



UNIVERSIDAD POLITÉCNICA DE PUEBLA

INGENIERIA EN INFORMÁTICA

PROYECTO DE ESTADIA PROFESIONAL

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA ADMINISTRACIÓN, ASIGNACIÓN DE
LABORATORIOS Y MATERIALES DE ÓPTICA**

ARROYO CORONA LUIS

ASESOR TÉCNICO

DR. FRANCISCO JAVIER RENERO CARRILLO

ASESOR ACADÉMICO

MC. REBECA RODRIGUEZ HUESCA

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS	iii
ÍNDICE DE TABLAS.....	iv
INTRODUCCIÓN	1
CAPITULO I.....	2
ANTECEDENTES DEL INSTITUTO NACIONAL DE ASTROFÍSICA ÓPTICA Y ELECTRÓNICA... 2	
1.1 ANTECEDENTES	2
1.2 Misión.....	3
1.3 Visión	3
1.4 Facultades.....	3
1.5 Coordinación de óptica.....	5
1.6 Misión de la coordinación	5
1.7 Descripción general del proyecto	6
1.7.1 Planteamiento del problema	6
1.7.2 Justificación.....	7
1.7.3 Objetivo general	7
1.7.4 Objetivos específicos.....	7
1.7.5 Alcances y limitaciones.....	7
CAPITULO II	9
METODOLOGÍA Y HERRAMIENTAS DE DESARROLLO.....	9
2.1 Metodología de desarrollo ágil de software	9
2.1.1 Programación extrema.....	9
2.1.2 Ventajas y desventajas de la programación extrema	10
2.1.2.1 Ventajas.....	10
2.1.2.2 Desventajas.....	10
2.1.2.3 Proceso de desarrollo.....	10
2.2 Herramientas de desarrollo.....	12
2.2.1 Android Studio	12
2.2.2 Sistema Operativo Android	13
2.2.3 SQLite	14
CAPITULO III	15

DESARROLLO DE APLICACIÓN MÓVIL PARA LA ADMINISTRACIÓN, ASIGNACIÓN DE LABORATORIOS Y MATERIALES DE ÓPTICA 15

 3.1 Definición del material 15

 3.2 Proceso de desarrollo..... 15

 3.2.1 Primera Fase: Exploración 15

 3.2.1.1 Historias de usuario 16

 3.2.2 Segunda Fase: Planificación de entrega 19

 3.2.3 Tercera fase: Iteraciones..... 20

 3.2.3.1 Primera Iteración 20

 3.2.3.2 Segunda Iteración 22

 3.2.3.3 Tercera Iteración..... 25

 3.2.3.4 Cuarta Iteración 31

 3.2.3.5 Quinta Iteración 32

 3.2.4 Cuarta fase: Producción..... 36

 3.2.5 Quinta fase: Mantenimiento 37

 3.2.6 Sexta fase: Muerte del proyecto 38

CONCLUSIONES 39

GLOSARIO..... 40

ANEXO 42

BIBLIOGRAFÍA..... 46

ÍNDICE DE FIGURAS

Figura 1.1 Organigrama general del INAOE	4
Figura 1.2 Organigrama de la institución	4
Figura 2.1 Valores de la Programación Extrema	9
Figura 3.1 Diseño de base de datos	20
Figura 3.2 Creación de Base de Datos	21
Figura 3.2 Ejecución de métodos	21
Figura 3.4 Ruta para añadir extensión	22
Figura 3.5 Pantalla de inicio	23
Figura 3.6 Menú Principal	24
Figura 3.7 Barra de navegación	24
Figura 3.8 Menú principal laboratorios	25
Figura 3.9 Alta de laboratorios	26
Figura 3.10 Baja de laboratorios	27
Figura 3.10 Consulta de laboratorios	28
Figura 3.11 Resultado de consulta	28
Figura 3.12 Menú principal de materiales	29
Figura 3.13 Alta de materiales	30
Figura 3.14 Inicio de sesión	31
Figura 3.15 Prueba de inicio de sesión	32
Figura 3.16 Menú para el registro de horarios	33
Figura 3.17 Asignación de laboratorio	34
Figura 3.18 Resultado de consulta y edición	35
Figura 3.19 Ruta para crear APK	35
Figura 3.20 Archivo APK	35
Figura 3.21 Permisos de Instalación	36
Figura 3.22 Aplicación instalada	37

ÍNDICE DE TABLAS

Tabla 1.1 Versiones del sistema operativo 12

Tabla 1.2: Historia de Usuario 1. Diseño de base de datos18

Tabla 1.3: Historia de Usuario 2. Gestión de base de datos.....18

Tabla 1.4: Historia de Usuario 3. Diseño de interfaces de usuario.....18

Tabla 1.5: Historia de Usuario 4. Módulo de laboratorios.....19

Tabla 1.6: Historia de Usuario 4. Módulo de laboratorios.....19

Tabla 1.7: Historia de Usuario 6. Manejo de sesiones.....19

INTRODUCCIÓN

Debido a la incesante necesidad actualizar y agilizar tareas y procesos que día a día van evolucionando, las instituciones, así como las diferentes organizaciones educativas se ven a la necesidad de generar tecnologías que cubran estas necesidades.

Las tecnologías de información y comunicación han ido evolucionando con el paso del tiempo, y es así como las necesidades de obtener software que pueda agilizar tareas y procesos son cada vez más solicitados en los sectores educativos.

Sin embargo, el desarrollo de software que pueda sustentar dichas problemáticas no sólo está sujeta a limitados dispositivos de computo sino también abarca a dispositivos móviles que sean capaces de procesar información.

Actualmente existen dispositivos móviles como teléfonos inteligentes y tabletas con diferentes sistemas operativos que cumplen con tareas específicas.

Los sectores educativos a partir del nivel medio superior y superior, apuestan por el desarrollo de aplicaciones móviles que faciliten la gestión de tareas, de tal modo que pueda ser utilizadas para cubrir dicha tarea.

Sin embargo, aún existen problemas de los cuales no todas las organizaciones educativas prevén o cumplen dentro de los procesos internos que lleven, como la gestión académica o activo fijo de dicha institución.

El presente trabajo aborda una de las problemáticas que se ha generado con el paso del tiempo han afectado a diversas instituciones de índole educativo, asimismo esto conlleva a generar herramientas tecnológicas que ayuden a la administración y control de laboratorios e inventario de estos mismos.

En este trabajo se muestra el desarrollo de una aplicación móvil asignada directamente hacia los laboratorios y materiales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).

CAPITULO I

ANTECEDENTES DEL INSTITUTO NACIONAL DE ASTROFÍSICA ÓPTICA Y ELECTRÓNICA

1.1 ANTECEDENTES

El Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) fue creado por decreto presidencial el 11 de noviembre de 1971 como un organismo descentralizado, de interés público, con personalidad jurídica y patrimonio propio, ubicado en Tonantzintla, Puebla, con los siguientes objetivos:

- Preparar investigadores, profesores especializados, expertos y técnicos en astrofísica, óptica y electrónica.
- Procurar la solución de problemas científicos y tecnológicos relacionados con las citadas disciplinas.
- Orientar sus actividades de investigación y docencia hacia la superación de las condiciones y resolución de los problemas del país.
- Con este decreto el INAOE tiene la facultad de impartir cursos y otorgar grados de maestría y doctorado en las diversas disciplinas que en él se desarrollan.

Luis Enrique Erro fundó el Observatorio Astrofísico Nacional de Tonantzintla. En aquel entonces, Tonantzintla se escogió como el lugar idóneo para la instalación del Observatorio, el cual cumplía con las exigentes normas de calidad como noches despejadas y en cantidad por año

Erro fue sustituido en la dirección del Observatorio por el doctor Guillermo Haro, bajo cuya dirección se convirtió en uno de los centros más importantes de América Latina por la calidad del trabajo científico que en él se llevaba a cabo.

El mismo Haro se dio cuenta de la importancia para el país de la óptica y la electrónica, por lo que en 1971 decidió fundar el INAOE.

En 1972 se fundó el Departamento de Óptica, y dos años después inició sus actividades el Departamento de Electrónica.

Desde su creación uno de los principales objetivos del INAOE ha sido la preparación de investigadores jóvenes, capaces de identificar y resolver problemas científicos y tecnológicos en astrofísica, óptica, electrónica y áreas afines. En 1972 se iniciaron los estudios de maestría en Óptica y en 1974 los de Electrónica.

En 1984 se inició el programa de doctorado en Óptica, y en 1993 los programas de doctorado en Electrónica; así como la maestría y doctorado en Astrofísica.

Finalmente, en agosto de 1998 se inició el programa de maestría y doctorado en Ciencias Computacionales.

1.2 Misión

Contribuir como centro público de investigación a la generación, avance y difusión del conocimiento para el desarrollo del país y de la humanidad, por medio de la identificación y solución de problemas científicos y tecnológicos y de la formación de especialistas en las áreas de Astrofísica, Óptica, Electrónica, Ciencias Computacionales y áreas afines.

1.3 Visión

El INAOE será un centro público de investigación con un alto liderazgo a nivel internacional en el ámbito de la investigación científica, el desarrollo tecnológico y la formación de recursos humanos dentro de las áreas de Astrofísica, Óptica, Electrónica, Ciencias Computacionales y áreas afines, comprometido con el desarrollo nacional a través de la promoción de valores sociales de solidaridad, creatividad y alta competitividad.

1.4 Facultades

A continuación, se mencionan las facultades con las que en base el INAOE trabaja:

- Desarrollar investigaciones e impartir enseñanzas para la consecución de los objetivos previstos.
- Organizar sus planes de investigación y enseñanza.
- Adoptar métodos adecuados para evaluar sus actividades de investigación y enseñanza.
- Conceder grados y otorgar diplomas.

Cada organización está conformada por una jerarquía dónde se establecen los roles o departamentos, a continuación, se muestra el organigrama general:



Figura 1.1 Organigrama general del INAOE

Sin embargo, el organigrama general del Instituto Nacional de Astrofísica, óptica y Electrónica se divide en sub géneros a partir de este que se muestra a continuación:

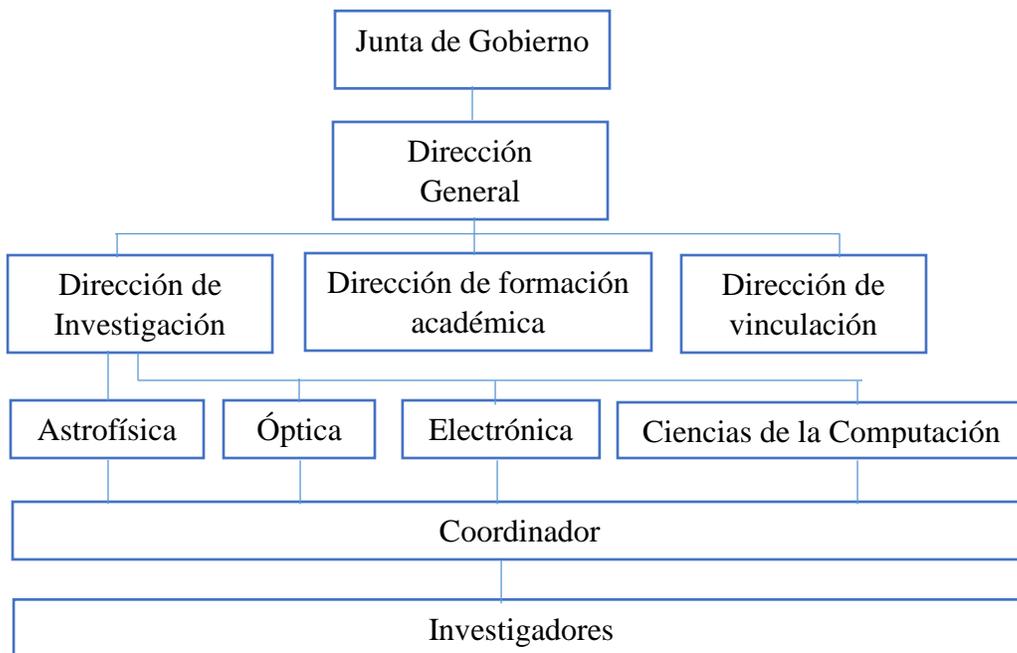


Figura 1.2 Organigrama de la institución

A pesar de que el INAOE cuenta con distintos programas educativos, el proyecto en general está basado en el departamento de Óptica, de la cual se muestra las características principales de esta coordinación.

1.5 Coordinación de óptica

La Coordinación de Óptica fue creada en 1972, justo después de la fundación del INAOE. Sus dos postgrados, Maestría y Doctorado en Ciencias con especialidad en Óptica, son los más antiguos del INAOE.

Actualmente el programa de maestría está catalogado como de Competencia Internacional, mientras que el de doctorado como Consolidado, en el Padrón Nacional de Posgrados de Calidad del CONACYT.

1.6 Misión de la coordinación

La misión de los investigadores asociados a la Coordinación de Óptica es:

- realizar investigación básica de vanguardia,
- realizar investigación aplicada orientada a satisfacer las necesidades de la sociedad y
- formar recursos humanos capaces de resolver problemas científicos y tecnológicos de alta relevancia.

Todo lo anterior dentro de las siguientes líneas generales de investigación:

- Biofotónica
- Fotónica
- Instrumentación Óptica y Metrología
- Óptica Cuántica
- Óptica Estadística
- Óptica Física
- Optoelectrónica
- Procesado de Imágenes

1.7 Descripción general del proyecto

En un entorno general, el INAOE cuenta con distintos departamentos educativos dónde cada uno administra los recursos de manera diferente, asimismo el departamento de Óptica cuenta con los laboratorios que se muestran a continuación:

- Biofotónica
- Ciencias De La Imagen
- Comunicaciones Ópticas Y Optoelectrónica
- Fibras Ópticas
- Holografía
- Imágenes Y Color Digital
- Instrumentación Óptica
- Microscopía Óptica y Metrología Dimensional
- Óptica Difractiva
- Óptica Biomédica
- Películas Delgadas

Asimismo, el departamento de Óptica cuenta con 11 laboratorios prácticos teóricos dónde la asignación para los diferentes cursos es de manera mecánica mediante registros sin una manera intuitiva de manejar la asignación tanto de laboratorios como del material destinado a cada laboratorio.

Se plantea el uso de herramientas de programación para dispositivos móviles para el desarrollo de una aplicación en el sistema operativo Android que ayude a mejorar el proceso de asignación de laboratorios, así como de su material.

1.7.1 Planteamiento del problema

En el Instituto Nacional de Astrofísica, Óptica y Electrónica las organizaciones de acuerdo a los diferentes ramos estudiantiles manejan de diferente forma el control y asignación de materiales y laboratorios. Esto se refleja en la cantidad de información plasmada en registros de los cuales se posee un gran inventario.

Por lo tanto, esto genera un tratamiento de información ineficiente debido al proceso de gestión para llevar a cabo la asignación de laboratorios y material.

Con un vasto número de laboratorios, así como de equipo, es necesario enfocar una solución óptima al tratado de este punto.

De tal modo, es necesario desarrollar una aplicación accesible para los usuarios que podrán hacer la correcta administración desde un dispositivo móvil orientado al sistema operativo Android.

1.7.2 Justificación

Hoy en día diferentes instituciones independientemente de los servicios de educación que ofrezcan, manejan sistemas de información, los cuales pueden ser adquiridos mediante licencias o desarrollados por agencias externas.

Garbarini, Ramiro W. Cigliuti, Pablo Bursztyn, Andrés P. M. Pollo Cattaneo, María Florencia (2013) mencionan que, la gestión de un laboratorio universitario tiene por objeto brindar servicios y recursos a sus destinatarios, docentes, alumnos e investigadores de su comunidad. La gestión debe enfocarse en satisfacer requerimientos: legales, de la institución en particular y de sus destinatarios. Para cumplir con estas premisas, se deben administrar de forma eficiente el personal, el equipamiento, y las herramientas que posee.

De tal modo, es posible llevar a cabo el desarrollo de un sistema que sea capaz de gestionar y mejorar la operatividad en el Instituto.

Con el desarrollo de aplicaciones móviles dado el caso, es posible el desarrollo de software de acuerdo a las necesidades. Se plantea el uso esté orientado hacia el sistema operativo Android debido a la facilidad de uso y la fácil adquisición de un dispositivo con este sistema operativo.

El impacto esperado es considerado, debido a que se espera una mejora en la operatividad en la gestión laboratorios, así como el equipo que está asignado al área de Óptica.

1.7.3 Objetivo general

Implementar una aplicación móvil para la asignación y administración de laboratorios y equipo del cuerpo académico de Óptica

1.7.4 Objetivos específicos

- Diseñar una base de datos para el almacenamiento de información acorde a los laboratorios y materiales
- Desarrollar una aplicación móvil en el sistema operativo Android que agilice el proceso de asignación de laboratorios
- Realizar 5 pruebas de usabilidad con los laboratorios del cuerpo académico de Óptica

1.7.5 Alcances y limitaciones

De acuerdo a los alcances y limitaciones del proyecto, están proyectados de la siguiente manera:

- Alcances

1. Asignación de laboratorios del cuerpo académico de óptica en un dispositivo móvil
2. La consulta de información será a través de cualquier dispositivo móvil con sistema operativo Android
3. Reducir el tiempo de asignación de laboratorios, así como la de materiales

- Limitaciones

1. La aplicación no estará disponible para otro sistema operativo
2. La aplicación sólo estará ofertada para el cuerpo académico de óptica
3. El tiempo de desarrollo es corto para la aplicación de la aplicación

CAPITULO II

METODOLOGÍA Y HERRAMIENTAS DE DESARROLLO

2.1 Metodología de desarrollo ágil de software

El realizar el desarrollo de un proyecto, lleva consigo buscar guías o lineamientos que nos digan los pasos a seguir para realizar un sistema de calidad, un conjunto de puntos a cubrir para obtener un resultado satisfactorio.

Para evaluar que metodología de desarrollo de software es conveniente de implementar, se necesitan evaluar diferentes puntos de vista como lo es la magnitud del proyecto, el tiempo estimado para el desarrollo y las etapas que este deba de pasar para poder llegar a una etapa final.

Una de las opciones a considerar para el desarrollo de software a corto plazo, es el uso de metodologías de desarrollo ágil.

De acuerdo con Amaya Balaguera (2013), las metodologías ágiles han ganado popularidad desde hace algunos años, ya que constituyen una buena solución para proyectos a corto plazo, en especial, aquellos proyectos en dónde los requisitos están cambiando constantemente, un ejemplo de esto son las aplicaciones para dispositivos móviles, debido a que éstas tienen que satisfacer una serie de características y condicionantes especiales, tales como: canal, movilidad, portabilidad, capacidades específicas de las terminales, entre otras, y aun cuando existen miles de aplicaciones para dispositivos móviles que corren en diferentes sistemas operativos.

2.1.1 Programación extrema

La *programación extrema* se define como un conjunto de pasos de diversas metodologías, acopladas de manera que sean pasos flexibles a seguir utilizadas con el uso común, para realizar un desarrollo más agradable.

De acuerdo con lo anterior, la razón para optar por el desarrollo ágil utilizando a la programación extrema como metodología, tiene 4 valores principales para poder llevar con éxito el desarrollo de software:

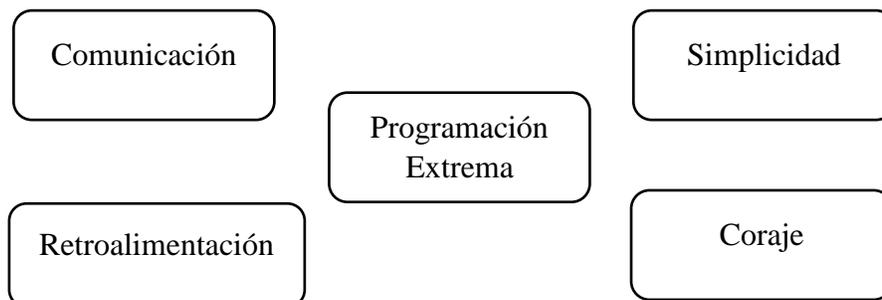


Figura 2.1 Valores de la Programación Extrema

2.1.2 Ventajas y desventajas de la programación extrema

2.1.2.1 Ventajas

Una de las ventajas de la programación extrema es que se adapta al desarrollo de sistemas pequeños y grandes, optimiza el tiempo de desarrollo, con los lineamientos que plantea esta metodología, se realizan versiones pequeñas de las funcionalidades del sistema, realizando entregas periódicas al usuario; este se vuelve parte del grupo de desarrollo por la constante comunicación e interacción que hay entre ambos durante el desarrollo del proyecto, permitiendo realizar cambios o ajustes en el código, a tiempo.

Asimismo, se puede añadir las pruebas que existen a lo largo del periodo de desarrollo, ya que de esta manera se verifica en periodos de entrega que, las funcionalidades previstas para las fechas de entrega cumplan con lo solicitado.

De esta manera se facilitan los cambios dentro de la aplicación de acuerdo a los puntos de vista recabados por parte del desarrollador, así como del cliente.

2.1.2.2 Desventajas

Como toda metodología, la programación extrema no está exenta de algunos problemas, como la definición del costo y el tiempo de desarrollo, el sistema va creciendo después de cada entrega al usuario y nadie puede decir que el usuario no querrá una función más al sistema.

Una de las desventajas más grandes de esta metodología es la que conlleva el desarrollo de sistemas de alta calidad, para esto es necesario la presencia constante del cliente lo cual es difícil de mantener constante el seguimiento por parte del usuario.

2.1.2.3 Proceso de desarrollo

De manera general, el proceso de desarrollo consiste en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El cliente determinará la importancia de cada una de las funcionalidades que tendrá el sistema.
- El programador estima el esfuerzo necesario para su implementación.
- El grupo de desarrollo da una estimación de tiempo necesario para su desarrollo.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El cliente es el personaje que establece el orden de la funcionalidad que debe tener el sistema

- El programador construye ese valor de negocio.
- Vuelve al primer paso.

En cada una de las iteraciones del proyecto, el grupo de desarrollo además del cliente aprenden de manera coordinada, además para agilizar la calidad del software no es recomendable la presión hacia el grupo de desarrolladores para realizar más trabajo de lo establecido.

El ciclo de vida ideal de la programación extrema consta de seis fases listadas a continuación:

1. Exploración
2. Planificación de la entrega (*Release*: Es la comodidad de entrega)
3. Iteraciones
4. Producción
5. Mantenimiento
6. Muerte del proyecto

Primera Fase: **Exploración**

En esta primera fase el cliente debe definir las historias de usuario, estas son una forma rápida de administrar los requisitos sin tener que elaborar gran cantidad de documentos y con poca administración.

Una *historia de usuario* es una tarjeta de papel donde se escribe de manera breve las características que el sistema debe poseer.

Estas historias de usuario deben ser desarrolladas en un plazo no mayor a tres semanas, para no superar el tiempo límite de una iteración.

Segunda Fase: **Planificación de la entrega**

Después de haber sido escritas las historias de usuario, el cliente deberá establecer la prioridad de cada una de ellas, con esto el grupo de desarrollo elaborará una estimación aproximada del esfuerzo necesario para su implementación, dando como resultado un cronograma de trabajo donde se especifican las fechas de entrega.

Tercera Fase: **Iteraciones**

El proceso de desarrollo incluye varias iteraciones antes de ser entregado, por lo que debe realizarse un plan de entrega dónde se especifica el número de iteraciones necesarias para el proceso de desarrollo.

Estas iteraciones clasifican un número determinado de historias de usuario que tendrán que ser implementadas dentro de cierto tiempo.

Cuarta Fase: **Producción**

Dentro de esta fase se realizan pruebas adicionales y revisiones antes de que el sistema sea instalado al cliente, así mismo se debe determinar si es necesario incluir nuevas características al sistema actual, esto depende de los resultados y pruebas anteriores.

Quinta Fase: **Mantenimiento**

Una vez que la primera versión se encuentra en producción, se debe mantener el sistema en funcionamiento al mismo tiempo que se desarrollan nuevas iteraciones.

Para realizar a cabo lo antes mencionado, se requiere de tareas de soporte para el cliente, asimismo, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción.

Sexta Fase: **Muerte del Proyecto**

En esta fase es cuando el cliente no tiene más historias para ser incluidas en el sistema, de esta manera se asegura que el sistema satisface las necesidades del cliente en aspectos de rendimiento y confiabilidad del sistema.

2.2 Herramientas de desarrollo

Para todo desarrollo de software, es necesario contar con las herramientas necesarias para llevarlo a cabo, a continuación, son mencionadas:

2.2.1 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA.

Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes:

- Sistema de compilación flexible basado en Gradle.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run, para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.

- Integración de plantillas de código y [GitHub](#), para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.
- Compatibilidad con C++

2.2.2 Sistema Operativo Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, *tablets* o tablefonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró.

Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.

Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS

Actualmente este sistema operativo ha lanzado su última versión 7.1 denominada Nougat, sin embargo, la versión más ejecutada a nivel de dispositivos es la Android 4.4 KitKat.



Figura 2.2 Logotipo oficial de Android

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	4.0%
4.2.x		17	5.7%
4.3		18	1.6%
4.4	KitKat	19	21.9%
5.0	Lollipop	21	9.8%
5.1		22	23.1%
6.0	Marshmallow	23	30.7%
7.0	Nougat	24	0.9%
7.1		25	0.3%

Tabla 1.1 Versiones del sistema operativo

2.2.3 SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB) escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia del sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica.

En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina local.

Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

CAPITULO III

DESARROLLO DE APLICACIÓN MÓVIL PARA LA ADMINISTRACIÓN, ASIGNACIÓN DE LABORATORIOS Y MATERIALES DE ÓPTICA

En este capítulo se define la metodología con la que se llevó a cabo el desarrollo de la aplicación móvil.

Para poder lograr que el desarrollo de software sea eficaz, es necesario elaborar una planificación total.

Como ya se ha mencionado con anterioridad, el presente trabajo pretende dar solución a la problemática del manejo y administración de la información de acuerdo a la asignación de laboratorios a grupos del instituto, así como la administración del material de laboratorio.

3.1 Definición del material

Para el desarrollo de software, se optó por el desarrollo en el sistema operativo de Android, esto es debido a que ofrece entornos de desarrollo gratuitos además de ser de código abierto.

Android es el sistema operativo de Google el cual respalda de recursos de código y soporte para la plataforma, siendo así una opción óptima para su elección.

Como IDE, se optó por Android Studio debido a que ofrece gran cantidad de virtualizaciones de dispositivos móviles, así como la descarga y manipulación de versiones del sistema operativo Android.

Para el almacenamiento de los datos, se optó en su primera versión realizarlo en el gestor de base de datos propio del sistema Android, SQLite debido a la respuesta de entrega de datos y poco uso de recursos de sistema.

En cuanto a recursos tecnológicos, la dirección de óptica puso a la disposición de una computadora con características de procesamiento que ofrece un procesador Intel Core i3 de 2.90 GHz, 12 GB de memoria RAM y 1 TB en disco duro con sistema operativo Windows 7 de 64 bits.

3.2 Proceso de desarrollo

3.2.1 Primera Fase: Exploración

En la primera fase para el desarrollo de la aplicación, es la interacción con el cliente, por lo que según las características de la metodología XP son necesarias las historias de usuario.

En la primera entrevista con el cliente, en este caso el Dr. Francisco Javier Renero Carrillo, se determinó que la aplicación sería capaz de cumplir con los siguientes puntos:

- Crear y respaldar una base de datos
- Conectarse a la base de datos tanto local como remota para la manipulación de la información.
- Realizar operaciones como: agregar, eliminar, editar y consultar los registros de dicha base de datos.
- Creación de módulos de usuario, laboratorios y materiales.
- Presentar una interfaz fácil de utilizar para el manejo de información.
- Minimizar la tarea de la asignación de materiales de laboratorio, así como la de los laboratorios.
- Minimizar el riesgo de manipulación de la información por personal no autorizado.

3.2.1.1 Historias de usuario

De los puntos tratados con anterioridad, se desprenden las primeras historias de usuario que son enlistadas a continuación:

- Diseño de una base de datos
- Diseño de interfaces
- Creación de módulo de usuarios
- Creación de módulo de laboratorios
- Creación de módulo de materiales
- Gestión de la base de datos
- Administración de la base de datos
- Manejo de sesiones de la aplicación
- Asignación de laboratorios

Sin embargo, un factor en contra al momento de definir las historias de usuario fueron los impedimentos extras que se mencionan a continuación:

- El instituto no pudo otorgar en tiempo y forma de un servidor para la instalación y administración de una base de datos.
- La base de datos no puede contemplar información de terceros sin la plena autorización por parte del departamento de servicios escolares el cual fue negado.
- El módulo de usuarios quedaría por el momento en modo de espera debido al tiempo, desarrollando sólo los módulos de laboratorios y materiales en su primera versión.

De acuerdo con la información recabada con anterioridad, las historias de usuario se presentan a continuación:

Historia de Usuario	
Número: 1	Usuario:
Nombre de la historia: Diseño de base de datos	
Iteración asignada: 1	
Programador responsable: Luis Arroyo Corona	
Descripción: Se muestra un diagrama de base de datos donde son mostradas las relaciones con las entidades.	
Observaciones:	

Tabla 1.2: Historia de Usuario 1. Diseño de base de datos

Historia de Usuario	
Número: 2	Usuario: Capturista
Nombre de la historia: Gestión de base de datos	
Iteración asignada: 1	
Programador responsable: Luis Arroyo Corona	
Descripción: Se realiza la inserción de la base de datos dentro del sistema, de esta manera se realizan las operaciones de cada módulo	
Observaciones:	

Tabla 1.3: Historia de Usuario 2. Gestión de base de datos

Historia de Usuario	
Número: 3	Usuario:
Nombre de la historia: Diseño de interfaces de usuario	
Iteración asignada: 2	
Programador responsable: Luis Arroyo Corona	
Descripción: Se realiza el diseño de las interfaces de la aplicación.	
Observaciones: Existe la posibilidad de que el diseño pueda variar ligeramente debido a las versiones del sistema operativo	

Tabla 1.4: Historia de Usuario 3. Diseño de interfaces de usuario

Historia de Usuario	
Número: 4	Usuario: Encargado de departamento
Nombre de la historia: Módulo de laboratorios	
Iteración asignada: 3	
Programador responsable: Luis Arroyo Corona	
Descripción: Se realiza el desarrollo del módulo de laboratorios dónde están las opciones de altas, bajas, consultas y modificaciones de la información de laboratorios.	
Observaciones:	

Tabla 1.5: Historia de Usuario 4. Módulo de laboratorios

Historia de Usuario	
Número: 5	Usuario: Encargado de departamento
Nombre de la historia: Módulo de materiales	
Iteración asignada: 3	
Programador responsable: Luis Arroyo Corona	
Descripción: Se realiza el desarrollo del módulo de laboratorios dónde están las opciones de altas, bajas, consultas y modificaciones de la información de laboratorios.	
Observaciones: En esta primera versión no contiene imágenes acerca de los materiales correspondientes	

Tabla 1.6: Historia de Usuario 4. Módulo de laboratorios

Historia de Usuario	
Número: 6	Usuario: Encargado de departamento
Nombre de la historia: Manejo de sesiones	
Iteración asignada: 4	
Programador responsable: Luis Arroyo Corona	
Descripción: El usuario del sistema puede ingresar a la aplicación mediante un usuario y contraseña establecido	
Observaciones:	

Tabla 1.7: Historia de Usuario 6. Manejo de sesiones

Historia de Usuario	
Número: 7	Usuario: Encargado de departamento
Nombre de la historia: Asignación de laboratorios	
Iteración asignada: 5	
Programador responsable: Luis Arroyo Corona	
Descripción: El usuario del sistema puede asignar horarios de uso a grupos en existencia	
Observaciones:	

3.2.2 Segunda Fase: Planificación de entrega

Una vez realizada la valoración de las historias de usuario y con el análisis debidamente requerido de acuerdo a las disposiciones, se presenta a continuación la tabla donde se especifica las fechas correspondientes al plan de entrega.

Número de Iteración	Número de historia	Fecha de entrega
1	1 y 2	02/02/2017
2	3	10/02/2017
3	4 y 5	10/03/2017
4	6	16/03/2017
5	7	22/03/2017

En la primera iteración se desarrollará las funcionalidades descritas en la primera historia de usuario, en esta sección se llevará a cabo el diseño de la base de datos además de la inserción de la base de datos en el sistema, así como su gestión.

En la segunda iteración, se crearán el diseño de la interfaz con la que el usuario mantendrá interacción.

En la tercera iteración se desarrollará la parte de mayor tiempo, esta es el desarrollo de los módulos de laboratorios y materiales, donde se generarán las acciones de agregar, eliminar, consultar y modificar los registros correspondientes a cada módulo.

En la cuarta iteración, se desarrollará el manejo de sesiones para asegurar la entrada y salida del usuario final.

En la última iteración, el usuario podrá asignar horarios de uso de laboratorio a los grupos existentes del departamento de Óptica en su primera versión.

3.2.3 Tercera fase: Iteraciones

3.2.3.1 Primera Iteración

Diseño de base de datos y gestión

En la primera iteración se procede al desarrollo de la primera historia de usuario, para ello se empieza por diseñar las tablas correspondientes a la base de datos.

Realizando el análisis requerido para establecer la estructura de las tablas de la base de datos, en la figura 3.1 se muestra la estructura gráfica general de ellas.

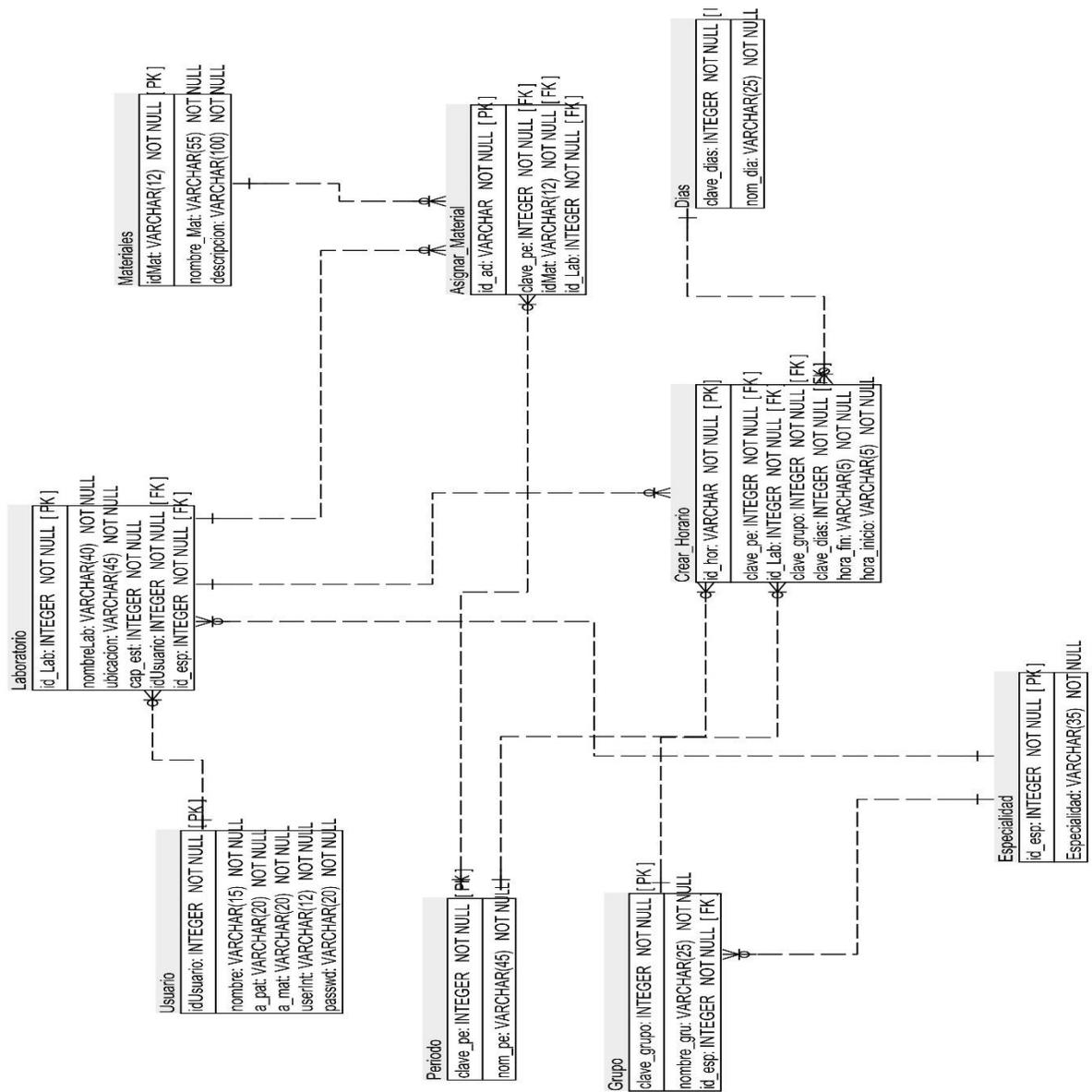


Figura 3.1 Diseño de base de datos

De acuerdo con el diseño previamente establecido de la base de datos, fué de la siguiente manera debido a que existe información que no pudo ser utilizada, asimismo se establecieron tablas que no generen redundancia.

Debido a que es una base de datos que al momento está implantada de manera local, SQLite permite la inserción de la base de datos creando una clave con la extensión .java dentro del proyecto que comparte los módulos de laboratorio y materiales.

Para crear y ejecutar la base de datos dentro del proyecto de Android Studio, se crea la siguiente clase importando las siguientes librerías y métodos:

```
BaseHelper onCreate()
1 package com.example.luis.basecrud;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7
8 public class BaseHelper extends SQLiteOpenHelper
9 {
10
11     String Usuario="CREATE TABLE USUARIO(idUsuario INTEGER PRIMARY KEY NOT NULL, nombre TEXT NOT NULL," +
12         " a_pat TEXT NOT NULL, a_mat TEXT NOT NULL, userInt TEXT NOT NULL, passwd TEXT NOT NULL)";
13
14     String Especialidad="CREATE TABLE ESPECIALIDAD(id_esp INTEGER PRIMARY KEY NOT NULL, Especialidad TEXT NOT NULL)";
15
16     String Laboratorio="CREATE TABLE LABORATORIO(id_Lab INTEGER PRIMARY KEY NOT NULL, nombreLab TEXT NOT NULL," +
17         " ubicacion TEXT NOT NULL, cap_est INTEGER NOT NULL, idUsuario INTEGER NOT NULL, id_esp INTEGER NOT NULL," +
18         " FOREIGN KEY (idUsuario) REFERENCES USUARIO (idUsuario), FOREIGN KEY (id_esp) REFERENCES Especialidad (id_esp)");
19
20     public BaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
21         super(context, name, factory, version);
22     }
23 }
```

Figura 3.2 Creación de Base de Datos

```
BaseHelper onCreate()
public BaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}

@Override
public void onCreate(SQLiteDatabase db)
{
    db.execSQL(Usuario);
    db.execSQL(Especialidad);
    db.execSQL(Laboratorio);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    db.execSQL("drop table if exist USUARIO");
    db.execSQL("drop table if exist ESPECIALIDAD");
    db.execSQL("drop table if exist Laboratorio");
    onCreate(db);
}
}
```

Figura 3.3 Ejecución de métodos

La clase denominada *BaseHelper* es una clase de java que crea tablas a través de cadenas de texto, las cuales pasa por parámetros hacia otros métodos, estos métodos ejecutan dichas instrucciones dependiendo de la solicitud dónde sean requeridos.

Para más información, parte del código fuente estará disponible en el anexo de este documento.

Una vez creado todas las cadenas de texto con las correspondientes interpretaciones SQL para la creación de tablas, es posible mandar valores mediante referencias dentro de la clase principal.

Este proceso no debe de ser mostrado hacia el usuario final, para ello es necesario de interfaces que cumplan con un diseño acorde a la institución como logotipo y colores.

3.2.3.2 Segunda Iteración

Diseño de interfaz de usuario

Para que una aplicación móvil pueda ser de utilidad, la interfaz de usuario tiene un papel importante ya que esta es la manera de presentación con la que el usuario estará familiarizado.

De esta manera y de acuerdo a los estándares para el diseño de aplicaciones universales por parte de Android, se implementó el estilo de diseño *Material Design* la cual es interpretada como una normativa de diseño enfocado en la visualización del sistema operativo Android, además en la web y en cualquier plataforma.

Sin embargo, es necesario modificar el proyecto, debido a que esta normatividad sólo aplica para las versiones del sistema operativo 5.0 en adelante.

Para ello, en la ruta App>build.gradle(Module: app) se añade una extensión para que una versión del sistema operativo inferior a Android 5.0 pueda mantener esta normatividad.

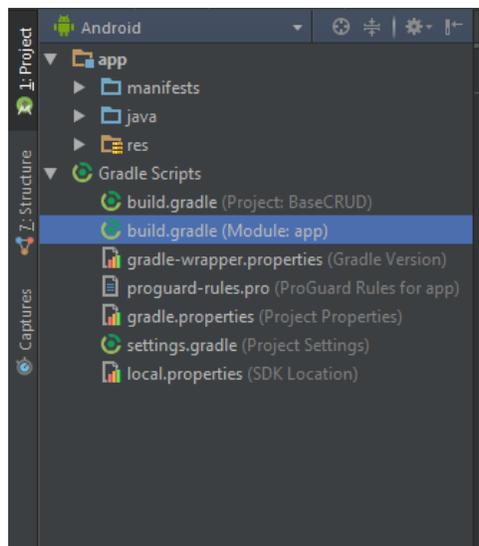


Figura 3.4 Ruta para añadir extensión

A continuación, se presenta la extensión para versiones de sistemas inferiores.

compile 'com.android.support.palette-v7:+'

Esta última línea es agregada al archivo build.gradle.

Google provee de información para el desarrollo de aplicaciones, provee además de código fuente herramientas para el diseño de aplicaciones, tal es el caso de la paleta de colores estandarizada por Google.

Es posible acceder mediante el siguiente enlace <https://material.io/guidelines/style/color.html> , además es posible adquirir iconografía libre de derecho de autor de la página de Google, a continuación se presenta el siguiente enlace: <https://material.io/icons/>.

De acuerdo con el logotipo y colores de la institución, la propuesta para la pantalla de presentación es la siguiente:



Figura 3.5 Pantalla de inicio

Como se puede observar en la imagen anterior, es posible agregar colores que identifiquen a la institución así como una barra de estado de la aplicación que indica cuando ha cargado totalmente para ser utilizada.

De acuerdo con el diseño de la interfaz, se plantea el uso de un menú principal sencillo donde el usuario pueda navegar de manera libre, es por ello que se optó por un menú centralizado además de una barra lateral para más aditamentos a la aplicación.



Figura 3.6 Menú Principal

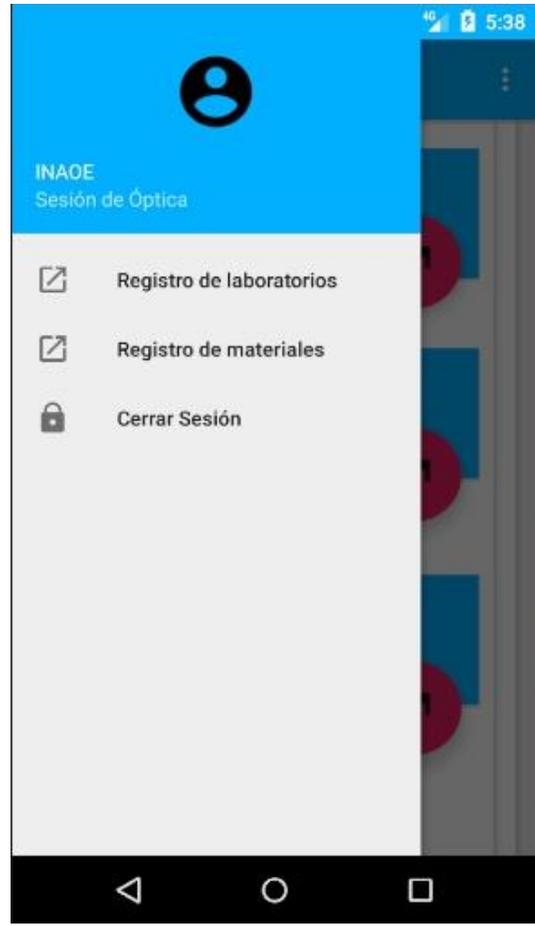


Figura 3.7 Barra de navegación

Se optó por este diseño debido a que muchas aplicaciones cuentan con diseños similares de navegación, de esta manera el usuario podrá familiarizarse rápidamente al manejar la aplicación.

Como se puede observar, cuenta con dos botones principales en el menú principal que cambiarán al catálogo de laboratorios o materiales.

En la barra lateral, es posible seguir el perfil del usuario, visualizando los registros de laboratorios o materiales con la opción de cerrar sesión.

De acuerdo a este diseño, se debe de mantener el diseño de la interfaz en todas las pantallas dando así una vista que respete el diseño principal.

3.2.3.3 Tercera Iteración

Módulo de laboratorios

Para esta historia de usuario, se toma en cuenta las tablas que serán usadas para el almacenamiento de los datos, de acuerdo a esto, primero es necesario registrar datos acerca de los laboratorios.



Figura 3.8 Menú principal laboratorios

Este menú respeta la integridad del menú principal adicionando de un botón más para las consultas que serán realizadas.

Dentro de la opción de consultas, se encuentra la opción para modificar la información de los laboratorios.

La navegación para el retroceso entre pantallas esta dado por la barra inferior de navegación que provee el dispositivo.



Figura 3.9 Alta de laboratorios

Para el alta de laboratorios, la información como el Nombre, Ubicación, Capacidad de estudiantes y la Especialidad son los datos iniciales con los que se registra un laboratorio.

El sistema proveerá de un identificador auto incrementable por cada registro.

El texto que aparece en cada caja de texto es debido a la propiedad *hint* la cual es una propiedad que permite escribir texto informativo dentro del componente, este último será borrado después de que el usuario escriba directamente sobre la caja de texto.

Todas las interfaces que requieran de la inserción de información poseen este tipo de comportamiento.

Al momento de oprimir el botón de aceptar, la aplicación manda a la base de datos los valores ingresados.

Ningún valor puede ser nulo, es por esto que cada caja de texto esta validada mediante condiciones de completado de texto.



Figura 3.10 Baja de laboratorios

De acuerdo a la inserción de información, la opción de Baja de laboratorios es de manera más sencilla para el usuario, esto es debido a que solo se cuenta con una caja de texto para escribir el nombre del laboratorio que desea ser eliminado.

Para facilitar el uso, la caja de texto cuenta con un modo de autocompletado para que de esta manera pueda seleccionar el ítem buscado sin tener que teclear o memorizar todo el nombre.

Por el momento, la muestra de información que aparecen dentro de la Baja de laboratorios es de manera tentativa, esto es debido a que mientras más información se presente dentro de la búsqueda de laboratorio para su correspondiente baja, esta mostrara un *ScrollView*, que es una herramienta que provee Android Studio para poder visualizar más contenido en una ventana con ciertas dimensiones para la visualización de información.

Esta herramienta es utilizada en otras ventanas para mantener estructura de las funciones de la aplicación.



Figura 3.10 Consulta de laboratorios

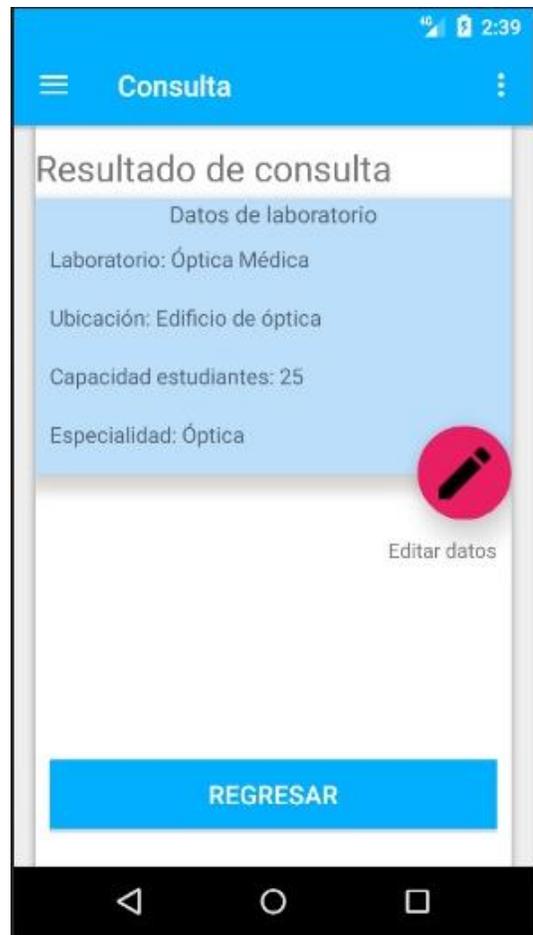


Figura 3.11 Resultado de consulta

La consulta de información para laboratorios posee una similitud con el formulario para Eliminar Laboratorio, sin embargo, este posee dos funciones más, al momento de escribir el nombre del laboratorio, este despliega las posibles opciones de manera de autocompletado y además redirecciona a otro formulario con el resultado de la consulta.

En la ventana de consulta, se visualizará en un recuadro la información del laboratorio, de tal manera que el usuario podrá editar dicha información tocando el botón con el dialogó Editar datos, este botón enviará a una ventana con campos editables.

Para realizar el proceso de autocompletado, parte del código puede ser consultado en la parte de anexos.

Módulo de materiales

Dentro de este módulo, se toma en cuenta las tablas que serán usadas para el almacenamiento de los datos, de acuerdo a esto, primero es necesario registrar datos acerca de materiales.

Para mantener una navegación similar, se toma en cuenta las ventanas realizadas con anterioridad del anterior módulo.



Figura 3.12 Menú principal de materiales

El menú mostrado para este módulo cuenta con los botones de Altas, Bajas y Consultas, respetando el diseño de otras ventanas y formularios.



Figura 3.13 Alta de materiales

La ventana mantiene el mismo diseño, sin embargo, varía en los campos para agregar texto donde sólo se presentan 3 cajas de texto con el Código de Inventario, el Nombre del material y una Descripción.

Como es obvio, cada ventana del módulo de Materiales se asemejará entre diseños realizados con anterioridad, con la finalidad de tener una navegación y ventanas similares.

Con esto es posible reutilizar herramientas de Android Studio, de tal modo que la aplicación final no consumirá demasiados recursos y espacio en memoria.

3.2.3.4 Cuarta Iteración

Manejo de sesiones

Para esta historia de usuario, se plantea el uso de un manejo de sesiones dónde el usuario pueda lograr tener una identificación con el sistema, esto es debido a la seguridad de la aplicación.

Para ello, es necesario utilizar un método de encriptación de claves para mejorar la seguridad de la aplicación.

El código para la encriptación e inicio de sesión se pueden consultar en la parte de anexos de este documento.



Figura 3.14 Inicio de sesión

La aplicación pide como requisitos un usuario y contraseña, estos serán dados por el administrador de la aplicación.

Esto es debido a la falta de tiempo de desarrollo para un portal web que gestione a los usuarios asignando y designando permisos, así como cambio de contraseñas.



Figura 3.15 Prueba de inicio de sesión

De acuerdo a lo anterior, el sistema valida los datos ingresados de acuerdo a la información que se encuentra en el registro, si esta es correcta, el sistema pasará hacia el menú principal de la aplicación.

3.2.3.5 Quinta Iteración

Para esta historia de usuario, el usuario puede hacer uso de la información antes almacenada dentro de la base de datos para la asignación de horarios de una manera intuitiva.

De esta manera puede gestionar la información de los horarios en uso de acuerdo a los días de la semana disponibles.



Figura 3.16 Menú para el registro de horarios

Como parte del diseño, el menú plantea el mismo aspecto que los menús encontrados dentro de los catálogos de laboratorios, así como de materiales.

Un aspecto importante es que, para acceder a este menú el usuario puede desplegar la barra lateral izquierda donde se encuentra dicho menú.

Para asignar un laboratorio, como campos principales de requiere del nombre del laboratorio al que se va a asignar el horario, así como en nombre del grupo y el periodo correspondiente.

Después, el usuario puede seleccionar los días en los que se requerirá el laboratorio y las horas de uso.

Por último, puede guardar los datos de la ventana presionado el botón de asignar.



Figura 3.17 Asignación de laboratorio

Para modificar los campos del horario, esto se hace mediante el ingreso del nombre del laboratorio, el resultado arrojará los horarios que han sido asignados al laboratorio que ha sido consultado, de esta manera es posible hacer una búsqueda mediante el nombre del laboratorio.

La caja de texto contiene el método de autocompletado como se ha trabajado con formularios anteriores.

Cuando la consulta es mostrada en pantalla, el usuario puede editar los datos que se muestran en la ventana, de esta manera la transición de ventanas reduce y el proceso es más rápido.



Figura 3.18 Resultado de consulta y edición

3.2.4 Cuarta fase: Producción

Una vez concluidas las iteraciones del proyecto, en las cuales se especifica claramente la forma en que se llevaron a cabo, el siguiente paso es el de verificar que cada una de las historias de usuario cumplan con las necesidades del usuario.

Es el momento de unificar las secciones correspondientes a la aplicación, para ello es necesario de generar el archivo .apk donde se podrá instalar de forma individual la aplicación.

En la sección de Android Studio, en la pestaña Build > Build APK se generará archivo de instalación.

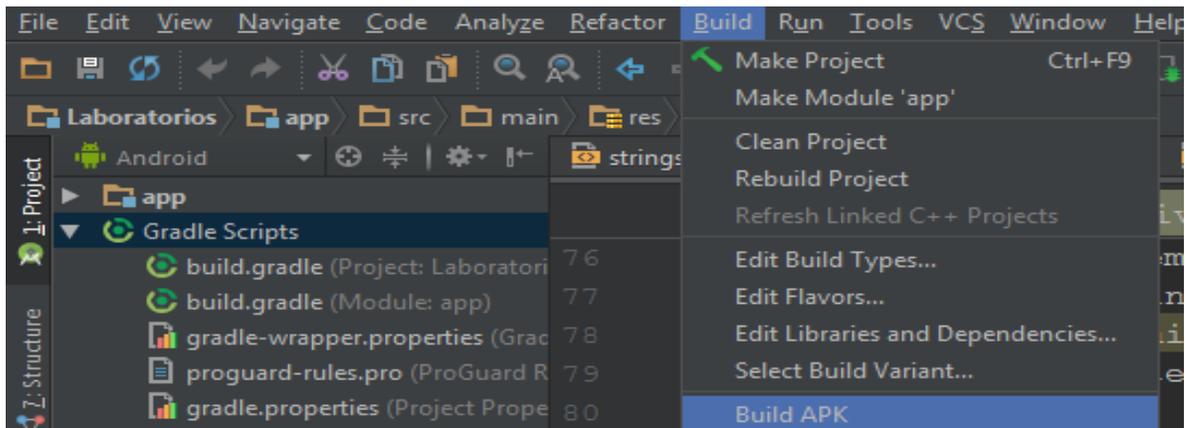


Figura 3.19 Ruta para crear APK

Una vez generado el archivo .APK, este puede ser encontrado en el siguiente directorio C:\Users\INAOE\Desktop\Proyecto\Laboratorios\app\build\outputs\apk.

La ubicación del archivo dependerá de la ruta que se haya establecido al momento de instalar Android Studio.

Nombre	Fecha de modifica...	Tipo	Tamaño
app-debug.apk	24/03/2017 10:20 a...	Archivo APK	2,010 KB

Figura 3.20 Archivo APK

La aplicación pedirá permisos para la instalación dentro del dispositivo móvil.

La opción instalar es seleccionada y se instala la aplicación.

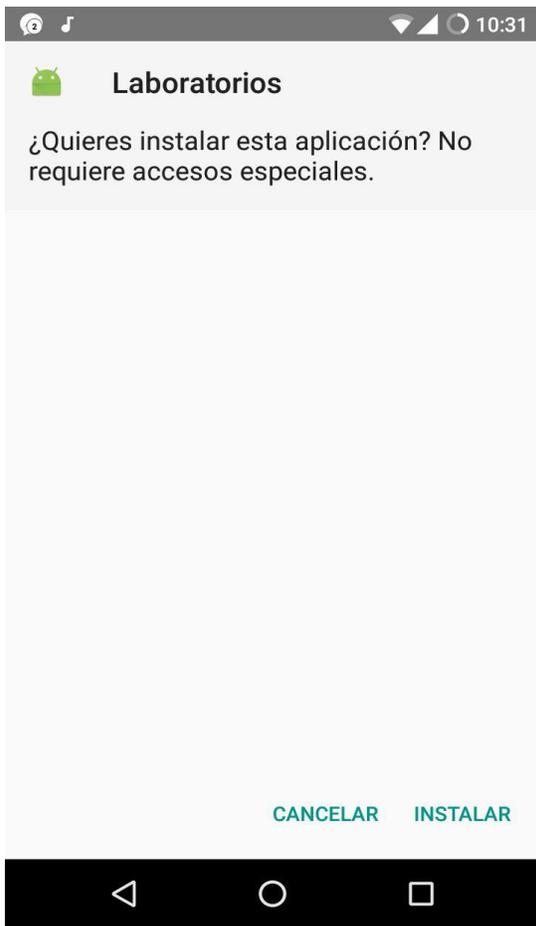


Figura 3.21 Permisos de instalación

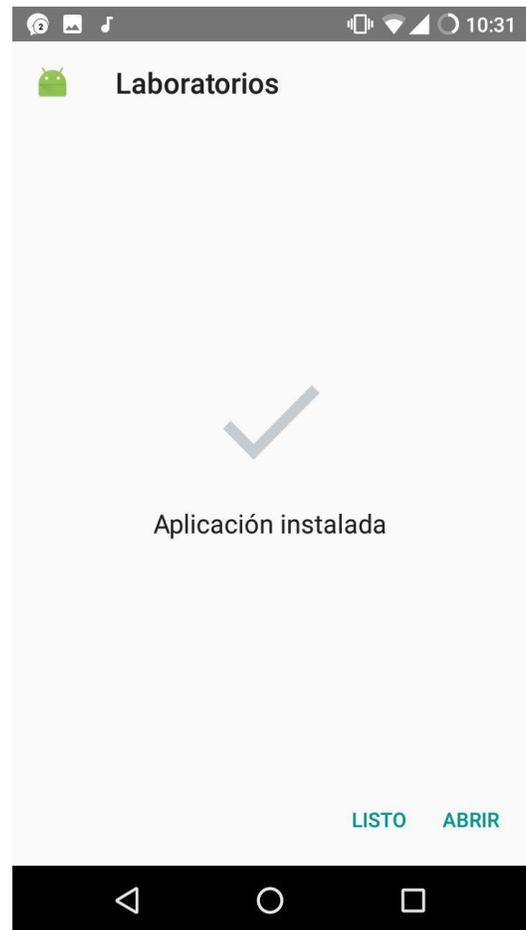


Figura 3.22 Aplicación instalada

3.2.5 Quinta fase: Mantenimiento

Una vez realizadas las pruebas de unificación e instalación, se procede a la entrega del sistema, para ello se realizó una cita con el cliente, con una fecha de anticipación, dejando en pendiente la capacitación del personal.

Una corrección a la aplicación es un error de margen en el apartado de la asignación de horarios donde las cajas de texto para ingresar la hora en la que serán ocupados los laboratorios se encuentran parcialmente desalineados, esta corrección se realizó cambiando la altura y anchura de los elementos en un formato XML.

3.2.6 Sexta fase: Muerte del proyecto

Así es como el desarrollo de una aplicación móvil para la administración, asignación de laboratorios y materiales de óptica llega a su fin de manera temporal, dado a que el cliente tiene una historia de usuario por añadir la cual es la asignación de materiales.

Esta última tuvo que quedar en planes a futuro debido al corto tiempo que hubo para el desarrollo de la aplicación.

CONCLUSIONES

El desarrollo de aplicaciones móviles que puedan ser utilizadas para la optimización de una tarea tiene ventajas y desventajas.

Una de las ventajas es poder realizar una tarea en diferentes lugares además de ser adaptada a distintos dispositivos que tengas ciertas características que cumplan para poder ejecutar dicha aplicación.

Sin embargo, no es posible de momento desarrollar aplicaciones de gran potencia de procesamiento que cubran procesos más sofisticados, haciendo del desarrollo móvil una opción no muy recurrente para este tipo de situaciones.

El soporte brindado por parte de las características de la plataforma de desarrollo, así como los diferentes soportes que tiene hace de Android un sistema móvil muy consultado.

Gracias a que Android es sistema operativo basado en java, algunos conocimientos del lenguaje en su gran mayoría fueron de utilidad para agilizar el proceso de desarrollo.

Por otra parte, el tiempo de asignación para culminar con la aplicación fue reducido considerablemente, dejando a un lado algunas funcionalidades que puedes completarse en un futuro.

Considero que la metodología de la programación extrema sirve en gran parte para desarrollar aplicaciones móviles que tengan un periodo de desarrollo considerable, pero no tan reducido como este.

Como trabajo a futuro, esta aplicación puede migrar de base de datos local a una base de datos con un gestor como Mysql que puede ser montado en un servidor como era un principio.

De esta manera ya no sería una base de datos local, además de brindar soporte a la aplicación mediante otros lenguajes de programación del lado del servidor como PHP.

Considero que aplique gran parte del conocimiento adquirido dentro de la Universidad y otra parte que adquiriré mediante prácticas externas, fortaleciendo mis conocimientos en un entorno real.

GLOSARIO

Conacyt: El Consejo Nacional de Ciencia y Tecnología (**Conacyt**) es un organismo público descentralizado del gobierno federal mexicano dedicado a promover y estimular el desarrollo de la ciencia y la tecnología en ese país. Tiene la responsabilidad oficial para elaborar las políticas de ciencia y tecnología nacionales.

Biofotónica: La biofotónica concierne al estudio, investigación y utilización del conocimiento sobre los biofotones. Las investigaciones tratan por lo general sobre puntos fundamentales de biofísica y materias relacionadas — por ejemplo, el metabolismo y el crecimiento y diferenciación de la célula, las relaciones entre la así llamada (luminiscencia retardada) y el espectro de las emisiones luminosas en la bioquímica de las macromoléculas en los tejidos vivos, etc.

Fotónica: La Fotónica es la ciencia y la tecnología de la generación, control y detección de fotones, que son partículas de luz.

Instrumentación Óptica: La instrumentación óptica sirve para procesar ondas de luz con el fin de mejorar una imagen para su visualización, y para analizar las ondas de luz (o fotones) para determinar propiedades características.

Metrología: Ciencia que estudia los sistemas de pesas y medidas.

Óptica Cuántica: La óptica cuántica es un campo de investigación que se ocupa la aplicación de la mecánica cuántica a fenómenos que implican la luz y sus interacciones con la materia.

Óptica Estadística: Surge al considerar el campo electromagnético como algo menos idealizado: supone distribuciones estocásticas para el campo electromagnético y la emisión de luz como un proceso aleatorio.

Óptica Física: La óptica física es la rama de la física que toma la luz como una onda y explica algunos fenómenos que no se podrían explicar tomando la luz como un rayo.

Optoelectrónica: La optoelectrónica es la rama de la electrónica que trata con la luz. Los dispositivos ópticos son aquellos que responden a la radiación de la luz, o que emiten radiación. Estos dispositivos responden a una frecuencia específica de radiación.

Procesado de Imágenes: El procesamiento de imágenes tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. La imagen puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente, por medio de monitores de televisión.

Fibras Ópticas: La fibra óptica es una delgada hebra de vidrio o silicio fundido que conduce la luz. Se requieren dos filamentos para una comunicación bi-direccional: TX y RX.

Holografía: Técnica fotográfica que permite obtener una imagen con un efecto óptico tridimensional mediante el uso de un rayo láser.

Microscopía Óptica: Se refiere al uso de microscopios que usen luz visible con el fin de visualizar u observar muestras.

Metrología Dimensional: La metrología dimensional incluye la medición de todas aquellas propiedades que se determinen mediante la unidad de longitud, como por ejemplo distancia, posición, diámetro, redondez, planitud, rugosidad, etc. La longitud es una de las siete magnitudes base del Sistema Internacional de Unidades (SI).

Óptica Difractiva: Una revisión al diseño y construcción de sistemas ópticos empleando lentes difractivas.

Óptica Biomédica: Es la rama de la óptica que se dedica a realizar investigaciones teóricas y experimentales de la interacción de la radiación óptica con los tejidos biológicos, incluyendo tanto el estudio de muestras ex-vivo de tejidos humanos como investigaciones in-vivo en pacientes para establecer metodologías propias de diagnóstico y de nuevas terapias basadas en el uso del láser y la radiación óptica en general.

Películas Delgadas: Las películas delgadas son estructuras sólidas, tan delgadas que se puede despreciar muchos efectos físicos en su grosor. La mayoría de las películas delgadas interactuarán con ondas, por lo cual su grosor debe ser del orden de la longitud de onda de la perturbación con la que interacciona.

Gradle: Es una herramienta para automatizar el proceso de construcción de proyectos a nivel jerárquico (compilar, prueba, empaquetado).

APK: Un archivo con extensión .apk (Aplicación empaquetada de Android) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para teléfonos inteligentes y tabletas

GitHub: Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Framework: Define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

ACID: En bases de datos se denomina ACID a las características de los parámetros que permiten clasificar las transacciones de los sistemas de gestión de bases de datos.

ANEXO

ANEXO PARA BASE DE DATOS

```
package com.example.luis.Laboratorio;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class BaseHelper extends SQLiteOpenHelper
{

    String Usuario="CREATE TABLE USUARIO(idUsuario INTEGER PRIMARY KEY NOT
NULL, nombre TEXT NOT NULL," +

        " a_pat TEXT NOT NULL, a_mat TEXT NOT NULL, userInt TEXT NOT NULL,
passwd TEXT NOT NULL)";

    String Especialidad="CREATE TABLE ESPECIALIDAD(id_esp INTEGER PRIMARY KEY
NOT NULL, Especialidad TEXT NOT NULL)";

    String Laboratorio="CREATE TABLE LABORATORIO(id_Lab INTEGER PRIMARY KEY
NOT NULL, nombreLab TEXT NOT NULL," +

        "ubicacion TEXT NOT NULL, cap_est INTEGER NOT NULL, idUsuario
INTEGER NOT NULL, id_esp INTEGER NOT NULL," +

        "FOREIGN KEY (idUsuario) REFERENCES USUARIO (idUsuario), FOREIGN
KEY (id_esp) REFERENCES Especialidad (id_esp))";

    String Periodo="CREATE TABLE PERIODO(clave_pe INTEGER PRIMARY KEY NOT NULL,
nom_pe TEXT NOT NULL)";

    String Grupo="CREATE TABLE GRUPO(clave_grupo INTEGER PRIMARY KEY NOT NULL,
nombre_gru TEXT NOT NULL," +

        "id_esp INTEGER NOT NULL, FOREIGN KEY(id_esp) REFERENCES
Especialidad (id_esp))";
```

```

String Materiales="CREATE TABLE MATERIALES(idMAT TEXT PRIMARY KEY NOT
NULL, nombre_Mat TEXT NOT NULL," +
    "descripcion TEXT NOT NULL)";

String Asignar_Material="CREATE TABLE Asignar_Material(id_ad TEXT PRIMARY
KEY NOT NULL, clave_pe INTEGER NOT NULL, idMat TEXT NOT NULL," +
    "id_Lab INTEGER NOT NULL, FOREIGN KEY (clave_pe) REFERENCES
Periodo(clave_pe)," +
    "FOREIGN KEY (idMat) REFERENCES Materiales (idMat), FOREIGN KEY
(id_Lab) REFERENTES" +
    "Laboratorio (id_Lab)";

String Dias="CREATE TABLE DIAS(clave_dias INTEGER NOT NULL, dias TEXT NOT
NULL)";

String Crear_Horario=("id_Hor TEXT PRIMARY KEY NOT NULL, clave_pe INTEGER
NOT NULL,id_Lab INTEGER NOT NULL," +
    "clave_grupo INTEGER NOT NULL, clave_dias INTEGER NOT NULL,
hora_inicio TEXT NOT NULL," +
    "hora_fin TEXT NOT NULL, FOREIGN KEY (clave_pe) REFERENCES
Periodo(clave_pe)," +
    "FOREIGN KEY (id_Lab) REFERENCES Laboratorio(id_Lab), FOREIGN KEY
(clave_grupo)" +
    "REFERENCES Grupo(clave_grupo), FOREIGN KEY (clave_dias)
REFERENCES dias(clave_dias)");

public BaseHelper(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}

@Override
public void onCreate(SQLiteDatabase db)

{
    db.execSQL(Usuario);
    db.execSQL(Especialidad);
    db.execSQL(Laboratorio);

```

```

        db.execSQL(Periodo);
        db.execSQL(Grupo);
        db.execSQL(Materiales);
        db.execSQL(Asignar_Material);
        db.execSQL(Dias);
        db.execSQL(Crear_Horario);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        db.execSQL("drop table if exist USUARIO");
        db.execSQL("drop table if exist ESPECIALIDAD");
        db.execSQL("drop table if exist Laboratorio");
        db.execSQL("drop table if exist Periodo");
        db.execSQL("drop table if exist Grupo");
        db.execSQL("drop table if exist Materiales");
        db.execSQL("drop table if exist Asignar_Material");
        db.execSQL("drop table if exist Dias");
        db.execSQL("drop table if exist Crear_Horario");
        onCreate(db);
    }
}

```

Anexo para encriptación

```
public static final String md5(final String s)
{
    final String MD5 = "MD5";
    try
    {
        MessageDigest digest = java.security.MessageDigest
            .getInstance(MD5);
        digest.update(s.getBytes());
        byte messageDigest[] = digest.digest();

        StringBuilder hexString = new StringBuilder();
        for (byte aMessageDigest : messageDigest)
        {
            String h = Integer.toHexString(0xFF & aMessageDigest);
            while (h.length() < 2)
                h = "0" + h;
            hexString.append(h);
        }
        return hexString.toString();

    } catch (NoSuchAlgorithmException e)
    {
        e.printStackTrace();
    }
    return "";
}
```

```
public String MD5(String md5)
{
    try
    {
        java.security.MessageDigest md =
        java.security.MessageDigest.getInstance("MD5");
        byte[] array = md.digest(md5.getBytes());
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < array.length; ++i)
        {
            sb.append(Integer.toHexString((array[i] & 0xFF) |
0x100).substring(1,3));
        }
        return sb.toString();
    } catch (java.security.NoSuchAlgorithmException e)
    {
    }
    return null;
}
```

BIBLIOGRAFÍA

- [1] Garbarini, R. W. “Implementación de un sistema de gestión de calidad y servicios en laboratorio universitario de Ingeniería en Sistemas de Información”. Universidad Tecnológica Nacional. Facultad Regional Buenos Aires , 2013
- [2] Balaguera, Y. D. "Metodologías ágiles en el desarrollo de aplicaciones". Universidad Pedagógica y Tecnológica de Colombia,2013.
- [3] URL: <http://inaoep.mx/> Página principal del Instituto Nacional de Astrofísica Óptica y Electrónica, en esta página se puede consultar información acerca del instituto, de las ofertas educativas
- [4] Roger S. Pressman “Ingeniería de software, un enfoque práctico. Séptima Edición University of Connecticut, 2010”