

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**Sistema de Control de Inventario móvil utilizando la  
Tecnología RFID**

**Caso: CODEROAD**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

**POSTULANTE:** Univ. Paul Wilker Landaeta Flores  
**TUTOR METODOLOGICO:** Lic. Freddy Miguel Toledo Paz  
**ASESOR:** Lic. Marcelo German Aruquipa Chambi

LA PAZ – BOLIVIA  
2017



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## **Dedicatoria**

Dedico el presente trabajo a las dos personas más importantes de mi vida que sin ellas este trabajo nunca se hubiera concluido.

A mi madre Reyna Flores quien me dio la vida y me inculco los valores del bien, por su infinito amor, el sacrificio y su ejemplo de fortaleza para luchar en la vida sin importar las circunstancias.

A mi hermana Fernanda Alvarez que me enseñó el valor de sonreír en los momentos malos y darme el apoyo constante durante todo este tiempo.

## **Agradecimientos**

Un trabajo de Proyecto de Grado, constituye un gran esfuerzo personal para alcanzar con éxito el objetivo propuesto, dicho trabajo no sería posible realizar sin el apoyo, ayuda, orientación, consejos, sugerencias, conocimientos y experiencias de otras personas.

Un agradecimiento muy especial a mi tutor metodológico, el Lic. Freddy Miguel Toledo Paz, por haberme brindado su apoyo y conocimiento, su experiencia y consejos sin los cuales no hubiera sido posible la culminación del trabajo.

De la misma manera un agradecimiento muy sincero a mi Asesor el Lic. Marcelo Aruquipa Chambi, por guiarme en las diferentes etapas de mi proyecto, por su asesoramiento, su colaboración y consejos, por sus observaciones y seguimiento constante, pero sobre todo por su paciencia y tiempo necesario para poder concluir este trabajo.

A la empresa CODEROAD SRL. Quienes me abrieron las puertas brindándome la confianza la elaboración del proyecto, el apoyo constante que me brindaron y la disponibilidad de tiempo durante la elaboración del proyecto.

Agradezco a mi familia, pero sobre todo a mi madre Reyna Flores y mi hermana Fernanda Alvarez por toda la colaboración en todas las etapas de mi vida.

A mis amigos y amigas, pero sobre todo a quienes me brindaron su amor y cariño, por no dejarme caer, por aceptarme como soy y así permitirme ser parte de sus vidas, por ayudarme a ser más fuerte, por enseñarme a ser mejor cada día, por darme la bendición de tenerlos en mi vida.

A todos muchas gracias, desde el fondo de mi corazón les deseo todo lo mejor, nunca olvidare, y que Dios los bendiga y los cuide siempre.

## Resumen

El presente proyecto es desarrollado para el área de tecnologías de información (IT) de la empresa CODEROAD SRL. con el objetivo de optimizar el control y la administración de los objetos informáticos de dicha empresa.

En el área de IT de la empresa CODEROAD SRL. se pudo observar varios problemas ya que la empresa se encontraba haciendo el control del inventario de manera manual usando códigos de barras para etiquetar los objetos para luego guardar la información en una hoja de cálculo la cual limitaba saber dónde se encontraban los objeto si estos llegan a moverse, el tiempo de lectura de los objetos era demasiado largo. Por lo tanto, todo este proceso se hacía de manera manual lo que provocaba que el tiempo de etiquetado de un objeto persona tomaba más de 2 minutos.

Para dar solución a los problemas existentes se desarrolla el presente proyecto de grado destinado a dar seguimiento y controlar el inventario de los objetos que entran dentro del área de IT, para tal objetivo se hace el uso de una aplicación móvil utilizando la tecnología RFID, otorgando información oportuna sobre el estado de los materiales y mecanismos mediante los cuales se puede solicitar, ingresar y administrar los objetos.

Para el desarrollo de la aplicación móvil se utiliza la metodología Mobile-D mediante sus diferentes fases ayudando a organizar y optimizar los procesos de planeación y desarrollo de la aplicación.

Una vez realizado el sistema se realizaron las pruebas de funcionamiento, en las cuales se puede constatar que la aplicación responde a los requerimientos de la institución ayudando a dar seguimiento al control de materiales otorgando información inmediata y confiable.

## **Abstract**

The current project was developed for the information technologies (IT) area of the CODEROAD SRL company with the main goal of improve and optimize the control and administration of all the computer objects into the company.

In the area of IT of the CODEROAD SRL company several problems could be observed since the company was controlling the inventory manually using Barcodes to tag the objects and then save the information in a spreadsheet which limited knowledge where the objects were the object are located and the reading time of objects was too long. Therefore, this whole process was done manually which caused the tagging time of a person object to take more than 2 minutes.

In order to solve the existing problems, the present degree project is developed to control the inventory of the objects that into within the area of IT, for that purpose We use of a mobile application using RFID technology, this application give us information on the state of materials and mechanisms through which objects can be requested, entered and managed.

For the development of the mobile application we use Mobile-D methodology, with its different phases, helping to organize and optimize the planning and development processes of the application.

Once the system has been completed and the functional test was performed into the company we verified that all requirements of the company has been completed, it's helping to follow the control of materials by providing immediate and reliable information.

## Tabla de Contenidos

CAPITULO I MARCO INTRODUCTORIO.....	1
1.1 INTRODUCCION .....	1
1.2 ANTECEDENTES .....	2
1.2.1 ANTECEDENTES INSTITUCIONALES .....	2
1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES.....	3
1.3 PLANTEAMIENTO DEL PROBLEMA .....	4
1.3.1 PROBLEMA CENTRAL .....	4
1.3.2 PROBLEMAS SECUNDARIOS.....	5
1.4 DEFINICION DE OBJETIVOS .....	5
1.4.1 OBJETIVO GENERAL.....	5
1.4.2 OBJETIVOS ESPECIFICOS.....	5
1.5 JUSTIFICACION .....	6
1.5.1 JUSTIFICACION SOCIAL.....	6
1.5.2 JUSTIFICACION TECNICA .....	6
1.5.3 JUSTIFICACION ECONOMICA .....	6
1.6 ALCANCES Y LIMITES.....	7
1.6.1 ALCANCES .....	7
1.6.2 LIMITES.....	7
1.7 APORTES.....	7
CAPITULO II MARCO TEORICO .....	9
2.1 INTRODUCCION .....	9
2.2 METODOLOGIA AGIL MOBILE-D .....	9
2.2.1 EXPLORACION .....	11
2.2.2 INICIALIZACION .....	15
2.2.3 PRODUCCION.....	16
2.2.4 ESTABILIZACION.....	17
2.2.5 PRUEBAS.....	18

2.3 IONIC .....	19
2.3.1 ¿QUE ES IONIC? .....	19
2.3.2 TIPOS DE APLICACIONES MOVILES .....	20
2.3.3 INTERFAZ USUARIO .....	24
2.4 ANGULARJS .....	24
2.5 ANDROID .....	25
2.5.1 ARQUITECTURA .....	25
2.5.2 VERSIONES.....	28
2.6 TECNOLOGIA DE AUTOIDENTIFICACION Y RFID .....	28
2.6.1 TECNOLOGIA DE AUTOIDENTIFICACION .....	29
2.6.2 TECNOLOGIA RFID.....	29
2.6.3 MIDDLEWARE RFID .....	30
2.6.4 ETIQUETA RFID.....	31
2.6.5 ANTENA RFID.....	31
2.6.6 LECTOR RFID .....	33
2.6.7 TIPO SE SISTEMAS DE RADIO FRECUENCIA .....	34
2.7 USABILIDAD EN APLICACIONES MÓVILES .....	35
2.7.1 METRICAS OBJETIVAS .....	35
2.7.2 METRICAS SUBJETIVAS.....	37
2.8 TECNICAS DE INVENTARIADO .....	38
2.8.1 DEFINICION DE CONTROL DE INVENTARIOS .....	39
2.8.2 IMPORTANCIA DEL INVENTARIO .....	39
2.8.3 TECNICAS DE INVENTARIO .....	39
2.9 ESTIMACION DE COSTOS DEL PROYECTO .....	40
2.9.1 LAS MÉTRICAS ORIENTADAS AL TAMAÑO .....	40
2.9.2 METODO COCOMO II .....	41
2.10 HERRAMIENTAS TECNOLOGICAS.....	42
2.10.1 GITHUB .....	43
2.10.2 WEBSTORM.....	43
CAPITULO III MARCO APLICATIVO .....	44

3.1 INTRODUCCION .....	44
3.2 FASE DE EXPLORACION .....	45
3.2.1 AMBIENTE FISICO .....	45
3.2.2 AMBIENTE TECNICO.....	45
3.2.3 RECOPIACION DE REQUERIMIENTOS .....	45
3.2.4 CASOS DE USO .....	46
3.3 FASE DE INICIALIZACIÓN .....	53
3.3.1 CONFIGURACIÓN INICIAL DEL PROYECTO.....	53
3.3.2 PLANIFICACIÓN INICIAL .....	53
3.4 FASE DE PRODUCTO.....	55
3.4.1 ITERACION 1 .....	55
3.4.2 ITERACION 2 .....	57
3.4.3 ITERACION 3 .....	63
3.5 FASE DE ESTABILIZACION.....	68
CAPITULO IV METRICAS DE CALIDAD Y SEGURIDAD .....	69
4.1. CALIDAD DE SOFTWARE.....	69
4.1.1. INTRODUCCION .....	69
4.1.2. DEFINICION Y ESPECIFICACION DE REQUERIMIENTOS DE CALIDAD.....	69
4.2 SEGURIDAD .....	73
4.2.1 RIESGOS .....	73
4.3 SEGURIDAD DE ACCESO .....	75
4.3.1. LA AUTENTICACION.....	75
4.3.2 SEGURIDAD BASADA EN LA SESION .....	76
CAPITULO V ANALISIS COSTO BENEFICIO.....	77
5.1 INTRODUCCION .....	77
5.2 COCOMO II .....	77
5.2.1 PUNTO DE FUNCION .....	77
5.2.2 COSTOS DEL SOFTWARE DESARROLLADO.....	82
5.2.3 COSTOS DE LA ELABORACION DEL PROYECTO .....	85
CAPITULO VI CONCLUSIONES Y RECOMENDACIONES .....	86

6.1 CONCLUSIONES .....	86
5.2 RECOMENDACIONES.....	86
BIBLIOGRAFIA .....	88

## Lista de tablas

Tabla 2.1: Simbología UML Fuente: (Salinas, 2010).....	14
Tabla 2.2: Ventajas de las Aplicaciones Nativas Fuente: Elaboración Propia .....	21
Tabla 2.3: Desventajas de las Aplicaciones Nativas Fuente: Elaboración Propia.....	21
Tabla 2.4: Ventajas de las Aplicaciones web móviles. Fuente: Elaboración Propia .....	22
Tabla 2.5: Desventajas de las Aplicaciones web móviles Fuente: Elaboración Propia.....	22
Tabla 2.6: Ventajas de las Aplicaciones híbridas Fuente: Elaboración Propia.....	23
Tabla 2.7: Desventajas de las Aplicaciones híbridas Fuente: Elaboración Propia .....	23
Tabla 2.8: Ventajas AngularJS Fuente: Elaboración Propia.....	24
Tabla 2.9: Versiones Soportadas Fuente: Elaboración Propia.....	28
Tabla 2.10: Categoría de un proyecto en función de sus líneas de código Fuente: (Baik, 2001). 41	
Tabla 2.11: Coeficientes COCOMO II Fuente (Boehm , 1995).....	42
Tabla 3.1: Lista de Requerimientos Fuente: Elaboración Propia .....	46
Tabla 3.2: Caso de Uso Ingreso a la Aplicación Fuente: Elaboración Propia .....	48
Tabla 3.3: Caso de Uso Asociar Objeto Fuente: Elaboración Propia.....	49
Tabla 3.4: Caso de Uso Obtener Reporte de Objetos Dañados Fuente: Elaboración Propia.....	50
Tabla 3.5: Caso de Uso Obtener reporte de Personal Fuente: Elaboración Propia.....	51
Tabla 3.6: Caso de uso Identificar Objeto Fuente: Elaboración Propia.....	52
Tabla 3.7: Métodos de la Librería TSL Fuente: Elaboración Propia.....	58
Tabla 4.1: Rango de Evaluación Fuente: Elaboración Propia .....	70
Tabla 4.2: Métricas Objetivas Fuente: Elaboración Propia .....	71
Tabla 4.3 Métricas Subjetivas Fuente: Elaboración Propia .....	72
Tabla 5.1: Entradas de la Aplicación Fuente: Elaboración Propia .....	78
Tabla 5.2: Salidas de la Aplicación Fuente: Elaboración Propia.....	78
Tabla 5.3: Peticiones de Usuario Fuente: Elaboración Propia.....	79
Tabla 5.4 Archivos de la aplicación Fuente: Elaboración Propia.....	79
Tabla 5.5: Interfaces Parámetros de medición Fuente: Elaboración Propia .....	80
Tabla 5.6: Factor de Complejidad Fuente: Elaboración Propia.....	81
Tabla 5.7: Factor de Ajuste Fuente: Elaboración Propia .....	82

Tabla 5.8: Punto de Función Fuente: Elaboración Propia .....	82
Tabla 5.9: Factor LCD/PF de lenguajes de programación Fuente: (QSM, 2017) .....	83
Tabla 5.10. Costos de elaboración del proyecto Fuente: Elaboración propia.....	85

## Lista de figuras

Figura 2.1: Estructura de la metodología Mobile-D Fuente: (Blanco, Camarero, Fumero, Werterski, & Rodriguez, 2009).....	10
Figura 2.2: Fase de Exploración Fuente: (Koskela, 2004).....	11
Figura 2.3: Proceso de Inicialización Fuente: (Koskela, 2004).....	15
Figura 2.4 Fase de Producción Fuente: (Koskela, 2004).....	16
Figura 2.5 Etapa de Estabilización Fuente: (Koskela, 2004).....	18
Figura 2.6. Etapa de Pruebas. Fuente: (Koskela, 2004).....	18
Figura 2.7 Estructura de Android Fuente: (Google, Google Developer, 2017).....	26
Figura 2.8: Antena directiva de alta densidad Fuente: (Clou, 2017).....	32
Figura 3.1: Proceso de Desarrollo Mobile-D Fuente: (Calle, 2014).....	44
Figura 3.2: Caso de Uso General Fuente: Elaboración Propia.....	47
Figura 3.3: Caso de uso Ingreso a la aplicación Fuente: Elaboración Propia.....	48
Figura 3.4: Caso de Uso Asociar Objeto Fuente: Elaboración Propia.....	49
Figura 3.5: Caso de Uso Obtener Reporte Objetos Dañados Fuente: Elaboración Propia.....	50
Figura 3.6: Caso de Uso Obtener Reporte Personal Fuente: Elaboración Propia.....	51
Figura 3.7: Caso de Uso Identificar Objeto Fuente: Elaboración Propia.....	52
Figura 3.8 Configuración del Proyecto Fuente: Elaboración Propia.....	54
Figura 3.9 Arquitectura del Proyecto Fuente: Elaboración Propia.....	54
Figura 3.10: Pantalla de Ingreso de la Aplicación Fuente: Elaboración Propia.....	56
Figura 3.11: Pantalla Final Ingreso de la Aplicación Fuente: Elaboración Propia.....	57
Figura 3.12: Pantalla de Tipo de Lectura Elaboración: Propia.....	59
Figura 3.13: Lista y selección de Usuario Fuente: Elaboración Propia.....	60
Figura 3.14: Pantalla de Asociación Fuente: Elaboración Propia.....	60
Figura 3.15: Pantalla Final de Selección de Reader Fuente: Elaboración Propia.....	61
Figura 3.16: Pantalla Final Selección de Perna Fuente: Elaboración Propia.....	62
Figura 3.17: Pantalla Final Confirmación de Asociación Fuente: Elaboración Propia.....	62
Figura 3.18: Diseño del Reporte de Depreciación Fuente: Elaboración Propia.....	64
Figura 3.19: Diseño de pantalla Información de Empleado Fuente: Elaboración Propia.....	64

Figura 3.20: Diseño de la Pantalla Reporte de Objetos Fuente: Elaboración Propia ..... 65

Figura 3.21: Diseño Final de la Pantalla Selección de Reportes Fuente: Elaboración Propia..... 66

Figura 3.22: Pantalla Final Reporte por Persona Fuente: Elaboración Propia ..... 66

Figura 3.23 Pantalla Final Reporte de Objetos Dañados Fuente: Elaboración Propia ..... 67

Figura 3.24: Pantalla Final Reporte de Depreciación de Objetos Fuente: Elaboración Propia .... 67

# CAPITULO I MARCO INTRODUCTORIO

## 1.1 INTRODUCCION

En los últimos años la tecnología de Identificación por Radiofrecuencia (RFID<sup>1</sup>) ha ido tomando relevancia, aplicándose en las diferentes áreas de la industria, sin embargo, la tecnología RFID no es una tecnología nueva, pues el inicio de esta tecnología fue desde el año 1948 (Stockman, 1948). En su forma básica la tecnología RFID está compuesta por un lector RFID y una etiqueta RFID, que es generalmente pequeña y se adhiere o incrusta dentro del objeto, animal o persona para su identificación.

Debido a la reducción de costos y a la facilidad de manipulación de la tecnología RFID las empresas han decidido incorporarla dentro de sus instalaciones, por ejemplo, la Agencia de Alimentos y Medicamentos (FDA<sup>2</sup>) recomienda a las compañías farmacéuticas hacer el uso de esta tecnología para controlar y prevenir la falsificación de medicamentos (FDA, 2016). Las granjas ya están haciendo el uso los tags<sup>3</sup> RFID ya que aprovechan que estos son sumamente pequeños y pueden ser utilizados encima o dentro de objetos pequeños o incluso animales, para poder tener así un control de la posición en la que se encuentran sus animales dentro del área de su granja y saber cuándo uno de ellos esté a punto de abandonarla. Además de hacer el seguimiento de los objetos también podemos hacer el uso de los tags para el control de inventario de los objetos de un lugar, este es el área donde más se está utilizando la tecnología RFID. Los supermercados incorporan en cada objeto de la tienda un tag RFID, donde es almacenado la información del objeto como

---

<sup>1</sup> RFID: Radio Frequency Identification.

<sup>2</sup> FDA: Food and Drug Administration.

<sup>3</sup> Tag: Etiqueta RFID.

por ejemplo precio, marca, peso, para que luego este sea leído por el lector y poder obtener la información de una manera automática, reconociendo cada característica del objeto además que las antenas de RFID reconocen en cada instante la posición en la que se encuentra el objeto.

En el presente proyecto de grado vamos a centrarnos en el control de inventarios y la búsqueda de objetos con el lector RFID para el área de IT de la empresa CODEROAD.

## **1.2 ANTECEDENTES**

### **1.2.1 ANTECEDENTES INSTITUCIONALES**

Coderoad es una empresa dedicada por más de 10 años al desarrollo de aplicaciones IoT<sup>4</sup> y Big Data<sup>5</sup> en la ciudad de La Paz, con una gran profundidad de antecedentes en el software Coderoad ofrece servicios para automotriz, salud, minoristas, petróleo, gas y muchos más.

Usando las tecnologías más actuales para el desarrollo de aplicaciones web, móviles e híbridas con un equipo de desarrollo que incluyen desarrolladores, Project managers, diseñadores y producto managers. Utilizan la metodología Ágil para dar solución a cada uno de los proyectos que cuenta.

Las instalaciones de la empresa cuentan con equipamiento de última generación el cual es otorgado a cada uno de sus trabajadores dependiendo el área en el que trabaja, como

---

<sup>4</sup> IoT: Internet en las cosas (internet of things).

<sup>5</sup>Big Data: Análisis de grandes cantidades de datos.

ser desktops, laptops, iPhone, Android, tabletas para que este pueda cumplir el trabajo que se le es asignado.

### **1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES**

Actualmente en la Universidad Mayor de San Andrés podemos encontrar las siguientes tesis relacionadas con el uso aplicativo de la tecnología RFID:

“Mecanismos de Personalización de etiquetas RFID con firma dinámica JetFD EPC” en el siguiente trabajo se propone una manera de encriptar la información dentro de la etiqueta RFID para que esta sea más segura y no pueda leerse la información de manera directa (Benitez, 2016).

“Diseño de un sistema identificador de placas de vehículos, utilizando tecnología RFID, Caso: Gobierno Autónomo Municipal de La Paz” El presente proyecto trata de un diseño de un sistema que identifique automáticamente las características más importantes de los automóviles al ingreso de las estaciones de servicio, dentro de estas características se encuentra la placa de control única de cada vehículo (Duran & Lozano, 2012).

“Registro y Control de activos fijos muebles en las unidades educativas, utilizando la tecnología RFID Caso: Gobierno Autónomo Municipal de La Paz- Unidad de Bienes Muebles” El presente trabajo propone la utilización de etiquetas RFID en cada mueble del establecimiento para poder luego realizar un control mediante el uso del reader para leer cada una de las etiquetas sin la necesidad de leer el código de barras y solamente usar la tecnología RFID (Quispe, Quispe, & Montecinos, 2015).

También podemos encontrar tesis relacionadas sobre el uso del RFID en otras universidades como por ejemplo.

“Formulación de proyectos para la implementación de sistemas de control de inventarios mediante tecnología RFID basado en PMI” presentada en la Universidad militar nueva granada de la Facultad de Ingeniería propone hacer el uso de la tecnología

RFID para el uso de inventarios usando la metodología PMI, para tal propósito cada objeto es etiquetado para luego ser escaneado y poder así generar el inventario de los objetos con los que cuenta (Pachon, 2015).

“Securing Assets with RFID” presentada en la universidad de Agder que propone solucionar al problema de pérdida de objetos de empresas que tienen una existencia de objetos masiva para el cual hacen el uso de la tecnología RFID y el código de producto (Chanrasekaram & Oustad, 2008).

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

En la actualidad el control de inventarios dentro de la empresa Coderoad se lleva a cabo usando la tecnología de Código de Barras, la cual es colocada en cada objeto que tiene la empresa. El código de Barras instalado en cada objeto es enlazado con las características del objeto, posición en la que se encuentra y la persona a la que fue asignada, toda esta información es guardada en un archivo para luego ser revisada cuando se necesita dicha información.

Este tipo de tecnología genera muchos problemas a la hora de hacer un control sobre el inventario dentro de la empresa. Por ejemplo, para obtener la información del código de barras debemos leer la etiqueta a una distancia corta y directa hacia el código, pero si esta etiqueta está dañada no podremos leerla correctamente, además si el objeto se encuentra a una distancia larga no podremos leerla.

La lectura de cada objeto debe ser uno por uno lo que provoca una pérdida de tiempo a la hora de inventariar una cantidad grande de objetos. Otro problema con el uso de la tecnología es no saber si un objeto fue llevado a otro lugar, este y los anteriores problemas pueden ser solucionados usando la tecnología RFID.

#### **1.3.1 PROBLEMA CENTRAL**

Por lo mencionado anteriormente podemos apreciar que la tecnología de código de barras aplicada en la empresa Coderoad tiene muchos problemas y limitantes a la hora de

hacer un control de inventarios, lo que provoca muchas veces la pérdida de información y tiempo, por lo cual nos surge la siguiente pregunta:

¿De qué manera podemos disminuir la pérdida de información y de tiempo a la hora de hacer un control de inventario utilizando la tecnología RFID?

### **1.3.2 PROBLEMAS SECUNDARIOS**

Los problemas secundarios identificados son los siguientes:

- ✓ Lectura de objetos uno por uno.
- ✓ No existe la forma de ver la información del objeto instantáneamente luego de ser escaneado.
- ✓ Se necesita apuntar directamente a la etiqueta de código de Barras para su lectura.
- ✓ No existe una forma de buscar una etiqueta de código de barras.

## **1.4 DEFINICION DE OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

El objetivo del presente proyecto es diseñar una aplicación móvil capaz de hacer el control de inventario del área de tecnología de Información (IT) de la empresa Coderoad, aplicando la tecnología RFID para tal propósito.

### **1.4.2 OBJETIVOS ESPECIFICOS**

Los objetivos específicos del presente trabajo son:

- ✓ Desplegar la información del objeto a partir de las etiquetas.
- ✓ Enlazar una etiqueta con un objeto y una persona.
- ✓ Diseñar e implementar una base de datos para el inventario.
- ✓ Analizar los reportes para el inventario.

## **1.5 JUSTIFICACION**

### **1.5.1 JUSTIFICACION SOCIAL**

Se conoce que el control de inventarios con el uso de la tecnología de código de barras puede muchas veces provocar que se pierda la información si la etiqueta se encuentra dañada con lo que no se puede llegar a conocer correctamente la información que esta contenía, por ello surge la idea de la construcción de un sistema que facilite la lectura de etiquetas sin la necesidad de tener que leer directamente a la etiqueta para tal objetivo se usa la tecnología RFID.

El Sistema beneficiará al control de inventario de la empresa Coderoad pues al momento de leer la etiqueta RFID este recibirá información exacta del objeto u objetos que está leyendo, además que se obtendrán datos de las personas a las que fue asignado dicho objeto, ayudando así a tener un registro más exacto de todos los objetos inventariados de la empresa.

### **1.5.2 JUSTIFICACION TECNICA**

Para una buena interacción entre el usuario y el software se necesita un alto manejo de usabilidad, una interfaz limpia y un sistema rápido, por lo cual para la implementación se propuso asociar las herramientas de software para la creación de la aplicación móvil con Ionic para la interfaz de usuario y el uso de Java para el manejo de los servicios y la base de datos.

### **1.5.3 JUSTIFICACION ECONOMICA**

El software realizado en el presente proyecto de grado, permite a la empresa Coderoad minimizar los costos del inventariado haciendo el uso etiquetas rfid vs etiquetas de código de barras para así poder trabajar con estas etiquetas un tiempo más largo, también ayuda a conocer el estado y la posición donde se encuentra cada objeto. Así, no extraviar los objetos y no gastar en la reposición del mismo.

## **1.6 ALCANCES Y LIMITES**

### **1.6.1 ALCANCES**

Los alcances del proyecto son los siguientes:

- ✓ El sistema contará con reportes para poder acceder a la información del inventario.
- ✓ El sistema contará con la información de todos los objetos que sean etiquetados.
- ✓ Se contará con un manejo de inventario usando tecnología RFID.
- ✓ El sistema será capaz de hacer una búsqueda con el lector RFID.

### **1.6.2 LIMITES**

#### **Límite Temático.**

En cuanto a los temas de aprendizaje solo se realizará la lectura simple y múltiple del Reader y la asociación de objetos con las etiquetas RFID.

#### **Límite Temporal.**

El límite temporal para el presente proyecto es de 6 meses para la entrega de los alcances y límites mencionados.

#### **Límite Geográfico.**

La aplicación será instanciada en los servidores de Coderoad, por lo cual para poder acceder a la aplicación se necesitará una conexión dentro de las instalaciones de la empresa y supervisada por los trabajadores de tal empresa.

## **1.7 APORTES**

El presente proyecto aporta al área de las aplicaciones del uso de la tecnología RFID, pudiendo ser utilizada como base para el uso en diferentes áreas donde esta tecnología pueda facilitar la automatización de procesos manuales de seguimientos de objetos.

Además de que podría ser utilizado como punto de referencia para nuevas aplicaciones sobre uso del internet en las cosas, pudiendo ser mejorado o ampliando el rango de la actual aplicación.



## **CAPITULO II MARCO TEORICO**

### **2.1 INTRODUCCION**

A lo largo de este capítulo vamos a definir los diferentes conceptos que nos encontraremos al hacer el uso de la tecnología RFID las ventajas y desventajas, pero también analizaremos la metodología<sup>6</sup> de desarrollo de nuestro sistema y las herramientas que usaremos para esta.

### **2.2 METODOLOGIA AGIL MOBILE-D**

La metodología ágil Mobile-D consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema (Amaya, 2013). Donde cada una de estas etapas tiene sus propias fases las cuales aplicaremos para tener una aplicación Mobile entregable y publicable para el usuario.

Mobile-D es una metodología creada el año 2004 que se desarrolló como parte de un proyecto de empresas de TI finlandesas llamada ICARUS, para el desarrollo de aplicaciones comerciales. Por lo cual los investigadores usaron prácticas habituales de desarrollo de software. (Blanco, Camarero, Fumero, Werterski, & Rodriguez, 2009).

Las ventajas de esta metodología son las siguientes:

- ✓ Un costo bajo al realizar un cambio en el proyecto.
- ✓ Entrega de resultados de manera rápida.
- ✓ Asegura el software adecuado en el momento adecuado.

---

<sup>6</sup> Conjunto de métodos que se siguen en una investigación científica, un estudio o una exposición doctrinal.

Mobile-D es una metodología ágil por lo cual los resultados serán rápidos, por el uso de pequeños grupos de trabajo por lo cual todos los integrantes del equipo de trabajo tienen que tener capacidades similares de trabajo.

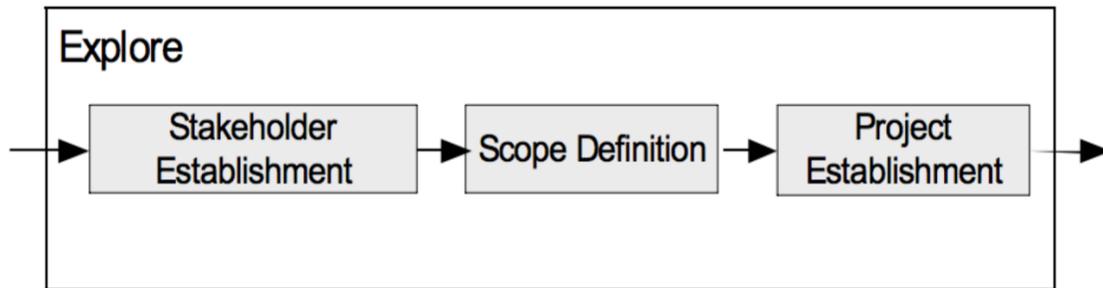
Como se mencionó anteriormente Mobile-D está basado en prácticas habituales de desarrollo de software como ser: eXtreme Programming (XP), Crystal methodologies y Rational Unified Process (RUP). Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron valiosos términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida. (Blanco, Camarero, Fumero, Werterski, & Rodriguez, 2009). Por lo cual el ciclo del proyecto se divide en cinco fases que son: exploración, inicialización, producción, estabilización y prueba de sistema como se ve en la figura 2.1.



*Figura 2.1: Estructura de la metodología Mobile-D*  
Fuente: (Blanco, Camarero, Fumero, Werterski, & Rodriguez, 2009)

### 2.2.1 EXPLORACION

Esta etapa está centrada a la planificación y a los conceptos del proyecto, para esta fase nosotros limitaremos los alcances de nuestro proyecto y vamos a establecer las funcionalidades de tal. Por lo cual el equipo en esta etapa debe hacer un plan y establecer las características del proyecto esto se realiza en tres fases figura 2.2.



*Figura 2.2: Fase de Exploración  
Fuente: (Koskela, 2004)*

#### **Establecimiento de Actores**

En esta fase se tiene que interactuar con el cliente el cual cuenta con el conocimiento de lo que se desea para la aplicación con lo cual se obtendrá la lista de requisitos de nuestra aplicación.

#### **Definición de Alcance**

En esta fase ya contamos con los requisitos de la aplicación por lo tanto se va planear y definir el alcance que tendrá nuestra aplicación.

#### **Establecimientos de proyecto**

Esta fase se encarga de definir y localizar los recursos tanto humanos como técnicos que se usaran en el desarrollo de nuestra aplicación.

## **El lenguaje unificado de modelado (UML)**

El lenguaje unificado de modelado (UML)<sup>7</sup>, es un estándar adoptado a nivel internacional por numerosos organismos y empresas para visualizar, especificar, construir y documentar un sistema.

Por lo tanto, UML se encarga de especificar o describir métodos o procesos que tiene un sistema antes de la elaboración del mismo, en otras palabras, en un lenguaje para mostrar las acciones que tendrá el sistema antes de la elaboración del mismo (Stevens & Pooley, 2002).

UML hace el uso de diagramas de casos de uso, para poder comunicar u analizar, los escenarios en los que el sistema o aplicación interactúa con personas, organizaciones o sistemas externos.

### **Diagrama de Casos de Uso**

Los casos de uso solamente se usan para representar los requisitos funcionales del sistema, como ser la forma, tipo y orden en el que el usuario (actor) interactúa con el sistema.

Para entender mejor de cómo se debe usar los casos de uso veremos cada uno de los elementos del mismo:

#### **Actor**

Es una clase de persona organización, dispositivo o componente de software externo que interactúa con el sistema, ver el símbolo en la tabla 2.1.

---

<sup>7</sup> UML: Unified Modeling Language por sus siglas en ingles.

## Caso de Uso

Un caso de uso representa las acciones que uno o varios de los actores realizan a fin de conseguir un objetivo determinado, ver símbolo en la tabla 2.1.

## Relaciones de uso

Las relaciones de uso sirven para relacionar un caso de uso con el actor que ejecuta dicha acción de los cuales podemos encontrar los siguientes:

- ✓ Asociación. Es la relación más básica que indica la invocación desde un actor o caso de uso a otro caso de uso. Ver símbolo en la tabla 2.1
- ✓ Dependencia o Instanciación. Es una forma muy particular de la relación entre clases, en la cual una clase depende del uso de otra, Ver símbolo en la tabla 2.1
- ✓ Generalización. Este tipo de relación es el más utilizado pues cumple una doble función **uso**<<uses>> cuando existe casos que son muy similares o de **herencia**<<extends>> cuando se tiene un conjunto de características de un caso de uso a otro Ver los símbolos en la tabla 2.1.

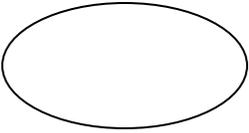
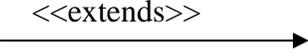
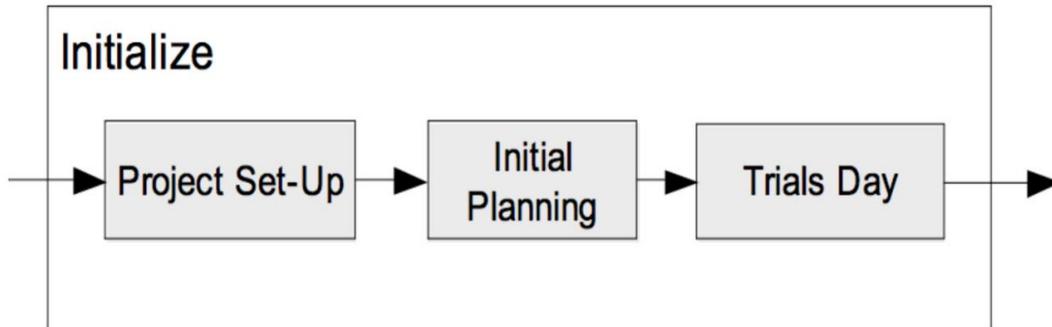
NOMBRE	SÍMBOLO
Actor	
Caso de Uso	
Asociación	
Generalización	
Extends	

Tabla 2.1: Simbología UML  
Fuente: (Salinas, 2010)

## 2.2.2 INICIALIZACION

Esta etapa el equipo de desarrollo debe encargarse de identificar todos los recursos necesarios para las siguientes etapas y se establece el entorno técnico. Esta etapa se divide en cuatro fases figura 2.3:



*Figura 2.3: Proceso de Inicialización  
Fuente: (Koskela, 2004)*

### **Puesta en marcha del proyecto**

Esta fase se encarga de configurar física y técnicamente los recursos para que el proyecto se puede colocar en un ambiente donde pueda ser monitoreado más fácilmente, entrenar al equipo en lo que sea necesario y especificar las formas de comunicación que tiene el equipo.

### **Planificación inicial**

Esta fase se encarga de planificar y explicar todos los requerimientos del sistema al equipo de desarrollo, planear el tipo de arquitectura tendrá la aplicación y la forma de afrontar a cada uno de los requerimientos adecuándolo a un patrón. Luego analizar la prioridad de cada requerimiento y encontrar un conjunto de requerimientos y poder así dividir por grupos de requerimientos y entregar al equipo de desarrollo.

## Día de Prueba

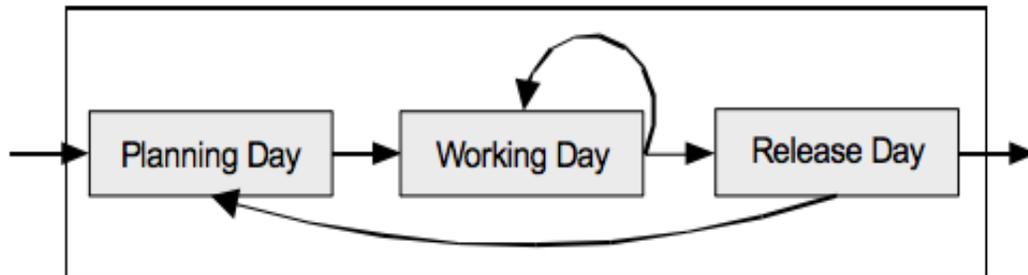
En esta fase se encarga de probar y configurar el entorno de desarrollo y estar listos para empezar el desarrollo de la aplicación, también tiene el propósito de crear el flujo básico de la aplicación y analizar qué problemas puede llegar a ocurrir en el transcurso del desarrollo de la aplicación.

## Día de Salida

Esta fase dependerá de la forma en que es implementada la aplicación por lo cual Mobile-D no especifica una descripción para dicha fase.

### 2.2.3 PRODUCCION

En esta etapa se repite la programación de 3 días hasta implementar todas las funcionalidades deseadas para el sistema. Aquí usamos el desarrollo dirigido por pruebas (*TDD*), para verificar el correcto funcionamiento de los desarrollos. Esta etapa se divide en tres fases que son figura 2.4.



*Figura 2.4 Fase de Producción*  
*Fuente: (Koskela, 2004)*

### **Día de planificación**

Esta fase se encarga de planificar la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes.

### **Día de Trabajo**

Esta fase se realiza el desarrollo de todo lo planificado en la anterior fase para luego integrar con el trabajo ya realizado en la aplicación. Se hace una integración continua para luego informar al cliente el trabajo hecho hasta ese punto.

### **Día de Entrega**

Esta fase tiene como objetivo integrar todo el sistema para luego realizar las pruebas de entrega y este apto para ser una aplicación de entrega. Cada prueba de entrega debe estar ligada a los requerimientos pedidos por el cliente para que finalmente se tenga una aplicación estable y lista para mostrar al cliente.

### **2.2.4 ESTABILIZACION**

Esta etapa está encargada de vincular todos los módulos separados en una sola aplicación. Por lo tanto, en esta fase se llevan a cabo las últimas integraciones de nuestra aplicación funcione completamente.

La estabilización tiene casi la misma estructura que la etapa producción con la única variante que existe una nueva fase llamada fase de documentación figura 2.5.

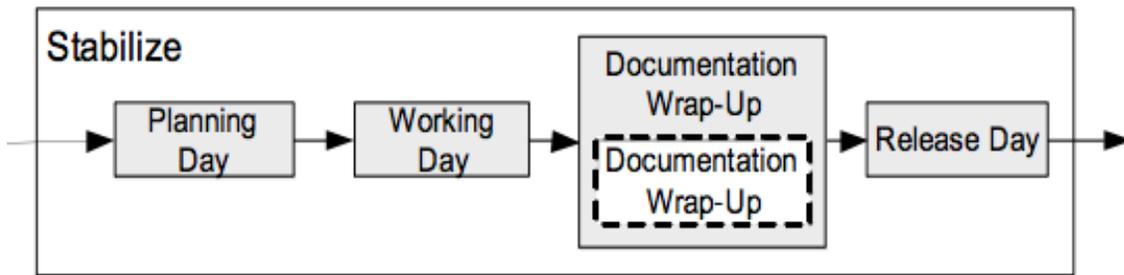


Figura 2.5 Etapa de Estabilización  
Fuente: (Koskela, 2004)

## Documentación

Esta fase tiene el objetivo de crear la documentación del módulo que se va integrando dentro de la aplicación principal por lo cual es considerada la más importante pues aparte de integrar toda la aplicación y tener una estable se genera la documentación de nuestra aplicación.

### 2.2.5 PRUEBAS

Tiene como meta principal tener una versión estable y plenamente funcional de nuestra aplicación. Para lo cual se ejecutan diferentes pruebas a lo largo de toda la aplicación y si es necesario resolver todos los errores que se vayan encontrando en nuestras pruebas. Esta etapa tiene las mismas fases que la etapa de producción como ser: Día de plan, día de trabajo y día de entrega figura 2.6.

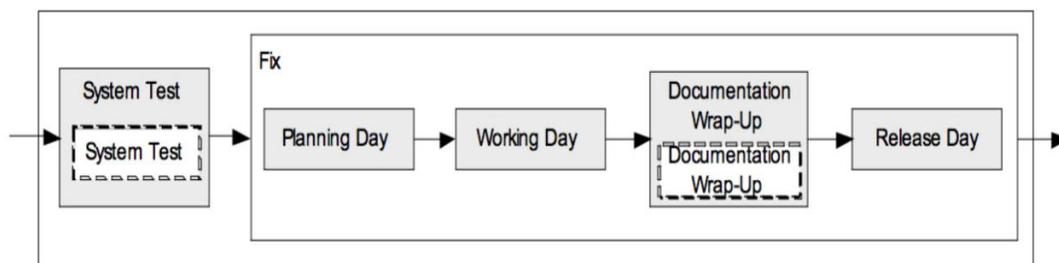


Figura 2.6. Etapa de Pruebas.  
Fuente: (Koskela, 2004)

Terminando esta etapa ya obtendremos una aplicación estable y publicable por lo cual daríamos terminado nuestra aplicación.

## **2.3 IONIC**

La construcción de aplicaciones móviles tiende a ser un gran valor para muchos desarrolladores, por lo cual con Ionic tú llegas a construir aplicaciones híbridas que son parecidas estéticamente a una aplicación móvil nativa. Una aplicación híbrida es un tipo de aplicación móviles que usa el navegador para mostrar su interface. Ionic es una herramienta que cuenta con utilidades disponibles para los desarrolladores puedan construir aplicación móvil híbrida de una forma muy rápida. Esta tecnología cuenta con el uso de HTML, CSS y JavaScript.

Lo que hace Ionic a la hora de construir una aplicación móvil es hacer el uso de una aplicación web embebida que luego será puesta dentro de una aplicación nativa que puede ser Android o iOS y para tal meta Ionic usa el apoyo de Cordova.

Ionic hace el uso de HTML, CSS y JavaScript por lo cual se ayuda de Angular para crear una página web ayudándose de diferentes directivas para los controles de manejo de la aplicación móvil (Wilken, 2016).

### **2.3.1 ¿QUE ES IONIC?**

Ionic es una combinación de tecnologías y herramientas para la creación de aplicaciones móviles híbridas de manera rápida, fácil y sencilla. Ionic está construido a base del Framework AngularJS y Cordova para el empaquetado de la aplicación nativa.

#### **Device**

Esto es el encargado de ejecutar la aplicación móvil. El dispositivo cuenta con las operaciones del sistema que pueden ser Android o iOS, usando por ejemplo localización GPS, lista de contactos o la cámara (Wilken, 2016).

## **Cordova app wrapper**

Esta es una aplicación nativa que carga una aplicación web. Cordova es una plataforma para la creación de aplicaciones móviles que pueden correr aplicaciones usando HTML, CSS y JavaScript dentro de una aplicación nativa. Este tipo de aplicaciones son las consideradas aplicaciones híbridas (Wilken, 2016).

## **Cordova JavaScript Api**

Este es el puente que se comunica entre la aplicación y el dispositivo haciendo el uso de las herramientas del dispositivo con el uso de JavaScript (Wilken, 2016).

## **AngularJS**

Esta es una aplicación web que controla las rutas y los funcionamientos de la aplicación. La aplicación web angular se ejecuta dentro de un componente WebView<sup>8</sup>.

## **Ionic**

Este provee al usuario las componentes para renderizar<sup>9</sup> la aplicación. Ionic está construido con base de AngularJS. Este incluye elementos tales como tabuladores, botones, navegadores, cabeceras entre otros.

### **2.3.2 TIPOS DE APLICACIONES MOVILES**

Para entender mejor lo que es una aplicación híbrida vamos a ver los tipos de aplicaciones móviles que existen. Hay tres tipos de aplicaciones básicas que son: aplicaciones nativas, sitios web móviles y aplicaciones móviles. En la figura X podemos comparar las diferentes arquitecturas que tienen cada uno de los tipos de aplicaciones.

---

<sup>8</sup> WebView: Clase que permite desplegar una página web.

<sup>9</sup> Renderizar: Proceso de generar una imagen o vídeo.

## Aplicaciones Nativas

Como su nombre indica, las aplicaciones nativas usan el lenguaje nativo en las que son programadas eso quiere decir que en el caso de aplicaciones Android debemos desarrollar en el lenguaje Java, mientras tanto si queremos desarrollar para iOS debemos hacerlo en Objective C o bien en Swift. Cada desarrollador debe usar el kit de desarrollo (SDK) brindado por cada compañía.

Tanto como iOS y Android brinda a la comunidad un conjunto de herramientas predefinidas para la construcción de aplicaciones web.

NRO.	VENTAJAS	
1	APIs Nativas	Las aplicaciones nativas usan las APIs directamente en la aplicación.
2	Performance	Tienen un alto nivel de velocidad a la hora de ejecutar los procesos de una aplicación.

*Tabla 2.2: Ventajas de las Aplicaciones Nativas  
Fuente: Elaboración Propia*

NRO.	DESVENTAJAS	
1	Requerimientos de lenguaje	Se requiere el conocimiento del lenguaje nativo de la aplicación.
2	Solo un tipo de aplicación	Solo se puede crear una aplicación con el mismo lenguaje de programación.

*Tabla 2.3: Desventajas de las Aplicaciones Nativas  
Fuente: Elaboración Propia*

## Sitios Web Móviles

Los sitios web móviles o aplicaciones web, trabajan muy bien dentro un explorador móvil. Estos sitios web son diseñados específicamente para un tamaño de pantalla de un dispositivo móvil. Muchos desarrolladores crean este tipo de aplicaciones web a partir del sitio web original adaptándolo a los diferentes tamaños para poder una mejor usabilidad.

Nro.	VENTAJAS	
1	Mantenibilidad	Son fáciles de mantener las aplicaciones móviles sin poner mucho esfuerzo a la hora de hacer cambios dentro de la página.
2	Sin Instalación	Como son aplicaciones móviles no necesitan ser instaladas dentro del dispositivo.
3	Multi-Plataforma	Como solo necesita un explorador móvil, esta será capaz de ser accesible en diferentes plataformas.

*Tabla 2.4: Ventajas de las Aplicaciones web móviles.  
Fuente: Elaboración Propia*

Nro.	DESVENTAJAS	
1	Sin acceso a las APIs	Por qué las aplicaciones web usan un explorador web estos no pueden acceder al uso de las APIs nativas de cada plataforma.
2	Limite en la Interfaz de Usuario	Es muy difícil crear aplicación amigable para el uso del touch de los dispositivos móviles.

*Tabla 2.5: Desventajas de las Aplicaciones web móviles  
Fuente: Elaboración Propia*

## Aplicaciones Híbridas

Las aplicaciones híbridas es una aplicación móvil capaz de fusionar ambos tipos de aplicaciones mencionadas anteriormente que son las aplicaciones móviles nativas y las aplicaciones web móviles. Esta usa una app wrapper para poder comunicarse con las APIs nativas de cada plataforma aparte de crear una página web q será mostrada dentro del componente WebView de cada plataforma.

Nro.	VENTAJAS	
1	Fácil y Rápido	Al ser una página web, este tipo de aplicaciones son rápidas y fácil de construir y mantener.
2	Acceso a las APIs	Gracias al wrapper podemos acceder a las APIs de cada plataforma.
3	Multi-Plataforma	Tú puedes crear una sola aplicación híbrida y tener la aplicación para diferentes tipos de plataformas.

*Tabla 2.6: Ventajas de las Aplicaciones híbridas  
Fuente: Elaboración Propia*

Nro.	DESVENTAJAS	
1	Límites de WebView	La aplicación solo puede usar dentro de un webview.
2	Acceso a las características nativas	Para poder acceder a las APIs nativas debemos hacer uso de diferentes plugins.
3	Control completo de la interface	Sin el conjunto de herramientas de Ionic, los desarrolladores tienden a crear todos los elementos de interfaz de usuario

*Tabla 2.7: Desventajas de las Aplicaciones híbridas  
Fuente: Elaboración Propia*

### 2.3.3 INTERFAZ USUARIO

Ionic cuenta con un conjunto de elementos que facilitan el manejo de la interfaz de usuario haciendo el uso de HTML similares a los elementos comunes dentro de las aplicaciones móviles nativas. Por ejemplo, podemos encontrar la barra de navegación, botones, listas, iconos, barra inferior, pestañas entre otros.

### 2.4 ANGULARJS

AngularJS es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje JavaScript con programación del lado del cliente.

HTML es ideal para declarar documentos estáticos, pero se tambalea cuando tratamos de utilizarlo para declarar vistas dinámicas en aplicaciones web. AngularJS le permite ampliar el vocabulario HTML para su aplicación. El medio ambiente resultante es extraordinariamente expresiva, legible y rápida de desarrollar.

NRO.	VENTAJAS
1	Facilidad de encontrar errores en el código
2	Calidad de software por el patrón utilizado por angularJS que es Modelo Vista Controlador MVC.
3	AngularJS es respaldado por Google

*Tabla 2.8: Ventajas AngularJS  
Fuente: Elaboración Propia*

## **2.5 ANDROID**

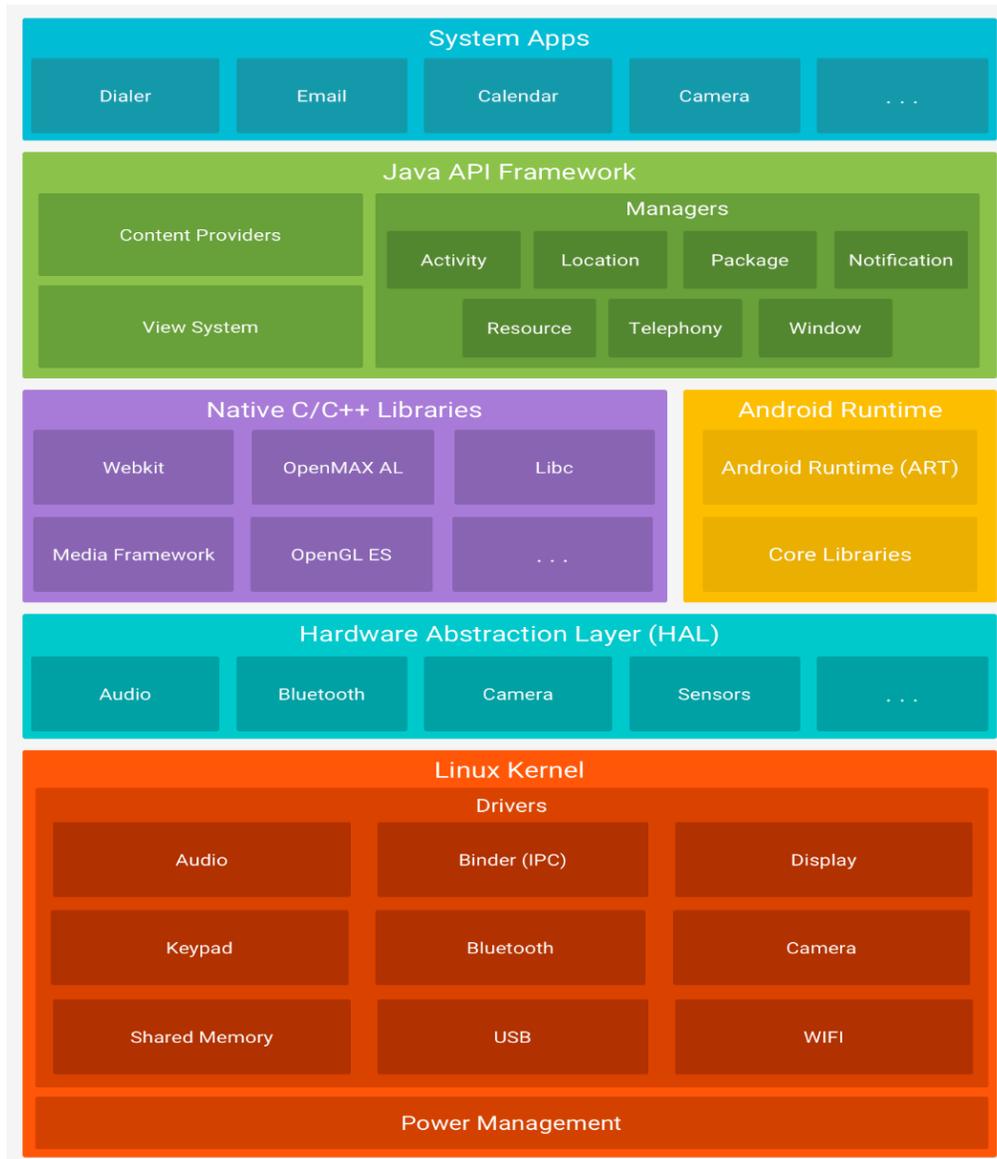
Android es un sistema operativo para dispositivos móviles como teléfonos inteligentes y tabletas basado en el núcleo de Linux. Es desarrollado por la Open Handset Alliance, la cual es dirigida por Google (Software de Comunicaciones, 2012).

El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. El éxito del sistema operativo Android se ha convertido en el sistema móvil más usado en el mundo actualmente, en el último evento de Google IO se anunció que Android cuenta 2 billones de dispositivos que usan Android (IO2017, 2017).

### **2.5.1 ARQUITECTURA**

Como vimos anteriormente Android está basado en el núcleo de Linux para una mejor apreciación de cómo está estructurado Android vea figura 2.7.





*Figura 2.7 Estructura de Android*  
*Fuente: (Google, Google Developer, 2017)*

## Núcleo de Linux

El uso del núcleo de Linux permite que Android aproveche funciones de seguridad claves y, al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un núcleo conocido.

## **Capa de Abstracción de Hardware (HAL)**

La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la API de Java de nivel más alto.

## **Tiempo de ejecución de Android**

Para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

## **Bibliotecas C/C++ nativas**

Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++.

## **Framework de la API de Java**

Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java.

## **Apps del sistema**

En Android se incluye un conjunto de apps centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos.

## 2.5.2 VERSIONES

A lo largo de la evolución de Android, este ha ido cambiando por diferentes versiones desde la 1.0 Apple Pie hasta la actual que es 7.0 Nougat. La tabla 2.8 muestra las versiones actuales de Android que soporta nuestra aplicación.

VERSION	NOMBRE	LOGO
4.4	KitKat	
5.0	Lollipop	
6.0	Marshmallow	
7.0	Nougat	

*Tabla 2.9: Versiones Soportadas*

*Fuente: Elaboración Propia*

## 2.6 TECNOLOGIA DE AUTOIDENTIFICACION Y RFID

Con el avance de la tecnología se fueron creando diferentes tipos de tecnología orientadas al sector de auto identificación y seguimiento de objetos. Para poder entender más sobre este tipo de tecnología vamos a ir viendo los principales tipos de tecnología

analizando las ventajas y desventajas de cada una poniendo un gran énfasis en la tecnología RFID y los componentes esenciales para poder aplicar este tipo de tecnología.

### **2.6.1 TECNOLOGIA DE AUTOIDENTIFICACION**

Actualmente encontramos una variedad de tecnologías para la identificación de diferentes objetos como ser personas, animales y productos. Una de estas tecnologías ha sido el principal exponente en el mercado, llamado código de barras, el cual fue implementado en la mayoría de los productos por la facilidad de manejo e implementación de tal tecnología. Sin embargo, las demás tecnologías fueron dejadas a un lado por la falta de información y el difícil manejo de ellas, pero actualmente se ha dado el boom de estas tecnologías gracias a el bajo precio y la documentación que cuenta ahora (Alvarado, 2008).

#### **Sistemas Biométricos**

Este tipo de tecnología usa características físicas para obtener la información de ciertos objetos. Algunas de las técnicas biométricas existen son:

- ✓ Reconocimiento de iris.
- ✓ Reflexión retinal.
- ✓ Geometría de la mano.
- ✓ Termografía mano, facial.
- ✓ Huellas dactilares.
- ✓ Patrón de la voz.
- ✓ Patrón de escritura.

### **2.6.2 TECNOLOGIA RFID**

La tecnología de identificación por radio frecuencia se ha convertido en uno de los pilares tecnológicos de avanzada para los países que efectúan procesos automatizados de gestión, control y trazabilidad de objetos (AIM, 2004).

En el esquema básico, la tecnología RFID contiene componentes esenciales: etiqueta (tag), una antena, un lector y un middleware, que permiten interrelacionar y obtener la información asociada al objeto.

Cuando se presenta el evento de identificación por radio frecuencia, entiéndase a este evento como la acción de los componentes para intercambio de información y detección por ondas de radio destinadas mediante un middleware RFID, donde se obtiene la información almacenada dentro de la etiqueta RFID.

### **2.6.3 MIDDLEWARE RFID**

Previo a la definición particular se toman en cuenta las siguientes definiciones descritas en el marco de recopilación bibliográfica:

“El middleware RFID, es un programa de software que se debe construir a partir de reglas de negocio que respondan a la satisfacción de necesidades de obtención de información (Ahson & Ilyas, 2008).”

“El middleware RFID es un software que conecta el hardware RFID con los sistemas implementados en una empresa (Hunt & Puglia, 2007).”

La tecnología de identificación por radio frecuencia en una entidad, permitirá gestionar el equipamiento RFID, ejecutar acciones y gestionar los eventos de detección, además de permitir la integración con otras aplicaciones bajo los criterios de:

- ✓ Resolver e identificar el código EPC<sup>10</sup>.
- ✓ Administrar lectores RFID.
- ✓ Formatear datos asociados a la etiqueta RFID.

---

<sup>10</sup> EPCIS: Servicio de Información de Código de Producto Electrónico.

#### **2.6.4 ETIQUETA RFID**

La etiqueta RFID es un dispositivo que puede ser fabricado en diferentes formas y tamaños, el cual puede ser adherido a un objeto, animal o persona. El cual contiene una antena que permite recibir y responder peticiones por radiofrecuencia desde un emisor/receptor RFID.

##### **Arquitectura.**

La etiqueta RFID o transportador está compuesta de una antena, un transductor radio y un chip. El propósito de la antena es permitirle al chip, que contiene cierta información, transmitir datos que están almacenados dentro de su memoria interna. La memoria puede ir variando de acuerdo al modelo de la etiqueta entre estas están:

- ✓ Solo Lectura: El código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.
- ✓ Lectura y Escritura: La información de identificación puede ser modificada por el lector.
- ✓ Anticolisión: Se trata de etiquetas especiales que permiten que un lector identifique varias al mismo tiempo.

#### **2.6.5 ANTENA RFID**

Este dispositivo por sus características puede ser clasificado por acción corta o acción larga, alta o baja densidad de campo que, conforme a las características y necesidad de aplicación, son variables a ser consideradas.

Las características de las antenas RFID, están definidas por su acción de irradiación en el espacio, este comportamiento se define como la proyección en el espacio de un haz cuya trayectoria es una onda llamada también patrón de irradiación o bulbo (Kaur & Sandhu, 2011), entonces conforme el diseño y construcción para la cual se aplicará se tienen antenas como:

## **Antenas Directivas**

Este tipo de antena denominada también de campo lejano es una antena diseñada para entornos donde sea necesario obtener un alto nivel de lecturas RFID, Con polarización circular esta antena está especialmente diseñada para entornos donde existe una gran densidad de etiquetas o cuando etiqueta está situada sobre productos densos, acuosos y de materiales compuestos Figura 2.8



*Figura 2.8: Antena direccional de alta densidad  
Fuente: (Clou, 2017)*

## **Antenas Threshold**

Las antenas RFID de tipo threshold de campo lejano es una antena diseñada para situaciones de barrera, delimitación de fronteras, pasos y lugares cerrados por metales, Dispone de un amplio haz de cobertura para maximizar la zona de acción. Instalada de forma lineal, puede formar una zona de umbral para detectar las etiquetas que lo cruzan.

## **Antenas Guardwall**

Las antenas RFID guardwall de campo semi-lejano, son antenas de doble polarización circular, una a derecha y otra a izquierda, provocando un intenso campo de acción dentro de ellas para poder penetrar en los embalajes y llegar a leer el artículo

unitario. Permitiendo emitir y recibir la información y señales entre los lectores y las etiquetas RFID. Utilizada una en frente de la otra en una cinta de transporte, es capaz de capturar todas las lecturas con los embalajes en movimiento, punto imprescindible en fabricación, almacenajes y distribución.

### **Antenas Brickyard**

Las antenas RFID brickyard de campo cercano fueron diseñadas para limitar la distancia de lectura de las etiquetas. Son ideales para trabajar en puntos de venta, tiendas de ropa, control de documentos, bibliotecas donde es necesario instalar un punto de lectura concreto y de corto alcance.

### **Antenas Guardrail**

La antena RFID guardrail de campo muy cercano es una antena diseñada para leer a distancias muy cortas principalmente para artículos unitarios, con alta y limitada zona de lectura, esta antena está especialmente diseñada para controles de acceso, venta de entradas(ticketing), control de documentos, control puntos de procesos y líneas de empaquetado a alta velocidad.

### **Antenas Universales de 8 dbi**

La antena RFID universal de campo lejano es una antena de polarización circular y de uso general, que permite emitir y recibir la información y señales entre los lectores y las etiquetas RFID, su alcance puede llegar hasta 12-18 metros en campo abierto y siempre en función de posición de la etiqueta RFID y de la superficie donde se encuentre instalado.

## **2.6.6 LECTOR RFID**

En la gama de lectores RFID existen dos tipos que resultan los más representativos: los lectores industriales y los lectores estándar (fijos y portátiles). La diferencia fundamental entre ambos tipos de equipos radica en el uso, siendo que las características

técnicas como procesamiento capacidad puertos de conexión, potencia de irradiación marcaran la diferencia (Chanrasekaram & Oustad, 2008).

También están los aspectos físicos, como la cantidad de puertos de conexión, interfaces de expansión protección control polvo y agua.

### **2.6.7 TIPO SE SISTEMAS DE RADIO FRECUENCIA**

Los sistemas RFID fueron diseñados para operar en tres ámbitos de mercado: Industrial, Científico y Medico (Kaur & Sandhu, 2011), al igual que los sistemas Wireless o Bluetooth, tienen rangos de operación en potencia y frecuencia, puesto que los mismos deben considerar aspectos de cobertura, amplitud de onda e irradiación de potencia además de los protocolos de irradiación para la intercomunicación entre lector y etiqueta (Stephen B & Miles, 2008), es por ello que independiente al tipo de encapsulamiento en el cual se encapsula el chip RFID para etiquetas de tipo Activo, Semi Activo y Pasivo.

#### **Sistemas RFID de Baja frecuencia**

Los sistemas de baja frecuencia (LF – *Low Frecuencias*) tienen un rango de operación entre los 125-134 KHz<sup>11</sup> mismos operan a corto alcance. Los sistemas RFID operan libremente sobre este ancho de banda para aplicaciones aeronáuticas y navegación marina (Finkenzeller, 2010).

#### **Sistemas RFID de Alta Frecuencia**

Los sistemas de alta frecuencia (HF–*High Frecuencias*) tienen un rango de operación entre 13-56 MHz<sup>12</sup>, los sistemas RFID HF similar al anterior facilitan un mayor

---

<sup>11</sup> KHz: Unidad de medición de frecuencias, representada a 1000 Hz (Hertz).

<sup>12</sup> MHz: Unidad de medición de frecuencia, representada a 1000 KHz (Kilo Hertz).

rango de aplicaciones puesto que la potencia de irradiación es menor y permite encapsular componentes electrónicos de menor tamaño (miniaturización) (Finkenzeller, 2010).

## **2.7 USABILIDAD EN APLICACIONES MÓVILES**

En las tecnologías móviles a diferencia de otras (Web, escritorio, TV digital), la usabilidad es un problema más significativo, esto es debido a que una gran mayoría de las aplicaciones móviles son difíciles de usar, son poco flexibles y no son robustas. Debido a la movilidad que permiten estos dispositivos las pruebas de usabilidad en un ambiente real de uso son difíciles de realizar. En primer lugar, puede ser complicado establecer estudios realistas que reflejan la riqueza del contexto de uso descrito anteriormente. En segundo lugar, está lejos de ser trivial aplicar técnicas de evaluación clásicas, cuando la prueba se realiza en condiciones reales de uso. Por último, las pruebas en un entorno real complican la recolección de datos y limitan el control sobre éstos y sobre el usuario, ya que el mismo se está moviendo físicamente en un contexto con un número de variables desconocidas que afectan potencialmente la evaluación. El trabajo realizado por Husain propone un modelo, llamado Mobile Goal Question Metric (mGQM) (Hussain, 2012), el cual define un conjunto de métricas para evaluar la usabilidad de las aplicaciones móviles. El modelo propone un conjunto de métricas objetivas y subjetivas, las que a continuación se detallan:

### **2.7.1 METRICAS OBJETIVAS**

Para el control de calidad de una aplicación móvil el modelo mGQM propone ejecutar las siguientes métricas objetivas:

- ✓ Tiempo requerido para introducir los datos: Esta métrica mide el tiempo tomado por el usuario para ingresar datos de entrada.
- ✓ Número de errores al teclear los datos: Esta métrica mide el número de errores cometidos, mientras el usuario introduce datos al utilizar la aplicación.
- ✓ Tiempo empleado para instalar: Esta métrica mide el tiempo empleado por cada aplicación que se instala en el dispositivo móvil.

- ✓ El número de interacciones durante la instalación de la aplicación: Esta métrica mide la cantidad de interacciones entre el usuario y el dispositivo móvil cuando se está instalando una aplicación.
- ✓ Tiempo necesario para aprender: Este indicador mide el tiempo empleado por los usuarios para aprender a utilizar la aplicación.
- ✓ Número de errores mientras aprende: Este indicador mide el número de errores cometidos por los usuarios cuando están aprendiendo a utilizar la aplicación.
- ✓ Número de errores: Este indicador mide si la aplicación es precisa o imprecisa.
- ✓ Tiempo necesario para completar la tarea: Este indicador medirá el tiempo empleado por el usuario para completar las tareas dadas.
- ✓ Número de tareas exitosas en el primer intento: Esta métrica indica el número de tareas terminadas con éxito en el primer intento.
- ✓ Número de tareas exitosas en el tiempo dado: Este indicador proporciona información sobre si los usuarios están familiarizados con la aplicación mediante el recuento del número de tareas completadas con éxito en un tiempo dado.
- ✓ Tiempo necesario para iniciar la aplicación: Este indicador mide el tiempo empleado por los dispositivos móviles para iniciar una aplicación.
- ✓ Tiempo necesario para responder: Esta métrica mide el tiempo empleado por una aplicación para responder a la entrada del usuario.
- ✓ Tiempo necesario para conectarse a la red (vía satélite o Wi-Fi): El tiempo empleado por la aplicación para conectarse a la red es una medida esencial porque refleja la satisfacción de los usuarios.
- ✓ Número de recursos del sistema mostrados: Esta métrica cuenta el número de recursos del sistema (nivel de la señal o de la batería) que se muestra durante el estudio de usabilidad.
- ✓ Número de solicitudes de actualización de la aplicación: Esta métrica cuenta el número de solicitudes de las aplicaciones para actualizar la versión.

- ✓ Porcentaje de la batería usada durante la instalación: Este indicador mide el porcentaje de batería que se utiliza cuando se instalan una aplicación en un dispositivo móvil.
- ✓ Porcentaje de la batería usada: Este indicador mide el porcentaje de la batería usada por una aplicación para una tarea determinada.

### **2.7.2 METRICAS SUBJETIVAS**

Para el control de usabilidad de la aplicación el modelo mGQM propone ejecutar las siguientes métricas subjetivas.

- ✓ Satisfacción con el teclado virtual: Este indicador mide el grado de satisfacción del usuario con el teclado proporcionado por la aplicación móvil o por el dispositivo móvil.
- ✓ Satisfacción con la salida: Este indicador mide el nivel de satisfacción con la salida proporcionada por las aplicaciones.
- ✓ Satisfacción con el proceso de instalación: Este indicador mide el nivel de satisfacción de los usuarios cuando instalan aplicaciones en los dispositivos móviles.
- ✓ Satisfacción con la optimización del tamaño de pantalla: Este indicador mide el nivel de satisfacción de los usuarios con la optimización del tamaño de la pantalla.
- ✓ Satisfacción con la ayuda: Este indicador mide el grado de satisfacción de los usuarios con la ayuda proporcionada por las aplicaciones.
- ✓ Satisfacción con los contenidos: Satisfacción con el contenido es una medida esencial para asegurar que los contenidos suministrados cumplirán las necesidades de los usuarios.
- ✓ Placer: Esta métrica mide el disfrute de los usuarios cuando utilizan las aplicaciones móviles.
- ✓ Satisfacción con la interfaz: Satisfacción con la interfaz es también una medida importante porque una buena interfaz atraerá a más usuarios a utilizar la aplicación.

- ✓ Seguridad durante la conducción: Este indicador mide el nivel de seguridad para las personas que utilizan las aplicaciones móviles mientras conducen.
- ✓ Facilidad para encontrar ayuda: Esta métrica mide cuán fácil es para los usuarios encontrar ayuda en las aplicaciones.
- ✓ Estrés: Esta métrica mide el nivel de estrés que sienten los usuarios cuando usan las aplicaciones móviles.
- ✓ Satisfacción con el indicador de señal: Esta medida indica la satisfacción del usuario con el indicador de señal proporcionada por las aplicaciones móviles.
- ✓ Satisfacción con el joystick virtual: Esta métrica indica el nivel de satisfacción de los usuarios con el joystick virtual.
- ✓ Satisfacción mientras aprenden: Este indicador mide el nivel de satisfacción de los usuarios cuando aprenden a utilizar las aplicaciones.
- ✓ Satisfacción con el texto: Este indicador mide el nivel de satisfacción de los usuarios con el texto que muestran las aplicaciones.
- ✓ Satisfacción con el sistema de navegación: Este indicador mide el grado de satisfacción con el sistema de navegación (como los usuarios pasan de un menú a otro menú o submenú).
- ✓ Satisfacción con la pantalla táctil: Esta métrica mide el nivel de satisfacción de los usuarios con la pantalla táctil.
- ✓ Satisfacción con el botón de menú: Esta medida indica el nivel de satisfacción de los usuarios con el botón de menú.

## **2.8 TECNICAS DE INVENTARIADO**

En un entorno empresarial se conoce como la gestión de inventario que es el proceso encargado de asegurar la cantidad exacta y los productos adecuados para que la organización, de tal manera que los trabajadores puedan asegurar todas las operaciones que se debe cumplir para la continuación de la elaboración del producto para con los clientes de la empresa.

### **2.8.1 DEFINICION DE CONTROL DE INVENTARIOS**

El control de inventario busca mantener disponible los productos que requiere la empresa. De acuerdo a (Ballou, 2005) “Los inventarios son acumulaciones de materias primas, provisiones, componentes, trabajo en proceso y productos terminados que aparecen en numerosos puntos a lo largo del canal de producción y de logística de una empresa.”

### **2.8.2 IMPORTANCIA DEL INVENTARIO**

Llevar un buen control del inventario de tu organización te muestra dinero tienes en espera actualmente. Es por eso que tener una buena gestión de tu inventario es crucial para el manejo y la producción de tu empresa o negocio. En otras palabras, un buen manejo de inventario te ahorrara dinero.

- ✓ Evita deterioros de producto
- ✓ Evita perdida de inventarios
- ✓ Ahorrar en costos de almacenamiento

### **2.8.3 TECNICAS DE INVENTARIO**

La gestión del inventario es una parte variable al implementar un negocio. El sistema óptimo es diferente para cada empresa. Sin embargo, cada empresa debe mejorar la técnica para el control de inventario para poder eliminar errores humanos, tanto como sea posible. Esto significa tener un sistema de manejo de control de inventario. Para tal objetivo se puede elegir entre distintas técnicas de inventario, la usada actualmente en la empresa Coderoad es una Auditoria regular.

#### **Auditoria Regular**

Los informes de los objetos existentes dentro de tu organización se deben manejar dentro de un registro físico o virtual para el cual podemos confiar en un software especializado para nuestro objetivo.

## **Inventario Físico**

Dentro de una organización se realiza un inventario anual físico de todos los objetos dentro de la organización para dar resultados de contabilidad e impuestos. Estos inventarios físicos solo se realizan una vez al año, pueden ser terriblemente perjudiciales para la empresa además de ser tediosos y gastar mucho tiempo. Si encuentras una discrepancia puede llegar a ser difícil encontrar donde se encuentra el problema ya que los resultados deben ser buscados en el registro de todo el año.

## **Punto de Comprobación**

A lo largo de todo el año se puede elegir un producto o un conjunto de productos para poder comprobarlo en un punto cualquiera, este proceso se llama hacer un producto de comprobación. Esto simplemente sirve para poder tener el control del objeto durante el lapso del año y eliminar los errores que puedan surgir al hacer el inventario físico.

## **Ciclo de Conteo**

En vez de hacer un inventario físico completo durante el fin de año o realizarlo una vez al año, podemos realizar ciclo de conteos durante todo el año como ser una vez a la semana, al mes o cada trimestre donde se deben elegir los elementos de mayor valor para su conteo. (Campbell, 2015)

## **2.9 ESTIMACION DE COSTOS DEL PROYECTO**

### **2.9.1 LAS MÉTRICAS ORIENTADAS AL TAMAÑO**

La elaboración de un software tiene diferentes factores importantes que se deben seguir, es importante analizar correctamente cada una de las métricas. Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso que la desarrollada. Estas se basan midiendo la cantidad de líneas de código LDC que contiene durante el desarrollo.

Para ello vamos a calificar el proyecto siguiendo la información de la tabla 2.10

CATEGORÍA	PROGRAMADORES	DURACIÓN	LÍNEAS DE CÓDIGO	EJEMPLO
Trivial	1	0-4 semanas	<1K	Utilidad de ordenación
Pequeño	1	1-6 semanas	1k-3k	Bibliotecas de funciones
Media	2-5	0.5 -2 semanas	3k- 50k	Compilador C
Grande	5-20	2-3 años	50k – 100k	Pequeño SO
Muy Grande	100-1000	4 – 5 años	100k – 1M	Grandes SO
Gigante	1000 – 5000	5 – 10 años	>1M	Sistema de Distribución

*Tabla 2.10: Categoría de un proyecto en función de sus líneas de código  
Fuente: (Baik, 2001)*

## 2.9.2 METODO COCOMO II

Una metodología que se encarga de medir el costo de un proyecto es el Costo modelo de construcción (COCOMO) se debe a Barry Boehm, y está orientada hacia las líneas de código. Este método está compuesto de 3 tipos:

- ✓ Modelo Básico.
- ✓ Modelo Intermedio.
- ✓ Modelo Avanzado.

Las cuales son aplicadas a tres diferentes tipos de software.

**Orgánico** Proyectos relativamente sencillos, menores a 50000 líneas de código.

**Semi-Acoplado** Proyectos intermedios en complejidad y tamaño.

**Empotrado** Proyecto bastantes complejos, en los cuales se cuenta con poca experiencia lo que quiere decir que son proyectos completamente innovadores.

Dado que sólo se va a emplear una variable para la estimación (la línea de código), se empleará CÓCOMO básico, ya que es un modelo uní variable estático, con lo que se obtiene una valoración objetiva del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que posee menos de 50.000 líneas de código.

La ecuación del esfuerzo de COCOMO básico tiene la siguiente forma:

**h=hombre, m=mes, h-m=hombre-mes, KLDC=cantidad de código**

	<b>ORGÁNICO</b>
Esfuerzo estimado(h-m)	$E_D = 2,4(KLDC)^{1,05}$
Tiempo Desarrollo(m)	$T_D = 2,5 (E_D)^{0,38}$
Productividad	$PR = LDC/E_D$
N medio de personas (h)	FSP (Full- Time equivalent Software Personel) $P_E = E_D/T_D$

*Tabla 2.11: Coeficientes COCOMO II*

*Fuente (Boehm , 1995)*

## 2.10 HERRAMIENTAS TECNOLOGICAS

Para la elaboración del presente proyecto se utilizaron las diferentes herramientas tecnológicas para un mejor ambiente de desarrollo.

### 2.10.1 GITHUB

GitHub es una plataforma de desarrollo inspirada en tu forma de trabajar. De código abierto a negocio, puede alojar y revisar código, gestionar proyectos y crear software junto con millones de otros desarrolladores al mismo tiempo (GitHub, 2017).

GitHub hace el uso de Git que es un sistema de control de versión distribuida libre y de código abierto diseñado para manejar todo, desde proyectos pequeños a muy grandes con rapidez y eficiencia.

### 2.10.2 WEBSTORM

WebStorm es un entorno de desarrollo integrado (IDE) para el desarrollo de aplicaciones web o móviles. Estas aplicaciones pueden ser desarrolladas en los siguientes tipos de lenguaje.

#### Web

- ✓ Angular.
- ✓ React.
- ✓ Vue.js.

#### Mobile

- ✓ Ionic.
- ✓ Cordova.
- ✓ React Native.

#### Server

- ✓ Node.js.
- ✓ Meteor.

#### Desktop

- ✓ Electron.



## CAPITULO III MARCO APLICATIVO

### 3.1 INTRODUCCION

En el presente capítulo se aplicarán los fundamentos teóricos mencionados anteriormente que se enfocan en el análisis y diseño de nuestro sistema, para tal objetivo nosotros estamos utilizando la metodología ágil Mobile-D para la cual debemos seguir un proceso de evaluación por sprint de acuerdo a la evolución del proyecto.

Para trabajar con Mobile-D exitosamente necesitamos entender correctamente como este trabaja, para ello vamos a ver su proceso de desarrollo.



*Figura 3.1: Proceso de Desarrollo Mobile-D*  
*Fuente: (Calle, 2014)*

## **3.2 FASE DE EXPLORACION**

Para la fase de exploración mostros haremos la planificación de los alcances y el desarrollo de la aplicación, para la cual nos centraremos en la búsqueda de las librerías necesarias para nuestro proyecto y las herramientas para el desarrollo del mismo.

### **3.2.1 AMBIENTE FISICO**

El ambiente físico en el que se trabajara para el desarrollo y testeo de la aplicación son los siguientes:

- ✓ Sony Xperia Z5 Premium
- ✓ Reader
- ✓ Laptop HP Envy Core i7
- ✓ Servidor

### **3.2.2 AMBIENTE TECNICO**

El ambiente técnico para el desarrollo de la aplicación utilizara las siguientes herramientas:

- ✓ WebStorm 2016.2
- ✓ Git versión 2.8.1 Windows.1
- ✓ Ionic v1
- ✓ Cordova Plugin Reader

### **3.2.3 RECOPIACION DE REQUERIMIENTOS**

Durante la fase de recopilación de requerimientos se hizo la creación de una lista de las tareas a realizar para el proyecto, donde se capturo los requerimientos de los usuarios, cada requerimiento es considerado como una tarea para nuestra fase de producción. Si bien nuestra lista ira creciendo con cada requerimiento, se debe hacer una priorización de los mismos, la cual será llevada por el propietario del proyecto de la empresa Coderoad.

<b>NRO.</b>	<b>DESCRIPCIÓN</b>	<b>ESTADO</b>
1	Diseñar de autenticación de usuario	Finalizado
2	Diseñar pantalla para asociar etiquetas con objetos	Finalizado
3	Diseñar pantalla para listar a las personas	Finalizado
4	Diseñar pantalla para asociar objeto con personas	Finalizado
5	Generar reporte de depreciación	Finalizado
6	Generar reporte de objetos dañados	Finalizado
7	Generar reporte de objetos	Finalizado
8	Correcciones de Bugs	Finalizado
9	Puerta en marcha del Sistema	Finalizado

*Tabla 3.1: Lista de Requerimientos  
Fuente: Elaboración Propia*

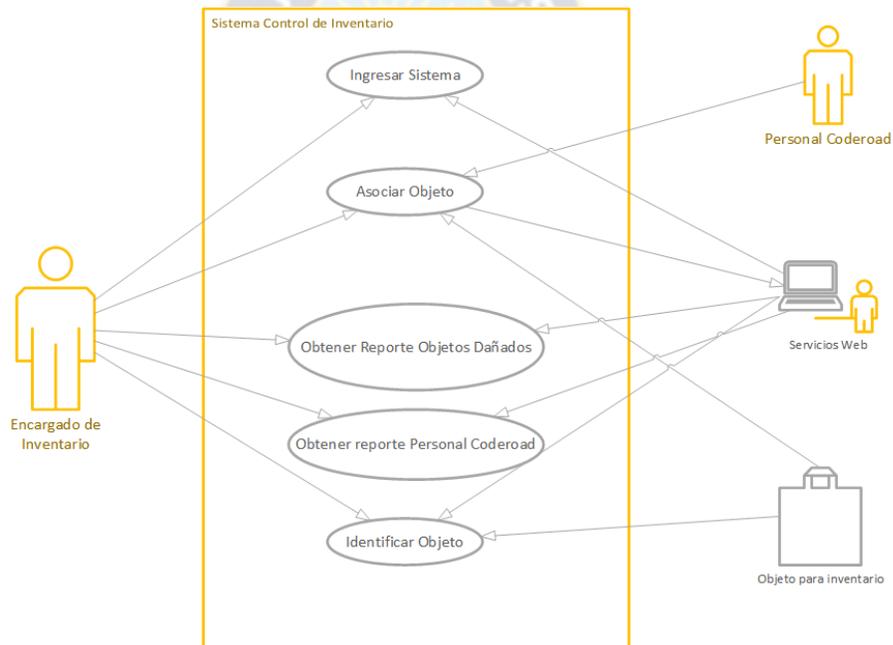
### **3.2.4 CASOS DE USO**

Para poder empezar el proyecto se necesita definir los casos de uso de la aplicación, que son definidos en el lenguaje UML definido en el anterior capítulo.

Para ello se analizará cada uno de los requerimientos de la empresa Coderoad para la aplicación del control de inventario listada en la Tabla 3.1.

El software desarrollado tiene como objetivo hacer el control de inventario de la empresa, haciendo el uso de la tecnología RFID, para ello el flujo del sistema parte desde la asociación de la etiqueta al objeto, donde se enlaza una etiqueta RFID a un objeto nuevo para ser guardada dentro de la base de datos. La siguiente fase es la asociación de un objeto-etiqueta a una persona, esta relación solo ocurrirá cuando un objeto sea entregado a una persona. Además, se contará con un conjunto de reportes sobre los estados de los objetos.

En la Figura 3.2 podemos observar el caso de uso general de la aplicación.



*Figura 3.2: Caso de Uso General  
Fuente: Elaboración Propia*

### **Ingreso a la aplicación**

En la figura 3.3 podemos apreciar el caso de uso para ingresar dentro de la aplicación, este ingreso sirve para poder determinar los reportes y acciones que dicho usuario puede acceder.



Figura 3.3: Caso de uso Ingreso a la aplicación  
Fuente: Elaboración Propia

Caso de Uso: Ingreso a la aplicación	
Actor: Encargado de Inventario	
Curso Normal	Alternativas
El usuario Ingresa sus credenciales	
El sistema obtiene la información básica sobre el usuario.	2.1 Si no está registrado, se le informa que debe pasar por el área de IT.
Accede a los flujos permitidos para el usuario.	

Tabla 3.2: Caso de Uso Ingreso a la Aplicación  
Fuente: Elaboración Propia

### Asociar Objeto

En la figura 3.4 podemos apreciar el caso de uso para la asociación de un objeto a una persona, para esto se necesita el proceso de lectura RFID del objeto y seleccionar a la persona a la que será asignada el usuario.

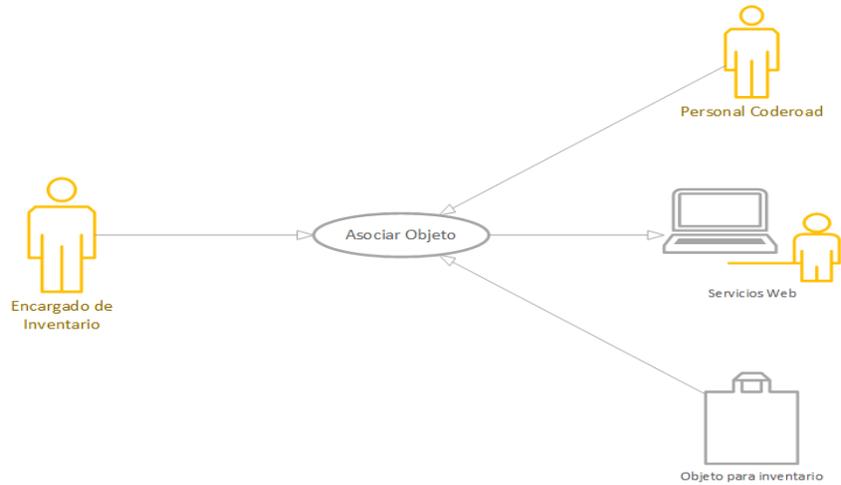


Figura 3.4: Caso de Uso Asociar Objeto  
Fuente: Elaboración Propia

Caso de Uso: Asociar Objeto	
Actor: Encargado de Inventario, Servicios Web, Objeto para inventario, Personal Coderoad.	
Curso Normal	Alternativas
Seleccionar Tipo de Lectura Código de Barras o RFID.	
Realizar la lectura del Objeto	
El sistema obtiene la información básica sobre el objeto.	3.1 Si no está registrado el objeto o el objeto ya fue asignado muestra una alerta en la pantalla.
Seleccionar la persona a Asociar.	3.2 Si no existe la persona se informa que debe pasar por el área de IT.
El sistema guarda la asociación de objeto-persona.	

Tabla 3.3: Caso de Uso Asociar Objeto  
Fuente: Elaboración Propia

## Obtener Reporte de Objetos Dañados

En la figura 3.5 podemos apreciar el caso de uso para la obtención del reporte de Objetos Dañados, para el cual el usuario debe ingresar las fechas de inicio y final para la consulta de los objetos dañados entre ese rango de fechas.



*Figura 3.5: Caso de Uso Obtener Reporte Objetos Dañados*  
*Fuente: Elaboración Propia*

Caso de Uso: Obtener Reporte de Objetos Dañados
Actor: Encargado de Inventario
Curso Normal
1) Seleccionar reporte de Objetos Dañados.
2) Seleccionar el Rango de Fechas.
3) El sistema obtiene la información del reporte desde las fechas dadas.

*Tabla 3.4: Caso de Uso Obtener Reporte de Objetos Dañados*  
*Fuente: Elaboración Propia*

### Obtener Reporte de Personal

En la figura 3.6 podemos apreciar el caso de uso para la obtención del reporte de personal, para el cual el usuario selecciona el reporte Personal para obtener la lista de todo el personal de la empresa.

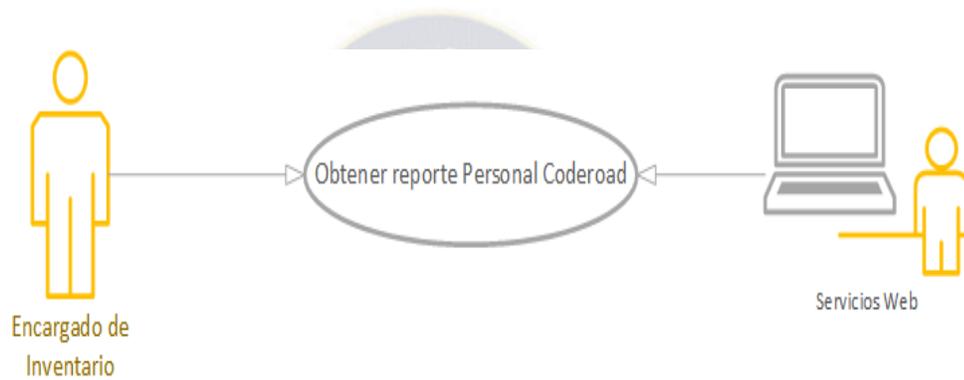


Figura 3.6: Caso de Uso Obtener Reporte Personal  
Fuente: Elaboración Propia

Caso de Uso: Obtener Reporte de Personal
Actor: Encargado de Inventario
Curso Normal
1) Seleccionar reporte de Personal.
2) El sistema obtiene la información del reporte desde las fechas dadas.

Tabla 3.5: Caso de Uso Obtener reporte de Personal  
Fuente: Elaboración Propia

## Identificar Objeto

En la figura 3.7 podemos apreciar el caso de uso para la identificación de un objeto por medio del RFID o Código de Barras, para el cual el usuario selecciona el tipo de Reader y leer la etiqueta del objeto.

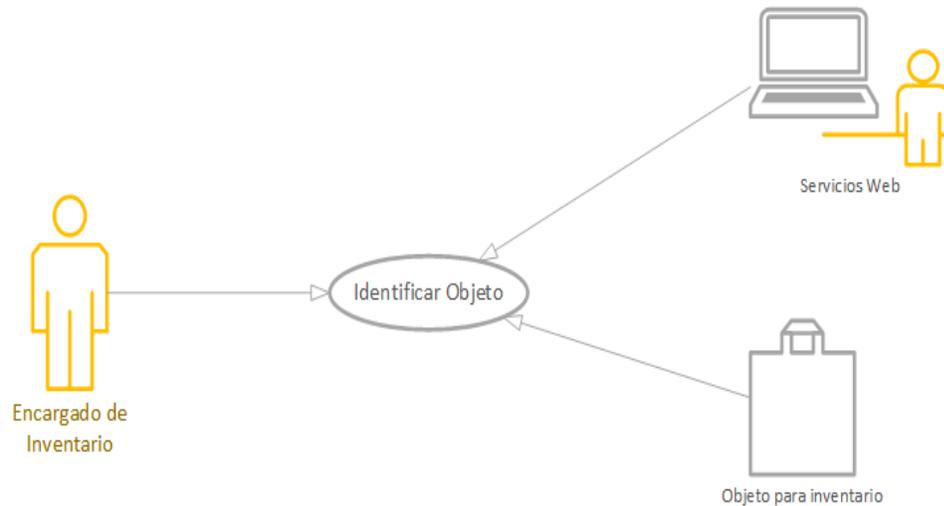


Figura 3.7: Caso de Uso Identificar Objeto  
Fuente: Elaboración Propia

Caso de Uso: Identificar Objeto	
Actor: Encargado de Inventario, Objeto para Inventario, Servicios Web	
Curso Normal	Alternativas
Seleccionar Tipo de Lectura Código de Barras o RFID.	
Realizar la lectura del Objeto	
El sistema obtiene la información básica sobre el objeto.	3.1 Si no está registrado el objeto o el objeto ya fue asignado muestra una alerta en la pantalla.

Tabla 3.6: Caso de uso Identificar Objeto  
Fuente: Elaboración Propia

### **3.3 FASE DE INICIALIZACIÓN**

Después de la obtención de requerimientos y el desarrollo de cada caso de uso del software, debemos definir la arquitectura de la aplicación para cumplir cada una de las funcionalidades mencionadas en los casos de uso.

#### **3.3.1 CONFIGURACIÓN INICIAL DEL PROYECTO**

Los objetivos principales que debe alcanzar la aplicación en el desarrollo son los siguientes:

- ✓ Conectar el Reader y el teléfono celular vía Bluetooth.
- ✓ Creación inicial del proyecto en Ionic para la aplicación Android.
- ✓ Implementar el flujo de asociación de Objeto-Etiqueta.
- ✓ Implementar el flujo de asociación de Objeto-Persona.
- ✓ Implementación de la librería para la conexión.

#### **3.3.2 PLANIFICACIÓN INICIAL**

En esta fase se planifica la arquitectura del proyecto, configuraciones iniciales y el diseño de las pantallas iniciales de nuestra aplicación vea figura 3.8.

#### **DISEÑO DE LA ARQUITECTURA DEL PROYECTO**

La arquitectura del proyecto está definida por una aplicación hecha en Ionic para Android que hace la asociación de objeto-etiqueta y objeto-persona, toda esta información será guardada en el servidor de la empresa Ver la Figura 3.9.

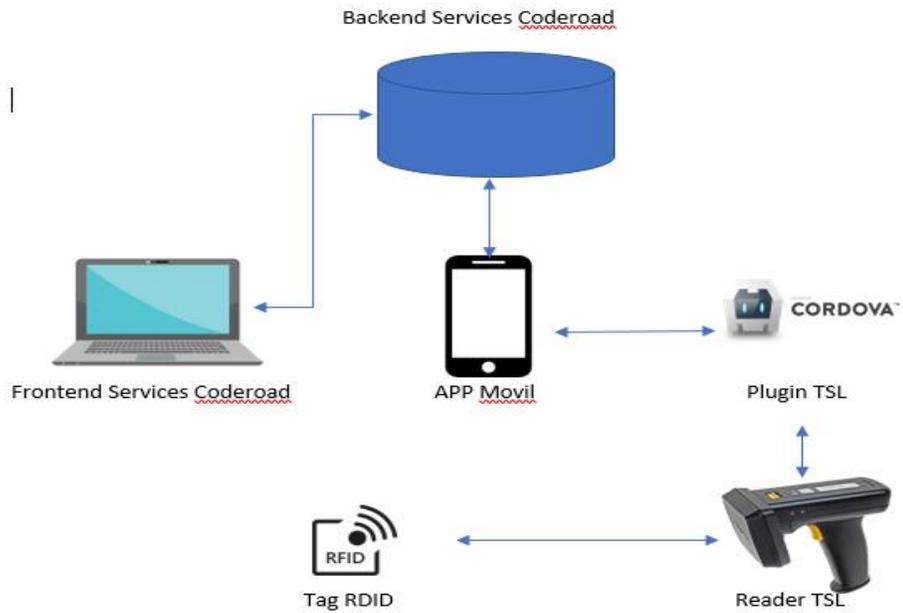


Figura 3.8 Configuración del Proyecto

Fuente: Elaboración Propia

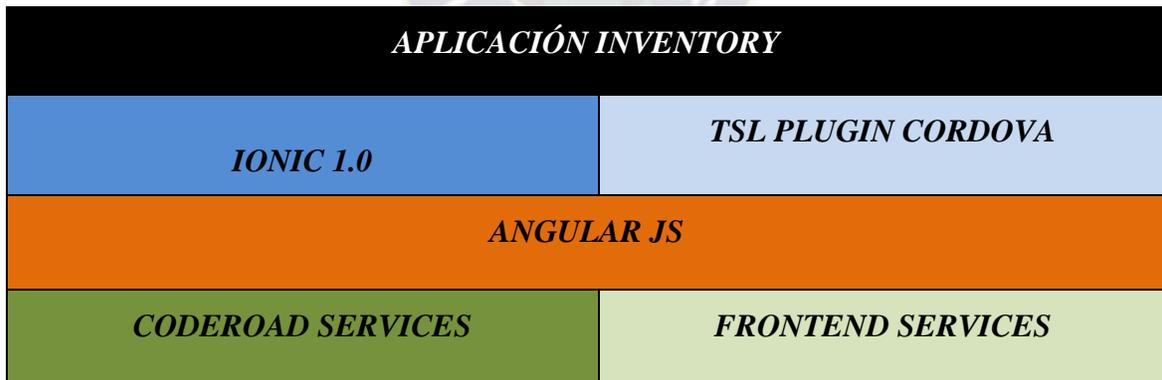


Figura 3.9 Arquitectura del Proyecto

Fuente: Elaboración Propia

### **3.4 FASE DE PRODUCTO**

Para esta fase del proyecto ya contamos con la estructura y el entorno de trabajos listos para hacer una división de las tareas similares que analizamos en la tabla 3.1 de requerimientos, para cumplir todos los requerimientos, se hizo la división de tres iteraciones que son detalladas a continuación.

#### **3.4.1 ITERACION 1**

##### **DÍA DE PLANIFICACIÓN**

En la iteración 1, se trabajó en la conexión de la aplicación con los servicios web proporcionados por la empresa CODEROAD para el guardado de los datos de la aplicación.

Los objetivos de la iteración 1 se muestran a continuación:

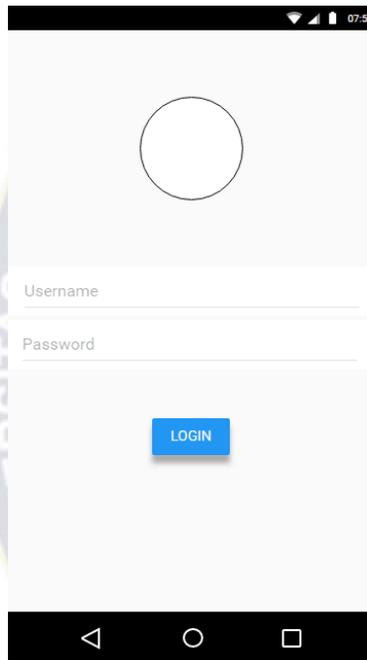
- ✓ Creación de módulo servicios para el consumo de los servicios web de la aplicación.
- ✓ Creación de servicios para el guardado de datos dentro de la aplicación Android usando PouchDB.
- ✓ Creación y diseño de las vistas para configuración del servidor y acceso a la aplicación.
- ✓ Creación de la persistencia en caso de que se pierda la conexión.

##### **DÍA DE TRABAJO**

El día de trabajo está conformado por dos semanas en la cual se desarrolló cada uno de los objetivos planteados en el día de la planificación.

## MODULO DE CONFIGURACION E INGRESO

Las siguientes imágenes son los diseños de las pantallas de configuración y acceso a la aplicación figura 3.10, que permitirá la configuración de la IP del servidor otorgado por la empresa y los usuarios que puedan acceder a ella.



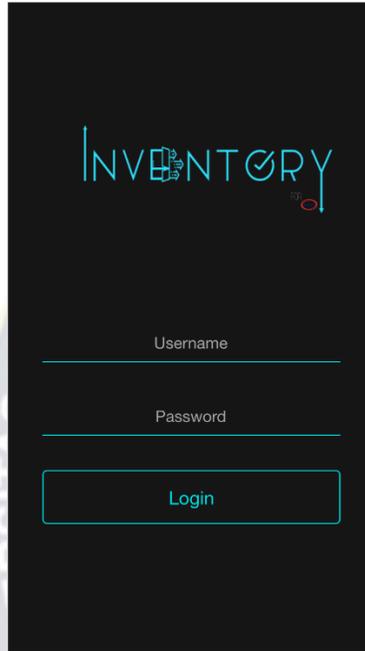
*Figura 3.10: Pantalla de Ingreso de la Aplicación  
Fuente: Elaboración Propia*

## DIA DE ENTREGA

El día de entrega podemos apreciar cada una de las tareas que se realizaron dentro del día de trabajo de la iteración como ser los métodos creados y las pantallas ya concluidas de nuestra aplicación.

## MODULO DE CONFIGURACION E INGRESO

Para este punto se creó la pantalla de ingreso de la aplicación haciendo un puente con los servicios otorgados por la empresa vea figura 3.11.



*Figura 3.11: Pantalla Final Ingreso de la Aplicación  
Fuente: Elaboración Propia*

### 3.4.2 ITERACION 2

#### DIA DE PLANIFICACION

Los objetivos de la iteración 2 son la creación de la conexión del Reader TSL con la aplicación Android para la lectura de etiquetas RFID y las pantallas donde se mostrará la información de la persona u objeto.

- ✓ Creación de la librería para la conexión del Reader TSL con Ionic.
- ✓ Diseño y creación de las pantallas para la asociación entre objetos.
- ✓ Diseño y creación de las pantallas detalles para ver la información del objeto.

## DIA DE TRABAJO

Durante las dos semanas de la iteración 2 se implementó la librería de conexión vía Bluetooth del Reader TSL con la aplicación Ionic además del diseño de las pantallas del flujo principal de la aplicación.

## LIBREARIA TSL

Para poder conectar el dispositivo móvil con Ionic se creó una librería TSL que permita conectar la aplicación móvil con el Reader TSL, esta librería tiene los comandos básicos para el manejo del Reader detallado en la Tabla 3.7.

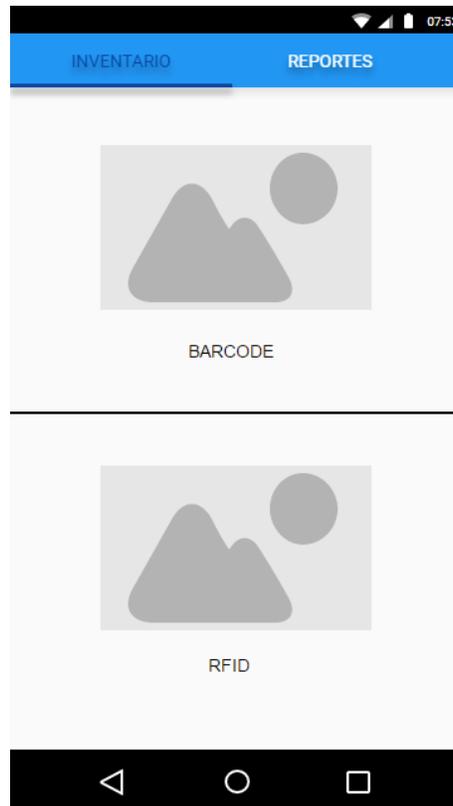
MÉTODO	DESCRIPCIÓN
OpenRFIDReader	Crea el puerto para la lectura de etiquetas RFID.
CloseRFIDReader	Cierra el puerto para la lectura de etiquetas RFID
OpenBarcodeReader	Abre la sesión para la lectura de código de barras.
CloseBarcodeReader	Cierra la sesión para la lectura de código de barras
ConnectDevice	Método para conectar el Reader con un dispositivo móvil.

*Tabla 3.7: Métodos de la Librería TSL*

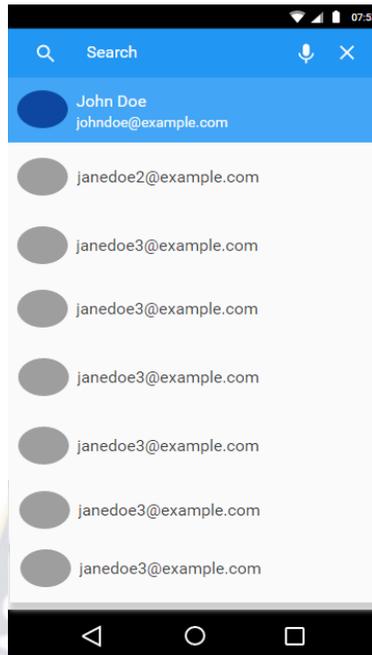
*Fuente: Elaboración Propia*

## FLUJO DE ASOCIACION

El flujo de asociación tiene como tarea asociar un objeto ya etiquetado con una persona, que consta de 3 pantallas que son: Elección de lectura de Tag figura 3.12, Lista y selección de Persona figura 3.13 y finalmente la pantalla de confirmación de asociación del objeto con la persona figura 3.14.



*Figura 3.12: Pantalla de Tipo de Lectura  
Elaboración: Propia*



*Figura 3.13: Lista y selección de Usuario*  
*Fuente: Elaboración Propia*



*Figura 3.14: Pantalla de Asociación*  
*Fuente: Elaboración Propia*

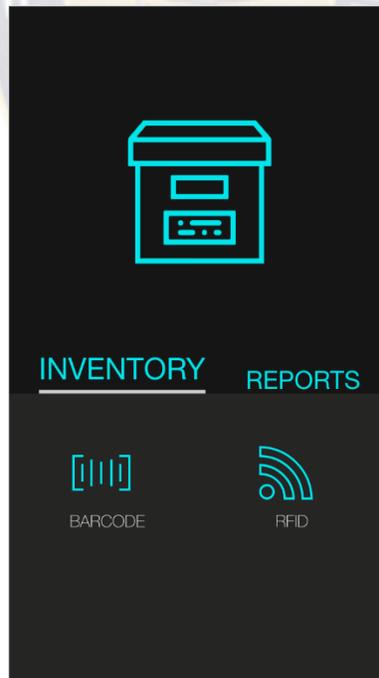
## DIA DE ENTREGA

A continuación, se muestran las pantallas terminadas del flujo principal de la aplicación.

## FLUJO DE ASOCIACIÓN

Para el flujo de asociación en la pantalla de elección de lectura se cuenta con dos opciones la primera opción hace el uso de la cámara del dispositivo móvil para la lectura de una etiqueta de código de barras, mientras la otra opción permite hacer una lectura de etiquetas RFID haciendo el uso de la librería TSL mencionada anteriormente vea figura 3.15.

Continuando con el flujo de asociación tenemos la pantalla de elección de personas (figura 3.16) y la pantalla de confirmación para la asociación del objeto vea figura 3.17.



*Figura 3.15: Pantalla Final de Selección de Reader  
Fuente: Elaboración Propia*

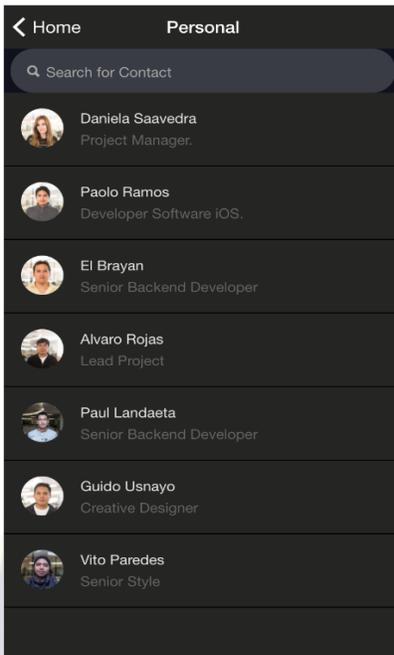


Figura 3.16: Pantalla Final Selección de Perna  
Fuente: Elaboración Propia

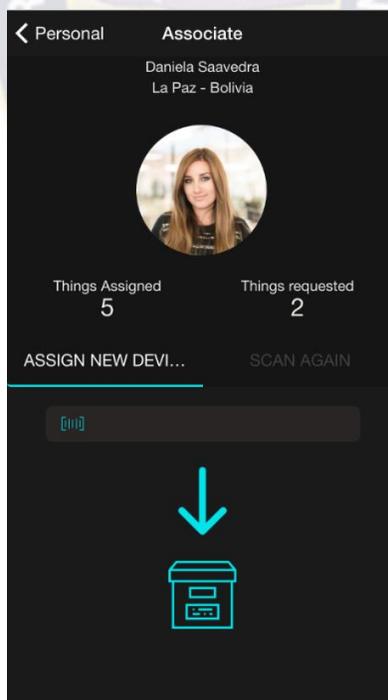


Figura 3.17: Pantalla Final Confirmación de Asociación  
Fuente: Elaboración Propia

### **3.4.3 ITERACION 3**

#### **DIA DE PLANIFICACION**

Esta es la última iteración del proyecto que tiene un lapso de tiempo de 3 semanas para la elaboración de los diferentes reportes que contiene la aplicación.

- ✓ Diseño de la pantalla de reportes de depreciación.
- ✓ Diseño de la pantalla de reportes de objetos dañados.
- ✓ Diseño de la pantalla de reportes objetos por persona.

#### **DIA DE TRABAJO**

Por ser la iteración más grande y con solo 3 tareas, se trabajó cada reporte una semana viendo las dificultades de cada uno a la hora de desplegar la información del reporte.

##### **Reporte de depreciación**

El fin de este reporte es listar todos los objetos con su depreciación correspondiente desde el día de la compra vea la figura 3.18.

##### **Reporte de Usuarios**

Dentro del reporte de usuarios podemos apreciar los objetos y la cantidad de cada uno que se dieron a un determinado empleado de la empresa vea figura 3.19.

##### **Reporte de Objetos Dañados**

El siguiente reporte se encarga de listar todos los objetos dañados con el nombre del ultimo propietario de dicho objeto si tuviera, vea figura 3.20.

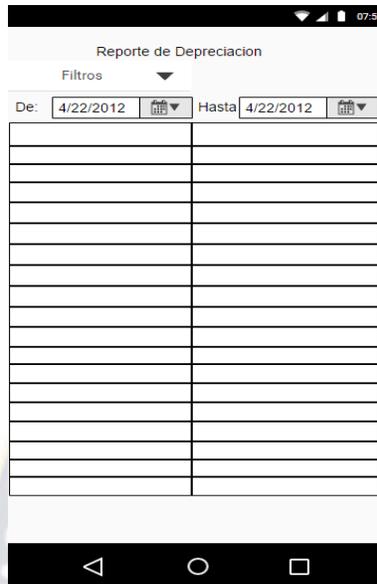
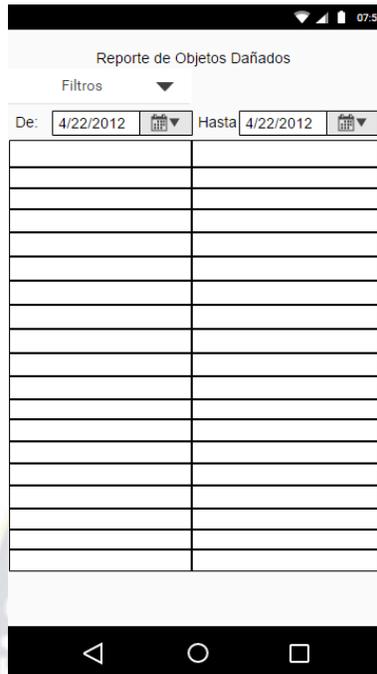


Figura 3.18: Diseño del Reporte de Depreciación  
Fuente: Elaboración Propia



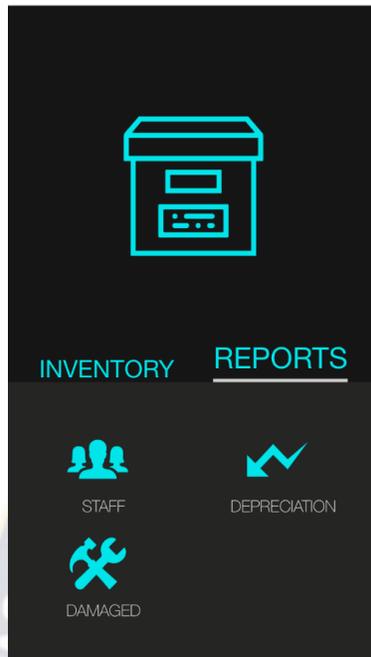
Figura 3.19: Diseño de pantalla Información de Empleado  
Fuente: Elaboración Propia



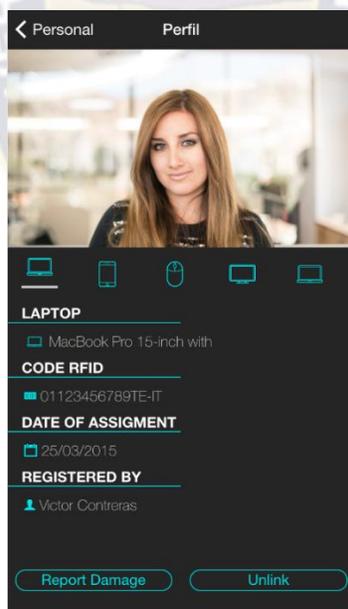
*Figura 3.20: Diseño de la Pantalla Reporte de Objetos*  
*Fuente: Elaboración Propia*

## **DIA DE ENTREGA**

A continuación, podemos apreciar cada una de las pantallas de reportes y el manejo de los filtros para cada uno.



*Figura 3.21: Diseño Final de la Pantalla Selección de Reportes  
Fuente: Elaboración Propia*



*Figura 3.22: Pantalla Final Reporte por Persona  
Fuente: Elaboración Propia*



Figura 3.23 Pantalla Final Reporte de Objetos Dañados  
Fuente: Elaboración Propia

Item	Date of purchase	Time in stock
MacBook Pro	Jan 13	2 Years
MacBook Pro	Jan 13	30 days
MacBook Pro	Jan 13	50 days
MacBook Pro	Jan 13	5 years
MacBook Pro	Jan 13	2 Years
MacBook Pro	Jan 13	30 days
MacBook Pro	Jan 13	50 days
MacBook Pro	Jan 13	5 years
MacBook Pro	Jan 13	2 Years
MacBook Pro	Jan 13	30 days
MacBook Pro	Jan 13	50 days
MacBook Pro	Jan 13	5 years
MacBook Pro	Jan 13	2 Years
MacBook Pro	Jan 13	30 days
MacBook Pro	Jan 13	50 days

15 records of 1000

Figura 3.24: Pantalla Final Reporte de Depreciación de Objetos  
Fuente: Elaboración Propia

### **3.5 FASE DE ESTABILIZACION**

Para esta fase se hizo las mejoras de la librería del Reader TSL, además se hizo unas mejoras dentro del rendimiento de la aplicación como ser la velocidad de respuesta a los servicios y el guardado de las imágenes dentro del dispositivo.



## **CAPITULO IV METRICAS DE CALIDAD Y SEGURIDAD**

### **4.1. CALIDAD DE SOFTWARE**

#### **4.1.1. INTRODUCCION**

La meta fundamental de la Ingeniería de Software es construir un software de alta calidad, por lo cual en este capítulo vamos a evaluar directamente nuestra aplicación móvil.

Como última etapa del proceso de desarrollo de la aplicación móvil usando la metodología Mobile-D tenemos la fase de pruebas, para tal objetivo nosotros usaremos métricas de control de calidad basados en el trabajo “USABILIDAD EN APLICACIONES MÓVILES” del Lic. Enríquez Juan Gabriel y la Dra. Casas Sandra Isabel (Enriquez & Casas , 2013). En el trabajo se considera 2 aspectos: Métricas Objetivas y Métricas Subjetivas.

#### **4.1.2. DEFINICION Y ESPECIFICACION DE REQUERIMIENTOS DE CALIDAD**

Esta fase se evalúa todos los requerimientos de calidad, a partir de un proceso de evaluación para lo cual se hace una mezcla de los dos aspectos mencionados anteriormente las métricas objetivas y las métricas subjetivas (modelo de enfoque mixto de calidad), con el fin de analizar, comparar y comprender cada uno de los atributos que existe dentro de la aplicación, todos los requerimientos deben responder a las necesidades y deseos de un perfil de usuario y dominio establecido en la primera fase de exploración.

Para una mejor comprensión vamos a definir el dominio de la aplicación, desde una vista de la evaluación, como a un sistema real o abstracto del universo que existe independiente del sistema de evaluación.

El rango para la evaluación de las preguntas planteadas para medir la calidad de nuestro software es detallado en la tabla 4.1

<b>RANGO</b>	<b>DESCRIPCIÓN</b>
0	No existe la característica
1-50	Malo
51-60	Regular
61-70	Optimo
70-100	Excelente

*Tabla 4.1: Rango de Evaluación  
Fuente: Elaboración Propia*

Para el análisis se hizo encuestas sobre el manejo de nuestro software esto para las primeras fases de la implementación del sistema y medir la usabilidad del mismo.

1. MÉTRICAS OBJETIVAS	DESCRIPCIÓN	NOTA	PERSONAS
1.1 Usabilidad de entrada de datos	1.1.1 Tiempo para introducir datos	85	10
	1.1.2 Errores al teclear los datos	90	10
1.2 Instalación	1.2.1 Tiempo empleado para Instalar	80	10
	1.2.2 Iteraciones para la instalación	80	10
	1.2.2.1 Permanencia de controles directos	90	10
	1.2.2.2 Permanencia de controles indirectos A	80	10
	1.2.2.4 Estabilidad	90	10
	1.2.3 Aspectos de estilo	95	10
	1.2.4. Preferencia estética	80	10
1.3 Misceláneas	1.3.1 Soporte de lenguaje extranjero	90	10
	1.3.2 Identificador de resolución de pantalla	85	10

*Tabla 4.2: Métricas Objetivas  
Fuente: Elaboración Propia*

Las métricas subjetivas son detalladas en la tabla 4.3

2. MÉTRICAS SUBJETIVAS	DESCRIPCIÓN	NOTA %	PERSONAS
2.1 Ayuda	2.1.1 Satisfacción con la ayuda	70	10
	2.1.2 Satisfacción con los contenidos	80	10
	2.1.3 Facilidad para encontrar ayuda	70	10
2.2 Navegación	2.2.1 Tamaño de la pantalla	95	10
	2.2.3 Satisfacción con la Interfaz	95	10
	2.2.3 Satisfacción con el texto	95	10
	2.2.4 Satisfacción con la pantalla táctil	100	10
2.3 Misceláneas	2.3.1 Satisfacción del teclado virtual	100	10
	2.3.2 Satisfacción del proceso de instalación	100	10
	2.3.3 Placer al usar la aplicación	85	10
	2.3.4 Estrés al usar la aplicación	58	10

*Tabla 4.3 Métricas Subjetivas  
Fuente: Elaboración Propia*

## **4.2 SEGURIDAD**

Seguridad en términos generales e informáticos es un estado que indica si el sistema está libre de peligro, daño o riesgo. Se entiende como daño o riesgo todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para una gran mayoría de los expertos el concepto de seguridad en la informática es ilusorio porque no existe un sistema 100% seguro.

Para que un sistema se pueda definir como seguro debe cumplir con 4 características.

### **4.2.1 RIESGOS**

El riesgo es la posibilidad de pérdida como consecuencia de un peligro o amenaza. Se refiere a la pérdida de confidencialidad, disponibilidad o integridad como el resultado de una amenaza a la seguridad de IT. Riesgos normalmente son calificados como de alta, media y baja severidad. A continuación, se identifica las principales clases de vulnerabilidades de las aplicaciones web.

La falta de autenticación Suficiente, comprobar correctamente las credenciales de autenticación y luego proporcionar acceso a una web en consecuencia la aplicación son las operaciones de suma importancia opera un servidor para llevar a cabo en la prestación de seguridad y privacidad. Antes de acceder a una aplicación web, un servidor debe exigir a los usuarios finales autenticarse y confirmar de hecho son quienes pretender ser.

La Administración de sesiones débil, es algo que la mayoría de los usuarios no son conscientes, pero esto es una metodología de seguridad esencial para frustrar a los hackers

intenten romper y tomar el control de una sesión. La idea es que un servidor sea capaz de verificar periódicamente la interacción del usuario.

El control de Acceso Débil, restringir o controlar el acceso a una aplicación, o para el caso de todos los importantes procesos y archivos, es el aspecto más importante de la seguridad.

Las fallas de inyección, la vulnerabilidad de inyección es causada por una falta de suficiente filtrado o ensayo de datos; es decir, la entrada de un cliente. Todos los datos que por la aplicación web inmediatamente. Esta es una clase de ataque que se basa en los datos de inyección en una aplicación web con el fin de facilitar la ejecución o interpretación de los datos maliciosos de una manera inesperada. Ejemplos de ataques dentro de esta clase incluyen cross-site scripting (XSS), inyección SQL, inyección de cabecera y muchos más. Dan lugar a la ejecución de código malicioso para robar información. En estos ataques, las víctimas son las aplicaciones web y la base de datos detrás de ellos.

La Vista de Datos no autorizada, se trata de una vulnerabilidad común, donde se da a conocer información sensible sobre el entorno de aplicaciones web. Esto puede ayudar a un hacker en el sondeo de los datos más sensibles en la preparación para un ataque. Surge la vulnerabilidad cuando un usuario no autorizado identifica un objeto, como un servidor o el nombre del archivo por un nombre específico.

El manejo de errores, esta es una variación sobre el tema de revelar lo que puede parecer inocua información a personas no autorizadas. En realidad, un hacker competente

podrá ser maliciosa puede ingresar intencionalmente datos anormales con el fin de obligar a los mensajes de error. Un atacante podría utilizar estos mensajes de error genéricos como los archivos y directorios para planear un ataque.

Cross-Site Scripting Attacks. Reciben una gran cantidad de la cobertura de noticias, principalmente por el aumento dramático en el uso de lenguaje de script. Los mismos viejos problemas se arrastran hasta incluso en estos relativamente nuevos lenguajes de script de filtrado insuficiente de los datos de entrada de los usuarios y por el contrario, la prohibición de todo tipo, pero que se espera de los datos. Esta vulnerabilidad XSS es causada por defectos en los lenguajes de script del lado del cliente como JavaScript y el lenguaje de programación HTML. Se puede surgir cuando las aplicaciones web aceptan los datos de entrada de los usuarios y dinámicamente incluirlo en páginas web sin correctamente validarla primero. Vulnerabilidades XSS permite a un atacante ejecutar comandos arbitrarios y mostrar contenido arbitrario en el navegador del usuario.

### **4.3 SEGURIDAD DE ACCESO**

Las soluciones de los riesgos mencionados anteriormente se detallan a continuación.

#### **4.3.1. LA AUTENTICACION**

En el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse se considera usar SSL sobre HTTP

(HTTPS) para transferir los datos y asegurarse de que el cifrado cubre las credenciales y el ID de sesión en todas las comunicaciones.

#### **4.3.2 SEGURIDAD BASADA EN LA SESION**

El modelo de seguridad basado en sesiones depende de una sesión en el servidor, de mantener la identidad de un usuario a través de las solicitudes, la sesión tendrá un tiempo de vida de 30 minutos si no hay actividad presente.



## **CAPITULO V ANALISIS COSTO BENEFICIO**

### **5.1 INTRODUCCION**

El análisis de costo y beneficio es una parte muy importante en la elaboración de un proyecto. El propósito, es mostrar a los usuarios del nuevo sistema, al igual que a los administradores de la organización, que los beneficios que se espera con la nueva aplicación superan los costos esperados.

### **5.2 COCOMO II**

Con el estudio de costo-beneficio del presente proyecto se determinará el valor de la inversión para el desarrollo e implementación de la aplicación móvil, combinando los métodos de estimación de esfuerzo, tiempo y costo que son las más utilizadas en el desarrollo de proyectos informáticos. Cabe mencionar que el costo total del proyecto desarrollado es la suma del costo de software de desarrollo, más el costo de implementación, más el costo de elaboración del proyecto realizado.

Para estimar el costo total del sistema se tomarán en cuenta los siguientes costos: Costo de la elaboración del proyecto, costos del software desarrollado, costos de la implementación del sistema.

#### **5.2.1 PUNTO DE FUNCION**

Primero se debe hallar el punto de función para poder obtener el costo del proyecto para ello vamos analizar cada una de las pantallas de nuestro sistema.

##### **Número de entradas de usuario**

Son los que proporcionan datos al sistema para realizar las distintas operaciones de altas, bajas, etc. vea tabla 5.1.

<b>NRO.</b>	<b>ENTRADAS DE USUARIO</b>
1	Pantalla de Ingreso.
2	Ingreso Etiqueta RFID.
3	Ingreso Etiqueta de Código de Barras.

*Tabla 5.1: Entradas de la Aplicación  
Fuente: Elaboración Propia*

### **Número de salidas de Usuario**

Son las que proporcionan información elaborada por el sistema que son gestionadas al usuario.

<b>NRO.</b>	<b>SALIDAS DE USUARIO</b>
1	Confirmación de datos de Usuario
2	Reporte Objetos Dañados
3	Reporte de Depreciación
4	Reporte de Objetos entregados
5	Lista de Personal
6	Lista de Objetos

*Tabla 5.2: Salidas de la Aplicación  
Fuente: Elaboración Propia*

### Número de Peticiones de Usuario

Es una entrada de tipo interactiva que proporciona una salida o respuesta.

NRO.	PETICIONES DE USUARIO
1	Autenticación de Usuario
2	Obtener Objeto Escaneado
3	Obtener Lista de Objetos Persona
4	Obtener Datos de Depreciación

*Tabla 5.3: Peticiones de Usuario  
Fuente: Elaboración Propia*

### Numero de Archivos

Es el grupo lógico de datos que puede ser parte de una base de datos.

NRO.	ARCHIVOS
1	Información del tag
2	Personal
3	Objeto

*Tabla 5.4 Archivos de la aplicación  
Fuente: Elaboración Propia*

Una vez contando con todos los datos sobre las entradas, salidas, etc. De nuestra aplicación pasamos a agruparlas para obtener el parámetro de medición vea la tabla 5.5.

PARAMETROS DE MEDICION	CUENTAS	FACTOR DE PONDERACIÓN			TOTALES
Nro. de entradas	3	3	4	6	9
Nro. de Salidas	6	4	5	7	30
Nro. de peticiones	4	5	4	6	20
Nro. de archivos	3	7	10	15	21
<b>TOTALES EN CUENTA</b>					<b>80</b>

*Tabla 5.5: Interfaces Parámetros de medición  
Fuente: Elaboración Propia*

### Calculo de factor de ajuste de la complejidad

Los valores son respondidos en la escala de 0 a 5, donde el 0 significa que no es importante y el 5 que es fundamental para la aplicación vea la tabla 5.6.

<b>FACTOR DE COMPLEJIDAD</b>	<b>VALOR</b>
Requiere copias de seguridad y recuperación	3
Necesita comunicación de datos	3
Existe funciones de procedimiento distribuido	2
Rendimiento critico	2
Se ejecuta en un entorno operativo existente	2
Se requiere entrada de datos en línea (on-line)	4
Transacciones de entradas en múltiples pantallas	3
Archivos maestros actualizados en línea (on-line)	4
Complejidad de valores del dominio de información	3
Complejidad del procedimiento interno	3
Código diseñado para la reutilización	4
Conversión/instalación en diseño	2
Instalaciones Múltiples	2
Aplicación diseñada para el cambio	4
<b>SUMA de <math>f() = \sum f_i()</math></b>	<b>41</b>

*Tabla 5.6: Factor de Complejidad*  
*Fuente: Elaboración Propia*

Ahora se calcula el factor de ajuste vea tabla 5.7.

<b>FACTOR DE AJUSTE</b>	<b><math>0.65 + 0.01 * \sum f_i()</math></b>
Factor de Ajuste	0.65 + 0.01 * 41
Factor de ajuste	<b>1.06</b>

*Tabla 5.7: Factor de Ajuste*  
*Fuente: Elaboración Propia*

Calculando el Punto Función con los datos obtenidos vea tabla 5.8.

<b>PUNTO FUNCION</b>	<b>CUENTA TOTAL* FACTOR DE AJUSTES</b>
Punto de Función	80*1.06
Punto de Función	<b>84.8</b>

*Tabla 5.8: Punto de Función*  
*Fuente: Elaboración Propia*

### 5.2.2 COSTOS DEL SOFTWARE DESARROLLADO

Se utilizó como punto función PF=84.8 a continuación, realizamos la conversión de punto función a miles de líneas de código mediante la siguiente tabla.

LENGUAJE	FACTOR LDC/PF
Java	53
<b>JavaScript</b>	47
Visual Basic	46
ASP	36
Visual C++	34
PHP	12
Ensamblador	320
C	150

*Tabla 5.9: Factor LCD/PF de lenguajes de programación  
Fuente: (QSM, 2017)*

Ahora aplicamos las formulas básicas de esfuerzo, tiempo calendario y personal requerido. Las fórmulas de COCOMO II que utilizaremos serán las siguientes:

$$E_D = 2,4(KLDC)^{1,05}$$

$$T_D = 2,5 (E_D)^{0,38}$$

Donde:

$E_D$ : Esfuerzo aplicado en personas por mes.

$T_D$ : Tiempo de desarrollo en mes.

KLDC = Número estimado de líneas de código distribuidas.

LDC = PF\*Factor LDC

LDC = 84.8\*47 = 3985

KLDC = 3.98

### **Orgánicos**

Relativamente sencillos y pequeños, en los que trabajan equipos pequeños con experiencia, sobre un conjunto de datos poco rígidos.

### **Semiacoplados**

Proyectos intermedios (en tamaño y complejidad) en los que participan equipos con variados niveles de experiencia, deben satisfacer requisitos medio rígidos.

### **Empotrados**

Proyectos bastantes complejos, en los que apenas se tienen experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

El proyecto al que se adecua es al Orgánico por lo tanto remplazando en las formulas anteriores tenemos:

$$E_D = 2.4 * (3.98)^{1.05} = 10.24 \text{ [personas/mes]}$$

$$T_D = 2.5 * (10.24)^{0.38} = 6.05 \text{ [meses]}$$

El personal requerido para el desarrollo del proyecto se obtiene de la siguiente formula

$$\text{Numero de programadores} = \frac{E_D}{T_D} = \frac{10.24}{6.05} = 1.69$$

Por lo tanto, se necesitan 2 programadores para el desarrollo del proyecto. El costo de salario por programador es de 200 \$us/mes. Por lo tanto, con este dato la estimación del costo del software se calculará con la siguiente formula.

$$\text{Costo Software} = \# \text{ de programadores} * \text{Salario de Programador} * \# \text{ de meses}$$

$$\text{Costo Software} = 2 * 300 * 7 = 4200 \text{ $us}$$

Por lo tanto, el costo de desarrollo del software es de 4200\$us

### 5.2.3 COSTOS DE LA ELABORACION DEL PROYECTO

Los costos de elaboración del proyecto se refieren a los costos de estudio del sistema en la etapa de análisis, estos costos se muestran en la siguiente tabla.

DESCRIPCIÓN	COSTO (\$US)
Análisis y diseñado del Proyecto	250
Material de escritorio	50
Otros	100
<b>TOTAL</b>	<b>400</b>

*Tabla 5.10. Costos de elaboración del proyecto  
Fuente: Elaboración propia*

Por lo tanto, el costo total para la elaboración del proyecto es 4600\$us.

## **CAPITULO VI CONCLUSIONES Y RECOMENDACIONES**

### **6.1 CONCLUSIONES**

El desarrollo e implementación de la Aplicación móvil para el control de inventario de la empresa Coderoad, fue acorde y seguimiento de los requisitos establecidos al principio de la propuesta con la empresa, cumpliendo todos los requerimientos y tareas de cada iteración, por lo cual no existió un retraso en el desarrollo de la aplicación, con todo esto se puede afirmar que el sistema cumple con los objetivos trazados y satisfacen las necesidades del cliente.

- ✓ Se ha desarrollado una librería capaz de interactuar entre el dispositivo móvil con el Reader TSL.
- ✓ Se ha desarrollado un flujo de asociación para poder asociar el objeto a inventariar con una persona.
- ✓ Se ha desarrollado un módulo donde el usuario puede encontrar toda la información acerca de los objetos dañados.
- ✓ Se ha desarrollado un módulo para que el usuario pueda leer Códigos de Barra.
- ✓ Se ha desarrollado un módulo para que el usuario pueda leer etiquetas RFID.

### **5.2 RECOMENDACIONES**

De acuerdo a los logros obtenidos del presente proyecto de grado, surgen algunas Recomendaciones del Sistema.

- ✓ Incorporar antenas RFID para obtener el lugar exacto de cada objeto sin necesidad que este sea leído por un Reader.
- ✓ Incorporar etiquetas RFID dentro del membrete de cada persona para hacer el marcado de entrada y salida del personal.
- ✓ Incorporar más módulos para completar la aplicación, como ser las estadísticas de objetos dañados.

- ✓ Para mejorar la usabilidad se debería realizar una retroalimentación de las críticas de los usuarios finales, para poder mejorarlo.
- ✓ Para mejorar el rendimiento de la aplicación se puede llevar el código de Ionic v1 a la versión de Ionic v2.



## BIBLIOGRAFIA

- Ahson, S., & Ilyas, M. (2008). *RFID HandBook*. United States of America: Taylor & Francis Group, LLC.
- AIM, G. (2004). AIM Global Standard for the use if the AIM RFID Mark and Index to Identify RFID Enabled labels. *AIM Global Standard for the use if the AIM RFID Mark and Index to Identify RFID Enabled labels*. AIM Inc.
- Alvarado, J. (Enero de 2008). Sistema de Control de Acceso con RFID. Mexico DF, Mexico .
- Amaya, Y. (23 de Julio de 2013). <http://www.uelbosque.edu.co/>. Obtenido de Universidad del Bosque:  
[http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista\\_tecnologia/volumen12\\_numero2/12Articulo\\_Rev-Tec-Num-2.pdf](http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista_tecnologia/volumen12_numero2/12Articulo_Rev-Tec-Num-2.pdf)
- Baik, J. (2001). *COCOMO II Model Definition Manual*. Center for Software Engineering, USC.
- Ballou, R. (2005). *Logistica: Administracion de la cadena de suministro*. Mexico: 5ta Edicion Pearson Prentice Hall.
- Benitez, A. A. (2016). *Mecanismo de personalizacion de etiquetas rfid con firma dinamica JetFD EPC*. La Paz: TESIS DE MAESTRIA.
- Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodriguez, P. (2009). <http://www.adamwesterski.com/>. Obtenido de Adam Westerski:  
[http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Boehm . (1995). *The COCOMO 2.0 Software Cost Estimation Model*.
- Calle, I. (2014). Diseño e implementacion de un aplicativo movil para la consulta de servicios por parte de los clientes de pequeña y mediana empresa de una telefonica de pereira. *Diseño e implementacion de un aplicativo movil para la consulta de servicios por parte de los clientes de pequeña y mediana empresa de una telefonica de pereira*. Colombia: Universidad Tecnológica de Pereira.
- Campbell, C. (Diciembre de 2015). *Shopify*. Obtenido de Shopify:  
<https://es.shopify.com/blog/71071813-advertencia-estas-perdiendo-dinero-al-no-utilizar-estas-8-tecnicas-de-manejo-de-inventario>
- Chanrasekaram, S., & Oustad, M. (23 de Mayo de 2008). Securing Assets with RFID. Grimstad, Noruega.

- Clou. (2017). Antena RFID. *Antena RFID*. Clou IOT Technologies Co.,Ltd.
- Cuadrat, C. (Julio de 2012). Estudio sobre Evaluacion de la usabilidad movil y propuesta de un metodo para tests de usabilidad cuantitativos basados en tecnicas de eyetracking. *Estudio sobre Evaluacion de la usabilidad movil y propuesta de un metodo para tests de usabilidad cuantitativos basados en tecnicas de eyetracking*. L rida, Espa a.
- Duran, B. G., & Lozano, C. (2012). Dise o de un sistema identificador de placas de vehiculos, utilizando tecnologia RFID. Caso: Gobierno Autonomo Municipal de La Paz. La Paz.
- Enriquez, J., & Casas, S. (23 de Junio de 2013). USABILIDAD EN APLICACIONES M VILES. *USABILIDAD EN APLICACIONES M VILES*. Argentina.
- FDA. (08 de Diciembre de 2016). Obtenido de Food and Drug Administration: [http://www.fda.gov/oc/initiatives/counterfeit/rfid\\_cpg.html](http://www.fda.gov/oc/initiatives/counterfeit/rfid_cpg.html)
- Finkenzeller, K. (2010). *RFID Handbook*. United State of America: Wiley.
- GitHub. (2017). *GitHub*. Obtenido de GitHub: <https://github.com/>
- Google. (2016). *Firebase*. Obtenido de Firebase: <https://firebase.google.com>
- Google. (2017). *Google Developer*. Obtenido de Google Developer: <https://developer.android.com/guide/platform/index.html?hl=es-419>
- Hunt, D., & Puglia, A. (2007). *RFID: A Guide to Radio Frequency Identification*. United State of America: WILEY.
- Hussain, A. (2012). Metric based evaluation of mobile devices. *Metric based evaluation of mobile devices*. M nchester, Inglaterra: University of Salford.
- IO2017. (Mayo de 2017). *Google IO*. Obtenido de Google IO: <https://events.google.com/io>
- Kaur, M., & Sandhu, M. (February de 2011). RFID Technology Pronciples, Advantages, Limitations & Its Applications. *RFID Technology Pronciples, Advantages, Limitations & Its Applications*. International Journal of Computer and Electrical Engineering.
- Koskela, J. (2004). Mobile-D: an agile approach for mobile. *Mobile-D: an agile approach for mobile*. New York.
- Pachon, L. (2015). Formulacion de proyectos para la implementacion de sistemas de control de inventarios mediante tecnologia RFID basado en PMI. Bogota, Colombia.
- QSM. (2017). *Quantitative Software Management*. Obtenido de Quantitative Software Management: <http://www.qsm.com/resources/function-point-languages-table>

Quispe, C., Quispe , E., & Montecinos, L. (2015). Registro y Control de activos fijos muebles en las unidades educativas, utilizando la tecnología RFID Caso: Gobierno Atunomo Municipal de La Paz- Unidad de Bienes Muebles. La Paz.

Salinas, E. (2010). *Aplicación de modelos UML en el diseño de una solución de inteligencia de negocio*. Valencia.

*Software de Comunicaciones*. (2012). Obtenido de Software de Comunicaciones.

Stevens, P., & Pooley, R. (2002). Utilización de UML en Ingeniería del Software con Objetos y Componentes. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. Madrid: Addison Wesley.

Stockman, H. (1948). *Communication by Means of Reflected Power*. Cambridge Field Station.

Wilken, J. (2016). *Ionic in Action*. United Stater of America: Manning Shelter Island.

