Componente para la reproducción por audio y diseño visual en la aplicación móvil iLex Notario

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Wendy Lara González

Eddy González Escandell

Tutores:

MSc. Yarina Amoroso Fernández

MSc. José Miguel Fabra Gallo

Ing. Yoannys Gustavo Dueñas Pérez

Ing. Felix Antonio Marrero Pentón

La Habana, Cuba

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2020.

Wendy Lara González	**Eddy González Escandell**
Autor	Autor
MSc. Yarina Amoroso Fernández	**MSc. José Miguel Fabra Gallo**
Tutor	Tutor
Ing. Yoannys Gustavo Dueñas Pérez	**Ing. Felix Antonio Marrero Pentón**
Tutor	Tutor

DEDICATORIA

A mi madre por ser mi guía, por todo su amor y por ser la mejor madre del mundo.

A mi hermanapequeña por cada pregunta de preocupación y ser su ejemplo a seguir.

A mi abuela por sus consejos y siempre estar para mí, a Adalberto por su apoyo durante toda esta etapa.

A ustedes gracias por estar siempre conmigo.

Wendy Lara González

A mi madre y a mi padre, por ser ese ejemplo incondicional de superación personal.

Por no permitir que nunca me hiciera falta nada, por demostrarme que con inteligencia y sacrificio todo se gana, por hacerme el hombre que soy.

A mi hermana, por darme ánimo para ser mejor persona cada día.

A todos ustedes por estar siempre conmigo.

Eddy González Escandell

AGRADECIMIENTOS

A la Universidad de las Ciencias Informáticas por formarme como una profesional y a la Revolución Cubana por darme la oportunidad de estudiar en ella.

A mi madre por todo su amor, cariño, apoyo incondicional y comprensión; por ser mi guía y la mejor madre del mundo.

A mi hermana Daiceli, que el hecho de saber que puedo ser un ejemplo para ella me mantiene firme.

A mi abuela Dayni por ser la primera persona en escucharme ante cada tristeza o alegría y por siempre estar para mí.

A mi padrastro Adalberto por ser mi apoyo y la persona que nunca faltó siempre que necesitaba ayuda.

A mi papá que de una forma u otra siempre me ha apoyado.

A mi novio Alfonso por todos sus consejos, apoyo incondicional, ejemplo a seguir y por hacerme muy feliz.

A mis suegros por todo su cariño, apoyo y hacerme parte de la familia.

A mis maravillosos tutores Fabra,Félix yYoannys por su gran ayuda, apoyo y dedicación para la realización de este trabajo de diploma.

A mi compañero de tesis, por su esfuerzo en el desarrollo de la investigación.

A todos los integrantes del proyecto, que en momentos difíciles tendieron su mano, en especial a Reinier y Yoslenys.

A mis compañeros de aula, en especial a los que venimos juntos desde el inicio, con los cuales he pasado excelentes e inolvidables momentos. Los quiero mucho.

A todos los profesores que han contribuido con mi formación profesional, en especial a esos que su preocupación fue constante, gracias por creer en mí.

A todos los que de una forma u otra han contribuido a la realización del presente trabajo. A todos aquellos que no he mencionado por falta de espacio pero que estuvieron ahí brindando su ayuda.

A todos, muchas gracias.

Wendy Lara González

A la Revolución cubana por hacerme un profesional y darme la oportunidad de estudiar en una de las universidades más prestigiosas del país. Gracias al pensamiento de nuestro Fidel Castro y la continuidad de su legado.

A mi madre, Maria Isabel, ya que este momento se hace realidad gracias a ella.

A mi padre, Ediberto, por darme esos empujones para que siempre sea mejor persona.

A mi hermana Thalia que siempre estuvo para mí en todo momento junto con su novio Daniel.

A mis compañeros de clases por compartir grandes momentos.

A Daniel, David y Abel por pasar esos momentos que todo joven universitario tiene que disfrutar.

En especial a mi novia Laura por estar este tiempo a mi lado y darme ese amor para sobrepasar los obstáculos que nos pone la vida.

A la FEU de la UCI, que me enseñó muchas cosas y en especial a querer más a mi país.

A mis tutores y el colectivo de profesores que me guiaron a este momento.

Y a todos los que de una manera u otra me ayudaron a que este sueño se hiciera realidad.

Eddy González Escandell

RESUMEN

iLex Notario es una aplicación móvil desarrollada para facilitar el acceso de la población a información oportuna y confiable sobre las notarías. Esta solución informática, responde a la necesidad de incrementar la calidad de los servicios notariales, dotando al ciudadano de información que mejora su experiencia de usuario y le permita exigir un servicio acorde a sus intereses. Si bien los usuarios y juristas reconocen la utilidad de esta aplicación móvil y su contribución al incremento de la calidad de los servicios jurídicos en el país, su impacto todavía es limitado ya que no está diseñada para facilitar la consulta a discapacitados visuales. En tal sentido, el presente trabajo tiene el objetivo de desarrollar un componente para la reproducción por audio y diseño visual en la aplicación móvil iLex Notario de manera que facilite la consulta de los contenidos a personas con discapacidad visual.Para guiar la propuesta de solución se aplicala metodología de desarrollo de software Proceso Unificado Ágil, en su variación para la Universidad de las Ciencias Informáticas. Además, se utilizaron las herramientas, lenguajes y tecnologías acordes a las necesidades de la solución, las líneas directrices de la universidad y las políticas de soberanía tecnológica definidas en el país. Utilizando métricas de validación del diseño ypruebas de software se pudo comprobar de forma cuantitativa la calidad de los artefactos obtenidos durante el desarrollo de la propuesta de solución.

Palabras claves: aplicaciones móviles, discapacitados visuales, iLex Notario

ABSTRACT

iLex Notary is a mobile app developed to facilitate the population's access to an on time and reliable informationregarding to notaries. This informatics' solution, answers to the necessity of increasing the quality for the notarial services, providing the citizens with information which improves its users experience and allow them to demand a service according to their interests. Although users and jurist are well awarded of the utility of this mobile app and its contribution to the enhancement of the quality of the juridical services along the country, it impact is still limited due to it is not designed to facilitate the visual handicapper's consult: restraining the reach of the information which it provides. In this meaning, the current work has as objective to develop a component for the audio reproduction and visual design on the iLex Notary mobile app in a fashion which facilitate the consulting of the contents for people with visual handicappers. To guide this proposal of solution it is applied the software development methodology AUP (Agile Unified Process) on its variation for the Informatics Science University. Besides, were used the tools, languages and technologies accordingly to the necessities of the solution, the Center/University guidelines and the politics for technological sobering defined in the country. Using design validation metrics and software test, the quality of the artifacts obtained during the development of the solution proposal could be quantitatively verified.

Key words: mobile apps, visual handicappers, iLex Notary

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

INTRODUCCIÓN

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha generado cambios en prácticamente todas las áreas y estructuras de la sociedad moderna; incluyendo las entidades jurídicas, las que deben actuar en nuevos escenarios donde la interacción entre los individuos está fuertemente mediada por las tecnologías y el acceso a la información es cada vez más necesario. En este contexto, las TIC se posicionan no sólo como un instrumento para mejorar el desempeñode este tipo de instituciones, sino que contribuyen a forjar nuevos modelos de gobierno electrónico basados en la gobernanza inteligente, el gobierno abierto y la innovación colaborativa. Además, habilitan a los ciudadanos de herramientas que le permiten incrementar su cultura jurídica y desempeñar un rol más activo en el proceso de toma de decisiones.

Ante estas nuevas realidades, Cuba desarrolla un proceso de informatización de la sociedad, encaminado a la aplicación ordenada y masiva de las tecnologías en la gestión de la información y el conocimiento, para satisfacer gradualmente las necesidades sociales (…) lograr más eficacia y eficiencia en los procesos, así como mayor generación de riqueza y aumento de la calidad de vida de los ciudadanos (MINCOM, 2018). Al respecto, el presidente de la República de Cuba Miguel Díaz-Canel Bermúdez ha planteado que existe una voluntad política de apoyar el proceso de informatización y que es preciso potenciar la soberanía, con software hechos en el país (Figueredo Concepción, 2018)

En la medida que se ha avanzado en este proceso, se empiezan a lograr resultados. Resaltando la presencia en internet de portales institucionales de todos los órganos, Organismos de la Administración Central del Estado (OACE) y los gobiernos territoriales. Creándose plataformas que permiten el pago electrónico de productos y servicios. Realizando inversiones e instalaciones para la mejora y ampliación de la infraestructura para el acceso a internet; incrementando las líneas móviles activas; el desarrollo de contenidos útiles en la web nacional; y las plataformas móviles que facilitan la interacción entre los órganos de justicia y la población.

En este último punto se destaca “iLex Notario”, desarrollada por el Centro de Gobierno Electrónico (CEGEL) de la Universidad de las Ciencias Informáticas (UCI) y la Dirección de Notarías del Ministerio de Justicia de la República de Cuba (MINJUS). Con el objetivo de facilitar el acceso de la población a información sobre las notarías, los servicios que ofrecen y

ampliar los canales de comunicación con sus directivos y funcionarios.Esta solución informática, responde a la necesidad de incrementar la calidad de los servicios notariales, dotando al ciudadano de información que mejore su experiencia de usuario y le permita exigir un servicio acorde a sus intereses.Haciendo uso de las tecnologías móviles por el nivel de penetración y aceptación que tienen en la población. En la versión de iLex Notario disponible en la actualidadlos usuarios pueden:

- Conocer una breve reseña sobre la historia de las notarías en Cuba.
- Conocer los servicios que brindan las notarías del país.
- Conocer los tipos de documentos que se redactan en las notarías.
- Acceder a una lista de preguntas frecuentes que pudieran esclarecer dudas en la población.
- Conocer los deberes y derechos de los notarios y de los ciudadanos.
- Consultar la legislación que regula la actividad notarial en Cuba.
- Consultar un directorio de todas las notarías por provincias y municipios, así como las unidades notariales pertenecientes a las sociedades civiles de derecho jurídico.
- Consultar información de contacto de la Dirección de Notarías del MINJUS y enviar correos electrónicos a los jefes de departamentos de las notarías provinciales.

Si bien usuarios y juristas reconocen la utilidad de esta aplicación móvil y su contribución al incremento de la calidad de los servicios jurídicos en el país, su impacto todavía es limitado ya que:

- La paleta de colores utilizada no favorece la consulta de los contenidos de la aplicación por personas con algún tipo de discapacidad visual.
- Las fuentes empleadas no favorecen la consulta de los contenidos de la aplicación por personas con algún tipo de discapacidad visual.
- El tamaño de letras empleado no favorece la consulta de los contenidos de la aplicación por personas con algún tipo de discapacidad visual.
- En caso de que la persona esté totalmente ciega no puede consultar la información de la aplicación.

Los elementos antes mencionados, evidencian la carencia de un diseño óptimo para que las personas con discapacidad visual hagan uso de iLex Notario.

La situación problemática antes descrita ha generado el siguiente**problema de investigación:**

¿Cómo gestionar la información en la aplicación móvil iLex Notariode manera que facilite la consulta de los contenidos a personas con discapacidad visual?

Este trabajo tiene como **objeto de estudio** el proceso de desarrollo de aplicaciones móviles y como **campo de acción** el desarrollo de aplicaciones móviles para discapacitados visuales.

Se establece como **objetivo general:** Desarrollar un componente para la reproducción por audio y diseño visual en la aplicación móvil iLex Notario de manera que facilite la consulta de los contenidos a personas con discapacidad visual.

Se definen, además, los siguientes **objetivos específicos:**

- Identificar los referentes teóricos en los que se sustenta la propuesta de solución sobre el desarrollo de aplicaciones móviles para discapacitados visuales.
- Realizar la identificación de requisitos, análisis y diseño de la propuesta de solución, como aproximación a la implementación.
- Implementar el componente para la reproducción de audio y diseño visual en la aplicación móvil iLex Notario.
- Validar el funcionamiento de la propuesta de solución aplicando métricas y pruebas de software.

Para cumplir con el objetivo propuesto se plantean las siguientes **tareas de la investigación:**

- Definición de los principales conceptos relacionados con el objeto de estudio y campo de acción.
- Análisis de sistemas similares en el contexto nacional e internacional.
- Descripción del proceso de desarrollo de software y su metodología.
- Caracterización de las tecnologías y herramientas a utilizar para el desarrollo de la solución.
- Especificaciónde los requisitos funcionales y no funcionales de la solución.
- Descripción de la arquitectura de software base para la implementación de la propuesta de solución.
- Caracterización y selección de los patrones de diseño más factibles para la propuesta de solución.
- Realización del diagrama de clases de diseñoy diseño de casos de pruebas.
- Implementación de la propuesta de solución.

- Ejecución de las pruebas de software para evaluar la calidad de la implementación.

Teniendo como **idea a defender:** Si se desarrolla el componente para la reproducción por audio y diseño visual en la aplicación móvil iLex Notario se facilita la consulta de los contenidos a personas con discapacidad visual.

En el desarrollo de la investigación se emplearon los siguientes métodos:

Métodos del nivel teórico:

- **Histórico-Lógico:** empleado en el estudio de la evolución de las herramientas utilizadas por personas con algún tipo de discapacidad visual para acceder a la información, y el comportamiento experimentado por las aplicaciones móviles en este proceso.
- **Modelación:**se utilizópara la realización de algunos diagramas en el proceso de desarrollo de software, haciendo una representación abstracta de la solución.
- **Analítico-Sintético:** utilizado en el estudio de las aplicaciones existentes para discapacitados visuales en Cuba y el mundo, partiendo de sus características y deficiencias para la realización de la investigación.

Métodos del nivel empírico:

- **Entrevista**: se utilizó para identificar el dominio del negocio, comprender la estructura y funcionamiento de la organización y detectar las carencias que dan origen a la investigación.

El presente trabajo de diploma está estructurado en 3 capítulos. En el capítulo 1se abordan los referentes teórico-metodológicos de la investigación. Se realiza un estudio comparativo de soluciones informáticas con similar dominio de actuación, para valorar su viabilidad en la solución del problema planteado y/o buenas prácticas para el desarrollo de nuevas soluciones.Se realiza un estudio de la metodología más factible para el desarrollo de la solución informática, así como las tecnologías, herramientas y entorno de desarrollo.

En el capítulo 2 se realiza una descripción de la propuesta de solución y sus principales funcionalidades. Se argumenta el uso de los patrones de diseño y se describen los artefactos que se generan a partir de la metodología seleccionada.

Finalmente, en el capítulo 3 se elaboran y ejecutan los casos de prueba para verificar el correcto funcionamiento de la aplicación. Se hace un análisis de los resultados obtenidos y se resuelven las no conformidades detectadas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordan los referentes teórico-metodológicos de la investigación. Se realiza un estudio comparativo de soluciones informáticas con similar dominio de actuación, para valorar su viabilidad en la solución del problema planteado y/o buenas prácticas para el desarrollo de nuevas soluciones. Se realiza un estudio de la metodología más factible para el desarrollo de la solución informática, así como las tecnologías, herramientas y entorno de desarrollo.

1.1 Conceptos asociados a la investigación

Un término vital para la comprensión del dominio del problema en cuestión es “discapacidad”; este abarca las deficiencias, las limitaciones de la actividad y las restricciones de la participación. Las deficiencias son problemas que afectan a una estructura o función corporal; las limitaciones de la actividad son dificultades para ejecutar acciones o tareas, y las restricciones de la participación son problemas para participar en situaciones vitales (OMS, 2020). Por consiguiente, la discapacidad es un fenómeno complejo que refleja una interacción entre las características del organismo humano y las características de la sociedad en la que vive.

Por su parte, la discapacidad visual se refiere a personas con deficiencias funcionales del órgano de la visión y de las estructuras y funciones asociadas, incluidos los párpados (OMS, 2020).La discapacidad visual puede limitar a las personas en la realización de tareas cotidianas y afectar su calidad de vida, así como sus posibilidades de interacción con el mundo circundante.La ceguera, la forma más grave de discapacidad visual, puede reducir la capacidad de las personas para realizar tareas cotidianas.Existen diferentes tipos de discapacidad visual según el grado de afectación a la capacidad de ver. Esta significación puede implicar desde una visión parcial hasta una ceguera o impedimento visual completo.

Así como la iluminación, el contraste y la ampliación de las imágenes ayudan a los discapacitados visuales a desarrollarse, existen instrumentos que facilitan el acceso a la información por este tipo de individuos: la pauta, la regleta, la máquina Perkins, entre otras(AGI, 2016).Al propósito de facilitar el acceso a la información de personas con algún tipo

de discapacidad visual también se suman las TIC; ejemplo de ello son las siguientes adaptaciones para ordenadores (AGI, 2016):

- **Síntesis de voz:** es una adaptación específica para ciegos, con salidas de voz.
- **Línea braille:**es un dispositivo que conectado al ordenador permite al ciego la lectura de la información que ofrece la pantalla del ordenador a través de una línea con células braille donde se transfiere el contenido de la pantalla línea a línea.
- **Lectores de pantalla:**existen programas que convierten el texto en voz sintetizada para que los ciegos sean capaces de escuchar los contenidos web. Estos programas son generalmente llamados lectores de pantalla, a pesar de que hacen algo más de lo que su nombre indica. Los lectores de pantalla permiten a los usuarios navegar a través de contenidos web de muchas maneras.

1.2 Estudio de sistemas similares u homólogos

Una vez realizada la investigación de los principales conceptos, se realiza un análisis de soluciones informáticas que guardan relación con el dominio del problema,con el objetivo de definir sus características y la viabilidad de uso en el contexto nacional. Para ello, se parte de la búsqueda en plataformas nacionales para la distribución de aplicaciones móviles, en las cuales no se pudo constatar la existencia de aplicaciones para discapacitados visuales que permitan consultar información notarial.

Usando otras plataformas como Google Play[1], reconocida a nivel internacional, tampoco se encontraron aplicaciones con funciones similares; no obstante, se estudiaron varias de estas solucionesorientadas a personas con algún tipo de discapacidad visual, independientemente de su objetivo y/o función:

Mobile Accesibility: aplicación Android[2]diseñada para personas ciegas que les permite usar su teléfono de una manera intuitiva, fácil y simple. Se trata de dos productos en uno pues es un conjunto de 10 aplicaciones (Teléfono, Contactos, SMS, Alarma, Calendario, Email, Web,

[1] Google Play es la tienda de aplicaciones creada por Google donde puedes encontrar juegos, películas, música, libros y más. Está disponible para cualquier dispositivo móvil que cuente con sistema operativo Android.

[2]Android es un sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil. De este modo, es posible encontrar tabletas (tablets), teléfonos móviles (celulares) y relojes equipados con Android, aunque el software también se usa en automóviles, televisores y otros dispositivos.

Dónde estoy, Aplicaciones y Ajustes) y también es un lector de pantallas para facilitar la navegación por la interfaz del móvil(Saez, 2014).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra informaciónsobre servicios notariales y requiere de internet para su utilización.

Dragon Dictation: es una aplicación para iPhone[3] y Android de reconocimiento de voz que permite dictar y ver al instante el mensaje o correo electrónico que se desea enviar. De hecho, es hasta cinco veces más rápido que escribir con el teclado(Saez, 2014).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra información sobre servicios notariales.

Talkback: herramientaAndroid pensada para que las personas con discapacidad visual puedan navegar por las aplicaciones y herramientas para así favorecer un uso independiente y autónomo del smartphone. Cuando TalkBack está activado, el dispositivo habla y describe cada uno de los elementos que son seleccionados o activados(Saez, 2014).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra información sobre servicios notariales y requiere internet para su utilización.

BrailleBack: es un servicio de accesibilidadAndroid complementario que ayuda a los usuarios invidentes a utilizar los dispositivos braille. Se utiliza junto con la aplicación TalkBack antes mencionada para ofrecer un servicio combinado de voz y sistema braille. Esta aplicación permite conectar una pantalla braille a través de Bluetooth e introducir texto con el teclado braille(Saez, 2014).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra información sobre servicios notariales y requiere internet para su utilización.

Voice Access: aunque se creó como ayuda a personas con movilidad reducida es una de las aplicaciones básicas paraciegos. Esta facilita utilizar un teléfono móvil Android usando comandos por voz. Así, se facilita que el anciano con problemas de visión utilice su dispositivo

[3] iPhone es una línea de teléfonos inteligentes de alta gama diseñada y comercializada por Apple Inc. Ejecuta el sistema operativo móvil iOS.

móvil; no solo para hacero recibir llamadas, también para acceder a otras aplicaciones para personas con problemas de visión o ceguera (assistiva, 2019).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra información sobre servicios notariales y requiere internet para su utilización.

Linguoo: esta aplicación Android es especialmente interesante para personas ciegas coninterésporlaactualidad y la cultura. También es adecuada para todo anciano sin visión que requiera distracciones durante su tiempo libre en casa(assistiva, 2019).

Los autores reconocen la utilidad de esta aplicación para personas con discapacidad visual, pero esta no muestra información sobre servicios notariales y requiere internet para su utilización.

Los elementos antes mencionados se resumen en una tabla comparativa:

Tabla 1. Tabla comparativa sobre aplicaciones estudiadas.

Aplicaciones	Accesible por discapacitados visuales	Requiere internet	Permite consultar servicios notariales
Mobile Accesibility	Sí	Sí	No
Dragon Dictation	Sí	No	No
Talkback	Sí	Sí	No
BrailleBack	Sí	Sí	No
Voice Access	Sí	Sí	No
Linguoo	Sí	Sí	No

Fuente: elaboración propia.

El procesamiento de esta información permitió constatar la carencia de aplicaciones móviles que se adecuen al contexto y dominio del problema planteado; además en su mayoría requieren de conexión a internet para el acceso a sus funcionalidades.Sin embargo, se

identificaron elementosen las herramientas nombradas que se deben tomar en cuenta en el desarrollo de soluciones informáticas para discapacitados visuales,entre ellos:

- El acceso a la aplicación debe ser inmediato o con el menor recorrido posible desde el arranque, y la salida de la misma, sencilla, aunque haya de ser verificada.
- No usar el texto para construir gráficos. No utilizar los caracteres para la creación de gráficos. Los lectores de pantalla interpretarán estos gráficos como texto y harán una lectura errática.
- Las aplicaciones deben proporcionar opciones para que el usuario elija el tipo de letra, su tamaño y el color de todos los controles de la interfaz. Dentro de las características del texto se incluyen también la fuente y el estilo (cursiva, negrita, etc.).
- El diseño de la interfaz gráfica de usuario debe permitir que los elementos que la componen, principalmente el texto y los controles, sigan siendo visibles y navegables cuando se modifican su tamaño.
- Deben proporcionarse combinaciones de colores predefinidas que hayan sido diseñadas teniendo en cuenta las necesidades de las personas con diversidad funcional visual.
- El usuario debe poder ajustar el volumen del sonido de la aplicación. Este requisito tiene relación con la funcionalidad del propio sistema operativo y del dispositivo.
- La lectura de texto es adecuada para personas con problemas de visión y también para aplicaciones de ámbito educativo.
- La aplicación debe proporcionar controles de reproducción para reproducir, pausar y parar. Cuando no se proporcionan estos controles y el contenido multimedia se activa por defecto automáticamente, los usuarios de lectores de pantalla pueden tener dificultades para controlar la información de salida que simultáneamente se produce en la interfaz.

1.3 Metodología de desarrollo

Según Sommerville, una metodología de desarrollo es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos(Sommerville, 2017).En el desarrollo de software la necesidad de organizar o estructurar de forma correcta y disciplinada, es uno de los factores más importantes

para evitar pérdidas de tiempo y recursos. Con el fin de prevenir tales errores es preciso definir una estrategia que ordene las posibles tareas a desarrollar, así como llevar a cabo una guía de cómo efectuar las actividades, procedimientos y pasos que se deben de seguir para el desarrollo de un software(Cano, 2003).

En el caso de la presente investigación, el proceso es guiado aplicando la metodología definida por la UCI para organizar su proceso de desarrollo de software,la cual es una variación del AUP, (por sus siglas en*Agile UnifiedProcess)*. AUP-UCI se fundamenta en las características propias de las metodologías de adaptarse a las necesidades de cada proyecto (equipo de desarrollo y recursos).

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, teniendo en cuenta el modelo de mejora de procesos de desarrollo de software (CMMI-DEV v1.3). El mismo constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad(Sánchez, 2015).

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación(Sánchez, 2015).

AUP establece cuatro fases que transcurren de manera consecutiva:

- **Inicio:**el objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:**el objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:**durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.
- **Transición:**el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

De igual forma se definen en la metodología 4 escenarios para la disciplina de Requisitos. Para el caso de la presente investigación se selecciona el escenario 4,pues el mismoaplica a la presente investigación,pues se ha evaluado el negocio a informatizar y como resultado se obtiene un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, como el que se debe desarrollar.

1.4 Herramientas, tecnologías y lenguajes

1.4.1 Herramienta CASE

Las herramientas CASE (por sus siglas en inglés *ComputerAided Software Engineering*),constituyen aplicaciones o programas informáticos destinados a aumentar el balance en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas sirven de ayuda en todos los aspectos del ciclo de vida del software en tareas como el diseño de proyectos, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras(Ambler, 2018).

En la presente investigación se propone utilizar como herramienta CASE profesional**Visual Paradigm**, que utiliza UML *(*por sus siglas en inglés*UnifiedModelingLanguage)* como lenguaje de modelado y que soporta el ciclo de vida del software, además de tener la ventaja de ser multiplataforma (Gaines, y otros, 2017). A continuación, se listan algunas características de la misma:

- Ofrece amplias características de modelado de casos de uso incluyendo la función completa de UML, diagrama de casos de uso y editor de flujo de eventos.
- Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.
- Proporciona abundantes tutoriales del Lenguaje de Modelado, demostraciones interactivas de UML y proyectos UML.
- Se integra con herramientas Java, como: Eclipse/IBM, NetBeansIDE, entre otras.
- Es apoyado por un conjunto de lenguajes tanto en la generación del código como en la Ingeniería Inversa(Gaines, y otros, 2017).

Lenguaje de modelado unificado UML

Lenguaje de modelado unificadoUML en su versión 2.0 se usa para especificar, visualizar,

construir y documentar artefactos de un sistema de software, no define un proceso de desarrollo específico, tan solo se trata de una notación. Se utiliza para detallar los artefactos en el sistema y definir un sistema de software (Pressman, 2010).El análisis de este lenguaje se realiza debido a que UML es la notación utilizada por la herramienta CASE que se emplea para la creación de los componentes.

1.4.2AndroidStudio

Es el Entorno de desarrollo integrado (IDE) proporcionado por Google para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio en su versión 3.6.1 ofrece funciones que aumentan la productividad durante la compilación de aplicaciones para Android, como las siguientes (Studio, 2016)**:**

- Sistema de compilación flexible basado en Gradle[4].
- Emulador rápido con varias funciones.
- Entorno unificado en el que se puede realizar desarrollos para todos los dispositivos Android.
- Aplicar cambios mientras la aplicación se ejecuta sin la necesidad de compilar una nueva aplicación.
- Integración de plantillas de código y GitHub, para ayudar a compilar funciones comunes de las aplicaciones e importar ejemplos de código.
- Gran cantidad de herramientas y marcos de trabajo de prueba.
- Herramientas para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.

Por todas las ventajas antes descritas se propone utilizar Android Studio y de esta forma garantizar la compatibilidad con iLex Notario y el resto de las aplicaciones de la suite iLex.

1.4.3Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el

[4]Gradle es un sistema de compilación que reúne las mejores prestaciones de otros sistemas de compilación. Está basado en JVM (Java Virtual Machine), lo que significa que puedes escribir tu propio script en java, y que Android Studio lo entenderá y lo usará.

programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra (MX, 2016). Fue desarrollado por Sun Microsystems[5], posteriormente adquirido por Oracle[6].

En la actualidad puede utilizarse de modo gratuito, pudiéndose conseguir sin problemas un paquete para desarrolladores que oriente la actividad de programar en este lenguaje. Puede ser modificado por cualquier usuario, circunstancia que lo convierte en lo que comúnmente se denomina “código abierto”. En la actualidad, este lenguaje de programación ha ganado una enorme popularidad como consecuencias de su portabilidad, simpleza y grandes posibilidades de utilización (MX, 2016).

Para el desarrollo de la propuesta de solución se propone el uso de este lenguaje y de esta forma garantizar la compatibilidad con iLex Notario y el resto de las aplicaciones de la suite iLex.

1.4.4Kotlin

Lenguaje oficial de Android creado como alternativa a Java, es un lenguaje de programación que Google ha colocado en primera línea para desarrollar Android en el futuro. Creado por JetBrains, se destaca por sus entornos de desarrollo integrado o IDEs para SQL, Java, JavaScript, Python y otros lenguajes. Entre sus más destacados productos, sobresalen IntelliJ IDEA o DataGrip. En el 2011, JetBrains decidió crear su propio lenguaje de programación, Kotlin, compatible con la máquina virtual Java, compilable a JavaScript y cuyo mayor logro ha sido que Google lo haya incluido entre los lenguajes de programación oficiales de Android, lista en la que hasta ahora solo figuraban Java y C++ (López, 2019).

Se propone la utilización de este lenguaje en su versión 1.3.70 junto a Java por ser el lenguaje oficial que utiliza Android para el desarrollo de sus aplicaciones y garantizar la compatibilidad con iLex Notario y el resto de las aplicaciones de la suite iLex.

1.4.5 Android SDK

SDK (por sus siglas en inglés *Software development kit*)o paquete de desarrollo de software, desde él se pueden emular softwares desarrollados para otras plataformas y es la base para el

[5]Sun Microsystems fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos.
[6]Oracle Corporation es una compañía de software que desarrolla bases de datos (OracleDatabase) y sistemas de gestión de bases de datos.

desarrollo de aplicaciones para la plataforma de Google. Este paquete o kit de desarrollo incluye las API (por sus siglas en inglés *ApplicationProgramming Interface*) Interfaz de Programación de Aplicaciones y herramientas necesarias para desarrollar las aplicaciones, utilizando Java como lenguaje de programación(Cepeda, 2014).

Para el desarrollo de la propuesta de solución se selecciona este paquete de desarrollo de software y así garantizar la compatibilidad con iLex Notario y el resto de las aplicaciones de la suite iLex.

1.4.6ProGuard

Para optimizar y modificarelcódigo Java una vez compilado se propone a utilizar la herramienta ProGuard. En una primera fase elimina las clases y métodos no utilizados. Luego optimiza el código resultante y por último lo ofusca[7]renombrando clases, métodos y campos con nombres poco legibles.Puede utilizarse mediante el típico asistente, invocarse desde línea de comandos (Guardsquare, 2016-2019).Es multiplataforma y tiene Licencia Pública General(GPL). Hace que sus aplicaciones Java y Android sean más pequeñas y más rápidas. También proporciona una protección mínima contra la ingeniería inversa al ofuscar los nombres de las clases, campos y métodos. Es usado de forma gratuita para procesar sus aplicaciones, comerciales o no comerciales. (Guardsquare, 2016-2019).

Por todas las ventajas antes descritas y para garantizar la compatibilidad con iLex Notario y el resto de las aplicaciones de la suite iLex,se decide utilizar ProGuard.

1.4.7 Notación de Objetos de JavaScript (JSON)

La Notación de Objetos de JavaScript JSON (por sus siglas en inglés *JavaScript ObjectNotation*) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript[8]. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java,

[7] Ofuscar, en el contexto del desarrollo de aplicaciones móviles, es la modificación que se realiza sobre el código fuente para que resulte más complicado de leer e interpretar por terceras personas.
[8]JavaScript (JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

JavaScript, Perl, Python, y otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos, constituido por dos estructuras(JSON, 2016):

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras(JSON, 2016).

Es utilizado para almacenar los datos de la aplicación iLex Notario en su versión inicial que luego estos serán enviados al componente desarrollado en la investigación.

1.5 Conclusiones del capítulo

El estudio de los principales conceptos asociados al dominio del problema facilitó la mejor comprensión del presente trabajo. El análisis comparativo de soluciones informáticas en el contexto nacional e internacional permitiódefinir las características yfuncionalidades necesarias a tener en cuenta para desarrollar el componente, lo que ratificó la necesidad de crear una solución para las necesidades de los discapacitados visuales en el contexto nacional.

Por otra parte, la selecciónde la metodología AUP en su variación para la UCIpone en línea la presente solución con las políticas que sigue la universidad para obtener productos demayor calidad que satisfagan las necesidades del cliente.Las tecnologías, herramientas y lenguajes estudiados, garantizan la compatibilidad con iLex Notario y otras aplicaciones de la suite iLex y el cumplimiento de las políticas de soberanía tecnológica de la UCI y el país.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

En el presente capítulo se realiza la descripción de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales, obteniéndose la base de la arquitectura del componente, así como particularidades del diseño e implementación. A partir de los requisitos identificados se modelan los diagramas de clases del diseño los cuales permiten obtener una visión más clara de la implementación. Además, se obtiene el diagrama de componentes que representa la relación y las dependencias entre los mismos.

2.1 Descripción de la propuesta de solución

El componenteque se propone tiene como objetivo brindar a los discapacitados visuales una solución que facilite la consulta de información de interés sobre las notarías, dígase:

- Conocer una breve reseña sobre la historia de las notarías en Cuba.
- Conocer los servicios que brindan las notarías del país.
- Conocer los tipos de documentos que se redactan en las notarías.
- Tener acceso a una lista de preguntas frecuentes que pudieran esclarecer dudas.
- Conocer los deberes y derechos de los notarios y de los ciudadanos.
- Conocer toda la legislación de las notarías de la República (Código de ética del notario, Ley No. 50-48 de las notarías estatales, Reglamento de la ley de las notarías, Instrucción conjunta No.1, Resolución No. 250, Resolución No. 130).
- Tener acceso a un directorio delas notarías por provincias y por municipios, así como las unidades notariales pertenecientes a las sociedades civiles de derecho jurídico.
- Conocer información de contacto de la dirección de notarías del MINJUS y enviar email a los jefes de departamentos de las notarías provinciales.

El proceso comienza cuando el usuario instala el componente y recibe una estructura de datos con toda la información de la aplicación iLex Notario, adaptándola a las necesidades funcionales de los discapacitados visuales. Este nuevo componente cuenta con una pantalla principal que tiene todas las opciones de la versióninicial de iLex Notario, pero con una nueva estructura, visualizada con botones cuadrados y de tamaño ajustable por el usuario.

Si el usuario accede a la pantalla de lectura, el componente facilita la reproducción del audio correspondiente al texto seleccionado, con las opciones de pausado,detención y acceso a la pantalla anterior. La aplicación cuenta con una pantalla de configuración que permite regresar a la versión inicial de iLex Notario, quitar el audio y cambiar el tamaño o el tipo de letra de lectura. En la siguiente ilustración se muestra la transformación que experimenta la información que se brinda al usuario después de instalado el componente.

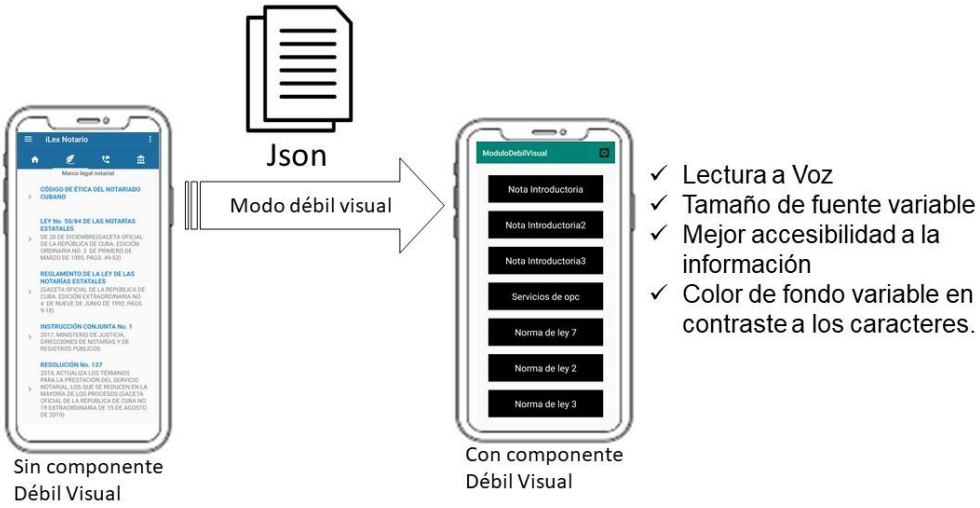


Ilustración 1. Propuesta de solución.

Fuente: elaboración propia.

2.2 Requisitos del software

Un requisito es la condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo. También se define como la condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación, u otra documentación formalmente impuesta(Sanchez, 2018).

2.2.1Técnicas de levantamiento de requisitos de software

El levantamiento de requisitos se refiere a la identificación y documentación de los requerimientos de un sistema, a partir de los usuarios, clientes o interesados. A la práctica también se le conoce como recopilación de requerimientos (Oficina de Proyectos de

Informática, 2018). Luego de un análisis de las técnicas existentes para darle solución al problema planteado se decide el empleo de las siguientes técnicas:

Análisis de documentación: consiste en obtener la información sobre los requerimientos funcionales y requerimientos no funcionales de software a partir de documentos que ya están elaborados(Oficina de Proyectos de Informática, 2018).Esta técnica se utilizó en el procesamiento de la documentación de la solución informática iLex Notario.

Entrevistas: esta técnica posee las siguientes características(Oficina de Proyectos de Informática, 2018):

- Se realiza con los usuarios o el cliente.
- Direccionan al usuario hacia aspectos específicos del requerimiento a levantar.
- Pueden ser entrevistas formales o informales.
- Las preguntas abiertas son útiles para identificar información faltante.
- Las preguntas cerradas son útiles para confirmar y validar información.

El éxito de las entrevistas depende del grado de conocimiento del entrevistador y entrevistado, disposición del entrevistado de suministrar información, buena documentación de la discusión y en definitiva de una buena relación entre las partes.Esta técnica se utilizó para identificar los requisitos funcionales; fundamentalmente con directivos del MINJUS y usuarios potenciales de la aplicación(Ver Anexo 6).

Tormenta de ideas: es una sesión de trabajo estructurada orientada para obtener la mayor cantidad de ideas posibles. Las reglas son importantes, por ejemplo, los criterios para evaluar ideas y asignarles un puntaje, no permitir las críticas a las ideas y limitar el tiempo de discusión. En una primera fase, se deben identificar la mayor cantidad de ideas, para luego evaluarlas (Oficina de Proyectos de Informática, 2018).Esta técnica fue empleada por los autores(equipo de proyecto) para recopilar la mayor cantidad de información posible parael desarrollo de la aplicación.

2.2.2Requisitos funcionales

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio(Pressman, 2010).

Los autores de la presente investigación, identificaron 17 (RF), en la Tabla 2 se muestra la especificación de estos requisitos.

Tabla 2. Requisitos funcionales (RF).

No.	Nombre	Descripción
RF 1.	Crear la estructura de los datos en la aplicación iLex Notario.	El componente permitirá mostrar los datos enviados desde la aplicación de prueba para débiles visuales.
RF 2.	Habilitar/Deshabilitar modo "Débil Visual" en la aplicación iLex Notario.	El componente permitirá habilitar o deshabilitar las interfaces para débiles visuales en la aplicación según se seleccione en la aplicación iLex Notario.
RF 3.	Visualizar pantalla de inicio para débil visual.	El componente permitirá mostrar la pantalla de inicio con botones cuadrados de 24 píxel.
RF 4.	Acceder a la pantalla de configuración.	El componente permitirá tener acceso a la configuración desde la pantalla principal.
RF 5.	Cambiar el tamaño de letra	El componente permitirá cambiar el tamaño de letra de las pantallas según requiera el usuario.
RF 6.	Cambiar el tipo de letra.	El componente permitirá cambiar el tipo de letra de las pantallas según requiera el usuario.
RF 7.	Cambiar el color de la letra.	El componente permitirá cambiar el color de la letra de las pantallas según requiera el usuario.
RF 8.	Acceder a la pantalla principal desde otras pantallas.	El componente permitirá acceder a la pantalla principal desde cualquier pantalla.
RF 9.	Acceder a la pantalla anterior.	El componente permitirá retroceder a la pantalla anterior desde la pantalla en la que se encuentre

		el usuario.
RF 10.	Visualizar pantalla de servicios notariales.	El componente permitirá mostrar la pantalla de servicios notariales con una lista de botones a lo largo de la pantalla.
RF 11.	Visualizar pantalla de lectura.	El componente permitirá mostrar el artículo seleccionado en la pantalla de lectura genérica.
RF 12.	Visualizar pantalla de directorio.	El componente permitirá mostrar la pantalla de directorio para tener acceso a los datos de las entidades notariales.
RF 13.	Reproducir lectura de textos.	El componente permitirá iniciar la lectura a voz en la pantalla de lectura.
RF 14.	Pausar lectura de textos.	El componente permitirá pausar la lectura de voz en la pantalla de lectura y al iniciar continúa su lectura en el texto pausado.
RF 15.	Detener lectura de textos.	El componente permitirá detener la lectura de voz en la pantalla de lectura y al iniciar lectura comienza en el texto inicial.
RF 16.	Leer nombre del botón seleccionado.	El componente permitirá leer el título del botón seleccionado en la pantalla para disminuir el esfuerzo de la vista.
RF 17.	Ajustar el volumen del sonido de la aplicación.	El componente permitirá ajustar el volumen como funcionalidad del propio sistema operativo y del dispositivo.

Fuente: elaboración propia.

2.2.3 Requisitos no funcionales

Los requisitos no funcionales(RnF) representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir (Pressman, 2010).

Restricciones deentorno:

RnF 1. Requisito deentorno 1

La aplicación será desarrollada para el sistema operativo Android.

Restricciones de eficiencia:

RnF 2. Requisito de eficiencia1

La versión de sistema operativo tendrá que ser 4.0 o superior.

RnF 3. Requisito de eficiencia2

El dispositivo deberá tener un procesador ARMv6 a 800 MHZ o superior.

RnF 4. Requisito de eficiencia3

El dispositivo móvil deberá tener disponible una memoria RAM mínimo de 256 MB y una memoria ROMmínimo de 256 MB.

RnF 5. Requisito de eficiencia4

El dispositivo móvil deberá tener disponible un espacio de almacenamiento mínimo de 50 MB.

RnF 6. Requisito de eficiencia5

El dispositivo debe tener instalado un paquete de audio en idioma español y seleccionado como lector principal.

Restricción de desarrollo:

RnF 7. Requisito de desarrollo 1

Elcomponente se debe desarrollar utilizando el lenguaje de programación Java y Kotlin.

Usabilidad:

RnF 8. Requisito de usabilidad 1.

En el componente se deben visualizar todos los mensajes en idioma español. La tipografía debe ser uniforme y de tamaño 24 píxel para los débiles visuales.

RnF 9. Requisito de usabilidad 2.

El componente debe ofrecer una interfaz amigable, fácil de operar para los discapacitados visuales. Igualmente tiene que mantener la línea de diseño establecida la cual mantiene la uniformidad y representatividad de la solución.

RnF 10. Requisito de usabilidad 3.

Las interfaces deben poseer un diseño sencillo, con la paleta de colores necesaria, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización del mismo.

Confiabilidad:

RnF 11. Requisito de confiabilidad 1.

El componente debe estar disponible para ser descargado en cualquier momento de la plataforma Apklis.

RnF 12. Requisito de confiabilidad 2.

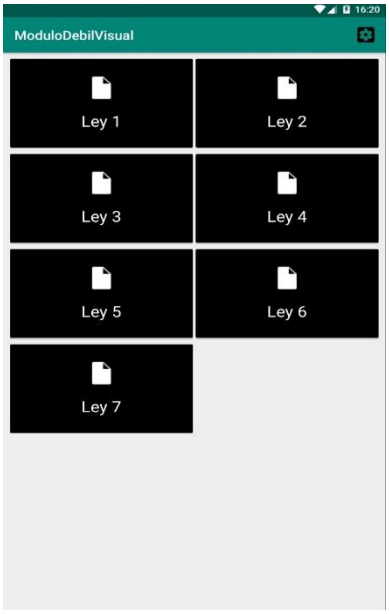
La información debe ser confiable, emitida por la Dirección de Notarías del MINJUS.

2.3 Historias de usuario (HU)

Constituyen descripciones cortasyesquemáticas que resumen la necesidad concreta de un usuario al utilizar un producto o servicio, así como la solución que la satisface. Su función principal es identificarproblemaspercibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas(Solvingadhoc, 2017).Para la propuesta de solución se definieron 17 historias de usuario, de las cuales se presentan, a manera de ejemplos, las asociadas a los requisitos funcionales 3 y 11.

Tabla 3. HU Visualizar pantalla de inicio para débil visual.

Número: 3	**Requisito:** Visualizar pantalla de inicio para débil visual.
Programador: Wendy Lara González	**Iteración Asignada:** 1
Prioridad: alta	**Tiempo Estimado:** 2 semanas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 10 días
Descripción: funcionalidad que permitirá mostrar la pantalla de inicio con botones cuadrados de 24 píxel. **Procedimiento:** campo de selección que se activa después de estar en la pantalla inicial para acceder a la opción que el usuario desea consultar.	

Botones: **Configuración:** permite acceder a la pantalla de configuración.
Observaciones: al enviar los datos de la aplicación iLex Notario la aplicación muestra la pantalla principal con las opciones deseadas en botones cuadrados de 24 píxel.


Fuente: elaboración propia.

Tabla 4. HU Visualizar pantalla de lectura.

Número: 11	**Requisito:**Visualizar pantalla de lectura.
Programador: Eddy González Escandell	**Iteración Asignada:** 1
Prioridad: alta	**Tiempo Estimado:** 2 semanas
Riesgo en Desarrollo: N/A	**Tiempo Real:** 9 días
Descripción: funcionalidad que permite mostrar el artículo seleccionado en la pantalla de lectura genérica. **Procedimiento:** campo de selección que se activa después de estar en la pantalla de	

lectura para comenzar a reproducir los textos.

Botones:

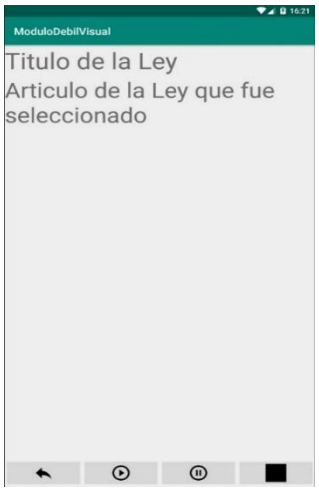
Reanudar: permite reproducir los textos de la pantalla de lectura.

Pausar: permite pausar la lectura de los textos.

Detener: permite detener y empezar al inicio la lectura de los textos.

Retroceder: permite retroceder a la pantalla anterior para realizar otra selección.

Observaciones: al seleccionar la pantalla de lectura comienza la lectura a voz de los textos de la aplicación. No es obligatorio.



Fuente: elaboración propia.

2.4 Arquitectura del componente

El diseño arquitectónico es la primera etapa en el proceso de construcción del software. Constituye el enlace crucial entre el diseño y la ingeniería de requisitos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos. Su salida consiste en un modelo que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación (Sommerville, 2011).

Se propone para el desarrollo de la investigación la utilización del patrón arquitectónico Modelo-Vista-Presentador (MVP), con el objetivo de simplificar el código y tener organizado el proyecto. MVP es un patrón arquitectónico que pretende codificar la interfaz de usuario lo más simple posible, teniendo el menor código. En su lugar, toda la lógica de la interfaz de usuario, se hace en una clase separada (que se conoce como Presentador), que no dependa en absoluto de los componentes de la interfaz gráfica y que, por tanto, es más fácil de realizar pruebas. Este patrón está compuesto por 3 elementos fundamentales:

- **La vista:**es la interfaz gráfica del usuario. Está compuesta por las ventanas y controles que forman la interfaz de la aplicación. Algunos de los elementos que componen la vista son:ConfiguracionActivity,LecturaActivity y MainActivity.
- **El modelo:**es donde se encuentran las clases de acceso a datos de la lógica del negocio. Ejemplos de algunas de estas clases son: Directorio, Telef, Opcion y Texto.
- **El presentador:**es una capa intermediaria entre la vistay el modelo de datos. Este permite conectar la interfaz gráfica con los datos, que los recupera del modelo y los devuelve a la vista formateados. En esta se encuentran:AdapterGridView y LectorDeTexto.

A continuación, se muestra la arquitectura de la propuesta de solución, la cual está en correspondencia con los elementos antes mencionados.

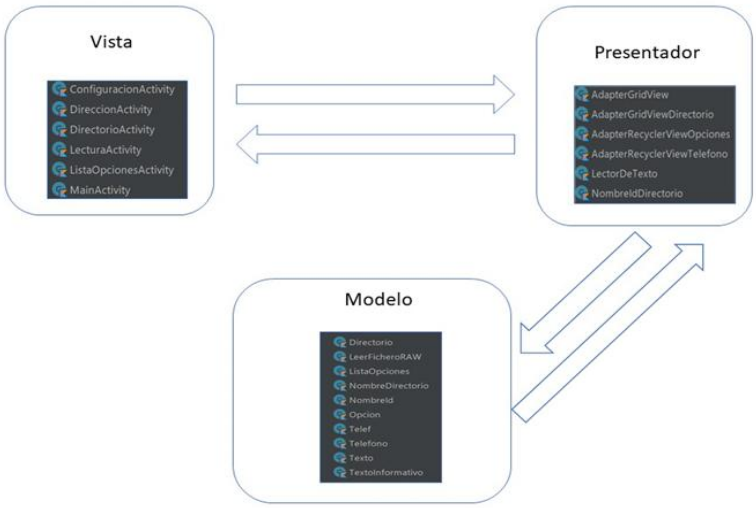


Ilustración 2. Arquitectura de la propuesta de solución.

Fuente: elaboración propia.

2.5 Patrones de diseño

Los patrones de diseño son técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias(Pressman, 2010).Existen clasificaciones para estos patrones entre los que se encuentran los Patrones de Software para Asignar ResponsabilidadesGRASP (por sus siglas en inglés *General ResponsibilityAssignment Software Patterns*) y los Patrones del Grupo de los CuatroGoF (por sus siglas en inglés *GangofFour)*. A continuación, se presentan los patrones de diseño que son utilizados para el desarrollo de la solución.

2.5.1 Patrones GRASP

Los patrones GRASP son una ayuda en el aprendizaje que permiten al desarrollador entender lo esencial del diseño de objetos y aplicar el razonamiento de una forma metódica, racional y explicable (Pressman, 2010). A continuación, se describen los utilizados en la propuesta de solución:

Patrón Controlador: sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado(Larman, 2016). Este patrón se evidencia en la solución a través de las clases controladorasAdapterGridView.kt, AdapterGridViewDirectorio.kt, AdapterRecyclerViewOpciones.kt, AdapterRecyclerViewTelefono.kt yLectorDeTexto.kt, NombreIdDirectorio.kt.

Patrón Experto: es el principio básico de asignación de responsabilidades. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada(Larman, 2016).Este patrón se pone en práctica en las clases entidades que son las expertas en información y las encargadas de manejar la lógica del negocio que comprenden cada uno de

los conceptos que se engloban en la propuesta de solución, ejemplos de estas clases son Teléfono.kt, Directorio.kt y TextoInformativo.kt.

Patrón Creador:ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases(Larman, 2016). Este patrón se evidencia en la propuesta de solución en las clasesConfiguracionActivity.kt, DireccionActiviy.kt, DirectorioActivity.kt, LecturaActivity.kt, ListaOpcionesActivity.kt y MainActivity.kt.

Patrón Bajo acoplamiento:este patrón se basa en la baja dependencia que debe existir entre las clases, además está estrechamente relacionado con los patrones Experto o Alta Cohesión(Larman, 2016), puesto que a cada clase se le asignan solamente las responsabilidades necesarias de manera que no dependan en gran medida de otras; por lo que también se encuentra presente en la propuesta de solución.

Patrón Alta cohesión:este patrón tiene como propósito asignar responsabilidades de manera que la cohesión siga siendo alta(Larman, 2016).Se evidencia a través de la interrelación que existe entre las clases controladora, las clases gestoras y las funcionalidades del componente, la primera encargada de manejar la lógica de presentación y el flujo de los datos provenientes de la vista y la segunda encargada de manejar la lógica del negocio de cada funcionalidad.

2.5.2 Patrones GoF

Los patrones GoF son soluciones concretas y técnicas basadas en la Programación Orientada a Objetos (POO). Se utilizan en situaciones frecuentes, debido a que se basan en la experiencia acumulada al resolver problemas reiterativos. Ayudan a construir software basado en la reutilización y a construir clases reutilizables (Pérez, 2013).A continuación, se describen los utilizados en la propuesta de solución:

Patrón Decorador: su principal objetivo es añadir responsabilidades a objetos concretos de manera dinámica y transparente sin afectar a otros objetos. Este patrón brinda más flexibilidad que la herencia estática y evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas(Gamma, y otros, 1994).Este patrón se evidencia en las clases a las que se añaden funcionalidades por ejemplo AdapterGridView.kt y AdapterGridViewDirectorio.kt.

Patrón Observador: permite observar los cambios producidos por un objeto. Es uno de los principales patrones de diseño utilizados en interfaces gráficas de usuario, ya que permite desacoplar al componente gráfico de la acción a realizar (Larman, 2016).Este patrón es

empleadoal enviar los datos de la vista al presentador, este último hace la petición de un servicio y se queda a la espera de la respuesta,que este servicio debe proveer para así emitir o no un evento.

Patrón Fabricación pura: se da en las clases que no representan un ente u objeto real del dominio del problema, sino que se ha creado intencionadamente para disminuir el acoplamiento, aumentar la cohesión y/o potenciar la reutilización del código. Las clases de fabricación pura casi siempre se dividen atendiendo a su funcionalidad, dicho con otras palabras, se confeccionan clases destinadas a conjuntos de funciones(Larman, 2016).Generalmente se considera que la fabricación es parte de la capa de servicios orientada a objetos de alto nivel en una arquitectura. Este patrón se evidencia en la clase LectorDeTexto.ktque posee uno de los servicios de la aplicación.

2.6 Diagrama de paquetes

Se utiliza en el desarrollo de la investigación con el objetivo de obtener una visión más clara del sistema de información orientado a objetos, organizándolo en subsistemas, agrupando los elementos del análisis, diseño o construcción y detallando las relaciones de dependencia entre ellos. El mecanismo de agrupación se denomina Paquete(Cillero, 2009-2020).A continuación, se muestra el diagrama de paquetes de la propuesta de solución.

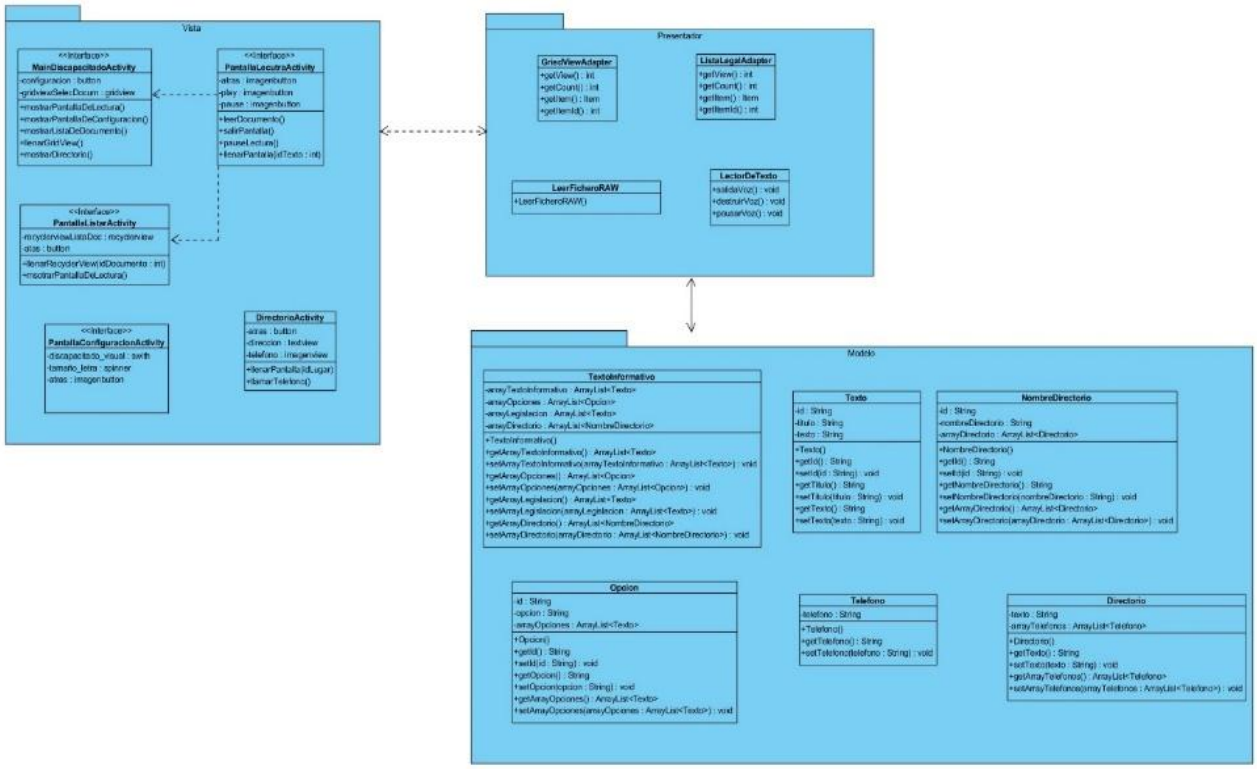


Ilustración 3. Diagrama de paquetes de la propuesta de solución.

Fuente: elaboración propia.

2.7 Estándar de codificación

Los estándares de codificación recomiendan estilos de programación, buenas prácticas y métodos para mantener el aspecto del código fuente. Incluyendo la organización de archivos, la identificación, los comentarios, las declaraciones, los espacios en blanco, las llaves de apertura y cerrado (Vega, 2017).

Para la implementación del componente se utilizó la nomenclaturaCamelCase, la cual es una convención de nomenclaturas en la que cada palabra compuesta se escribe con mayúscula, excepto la primera palabra (TechTerms, 2019).Esta nomenclatura es recomendada por Java, representa la unión entre UpperCamelCase (PascalCase) y lowerCamelCase (camelCase). La primera para la notación de clases y la segunda para variables y métodos.

Clases e interfaces:la inicial en mayúscula ya sea simple o compuesto el nombre de la clase.Ejemplo:

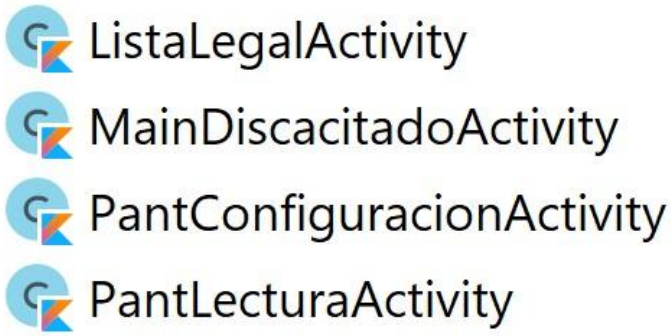


Ilustración 4. Declaración de clases

Fuente: elaboración propia.

Métodos: la primera letra de la primera palabra en minúsculas, el resto de las palabras empiezan por mayúsculas.Ejemplo:

```
override fun onCreateOptionsMenu(menu: Menu): Boolean {...}

override fun onOptionsItemSelected(item: MenuItem): Boolean {...}
```

Ilustración 5. Declaración de métodos

Fuente: elaboración propia.

Variables: deben comenzar por minúscula y no contener caracteres extraños como tildes o guiones bajos.Ejemplos:

```
private lateinit var textArticulo: TextView
private lateinit var butonAtras: ImageButton
```

Ilustración 6. Declaración de variables.

Fuente: elaboración propia.

2.8 Conclusiones del capítulo

El desarrollo de los artefactos propuestos por la metodología AUP-UCI permitió organizar el desarrollo y obtener una solución acorde a las necesidades de los clientes. El levantamiento de requisitos permitió identificar los requerimientos para darle solución al problema planteado.La arquitectura del componente posibilitó codificar la interfaz lo más simple posible y tener organizado el proyecto.

Por otra parte, los patrones de diseño garantizaron la obtención de una solución con poca dependencia entre clases, flexible al mantenimiento y a la introducción de cambios. De igual manera, elempleo de los estándares de codificación facilitó la lectura, comprensión y mantenimiento del código por parte de los desarrolladores.

CAPÍTULO 3: VALIDACIÓN Y PRUEBA

En este capítulo se evalúa el grado de calidad y fiabilidad de los requisitos obtenidos; se abordan las métricas aplicadas para validar el diseño de la solución. Además, se aplican pruebas de software para verificar y revelar la calidad de la solución propuesta antes de ser entregada al cliente.

3.1 Técnicas de validación de requisitos

Con el objetivo de ratificar que los requisitos del software obtenido definen la solución que el cliente necesita, se llevó a cabo un proceso de validación de los mismos, empleando las siguientes técnicas:

- **Revisiones de los requisitos:**se realizaron revisiones a cada uno de los requisitos por parte del equipo de desarrollo. Las revisiones internas generaron un total de 6 no conformidades de tipo técnicas y de ortografía, las cuales fueron corregidas satisfactoriamente. Con el equipo de análisis y líder de proyecto se realizó una revisión en la cual se añadieron detalles a 4 requisitos funcionales y se aprobaron finalmente los mismos.
- **Construcción de prototipos:**se mostró al cliente un modelo ejecutable que permitió tener una visión preliminar de cómo se verían las funcionalidades del componente; se comprobó la satisfacción y aprobación del cliente.

3.1.1Métricas aplicadas a los requisitos

Con el objetivo de medir la calidad de la especificación de los requisitos se aplicó una de las métricas propuestas para la metodología AUP-UCI, Calidad de la especificación (CE).Para obtener cuán entendibles son los requisitos, primeramente, se calcula el total de requisitos de la especificación como se muestra a continuación:

Nr: el total de requisitos de especificación.

Nf: cantidad de requisitos funcionales.

Nnf: cantidad de requisitos no funcionales.

Nr= Nf + Nnf

Como resultado de la sustitución de los valores, se obtiene:

Nr = 17 + 12

Nr = 29

Para determinar, finalmente, la Especificidad de los Requisitos (ER) en cuanto a la ausencia de ambigüedad en los mismos se realiza la siguiente operación:

ER = Nui / Nr

Donde **Nui**es el número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca de 1 esté el valor de **ER**, menor será la ambigüedad.

Para el caso de los requisitos obtenidos, 5 produjeron contradicción en las interpretaciones. Sustituyendo las variables se obtiene:

ER = 24 / 29

ER = 0.82

Arrojando un resultado final satisfactorio, indicando que el grado de ambigüedad de los requisitos es bajo (18%) ya que el 82% son entendibles. Los requisitos ambiguos fueron modificados y validados para garantizar una correcta interpretación.

3.2Validación del diseño

Con el objetivo de validar el diseño realizado se aplicaron las siguientes métricas de diseño: Relaciones entre Clases (RC) y Tamaño Operacional de Clase (TOC).

3.2.1 Relación entre clases (RC)

Permite evaluar el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar una clase teniendo en cuenta las relaciones existentes entre ellas(Pressman, 2010). La ilustración que se muestra a continuación representala cantidad de clases por cantidad de relaciones de usos que poseen.

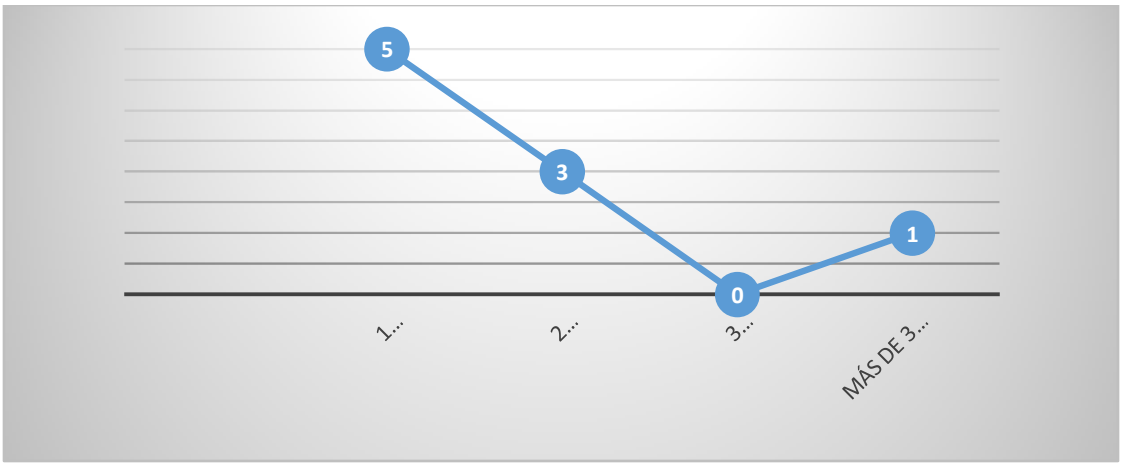


Ilustración 7.Clases y cantidad de relaciones de usos.

Fuente: elaboración propia.

En la gráfica se observa como 5 clases presentan 1 sola dependencia, 3 clases presentan 2 dependencias, no existe ninguna clase con 3 dependencias y existe 1 clase con más de tres dependencias.

A continuación, se muestra en porciento (%)el nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases al aplicar la métrica RC.

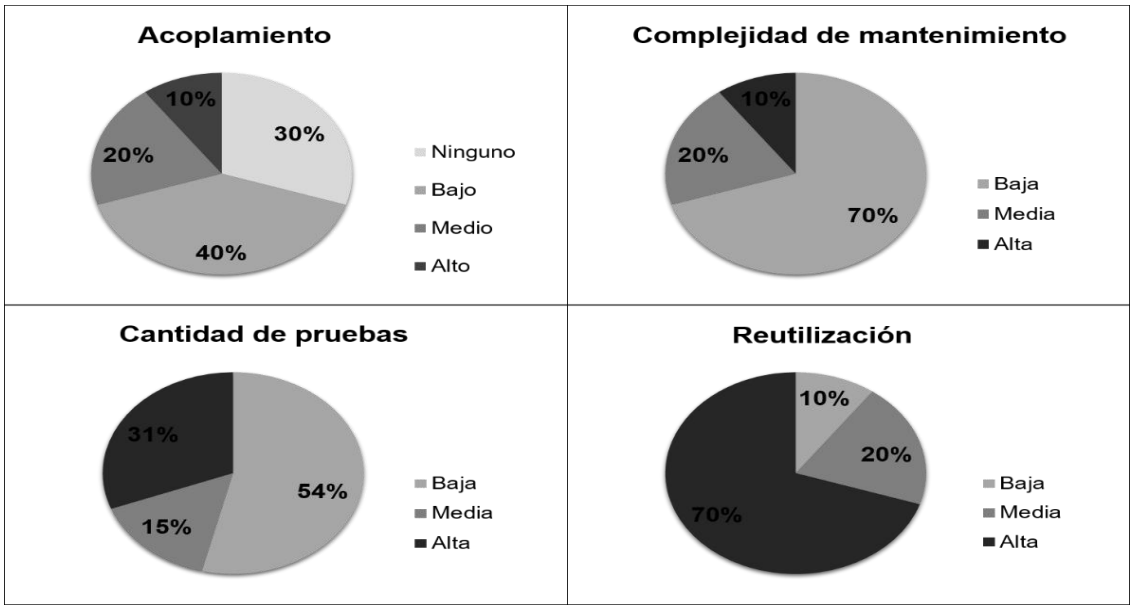


Ilustración 8.Nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases

Fuente: elaboración propia.

Una vez aplicada la métrica y teniendo en cuenta el umbral definido para validar el diseño (Bien [0.1; 0.3], Regular [0.4; 0.7] y Mal [0.8; 1]), se obtiene como resultado que las clases del diseño promueven el bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas no son altas y en consecuencia el grado de reutilización es mayor.

3.2.2 Tamaño operacional de clase (TOC)

Permite medir la responsabilidad, la complejidad de implementación y la reutilización de las clases del diseño. Es importante destacar que, para esta métrica, la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que, a mayor responsabilidad y complejidad de implementación de una clase, menor es su nivel de reutilización (Pressman, 2010).En la ilustración que se muestra a continuación se evidencian los resultados obtenidos de la aplicación de la métrica TOC.

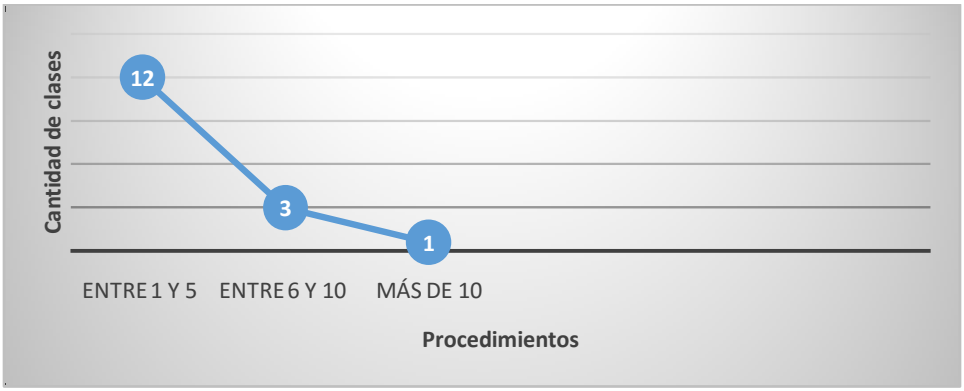


Ilustración 9.Cantidad de clases y procedimientos que contienen.

Fuente: elaboración propia.

En la gráfica se observa que existen 12 clases que presentan entre 1 y 5 procedimientos, 3 clases que tienen entre 6 y 10 procedimientos, y 1 clase que tiene más de 10procedimientos.

A continuación, se representa en porciento (%)el nivel de responsabilidad, complejidad de implementación y reutilización de las clases al aplicar la métrica TOC.

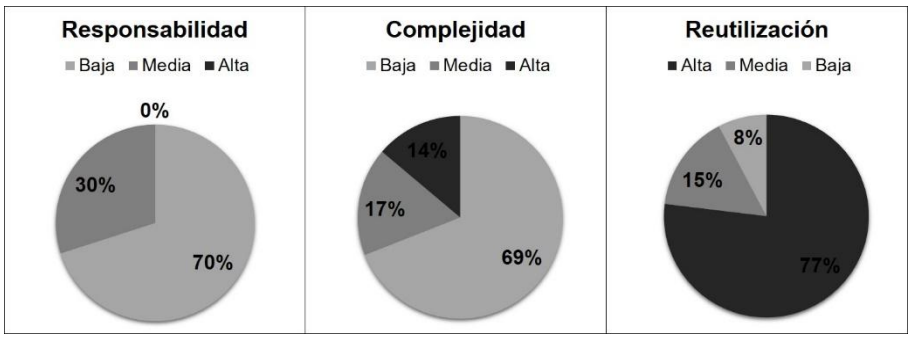


Ilustración 10.Nivel de responsabilidad, complejidad de implementación y reutilización de las clases.

Fuente: elaboración propia.

Luego de aplicada la métrica se observa que las clases del diseño no se encuentran sobrecargadas en cuanto a responsabilidades pues el 70% son bajas y el nivel de complejidad de las mismas no es elevado con un 69% bajo, teniendo una alta reutilización de un 77%.

De manera general, los resultados obtenidos de la aplicación de las métricas TOC y RC demuestran que el diseño no es complejo, que las clases presentan bajo acoplamiento y un alto grado de reutilización, lo que trae consigo que las clases puedan reutilizarse favoreciendo la implementación.

3.3 Pruebas de software

La disciplina de pruebas constituye el proceso que permite verificar y revelar la calidad de un producto software antes de ser entregado al cliente. Es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas(Institute Puig Castelar, 2018). La metodología escogida define las pruebas como una de las disciplinas a tener en cuenta.

3.3.1 Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas(Sánchez, 2015).

Pruebas unitarias

Las pruebas unitarias se realizan sobre las funcionalidades internas de un módulo y se encargan de comprobar los caminos lógicos, ciclos (bucles) y condiciones que debe cumplir el programa(Pressman, 2010).

Para aplicar las pruebas unitarias, los autores de la presente investigación deciden utilizar el método de Caja blanca. Con el empleo de este método es posible desarrollar casos de prueba que garanticen la ejecución, al menos una vez, de los caminos independientes (Pressman, 2010). Para aplicar este método se define la técnica de ruta básica.

Técnica de ruta básica

La técnica de ruta básica tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica de prueba debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada camino independiente sea ejecutado por lo menos una vez en el sistema (Pressman, 2010).

Pressman propone como estrategia para aplicar la ruta básica, realizar un análisis de la complejidad ciclomática de cada procedimiento que componen las clases del sistema; una vez concluido este paso se selecciona el método con valor de contener errores, además de que ofrece una medida del número de pruebas que deben diseñarse para validar la correcta implementación de una determinada función.

Para obtener los casos de prueba a partir de la técnica ruta básica, se debe construir el grafo de flujo para cada uno de los métodos desarrollados, a continuación, se muestra un ejemplo de la aplicación de la técnica correspondiente al código delmétodo:*mostrarGridView()* que es uno de los métodos más complejos desde el punto de vista de implementación (Ver Ilustración 11).

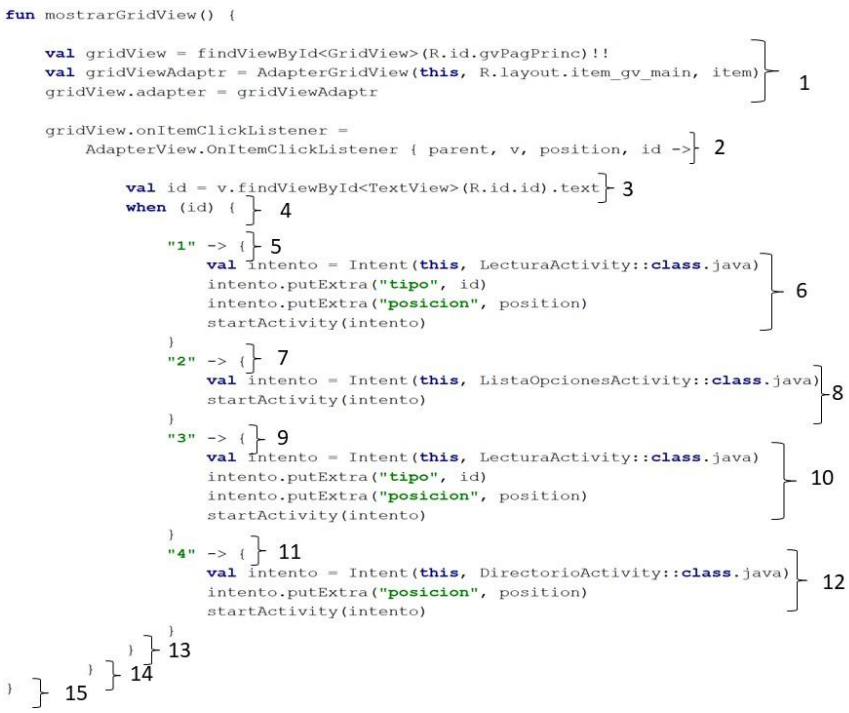


Ilustración 11.Método mostrarGridView().

Fuente: elaboración propia.

A continuación, se muestra el grafo de flujo obtenido a partir del método anteriormente descrito.

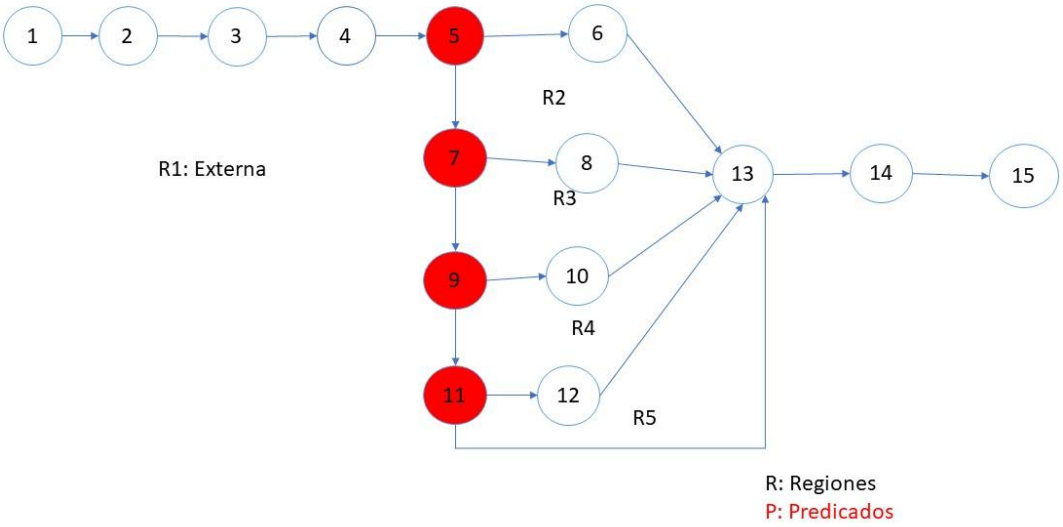


Ilustración 12. Grafo del flujo del método mostrarGridView().

Fuente: elaboración propia.

Luego se determina la complejidad ciclomática V (G) del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática puede ser calculada de 3 formas:

1. **V(G)** = R
2. **V(G)** = E – N + 2
3. **V (G)** = P + 1

Conociendo que:

- **G:** Grafo de flujo (grafo)
- **R:** El número de regiones contribuye a estimar el valor de la complejidad ciclomática
- **E:** Número de aristas
- **V(G):** Complejidad ciclomática
- **N:** Número de nodos del grafo
- **P:** Número de nodos predicados (nodos de donde parten al menos dos aristas)

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

V(G) = R **V(G)** = E – N + 2 **V(G)** = P + 1

V(G) = 5**V(G)** = 18 – 15 + 2 **V(G)** = 4 + 1

V(G) = 5**V(G)** = 5

Una vez calculada la complejidad ciclomática, el valor obtenido representa el límite superior de pruebas que debe aplicarse (Pressman, 2010). Por lo que los conjuntos de caminos básicos son:

Camino básico 1: 1-2-3-4-5-6-13-14-15
Camino básico 2: 1-2-3-4-5-7-8-13-14-15
Camino básico 3: 1-2-3-4-5-7-9-10-13-14-15
Camino básico 4: 1-2-3-4-5-7-9-11-12-13-14-15
Camino básico 5: 1-2-3-4-5-7-9-11-13-14-15

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos, en cada una de las funciones de la solución. A continuación, se presenta-a manera de ejemplo-el caso de prueba definido para la ruta independiente 1.

Tabla 5. Caso de prueba para ruta independiente 1.

Descripción	Muestra los documentos jurídicos y la pantalla de lectura, luego el usuario debe seleccionar un documento jurídico.
Condición de ejecución	Los datos se hayan cargado correctamente.
Entradas	-
Resultados esperados	Se debe mostrar en la pantalla de lectura el documento jurídico elegido por el usuario.

Fuente: elaboración propia.

Para comprobar la fiabilidad del código fuente se realizaron dos iteraciones completas en busca de errores de codificación, como condición de parada se tuvo en cuenta lo definido como resultado esperado en los casos de prueba. Saliendo errores de codificación, puesto que se emitía a pantallas no puestas en los casos de prueba. Se realizó una segunda iteración donde se erradicaron las No conformidades, por lo que se puede concluir que luego de realizar la

prueba de caja blanca, donde se diseñaron y ejecutaron los casos de prueba correspondientes, se logró asegurar el cumplimiento del proceso de mejora del código.

Pruebas funcionales

Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que se va a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas(Pressman, 2010).

Método de Caja negra

El método de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La misma no es una alternativa a las técnicas de método de caja blanca,más bien se trata de un enfoque complementario que intenta descubrir otros tipos de errores. Estas pruebas permiten encontrar (Pressman, 2010):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para llevar a cabo el método de Caja negra se utiliza la técnica de Partición equivalente.

Técnica de prueba: Partición equivalente

Esta técnica divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. El método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada. (Pressman, 2010)

Para aplicar esta técnica, se debe realizar el diseño de casos de prueba (DCP) con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software.Un DCP es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuales un analista determinará si una aplicación o una característica de éstos es parcial o completamente satisfactoria(Pressman, 2010).

Con el objetivo de comprobar el funcionamiento de los componentes, se realizaron un total de 3 iteraciones de pruebas, obteniéndose los resultados que se muestran en la siguiente ilustración:

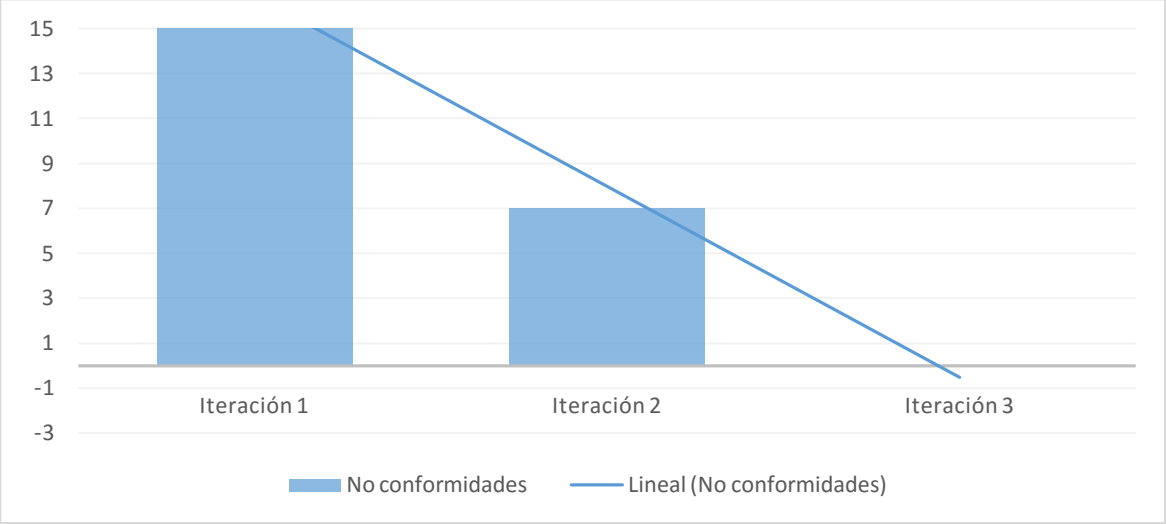


Ilustración 13.Cantidad de no conformidades por iteraciones.

Fuente: elaboración propia.

En una primera iteración se detectaron un total de 15 no conformidades (9 de validación, 4 de ortografía y 2 de interfaz). En la segunda iteración se manifestaron 7 no conformidades donde aún existían 5 errores de validación y 2 errores ortográficos. Finalmente, en una tercera iteración se obtuvieron resultados satisfactorios.

3.4Conclusiones del capítulo

La aplicación de técnicas de validación de requisitos permitió ratificar la correspondencia de los mismos con las solicitudes del cliente. Por su parte, la validación del diseño mediante las métricas TOC y RC permitió obtener, el grado de complejidad de implementación, mantenimiento, responsabilidad, reutilización, acoplamiento y la cantidad de pruebas necesarias para realizar a las clases. Por su parte, el método de caja blanca permitió comprobar internamente las funciones de los componentes, facilitando la detección de no conformidades para su corrección.Además, el método de caja negra permitió comprobar que las funciones son operativas a través de la interfaz del software.

CONCLUSIONES GENERALES

A partir de la investigación realizada y de los resultados obtenidos se concluye que:

- La elaboración del marco teórico en los que se sustenta la investigación, facilitó la adquisición de los conocimientos necesarios para el desarrollo de la solución.
- La especificación de los requisitos, el análisis y diseño del componente, permitieron obtener una aproximación para concebir los elementos necesarios de la implementación de la propuesta de solución.
- Con el desarrollo del componente para la reproducción por audio y diseño visual en la aplicación móvil iLex Notariose logra facilitar la consulta de los contenidos de esta aplicación porpersonas con discapacidad visual.
- Con los resultados obtenidos a través de las métricas de validación del diseño y las pruebas de software se pudo comprobar de forma cuantitativa la calidad de los artefactos obtenidos durante el desarrollo de la propuesta de solución.

RECOMENDACIONES

En correspondencia con las conclusiones a las que se ha arribado, se sugieren las recomendaciones siguientes:

- Se recomienda la integración del componente propuesto al resto de las aplicaciones móviles de la suite Lex-Cuba.
- Se recomienda utilizar el componente propuesto en otras aplicaciones móviles que brinden información de interés a la población y puedan ser usadas por discapacitados visuales.

BIBLIOGRAFÍA

AGI. 2016. [Online] 2016. https://www.agielkartea.org/informaci%C3%B3n-de-inter%C3%A9s/recursos.html.

Ambler, Scott W. 2018. Algunos lineamientos útiles para trabajar y seleccionar herramientas CASE de manera ágil. *Agile Module.* [Online] 2018. http://www.agilemodeling.com/essays/simpleTools.htm#SelectingCASE.

assistiva. 2019. Las aplicaciones más útiles para ciegos. *Las aplicaciones más útiles para ciegos.* [Online] Septiembre 23, 2019. https://assisvita.com/aplicaciones-para-ciegos/.

CALLEJA, MANUEL ARIAS. 2012. Estándares de codificación. *Estándares de codificación.* [Online] 2012. http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf.

Cano, José Hilario, Patricio Letelier Torres, Grupo ISSI. 2003. Metodologías ágiles en el desarrollo de software. [Online] noviembre 12, 2003. http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf.

Cepeda, Vicente. 2014. Que es y para que sirve SDK. *Que es y para que sirve SDK.* [Online] Abril 9, 2014. https://www.psafe.com/es/blog/que-es-y-que-sirve-el-sdk/.

Cillero, Manuel. 2009-2020. Mi circunstancia digital. *Mi circunstancia digital.* [Online] 2009-2020. https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-paquetes/.

Codina, Lluís. 2019. Dispositivos móviles 2019: plataformas, equipos, aplicaciones y fuentes de información. *Dispositivos móviles 2019: plataformas, equipos, aplicaciones y fuentes de información.* [Online] junio 18, 2019. https://www.lluiscodina.com/dispositivos-moviles-2019/.

Consultoria, Formacion. 2018. Instituto Superior de Ciencias Sociales y Sociosanitarios. *Instituto Superior de Ciencias Sociales y Sociosanitarios.* [Online] 2018. https://www.isesinstituto.com/discapacidad.

Copyright. 2020. Diccionario de la lengua española. *Diccionario de la lengua española.* [Online] 2020. https://www.wordreference.com/definicion/deficiencia.

Española, Real Academia. 2020. [Online] 2020.

Figueredo Concepción, Doimeadios y Perez. 2018. 2018.

Gaines, Jeff, Boyd, Geraldine and Copley, Della. 2017. Visual Paradigm Online. *Visual Paradigm Online.* [Online] 2017. https://online.visual-paradigm.com/es/features/.

Gamma, Erich, et al. 1994.*"Design Patterns: Elements of Reusable Object Oriented Sofware".* s.l. : Grady Booch, 1994.

Gardey, Julian Perez Porto y Ana. 2014.[Online] 2014. https://definicion.de/ofuscacion/.

González, Santiago Gil. 2013. Cómo hacer Apps accesibles. *Cómo hacer Apps accesibles.* [Online] Febrero 2013. http://www.ceapat.es/InterPresent1/groups/imserso/documents/binario/appsaccesibles.pdf.

Guardsquare. 2016-2019. El optimizador de codigo abierto para Java bytecode. [Online] 2016-2019. https://www.guardsquare.com/es/productos/proguard.

Institute Puig Castelar. 2018. Pruebas de aplicaciones web. *Pruebas de aplicaciones web.* [Online] 2018. https://elpuig.xeill.net/Members/vcarceler/asix-m09/uf1/nf1/a5.

JSON. 2016. JSON. [Online] Noviembre 22, 2016. http://www.json.org/json-es.html.

Larman, Craig. 2016.*UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 3ra . s.l. : Prentice-Hall, 2016.

López, Jose Maria. 2019. Kotlin, el lenguaje oficial de Android que quiere comerse a Java. *Kotlin, el lenguaje oficial de Android que quiere comerse a Java.* [Online] junio 9, 2019. https://hipertextual.com/2019/06/kotlin-lenguaje-android-java.

Mimenza, Oscar Castillero. 2019. Psicología y Mente. *Psicología y Mente.* [En línea] 2019. https://psicologiaymente.com/salud/tipos-de-discapacidad.

MINCOM. 2018. 2018.

MX, Java. Definicion. 2016. Que es el lenguaje de programacion JAVA. *Que es el lenguaje de programacion JAVA.* [En línea] 23 de November de 2016. http://definicion.mx/java/Definición de Java.

Oficina de Proyectos de Informática. 2018. 7 Técnicas de levantamiento de requerimientos software . *7 Técnicas de levantamiento de requerimientos software .* [Online] agosto 3, 2018. http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html.

OMS. 2014. 10 datos sobre la ceguera y la discapacidad visua. *10 datos sobre la ceguera y la discapacidad visua.* [Online] Agosto 2014. https://www.who.int/features/factfiles/blindness/es/.

—. 2020. OMS. *OMS.* [Online] 2020. https://www.who.int/topics/disabilities/es/.

Pérez, Mariñan. 2013. Patrones de Diseño. [Online] 2013. https://www.ecured.cu/Patrones_Gof.

Pressman, ROGER R. 2017. Ingenieria del Software. Un Enfoque Practico - Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF. *Ingenieria del Software. Un Enfoque Practico - Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF.* [Online] June 1, 2017. http://s3.amazonaws.com/academia.edu.documents/45525376/Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1496273138&Signature=GJvs9eIIJ0pb9hnGQlz%2F6hIyeac%3D&response-content-disposition=inline%3B%20fil.

—. 2017. Un Enfoque Practico - Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF. *Un Enfoque Practico - Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF.* [Online] Junio 1, 2017. http://s3.amazonaws.com/academia.edu.documents/45525376/Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1496273138&Signature=GJvs9eIIJ0pb9hnGQlz%2F6hIyeac%3D&response-content-disposition=inline%3B%20fil.

Pressman, Roger S, Ph.D. 2010.*Ingenieria de Software. Un enfoque práctico. Septima Edición.* Mexico, D.F : The Me Graw Hill Companies, 2010.

S.L, Puntodis. 2019. PUNTODIS. *PUNTODIS.* [Online] 2019. https://puntodis.com/featured_item/discapacidad-visual/.

Saez, Maria Paul. 2014. Aplicaciones para personas con discapacidad. *Aplicaciones para personas con discapacidad.* [Online] December 5, 2014. https://blogthinkbig.com/aplicaciones-para-personas-con-discapacidad.

Sanchez, Ruby. 2018. Especificación de Requisitos Software según el estándar de IEEE 830. [Online] octubre 2, 2018.

https://www.academia.edu/6647065/Especificaci%C3%B3n_de_Requisitos_Software_seg%C3%BAn_el_est%C3%A1ndar_de_IEEE_830.

Sánchez, Tamara Rodríguez. 2015.*Metodología de desarrollo para la Actividad productiva de la UCI. .* La Habana. Cuba : s.n., 2015.

ServiSoftCorp. 2010. [Online] 2010. https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/.

Sommerville. 2017. Modelos Y Metodologías Para El Desarrollo De Software. *Modelos Y Metodologías Para El Desarrollo De Software.* [Online] Junio 6, 2017. http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm.

Sommerville, Ian. 2011.*Software engineering, Ninth Edition.* 2011.

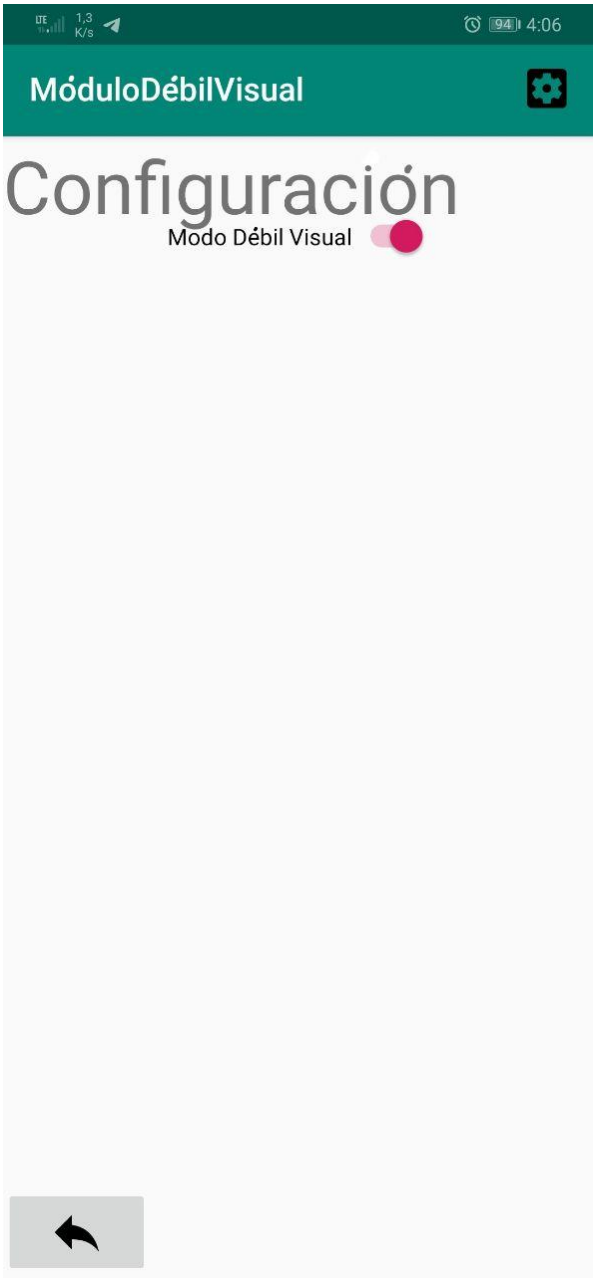
Studio, Android. 2016. Conoce a Android Studio. *Conoce a Android Studio.* [Online] November 22, 2016. https://developer.android.com/studio/intro/index.html.

TechTerms. 2019. TechTerms. [Online] 10 7, 2019. www.google.com/amp/s/techterms.com/amp/definition/camelcase.

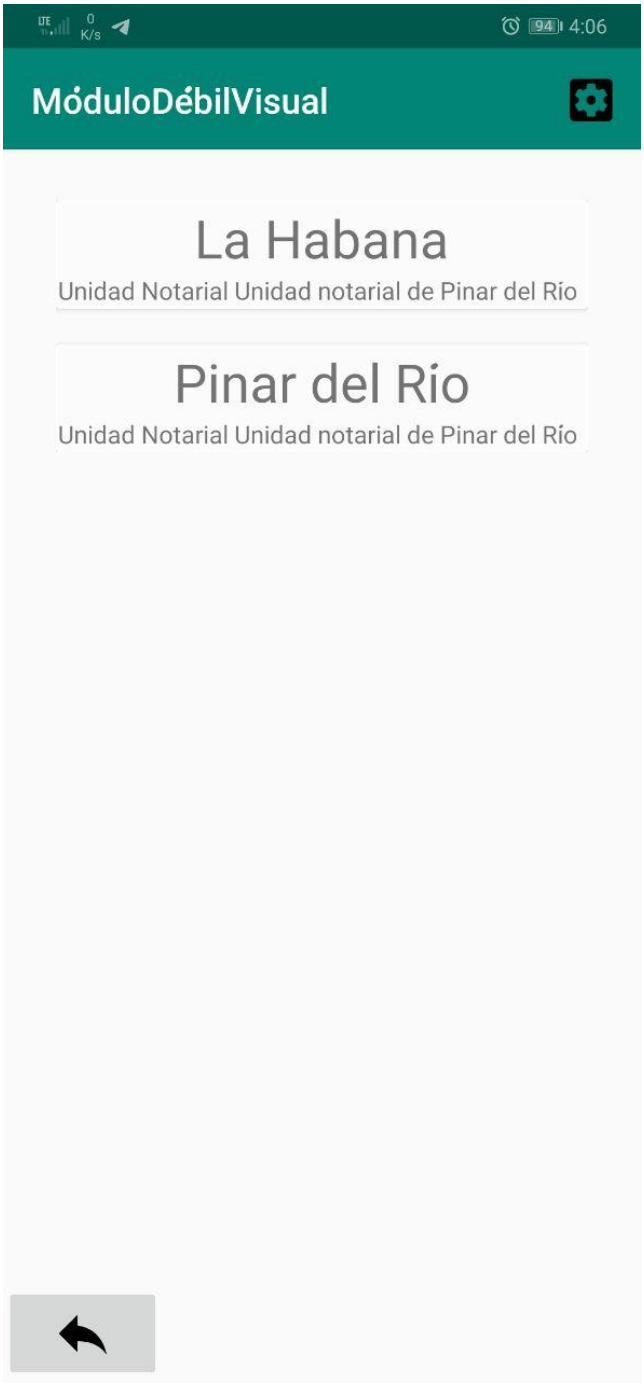
Vega, Adrian Alonso. 2017. Convención de nombres: desde el CamelCase hasta el kebab-case. *Medium.* [Online] 6 11, 2017. https://medium.com/@alonsus91/convenci%C3%B3n-de-nombres-dessde-el-camelcase-hasta-el-kebab-case-787e56d66023.

ANEXOS

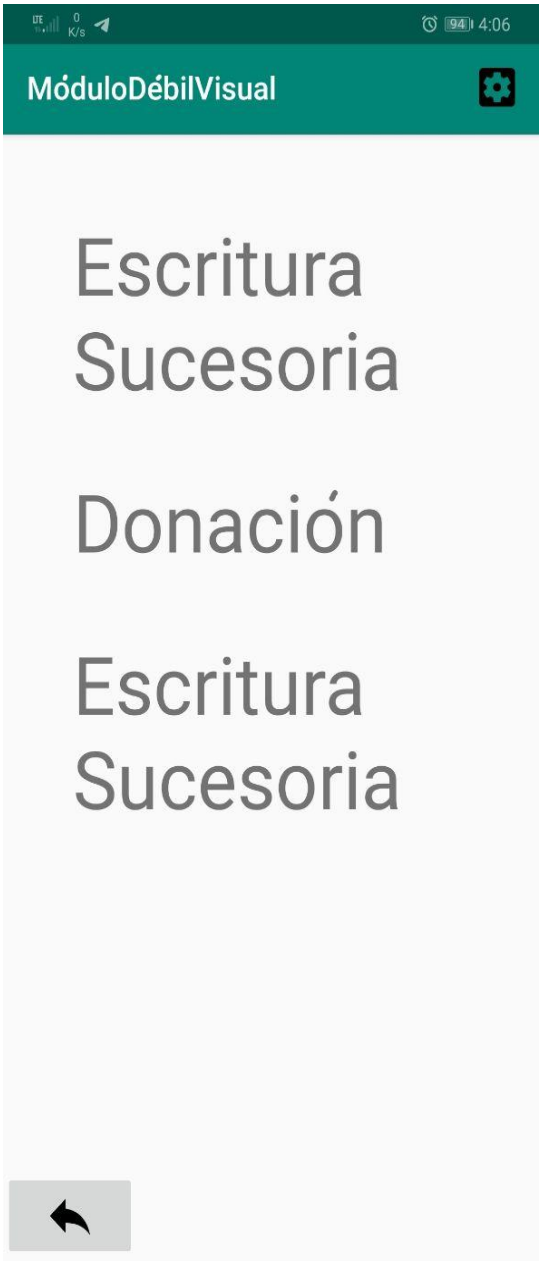
ANEXO 1. Pantalla de configuración de la aplicación.



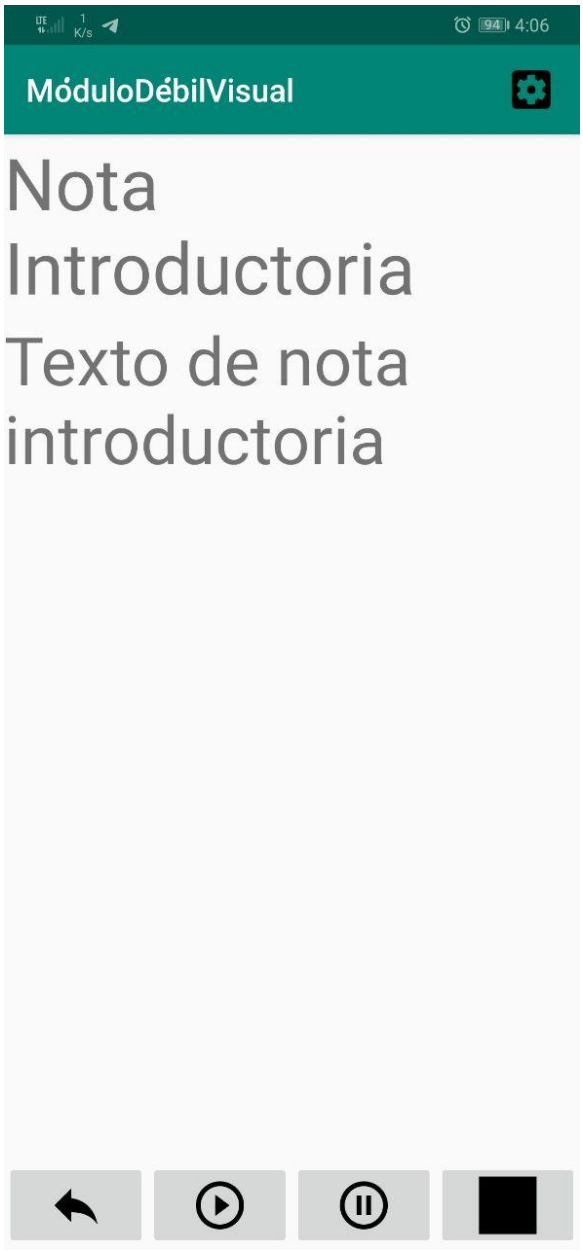
ANEXO 2. Pantalla de directorio de la aplicación.



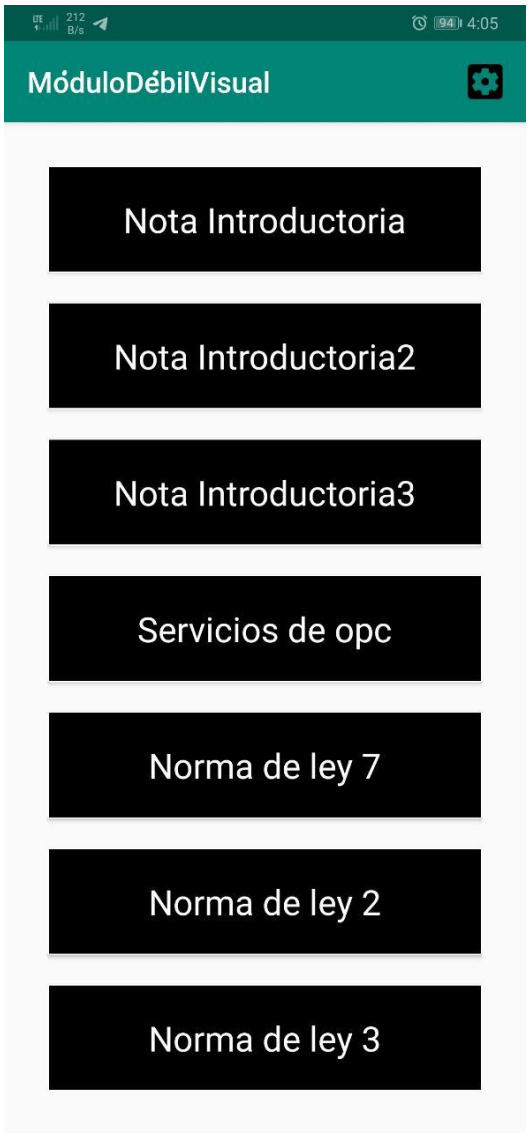
ANEXO 3. Pantalla de listado legislativo de la aplicación.



ANEXO 4. Pantalla de lectura de la aplicación.



ANEXO 5. Pantalla inicial de la aplicación.



ANEXO 6. Entrevista.

Guía de preguntas utilizadas en el desarrollo de la entrevista con los profesionales delMINJUS.

1. ¿El usuario podrá cambiar el tamaño de la letra?
2. ¿El usuario podrá cambiar el tipo de letra?
3. ¿El usuario podrá cambiar el color de la letra?
4. ¿El usuario podrá acceder a la pantalla principal desde cualquier otra pantalla que se encuentre?
5. ¿El usuario podrá reproducir la lectura de los textos cuando desee?
6. ¿El usuario podrá pausar la lectura de los textos cuando desee?
7. ¿El usuario podrá detener la lectura de los textos cuando desee?
8. ¿El usuario podrá ajustar el volumen de los textos cuando desee?