1102390133
IVA:	12%	12%
Subtotal:	12	12
Salud:	12	12
Vestimenta:	0	0
Educación:	0	0
Alimentación:	0	0
Vivienda:	0	0
Varios:	0	0
Total \$:	12,00	12,00
Botón	CARGAR DATOS	CARGAR DATOS

Figura 4. 3 Validación campo a campo adquisición manual de información

Fuente: Autores

Elaboración: Autores.

4.4 Adquisición de datos mediante escaneo de código QR

Se comprueba que el código decodificado contenga un código de acceso y este estructurado de tal manera dicha cadena.

Si se escanea un QR no válido (cualquier QR que no contenga el código de acceso) se presenta una alerta notificando al usuario que el código QR no es válido, en caso de que contenga el código de acceso y no esté estructurado de la forma correcta se presenta el mismo mensaje.

Finalmente, si el escaneo es realizado a un código que contenga la estructura propuesta, la información se envían automáticamente a la base de dato.

La figura 4.4 muestra los mensajes de notificación en caso de código valido y código erróneo.

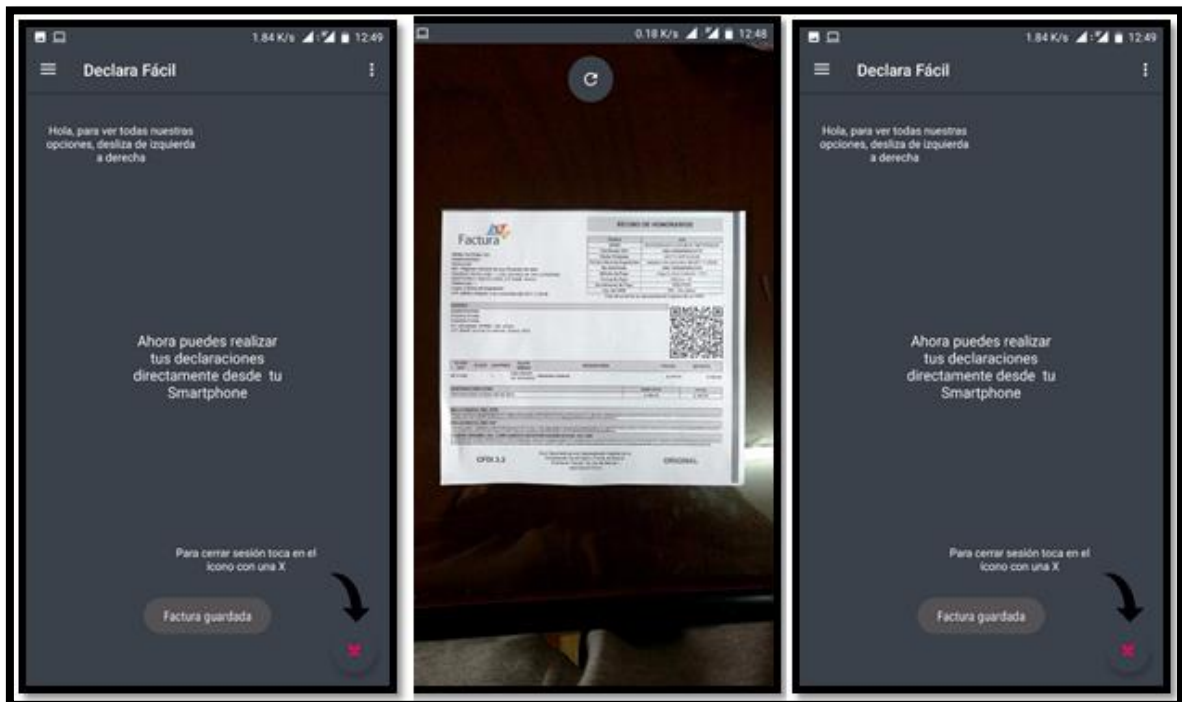


Figura 4. 4 Validación código QR

Fuente: Autores

Elaboración: Autores.

4.3.1 Adquisición de datos mediante reconocimiento óptico de caracteres.

En el capítulo II, se realizó un estudio para la selección del motor OCR, se analizaron un total de 96 facturas en distintas condiciones, dando como resultado general una efectividad de la precisión general del 75% y para la condición recomendada una precisión del 95%.

Previo al envío de información a la base de datos el usuario puede corroborar que todos los datos extraídos mediante reconocimiento óptico de caracteres sean los correctos y a si vez deducir los gastos personales y luego adjuntar la información de la factura a la base de datos.

La figura 4.5 presenta la validación realizada por el usuario previo al envío de información.

FACTURA No. 001001-000000172	
Cliente:	PATRICIA LUDEA GONZALEZ
RUC:	1103997530
Domicilio:	CUERO Y
Fecha:	Loja, 27/08/2016
SUBTOTAL	5.37
TARIFA 0	2.20
IVA 14%	0.44
TOTAL	5.81

PROCESAR IMAGEN

Nombre: PATRICIA LUDEA GONZAL

Factura N°: 001001-000000172

Fecha: 27/08/2016

CI/RUC Cliente: 1103997530

Proveedor: _____

CI/RUC Proveedor: _____

Subtotal: 5.37

Salud: _____

Vestimenta: _____

Educación: _____

Alimentación: _____

Vivienda: _____

Varios: _____

Total \$: 0.00

GUARDAR DATOS OCR

Figura 4. 5 Validación reconocimiento óptico de caracteres

Fuente: Autores

Elaboración: Autores.

4.5 Adquisición de datos mediante escaneo de código QR

El módulo se encarga de realizar consultas dentro de la base de datos, de esta forma se verifica que exista información para su posterior descarga y procesamiento.

En el caso de que exista información, se podrá visualizar en gráficos estadísticos (figura 4.6 izquierda) con opción a generar un archivo en formato Microsoft Excel (figura 4.6 derecha) con el resumen de gastos deducibles, este se almacena dentro del directorio "DeclaraFacil" en el dispositivo móvil. En el caso de que no exista información en la base de datos, se presentara un mensaje notificando que no existen datos almacenados.

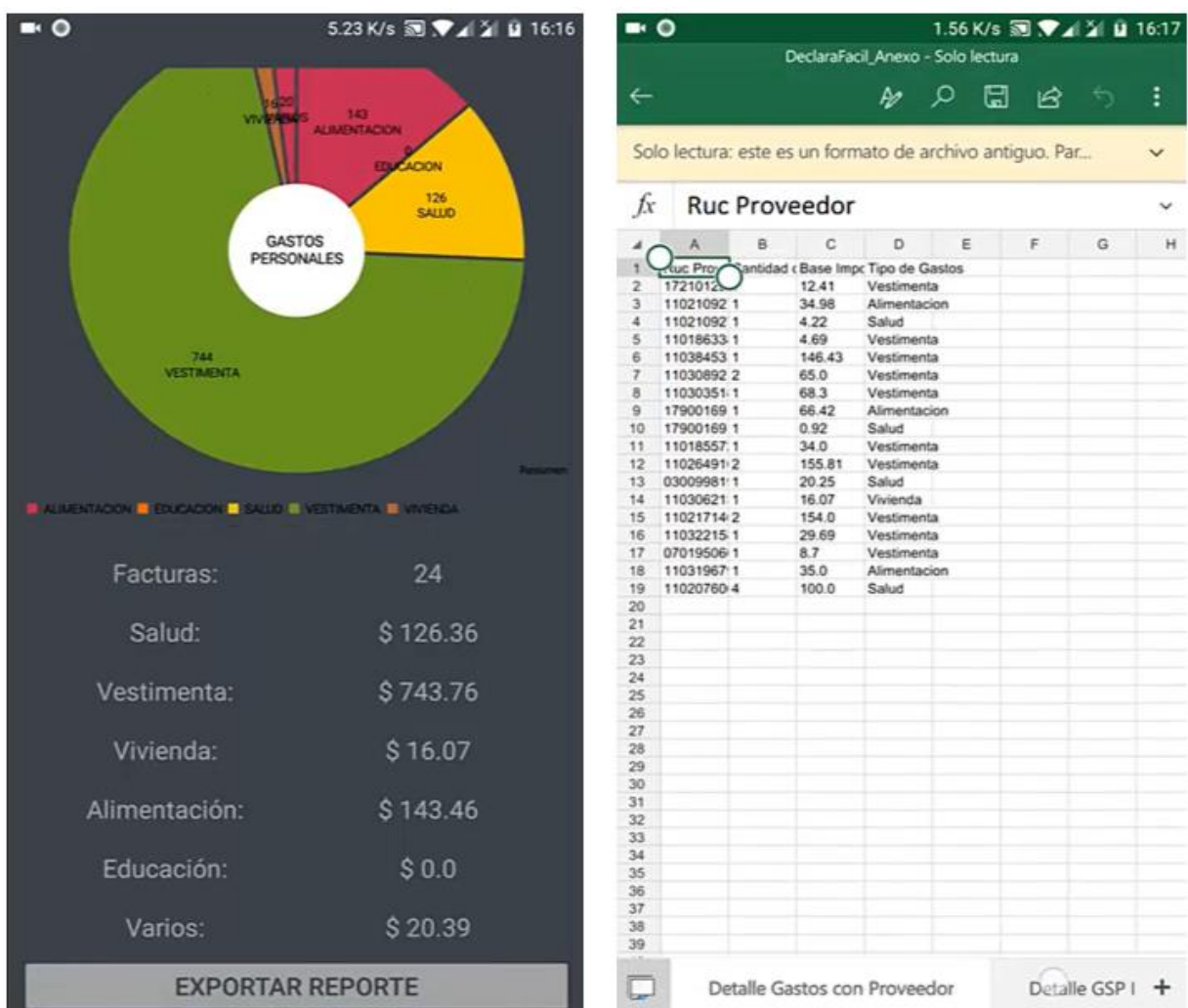


Figura 4. 6 Validación Módulo de visualización y generación de anexo

Fuente: Autores

Elaboración: Autores.

4.6 Manual de usuario

Para mostrar el funcionamiento completo del aplicativo, se realizó un video explicativo, mostrando todas las características implementadas en el mismo, el enlace del video se encuentra almacenado en la plataforma YouTube y el enlace del mismo se encuentra en el Anexo

CONCLUSIONES

- Se logro diseñar e implementar una aplicación móvil en Android para la adquisición de información tributaria de facturas físicas haciendo uso de tecnologías como: Base de datos, OCR y QR.
- La investigación realizada facilitó la elección de los motores OCR, la mayoría de sistemas que realizan reconocimiento óptico de caracteres usan Google Mobile Vision o Tesseract OCR.
- El análisis comparativo realizado entre los motores OCR tanto en tiempos de procesamiento como en porcentajes de error usando fotos es diferentes condiciones, facilito la elección del motor a usar, los resultados de estas pruebas fueron favorables para Google Mobile Vision que es hasta 5 veces más rápido y 40% más preciso que Tesseract OCR.
- Gracias a la aplicación de un pre-procesado: uso de corrección de perspectiva, filtrado y eliminación de ruido, se logró disminuir el margen de error de 26% a 4% en textos, lo que ayuda notablemente a la confiabilidad de la extracción de datos mediante OCR.
- Firebase permitió incorporar servicios como: Autenticación y Base de datos haciendo uso de una única plataforma para administración y haciendo uso de reglas de seguridad en Firebase Database se puede delimitar el acceso a la información de cada usuario.
- El método de adquisición de información más rápido y efectivo es el de lectura de QR, ya que toda la información contenida en el código QR viene validada desde su impresión.
- Se implementó funcionalidades adicionales, como la visualización de montos por rubro en gráficos estadísticos, número de facturas almacenadas, etc.

RECOMENDACIONES

- Para la selección del motor OCR, se puede analizar más condiciones de imagen que las expuestas en este trabajo.
- Realizar fotos con buenas condiciones de luz, de preferencia las fotos deben estar tomadas a máximo 15 centímetros de distancia.
- Firebase Database funciona muy bien; sin embargo, no es completamente gratuita, existen limitaciones que debe tener en cuenta. Además, se debe pensar en cómo se va a mostrar los datos antes de crear la estructura de misma.

BIBLIOGRAFÍA

Alonso, R. R., & Torres, V. D. (2014). Reconocimiento de Caracteres en Imagen Digital, una Aplicación al Reconocimiento Automático de Placas de Vehículos. *FPUNE Scientific*, 2(2). Retrieved from http://www.une.edu.py:82/fpune_scientific/index.php/fpunescientific/article/view/41

Álvarez Durán, M. A. (2014). Análisis, diseño e implementación de un sistema de control de ingreso de vehiculos basado en visión artificial y reconocimiento de placas en el parqueadero de la Universidad Politécnica salesiana-Sede Cuenca., 144. Retrieved from <https://dspace.ups.edu.ec/bitstream/123456789/7060/1/UPS-CT003790.pdf>

ARCOTEL. (2017). 46,4% de usuarios del Servicio Móvil Avanzado poseen un smartphone | Agencia de Regulación y Control de las Telecomunicaciones | Ecuador. Retrieved July 14, 2018, from <http://www.arcotel.gob.ec/464-de-usuarios-del-servicio-movil-avanzado-poseen-un-smartphone/>

Avison, D. E., & Fitzgerald, G. (Guy). (2003). *Information systems development: methodologies, techniques and tools*. McGraw-Hill. Retrieved from <https://eprints.soton.ac.uk/35879/>

Báez López, D. (2006). *MATLAB con aplicaciones a la ingeniería, física y finanzas*. Alfaomega. Retrieved from <https://www.casadellibro.com/libro-matlab-con-aplicaciones-a-la-ingenieria-fisica-y-finanzas/9789701511374/1187922>

Baharlou, S. M., Hemayat, S., Saberhari, A., & Yaghoobi, S. (2015). Fast and adaptive license plate recognition algorithm for Persian plates. In *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)* (pp. 1–6). IEEE. <https://doi.org/10.1109/PRIA.2015.7161638>

Baz Alonso, A., Ferreira Artime, I., Álvarez Rodríguez, M., & García Baniello, R. (n.d.). *Dispositivos móviles*. Retrieved from <http://isa.uniovi.es/docencia/SIGC/pdf/telefoniamovil.pdf>

Bhaskar, S., Lavassar, N., & Green, S. (2010). Implementing Optical Character Recognition on the Android Operating System for Business Cards, 5. Retrieved from https://pdfs.semanticscholar.org/1bf3/c55d17214452970497a0915ee945464d5742.pdf?_ga=2.168931107.344121899.1531408526-1977943489.1531408526

Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013). PhotoOCR: Reading Text in Uncontrolled Conditions. In 2013 IEEE International Conference on Computer Vision (pp. 785–792). IEEE. <https://doi.org/10.1109/ICCV.2013.102>

Cattaneo, C. A., Larcher, L. I., Ruggeri, A. I., Herrera, A. C., & Bionani, E. M. (2011). Métodos de Umbralización de Imágenes Digitales Basados en Entropía de Shannon y Otros. *Mecánica Computacional*, 30(36), 2785–2805. Retrieved from <https://cimec.org.ar/ojs/index.php/mc/article/view/3951>

Ciencias Empresariales, F. DE, & Andrea Ramírez Norambuena Christopher Edgardo Arevalo Gonzales, V. (2010). Desarrollo de aplicación móvil en plataforma Android en apoyo a visitas médicas, 133. Retrieved from [http://werken.ubiobio.cl/html/documentos/memorias_bcm_errores/Memorias con Error/M\(DC\) 001.6 Ar34 2010/Proyecto de titulo/Informe/InformeProyectoTitulo.pdf](http://werken.ubiobio.cl/html/documentos/memorias_bcm_errores/Memorias con Error/M(DC) 001.6 Ar34 2010/Proyecto de titulo/Informe/InformeProyectoTitulo.pdf)

Daniela, J., Salgueiro, A., De La, A., Gascueña, P., Sánchez Muñoz, E., Botella, G., ... Martín Hernández, A. (2012). Procesado de imágenes para plataformas móviles. Retrieved from https://eprints.ucm.es/22577/1/Procesado_de_imagenes_para_plataformas_moviles.pdf

Developers Android. (n.d.). Arquitectura de la plataforma | Android Developers. Retrieved July 11, 2018, from <https://developer.android.com/guide/platform/?hl=es-419>

Do, H. N., Vo, M.-T., Vuong, B. Q., Pham, H. T., Nguyen, A. H., & Luong, H. Q. (2016). Automatic license plate recognition using mobile device. In 2016 International Conference on Advanced Technologies for Communications (ATC) (pp. 268–271). IEEE. <https://doi.org/10.1109/ATC.2016.7764786>

Garcia Santillán, I. D. (2008). *Vision Artificial y Procesamiento Digital de Imagenes Usando Matlab*. Retrieved from <https://es.scribd.com/doc/226655280/Vision-Artificial-y-Procesamiento-Digital-de-Imagenes-Usando-Matlab>

Gasca Mantilla, M. C., Camargo Ariza, L. L., & Medina Delgado, B. (2013). Metodología para el desarrollo de aplicaciones móviles | Gasca Mantilla | Tecnura. Retrieved December 7, 2017, from <http://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>

gcfaprendelibre.org. (2016). iPad - Sistema operativo Móvil iOS. Retrieved July 12, 2018, from https://www.gcfaprendelibre.org/tecnologia/curso/ipad/caracteristicas_generales_del_ipad/3.do

Gomezjurado, J., Núñez, J., Cordero, J., & Uyaguari, F. (2014). HISTORIA DE LAS TELECOMUNICACIONES EN EL ECUADOR. Retrieved from <http://corporativo.cnt.gob.ec/wp-content/uploads/2014/07/LIBRO-CNT-WEB.pdf>

González, R. C., & Wintz, P. (1996). Filtrado de imágenes, 24. Retrieved from <http://informatica.uv.es/doctorado/AIRF/ParteAI/tema3.pdf>

Heliński, M., Kmiecik, M., & Parkoła, T. (2012). Report on the comparison of Tesseract and ABBYY FineReader OCR engines, 24. Retrieved from http://dl.psnc.pl/download/PSNC_Tesseract-FineReader-report.pdf

INEC. (2016). Tecnologías de la Información y Comunicaciones (TIC'S) 2016, 39. Retrieved from http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2016/170125.Presentacion_Tics_2016.pdf

Javier, C., & Fernández, S. (2008). Reconocimiento Óptico de Caracteres (OCR). Retrieved from <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf>

Jiang, H., Gonnot, T., Yi, W.-J., & Saniie, J. (2017). Computer vision and text recognition for assisting visually impaired people using Android smartphone. In 2017 IEEE International Conference on Electro Information Technology (EIT) (pp. 350–353). IEEE. <https://doi.org/10.1109/EIT.2017.8053384>

Kongtahn, A., Minsakorn, S., Yodchaloemkul, L., Boontarak, S., & Phongsuphap, S. (2014). Medical document reader on Android smartphone. In 2014 Third ICT International Student Project Conference (ICT-ISPC) (pp. 65–68). IEEE. <https://doi.org/10.1109/ICT-ISPC.2014.6923219>

Lamar, C. (1999). UML y Patrones (2nd ed.). Retrieved from <http://www.fmonje.com/UTN/ADES - 208/UML y Patrones 2da Edicion.pdf>

León Aucapiña, Y. J. (2017). Diseño e implementación de una aplicación para la detección de placas vehiculares a partir de imágenes, 98. Retrieved from http://dspace.utpl.edu.ec/bitstream/123456789/19717/1/León_Aucapiña%2C_Yosselin_Jackeline_TESIS.pdf

López Airam, V. (2007). Detección de trayectorias y reconocimiento de objetos regulares para el control por visión artificial de un robot móvil., 91. Retrieved from https://tesis.ipn.mx/jspui/bitstream/123456789/12629/1/Detección_de_trayectorias_y_reconocimiento_de_objetos_regulares_para_el_control_por_visión_artificial_de_un_robot_móvil.pdf

Lozano, M. (2018). ¿Qué es un Smartphone? Conoce todos los detalles | WhistleOut. Retrieved June 21, 2018, from <https://www.whistleout.com.mx/CellPhones/Guides/que-es-un-smartphone>

Magalhaes, L., Ribeiro, B., Alves, N., & Guevara, M. (2017). A three-staged approach to medicine box recognition. In 2017 24o Encontro Português de Computação Gráfica e Interação (EPCGI) (pp. 1–7). IEEE. <https://doi.org/10.1109/EPCGI.2017.8124317>

Ocampo Carrion, J. L. (2011). Reconocimiento de caracteres de una placa de automovil haciendo uso de redes neuronales en Matlab. Retrieved from <http://repositorio.espe.edu.ec/xmlui/handle/21000/4939>

Palomino, N. L. S., & Roman Concha, U. N. (2014). Revista de investigación de sistemas e informática. Revista de investigación de Sistemas e Informática (Vol. 6). Universidad Nacional Mayor de San Marcos. Revista de investigación de Sistemas e Informática (Vol. 6). Universidad Nacional Mayor de San Marcos. Retrieved from <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3299>

Pungacho Sánchez, L. D. (2017). “Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.” Retrieved from <http://repositorio.puce.edu.ec/handle/22000/13481>

Ramiah, S., Liong, T. Y., & Jayabalan, M. (2015). Detecting text based image with optical character recognition for English translation and speech using Android. In 2015 IEEE Student Conference on Research and Development (SCOReD) (pp. 272–277). IEEE. <https://doi.org/10.1109/SCORED.2015.7449339>

Ramírez-Manzanares, A. (2014). Reporte de Búsqueda, detección y conteo de objetos, 40. Retrieved from https://www.cimat.mx/~alram/VC/ramirez_segObjDetecc.pdf

Ravina, M., Supriya, I., & Nilam, D. (2013). Optical Character Recognition, 4. Retrieved from <http://code.google.com/p/tesseract-ocr>.

Sanchez Tirado, S. A. (2006). Integración de tecnicas de manejo de la imagen digital para la web y aplicaciones multimedia.

Servicio de Rentas Internas del Ecuador. (2018). Detalle Noticias - Servicio de Rentas Internas del Ecuador. Retrieved June 21, 2018, from <http://www.sri.gob.ec/web/guest/detalle-noticias?idnoticia=524>

Servicio de Rentas Internas del Ecuador (SRI). (n.d.). Guía Práctica declaración del Impuesto a la Renta. Retrieved November 21, 2017, from <http://www.sri.gob.ec/web/guest/384>

Singh, S. (2013). Journal of Emerging Trends in Computing and Information Sciences Optical Character Recognition Techniques: A Survey, 4(6). Retrieved from <http://www.cisjournal.org>

sites.google.com. (n.d.). 2.2. Arquitectura Android - Software de Comunicaciones. Retrieved July 11, 2018, from <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Telesemana. (2017). Estadísticas: telecomunicaciones en Ecuador | TeleSemana.com. Retrieved July 17, 2018, from <http://www.telesemana.com/panorama-de-mercado/ecuador/>

Vuong, B. Q., & Do, H. N. (2014). Design and implementation of multilanguage name card reader on Android platform. In 2014 International Conference on Advanced Technologies for Communications (ATC 2014) (pp. 528–531). IEEE. <https://doi.org/10.1109/ATC.2014.7043445>

ANEXOS

ANEXO A.

Resultados obtenidos con respecto al porcentaje de error en diferentes tipos de facturas, utilizando Google Vision.

GOOGLE VISION					
Condiciones\tipos	Factura 1	Factura 2	Factura 3	Factura 4	%Error de evento
Clara cerca	7,39%	4,95%	4,14%	7,05%	26,21%
Clara lejos	100,00%	99,01%	4,94%	82,19%	71,53%
Ángulo claro cerca	9,98%	3,39%	4,30%	8,35%	6,51%
Ángulo claro lejos	99,08%	94,34%	92,83%	7,79%	73,51%
Obscuro cerca	10,17%	1,27%	17,20%	9,65%	9,57%
Obscuro lejos	90,94%	1,27%	4,94%	6,31%	25,86%
Ángulo obscuro cerca	8,87%	1,27%	6,21%	20,04%	9,10%
Ángulo obscuro lejos	12,20%	0,85%	2,87%	14,84%	7,69%

	Nº CHARACTERS	Nº ACIERTOS	% ERROR	%Error de evento
Claras cerca factura 1	541	501	7,39	5,88
Claras cerca factura 2	707	672	4,95	
Claras cerca factura 3	628	602	4,14	
Claras cerca factura 4	539	501	7,05	

Claras lejanas facturas 1	541	0	100	%Error de evento
---------------------------	-----	---	-----	------------------

Claras lejanas facturas 2	707	7	99,01	71,53
Claras lejanas facturas 3	628	597	4,94	
Claras lejanas facturas 4	539	96	82,19	
Ángulo cerca factura 1	541	487	9,98	%Error de evento
Ángulo cerca factura 2	707	683	3,39	6,51
Ángulo cerca factura 3	628	601	4,30	
Ángulo cerca factura 4	539	494	8,35	
Ángulo lejanas factura 1	541	5	99,08	%Error de evento
Ángulo lejanas factura 2	707	40	94,34	73,51
Ángulo lejanas factura 3	628	45	92,83	
Ángulo lejanas factura 4	539	497	7,79	
Ángulo cercano obscuro factura 1	541	493	8,87	%Error de evento
Ángulo cercano obscuro factura 2	707	698	1,27	9,10
Ángulo cercano obscuro factura 3	628	589	6,21	
Ángulo cercano obscuro factura 4	539	431	20,04	
Lejana obscuro factura 1	541	49	90,94	%Error de evento

Lejana obscuro factura 2	707	698	1,27	25,86
Lejana obscuro factura 3	628	597	4,94	

Lejana obscuro factura 4	539	505	6,31	
Ángulo lejano obscuro factura 1	541	475	12,20	%Error de evento
Ángulo lejano obscuro factura 2	707	701	0,85	7,69

Ángulo lejano obscuro factura 3	628	610	2,87	
Ángulo lejano obscuro factura 4	539	459	14,84	
Cerca obscuro factura 1	541	486	10,17	%Error de evento
Cerca obscuro factura 2	707	698	1,27	9,57
Cerca obscuro factura 3	628	520	17,20	
Cerca obscuro factura 4	539	487	9,65	

Resultados obtenidos referentes al porcentaje de error usando varios tipos de facturas con el motor Tesseract OCR.

TESSERACT OCR					
Condiciones\tipos	Factura 1	Factura 2	Factura 3	Factura 4	%Error de evento
Clara cerca	83,87%	86,56%	85,67%	85,53%	85,37%
Clara lejos	74,86%	64,78%	68,63%	45,27%	63,39%

Ángulo claro cerca	68,02%	85,86%	76,91%	55,29%	71,52%
--------------------	--------	--------	--------	--------	--------

Ángulo claro lejos	64,33%	78,22%	61,31%	67,35%	67,80%
Obscuro cerca	52,68%	56,86%	49,04%	57,33%	53,93%
Obscuro lejos	51,76%	50,64%	60,19%	71,61%	58,55%
Ángulo obscuro cerca	84,29%	96,61%	85,67%	83,86%	87,61%
Ángulo obscuro lejos	83,18%	93,49%	80,89%	85,53%	85,77%

condiciones fotos	Nº caracteres	Nº aciertos	% error	%Error de evento
Claros cerca factura 1	541	88	83,73	85,37
Claros cerca factura 2	707	95	86,56	
Claros cerca factura 3	628	90	85,67	
Claros cerca factura 4	539	78	85,53	
Claros lejanas facturas 1	541	136	74,86	%Error de evento
Claros lejanas facturas 2	707	249	64,78	63,39
Claros lejanas facturas 3	628	197	68,63	
Claros lejanas facturas 4	539	295	45,27	
Ángulo cerca factura 1	541	173	68,02	%Error de evento
Ángulo cerca factura 2	707	100	85,86	71,52

Ángulo cerca factura 3	628	145	76,91
Ángulo cerca factura 4	539	241	55,29

Ángulo lejanas factura 1	541	193	64,33	%Error de evento
Ángulo lejanas factura 2	707	154	78,22	67,80

Ángulo lejanas factura 3	628	243	61,31	
Ángulo lejanas factura 4	539	176	67,35	
Ángulo cercano obscuro factura 1	541	256	52,68	%Error de evento
Ángulo cercano obscuro factura 2	707	305	56,86	53,98
Ángulo cercano obscuro factura 3	628	320	49,04	
Ángulo cercano obscuro factura 4	539	230	57,33	
Lejana obscuro factura 1	541	261	51,76	%Error de evento
Lejana obscuro factura 2	707	349	50,64	58,55
Lejana obscuro factura 3	628	250	60,19	
Lejana obscuro factura 4	539	153	71,61	
Ángulo lejano obscuro factura 1	541	85	84,29	%Error de evento
Ángulo lejano obscuro factura 2	707	24	96,61	87,61
Obscuro lejano obscuro factura 3	628	90	85,67	
Obscuro lejano obscuro factura 4	539	87	83,86	

Cerca obscuro factura 1	541	91	83,18	%Error de evento
Cerca obscuro factura 2	707	46	93,49	85,77
Cerca obscuro factura 3	628	120	80,89	

Cerca obscuro factura 4	539	78	85,53	
-------------------------	-----	----	-------	--

ANEXO B

Tablas de ponderación en base a pruebas de funcionalidad y requerimientos por parte de los desarrolladores

Cuadro de ponderación de librería Google Mobile Vison					
Preguntas	Valor 1	Valor 2	Valor 3	Valor 4	Valor 5
1.- Tiempos de procesamiento					X
2.- Digitalización del texto				X	
3.- Facilidad de integración					X
4.- Documentación					X
5.- Desarrollo de la comunidad					X
6.- Funcionalidad					X
Total	29				

1	2	3	4	5
Muy lenta	Lenta	Medianamente rápida	Rápida	Muy rápido
Muy mala	Mala	Normal	Buena	Muy buena

Muy difícil	Difícil	Neutral	Fácil	Extremadamente fácil
-------------	---------	---------	-------	----------------------

Escasa información	Poca información	información necesaria	Mucha información	Abundante información
--------------------	------------------	-----------------------	-------------------	-----------------------

Escaso desarrollo	Poco desarrollo	información necesaria	Mucha información	Abundante desarrollo
Malo	Promedio	Bueno	Muy bueno	Excelente

Cuadro de ponderación de librería Tesseract					
Preguntas	Valor 1	Valor 2	Valor 3	Valor 4	Valor 5
1.- Tiempos de procesamiento			X		
2.- Digitalización del texto			X		
3.- Facilidad de integración		X			
4.- documentación					X
5.- Desarrollo de la comunidad					X
6.- funcionalidad				X	
Total	22				

ANEXO C

Test de usabilidad aplicado a los participantes del aplicativo

Instrucciones:

A continuación, se le prestan el prototipo del aplicativo “DECLARA FACIL”, el motivo de esta encuesta es dar a conocer que tan intuitiva es.

Usted debe marcar con una X si la tarea que se le encomendó es fácil, normal o difícil de cumplir.

Para la realización de la misma el equipo evaluador le presentará un video explicativo y luego le dará los comprobantes y la aplicación para que pueda realizar las actividades.

Actividad a medir	Fácil	Normal	Difícil
1.- Cree un usuario nuevo			
2.- Inicie sesión con el usuario antes creado			
3.-Cargue una factura usando el método de ingreso manual			
4.-Cargue una factura usando el método QR			
5.-Cargue una factura usando el método OCR			
6.-Visualice su estado actual de declaración			
7.-Genere el anexo			
8.-Cierre su sesión			

Gracias

ANEXO D

Video explicativo de uso de la aplicación móvil: <https://youtu.be/MAfUxR3DXek>