



Universidad de las Ciencias Informáticas

Centro de Gobierno Electrónico

Facultad 3

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

**iLex. Publicidad normativa en su
móvil**

Autor: Felix A. Marrero Pentón

Tutor (es): MsC. Yarina Amoroso Fernández

Ing. Doris Maza Oval

La Habana, junio 2017

Pensamiento

"El desconocimiento de la Ley no exime de su cumplimiento"

Precepto legal universal



Declaración de autoría

Declaro que soy el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor:

Felix A. Marrero Pentón _____

Firma del Tutor:

MsC. Yarina Amoroso Fernández _____

Firma del Tutor:

Ing. Doris Maza Oval _____

Datos de contacto

Tutor: MsC. Yarina Amoroso Fernández

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Centro de trabajo: Centro de Gobierno Electrónico. Departamento de Práctica Profesional.

Correo electrónico: yarina@uci.cu

Tutor: Ing. Doris Maza Oval

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Centro de trabajo: Centro de Gobierno Electrónico. Departamento de Práctica Profesional.

Correo electrónico: dmaza@uci.cu

Autor: Felix A. Marrero Pentón

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: felixmarrero@nauta.cu

Dedicatoria

*A mi madre principalmente, por el sacrificio de tantos años.
Por haber puesto mi bienestar personal ante todas las cosas y
por la educación ejemplar que me ha dado y me ha
convertido en el hombre que hoy soy.*

*A mi padre, por ser un ejemplo para mí. Por darme siempre su
apoyo y confianza.*

A los dos por estar siempre conmigo.

Agradecimientos

*A mi madre, Maricel, ya que este sueño se ha hecho realidad
gracias a ella.*

A mi padre, Jorge, por el apoyo que siempre me ha dado.

*A mis tías Grisel y Fela, por ser como segundas madres para
mí.*

A mi hermana Lili por estar siempre ahí cuando lo necesite.

A mi familia en general.

*A los compañeros de la Facultad Regional de Ciego que
quedaron allá.*

*Y a Claudia, Yanser, Rolando, Camilo, Alejandro, Arlene,
Julio, Rafa, Lizandra, Doris y Daímary que vinieron conmigo
y cuyos nombres no podía dejar de mencionar.*

A las nuevas amistades que encontré aquí al llegar.

*A los profes del centro CEGEL por su apoyo brindado en estos
dos años.*

*Y a todos los que de una manera u otra ayudaron a que este
sueño se cumpliera.*

Resumen

Conocer el Derecho constituye un derecho ciudadano, sin embargo, entre la población cubana, particularmente entre los jóvenes se ha evidenciado cierto desconocimiento de la Ley. En el presente trabajo se propone el desarrollo de la aplicación "iLex" para entornos móviles con sistema operativo Android. Dicha aplicación persigue difundir la información legal a partir de la Gaceta Oficial de la República de Cuba y sus fondos documentales, así como favorecer de manera más amplia el cumplimiento a la publicidad normativa. La aplicación tiene como objetivo poner al alcance de todos las Leyes, decretos, resoluciones, reglamentos y demás disposiciones generales de los órganos del estado. El significado de iLex está dado por la unión de las palabras "i" de información y "Lex" que proviene del latín y significa Ley. Dicha aplicación en su versión 2.0, se desarrolló utilizando como entorno de desarrollo Android Studio, publicado de forma gratuita a través de la Licencia Apache 2.0 y como metodología de desarrollo en la solución Scrum. iLex es propiedad intelectual del Centro de Gobierno Electrónico (CEGEL) de la Facultad 3 perteneciente a la Universidad de las Ciencias Informáticas (UCI) y así consta registrado.

Palabras claves: aplicación, divulgación, ilex, legislación, móvil, publicidad

Abstract

Knowing the law is a civil right, however, among the Cuban population, particularly among young people has shown a certain ignorance of the law. In this paper it is propose the development of the application "iLex" for mobile environments with Android operating system. Said application seeks to disseminate legal information from the Official Gazette of the Republic of Cuba and its documentary funds, as well as to favor in a broader way the fulfillment to the normative publicity. The purpose of the application is to make available to all Laws, decrees, resolutions, regulations and other general provisions of state organs. The meaning of iLex is given by the union of the words "i" of information and "Lex" that comes from Latin and means Law. This application in its version 2.0, was developed using as an Android Studio development environment, published for free through the Apache 2.0 License and as development methodology in the solution Scrum. ilex is the intellectual property of the Center for Electronic Government (CEGEL) Faculty 3 belonging to the University of Information Sciences (UCI) and thus has recorded.

Keywords: application, divulgation, ilex, legislation, mobile, publicity

Índice

Introducción	1
Capítulo I. Fundamentación Teórica	6
1.1 Introducción.....	6
1.2. Conceptos asociados a la investigación	6
1.2.1 Publicidad	6
1.2.2 Publicidad Normativa	6
1.2.3 Divulgar y Difundir.....	7
1.2.4 Difusión o Divulgación Normativa	7
1.2.5 Dispositivos Móviles.....	7
1.3 Estudio del estado arte.....	7
1.3.1 Soluciones internacionales	7
1.3.2 Soluciones nacionales	9
1.3.3 Análisis de las soluciones encontradas.....	10
1.4 Metodología de Desarrollo y Tecnologías Seleccionadas.....	11
1.4.1 Metodología de desarrollo.....	11
1.4.2 Herramienta de Modelado.....	15
1.4.3 Sistema Operativo	15
1.4.4 Entorno de Desarrollo Integrado	16
1.4.5 Lenguajes de Programación	17
1.4.6 Android SDK	17
1.4.7 JSON	18
1.4.8 SQLite.....	18
1.5 Pruebas de Software	19
1.5.1 Casos de Pruebas	20
1.5.2 Pruebas de Caja Blanca (PCB).....	20
1.5.3 Pruebas de Caja Negra (PCN).....	20
1.6 Conclusiones del Capítulo	21
Capítulo II. Análisis y diseño de la solución propuesta.....	22

2.1 Introducción.....	22
2.2 Propuesta de Solución	22
2.3 Roles del Proyecto	23
2.4 El Sprint.....	23
2.5 Requisitos del Sistema	23
2.6 Descripción de Historias de Usuarios	25
2.7 Patrón Arquitectónico	28
2.8 Patrones de Diseño.....	29
2.8.1 Patrones GRASP	30
2.8.2 Patrones GOF.....	31
2.9 Almacenamiento de los datos.....	31
2.10 Estándar de Codificación.....	32
2.11 Conclusiones del Capítulo	34
Capítulo III. Validación del producto.....	35
3.1 Introducción.....	35
3.2 Validación de la Solución	35
3.2.1 Resultados de las Pruebas de Caja Blanca	35
3.2.2 Resultados de las Pruebas de Caja Negra.....	42
3.3 Validación de las variables de investigación.....	45
3.4 Comparación cualitativa entre las versiones 1.0 y 2.0 de iLex.....	46
3.5 Conclusiones del Capítulo	48
Conclusiones	50
Recomendaciones	51
Referencias Bibliográficas	52
Anexos	56
Anexo 1.....	56
Anexo 2.....	57
Anexo 3.....	58
Anexo 4.....	58

Anexo 5.....	59
Anexo 6.....	59
Anexo 7.....	60
Anexo 8.....	60
Anexo 9.....	61
Anexo 10.....	61
Anexo 11.....	62
Anexo 12.....	62
Anexo 13.....	63
Anexo 14.....	63
Anexo 15.....	64
Anexo 16.....	64

Índice de Figuras

Figura 1: Evolución de las líneas móviles.	2
Figura 2: Comparación entre el uso de sistemas operativos.	15
Figura 3: Versiones de SQLite según las API.	19
Figura 6: Representación del patrón Modelo Vista Presentador.	28
Figura 7: Diagrama de paquetes.	29
Figura 8: Dependencia de la clase VerDocumento.java.	30
Figura 9: Interfaz de la clase ListarCategorias.java.	31
Figura 10: Declaración de variables.	32
Figura 11: Espacios en blanco.	33
Figura 12: Nombre de las clases.	33
Figura 13: Nombre de los métodos.	33
Figura 14: Nombres de las variables.	34
Figura 15: Resultado SonarQube. Bugs y vulnerabilidades.	36
Figura 16: Resultado SonarQube. Líneas de código.	37
Figura 17: Resultado SonarQube. Complejidad.	37
Figura 18: Método dameParrafo().	38
Figura 19: Método llenarInciso().	39
Figura 20: Métodos aux1() y aux()2.	39
Figura 21: Método silncisoVacio().	39
Figura 22: Método siParrafoVacio().	40
Figura 23: Método dameParrafo() optimizado.	40
Figura 24: Resultado SonarQube. Código duplicado.	40
Figura 25: Método salirDeiLex().	41
Figura 26: Método llenarSpinner ().	41
Figura 27: Listado de Leyes.	42
Figura 28: Cambiar tamaño de texto.	43
Figura 29: Gráfico. Resultado de pruebas funcionales.	44
Figura 30: Resultado de encuesta en sitio www.humanos.uci.cu	45
Figura 31: Resultado de encuesta en sitio www.jorgen.cubava.cu	46

Índice de Tablas

Tabla 1: Roles del proyecto.	23
Tabla 2: Requisitos funcionales.	24
Tabla 3: Historia de Usuario No. 5.	26
Tabla 4: Historia de Usuario No. 20.	27
Tabla 5: Bugs y vulnerabilidades.	36
Tabla 6: Comparación entre versiones.....	48

Introducción

El diario Juventud Rebelde, en su edición del 5 de abril del 2015, publica como primera reflexión en el artículo “Jóvenes de Ley” lo siguiente: *“El Derecho no puede estar solo en blanco y negro; requiere que se le traiga desde los textos hacia la realidad. Y como en la juventud se está formando la ciudadanía de hoy, en ella debe primar la cultura jurídica como expresión de la conciencia ciudadana (1)”*.

En Cuba, la cultura jurídica, medida en el acceso y conocimiento de la Ley, arroja cifras que deben ser premisas a la hora de precisar la instrumentación de mecanismos idóneos de difusión, así como mejorar los existentes. Tal afirmación se sustenta en los resultados obtenidos mediante una encuesta (ver anexo 1) realizada a estudiantes, profesores, trabajadores y especialistas de la Universidad de las Ciencias Informáticas (UCI) con el objetivo de estudiar el nivel de conocimiento existente sobre la Constitución de la República de Cuba. Se tomó esta Ley porque es la Ley fundamental que establece los principios que rigen el orden legal de la sociedad cubana, así como los deberes, derechos y libertades para todos los ciudadanos y las responsabilidades del Estado. De la muestra escogida de 213 personas, un 60,65 % reconoce no haber leído nunca la Constitución. De igual modo, un elevado por ciento de los encuestados manifestó, tener poco o no tener conocimiento alguno acerca de la Constitución de la República de Cuba. Dichos resultados pueden ser extrapolados a la toda la sociedad cubana debido a la variedad de personas de diferentes provincias que se encuentran en la institución.

Una de las causas principales del desconocimiento de la legislación es la existencia de deficiencias en los medios de publicidad normativa actuales. Actualmente la divulgación popular del Derecho se ve afectada por el modo en que prevalece el proceso de edición y publicación de las Leyes en el país.

La Gaceta Oficial de la República de Cuba es el órgano oficial del estado para publicar las Leyes, el artículo 77 de la Constitución de la República de Cuba dice lo siguiente: *“Las leyes, decretos-leyes, decretos y resoluciones, reglamentos y demás disposiciones generales de los órganos del Estado, se publican en la Gaceta Oficial de la República”*. La misma puede encontrarse en papel (ejemplo los boletines de la gaceta que se pueden encontrar en los estancillos de correo) o en formato digital a través de un sitio web (por ejemplo el sitio oficial de la Gaceta¹ y el sitio de la Asamblea Nacional del Poder Popular²). Todo esto trae aparejados contratiempos propios del tipo de soporte. En el caso del papel el Estado incurre en costos asociados a la impresión de los

¹ www.gacetaoficial.gob.cu

² www.parlamentocubano.cu

boletines. Por ejemplo, cada año, editoriales como la de la Organización Nacional de Bufetes Colectivos (ONBC) y la Ignacio Agramonte, del Ministerio de Justicia, entre otras, dedican recursos a imprimir textos sobre materia jurídica; y no siempre la cantidad de textos impresos son suficientes para satisfacer a toda la población. Además, no todas las personas cuentan con a la infraestructura necesaria para acceder a la intranet nacional y visitar los sitios web antes mencionados. Los problemas anteriormente descritos evidencian las dificultades que existen a la hora de hacer llegar la Ley a las personas y el poco alcance que tiene el canal oficial de publicidad normativa actual.

En la actualidad las personas asumen con total normalidad la presencia de las tecnologías en la sociedad, conviven con ellas y las adoptan sin dificultad para su uso cotidiano, utilizándolas como herramientas de interacción, información, comunicación y conocimiento. En Cuba en los últimos años la telefonía celular ha tenido un gran auge y desarrollo. Según datos proporcionados por la propia Empresa de Telecomunicaciones de Cuba (ETECSA), en mayo del 2017 había ya 4,2 millones de líneas móviles contratadas por la población (2).

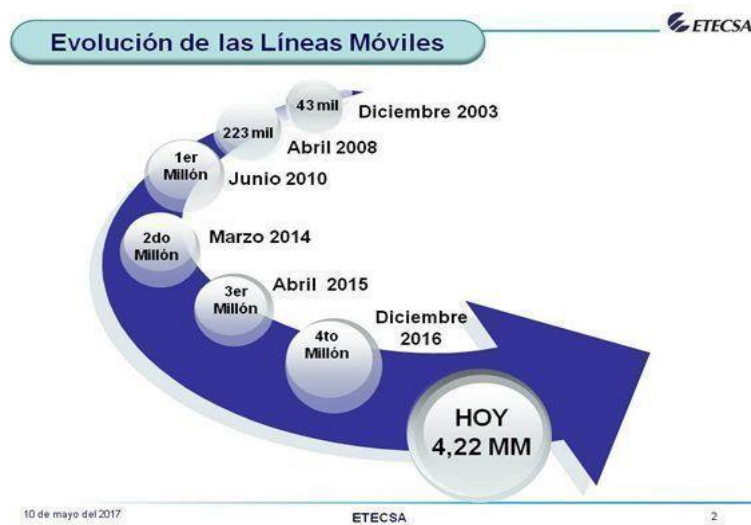


Figura 1: Evolución de las líneas móviles.

Al respecto la mayoría de las personas encuestadas tenían en su poder un teléfono móvil, pues de la muestra solo 50 dijeron no tenerlo, pero señalaron estar dispuestos a adquirirlo en un corto período de tiempo. Por otro lado, nuestro país se encuentra en un proceso de informatización de la sociedad. El uso de la tecnología móvil ha crecido considerablemente, sobre todo la utilización de los llamados teléfonos inteligentes. Esto confirma que la tecnología móvil puede surgir como un método novedoso para el dominio cubano en el cumplimiento a la divulgación popular del Derecho. Sobre la base de tal resultado, pudiera ser de interés para la dirección del país la utilización de las

potencialidades de la telefonía celular en el amplio espectro de las nuevas Tecnologías de la Informática y las Comunicaciones (TICs).

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

¿Cómo contribuir en la ampliación del canal de publicidad normativa de manera que se propicie un acercamiento mayor de la población a las disposiciones legales del país?

Este trabajo tiene como **objeto de estudio** el desarrollo de un canal de publicidad normativa y como **campo de acción** la Informatización de un canal de publicidad normativa.

Para ello se identifica como **objetivo general**: Desarrollar una aplicación móvil para la instrumentación de un canal de publicidad normativa que propicie un acercamiento mayor de la población a las disposiciones legales del país.

Objetivos específicos:

1. Definir el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos para el desarrollo de la solución.
2. Realizar el diseño e implementación de la solución para obtener los componentes de software asociados con la publicidad normativa.
3. Valorar el nivel de aceptación y efectividad de la solución propuesta y la realización de pruebas de caja negra y caja blanca.

Para cumplir con el objetivo propuesto y resolver la situación problemática mencionada, se plantean las siguientes **tareas de la investigación**:

- Análisis del estado de arte de los canales de publicidad normativa.
- Levantamiento de las funcionalidades de los procesos de negocio.
- Caracterización de la metodología, las tecnologías y las herramientas a utilizar para el desarrollo de la solución.
- Diseño de la estructura y comportamiento de los componentes de los procesos identificados.
- Implementación de los componentes diseñados.
- Validación de la implementación realizada.
- Validación de las variables de la investigación.

Planteándose como **posible resultado**: Obtención de una aplicación móvil para la instrumentación de un canal de publicidad normativa.

Siendo la **idea a defender**: El desarrollo de una aplicación móvil para la instrumentación de un canal de publicidad normativa, propiciará un acercamiento mayor de la población a las disposiciones legales del país.

Para dar cumplimiento a las tareas de investigación se emplearon los siguientes métodos de investigación:

Métodos Teóricos:

- **Histórico-Lógico:** Utilizado para el estudio de soluciones informáticas existentes y lo que se pudiese tomar de ellas para resolver el problema planteado.
- **Análisis Documental:** Utilizado en el estudio de la documentación existente referente al problema planteado que pudiera aportar ideas importantes para la investigación.

Métodos Empíricos:

- **Encuesta:** Utilizada al inicio de la investigación como vía para medir el nivel de conocimiento de la población sobre la legislación del país y la facilidad de acceso de las personas a la misma.

Estructura del Documento:

Como continuación de lo antes expuesto, el estudio se divide en tres capítulos:

Capítulo I: Se da a conocer la fundamentación teórica de la investigación, así como los principales conceptos que contribuyen a mejorar el entendimiento del problema en cuestión. Se realiza un estudio del estado del arte de las tecnologías ya existentes que de alguna manera pudieran darle solución al problema planteado. Se realiza una comparación entre las mismas donde se dan a conocer sus deficiencias y por qué no son óptimas para alcanzar la solución deseada. Se define la metodología de desarrollo utilizada, así como las tecnologías, herramientas y entorno de desarrollo.

Capítulo II: En este capítulo se realiza una descripción de la aplicación propuesta como solución y sus principales funcionalidades. Se argumenta el uso de los patrones de diseño y se detalla las particularidades de la metodología seleccionada.

Capítulo III: En este capítulo se elaboran y ejecutan los casos de prueba para verificar el correcto funcionamiento de la aplicación. Se hace un análisis de los resultados

obtenidos, se resuelven las No Conformidades detectadas y se valida el resultado de la investigación.

Capítulo I. Fundamentación Teórica

1.1 Introducción

En este capítulo se abordan los principales conceptos asociados a la investigación. Se realiza un estudio de las soluciones informáticas existentes para plataformas móviles con sistema operativo Android³ que pudieran dar respuesta al problema planteado. Además, se realiza una comparación entre dichas soluciones para identificar las deficiencias de las mismas y lo que se puede tomar de ellas. Por último, se dan a conocer las tecnologías y metodologías para el desarrollo de la solución propuesta en el objetivo general.

1.2. Conceptos asociados a la investigación

1.2.1 Publicidad

Según la Real Academia de la Lengua Española (RAE) publicidad es difusión o divulgación de información, ideas u opiniones de carácter político, religioso, comercial, etc., con la intención de que alguien actúe de una determinada manera, piense según unas ideas o adquiera un determinado producto (3).

1.2.2 Publicidad Normativa

La publicidad normativa es el principio exigido por la seguridad jurídica, que permite a los ciudadanos conocer las disposiciones normativas que están obligados a cumplir. La publicidad de las normas constituye uno de los pilares del Estado de Derecho. Dado que «la ignorancia de las Leyes no excusa de su cumplimiento», ha de facilitarse el conocimiento de las normas por los ciudadanos, para lo cual éstas se publican en el Diario Oficial o Gaceta Oficial (4).

Visto de otro ángulo, la publicación de la norma es el primer acto de ejecución de la "Ley". Existen más o menos estandarizadas fórmulas de publicidad normativa que se expresan generalmente en las disposiciones finales de los textos normativos en formas tan arraigadas como "PUBLÍQUESE en la Gaceta Oficial de la República para general conocimiento (...)" e incluso en ocasiones se indica a continuación "la que entrará en vigor a partir de (...)".

De este modo, la publicidad constituye pues una condición para fijar el comienzo inmediato de la vigencia de tales disposiciones y la base del carácter presuntivo del

³ Android es el nombre de 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 otras máquinas.

conocimiento de la disposición por el hecho mismo de su publicación en un medio de difusión escrito de carácter oficial creado al efecto(4).

1.2.3 Divulgar y Difundir

Según la RAE es hacer que un hecho, una noticia, una lengua, un conjunto de conocimientos, etc., llegue a conocimiento de muchas personas (5).

1.2.4 Difusión o Divulgación Normativa

Según el concepto de divulgar y difundir, la difusión o divulgación normativa se refiere a la correcta distribución de las normas jurídicas de un país por diferentes vías o canales de difusión normativa, ya sea periódicos, sitios web, aplicaciones móviles entre otros, con el objetivo de que de una manera simple las mismas estén al alcance de todos los ciudadanos.

1.2.5 Dispositivos Móviles

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales (6).

Entre los dispositivos móviles más utilizados encontramos las Tablet (“Tablet”) y los teléfonos inteligentes (“Smartphone”).

1.3 Estudio del estado del arte

Para toda investigación resulta de vital importancia que se realice un estudio de los antecedentes de la misma tanto a nivel nacional como internacional. Por este motivo a continuación se presentan distintas vías para acceder a la información legislativa de diferentes países utilizando las TICs.

1.3.1 Soluciones internacionales

Este trabajo se propone desarrollar un producto que acerque más a los ciudadanos del país a la legislación cubana. La sencillez y la movilidad del mismo son premisas para alcanzar el resultado deseado. Además, debe contar con la mayor cantidad de información legal para que la misma este a disposición del usuario en todo momento. A nivel internacional existen diferentes productos que se han tomado como referencia para la realización de iLex, dichos productos representan un avance en el campo de la divulgación jurídica en sus respectivos países. Entre estos se encuentran:

LEX NOVA Apk:

Lex Nova ofrece a los profesionales jurídicos y empresariales toda la información legislativa que necesitan para su trabajo diario. Actualizándose automáticamente, esta aplicación le da acceso a la legislación española vigente (7).

La aplicación contiene la selección de normas más importantes de su rama del Ordenamiento Jurídico. Incluye además otros contenidos de gran valor (7):

- Notas de concordancia: detalle de otros artículos o normas relacionados.
- Notas de actualización: detalle de cuándo una norma ha sido modificada y cómo.
- Notas de redacción original: donde se incluyen los textos no vigentes de la norma.

vLex:

vLex es una biblioteca virtual especializada en Derecho que contiene todo lo que un estudiante o profesional pueda necesitar en el ejercicio de sus labores. Su estructura es ordenada para un fácil uso: legislación, jurisprudencia, doctrina y modelos de contratos, también es importante la diferenciación de la búsqueda por países y épocas y la facilidad de buscar también bajo todos los criterios a la vez (8).

En un primer momento, vLex ofreció información jurídica gratuita de España y América Latina y se financiaba con publicidad. Posteriormente, a medida que se iban incorporando contenidos premium (libros, revistas, contratos), se desarrollaron productos de información jurídica por suscripción. Primero fueron para España y más adelante para el resto del mundo (9).

Ofrece acceso a Legislación, jurisprudencia y Doctrina de 28 países latinoamericanos y la información publicada por organismos como Mercosur, Pacto Andino y Corte Interamericana de derechos humanos. Acceso a más de 1.900 títulos de libros y revistas a texto completo de las principales editoriales comerciales y academias de la región en una base de datos en permanente crecimiento (10).

Además, permite acceder a contenidos exclusivos de alto valor editorial elaborados por los autores de más prestigio de cada país (más de 130 países) y con garantía de actualización diaria y permanente. Todos los documentos de vLex contienen un cuidadoso análisis jurídico elaborado por un equipo de expertos, lo que te permite llegar rápidamente al resultado más preciso y actualizado (11).

Mi abogado Apk (ARAG):

Es una aplicación para consultar dudas legales relacionadas con temas de Vivienda, Consumo y Economía doméstica de manera muy sencilla e inmediata. La aplicación permite obtener las respuestas de un abogado ARAG⁴ a las dudas legales más habituales y si se necesita más información permite llamar y hablar directamente con un abogado de ARAG para resolver las dudas (12).

Todas estas soluciones internacionales se presentan de manera eficiente para resolver problemas a la hora de acceder a la legislación de distintos países por lo que se toman como guía a la hora de implementar la solución deseada para nuestro país.

1.3.2 Soluciones nacionales

El principal objetivo de este trabajo es ayudar a mejorar la divulgación de las normas legales cubanas ampliando el canal **oficial** de publicidad normativa para el Ministerio de Justicia de la República de Cuba (MINJUS). Entre las soluciones nacionales que resuelven este problema se encuentran:

Gaceta Oficial de la República de Cuba:

Es el órgano oficial de publicidad normativa reconocido constitucionalmente. El Ministerio de Justicia es el responsable de su publicación y es el custodio de su archivo central (13).

Sitio Web de la Gaceta:

La Gaceta Oficial cuenta con un sitio digital (<https://www.gacetaoficial.gob.cu>) en el que se publican periódicamente las Leyes y normas de la nación. Dicho sitio presenta el inconveniente de que no todos los ciudadanos pueden acceder al mismo con facilidad, pues no se cuenta con la infraestructura para que la población tenga acceso a la intranet nacional de manera prolongada.

Boletines de la Gaceta:

Los boletines de la Gaceta circulan de manera impresa por el país, los mismos pueden ser encontrados en los estancillos de periódicos o en la oficina de correo postal. Estos boletines se presentan como una forma más para que la población pueda acceder a las publicaciones normativas de los órganos oficiales del estado. De igual manera presentan inconvenientes que se pueden medir en la cantidad de boletines, la periodicidad de las publicaciones, así como el costo que representa para el estado la impresión de los mismos.

⁴ ARAG son las iniciales de Allgemeine Rechtsschutz-Versicherungs-AG, es decir, Compañía de Seguros de Defensa Jurídica S.A.

xCuba Apk:

También tomamos como ejemplo a **xCuba**, una aplicación creada por jóvenes para conocer el sistema político y electoral cubano. Disponible para sistema Android contiene textos, audios, infografías, imágenes y frases para consultar, marcar y compartir en las redes sociales. En ella se puede encontrar la Constitución, la Ley Electoral, la organización del Estado cubano, la historia del Poder Popular, entre otros documentos (14).

Dicha aplicación surge para dar a conocer de una manera más eficiente el sistema político y electoral cubano, pero no aborda otras Leyes que son de igual importancia para la sociedad cubana. Aunque la misma si se estudió con el objetivo de tomar nuevas ideas para incluirlas en el trabajo, como por ejemplo la posibilidad de añadir notas en los artículos.

iLex Constitución v1.0:

iLex en su versión 1.0 es una aplicación para dispositivos móviles con sistema operativo Android que permite consultar el texto de la Constitución de la República de Cuba de forma rápida y práctica, a fin de que resulte más fácil a nuestro pueblo el conocimiento de la misma. Este software tiene como objetivo favorecer el acceso a la legislación cubana (solo la constitución) como requisito imprescindible para elevar la cultura jurídica y el conocimiento del ordenamiento jurídico. Todo ello representa un salto revolucionario respecto a las publicaciones impresas y a tono con la actual sociedad de la información.

Si bien iLex Constitución v1.0 es la principal base para la realización de este trabajo, cuenta con el inconveniente de que su estructura y diseño se adaptan únicamente a la Constitución de la República de Cuba. Esto impide que se le puedan adicionar otras Leyes por lo que no cumple con los objetivos principales del trabajo. Debido a esto se precisa realizar un producto totalmente nuevo que permita adicionar cualquier Ley que esté presente en la legislación cubana, no obstante, parte de su diseño visual se reutiliza a la hora de realizar el proyecto.

1.3.3 Análisis de las soluciones encontradas

Uno de los objetivos de este trabajo es la creación de un producto totalmente cubano, que responda a los intereses del MINJUS y le sirva a este como una herramienta más para ampliar su canal oficial de publicidad normativa. Además, garantizando un producto propio se garantiza también la integridad de la información a difundir. Es por ello que las soluciones internacionales antes presentadas se toman solo como base para la realización del trabajo. De estas se estudió la forma de organización de las Leyes, la

estructura de las aplicaciones, las diferentes opciones que brindan las mismas, así como los elementos más distintivos de su diseño visual.

A su vez se demuestra que las soluciones nacionales no satisfacen las necesidades actuales de este trabajo por los inconvenientes antes mencionados.

Por estas razones y teniendo en cuenta el avance constante de la telefonía móvil en el país, la aplicación iLex v2.0 se presenta como un producto innovador que permitirá tanto a profesionales como a personas naturales disponer de la legislación cubana en su propio Smartphone o Tablet, siempre y cuando dispongan de los mismos.

1.4 Metodología de Desarrollo y Tecnologías Seleccionadas

1.4.1 Metodología de desarrollo

Según la RAE una metodología es un conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal (15).

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 (16).

Existen diferentes metodologías de desarrollo que han sido en los últimos años herramientas de apoyo para el desarrollo de software. Las metodologías de desarrollo de software están divididas en dos grupos que se conocen como, metodologías tradicionales y metodologías ágiles. Las tradicionales, pensadas para hacer un uso exhaustivo de documentación durante todo el ciclo del proyecto son recomendadas para los proyectos con grandes equipos de desarrollo. Por otro lado, las ágiles hacen énfasis en la capacidad de dar respuesta a los cambios, promoviendo el trabajo en equipo y manteniendo una buena relación con el cliente (17).

El desarrollo de aplicaciones para proveer servicios móviles, difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos móviles, también difieran de las del software clásico (18).

Es por ello que se decide utilizar una metodología ágil, pues son una excelente alternativa para guiar proyectos de desarrollo de software de tamaño reducido, como es el caso de las aplicaciones para dispositivos móviles. Además, poseen una gran facilidad de adaptación y flexibilidad; pero estas necesitan ser adaptadas a las características especiales de los dispositivos con el fin de obtener productos de calidad.

Entre el grupo de metodologías ágiles más utilizadas se encuentran:

Agile Unified Process (AUP): Proceso Unificado Ágil en español, se describe como una metodología fácil de entender para el desarrollo de aplicaciones de software de negocio, utilizando técnicas ágiles y conceptos aun fieles a los de RUP, por lo tanto, es una versión simplificada del Rational Unified Process (RUP). Esta metodología, plantea un ciclo de vida iterativo, que se basa en la ampliación y refinamiento sucesivo del sistema, mediante múltiples iteraciones con retroalimentación cíclica y adaptación como elementos principales que dirigen para converger hacia un sistema adecuado (19).

Extreme Programming (XP): Programación extrema en español, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (20).

Crystal Methodologies: Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas (20).

Adaptive Software Development (ASD): Desarrollo Adaptable de Software en español. Sus principales características son el desarrollo iterativo, orientado a los componentes de software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo (20).

Feature Driven Development (FDD): Desarrollo Basado en Funciones en español. Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software (20).

Lean Development (LD): Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios (20).

Las metodologías ágiles anteriormente descritas pudieran darle solución al problema, pero por las características del software a desarrollar se determinó el uso del framework⁵ de desarrollo ágil Scrum. Esta metodología es ideal para proyectos pequeños con pocos integrantes en su equipo de desarrollo y presenta un alto grado de flexibilidad y adaptabilidad ante los cambios. Además, es una metodología con gran presencia en internet según el número de resultados obtenidos en las búsquedas por Yahoo, Google y Microsoft Live (21). Por otro lado, iLex Constitución v1.0 utilizó esta metodología en su implementación por lo que se decide continuar el desarrollo del proyecto bajo la misma.

En Scrum el trabajo es estructurado en ciclos de trabajo llamados Sprint, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente priorizados, llamados historias de usuarios, para que las características que sean desarrolladas primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto terminado y potencialmente comerciable (22).

Scrum más que una metodología de desarrollo de software, es una forma de auto-gestión de los equipos de desarrollo. Permitiendo seguir de forma clara el avance de las tareas realizadas, de forma que se pueden observar progresos en poco tiempo de trabajo. Además, es indicado para pequeños equipos de desarrollo y se orienta a una entrega rápida de resultados y una alta flexibilidad.

Scrum maneja de forma empírica la evolución del proyecto con las siguientes tácticas (23):

Revisión de las iteraciones:

⁵ La palabra inglesa "**framework**" (infraestructura, armazón, marco) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

- Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Es por tanto la duración del sprint, el período de tiempo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto.

Desarrollo incremental:

- El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar.
- ¿Por qué predecir la versión definitiva de algo que va a estar evolucionando de forma continua? Scrum considera a la inestabilidad como una premisa, y adopta técnicas de trabajo para facilitar la evolución sin degradar la calidad de la arquitectura y permitir que también evolucione durante el desarrollo.

Auto organización:

- Son muchos los factores impredecibles en un proyecto. La gestión predictiva asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución.
- En Scrum los equipos son auto organizados, con un margen de maniobra suficiente para tomar las decisiones que consideren oportunas.

Colaboración:

- Es un componente importante y necesario para que a través de la auto organización se pueda gestionar con solvencia la labor que de otra forma realizaría un gestor de proyectos.
- Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

Dentro de los artefactos que genera Scrum se encuentran (24):

Pila del producto: (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.

Pila del sprint: (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.

Sprint: nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time boxing).

Incremento: resultado de cada sprint.

1.4.2 Herramienta de Modelado

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

Visual Paradigm from UML 8.0 Enterprise Edition:

El Lenguaje Unificado de Modelado (UML), permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra el desarrollo de un software.

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Una de sus ventajas sobre las demás herramientas CASE es su condición multiplataforma, por lo que tiene la capacidad de ejecutarse sobre diferentes Sistemas Operativos. Es fácil de usar y presenta una agradable interfaz para interactuar con el usuario (25).

1.4.3 Sistema Operativo móvil

A la hora de desarrollar software hay que prestarle vital atención al sistema operativo que va a soportar la aplicación a desarrollar. Un Sistema Operativo (SO) es el software básico de una computadora que provee una interfaz entre el resto de programas del computador, los dispositivos hardware y el usuario. Las funciones básicas del SO son administrar los recursos de la máquina, coordinar el hardware y organizar archivos y directorios en dispositivos de almacenamiento (26). Entre los SO más utilizados para dispositivos móviles se encuentran Android, iOS, Windows Phone y BlackBerry. Gartner Inc. empresa consultora y de investigación de las tecnologías de la información desde el 2012 viene previendo un alto uso de Android a nivel internacional.

Comparación de sistemas operativos en millones de dispositivos

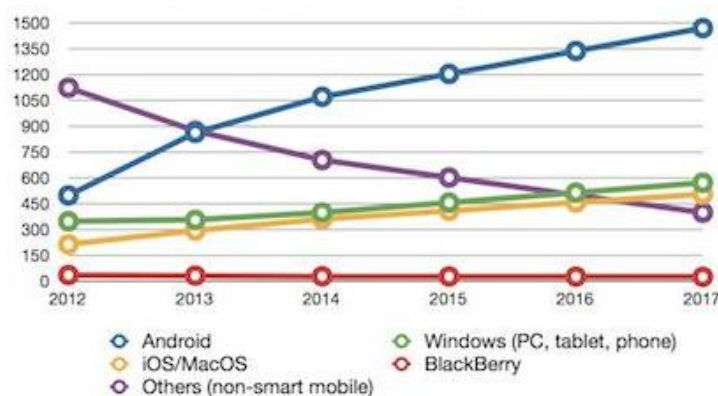


Figura 2: Comparación entre el uso de sistemas operativos.

Si bien iOS es el segundo SO más utilizado, Android reina en el mercado controlando el 82% a nivel mundial (27). Es posible encontrar tabletas (tablets), teléfonos móviles (celulares) y relojes equipados con Android, aunque el software también se usa en automóviles, televisores y otras máquinas. Además, al ser software libre es normal que muchos desarrolladores lo prefieran para realizar sus proyectos. Es por ello que se decide realizar este proyecto para dispositivos que utilicen Android, con el objetivo de que llegue a la mayor cantidad de personas posibles.

1.4.4 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado, del inglés Integrated Development Environment (IDE), es un programa informático compuesto por un conjunto de herramientas de programación. Los IDE proveen un marco de trabajo para los lenguajes de programación (28).

Android Studio:

Es el 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 ofrece funciones que aumentan la productividad durante la compilación de apps para Android, como las siguientes (29):

- Sistema de compilación flexible basado en Gradle⁶.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run, para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub, para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK.

⁶ Gradle es un sistema de compilación que reúne en un solo 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.

1.4.5 Lenguajes de Programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación (30).

Java como lenguaje nativo en Android:

Java es un lenguaje de programación orientado a objetos que se incorporó al ámbito de la informática en los años noventa. La idea de Java es que pueda realizarse programas con la posibilidad de ejecutarse en cualquier contexto, en cualquier ambiente, siendo así su portabilidad uno de sus principales logros. Fue desarrollado por Sun Microsystems⁷, posteriormente adquirido por Oracle⁸. 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 cualquiera, 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, su simpleza y sus grandes posibilidades de utilización (31).

Está especialmente extendido su uso en dispositivos móviles (31).

1.4.6 Android SDK

Android SDK siglas en inglés de Software Development Kit (kit de desarrollo de software) es un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, en este caso para Android. Este paquete o kit de desarrollo incluye las API siglas en inglés de Application Programming Interface (Interfaz de Programación de Aplicaciones) y herramientas

⁷ Sun Microsystems fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos.

⁸Oracle Corporation es una compañía de software que desarrolla bases de datos (Oracle Database) y sistemas de gestión de bases de datos. Cuenta, además, con herramientas propias de desarrollo para realizar aplicaciones, como Oracle Designer, Oracle JDeveloper y Oracle Developer Suite.

necesarias para desarrollar las aplicaciones, utilizando Java como lenguaje de programación (32).

1.4.7 JSON

(JavaScript Object Notation - Notación de Objetos de JavaScript) 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 JavaScript⁹. 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, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (33).

JSON está constituido por dos estructuras (33):

- 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 un 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(33).

1.4.8 SQLite

SQLite es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Permite almacenar información en dispositivos de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser un teléfono celular. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel (34).

SQLite se esfuerza por proporcionar almacenamiento de datos local para aplicaciones y dispositivos individuales. También hace hincapié en la economía, fiabilidad,

⁹ JavaScript (JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

independencia y simplicidad (34). Android incorpora SQLite en sus API¹⁰ por defecto para llevar a cabo de manera sencilla todas las tareas que se necesitan realizar en un proyecto. Como se muestra en la figura No. 3, las versiones de SQLite dependen de las versiones de la API (35).

Android API	SQLite Version
API 24	3.9
API 21	3.8
API 11	3.7
API 8	3.6
API 3	3.5
API 1	3.4

Figura 3: Versiones de SQLite según las API.

DB Browser for SQLite

DB Browser for SQLite es una herramienta de código abierto y de alta calidad para crear, diseñar y editar archivos de bases de datos compatibles con SQLite. Diseñada para usuarios y desarrolladores que desean crear bases de datos, así como buscar y editar datos. Utiliza una interfaz similar a una hoja de cálculo y no se necesita aprender comandos SQL complicados (36).

1.5 Pruebas de Software

En un proyecto de desarrollo de software los errores (bugs en inglés) puede presentarse en cualquiera de las etapas del ciclo de vida del mismo. Aun cuando se intente detectarlos después de cada fase utilizando técnicas como la inspección, algunos errores permanecen sin ser descubiertos. Por lo tanto, es muy probable que el código final contenga errores de requerimientos y diseño, adicionales a los introducidos en la codificación. Para ello las pruebas de software son los procesos que permiten verificar y revelar la calidad o el buen funcionamiento de un producto de software. Estas se integran dentro de las diferentes fases del ciclo del desarrollo de software con el objetivo de controlar el nivel de calidad y el correcto funcionamiento del producto (37).

¹⁰ Interfaz de programación de aplicaciones (API por sus siglas en inglés) no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como, por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc.

1.5.1 Casos de Pruebas

El diseño de casos de prueba puede requerir tanto esfuerzo como el propio diseño inicial del producto, se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la menor cantidad de esfuerzo y tiempo posible. Un caso de prueba se puede definir como un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados. Los casos de prueba son desarrollados para cumplir un objetivo en particular o una función esperada (38). Tienen como misión verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en la especificación de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

1.5.2 Pruebas de Caja Blanca (PCB)

Estas pruebas se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Las mismas (39):

- Verifican la correcta implementación de las unidades internas, las estructuras y sus relaciones.
- Hacen énfasis en la reducción de errores internos.
- Los métodos de caja blanca o estructurales permiten derivar casos de prueba que:
 1. Garanticen que todas las rutas independientes dentro del módulo se ejecuten al menos una vez.
 2. Ejecuten los lados verdadero y falso de todas las decisiones lógicas.
 3. Ejecuten todos los ciclos dentro y en sus límites operacionales.
 4. Ejerciten las estructuras de datos internas para asegurar su validez.

1.5.3 Pruebas de Caja Negra (PCN)

Estas pruebas tienen como objetivo estudiar el software desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. Las mismas(39):

- Verifican el correcto manejo de funciones externas soportadas por el software.
- Verifican que el comportamiento observado se apege a las especificaciones del producto y a las expectativas del usuario.

- Los casos de prueba se construyen a partir de las especificaciones del sistema.
- Los métodos de caja negra o funcionales permiten derivar casos de prueba que buscan encontrar los siguientes tipos de errores:
 1. Funciones incorrectas o faltantes.
 2. Errores de interfaz.
 3. Errores en estructuras de datos o en acceso a BD externas.
 4. Errores de comportamiento o desempeño.
 5. Errores de inicialización o término.

1.6 Conclusiones del Capítulo

En este capítulo la realización del estudio de los sistemas homólogos permitió determinar las principales características y funcionalidades de los sistemas de divulgación legal existentes a nivel tanto nacional como internacional. Además, la descripción de los principales conceptos posibilitó adquirir una mayor comprensión de los temas relacionados con el objeto de estudio y la necesidad de implementar una aplicación para dispositivos móviles que permita ampliar el canal oficial de divulgación legal en el país. Por otra parte, el análisis sobre las potencialidades, ventajas y desventajas de las diferentes metodologías, herramientas y tecnologías permitió alcanzar los conocimientos necesarios para seleccionar las adecuadas en el desarrollo de la solución. Por lo que, se decide utilizar como entorno de desarrollo integrado Android Studio, como metodología a utilizar Scrum, y como lenguaje de programación Java.

Capítulo II. Análisis y diseño de la solución propuesta

2.1 Introducción

En el presente Capítulo se analiza y diseña la solución propuesta guiada por la metodología de desarrollo Scrum. Se describen las diferentes características que debe cumplir el sistema a través de los requisitos funcionales y no funcionales identificados, cada uno de ellos asociados a su Sprint correspondiente. Se describen las Historias de Usuario ejemplificando dos de ellas de las 26 existentes. Además, se defiende la arquitectura Modelo-Vista-Presentador (MVP) como una buena práctica de patrón arquitectónico a la hora de desarrollar la aplicación.

2.2 Propuesta de Solución

El presente trabajo propone el desarrollo de la versión 2.0 de la aplicación para dispositivos móviles iLex. En su primera versión iLex ponía al alcance de todos los ciudadanos el texto íntegro de la Constitución de la República de Cuba. iLex 2.0 pretende ampliar sus servicios ofreciendo al usuario la posibilidad de consultar la legislación cubana en su totalidad, dígase Leyes, Decretos y Decretos-Ley.

Que se mantiene en ambas versiones:

1. Se reutilizó la interfaz visual para la visualización de los artículos.
2. Se reutilizó el componente para el cambio de tamaño y color de la fuente del texto.
3. Se reutilizó el método para la búsqueda local en los artículos.
4. Se reutilizó el componente para la edición de secciones de texto.
5. Se mantiene la opción de añadir artículos a favoritos.

Novedades de la versión 2.0:

1. Soporte para más Leyes.
2. Permite visualizar los títulos de las categorías (ejemplo de categorías, Libros, Títulos, Capítulos, Secciones).
3. Adición de notas a los artículos.
4. Migas de pan (para saber en qué parte del documento de Ley se encuentra el usuario).
5. Enviar artículos por aplicaciones de mensajería.
6. Descargar nuevas Leyes.

7. Suprimir Leyes que ya no sean de interés para el usuario.
8. Continuar la lectura por donde la dejo el usuario.

Todas estas características tratan de hacer más dinámica la relación entre los usuarios y la legislación del país. Además, presentan a iLex como una aplicación atractiva para los jóvenes de hoy en día que tan apegados están a las nuevas tecnologías.

2.3 Roles del Proyecto

La metodología Scrum define roles a cada persona que interactúa en el proceso de desarrollo del producto. En la tabla No.1 se muestran los roles del proyecto.

Tabla 1: Roles del proyecto.

Roles	
Propietario del producto	Universidad de las Ciencias Informáticas
Equipo de desarrollo	Felix A. Marrero Pentón
Scrum Master	MsC. Yarina Amoroso Fernández
Interesados	CEGEL, Unión de Juristas de Cuba, MINJUS, personas jurídicas o naturales de la República de Cuba.

2.4 El Sprint

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos de duración durante el cual se crea un incremento de producto "Terminado" utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprint es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior(24).

2.5 Requisitos del Sistema

En la ingeniería de software tradicional, los requisitos del sistema forman parte del proceso de adquisición, siendo por tanto responsabilidad del cliente la definición del problema y de las funcionalidades que debe aportar la solución. En Scrum el cliente (propietario del producto) comparte su visión con todo el equipo, y la pila del producto se realiza y evoluciona de forma continua con los aportes de todos (40).

Scrum, emplea dos formatos para registrar los requisitos:

Pila del producto (Product Backlog):

La pila del producto registra los requisitos vistos desde el punto de vista del cliente. Está formada por la lista de funcionalidades o "Historias de Usuario" que desea obtener el

cliente, ordenadas por la prioridad que el mismo da a cada una. Una pila del producto es un documento vivo, que evoluciona constantemente durante el desarrollo(24).

Pila del sprint (Sprint Backlog):

La pila del sprint refleja los requisitos vistos desde el punto de vista del equipo de desarrollo. Está formada por la lista de tareas en las que se descomponen las historias de usuario que se van a llevar a cabo en el sprint.

Estos dos últimos junto con el **incremento** conformaría los **artefactos** generados por la metodología Scrum, donde el incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada al cliente(24). A continuación, la tabla No. 2 muestra la lista de requisitos funcionales asociados cada uno de ellos a su Sprint.

Tabla 2: Requisitos funcionales.

No. Sprint	Duración (semanas)	Descripción
1	4	<i>Investigación previa del problema.</i>
2	3	<i>1. Desarrollar estructura JSON genérica. 2. Implementar estructura JSON genérica en la aplicación.</i>
3	3	<i>3. Desarrollar Menú Lateral. 4. Desarrollar Seleccionar Documento. 5. Desarrollar Listar Categorías. 6. Desarrollar Ver Documento.</i>
4	4	<i>7. Desarrollar Buscar en Artículo. 8. Desarrollar Agregar o Eliminar Favoritos. 9. Desarrollar Listar Favoritos. 10. Desarrollar Panel de Edición. 11. Poner Texto Seleccionado en Negrita. 12. Poner Texto Seleccionado en Cursiva. 13. Cambiar Color de Texto Seleccionado. 14. Ver nota. 15. Editar nota. 16. Enviar Artículo. 17. Desarrollar ¿Dónde estoy? (Migas de pan). 18. Guardar Personalización de la Fuente. 19. Eliminar Personalización de la Fuente.</i>

5	1	20. Desarrollar Cambiar Tamaño de Texto. 21. Desarrollar Cambiar Color de Texto. 22. Desarrollar Navegar entre Páginas.
6	4	23. Desarrollar Suprimir Documento. 24. Desarrollar Descargar Documento.
7	1	25. Desarrollar Continuar Lectura. 26. Desarrollar Menú Inferior.

Además de los requisitos funcionales, están los **no funcionales**, de estos últimos lo único que debe cumplirse para poder hacer un buen uso de iLex es contar con:

- **Sistema Operativo (SO):** Android.
- **Versión del SO:** 4.0 o superior.
- **Procesador:** ARMv6 a 800 Mhz o superior.
- **Memoria RAM:** mínimo 256 Mb.
- **Memoria ROM:** mínimo 256 Mb.
- **Espacio en Almacenamiento:** mínimo 30 Mb.

No es imprescindible que sea un teléfono móvil, también se pueden emplear tabletas o usar un pc haciendo uso de un emulador de Android.

2.6 Descripción de Historias de Usuarios

Las Historias de Usuario (HU) son utilizadas en los métodos ágiles para la especificación de requisitos, son una descripción breve de una funcionalidad del software o requisito funcional tal y como la percibe el usuario. Describen lo que el cliente o el usuario quiere que se implemente y se escriben con una o dos frases utilizando el lenguaje común del usuario. Cada historia de usuario debe ser limitada para poder memorizarla fácilmente (41).

La estructura de una HU está compuesta por:

Número: Se especifica un identificador para cada HU.

Nombre del requisito: Se especifica el nombre del requisito funcional asociado a la HU.

Nombre del programador: Se especifica el nombre del programador que va a desarrollar el requisito asociado a la HU.

Iteración Asignada: En el caso de la metodología utilizada (SCRUM) se le asigna a cada HU el Sprint al que pertenecen.

Prioridad: Se especifica la prioridad del requisito asociado a la HU que puede ser alta, media o baja en dependencia del negocio.

Tiempo estimado: Se especifica el tiempo que se estima demorará la realización del requisito asociado a la HU.

Tiempo real: Se especifica el tiempo real que demora la realización del requisito asociado a la HU.

Riesgo en desarrollo: Se especifican los riesgos que pudiera tener la realización del requisito asociado a la HU.

Descripción: Se describe con lujo de detalle en que consiste el requisito asociado a la HU.

Observación: Se plantea alguna observación que sea de relevancia.

Prototipo elemental de interfaz gráfica de usuario: Se muestra una imagen de cómo quedaría la interfaz gráfica del requisito asociado a la HU.

A continuación, se muestran en las tablas No. 3 y No. 4 dos HU de un total de 26 que se definieron en la realización del trabajo.

Tabla 3: Historia de Usuario No. 5.

Número:5	
Número:5	Nombre del requisito: Listar categorías.
Programador: Felix Antonio Marrero Pentón	Iteración Asignada: Sprint 3
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: Posibles cambios o modificaciones en las Leyes o en su estructura.	Tiempo Real: 1 semana
Descripción: Se muestra un listado con todas las categorías del documento consultado.	
Observaciones: Las categorías pueden ser libro, título, capítulo, sección, artículo, etc.	

Prototipo elemental de interfaz gráfica de usuario:

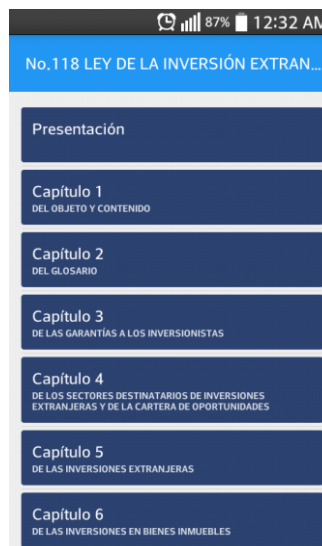


Tabla 4: Historia de Usuario No. 20.

Historia de Usuario	
Número: 20	Nombre del requisito: Cambiar Tamaño de Texto
Programador: Felix Antonio Marrero Pentón	Iteración Asignada: Sprint 5
Prioridad: Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo:	Tiempo Real: 1 día
Descripción: La aplicación debe permitirle al usuario aumentar y disminuir el tamaño del texto a través de la barra que se muestra en el inferior de la vista.	
Observaciones:	
<h3>Prototipo elemental de interfaz gráfica de usuario:</h3> <div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> </div> <div style="width: 45%;"> </div> </div>	

Para ver las demás historias de usuarios revisar el documento ***Historias de usuario iLex 2.0***.

2.7 Patrón Arquitectónico

Todo proyecto bien desarrollado debe basarse en un patrón arquitectónico con el objetivo de simplificar el código, tener organización en el proyecto y garantizar la correcta realización de test unitarios a la aplicación. Android no se excluye de esto, es por eso que para la realización de la aplicación iLex se plantea la utilización del patrón arquitectónico Modelo-Vista-Presentador MVP (42).

MVP es un patrón arquitectónico que surge para ayudar a realizar pruebas automáticas de la interfaz gráfica, para ello la idea es codificar la interfaz de usuario lo más simple posible, teniendo el menor código posible, de forma que no merezca la pena probarla. 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.

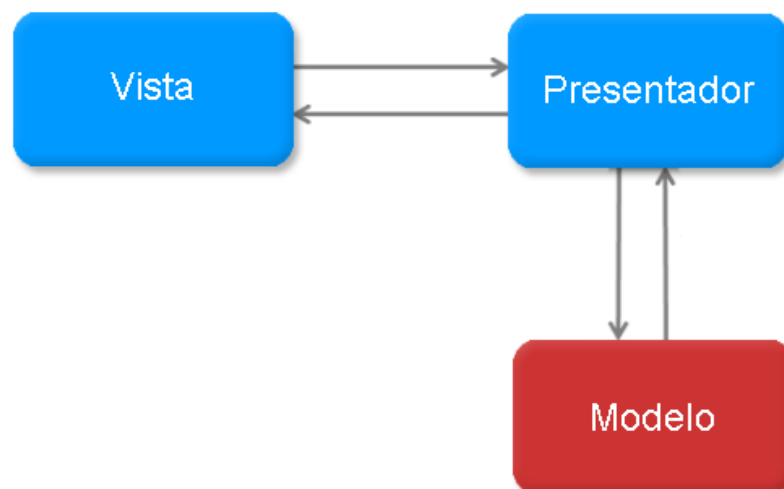


Figura 4: Representación del patrón Modelo Vista Presentador.

Básicamente este patrón consiste en 3 componentes:

- **La vista:** Es la interfaz gráfica del usuario. Está compuesta de las ventanas y controles que forman la interfaz la aplicación.
- **El modelo:** Es donde se encuentran las clases de acceso a datos de la lógica del negocio.
- **El presentador:** Es una capa intermediaria entre la vista y el modelo de datos. Este permite conectar la interfaz gráfica con los datos, que los recupera del modelo y se los devuelve a la vista formateados.

El diagrama de paquetes de la figura No. 7 muestra el uso de la arquitectura MVP en el proyecto.

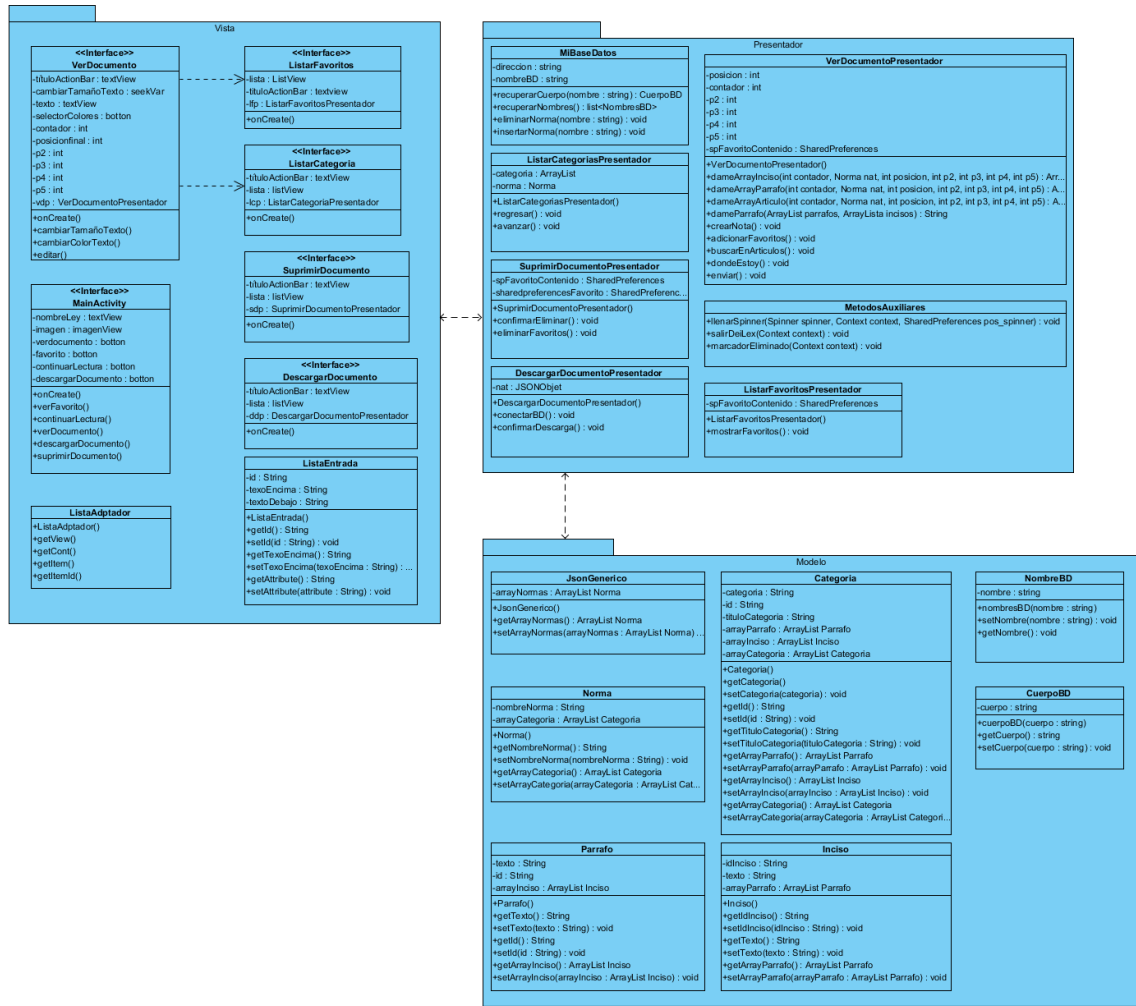


Figura 5: Diagrama de paquetes.

2.8 Patrones de Diseño

Cuando existen situaciones o problemas comunes y ampliamente observados, surgen soluciones que pueden aplicarse, basándose en una resolución anterior. Durante el diseño Orientado a Objetos es frecuente encontrarse repetidamente con estos tipos de problemas, es por ello que para analizar, compartir y documentar el conocimiento sobre estos problemas se han desarrollado los patrones de diseño.

Existen clasificaciones para estos patrones entre los que se encuentran los Patrones Generales de *Software* para Asignar Responsabilidades (GRASP por sus siglas en inglés) y los Patrones del Grupo de los Cuatro (GOF). A continuación, se presentan los patrones de diseño que son utilizados para el desarrollo de la solución.

2.8.1 Patrones GRASP

Los patrones GRASP nos dan unos principios generales para asignar responsabilidades a objetos, expresados en forma de patrones (43).

A continuación, se explican los principales patrones GRASP utilizados (43):

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). En una clase, que contiene toda la información necesaria para realizar la labor es donde se debe implementar el método. Este patrón se evidencia en la clase VerDocumento.java pues como es la que va a mostrar los artículos como tal, es la que se encarga de la personalización del texto.

Creador: Define quien es el encargado (o quien debería tener la responsabilidad) de crear un determinado objeto. Asigna la responsabilidad a una clase de crear objetos, práctica muy frecuente en los sistemas orientados a objetos. Entre sus beneficios se encuentra el brindar soporte a un bajo acoplamiento lo que trae como consecuencia una menor dependencia respecto al mantenimiento y la reutilización en mayor medida. Se pone de manifiesto en la clase MiBaseDatos.java.

Bajo Acoplamiento: Debe haber pocas dependencias entre las clases. El patrón propone el diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas. Por supuesto es casi imposible que no existan dependencias entre algunas clases de un proyecto. Como se muestra en la figura No. 8 en el software realizado solamente depende la clase VerDocumento.java de las clases ListarFavoritos.java y ListarCategoria.java.

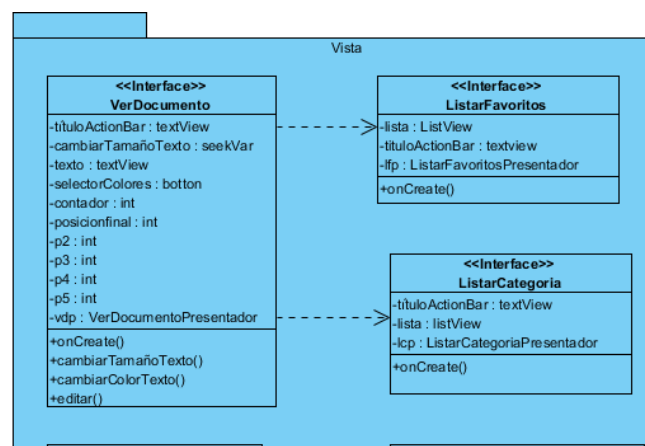


Figura 6: Dependencia de la clase VerDocumento.java.

Alta Cohesión: Asigna las responsabilidades en función de mantener la complejidad dentro de los límites posibles. La cohesión es una medida que refleja cuán enfocadas y relacionadas se encuentran las responsabilidades en una clase. Las clases que

presentan una alta cohesión se caracterizan por poseer responsabilidades estrechamente relacionadas que no hagan un enorme trabajo.

2.8.2 Patrones GOF

Los patrones del Grupo de los Cuatro (GOF¹¹ por sus siglas en inglés) se dividen en tres categorías: patrones de creación, patrones estructurales y patrones de comportamiento. A continuación, se describe el patrón utilizado en la solución (44):

Adaptador (del inglés **Adapter**): convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permite que cooperen clases que de otra manera no podrían por tener interfaces incompatibles (44). Como se aprecia en la figura No. 9 la clase ListaAdaptador.java posibilita que la interfaz de la clase ListarCategorias.java se combine con la interfaz de la clase ListaEntrada.java y tome esta forma:



Figura 7: Interfaz de la clase ListarCategorias.java.

2.9 Almacenamiento de los datos

Para el almacenamiento de los datos se utilizó una base de datos SQLite con el objetivo de almacenar el contenido en la aplicación de manera persistente, es decir, que no se pierdan los datos cuando cerramos la aplicación. Dicha base de datos se compone por una tabla de nombre “**norma**” que contiene dos tuplas: “**nombre**” de tipo *string* como “*llave primaria*” donde se almacena el nombre de la Ley y “**cuerpo**” también de tipo *string* donde se almacena una estructura *json* que representa el cuerpo de la Ley como tal.

En un principio se manejaba la idea de crear la base de datos externa a la aplicación. De esta manera el usuario al instalar la aplicación tendría buscar la ruta a la base de datos que se encontraría en la memoria interna o externa del dispositivo. Esta idea se desechó pues sería muy fácil para una persona acceder a los datos, modificarlos y luego difundir la aplicación con información falsa. Es por ello que se optó por la opción de

¹¹ Gang of Four (Grupo de los Cuatro): Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

introducir la base de datos internamente en la aplicación pues de esta manera se hace más difícil modificar su contenido y luego compartir la aplicación a otras personas con información incorrecta.

2.10 Estándar de Codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (45).

Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto permite que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea fácil de mantener (46).

Para la realización de iLex se definió el siguiente estándar de codificación:

Número de declaraciones por línea:

Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

Ejemplo:

```
public int bandera = 0;
public static String colorStatic;
public static String textStatic;
public static String sizeStatic;
public boolean hereEdit = false;
public CustomEditText customEditText;
```

Figura 8: Declaración de variables.

Espacio en blanco:

Se debe usar una línea en blanco entre:

- Métodos.
- Variables locales de un método y la primera sentencia.
- Entre diferentes secciones lógicas dentro de un fichero (más legibilidad).

Ejemplo:

```
private void setToolbar() {
    toolbar = (Toolbar) findViewById(R.id.toolbar);

    if (toolbar != null) {
        setSupportActionBar(toolbar);
    }
}
```

Figura 9: Espacios en blanco.

Asignación de nombres:

Cada tipo de elemento debe nombrarse con una serie de reglas determinadas.

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

Ejemplo:













-   ListaAdaptador
-   ListaEntrada
-   ListarCategoriasActivity
-   ListarFavoritosActivity
-   MainActivity
-   MiBaseDatos

Figura 10: Nombre de las clases.

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

Ejemplo:

```
private void setToolbar() {...}

private void initView() {...}
```

Figura 11: Nombre de los métodos.

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

Ejemplos:

```
public int contador = 1;  
public int posicion = 0;
```

Figura 12: Nombres de las variables.

2.11 Conclusiones del Capítulo

En este Capítulo se analizó la propuesta de solución presentada. Basándose en la metodología seleccionada, se definieron los roles del proyecto y requisitos del sistema, tanto funcionales como no funcionales. Se realizó una muestra de las historias de usuarios y se definió el patrón arquitectónico y los patrones de diseño a utilizar.

Capítulo III. Validación del producto

3.1 Introducción

El presente capítulo aborda la fase de la realización de las pruebas de software a la solución propuesta. Se describen las no conformidades encontradas en las pruebas y sus soluciones con el objetivo de asegurar la calidad y el buen funcionamiento del producto. Por último, se valida la solución y se realiza una comparación cualitativa entre la solución de la presente investigación y su versión anterior.

3.2 Validación de la Solución

Ian Sommerville en su libro Ingeniería del Software 7ma edición define que la validación tiene como objetivo: “asegurar que el sistema de software satisface las expectativas del cliente”. Así como también establecer la seguridad de que el software está hecho para un propósito, o sea, debe ser lo suficientemente bueno para su uso pretendido. A continuación, se describen los resultados de las pruebas realizadas para garantizar la calidad de la solución (47).

3.2.1 Resultados de las Pruebas de Caja Blanca

Se utilizó la herramienta SonarQube para la realización de las pruebas de caja blanca. Esta es una plataforma para evaluar código fuente. Es software libre y usa diversas herramientas de análisis estático de código fuente como Checkstyle, PMD o FindBugs para obtener métricas que pueden ayudar a mejorar la calidad del código de nuestros programas. Además, tiene soporte para más de 20 lenguajes de programación entre los que se encuentran Java, C#, C / C++, PL / SQL, Cobol, ABAP, Python, JavaScript (48).

Algunas de las métricas que usa SonarQube para analizar el código fuente son:

Bugs y Vulnerabilidades: Los Bugs (errores) y vulnerabilidades pueden tratarse de malas declaraciones en las variables.

Tamaño: Número de líneas de código, sin contar los comentarios.

Complejidad: Cuando se habla de complejidad en SonarQube se está hablando de complejidad ciclomática, que según Thomas McCabe en 1976 argumentaba como la complejidad del código puede obtenerse desde su flujo de control, o dicho de una manera más exhaustiva del número de rutas linealmente independientes del código.

Código Duplicado: Por defecto se considera una duplicación si 10 líneas sucesivas de código Java están repetidas, pero se puede modificar este límite a través de los ajustes de la herramienta.

Se analizó el código fuente de manera general, haciendo énfasis en las clases principales como son VerDocumento.java y ListarCategorias.java.

Los resultados obtenidos al analizar iLex con la herramienta SonarQube son los siguientes:

La herramienta detectó 16 bugs y 12 vulnerabilidades en el código fuente.

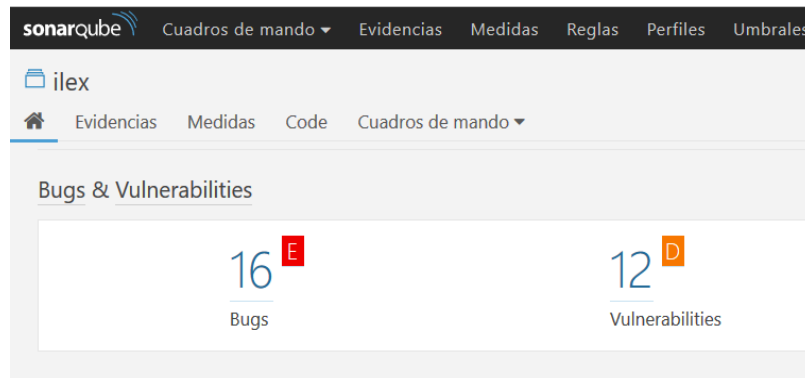


Figura 13: Resultado SonarQube. Bugs y vulnerabilidades.

Entre los bugs y vulnerabilidades detectados se encontraban, variables incorrectamente nombradas, así como variables declaradas como públicas innecesariamente. A continuación, la tabla No. 5 muestra algunos de los errores encontrados.

Tabla 5: Bugs y vulnerabilidades.

No.	Bugs	Clase	Vulnerabilidades	Clase
1	SharedPreferences <code>pos_spinner;</code>	class VerDocumentoActivity	<code>public int p2 = 0;</code>	class VerDocumentoActivity
2	<code>int</code> <code>posicion_spinner;</code>		<code>public int p3 = 0;</code>	
3	<code>int</code> <code>bd_size;</code>		<code>public int posicion = 0;</code>	
4	String <code>subtitulo_favorito = "";</code>		<code>public int contador = 1;</code>	
7	SharedPreferences <code>prefe_anterior;</code>	class ListarCategoriasActivity	<code>public static</code> String <code>colorStatic;</code>	
8	<code>int</code> <code>var_anterior = prefe_anterior.getInt("var_anterior", 0);</code>		<code>public static</code> String <code>textStatic;</code>	
9	<code>edmarc.putIn("key_marcador", 0);</code>		<code>public</code> Norma <code>nat;</code>	

10	<pre>int aux = prefeg.getInt("posicion_1", 0);</pre>		<pre>public int contador;</pre>	
----	--	--	---------------------------------	--

Además, la herramienta detectó un total de 7 368 líneas de código en todo el proyecto de manera general y una complejidad ciclomática general de 1 270.

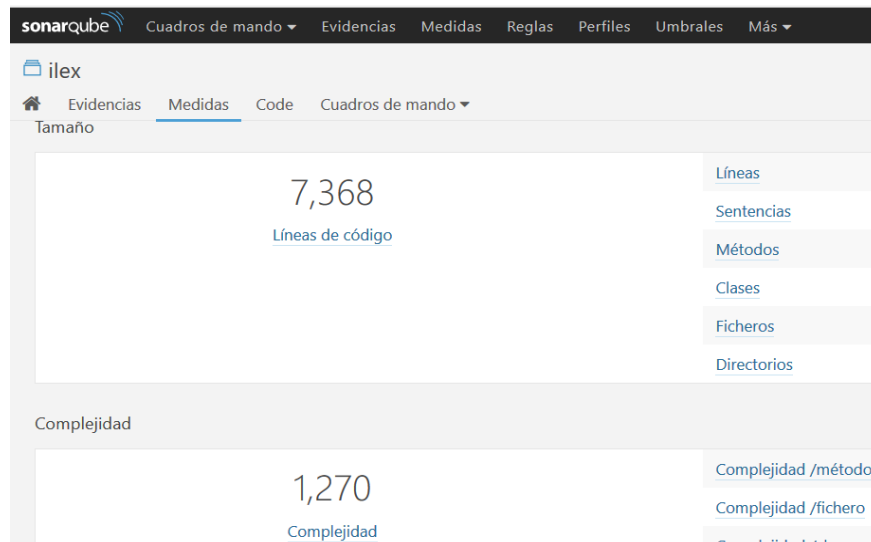


Figura 14: Resultado SonarQube. Líneas de código.

Lo que se refiere a la complejidad por métodos, 23 de estos superaban el máximo de complejidad establecido que es de 10, mostrando una complejidad de 12.

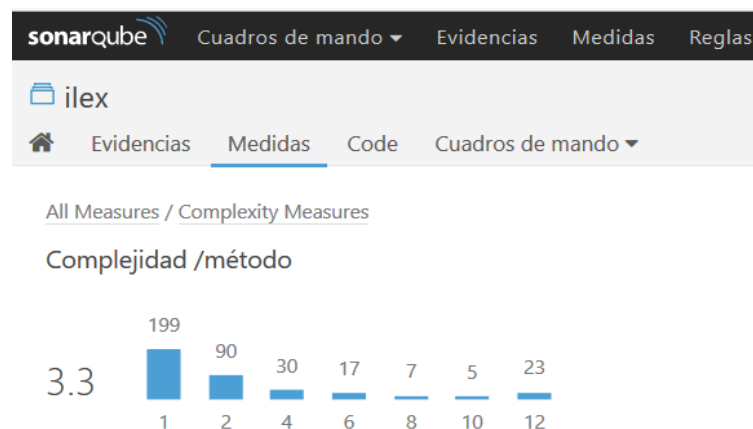


Figura 15: Resultado SonarQube. Complejidad.

A este problema se le dio solución optimizando los métodos con el objetivo de que no contengan líneas de código duplicadas ni ciclos repetidos innecesariamente. Un ejemplo con el método **dameParrafo()** de la clase `VerDocumentoPresentador.java` que se encarga de mostrar el contenido del artículo seleccionado. En un principio el método era el que se muestra en la figura No. 18.

```

public String dameParrafo(ArrayList<Parrafos> pa, ArrayList<Inciso> inc) {
    String parrafo = "";
    if (inc == null) {
        for (int p = 0; p < pa.size(); p++) {
            if (pa.get(p).getArray_Inciso() == null) {
                parrafo = parrafo.concat(pa.get(p).getTexto());
                parrafo = parrafo.concat("\n" + "\n");
            } else {
                parrafo = parrafo.concat(pa.get(p).getTexto());
                parrafo = parrafo.concat("\n" + "\n");

                ArrayList<Inciso> n = pa.get(p).getArray_Inciso();
                String incisos = "";
                for (int i = 0; i < n.size(); i++) {
                    incisos = incisos.concat(n.get(i).getId_inciso() + " ");
                    ArrayList<Parrafos> pi = n.get(i).getArray_parrafo();
                    for (int y = 0; y < pi.size(); y++) {
                        incisos = incisos.concat(pi.get(y).getTexto());
                        incisos = incisos.concat("\n" + "\n");
                    }
                }
                parrafo = parrafo.concat(incisos);
            }
        }
    }
}

if (pa == null) {
    for (int p = 0; p < inc.size(); p++) {
        boolean bool = false;
        ArrayList<Parrafos> parr = inc.get(p).getArray_parrafo();
        for (int y = 0; y < parr.size(); y++) {
            if (inc.get(p).getArray_parrafo().get(y).getArray_Inciso() == null) {
                if (!bool) {
                    parrafo = parrafo.concat(inc.get(p).getId_inciso() + ". ");
                    bool = true;
                }
                parrafo = parrafo.concat(inc.get(p).getArray_parrafo().get(y).getTexto());
                parrafo = parrafo.concat("\n" + "\n");
            } else {
                parrafo = parrafo.concat(inc.get(p).getId_inciso() + ". ");
                bool = true;
                parrafo = parrafo.concat(inc.get(p).getArray_parrafo().get(y).getTexto());
                parrafo = parrafo.concat("\n" + "\n");

                ArrayList<Inciso> n = parr.get(y).getArray_Inciso();
                String incisos = "";
                for (int i = 0; i < n.size(); i++) {
                    incisos = incisos.concat(n.get(i).getId_inciso() + " ");
                    ArrayList<Parrafos> pi = n.get(i).getArray_parrafo();
                    for (int y2 = 0; y2 < pi.size(); y2++) {
                        incisos = incisos.concat(n.get(i).getArray_parrafo().get(y2).getTexto());
                        incisos = incisos.concat("\n" + "\n");
                    }
                }
                parrafo = parrafo.concat(incisos);
            }
        }
    }
}

return parrafo;
}

```

Figura 16: Método dameParrafo().

Las llaves de colores indican las líneas de códigos duplicadas innecesariamente. Las líneas de código marcadas con las llaves rojas contienen ciclos anidados que aumentan la complejidad ciclomática del método, por lo que se decide realizar en un método aparte **llenarInciso()** para esta sección de código.

```

public String llenarInciso(String parrafo, ArrayList<Parrafos> parr, int y){
    ArrayList<Inciso> n = parr.get(y).getArray_Inciso();
    String incisos = "";
    for (int i = 0; i < n.size(); i++) {
        incisos = incisos.concat(n.get(i).getId_inciso() + " ");
        ArrayList<Parrafos> pi = n.get(i).getArray_parrafo();
        for (int y2 = 0; y2 < pi.size(); y2++) {
            incisos = incisos.concat(n.get(i).getArray_parrafo().get(y2).getTexto());
            incisos = incisos.concat("\n" + "\n");
        }
    }
    return parrafo = parrafo.concat(incisos);
}

```

Figura 17: Método llenarInciso().

Además, las secciones de texto marcadas por las llaves azules y verdes también se implementaron en métodos aparte, **aux1()** y **aux2()** respectivamente como se muestra en la figura No. 20.

```

public String aux1(String parrafo, ArrayList<Parrafos> pa, int p){
    parrafo = parrafo.concat(pa.get(p).getTexto());
    return parrafo = parrafo.concat("\n" + "\n");
}

public String aux2(String parrafo, ArrayList<Inciso> inc, int p, int y){
    parrafo = parrafo.concat(inc.get(p).getArray_parrafo().get(y).getTexto());
    return parrafo = parrafo.concat("\n" + "\n");
}

```

Figura 18: Métodos aux1() y aux2().

Por otro lado, el código marcado con las llaves azul claro y naranja se implementó en dos métodos auxiliares más, **siIncisoVacio()** figura No.21 y **siParrafoVacio()** figura No. 22 respectivamente:

```

public String siIncisoVacio(ArrayList<Parrafos> pa, ArrayList<Inciso> inc, String parrafo){
    if (inc == null) {
        for (int p = 0; p < pa.size(); p++) {
            if (pa.get(p).getArray_Inciso() == null) {
                parrafo = aux1(parrafo, pa, p);
            } else {
                parrafo = aux2(parrafo, pa, p);
            }
            parrafo = llenarInciso(parrafo, pa, p);
        }
    }
    return parrafo;
}

```

Figura 19: Método siIncisoVacio().

```

public String siParrafoVacio(ArrayList<Parrafos> pa, ArrayList<Inciso> inc, String parrafo){
    if (pa == null) {
        for (int p = 0; p < inc.size(); p++) {
            boolean bool = false;
            ArrayList<Parrafos> parr = inc.get(p).getArray_parrafo();
            for (int y = 0; y < parr.size(); y++) {
                if (inc.get(p).getArray_parrafo().get(y).getArray_Inciso() == null) {
                    if (!bool) {
                        parrafo = parrafo.concat(inc.get(p).getId_inciso() + ". ");
                        bool = true;
                    }
                    parrafo = aux2(parrafo, inc, p, y);
                } else {
                    parrafo = parrafo.concat(inc.get(p).getId_inciso() + ". ");
                    bool = true;
                    parrafo = aux2(parrafo, inc, p, y);
                }
                parrafo = llenarInciso(parrafo, parr, y);
            }
        }
    }
    return parrafo;
}

```

Figura 20: Método siParrafoVacio().

Estos métodos ya realizan las llamadas a los métodos **llenarInciso()**, **aux1()** y **aux2()** como se ve en las secciones sombreadas. Al final el método **dameParrafo()** quedó como se observa en la figura No. 23.

```

public String dameParrafo(ArrayList<Parrafos> pa, ArrayList<Inciso> inc) {
    String parrafo = "";
    parrafo = siIncisoVacio(pa, inc, parrafo);
    parrafo = siParrafoVacio(pa, inc, parrafo);
    return parrafo;
}

```

Figura 21: Método dameParrafo() optimizado.

Además de eso el software detectó un 18% de código duplicado.

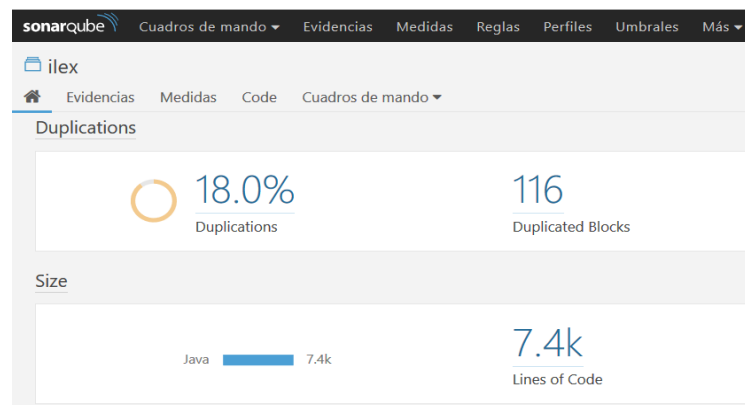


Figura 22: Resultado SonarQube. Código duplicado.

Este problema se le dio solución reutilizando el código, en lugar de tener el mismo método en varias clases, una clase auxiliar contiene un único método al que acceden las demás clases. Por ejemplo, los métodos **salirDeiLex()** y **llenarSpinner()** que se muestran en las figuras No. 25 y No. 26 respectivamente.

```

public void salirDeiLex(Context context) {
    mMaterialDialog = new MaterialDialog(context);
    mMaterialDialog.setTitle("Confirmación")
        .setMessage("¿Está seguro que desea salir de iLex?")
        .setPositiveButton("Aceptar", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                System.exit(0);
                mMaterialDialog.dismiss();
            }
        })
        .setNegativeButton(
            "Cancelar", (v) -> {
                mMaterialDialog.dismiss();
            }
        )
        .setCanceledOnTouchOutside(false)
        .setOnDismissListener(
            new DialogInterface.OnDismissListener() {
                @Override
                public void onDismiss(DialogInterface dialog) {
                }
            }
        )
        .show();
}

```

Figura 23: Método salirDeiLex().

```

public void llenarSpinner(Spinner spinner, Context context, final SharedPreferences pos_spinner) {
    MiBaseDatos MDB = new MiBaseDatos(context);
    int bdsiz = MDB.recuperarNOMBRES().size();
    String[] nombres = new String[bdsiz];
    for (int i = 0; i < bdsiz; i++) {
        nombres[i] = MDB.recuperarNOMBRES().get(i).getNOMBRE();
    }
    spinner.setAdapter(new ArrayAdapter<String>(context, R.layout.spinner_item, nombres));
    spinner.setSelection(pos_spinner.getInt("posicion_spinner", 0));
    spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
            SharedPreferences.Editor editor = pos_spinner.edit();
            editor.putInt("posicion_ra", position);
            editor.putInt("posicion_spinner", position);
            editor.commit();
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
}

```

Figura 24: Método llenarSpinner ().

Como se puede apreciar son métodos algo extenso que se estaban implementado de manera innecesaria en las clases VerDocumentoPresentador.java, ListarFavoritosPresentador.java, DescargarDocumentoPresentador.java, SuprimirDocumentoPresentador.java y MainActivity.java. Al implementar dichos métodos en una clase auxiliar a la que acceden las demás, el código dejó de ser duplicado innecesariamente.

3.2.2 Resultados de las Pruebas de Caja Negra

En la realización de las pruebas de caja negra se utilizaron las descripciones de las historias de usuarios como casos de pruebas, por ejemplo, la HU número 4: Seleccionar Documento en su descripción dice lo siguiente:

Descripción: *Al desplegar el menú lateral, se debe presionar sobre una franja rectangular oscurecida que aparece en el mismo y que contiene el nombre de un documento de Ley. Inmediatamente se desplegará un listado con los documentos de Ley existentes en la aplicación para que el usuario seleccione cual desea consultar.*

El resultado se considera satisfactorio si la aplicación muestra lo que describe la HU. En este caso, es evidente que se despliega un listado con las Leyes que contiene la aplicación como se muestra en la figura No. 27.



Figura 25: Listado de Leyes.

Otro ejemplo de casos de prueba con la HU número 20: Cambiar Tamaño de Texto. Su descripción dice lo siguiente:

Descripción: La aplicación debe permitirle al usuario aumentar y disminuir el tamaño del texto desplazando la barra que se muestra en el inferior de la vista a la derecha o a la izquierda respectivamente.

En este otro caso el resultado que se muestra en la figura No. 28 es satisfactorio. Como indica la descripción de la HU, el tamaño de texto aumenta cuando se desplaza la barra hacia la derecha y disminuye en caso contrario,.

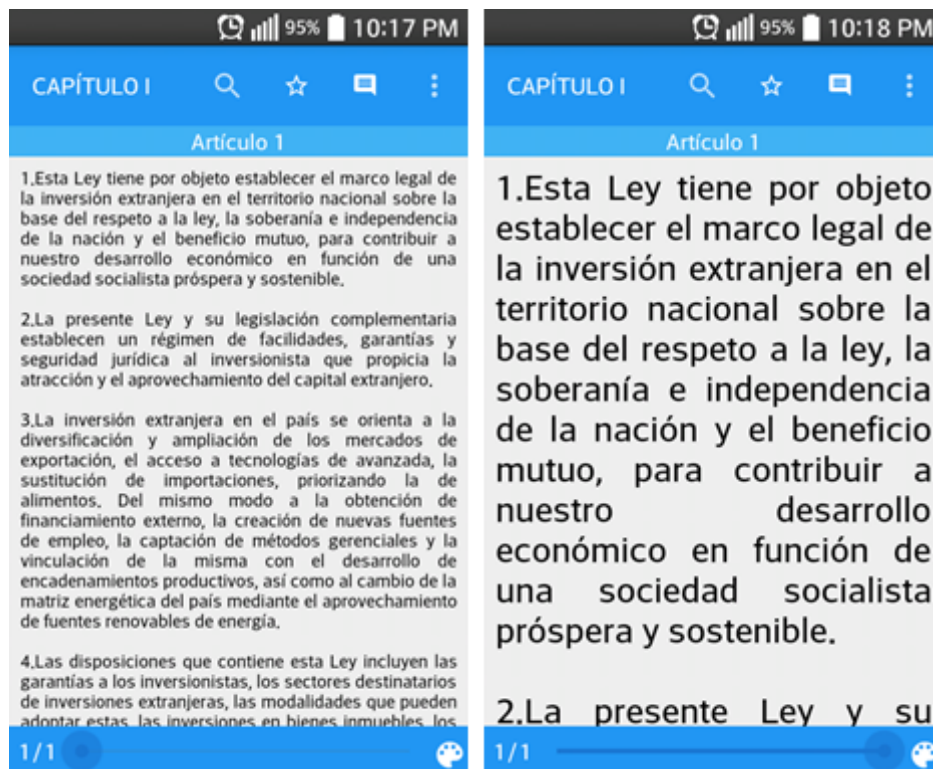


Figura 26: Cambiar tamaño de texto.

No Conformidades (NC)

Según la norma ISO 9000:2005, una No Conformidad es un incumplimiento de un requisito del sistema, sea este especificado o no. Se conoce como requisito una necesidad o expectativa establecida, generalmente explícita u obligatoria (49).

La no conformidad en un sistema de calidad, se origina cuando se está incumpliendo con los requisitos del manual de calidad. Además, también se toma como no conformidad faltas ortográficas o errores en el diseño que puedan entorpecer el uso de la aplicación por parte de los usuarios. En la figura No. 29 se muestra un gráfico con el número de no conformidades encontradas por iteraciones en iLex.

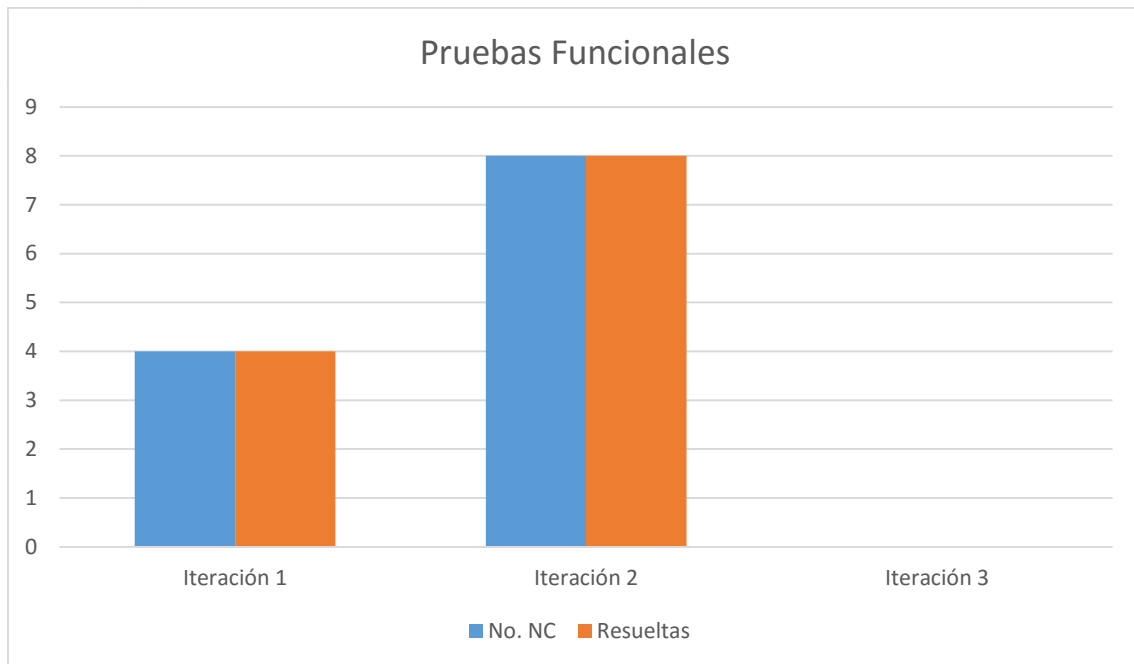


Figura 27: Gráfico. Resultado de pruebas funcionales.

En la primera iteración se encontraron 4 NC:

- En la interfaz Edición de nota, el botón Cancelar y Borrar realizan lo mismo.
- En la interfaz Edición de nota, en el campo para escribir el texto, poner Entre la nota, y no “entre nota”.
- En el Menú Lateral aparecen las segundas palabras con mayúsculas. Éstas deberían estar en minúsculas. Por ejemplo, sustituir “Continuar Lectura” por “Continuar lectura”.
- Falta de ortografía.

Se le dio solución a las 4. Luego se realizó una segunda iteración donde se encontraron 8 NC:

- No hay botón para volver atrás, cuando se están listando las categorías el usuario solo puede volver atrás con los botones físicos del móvil.
- El buscar en los artículos no distingue entre palabras con tilde o sin tilde.
- Cambiar el nombre del botón “Inicio” en el menú lateral por “Portada”, para evitar confusiones al usuario.
- El tamaño del texto “Documentos legales” en el menú lateral era muy pequeño.
- La demás encontradas eran faltas ortográficas.

Fueron resueltas las 8 NC, y por último la tercera iteración arrojó 0 NC permitiendo que el departamento de calidad emitiera el documento correspondiente para la liberación del producto (ver Anexo 2).

3.3 Validación de las variables de investigación

Con el objetivo de conocer el nivel de aceptación por parte de la población se publicó la aplicación iLex 2.0 en varios sitios de renombre nacional como son **www.humanos.uci.cu** (Comunidad del software libre en cuba con sede en la UCI) y **www.jorgen.cubava.cu** considerado por muchos la comunidad android de cuba. En estos sitios se tiene en cuenta la cantidad de descargas, los comentarios de los usuarios y los datos arrojados por una encuesta anexada a la publicación con la siguiente pregunta:

Cree usted que para propiciar un acercamiento mayor de la población a la legislación del país la aplicación iLex es:

- Excelente
- Buena
- Regular
- Mala

Estos son los datos arrojados por el sitio **www.humanos.uci.cu**:

Hasta el momento como se aprecia en la figura No. 30 se contaba con un total de 440 descargas, y de 122 votos un 43% afirma que la aplicación es Excelente, un 28% que es Buena, un 16% que es Regular y un 13% que es Mala.



Figura 28: Resultado de encuesta en sitio www.humanos.uci.cu.

Estos son los datos arrojados por el sitio www.jorgen.cubava.cu:

En este sitio se desconoce la cantidad de descargas y de un total 53 votos un 57% de los usuarios afirman que la aplicación es Excelente, un 36% que es Buena, un 8% que es Regular y un 0% que es Mala, ver figura No. 31.

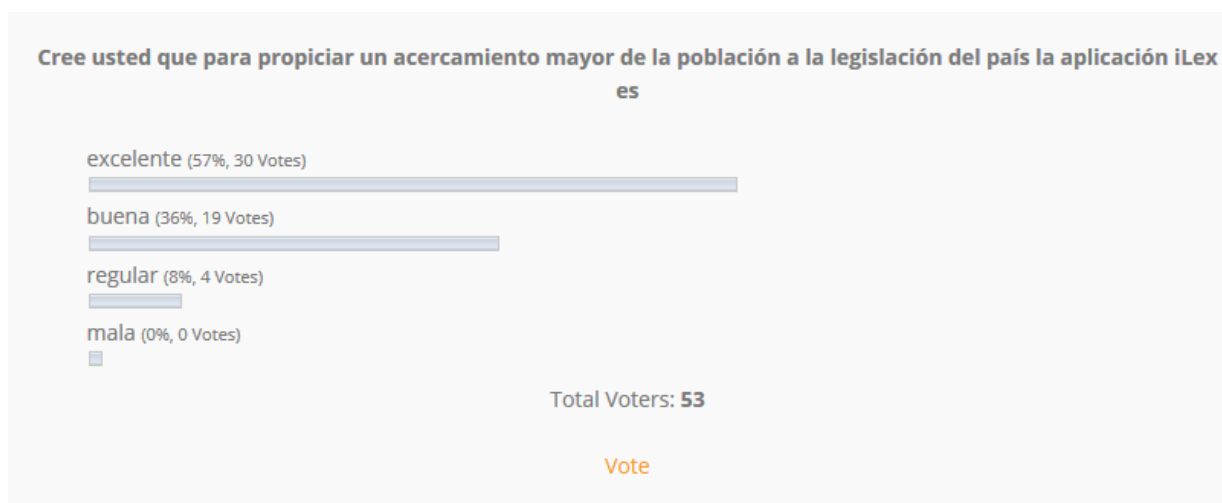


Figura 29: Resultado de encuesta en sitio www.jorgen.cubava.cu.

La aplicación iLex fue presentada además en la XI Conferencia de Derecho e Informática realizada del 16 al 19 de mayo del 2017 en la Habana, Cuba. Los participantes en el evento, en su mayoría juristas, tuvieron una opinión satisfactoria de la misma.

3.4 Comparación cualitativa entre las versiones 1.0 y 2.0 de iLex

Con el objetivo de realizar una comparación entre las dos versiones de la aplicación iLex, se usa el método cualitativo para resaltar las diferencias y novedades entre ambas versiones. La metodología cualitativa, como indica su propia denominación, tiene como objetivo la descripción de las cualidades de un fenómeno. Busca un concepto que pueda abarcar una parte de la realidad. No se trata de probar o de medir en qué grado una cierta cualidad se encuentra en un cierto acontecimiento dado, sino de descubrir tantas cualidades como sea posible (50).

Los **indicadores** de la comparación son los siguientes:

1. Soporte para varias Leyes.
2. Visualización de los títulos de las categorías (categorías se refiere a Libros, Títulos, Capítulos, Secciones).
3. Búsquedas Generales.

4. Edición de secciones de texto (Poner texto en negrita y cursiva, cambiar color de fuente).
5. Añadir favoritos.
6. Búsquedas locales en los artículos.
7. Cambiar tamaño de la fuente.
8. Cambiar color de la fuente.
9. Añadir notas.
10. Migas de Pan (Permite ver en qué parte del documento legal se encuentra el usuario).
11. Enviar artículo (compartir artículo a través de correo, SMS etc.).
12. Descargar nuevas Leyes.
13. Suprimir Leyes.
14. Continuar Lectura.

Como se evidencia en la tabla No. 6 el desarrollo de la nueva versión permitió aumentar las funcionalidades de iLex a fin de mejorar el producto. Permitir soporte para varias Leyes fue una de las causas que motivó el desarrollo de esta segunda versión (ver anexo 3) pues iLex 1.0 solo ofrecía el texto de la Constitución. De un fichero de texto almacenado internamente dentro de la aplicación, iLex paso a presentar una estructura genérica, capaz de adaptarse a cualquier Ley existente en la legislación del país. Además, el uso de una base de datos SQLite aumentó considerablemente la velocidad de consulta de la Leyes. Por otro lado, el usuario puede, inmediatamente después de que se publique una Ley nueva, conectarse a una red wifi, descargarla y desconectarse para trabajar offline (ver anexo 14). Esto último, junto con la opción de suprimir documentos (ver anexo 15) le dan al usuario la posibilidad de escoger que disposiciones legales desea tener es su dispositivo móvil.

En la primera versión, varios usuarios se quejaban de que no podían visualizar el título de los capítulos y se les hacía trabajoso navegar por la Ley, a este problema se le da solución en iLex 2.0 (ver anexo 4) haciendo más fácil la navegación por las Leyes. El único inconveniente de la nueva versión es que no permite la realización búsquedas generales, funcionalidad que si se encuentra en la primera versión (ver anexo 5).

En lo que se refiere a los indicadores 4 (ver anexo 6), 6 (ver anexo 8), 7 (ver anexo 9) y 8 (ver anexo 10) se encuentran presentes en ambas versiones pues representan los elementos reutilizados de la primera versión. En iLex 1.0 la lista de artículos añadidos como favoritos se encontraba en el menú lateral y solo mostraba el número del artículo. En la segunda versión la lista se muestra en una actividad aparte, donde además del número del artículo también se muestra el nombre de la Ley al que pertenece dicho artículo así como la ruta donde se encuentra ubicado (ver anexo 7).

Entre las nuevas funcionalidades la adición de notas permite al usuario añadir una nota descriptiva a un artículo determinado (ver anexo 11). También está la opción ¿Dónde estoy? (migas de pan) que permite saber en qué parte del documento de Ley se encuentra el usuario (ver anexo 12), así como la funcionalidad de enviar un artículo a través de las aplicaciones de mensajerías existentes en el dispositivo (ver anexo 13). Por último, iLex 2.0 permite continuar la lectura por donde el usuario la dejó la última vez, haciendo más fácil encontrar el último artículo que leyó el mismo (ver anexo 16).

Tabla 6: Comparación entre versiones.

Indicador	Versión 1.0	Versión 2.0
1	no	sí
2	no	sí
3	sí	no
4	sí	sí
5	sí	sí
6	sí	sí
7	sí	sí
8	sí	sí
9	no	sí
10	no	sí
11	no	sí
12	no	sí
13	no	sí
14	no	sí

3.5 Conclusiones del Capítulo

La realización de las pruebas de caja negra permitió encontrar no conformidades y verificar así el correcto funcionamiento de los requisitos del sistema descritos en la

solución. Con la realización de las pruebas de caja blanca se pudo analizar más detalladamente el código fuente de la aplicación y mejorar el rendimiento de la misma dando solución a los problemas encontrados. La publicación de la aplicación en sitios de internet le dio visibilidad a la misma y se pudo medir de cierta manera la satisfacción de los usuarios sobre el trabajo realizado.

Conclusiones

De manera general, durante el proceso de investigación e implementación de la aplicación para dispositivos móviles iLex, se pudo llegar a las siguientes conclusiones:

- Existe desconocimiento en la población cubana sobre la legislación del país, sobre todo en los jóvenes. Además, se evidencian deficiencias en las vías actuales de divulgación, siendo este uno de los factores que conllevan al desconocimiento de la población sobre los temas legislativos de la nación.
- La aplicación iLex en su versión 2.0 representa con respecto a su primera versión un avance en los temas de publicidad legislativa en el país. Además, se presenta como un producto innovador, que contribuirá a ampliar el canal oficial de publicidad normativa y mejorar el acercamiento de la población, en especial de los jóvenes a la legislación cubana.
- La realización de las pruebas de software permitió detectar errores en el producto para darle solución a los mismos y que la aplicación cuente con la calidad necesaria. Por otro lado, el nivel de aceptación de los usuarios respecto a la aplicación iLex fue satisfactorio en sentido general.

Recomendaciones

Se recomienda para investigaciones futuras:

- Incrementar el número de funcionalidades que presenta la aplicación.
- Aumentar la cantidad de disposiciones legales que brinda la aplicación hasta llegar a la totalidad de las que existen en el país.
- Implementar un algoritmo de búsqueda general y por Leyes con el objetivo de facilitar la búsqueda de contenidos en la aplicación.

Referencias Bibliográficas

1. Jóvenes de ley - Cuba - Juventud Rebelde - Diario de la juventud cubana. [online]. [Accessed 14 December 2016]. Available from: <http://www.juventudrebelde.cu/cuba/2015-04-05/jovenes-de-ley/>
2. ETECSA tiene la palabra (+ Infografías y Video) | Cubadebate. [online]. [Accessed 11 May 2017]. Available from: <http://www.cubadebate.cu/especiales/2017/05/11/etecsa-tiene-la-palabra-infografia-y-video/>
3. Real Academia Española. Diccionario Usual. [online]. [Accessed 10 June 2017]. Available from: <http://dle.rae.es/srv/search?m=30&w=publicidad>
4. Publicidad de las normas. [online]. [Accessed 8 December 2016]. Available from: <http://www.encyclopedia-juridica.biz14.com/d/publicidad-de-las-normas/publicidad-de-las-normas.htm>
5. Real Academia Española. Diccionario Usual. [online]. [Accessed 10 June 2017]. Available from: <http://dle.rae.es/srv/search?m=30&w=difundir>
6. QUE SON DISPOSITIVOS MOVILES | admsaludv. [online]. [Accessed 8 December 2016]. Available from: <https://admsaludv.wordpress.com/59-2/>
7. Normas Políticas de España – contenidos jurídicos siempre actualizados apk. [online]. [Accessed 23 November 2016]. Available from: <http://normas-politicas-de-espaa-contenidos-juridicos-siempre-actualiza.freeapkdownload.org/>
8. vLex América Latina. [online]. [Accessed 23 November 2016]. Available from: <http://latam.vlex.com/>
9. Historia de vLex. [online]. [Accessed 23 November 2016]. Available from: <http://latam.vlex.com/nosotros/historia>
10. Producto vLex América Latina. [online]. [Accessed 23 November 2016]. Available from: <http://latam.vlex.com/productos/profesionales/latam-pa-vlex-america-latina>
11. Por qué vLex. [online]. [Accessed 23 November 2016]. Available from: <http://latam.vlex.com/por-que-vlex>
12. Mi Abogado - Aplicaciones de Android en Google Play. [online]. [Accessed 23 November 2016]. Available from: <https://play.google.com/store/apps/details?id=com.arag.miabogado&hl=es>
13. Gaceta Oficial, de la Republica de Cuba. Publicación Digital. [online]. [Accessed 23 November 2016]. Available from: https://www.gacetaoficial.gob.cu/html/que_es.html
14. X Cuba – Aplicación móvil para Android | Parlamento Cubano. [online]. [Accessed 26 April 2017]. Available from: <http://www.parlamentocubano.cu/index.php/x-cuba-aplicacion-movil-para-android/>
15. Real Academia Española. Diccionario Usual. [online]. [Accessed 10 June 2017]. Available from: <http://dle.rae.es/srv/search?m=30&w=metodolog%C3%ADa>

16. Modelos Y Metodologías Para El Desarrollo De Software. [online]. [Accessed 6 June 2017]. Available from: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>
17. FIGUEROA, Roberth G, SOLÍS, CAMILO J and CABRERA, ARMANDO A. Metodologías tradicionales vs. Metodologías ágiles. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. [online]. Available from: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>.
18. RAHIMIAN, V and RAMSIN, R. Designing and agile methodology for mobile software development: a hybrid ethod engineering approach. Second International Conference on Research Challenges in Information Science. [online]. 6 June 2008. Available from: [http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4632123&number%3D4620134%26sortType%3Das_c_p_Sequence%26fiter%3DAND%28p_IS_Number%3A4632084%29%26pageNu mber%3D2](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4632123&number%3D4620134%26sortType%3Das_c_p_Sequence%26fiter%3DAND%28p_IS_Number%3A4632084%29%26pageNumber%3D2).
19. JOSE GERMÁN NÚÑEZ MORI. *Tesis de Maestria. USABILIDAD EN METODOLOGÍAS ÁGILES* [online]. 2010. [Accessed 7 June 2017]. Available from: https://www.fi.upm.es/catedra-ibmrational/sites/www.fi.upm.es.catedra-ibmrational/files/Tesis_MarcoAgilTrabajo.pdf
20. GRUPO ISSI INGENIERIA DE SOFTWARE Y SISTEMAS INFORMÁTICOS. Metodologías ágiles en el desarrollo de software. [online]. 12 November 2013. [Accessed 7 June 2017]. Available from: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
21. LUIS MIGUEL EGAS and JUAN XAVIER JÁTIVA. Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. [online]. 2010. [Accessed 7 June 2017]. Available from: <https://repositorio.espe.edu.ec/jspui/bitstream/21000/8771/1/AC-ESPEL-SOF-0004.pdf>
22. Scrum: ¿Qué es Scrum? | Aplicando Scrum. [online]. [Accessed 24 February 2017]. Available from: <http://www.aplicandoscrum.com/scrum/>
23. Palacio Juan. Gestión de proyectos Scrum Manager. [online]. [Accessed 22 November 2014]. Available from: http://www.scrummanager.net/files/sm_proyecto.pdf
24. Artefactos - Scrum Manager BoK. [online]. [Accessed 14 December 2016]. Available from: <http://www.scrummanager.net/bok/index.php?title=Artefactos>
25. Visual Paradigm para UML. [online]. [Accessed 22 November 2016]. Available from: <http://www.software.com.ar/p/visual-paradigm-para-uml>
26. Definicion de sistema Operativo « TECNOLOGÍA E INFORMÁTICA. [online]. [Accessed 9 June 2017]. Available from: <https://solvasquez.wordpress.com/2011/01/24/definicion-de-sistema-operativo/>
27. Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016. [online]. [Accessed 9 June 2017]. Available from: <https://www.gartner.com/newsroom/id/3415117>

28. Definición de IDE - Diccionario de Términos técnicos de Internet - Glosario. [online]. [Accessed 10 June 2017]. Available from: <http://tecnologia.glosario.net/terminos-tecnicos-internet/ide-860.html>
29. Conoce Android Studio | Android Studio. [online]. [Accessed 22 November 2016]. Available from: <https://developer.android.com/studio/intro/index.html>
30. Robert W. Sebesta. *CONCEPTS OF PROGRAMMING LANGUAGES. 10.* New Jersey.
31. Java. *Definición MX* [online]. [Accessed 23 November 2016]. Available from: <http://definicion.mx/java/Definición de Java>
32. Definición de SDK. [online]. [Accessed 13 April 2017]. Available from: <http://www.alegsa.com.ar/Dic/sdk.php>
33. JSON. [online]. [Accessed 22 November 2016]. Available from: <http://www.json.org/json-es.html>
34. About SQLite. [online]. [Accessed 8 June 2017]. Available from: <https://www.sqlite.org/about.html>
35. android.database.sqlite | Android Developers. [online]. [Accessed 11 June 2017]. Available from: <https://developer.android.com/reference/android/database/sqlite/package-summary.html>
36. DB Browser for SQLite. [online]. [Accessed 8 June 2017]. Available from: <http://sqlitebrowser.org/>
37. Introducción a las pruebas del software - Monografias.com. [online]. [Accessed 12 June 2017]. Available from: <http://www.monografias.com/trabajos107/introduccion-pruebas-del-software/introduccion-pruebas-del-software.shtml>
38. ROGER R. PRESSMAN. Ingeniería del Software. Un Enfoque Practico - Ingeniería.de.software.enfoque.practico.7ed.Pressman.PDF. [online]. [Accessed 1 June 2017]. Available from: <http://s3.amazonaws.com/academia.edu.documents/45525376/Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1496273138&Signature=GJvs9eIIJ0pb9hnGQIz%2F6hlyeac%3D&response-content-disposition=inline%3B%20filename%3DIngenieria.de.software.enfoque.practico..pdf>
39. DR. EDUARDO A. RODRIGUÉZ TELLO. Importancia de las pruebas de software. [online]. 30 March 2011. [Accessed 13 April 2017]. Available from: <http://www.tamps.cinvestav.mx/~ertello/swe/swTestingTecZacatecas.pdf>
40. KEN SCHWABER and JEFF SUTHERLAND. *La Guía de Scrum.* 2016. La Guía Definitiva de Scrum: Las Reglas del Juego.
41. ALEXANDER MENZINSKY, GERTRUDIS LÓPEZ and JUAN PALACIO. *Scrum manager* [online]. [no date]. Available from: [http://www.scrummanager.net/bok/Derechos regitrados en Safe Creative n° de registro: 1607208414838](http://www.scrummanager.net/bok/Derechos%20regitrados%20en%20Safe%20Creative%20n%C3%B3%20de%20registro%201607208414838)

42. JENNY GERMANIA CARRERA GUANOLUISA. *ANÁLISIS COMPARATIVO DE LA PRODUCTIVIDAD ENTRE LOS PATRONES DE DISEÑO MODELO VISTA CONTROLADOR (MVC) Y MODELO VISTA PRESENTADOR (MVP) APLICADO AL DESARROLLO DEL SISTEMA NÓMINA DE EMPLEADOS Y ROL DE PAGOS DE LA "DISTRIBUIDORA SORIA C.A* [online]. Riobamba – Ecuador, 2014. [Accessed 6 June 2017]. Available from: <http://dspace.esPOCH.edu.ec/bitstream/123456789/3583/1/18T00569.pdf>
43. INGENIERÍA DEL SOFTWARE II. Patrones de Diseño. [online]. 2011. [Accessed 3 May 2017]. Available from: <http://www.cartagena99.com/recursos/alumnos/apuntes/Patrones%20de%20Diseño.pdf>
44. CARLOS PLATERO. Apuntes de Informática Industrial. Patrones de diseño GoF. [online]. [Accessed 6 June 2017]. Available from: http://www.elai.upm.es/moodle/pluginfile.php/2716/mod_resource/content/6/cap6_Dise%C3%B1o1.6b.pdf
45. Revisiones de código y estándares de codificación. [online]. [Accessed 3 May 2017]. Available from: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx)
46. MANUEL ARIAS CALLEJA. *Estándares de codificación* [online]. 2012. [Accessed 3 May 2017]. Available from: <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>
47. IAN SOMMERVILLE. *Ingeniería del Software 7ma. Ed.* [online]. [no date]. [Accessed 8 June 2017]. Available from: <http://zeus.inf.ucv.cl/~bcrawford/EnfoquesDeDesarrolloDeSwYLenguajesDeModelado/Ingeniería%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf>
48. Continuous Code Quality | SonarQube. [online]. [Accessed 9 May 2017]. Available from: <https://www.sonarqube.org/>
49. AEC - No Conformidad. [online]. [Accessed 27 April 2017]. Available from: <https://www.aec.es/web/guest/centro-conocimiento/no-conformidad>
50. Investigación cualitativa y cuantitativa - Diferencias y limitaciones - Monografias.com. [online]. [Accessed 7 June 2017]. Available from: <http://www.monografias.com/trabajos38/investigacion-cualitativa/investigacion-cualitativa.shtml>

Anexo 1

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Encuesta para estudiar el nivel de conocimiento sobre la Constitución de la República de Cuba

Edad: _____ Sexo: _____

Ubicación Laboral:

Trabajador: _____ Estudiantes: _____ Profesor: _____

Especialista: _____

Nivel de Escolaridad:

Primaria: _____ Secundaria: _____ Preuniversitario: _____

Universitario: _____ T.M: _____

¿Ha estudiado la Constitución de la República de Cuba en algún curso de sus estudios regulares?

Sí: _____ No: _____ ¿EnCuál? : _____

¿Ha leído la Constitución de la República de Cuba?

Todo: _____ En Parte: _____ Nada: _____

¿Si la leyó, la entendió?

Todo: _____ En Parte: _____ Nada: _____

¿Sería bueno que le hubieran explicado?

Sí: _____ No: _____

¿Sabes dónde podrías adquirir la Constitución de la República de Cuba?

Sí: _____ No: _____

Anexo 2

Artefacto	Versión	Estado final	Cantidad iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App iLex Publicidad normativa en su móvil	2.0	0	2	Evaluación dinámica Pruebas de Funcionalidad	11/05/2017

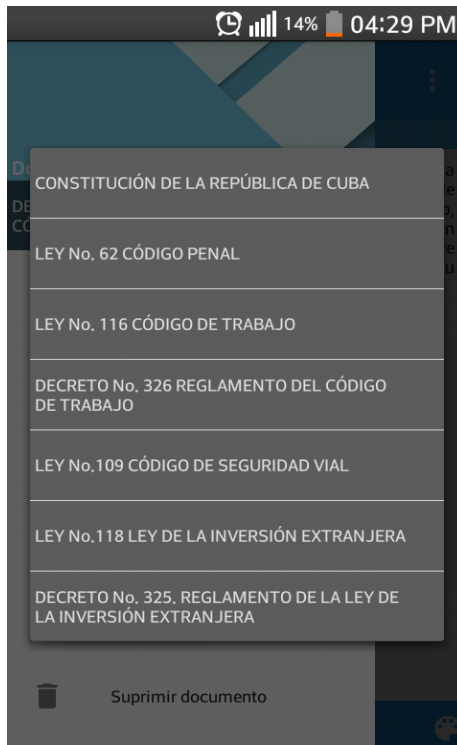
MSc. Yordania García Leiva
 Asesor de Calidad CEGEL

Ing. Elizabeth Enríquez Gusado
 Responsable de la liberación

Felix Antonio Marrero Piñón
 Autor

1

Anexo 3



v 2.0

Anexo 4

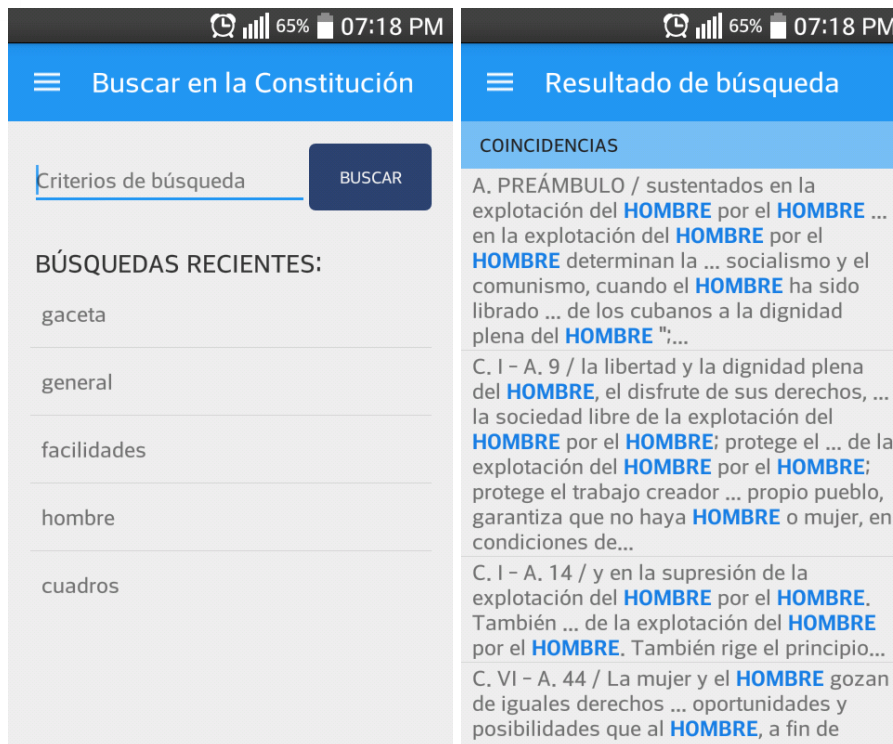


v 1.0



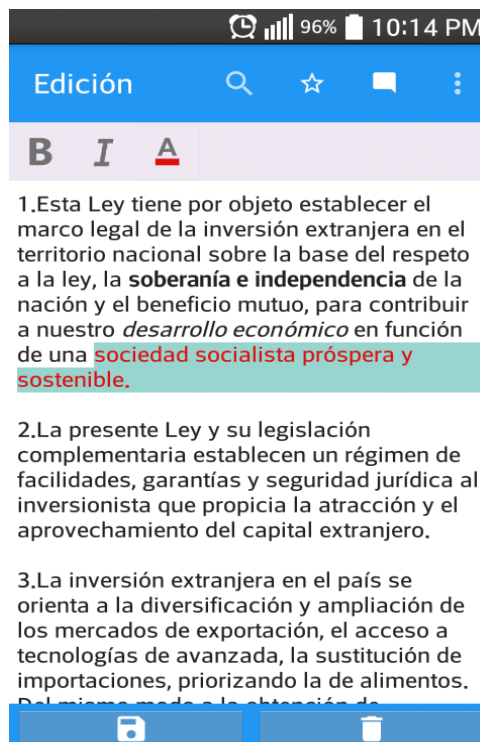
v 2.0

Anexo 5



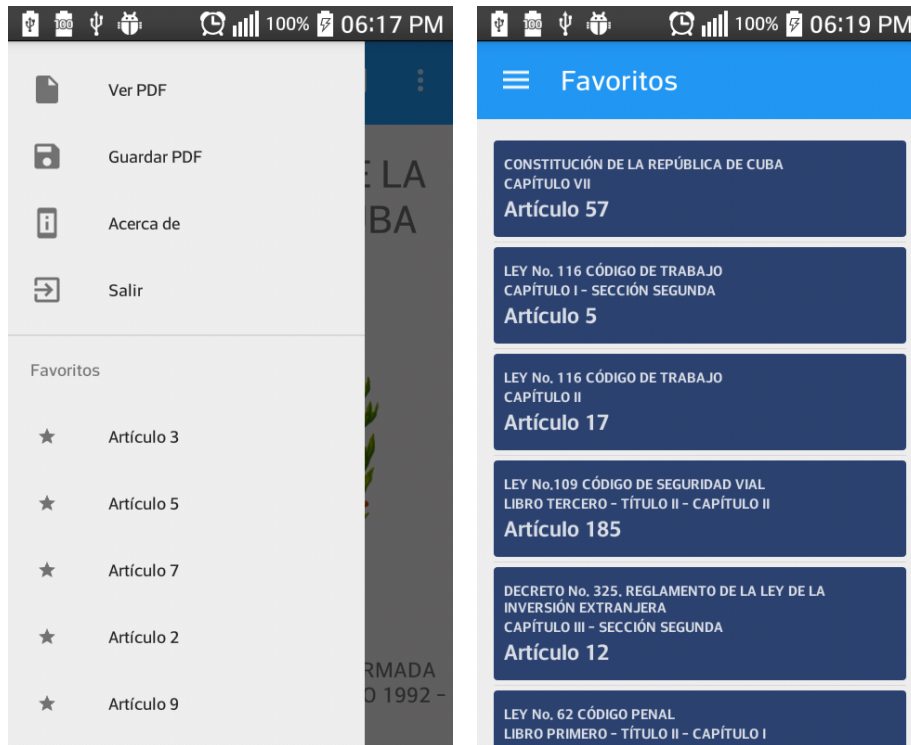
v 1.0

Anexo 6



Ambas versiones

Anexo 7



v 1.0

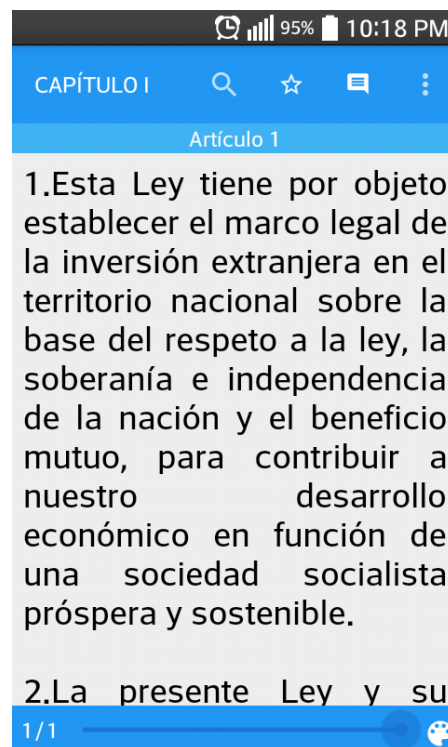
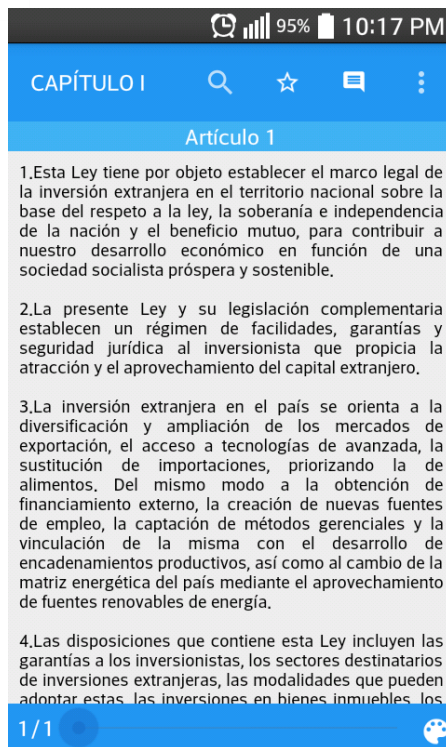
v 2.0

Anexo 8



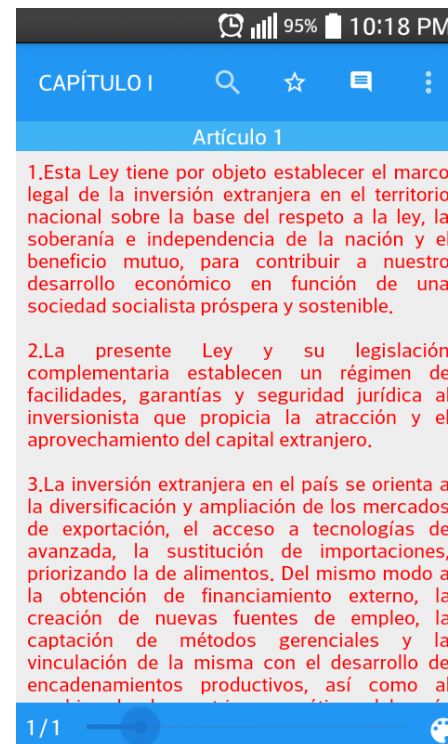
Ambas versiones

Anexo 9



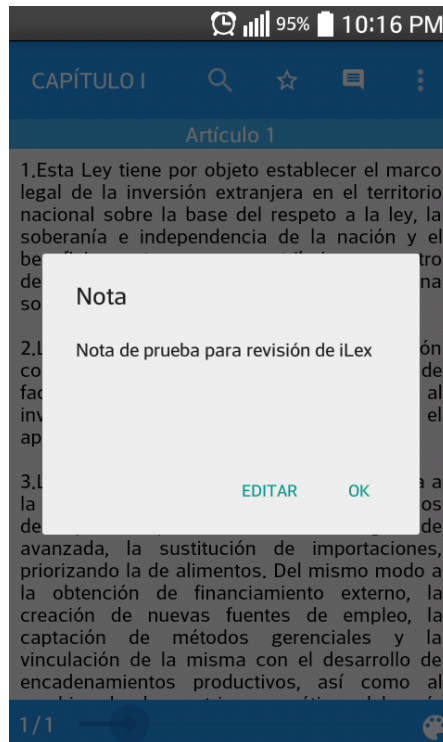
Ambas versiones

Anexo 10



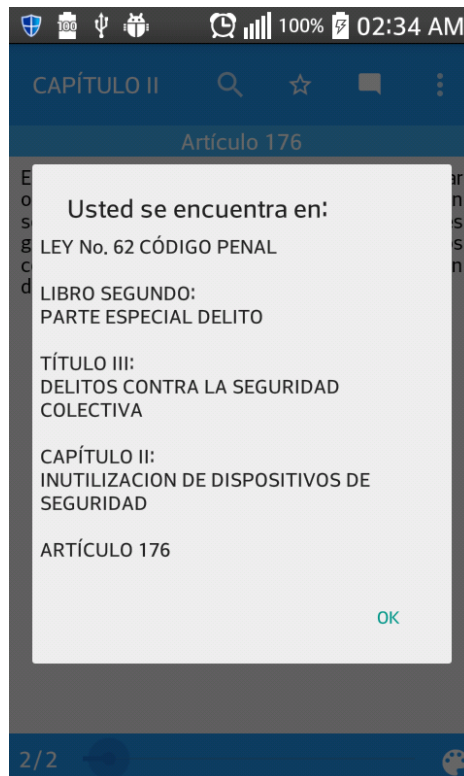
Ambas versiones

Anexo 11



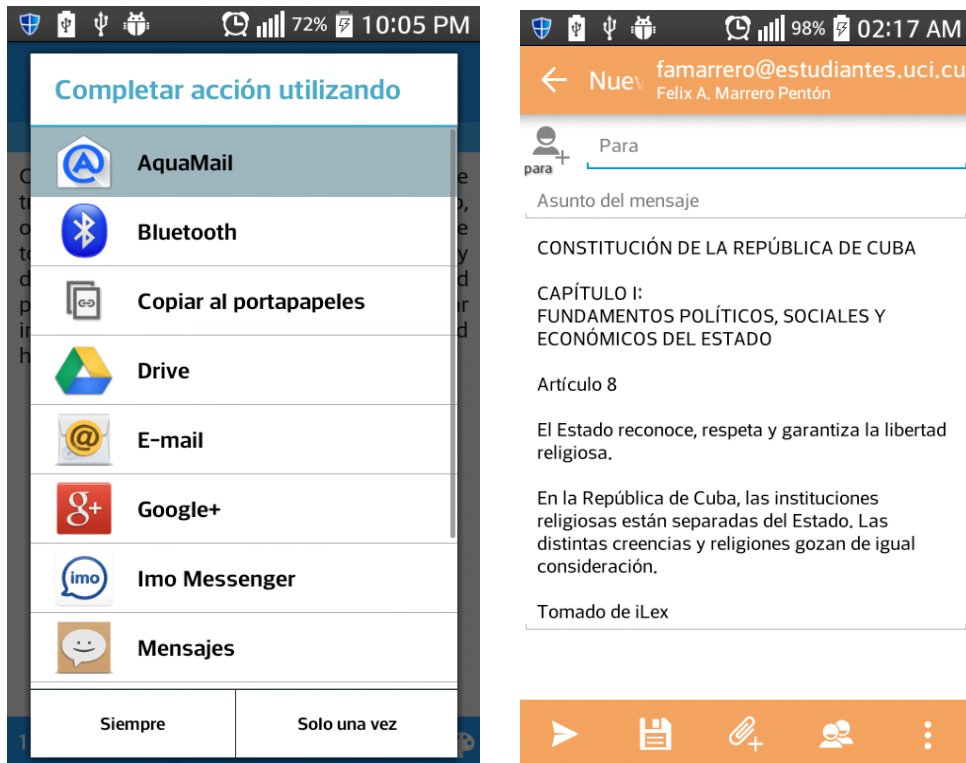
v 2.0

Anexo 12



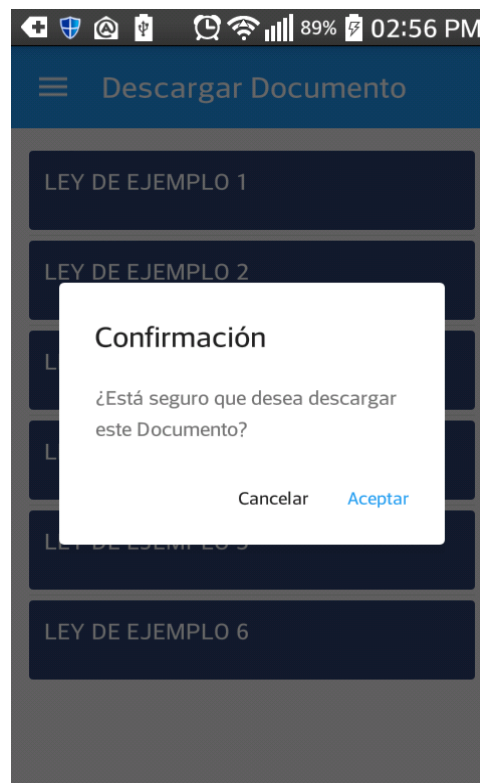
v 2.0

Anexo 13



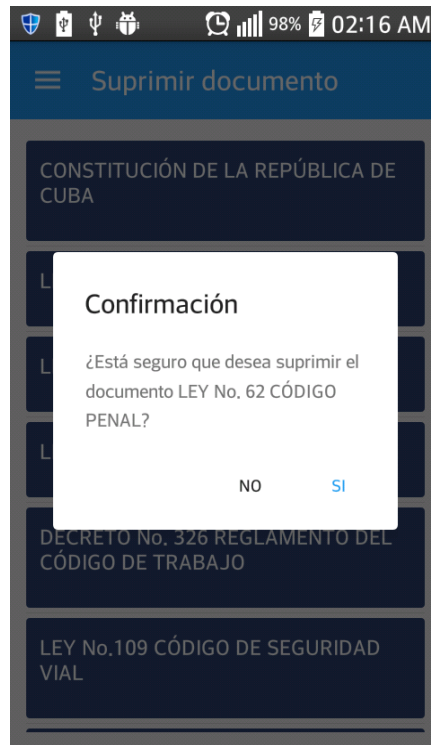
v 2.0

Anexo 14



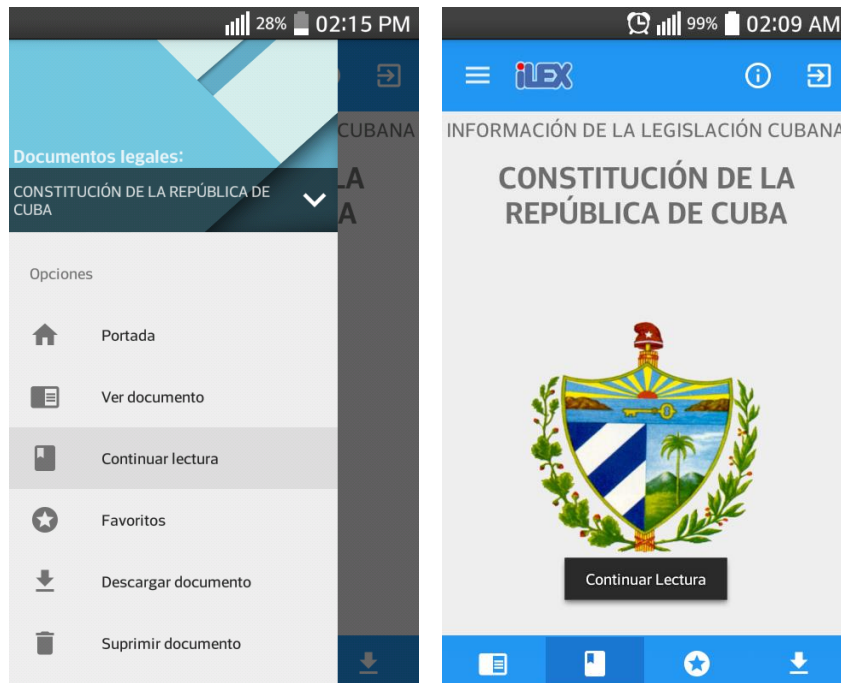
v 2.0

Anexo 15



v 2.0

Anexo 16



v 2.0