



*Universidad de Sancti Spiritus “JOSÉ MARTÍ PÉREZ”
Facultad De Ciencias Técnicas y Económicas*

*Aplicación ANDROID para gestionar la información de los
recursos tangibles del Departamento de Informática de la
Universidad de Sancti Spiritus.*

*TRABAJO DE DIPLOMA PARA OPTAR POR EL
TÍTULO DE INGENIERO EN INFORMÁTICA.*

Autor: Pablo Enrique Rocío Cruz

Tutor: Ing. Renides Leyva Ramírez

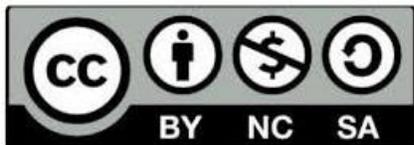
*Sancti Spiritus
2022*

Copyright©UNISS

Este documento es Propiedad Patrimonial de la Universidad de Sancti Spíritus “José Martí Pérez”, y se encuentra depositado en los fondos del Centro de Recursos para el Aprendizaje y la Investigación “Raúl Ferrer Pérez”, subordinado a la Dirección General de Desarrollo 3 de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información, contacte con:

Centro de Recursos para el Aprendizaje y la Investigación “Raúl Ferrer Pérez”. Comandante Manuel Fajardo s/n, esquina a Cuartel, Olivos 1. Sancti Spíritus. Cuba. CP. 60100

Teléfono: **41-334968**

Pensamiento

“(...) Un arquitecto, un ingeniero de cualquier especialidad que conozca la computación, puede multiplicar muchas veces su productividad y hacer las cosas con más exactitud y eficiencia (...)”

Fidel Castro

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título:
_____ y
reconocemos a la Universidad de Sancti Spíritus los derechos patrimoniales de la
misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días
del mes de _____ del año _____.

<nombre autor>

Firma del Autor

<nombre tutor>

Firma de Tutor

DEDICATORIA

A mis padres, abuelos y cada miembro de mi familia por confiar en mí y por su apoyo diario,

AGRADECIMIENTOS

A mis padres y abuelos por el apoyo a lo largo de la vida.

A mi tutor por toda la ayuda que me brindó y por todo lo que pude aprender de él.

A mi novia por su apoyo incondicional día a día.

Al colectivo de profesores de la carrera por su formación profesional.

A mis compañeros de estudio que tanto me han ayudado.

A todos aquellos que participaron de una forma o de otra en la realización de esta investigación.

A todos muchas gracias...

Resumen

En el mundo de hoy el desarrollo de nuevas tecnologías para los dispositivos móviles es un mercado que está en pleno auge. Dentro de ellas se encuentran las aplicaciones para el sistema operativo Android. La Universidad de Sancti Spíritus no queda fuera en el desarrollo de las nuevas tecnologías. Es por esto que en la facultad de Ciencias Técnicas y Económicas, en el Departamento de Ingeniería Informática hizo falta la realización de una herramienta que permitiera el control de los medios y recursos con los que cuenta el local. Por esta razón se hace indispensable el desarrollar una aplicación que permitiera de forma fácil y usando la novedosa tecnología de QR se pudiera inventariar todos los AFT, útiles y herramientas del departamento. Este trabajo se basa en desarrollar una aplicación para sistema operativo Android, con el propósito de solucionar las limitaciones que existían hasta el momento en el departamento para realizar los inventarios. Utilizando diferentes tecnologías como el ambiente de desarrollo integrado Android Studio, el lenguaje de programación Android, y sustentado bajo la metodología de desarrollo de software RUP.

Abstract

In today's world, the development of new technologies for mobile devices is a booming market. Among them are applications for the Android operating system. The University of Sancti Spíritus is not left out in the development of new technologies. That is why in the Faculty of Technical and Economic Sciences, in the Department of Informatics Engineering, it was necessary to develop a tool that would allow the control of the means and resources available in the premises. For this reason it is essential to develop an application that would allow easily and using the new QR technology to inventory all the AFT, tools and tools of the department. This work is based on developing an application for Android operating system, in order to solve the limitations that existed so far in the department to perform inventories. Using different technologies such as the integrated development environment Android Studio, the Android programming language, and supported under the RUP software development methodology.

Tabla de Contenido

Resumen.....	7
Abstract	8
Introducción:	1
CAPÍTULO I: FUNDAMENTOS TEÓRICOS Y METODOLÓGICOS DEL EMPLEO DE HERRAMIENTAS INFORMÁTICAS PARA LA GESTIÓN DE LA INFORMACIÓN PARA EL CONTROL DE LOS ACTIVOS FIJOS TANGIBLES Y LOS ÚTILES Y HERRAMIENTAS EN EL DEPARTAMENTO DE INGENIERÍA INFORMÁTICA DE LA UNISS.....	7
Introducción:	7
1.1 Proceso de gestión de la información en el departamento de ingeniería informática de la UNISS 7	
1.2 Aplicaciones Android.....	8
1.2.1 Ventajas de las aplicaciones Android	9
1.3 Arquitectura del Sistema Operativo Android.....	10
1.3.1 Componentes de software en el desarrollo de aplicaciones Android	13
1.3.2 Persistencia de datos en el desarrollo de aplicaciones Android	14
1.4 Metodologías utilizadas para el diseño.....	14
1.4.1 Metodologías ágiles	15
1.4.2 Metodología RUP y Ciclo de Vida	15
1.5 Lenguaje Unificado de Modelado (UML)	17
1.6 Herramientas útiles en el diseño e implementación de una aplicación Android	17
1.6.1 Android Studio.....	17
1.6.2 Herramientas para el diseño.	18
1.6.3 SQLite	20
1.6.4 Visual Paradigm.....	21
1.7 Conclusiones del capítulo.....	21
Capítulo II: DESCRIPCIÓN DEL SISTEMA PROPUESTO DE LA APLICACIÓN PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID PARA LA GESTIÓN DE LA INFORMACIÓN PARA EL CONTROL DE LOS ACTIVOS FIJOS TANGIBLES Y LOS UTILES Y HERRAMIENTAS EN EL DEPARTAMENTO DE INGENIERÍA INFORMÁTICA DE LA UNISS.....	22
2.1 Introducción	22
2.2.1 Reglas del negocio.....	23

2.2.2 Trabajadores del negocio	23
2.3.1 Casos de Uso del negocio	24
2.3.2 Descripción del Caso de Uso del Negocio	24
2.4 Modelo del Sistema.....	25
2.4.1 Diagrama de Actividades.....	26
2.4.2 Diagrama de clases del modelo de objeto.....	26
2.5 Requerimientos del sistema.....	27
2.5.1 Requisitos funcionales.....	27
2.5.2 Requisitos no funcionales	28
2.6 Modelo de Caso de Uso del Sistema	29
2.7 Análisis y Diseño del Sistema	32
2.7.1 Diagrama de clases del diseño	32
2.8 Diagrama de Colaboración	33
2.9 Diseño de la Base de Datos	34
2.9.1 Diagrama Entidad-Relación de la base de datos.....	34
2.9.2 Diagrama de clases persistente.....	35
2.9.3 Modelo de Datos Físicos	35
2.10 Conclusiones del capítulo.....	36
CAPÍTULO III: CONSTRUCCIÓN DEL SISTEMA PROPUESTO DE LA APLICACIÓN PARA DISPOSITIVOS MOVILES CON SISTEMA OPERATIVO ANDROID PARA AYUDAR A LA GESTIÓN DE LOS ACTIVOS FIJOS TANGIBLES Y LOS ÚTILES Y HERRAMIENTAS DEL DEPARTAMENTO DE INGIENERIA INFORMÁTICA DE LA UNISS	37
3.1 Introducción	37
3.2 Principios del Diseño	37
3.3 Modelo de Implementación.....	38
3.3.1 Diagrama de Despliegue.....	38
3.3.2 Diagrama de componentes	39
3.4 Estrategia de pruebas.....	39
3.4.1 Pruebas de Caja Negra	40
3.4.2 Ventajas y desventajas de las pruebas de Caja Negra	41
3.4.3 Realización de las pruebas de Caja Negra.....	42
3.5 Conclusiones del capítulo.....	45

Conclusiones	46
Recomendaciones	46
Referencias.....	47
Anexos.....	50

Índice de Tablas

<i>Tabla 1 Trabajador del negocio</i>	23
<i>Tabla 2 Caso de Uso del Negocio</i>	25
<i>Tabla 3 Caso de uso Gestionar Inventario</i>	30
<i>Tabla 4 Caso de uso Gestionar AFT</i>	30
<i>Tabla 5 Caso de uso Gestionar útil</i>	31
<i>Tabla 6 Prueba de Caja de Negra Para el CU Insertar Inventario</i>	43
<i>Tabla 7 Prueba de Caja de Negra Para el CU Insertar Inventario</i>	44

Índice de Figuras

<i>Figura 1 Arquitectura del Sistema Operativo Android.....</i>	10
<i>Figura 2 Caso de Uso del Negocio</i>	24
<i>Figura 3 Diagrama de actividades.....</i>	26
<i>Figura 4 Diagrama de clases del modelo de objeto</i>	27
<i>Figura 5 Modelo de Caso de Uso del Sistema</i>	29
<i>Figura 6 Diagrama de clases de diseño Insertar Inventario.....</i>	33
<i>Figura 7 Diagrama de clases de diseño Eliminar Inventario.....</i>	33
<i>Figura 8 Diagrama de Colaboración</i>	33
<i>Figura 9 Diagrama Entidad-Relación</i>	34
<i>Figura 10 Diagrama de clases persistente</i>	35
<i>Figura 11 Diseño Físico de la BD</i>	36
<i>Figura 12 Diagrama de Despliegue.....</i>	38
<i>Figura 13 Diagrama de Componentes</i>	39
<i>Figura 14 Inicio de la Aplicación Inventario</i>	45
<i>Figura 15 Acceso a Inventario dentro de la aplicación</i>	45

Introducción:

En el mundo que conocemos hoy, mediante el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), las organizaciones han optimizado sus habilidades, destrezas y conocimientos para la utilización de las mismas. Por ello, han implementado cambios en su estrategia laboral a la hora de enfrentar los retos que se les presentan. En este sentido, se utilizan nuevas técnicas para lograr el cumplimiento de los objetivos trazados por la persona que las emplee, lo cual está relacionado directamente con la calidad del talento humano.

Su función principal es facilitar el acceso a la información fácil y rápida en cualquier formato, esto es posible a través de la inmaterialidad; es decir de la digitalización de la información para almacenarla en grandes cantidades o tener acceso aún si está en dispositivos lejanos. (University, 2020)

En segundo lugar, la inmediatez; pues la información tiene la capacidad de ser compartida instantáneamente. Aunque la característica más importante es que permite la comunicación bidireccional entre varias personas. Con el desarrollo de la tecnología y la búsqueda de nuevas formas de comunicación más eficiente, las TIC se han posicionado como uno de los pilares básicos de la sociedad.

Cuba transita desde hace algunos años por un proceso que se ha definido como informatización de la sociedad: unos de los tres pilares que respalda la gestión gubernamental. Las acciones realizadas, si bien aún no alcanza la magnitud que demanda el desarrollo del país, han propiciado avances incuestionables en el gobierno y comercio electrónico.

El país, evoluciona y transita por un camino que conducen a perfeccionar y ampliar los procesos para obtener de ellos el máximo provecho. En ese camino la Universidad de Sancti Spíritus (UNISS) asume nuevos preceptos que la llevan hacia la transformación digital: un nuevo proyecto donde se integran las tecnologías digitales en todos los ámbitos universitarios.

El auge con que se presentan las TIC en el mundo de la enseñanza toca las puertas de las instituciones universitarias y sin pretender profesar fe ciega en las virtudes de la llamada sociedad del conocimiento es evidente que la informatización será uno de los retos que tendrán que asumir estas instituciones a fin de integrar las ventajas elocuentes que tienen estos dispositivos tecnológicos para el desarrollo de su misión y fines tanto educativos como sociales. (Díaz Lazo, Pérez Gutiérrez, & Florido Bacallao, 2011)

La UNISS sumergida en este proceso de informatización y desarrollo, ha logrado dar pasos firmes en la informatización de disímiles procesos, los cuales tienen como único objetivo mejorar y ayudar en el proceso docente y no docente. Un ejemplo de esto es la aplicación web “Moodle”, la cual permite la fácil comunicación y conexión entre alumnos y profesores.

El departamento de Ingeniería Informática, que pertenece a la facultad de Ciencias Técnicas y Económica de la UNISS no queda exento de este proceso de transformación digital, ya que, por ser el departamento de Ingeniería Informática, tiene que ser pionero en este proceso. El departamento cuenta ya con XII graduaciones, desde su fundación, ha crecido cada año en su matrícula y con esto el número de profesores, esto ha conllevado a que se complejice la gestión de la información administrativa; esta problemática ha llevado a buscar soluciones que resuelvan la contradicción que genera el aumento de la información y la manera de gestionarla de manera eficiente, rápida y segura, de digitalmente, mediante la utilización de un teléfono móvil, Tablet, otro dispositivo.

En medio de este proceso de informatización; los dispositivos móviles, juegan un rol fundamental ya que permiten la constante interrelación y comunicación entre personas, sociedad, empresas y los demás actores del mundo moderno. Es por esta razón que estos dispositivos móviles se han adaptado a nuestras vidas. Lo que ha provocado el surgimiento de numerosas aplicaciones orientadas al uso de la explotación de las capacidades de estos dispositivos.

Dentro de estas aplicaciones, destacan campos en los cuales ha habido aportaciones singulares que explotan las capacidades técnicas y el contexto de uso en los dispositivos móviles como son los códigos QR, los códigos QR (del inglés Quick Response code), son códigos bidimensionales también llamados “Código de Respuesta Rápida”, los cuales son capaces de contener información de distintos tipos y son legibles mediante una cámara y un software de lectura para código QR. Estos códigos gracias a su fácil manejo y su alta confiabilidad son usados con más frecuencia, estos permiten tener todos los objetos identificados de manera que podamos llevar un control eficaz. (Aguilar García , 2018)

Aspecto que se tomará en cuenta para la informatización de documentos, que es un concepto relativamente nuevo, que ha proliferado hace algunos años, y cada vez hace más evidente y necesario la inserción de este en nuestra sociedad, en nuestras empresas, en nuestras universidades, etc.

Hace algunos años, en los centros antes mencionados, realizar algo tan sencillo como un chequeo de inventario era un proceso largo y difícil, pero los sistemas informáticos nos ayudan a ahorrar tiempo y obtener de manera segura y organizada grandes volúmenes de información, facilitando las consultas de la misma, ya que estos están diseñados para ayudar y facilitar el trabajo de los usuarios.

En el departamento de Ingeniería Informática de la UNISS la gestión de la información administrativa tanto interna, como la suministrada a los niveles superiores presenta dificultades en la gestión de información de los Activos Fijos Tangibles y los Útiles y Herramientas las cuales se traducen en forma de manifestaciones, reveladas luego de la realización de una entrevista al jefe de departamento. La información antes mencionada se encuentra en formatos duro y en hojas Excel, por esto se tiene como resultado una demora ocasional en la entrega de información, dificultad para dar una respuesta inmediata a solicitudes de información, poco control de los medios básicos por tener falta de información y con esto extravíos de los mismos, duda en la veracidad de la información, falta de actualización de la información por ser un proceso difícil y engorroso, la información puede estar expensa cambios o duplicidad.

Teniendo en cuenta la problemática antes mencionada, se plantea el siguiente **problema a resolver**: ¿Cómo mejorar el proceso de control y gestión de los Activos Fijos Tangibles (AFT) y útiles y herramientas del departamento de Ingeniería Informática de la UNISS?

Es por ello que el **objeto de estudio** de la presente investigación se centra en el proceso de control y gestión de los Activos Fijos Tangibles y útiles y herramientas.

La investigación se enmarca en el **campo de acción** el proceso de control y gestión de los AFT y útiles y herramientas del departamento de Ingeniería Informática de la UNISS.

Se ha planteado como **objetivo general** desarrollar una aplicación para dispositivos móviles con sistema operativo Android que mediante el código QR agilice el proceso del control y gestión de los AFT y Útiles y Herramientas de forma segura y eficiente.

Para dar cumplimiento a este objetivo general se plantean las siguientes **preguntas de investigación**:

1. ¿Cuáles son los fundamentos teóricos y metodológicos que sustentan la elaboración de una aplicación para dispositivos móviles para la gestión de la información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS?
2. ¿Cómo diseñar una aplicación para dispositivos móviles con sistema operativo Android que perfeccione la gestión de la información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS?

¿Cómo implementar una aplicación para dispositivos móviles con sistema operativo Android que perfeccione la gestión de la información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS?

Para dar respuesta a estas preguntas de investigación de plantearon las siguientes **tareas de investigación**:

1. Determinar los fundamentos teóricos-metodológicos que sustentan el desarrollo de una aplicación para dispositivos móviles para la gestión de la

información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS.

2. Diseñar una aplicación para dispositivos móviles con sistema operativo Android para la gestión de la información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS.

Implementar una aplicación para dispositivos móviles con sistema operativo Android para la gestión de la información de los recursos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la UNISS.

Para apoyar el desarrollo de la investigación se emplean los siguientes **métodos científicos**:

Métodos Teóricos:

✓ Analítico Sintético: Se realizó un análisis de los documentos y la información referente a la forma en que se realiza la evaluación de desempeño en la gestión y control de documentos en el departamento de ingeniería informática de la UNISS, lo que permitió la identificación de las dificultades con que cuenta el proceso.

Métodos Empíricos:

Entrevista no estructurada: Se realizaron varias entrevistas abiertas al cliente, con el objetivo de obtener la información necesaria sobre el proceso de gestión y control de los AFT y medios básicos, lo que contribuyó a la definición de las funcionalidades necesarias para que el sistema a desarrollar sea oportuno ante este escenario en particular.

✓ Observación científica abierta: Permitted obtener conocimiento de cómo se realiza el proceso de gestión y control, para lograr su mejoramiento con el nuevo sistema. Para ello se tuvieron en cuenta los siguientes indicadores:

- Tiempo que demora la organización de la información contenida en formato duro y en formato digital

- Forma en la que se realiza el control de los AFT y de los medios básicos en el departamento de Ingeniería Informática de la UNISS.

El trabajo que se presenta está estructurado de la forma siguiente: introducción, tres capítulos, que incluyen todo lo relacionado con el trabajo investigativo, así como el diseño e implementación del sistema, conclusiones de cada capítulo y las generales, recomendaciones, bibliografía y anexos:

En el Capítulo I: En este capítulo se dan a conocer varios conceptos importantes que se deben tener en cuenta para comprender el objetivo del proyecto y se exponen los conceptos y algunas características de las herramientas que se utilizarán para la realización de la aplicación que se pretende realizar y así se comprenderá el porqué de su uso.

En el Capítulo II: En este capítulo se describe el sistema a desarrollar. Además de algunos modelos como: el modelo del negocio, reglas del negocio, modelo de requerimientos, modelo de clases de análisis, diagramas de casos de uso y el modelo de base de datos.

En el Capítulo III: En este último capítulo se expondrán datos referentes a la interfaz, seguridad y ayuda del sistema entre otros conceptos referentes a la aplicación ya implementada anteriormente. También se colocarán imágenes de la interfaz del producto final para que el cliente conozca de primera mano lo que va a consumir.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS Y METODOLÓGICOS DEL EMPLEO DE HERRAMIENTAS INFORMÁTICAS PARA LA GESTIÓN DE LA INFORMACIÓN DEL CONTROL DE ACTIVOS FIJOS TANGIBLES Y ÚTILES Y HERRAMIENTAS EN EL DEPARTAMENTO DE INGENIERÍA INFORMÁTICA DE LA UNISS.

Introducción:

En el presente capítulo se abordarán los principales conceptos y principios teóricos que sustentan la investigación, los cuales estarán avalados por referencias bibliográficas actualizadas y la investigación de los fundamentos teóricos sobre referentes a la Gestión de la información para el control de los activos fijos tangibles y los útiles y herramientas en el departamento de Ingeniería Informática de la Uniss, teniendo en cuenta los conceptos asociados al objeto de estudio y al campo de acción basados en el problema de esta investigación. También se explican detalladamente los aspectos a tener en cuenta para el desarrollo de la solución propuesta, la selección de las herramientas, librerías y metodología a utilizar.

1.1 Proceso de gestión de la información en el departamento de ingeniería informática de la UNISS

La gestión de información no es un fenómeno nuevo, fue algo que siempre le preocupó al hombre, desde el momento mismo en que se dio cuenta que aprovechar únicamente sus habilidades le limitaba con respecto a aquellos que habían logrado hacerse del conocimiento de las experiencias colectivas. (Pérez Rodríguez & Coutín Domínguez, 2005)

Sobre gestión de información, el Diccionario Ilustrado de la Lengua Española plantea: “La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades” (Diccionario ilustrado de la Lengua Española).

Se puede conceptualizar la gestión de información como un proceso mediatizado por un conjunto de actividades que permite la obtención de información, lo más pertinente,

relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización.

El concepto de gestión está basado en una mejora continua e ininterrumpida, en el que se diseñan estrategias que garantizan el perfeccionamiento permanente, que incluye a todos los recursos humanos (sean del nivel jerárquico que sean) de una organización. (Saunders Vázquez, 2020)

Dichas fuentes coinciden en que la gestión de la información es un ciclo inacabable donde se obtiene, se procesa, se aplica y se logra como resultado, una información enriquecida.

En la actualidad se utiliza con frecuencia el término de Gestión de la Información, por lo que hay que diferenciar la información para la gestión y la gestión de la información, que son dos conceptos diferentes. La información para la gestión es un tipo de información (los datos) y la gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. La información para la gestión es la información necesaria para tomar decisiones de gestión. (Alonso Arévalo, 2007)

Por tanto, la gestión de la información en una organización tiene como objetivo principal proveer a la entidad u organización la capacidad de ofrecer una información pertinente, inmediata, precisa y confiable, tanto en un nivel interno de la administración como en el suministro de la misma al exterior, a personas implicadas en el proceso.

Actualmente, se ha creado un desbalance entre los grandes volúmenes de información y la inmediatez que se requiere para su procesamiento, lo que impone a las organizaciones un nuevo reto, para el que sin dudas las tecnologías de la información y las comunicaciones emergen como un recurso a valorar para alcanzar el necesario equilibrio.

1.2 Aplicaciones Android

Investigando sobre sus orígenes no existe un criterio único aceptado por la comunidad tecnológica sobre el origen de las aplicaciones como tal. Sin embargo, se puede situar

en las primeras aplicaciones de video juego, tonos de llamadas, calendario y agendas implementados en los teléfonos celulares o móviles de segunda generación de los años 90. (Fuentes, 2018)

Las aplicaciones móviles o también llamadas aplicaciones telefónicas, es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, Tablet, y otros dispositivos móviles: Este tipo de aplicaciones permite al usuario efectuar un variado conjunto de tareas, ya sean profesional, de ocio, educativas, de acceso a los servicios, etcétera, facilitando la gestión de las actividades a desarrollar.

Las aplicaciones móviles pueden aprovechar mucho más el contexto en el que se ejecutaran, sobre todo si se comparan con las aplicaciones tradicionales. Ellos se deben a diferentes factores entre los que se encuentra las capacidades actuales en hardware de los dispositivos o la capacidad de acceder a la información del usuario a la que el propio dispositivo tiene acceso. (Fuentes, 2018)

La usabilidad de las aplicaciones móviles no parece ser un problema ya que estas son fáciles de usar, aumenta la productividad, reduce el costo y la curva de aprendizaje y mejora la satisfacción del usuario

1.2.1 Ventajas de las aplicaciones Android

Entre las ventajas más notables del sistema operativo Android se encuentran (<http://www.staffcreativa.pe>. [En línea], s.f.)

- Da completa libertad al propietario de un terminal a instalar cualquier aplicación en el formato APK.
- Google liberó la mayoría de su código bajo la licencia apache, una licencia libre y de código abierto.
- Cuenta con una de las comunidades más grande de desarrolladores.
- Dos tercios de sus aplicaciones son gratuitas y disponibles en su tienda oficial de aplicaciones: Google Play.
- Es completamente personalizable.
- Posee sistema de multitarea inteligente.

- Permiten un diseño responsive que se adapta a todo tipo de tamaños de pantalla y dispositivos: ordenadores, portátiles, tablets, teléfonos móviles, etc.

Entre sus desventajas se pueden mencionar:

- El acceso a las actualizaciones de cada versión, depende del fabricante.
- La mayoría de software malicioso creados para Linux están destinados a dispositivos con este sistema operativo.
- Los dispositivos con sistema operativo Android tienen entre sus variantes de establecer conexión con otros dispositivos el bluetooth.

Por todo lo expuesto anteriormente se aprecia que las aplicaciones Android son cada vez más usadas lo que ha facilitado la gestión de la información en el mundo moderno y en nuestro contexto nacional.

1.3 Arquitectura del Sistema Operativo Android

En las siguientes líneas se dará una visión global por capas de cuál es la arquitectura empleada en Android. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores, tal como muestra la siguiente figura 1. (Comunicaciones, 2021)

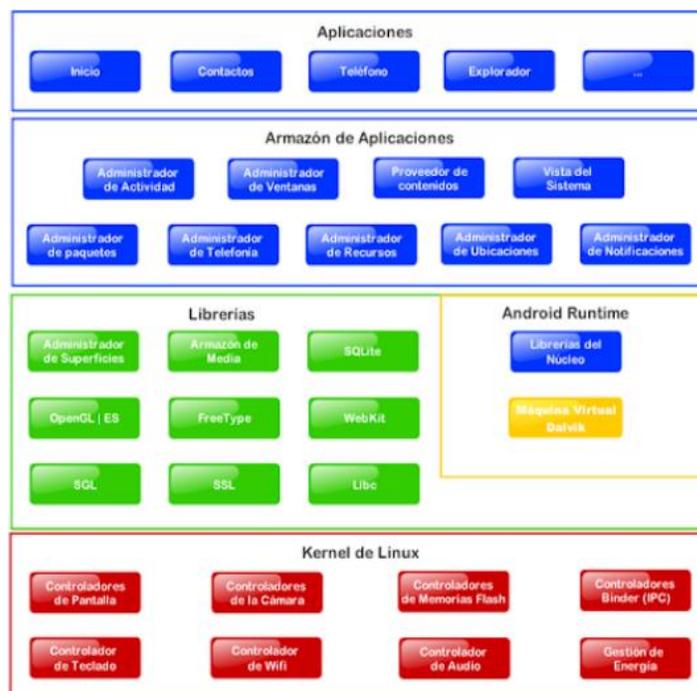


Figura 1 Arquitectura del Sistema Operativo Android

- **Aplicaciones:** Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.
- **Framework de Aplicaciones:** Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel.

Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **-Activity Manager:** Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.
- **-Window Manager:** Gestiona las ventanas de las aplicaciones y utiliza la librería Surface Manager.
- **Telephone Manager:** Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.).
- **Content Provider:** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta API la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- **-View System:** Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, "check-boxes", tamaño de ventanas, control de las interfaces mediante teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.
- **-Location Manager:** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- **-Notification Manager:** Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android

denominada Intent, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.

- ***XMPP Service***: Colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

Librerías: La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android. (Comunicaciones, 2021)

Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- ***Librería libc***: Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- ***Librería Surface Manager***: Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- ***OpenGL/SL y SGL***: Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. *OpenGL/SL* maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, *SGL* proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.
- ***Librería Media Libraries***: Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- ***FreeType***: Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- ***Librería SSL***: Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.

- **Librería *SQLite*:** Creación y gestión de bases de datos relacionales.
- **Librería *WebKit*:** Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Tiempo de ejecución de Android: Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Éste lo constituyen las *Core Libraries*, que son librerías con multitud de clases Java y la máquina virtual *Dalvik*.

Núcleo Linux: Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluye un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro de este *kernel* de Linux embebido en el propio Android.

1.3.1 Componentes de software en el desarrollo de aplicaciones Android

Como componentes de software importantes en la arquitectura Android se encuentran las actividades (*activities*), las vistas (*view*) y los *intents*. Las actividades representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otra aplicación.

Los objetos *view* son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los controles de Java o .NET. De inicio, Android pone a disposición de los desarrolladores una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

Un *intent* es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre

distintas aplicaciones. Mediante un *intent* se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc. (sgoliver, 2020)

1.3.2 Persistencia de datos en el desarrollo de aplicaciones Android

SQLite es un motor de bases de datos muy popular en la actualidad por ser de pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto por ser de código abierto. Android incorpora todas las herramientas necesarias para la creación y gestión de bases de datos SQLite, y entre ellas una completa API para llevar a cabo de manera sencilla todas las tareas necesarias.

En Android, la forma típica para crear, actualizar, y conectar con una base de datos SQLite se efectúa a través de una clase auxiliar llamada *SQLiteOpenHelper*, o para ser más exactos, de una clase propia que derive de ella y que debemos personalizar para adaptarnos a las necesidades concretas de nuestra aplicación.

1.4 Metodologías utilizadas para el diseño.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros.

Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software.

Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. (Montoya Suarez & Sepulveda Castaño, 2017)

1.4.1 Metodologías ágiles

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento).

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. (Montoya Suarez & Sepulveda Castaño, 2017)

1.4.2 Metodología RUP y Ciclo de Vida

La compañía norteamericana *Rational Software Corporation* creó, a mediados de 1998, el Proceso Unificado de Desarrollo (*Rational Unified Process* en inglés, habitualmente resumido como RUP). RUP es una metodología que tiene como objetivo ordenar y estructurar el desarrollo de software, en la cual se tienen un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema. RUP es además

un proceso basado en los modelos en Cascada y por Componentes, por lo cual es dirigido por los casos de uso, es centrado en la arquitectura, iterativo e incremental, lo que es fundamental para el proceso de desarrollo de software. (MÉNDEZ & MARGARITA, 2006)

Para Meza (2014, p.23), es un proceso de desarrollo de software que junto con el lenguaje UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Forma disciplinaria de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Su virtud principal es asegurar la producción software de alta calidad, apropiada a las necesidades del usuario final, dentro de un cronograma y un presupuesto predecible.

Además, Meza (2014 p.25), menciona las siguientes características importantes de la metodología RUP:

- ✓ Es un proceso iterativo e incremental, basado en el refinamiento sucesivo del sistema.
- ✓ Es un proceso controlado, donde juegan un papel de primordial importancia en la gestión de requisitos y el control de los cambios.
- ✓ Basado en la construcción de modelos visuales del sistema.
- ✓ Centrado en el desarrollo de la arquitectura, por lo que maneja el concepto de desarrollo basado en componentes.
- ✓ Soporta técnicas orientadas a objetos y en particular el uso UML. Incluye artefactos (que son los productos tangibles del proceso como, por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel, que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

La metodología RUP emplea el Lenguaje de Modelado Unificado (*Unified Modelling Language*, UML). UML es el lenguaje gráfico de modelaje estándar de la industria para especificar, visualizar, construir y documentar los artefactos de los sistemas de software. Permite escribir planos de software, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como

expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Ventajas de la Metodología RUP (Kruchten, 2004):

- Requiere conocimientos del proceso y de UML.
- Progreso visible en las etapas tempranas.
- El uso de iteraciones.
- Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño.

1.5 Lenguaje Unificado de Modelado (UML)

UML (Unified Modeling Language) fue adoptado como estándar del Object Management Group (Grupo Gestor de Objetos) en 1997 debido a que representa una colección de las mejores prácticas de ingeniería que han sido probadas con éxito en el modelado de sistemas. Es un lenguaje para la especificación, visualización, construcción y documentación de sistemas, no solo de software. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, pues ha sido diseñado para modelar cualquier tipo de soluciones informáticas, arquitectura o cualquier otra rama

1.6 Herramientas útiles en el diseño e implementación de una aplicación Android

Para el desarrollo de una aplicación para Android se hace necesario la utilización de múltiples herramientas que faciliten el proceso de desarrollo. A continuación, se describen algunas de ellas.

1.6.1 Android Studio

Android Studio es un entorno de desarrollo integrado para el sistema operativo Android lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno Eclipse, hasta ahora el IDE más utilizado. Al crear un nuevo proyecto en Android Studio, la estructura del proyecto aparece con

casi todos los archivos dentro del directorio SRC, un cambio a un sistema de generación basado *Gradle* que proporcionará una mayor flexibilidad para el proceso de construcción. Además, gracias a su sistema de emulación integrado, Android Studio permite ver los cambios que realizamos en nuestra aplicación en tiempo real, pudiendo además comprobar cómo se visualiza en diferentes dispositivos Android con distintas configuraciones y resoluciones de forma simultánea. (Android, s.f.)

Entre las muchas características de Android Studio destacan sus herramientas de empaquetado y etiquetado de código para organizarnos al implementar grandes cantidades de código, sirviéndose además de un sistema *drag & drop* para mover los componentes a través de la interfaz de usuario. Además, este nuevo entorno cuenta con *Google Cloud Messaging*, una funcionalidad con la que podremos enviar datos desde el servidor a terminales Android a través de la nube, siendo esta una forma de enviar notificaciones *Push* a nuestras apps. Por otro lado, también nos ayudará en la localización de nuestras aplicaciones, dándonos una forma más visual de seguir programando y controlar el flujo de la aplicación.

Android Studio utiliza Java, que es un lenguaje de desarrollo de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales. Incluye una combinación de características que lo hacen único y está siendo adoptado por multitud de fabricantes como herramienta básica para el desarrollo de aplicaciones comerciales de gran repercusión (Schildt, 2005).

¿Qué ofrece Android Studio?

- Un entorno de desarrollo en Java claro y robusto.
- Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- Asistentes y plantillas para los elementos comunes de programación en Android.
- Un completo editor con muchas herramientas extra para agilizar el desarrollo de nuestras aplicaciones.

1.6.2 Herramientas para el diseño.

En la actualidad las aplicaciones para lograr éxito deben ser portadoras de estética y en este sentido tanto el *Photoshop* como el *Material Design* constituyen dos herramientas indispensables para lograr efectos atractivos.

En el caso del *Material Design* este es un diseño donde la profundidad, las superficies, los bordes, las sombras y los colores juegan un papel principal. Recibe su nombre por estar basado en objetos materiales, es decir, son piezas colocadas en un espacio (lugar) y con un tiempo (movimiento) determinado.

Precisamente este diseño basado en objetos es una manera de intentar aproximarse a la realidad, algo que en un mundo donde todo es táctil y virtual es difícil. *Material Design* se guía por las leyes de la física, donde las animaciones son lógicas, los objetos se superponen, pero no puedan atravesarse el uno al otro.

La utilización de *Material Design* permite lo siguiente:

- Elementos ordenados e imágenes claras. Tiene un diseño con una tipografía clara, casillas bien ordenadas, colores e imágenes llamativos para no perder el foco y un sentido del orden y la jerarquía muy marcado. Estas ideas ya se aplican en muchos diseños, pero en *Material Design*, Google ha creado normas muy claras de cómo llevarlo a la práctica.

- Luz y sombras que dan sensación de jerarquía. Uno de los elementos claves es la luz y las sombras. Una iluminación realista proporciona indicios de cómo se comportará un elemento y en qué nivel se encuentra.

- El movimiento como forma de guiar al usuario.

El movimiento es otro elemento clave, ya que, al parpadear un objeto, no pretende otra cosa que llamar la atención; un elemento que se expande, significa que se acaba de abrir. Todos estos movimientos se crean en una dirección determinada, explicada en la guía de *Material Design*. Esto se traslada a *Android*, básicamente, delimitando con claridad el tipo de menú, los botones, los tipos de imágenes a elegir, así como sus animaciones (Vivas, 2015).

Con respecto al *Photoshop*, este programa ha dejado de ser una herramienta únicamente usada por diseñadores y fotógrafos, para ser empleada en multitud de disciplinas que trabajan los diseños web, composición de imágenes en mapa de bits, estilismo digital, fotocomposición, edición y grafismos de video, además cualquier otra actividad que requiera el tratamiento de imágenes digitales. Como parte de su uso extendido es ampliamente utilizado en el retoque y perfeccionamiento de las galerías de fotos contenidas en las aplicaciones (Margulis, 2005).

1.6.3 SQLite

Es un ligero motor de bases de datos de código abierto, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla. El sistema operativo Android, cuenta con SQLite como motor de base de datos por defecto (Sqlite-wince, s.f.).

A diferencia de otros Sistemas gestores de bases de datos como **MySQL**, **SQL Server** y **Oracle DB**, **SQLite** tiene las siguientes ventajas:

- **No requiere el soporte de un servidor:** SQLite no ejecuta un proceso para administrar la información, si no que implementa un conjunto de librerías encargadas de la gestión.
- **No necesita configuración:** Libera al programador de todo tipo de configuraciones de puertos, tamaños, ubicaciones, etc.
- **Usa un archivo para el esquema:** Crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad, ya que los datos de las aplicaciones Android no pueden ser accedidos por contextos externos.
- **Es de Código Abierto:** Esta disponible al dominio público de los desarrolladores al igual que sus archivos de compilación e instrucciones de escalabilidad.

Es por eso que **SQLite** es una tecnología cómoda para los dispositivos móviles. Su simplicidad, rapidez y usabilidad permiten un desarrollo muy amigable

En su versión 3, SQLite permite bases de datos de hasta 2 Tb de tamaño, y también permite la inclusión de campos tipo BLOB. Los BLOB (*Binary Large Objects*, objetos

binarios grandes), son elementos utilizados en las bases de datos para almacenar datos de gran tamaño generalmente imágenes u otros archivos multimedia.

1.6.4 Visual Paradigm

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: *Enterprise*, *Professional*, *Community*, *Standard*, *Modeler* y *Personal*. Existe una alternativa libre y gratuita de este software, la versión *Visual Paradigm UML 6.4 Community Edition*. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (*Visual Paradigm online*, s.f.).

1.7 Conclusiones del capítulo

La revisión de la bibliografía permitió profundizar en los distintos procesos de la gestión de la información en el departamento de Ingeniería Informática de la Universidad de Sancti Spíritus. Además, a partir de la bibliografía consultada se han podido identificar los elementos a utilizar en el desarrollo de aplicaciones para teléfonos inteligentes con sistema operativo Android, lo cual ha permitido discernir el tipo de aplicación a desarrollar, así como las herramientas indispensables para el diseño e implementación de la aplicación móvil como medio virtual para apoyar el proceso de control de los activos fijos tangibles y los útiles y herramientas del Departamento de Informática.

Capítulo II: DESCRIPCIÓN DEL SISTEMA PROPUESTO DE LA APLICACIÓN ANDROID PARA LA GESTIÓN DE LOS ACTIVOS FIJOS TANGIBLES Y LOS ÚTILES Y HERRAMIENTAS DEL DEPARTAMENTO DE INGENIERIA INFORMÁTICA DE LA UNISS.

2.1 Introducción

En el presente capítulo se modelan las ideas principales para el desarrollo de una aplicación para dispositivos móviles con sistema operativo Android, cuyo objetivo es el de ayudar y facilitar al control y la gestión de los AFT, Útiles y Herramientas en el Departamento de Ingeniería Informática de la UNISS. La metodología de desarrollo de software *Rational Unified Process* (RUP sus siglas en inglés) es la utilizada para dar la solución, la cual se lleva a cabo a través de diagramas. Para la realización de este modelo de negocio, se realiza un estudio del negocio identificando las reglas del negocio, los actores y trabajadores que intervienen, el diagrama de caso de uso del negocio, diagrama de actividades y el modelo de objetos, lo que permite una mejor comprensión de la problemática a tratar.

2.2 Modelado del Negocio

El Modelado del Negocio el cual se lo realiza con mayor énfasis en la fase conceptual de la metodología *Rational Unified Process* (RUP) tiene como objetivos comprender la estructura, la dinámica de la organización, problemas actuales, identificar posibles mejoras y comprender los procesos. Utiliza el Modelo de Casos de Uso del Negocio para describir los procesos del negocio y los clientes. (Alderson Luna, 2021)

El negocio a modelar gestiona los inventarios que ocurren dentro del Departamento de Ingeniería Informática de la UNISS. El proceso inicia cuando el actor del negocio, dígase el Jefe de Departamento o un trabajador designado del departamento decide realizar un conteo de AFT, útiles y herramientas dentro del local que ocupa en Departamento. Esto puede ocurrir por varios motivos, el primero por indicaciones de la administración de la facultad y del centro, o por un simple chequeo de los recursos con los que cuenta el departamento ya sea para un cierre por fin de año o por receso docente de vacaciones.

2.2.1 Reglas del negocio

Las reglas de negocio son reglas que están bajo la jurisdicción del negocio (Ross, 2011), es decir, es en el negocio que estas reglas pueden ser manipuladas por quienes actúan y tienen la autorización en él. Son criterios para guiar la actividad diaria de un negocio, para moldear las regulaciones judiciales operacionales, o para realizar decisiones operacionales del negocio. Los términos y hechos sobre los que se componen no están relacionados con el potencial sistema que pueda automatizarlas, sino solo con los conceptos del negocio.

Plantean otros autores que las reglas del negocio constituyen un conjunto de políticas o patrones a ser cumplidos con el fin de garantizar el adecuado funcionamiento del proceso del negocio. En el ambiente de trabajo asociado al sistema se han determinado las siguientes reglas del negocio (Nuñez Arcia & Boggiano Castillo, 2015):

Reglas del negocio a considerar:

- RN1: Revisar los AFT, Útiles y herramientas: El usuario debe revisar todos los AFT, Útiles y herramientas para ver las acciones que se tienen que realizar
- RN2: Realizar las acciones de gestión: El usuario le realiza todas las acciones correspondientes a los AFT, Útiles y herramientas en el tiempo establecido.

2.2.2 Trabajadores del negocio

Un trabajador del negocio representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades. (Mamani Rodríguez & La Serna Palomino, 2011)

En la tabla 1 se muestran el trabajador que interviene en el negocio antes descrito.

Tabla 1 Trabajador del negocio

Actor	Descripción
-------	-------------

Jefe de Departamento	Es el que lleva a cabo la actualización del registro del control de los medios dígame AFT o Útil en el departamento
----------------------	---

2.3.1 Casos de Uso del negocio

Un Caso de Uso del Negocio define qué debe ocurrir en el negocio cuando este se realiza, describe el comportamiento de una sucesión de acciones que produce un resultado de valor para un Actor particular del negocio. Es decir, un Caso de Uso del Negocio describe una secuencia de acciones realizadas en el negocio que produce un resultado de valor observable para un actor individual del negocio. Por tanto, desde la perspectiva de un actor individual, un caso de uso del negocio define el flujo de trabajo completo que produce los resultados deseados (Zamuriano, 2011).

Los diagramas de casos de uso del negocio constituyen una representación gráfica de un conjunto de elementos tales como actores y casos de uso, así como las relaciones y dependencias que se establecen entre ellos.

El caso de uso del negocio determinado es:

- Gestión de los Activos Fijos Tangibles y Útiles y Herramientas.

En la siguiente imagen se muestra el Caso de Uso que guiara el desarrollo de este trabajo.

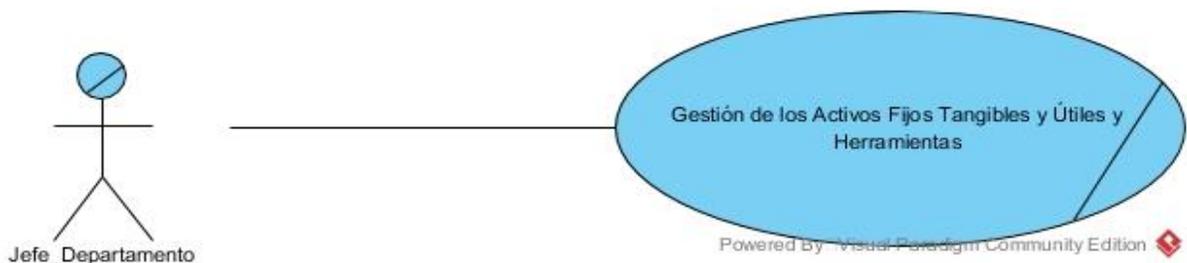


Figura 2 Caso de Uso del Negocio

;

En la tabla 2 se muestra la descripción del caso de uso del negocio.

Tabla 2 Caso de Uso del Negocio

Caso de Uso del Negocio		Gestión de los Activos Fijos Tangibles y Útiles y Herramientas
Actores	Jefe de Departamento	
Propósito	Consultar todos los AFT y los Útiles y Herramientas del Departamento	
Resumen El caso de uso comienza cuando el Jefe de Departamento necesita realizar un inventariado de los AFT y los Útiles para actualizar el Registro de Control de los Medios		
Casos de uso asociados		
Curso Normal de los eventos		
Acción del Actor		Respuesta del negocio
1. El Jefe de Departamento recibe el Registro de Control de los Medios		
		2. Realiza el levantamiento de los AFT y los Útiles y Herramienta del Departamento para actualizar el Registro de Control de los Medios

2.4 Modelo del Sistema

El Modelo del Sistema se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define como estos requisitos se cumplirán, la estructura que debe darse al sistema de software para que se haga realidad. Es una descripción del sistema muy cercana a la codificación. (Cristi'a, 2014)

2.4.1 Diagrama de Actividades

El Diagrama de Actividad es un diagrama de flujo del proceso multi-propósito que se usa para modelar el comportamiento del sistema. Los diagramas de actividad se pueden usar para modelar un Caso de Uso, o una clase, o un método complicado (Press., 2018).

En la figura 3 se muestra el diagrama de actividades correspondiente a la aplicación.

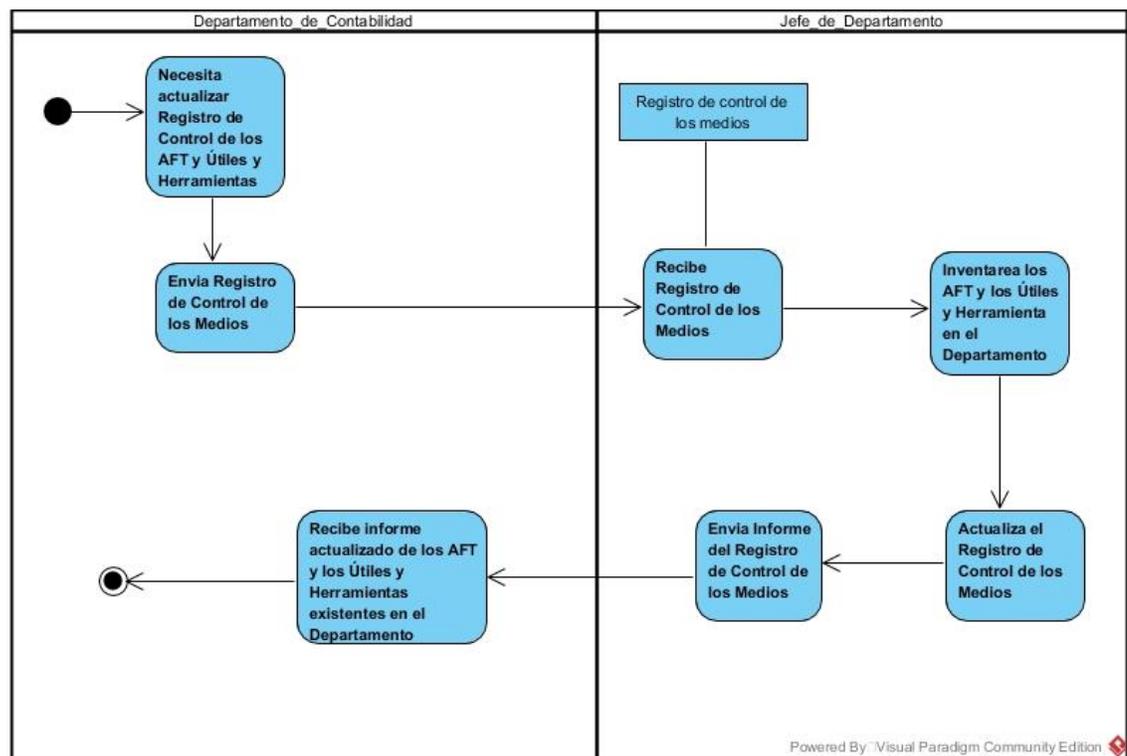


Figura 3 Diagrama de actividades

2.4.2 Diagrama de clases del modelo de objeto.

El modelo de objetos del negocio se utiliza para describir la participación de los trabajadores y entidades del negocio, y su colaboración en la realización del negocio. Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.

(Hernández González, 2015) En la siguiente figura (Fig. 4) se muestra el diagrama de clases del modelo de objeto de este trabajo.

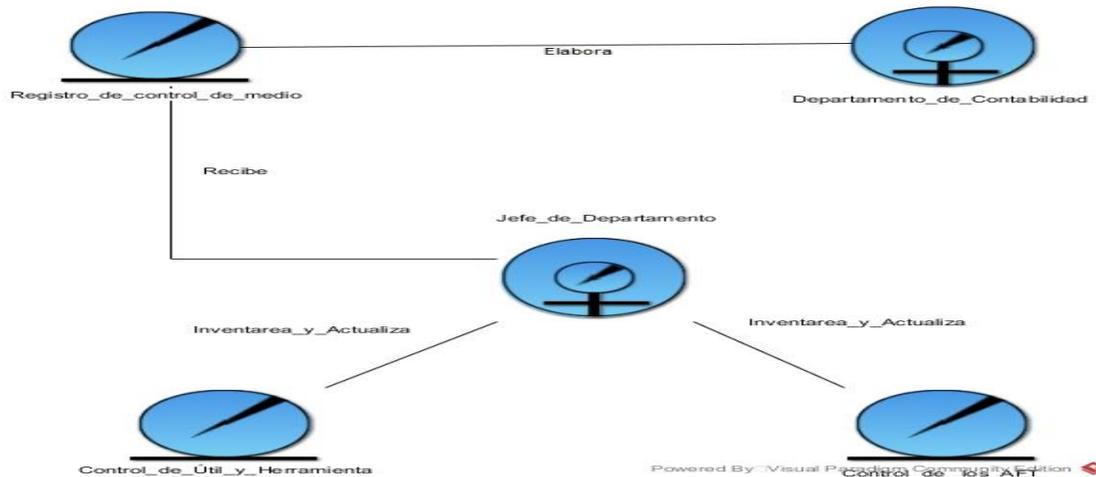


Figura 4 Diagrama de clases del modelo de objeto

2.5 Requerimientos del sistema

Esta disciplina toma el modelo de objetos de negocio y el modelo de caso de uso de negocio como entrada principal y trata de entender el problema que se quiere resolver por medio del sistema que se va a desarrollar. En el siguiente paso el analista junto con los interesados en el desarrollo del sistema define las necesidades de los clientes las cuales se plasman en lenguaje natural. (Hernández González, 2015)

2.5.1 Requisitos funcionales

Los requerimientos funcionales son los que se encargan de definir lo que la herramienta de software debe hacer. Definen los alcances del sistema en cuanto a las acciones que debe de realizar, y en cuanto a la transferencia de datos entre todas las diferentes funciones del sistema (Sommerville, 2015).

Posterior a un análisis con los clientes de la aplicación, se definieron los requisitos funcionales (RNF#) que debe cumplir la aplicación.

En específico la aplicación debe ser capaz de:

RF1: Leer y generar un código QR

RF2: Añadir la información sobre un nuevo código QR

RF3: Eliminar la información sobre un código QR

RF4: Consultar Información

RF4.1: Consultar Información técnica sobre los AFT, Útiles y Herramientas

RF4.2: Consultar Información sobre la cantidad de AFT, Útiles y Herramientas que tiene el departamento

RF4.3: Consultar en que fechas se hacen los registros y se realizan los movimientos de los AFT, Útiles y Herramientas

RF5: Ver galería de imágenes de los QR guardados.

2.5.2 Requisitos no funcionales

Los requerimientos no funcionales son aquellos que definen lo que la herramienta de software debe tener en cuanto a apariencia, sensación, operabilidad, y mantenimiento (Sommerville, 2015).

Los requisitos no funcionales (RNF) identificados para esta aplicación son:

RNF1: Usabilidad: La aplicación estará orientada para ser utilizada por diferentes tipos de usuarios. La usabilidad de esta aplicación se garantiza con un diseño sencillo que permita un uso intuitivo de la misma.

RNF2: Sistema Android: El dispositivo debe contar con el sistema operativo Android en su versión 5.0 o superior.

RNF3: Capacidad: El dispositivo móvil debe contar con un mínimo de 25 Mb de capacidad interna para almacenar la aplicación.

RNF4: Rendimiento: Al usuario hacer una determinada acción, el tiempo de respuesta de una gran cantidad de dispositivos de diversos modelos no sobrepasa los 2 segundos.

RNF5: Hardware: La solución alcanzada no requiere un hardware con grandes prestaciones. La aplicación funciona correctamente con los siguientes requisitos: dispositivos móviles con un mínimo de 512 Mb de memoria RAM y un microprocesador Dual Core o superior.

RNF6: Seguridad: Se hace necesario otorgarle permisos a la aplicación para que esta sea capaz de realizar las funciones de llamadas y permisos de almacenamiento del dispositivo.

2.6 Modelo de Caso de Uso del Sistema

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

El análisis efectuado sobre los requerimientos funcionales para la solución, permitió definir los casos de uso necesarios con el fin de satisfacer las necesidades del cliente. Además, este análisis permitió que cada caso de uso pudiera asociarse al actor que lo realiza. En la figura 5 se muestra el diagrama de caso de uso del sistema.

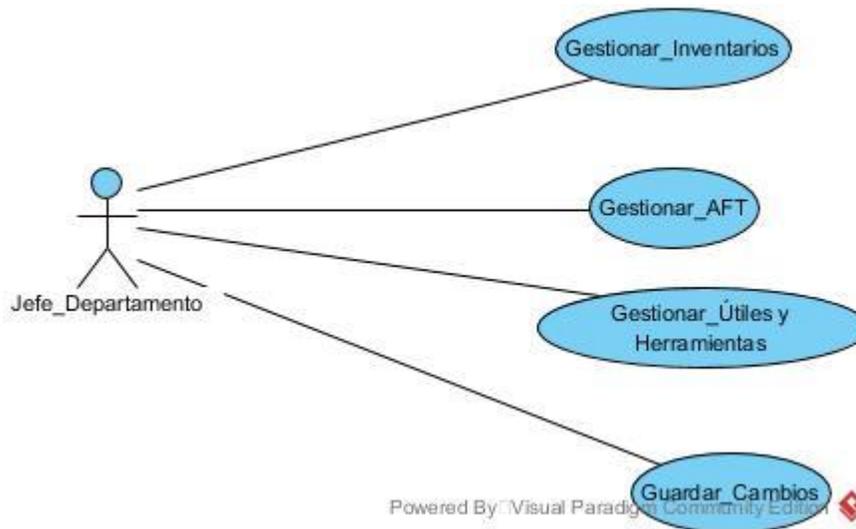


Figura 5 Modelo de Caso de Uso del Sistema

En las tablas 3,4 y 5 se muestran la descripción del caso de uso del sistema.

Tabla 3 Caso de uso Gestionar Inventario

Caso de Uso del Sistema		Gestionar Inventario
Actores	Jefe de Departamento	
Propósito	Insertar el inventario con sus datos para posteriormente realizar el levantamiento de AFT, útiles y herramientas.	
Resumen El caso de uso comienza cuando el jefe de departamento va a la aplicación y selecciona inventario y luego procede a añadir un inventario.		
Casos de uso asociados		Gestionar AFT Gestionar Útiles y Herramientas
Curso Normal de los eventos		
Acción del Actor		Respuesta del negocio
1. El jefe de departamento tiene que ir a la aplicación la acción añadir inventario.		
2. El jefe de departamento selecciona el botón de Agregar Inventario.		
		3. Se agrega el inventario en la base de datos de la aplicación.
Prioridad		Crítica.
Mejoras		

Tabla 4 Caso de uso Gestionar AFT

Caso de Uso del Sistema		Gestionar AFT
Actores	Jefe de Departamento	
Propósito	Insertar un AFT con sus datos para posteriormente realizar el levantamiento de AFT, útiles y herramientas.	

Resumen El caso de uso comienza cuando el jefe de departamento va a la aplicación y selecciona AFT y luego procede a añadir cada uno de los datos para este tipo de medio.	
Casos de uso asociados	
Curso Normal de los eventos	
Acción del Actor	Respuesta del negocio
4. El jefe de departamento tiene que ir a la aplicación y presionar la acción añadir AFT.	
5. El jefe de departamento selecciona el botón de Agregar AFT.	
	6. Se agrega un AFT en la base de datos de la aplicación.
Prioridad	Crítica.
Mejoras	

Tabla 5 Caso de uso Gestionar útiles

Caso de Uso del Sistema	Gestionar Útiles
Actores	Jefe de Departamento
Propósito	Insertar algún útil o herramienta del departamento para posteriormente inventararlo en la aplicación en los diferentes inventarios que se realicen.
Resumen El caso de uso comienza cuando el jefe de departamento va a la aplicación y selecciona Útil y luego procede a añadir una herramienta o útil del departamento.	
Casos de uso asociados	
Curso Normal de los eventos	
Acción del Actor	Respuesta del negocio

7. El jefe de departamento tiene que ir a la aplicación la acción añadir Útil.	
8. El jefe de departamento selecciona el botón de Agregar Útil.	
	9. Se agrega el inventario en la base de datos de la aplicación.
Prioridad	Crítica.
Mejoras	

2.7 Análisis y Diseño del Sistema

En el Análisis y Diseño, se transforman los requerimientos en un diseño de sistema, se desarrolla una arquitectura robusta y se adapta el diseño para concordar con el ambiente de implementación. Esta disciplina se caracteriza por su esfuerzo en el modelado, ya que se transforman los requerimientos en clases y relaciones, además se añaden las condiciones de los requerimientos no-funcionales, se diseña de la base de datos y se hace la identificación de componentes.

2.7.1 Diagrama de clases del diseño

Una clase de diseño es una abstracción de una clase o construcción similar en la implementación del sistema. Un diagrama de clases es una colección de elementos declaratorios del modelo, como clases, tipos y sus relaciones; conectados unos a otros y a sus contenidos en forma de grafo (Jacobson, Booch, & Rumbaugh, 2000)

A continuación, se muestran los diagramas de clases del diseño referente a los casos de uso Insertar Inventario y Eliminar Inventario (Fig. 6 y Fig. 7).

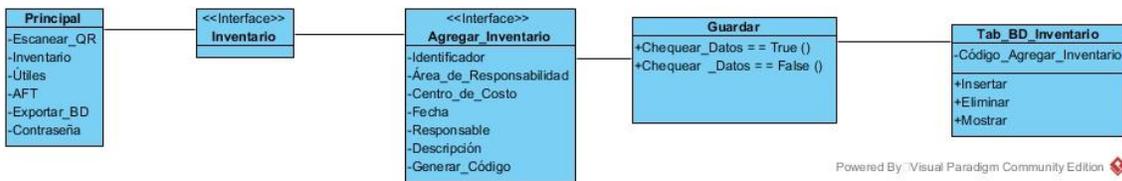


Figura 6 Diagrama de clases de diseño Insertar Inventario

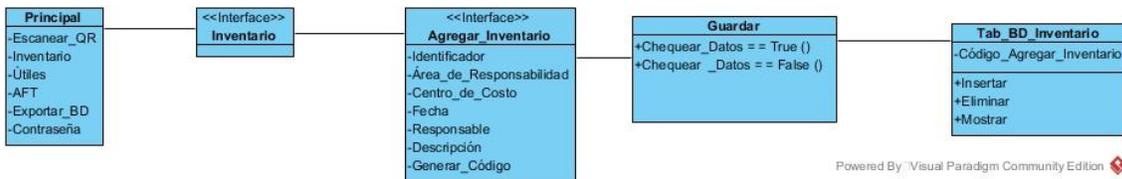


Figura 7 Diagrama de clases de diseño Eliminar Inventario

2.8 Diagrama de Colaboración

Un diagrama de colaboración describe las interacciones entre objetos y sus asociaciones, estas interacciones no se dan en un tiempo definido ni en un orden cronológico directo. En un diagrama de colaboración para que un objeto le mande un mensaje a otro, tiene que haber un “enlace” entre los dos, lo que significa que los objetos están asociados y por lo tanto que se pueden mandar mensajes entre ellos. (Gutiérrez Corea & Urrutia Zambrana, 2007)

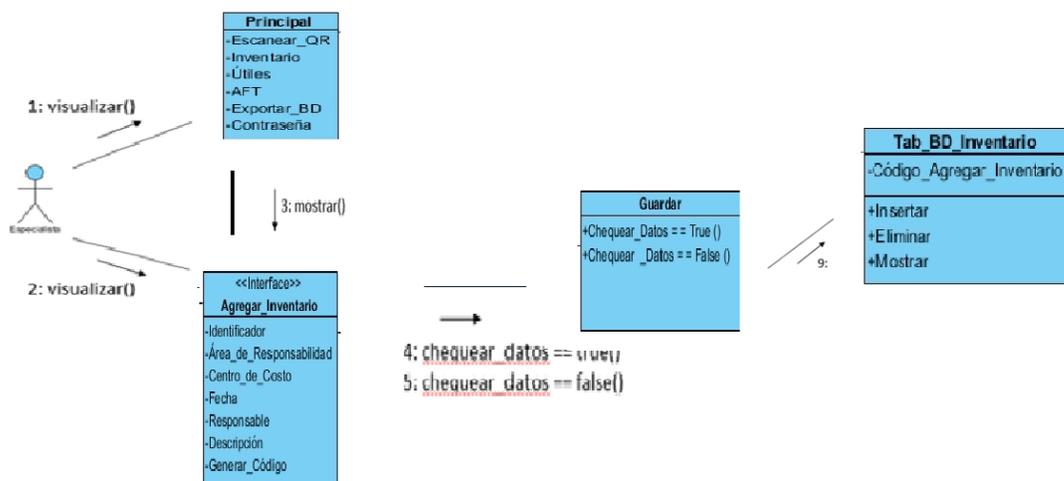


Figura 8 Diagrama de Colaboración

2.9 Diseño de la Base de Datos

El modelo conceptual de datos es una representación de un sistema, hecho de la composición de conceptos que se utilizan para ayudar a las personas a conocer, comprender o simular un tema que representa el modelo, incluye las entidades importantes y las relaciones entre ellos (Huertas López, 2022)

2.9.1 Diagrama Entidad-Relación de la base de datos.

La segunda fase del diseño de la base de datos se denomina diseño lógico de la base de datos, el cual tiene como resultado la creación de un esquema lógico de los datos para la parte de la empresa que queremos modelar. En la figura 9 se muestra el diagrama de entidad - relación de la base de datos propuesta.

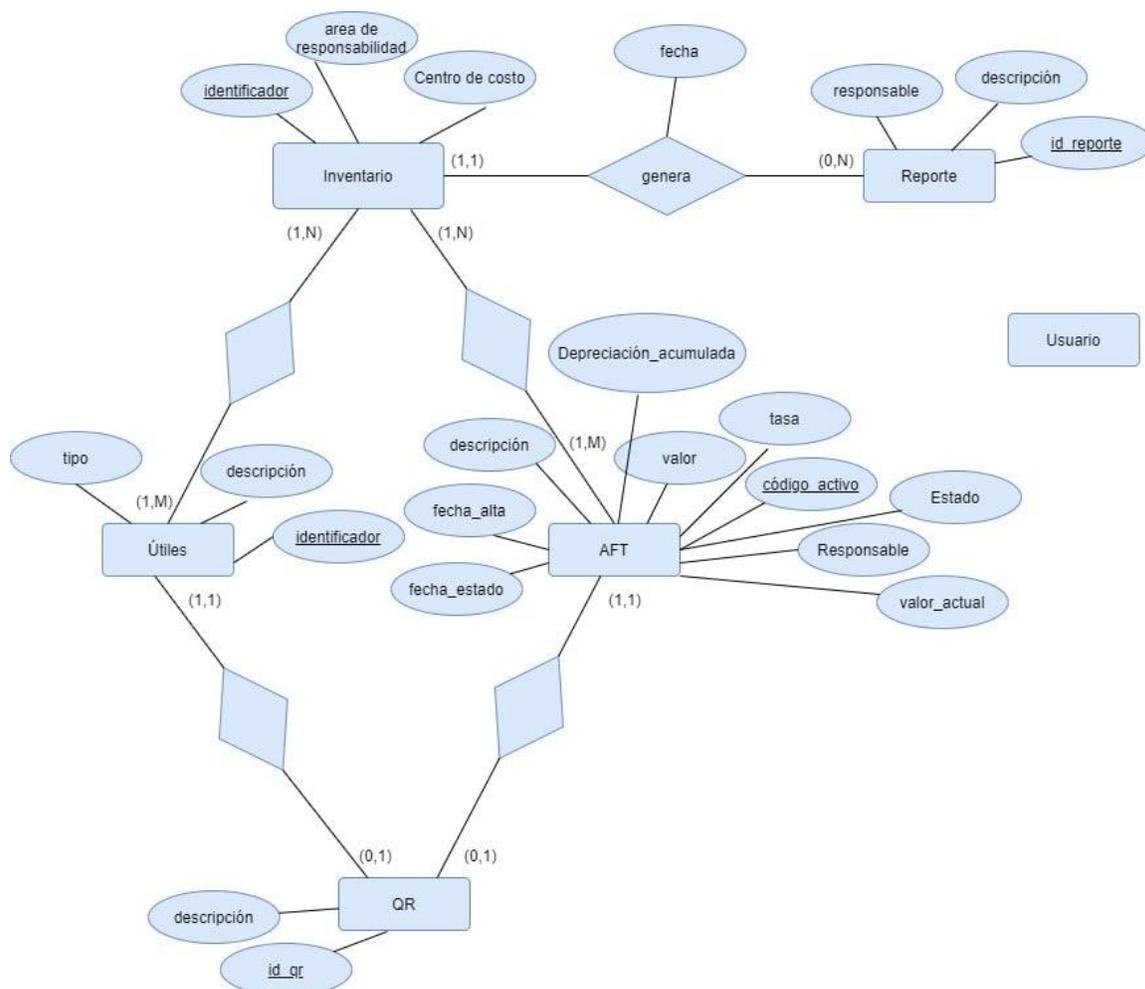


Figura 9 Diagrama Entidad-Relación

2.9.2 Diagrama de clases persistente

Las entidades de clases son usadas para representar las tablas de la base de datos. Son clases cuyos atributos de clases corresponden a las columnas de las tablas, usualmente el mapeo de cada campo de las tablas se realiza a través de anotaciones.

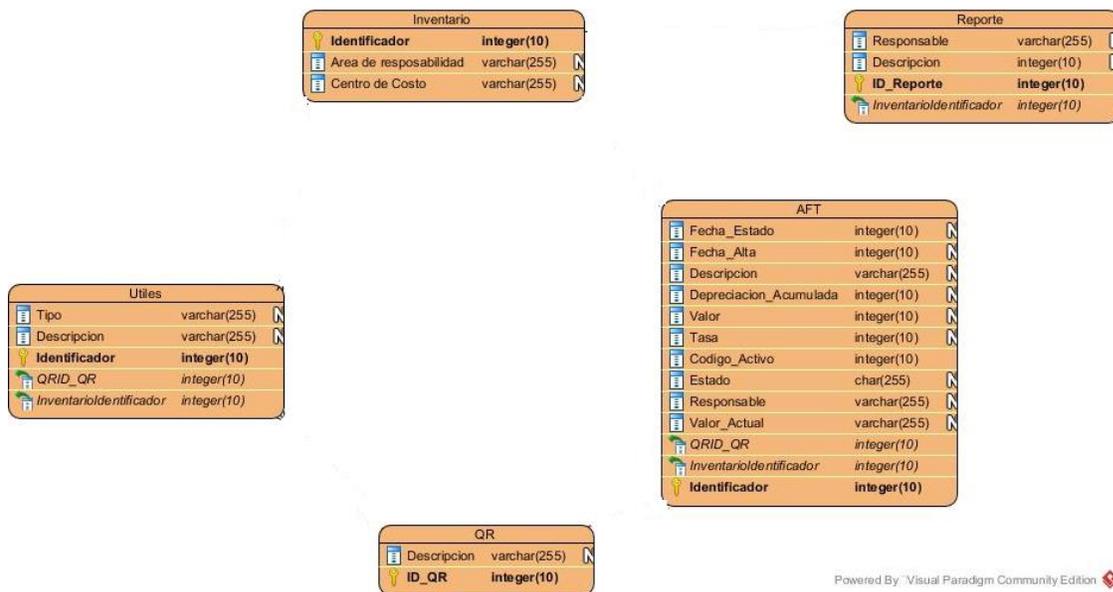


Figura 10 Diagrama de clases persistente

2.9.3 Modelo de Datos Físicos

El modelo de datos físicos representa como se construirá el modelo de base de datos. Un modelo de base de datos física muestra todas las estructuras de tablas, incluidas el nombre de columna, el tipo de datos de columna, la clave principal, la clave externa y las relaciones entre las tablas.

Entre las características de un modelo de datos físicos se incluyen:

- especificación de todas las tablas y columnas.
- las claves externas se usan para identificar relaciones entre tablas.
- la desnormalización puede ocurrir según los requisitos del usuario.

Las consideraciones físicas pueden hacer que el modelo de datos físicos sea bastante diferente del modelo de datos lógicos. En la figura 11 se muestra el modelo físico de la base de datos propuesta para el desarrollo de la aplicación

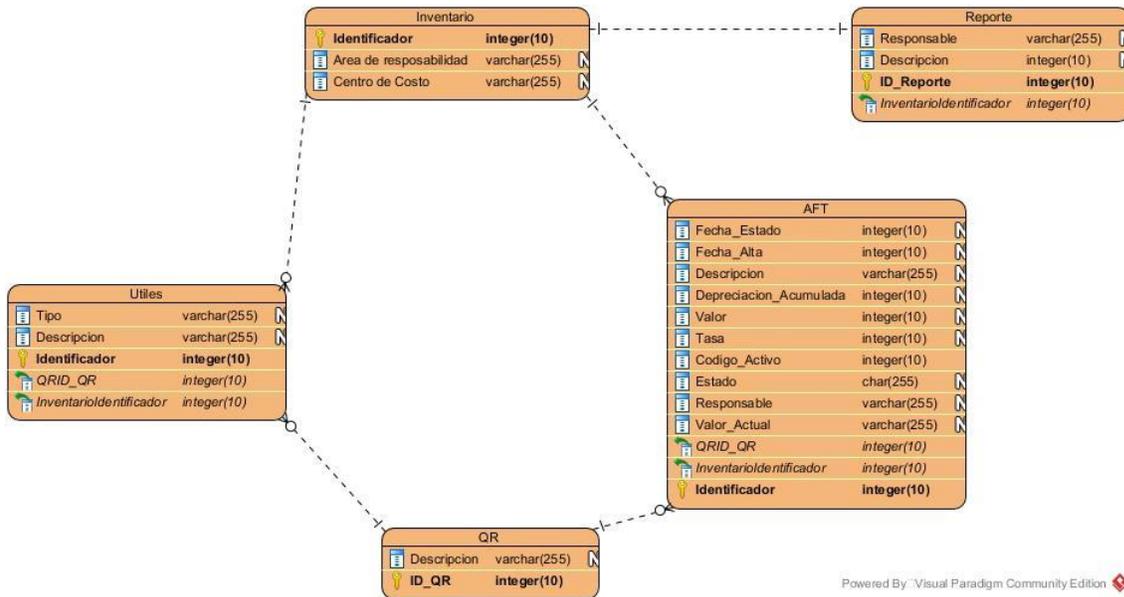


Figura 11 Diseño Físico de la BD

2.10 Conclusiones del capítulo.

La utilización de la metodología RUP permitió realizar el análisis y diseño de la aplicación para dispositivos móviles con sistema operativo Android “Inventarios”, a través del análisis realizado del modelo del negocio, se elaboró el diagrama de caso de uso del negocio. Luego, se especificaron también los requerimientos funcionales y no funcionales de la aplicación y se realizó una descripción general del modelo de sistema, identificando los actores y casos de uso y se generan otros artefactos importantes en el diseño de la aplicación. Además, se realizó el diseño de la base de datos que almacenará los datos persistentes de la aplicación. Esto permitió una adecuada descripción de las funcionalidades de la aplicación a implementar, así como la documentación de cada respuesta a las necesidades del cliente.

CAPÍTULO III: CONSTRUCCIÓN DEL SISTEMA PROPUESTO DE LA APLICACIÓN ANDROID PARA LA GESTIÓN DE LOS ACTIVOS FIJOS TANGIBLES Y LOS ÚTILES Y HERRAMIENTAS DEL DEPARTAMENTO DE INGIENERIA INFORMÁTICA DE LA UNISS.

2.1 Introducción

La parte del proceso de desarrollo de software cuyo principal propósito es decidir cómo se llevará a cabo el sistema, es el diseño. Durante esta etapa se toman decisiones importantes que conllevan al efectivo cumplimiento de los requerimientos funcionales y la obtención de un software con calidad. En el presente capítulo se plasman los resultados de la etapa del diseño. Se presentan diagramas como, por ejemplo: diagrama de despliegue, de componentes y de navegación de la aplicación. Además, se describen los principios de diseño aplicados.

3.2 Principios del Diseño

El diseño de la interfaz de una aplicación, el formato de los reportes, la concepción de la ayuda, el tratamiento de errores y la seguridad, tienen gran influencia en el éxito o fracaso de una aplicación. A continuación, se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión:

- **Diseño de la interfaz del sistema:** La interfaz de la aplicación se concibe con la mayor sencillez posible, de manera que el trabajo con el sistema sea fácil y ameno. Se utilizan colores agradables a la vista y una letra legible para su rápida comprensión, además se utiliza un lenguaje conocido por el usuario, sin emplear términos informáticos. El software consta de menús que donde aparecen de forma permanente todas las funcionalidades agrupadas de acuerdo a su naturaleza.
- **Tratamiento de errores:** El sistema propuesto valida constantemente la información, con el propósito de minimizar las posibilidades de introducir errores. En caso de que algún evento suceda se le comunica al usuario a través de un mensaje de alerta en un lenguaje fácil de comprender.

- **Concepción General de la ayuda:** El sistema cuenta con una ayuda capaz de explicar al usuario cómo es la manipulación y funcionamiento de la aplicación. El usuario podrá navegar por cada uno de ellos y consultarlo en cualquier momento ya que se encuentra ubicada en el menú principal de la aplicación.
- **Seguridad:** El sistema no mantiene un fuerte mecanismo de seguridad, cualquier usuario con un dispositivo móvil con sistema operativo Android podrá utilizarlo. Las personas encargadas de operar el sistema tendrán permiso de acceso a todas las funcionalidades de la aplicación.

3.3 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo del diseño, se implementan en términos de componentes, describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados y cómo dependen unos componentes de otros (Jacobson, Booch, & Rumbaugh, 2000).

2.3.1 Diagrama de Despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuyen las funcionalidades entre los nodos de cómputo (Jacobson, Booch, & Rumbaugh, 2000). A continuación, se muestra en la siguiente figura el diagrama de despliegue correspondiente al presente trabajo.

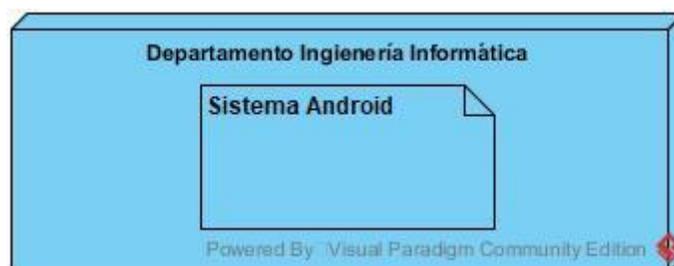


Figura 12 Diagrama de Despliegue

3.3.2 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Debido a que son más parecidos a los diagramas de casos de usos, estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes (SYSTEMS, 2015).

En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema. En la figura 13 se muestra el diagrama de componentes para la aplicación donde se muestran las bibliotecas utilizadas para el diseño de la aplicación.

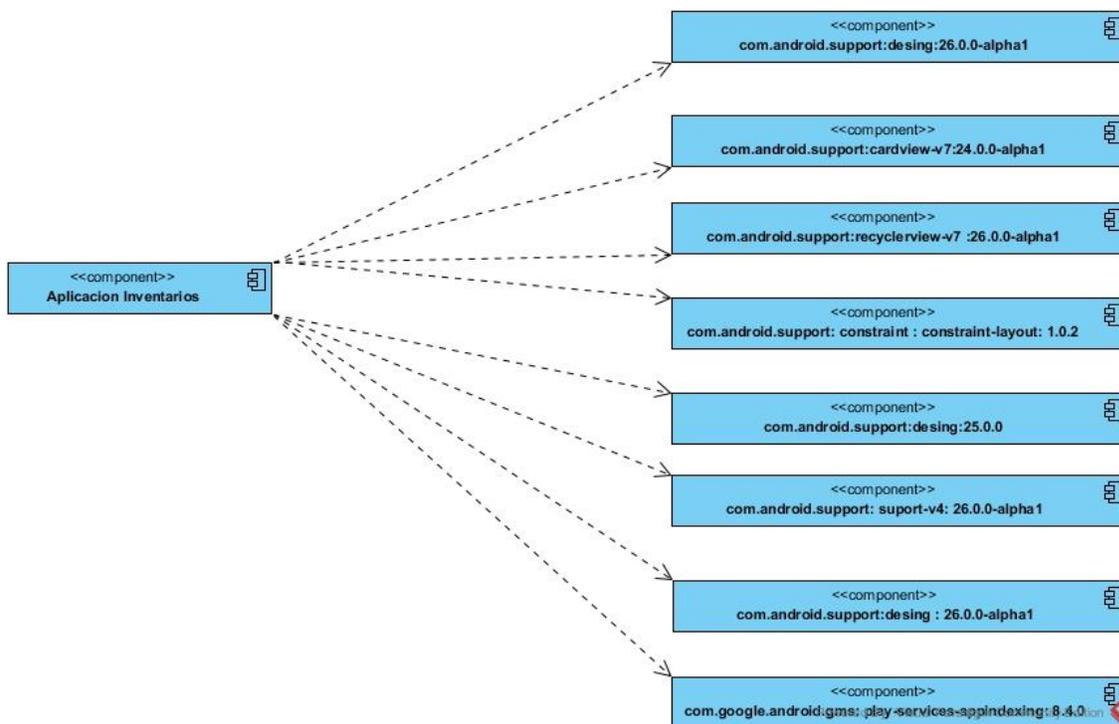


Figura 13 Diagrama de Componentes

2.4 Estrategia de pruebas

La fase de prueba es de vital importancia en el desarrollo de un software, ya que es la vía más factible para validar la información necesaria sobre la calidad del producto. El proceso de pruebas de la aplicación varía según el contexto de la misma. Por tanto, es

muy importante trazarse una estrategia de pruebas que corresponda con el producto desarrollado.

Cada metodología de pruebas tiene su propia definición para la estrategia y para el enfoque de pruebas. La estrategia de pruebas es la descripción de los diferentes niveles de pruebas y las pruebas que deben realizarse en cada uno de ellos en una organización o conjunto de proyectos. El enfoque de pruebas es la concreción de la estrategia de pruebas en un proyecto concreto. Este enfoque incluye las decisiones de prueba tomadas para alcanzar los objetivos del proyecto, el análisis de riesgos del producto, los criterios de entrada y salida, las técnicas de prueba seleccionadas, etc.

Es necesario aplicar una estrategia de pruebas ya que el tiempo disponible para probar es limitado. No se puede dedicar a todo el mismo esfuerzo de pruebas. Esto significa que hay que tomar decisiones (argumentadas) relacionadas con la profundidad de pruebas. La estrategia de pruebas también nos ayudara a utilizar nuestros recursos de pruebas de la forma más efectiva y eficiente posible a lo largo de todo el proyecto. La estrategia de pruebas se basa en los riesgos. Si el software entregado es muy crítico para el negocio, será necesario realizar pruebas intensivas. Por el contrario, si el sistema no desempeña un papel importante en el negocio, no será necesario realizar pruebas con demasiada profundidad ni cobertura. El objetivo de la estrategia de pruebas es determinar qué, cómo y cuándo (en qué nivel de pruebas) hay que probar para encontrar el mayor número de defectos posible con el mínimo coste y reduciendo al máximo el nivel de riesgo del producto.

2.4.1 Pruebas de Caja Negra

Para entregar un software de calidad, es necesario asegurar que cada uno de los componentes funciona correctamente, de tal manera que el usuario no se encuentre con sorpresas inesperadas al momento de utilizar la aplicación. Para esto existen pruebas que permiten al equipo de desarrollo asegurar que cada una de las funcionalidades ofrecidas cumpla debidamente con lo requerido. Las pruebas de caja negra se centran en el funcionamiento práctico del software y sus funcionalidades, obviando los detalles del código en particular. Las pruebas de caja negra son pruebas

funcionales dedicadas a «mirar» en el exterior de lo que se prueba. Estas pruebas se denominan de varias formas, pruebas de caja «opaca», pruebas de entrada/salida, pruebas inducidas por datos, etc. los sinónimos son muchos y muy variados (López, 2005).

Para realizar las pruebas de caja negra se limitan a que el probador intente con datos de entrada y estudie como salen, sin preocuparse de lo que ocurre en el interior. Principalmente, se centran en módulos de interfaz de usuario (pantalla, ficheros, canales de comunicación, etc.) pero suelen ser útiles en cualquier modulo ya que todos o la mayoría tienen datos de entrada y salida que se pueden comprobar y verificar. Como cualquier otra prueba, las de caja negra se apoyan y basan en la especificación de requisitos y documentación funcional, estos requisitos suelen ser más complejos que los internos.

2.4.2 Ventajas y desventajas de las pruebas de Caja Negra

Lo más deseable a la hora de realizar pruebas de caja negra es realizar una cobertura completa, pero, en la mayoría de los casos no es suficiente, siempre hay que combinarlas con pruebas de integración, ya que por mucho que funcionen los datos de entrada/salida, por dentro o en terceros sistemas, pueden existir defectos que no se están teniendo en cuenta. Estos defectos pueden no acarrear problemas a corto plazo, pero a lo largo del tiempo aparecerán. Por estas razones siempre se recomienda que se realicen todos los tipos de pruebas, tanto de caja negra como de integración, así como unas buenas pruebas de regresión y de compatibilidad. Siempre las pruebas funcionales tienen que estar completas y cubrir todas las funcionalidades posibles, solo así, se conseguirá una verdadera calidad del software y tus clientes podrán respirar tranquilos.

Entre los errores que permite encontrar las pruebas de caja negra se encuentran:

- Funciones incorrectas.
- Errores de interfaz.
- Errores en la base de datos externa o estructuras de datos.
- Problemas de rendimiento.

- Problemas de inicialización o terminación.

Pero las pruebas de caja negra no se realizan de forma arbitraria, sino que para seleccionar los casos a analizar existen diferentes criterios a tener en cuenta como son:

- Particiones de Equivalencia.
- Análisis de Valores Límite.
- Métodos Basados en Grafos.
- Pruebas de Comparación.
- Análisis Causa-Efecto.

3.4.3 Realización de las pruebas de Caja Negra

Con el objetivo de mejorar la experiencia del usuario final y, de identificar si existe algún error en la aplicación se ejecutan las pruebas de caja negra. La presente investigación describe una solución al proceso de prueba después de la implementación de los requisitos definidos para lo cual se realizan pruebas de interfaz utilizando el método de partición equivalencia. Resulta importante resaltar que para este tipo de pruebas no es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

- **Condiciones de ejecución:**
 1. La aplicación InventarioApp debe estar instalada en el dispositivo móvil con sistema operativo Android.
 2. El dispositivo debe contar con un mínimo de 25 Mb de memoria libre.
- **Caso de Uso: Insertar Inventario**

El jefe de departamento desea insertar un nuevo inventario en una fecha determinada para así poder comparar la hoja de inventario bajada desde la administración de la facultad o bien verificar que se encuentre en orden el registro de inventario de AFT, útiles y herramientas del departamento. Para ello, accede a la interfaz de usuario Inventario.

Leyenda de la Tabla de Prueba:

- V: indica valido.

- I: indica invalido.

En la tabla 6 se muestra la prueba de caja negra del caso de uso Insertar Inventario

Tabla 6 Prueba de Caja de Negra Para el CU Insertar Inventario

Id del escenario	Escenario	Variable 1 Permisos	Respuesta del Sistema	Resultado de la Prueba
EC 1	EC 1.1: El usuario desea insertar un inventario por primera vez.	V	El sistema chequea los permisos de escritura y guarda el inventario en la base de datos dentro del almacenamiento interno del teléfono y muestra la vista de los inventarios con el inventario ya insertado.	Prueba superada con éxito
		I	El sistema lanza una alerta notificando que tienes que insertar los valores adecuados en los campos que conforman el inventario.	
	EC 1.2: El usuario desea acceder nuevamente al inventario insertado.	V	El sistema comprueba que existe el inventario y lo muestra al usuario.	Prueba superada con éxito
		I	El sistema lanza una alerta notificando que no existe ningún inventario.	

Tabla 7 Prueba de Caja de Negra Para el CU Insertar Inventario

Id del escenario	Escenario	Variable 1 Permisos	Respuesta del Sistema	Resultado de la Prueba
EC 1	EC 1.1: El usuario desea insertar un AFT por primera vez.	V	El sistema chequea los permisos de escritura y guarda el AFT en la base de datos dentro del almacenamiento interno del teléfono y muestra la vista de los inventarios con el inventario ya insertado.	Prueba superada con éxito
		I	El sistema lanza una alerta notificando que tienes que insertar los valores adecuados en los campos que conforman el AFT.	
	EC 1.2: El usuario desea acceder Nuevamente al AFT insertado.	V	El sistema comprueba que existe el AFT y lo muestra al usuario.	Prueba superada con éxito
		I	El sistema lanza una alerta notificando que no existe ningún AFT.	

Algunas capturas realizadas durante la aplicación de las pruebas de caja negra

En las figuras 14 y 15 se muestran las capturas de pantallas realizadas a la aplicación InventarioApp durante el proceso de prueba. En la figura 14 se muestra la vista principal de la aplicación, en la que se puede apreciar el inventario, así como Reportes, Útiles,

AFT, Exportar BD y la sección para modificar la contraseña de los usuarios que acceden a la aplicación en cuestión. Una vez dentro de Inventario (Figura 15) se puede

visualizar todos los elementos que conforman la realización de un proceso de inventario en el departamento.



Figura 14 Inicio de la Aplicación Inventario



Figura 15 Acceso a Inventario dentro de la aplicación

3.5 Conclusiones del capítulo

Durante el desarrollo del presente capítulo se elaboró el diagrama de componente en el cual se graficó de manera clara las relaciones entre los subsistemas, los componentes y la base de datos; el diagrama de despliegue y el modelo conceptual y físico de la base de datos. Además, se describieron los principios de diseño seguidos, específicamente el diseño de la interfaz de usuario, los estándares de codificación, la concepción de la ayuda, el tratamiento de excepciones y la seguridad del sistema.

Además, se pudo profundizar en la diversidad de métodos existentes para la realización de pruebas a los productos de software, lo que permitió determinar las pruebas específicas que podrían ser utilizadas para la comprobación de la aplicación InventarioApp. Los resultados obtenidos de dichas pruebas han permitido comprobar que el producto cuenta con calidad y que a la vez resulta de muy fácil uso por parte de los usuarios.

Conclusiones

Finalizada la investigación y con el resultado del producto de software, se le da solución al problema planteado y cumplimiento a los objetivos propuesto para el presente trabajo de diploma. Al término del mismo, se concluye que:

1. Se identificaron los elementos a utilizar en el desarrollo de aplicaciones para dispositivos móviles con sistema operativo Android y las herramientas indispensables para el diseño e implementación de la aplicación para el apoyo en la Gestión y control de la información en el departamento de Ingeniería Informática de la UNISS.
2. Se utilizó la metodología de desarrollo RUP ya que resalta por su corto tiempo de duración, un equipo reducido y gran cantidad de información disponible y aceptación, por lo que se determinó que era la opción para este trabajo.
3. Las pruebas realizadas a la solución permitieron asegurar la calidad y el correcto funcionamiento de la aplicación cumpliendo con las expectativas del cliente.

Recomendaciones

Una vez finalizado el trabajo investigativo, donde se hace referencia a la utilización de las nuevas Tecnologías de la Información y las Comunicaciones desde el departamento de Informática de la Facultad de Ciencias Técnicas y Económicas de la Universidad de Sancti Spíritus José Martí se observa que la aplicación para dispositivos móviles con sistema operativo Android resultado de la misma se podría extender su uso hacia otros departamentos de la facultad y de la universidad, aun cuando haya que agregarle nuevos elementos a la misma para que su uso sea más general. Por tanto, se recomienda:

1. Ampliar su uso, y extenderlo hacia otros departamentos de la Facultad de Ciencias Técnicas y Empresariales, en busca de una mayor aceptación del software y de probar su utilidad en la gestión de la información referente a los inventarios de AFT, útiles y herramientas dentro de estos locales.

Referencias

1. Montoya Suarez, L., & Sepulveda Castaño, J. (10 de 2017). *Metodologías ágiles de desarrollo, uso y beneficio*. Obtenido de Metodologías ágiles de desarrollo, uso y beneficio: https://www.researchgate.net/publication/331530359_Capitulo_13_Metodologias_agiles_de_desarrollo_uso_y_beneficio
2. Huertas López, T. E. (2022). *Diseño de un modelo de gestión. Base científica y práctica para su elaboración*. Universidad y sociedad.
3. Aguilar García , T. (2018). *Que son los Códigos QR?* Obtenido de Que son los Códigos QR?: <https://docplayer.es/72039610-Que-son-los-codigos-qr.html>
4. Alderson Luna, O. (2021). *El modelado del negocio utilizando la metodología Rational Unified Process (RUP)*. Obtenido de El modelado del negocio utilizando la metodología Rational Unified Process (RUP): <https://www.gestiopolis.com/modelado-del-negocio-rational-unified-process-rup/>
5. Alonso Arévalo, J. (2007). *Gestión de la Información, gestión de*. Salamanca: Universidad de Salamanca.
6. Android, A. (s.f.). *Android Studio v1.0: características y comparativa con Eclipse*. Obtenido de Android Studio v1.0: características y comparativa con Eclipse: <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
7. Comunicaciones, S. d. (2021). *Arquitectura Android*. Obtenido de Arquitectura Android: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
8. Cristiá, M. (2014). *Introducción a la Ingeniería de Requerimientos*. Rosario, Argentina.
9. Díaz Lazo, J., Pérez Gutiérrez, M., & Florido Bacallao, D. (2011). IMPACT OF INFORMATION TECHNOLOGY AND COMMUNICATIONS (ICT) TO REDUCE THE DIGITAL DIVIDE IN TODAY'S SOCIETY. *Cultivos Tropicales*.
10. Fuentes, I. (13 de 5 de 2018). *Origen de las aplicaciones móviles*. Obtenido de Origen de las aplicaciones móviles: <https://11aplicacionesmoviles4amp.wordpress.com/2018/05/13/origen-de-las-aplicaciones-moviles/>

11. González, A. (2005). *Modelamiento del negocio*. Centro de Estudios de Ingeniería de Sistemas (CEIS).
12. Gutiérrez Corea, F. V., & Urrutia Zambrana, A. J. (2007). *Propuesta de sistema Geoinformático con representación de escenarios para auxiliar en la nueva metodología propuesta por funcionarios de INETER y la UNI para el estudio a gran escala de la vulnerabilidad y daños debido a sismos en las edificaciones*. Managua.
13. Hernández González, A. (2015). *Identificación del proceso de negocio*. La Habana: Red de revistas de America Latina.
14. <http://www.staffcreativa.pe>. [En línea]. (s.f.). Obtenido de <http://www.staffcreativa.pe>. [En línea]: <http://www.staffcreativa.pe/blog/android->
15. Jacobson, Booch, & Rumbaugh. (2000). *El proceso unificado de desarrollo*. Madrid: Addison Wesley.
16. Jacobson, I., Booch, g., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Addison Wesley.
17. López. (2005). *Pruebas de caja negra*.
18. Luna, A., & Celis, O. (2013, diciembre 19). Modelado del negocio Rational Unified Process (RUP). Obtenido de <https://www.gestiopolis.com/modelado-del-negocio-rational-unified-process-rup/>
19. Mamani Rodríguez, Z., & La Serna Palomino, N. (2011). Modelamiento del negocio desde el enfoque de un proceso de desarrollo de software. *REVISTA DE INVESTIGACION DE SISTEMAS E INFORMATICA*.
20. MÉNDEZ , N., & MARGARITA, E. (2006). *MODELO DE EVALUACIÓN DE METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE*. Caracas: UNIVERSIDAD CATÓLICA ANDRÉS BELLO.
21. MURILLO, R. P. (05 de Septiembre de 2019). ARQUITECTURA PERVASIVA CON TECNOLOGÍAS WebRTC HÍBRIDAS PARA EL DESARROLLO DE UN FRAMEWORK MODELO VISTA CONTROLADOR DE TIEMPO REAL. Puno, Perú: Repositorio Institucional UNA-PUNO.
22. Nuñez Arcia, Y., & Boggiano Castillo, M. B. (2015). *Generación automática de reglas de negocio en bases de datos para la implementación del sistema de información SIMCO*. Villa Clara: GECONTEC.

23. Pérez Rodríguez, L., & Coutín Domínguez, M. (2005). La gestión del conocimiento: un nuevo enfoque en la gestión empresarial. *ACIMED*.
24. Saunders Vázquez, A. (2020). *PROPUESTA DE BASES PARA EL DISEÑO DE UN SISTEMA DE GESTIÓN ESTRATÉGICA DE INFORMACIÓN PARA LA DIRECCIÓN DE ENERGÍA RENOVABLE DEL MINBAS*. Ciudad de la Habana: Enumet.
25. Schildt, H. (2005). *Java™: A Beginner's Guide*. New York: McGraw-Hill. New York.
26. sgoliver. (2020). *Componentes de una aplicación Android*. Obtenido de Componentes de una aplicación Android:
<https://www.sgoliver.net/blog/componentes-de-una-aplicacion-android/>
27. Sommerville, I. (2015). *Requerimientos del Software*.
28. *Sqlite-wince*. (s.f.). Obtenido de Sqlite-wince: <http://sqlite-wince.sourceforge.net>
29. SYSTEMS, S. (2015). *SPARX SYSTEMS*. Obtenido de <https://sparxsystems.com/resources/tutorials/uml2/component-diagram.html>
30. University, A. S. (9 de 7 de 2020). *Universidad Latina de Costa Rica*. Obtenido de ¿Qué son las TIC y para qué sirven?
31. *Visual Paradigm online*. (s.f.). Obtenido de Visual Paradigm online:
<https://online.visual-paradigm.com>

Anexos

