UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Aplicación de Android para la gestión de un cuaderno digital para los estudiantes de la Universidad Católica Andrés Bello

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: Diecinueve (19) puntos.

J U R A D O E X A M I N A D O R

Firma: Fanny Hernández	Firma:	Firma: Gloria Tarrío V.
Nombre: Fanny Hernández	Nombre: Henry Martínez	Nombre: Gloria Tarrío V.

REALIZADO POR	**Díaz Pérez, Pedro Emilio**
TUTOR EMPRESARIAL	**Ing. Benal Vera, Roquefelix**
TUTOR ACADÉMICO	**Ing. Tarrío Villaverde, Gloria**
FECHA	**Julio 2017**

UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Aplicación de Android para la gestión de un cuaderno digital para los estudiantes de la Universidad Católica Andrés Bello.

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR	Díaz Pérez, Pedro Emilio
TUTOR EMPRESARIAL	Ing. Benal Vera, Roquefelix
TUTOR ACADÉMICO	Ing. Tarrío Villaverde, Gloria
FECHA	Julio 2017

ÍNDICE GENERAL

ÍNDICE DE FIGURAS VII

ÍNDICE DE TABLAS VIII

AGRADECIMIENTOS IX

SINOPSIS X

CAPÍTULO I – NECESIDADES DE LA EMPRESA 1

I.1 Necesidades de la empresa 1

I.2 Solución propuesta 2

I.3 Objetivo General 3

I.4 Objetivo Específicos 3

I.4.1 Aporte Tecnológico 3

I.4.2 Aporte Funcional 4

I.5 Alcance y limitaciones 4

I.5.1 Alcance 4

I.5.1.1 Aporte Tecnológico 12

I.5.1.1 Aporte Funcional 13

I.5.2 Limitaciones 14

I.6 Justificación 15

CAPÍTULO II – MARCO TEÓRICO **16**

II.1 Universidad Católica Andrés Bello 16

II.2 Aplicaciones móviles 17

II.2.1 Tipos de aplicaciones móviles 17

II.2.1.1 Aplicaciones nativas 18

II.2.1.2 Aplicaciones web 19

II.2.1.3 Aplicaciones híbridas 20

II.3 Android 21

II.3.1 Características de Android 22

II.3.2 Arquitectura de Android 24

II.3.3 Fundamentos de una aplicación Android 25

II.3.3.1 Componentes de una aplicación Android 27

II.3.3.2 El archivo de manifiesto de una aplicación Android 28

II.3.3.3 Recursos de una aplicación Android 30

II.3.4 Android Studio 31

II.4 UCABdroide 32

II.5 Java 32

II.6 Patrones de diseño 32

II.6.1 Modelo-Vista-Presentador (MVP) 33

II.6.2 Repositorio 34

II.6.3 Inyección de dependencias 34

CAPÍTULO III – MARCO METODOLÓGICO 35

III.1 Scrum 35

III.1.1 Características de Scrum 36

III.1.2 Proceso de Scrum 36

III.1.3 Roles de Scrum 38

III.1.4 Artefactos de Scrum 38

III.2 Justificación de la metodología 39

III.3 Planificación de la implementación de la solución 41

CAPÍTULO IV – DESARROLLO 45

IV.1 Sprint 0 45

IV.1.1 Inicio 45

IV.1.2 Product Backlog 46

IV.1.3 Diseño e implementación de almacén de datos 48

IV.1.3.1 Creación del Modelo Entidad-Relación 48

IV.1.3.1 Creación de la base de datos 48

IV.1.4 Desarrollo de librería de respaldo y sincronización con la nube 49

IV.2 Sprint 1 50

IV.2.1 Rediseño del proceso de apuntes con dispositivos móviles 50

IV.2.2 Desarrollo de módulo de gestión de notas y apuntes 51

IV.3 Sprint 2 67

IV.3.1 Diseño de módulo de galería de archivos 67

IV.3.2 Desarrollo de control de visualización de archivos 68

IV.3.3 Desarrollo de control de asociación de archivos 70

IV.3.4 Desarrollo de widget para captura de foto o nota de voz 71

IV.3.5 Diseño e implementación de estructura de etiquetas 72

IV.4 Sprint 3 73

IV.4.1 Diseño de estructura de archivo de cuadernos 73

IV.4.2 Desarrollo de control de exportación e importación de cuadernos 74

IV.4.3 Desarrollo de control de sincronización y respaldo de cuadernos 75

IV.4.4 Desarrollo de control de exportación de cuaderno como archivo PDF 78

IV.5 Sprint 4 78

IV.5.1 Integrar UCABdroide las aplicaciones instaladas en el dispositivo 78

IV.5.2 Desarrollo de control para compartir desde la galería del cuaderno con estudiantes de un curso 79

CAPÍTULO V – RESULTADOS **80**

V.1 Resultados según Objetivos 80

V.2 Aporte Tecnológico 82

V.3 Aporte Funcional 82

CAPÍTULO VI – CONCLUSIONES Y RECOMENDACIONES **84**

VI.1 Conclusiones 84

VI.2 Recomendaciones 86

BIBLIOGRAFÍA **87**

Referencias electrónicas 87

Referencias de libros 88

APÉNDICE **89**

Apéndice A: Figuras 89

A1. Módulo de gestión de notas y apuntes 89

A2. Módulo de galería de archivos 91

A3. Módulo de respaldo y sincronización 94

A4. Módulo para compartir contenido 96

ÍNDICE DE FIGURAS

Figura 1. Pila de software de Android 26

Figura 2. Archivo de manifiesto “AndroidManifest.xml” 29

Figura 3. Modelo Vista Presentador 33

Figura 4. Metodología Scrum 39

Figura 5. Diagrama entidad-relación 49

Figura 6. Comparación de flujo de toma de apuntes con móviles 52

Figura 7. Paneles de edición de cuaderno digital 56

Figura 8. Primera implementación de trazo a mano alzada 59

Figura 9. Segunda implementación de trazo a mano alzada 59

Figura 10. Curva de Bézier 60

Figura 11. Implementación final de trazo a mano alzada 61

Figura 12. Barra de herramientas de cuaderno 63

Figura 13. Control de pincel 64

Figura 14. Control de color 64

Figura 15. Barra de herramientas de cuaderno colapsada 66

Figura 16. Sistema de archivos UCABdroide 68

Figura 17. Archivo ".ucab" 74

ÍNDICE DE TABLAS

Tabla 1. Aplicaciones nativas 19

Tabla 2. Sprint Backlog 44

Tabla 3. Product Backlog 47

AGRADECIMIENTOS

El principal agradecimiento, el que me ha guiado en todo momento y no me ha abandonado, Dios.

A mi padre y madre, quienes siempre han estado conmigo, me han forjado como persona y se han esforzado en brindarme educación. Muchos de mis logros, incluyendo este, se lo debo a ustedes.

A mis hermanos y familia, en los que también han contribuido en mi crecimiento como persona y profesional, enseñándome cosas invaluables.

A la universidad y mis profesores, por las enseñanzas, apoyo y oportunidades que me han brindado a lo largo de la carrera.

A mis amigos y compañeros de la carrera, quienes siempre estuvieron presente cuando necesitaba apoyo, por sus valiosas ayudas y aportes a lo largo de la carrera, y por todos los valiosos momentos vividos.

A todos ellos, gracias.

SINOPSIS

El presente Trabajo Instrumental de Grado titulado "Aplicación de Android para la gestión de un cuaderno digital para los estudiantes de la Universidad Católica Andrés Bello", se desarrolló bajo los lineamientos de la Dirección de Tecnología de Información (DTI) y tiene como fin optimizar el proceso mediante el cual los estudiantes de la universidad llevan a cabo los apuntes de clase con dispositivos móviles.

Desde hace un tiempo los estudiantes de la Universidad Católica Andrés Bello han tenido la necesidad de guardar en formato digital las anotaciones que se hacen en el pizarrón de clases, es por ello que cada día es más común ver estudiantes tomando fotos con sus teléfonos inteligentes o tabletas al pizarrón de clases y van dejando a un lado el cuaderno. En este sentido se desarrolla una aplicación que permita a los estudiantes hacer una sustitución, quizás no definitiva, pero sí en gran medida del uso del cuaderno físico por un cuaderno digital el cual le permitirá tomar las notas con menor esfuerzo y mantener el material digital organizado.

Se utilizó SCRUM, ágil y flexible, con desarrollo iterativo e incremental, el uso de bloques de trabajo (sprint) fue fundamental para concretar los objetivos planificados.

Finalmente, el entorno de desarrollo integrado de la aplicación fue Android Studio, sobre el lenguaje de programación Java, usando una base de datos SQLite para el almacenamiento de los datos.

CAPÍTULO I – NECESIDADES DE LA EMPRESA

I.1 Necesidades de la empresa

La tecnología es el conjunto de conocimientos y técnicas que, aplicados de forma lógica y ordenada, permiten al ser humano modificar su entorno material o virtual para satisfacer sus necesidades. Esto se ha notado con el aumento del uso de las herramientas tecnológicas en los salones de clases Universidad Católica Andrés Bello por parte de los alumnos.

Una de las conductas más notorias es la toma de fotos al pizarrón de clases o grabar la clase como nota de voz con su teléfono celular mientras el profesor va realizando sus anotaciones y dictando la clase, esto debido a la necesidad de guardar las notas de manera digital para luego apoyarse en las mismas a la hora de estudiar. Además, esta nueva práctica, permite que los estudiantes compartan el material captado de la clase entre ellos, pudiendo visualizar contenido de otras secciones o inclusive, de semestres anteriores.

Sin embargo, además de estos beneficios, esta práctica trae consigo algunas complicaciones como lo es la búsqueda y clasificación de cada fotografía y/o notas de audio y la sincronización con otros dispositivos de una manera simple.

La Universidad Católica Andrés Bello desea poder brindar a sus estudiantes una herramienta digital que le permita la organización, respaldo e

intercambio entre alumnos del material capturado en clases para poder facilitar el proceso de aprendizaje.

Adicionalmente la universidad busca poder ofrecer un sistema de cuaderno digital. Este sistema permitirá al estudiante mediante controles como de texto o trazos a mano alzada, poder realizar apuntes de la clase con la habilidad de asociar material gráfico y notas de audio.

I.2 Solución propuesta

Se propone desarrollar un módulo adicional e integrarlo con UCABdroide que tenga como objetivo principal poder bridarle al estudiante una herramienta digital para la toma de apuntes en clase.

Este módulo debe ser capaz de crear notas para que el estudiante tenga la opción de llevar sus apuntes de forma digital. La captación de estas notas debe realizarse con el teclado del dispositivo o a mano alzada, como prefiera el estudiante. Además, estas notas tendrían la opción de asociar contenido multimedia, donde el estudiante puede complementar las notas con imágenes, audio, documentos pdf, entre otros.

Se propone adicionalmente que el módulo le permita al estudiante organizar y visualizar el contenido digital capturado, bien sea por materia, tipo o fecha de creación mediante la implementación de una galería. Además, la galería podrá exportar e importar contenido con otras aplicaciones.

Como complemento a esta propuesta, se desea poder brindarle al estudiante un medio para la sincronización y respaldo de las notas realizadas, donde el estudiante, usando su cuenta personal, tenga la opción de realizar estas operaciones.

I.3 Objetivo General

Desarrollar una aplicación de Android para la gestión de un cuaderno digital para los estudiantes de la Universidad Católica Andrés Bello.

I.4 Objetivo Específicos

1. Desarrollar módulo de gestión de notas y apuntes.
2. Desarrollar módulo de galería de archivos.
3. Desarrollar módulo de respaldo y sincronización.
4. Desarrollar módulo para compartir contenido.
5. Desarrollar una librería que se encargue de facilitar la sincronización y respaldo con servicios de almacenamiento en la nube.

I.4.1 Aporte Tecnológico

6. Integrar con otras aplicaciones instaladas en el dispositivo para compartir contenido con las mismas.

I.4.2 Aporte Funcional

7. Rediseñar el proceso mediante el cual se llevan a cabo los apuntes de clase con dispositivos móviles.

I.5 Alcance y limitaciones

I.5.1 Alcance

1. Desarrollar módulo de gestión de notas y apuntes.

    a. Desarrollar un módulo de gestión de notas y apuntes de clases.

        Este módulo se encargará de brindar al estudiante una serie de controles para la creación y manipulación de notas que se deban realizar en un cuaderno, estos controles son:

        - Control de trazo a mano alzada: Con este control el estudiante podrá realizar trazos sobre el cuaderno usando su dedo o un lápiz para tabletas y teléfonos inteligentes. Este control contará con las siguientes opciones de personalizado:

            - Color.
            - Grosor.
            - Resaltado (opacidad).

- Control de Texto: Con este control el estudiante podrá insertar texto con el teclado del dispositivo en una página del cuaderno. Este control contará con las siguientes opciones de personalizado:
    - Fuente.
    - Aumentar tamaño de fuente.
    - Disminuir tamaño de fuente.
    - Color de la fuente.
    - Colocar el texto en negrita.
    - Colocar el texto en cursiva.
    - Colocar un subrayado al texto.
    - Cambiar el tamaño del cuadro de texto.
- Control de flecha: Con este control el estudiante podrá insertar una figura de flecha. Este control contará con las siguientes opciones de personalizado:
    - Color.
    - Rotar flecha.
    - Aumentar tamaño de la flecha.

- Disminuir tamaño de la flecha.

- Control de círculo: Con este control el estudiante podrá insertar una figura de círculo. Este control contará con las siguientes opciones de personalizado:

  - Color.

  - Grosor del borde.

  - Cambiar el tamaño del círculo.

- Control de cuadrado: Con este control el estudiante podrá insertar una figura de cuadrado. Este control contará con las siguientes opciones de personalizado:

  - Color.

  - Cambiar el tamaño del cuadrado.

  - Grosor del borde.

- Control de imagen: Con este control el estudiante podrá realizar fotos desde la cámara o importar imágenes de la galería del dispositivo e insertarlas en el cuaderno o asociarlas a una materia. El control contará con las siguientes características:

- Permitir tomar una fotografía y agregarla como fondo de una página o como una imagen dentro de una página.
- Permitir elegir una fotografía bien sea desde la galería de fotos del dispositivo o desde la galería de fotos de la materia y colocarla como fondo de página o como una imagen dentro de una página.
- Permitir cambiar el tamaño de la imagen o rotarlo.

- Control de nota de audio: Con este control el estudiante podrá grabar una nota de audio y asociarla a una clase o a una materia.

2. Desarrollar módulo de galería de archivos.

   a. Desarrollar una galería para visualizar los archivos asociados a una materia.

      Se creará una galería donde el estudiante podrá visualizar todos los documentos (imágenes, archivos y audio) asociados a la materia o ver dichos documentos por clase. El estudiante podrá seleccionar uno o más archivos para poder eliminarlos o desmarcar toda la selección hecha. También podrá mover un archivo de una clase a otra después de seleccionarlo.

b. Desarrollar un control para asociar archivos y contenido multimedia (imágenes y audio) almacenados en el dispositivo a una materia.

Adicionalmente se desarrollará e implementará un control que permita asociar archivos y contenidos multimedia almacenados en el dispositivo a una materia o clase en particular. Para lograr este control, se implementarán las siguientes funcionalidades:

- Por materia: A través de esta funcionalidad el estudiante desde la galería de la materia podrá asociar archivos, fotos tomadas desde la cámara o notas de voz del dispositivo a una materia.

- Por clase: A través de esta funcionalidad el estudiante desde la galería de la materia podrá asociar archivos, fotos tomadas desde la cámara o notas de voz del dispositivo a una clase en particular.

c. Diseñar e implementar una estructura de etiquetas y palabras claves asociadas a archivos y clases que permita luego, mediante un control, realizar búsquedas.

El módulo contará con una estructura de etiquetas y palabras claves donde el estudiante podrá asociar dichas etiquetas a archivos y clases que se encuentren en la aplicación para luego,

mediante un motor de búsqueda y una interfaz donde se podrá visualizar todos los contenidos asociados a dichas palabras claves.

d. Desarrollar un widget de escritorio que permita tomar una foto o una nota de voz y asociarlo a una materia en la que esté inscrito el estudiante.

Como complemento de este módulo, se desarrollará e implementará un widget de escritorio que permita al estudiante tomar una foto o una nota de voz desde de la cámara o grabadora de UCABdroide y asociarla a una materia sin necesidad de tener que entrar a la aplicación como tal para poder asociar el archivo.

3. Desarrollar módulo de respaldo y sincronización.

a. Diseñar e implementar un almacén de datos local, que permita almacenar los cuadernos y todos los objetos relacionados con el mismo.

Para el desarrollo e implementación de este objetivo es necesario diseñar y construir un diagrama entidad-relación que especifique la estructura de la base de datos relacional en el dispositivo que permita almacenar los cuadernos y todos los objetos relacionados con el mismo.

b. Desarrollar un control que permita exportar los cuadernos y todos los objetos relacionados a un archivo local en el dispositivo.

Se desarrollará un control que permita al estudiante exportar los cuadernos y objetos relacionados a un archivo local en el dispositivo. Se debe diseñar una estructura que permita el logro de este sub-módulo.

c. Desarrollar un control que permita importar los cuadernos del estudiante desde un archivo local en el dispositivo.

A través de esta funcionalidad, el estudiante podrá importar a UCABdroide los cuadernos y objetos relacionados dentro de un archivo que cumpla con la estructura diseñada previamente en el control de exportación.

d. Desarrollar un control que permita la sincronización y respaldo de cuadernos, notas y apuntes.

Utilizando la librería desarrollada para el respaldo y sincronización con servicios de almacenamiento en la nube y específicamente Google Drive, brindar al estudiante la opción de guardar y recuperar sus cuadernos y apuntes realizados. Además, el estudiante podrá elegir el modo de sincronización: Periódica o manual.

e. Desarrollar un control que permita exportar un cuaderno o una clase como archivo PDF.

Se requiere desarrollar e implementar un control que permita al estudiante exportar a la memoria del dispositivo, uno o varios cuadernos, clases o páginas en formato PDF.

4. Desarrollar módulo para compartir contenido.

   a. Integrar UCABdroide con el resto de las aplicaciones instaladas en el dispositivo mediante el uso del menú compartir de Android y los widgets de escritorio.

      A través de esta funcionalidad, el usuario podrá compartir archivos de otras aplicaciones con UCABdroide mediante el uso del menú compartir del sistema operativo Android, donde el estudiante podrá asociarlo a una materia y después a cualquier clase o a los documentos comunes de dicha materia.

   b. Desarrollar un control que permita compartir desde la galería del cuaderno con uno o varios estudiantes de un curso.

      Este módulo contará con una funcionalidad que permita compartir archivos asociados a una materia con uno o varios estudiantes del curso. Una vez que se haya compartido el archivo, a los otros estudiantes recibirán una notificación de UCABdroide indicando que se le ha compartido un archivo, imagen o nota de voz y le saldrá la opción de descarga.

5. Desarrollar una librería que se encargue de facilitar la sincronización y respaldo con servicios de almacenamiento en la nube

Se requiere desarrollar una librería que contenga API y recursos necesarios basada en el framework Android, para el manejo de sincronización y respaldo de archivos con servicios de almacenamiento en la nube. Entre estos servicios se incluyen Drive, Dropbox y Box. La librería debe ser capaz de implementar las interfaces de cada uno de estos servicios y ofrecer una interfaz unificada para el que referencie esta librería independientemente del servicio de almacenamiento escogido. Esta debe ser reusable e independiente a las aplicaciones que la referencien.

I.5.1.1 Aporte Tecnológico

6. Integrar con otras aplicaciones instaladas en el dispositivo para compartir contenido con las mismas.

En el ecosistema de aplicaciones para el sistema operativo Android es común que las mismas interactúen e intercambien contenidos entre ellas, el mismo sistema operativo provee algunas funciones para facilitar estos intercambios entre aplicaciones instaladas en el dispositivo, por ello se propone desarrollar una interfaz que le permita a la aplicación el intercambio de diferentes tipos de archivos con otras aplicaciones instaladas en el dispositivo mediante el uso del API nativo de Android "*Share*". Las aplicaciones con las cuales el usuario podrá intercambiar

contenido vienen dadas por el tipo de media (*mime type*) que éstas definan en su "*manifest*". Por ejemplo, el usuario podría recibir una imagen por diferentes vías como bluetooth, aplicaciones de mensajerías de texto o incluso mediante correo electrónico y relacionar la misma con un cuaderno de la aplicación directamente desde el menú compartir del sistema operativo Android. El usuario también podría tomar una foto desde la cámara del dispositivo, compartirla directamente con una aplicación de edición de fotos para modificarla y luego asociar la foto modificada con el cuaderno digital de una de sus materias, todo esto con la finalidad de que la aplicación no quede aislada del resto de las aplicaciones y proporcionar al usuario mayor productividad y aprovechar las ventajas del sistema operativo Android.

I.5.1.1 Aporte Funcional

7. Rediseñar el proceso mediante el cual se llevan a cabo los apuntes de clase con dispositivos móviles

Desde hace un tiempo los estudiantes y profesores de la Universidad Católica Andrés Bello han tenido la necesidad de guardar en formato digital las anotaciones que se hacen en el pizarrón de clases, es por ello que cada día es más común ver estudiantes tomando fotos con sus teléfono inteligentes o tabletas al pizarrón de clases y van dejando a un lado el cuaderno, una vez llegada la hora de estudiar encontrar las fotos de una clase en particular puede convertirse en una tarea realmente difícil

para los estudiantes ya que cada día la cantidad de fotos van aumentando con cada clase y con cada foto que fuera de las aulas los estudiantes van tomando en su día a día, es por ello que este nuevo módulo de UCABdroide busca de alguna manera simplificar y mejorar la clasificación de las fotos correspondientes a cada clase y materia.

Esta nueva versión de UCABdroide busca una sustitución quizás no definitiva pero sí en gran medida por parte del estudiante del uso del cuaderno físico por un cuaderno digital el cual le permitirá tomar las notas con menor esfuerzo y prestar más atención en la clase mientras el profesor explica. De esta manera también es más fácil para el profesor hacer llegar todas estas anotaciones y explicaciones a los alumnos que no hayan asistido a su clase ya que la herramienta permitirá a los estudiantes compartir estas anotaciones entre alumnos del mismo curso.

I.5.2 Limitaciones

- Los usuarios finales serán los estudiantes de la UCAB.
- Las nuevas funcionalidades de UCABdroide desarrolladas en este proyecto solo tendrán compatibilidad con Android Icecream Sandwich 4.0.3 (API 15) y superior.
- Para la sincronización y respaldo de cuadernos, notas y apuntes se utilizará como almacenamiento en la nube Google Drive.

- Para compartir con otros estudiantes, se utilizará Google Drive como medio de almacenamiento de los archivos.

- Las pruebas funcionales se realizarán en los siguientes dispositivos Android:

    - API 15 - Dispositivo Virtual.

    - API 16 - Samsung Galaxy Tab 3 Kids.

    - API 19 - Samsung Galaxy Note 2.

    - API 21 - LG G3.

    - API 22 - Google Nexus 5.

I.6 Justificación

La implementación de este módulo de UCABdroide permitirá a los estudiantes, de manera sencilla y organizada, capturar contenido relevante durante la clase, donde posteriormente podrá consultar fácilmente a la hora de estudiar. Además, posee un módulo para la captura de notas en donde los mismos podrán crear apuntes de clase y asociar material capturado, reduciendo tiempo y esfuerzos que pueda tomar al realizarlo de forma tradicional y logrando que los alumnos puedan prestar atención al profesor.

Como complemento, estos apuntes digitales, fotos y audio relacionados a una misma clase podrán ser compartidos, permitiendo a otros estudiantes, del mismo curso o no, acceso al material presentado durante la clase.

CAPÍTULO II – MARCO TEÓRICO

Este capítulo presenta los conceptos teóricos y las tecnologías relevantes para el desarrollo de la solución.

II.1 Universidad Católica Andrés Bello

La Universidad Católica Andrés Bello es una institución de educación superior de la Compañía de Jesús. Su fundación fue decretada por el Episcopado Venezolano en el año de 1951 y realizada en Caracas el año de 1953 por la Compañía de Jesús, a quien pertenece a perpetuidad. Tiene su sede principal en Caracas y núcleos en Los Teques, Puerto Ordaz, Coro y próximamente en la Isla de Margarita.

Es una Institución sin fines de lucro; la fuente de sus ingresos son los estipendios provenientes de matrículas y pensiones estudiantiles, los aportes, donaciones, herencias o legados de personas y comunidades que quieran vincular su nombre a la Institución, y los derivados de cualquier convención lícita cuya celebración se considere.

La universidad está conformada por un total de cinco facultades:

- Facultada de Ciencias Económicas y Sociales.
- Facultad de Derecho.
- Facultad de Humanidades y Educación.

- Facultad de Ingeniería.
- Facultad de Teología.

Es la tercera universidad de Venezuela de acuerdo con el QS World University Rankings® para el año académico 2016. [1]

II.2 Aplicaciones móviles

Una aplicación móvil es una aplicación informática que es diseñada para ser ejecutada en teléfonos inteligentes, tabletas, relojes, televisores y otros dispositivos móviles [2]. El desarrollo de estas aplicaciones requiere tener en cuenta que no todos los dispositivos poseen la misma configuración, hardware, versiones y limitaciones del sistema operativo, densidad y resolución de la pantalla, entre otros.

Durante la etapa de desarrollo de la aplicación, suele usarse IDEs y emuladores de dispositivos que permitan verificar el comportamiento de la misma, para las distintas configuraciones mencionadas.

II.2.1 Tipos de aplicaciones móviles

Una de las decisiones clave que se debe tomar al implementar una aplicación para móviles es decidir qué tipo de aplicación satisface mejor su objetivo. Tener una comprensión básica de los tipos de aplicaciones, incluidos los pros y los contras, ayuda en el éxito de la misma.

II.2.1.1 Aplicaciones nativas

Una aplicación nativa es aquella que se ha desarrollado para su uso en una plataforma en particular mediante el *software development kit* o *SDK* de la misma usando el lenguaje de programación soportado, por ejemplo, para desarrollar en Android se utiliza Java, C++ o Kotlin, en iOS Objective-C o Swift, y en Windows Phone se usa C# (Ver Tabla 1).

Debido a que las aplicaciones nativas están escritas para una plataforma específica, pueden interactuar y aprovechar las características del sistema operativo y otro software que normalmente se instala en esa plataforma. Además, tienen la capacidad de utilizar hardware del dispositivo, lo que significa que las aplicaciones nativas pueden aprovechar la última tecnología disponible en dispositivos móviles, como un sistema de posicionamiento global (GPS) y cámara. Al estar diseñadas directamente para el software del terminal, tendrán un rendimiento optimizado, así como una interfaz mucho más adaptada al sistema operativo al cual el usuario está acostumbrado.

Las aplicaciones nativas son actualizadas frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras.

- Ventajas
    - Velocidad, rendimiento e interfaz de usuario optimizados.
    - Funciona sin conexión a Internet.

Sistema Operativo	Fabricante	Lenguaje
Android	Google	Java, C++ y Kotlin
iOS	Apple	Objective-C y Swift
Windows Phone	Microsoft	C#

Tabla 1. Aplicaciones nativas
Fuente: Elaboración propia

- Desventajas
    - Debe crear una aplicación específica para cada sistema operativo.
    - Toma más tiempo para desarrollar e implementar.
    - Costes de desarrollo más altos.

II.2.1.2 Aplicaciones web

Son aquellas desarrolladas usando lenguajes para el desarrollo web como lo son HTML, CSS y JavaScript, además de un framework para el desarrollo de aplicaciones web, como por ejemplo jQuery. Se podría decir que este tipo de aplicaciones es muy usado para brindar accesibilidad a la información desde cualquier dispositivo, sin importar el sistema operativo, ya que solo se necesita contar con un navegador para acceder a ésta.

- Ventajas
    - No es necesario distribuir utilizando las tiendas de aplicaciones.
    - Funciona en cualquier dispositivo con un navegador, pero la experiencia varía.
    - Menores costos de implementación.
- Desventajas
    - Rendimiento más lento.
    - Se requiere conexión a Internet.

II.2.1.3 Aplicaciones híbridas

Una aplicación híbrida es una aplicación para móviles que contiene un *WebView* (esencialmente una instancia de navegador aislada) para ejecutar una aplicación web dentro de una aplicación nativa, utilizando un contenedor de aplicación nativo que puede comunicarse con la plataforma del dispositivo nativo y la vista web. Esto significa que las aplicaciones web pueden ejecutarse en un dispositivo móvil y tener acceso al dispositivo, como las funciones de la cámara o del GPS.

Las aplicaciones híbridas son posibles debido a las herramientas que se han creado que facilitan la comunicación entre el *WebView* y la plataforma nativa. Estas herramientas no forman parte de las plataformas oficiales de iOS o

Android, pero son herramientas de terceros como *Apache Cordova*. Cuando se crea una aplicación híbrida, se compilará, transformando su aplicación web en una aplicación nativa.

- Ventajas
    - Guarda contenido, por lo que funciona sin conexión a internet.
    - Descargable desde las tiendas de aplicaciones.
    - Implementación de multiplataforma y costos mucho más bajos que las aplicaciones nativas.
- Desventajas
    - El rendimiento de ejecución no es tan bueno como las aplicaciones nativas.
    - En modo *offline,* puede presentar inconsistencias.
    - El contenedor nativo debe ser implementado para cada plataforma.

II.3 Android

Es un sistema operativo para dispositivos móviles de código abierto desarrollado por Google, basado en el kernel de Linux y diseñado principalmente para dispositivos móviles táctiles como teléfonos y tabletas, y en la actualidad, llegando a relojes, televisores y automóviles. Es uno de los sistemas operativos para dispositivos móviles más importante del mundo. [3]

Android permite a los desarrolladores crear distintas aplicaciones aprovechando la variedad de características y utilidades de los dispositivos. Por ejemplo: al crear una aplicación con Android, se podrá implementar alguna funcionalidad que implique usar la función de llamada del teléfono, o la función de mensajería de texto, la de la cámara de foto, etc. La idea es que se pueda crear todo tipo de aplicaciones usando todas las funciones del teléfono que se requiera, y combinándolas como se quiera.

II.3.1 Características de Android

A continuación, se lista algunas de las características más relevantes del sistema operativo: [4]

- **Mensajería**: SMS y MMS, así como *Google Cloud Messaging* y *Firebase*, son algunas de las formas de mensajes en Android.
- **Navegador web integrado**: Basado en el motor de renderizado de código abierto *Blink*, previamente conocido como *WebKit*.
- **Multi-táctil**: Android ofrece una variedad de gestos mediante esta funcionalidad.
- **Multi-tarea**: Es posible ejecutar aplicaciones en tiempo simultaneo, con un manejo adecuado de recursos compartidos.

- **Play Store**: Catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas en dispositivos Android sin la necesidad de una computadora.

- **Conectividad**: Soporte para tecnologías de comunicación como GSM, EDGE, CDMA, UTMS, LTE, Bluetooth, entre otros.

- **Multimedia**: Android ofrece soporte de imágenes, audio y video para gran variedad de formatos multimedia.

- **Almacenamiento externo**: Es posible utilizar almacenamiento externo microSD con formato del sistema de archivo FAT32, Ext3 y Ext4.

- **Soporte para Java**: A pesar de que el lenguaje oficial para el desarrollo en Android es Java, no existe JRE y todo bytecode de Java no podrá ejecutado. En su lugar, Android cuenta con el Android Runtime (mejor conocido por sus siglas ART) que no es más que una máquina virtual especializada diseñada específicamente para Android y optimizada para dispositivos móviles con batería con memoria limitada y CPU. Las clases de Java (.class) pasan por un proceso de transformación en tiempo de compilación llamado DEX que transforma el bytecode de Java a bytecode compatible con ART.

- **Almacenamiento**: Android viene de fábrica con SQLite, un manejador de base de datos relacional ligera.

- **Entorno de desarrollo**: Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado oficial es Android Studio.

II.3.2 Arquitectura de Android

El sistema operativo Android es una pila de componentes de software que se divide en seis secciones y cinco capas principales, como se muestra a continuación en el diagrama de arquitectura (Ver Figura 1). Esta distribución permite acceder a las capas más bajas mediante el uso del *Java API Framework* y librerías nativas para que así el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware de los teléfonos. A continuación, explicamos cada una de las capas que conforman al sistema operativo:

- **Apps del sistema (System Apps):** En la capa superior de la pila de software de Android se encuentran todas las aplicaciones de Android. Ésta en la única capa donde se permite que las aplicaciones desarrolladas sean instaladas. Ejemplos de tales aplicaciones son Contactos Libros, Navegador, Juegos, etc.

- **Framework de la API en Java (Java API Framework)**: En la parte superior de las bibliotecas nativas y android runtime, se encuentra el framework de Android. Todo el conjunto de funciones de Android está disponible mediante API escritas en el lenguaje Java. Estas API son los

cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como interfaz de usuario, telefonía, recursos, ubicaciones, proveedores de contenido (datos) y administradores de paquetes. Proporciona una gran cantidad de clases e interfaces para el desarrollo de aplicaciones Android.

II.3.3 Fundamentos de una aplicación Android

Las aplicaciones Android están escritas en los lenguajes de programación Java o Kotlin (es posible usar C/C++ como complemento). Las herramientas del SDK de Android compilan el código junto con cualquier archivo de recursos y data en un APK (siglas en inglés de *Android Package)*, el cual es un archivo con extensión ".apk". Un archivo APK contiene todo lo necesario para que la aplicación sea instalada en el dispositivo.

Cada aplicación Android vive dentro de su entorno aislado (del inglés *sandbox*) protegido por las siguientes características de seguridad:

- Android es un sistema operativo basado en Linux con capacidad de multi-usuario en el cuál, cada aplicación, es un usuario distinto.

- Por defecto, el sistema asigna a cada aplicación un identificador de usuario distinto (que no es conocido por aplicación). El sistema establece permisos para todos los archivos de una aplicación de modo que sólo el identificador de usuario asignado a esa aplicación pueda acceder a ellos.

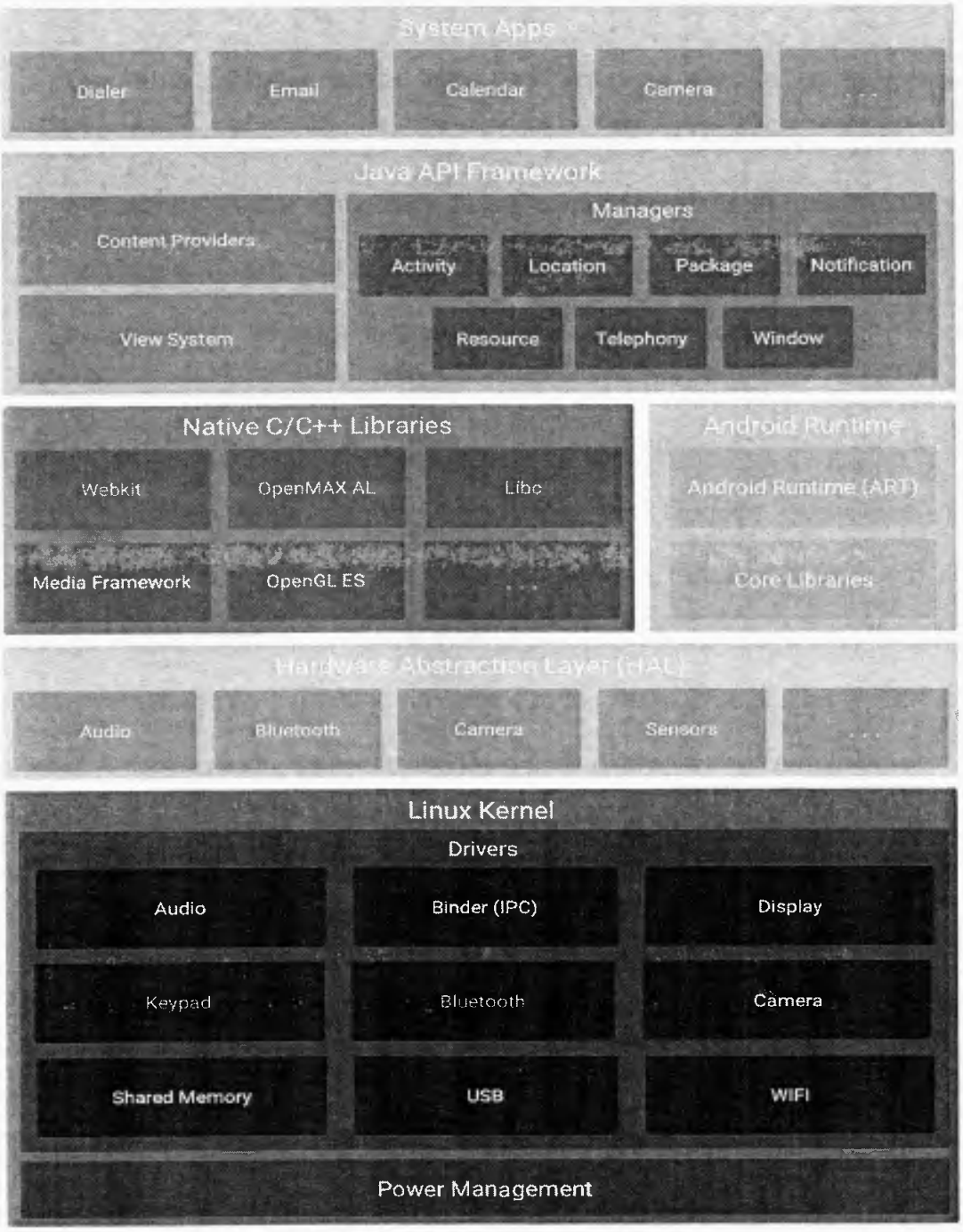

Figura 1. Pila de software de Android
Fuente: Android Developers [5]

- Cada proceso tiene su propia instancia de ART o *Android Runtime*, permitiendo que el código se ejecute aislado de otras aplicaciones.

- De forma predeterminada, cada aplicación se ejecuta en su propio proceso de Linux. El sistema Android inicia el proceso cuando es necesario ejecutar cualquiera de los componentes de la aplicación y, a continuación, cierra el proceso cuando ya no es necesario o cuando el sistema debe recuperar la memoria para otras aplicaciones.

II.3.3.1 Componentes de una aplicación Android

Los componentes de una aplicación son bloques esenciales de construcción de la misma, ya que cada componente representa un punto de entrada a través del cual el sistema o un usuario puede ingresar a la aplicación. Algunos componentes dependen de otros y todos están incluidos en el SDK de Android. A continuación, se presentan los 4 tipos diferentes de componentes:

- **Actividades (Activities)**: Una actividad representa una sola pantalla con una interfaz de usuario.

- **Servicios (Services)**: Un servicio es un componente que se ejecuta en segundo plano para realizar operaciones de larga duración o para realizar trabajos en procesos remotos (ejemplo: otras aplicaciones). Un servicio no proporciona una interfaz de usuario.

- **Proveedores de contenido (Content Providers)**: Un proveedor de contenido administra un conjunto compartido de datos de aplicación.

Puede almacenar los datos en el sistema de archivos, en una base de datos SQLite, en la web o en cualquier otra ubicación de almacenamiento permanente a la que pueda acceder su aplicación. A través del proveedor de contenido, otras aplicaciones pueden consultar o incluso modificar los datos (si el proveedor de contenido lo permite).

- **Receptores de mensajes (Broadcast Receivers)**: Un receptor de mensajes es un componente que responde a anuncios del sistema, por ejemplo, una emisión que anuncia que la pantalla se ha apagado, que la batería está baja o que se ha capturado una imagen. Las aplicaciones también pueden iniciar transmisiones.

II.3.3.2 El archivo de manifiesto de una aplicación Android

Antes de que el sistema Android pueda iniciar una aplicación, el sistema debe revisar que el componente existe, leyendo el archivo de manifiesto de la aplicación "AndroidManifest.xml" (Ver Figura 2). La aplicación debe declarar todos sus componentes en este archivo, que debe estar en la raíz del directorio del proyecto de la aplicación.

Además de declarar todos los componentes de una aplicación, el archivo de manifiesto sirve para declarar información importante de la aplicación como son:

- Los permisos que requiere la aplicación, como acceso a internet o a la ubicación del dispositivo

- El nivel de API mínimo necesario para el funcionamiento de la aplicación. Ciertos componentes del SDK de Android no están disponibles en todas las versiones y es necesario declarar la mínima versión de Android en la cual la aplicación es compatible y esta verificación se realiza cuando se instala.

- Características de hardware que la aplicación necesita. Si la aplicación requiere obligatoriamente que el dispositivo, por ejemplo, cuente con bluetooth o cámara, se declara en este archivo para que sea verificado cuando la aplicación se esté instalando.

- Entre otros.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.test.appstructure"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="4" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MainActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figura 2. Archivo de manifiesto “AndroidManifest.xml”
Fuente: Elaboración propia

II.3.3.3 Recursos de una aplicación Android

Además de código, una aplicación puede requerir recursos independientes al código fuente, como audio, imágenes y otros elementos relacionados con la presentación de la misma, por ejemplo: animaciones, menús, estilos, texto, colores y diseño de la interfaz de usuario del componente de actividad. Estos elementos representativos son declarados en un archivo de tipo XML, aunque estas representaciones pueden ser construidas en el código Java.

Por cada recurso que es declarado en el proyecto, la herramienta de compilación de Android define un identificador único de recurso, con la finalidad de poder ser referenciados desde el código u otros archivos de recurso XML.

Uno de los aspectos más importantes de proveer recursos que son independientes al código, es la capacidad de ofrecer recursos alternativos para las distintas configuraciones de dispositivos. Por ejemplo, al definir texto de interfaz de usuario en XML, es posible traducir el mismo a otros idiomas y guardarlas en archivos independientes. Luego, en función de un calificador de idioma que se anexe al nombre del directorio de recursos (como *res/values-fr/* para los valores de texto en francés) y a la configuración de idioma del usuario, el sistema Android aplica los textos en el idioma correspondientes a la interfaz de usuario.

II.3.4 Android Studio

Basado en entorno de desarrollo integrado de código abierto IntelliJ Community Edition, fabricado por la empresa JetBrains, es el IDE oficial para el desarrollo en la plataforma Android desde junio de 2014 y disponible para Windows, GNU/Linux y Macintosh OS X. [6]

Posee una gran variedad de características que simplifican el proceso de desarrollo al programador, entre algunas se mencionan:

- Soporte de compilación basado en *Gradle*.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas lint para obtener rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Integración *ProGuard* y capacidades de firma de aplicaciones.
- Asistentes basados en plantillas para crear diseños y componentes comunes para Android.
- Un editor de diseño que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario, opción para pre-visualizar diseños en configuraciones de pantalla múltiple. [7]
- Soporte para crear aplicaciones Android Wear (reloj) y Android TV (televisor).

- Soporte integrado para *Google Cloud Platform*, que permite la integración con *Firebase Cloud Messaging* (anteriormente *Google Cloud Messaging*) y *Google App Engine*. [6]

II.4 UCABdroide

UCABdroide, la aplicación Android para ucabistas desarrollada por la Dirección de Tecnologías de Información (DTI) de la UCAB, donde les ha permitido a los estudiantes verificar de forma sencilla la información académica de mayor importancia. Los alumnos pueden verificar su horario, notas, retenciones y turno de inscripción, entre otras cosas, además de mantenerse al día sobre los eventos que se realizan en la universidad [8]. Es la aplicación sobre la cual se desarrollará e implementará este trabajo de grado.

II.5 Java

"Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems". En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. [9]

II.6 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación.

Aunque cada aplicación es única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas etc.

II.6.1 Modelo-Vista-Presentador (MVP)

Este patrón de diseño busca separar la interfaz gráfica de usuario con la lógica de la aplicación. Está conformada por 3 componentes (Ver figura 3): La **vista**, que se encarga de mostrar la información al usuario y de interactuar con él para hacer ciertas operaciones. Luego el **modelo**, ignorante de cómo la información es mostrada al usuario, realiza toda la lógica de las aplicaciones usando las entidades del dominio. Y por último tenemos al **presentador** que es el que "presenta" a ambos actores sin que haya ningún tipo de dependencia entre ellos.

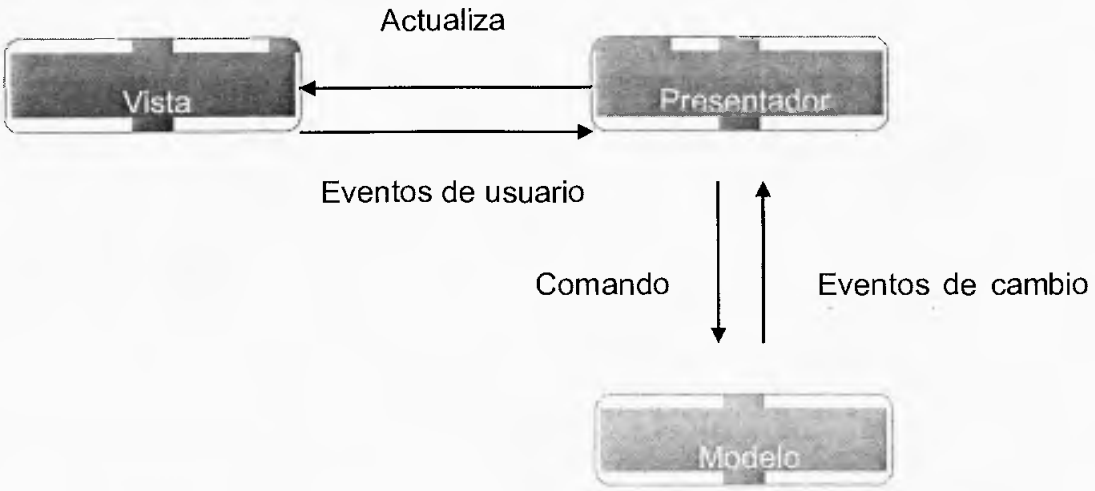

Figura 3. Modelo Vista Presentador
Fuente: Elaboración propia

II.6.2 Repositorio

El repositorio es, de hecho, el punto de entrada para que la aplicación acceda a la capa de dominio. Para decirlo simplemente, el repositorio permite que todo el código utilice objetos sin tener que saber cómo se persiste el objeto. El repositorio contiene todo el conocimiento de la persistencia, incluida la asignación de tablas a objetos. Esto proporciona una vista más orientada a objetos de la capa de persistencia y hace que el código de asignación sea más encapsulado.

II.6.3 Inyección de dependencias

Este patrón especifico de la programación orientada a objetos, basada en los conceptos de modularidad y reutilización, y derivado del patrón Inversión de Control; consiste en suministrar a las instancias de las clases, todas las dependencias a otras ya instanciadas en lugar que sea ésta la que cree los objetos.

CAPÍTULO III – MARCO METODOLÓGICO

Este capítulo tiene como objetivo describir la metodología implementada en el desarrollo de la aplicación de Android para la gestión de un cuaderno digital para los estudiantes de la Universidad Católica Andrés Bello (UCAB), dicha metodología es Scrum.

III.1 Scrum

"Scrum es una metodología de desarrollo ágil, definiendo la forma de abordar un proceso de desarrollo de software de forma ágil y liviana a través de la descripción de un conjunto de roles, componentes y organización de la actividad diaria" [15].

Scrum es un marco de trabajo en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos [16].

En Scrum el proceso de un proyecto se ejecuta en bloques cortos y fijos (iteraciones de un mes natural y hasta dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

III.1.1 Características de Scrum

- Es una metodología ágil.
- Es liviano, iterativo e incremental.
- Se realizan entregas funcionales.
- Simplicidad.
- Motivación y responsabilidad.
- Adaptativo y flexible.
- Mitigación de riesgos.
- Resultados al cambio.

III.1.2 Proceso de Scrum

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevos requerimientos listos para ser usados [17].

En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevos requerimientos priorizando siempre aquellos que aporten mayor valor de negocio [17].

- **Sprint Planning**: Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de tiempo que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir [17].

- **Sprint**: Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo. [17]

- **Daily Scrum**: Reunión diaria de máximo 15 minutos en la cual el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos [17].

- **Sprint Review**: Lista de las tareas necesarias para llevar a cabo las historias del sprint [17].

- **Sprint Retrospective**: Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discuten acerca de cómo perfeccionarlos [17].

III.1.3 Roles de Scrum

- **Product Owner**: Es la persona que habla por el cliente, el mismo asegura que se cumpla las expectativas. Es el responsable del proyecto. Identifica los elementos del Product Backlog.

- **Scrum Master**: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI [17].

- **Development Team**: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a las que se comprometen al inicio de cada sprint [17].

III.1.4 Artefactos de Scrum

- **Product Backlog**: Es una lista ordenada de todo lo que podría ser necesario para el producto, y la única fuente de requerimientos para cualquier cambio a realizarse. Contiene todas las características funcionales, requerimientos, mejoras y correcciones del producto, y otros atributos como descripción, ordenación y estimación. [18]

- **Sprint Backlog**: Es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint, más un plan para entregar el Incremento y conseguir el Objetivo. "Define el trabajo que se llevará a cabo para convertir elementos de la pila en un incremento hecho". [18]

- **Product Increment**: Parte del producto desarrollada en un sprint, en condiciones de ser usada. Resultado de cada sprint. [19]

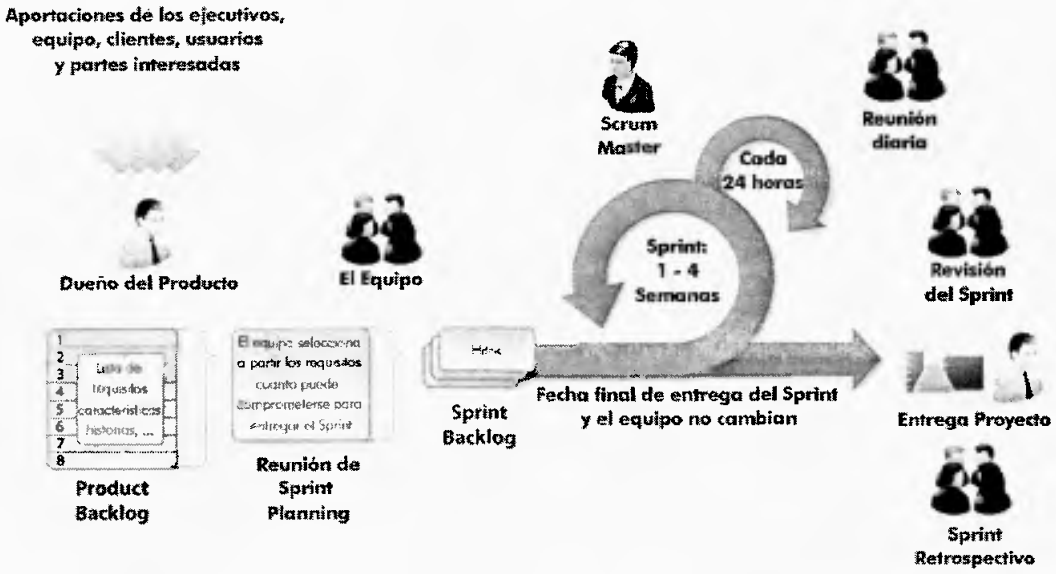

Figura 4. Metodología Scrum
Fuente: Isla Visual, 2014

III.2 Justificación de la metodología

Se escogió como metodología de trabajo Scrum por todos sus beneficios. Entre ellos, dado que es una metodología ágil y se trabaja colaborativamente en equipo, se agilizan los tiempos y la aceptación de cada entrega, ya que éstas han sido revisadas por los involucrados por parte de la Dirección de Tecnologías de Información (DTI) de la UCAB.

Las reuniones cortas diarias (Daily Scrum), las revisiones y retrospectiva presentando un demo con las nuevas funcionalidades fueron las otras características que formaron parte en la toma de decisión de Scrum como metodología de trabajo.

En el Scrum diario se reúne todo el equipo y se planifica el trabajo, con esta reunión también se trata de encontrar las funcionalidades que estén deficientes y en cómo se pueden mejorar. Al finalizar cada sprint se hace entregan entregas funcionales y se planifican las pautas para continuar el desarrollo.

Estas reuniones son de gran importancia ya que, de encontrar alguna deficiencia o alguna duda en los requerimientos, se planificaba una reunión directamente con la DTI y los distintos docentes que al final, serán los usuarios de la aplicación.

Dada la magnitud del proyecto y por la cantidad de estudiantes que serán los usuarios finales, cada conclusión obtenida de las reuniones es importante para lograr la satisfacción de estos. Scrum es adaptable al cambio, esto quiere decir que aparte de seguir un plan de trabajo, se permite la mejora constante del producto, se pueden realizar modificaciones cuando sean solicitadas y no al momento de la entrega final, siendo una metodología adaptativa. Este aspecto es fundamental, aunque existe una visión clara del alcance del sistema, es muy

probable que de las reuniones con los usuarios surjan funcionalidades adicionales que pudieran ser incluidas en el proyecto.

III.3 Planificación de la implementación de la solución

A continuación, se describe la dinámica de trabajo implementada en el desarrollo de la solución. Se listará por cada Sprint el conjunto de actividades realizadas.

Sprint	Actividades	Entregables	Objetivos
0	• Levantamiento de información sobre aplicación UCABdroide. • Selección de herramientas de desarrollo. • Migración del proyecto existente a nuevo entorno de desarrollo. • Definición de estructuras para el módulo de cuaderno digital.	• Product Backlog del proyecto. • UCABdroide en nuevo entorno de desarrollo. • Diagrama entidad-relación de base de datos. • Librería de sincronización y respaldo con servicios de	• 5

	• Diseño e implementación de almacén de datos en el dispositivo móvil. • Desarrollo de librería de respaldo y sincronización con la nube.	almacenamiento en la nube.	
1	• Rediseño del proceso de apuntes con dispositivos móviles. • Desarrollo de módulo de gestión de notas y apuntes.	• Módulo de gestión de notas y apuntes.	• 7 • 1
2	• Diseño de módulo de galería de archivos. • Desarrollo de control de visualización de archivos.	• Módulo de galería de archivos.	• 2

	• Desarrollo de control de asociación de archivos. • Desarrollo de widget para captura de foto o nota de voz. • Diseño e implementación de estructura de etiquetas.		
3	• Diseño de estructura de archivo de cuadernos. • Desarrollo de control para la exportación e importación de cuadernos. • Desarrollo de control de sincronización y respaldo de cuadernos. • Desarrollo de control de exportación de	• Módulo de respaldo y sincronización.	• 3

	cuaderno como archivo PDF		
4	• Integrar UCABdroide con aplicaciones instaladas en el dispositivo. • Desarrollo de control para compartir desde la galería del cuaderno con estudiantes de un curso.	• Módulo para compartir contenido.	• 4 • 6

Tabla 2. Sprint Backlog
Fuente: Elaboración propia

CAPÍTULO IV – DESARROLLO

En este capítulo se explica cómo usando la metodología descrita anteriormente se logra la solución planteada.

IV.1 Sprint 0

IV.1.1 Inicio

El objetivo principal de este sprint es la transferencia de información y análisis de la misma, entre los involucrados del equipo Scrum. En este punto se definen las siguientes reuniones, reconocimiento del equipo, roles de cada integrante, entre otros.

Adicionalmente, se planificó una reunión con las personas encargadas del desarrollo de UCABdroide para conocer el proyecto y estructura, de manera de seguir los mismos estándares y lineamientos para así tener uniformidad entre los desarrollos creados.

Para la aplicación a implementar se decide usar el lenguaje de programación Java y el entorno de desarrollo integrado (IDE) Android Studio, el cuál brinda al usuario ciertas características que agilizan el desarrollo de aplicaciones como señalamiento de código no seguro, depuración, refactorización, entre otros. Debido a que UCABdroide fue programado en Eclipse (un IDE obsoleto para el momento de inicio de este sprint) en favor de Android Studio, se migró el código base al nuevo entorno.

Para la aplicación se decide usar una librería ORM que trabaje con la base de datos SQLite, la cual viene integrada con el sistema Android. Este tipo de herramientas elimina la escritura de consultas SQL para interactuar con SQLite, se encarga de crear base de datos (comandos DDL), gestiona relaciones de objetos y proporciona API claras y simples para operaciones de base de datos.

IV.1.2 Product Backlog

En la Tabla 3 se puede observar el Product Backlog del proyecto.

Product Backlog
La aplicación debe ser parte de UCABdroide.
Seguir los estándares de UCABdroide, tanto de interfaz gráfica como patrones de diseño MVP, inyección de dependencias y repositorio.
El desarrollo debe ser compatible hasta la versión 15 de Android.
La aplicación permitirá a los estudiantes la gestión de notas y apuntes.
La aplicación permitirá a los estudiantes la gestión de galería de archivos, imágenes y audio.

La aplicación permitirá a los estudiantes la gestión de respaldo y sincronización de contenido.
La sincronización de archivos, imágenes y audio con la nube podrá realizarse usando datos del operador (EDGE, 3G, LTE) si y solo si el estudiante desea realizar la operación de forma manual.
La sincronización debe ser con Google Drive utilizando la cuenta Google que desee el estudiante.
La aplicación permitirá a los estudiantes compartir contenido con otros.

Tabla 3. Product Backlog
Fuente: Elaboración propia

Para el desarrollo de este proyecto se definieron los roles y sus responsables:

- Product Owner
    - Benal, Roquefelix (jefe de la unidad de desarrollo).
- Scrum Master
    - Benal, Roquefelix (jefe de la unidad de desarrollo).
- Scrum Team

- Bereciartu, Víctor (coordinador de servicios web).
- Esperanza, María Luis (jefe de la unidad de calidad).
- Díaz, Pedro (desarrollador).

IV.1.3 Diseño e implementación de almacén de datos

IV.1.3.1 Creación del Modelo Entidad-Relación

A partir del levantamiento de información surgió la necesidad de la creación del modelo entidad-relación. Éste debe representar las estructuras definidas para el módulo de cuaderno digital, como lo son: cuadernos, páginas, notas de texto, contenido multimedia asociados y trazos. En la Figura 5 se puede observar el diagrama entidad-relación resultante.

IV.1.3.1 Creación de la base de datos

Partiendo del modelo entidad-relación, se construyó y relacionó los objetos Java que representen a éste modelo, para que la herramienta ORM, asegurando el respeto a todas las restricciones de integridad establecidas para lograr la consistencia de la información, cree la base de datos local en tiempo de ejecución (de no existir). A lo largo del desarrollo, la base de datos estuvo en constantes cambios hasta finalizar la solución planteada en este Trabajo Instrumental de Grado.

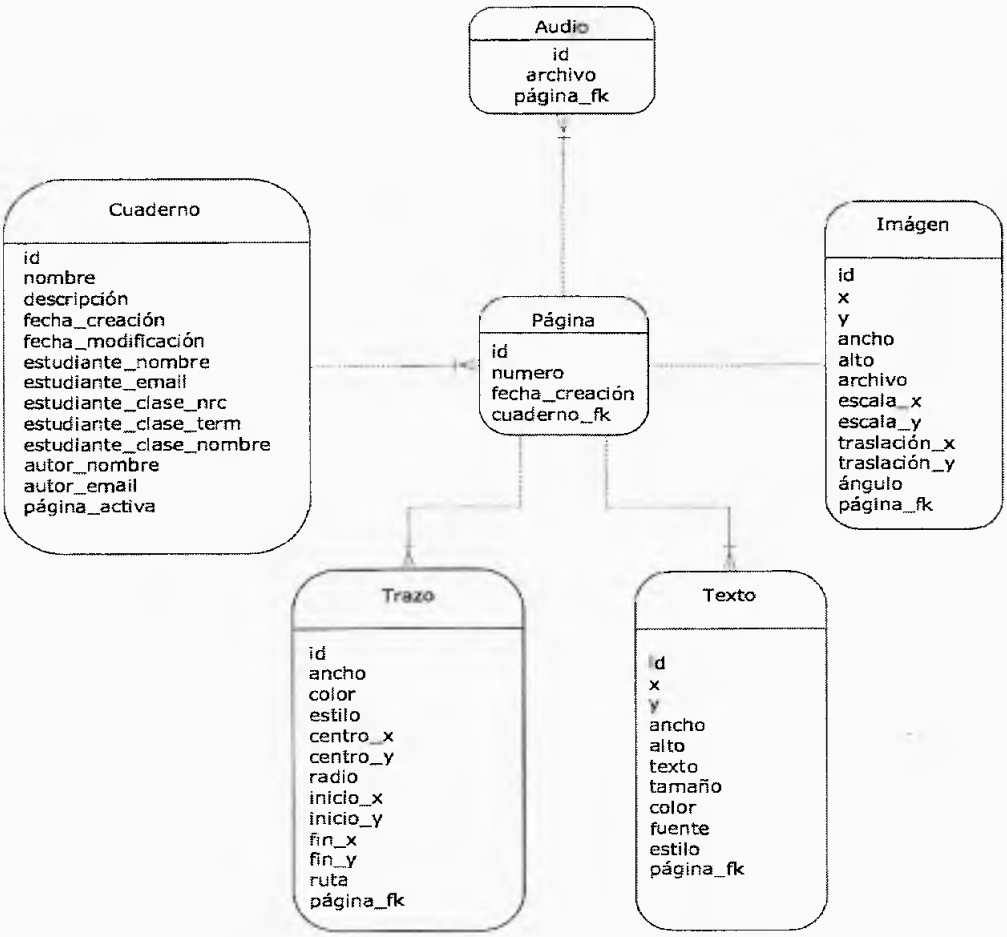

Figura 5. Diagrama entidad-relación
Fuente: Elaboración propia

IV.1.4 Desarrollo de librería de respaldo y sincronización con la nube

Se desarrolló una librería dependiente al framework de Android (conocidas como "aar" por sus siglas Android ARchive) que se encarga de facilitar la sincronización y respaldo con servicios de almacenamiento en la nube.

La librería implementa las interfaces de los servicios Google Drive, Dropbox y Box y ofrece una interfaz unificada para que, la aplicación que referencie esta librería, pueda utilizar el API independientemente del servicio de almacenamiento escogido.

Esta librería no está implementada en el mismo workspace de UCABdroide, aunque la aplicación la referencia para ser utilizada por el módulo de sincronización y respaldo.

IV.2 Sprint 1

Para realizar el rediseño del proceso de apuntes de los estudiantes con ayuda de dispositivos móviles, se analizó cómo se realiza el proceso actual. Para ello, se llevó a cabo una serie de reuniones con algunos estudiantes que forman parte del DTI.

IV.2.1 Rediseño del proceso de apuntes con dispositivos móviles

Se detectó que los estudiantes que usan los dispositivos móviles durante el proceso de toma de apuntes de una clase para capturar fotos del pizarrón como complemento a lo que escriben en cuaderno físico o como reemplazo del mismo, deben abrir la aplicación nativa de cámara, capturar la foto y luego, si el dispositivo lo permite, debe organizar las fotos de tal manera que a la hora de estudiar sepa dónde encontrarlas. La organización del material puede hacerse durante o después de clase: durante la clase el estudiante puede perder el hilo al profesor mientras mueve las fotos a una carpeta donde las agrupe y después

de clase, se corre con el riesgo que se olvide de hacerlo y depende de su memoria para poder decir que foto pertenece a que clase. Para ambos casos requiere tiempo.

Para reducir a cero el tiempo que emplea el estudiante en agrupar y categorizar el material digital capturado en clase, desde la aplicación desarrollada, se podrá realizar fotos y asociarlas a una clase o materia de forma automática.

En la Figura 6 se puede ver, a través de la comparación de diagramas de flujo, la diferencia entre el proceso actual y el nuevo proceso de apuntes con dispositivos móviles.

IV.2.2 Desarrollo de módulo de gestión de notas y apuntes

El módulo de gestión de notas y apuntes es, por decirlo de alguna manera, el corazón de este trabajo de grado, donde el estudiante podrá realizar apuntes y notas utilizando la aplicación UCABdroide.

Después de sostener reuniones con el equipo de trabajo de DTI, se definió los elementos que conforman este módulo y como debe ser el proceso de administración de notas y apuntes, de tal forma, que el usuario pueda a través de una interfaz sencilla, crear, eliminar y editar cuadernos.

Uno de los objetivos que busca el módulo gestión de notas y apuntes es que los cuadernos queden organizados por materia para facilitar a los estudiantes consultarlos de forma rápida.

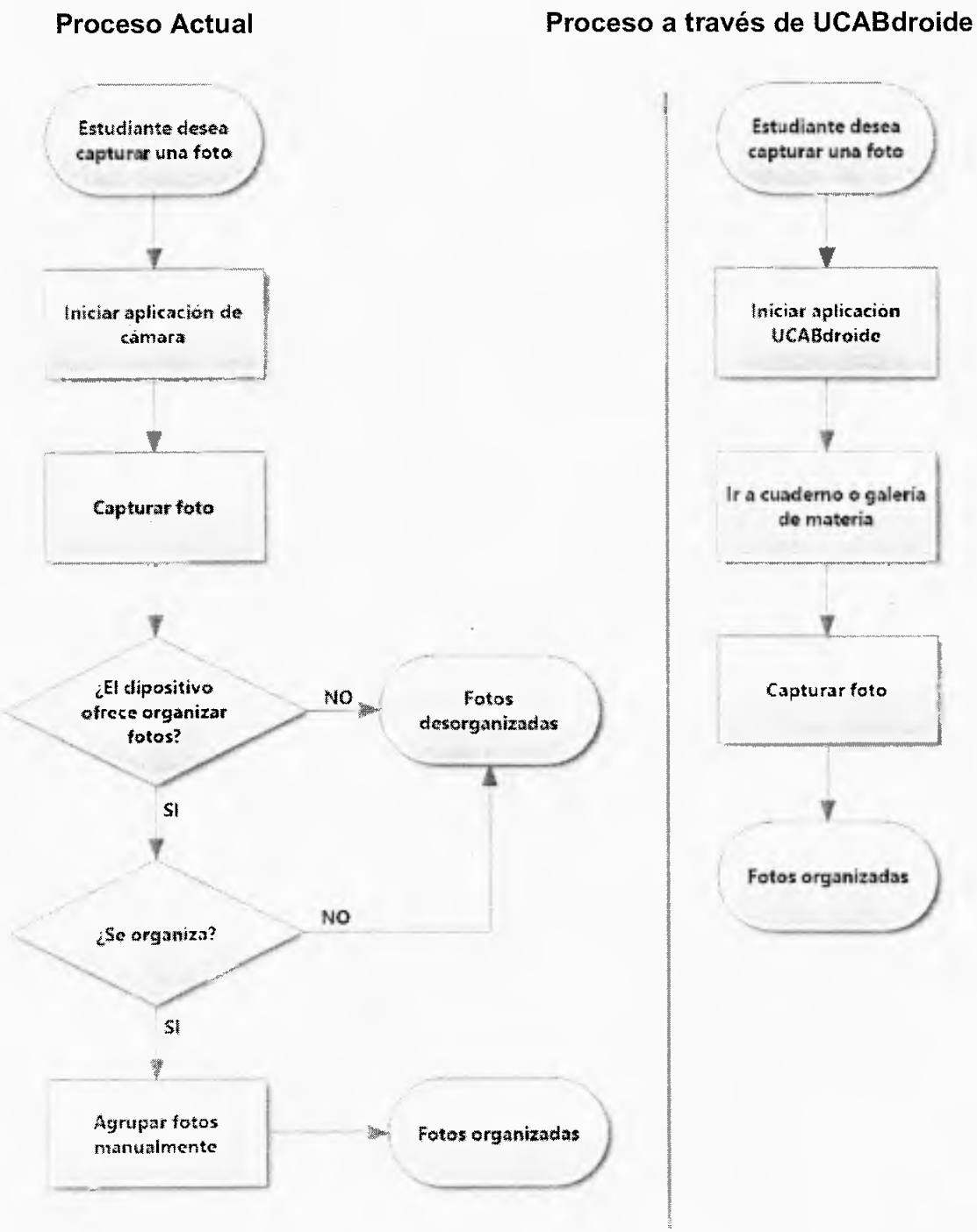

Figura 6. Comparación de flujo de toma de apuntes con móviles
Fuente: Elaboración propia

Para logarlo, la pantalla de gestión de cuadernos debe conocer la asignatura para la cual se está administrando las notas. Es por esto que se utilizó la pantalla principal de UCABdroide, la cual muestra la lista de materias que el estudiante está cursando actualmente, como punto de entrada al mismo, permitiéndole conocer la asignatura para la cual se está gestionando las notas. El estudiante podrá acceder a la pantalla abriendo el menú asociado a cada una de las materias, donde observará la opción "Ir a cuaderno" (ver Apéndice A.1 Figura 1).

En la primera pantalla de este módulo (ver Apéndice A.1 Figura 2) se presenta la lista de materias relacionadas a la asignatura. Por medio del patrón de repositorio que forma parte del modelo, el presentador de la vista consulta y muestra los cuadernos existentes del estudiante para dicha materia. Esta lista se presenta de forma ordenada, mostrando las notas con fecha de creación reciente de primero con la finalidad de que el usuario tenga referencia cronológica de los cuadernos creados.

Para cada uno de los elementos de la lista, se brinda información relevante acerca del cuaderno: nombre, período para el cual fue creado, la materia a la que pertenece, número de páginas y fecha de creación y última edición.

Adicionalmente, cada ítem posee un menú contextualizado, donde el estudiante podrá realizar acciones sobre cada uno de los cuadernos:

- **Exportar**: El cuaderno y todas las estructuras relacionadas al mismo, son serializados en un archivo de formato ".ucab" que se crea en la memoria

del dispositivo, el cual podrá ser importado nuevamente o compartido a otros estudiantes.

- **Exportar PDF**: El cuaderno y todas las estructuras visuales relacionadas al mismo, son representados en un documento PDF que se crea en la memoria del dispositivo. El usuario podrá en cualquier plataforma, consultar el cuaderno usando el visualizador de documentos PDF de preferencia.

- **Renombrar**: El estudiante podrá renombrar en cualquier momento, el título del cuaderno.

- **Borrar**: Por último, el usuario puede borrar cuando considere conveniente, el cuaderno y todos los elementos que éste contiene. Esta operación es permanente y se pide confirmarla para proceder. El estudiante depende de *backups* realizados previamente al mismo de querer restaurarlo.

También se agregó en esta pantalla un botón flotante (ver Apéndice A.1 Figura 2) con dos funcionalidades:

- **Crear cuaderno**: Por medio de esta funcionalidad, se muestra un diálogo donde pide el título del cuaderno a crear. Una vez confirmado, se crea un nuevo registro en la tabla de cuadernos y se actualiza la lista de los mismos.

- **Crear nota de audio**: Al hacer seleccionar esta funcionalidad, se inicia un proceso de segundo plano utilizando uno de los componentes de Android

llamado "servicio", donde inmediatamente se comienza a grabar audio utilizando el micrófono del dispositivo. El estudiante notará una notificación persistente en la barra de estado de Android indicando que se está grabando audio y con un botón para detener y guardar la misma. Este servicio es independiente al estado de la aplicación, es decir, seguirá corriendo en segundo plano, aunque la aplicación sea cerrada.

Al hacer *click* sobre un cuaderno, la aplicación lleva al estudiante a la segunda pantalla de este módulo, que es donde es estudiante podrá, mediante una serie de controles, realizar notas y apuntes de clase.

Esta pantalla consta de una serie de paneles que permite al usuario crear y manipular las notas de clase (ver Figura 7). Para la realización de esta ventana, se implantaron vistas (paneles) personalizados para lograr la funcionalidad deseada:

- **Panel A – Borrador**: Este control es de tipo mano alzada, es decir, por donde pase el dedo, borra los trazos realizados. El borrador implementa un complejo algoritmo de intersección y exclusión de los trazos realizados por la herramienta lápiz que sean comunes con los trazos realizados por esta herramienta. De no estar seleccionado, el *input* de trazo a mano alzada del usuario es de tipo lápiz.

- **Panel B – Manipulación de contenido**: Este control permite indicar a cuál tipo de contenido que conforman una página (texto, imagen y trazos), se

le realizaran *gestures* para su manipulación. Debido a que dichos contenidos pueden estar solapados, es necesario contar con un panel que permita seleccionar el elemento deseado (ver Apéndice A.1 Figura 3).

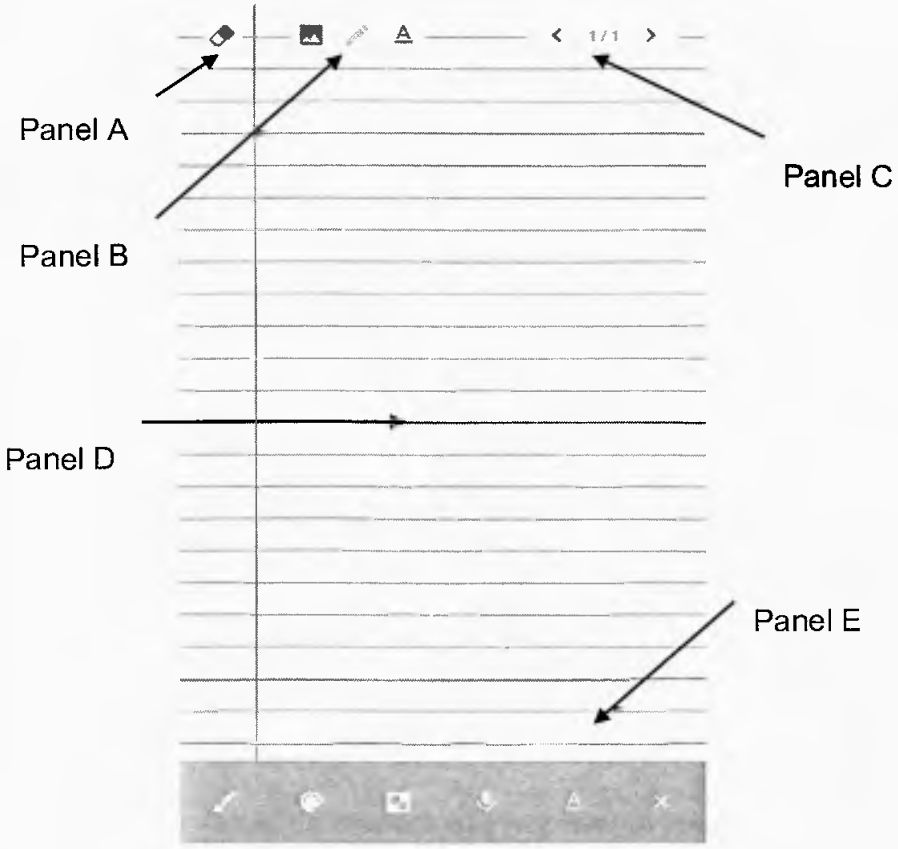

Figura 7. Paneles de edición de cuaderno digital
Fuente: Elaboración propia

A continuación, se describe cada componente que conforma el panel, de izquierda a derecha como se aprecia en la Figura 7:

- **Imagen**

Mediante esta funcionalidad, el usuario es capaz de aplicar ciertas transformaciones a la imagen contenida en la página. Este control implementa detección de *gestures* del usuario para realizar dichas transformaciones:

- Usando un solo dedo podrá ajustar la posición en el eje de coordenadas en el plano del canvas (panel D) al gusto.
- Usando el *gesture pinch-in* y *pinch-out* se disminuye y aumenta respectivamente, el tamaño de la imagen en pantalla.
- Usando el *gesture* de rotación, se ajusta el ángulo deseado para la imagen.

Cabe destacar que todas estas transformaciones aplicadas a la imagen asociada a una página del cuaderno son almacenadas en base de datos, de manera que la próxima vez que se visualice, conserve el último estado.

Por razones de rendimiento, se tiene como limitación una imagen por página. Durante las reuniones previas a la implementación de este módulo, se analizó y se llegó a la

conclusión que mantener varias imágenes en memoria resulta muy costo y en dispositivos de gama media y baja, los efectos de lentitud de la aplicación pueden “dañar”, por así decirlo, la experiencia del usuario.

- **Lápiz**

Este control permite indicar que los *gestures* a realizar serán de tipo mano alzada. Usando el dedo se podrá dibujar en el canvas del panel D, la ruta por donde pase el mismo. Es estilo, grosor y color de los trazos son configurados por medio del panel E.

La implementación de esta funcionalidad requirió tener conceptos fundamentales de computación gráfica y conocer en profundidad como Android maneja los eventos de movimiento.

En una primera implementación se realizó una vista personalizada que capture los eventos de movimiento, conformado por coordenadas x/y, y uniendo los puntos mediante líneas. Se obtuvo como resultado lo que se muestra en la Figura 8.

Figura 8. Primera implementación de trazo a mano alzada
Fuente: Square Corner Blog

Aunque fue relativamente fácil de implementar, el resultado no fue el deseado. Las curvas son cuadradas debido a la poca cantidad de eventos de movimientos (puntos x/y) reportados por Android. Luego se descubrió que los eventos que son entregados vienen por batches, y que habría que consultar con el historial de movimiento ocurrido entre los mismos para obtener todos los puntos y utilizarlos para obtener un trazo más suave. En la Figura 9 se observa la mejora.

Figura 9. Segunda implementación de trazo a mano alzada
Fuente: Square Corner Blog

Aunque se mejoró notablemente la calidad del trazo, aún hubo la necesidad de perfeccionarlos. Un punto del plano puede definirse por coordenadas. Por ejemplo, un punto A tiene unas coordenadas (x1, y1) y a un punto B le corresponde (x2, y2). Para trazar una recta entre ambos basta con conocer su posición. Pero si en lugar de unir dos puntos con una recta se unen con una curva, surgen los elementos esenciales de una curva Bézier; los puntos se denominan «puntos de anclaje» o «nodos».

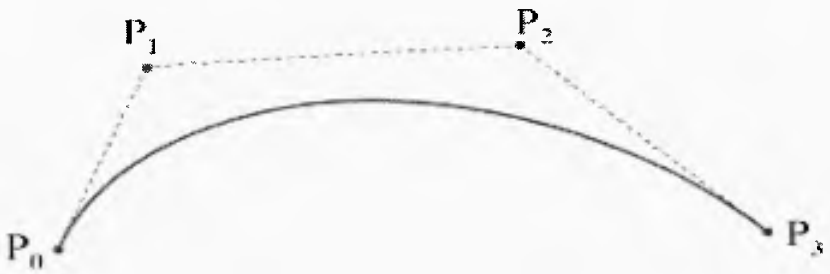

Figura 10. Curva de Bézier
Fuente: Wikipedia

La forma de la curva se define por unos puntos invisibles en el dibujo, denominados «puntos de control», «manejadores» o «manecillas» (ver Figura 10). Y dado a que Android reporta el historial completo de coordenadas de trazas, se decidió usar esta técnica para suavizar aún más las curvas de los trazos. El resultado final puede apreciarse en la Figura 11.

- **Texto**

Por medio de este control, el usuario podrá ajustar la posición y dimensiones del cuadro de texto en el canvas mediante el uso de *gestures* de un solo dedo. Como limitante, únicamente se podrá manejar un cuadro de texto por página y es posible, como mejora de este módulo, introducir la posibilidad de manejar múltiples.

Figura 11. Implementación final de trazo a mano alzada
Fuente: Square Corner Blog

- **Panel C – Control de página**: Durante las reuniones del sprint con el equipo de trabajo sobre el diseño del módulo, se decidió crear un panel donde el usuario pueda crear páginas y navegar entre ellas.

Una vez cargada la pantalla, el usuario podrá observar el número de página en la que se encuentra y dos fechas, izquierda y derecha. La flecha derecha, además de permitir la navegación a páginas más recientes, permite la creación de nuevas páginas solo cuando, estando en

la última de ellas, es presionada. La fecha izquierda solo tiene como funcionalidad navegar a páginas más antiguas.

Una página puede contener una imagen, un cuadro de texto y los trazos que desee el usuario, cada uno de ellos posee atributos distintos que deben ser almacenados en base de datos. La carga de la misma es una operación muy costosa, ya que requiere lectura de base de datos, procesar la información y renderizarla en la pantalla. Hacer esto en el *UI thread* (hilo de interfaz gráfica - hilo por defecto donde corre una aplicación Android), resultó en un programa no responsivo mientras se ejecutará esta tarea. Se implementó un proceso asíncrono, donde fuera del *UI thread*, se prepara la información a cargar lista para ser renderizada y con ayuda del patrón de observador, obtener los resultados de forma inmediata

- **Panel D – Canvas**: Esta sección de pantalla es donde se representan las notas, apuntes e imágenes realizadas en páginas de cuadernos.

Además de ser el encargado del renderizado de información, tiene la responsabilidad de informar al panel B sobre *gestures* realizados sobre el mismo. Está compuesto por 3 capas invisibles superpuestas, donde cada una trabaja con un tipo de contenido distinto: trazos, imagen y texto. La primera de ellas se encarga de registrar los trazos realizados con el dedo y renderizar las trazas. La segunda tiene como responsabilidad registrar los gestures sobre la imagen y delegarlos al panel B. Por último, la capa de texto detecta click sobre la pantalla para luego invocar al

teclado digital del dispositivo y que el usuario pueda introducir información por esta vía.

- **Panel E – Barra de herramientas**: Se desarrolló e implemento una barra de herramientas que consta de una serie de componentes que le permite al usuario capturar una imagen y asociarla a la página, iniciar una grabación de audio, definir el tipo de traza a utilizar definiendo color, ancho y tipo de fuente (ver Figura 12). La barra de herramienta consta de los siguientes componentes:

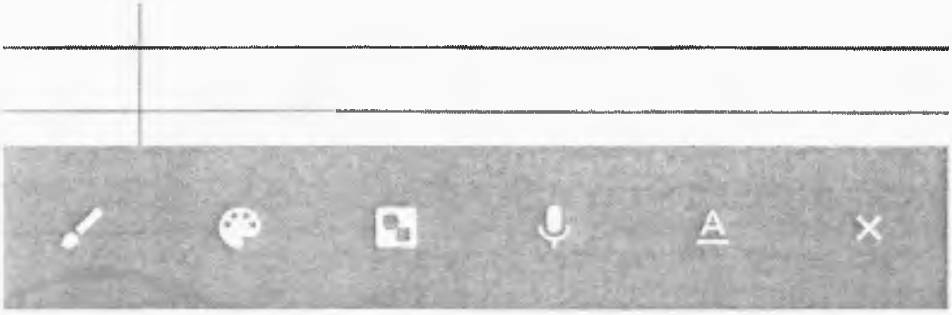

Figura 12. Barra de herramientas de cuaderno
Fuente: Square Corner Blog

  - **Control de pincel**

    El usuario podrá definir cuál tipo de traza utilizará mediante este control. Una vez que el mismo haga click en ícono, se muestra un diálogo donde se puede seleccionar la traza y configurar el grosor de la misma (ver Figura 13). Entre las trazas disponibles se encuentran el estilo libre, borrador, círculo, cuadrado, línea recta.

Adicionalmente se ofrece una vista interactiva que muestra la configuración seleccionada, con el fin de evitar el engorroso proceso de "prueba-y-error" buscando la configuración deseada.

- **Control de color**

Adicionalmente, el usuario cuenta con la posibilidad de ajustar el color a detalle; tono, contraste y opacidad (ver Figura 14). La configuración de color no está relacionada al tipo de traza o texto que se esté usando, es decir, el usuario puede cambiar entre los distintos tipos de trazas preservando la configuración del color.

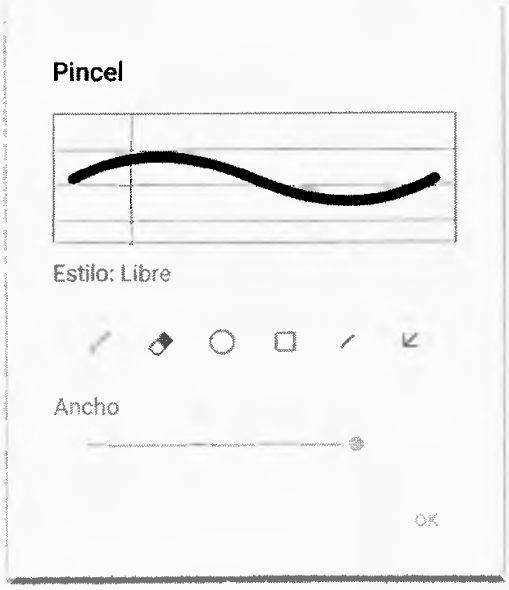

Figura 13. Control de pincel
Fuente: Elaboración propia

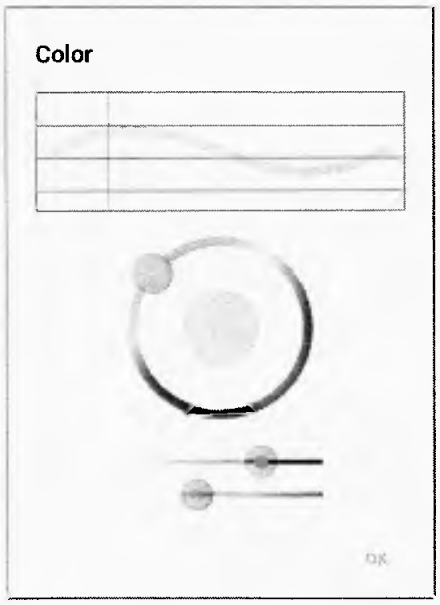

Figura 14. Control de color
Fuente: Elaboración propia

- **Control de imagen**

Al hacer click en el ícono, se muestra un diálogo donde el usuario deberá indicar la fuente de la imagen a insertar en la página.

Entre las fuentes posibles se encuentran: tomar una foto usando la cámara del dispositivo, importar una imagen existente en el dispositivo utilizando la aplicación nativa de galería del mismo e seleccionar una imagen previamente importada por UCABdroide (ver Apéndice A.1 Figura 4).

Cabe señalar que al capturar una imagen utilizando la cámara del dispositivo, la misma queda almacenada dentro del sistema de archivos de UCABdroide implementado por el módulo de galería.

- **Control de audio**

Este control posee la misma funcionalidad implementada en la pantalla previa, donde se listan los cuadernos existentes para una materia. Durante las reuniones se decidió implementar este acceso rápido en la barra de herramientas para evitar que el usuario tenga que cerrar el cuaderno para iniciar una grabación de audio. Vale recordar que el proceso de representar el contenido de

una página es costoso, y este acceso rápido le ahorra segundos al estudiante.

- **Control de texto**

Por último, se encuentra en control de texto de la barra de herramientas. Este control es similar al control de pincel, solo que, en lugar de elegir la traza a utilizar y configurar su ancho, este diálogo le permite configurar el tipo de fuente de texto y tamaño. Como limitante, los tipos de fuentes deben ser incluidos en el APK de UCABdroide y solo contará por el momento con 3 tipos distintos.

La barra de herramienta cuenta con la funcionalidad además de colapsar en un botón flotante con el fin de maximizar la visibilidad del contenido renderizado en el panel D. En la Figura 12 se parecía la barra expandida mientras que en la Figura 15 se presenta colapsada.

Figura 15. Barra de herramientas de cuaderno colapsada
Fuente: Elaboración propia

IV.3 Sprint 2

IV.3.1 Diseño de módulo de galería de archivos

Durante este sprint se llevó a cabo una serie de reuniones con el equipo de trabajo donde se discutió el diseño del módulo de galería de archivos y los componentes que lo conforman.

Este módulo está enfocado en brindar al estudiante una serie de controles que le ayude con la organización de contenido multimedia. Es por ello que se decidió agrupar los mismos por materia, ya que a la hora de estudiar podrá encontrar todo el material relacionado en un mismo lugar. Para lograrlo, se utilizó la pantalla principal de UCABdroide, la cual muestra la lista de materias que el estudiante está cursando actualmente, y se agregó al menú de cada una la opción “Ir a galería” donde se mostrarán los archivos asociados a la asignatura.

Android tienen dos áreas de almacenamiento de archivos: almacenamiento "interno" y "externo". Se discutió cual tipo de almacenamiento a utilizar comparando ventajas y desventajas de cada uno. El almacenamiento “interno” siempre estará disponible pero los archivos solo podrán ser accesibles por la aplicación (ver II.3.3 Fundamentos de una aplicación Android) y si UCABdroide es desinstalado, estos archivos serán borrados. Por otro lado, el almacenamiento “externo” puede que no esté disponible, pero no se borrarán los archivos cuando se desinstale la aplicación ya que este tipo de memoria es compartida y puede ser accesible por otras aplicaciones. Se decidió utilizar la

memoria "externa" del dispositivo con el fin de que el estudiante conserve su material incluso si decide desinstalar la aplicación.

Con el fin de mantener los archivos lo más organizado posible y poder clasificar los archivos asociados por materia, se diseñó e implementó una estructura de carpetas en el almacenamiento externo. Esta estructura como primer nivel una carpeta llamada "UCABdroide", seguidamente en su interior estarán las carpetas con el código de periodo cursado, por último, las carpetas con el nombre de la materia cursada por el estudiante dentro del período correspondiente, donde estarán todo el contenido relacionado a la asignatura (ver la Figura 16).

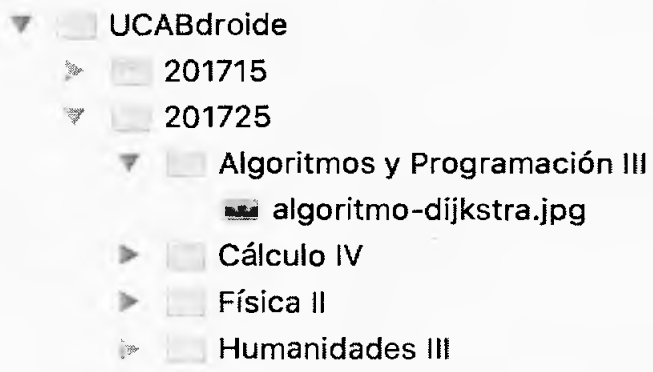

Figura 16. Sistema de archivos UCABdroide
Fuente: Elaboración propia

IV.3.2 Desarrollo de control de visualización de archivos

Mediante este control el estudiante podrá visualizar el material que ha asociado. Para acceder al mismo, debe hacer click en "Ir a galería" ubicado en el

menú de la lista de materias de UCABdroide. Está conformado por dos pantallas: la primera muestra una lista todos los archivos para la materia, que al hacer *click* en un ítem, lleva a la segunda pantalla donde se visualiza en pantalla completa el elemento seleccionado.

- Pantalla de lista de archivos

  Por medio de la misma el usuario podrá visualizar todos los archivos asociados a la materia. La misma hace uso del sistema de carpetas diseñado para este módulo para determinar cuáles son dichos archivos.

  Por cada ítem de la lista, se muestra el *thumbnail* de la imagen en caso de serlo, o un ícono indicando el tipo de archivo para el resto. Además, se muestra el nombre y la fecha de creación (ver Apéndice A.2 Figura 1). Esta lista está ordenada mostrando ítems más recientes de primero de manera que el usuario tenga un orden cronológico de la materia.

  Cada ítem, tendrá un conjunto de acciones asociado a la manipulación del mismo. Entre este conjunto, se podrá compartir, renombrar, borrar o ver información correspondiente del archivo (ver Apéndice A.2 Figura 2). Adicionalmente, se permite que usuario haga selección múltiple de ítems para realizar dichas operaciones por *batch*.

- Pantalla de vista de archivos

  Cada ítem de la lista desarrollada anteriormente, podrá ser visualizada en una nueva pantalla en tamaño completo al hacer click (ver Apéndice A.2 Figura 3). Esta pantalla permite acciones de *swipe* o deslizamiento para cambiar el archivo a visualizar sin la necesidad de navegar a la pantalla anterior para seleccionar un nuevo ítem (ver Apéndice A.2 Figura 4). Además, permite hacer zoom a imágenes con *gestures* de *pinch-in* para alejamiento, y *pinch-out* para acercamiento de la misma.

IV.3.3 Desarrollo de control de asociación de archivos

Se desarrolló e implementó un control que se encarga de asociar contenido multimedia y archivos, independientemente de su tipo o extensión, a la materia deseada. Para logar este control, se implementó las siguientes funcionalidades:

- Por materia: En la pantalla de lista de archivos por materia, se agregó un botón a la barra de herramientas de la misma, donde al hacer click, se muestra un diálogo preguntando al usuario la fuente de proveniencia del archivo o imagen: usando cámara del dispositivo, seleccionando una imagen a través de la aplicación nativa de galería de Android o por medio del explorador de archivos nativo del sistema operativo (ver Apéndice A.2 Figura 5). Una vez indicado, el control se encarga de copiar el archivo en

el sistema de carpetas UCABdroide, se manera que quede asociado a la asignatura.

- Por clase: El estudiante desde la pantalla de cuaderno, podrá asociar imágenes a una clase. Similar al punto anterior, el usuario podrá escoger el archivo o usar la cámara del dispositivo y este se copia en el sistema de carpetas de UCABdroide.

IV.3.4 Desarrollo de widget para captura de foto o nota de voz

Como complemento a este módulo, se desarrolló widgets de escritorio de acceso rápido a funcionalidades existentes implementadas en módulos anteriores de este trabajo instrumental de grado, como lo son la captura de foto y nota de voz (ver Apéndice A.2 Figura 6). Para llevar acabó este requerimiento, se decidió implementar dos widgets distintos:

- Widget de captura de foto: Una vez que el usuario haga click en este control, inmediatamente se accede a la cámara del dispositivo para realizar la captura de una foto. Al confirmar la captura, el widget re-direcciona al estudiante a la aplicación UCABdroide, donde por medio de un diálogo de elección de materia, es asociada a la misma.

- Widget de captura de audio: Al hacer click en este control, se ejecuta el servicio de grabación de notas de voz. El usuario notará una notificación persistente de sistema indicando el estado de grabación. Al detener la

grabación, se ejecuta el procedimiento de asociación a una materia también utilizado por el widget de captura de foto.

Ambos widgets reciben el contenido multimedia como una ruta al archivo temporal utilizando receptores de mensajes descritos en II.3.3.1 Componentes de una aplicación Android, para posteriormente ser copiados al sistema de carpetas de UCABdroide.

IV.3.5 Diseño e implementación de estructura de etiquetas

Por último, luego de reuniones con el equipo de trabajo, se implementó una estructura de etiquetas y palabras claves. Con esta funcionalidad se busca permitir clasificar archivos de contexto similar. Por ejemplo; guías, cuestionarios, ejercicios resueltos o láminas de una o varias sesiones de clase de cálculo, podrán ser etiquetados con la palabra "métodos de integración". A la hora del que el estudiante requiera estudiar o repasar todo el contenido de estas sesiones, cuenta con la opción de buscar por dicha etiqueta y visualizar todo el este material relacionado. Este control consta de dos elementos:

- Asignación de etiqueta: En la pantalla de lista de archivos de asociados a una materia, se implementó un icono de etiqueta en la barra de herramientas de la misma, que se activará cuando el usuario seleccione uno o varios archivos. Esta selección representa a los archivos que serán etiquetados, luego de que el usuario introduzca la palabra clave a través de un diálogo.

- Consulta de etiqueta: Para consultar contenido multimedia y archivos por palabra clave, se implementó un botón de lupa en la barra de herramientas de la pantalla principal de UCABdroide, donde podrá introducir por medio de teclado de dispositivo, dicha etiqueta y posteriormente, en una nueva pantalla, visualizar el material.

IV.4 Sprint 3

IV.4.1 Diseño de estructura de archivo de cuadernos

El módulo de sincronización y respaldo, que es el objetivo y entregable de este sprint, se basa en dar funcionalidades que complementen el módulo de gestión de notas y apuntes: exportación e importación, así como sincronización y respaldo de cuadernos.

Para lograr los objetivos, fue necesario diseñar e implementar, un proceso que, de alguna forma, permita el traspaso de la información contenida en base de datos, a un archivo portátil. Después de sostener reuniones con el equipo de trabajo, se llegó al acuerdo de serializar dicha información utilizando el formato ligero de intercambio de datos JSON en un archivo de formato ".ucab".

Un cuaderno, entre otros elementos, puede estar compuesto de archivos binarios (como imágenes y audio). El mayor retro que se presentó, una vez tomada la decisión, fue como serializarla y representar dichos binarios en un archivo JSON. Usando un esquema de codificación de binario a texto similares que representan datos binarios en un formato de cadena ASCII, fue posible incluir

los mismos en el archivo ".ucab". En la Figura 17 se muestra una fracción de un cuaderno serializado JSON donde se puede apreciar un binario representado en una cadena de caracteres. Como resultado, se desarrolló un algoritmo para la serialización y deserialización entre cuadernos almacenados en base de datos y archivos ".ucab".

cuaderno-201725-emprendimiento-12-07-2017-04-11-18-936.ucab

```
{
  "author_name": "Díaz Pérez, Pedro Emilio",
  "author_email": "pediaz.11@est.ucab.edu.ve",
  "class_nrc": "29024",
  "class_term": "201725",
  "class_name": "Emprendimiento",
  "note_id": 2,
  "note_name": "Test",
  "note_description": "",
  "note_created": 1495898113825,
  "note_modified": 1499777482950,
  "note_pages": [
    {
      "page_created": 1495898113885,
      "page_image": {
        "image_base64":
        "iVBORw0KGgoAAAANSUhEUgAAAeAAAAHgCAIAAADytinCA
        5jTReES46/uXeZy8ZTfuPrZtL1u2Olu2rNnbz/7v/+v/PB
```


Figura 17. Archivo ".ucab"
Fuente: Elaboración propia

IV.4.2 Desarrollo de control de exportación e importación de cuadernos

Mediante este control, los estudiantes cuentan con la posibilidad de compartir sus notas y apuntes, incluyendo el contenido multimedia asociado, entre ellos.

La funcionalidad de exportar un cuaderno se implementó en la pantalla de cuadernos de la materia. En ella, se implementó en el menú contextual de cada cuaderno, la opción "Exportar" (ver Apéndice A.3 Figura 1). Al hacer click, se inicia el servicio de exportación (un servicio es un proceso independiente del estado la aplicación) que se ejecuta fuera del hilo principal para evitar que la aplicación se haga no responsiva. El proceso se encarga de extraer el contenido relacionado al cuaderno en cuestión y serializarlo en un archivo ".ucab" que poster a su creación, es asociado como archivo de la materia del cuaderno. Al finalizar, el usuario es notificado del resultado del proceso.

Otra funcionalidad que brinda este control es la capacidad de importar un cuaderno previamente exportado. La aplicación, mediante el archivo de manifiesto de Android, notifica al sistema que ésta es la responsable de manejar archivos ".ucab". Cuando un usuario intente abrir un archivo con este formato, se le delega la acción a UCABdroide, y éste seguidamente al control para iniciar el proceso de importación (ver Apéndice A.3 Figura 2). El proceso utiliza los mismos componentes que el control de exportación de un cuaderno; se levanta un servicio en un *background thread* donde se deserializa los cuadernos contenido en el archivo para que finalmente sean almacenado de base de datos.

IV.4.3 Desarrollo de control de sincronización y respaldo de cuadernos

Con esta funcionalidad se le brinda al estudiante la opción de guardar y recuperar sus cuadernos y apuntes realizados por medio de servicio de almacenamiento en la nube. Esto además trae como ventaja para el usuario, la

facilidad de poder mover sus cuadernos entre los dispositivos Android que posea. Además, contará con la opción de realizar respaldos automáticamente durante intervalos de tiempo fijos.

Una de las limitantes que surgió durante el diseño de este control, es que el usuario sólo podrá utilizar el servicio de almacenamiento Google Drive, aunque la DTI planea dar soporte a futuro a servicios como Dropbox y Box. Adicionalmente, se acordó que los respaldos automáticos se realicen con conexión Wifi y mientras se esté cargando el dispositivo, con el fin de evitar consumos indeseados de batería y datos móviles. Y como última limitante, solo se manejará un respaldo a la vez, es decir, no habrá un historial de todas las sincronizaciones realizadas donde el usuario podrá decidir cuál usar, sino que una sincronización sobre-escribe a la anterior.

Antes de que el usuario pueda respaldar y sincronizar sus apuntes, es necesario indicar la cuenta Google a utilizar y la frecuencia con la que se realizaran los respaldos automáticos (ver Apéndice A.3 Figura 3). Para ello, se implementó una pantalla la cuál es accesible mediante la opción "Respaldo" menú principal, la cual está compuesta por dos paneles:

- Panel informativo: Esta sección muestra al usuario información de la sincronización realizada como tamaño y tipo de servicio utilizado (para el momento de realización de trabajo de grado, siempre será Google Drive). Y, por último, se muestran los siguientes botones:

- Respaldar: Al hacer click en este botón, se inicia el proceso de exportación desarrollado en el control anterior. Una vez que éste genere el archivo ".ucab" con todos los cuadernos serializados, se procede a utilizar el API de la librería de sincronización y respaldo con servicios de almacenamiento en la nube implementada previamente.

- Restaurar: Por medio de este botón, se inicia el proceso de recuperación de la última sincronización realizada en Google Drive mediante la librería de sincronización y respaldo. Se inicia un servicio en un *background thread* donde se descargar el archivo ".ucab" de Google Drive y una vez recuperado, se le delega la acción de restablecer el archivo de respaldo al control de importación desarrollado para este módulo.

- Panel de configuración: Por medio de este panel, el usuario configura la cuenta Google con la cuál desee almacenar los respaldos. Por mecanismos de seguridad de Google, el estudiante deberá autorizar a UCABdroide para el acceso a su cuenta Google Drive y este proceso es realizado por Android.

  Adicionalmente, el estudiante podrá escoger la frecuencia con la que se realizaran los respaldos automáticos: "diario", "semanal" o "mensual". También se brinda la opción de deshabilitar las mismas al seleccionar "nunca".

IV.4.4 Desarrollo de control de exportación de cuaderno como archivo PDF

Como último control que conforma este módulo, se desarrolló e implemento un control que permite convertir uno o varios cuadernos de clase a un archivo PDF.

En la pantalla de la lista de cuadernos por materia, podrá contar con la opción "Exportar como PDF" en el menú contextualizado de cada cuaderno (si se requiere exportar solo éste) o por medio de la barra de herramientas (ver Apéndice A.3 Figura 4) de dicha pantalla (de querer exportar todos los cuadernos). Este control carga el cuaderno y todos los objetos relacionados al mismo, y representarlos en un canvas de memoria. El algoritmo que convierte un canvas a un documento PDF está implementada en el SDK de Android y se decidió utilizarlo.

IV.5 Sprint 4

IV.5.1 Integrar UCABdroide las aplicaciones instaladas en el dispositivo

A través de esta funcionalidad, el usuario podrá compartir archivos de otras aplicaciones con UCABdroide mediante el uso del menú compartir del sistema operativo Android (ver Apéndice A.4 Figura 1), donde el estudiante podrá asociarlo a una materia y después a cualquier clase o a los documentos comunes de dicha materia. Para la integración de UCABdroide con el resto de las aplicaciones instaladas en el dispositivo, fue necesario definir en el manifiesto que esta aplicación puede recibir uno o varios archivos, sin importar el tipo,

mediante el componente intent-filter. Una vez que Android inicie UCABdroide indicando que se está compartiendo un archivo, se ejecuta el control de asociación de archivos implementado previamente en el módulo de gestión de galería (ver Apéndice A.4 Figura 2).

IV.5.2 Desarrollo de control para compartir desde la galería del cuaderno con estudiantes de un curso

Este módulo cuenta con una funcionalidad de permitir compartir archivos asociados a una materia con uno o varios estudiantes del curso. Una vez que se haya compartido el archivo, los otros estudiantes del curso recibirán una notificación de UCABdroide indicando que se le ha compartido un archivo, imagen o nota de voz y le saldrá la opción de descarga.

Como limitante de este control, se utiliza el servicio de almacenamiento Google Drive para subir y descargar los archivos. El estudiante debe configurar en la aplicación la cuenta Google a usar para poder brindarle esta funcionalidad.

Para lograr acabo este control se solicita al servicio web de registro académico de la UCAB, la lista de los estudiantes que conforman el curso con sus datos e identificador de GCM asignado para que, una vez subido el archivo seleccionado a Google Drive y teniendo el link público del mismo para la descarga, se le indique a GCM que debe mandar una notificación que incluya ese link, a los usuarios registrados con los identificadores seleccionados. Una vez que se recibe una notificación de este tipo, se descarga e importa el archivo compartido utilizando controles previamente desarrollados.

CAPÍTULO V – RESULTADOS

Desarrollando las iteraciones fijadas por la metodología SCRUM y garantizando que la aplicación funcione correctamente, se logró cumplir con cada uno de los objetivos planteados.

A continuación, se detallarán los resultados considerando los objetivos específicos propuestos para este Trabajo Instrumental de Grado.

V.1 Resultados según Objetivos

- **Desarrollar módulo de gestión de notas y apuntes**

  Luego de revisar y rediseñar el proceso de nota de apuntes de clase en reuniones con estudiantes que forman parte del DTI, se obtuvo como resultado el módulo de gestión de notas y apuntes. El usuario (en este caso el estudiante de la universidad) podrá realizar y organizar notas de clase desde la aplicación UCABdroide y asociar material multimedia a ella. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 1. (Ver Apéndice A1).

- **Desarrollar módulo de galería de archivos**

  Luego de reuniones con el equipo de trabajo se identificó la necesidad de un módulo que complemente al de notas y apuntes. Como resultado se desarrolló el módulo de galería, donde por medio de una pantalla, el usuario podrá visualizar el material capturado durante una

clase agrupado por materia, además de poder agregar y eliminar fotos y otros archivos a la misma sin necesidad de crear un cuaderno. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 2. (Ver Apéndice A2).

- **Desarrollar módulo de respaldo y sincronización**

De manera de poder brindarle al estudiante un mecanismo para guardar y recuperar cuadernos de clase con material multimedia asociado, se desarrolló el módulo de respaldo y sincronización. El estudiante podrá exportar e importar dichos cuadernos al almacenamiento local del dispositivo y a través del servicio de almacenamiento en la nube Google Drive. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 3. (Ver Apéndice A3).

- **Desarrollar módulo para compartir contenido**

Por último, fue necesario poder ofrecer al estudiante la opción de compartir sus cuadernos, imágenes, audio y otros archivos relacionados a una materia con otros. Para ello se desarrolló el módulo de compartir contenido, mediante el cual el usuario puede exportar e importar contenido con otras aplicaciones disponibles en el sistema y, además, compartirlo directamente con otros estudiantes sin depender de otras aplicaciones. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 4. (Ver Apéndice A4).

- **Desarrollar una librería que se encargue de facilitar la sincronización y respaldo con servicios de almacenamiento en la nube**

Después de realizar un análisis durante reuniones de con el equipo de trabajo, se detectó la posibilidad de que la DTI pueda necesitar para otros proyectos Android, la funcionalidad de sincronizar y respaldar archivos con Google Drive. Como resultado de desarrolló una librería independiente a UCABdroide donde se implementa y unifica interfaces de servicios de almacenamiento en la nube Google Drive, Dropbox y Box. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 0.

V.2 Aporte Tecnológico

- **Integrar con otras aplicaciones instaladas en el dispositivo para compartir contenido con las mismas**

Se diseñó e implementó un algoritmo en la aplicación móvil que permite la comunicación con otras aplicaciones instaladas en el dispositivo para compartir archivos. La comunicación es posible gracias al archivo manifiesto de Android de la aplicación. Este resultado se detalla en el capítulo IV en desarrollo del Sprint 4.

V.3 Aporte Funcional

Para este punto se tomó en cuenta cómo se lleva a cabo el proceso actual de la nota de apuntes de clase con apoyo de dispositivos móviles por los estudiantes, este proceso se detalla en el capítulo IV en el

desarrollo del Sprint 1. Buscando mejorar el proceso, se diseñó la sección de cuaderno digital y galería en la aplicación con el fin de ayudar al estudiante con la organización del material capturado durante clase con dispositivos móviles.

CAPÍTULO VI – CONCLUSIONES Y RECOMENDACIONES

VI.1 Conclusiones

A lo largo de todo el desarrollo de este Trabajo Instrumental de Grado y el conocimiento del proceso del estudiante de la Universidad Católica Andrés Bello para realizar notas y apuntes de clase mediante el uso de dispositivos móviles, tomando en cuenta cada uno de los resultados obtenidos se desglosan una serie de conclusiones:

- Como consecuencia del uso de la metodología SCRUM, se manipularon los cambios durante el transcurso del desarrollo de forma que pudiera ser tomado en cuenta, ya que en cada Scrum diario se discuten mejoras que lograron de la aplicación un producto final de calidad. Tal es el caso del cambio en donde se decidió migrar UCABdroide hacia otra plataforma de desarrollo que permitió desenvolver la solución en un ambiente favorable, ya que Android Studio es el IDE oficial y recomendado por Android Developers y es el número uno en manejo de aplicaciones en Android permitiendo estructurar y organizar el proyecto de una manera fácil.

- Con el uso del cuaderno digital por parte de los estudiantes para realizar la captura de contenido dictado en clase, se estima que se reduce considerablemente el tiempo que se necesita para la organización del contenido.

- La aplicación desarrollada en este Trabajo Instrumental de Grado permite el aprendizaje colaborativo entre estudiantes. Compartir apuntes entre los alumnos a través de un cuaderno digital genera ventajas en cuanto a la discusión y complemento de contenidos. Adicionalmente, reduce el riesgo de pérdida de apuntes para el que los presta y reduce gastos de reproducción al que los recibe tanto en tiempo como en costo.

- Sincronizar cuadernos con Google Drive reduce la posibilidad de perder notas y apuntes realizados en caso de que el dispositivo sea hurtado o extraviado.

- El uso de aplicaciones que permiten hacer bosquejos de interfaces de usuario agilizó el proceso de creación de pantallas durante el desarrollo de la aplicación.

- La implementación de los patrones MVP, inyección de dependencias, repositorio y observador, facilitó la organización del código durante el desarrollo del proyecto, brindando facilidad en el mantenimiento y escalabilidad.

- El uso de librerías para componer programas asincrónicos y basados en eventos mediante el uso del patrón de observables, como RxJava, facilitan al desarrollador la manipulación y manejo de información.

VI.2 Recomendaciones

- Debido a que no todos los estudiantes de la universidad poseen dispositivos Android, se recomienda realizar el desarrollo de aplicaciones móviles para otros sistemas operativos como iOS y Windows Phone.

- Se recomienda implementar un control que permita visualizar los cuadernos digitales realizados para otras materias y/o períodos.

- Dar soporte en el cuaderno digital a más de una imagen y cuadro de texto por página. De esta forma el estudiante no se verá obligado a crear una página nueva de necesitar insertar dichos elementos.

- Se recomienda al DTI la implementación de un backend donde se maneje el almacenamiento y la distribución de archivos a compartir directamente con otros estudiantes, de esta forma se puede controlar y sancionar a aquellos que envíen material indebido a otros usuarios.

- Cuando un compañero de clase comparte contenido digital utilizando la aplicación, los usuarios destino recibirán una notificación de este evento. Se recomienda poder brindar la opción al usuario de bloquear estas notificaciones entrantes, donde el mismo decida cuales personas podrán enviarle contenido.

- Se recomienda ofrecer servicios de almacenamiento en la nube como Dropbox y Box, donde el usuario tenga otras alternativas a Google Drive donde hacer respaldos en el módulo de sincronización y respaldo.

BIBLIOGRAFÍA

Referencias electrónicas

- [1] TopUniversities (2017). QS World University Rankings 2015/2016 [En línea]. Disponible en: https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2016

- [2] Wikipedia (2017). Aplicaciones Móvil [En línea]. Disponible en: https://es.wikipedia.org/wiki/Aplicacion_movil

- [3] Wikipedia (2017). Android [En línea]. Disponible en: https://en.wikipedia.org/wiki/Android_(operating_system)

- [4] Wikipedia (2017). Lista de características en Android [En línea]. Disponible en: https://en.wikipedia.org/wiki/List_of_features_in_Android

- [5] Google (2017). Arquitectura de la plataforma [En línea]. Disponible en: https://developer.android.com/guide/platform/index.html

- [6] Google (2017). Android Studio [En línea]. Disponible en: https://developer.android.com/studio/index.html

- [7] Olanoff, Drew (2013). Google launches Android Studio and new features for developer console, including beta releases and staged rollout [En línea]. Disponible en: https://techcrunch.com/2013/05/15/google-launches-android-studio-a-development-tool-for-apps

- [8] Google Playstore (2017). UCABdroide [En línea]. Disponible en: https://play.google.com/store/apps/details?id=ve.edu.ucab.proyectoucab&hl=en

- [14] Oracle (2001). ¿Que es Java? [En línea]. Disponible en: https://www.java.com/es/download/faq/whatis_java.xml

- [15] Vass Digital (2012). Manual de referencia [En línea]. Disponible en: http://www.vassdigital.com/scrum-la-metodologia-de-desarrollo-agil-por-excelencia

- [16] Proyectos ágiles (2014). ¿Qué es Scrum? [En línea]. Disponible en: http://proyectosagiles.org/que-es-scrum/

- [17] Softteng (2015). Proceso y Roles de Scrum. [En línea]. Disponible en: https://www.softeng.es/es-es/empresa/metodologias-de- trabajo/metodologia-scrum/proceso-roles-de-scrum.html

- [18] Ken Schwaber y Jeff Sutherland (2013). La Guía de Scrum [En línea]. Disponible en: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf

- [19] Scrum Manager (2014). Gestión de proyectos Scrum Manager [En línea]. Disponible en: http://www.scrummanager.net/files/sm_proyecto.pdf

Referencias de libros

- [20] Jesús Tomás Girones. (2014). El Gran Libro de Android. (pp. 9 - 42). USA: ELSEVIER Digital Press.

APÉNDICE

Apéndice A: Figuras

A1. Módulo de gestión de notas y apuntes

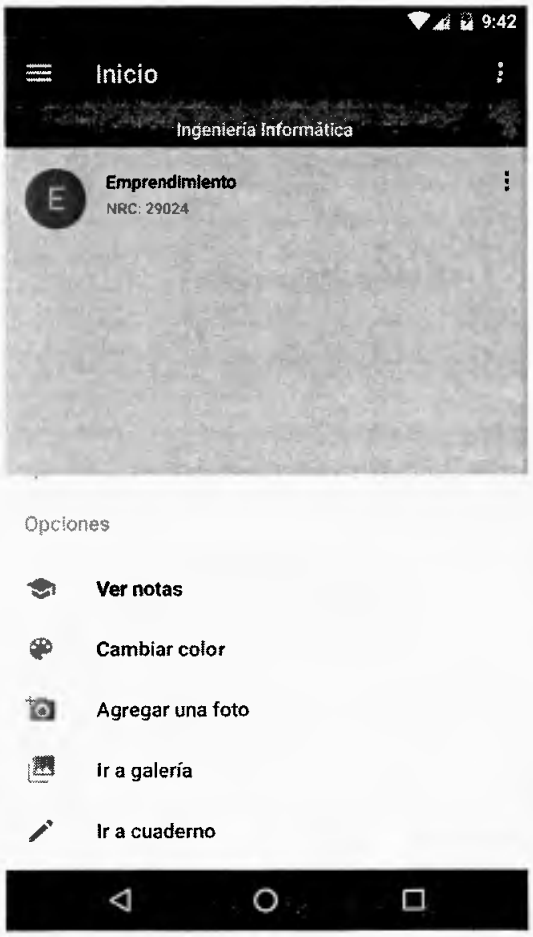

Figura A1 - 1
Ir a cuaderno
Fuente: Elaboración propia

Figura A1 - 2
Lista de cuadernos por materia
Fuente: Elaboración propia

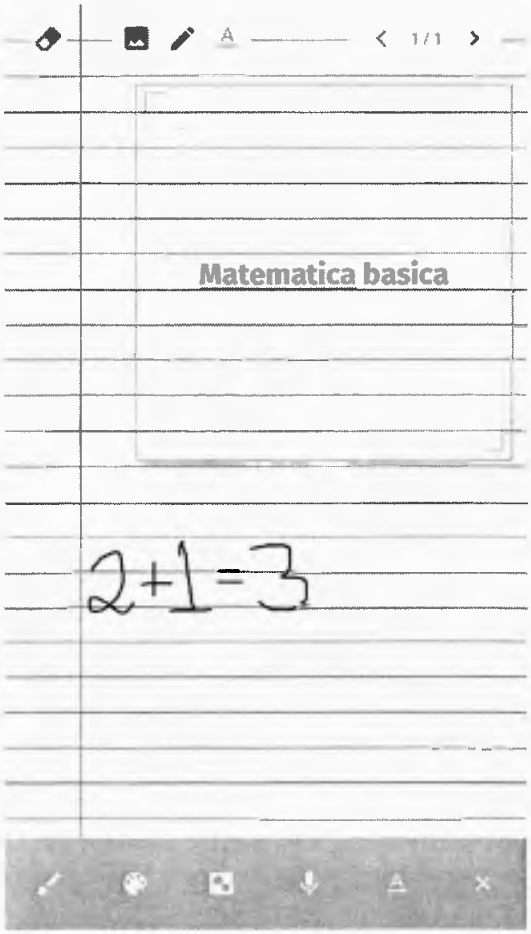

Figura A1 - 3
Canvas con trazos a mano alzada y texto
Fuente: Elaboración propia

Figura A1 - 4
Selección de fuente de imagen a importar
Fuente: Elaboración propia

A2. Módulo de galería de archivos

Figura A2 - 1
Lista de archivos por materia
Fuente: Elaboración propia

Figura A2 - 2
Acciones de archivos
Fuente: Elaboración propia

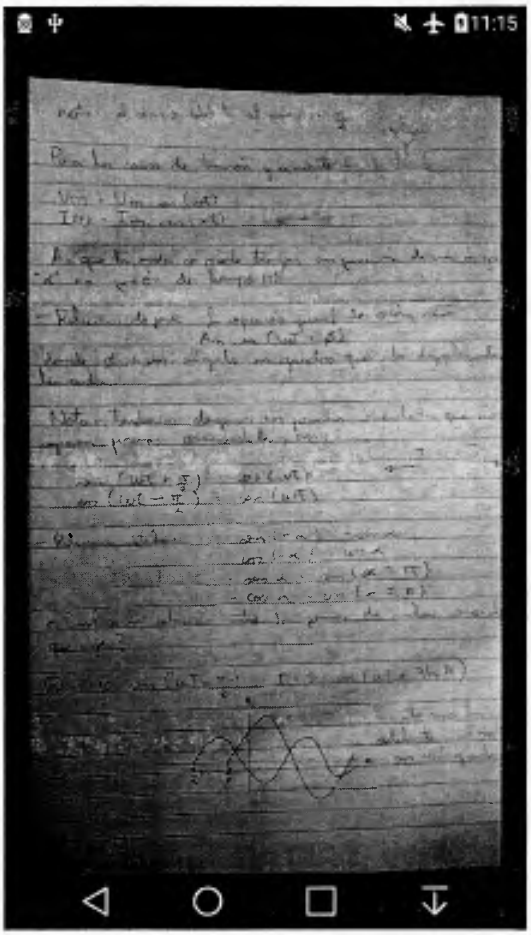

Figura A2 - 3
Vista pantalla completa de imagen
Fuente: Elaboración propia

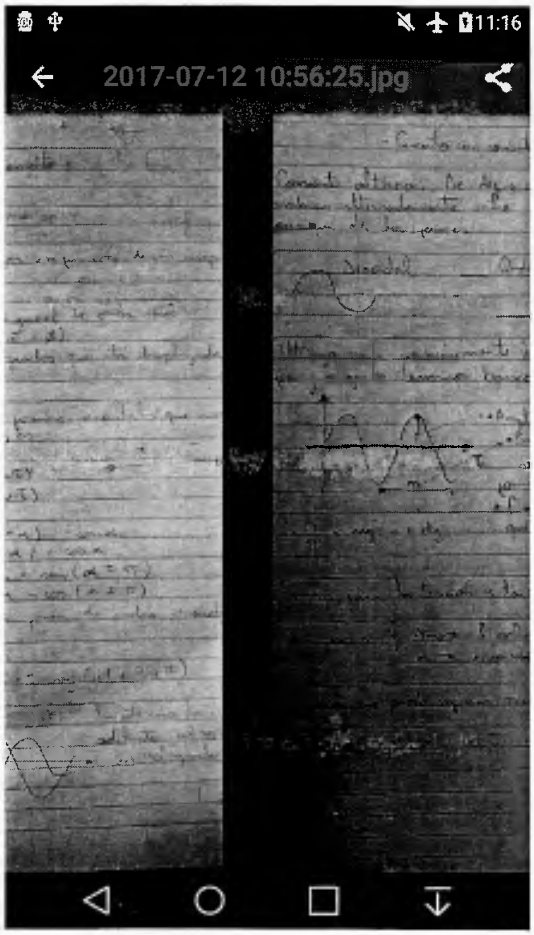

Figura A2 - 4
Swipe entre imágenes
Fuente: Elaboración propia

Figura A2 - 5
Asociación de archivos por materia
Fuente: Elaboración propia

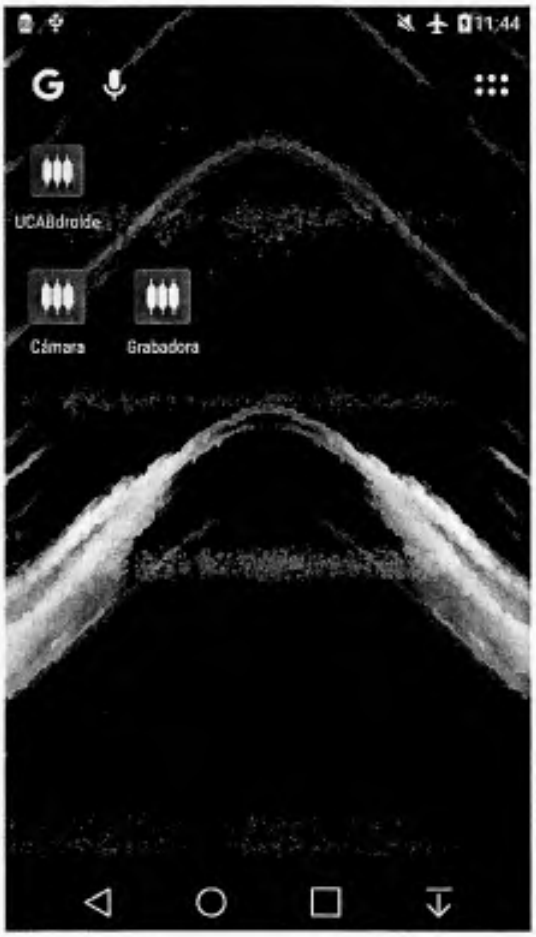

Figura A2 - 6
App de UCABdroide con widgets
Fuente: Elaboración propia

A3. Módulo de respaldo y sincronización

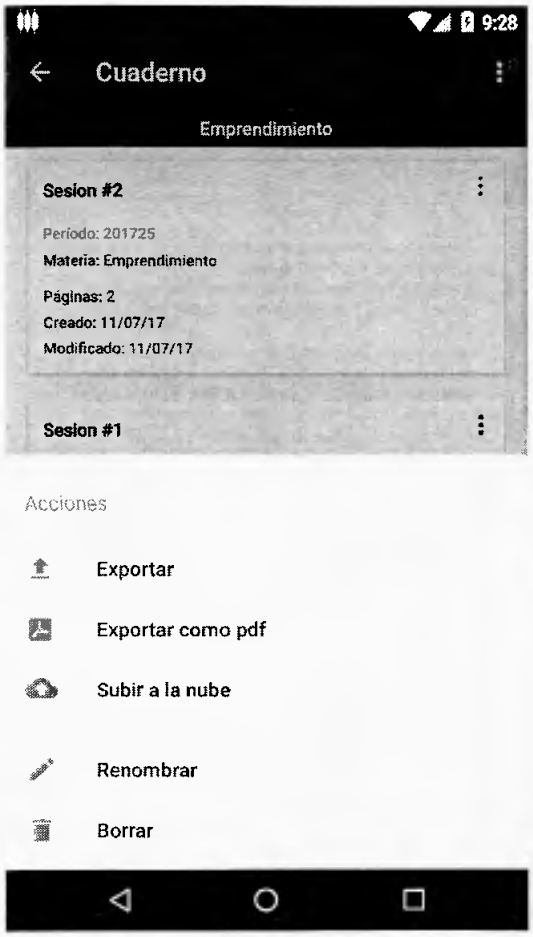

Figura A3 - 1
Exportación de un cuaderno
Fuente: Elaboración propia

Figura A3 - 2
Importación de un cuaderno
Fuente: Elaboración propia

Figura A3 - 3
Sincronización y respaldo con Google Drive
Fuente: Elaboración propia

Figura A3 - 4
Exportar como PDF
Fuente: Elaboración propia

A4. Módulo para compartir contenido

Figura A4 - 1
Compartir un archivo con UCABdroide desde otra aplicación
Fuente: Elaboración propia

Figura A4 - 2
UCABdroide realizando importación de un archivo compartido
Fuente: Elaboración propia

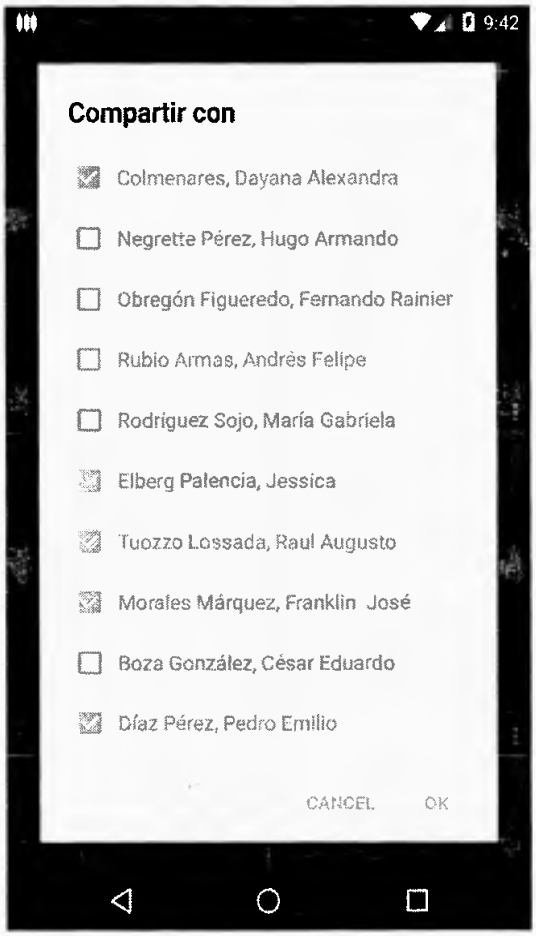

Figura A4 - 3
Compartir un archivo con otros compañeros de clase
Fuente: Elaboración propia