

# UNIVERSIDAD PÚBLICA DE EL ALTO

Creada por ley 2115 de 5 de septiembre de 2000 y autónoma por ley 2556 de 12 de noviembre de 2003

## CARRERA DE INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

APLICACIÓN MOVIL DE CONTROL DE INVENTARIOS  
MEDIANTE CODIGO QR  
CASO: CENTRO DE DESARROLLO INTEGRAL BO0450 FUNDACION  
LADISLAO CALANI

Para optar al título de Licenciatura en Ingeniería de Sistemas  
**Mención: INFORMATICA Y COMUNICACIONES**

**Postulante: Rogerd Patty Quispe**

**Tutor Metodológico: Lic. Ing. Marisol Arguedas Balladares**

**Tutor Especialista: Lic. Fredy Alanoca Coareti**

**Tutor Revisor: Lic. Ing. Ramiro Kantuta Limachi**

**EL ALTO – BOLIVIA**

## **DEDICATORIA**

*A Dios que siempre me acompaño en cada momento de mi vida.*

*A mis padres Felipe y Lidia María, por todo el amor, orientación, sus palabras, apoyo y los valores inculcados desde mi infancia.*

*A toda mi familia por apoyarme en los momentos buenos y malos de mi vida, en especial a mis hermanitas Deicy, Lizeth, Limber, Lina y mis queridos sobrinos Jeimy y Reichell. Con esfuerzo se logra todo lo que se puede soñar.*

## **AGRADECIMIENTOS**

*En primer lugar agradezco a Dios por permitirme lograr un objetivo más en mi vida*

*Quiero agradecer a todas las personas que involucran el presente trabajo, mis agradecimientos en especial a la Ing. Marisol Arguedas Balladares por la orientación, apoyo incondicional y seguimiento al presente escrito.*

*Gracias al Lic. Fredy Alanoca Coareti, por la confianza depositada en mi persona, observaciones y sugerencias que permitieron mejorar el presente trabajo, también por su disposición en el seguimiento del trabajo propuesto.*

*Gracias al Ing. Ramiro Kantuta Limachi por su tiempo en la revisión por las observaciones al escrito, también debo resaltar sus concejos y comentarios que me permitieron mejorar y adecuar mi trabajo correctamente.*

*A la Carrera de Ingeniería de Sistemas, Director Ing. David Carlos y mis queridos Docentes.*

*Al C.D.I. BO0450 Fundación Ladislao Calani, Directora Daniela Calani, por su apoyo incondicional.*

*A mis amigos y amigas que me apoyaron y me animaron en cada momento de mi vida.*

*A todos ellos muchísimas Gracias...*

## **RESUMEN**

Este proyecto explica el diseño e implementación de una aplicación para dispositivos móviles para ser utilizados como un medio de control de inventarios mediante el código QR, facilitando reportes de estados y reportes de préstamos reales de activos de la institución. Para este propósito, existen dos tipos de usuarios el administrador, el personal de la fundación y los beneficiarios quienes proporcionan la información. El control se realiza cada vez que ingresa un activo nuevo, cuando se hace algún préstamo o cuando se realiza auditoria, mediante el escaneo del código QR que se le asignara a cada activo. La aplicación tiene que tener una conexión a internet, para poder interactuar con la base de datos. Tanto así como información e imagen de los activos. En el escaneo le permite al usuario ver la foto del activo con se registró, y le provee información real y certera de dicho activo.

Los resultados muestran una buena aceptación y que es de utilidad para la institución, ya que es intuitiva y con una agradable interfaz de usuario.

## Tabla de Contenidos

1.	CAPITULO 1 MARCO REFERENCIAL.....	1
1.1.	Introducción.....	1
1.2.	Antecedentes.....	2
	Institucionales.....	2
	Internacionales.....	3
	Nacionales.....	4
1.3.	Problema Principal.....	5
	Problemas específicos.....	5
1.4.	Objetivo General.....	6
	Objetivos específicos.....	6
1.5.	Limites.....	6
1.6.	Alcances.....	7
1.7.	Justificación.....	7
	Justificación técnica.....	7
	Justificación económica.....	7
	Justificación Social.....	8
1.8.	Aportes.....	8
2.	CAPÍTULO II MARCO TEORICO.....	10
2.1.	Tecnología Móvil.....	10
	Sistemas Operativos para Móviles.....	11
	Android.....	12
	2.1.1.1. Arquitectura de Android.....	13
	2.1.1.1.1. Kernel de Linux.....	13
	2.1.1.1.2. Capa de abstracción de hardware (HAL).....	14
	2.1.1.1.3. Tiempo de ejecución de Android.....	14
	2.1.1.1.4. Bibliotecas C/C++ nativas.....	15
	2.1.1.1.5. Framework de la Java API.....	15
	2.1.1.1.6. Apps del sistema.....	16
	2.1.1.2. Estructura o componente de una aplicación Android.....	16
	2.1.1.3. SDK de Android.....	18
	2.1.1.5. SQLite.....	19
	2.1.1.6. Breve descripción de las versiones de Android.....	19
2.2.	Tecnologías WEB.....	20
	PHP.....	20
	MySQL.....	23
2.3.	Inventarios.....	26
	Tipos de inventarios.....	27
	El inventario Físico.....	27

El inventario de Bienes intangibles.....	28
Sistemas de control de inventario.....	28
Inventario periódico.....	28
Inventario permanente.....	28
Concluir el inventario satisfactoriamente.....	29
2.4. Código QR.....	29
Características Generales.....	30
Tipos de información que puede almacenarse en los códigos QR.....	30
Principales funciones para el código QR.....	31
Almacenamiento y difuncion.....	31
Diferencias entre Códigos QR y Códigos de Barra.....	31
2.5. Metodología XP (Extreme Programming).....	32
Proceso XP.....	34
Fases de Desarrollo de la Programación Extrema.....	35
2.6. Lenguaje De Modelado Unificado (UML).....	37
Concepto sobre UML.....	37
UML    38	
Diagramas del UML.....	40
2.7. Modelo de calidad MCCALL.....	46
Cómo emplear el modelo de MCCALL.....	51
2.8. Modelo Constructivo De Costos (COCOMO II).....	53
Utilización del modelo de composición de aplicaciones.....	53
Modelo de coste de composición de aplicaciones.....	53
2.9. Pruebas (Diseño de casos de Prueba).....	58
2.9.1. Pruebas de Caja Negra.....	59
2.9.2. Pruebas de caja Blanca.....	59
3. CAPITULO III MARCO APLICATIVO.....	60
3.1. Arquitectura de la Aplicación.....	60
Web Service (Servicio Web).....	60
3.2. Fase de Planificación.....	62
Descripción de los interesados.....	62
Requerimientos funcionales del proyecto.....	63
Requerimientos no funcionales del sistema.....	66
Restricciones.....	68
3.3. Fase de Diseño.....	70
Diagrama de casos de uso.....	70
Diagrama de Clases.....	76
3.4. Diseño de Bocetos de las Pantallas de la Aplicación.....	77
Bocetos de la aplicación Móvil.....	78
3.5. Herramientas para la codificación.....	86

Requerimientos de desarrollo.....	86
3.6. Descripción de Pruebas.....	87
Elaboración y Ejecución del plan de pruebas.....	87
Análisis de Resultados.....	89
3.7. Pruebas de Caja Negra.....	90
Inicio de Sesión.....	90
Visualización de Inventarios.....	91
Visualización de activos Prestados.....	91
Visualizar Beneficiarios de préstamos.....	91
Generalización de Reportes.....	91
Generalización de Códigos QR.....	92
Registro de Nuevo Activo.....	92
Registro de Nuevo Préstamo.....	92
Registro de Nuevo Beneficiario/Cliente.....	93
Escaneo del Código QR.....	93
3.8. Prueba de Caja Blanca.....	93
Gestión de Usuarios.....	94
4. CAPITULO IV CALIDAD Y SEGURIDAD.....	95
4.1. ANALISIS DE CALIDAD.....	95
FACTORES DE CALIDAD MCCALL.....	95
Capacidad: Características de operación.....	97
Capacidad: Capacidad de soportar cambios.....	98
Capacidad: Adaptabilidad a nuevos entornos.....	99
Medida de Calidad.....	100
Cálculos de Calidad.....	101
5. CAPITULO V COSTO BENEFICIO.....	102
5.1. Análisis de Costos.....	102
Modelo de coste de composición de aplicaciones.....	102
5.2. BENEFICIO ECONOMICO.....	107
6. CAPITULO VI CONCLUSIONES Y RECOMENDACIONES.....	108
6.1. Conclusiones.....	108
6.2. Recomendaciones.....	109
BIBLIOGRAFIA.....	110
MANUAL DEL USUARIO.....	113
MANUAL TECNICO.....	116
Características de Software y Hardware.....	116

	viii
Descripción de Software.....	116
Descripción de Hardware .....	116
Instalación de la Aplicación.....	117
ARBOL DE PROBLEMAS .....	119
ARBOL DE OBJETIVOS .....	120

**INDICE DE FIGURAS**

<b>Figura 2.1</b> Arquitectura Android	13
<b>Figura 2.2</b> Página de Descarga de SDK Android	19
<b>Figura 2.3</b> Versiones de Android y lanzamientos	20
<b>Figura 2.4</b> Comparación de Código de Barras y Código QR	33
<b>Figura 2.5</b> Características de la Programación Extrema	35
<b>Figura 2.7</b> Reglas de UML	40
<b>Figura 2.8</b> Clase abstracta	43
<b>Figura 2.9</b> Asociaciones	43
<b>Figura 2.10</b> Multiplicidad	44
<b>Figura 2.11</b> Composición y agregación	44
<b>Figura 2.12</b> Generalización	45
<b>Figura 2.13</b> Diagrama de objeto	45
<b>Figura 2.14</b> Diagrama de casos de uso	46
<b>Figura 2.15</b> diagrama de Estados o Actividades	47
<b>Figura 2.16</b> Diagrama de Secuencias	48
<b>Figura 3.1</b> Arquitectura de la aplicación	62
<b>Figura 3.2</b> Casos de Uso General del Sistema de difusión de información institucional	74
<b>Figura 3.3</b> Diagrama de Clases General	79
<b>Figura 3.5</b> Página oficial de Balsamiq Mockups para prototipos	80
<b>Figura 3.6</b> Prototipo Pantalla de Inicio de Sesión	81
<b>Figura 3.7</b> Prototipo de Pantalla Principal Administrativo	81
<b>Figura 3.8</b> Prototipo de Pantalla registro de Usuarios	82
<b>Figura 3.9</b> Prototipo de Pantalla principal de Personal	82
<b>Figura 3.10</b> Prototipo de Pantalla de Registro de Activo	83
<b>Figura 3.11</b> Prototipo de Pantalla (Toma de Fotografía del Activo)	83
<b>Figura 3.12</b> Prototipo de Pantalla de registro de Préstamo de activo	84
<b>Figura 3.13</b> Prototipo de pantalla, Registro de Prestamista	84
<b>Figura 3.14</b> Prototipo de la Realización del Inventario	85
<b>Figura 3.15</b> Prototipo de Reportes del Inventario	85

	x
<b>Figura 3.16</b> Prototipo de Reportes de Prestamos	86
<b>Figura 3.17</b> Prototipo de Reporte de Beneficiarios de Préstamo	86
<b>Figura 3.18</b> Prototipo de Reporte de Usuarios	87
<b>Figura 3.19</b> Prototipo de Planilla de Impresión de Códigos QR	87
<b>Figura 3.20</b> Prototipo de Escaneo de Activo por QR	88
<b>Figura 3.20</b> Prototipo de Reconocimiento de Activo	88
<b>Figura 4.1</b> Relaciones entre las capacidades, factores y métricas según McCall	95

**INDICE DE TABLAS**

<b>Tabla 2.1</b> Criterios de McCall	49
<b>Tabla 2.2</b> Calidad-Precio de los factores de McCall	54
<b>Tabla 2.3</b> Complejidad asociada a las instancias de objetos	57
<b>Tabla 2.4</b> Pesos asociados a los niveles de complejidad	58
<b>Tabla 2.5</b> Ratio de Productividad PROD	59
<b>Tabla 2.6</b> Factores de escala para el modelo de COCOMO II	60
<b>Tabla 2.7</b> Valores de los Factores de escala para el modelo de COCOMO II.	60
<b>Tabla 3.1</b> Matriz de Stakeholders del Proyecto	64
<b>Tabla 3.2</b> Descripción de Smartphone con los requisitos para la aplicación	71
<b>Tabla 3.3</b> Actores del sistema de difusión de información institucional	73
<b>Tabla 3.4</b> Especificación de caso de Uso Numero 1. Ingreso a la aplicación	75
<b>Tabla 3.5</b> Especificación de caso de Uso Numero 2. Registro de Usuarios	75
<b>Tabla 3.6</b> Especificación de caso de Uso Numero 3. Agregar activos	76
<b>Tabla 3.7</b> Especificación de caso de Uso Numero 4. Registro de Prestamos	76
<b>Tabla 3.8</b> Especificación de caso de Uso Numero 5. Registro de prestamistas	77
<b>Tabla 3.9</b> Especificación de caso de Uso Numero 6. Realización de Inventario	77
<b>Tabla 3.10</b> Especificación de caso de Uso Numero 7. Reportes	78
<b>Tabla 3.11</b> Especificación de caso de Uso Numero 8. Registrar devoluciones	79
<b>Tabla 3.12</b> Gamas de Smartphone 2020	91
<b>Tabla 3.13</b> Análisis de resultado del plan de pruebas	92
<b>Tabla 4.1</b> Puntuación de métricas según usuarios del sistema	94
<b>Tabla 4.2</b> Características de operación	96
<b>Tabla 4.3</b> Capacidad de soportar cambios	97
<b>Tabla 4.4</b> Adaptabilidad a nuevos entornos	98
<b>Tabla 4.5</b> Medida de Calidad y Valoración	99
<b>Tabla 5.1</b> Valores para los factores de escala	104
<b>Tabla 5.2</b> Costos de Hardware y Software	105

# 1. CAPITULO 1

## MARCO REFERENCIAL

### 1.1. Introducción.

Los dispositivos móviles se convirtieron en una herramienta indispensable para los universitarios, administrativos, colegiales, etc. Puesto que según estudios pasan una media de tres a cuatro horas diarias (Rius, 2016), razón por la cual es importante desarrollar aplicaciones móviles que sean de utilidad y una herramienta para agilizar las cosas. En la vida universitaria desde que se ingresa a la carrera de Ingeniería de Sistemas, siempre se sueña con el día en que se pueda ser un desarrollador de Sistemas, ya sea en la creación o en el mejoramiento de estas, empleando todo lo aprendido en el transcurso de los semestres, y realimentando los conocimientos de una manera práctica.

Para desarrollo de este trabajo se comienza por reconocer los principales problemas de la Organización y sus respectivas causas en cada una de sus áreas: en conjunto con la Directora de la fundación, se llega a identificar las áreas de compra, su almacenamiento, préstamo de activos por parte de Beneficiarios y Personal, y la pérdida de tiempo al procesar estas, como causante de la mayoría de los problemas y por ende son estas las primeras que deben ser mejoradas en pos de solucionar sus problemas.

Diseñar la aplicación móvil de inventarios basados en las necesidades de la institución. Se trata de ofrecer a la Organización un futuro de desarrollo, la aplicación móvil en plataforma Android, se constituirá en una herramienta apropiada y calificada, con un fácil uso, para cumplir con los objetivos propuestos de mejoramiento para su control de inventarios, reportes actualizados, identificación directa mediante una fotografía.

El proyecto tiene por objeto desarrollar un sistema de control de inventarios para el entorno Android mediante el escaneo de QR. Para lograr el cometido, se utilizará XP (Extreme Programming) como metodología, UML para el modelado, además de la tecnología JAVA,

PHP, phpMySQL y MySQL para la parte de la programación, y las herramientas Android Studio, Visual Studio Code,

## **1.2. Antecedentes.**

### **1.2.1. Institucionales.**

Compassion International, se creó un 26 de septiembre en 1952 es una ONG que apoya a niños y familias en situación de pobreza mediante sociedad con iglesias cristianas evangélicas. Mediante tres programas: CSP-Supervivencia Infantil (llegan a Mamas embarazadas), CDSP-Patrocinio (niños y adolescentes) y GAP-Liderazgo (jóvenes universitarios). A fin de que puedan conocer el evangelio y vivir una vida plena en base a principios bíblicos.

Esta ONG tiene la Misión de Responder a la gran comisión, Compassion International Existe como defensora, para liberarlos de su pobreza espiritual, económica, social y física y para ser capacitarlos a fin de que lleguen a ser cristianos adultos, responsables y realizados.

El Centro de Desarrollo Integral, conocido por su sigla CDI BO0450 es una fundación que apoya a niños, niñas, adolescentes, jóvenes y señoritas universitarios. Su Misión es impulsar a los graduados más sobresalientes del programa de Patrocinio hacia el cumplimiento del potencial que Dios les dio. Los estudiantes son equipados para tener una profesión, son mentoreados en su vida cristiana y se inculca en ellos una pasión por servir que transforma sus comunidades y naciones. Su Visión es que el CDI BO0450 será el ministerio de Liderazgo más importante. Los graduados serán reconocidos por su carácter cristiano e influencia se movilizarán para terminar con el ciclo de pobreza que quebranta naciones. Para el 2025 el CDI BO0450 crecerá hasta tener 2000 beneficiarios. Tendrá una escuela y colegio, basado en principios y valores cristianos, formando Estudiantes comprometidos con la sociedad.

### **1.2.2. Internacionales.**

(Márquez Rodríguez y Ponguillo Quinde, 2012), con un proyecto de grado titulada: “Aplicación De Un Sistema De Inventario Para El Control De Productos De La Empresa Kast S.A.”, El siguiente proyecto de tesis trata sobre la Aplicación de un sistema de Inventarios en el control de productos de la Empresa KAST S.A., Distribuidora y ventas de pinturas de la ciudad de Guayaquil. El problema presentado se dio por no haber tenido existencias, lo que provocó la pérdida de un cliente y el incremento de los costos. Se observó que dentro del área de bodega las operaciones del flujo de entradas y salidas del producto no las realiza el personal competente, por lo tanto se decide plantear la propuesta planteada para mejorar los procesos administrativos, así como la elaboración de reportes relacionados que se efectúan en dicho departamento, permitiéndole a la organización tener un rendimiento óptimo, facilitar el trabajo al personal que labora en dicho departamento y posteriormente y así poder aumentar el nivel de servicio saber la exactitud de las unidades suficientes que se tienen y así poder aumentar la utilidad de la empresa.

(Gonzales torrado y Sanchez Barajas, 2010), con la Tesis titulada: “Diseño De Un Modelo De Gestión De Inventarios Para La Empresa Importadora De Vinos Y Licores Global Wine And Spirits Ltda.”, El presente Trabajo de Grado titulado “Diseño de un modelo de gestión de inventarios para la empresa importadora de vinos y licores Global Wine and Spirits LTDA” surge de la necesidad de mejorar la situación de ésta empresa, caracterizada por los problemas fundamentales de desabastecimiento de mercancía y roturas de inventario presentes en la cadena de suministros, que en conjunto generan altos niveles de demanda insatisfecha.

A los problemas mencionados, se adiciona una situación de iliquidez ocasionada por las políticas de desembolso para compras y recaudo de ventas establecida, que afecta dos procesos fundamentales del flujo del producto dentro de la cadena de suministros: el despacho de la mercancía por la casa proveedora, y la nacionalización de la mercancía para su comercialización dentro del país. El análisis del panorama descrito, establece el diseño

de un modelo de inventarios integral, que proporcione mejoras a la situación operacional y financiera de la empresa, desde el proceso de la elaboración del pronóstico para la gestión de las órdenes de compra, hasta su distribución regional.

(Alfonso Fernández y Díaz Guzmán, 2008), con el proyecto de grado titulado: “Sistema De Control De Inventarios Y Facturación De La Licorería Tropical Club Girardot”, el proyecto se generara un cambio mayor que agilizara los procesos de ventas y pago de los productos que la licorería tiene al ofrecimiento y/o antojo de los clientes, además se obtendrá facilidad, eficacia y confiabilidad al momento de realizar, manipular y tener en cuenta el inventario, debido a que actualmente, este se toma manualmente y por obvias razones, el proceso se toma más lento. Por otro lado, también se pueden generar estrategias de beneficio, cuyos favorecidos serán los clientes que frecuenten repetidamente la licorería aquellos que demuestran preferencias de elección al venir a este establecimiento y más de no otro.

### **1.2.3. Nacionales.**

(Hugo Hernan Aquino Achumiri, 2018), con el proyecto de grado titulado “Aplicación web para el registro y control de compras, ventas e inventarios, para la asociación de productores en metalmecánica sur, METAL SUR”, En la asociación se desarrolló e implemento una aplicación web que permite realizar el registro de sus compras, ventas y realizar un control adecuado de su inventario de forma automatizada. Con la implementación del nuevo sistema, se logró mejorar el proceso de registro de compras, ventas y el manejo de su inventario de los materiales y artículos existentes. Toda esta información se encuentra almacenada en una base de datos, además se logró reducir el tiempo en la elaboración de reportes con respecto al número de existencias en el inventario. Para alcanzar este objetivo, nos apoyamos en la metodología de desarrollo de software ágil XP (programación extrema) por su gran adaptabilidad en todo el proceso de desarrollo. Cuenta con un modelo de desarrollo incremental con entregas rápidas y retroalimentación diaria por parte del cliente.

(Jesús Hermogenes Valle Quispe, 2009), con el proyecto de grado titulado: “Sistema de control y seguimiento de inventario de fármacos para la Clínica San Damián”, el proyecto automatizo las actividades cotidianas en la organización, cabe mencionar el vertiginoso avance de las telecomunicaciones y el progreso que han experimentado las ciencias informáticas que obliga a estar a tono y entrar al moderno mundo de la tecnología, ser competitivos y no que darse relegados en las tareas que proporcionan beneficios para proyectarse al futuro. El análisis hecho en el almacén, han identificado claramente falencias que afectan el funcionamiento de la Clínica “San Damián”, situación que se ha hecho evidente en la forma como se ejecutan los procesos y funciones propias de estas áreas. Son los aspectos importantes en la elaboración y diseño del sistema de información, así satisfacer los requerimientos de los usuarios y mejoras las tareas de inventarios de la clínica que se especializa en el área de salud.

### **1.3. Problema Principal.**

El Centro de Desarrollo Integral BO0450 fundación Ladislao Calani, en la actualidad realiza el control de entrada y salida de los activos de la Institución, de manera manual y en el transcurrir del tiempo, los registro y préstamos que se realiza a beneficiarios o Personal de la fundación para realizar actividades, los activos no son devueltos en el plazo establecido el cual causa inconvenientes al momento de encontrarlos y pérdida económica en la reposición del activo. Y además que afecta en el trabajo de Auditoria, se necesita un control absoluto de los mismos.

#### **Problemas específicos.**

- No existe control de los activos ya que es manual.
- No es posible obtener información de los activos y el estado de los mismos.
- No se tiene información de los infractores, es decir, los que no devolvieron los activos.
- No es posible obtener informes de los préstamos realizados.
- Al realizar la auditoria hay mucha demora en la ubicación de los activos de la Organización.

- No hay respaldo de los procesos manuales y propensos a sufrir pérdidas en el proceso de auditoría.

#### **1.4. Objetivo General.**

Implementar una Aplicación Móvil de Control De Inventarios Mediante Código QR del Centro de Desarrollo Integral BO0450 Fundación Ladislao Calani para facilitar y tener una información actualizada y controlada de los activos.

#### **Objetivos específicos.**

- Diseñar un modelo Relacional para el registro de activos y demás procesos.
- Construir un módulo que emita información de los activos y el estado de los mismos.
- Desarrollar un módulo para el control de los infractores
- Construir un módulo que emita información de activos prestados y sus beneficiarios.
- En el modelo Relacional incluir la ubicación actual del activo y con qué persona está asignada.
- Construir un módulo de información de inventarios para auditorías.

#### **1.5. Limites**

El presente proyecto controla el inventario del Centro de Desarrollo Integral Bo0450 Fundación Ladislao Calani, mediante el escaneo del código QR que tendrá cada activo.

- La aplicación no se encontrara en Play Store o tiendas de aplicaciones.
- No realiza ventas de Activos.
- No realiza el control del personal de la institución.
- No genera depreciaciones.

## **1.6. Alcances**

Los alcances del proyecto abarcan los siguientes módulos:

- Módulo de Acceso y sincronización de sistema. Este se encarga del acceso de solamente al encargado de inventario.
- Módulo de Registro de nuevos activos. Este se encarga de actualizar la base de datos, y tomarlas una imagen.
- Módulo de Registro de Prestamos. Este se encarga de actualizar la lista de préstamos, escaneándola con el código QR.
- Módulo de Etiquetación de Inventario. Este se encarga de poder procesar los códigos QR con la base de datos.
- Módulo de Presentación de reportes. Este es el encargado de generar resultados debidamente organizados del inventario actualizado.
- Módulo de Presentación de Reportes de Préstamos. Este es el encargo de sincronizar los activos con los préstamos, y generar los Reportes de los mismos.

## **1.7. Justificación.**

Las justificaciones son desarrolladas de acuerdo a tres aspectos: técnica, económica y social.

### **Justificación técnica.**

El proyecto a desarrollar, se realiza por la necesidad que tiene la Institución, ya que no cuenta con un buen control de activos, optimizando así los servicios que presta el mismo.

El sistema realiza un control de inventarios, mediante el escaneo del código QR.

### **Justificación económica.**

El CDI BO0450 cuenta con una Tableta inteligente, que cumple con los requerimientos para que la aplicación móvil pueda tener una perfecta incorporación para su uso, y no generar más gastos.

Al automatizar el proceso de administración de los inventarios, ya no se requieren material físico, un fin de papeles, notas y fichas de préstamos. Con el este sistema disminuirán de manera considerable permitiendo ahorrar estos y otros recursos.

Además, la aplicación dará reportes de los beneficiarios o del personal que se tomaron prestados algún activo de la institución, se disminuirán aquellos montos de recaudo del personal encargado para su reparación del mismo, que se desaparecían porque no había un responsable.

### **Justificación Social**

La institución lograra minimizar perdidas y maximizar los ingresos con los activos para el bienestar y mejor aprendizaje de los niños beneficiarios.

El presente proyecto promoverá al desarrollo tecnológico, mejorando, facilitando e incentivando a la creatividad. El CDI BO0450 es una organización de servicio, está orientada al beneficio social de los niños, niñas, adolescentes y jóvenes. Al realizar este tipo de mejoras en el sistema de inventarios, la calidad educativa que se ofrece los mayores beneficiados son los beneficiarios, la institución (Directora, Patrocinio y tutores) y la iglesia.

### **1.8. Aportes**

Gracias a los conocimientos de la Ingeniería de Sistemas, hoy tenemos muchos sistemas de control de inventarios que realizan tareas que anteriormente tenían que hacerse manualmente y consumían una gran cantidad de tiempo, por eso el desarrollo de este sistema como la primera herramienta y no la última que contara la institución, para su desarrollo tecnológico disfrutando así de operaciones automatizadas, sistematizadas y optimizadas.

Además de desarrollar nueva tecnología para controlar mejor los inventarios incentivamos al uso de nuevas herramientas como es el Código QR, nos permite tener una mejor veracidad de activos.

Diseño, desarrollo e implementación de sistemas, aplicaciones, modelos u otros. Gracias a ello, las empresas pueden disfrutar de operaciones con mejores productos y servicios. Así mismo mejora el proceso de toma de decisiones, lo que es vital para las instituciones que esperan perdurar en el tiempo y ser estables.

El diseño e implementación de programas que facilitan y optimizan procesos de enseñanza y aprendizaje.

Todo el trabajo de Grado es único y es flexible, escalable. Características que los sistemas gratuitos y otros no tienen.

## 2. CAPÍTULO II

### MARCO TEORICO

#### 2.1. Tecnología Móvil.

(SUPERTEL, Revista Institucional, 2012) La tecnología móvil está directamente ligada a la comunicación o telefonía móvil es a la que nos vamos a referir para el presente proyecto.

“Las comunicaciones móviles sin duda alguna han experimentado un enorme crecimiento desarrollándose diversas tecnologías y sistemas para dar servicios de comunicación inalámbrica. En el Ecuador el servicio móvil inicia a finales de 1993 con la entrada en el mercado de CONECEL S.A. (Portada Celular, luego CLARO) y OTECEL S.A. (Al inicio Bellsouth y actualmente denominada Movistar), manteniéndose el dominio de estas dos empresas hasta el año 2003 cuando entro en operaciones una tercera operadora TELECSA (al inicio Alegro Actualmente CNT E.P.)”.

(SUPERTEL, Revista Institucional, 2012) En el pasado las empresas de telecomunicaciones brindaban un solo servicio: telefonía, audio y video por suscripción, servicios portadores y servicios de valor agregado. En la actualidad los servicios han convergido de tal manera que un mismo proveedor de servicios dentro de una misma infraestructura de telecomunicaciones, puede brindar múltiples servicios. Esta convergencia tecnológica de un dispositivo electrónico que proporciona funcionalidades de telefonía y ofrece asistencias iguales a las de una computadora, hace posible el concepto de Smartphone (teléfonos inteligentes) con los que se puede realizar compras, enviar y recibir correos electrónicos, escuchar música, ver videos, acceder a redes sociales, además de hablar por teléfono, es posible con un solo dispositivo. La gran aceptación del mercado hacia estos aparatos ha permitido que las empresas líderes en tecnología vean atractiva esta plaza, direccionando su trabajo al desarrollo de aplicaciones personales y empresariales que satisfagan al usuario.

Definitivamente la posibilidad de acceso a internet (internet móvil) es el factor que más ha incidido para que los Smartphone logren tener el nivel de penetración alcanzado en el mercado,

la sensación de conectividad que ofrecen a las redes sociales, así como el acceso a mensajería instantánea hace que los usuarios se ven tentados por estos dispositivos. Por otra parte, las numerosas aplicaciones disponibles para teléfonos inteligentes le dan un valor agregado para quienes buscan entretenimiento en dispositivos móviles.

(Vique, 2012) Todas estas aplicaciones se ejecutan dentro de un ecosistema extendiendo varios factores que lo afectan como son la infraestructura de la aplicación, el sistema operativo, los métodos de entrada de información, los propios usuarios, los canales de distribución de la aplicación, etc. En el caso de las aplicaciones móviles, el ecosistema es aún más heterogéneo que en el resto de desarrollos. Pueden ejecutarse en diferentes tipos de dispositivos, ya sean un móvil antiguo o bien en un actualizado ya sea un Smartphone o una Tablet los cuales se encuentran diseñados bajo una plataforma informática y dependiendo de la marca, en diferentes sistemas operativos, por lo tanto para conseguir un desarrollo satisfactorio es ideal conocerlos más profundamente.

### **Sistemas Operativos para Móviles.**

Detallaremos algunos conceptos tomados (ver bibliografías) de un sistema operativo móvil.

(Wikipedia, Sistema operativos mobile, 2014) “un sistema operativo (SO) móvil controla un dispositivo móvil al igual que los PC utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.”

(World, 2017) “el sistema operativo (SO) móvil de un teléfono o tableta realiza la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez, lo ejecuta con mayor inteligencia. Una de las cualidades más atractivas de un sistema operativo móvil es la rapidez con la que en general se desempeña.”

(El País, 2013) “A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el segundo trimestre de 2014 era el siguiente:

- **Android** “84,7 (En países como España las diferencias son más significativas, donde Android tiene el 90% de la cuota de mercado)”
- **iOS** 11,7%
- **Windows Phone** 2,5%
- **BlackBerry OS** 0,5%
- **Otros** 0,6%
- **Firefox OS** Disponible
- **Ubuntu Touch** (que está en desarrollo)

(Zona, 2014) Android tiene la mayor cuota, desde enero 2011, con más de la mita del mercado, experimento un creciente aumento y en solo dos años (2009 a comienzos de 2011) han pasado de ser el SO móvil más utilizado”

### **Android.**

“Android es una solución completa de software de código libre (GNU Linux3) para teléfonos y dispositivos móviles. Es un paquete que engloban un sistema operativo, un “Runtime4” de ejecución basado en Java5 , un conjunto de librerías en bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Android se distribuye bajo una licencia libre que permite la integración con soluciones de código propietario”.

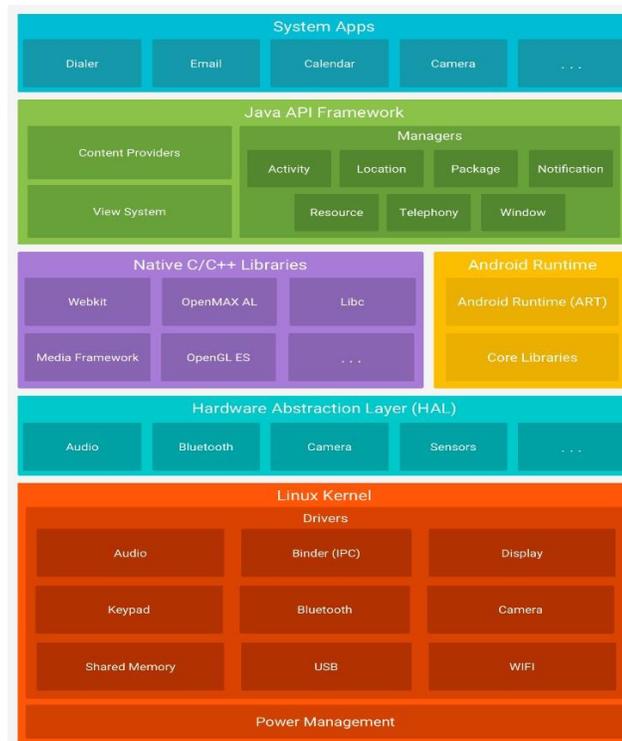
“Android surge como resultado de la “Open Handset Alliance” (Open Handset Alliance, 2014) un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil y un compromiso de comercializar principalmente por Google (tras la compra de Android Inc. En 2015) (Blanco, 2012)” “como se había descrito anteriormente, Android está

basado en Linux, disponiendo de un Kernel6 en este sistema y utilizando una máquina virtual sobre este Kernel que es la responsable de convertir el código escrito en Java de”.

### 2.1.1.1. Arquitectura de Android.

(Developers.android.com, 2018) Androide es una pila de software de código abierto basado en Linux para una variedad de dispositivos y factores de forma. En el siguiente diagrama se muestran los componentes principales de la plataforma Android.

*Figura 2.1 Arquitectura de Android*



Fuente: (Elaboración propia)

#### 2.1.1.1.1. Kernel de Linux.

La base de la plataforma Android es el kernel de Linux. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocessos y la administración de memoria de bajo nivel.

El uso del kernel 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 kernel conocido.

#### **2.1.1.1.2. Capa de abstracción de hardware (HAL)**

Capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la Java API de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de bluetooth. Cuando el framework de una API realiza una llamada para acceder a hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.

#### **2.1.1.1.3. 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.

Estas son algunas de las funciones principales del ART:

- Compilación ahead-of-time (AOT) y just-in-time (JIT);
- Recolección de elementos no usados (GC) optimizada;
- Mejor compatibilidad con la depuración, como un generador de perfiles de muestras dedicado, excepciones de diagnóstico detalladas e informes de fallos, y la capacidad de establecer puntos de control para controlar campos específicos.

Antes de Android 50 (nivel de API 21), Dalvik era el tiempo de ejecución del sistema operativo. Si tu app se ejecuta bien en el ART, también debe funcionar en Dalvik, pero es posible que no suceda lo contrario.

En Android también se incluye un conjunto de bibliotecas de tiempo de ejecución centrales que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java; se incluyen algunas funciones del lenguaje Java 8, que el framework de la Java API usa.

#### **2.1.1.1.4. 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++. La plataforma Android proporciona la API del framework de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps. Por ejemplo, puedes acceder a OpenGL ES a través de la Java OpenGL 7API del framework de Android para agregar a tu app compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D.

Si desarrollas una app que requiere C o C++, puedes usar el NDK 8de Android para acceder a algunas de estas bibliotecas de plataformas nativas directamente desde tu código nativo.

#### **2.1.1.1.5. Framework de la Java API.**

Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- Un sistema de vista enriquecido y extensible que puedes usar para compilar la IU de una app; se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de recursos que te brinda acceso a recursos sin código, como strings localizadas, gráficos y archivos de diseño.

- Un administrador de notificaciones que permite que todas las apps muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que administra el ciclo de vida de las apps y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las apps accedan a datos desde otras apps, como la app de Contactos, o compartan sus propios datos.

Los desarrolladores tienen acceso total a las mismas API del framework que usan las apps del sistema Android.

#### **2.1.1.1.6. 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. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elige instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Settings del sistema).

Las apps del sistema funcionan como apps para los usuarios y brindan capacidades claves a las cuales los desarrolladores pueden acceder desde sus propias apps. Por ejemplo, si en tu app se intenta entregar un mensaje SMS, no es necesario que compiles esa funcionalidad tú mismo; como alternativa, puedes invocar la app de SMS que ya está instalada para entregar un mensaje al receptor que especifiques.

#### **2.1.1.2. Estructura o componente de una aplicación Android.**

En Java o .NET estamos acostumbrados a manejar conceptos como ventana, control, eventos o servicios como los elementos básicos en la construcción de una aplicación.

Pues bien, en Android vamos a disponer de esos mismos elementos básicos, aunque con un pequeño cambio en la terminología y el enfoque. Repasemos los componentes principales que

pueden formar parte de una aplicación Android (Por claridad, y para evitar confusiones al consultar documentación en inglés, intentaré traducir lo menos posible los nombres originales de los componentes). (SGOLIVER, 2010)

- **Activity:** Las actividades (activities) representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana o pantalla en cualquier otro lenguaje visual.
- **View:** Las vistas (view) son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los controles de Java o .NET.
- **Service:** Los servicios (service) son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son similares a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo, actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (p.ej. actividades) si se necesita en algún momento la interacción con del usuario.
- **Content Provider:** Un proveedor de contenidos (content provider) es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido.
- **Broadcast Receiver:** Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: “Batería baja”, “SMS recibido”, “Tarjeta SD insertada”,...) o por otras aplicaciones (cualquier aplicación puede generar mensajes (intents, en terminología Android) broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo).
- **Widget:** Los widgets son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (home screen) del dispositivo Android y recibir

actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

- **Intent:** Un intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

### 2.1.1.3. SDK de Android.

Como se puede ver anteriormente, Android es una plataforma de software libre, por lo que cuenta con Kit de desarrollo de Software o SDK disponible para todo desarrollador o para alguien que desee experimentar que incluye: el conjunto completo del API soportado por este sistema para crear, probar y depurar aplicaciones para Android. Podemos descargarlo de la página web de Android: <http://developer.android.com/intl/es/sdk/index.html>.

Figura 2.2 Página de Descarga de SDK Android



Fuente: (<http://developer.android.com/intl/es/sdk/index.html>)

### **2.1.1.5. SQLite.**

(Wikipedia, SQLITE, 2014) “SQLite es un sistema gestor de base de datos relaciona (RDBMS). Lo que hace único a SQLite es que se considera una solución embebida. La mayoría de los sistemas de gestión de base de datos como Oracle, Mysql y SQL Server son procesos de servidor autónomos que se ejecutan independientemente. SQLite es en realidad una librería que esta enlazada dentro de las aplicaciones. Todas las operaciones de base de datos se manejan dentro de la aplicación. Todas las operaciones de base datos se manejan dentro de la aplicación mediante llamadas y funciones contenidas en la librería SQLite, hace mucho más fácil tratar grandes bases de datos en comparación con otras soluciones más convencionales de base de datos.”

SQLite se basa en el Lenguaje Estructurado de Consultas (SQL11); el mismo lenguaje que utilizan las mayorías de RDBMS. Es una gran ventaja si se conoce SQL ya que la mayoría de las sentencias se formulan de la misma forma. Android tiene integrado en el propio sistema una API completa que nos permite manejar la Base de Datos en SQLite.

### **2.1.1.6. Breve descripción de las versiones de Android.**

El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial (de prueba), Android 1.0, fue lanzada en septiembre de 2008. Android es un sistema operativo móvil desarrollado por Google y la Open Handset Alliance, y ha visto un número de actualizaciones a su sistema operativo base desde su lanzamiento original. Estas actualizaciones típicamente corrigen fallos de programa y agregan nuevas funcionalidades. Desde abril de 2009, las versiones de Android han sido desarrolladas bajo un nombre en clave y sus nombres siguen un orden alfabético:

Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat, Oreo, Pie, Android 10 y el último lanzamiento del Android 11 en el 8 de septiembre de 2020.

Figura 1.3 Versiones de Android y lanzamientos

Nombre	Número de versión	Fecha de lanzamiento	Nivel de API
Sin nombre en clave oficial	1.0	23 de septiembre de 2008	1
	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1.2	15 de junio de 2016	24 – 25
Oreo	8.0 – 8.1	21 de agosto de 2017	26 – 27
Pie	9.0	6 de agosto de 2018	28
Android 10	10.0	3 de septiembre de 2019	29
Android 11	11.0	8 de septiembre de 2020	30

Fuente: (Elaboración propia)

## 2.2. Tecnologías WEB

### PHP

PHP, acrónimo de PHP Hypertext Preprocessor, es un lenguaje de programación que se utiliza principalmente para el desarrollo de sitios web, pero que para muchos es un lenguaje de propósito general, y el uso que se le da dependerá en parte de los que necesite el programador. Entre las características que hacen de PHP un lenguaje popular y muy poderoso para desarrollar aplicaciones, podemos citar las siguientes:

- Programación de páginas dinámicas en servidores.
- Programación de aplicaciones de escritorio con GTK (PHPGTK).
- Soporte para trabajar con múltiples bases de datos.

- Soportes para múltiples plataformas
- Soporte para múltiples Servidores
- Facilidad de aprendizaje.
- Portabilidad de código entre diferentes plataformas.
- Total libertad para distribuir las aplicaciones.

Para obtener una copia, deberemos ingresar a su sitio web, **www.php.net-downloads**, y seleccionar la opción acorde con nuestro sistema operativo.

#### **2.2.1.1. PHP 7.4.12**

Actualmente PHP va por la versión de 7.4.12, que se encuentra disponible en su página, esta es una versión de revisión de seguridad que contiene una corrección de seguridad y muchas correcciones de errores.

#### **2.2.1.2. Extensiones en PHP**

Cuando programamos en un lenguaje, no solo en PHP, nos valemos de funciones o procedimientos, ya sea para resolver problemas o modular el código y hacerlo más legible o reutilizable sin escribir lo mismo más de una vez.

Las extensiones no son ni más ni menos que conjuntos de funciones disponibles para programar, pero para poder ser más precisos podríamos dividir las funciones en dos grupos. Por un lado, están las que vienen incorporadas con el lenguaje, llamadas **built in**, Por otro lado, tenemos las que se encuentran en las bibliotecas añadidas, que debemos incluir en el sistema de manera específica, estas son las extensiones o bibliotecas. Se podría decir que para utilizar ciertas funciones hay que extender el lenguaje. Las funciones de PHP pueden agruparse por funcionalidad: podríamos encontrar con una extensión para manipular cadenas de caracteres, otra para acceder a base de datos, otra para trabajar con archivos XLS y demás. (Minera, 2011)

### 2.2.1.3. Base de Datos

El lenguaje de PHP soporta, entre otras, el manejo de las siguientes bases de datos:

- dBase
- Informix
- InterBase
- MS SQL Server
- MySQL
- MsqI
- Oracle
- PostgreSQL
- SyBase

Además con ODBC se puede acceder a casi cualquier base de datos existente en el mercado. Este brinda un conjunto de comandos que son traducidos a instrucciones específicas de una base de datos en particular a través de driver provistos por esta.

Es claro que utilizar funciones nativas da más réditos en cuanto a la velocidad de respuesta, comparado con trabajar con algún mediador tipo ODBC. Por otro lado, cuando se emplean base de datos con una gran cantidad de prestaciones como Oracle y se utiliza ODBC, OLE, ADO, etc., se pierde gran parte del poder, puesto que hay funciones propias de la base de datos que no se pueden utilizar con un mediador genérico como estos, por no estar implementadas. Por lo tanto, si se quiere utilizar todo el potencial de la base de datos, es preferible acceder con funciones nativas, como las que ofrece PHP en sus extensiones.

Por ejemplo para el caso del desarrollo del sistema, se utilizara PostgreSQL y php ya provee de una extensión (php\_pgsqI.dll) para que podamos hacerlo de modo nativo.

## MySQL

MySQL es un gestor de bases de datos, y actualmente uno de los más usados y reconocidos del mercado. Especialmente en lo que se refiere a desarrollo web, está clasificada como la base de datos de código abierto más popular del mundo.

Fue desarrollada inicialmente por MySQL AB, luego adquirida por Sun Microsystems y luego posteriormente por Oracle Coporation en 2010.

Está desarrollado mayormente por una mezcla de los lenguajes C y C++. Es uno de los cuatro componentes del paquete de desarrollo LAMP, para Linux (corre perfecto en cualquier distro Linux, y se hizo muy popular al hacerlo en RedHat Enterprise), y WAMP, para Windows Server.

MySQL es utilizado por sitios webs muy populares y de gran tamaño. Entre estos sitios destacados, podemos nombrar algunos ejemplos como: Youtube, Wikipedia, Facebook, Google, Flickr y Twitter.

Es mayormente utilizando en conjunción con servidores web donde se lo encuentra relacionado a aplicaciones web o CMS para sitios online, como Wordpress, Joomla!, Drupal, entre otros. Está muy ligado a PHP en lo que se refiere a este tipo de desarrollos.

Es una base de datos que presenta rapidez en lectura, sobre todo cuando se utilizan ciertos motores como MylSAM o InnoDB. A pesar de ello, se puede presentar problemas el utilizarla en entornos o desarrollos que presenten una alta concurrencia de modificación.

Actualmente, las versiones de MySQL que se siguen desarrollando y recibiendo soporte son aquellas posteriores a la 5.5.0 M2.

### **2.2.1.4. Características de MySQL**

MySQL es un sistema de base de datos relacional, lo que quiere decir que archiva datos en tablas separadas en lugar de guardar todos los datos en un gran archivo, lo que le permite tener

mayor velocidad y flexibilidad. Estas tablas están relacionadas de formas definidas, por lo que se hace posible combinar distintos datos en varias tablas y conectarlos.

Cualquier persona puede utilizar MySQL y crear modificaciones bajando el código fuente con el fin de ajustarlo a su medida y para su uso, ya que es un software de código abierto. Se establece el GPL (General Public License) en donde se define lo que un usuario puede realizar o no con el software. Si un usuario no se ajusta a las medidas del GPL o requiere el uso del software para aplicaciones comerciales, se puede comprar una versión comercial licenciada.

**Algunas características de MySQL son:**

- Permite escoger múltiples motores de almacenamiento para cada tabla.
- Agrupación de transacciones, pudiendo reunir las de forma múltiple desde varias conexiones con el fin de incrementar el número de transacciones por segundo.
- Conectividad segura.
- Ejecución de transacciones y uso de claves foráneas.
- Presenta un amplio subconjunto del lenguaje SQL.
- Replicación
- Disponible en casi todas las plataformas o sistemas.
- Búsqueda e indexación de campos de texto.
- Utiliza varias herramientas para portabilidad.
- Tablas hash en memorias temporales
- Uso de tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- Ofrece un sistema de contraseñas y privilegios seguros de verificación basada en el host y tráfico de contraseñas encriptado al conectarse a un servidor.
- Uso de multihilos mediante hilos de kernel.
- Soporta gran cantidad de datos, incluso con más de 50 millones de registros.

- En las últimas versiones, se permiten hasta 64 índices por tablas. Cada índice puede consistir desde 1 a 16 columnas o partes de columnas. El máximo ancho de límite son de 1000 bytes.

### **2.2.1.5. Ventajas y desventajas de MySQL**

A continuación enumeramos algunas de las ventajas que presenta MySQL (sobre todo comparables frente a su eterno rival, PostgreSQL), especialmente aquellas que la han convertido en una de las BDs más utilizadas para desarrollos web.

#### **Ventajas:**

- MySQL es de uso libre y gratuito.
- Software con Licencia GPL.
- Bajo costo en requerimientos para la elaboración y ejecución del programa.
- No se necesita disponer de Hardware o Software de alto rendimiento para la ejecución del programa.
- Velocidad al realizar las operaciones y buen rendimiento.
- Facilidad de instalación y configuración.
- Soporte en casi el 100% de los sistemas operativos actuales.
- Baja probabilidad de corrupción de datos.
- Entorno con seguridad y encriptación.

#### **Desventajas:**

- Al ser de Software Libre, muchas de las soluciones para las deficiencias del software no están documentados ni presentan documentación oficial.
- Muchas de sus utilidades tampoco presentan documentación.
- Se debe controlar/monitorizar el rendimiento de las aplicaciones en busca de fallos.

### 2.3. Inventarios.

El inventario es el documento en contabilidad que consiste en una relación detallada, ordenada y valorada de todos los bienes, derechos y deudas de una empresa. Sirve para comprobar cuáles son los elementos que componen el patrimonio de una empresa en un momento determinado. Hay que tener en cuenta que dependiendo de las necesidades del negocio deberá realizarse diaria, semanal o mensualmente. Antiguamente, los inventarios se realizaban por medios físicos (en un papel) pero ahora, lo más común, es que se mantengan de manera centralizada en bases de datos.

Gracias al inventario se pueden corregir posibles errores anotados en los registros de entradas y salidas. Las tres funciones principales del inventario son:

- En la base de las cuentas anuales.
- Sirve para detectar posibles pérdidas por robo, mermas, etc.
- Con él se pueden corregir posibles errores registrados en la contabilidad de existencias.

Para llevar a cabo correctamente un inventario es importante tener en cuenta que la información contable, al tener una finalidad explicativa, debe cumplir con una serie de características:

- **Integridad:** hay que incluir todas las existencias con su cantidad correcta.
- **Precisión:** la naturaleza, la cantidad y el valor de los activos deben ser enumerados correctamente.
- **Eficiencia:** en aquellos casos en los que no pueda determinarse la cantidad exacta de un activo, esta puede ser estimada.
- **Verificabilidad:** esta característica permite que la información pueda ser revisada con posterioridad a su emisión.
- **Puntualidad:** los empresarios tienen que presentar el libro de inventario en un momento determinado, normalmente durante el plazo previsto para ello.

- **Partida individual:** básicamente, cada activo y cada deuda se registran individualmente en el inventario.

### **Tipos de inventarios.**

Es importante tener en cuenta que las empresas disponen de activos tangibles. Los activos tangibles son, como su propio nombre indica, aquellos que se pueden tocar porque tienen una forma física. Los intangibles, en cambio son aquellos que no son materiales pero que construyen recursos muy valiosos para las empresas. Entre los diferentes tipos de activos intangibles podemos encontrar, por ejemplo, las patentes, las marcas, los derechos de autor, etc.

Dependiendo de si los activos son tangibles (físicos) o intangibles, el tipo de inventario que debemos, realizar será diferente:

#### **El inventario Físico.**

Es el inventario propiamente dicho. Consiste en verificar las existencias de materiales, equipo, muebles e inmuebles con los que cuenta la compañía. Como mínimo, debe llevarse a cabo una vez en cada ejercicio, para comprobar si las cantidades registradas en la base de datos coinciden con la cantidad física real que hay en los almacenes.

Una vez se sepa la cantidad física, esta debe registrarse en la contabilidad. En definitiva, se trata hacer un recuento: hay que contar y anotar todos y cada uno de los bienes o mercancías de la empresa. En el caso que proceda, también habrá que pesar y medir.

Es sumamente importante que los empleados, a pesar de la tediosa carga de trabajo que se supone, procedan con cuidado. No se debe olvidar que el inventario es un mecanismo de control: de lo contrario, estarás comparando una lista supuestamente errónea con otra inexacta y obtendrás dudosos resultados.

### **El inventario de Bienes intangibles.**

Existen bienes que no pueden contarse porque no son físicos; son conocidos como bienes intangibles. Como los activos intangibles, los activos corrientes y las deudas no se pueden contar físicamente, deben registrarse en un inventario contable. También se incluyen los saldos de las cuentas bancarias, así como las cuentas por cobrar y los pasivos.

Asimismo, forma parte de él el llamado “capital intelectual de la empresa”, que son activos intangibles de una empresa basados en el conocimiento y que poseen y pueden producir valor (patentes, software y marcas).

### **Sistemas de control de inventario.**

Existen dos sistemas de control de inventarios:

- **Periódico**, que se lleva a cabo administrativamente y
- **Permanente**, que se realiza periódicamente mediante el recuento físico.

#### **Inventario periódico.**

Tal y como su nombre indica, por medio de este sistema se recuentan, físicamente y cada cierto tiempo, las existencias propiedad de la empresa con independencia de donde se encuentren localizadas. La ventaja de este sistema es que su coste es bajo; la desventaja, que las existencias solo pueden conocerse por medio de un recuento físico y ello plantea inconvenientes como la necesidad de cerrar los almacenes mientras se lleva a cabo el recuento. Además, esto da lugar a una carencia de información del valor de este activo durante los periodos en que no se realice el inventario y, consiguientemente el desconocimiento del resultado.

#### **Inventario permanente.**

Bajo un sistema de inventario permanente, las cuentas de inventarios y las bases de datos se actualizan automáticamente cada vez que se recibe o vende un producto. El desarrollo del

inventario permanente ha sido posible gracias a las aplicaciones informáticas, pues requiere unos medios superiores al inventario físico. Estos sistemas hacen un seguimiento al instante de cada movimiento del inventario y envían actualizaciones electrónicas a las bases de datos centrales desde cualquier lugar. Es decir, permite el conocimiento en todo momento del valor de las existencias.

### **Concluir el inventario satisfactoriamente.**

Nunca está de más tu contable este presente durante la realización del inventario, puesto que así podrá controlar el proceso, su ejecución y dar fe de su visita. También es importante que cuides las listas de inventario para que estén ordenadas; de esta forma conseguirás crear una buena impresión a los funcionarios.

### **2.4. Código QR.**

Un código QR (del inglés *Quick Response code*, “código de respuesta rápida”) es la evolución del código de barras. Es un módulo para almacenar información en un matriz de puntos o en un código de barras bidimensional.

La matriz se lee en el dispositivo móvil por un lector específico (lector QR) y de forma inmediata nos lleva a un aplicación en internet y puede ser un mapa de localización, un correo electrónico, una página web o un perfil en una red social. Fue creado en 1994 por la compañía japonesa Denso Wave, subsidiaria de Toyota. Presenta tres cuadrados en las esquinas que permiten detectar la posición del código al lector. El objetivo de los creadores (en equipo de dos personas en Denso Wave, dirigido por Masahiro Hara) fue que el código permitiera que su contenido se leyera a alta velocidad. Los códigos QR son muy comunes en Japón, donde son el código bidimensional más popular ahora se está haciendo en todo el mundo.

Actualmente, el código QR es el más famoso de código de barras 2D en el mundo y su éxito se remonta a la década del 2000 en Japón, donde se convirtieron en un estándar y hasta 2010 comenzaron la expansión en los Estados Unidos y Europa. Principalmente en anuncios. Como

dato adicional, Unitag.io (S.F.) afirma que para 2011 en promedio de 5 Códigos QR eran escaneados diariamente por cada japonés, esto supero por mucho al promedio de SMSM enviados al día.

Un detalle importante sobre el código QR es que, a diferencia de otros formatos de códigos de barras bidimensionales como el BIDI, su código es abierto y sus derechos de patente (propiedad de Denso Wave) no se ejercen.

### **Características Generales.**

La estructura general de un código QR en una matriz bidimensional de módulos de dos colores contrastados en principio, blancos y negros. Hay varias versiones de códigos QR según la cantidad de módulos que forman la matriz: van desde la versión 1 (con una matriz de 21 x 21 módulos) hasta la versión 10 (con 177 x 177 módulos). Las versiones de más módulos admiten mayor cantidad de informaciones en el código. Los códigos más extendidos para el uso del público en general suelen ser los de 25 x 25 y 29 x 29, para captura desde el teléfono móvil o celular en cualquier situación.

### **Tipos de información que puede almacenarse en los códigos QR**

- Información de contacto (vCard). Nombre, compañía, teléfono, dirección postal, etc.
- Una dirección web (URL). Por ejemplo al blog, a la web o incluso a tu perfil en Facebook, twitter, linkld, YouTube, etc.
- Una dirección email.
- Un mensaje tipo SMS.
- Un número de teléfono para realizar una llamada.
- Un texto de hasta 350 caracteres.
- Parámetros de acceso a una red Wi-Fi.
- Datos de un evento para calendario (día y hora de comienzo/fin, nombre del evento, etc.).

- Una geolocalización para verla en un mapa.

### **Principales funciones para el código QR.**

Son múltiples los usos de los códigos QR. Aquí solo te mencionaremos los mas notables:

#### **Almacenamiento y difuncion.**

Como sustituye al tradicional código de barras su uso es muy parecido al anterior, aunque llevado al terreno digital en el que nos movemos. En pocas palabras, sirve para almacenar información y difundirla de manera instantánea, sencilla e, incluso, creativa. Sin embargo, su uso se asocia orientado únicamente al consumidor y no tanto al vendedor. Pero debemos tener claro que además de difundir información es una gran fuente de retroalimentación.

Con frecuencia, los códigos QR nos redirigen a una página web concreta donde encontraremos información de todo tipo, cabe la posibilidad de que descifremos información de tipo audio visual o archivos listos para descargar en PDF. Sea en la forma que sea, su uso tiene mucho que aportarnos. Solo enriquecerá tu información, aunque todo tiene sus pros y sus contras, debemos estar en alerta de códigos QR con contenido malicioso.

#### **Diferencias entre Códigos QR y Códigos de Barra.**

A diferencia de los códigos de barra, los códigos QR almacenan la información en dos dimensiones. Esto permite que tengan una mayor capacidad de almacenamiento.

Los códigos QR Micro pueden almacenar la misma información que los Códigos de Barra pero ocupando solamente un décimo de lugar.

*Figura 2.4 Comparación de Código de Barras y Código QR*



Fuente: (Elaboración propia)

Al ser una simbología desarrollada en Japón, los códigos QR son capaces de encriptar JIS nivel q 1 y nivel 2 de caracteres Kranji. En el caso Japonés, un Kana or Kanjin es eficiente encriptado en 13 bits, permitiendo que el QR almacena menos un 20 % más que un código de dos dimensiones (Código de Barra).

### **2.5. Metodología XP (Extreme Programming).**

“La programación extrema o Extreme Programming (XP)” es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck.

Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

“Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de la vida del software. Esta metodología busca eliminar actividades relacionadas con la elaboración de algunos documentos de especificaciones que no tienen relación directa con el resultado final del software. Se base en los siguientes valores principales:”

- **“Comunicación”**: La necesidad de los desarrolladores de intercambiar ideas e informar sobre el proyecto ya sea con los directores del proyecto o los usuarios de forma confiable, y fácil. La comunicación debe ser continua y rápida.
- **“Sencillez”**: Cuando se tenga que elegir entre varias opciones, en lo posible elegir soluciones simples, sin que esto signifique aplicar enfoques simplistas; la programación

extrema define un diseño simple en el que se realice el menor número posible de clases, métodos y que no tenga código duplicado.

- **“Retroalimentación”**: debe ser rápida en todos los niveles principalmente se consigue ejecutando y probando el código, porque las entregas tempranas y frecuentes son muy importantes.
- **“Valor”**: Todas las personas que participen en el proyecto deben tener la capacidad de expresar su valoración sobre el proyecto. Deberían ser abiertos y dejar que todos revisen e incluso modificasen su trabajo. Los cambios no deberían ser vistos con terror y los desarrolladores deberían tener el valor de encontrar mejores soluciones y modificar el código siempre que sea necesario.
- **“Respeto”**: Debe manifestarse en diversas formas y situaciones, son la base para una buena relación y cooperación entre todos los componentes del equipo de trabajo.

La Programación Extrema se basa en:

- **Desarrollo de iteraciones**: En cada iteración se agregan nuevas funcionalidades, o se corrigen errores generando distintas versiones.
- **Pruebas unitarias continuas**: Estas pruebas están orientadas a comprobar que la aplicación mantenga sus funcionalidades.
- **Programación en pareja**: se hace esto con el fin de que el código se discuta y revise mientras se desarrolla el programa, basado en que los dos programadores puedan complementarse, generando código de mejor calidad con menos errores.
- **Interacción entre los desarrolladores y el usuario**: Se minimizara el esfuerzo de ambas partes, pues se podrá tener una mejor comprensión de los problemas o necesidades de los usuarios y las soluciones que puedan brindar los desarrolladores.
- **Refactorización del código**: Busca hacer el código más fácil y mantenerle, pero debe garantizar su correcto funcionamiento manteniendo las pruebas unitarias.



En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirá los plazos.

### **Fases de Desarrollo de la Programación Extrema.**

**“1ª Fase: Planificación del proyecto:** El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

**2ª Fase: Diseño:** La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e impleméntable que a la larga costará menos tiempo y esfuerzo desarrollar.

Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.

**3ª Fase: Codificación:** Como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código.

**4ª Fase: Pruebas:** Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

El uso de los test en X.P es el siguiente:

- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

- Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa.
- Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido."

## **2.6. Lenguaje De Modelado Unificado (UML).**

### **Concepto sobre UML.**

En todas las disciplinas de la Ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de "algo". Ese "algo" puede existir, estar en un estado de desarrollo o estar, todavía, en un estado de planeación. Es en este momento cuando los diseñadores del modelo deben investigar los requerimientos del producto terminado y dichos requerimientos pueden incluir áreas tales como funcionalidad, performance y confiabilidad. Además, a menudo, el modelo es dividido en un número de vistas, cada una de las cuales describe un aspecto específico del producto o sistema en construcción.

El modelado sirve no solamente para los grandes sistemas, aun en aplicaciones de pequeño tamaño se obtienen beneficios de modelado, sin embargo es un hecho que entre más grande y más complejo es el sistema, más importante es el papel de que juega el modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad".

UML es una técnica para la especificación sistemas en todas sus fases. Nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores y, precisamente, los padres

de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en enero de 1997 y ha sido utilizado con éxito en sistemas construidos para toda clase de industrias alrededor del mundo: hospitales, bancos, comunicaciones, aeronáutica, finanzas, etc.

Los principales beneficios de UML son:

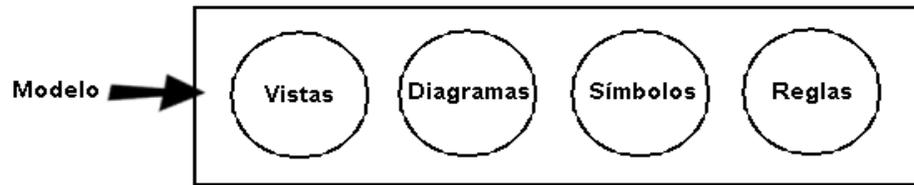
- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

## **UML**

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un lenguaje de modelado. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelo  $\frac{3}{4}$  los símbolos utilizados en los modelos  $\frac{3}{4}$  y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas.

Figura 2.7 Reglas de UML  
UML



Fuente: (Programming s.f.)

**Vistas:** Las vistas muestran diferentes aspectos del sistema modelado. Una vista no es una gráfica, pero sí una abstracción que consiste en un número de diagramas y todos esos diagramas juntos muestran una "fotografía" completa del sistema.

Las vistas también ligan el lenguaje de modelado a los métodos o procesos elegidos para el desarrollo. Las diferentes vistas que UML tiene son:

- Vista Use-Case: Una vista que muestra la funcionalidad del sistema como la perciben los actores externos.
- Vista Lógica: Muestra cómo se diseña la funcionalidad dentro del sistema, en términos de la estructura estática y la conducta dinámica del sistema.
- Vista de Componentes: Muestra la organización de los componentes de código.
- Vista Concurrente: Muestra la concurrencia en el sistema, direccionando los problemas con la comunicación y sincronización que están presentes en un sistema concurrente.
- Vista de Distribución: muestra la distribución del sistema en la arquitectura física con computadoras y dispositivos llamados nodos.

**Diagramas:** Los diagramas son las gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que son utilizados en combinación para proveer todas las vistas de un sistema: diagramas de caso de uso, de clases, de objetos, de estados, de secuencia, de colaboración, de actividad, de componentes y de distribución.

**Símbolos o Elementos de modelo:** Los conceptos utilizados en los diagramas son los elementos de modelo que representan conceptos comunes orientados a objetos, tales como clases, objetos

y mensajes, y las relaciones entre estos conceptos incluyendo la asociación, dependencia y generalización. Un elemento de modelo es utilizado en varios diagramas diferentes, pero siempre tiene el mismo significado y simbología.

**Reglas o Mecanismos generales:** Proveen comentarios extras, información o semántica acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

### **Diagramas del UML.**

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. (Blogspot RUP, s.f.) A continuación se describirán los diagramas más comunes del UML y los conceptos que representan:

#### **2.6.1.1. Diagrama de clases.**

Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. Un ejemplo puede ser la clase “Aviones” que tiene atributos como el “modelo de avión”, “la cantidad de motores”, “la velocidad de crucero” y “la capacidad de carga útil”. Entre las acciones de las cosas de esta clase se encuentran: “acelerar”, “elevarse”, “girar”, “descender”, “desacelerar”. Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre sí.

**Clase Abstracta:** Las clases se representan con rectángulos divididos en tres áreas: la superior contiene el nombre de la clase, la central contiene los atributos y la inferior las acciones.

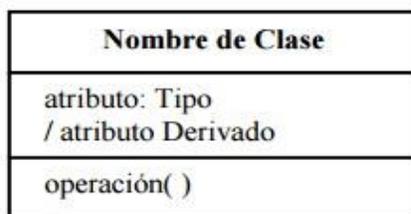


Figura 2.8 Clase abstracta (Programming s.f.)

**Asociaciones:** Las asociaciones son las que representan a las relaciones estáticas entre las clases. El nombre de la asociación va por sobre o por debajo de la línea que la representa. Una flecha rellena indica la dirección de la relación. Los roles se ubican cerca del final de una asociación. Los roles representan la manera en que dos clases se ven entre ellas. No es común el colocar ambos nombres, el de la asociación y el de los roles a la vez. Cuando una asociación es calificada, el símbolo correspondiente se coloca al final de la asociación, contra la clase que hace de calificador.

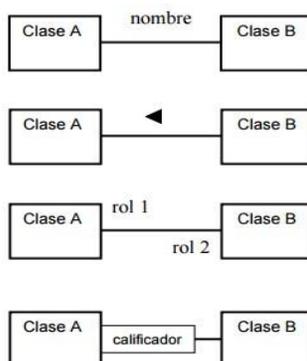


Figura 2.9 Asociaciones (Programming s.f.)

**Multiplicidad:** Las notaciones utilizadas para señalar la multiplicidad se colocan cerca del final de una asociación. Estos símbolos indican el número de instancias de una clase vinculadas a

una de las instancias de la otra clase. Por ejemplo, una empresa puede tener uno o más empleados, pero cada empleado trabaja para una sola empresa solamente.

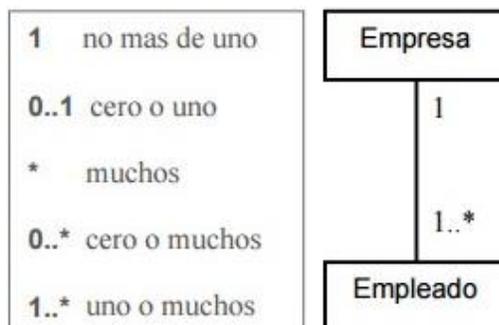


Figura 2.10 Multiplicidad (Programming s.f.)

**Composición y Agregación:** Composición es un tipo especial de agregación que denota una fuerte posesión de la Clase “Todo”, a la Clase “Parte”. Se grafica con un rombo diamante relleno contra la clase que representa el todo. La agregación es una relación en la que la Clase “Todo” juega un rol más importante que la Clase "Parte", pero las dos clases no son dependientes una de otra. Se grafica con un rombo diamante vacío contra la Clase “Todo”.

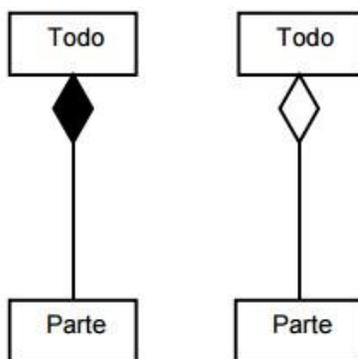


Figura 2.11 Composición y agregación (Programming s.f.)

**Generalización:** Generalización es otro nombre para herencia. Se refiere a una relación entre dos clases en donde una Clase “Específica” es una versión especializada de la otra, o Clase “General”. Por ejemplo, Honda es un tipo de auto, por lo que la Clase “Honda” va a tener una relación de generalización con la Clase “Auto”.

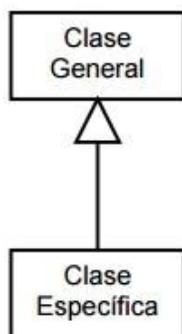


Figura 2.12 Generalización (Programming s.f.)

### 2.6.1.2. Diagrama de objetos.

Los Diagramas de Objetos están vinculados con los Diagramas de Clases. Un objeto es una instancia de una clase, por lo que un diagrama de objetos puede ser visto como una instancia de un diagrama de clases. Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases.

#### Nombre de los objetos

Cada objeto es representado como un rectángulo, que contiene el nombre del objeto y su clase subrayadas y separadas por dos puntos.

**Atributos:** Como con las clases, los atributos se listan en un área inferior. Sin embargo, los atributos de los objetos deben tener un valor asignado.

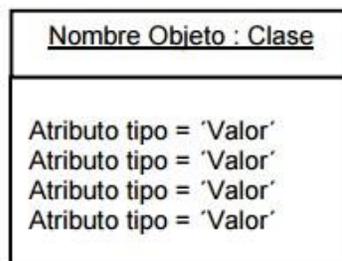


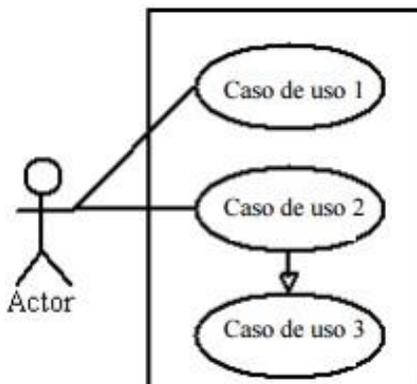
Figura 2.13 Diagrama de objeto (Programming s.f.)

### 2.6.1.3. Diagrama de casos de uso.

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

- **Sistema:** El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del sistema.
- **Casos de Uso:** Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.
- **Actores:** Los actores son los usuarios de un sistema.
- **Relaciones:** Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas "incluir" o "extender." Una relación "incluir" indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación "extender" indica opciones alternativas para un cierto caso de uso.

Figura 2.14 Diagrama de casos de uso



Fuente: (Elaboracion propia)

#### 2.6.1.4. Diagrama de estados.

En cualquier momento, un objeto se encuentra en un estado particular, la luz está encendida o apagada, el auto en movimiento o detenido, la persona leyendo o cantando, etc. El diagrama de estados UML captura esa pequeña realidad.

- **Estado:** El estado representa situaciones durante la vida de un objeto. Se representa con un rectángulo que tiene sus esquinas redondeadas.
- **Transición:** Una flecha representa el pasaje entre diferentes estados de un objeto. Se etiqueta con el evento que lo provoca y con la acción resultante.

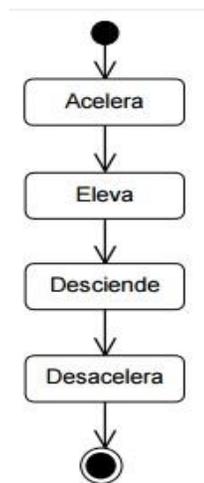


Figura 2.15 diagrama de Estados o Actividades (Programming s.f.)

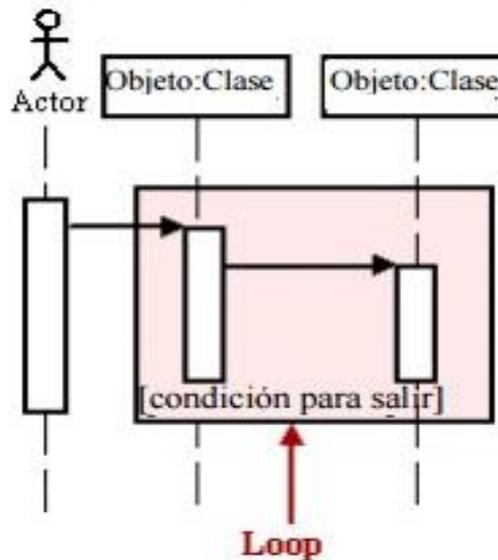
#### 2.6.1.5. Diagrama de secuencias.

Los diagramas de clases y los de objetos representan información estática. No obstante, en un sistema funcional, los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.

- **Rol de la Clase:** El rol de la clase describe la manera en que un objeto se va a comportar en el contexto. No se listan los atributos del objeto.

- **Activación:** Los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.
- **Mensajes:** Los mensajes son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos. Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.
- **Líneas de Vida:** Las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.
- **Destrucción de Objetos:** Los objetos pueden ser eliminados tempranamente usando una flecha etiquetada "<>" que apunta a una X.
- **Loops:** Una repetición o loop en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el loop se coloca en la parte inferior entre corchetes.

Figura 2.16 Diagrama de Secuencias



Fuente: (Programming s.f.)

## 2.7. Modelo de calidad MCCALL.

(Monografias.com, s.f.) El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto, basándose en once

factores de calidad organizados en torno a los tres ejes y a su vez cada factor se desglosa en otros criterios:

*Tabla 2.1 Criterios de McCall*

<b>Puntos de Vista o ejes</b>	<b>Factor</b>	<b>Criterios</b>
OPERACIÓN DEL PRODUCTO	Facilidad de uso	<ul style="list-style-type: none"> <li>• Facilidad de operación: Atributos del software que determinan la facilidad de operación del software.</li> <li>• Facilidad de comunicación: Atributos del software que proporcionan entradas y salidas fácilmente asimilables.</li> <li>• Facilidad de aprendizaje: Atributos del software que facilitan la familiarización inicial del usuario con el software y la transición del modo actual de operación.</li> <li>• Formación: El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.</li> </ul>
	Integridad	<ul style="list-style-type: none"> <li>• Control de accesos. Atributos del software que proporcionan control de acceso al software y los datos que maneja.</li> <li>• Facilidad de auditoría: Atributos del software que facilitan la auditoría de los accesos al software.</li> </ul> <p>Seguridad: La disponibilidad de mecanismos que controlen o protejan los programas o los datos.</p>

	Corrección	<ul style="list-style-type: none"> <li>• Completitud: Atributos del software que proporcionan la implementación completa de todas las funciones requeridas.</li> <li>• Consistencia: Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación.</li> <li>• Trazabilidad o rastreabilidad: Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto.</li> </ul>
OPERACIÓN DEL PRODUCTO	Fiabilidad	<ul style="list-style-type: none"> <li>• Precisión: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.</li> <li>• Consistencia.</li> <li>• Tolerancia a fallos: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales.</li> <li>• Modularidad: Atributos del software que proporcionan una estructura de módulos altamente independientes.</li> <li>• Simplicidad: Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible.</li> <li>• Exactitud: La precisión de los cálculos y del control.</li> </ul>

	Eficiencia	<ul style="list-style-type: none"> <li>• Eficiencia en ejecución: Atributos del software que minimizan el tiempo de procesamiento.</li> <li>• Eficiencia en almacenamiento: Atributos del software que minimizan el espacio de almacenamiento necesario.</li> </ul>
REVISION DEL PRODUCTO	Facilidad de mantenimiento	<ul style="list-style-type: none"> <li>• Modularidad.</li> <li>• Simplicidad.</li> <li>• Consistencia.</li> <li>• Concisión: Atributos del software que posibilitan la implementación de una función con la menor cantidad de códigos posible.</li> <li>• Auto descripción: Atributos del software que proporcionan explicaciones sobre la implementación de las funciones.</li> </ul>
	Facilidad de prueba	<ul style="list-style-type: none"> <li>• Modularidad.</li> <li>• Simplicidad.</li> <li>• Auto descripción.</li> <li>• Instrumentación: Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores.</li> </ul>
	Flexibilidad	<ul style="list-style-type: none"> <li>• Auto descripción.</li> <li>• Capacidad de expansión: Atributos del software que posibilitan la expansión del software en cuanto a capacidades funcionales y datos.</li> <li>• Generalidad: Atributos del software que</li> </ul>

		<p>proporcionan amplitud a las funciones implementadas.</p> <ul style="list-style-type: none"> <li>• Modularidad.</li> </ul>
	Reusabilidad	<ul style="list-style-type: none"> <li>• Auto descripción.</li> <li>• Generalidad.</li> <li>• Modularidad.</li> <li>• Independencia entre sistema y software: Atributos del software que determinan su dependencia del entorno operativo.</li> <li>• Independencia del hardware: Atributos del software que determinan su dependencia del hardware.</li> </ul>
	Intoperabilidad	<ul style="list-style-type: none"> <li>• Modularidad.</li> <li>• Compatibilidad de comunicaciones: Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar.</li> <li>• Compatibilidad de datos: Atributos del software que posibilitan el uso representaciones de datos estándar.</li> <li>• Estandarización en los datos: El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa.</li> </ul>
	Portabilidad	<ul style="list-style-type: none"> <li>• Auto descripción.</li> <li>• Modularidad.</li> <li>• Independencia entre sistema y software.</li> <li>• Independencia del hardware.</li> </ul>

Fuente: (Monografias.com s.f.)

### **Cómo emplear el modelo de MCCALL.**

Antes de comenzar a utilizar el modelo de McCall hay que seguir las siguientes pautas:

- Se aceptan los factores, criterios y métricas que propone el modelo.
- Se aceptan las relaciones entre factores y criterios, y entre criterios y métricas.
- Se selecciona un subconjunto de factores de calidad sobre los que aplicar los requisitos de calidad establecidos para el proyecto.

Al comienzo del proyecto habrá que especificar los requisitos de calidad del producto software, para lo cual se seleccionarán los aspectos inherentes a la calidad deseada del producto, teniendo que considerarse para ello:

- Las características particulares del propio producto que se está diseñando: por ejemplo, su ciclo de vida que si se espera que sea largo implicará un mayor énfasis en la facilidad de mantenimiento y la flexibilidad, o bien si el sistema en desarrollo está destinado a un entorno donde el hardware evoluciona rápidamente implicará como requisito su portabilidad.
- La relación calidad-precio, que puede evaluarse a través del coste de cada factor de calidad frente al beneficio que proporciona. La siguiente tabla muestra la relación calidad-precio para cada factor considerado:

*Tabla 2.2 Calidad-Precio de los factores de McCall*

Factor	Beneficio / coste
Corrección	alto
Fiabilidad	alto
Eficiencia	bajo
Integridad	bajo
Facilidad de uso	medio

Facilidad de mantenimiento	de alto
Facilidad de prueba	alto
Flexibilidad	medio
Portabilidad	medio
Reusabilidad	medio
Interoperabilidad	bajo

Fuente: (Elaboración Propia)

- La determinación de las etapas del ciclo de vida donde es necesario evaluar cada factor de calidad para conocer en cuales se dejan sentir más los efectos de una calidad pobre con respecto a cada uno de los factores.
- Las propias interrelaciones entre los factores debido a que algunos factores pueden entrar en conflicto entre sí: por ejemplo, la eficiencia plantea conflictos prácticamente con todos los demás factores de calidad. La interacción entre los diversos factores a evaluar queda reflejada en la tabla I que indica la dependencia entre los factores de McCall.

También habrá que establecer valores deseables para los criterios, para lo cual se emplearán datos históricos, el promedio en la industria y con ellos se concretarán los valores finales y otros intermedios o predictivos en cada período de medición durante el desarrollo, así como unos valores mínimos aceptables. La explicación para cualquier selección o decisión deberá ser adecuadamente documentada.

En la fase de desarrollo será necesario implementar las métricas elegidas, analizar sus resultados y tomar medidas correctivas cuando los valores obtenidos estén por debajo de los mínimos aceptables.

Una vez finalizado el proyecto será necesario contrastar las medidas predictivas utilizadas y comprobar si, en efecto, se pueden tomar como indicadores de los valores finales.

## **2.8. Modelo Constructivo De Costos (COCOMO II).**

COCOMO II, es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo de software. Está asociado a los ciclos de vida modernos, apunta hacia los proyectos de software de los 90 y de la primera década del 2000.

### **Utilización del modelo de composición de aplicaciones.**

Este modelo se dirige a aplicaciones que están demasiado diversificadas para crearse rápidamente en una herramienta de dominio específico, (como una hoja de cálculo) y que todavía no se conocen suficientemente como para ser compuestas a partir de componentes ínter operable. Ejemplos de estos sistemas basados en componentes son los creadores de interfaces gráficas para usuario, base de datos o gestores de objetos, middleware para proceso distribuido o transaccional, manejadores hipermedia, buscadores de datos pequeños y componentes de dominio específico tales como paquetes de control de procesos financieros, médicos o industriales.

Dado que el Modelo De Composición De Aplicaciones incluye esfuerzos de prototipo para resolver asuntos potenciales de alto riesgo tales como interfaces de usuario, interacción software/sistema, ejecución o grado de madurez tecnológica, los costes de este tipo de esfuerzo se estiman mejor mediante dicho modelo.

### **Modelo de coste de composición de aplicaciones.**

#### **2.8.1.1. Introducción a los Puntos Objeto.**

Los puntos objeto son el recuento de pantallas, informes, módulos de lenguajes de 3ra generación desarrollados en la aplicación, cada uno ponderado mediante un factor de complejidad de tres niveles (simple, medio y complejo).

La estimación de puntos objeto es un enfoque relativamente nuevo de medida de software, pero encaja bien en las prácticas del sector de la Composición de Aplicaciones. También encaja muy bien en los esfuerzos de prototipado asociados, basado en el uso de herramientas ICASE que proporcionan constructores de interfaces gráficos de usuario, herramientas de desarrollo de software; y en general, infraestructura que puede componerse y componentes de aplicación. En estas áreas se ha comparado bien con la estimación de Puntos de Función en un conjunto de aplicaciones no triviales (pero todavía limitadas).

#### **2.8.1.2. Procedimiento de Obtención de Puntos Objeto.**

La definición de los términos utilizados en los Puntos Objetos es la siguiente:

- **OP:** Puntos Objeto.
- **NOP:** Nuevos Puntos Objeto (Cantidad de Puntos Objeto ajustado por la reutilización).
- **Srvr:** Número de tablas de datos del servidor (mainframe ó equivalente) usadas junto a la pantalla o informe.
- **Clnt:** número de tablas de datos del cliente (estación de trabajo personal) usadas junto con la pantalla o el informe.
- **%reuse:** Porcentaje de pantallas, informes y módulos 3GL reutilizados a partir de aplicaciones anteriores prorrateadas por grado de reutilización.

Hemos de destacar que el uso del término “Objeto” en los puntos objeto define Pantallas, Informes y módulos 3GL como objetos.

#### **Paso 1. Hacer el Recuento de Objetos.**

Estimar el número de pantallas, informes y componentes de las que consta esta aplicación, suponer las definiciones estándar de los objetos en el entorno ICASE correspondiente.

## Paso 2. Clasificar Cada Instancia de Objeto.

Clasificar cada instancia de objeto dentro de niveles de complejidad simple, media y difícil dependiendo de los valores de las dimensiones de las características.

Tabla 2.3 Complejidad asociada a las instancias de objetos

Para Pantallas				Para Informes			
Nº de vistas que contiene	# y fuente de tablas de datos			Nº de secciones que contiene	# y fuente de tablas de datos		
	Total < 4 (<2 svr <3cInt)	Total < 8 (2/3 svr 3-5 cInt)	Total 8+ (>3 svr >5cInt)		Total < 4 (<2 svr <3cInt)	Total < 8 (2/3 svr 3-5 cInt)	Total 8+ (>3 svr >5cInt)
< 3	Simple	Simple	Medio	0 ó 1	Simple	Simple	Medio
3 – 7	Simple	Medio	Difícil	2 ó 3	Simple	Medio	Difícil
> 8	Medio	Difícil	Difícil	4 ó +	Medio	Difícil	Difícil

Fuente: (Campuchino, 2000)

## Paso 3. Pesar el Número de Cada Celda.

El peso refleja el esfuerzo relativo que se requiere para implementar una instancia de ese nivel de complejidad, para determinar el peso se usa la tabla siguiente:

Tabla 2.4 Pesos asociados a los niveles de complejidad

Tipo de Objeto	Complejidad – Peso		
	Simple	Medio	Difícil
Pantalla	1	2	3
Informe	2	5	8
Componente 3GL			10

Fuente: (Campuchino, 2000)

#### **Paso 4. Determinar Puntos Objeto.**

Para determinar los Puntos Objeto se debe sumar todas las instancias pesadas para conseguir un número. El recuento de puntos objeto.

#### **Paso 5. Porcentaje de Reutilización.**

Estimar el porcentaje de reutilización que se espera lograr en este proyecto. Calculando los nuevos Puntos Objeto a desarrollar.

$$NOP = \frac{(OP) \times (100 - \%Reuse)}{100}$$

#### **Paso 6. Determinar Ratio de Productividad.**

$$PROD = \frac{NOP}{Meses - persona}$$

*Tabla 2.5 Ratio de Productividad PROD*

<b>Experiencia y capacidad de los desarrolladores</b>	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
<b>ICASE madurez y capacidad</b>	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
<b>PROD</b>	4	7	13	25	50

Fuente: (Campuchino, 2000)

#### **Paso 7. Cálculo del valor Meses-Persona.**

Para realizar el cálculo del valor Meses-Persona (esfuerzo) se utiliza la ecuación:

$$MM = \frac{NOP}{PROD}$$

### 2.8.1.3. Ahorro y gasto software de escala (B).

$$B = 0.91 + 0.01 \times \sum_{j=1}^5 SF_j$$

Los modelos de estimación de coste del software a menudo tienen un factor exponencial para considerar los gastos y ahorros relativos de escala encontrados en proyectos software de distinto tamaño. El exponente B se usa para capturar estos efectos. El valor de B es calculado mediante la ecuación anterior.

El exponente B se obtiene mediante los denominados drivers de escala. La selección de drivers de escala se basa en la razón de que ellos son un recurso significativo de variación exponencial en un esfuerzo o variación de la productividad del proyecto. Cada driver de escala tiene un rango de niveles de valores desde Muy Bajo hasta Extra Alto, cada nivel de valores tiene un peso, SF, y el valor específico del peso se llama factor de escala. Un factor de escala de un proyecto, SF<sub>J</sub> se calcula sumando todos los factores y se usa para determinar el exponente de escala, B.

Tabla 2.6 Factores de escala para el modelo de COCOMO II

Factores de Escala (SF <sub>J</sub> )	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	Completamente sin precedentes	Prácticamente sin precedentes	Casi sin precedentes	Algo familiar	Muy familiar	Completamente familiar
FLEX	Riguroso	Relajación ocasional	Algo de relajación	Conformidad general	Algo de conformidad	Metas generales
RESL	Poco (20 %)	Algo (40 %)	A menudo (60 %)	Generalmente (75 %)	En su mayor parte (90 %)	Por completo (100 %)
TEAM	Interacciones muy difíciles	Algo de dificultad en las interacciones	Interacciones básicamente cooperativas	Bastante cooperativo	Altamente cooperativo	Completas interacciones
PMAT	Peso medio de respuestas "SI" para el cuestionario de madurez CMM					

Tabla 2.7 Valores de los Factores de escala para el modelo de COCOMO II.

Factores de Escala ( $W_i$ )	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Fuente: (Capuchino 2000)

#### 2.8.1.4. Valores de tiempo de desarrollo.

La versión inicial de COCOMO II proporciona una capacidad de estimación de tiempo simplemente similar a las de COCOMO. La ecuación siguiente es la ecuación inicial de tiempos base para las tres etapas de COCOMO II es:

$$TDEV = \left[ 3.67 \times MM^{0.28+0.2 \times (B-1.01)} \right] \times \frac{SCED\%}{100}$$

Dónde:

TDEV es el tiempo en meses desde la determinación de una línea base de requisitos del producto hasta que se completa una actividad de aceptación que certifica que el producto satisface los requisitos. MM es la estimación de meses-persona, excluyendo el estimador de esfuerzo SCED. B es la suma de los factores de escala del proyecto SCED % es el porcentaje de compresión/expansión en el multiplicador de esfuerzo SCED.

#### 2.9. Pruebas (Diseño de casos de Prueba)

Para la realización de estas es necesario la creación de casos de prueba especificando la forma de probar el sistema como un todo, (Sommerville, 2005). Esto incluye:

- Realizar pruebas de instalación en la plataforma de producción definida.

- Prueba de configuración
- Pruebas negativa , encontrar debilidades del sistema
- Pruebas de tensión o de estrés al no existir recursos suficientes
- Prueba de integración del sistema

### **2.9.1. Pruebas de Caja Negra**

También denominadas pruebas de comportamiento, se central los requisitos funcionales del sistema. Esta prueba permite al ingeniero obtener conjunto de condiciones de entrada que ejerciten los requisitos funcionales del sistema, (Sommerville, 2005).

Este tipo de pruebas intentan encontrar errores de tipo:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores de estructuras de datos
- Errores de rendimiento
- Errores de inicio y fin

### **2.9.2. Pruebas de caja Blanca**

Denominada también prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control de diseño para obtenerlo, (Somerville, 2005).

Con estas pruebas se pretenden:

- Garantizar que se ejecuta al menos una vez todos los caminos independientes de cada modulo
- Ejerciten todas la decisiones lógicas en sus vertientes verdadero y falsa
- Ejecuten todos los bucles en sus limites

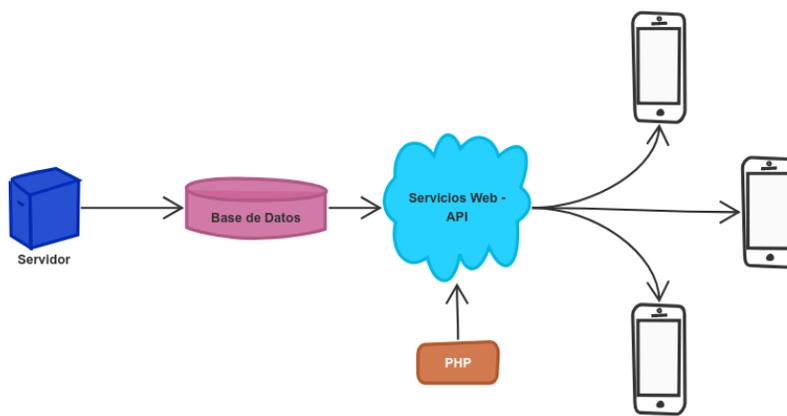
### 3. CAPITULO III

#### MARCO APLICATIVO

#### 3.1. Arquitectura de la Aplicación

El proyecto a desarrollar se divide en dos partes. La primera parte es la aplicación para Android y para su funcionamiento necesitamos de un dispositivo móvil Android, con la aplicación instalada y con acceso a internet por lo que se podrá consultar desde cualquier parte del mundo, esta se conecta al Servicio Web (WebService específicamente JSON) que estará publicada en internet mediante el Servidor de la Institución, que a su vez se comunica de los pedidos e intercambian información.

Figura 2.1 Arquitectura de la aplicación



Fuente: (Elaboración Propia)

#### Web Service (Servicio Web)

Un servicio web (Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares como Internet. La intemporalidad se consigue mediante la adopción de estándares abiertos. Las

organizaciones OASIS17 y W3C18 son los comités responsables de la arquitectura y reglamentación de los servicios Web. Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados. (Wikipedia, Servicio Web, 2015)

#### **3.1.1.1. Ventajas de los Servicios Web**

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web promueven los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y sistemas o programas de diferentes compañías ubicadas en diferentes lugares del mundo puedan ser combinados fácilmente para proveer servicios integrados.

Ahora entre más datos se estén intercambiando entre las aplicaciones, los mensajes crecerán más y el tiempo de entrega se irá incrementando. Esta situación ha hecho que haya surgido un nuevo formato para el intercambio de datos con el fin de hacer más ligeros los mensajes de intercambio. Este formato es JSON.

#### **3.1.1.2. JSON (JavaScript Object Notation)**

“JSON (JavaScript Object Notation – Notación de Objetos de JavaScript) es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML<sup>19</sup>. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML. También es frecuente el uso de JSON y XML en la misma alternativa.” (Json.org, s.f.) . Así es que para su configuración se debe descargar 2 librerías que nos ayudaran a simplificar el código necesario para consumir un Web Service SOAP con formato JSON

- **Gson** es una biblioteca Java que se puede utilizar para convertir objetos Java en su representación JSON. También se puede utilizar para convertir una JSON a una equivalente de objetos Java.
- **Volley** es una biblioteca ligera y eficiente para web services basados en el protocolo SOAP.

La metodología XP consta de 4 fases que se mencionó en el Capítulo II y que ahora se aplicara cada uno de ellas.

### 3.2. Fase de Planificación

#### Descripción de los interesados.

En la siguiente matriz se puede apreciar cuales son los interesados en el proyecto de desarrollo de sistema de difusión de información institucional, para tener una mejor interacción y comunicación.

*Tabla 3.1 Matriz de Stakeholders del Proyecto*

NOMBRE	CARGO	INSTITUCION	RELEVANCIA PROYECTO (1-5)
Hna. Ruth Daniela Calani Ticona	Directora de Fundación	CDI-BO0450 Fundación Ladislao Calani	5
Pastor y Hermanos lideres	Comité de Fundación	CDI-BO0450 Fundación Ladislao Calani	5

Profesional	Responsable de Patrocinio	CDI-BO0450 Fundación Ladislao Calani	5
Consultora	Responsable de Contabilidad	CDI-BO0450 Fundación Ladislao Calani	5
Profesional	Coordinador Programa Pre gap y Gap	CDI-BO0450 Fundación Ladislao Calani	5
Tutores	Personal a cargo de Niños, niñas, adolescentes	CDI-BO0450 Fundación Ladislao Calani	5
Población Infante-Jóvenes	Patrocinados	CDI-BO0450 Fundación Ladislao Calani	3
Padres de Familia	Familias de Beneficiarios	CDI-BO0450 Fundación Ladislao Calani	1

Fuente: (Elaboración propia)

Todos cumplen un papel esencial dentro del sistema a desarrollar, unos más que otros.

### **Requerimientos funcionales del proyecto.**

La aplicación estará compuesta de 7 partes o módulos en las cuales vamos a detallar cada una de las opciones con las que contara:

### **Autenticación de usuario.**

- Pre registró con datos de los usuarios más la contraseña de acceso.
  - Nombre y apellido
  - Contraseña.
  - Tipo de usuario
- Acceso a la app de acuerdo al tipo de usuario.
  - Director y el Responsable de Patrocinio
  - Crean modifican y eliminan
  - Los demás solo registran
  - Cerrar sesión

### **Registro de activos.**

- Llenado de datos específicos del activo.
  - Nombre, precio, tipo, color, estado/calidad.
  - Fecha de creación, y modificación que se pueda realizar
  - Registro de modificaciones
- Almacenamiento de fotografía.
  - Uso de la cámara del Smartphone Android.
  - Zoom de Imagen
  - Un campo para la ubicación de la foto del activo
  - Las fotografías son subidas a un servidor.
- Creación de un código QR.
  - Se generara un código QR por cada activo

### **Creación de Códigos QR.**

- Se genera un código QR por cada activo
  - El Código QR contiene Datos del Activo

- Lector Escaneo del Código QR
  - Búsqueda del activo por medio del escaneo

### **Registro de préstamos de activos.**

- Registro de usuarios beneficiados con el préstamo.
  - Nombre y apellidos
  - Cedula de identidad
  - Dirección celular
  - Ubicación donde se encontrara el activo.
- Registro del Responsable del préstamo
  - Importación de datos de Usuarios.
- Registro de la salida del activo.
  - Escaneo del código QR del activo
  - Asignación de un código de préstamo
  - Registro del Estado/calidad del activo
  - Registro del propósito de préstamo
  - Fecha automática de préstamo
  - Registro de fecha de devolución
  - Registro de fecha límite de devolución.

### **Realización de Inventarios.**

- Escaneo del código QR del activo
  - Búsqueda de la foto de activo
  - Características infalibles del activo
- Asignación del Estado del activo.
  - Modificación del campo estado del Activo
- Registro de fechas del inventario.

**Registro de Activos devueltos.**

- Escaneo del código QR del activo
  - Búsqueda de la foto de activo
  - Características infalibles del activo
- Asignación del Estado del activo.
  - Modificación del campo estado del Activo
- Registro de fechas de la devolución.

**Generalización de Reportes.**

- Reportes finales de los activos
  - PDF detallado de los activos (Auditoria)
  - Filtros de datos
- Reportes de Activos Prestados
  - PDF detallado de los activos
  - Filtros de datos
- Reportes de Prestamistas
  - PDF detallado de los activos
  - Filtros de datos

**Backup's de Bases de datos.**

- Copias de seguridad de la Base de datos
  - Antes de cada modificación
  - Guardado de forma local y en el servidor.

**Requerimientos no funcionales del sistema.**

**Rendimiento de la aplicación.****Seguridad.**

- Requisitos de conexión. (Debe verificar que sea un usuario creado para la aplicación por lo que la consulta directamente en la base de datos para acceder a la aplicación.). Este caso aplica para todos los usuarios.

**Disponibilidad.**

- La aplicación estará disponible el 100% del tiempo, ya que se trata de una aplicación nativa que se instalara en el dispositivo móvil.
- La aplicación dependerá de una conexión a internet o plan de datos para la autenticación de usuario, creación o modificación de activos, registro de préstamos y otros. En caso de no tener conexión a internet la misma seguirá teniendo funcionalidad, en el buscador de activos mediante el escaneo.
- La aplicación solo se instalara en la Tablet de la institución o en Smartphone autorizados por el comité y el pastor, ya que al ser una aplicación de difusión de información institucional va dirigido a solo usuarios interesados en consultar, registrar y/o modificar.

**Mantenibilidad.**

El sistema estará constantemente en mantenimiento ya que se podría agregar nuevas funcionalidades, modificaciones o correcciones.

Contará con un sistema de Backup's de la base de datos antes de que se haga alguna modificación.

**Portabilidad**

Compatibilidad con plataformas: En el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0.3 ya sea en un Tablet o un Smartphone.

## **Operabilidad.**

La aplicación podrá ser operado por todo usuario que tenga instalada la aplicación.

Para poder acceder a las distintas opciones el usuario debe autenticarse o registrarse.

El sistema presenta dos tipos de vistas dependiendo del tipo de usuario que se autentifique.

## **Restricciones**

El dispositivo móvil en donde se ejecutará la aplicación para su correcto funcionamiento deberá tener los siguientes requisitos mínimos, debido a que necesitamos velocidad de procesamiento al manejar una gran cantidad de información, además necesitaremos almacenamiento para la base de datos local donde se almacena la información descargada y para la descarga de distintos archivos multimedia que el usuario descargue.

- Procesador: 1 GHz Dual Core
- Memoria RAM: 512 MB
- Almacenamiento: 30 MB a 45 MB dependiendo del dispositivo.
- Pantalla de 4" pulgadas en adelante.
- Sistema operativo: Android 4.0.3 o superior

Se ha considerado la versión de Android 4.0.3 Ice Cream Sandwich o superior ya que la aplicación será programada con el API 15 como mínima, correspondiente a esta versión.

Para poder realizar una publicación exitosa, enviar notificaciones y que se guarde en la base de datos en el dispositivo móvil se necesitara de una conexión a internet ya sea:

- Wi-Fi
- Plan de Datos móviles (3G24 o superior)

Para la parte del sistema web, los requisitos mínimos serian:

- Computadora Dual Core o superior.
- Memoria 512MB RAM, 100 GB Disco Duro o superior.
- Navegador Google Chrome v.50 o superior.
- Conexión estable a internet.

Se detallara algunas marcas y modelos de Tablets y Smartphones más reconocidas, las cuales cumplen los requisitos mínimos y en las cuales la aplicación desarrollada, no tendría problemas en la ejecución ya sea por problemas de procesamiento o de visualización, además detallaremos algunas características y un precio de mercado aproximado a la fecha (Marzo 2016).

*Tabla 3.2 Descripción de smartphones con los requisitos para la aplicación*

<b>Dispositivo</b>	<b>Procesador</b>	<b>Memoria</b>	<b>Versión de Android</b>	<b>Precio estimado</b>
<b>Samsung Galaxy J1</b>	Dual Core 1.2 GHz	510 MB	4.3	710 Bs
<b>Huawei P8 lite</b>	Kirin 625 1.3GHz	2 GB	6.1	1200 Bs
<b>Xiaomi Mi MIX 2</b>	Qualcomm Snapdragon 835	6 GB	7.1	2749,11 Bs
<b>Samsung Galaxy J5</b>	Xynos 7870 1.60 GHz	4 GB	7	1508,91 Bs

Fuente: (Elaboracion propia – Mercado Uyustus)

Cabe recalcar que todas las restricciones descritas y analizadas anteriormente es solo una sugerencia para el óptimo funcionamiento de la aplicación es decir que se rápida y tenga buena visualización, y no quita o restringe que la aplicación pueda funcionar en dispositivos de menor gama o incluso en Smartphone que tiene más pequeña que una Tablet.

### **3.3. Fase de Diseño.**

Se elaborarán diseños breves que sirven de referencias para la implementación. Otra práctica fundamental de la metodología de programación Extrema (XP) es utilizar diseños tan simples como sea posible. El principio es utilizar el diseño más sencillo que consiga que todo funcione evitando diseñar características extra y que tomaran demasiado tiempo.

Para esta fase vamos a emplear el modelado UML ya que se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. UML ofrece varios diagramas en los cuales modelar sistemas, los desarrolladores lo eligen puesto que es un lenguaje que ayuda a distinguir los problemas y soluciones e la construcción del sistema y principalmente porque es el lenguaje de modelado de software más conocido y utilizado.

En nuestro proyecto hemos optado en la realización de los Diagramas más relevantes como son:

- Diagramas de caso de uso
- Diagramas de clases
- Diagramas de secuencias

A parte de los modelos UML vamos a complementar a nuestro diseño con modelos como el Arquitectónico con el objeto de ofrecer una visión simplificada del sistema. Y el de Navegación el cual nos ayudara en la comprensión del orden de presentaciones de las pantallas de nuestro sistema con los contenidos y vínculos.

#### **Diagrama de casos de uso.**

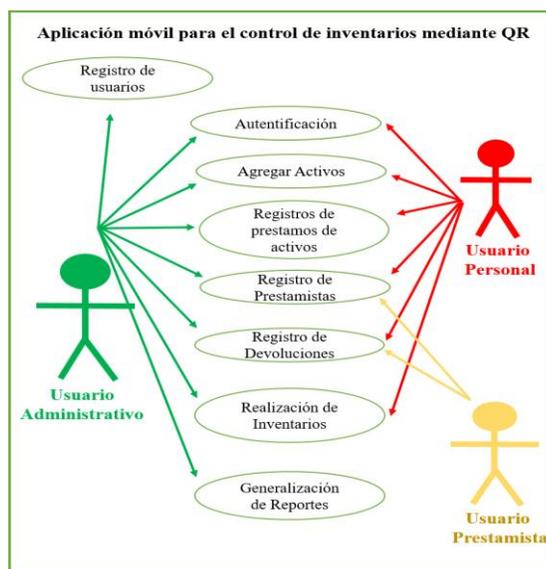
Tabla 3.3 Actores del sistema de difusión de información institucional

Actores de la Aplicación móvil de control de inventarios mediante código QR	
Usuario Administrativo	Usar el sistema, agregar, modificar eliminar Activos, registrar, modificar prestamos, e involucrados. Realizar inventarios y generar reportes.
Usuario personal	Usar el sistema, agregar Activos, registrar prestamos, e involucrados. Realizar inventarios.
usuario Prestamista	Usar el sistema, para el registro de sus datos personales

### 3.3.1.1. Caso de uso general, Aplicación móvil de control de inventarios mediante código QR.

Vamos a ver como el Usuario General interactúa con el escenario del sistema por lo consiguiente aparecerán muchos casos de uso, que luego vamos a irlos especificando uno a uno:

Figura 3.2 Casos de Uso General del Sistema de difusión de información institucional MOVIL



Fuente: (Elaboracion Propia)

## Especificaciones de Casos de Usos:

Tabla 3.4 Especificación de caso de Uso Numero 1. Ingreso a la aplicación

CASO DE USO NUMERO 1											
<b>Nombre</b>	Autenticacion										
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> </ul>										
<b>Descripcion</b>	Identificar al usuario para el uso de la aplicación										
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Tener un usuario</li> <li>• Que la aplicación se haya ejecutado</li> </ul>										
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>La aplicación despliega ventana de inicio de sesion.</td> </tr> <tr> <td>2</td> <td>El usuario ingresa con su usuario y contraseña</td> </tr> <tr> <td>3</td> <td>La aplicación valida los datos en la base de datos</td> </tr> <tr> <td>4</td> <td>la aplicación muestra la interfaz grafica para el usuario</td> </tr> </tbody> </table>	Paso	Accion	1	La aplicación despliega ventana de inicio de sesion.	2	El usuario ingresa con su usuario y contraseña	3	La aplicación valida los datos en la base de datos	4	la aplicación muestra la interfaz grafica para el usuario
	Paso	Accion									
	1	La aplicación despliega ventana de inicio de sesion.									
	2	El usuario ingresa con su usuario y contraseña									
	3	La aplicación valida los datos en la base de datos									
4	la aplicación muestra la interfaz grafica para el usuario										
<b>Precondicion</b>	Si el usuario digita mal su clave no tendra acceso a la interfaz, posteriormente se podra realizar el cambio de contraseña.										

Tabla 3.5 Especificación de caso de Uso Numero 2. Registro de Usuarios

CASO DE USO NUMERO 2											
<b>Nombre</b>	Registro deUsuarios										
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> </ul>										
<b>Descripcion</b>	Realizar el registro de los usuarios personal										
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>										
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar: Registrar Personal</td> </tr> <tr> <td>2</td> <td>Se mostrara una interfaz con espacios para llenar</td> </tr> <tr> <td>3</td> <td>Completar el registro, y su contraseña.</td> </tr> <tr> <td>4</td> <td>Seleccionar guardar, verificando los datos.</td> </tr> </tbody> </table>	Paso	Accion	1	Seleccionar: Registrar Personal	2	Se mostrara una interfaz con espacios para llenar	3	Completar el registro, y su contraseña.	4	Seleccionar guardar, verificando los datos.
	Paso	Accion									
	1	Seleccionar: Registrar Personal									
	2	Se mostrara una interfaz con espacios para llenar									
	3	Completar el registro, y su contraseña.									
4	Seleccionar guardar, verificando los datos.										
<b>Precondicion</b>	El usuario administrativo podra modificar al Usuario personal.										

Tabla 3.6 Especificación de caso de Uso Numero 3. Agregar activos

CASO DE USO NUMERO 3													
<b>Nombre</b>	Agregar Activos												
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> </ul>												
<b>Descripcion</b>	Registrar nuevos activos de la institucion.												
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>												
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar: Registrar Activos</td> </tr> <tr> <td>2</td> <td>Se mostrara una interfaz con espacios para llenar</td> </tr> <tr> <td>3</td> <td>Completar el registro, y la fotografia del activo registrado</td> </tr> <tr> <td>4</td> <td>Seleccionar guardar, verificando los datos.</td> </tr> <tr> <td>5</td> <td>Luego se generara un PDF con los Codigos Qr de Activos.</td> </tr> </tbody> </table>	Paso	Accion	1	Seleccionar: Registrar Activos	2	Se mostrara una interfaz con espacios para llenar	3	Completar el registro, y la fotografia del activo registrado	4	Seleccionar guardar, verificando los datos.	5	Luego se generara un PDF con los Codigos Qr de Activos.
	Paso	Accion											
	1	Seleccionar: Registrar Activos											
	2	Se mostrara una interfaz con espacios para llenar											
	3	Completar el registro, y la fotografia del activo registrado											
4	Seleccionar guardar, verificando los datos.												
5	Luego se generara un PDF con los Codigos Qr de Activos.												
<b>Precondicion</b>	Si hubiese algun observacion del registro, el usuario administrativo podra modificar el registro del activo.												

Tabla 3.7 Especificación de caso de Uso Numero 4. Registro de Prestamos de activos

CASO DE USO NUMERO 4															
<b>Nombre</b>	Registros de prestamos de activos														
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> </ul>														
<b>Descripcion</b>	Registrar la salidas de activos, en calidad de prestamo														
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>														
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Registrar al beneficiario del Prestamo</td> </tr> <tr> <td>2</td> <td>Seleccionar: Registrar Prestamo</td> </tr> <tr> <td>3</td> <td>Se abra un interfaz para escanear elCodigo Qr del activo.</td> </tr> <tr> <td>4</td> <td>Una vez reconocida el Qr, se mostrara la foto del activo.</td> </tr> <tr> <td>5</td> <td>Registrar el estado del activo.</td> </tr> <tr> <td>6</td> <td>Seleccionar guardar, verificando los datos.</td> </tr> </tbody> </table>	Paso	Accion	1	Registrar al beneficiario del Prestamo	2	Seleccionar: Registrar Prestamo	3	Se abra un interfaz para escanear elCodigo Qr del activo.	4	Una vez reconocida el Qr, se mostrara la foto del activo.	5	Registrar el estado del activo.	6	Seleccionar guardar, verificando los datos.
	Paso	Accion													
	1	Registrar al beneficiario del Prestamo													
	2	Seleccionar: Registrar Prestamo													
	3	Se abra un interfaz para escanear elCodigo Qr del activo.													
	4	Una vez reconocida el Qr, se mostrara la foto del activo.													
5	Registrar el estado del activo.														
6	Seleccionar guardar, verificando los datos.														
<b>Precondicion</b>	Si hubiese alguna observacion del registro, el usuario administrativo podra modificar el registro del Prestamo.														

Tabla 3.8 Especificación de caso de Uso Numero 5. Registro de prestamistas

CASO DE USO NUMERO 5											
<b>Nombre</b>	Registros de prestamistas										
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> <li>• Usuario Prestamista</li> </ul>										
<b>Descripcion</b>	Registrar a Beneficiarios de activos en calidad de Prestamo										
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>										
<b>Flujo del sistema</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Previamente autorizada el Prestamo</td> </tr> <tr> <td>2</td> <td>Seleccionar: Registrar Prestamo</td> </tr> <tr> <td>3</td> <td>Se abra un interfaz para llenar datos del beneficiario</td> </tr> <tr> <td>4</td> <td>Seleccionar guardar, verificando los datos.</td> </tr> </tbody> </table>	Paso	Accion	1	Previamente autorizada el Prestamo	2	Seleccionar: Registrar Prestamo	3	Se abra un interfaz para llenar datos del beneficiario	4	Seleccionar guardar, verificando los datos.
	Paso	Accion									
	1	Previamente autorizada el Prestamo									
	2	Seleccionar: Registrar Prestamo									
	3	Se abra un interfaz para llenar datos del beneficiario									
4	Seleccionar guardar, verificando los datos.										
<b>Precondicion</b>	Si hubiese alguna observacion del registro, el usuario administrativo podra modificar el registro del Beneficiario del Prestamo.										

Tabla 3.9 Especificación de caso de Uso Numero 6. Realización de Inventario

CASO DE USO NUMERO 6															
<b>Nombre</b>	Realizacion de Inventarios														
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> </ul>														
<b>Descripcion</b>	Comienzo del proceso de inventarios, mediante elCodigo QR														
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>														
<b>Flujo del sistema</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar: Realizar Inventario.</td> </tr> <tr> <td>2</td> <td>Se abra un interfaz para escanear elCodigo Qr del activo.</td> </tr> <tr> <td>3</td> <td>Una vez reconocida el Qr, saldra la foto del activo</td> </tr> <tr> <td>4</td> <td>Corroborada con las características y la foto</td> </tr> <tr> <td>5</td> <td>Registrar el estado del activo.</td> </tr> <tr> <td>6</td> <td>Seleccionar guardar, y seguira habilitado el escaner de QR.</td> </tr> </tbody> </table>	Paso	Accion	1	Seleccionar: Realizar Inventario.	2	Se abra un interfaz para escanear elCodigo Qr del activo.	3	Una vez reconocida el Qr, saldra la foto del activo	4	Corroborada con las características y la foto	5	Registrar el estado del activo.	6	Seleccionar guardar, y seguira habilitado el escaner de QR.
	Paso	Accion													
	1	Seleccionar: Realizar Inventario.													
	2	Se abra un interfaz para escanear elCodigo Qr del activo.													
	3	Una vez reconocida el Qr, saldra la foto del activo													
	4	Corroborada con las características y la foto													
	5	Registrar el estado del activo.													
6	Seleccionar guardar, y seguira habilitado el escaner de QR.														
<b>Precondicion</b>	Si hubiese alguna observacion del registro, el usuario administrativo														

Tabla 3.10 Especificación de caso de Uso Numero 7. Generalización de Reportes

CASO DE USO NUMERO 7													
<b>Nombre</b>	Generalizacion de Reportes												
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Personal</li> </ul>												
<b>Descripcion</b>	La aplicación genera reportes a travez de la base de datos.												
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>												
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar: Generar reportes.</td> </tr> <tr> <td>2</td> <td>Se abra una interfaz con pestañas.</td> </tr> <tr> <td>3</td> <td>Reportes: Inventarios, Activos, Prestamos y Prestamistas.</td> </tr> <tr> <td>4</td> <td>Seleccionar: Exportar a PDFs.</td> </tr> <tr> <td>5</td> <td>Seleccionar Salir.</td> </tr> </tbody> </table>	Paso	Accion	1	Seleccionar: Generar reportes.	2	Se abra una interfaz con pestañas.	3	Reportes: Inventarios, Activos, Prestamos y Prestamistas.	4	Seleccionar: Exportar a PDFs.	5	Seleccionar Salir.
	Paso	Accion											
	1	Seleccionar: Generar reportes.											
	2	Se abra una interfaz con pestañas.											
	3	Reportes: Inventarios, Activos, Prestamos y Prestamistas.											
4	Seleccionar: Exportar a PDFs.												
5	Seleccionar Salir.												
<b>Precondicion</b>	Los reportes se actualizaran según la base de datos vaya modificando.												

Tabla 3.11 Especificación de caso de Uso Numero 8. Registrar devoluciones

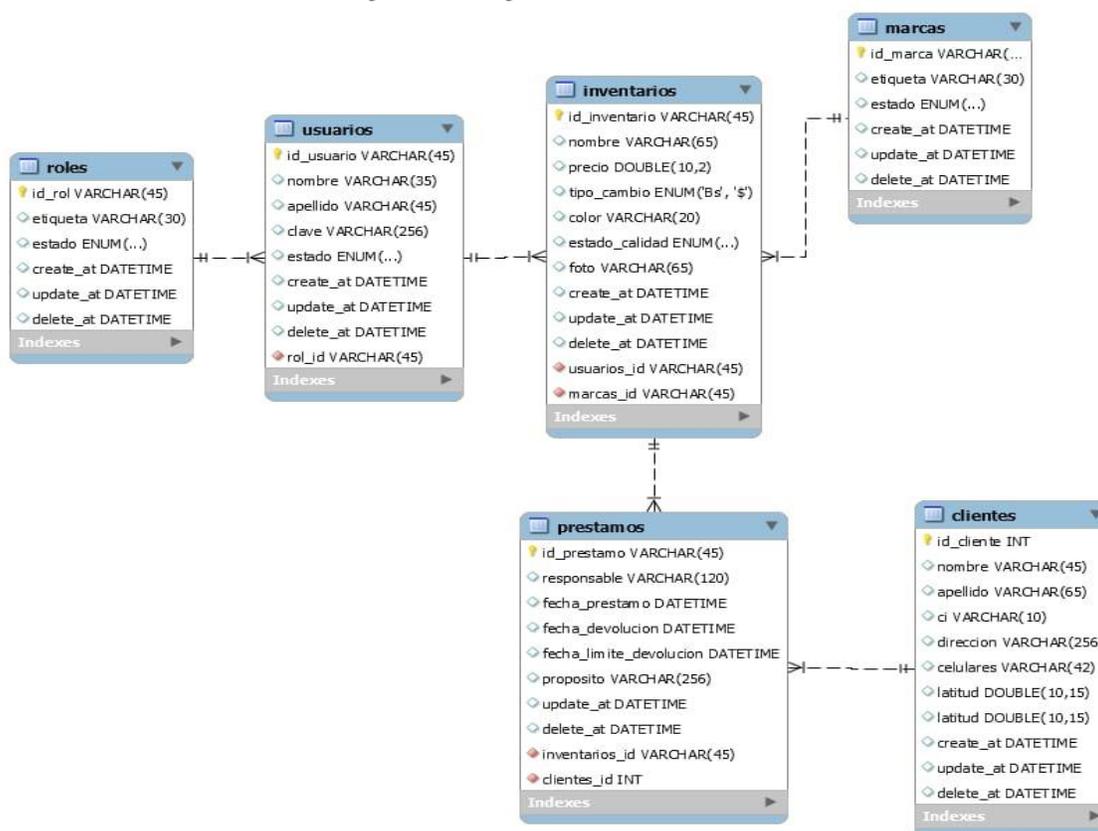
CASO DE USO NUMERO 8															
<b>Nombre</b>	Registrar devoluciones														
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Usuario Administrativo</li> <li>• Usuario Prestamista</li> </ul>														
<b>Descripcion</b>	Se realiza el registro del activo devuelto, en el mismo estado.														
<b>Precondicion</b>	<ul style="list-style-type: none"> <li>• Conexión a Base de datos Satisfactoria</li> <li>• Que la aplicación se haya ejecutado</li> </ul>														
<b>Flujo del sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Accion</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar: Registro de Devoluciones</td> </tr> <tr> <td>2</td> <td>Se abra un interfaz para escanear elCodigo Qr del activo.</td> </tr> <tr> <td>3</td> <td>Una vez reconocida el Qr, saldra la foto del activo</td> </tr> <tr> <td>4</td> <td>Corroborada con las características y la foto</td> </tr> <tr> <td>5</td> <td>Registrar el estado del activo.</td> </tr> <tr> <td>6</td> <td>Seleccionar guardar, y seguira habilitado el escaner de QR.</td> </tr> </tbody> </table>	Paso	Accion	1	Seleccionar: Registro de Devoluciones	2	Se abra un interfaz para escanear elCodigo Qr del activo.	3	Una vez reconocida el Qr, saldra la foto del activo	4	Corroborada con las características y la foto	5	Registrar el estado del activo.	6	Seleccionar guardar, y seguira habilitado el escaner de QR.
	Paso	Accion													
	1	Seleccionar: Registro de Devoluciones													
	2	Se abra un interfaz para escanear elCodigo Qr del activo.													
	3	Una vez reconocida el Qr, saldra la foto del activo													
	4	Corroborada con las características y la foto													
5	Registrar el estado del activo.														
6	Seleccionar guardar, y seguira habilitado el escaner de QR.														
<b>Precondicion</b>	Si hubiese alguna observacion del registro, el usuario administrativo podra modificar el registro del Prestamo.														

## Diagrama de Clases.

El diagrama que se mostrara a continuación está directamente relacionado con las clases que se utilizan en la aplicación móvil, ya que para poder contribuir o desarrollar la misma fue necesario crear nuestras propias tablas en la base de datos de nuestro proyecto Android, es importante recalcar que algunas tablas y clases son las mismas que existen en la base de datos principal de donde extraemos los datos, algunas otras son creadas de acuerdo a la necesidad de nuestro proyecto.

El en siguiente diagrama vamos a ver todas las clases con sus respectivos atributos y como están relacionados unas con otras.

Figura 3.3 Diagrama de Clases General



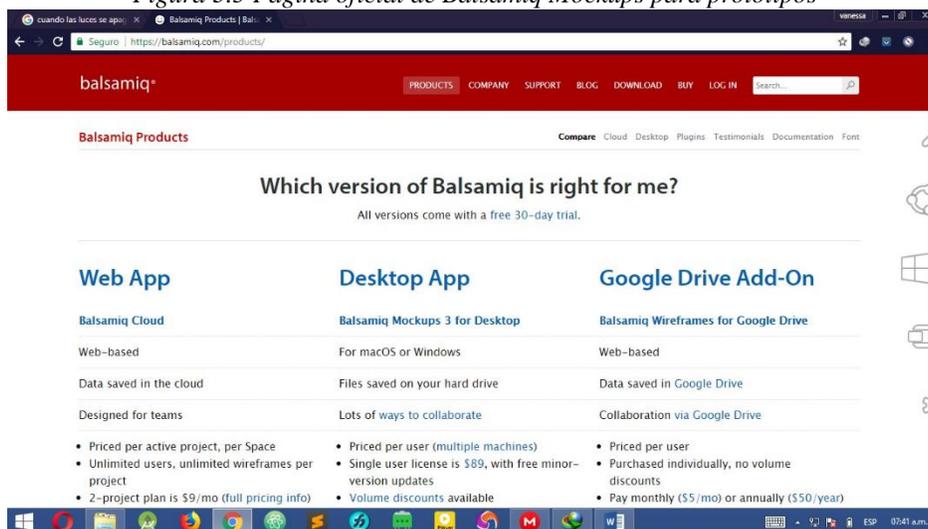
Fuente: (Elaboración Propia)

### 3.4. Diseño de Bocetos de las Pantallas de la Aplicación.

Para la parte de diseño de prototipos de nuestro proyecto cabe recalcar que la realizaremos gracias a la ayuda de una herramienta de escritorio (offline) llamado Balsamiq Mockups, que lo podemos descargar de su página oficial:

<https://balsamiq.com/products/>, no requiere ningún tipo de registro para su uso.

*Figura 3.5 Pagina oficial de Balsamiq Mockups para prototipos*



Fuente: (Elaboración Propia)

Al ser una herramienta de nivel intermedio avanzado sirve para realizar la creación de bocetos para Distintos tipos de plataformas como: iOS, Android, Windows Phone, desktop, Web, etc. Los elementos que se agrega a la aplicación ofrecen un acabado de dibujo a mano alzada, su principal ventaja es que es gratuita. La interacción de sus prototipos no va más allá de enlaces entre pantallas. Para ideas básicas, para diseño rápido y para un primer boceto, resulta muy eficiente para lo que buscamos en el proyecto.

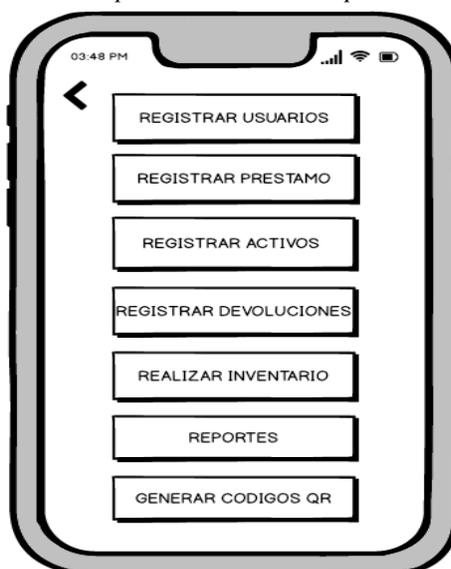
## Bocetos de la aplicación Móvil.

Figura 3.6 Prototipo Pantalla de Inicio de Sesión



Fuente: (Elaboración Propia)

Figura 3.7 Prototipo de Pantalla Principal Administrativo



Fuente: (Elaboración Propia)

Figura 3.8 Prototipo de Pantalla registro de Usuarios



04:04 PM

<

REGISTRO DE USUARIO

Nombre:

Apellido:

Estado:

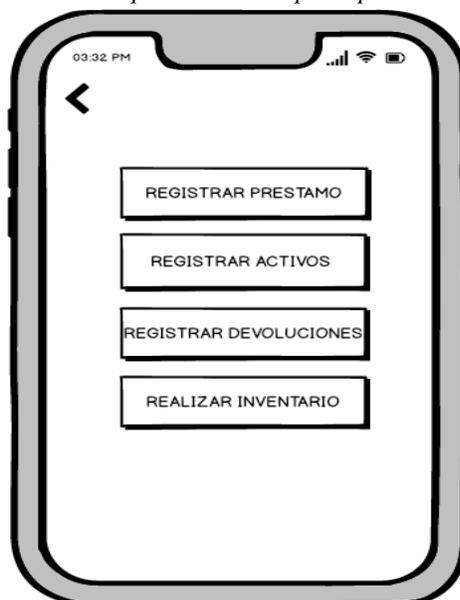
Clave:

+    

Detailed description: This is a mobile app prototype for a user registration screen. At the top, the status bar shows the time as 04:04 PM, along with signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow icon. The main title is 'REGISTRO DE USUARIO'. There are four input fields: 'Nombre:', 'Apellido:', 'Estado:', and 'Clave:'. At the bottom, there are two icons: a plus sign in a circle and a floppy disk icon.

Fuente: (Elaboración Propia)

Figura 3.9 Prototipo de Pantalla principal de Personal



03:32 PM

<

REGISTRAR PRESTAMO

REGISTRAR ACTIVOS

REGISTRAR DEVOLUCIONES

REALIZAR INVENTARIO

Detailed description: This is a mobile app prototype for a main menu screen for staff. At the top, the status bar shows the time as 03:32 PM, along with signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow icon. The screen contains four rectangular buttons stacked vertically: 'REGISTRAR PRESTAMO', 'REGISTRAR ACTIVOS', 'REGISTRAR DEVOLUCIONES', and 'REALIZAR INVENTARIO'.

Fuente: (Elaboración Propia)

Figura 3.10 Prototipo de Pantalla de Registro de Activo



03:32 PM

Nombre:

Precio:

Color:

Marca/Industria:

Estado:

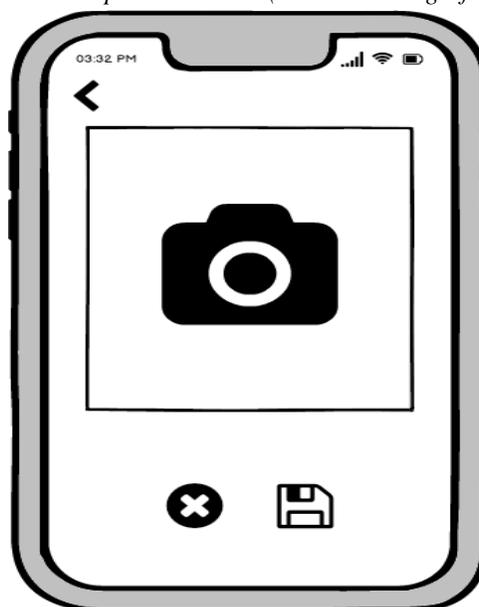
Regular  
Bueno  
Excelent

Nombre | Precio | Color | Marca | Estado | X

+ 

Fuente: (Elaboración Propia)

Figura 3.11 Prototipo de Pantalla (Toma de Fotografía del Activo)



Fuente: (Elaboración Propia)

Figura 3.12 Prototipo de Pantalla de registro de Préstamo de activo



03:32 PM

<

Tutor/Resp:

REGISTRAR PRESTAMISTA

Fecha de devolucion:  / / 

Proposito:

| Tutor | Resp | Fecha | Proposito | Estado | X |

Fuente: (Elaboración Propia)

Figura 3.13 Prototipo de pantalla, Registro de Prestamista



04:01 PM

<

REGISTRO DE PRESTAMISTA

Nombre:

Apellido:

C.I.:

Celular:

Direccion:



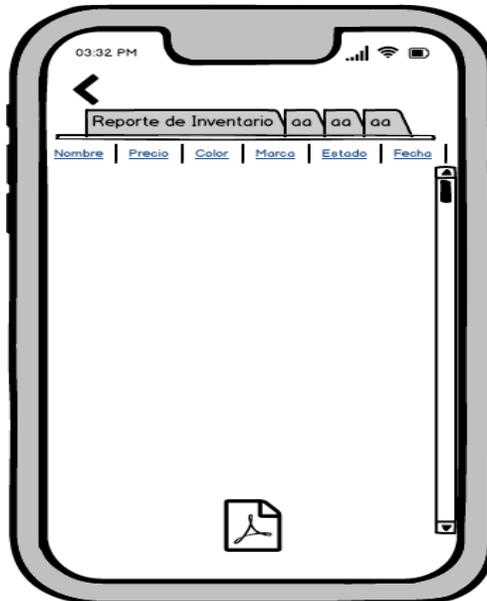
Fuente: (Elaboración Propia)

Figura 3.14 Prototipo de la Realización del Inventario



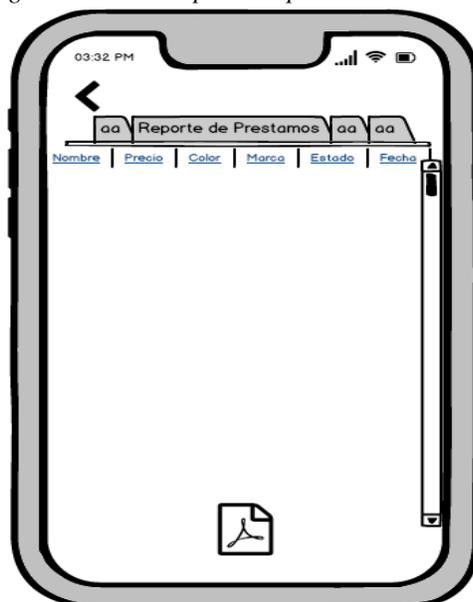
Fuente: (Elaboración Propia)

Figura 3.15 Prototipo de Reportes del Inventario



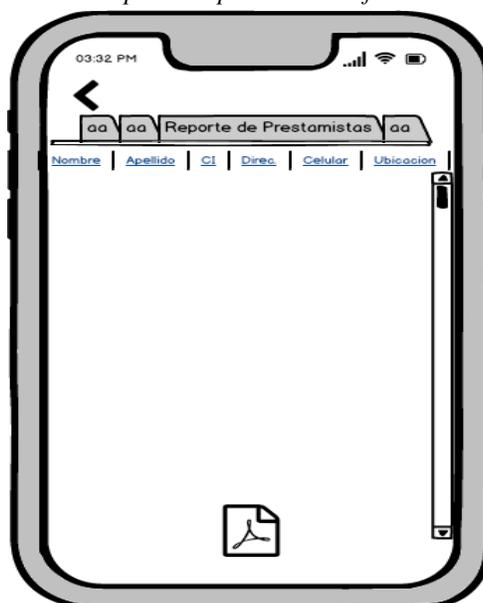
Fuente: (Elaboración Propia)

Figura 3.16 Prototipo de Reportes de Prestamos



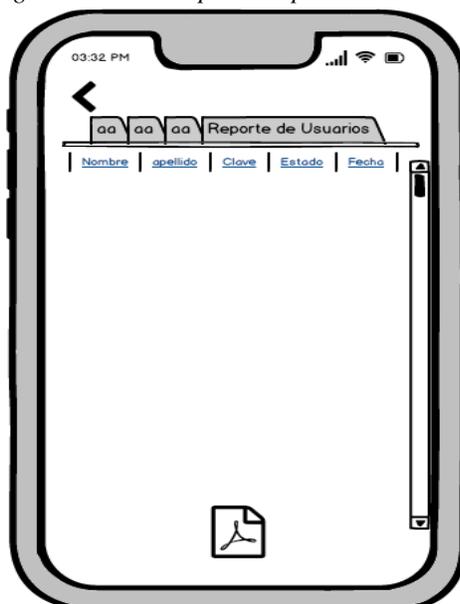
Fuente: (Elaboración Propia)

Figura 3.17 Prototipo de Reporte de Beneficiarios de Préstamo



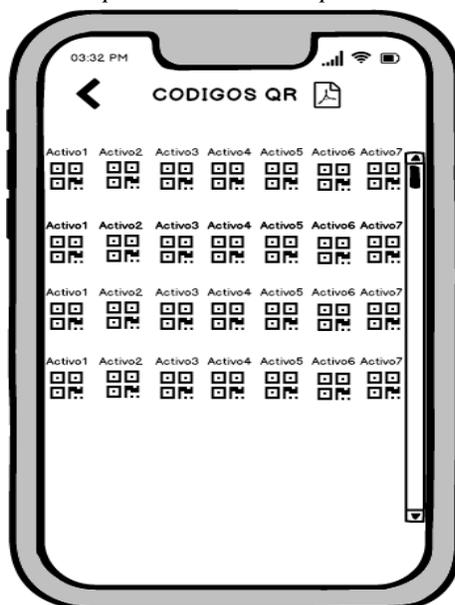
Fuente: (Elaboración Propia)

Figura 3.18 Prototipo de Reporte de Usuarios



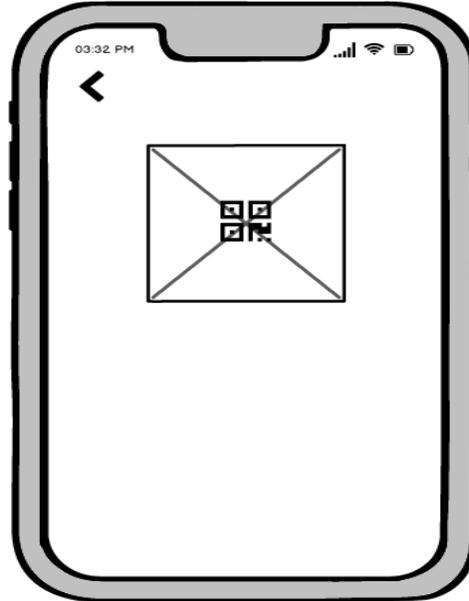
Fuente: (Elaboración Propia)

Figura 3.19 Prototipo de Planilla de Impresión de Códigos QR



Fuente: (Elaboración Propia)

Figura 3.20 Prototipo de Escaneo de Activo por QR



Fuente: (Elaboración Propia)

Figura 3.20 Prototipo de Reconocimiento de Activo



Fuente: (Elaboración Propia)

### 3.5. Herramientas para la codificación.

Para esta parte vamos a ver los requerimientos de desarrollo que se necesitan para crear un proyecto y empezar a programar en Android, así como una breve descripción del código de las clases más relevantes dentro de la aplicación como son:

- La clase Java MainActivity (Clase principal, que llama diferentes objetos)
- La clase Java Login (Conexión y autenticación en la Base de Datos mediante Web Service)
- El archivo XML activity\_main (Crear interfaz para el usuario)

#### Requerimientos de desarrollo.

En el Capítulo I, Sección de Ambientes de desarrollo, se revisa la selección del SGBD y lenguajes de programación a utilizar, ahora detallaremos resumidamente las herramientas, además de servicios o aplicaciones que se necesitan para empezar a desarrollar un proyecto en Android.

- Entorno de Desarrollo para la Aplicación
  - JDK Java.
  - Android Studio (IDE)
  - ADT (Plugins)
  - Android SDK (Librerías y Paquetes)
  - MariaDB (Base de Datos)
  - ADV (Emulador Android)
  - Web Service (JSON)
- Entorno de Acceso al servidor y base de datos
  - MySQL
  - PhpAdmin

### **3.6. Descripción de Pruebas.**

Una vez finalizada la fase de codificación y para la comprobación del correcto funcionamiento del Sistema en cada uno de sus Menús, se realizaran las pruebas respectivas tomando en cuenta los aspectos siguientes:

- Para la primera prueba Funcional vamos a elaborar una pequeña síntesis sobre las “historia de usuarios” al momento de utilizar la aplicación para lo cual se ha tomado en cuenta una Autenticación Correcta y una autenticación incorrecta de usuario al momento de iniciar sesión.
- En segundo lugar vamos a probar el sistema en una Smartphone de gama media y baja, que cumplan los requisitos mínimos de nuestra aplicación Android. Lo realizaremos con la Directora y el personal, con datos reales para ver los tiempos de repuestas.

#### **Elaboración y Ejecución del plan de pruebas.**

Se lo realizara en un Smartphone de gama media y baja de las siguientes características:

Smartphone de gama media

- Huawei P20 Lite
- Procesador Kirin 625 1.8GHz
- 3GB RAM
- Android 8.0

Smartphone de gama baja

- Samsung J1
- Procesador Spreadtrum dual-core 1.2GHz
- 512MB RAM
- Android 4.4

Tabla 3.12 Gamas de smarphone 2020

Smartphone de Gama media		Smartphone de Gama baja	
Huawei p20 lite		Samsung J1	
Motorola moto g6		Huawei p8 lite	
Honor 10		Sony Xperia L1	
Samsung Galaxy A8		Xiaomi 5a	
Xiaomi redmi 5		Motorola g5	

Fuente: (Elaboración propia)

## Análisis de Resultados

Este análisis en base a la segunda prueba funcional lo detallamos en la siguiente tabla.

*Tabla 3.13 Análisis de resultado del plan de pruebas*

<b>Control de Inventarios</b>			
<b>1</b>	<b>Inicio de Sesión</b>	Administrador y Personal	El logo funciona correctamente, la aplicación se demora un leve momento ya que tiene que cargar toda la información.
<b>2</b>	<b>Visualizar Inventario</b>	Administrador y Personal	Todo el inventario aparece en el inicio de Sesión, se carga con una pequeña imagen de referencia.
<b>3</b>	<b>Visualizar Activos Prestados</b>	Administrador y Personal	En la pestaña prestamos carga óptimamente, en cuanto a las descripciones de cada una de 1 a 5 segundo dependiendo de la velocidad de internet.
<b>4</b>	<b>Visualizar Beneficiarios de Prestamos</b>	Administrador y Personal	En cada salida de activo por préstamo, se hace el registro del beneficiario de dicho activo. El listado carga óptimamente.
<b>5</b>	<b>Generalización de Reportes</b>	Administrador y Personal	La generalización de Reporte es muy bueno, no tarda en descargarse en el dispositivo en formato PDF, para su visualización o impresión de dichos Reportes.
<b>6</b>	<b>Generalización de Códigos QR</b>	Administrador y Personal	De igual manera se descarga en el dispositivo, en formato PDF de todos los activos existentes para su impresión.

7	<b>Información</b>	Administrador y Personal	Se carga los datos y envía datos de manera óptima y sin inconvenientes. Hay una perfecta interacción con la base de datos.
8	<b>Escaneo de Código QR</b>	Administrador y Personal	Se habilita la aplicación de cámara, que hace el escaneo del Código QR, realiza un reconociendo perfecto del activo además que muestra una fotografía, para su veracidad.

Fuente: (Elaboración Propia)

Las Pruebas por lo general indican cuando el software está terminado. Si las pruebas funcionan la codificación ha terminado.

### **3.7. Pruebas de Caja Negra**

En la evaluación fundamental del sistema se observaron aspectos de funcionalidad, operativa aceptación de entradas, resultado, etc.

Se procede a realizar una evaluación de acuerdo a los módulos para cortejar los resultados que entrega el sistema ya sea mediante reportes o interfaces de resultados.

#### **Inicio de Sesión**

Datos de Ingreso: se ingresa con un login y una contraseña. Asignada por el administrador.

Prueba: Se espera que la aplicación realice las respectivas validaciones para el ingreso a sus configuraciones.

Resultado: La aplicación no permite el ingreso de caracteres fuera del formato de entrada.

### **Visualización de Inventarios**

Datos de ingreso: al iniciar la sesión debe cargar la lista de inventarios

Prueba: Se espera que la aplicación vaya actualizándose constantemente y mostrar todos los activos registrados, además con la imagen y el código QR perteneciente al activo.

Resultado: Muestra óptimamente todos los registros y fotografías de cada inventario.

### **Visualización de activos Prestados**

Datos de ingreso: al seleccionar la pestaña de Préstamos,

Prueba: Se espera que la aplicación vaya actualizándose constantemente una lista/Historial de todos los préstamos realizados, además que se carga un pequeño mensaje de color rojo. De beneficiarios que aún deben los activos.

Resultado: Muestra óptimamente todos los registros y los permite modificar cuales fuera su intención de hacerlo.

### **Visualizar Beneficiarios de préstamos**

Datos de ingreso: al seleccionar la pestaña de Clientes,

Prueba: Se espera que la aplicación vaya actualizándose constantemente una lista de todos los Clientes que se prestaron, además se podrá tener una información más certera

Resultado: Muestra óptimamente todos los registros y los permite modificar cuales fuera su intención de hacerlo.

### **Generalización de Reportes**

Datos de ingreso: al seleccionar Reporte ya sea del inventario o de préstamos

Prueba: Se espera que la aplicación haga una descargar local a la memoria interna de un archivo en formato PDF. Ya configurado en Tam/Carta.

Resultado: se descarga correctamente a la memoria del dispositivo el archivo PDF donde se visualiza lo requerido por el usuario.

### **Generalización de Códigos QR**

Datos de ingreso: al seleccionar Reporte QR

Prueba: Se espera que la aplicación haga una descargar local a la memoria interna de un archivo en formato PDF. Ya configurado en Tam/Carta. Los códigos QR de los activos.

Resultado: se descarga correctamente a la memoria del dispositivo el archivo PDF donde se visualiza lo requerido por el usuario.

### **Registro de Nuevo Activo**

Datos de ingreso: se ingresa al módulo de registro de nuevo activo.

Prueba: se espera que el sistema despliegue todos los registros de activos que ingresa al módulo los cuales van a poder ser seleccionados y/o modificados por si fuera necesario.

Resultado: La aplicación muestra todos los registros constantemente se actualizan para hacer la visualización de los nuevos registros.

### **Registro de Nuevo Préstamo**

Datos de ingreso: se ingresa al módulo de registro de nuevo préstamo.

Prueba: se espera que el sistema despliegue todos los registros de activos que ingresa al módulo los cuales van a poder ser seleccionados y/o modificados por si fuera necesario.

Resultado: La aplicación muestra todos los registros constantemente se actualizan para hacer la visualización de los nuevos registros.

### **Registro de Nuevo Beneficiario/Cliente**

Datos de ingreso: se ingresa al módulo de registro de nuevo Beneficiario/Cliente.

Prueba: se espera que el sistema despliegue todos los registros de activos que ingresa al módulo los cuales van a poder ser seleccionados y/o modificados por si fuera necesario.

Resultado: La aplicación muestra todos los registros constantemente se actualizan para hacer la visualización de los nuevos registros.

### **Escaneo del Código QR**

Datos de ingreso: al seleccionar Escanear QR

Prueba: Se espera que la aplicación habilite la cámara para hacer escanear o leer de manera alzada el código QR e identificar los datos de dicho activo.

Resultado: Óptimamente reconoce los QR de los activos, también que tiene una prueba de error con distintos QR, simplemente no hace nada hasta encontrar el QR familiar por la aplicación.

### **3.8. Prueba de Caja Blanca**

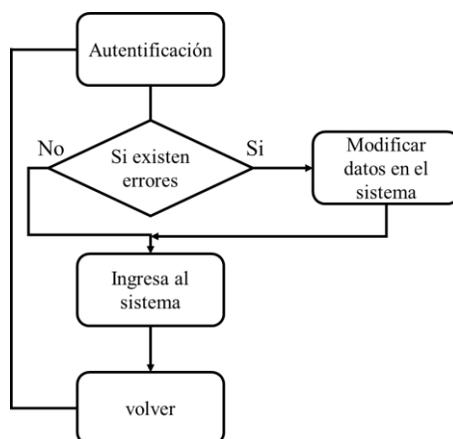
Para las pruebas de caja blanca se proporcionara una medición cuantitativa de la complejidad lógica del programa. La complejidad ciclomatica de un grado de flujo  $V(G)$  establece el número de caminos independientes:

- El número de regiones del grafico del flujo
- $V(G)=A-N+2$ , donde A es el número de aristas y N es el número de nodos.
- $V(G)=P+1$ , donde P es el número de nodos predicado

## Gestión de Usuarios

Se vio necesario realizar pruebas de caja blanca para el módulo de ingreso al autenticación de usuario del sistema debido al flujo grama que presentaba como se puede mostrar en la figura.

Figura Prueba de Caja blanca para la autenticación de usuario



Fuente: (elaboración propia)

Muestra la complejidad ciclomatica de un módulo de autenticación de usuario, por lo que reemplazando el número de predicados en la segunda fórmula se obtiene.

$$V(G) = 1 + 1 = 2$$

Este valor determina dos caminos independientes, dos casos de prueba para la autenticación de usuarios, por lo tanto se realizara para pruebas necesarias para cada uno de los caminos como se muestra en la siguiente tabla.

Camino 1	Camino 2
<ol style="list-style-type: none"> <li>1. Ingresar datos en el formulario</li> <li>2. Modifica datos en el sistema para su ingreso</li> </ol>	<ol style="list-style-type: none"> <li>1. Ingresar datos en el formulario.</li> <li>2. Verifica y valida si existen datos incorrectos en el formulario.</li> <li>3. Ingresa al sistema</li> <li>4. Vuelve a ingresar datos para un nueva sesión.</li> </ol>

## 4. CAPITULO 1V CALIDAD Y SEGURIDAD

### 4.1. ANALISIS DE CALIDAD

#### FACTORES DE CALIDAD MCCALL

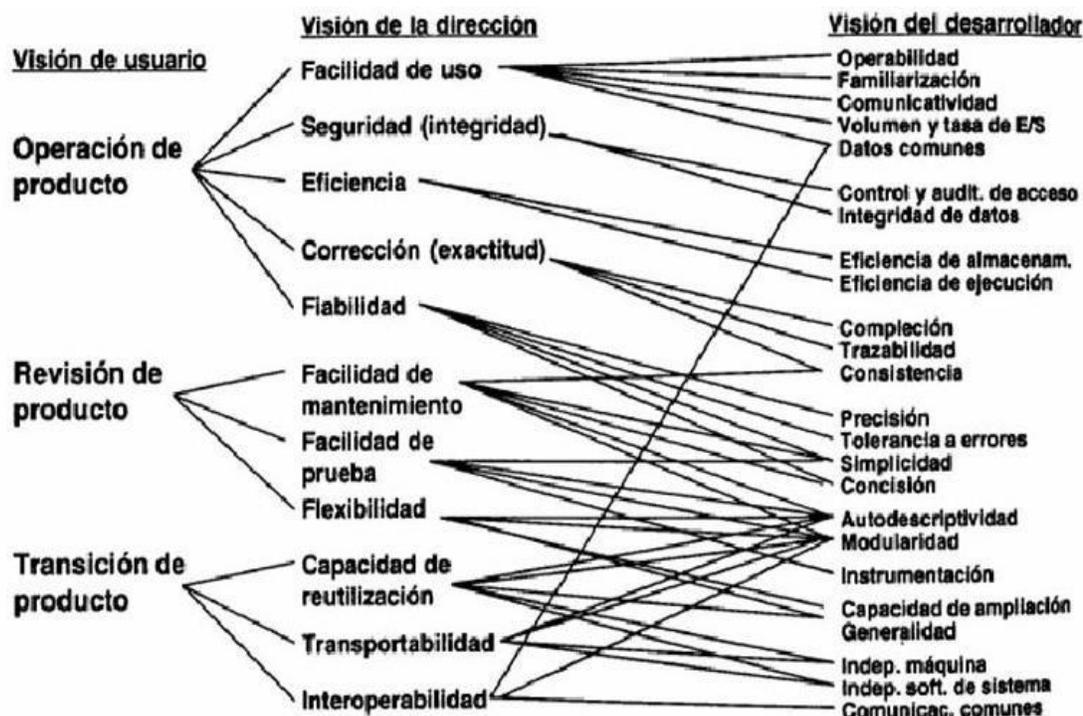
*Tabla 4.1 Puntuación de métricas según usuarios del sistema*

METRICAS	USUARIO 1	USUARIO2	PROMEDIO
<b>Compleitud</b>	9	8	<b>8,50</b>
<b>Trazabilidad</b>	9	8	<b>8,50</b>
<b>Consistencia</b>	9	9	<b>9,00</b>
<b>Simplicidad</b>	8	9	<b>8,50</b>
<b>Exactitud</b>	10	10	<b>10,00</b>
<b>Tolerancia a errores</b>	9	9	<b>9,00</b>
<b>Integridad de datos</b>	9	9	<b>9,00</b>
<b>Facilidad de auditoria</b>	8	8	<b>8,00</b>
<b>Control de acceso</b>	7	8	<b>7,50</b>
<b>Eficiencia de almacenamiento</b>	8	8	<b>8,00</b>
<b>Eficiencia de ejecución</b>	9	8	<b>8,50</b>
<b>Tolerabilidad</b>	9	9	<b>9,00</b>
<b>Familiarización</b>	8	9	<b>8,50</b>
<b>Comunicabilidad</b>	9	8	<b>8,50</b>
<b>Volumen y tasa de E/S</b>	7	8	<b>7,50</b>
<b>Estandarización de datos</b>	9	9	<b>9,00</b>
<b>Concisión</b>	6	7	<b>6,50</b>
<b>Instrumentación</b>	9	9	<b>9,00</b>
<b>Simplicidad</b>	8	9	<b>8,50</b>
<b>Modularidad</b>	7	9	<b>8,00</b>
<b>Auto-Descriptiva</b>	5	7	<b>6,00</b>
<b>Capacidad de ampliación</b>	7	5	<b>6,00</b>
<b>Generalidad</b>	5	8	<b>6,50</b>
<b>Independencia de hardware</b>	5	6	<b>5,50</b>
<b>Independencia de sistema</b>	8	7	<b>7,50</b>
<b>Normalización de comunicación</b>	8	6	<b>7,00</b>

Fuente: (Elaboracion Propia)

El esquema de graduación propuesto por McCall va en una escala de 0 (bajo) a 10 (alto). Se califican solo las métricas que corresponden al factor.

Figura 4.1. Relaciones entre las capacidades, factores y métricas según McCall



Fuente: (<https://slideplayer.es/slide/124874>)

Antes de comenzar a utilizar el modelo de McCall hay que seguir las siguientes pautas:

- Se aceptan los factores, criterios y métricas que propone el modelo.
- Se aceptan las relaciones entre factores de calidad sobre los que aplicar los requisitos de calidad establecidos para el proyecto.
- Se selecciona un subconjunto de factores de calidad sobre los que aplicar los requisitos de calidad establecidos para el proyecto.

Se tomara como guía de la determinación de la relación/Conste en la **Tabla 2.2**.

## Capacidad: Características de operación

Tabla 4.2 Características de operación

CARACTERÍSTICAS DE OPERACIÓN					
Métrica	FACTOR				
	Correccion	Fiabilidad	Seguridad	Eficiencia	Facilidad uso
Compleitud	8,5				
Trazabilidad	8,5				
Consistencia	9				
Simplicidad		8,5			
Exactitud		10			
Tolerancia a errores		9			
Integridad de datos			9		
Facilidad de auditoria			8		
Control de acceso			7,5		
Eficiencia de almacenamiento				8	
Eficiencia de ejecución				9	
Tolerabilidad					9
Familiarización					8,5
Comunicabilidad					9
Volumen y tasa de E/S					7,5
Estandarización de datos					9
<b>Promedio</b>	<b>8,67</b>	<b>9,17</b>	<b>8,17</b>	<b>8,50</b>	<b>8,60</b>
<b>PROMEDIO GENERAL</b>	<b>8,62</b>				

Fuente: (Elaboración Propia)

La característica de Operación tiene un puntuación de 8,62 en una escala de 0 al 10, lo cual indica que el factor de calidad es alto.

## Capacidad: Capacidad de soportar cambios

Tabla 4.3 Capacidad de soportar cambios

CARACTERÍSTICAS DE REVISION			
Metrica	FACTOR		
	Facilidad de Mantenimiento	Facilidad Prueba	Flexibilidad
Consistencia	8,67		
Concisión	6,5		
Instrumentación		9	
Simplicidad	8,5	8,5	
Modularidad	7,67	7,67	7,67
Auto - Descriptiva	6	6	6
Capacidad de ampliación			6
Generalidad			6,5
<b>Promedio</b>	<b>7,46</b>	<b>7,79</b>	<b>6,54</b>
<b>PROMEDIO GENERAL</b>			<b>7,27</b>

Fuente: (Elaboración propia)

La característica de revisión tiene una puntuación de 7.4 en una escala de 0 al 10, lo cual indica que el factor de calidad es alto.

## Capacidad: Adaptabilidad a nuevos entornos

Tabla 4.4 Adaptabilidad a nuevos entornos

CARACTERÍSTICAS DE REVISION			
Metrica	FACTOR		
	Reusabilidad	Portabilidad	Interoperabilidad
Generalidad	6,50		
Auto - Descriptiva	6,00	6,00	
Independencia de hardware	5,50	5,50	
Independencia de sistema	7,50	7,50	
Modularidad	7,27	7,27	7,27
Estandarización de datos			9,00
Normalización de comunicación			7,00
<b>Promedio</b>	<b>6,55</b>	<b>6,57</b>	<b>7,76</b>
<b>PROMEDIO GENERAL</b>			<b>6,96</b>

Fuente: (Elaboración Propia)

La característica de transmisión tiene una puntuación de 6,96 en una escala de 0 al 10, lo cual indica el factor de calidad es alto.

## Medida de Calidad

*Tabla 4.5 Medida de Calidad y Valoración*

<b>Factor</b>	<b>Beneficio / coste</b>	<b>Valoración</b>
Corrección	alto	<b>8,89</b>
Fiabilidad	alto	<b>8,91</b>
Eficiencia	bajo	<b>7,78</b>
Integridad	bajo	<b>8,5</b>
Facilidad de uso	medio	<b>8,6</b>
Facilidad de mantenimiento	alto	<b>7,6</b>
Facilidad de prueba	alto	<b>7,83</b>
Flexibilidad	medio	<b>6,75</b>
Portabilidad	medio	<b>6,8</b>
Reusabilidad	medio	<b>6,92</b>
Interoperabilidad	bajo	<b>8,22</b>
<b>PROMEDIO</b>		<b>7,89</b>

Fuente: (Elaboracion propia)

### Cálculos de Calidad

Para hallar la puntuación final de calidad de software se promediaran los resultados de los factores:

Para calcular la valoración de la calidad del sistema se utiliza esta fórmula.

$$\mathbf{Medida} = \frac{(5 \times \text{Operacion}) + (3 \times \text{Revision}) + (3 \times \text{Transicion})}{11}$$

Donde las variables tienen los siguientes valores:

$$\mathbf{Operación} = 8,62$$

$$\mathbf{Revisión} = 7,27$$

$$\mathbf{Transición} = 6,96$$

Entonces:

$$\mathbf{Medida} = \frac{(5 \times 8,62) + (3 \times 7,27) + (3 \times 6,96)}{11}$$

$$\mathbf{Medida} = 7,8$$

El valor de la puntuación de calidad de software final es de 7,8, en la calidad escala de 0 a 10 este valor indica que el SE DMC.

## 5. CAPITULO V

### COSTO BENEFICIO

#### 5.1. Análisis de Costos

Para determinar el costo del sistema y su tiempo de desarrollo utilizaremos la técnica de estimación de tiempo y coste de software llamada COCOMO II, para lo cual utilizaremos el modelo siguiente:

#### Modelo de coste de composición de aplicaciones

##### a) Recuento de objetos

- Numero de pantallas = 18
- Numero de Informes = 5

##### b) Clasificación de cada instancia de objeto

La clasificación de cada instancia de objeto se realiza dentro de los niveles de complejidad simple y difícil.

##### - Pantallas:

- Complejidad Simple: “Inicio de Sesión”, “Pestañas de Inventario, Prestamos y Clientes”. “Registro de Marca” = 3
- Complejidad Medio: “Registro de Usuarios”, “Registro de Inventarios”, “Registro de Prestamos”, “Registro de Beneficiarios de Activos prestados”, “Vista de Inventarios”, “Vista de Prestamos”, “Vista Clientes”, “Vista Marcas”, “Modificación de Inventario”, “Modificación de Clientes”, “Modificación de Prestamos”, “Modificación de Marcas”, “Toma de Fotografías”= 13
- Complejidad Difícil: “Autenticación por huella de digital”, “Escaneo del código QR” = 2

- **Informes**

- Complejidad Medio: “Reportes del Inventario”, “Historial de Prestamos por Cliente”, “Historial de Prestamos por Inventario”, “Historial de Prestamos por Responsable”, “Reportes de código QR del inventario para su impresión”  
= 5

c) **Determinar puntos Objeto**

Pensando el número de cada celda usando la **Tabla 2.4**, este peso representara el esfuerzo relativo que se requiere para implementar una instancia.

$$\begin{aligned}
 OP &= \sum (PANTALLAS \times COMPLEJIDAD) \\
 &+ \sum (INFORMES \times COMPLEJIDAD) \\
 &+ \sum (COMPONENTES 3 GL \times COMPLEJIDAD)
 \end{aligned}$$

$$OP = [(3 \times 1) + (13 \times 2) + (2 \times 3)] + [5 \times 5] + [0 \times 10]$$

$$OP = [3 + 26 + 6] + [25] = \mathbf{60}$$

d) **Estimar el porcentaje de reutilización**

En el sistema se estima un porcentaje de reutilización de código el 30% debido a las características que poseen los objetos, y el mismo sistema para realizar las inferencias. Con este Porcentaje de Reutilización calcularemos los nuevos puntos Objeto NOP:

$$NOP = \frac{(OP) \times (100 - \%Reutilizacion)}{100}$$

$$NOP = \frac{(60) \times (100 - 30)}{100}$$

$$NOP = 42$$

e) **Determinar una ratio de productividad PROD**

La ratio de productividad está dada por la **Tabla 2.5** y está expresado en NOP/Meses-persona.

Como la experiencia y la capacidad de los desarrolladores de media, escogeremos el valor Nominal con un valor de 13. Entonces:

$$PROD = 13 \text{ NOP/MesesPersona}$$

f) **Calculo de valor Meses-Personal estimado**

La fórmula básica para obtener una estimación de esfuerzo es:

$$MM = \frac{NOP}{PROD}$$

$$MM = \frac{42}{13} = 3,23 \Rightarrow 3 \text{ Meses} - \text{persona}$$

g) **Calculo del tiempo de Desarrollo**

Además del esfuerzo se debe estimar el tiempo en el que el sistema podrá ser terminado.

Entonces:

$$TDEV = \left[ 3.67 \times MM^{0.28+0.2 \times (B-1.01)} \right] \times \frac{SCED\%}{100}$$

Tomaremos un alargamiento de calendario NOMINAL por las características del sistema con los cual el valor de SCED% es de 100%

$$b = 0.91 + 0.01 \times \sum_{j=1}^5 SF_j$$

Los valores para los factores de escala que tomaremos (Según la **Tabla 2.7**) son los siguientes:

Tabla 5.1 Valores para los factores de escala

FACTOR DE ESCALA SF	NIVEL	VALOR
PREC	Alto	3.72
FLEX	Bajo	4.05
RESL	Nominal	4.24
TEAM	Alto	2.19
PMAT	Nominal	4.68
<b>TOTAL SF</b>		<b>18.88</b>

Fuente: (Elaboración propia)

$$B = 0.91 + 0.01 \times 18.88$$

$$B = 0.91 + 0.1888$$

$$B = 1.1$$

Tenemos:

$$TDEV = (3.67 \times 3^{0.28+0.2x(1.1-1.01)}) \times \frac{100}{100}$$

$$TDEV = (3.67 \times 3^{0.98}) \times 1$$

$$TDEV = 5.1 \text{ Meses} \Rightarrow 155 \text{ Dias aproximadamente}$$

#### h) **Calculo del Personal requerido**

Para calcular la cantidad de personal que se requiere para el desarrollo del sistema se utiliza la fórmula

$$Pr = \frac{MM}{TDEV}$$

Reemplazando tenemos:

$$Pr = \frac{3 \text{ Meses} - \text{Personas}}{5.1 \text{ Meses}} = 0.6 \text{ personas} \Rightarrow 1 \text{ persona}$$

#### i) **Calculo del costo de desarrollo de software**

El cálculo del costo de desarrollo de software se calcula mediante la fórmula:

$$CS = Pr \times D \times SP$$

Donde:

CS: Costo del desarrollo de software

Pr: Personal requerido

TDEV: tiempo requerido para el desarrollo

SP: sueldo del personal

Teniendo en cuenta el mercado actual se le asignara un sueldo de 200 Dólares mensuales a los desarrolladores (Personal) obteniendo

$$CS = 1 \times 5.1 \text{ Mes} \times \frac{200 \$us}{\text{Mes}} = 1020 \$us$$

j) **Hardware y Software requerido**

Los requisitos mínimos de Hardware y Software para el sistema son:

- Procesador: 1Ghz Dual Core
- Memoria RAM: 512 MB
- Almacenamiento: 30 MB a 8% MB dependiendo del dispositivo
- Pantalla de 4" pulgadas en adelante
- Sistema operativo: Android 4.1 o Superior
- GPS con soporte A – GPS<sup>25</sup>

En las siguientes tablas detallamos el precio de un SmartPhone para realizar el cálculo correspondiente.

*Tabla 5.2 Costos de Hardware y Software*

<b>Partes</b>	<b>Precio \$us</b>
Samsung Galaxy J1: Es un sencillo Smartphone Android con características de gama baja, que incluye una pantalla de 4.3 pulgadas, cámara de 5 megapíxeles, 512MB de RAM, 4GB de almacenamiento interno, Wi-fi, bluetooth, GPS, conectividad HSPA y corre con Android 4.4.	80
<b>TOTAL</b>	<b>80</b>

Fuente: (Mercado Uyustus)

**k) Calculo del Costo del Sistema**

El costo total del software está dado por la fórmula:

$$CTS = CS + HW + SW$$

$$CTS = 1020 + 80$$

$$CTS = 1100$$

EL Costo total del sistema es de 1100 Dólares, este monto nos sirve solo de parámetro económico.

El sistema de difusión de información institucional basada en las plataformas Móvil trae consigo una inversión inicial 1100 Dólares, hay que considerar que los beneficios que este sistema trae son bastantes y se concluye vale la pena invertir en el sistema.

**5.2. BENEFICIO ECONOMICO**

El beneficio económico que se lograra con la implementación del presente trabajo de grado, es muy alto ya que al poseer este sistema no habrá inventario que se pierda. Se tendrá un historial justificado del proceso de deterioro de cada activo. Se podrá hacer compras eficientes de marcas que les han resultado buenos y duraderos. Por ultimo ayudara mucho en el proceso de Auditoria de activos. Porque el uso de la aplicación es muy fácil y didáctica y certera en el reconocimiento de los activos. La realización del presente proyecto tiene como promedio 3 meses

## **6. CAPITULO VI CONCLUSIONES Y RECOMENDACIONES**

### **6.1. Conclusiones**

Se desarrolló un Modelo Relacional para el registro de activos y demás procesos que necesitaba la institución, cumpliendo así con todo lo requerido y dando algunos aportes que permitirán que la institución no tenga problemas y que el control de estos activos no sea tedioso (con más responsabilidad), sino que más bien se acomode de una manera fácil y entretenida.

Se construyó un modelo que emita reportes de los activos y el estado de los mismos, la aplicación cuenta con muchos reportes en formato PDF de descarga rápida al dispositivo y una manera visual concreta.

Se desarrolló un módulo para el Registro de los infractores, esto para tener el control exacto de dichos individuos, teniendo así la información necesaria para ubicarlos y darles la pena que merecen según las normas de la institución.

Se construyó también un módulo que emita reportes de activos prestados y sus beneficiarios, igualmente de descarga rápida y fácil al dispositivo. Además que se los puede hacer la impresión en físico porque está configurado

Se construyó en el Modelo Relacional la ubicación de los activos y la persona con que estará asignada, donde se guarda la longitud y latitud.

El peso de la aplicación APK, es de 5,4 MB permitiendo así la instalación rápida de 5 a 30 segundos dependiendo de las características del dispositivo, la aplicación es entretenida al realizar el control de los activos, es práctico, muy útil, sencillo, y solo necesita una conexión de internet de mínimo de ancho de banda de 100 KB/segundo, para tener una estable conexión con la base de datos.

En lo referente al desarrollo de aplicaciones en Android al ser un sistema operativo libre (Open Source) es decir nos brinda la libertad a los usuarios de adquirirlo y usarlo, puedo concluir que nos fue mucho más fácil la implementación en este sistema operativo ya que permitió desarrollar aplicaciones nativas con herramientas gratuitas y potentes como Android Studio y el SDK de Android , también de alguna manera se nos facilitó ya que Android emplea lenguajes *Java*, con el cual ya estaba familiarizado durante el transcurso de nuestra carrera.

## **6.2. Recomendaciones**

Con la finalización del presente proyecto se pueden efectuar las siguientes recomendaciones:

Una vez implementado la aplicación en el Centro de Desarrollo Integral BO0450 Fundación Ladislao Calani, se recomienda que este sea manipulado por un personal idóneo en el manejo del mismo, ya que de la eficiente manipulación se alcanzara los objetivos planteados en el proceso de registro y control de inventarios.

Se recomienda realizar un mantenimiento preventivo a Smartphone donde será instalado el nuevo sistema. Para evitar fallas durante el funcionamiento de esta.

También se recomienda seguir modificando o mejorando la aplicación Android para que sea más completa ya que al ser una versión inicial carece de algunas funcionalidades como por ejemplo, respaldar datos en tiempo real, implementar depreciaciones, modificar la interfaz gráfica para una mayor amigabilidad entre dispositivo y usuario, recibir notificaciones de artículos que estén por caducar las garantías y optimizar más el gasto de datos (MB) cuando cargue o se suba la información de activos o clientes.

El técnico encargado realizara el Backup mediante

**BIBLIOGRAFIA**

- Alarcon, F. (2010). Desarrollo de sistemas de información. Una metodología basada en el modelado. España: Editorial UPC. BASIC, V. (s.f.). docs.microsoft.com. Obtenido de <https://docs.microsoft.com/enus/dotnet/visual-basic/>
- Blanco, C. P. (2018). Metodología del desarrollo ágil para sistemas móviles. Obtenido de [http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Bilioguias-Biblioteca de la CEPAL. (Julio 2020). Que son los Códigos QR. Obtenido de <https://www.biblioguias.cepal.org/QR>
- VibLabelSRL. (2017). Códigos QR vs Códigos de Barra. Obtenido de <https://www.vib-label.com/index.php>
- Blogspot RUP. (s.f.). Obtenido de <http://rupmetodologia.blogspot.com/search?q=UML>
- California, U. o. (s.f.). Obtenido de [http://www.sunset.usc.edu/csse/research/cocomoii/cocomo\\_main.html](http://www.sunset.usc.edu/csse/research/cocomoii/cocomo_main.html)
- Campuchino, A. M. (2000). Estimación de Software. Universidad de Madrid.
- Developers.android.com. (2018). Developers. Obtenido de <https://developer.android.com/guide/platform/index.html?hl=es-419#hal>
- El Pais, T. (Octubre de 2013). España abraza los Android y desprecia los Windows. Obtenido de <http://tecnologia.elpais.com/tecnologia/2012/10/01/actualidad1380620296?338056.html>

- Guerrero, M. (03 de 11 de 2015). Blog diario.com. Obtenido de <http://manuelguerrero.blogspot.es/1446543763/metodologia-mobile-d-paradesarrollos-de-aplicaciones-moviles/>
- Wikipedia. (2015). Servicio Web. Obtenido de [http://es.wikipedia.org/wiki/Internet\\_Information\\_Services](http://es.wikipedia.org/wiki/Internet_Information_Services)
- Wolrd, p. (Noviembre de 2017). Sistemas operativos móviles: comunicación en tiempo real. Obtenido de <http://www.peworld.com.mx/Articulos/20734.jtml>
- Zona, M. (Agosto de 2014). Mercado dominado por iOS y Android mientras Windows Phone retrocede. Obtenido de <http://www.movilzona.es/2014/08/15/el-964-del-mercado-dominado-por-iosy-android-mientras-windows-phone/>.
- J. Joskowicz. (2018). Reglas y Practicas en Xtreme programming L. Echeverry y L. Delgado. Metodología ágil XP al desarrollo de software. Obtenido de <http://programacionextrema.tripod.com>
- SUPERTEL, Revista Institucional. (2012). Evolución de la telefonía Móvil en el Ecuador (Vol. n°16).
- UMSA, D. d. (2016). UMSA Hageo. Obtenido de <https://play.google.com/store/apps/details?id=bo.umsa.umsahageo>
- Vique, R. R. (2012). Métodos para el desarrollo de aplicaciones móviles. Cataluña
- Web Programmign. (s.f.). Obtenido de <https://www.google.com.bo/url?Fdesarrollode-software-ciclo-de-vida-rup-rational-unified-process>
- Monografias.com. (s.f.). Obtenido de <http://www.m.monografias.com/call/call> Moonfwd. (09 de 2017). UPN MOVIL. Obtenido de <https://play.google.com/store/apps/details?id=com.moofwd.upn>

Netvalle, D. d. (2016). Univalle Movil. Obtenido de

<https://play.google.com/store/apps/details?id=edu.univalle.siumovil>

Open Handset Alliance. (2014). Obtenido de <http://www.openhandsetalliance.com>

Paz, U. A. (2016). UCBMovil. Obtenido de

<https://play.google.com/store/apps/details?id=ucb.edu.bo.ucbmovil>

PHP. (s.f.). aprenderaprogramar.com. Obtenido de

[https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article  
&id=492:iq ue-es-php-y- ipara-que-sirve-un-potente-lenguaje-deprogramacion-para-  
crear-paginas-web-cu00803b&catid=70&Itemid=193](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:iq ue-es-php-y- ipara-que-sirve-un-potente-lenguaje-deprogramacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193)

Programming, E. (s.f.). Exteme Programming: A gentle introduction. Obtenido de

<http://www.extremeprogramming.org>

Rius, M. (27 de 01 de 2016). La Vanguardia. Obtenido de

[http://www.lavanguardia.com/vida/20160127/301705242421/uso-del-movilconsumo-  
digital-penetracion-tabletas.html](http://www.lavanguardia.com/vida/20160127/301705242421/uso-del-movilconsumo-digital-penetracion-tabletas.html)

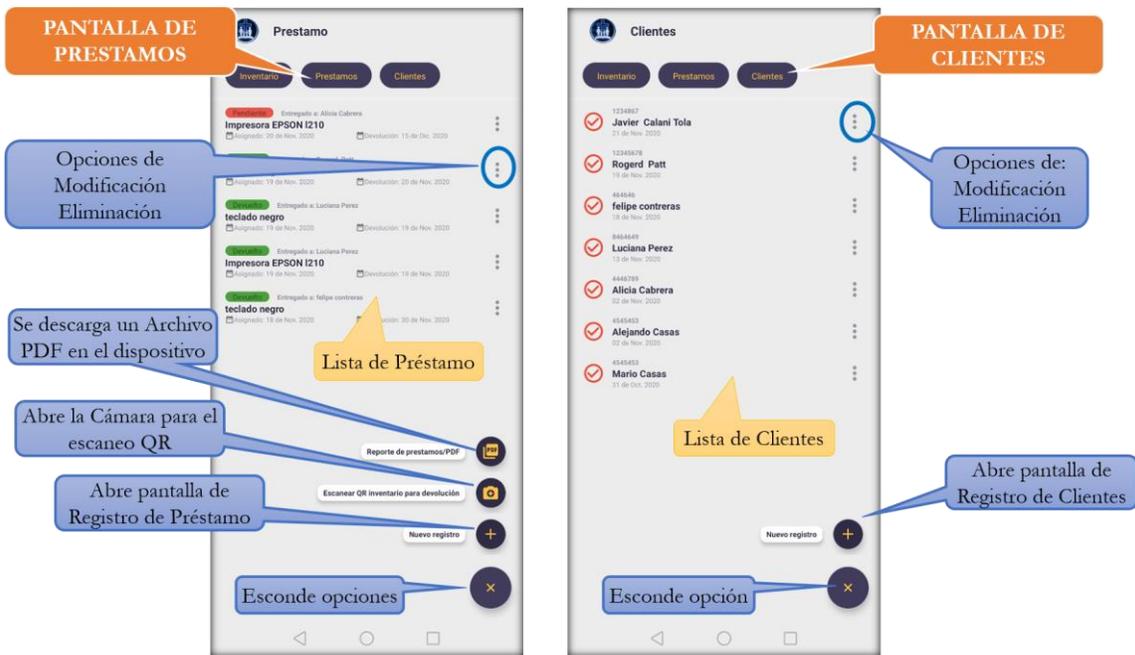
Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. (s.f.). Libro sobre UWE: “WebEngineering: Modelling and Implementing Web Applications”.

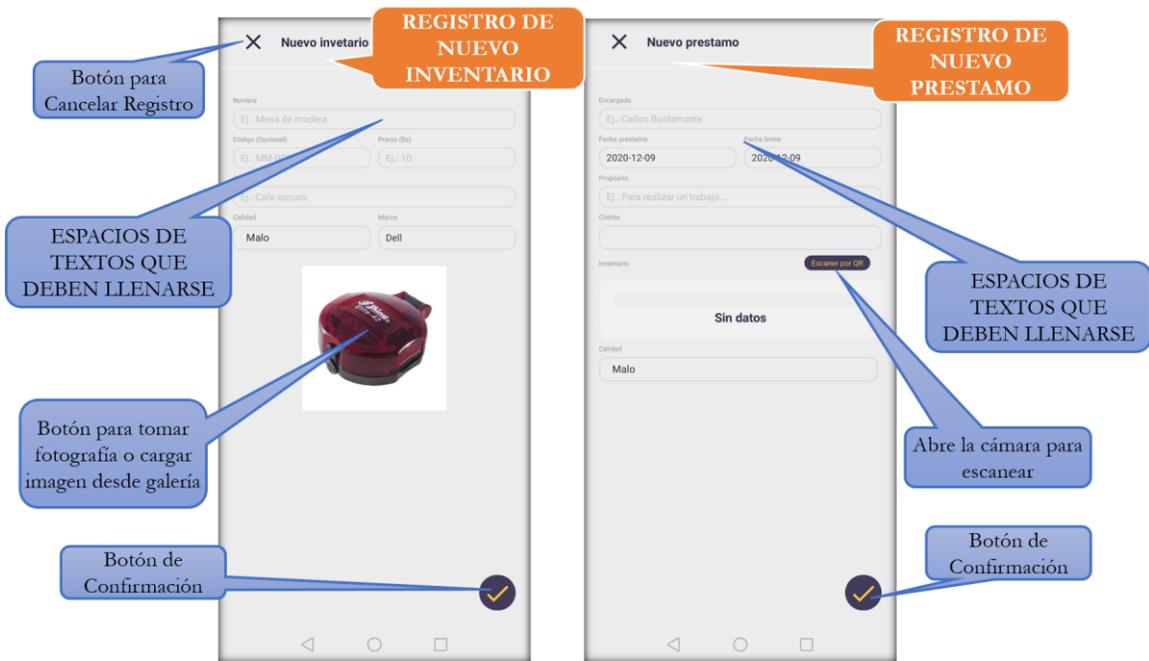
Rouse, M. (01 de 2015). SearchDataCenter MYSQL. Obtenido de

<http://searchdatacenter.techtarget.com/es/definicion/MySQL>

SGOLIVER. (11 de 08 de 2010). Obtenido de [http://www.sgoliver.net/blog/componentes-de-  
una-aplicacion-android/](http://www.sgoliver.net/blog/componentes-de-una-aplicacion-android/)

# MANUAL DEL USUARIO







**Reporte de inventario**

Se detalla todo el inventario disponible  
Fecha del reporte: 2020-12-09 18:28:55

N°	Fotografía	Nombre	Color	Cantidad	Marca
#0		Monitor Dell 15" Codigo:0069	Negro con Plomo	Bueno	Dell
#1		teclado negro Codigo:0069	negro	Bueno	Sin marca
#2		Impresora EPSON 1210 Codigo:	Negro	Regular	Epson
#3		mouse Codigo:7766	negro	Bueno	asus

**VISUALIZACION DE REPORTE PDF TAM/CARTA**

## MANUAL TECNICO

### Características de Software y Hardware

#### Descripción de Software

Las características del software que serán necesarios para el funcionamiento del presente proyecto se muestran a continuación.

Sistema operativo	Android 4.1 (en adelante)
Ancho de Banda	100 KB/segundo

#### Descripción de Hardware

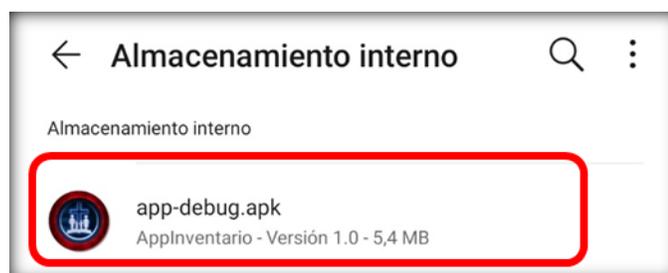
Las características del Hardware que serán necesarios para el funcionamiento del presente proyecto se muestran a continuación.

Procesador	Dual Core 1,2 GHz
Cámara	5 Mega Pixeles
Pantalla	480x800 pixeles (4,3 pulgadas)
Almacenamiento	100 MB (mínimo)
Memoria	512 MB
Batería	1500 mAh

## Instalación de la Aplicación

a) **Paso 1:** Realizar la transferencia del archivo APK (AppInventario – 5,4 MB), al Smartphone a instalarse. Por medio de Bluetooth, Wifi-Direct, Cable USB, u otro medio de transferencia de Archivos.

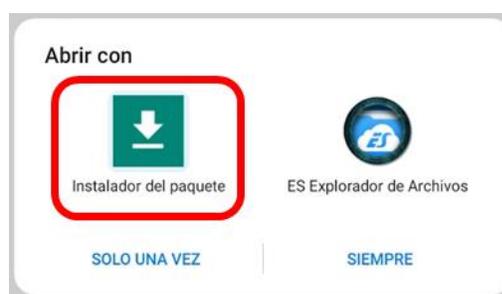
b) **Paso 2:** Localizar el Archivo APK en el Almacenamiento del Smartphone a instalarse.



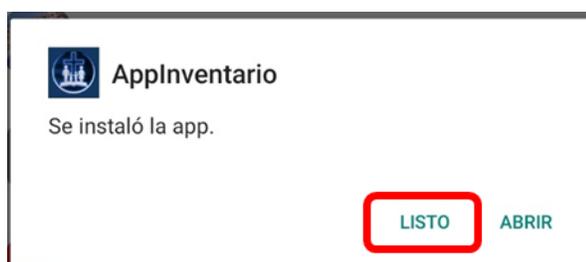
c) **Paso 3:** Clic en el icono del Archivo APK, Dar el permiso necesario para Instalar dicho APK (De acuerdo a las configuraciones del Dispositivo Móvil).



d) **Paso 4:** Elegir la Opción “Instalador de Paquetes” y clic en “Aceptar o Instalar”, en seguida la Instalación empezara su instalación en el Dispositivo.



e) **Paso 5:** Una vez terminada la instalación (5 a 30 segundos dependiendo a las características del Smartphone). Vamos a dar en “Listo”.

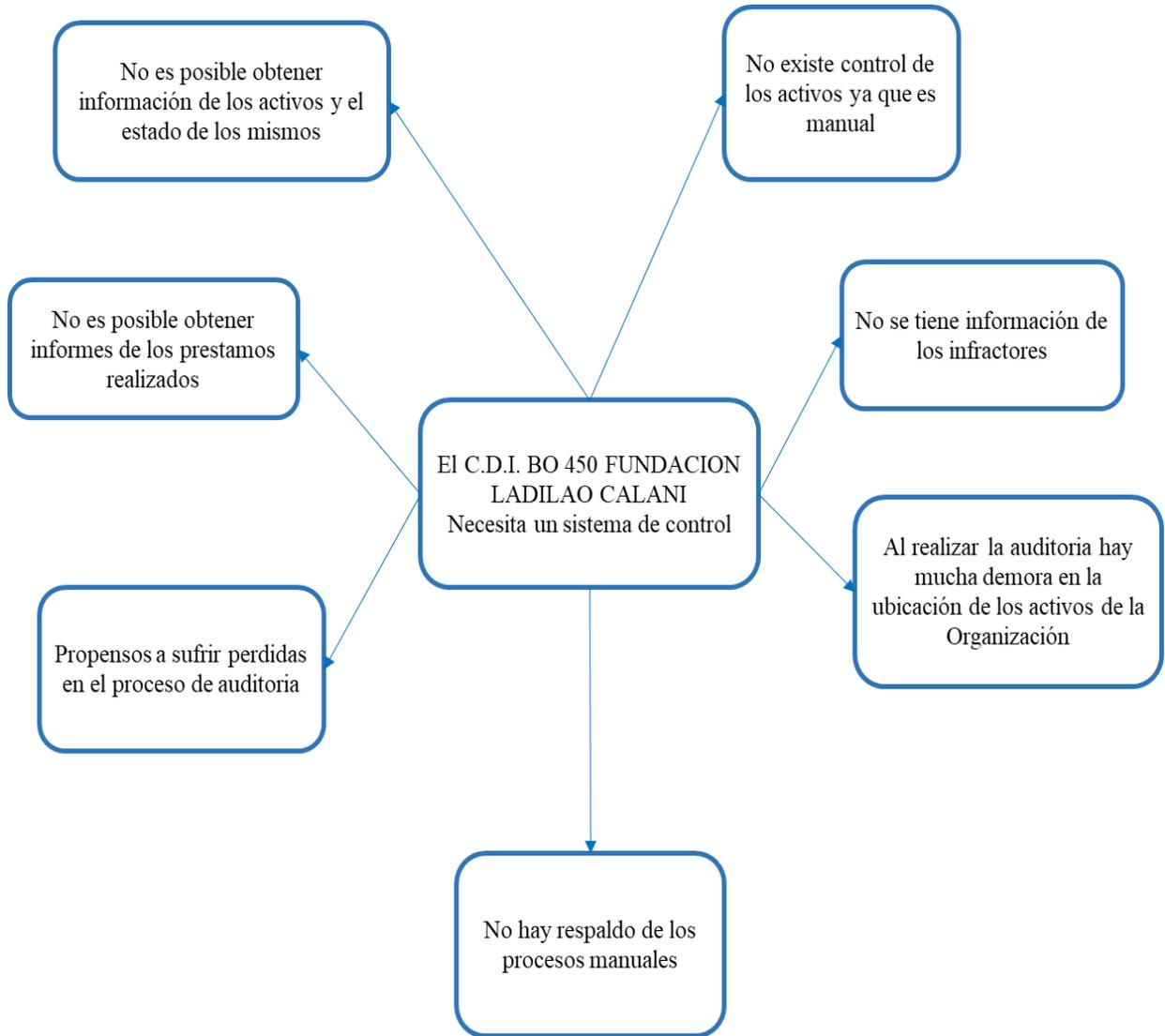


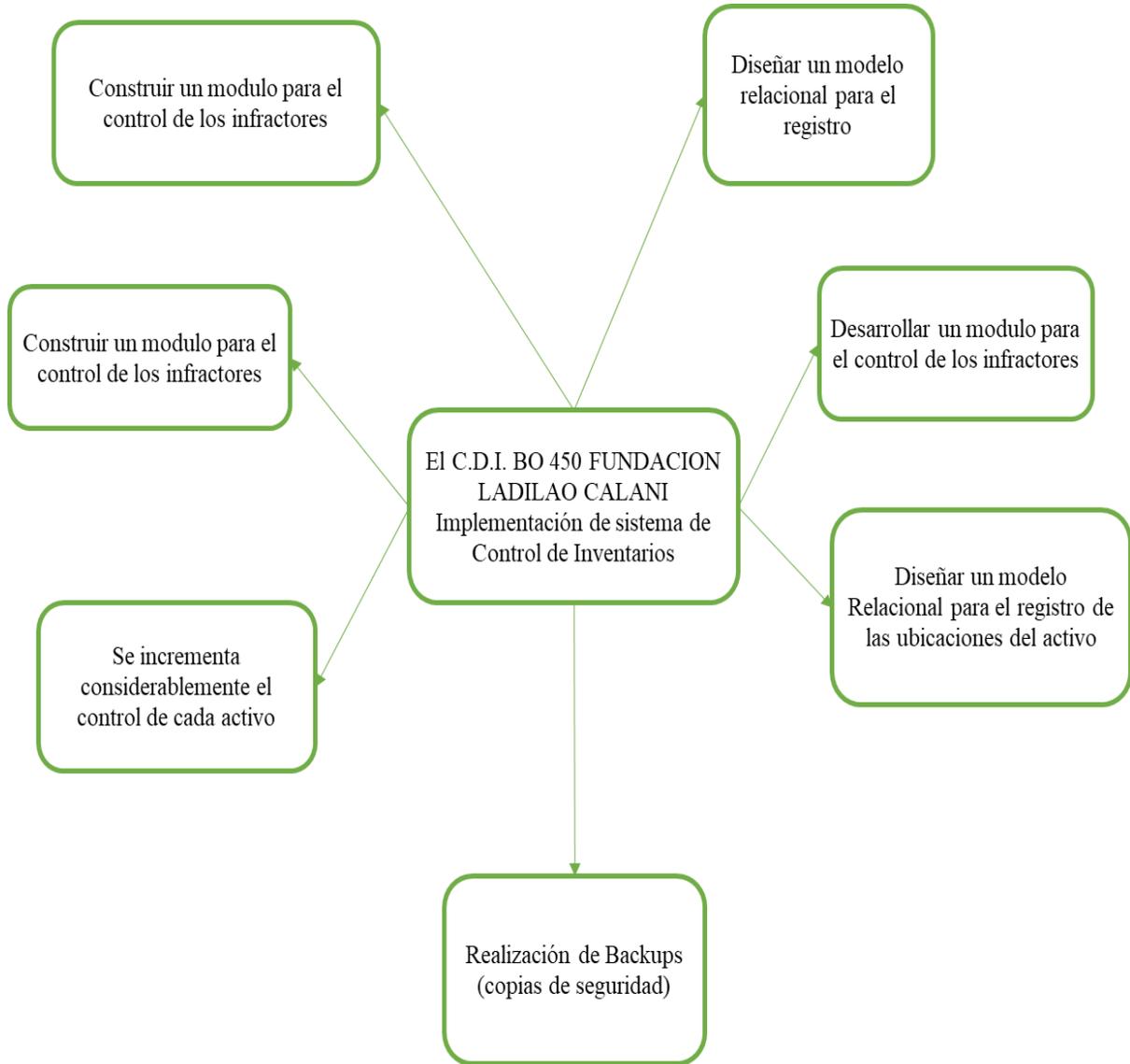
f) **Paso 6:** La aplicación “AppInventario” está lista para usarse.



g) **Paso 7:** No olvide Tener Conexión a internet para la interacción con la aplicación.  
Mínimo un ancho de banda de 100 MB/segundo.

**ARBOL DE PROBLEMAS**



**ARBOL DE OBJETIVOS**

## CUESTIONARIO

Para determinar las métricas de calidad del sistema/Aplicación “AppInventario”

**Nombre:** \_\_\_\_\_ **Fecha:** \_\_\_\_/\_\_\_\_/\_\_\_\_/

METRICAS	Calificación 0-10
Compleitud	
Trazabilidad	
Consistencia	
Simplicidad	
Exactitud	
Tolerancia a errores	
Integridad de datos	
Facilidad de auditoria	
Control de acceso	
Eficiencia de almacenamiento	
Eficiencia de ejecución	
Tolerabilidad	
Familiarización	
Comunicabilidad	
Volumen y tasa de E/S	
Estandarización de datos	
Concisión	
Instrumentación	
Simplicidad	
Modularidad	
Auto-Descriptiva	
Capacidad de ampliación	
Generalidad	
Independencia de hardware	
Independencia de sistema	
Normalización de comunicación	

Ciudad de El Alto, 22 de Noviembre de 2020

Señor:  
**Ing. Carlos David Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**  
Presente.-

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“APLICACIÓN MOVIL DE CONTROL DE INVENTARIOS MEDIANTE CODIGO QR, CASO: CENTRO DE DESARROLLO INTEGRAL BO0450 FUNDACION LADISLAO CALANI”**, elaborado por el universitario: **ROGERD PATTY QUISPE**, con cedula de identidad: **9952172 L.P.** y registro universitario: **12000678**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



.....  
Ing. Marisol Arguedas Balladares  
TUTOR METODOLÓGICO

Ciudad de El Alto, 20 de Noviembre de 2020

Señor:

**Ing. Carlos David Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**  
Presente.-

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“APLICACIÓN MOVIL DE CONTROL DE INVENTARIOS MEDIANTE CODIGO QR, CASO: CENTRO DE DESARROLLO INTEGRAL BO0450 FUNDACION LADISLAO CALANI”**, elaborado por el universitario: **ROGERD PATTY QUISPE**, con cedula de identidad: **9952172 L.P.** y registro universitario: **12000678**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



.....  
Ing. Ramiro Kantuta Limachi  
**TUTOR REVISOR**

Ciudad de El Alto, 20 de Noviembre de 2020

Señor:

**Ing. Carlos David Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**  
Presente.-

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **"APLICACIÓN MOVIL DE CONTROL DE INVENTARIOS MEDIANTE CODIGO QR, CASO: CENTRO DE DESARROLLO INTEGRAL BO0450 FUNDACION LADISLAO CALANI"**, elaborado por el universitario: **ROGERD PATTY QUISPE**, con cedula de identidad: **9952172 L.P.** y registro universitario: **12000678**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.

  
.....  
Lic. Fredy Alanoca Coareti  
TUTOR ESPECIALISTA

-



Las Asambleas de Dios de Bolivia  
P.J. 101452/61 R.S. 215651/95  
**Ministerio Evangelico "BETHEL"**  
**Centro de Desarrollo Integral**  
**C. D. I. BO - 450**



Ciudad de El Alto, 23 de noviembre de 2020

*La suscrita DIRECTORA DE FUNDACION, en uso de sus atribuciones.*

**CERTIFICA:**

Que, siendo auspiciantes del proyecto de grado del Sr. **ROGERD PATTY QUISPE**, con Cedula de Identidad: **9952172 L.P.** y Registro Universitario: **12000678**, quien desarrolló su trabajo con el tema **"APLICACIÓN MOVIL DE CONTROL DE INVENTARIOS MEDIANTE CODIGO QR, CASO: CENTRO DE DESARROLLO INTEGRAL BO0450 FUNDACION LADISLAO CALANI"**, es grato informar que el proceso de implementación de la aplicación ha superado satisfactoriamente, asimismo, las pruebas técnicas y los requerimientos de la institución, por lo que extendiendo el presente documento como constancia de aceptación y finalización del proyecto.

Autorizo al Sr. **ROGERD PATTY QUISPE**, puede hacer uso de este documento para los fines pertinentes en la Universidad Pública de El Alto.

Atentamente.



*[Handwritten Signature]*  
P. Daniela Calape Toroma  
DIRECTORA BO-450  
FUNDACION "LADISLAO CALANI"

