



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**TEMA:**

**MIGRACIÓN DE SISTEMA “CONTROL DE BIENES” DE  
ORACLE FORMS A UNA ARQUITECTURA (JSF) JAVA PARA LA  
EMPRESA MUNICIPAL DE MOVILIDAD Y OBRAS PÚBLICAS  
(EMMOP-Q)**

**AUTOR:**

**EDISON STALIN PUMASUNTA CHUNGANDRO**

**TUTOR:**

**INGENIERO. MARIO RUBEN PÉREZ CARGUA**

**QUITO, ECUADOR**

**AÑO:2020**

## DECLARACIÓN DE AUTORÍA

El documento de tesis con título: “MIGRACIÓN DE SISTEMA “CONTROL DE BIENES” DE ORACLE FORMS A UNA ARQUITECTURA (JSF) JAVA PARA LA EMPRESA MUNICIPAL DE MOVILIDAD Y OBRAS PÚBLICAS (EMMOP-Q)”, ha sido desarrollado por el señor Edison Stalin Pumasunta Chungandro con C.C. No. 1719599464 persona que posee los derechos de autoría y responsabilidad, restringiéndose la copia o utilización de la información de esta tesis sin previa autorización.



---

Edison Stalin Pumasunta Chungandro

# UNIVERSIDAD TECNOLÓGICA ISRAEL

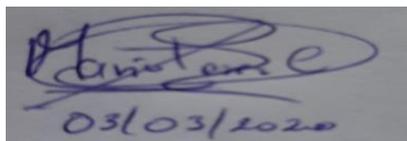
## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación certifico:

Que el trabajo de titulación “MIGRACIÓN DE SISTEMA “CONTROL DE BIENES” DE ORACLE FORMS A UNA ARQUITECTURA (JSF) JAVA PARA LA EMPRESA MUNICIPAL DE MOVILIDAD Y OBRAS PÚBLICAS (EMMOP-Q)”, presentado por Edison Stalin Pumasunta Chungandro, estudiante de la Carrera Ingeniería en Sistemas Informáticos, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., 06 de marzo del 2020

TUTOR



03/03/2020

-----  
Ing. Mario Pérez

## **AGRADECIMIENTOS**

Agradezco en primer lugar a Dios por permitirme tener vida y salud, a mis padres que son los pilares fundamentales en mi educación, también mis agradecimientos a cada uno de los docentes en mis años de formación académica, no me quedo sin mencionar la gran oportunidad que la universidad Israel me otorgo al permitirme encaminar en la vida de estudiante basado en el área Ingeniera en Sistemas Informáticos desarrollándome en el ámbito laboral paso a paso forjando mis conocimientos de especialidad y aprendiendo día a día por medio de nuevas tendencias a la nueva era tecnológica.

Agradecemos a Dios por bendecirnos la vida, por guiarnos a lo largo de nuestra existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad.

Gracias a mis padres: Segundo Pumasunta y Pilar Chungandro; por ser los principales promotores en perseguir mis sueños, por confiar y creer en mis expectativas, por los consejos, valores y principios que me han inculcado.

Agradecer a los docentes de Ingeniera en Sistemas Informáticos de la Universidad Israel, por haber compartido sus conocimientos a lo largo de la preparación de nuestra profesión, de manera especial, al ing. Mario Pérez tutor de mi proyecto de integrador quien ha guiado con su paciencia, y su rectitud como docente, y a los funcionarios del área DTI - EPMMOP por su valioso aporte para realizar el proyecto.

## **DEDICATORIA**

El presente trabajo lo dedico principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en una persona humilde y de principios. Ha sido un orgullo y un privilegio el ser su hijo, son los mejores padres.

A mis hijos por estar presentes en mi pensamiento y por ser un apoyo moral eterno de lucha por ustedes, brindándome su sonrisa transmitiéndome un don de entrega a mis deseos de superación.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos

## TABLA DE CONTENIDOS

RESUMEN .....	x
ABSTRACT.....	xi
INTRODUCCIÓN .....	1
Antecedentes de la situación objeto de estudio.....	1
Planteamiento del problema.....	1
Justificación .....	2
Objetivos.....	2
General.....	2
Objetivos específicos.....	2
Descripción de los capítulos .....	3
1    CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
1.1    Estado del arte .....	5
1.2    Lógica del negocio .....	113
1.3    Herramientas técnicas (componentes técnicos con los que se desarrolla).....	114
1.4    Alternativas de solución.....	19
2    CAPÍTULO 2. MARCO METODOLÓGICO.....	20
2.1    Tipo de investigación.....	20
2.1.1    Metodología seleccionada.....	20
2.2    Recopilación de información .....	21
2.2.1    Técnicas de recopilación de información.....	21
3    CAPÍTULO 3. PROPUESTA.....	26
3.1    Diagramas de procesos .....	26
3.2    Factibilidad técnica.....	28
3.3    Factibilidad operacional.....	28

3.4	Factibilidad económica-financiera .....	28
3.5	Especificación de requerimientos.....	28
3.5.1	Ámbito del software .....	29
3.5.2	Funciones del producto .....	29
3.5.3	Características de los usuarios del sistema .....	33
3.5.4	Restricciones de desarrollo .....	34
3.5.5	Requisitos.....	35
4	CAPÍTULO 4. IMPLEMENTACIÓN .....	38
4.1	Diseño .....	38
4.2	Esquema de la base de datos .....	38
4.3	Diagrama de la arquitectura del sistema .....	38
4.4	Diseño de interfaces.....	39
4.5	Estándares de programación utilizados.....	39
4.6	Implementación .....	41
4.7	Pruebas .....	55
4.7.1	Pruebas de funcionalidad (Aceptación de usuario) .....	55
4.7.2	Pruebas de rendimiento (Aceptación técnica).....	63
4.7.3	Pruebas de carga y estrés (Aceptación técnica).....	64
4.7.4	Plan de implementación.....	65
4.7.5	Requerimientos de implementación .....	66
4.7.6	Manual de usuario .....	66
4.7.7	Manual técnico .....	66
4.7.8	Plan de capacitación .....	67
	CONCLUSIONES .....	68
	RECOMENDACIONES .....	69
	REFERENCIAS BIBLIOGRÁFICAS .....	70

ANEXOS .....	70
Esquema de base de datos .....	71
Validación y verificación metodológica XP.....	75
Manual de Usuario.....	79
Manual Técnico .....	83

## LISTA DE FIGURAS

<i>Figura 1.</i> Estructura de Spring Modelo-Vista-Controlador. Tomado de F curso Learn Spring, centrado en los fundamentos de Spring 5 y Spring Boot. ....	17
<i>Figura 2.</i> Diagramas de Procesos (limitados tecnológicamente) <b>Error! Marcador no definido.</b>	
<i>Figura 3.</i> Diagrama de procesos(Automatizado).....	27
<i>Figura 4.</i> Diagrama de la arquitectura del sistema Custodia Bienes. Tomado de: <a href="https://www.codejava.net/frameworks/spring/how-to-enable-spring-mvc-in-webxml...">https://www.codejava.net/frameworks/spring/how-to-enable-spring-mvc-in-webxml...</a>	38
<i>Figura 5.</i> Estructura general de la metodología MADIISH. Tomado de: Silva, D. T., Galván, J. D. O., Gómez, H. A., & López, R. R. Una Propuesta de Metodología para la Migración de Sistemas Heredados .....	42

## LISTA DE TABLAS

Tabla 1. <i>Perfiles de usuario</i> .....	34
Tabla 2. <i>Roles Metodología XP</i> .....	3; <b>Error! Marcador no definido.</b>
Tabla 3. <i>Guía de Migración</i> .....	43
Tabla 4. <i>Descripción de las hojas de la presente guía...</i> <b>Error! Marcador no definido.</b>	
Tabla 5. <i>Datos generales del proyecto</i> .....	45
Tabla 6. <i>Datos sistema/software a migrar</i> .....	45
Tabla 7. <i>Madiish fase C</i> .....	46
Tabla 8. <i>Madiish fase 1</i> .....	47
Tabla 9. <i>Madiish fase 2</i> .....	48
Tabla 10. <i>Madiish fase 3</i> .....	48
Tabla 11. <i>Madiish fase 5</i> .....	49
Tabla 12. <i>Madiish fase T</i> .....	50

## **RESUMEN**

En el hoy día de los negocios hace que las organizaciones empresariales requieran sistemas de información que se adapten al cambio mediante su necesidad. Por lo tanto, el sistema Control de Bienes se convierten en un requisito indispensable para la organización, necesidades que se ven reflejadas en elaborar estrategias para aprovechar estos activos con su inherente valor organizacional. Bajo esta premisa, el siguiente trabajo tiene por objetivo facilitar un conjunto de pautas para la organización que involucran planes de migración de aplicaciones, especialmente las desarrolladas bajo la plataforma Oracle Forms 6i hacia una arquitectura Spring MVC que permita acoplar la lógica de negocio, presentación, acceso a datos y aprovechando las potencialidades del Framework Eclipse ID abierto de Desarrollo de Aplicaciones.

El presente trabajo de migración se convierte, en una herramienta de orientación para la organización que desean involucrar la reingeniería en sus proyectos de evolución de aplicaciones tecnológicas actuales, como es cliente-servidor enfocado a las arquitecturas multicapa.

### **PALABRAS CLAVES**

migración de aplicaciones, Oracle Forms, Spring MVC, Eclipse ID, arquitecturas multicapa.

## **ABSTRACT**

In business today, business organizations require information systems that adapt to change through their needs. Therefore, the Asset Control system becomes an indispensable requirement for the organization, needs that are reflected in developing strategies to take advantage of these assets with their inherent organizational value. Under this premise, the following work aims to provide a set of guidelines for the organization that involve application migration plans, especially those developed under the Oracle Forms 6i platform towards a Spring MVC architecture that allows coupling business logic, presentation, data access and taking advantage of the potential of the Open Eclipse ID Framework for Application Development.

This migration work becomes an orientation tool for the organization that wants to involve reengineering in their projects for the evolution of current technological applications, such as a client-server focused on multilayer architectures.

## **KEYWORDS**

application migration, Oracle Forms, Spring MVC, Eclipse ID, multilayer architectures.

## **INTRODUCCIÓN**

### **Antecedentes de la situación objeto de estudio**

La Empresa Pública Metropolitana de Movilidad y Obras Públicas (EPMMOP), según la Ordenanza Metropolitana No. 309 emitida el 16 de abril del 2010, en el ámbito de sus competencias gestiona y administra toda la infraestructura de vías y espacios públicos en el Distrito Metropolitano de Quito.

Con fecha 28 de febrero de 1994 y según Ordenanza No.3074 se constituyó la Empresa Metropolitana de Obras Públicas (EMOP-Q).

En abril de 2008 la EMOP-Q se unifica con la Empresa Metropolitana de Servicios y Administración de Transporte (EMSAT), y se crea la Empresa Municipal de Movilidad y Obras Públicas (EMMOP-Q) según Ordenanza No.251.

Con fecha 16 de abril de 2010 y según Ordenanza No.309; se crea la Empresa Pública Metropolitana de Movilidad y Obras Públicas (EPMMOP), conforme a la Ley Orgánica de Empresas Públicas.

La Empresa Pública Metropolitana de Movilidad y Obras Publicas cuenta con aplicativos propios limitados tecnológicamente, por lo cual la migración tiene por finalidad, el traslado a una nueva plataforma, conservando su funcionalidad y datos originales.

### **Planteamiento del problema**

En él ahora la Empresa Pública Metropolitana de Movilidad y Obras Públicas (EPMMOP), cuenta con un sistema “Control de Bienes” en un framework Oracle Forms ambiguo que se dispone en la Unidad Administrativa, debido al gran desarrollo institucional y crecimiento en sus diferentes ámbitos se implementó dependencias distribuidos en todo Quito, manejando el control de bienes mediante una forma manual o

a espera de personal autorizado, por la no existencia y falta de soporte técnico en algunas aplicaciones corporativas y bases de datos que posee (EMMOP-Q).

## **Justificación**

Con el uso de las tecnologías de información apropiadas al desarrollo de software se propone alcanzar una migración de sistemas Informáticos que traerá mejora en procesos internos de la Empresa Municipal de Movilidad y Obras Públicas (EMMOP-Q).

Mediante la actualización a una nueva tecnología de mejora, su modelo arquitectónico que se plantea puede resolver varios puntos de estrategias de negocio logrando añadir procesos para una integración de nuevos módulos enfocados hacia un futuro mediante estrategias y de acuerdo con la necesidad de la institución.

Los módulos en el proceso de migración llevarían un trabajo de reingeniería el cual otorga una flexibilidad de una arquitectura resultante, basado en un mantenimiento el cual el sistema podrá ser simplificado o añadiendo nuevas fusiones mediante la coordinación ordenada de actividades.

## **Objetivos**

### **General**

Ejecutar el desarrollo de la migración de sistema “Control de Bienes” de la plataforma Oracle Forms hacia una arquitectura multicapa Java que permitirá optimizar procesos en distintas dependencias manteniendo un nivel de desarrollo tecnológico con facilidades de almacenamiento, control y existencia de los bienes.

### **Objetivos específicos**

- Análisis de proceso actual.
- Utilizar una metodología de migración, de desarrollo para el módulo núcleo proyecto Control de Bienes.
- Creación de Arquitectura java.

- Migrar la lógica de negocio a nivel de aplicación para hacer independiente de la BDD.

## **Descripción de los capítulos**

### **Capítulo 1.**

En el presente y hacia un futuro encaminado que toda organización empresarial debe surgir de acuerdo a la nueva era tecnológica conjunto con las necesidades, para adaptarse e incluirse como una empresa de competencia con el fin de satisfacer las necesidades humanas.

Para ello el entorno basado en el Plan Estratégico de Tecnologías de la Información deben evolucionar hacia nuevos requerimientos, esto acarrea hacia una reingeniería de aplicaciones corporativas, tomando en cuenta desde su metodología de migración integrando dentro de su ámbito fases de metodología de investigación pasando a una metodología de desarrollo de programación ágil.

Sin dejar atrás la necesidad de herramientas tecnológicas de desarrollo de software como es IDE eclipse luna, integrando un acceso de conexión de BDD en una arquitectura modelo / vista / controlador.

### **Capítulo 2.**

Para la implementación de la migración se llevó una metodología de investigación tipo inductivo que es una forma relativamente flexible de copilar y analizar información con la exploración de la misma, mediante varias fuentes para obtener una generalización de la veracidad de información contando con la necesidad de los usuarios por mantener un Sistema “ Control de Bienes” en todas las dependencias, para lograr la generalización de información aplicando el método de investigación al sistema se toma una parte de la población de la EPMMOP, realizando una encuesta con el cual se parte de hechos particulares hacia una nueva expectativa en base al control en todas las dependencias.

### **Capítulo 3.**

Basado en la reingeniería del sistema Control de Bienes hacia una migración con la cual permitirá optimizar tiempos y costos que beneficiarán a la institución de la misma manera el poder obtener soporte del personal capacitado con funciones del producto basado en la lógica del negocio y plasmado en el área Administrativa encargada del manejo de todos los bienes empresariales con los que cuenta en distintas dependencias verificando su registro de estado y novedades del mismo.

Esto lo podremos lograr mediante el servicio en el portal interno de la empresa, así manejar una misma información ordenada y sin pérdida de documentación será un sistema independiente de control el cual tendrá una operación con procedimientos propuestos.

#### **Capítulo 4.**

Manejados por un estándar de programación en el área de sistemas corporativos se manejará por un modelo de interfaz gráfica, con una arquitectura de desarrollo Spring MVC donde se puede obtener beneficios y reestructuras de acuerdo a la forma y necesidades del usuario, todo esto con el fin de conseguir la migración al sistema “Control de Bienes” JSF utilizando metodologías como de migración dentro con una investigación y desarrollo ágil mencionada anteriormente con procedimientos propuesto y fases en la cual incluyen la fase de implementación.

# CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

## 1.1 Estado del arte

Sistemas Heredados (SH) son aplicaciones informáticas que tienen su periodo de funcionamiento dentro de las organizaciones empresariales, que abarcan una tendencia de desarrollo conjunto con la empresa y que con el paso del tiempo a pesar de su desarrollo la aplicación llega al final donde la organización toma la decisión de mantenerse o evolucionar de plataforma mediante una migración dirigida a nuevas tecnologías.

Causas de migración:

- La no existencia de soporte técnico para la herramienta de desarrollo de algunas aplicaciones actuales en producción.
- Normativas vigentes.
- Costos de mantenimiento.
- Imposibilidad de crecimiento y estabilidad tecnológica.

A los problemas anteriores se adiciona en muchas ocasiones el aislamiento de los orígenes de datos (si se carece de un gestor de datos relacional), lo cual puede provocar inconsistencias en los resultados de búsquedas, redundancia y deficiencia en la confiabilidad de los datos. Durante la migración de un sistema heredado, frecuentemente no se cuenta con manuales o esquemas de la arquitectura del sistema, o algún modelo que permita visualizar el flujo de los procesos para comprender el negocio, lo que dificulta el proceso de migración, recurriendo a periodos de análisis donde se deben definir los procesos y diseñar los esquemas de datos. Durante este proceso es necesaria la elaboración de módulos interoperables entre el nuevo sistema y el sistema heredado para permitir el flujo de datos mientras se realiza la migración.

Las organizaciones de TI están bajo creciente presión para reducir costes y aumentar su capacidad de reacción a las demandas del negocio, por lo que los sistemas se convierten en un problema para las empresas debido a que, en la mayoría de los casos,

su mantenimiento es costoso además de la falta de respuesta oportuna a las necesidades actuales de las empresas.

## **METODOLOGÍAS DE SOFTWARE**

Una Metodología de desarrollo de software es el proceso que se debe seguir para la solución de una aplicación informática, para lograr dicha metodología tiene que ver con varios puntos estratégicos como son: comunicación (intercambio de información), manipulación de modelos enfocados a una arquitectura con el fin de hacerlo más predecible y eficiente.

Partiendo desde lo planeado en las primeras fases hasta obtener el propósito con un proceso disciplinado de desarrollo adaptando características de acuerdo a cada proyecto, dicho en que consiste la metodología cuenta con dos grupos ágiles y robustas.

### **Metodologías Robustas**

Se basan en una planificación con su respectiva documentación exhaustiva en cumplimiento de todo el proceso desde su inicio hasta obtener el resultado final del proyecto, entre algunas de estas metodologías robustas se encuentran:

- Microsoft Solution Framework (MSF).
- Métrica 3.
- Proceso Unificado de Desarrollo (RUP).

Entre estas metodologías señaladas cuenta con ítems a seguir para su desarrollo:

1. Análisis de los requisitos: Consiste en documentar lo que el software deberá realizar al término de su migración.
2. Diseñar el sistema y programa: Es la realización de un prototipo y los algoritmos a utilizar sin codificar.
3. Codificar: Se realiza la escritura del código necesario para el desarrollo del software.

4. Ejecución de pruebas: Para valorar y localizar posibles errores o validaciones que no hayan sido consideradas. De igual forma permitirá conocer los tiempos de ejecución y la veracidad de los resultados obtenidos.
5. Verificación: Una vez que fue probado por el programador, se instalará para que el usuario realice pruebas reales.
6. Mantenimiento del sistema nuevo: Normalmente no se prueban todos los posibles casos por lo que siempre habrá que corregir errores y realizar actualizaciones.
7. Amplia documentación en todo momento.

### **Metodologías Ágiles**

Se basan en una comunicación estrecha en los participantes y se orientan en el resultado del producto y no en la documentación, proponiendo cambios de último momento entre estas metodologías:

- Extreme Programing (XP).
- Scrum.
- Crystal Metodología.

En cambio, las metodologías ágiles consisten principalmente en:

1. Trabajar para obtener software funcional en lugar de demasiada documentación.
2. Colaboración con el usuario para la comprensión rápida de sus procesos de negocios.
3. Se tiene la posibilidad de hacer cambios de planes en cualquier punto del proyecto evitando la planeación extensa, lo que permite iniciar la programación.

Sin embargo, aunque sus ventajas son interesantes, estas metodologías también presentan inconvenientes que hay que asumir cuando se decide trabajar con ellas. Estas son:

1. Falta de documentación del diseño.

2. Problemas de comunicación debido a la participación del usuario donde se pueden dar malas interpretaciones en lo hablado y no documentado.
3. Existe una fuerte dependencia de las personas, si el usuario no tiene el tiempo suficiente disponible puede alentar el desarrollo del proyecto.
4. Falta de reusabilidad derivada de la falta de documentación.

Estos inconvenientes son contemplados y existe un enfoque especial a los mismos dentro de la metodología de migración **MADIISH**.

## **METODOLOGÍAS PARA LA MIGRACIÓN DE SISTEMAS HEREDADOS**

Actualmente la tecnología ha ido evolucionando y a para ello se han diseñado varias metodologías de migración de aplicaciones que se basa en el sustento del plan y fases para el desarrollo de todo el proyecto.

### **Tipos de metodologías de migración de SH:**

- Migración de Sistemas Heredados a la nube informática.
- Migración de Sistemas heredados fundamentada en el uso de herramientas de KDD (knowledge discovery in databases).
- Migración de Sistemas Heredados Madiish.

### **Migración de Sistemas Heredados a la nube informática.**

Se refiere a la externalización (virtualización) en el cual se acuerdan pactos de servicios a terceros, se enfoca directamente en sustitución de componentes, migración dividida a la aplicación por sectores en módulos, migración íntegra, y convertir la aplicación en la nube informática, intervinientes durante el proceso de traslado el consumidor, proveedor y desarrollador del servicio.

Esto se realiza mediante la transformación de aplicaciones tradicionales basadas en formularios, basada en una arquitectura orientada a servicios. Para la mecánica de la

metodología consiste en recursos físicos y virtuales para optimizar al máximo la utilización.

“Actualmente, antes de migrar a la nube informática, se transforma primero las aplicaciones tradicionales a aplicaciones basadas en SOA, ya que esta arquitectura es flexible a la composición de servicios. SOA encapsula la lógica del negocio a través de servicios web (WSs) que se comunican entre sí por medio de métodos y protocolos estándares de intercambio de mensaje, que facilita la composición de servicios” (Zalazar, 2014, pág. 71).

### **Migración a la nube informática Técnicas:**

- **Substitución de la aplicación.**

Reemplaza parte de la aplicación por algún estándar en el mercado en este proceso puede haber pérdida de control de información, creación de interfaz con otros componentes.

- **Conversión de la aplicación.**

### **Transforma aplicaciones a través de un motor conversión y mapeo de datos.**

#### **Tipo de conversión:**

Conversión de aplicaciones (programas): es la transformación desarrollada en base algoritmos y software (aplicaciones), modificando los aspectos de programación conservando su funcionalidad y estructura.

Conversión de datos: es la transformación de datos según el esquema hacia un formato destino.

Conversión de esquemas: convierte una estructura de BDD a una nueva estructura de BDD equivalente.

- **Virtualización de la aplicación (programa).**

“Consiste básicamente en generar un contenedor virtual de la aplicación y trasladar todos sus componentes sin modificar los códigos fuente y el esquema de datos. Es decir, se considera al sistema heredado como una caja negra, donde se encapsulan todas sus capas, y durante la interacción del usuario con el sistema se analizan las entradas y salidas, para generar un *wrapper* (empaquetador) que pueda ser el nexo el sistema heredado encapsulado y una nueva capa de presentación desplegada en la nube informática” (Zalazar, 2014, pág. 72).

- **Generación de una nueva aplicación(Programa).**

“Se trata de programar una nueva aplicación compatible con los entornos de nube informática. Esta solución deberá realizar una reingeniería de la aplicación para obtener la definición de sus procesos y funciones, y generar una nueva aplicación para la nube informática. Como parte de esta reingeniería se deberá analizar de aquellas funcionalidades no utilizadas, redundantes, caducadas, con necesidades de mejoras, y que pueden ser consolidadas en otra funcionalidad” (Zalazar, 2014, pág. 72).

### **Migración de Sistemas heredados fundamentada en el uso de herramientas de KDD (knowledge discovery in databases).**

En la actualidad se han diseñado varias metodologías para la migración de un Sistema Heredado (SH), tal es el caso de la “metodología de apoyo basada en el uso de las herramientas KDD (Knowledge Discovery in Databases) donde se señala que uno de los factores de éxito de un proyecto de migración es el entendimiento del SH, esto es, entender tanto el modelo de datos como el modelo de negocios que trata de cubrir el mismo. Mediante esta estrategia se pretende reconstruir algunos aspectos básicos del SH a migrar, de modo que sea posible entender el modelo de datos del SH, entender el modelo de negocios que intentaba cubrir el SH y determinar el nivel de calidad de los datos del SH” (Silva, 2018, pág. 42).

Esta metodología propone afirmar la reparación de requisitos del SH, apoyada en herramientas de minería de datos; dichas herramientas basadas en un esquema a lo cual se identifica la estructura y el objetivo de la realización de la minería de datos, la cual pretende “excavar” entre los datos para hallar información encubierta y que eventualmente sea de gran beneficio en la toma de decisiones. La técnica de la minería de datos se basa principalmente en los siguientes incisos:

- Clasificación de datos.
- Preprocesamiento o enriquecimiento de datos.
- Evolución o codificación de datos.
- Extracción del conocimiento (minería de datos).
- Evaluación de resultados.

La propuesta menciona la calidad de la investigación de datos en el desarrollo de la metodología basada en herramientas KDD tiene como fundamento principal que la información continuamente estará presente y esa es la diferencia que ha sido utilizada en dicha exploración, por lo que se menciona en la propuesta misma: si se posee los datos, serán la fuente principal sobre el SH. Por lo que la clave es el cómo se puede obtener el conocimiento del SH que supone está supuesto en datos, “cuyo objetivo concurre, justamente, encontrar conocimiento oculto entre los datos. Las etapas que comprende la propuesta de la metodología anterior son:

- Etapa 1: Obtención de los datos del SH y la documentación asociada a ellos.
- Etapa 2: Selección de una herramienta de KDD.
- Etapa 3: Compatibilización del formato de los archivos o tablas con las entradas para la herramienta KDD y selección de atributos.
- Etapa 4: Análisis de los datos utilizando funciones de preprocesamiento.
- Etapa 5: Minado de los datos.
- Etapa 6: Chequeo de los resultados con el usuario.
- Etapa 7: Reconstrucción del modelo de datos del SH” (Silva, 2018, pág. 42).

Dichas etapas generan salida de información que determinará el proceso de la etapa inmediata; a su vez, continuamente se evalúan los resultados para verificar su integridad y que exista confiabilidad en los mismos.

Por otro lado, como parte de los trabajos de migración de sistemas heredados y las nuevas tendencias de desarrollo de software orientado a arquitecturas basadas en la nube informática, se propuso un esquema particularmente para la migración de sistemas hacia la nube. El autor menciona que, “en el flujo de trabajo propuesto, no se considera el análisis de seguridad como una tarea independiente, sino integrada a cada una de las tareas involucradas a la migración. Es por esto por lo que el consumidor tiene la responsabilidad de alinear cada actividad de la migración a sus políticas de seguridad y asegurarse que el contrato del proveedor abarque estas políticas” (Zalazar, 2014, pág. 73).

## **METODOLOGÍA MADIISH**

En esta metodología por naturaleza tiene una forma incremental e iterativa en cada una de sus iteraciones en cada una de sus fases. En cuanto a procesos ágiles por su organización y entrega del trabajo continuamente, mediante una investigación del sistema heredado se puede contar con la obtención de información y definiciones de módulos que integran al SH.

Mediante la fase de esta metodología se puede realizar un análisis de acuerdo a la selección de módulos principales en los que abarcan información categorizada, planteando requerimientos y mejoras comprendiendo funciones del negocio

La metodología MADIISH está basada en las principales características de las siguientes metodologías:

- Metodología iterativa debido a que se pueden realizar modificaciones constantes sobre un mismo subsistema hasta la entrega satisfactoria al usuario final.
- Metodología incremental ya que se migra subsistema por subsistema.

- Metodología ágil por la participación constante del cliente para reducir la documentación.
- La metodología MADIISH presenta una estructura propia de migración a nuevas tecnologías.

Luego de analizar las metodologías de migración enfocadas en una nueva arquitectura que sea de respuesta rápida y eficiente durante el proceso de desarrollo del proyecto, cada una contando con su respectiva información de trabajo y desarrollo, así como metodología de investigación en base la recolección y minería de información, siendo parte de una metodología de migración de software información que proviene de un antecedente del Sistema Heredado y sus estructuras de migración.

Para el presente proyecto se procede a seleccionar la metodología de migración MADIISH que abarca en gran ámbito todos los procesos de trabajo a desarrollar contando con un plan guía enfocado en etapas contempladas en la propuesta.

## **1.2 Lógica del negocio**

La empresa EPMMOP realiza procedimiento de Control de Bienes que dispone actualmente en un sistema de Oracle Forms que se dispone solo en matriz ya que en tiempo anteriores se disponía de bienes en cuanto a una dirección, al transcurrir el tiempo la empresa ha desarrollado varias áreas estratégicas de trabajo viendo la necesidad de adquisición de bienes muebles de trabajo del Distrito Metropolitano de Quito, a su vez el incremento de empleados para su desarrollo en el ámbito laboral debe ser verificado con el estado del bien entregado (Custodio/Persona).

En las distintas dependencias anexas se puede llevar el control de bienes y estado de los mismo mediante documentación a verificar en sistema custodio de Oracle Forms con el cual se ingresa la información de distintas gerencias, áreas, direcciones y lugares de trabajo.

La empresa EPMMOP-Q cuenta con un portal interno que puede manejar en repositorio distintos sistemas para todos los lugares de trabajo, desea llevar a cabo un plan

de migración de distintos sistemas de control y módulos de información que se manejan actualmente mediante Oracle Forms a nivel matriz, a su vez poseer en el portal interno(WEB) para disposición de todas las dependencias de trabajo y mantener una información unificada y sin traspapeles.

### **1.3 Herramientas técnicas**

#### **IDE Eclipse (java)**

Es un entorno de desarrollo integrado, de una generación de alto nivel de “Código abierto y Multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como Aplicaciones de Cliente Enriquecido, entorno de desarrollo integrado, opuesto a las aplicaciones Cliente-liviano, entorno de desarrollo integrado basadas en navegadores.

Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar” (Eclipse, entorno de desarrollo integrado, s.f.).

Desde que Sun Microsystems introdujo Java en 1995, los proponentes han buscado formas de mejorar la suerte de la tecnología. Un enfoque es crear un entorno de desarrollo integrado que facilite el trabajo con Java. Los partidarios esperaban que un IDE hiciera que Java fuera más competitivo con el popular Visual Studio .NET de Microsoft, que proporciona un entorno para herramientas de software integradas y fáciles de usar que atraen a los muchos desarrolladores de aplicaciones empresariales que no son programadores intransigentes.

Esto ha desencadenado una batalla entre varios IDE de Java, incluidos JBuilder de Borland, Visual J # de Microsoft, JDeveloper de Oracle y NetBeans de Sun. Un participante fue Eclipse, que IBM desarrolló y entregó en 2001 a la Eclipse Foundation, una organización sin necesidad de lucro, para que se administre como una plataforma de

código abierto. Además de proporcionar un IDE, Eclipse automatiza numerosas funciones que los desarrolladores de otra manera manejarían el código.

Eclipse ha obtenido tanto apoyo que muchos observadores de la industria dicen que ahora es el jugador clave de herramientas Java. Su uso es económico y hace que sea mucho más fácil integrar sus herramientas entre sí.

### **PrimeFaces**

Es uno de los frameworks web con mayor aceptación hoy día y que muchos programadores web han puesto en práctica con resultados favorables, es una librería de componentes visuales Open Source para JSF, así como IceFaces o RichFaces. Según DevRates.com (Es un Sitio Web que, mediante la opinión de los desarrolladores, crean matrices de opinión acerca de todo lo concerniente a la programación), Primefaces es el número dos en el top de los diez frameworks más populares del mundo.

“Es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web. Los componentes de PrimeFaces cuentan con soporte nativo de Ajax, pero no se encuentra implícito, de tal manera que se tiene que especificar que componentes se deben actualizar al realizar una petición proporcionando así mayor control sobre los eventos. Cuenta también con un módulo adicional TouchFaces para el desarrollo de aplicaciones web para dispositivos móviles con navegadores basados en WebKit. Las principales características de PrimeFaces son:

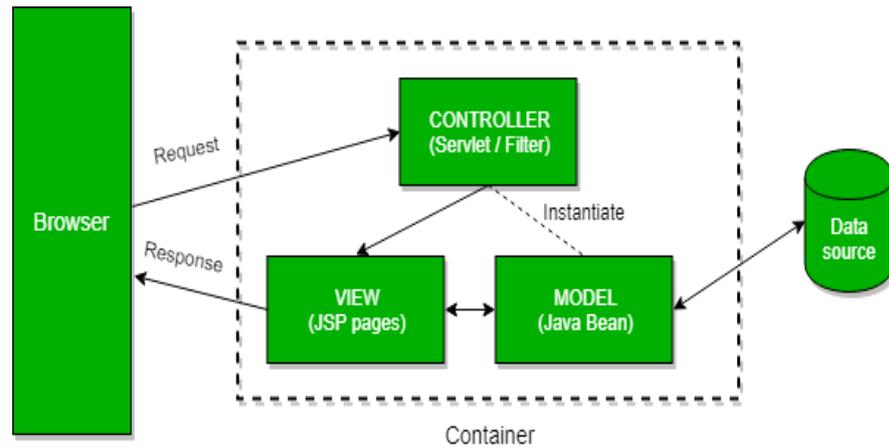
- Soporte nativo de Ajax, incluyendo Push/Coment.
- Kit para crear aplicaciones web móviles.
- Es compatible con otras librerías de componentes como Jboss RichFaces.
- Uso de JavaScript no intrusivo.
- Es un proyecto open source, activo y estable” (Pech-May, 2010).

## Spring MVC

El framework tiene un conjunto de interfaces que después se implementan para proporcionar la funcionalidad correspondiente.

Las interfaces están acopladas claramente al Servlet Api, la clase DispatcherServlet está en el front controller y es responsable de delegar y coordinar el control entre varias interfaces en la fase de ejecución durante una petición Http, “las interfaces más importantes definidas en Spring MVC y sus responsabilidades son:

- **HandlerMapping:** permite manejar peticiones de entrada.
- **HandlerAdapter:** ejecución de objetos que permiten manejar las peticiones entrantes.
- **Controller:** está entre el modelo y la vista, y permite manejar peticiones entrantes y redirigirlas a la respuesta adecuada.
- **Vista:** responsable de retornar una respuesta al cliente.
- **ViewResolver:** selecciona una vista basada en un nombre lógico de la vista.
- **HandlerInterceptor:** intercepta las peticiones entrantes, es comparable pero no igual a los filtros de Servlet.
- **LocaleResolver:** resuelve y opcionalmente salva el locale de un usuario individual.
- **MultipartResolver:** facilita trabajar con ficheros de subida wrapping peticiones de entrada” (PLANA, 2018, pág. 39).



- **Figura 1.** Estructura de Spring Modelo-Vista-Controlador. Tomado de F curso Learn Spring, centrado en los fundamentos de Spring 5 y Spring Boot.

## Oracle

“Es básicamente una herramienta cliente/servidor para la gestión de base de datos, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general” (Sistemas Manejadores de Base de Datos, 2009).

## PL/SQL

Oracle SQL Developer “es una aplicación gráfica que permite ejecutar las consultas o scripts SQL, gestionar los objetos de una base de datos (tablas, vistas, etc.)” (Torres Gonzales, 2017, pág. 10) y desarrollar y personalizar los programas PL/SQL.

PL/SQL es el lenguaje procedimental de ORACLE. Es una extensión de SQL, el cual es, por su parte, un lenguaje estándar basado en la teoría de conjuntos.

El interés del lenguaje PL/SQL está en poder mezclar la potencia de las instrucciones SQL con la flexibilidad de un lenguaje procedimental en un mismo programa.

Estos programas pueden ser ejecutados directamente por las herramientas de ORACLE (bloques anónimos) o a partir de objetos de la base de datos (procedimientos almacenados y paquetes).

Las ventajas del lenguaje PL/SQL son diversas:

- Integración de SQL: se pueden utilizar las instrucciones DML, las de control de transacciones y las funciones SQL prácticamente con la misma sintaxis.
- Funcionalidades suplementarias: la gestión de cursores y el tratamiento de errores ofrecen nuevas posibilidades de procesamiento.
- Mejora del rendimiento: se pueden agrupar varias instrucciones en una misma unidad (bloque) que sólo dará lugar a un "acceso" a la base de datos (en lugar de un acceso por cada instrucción).
- Integración en los productos de ORACLE: los bloques o procedimientos PL/SQL son compilados y ejecutados por el "motor" de PL/SQL. Este motor está integrado en el motor de la base de datos, así como en determinadas herramientas.

#### 1.4 Alternativas de solución

“PLSQL WIZARD. Es un software que permite la migración del código fuente PL/SQL a Java, distribuido por informatikatelier. Esta herramienta tiene una interfaz muy fácil de utilizar para el usuario. La forma de utilizar este software es pegando el código fuente de PL/SQL y la herramienta lo convierte a código fuente Java, mostrando el resultado final en una ventana al lado derecho del código fuente de PL/SQL, lo que se puede intuir que, para algunas aplicaciones, como suele ser muy frecuente en PL/SQL, en donde un paquete tiene muchas de líneas de código, es posible que se generen problemas con el software” (Madera Villada, 2012).

“SWISSQL-ORACLE TO JAVA.8 Esta herramienta al igual que la anterior permite la migración de código fuente PL/SQL a Java, esta herramienta es distribuida por Adventnet. Para realizar una migración de código fuente se debe realizar de la siguiente

forma: se debe adicionar el fichero donde se encuentre el código PL/SQL luego esta herramienta se encarga de leer el contenido y realizar la migración y este resultado es almacenado en archivos (clases) java” (Madera Villada, 2012).

“ORINDABUILD. Al igual que las otras, permite la migración de código fuente de PL/SQL a Java y es distribuida por Orinda Software. Para realizar una migración con esta herramienta, es necesario primero establecer una conexión a la base de datos, el siguiente paso a seguir es ubicar los procedimientos o funciones (código fuente), después de seleccionar el código fuente la herramienta se encarga de leer y genera el equivalente en java, con sus respectivas clases .java” (Madera Villada, 2012).

Luego de analizar las herramientas alternativas para migrar código fuente de Oracle Forms a una arquitectura multicapa se puede definir varios beneficios por parte del software abierto que nos provee IDE Eclipse de un entorno complejo tanto para su ejecución como compilación mediante varias herramientas anexas y necesarias para desarrollar un tipeado de lenguaje con una estructura en nuestro caso un entorno Spring MVC.

En especial Java lenguaje orientado a objetos y que por hoy día ocupa varios proyectos de organizaciones empresariales en desarrollo de la tecnología encaminados a una apertura WEB, contemplando modificar el código fuente en aceptación de la reingeniería de software aplicados a Sistemas Heredados.

## **CAPÍTULO 2. MARCO METODOLÓGICO**

### **2.1 Tipo de investigación**

En el presente desarrollo se enfoca en un proceso de investigación inductivo es una forma relativamente flexible basada en recopilar y analizar datos mediante la exploración obtenidos de distintas fuentes partiendo de premisas particulares. La investigación “implica el uso de herramientas informáticas, estadísticas, y matemáticas para obtener resultados, un tipo de investigación que depende de la recopilación de datos, de conducta u observaciones que pueden interpretarse de una forma” (¿Qué es la investigación cuantitativa?, s.f.) generalizada para obtener teorías o hipótesis.

#### **2.1.1 Metodología seleccionada**

“El método inductivo es un proceso utilizado para poder sacar conclusiones generales partiendo de hechos particulares. Es el método científico más usado.

El inductivismo va de lo particular a lo general. Es un método que se basa en la observación, el estudio y la experimentación de diversos sucesos reales para poder llegar a una conclusión que involucre a todos esos casos. La acumulación de datos que reafirmen nuestra postura es lo que hace al método inductivo.

Es un método que se basa en la observación, el estudio y la experimentación de diversos sucesos reales, para poder llegar a una conclusión que involucre a todos esos casos. La acumulación de datos que reafirmen nuestra postura es lo que hace al método inductivo” (¿Qué es el método inductivo?, s.f.).

En el desarrollo de la investigación del sistema control de bienes, el método inductivo es el que más características presenta ya que cuenta con una conclusión general, el cual es administrar el control de bienes a disposición de los empleados en todas las dependencias.

En el desarrollo de la investigación del sistema el método inductivo es el que más características presenta ya que cuenta con una conclusión general la custodia en todos los campamentos donde la empresa dispone de bienes a disposición de los empleados.

### **¿Qué es una Encuesta?**

La técnica de encuesta es ampliamente utilizada como procedimiento de investigación, ya que permite “obtener y elaborar datos de modo rápido y eficaz.

La encuesta permite aplicaciones masivas, que mediante técnicas de muestreo adecuadas pueden hacer extensivos los resultados a comunidades enteras” (León, 2019, pág. 23).

“Como tal, una encuesta está constituida por una serie de preguntas que están dirigidas a una porción representativa de una población, y tiene como finalidad averiguar estados de opinión, actitudes o comportamientos de las personas ante asuntos específicos” (ENCUESTA, s.f.).

## **2.2 Recopilación de información**

Determinar de qué manera la migración a una arquitectura multicapas (MVC) de un sistema de la organización, permitirá el avance futuro tecnológico mediante encuestas, como resultado mejor procesamiento de información, recolección de datos en la entidad pública EPMMOP.

### **2.2.1 Técnicas de recopilación de información**

Determinar el tamaño de la muestra que se va a seleccionar es un paso importante para el estudio de investigación de migración de aplicaciones informáticas.

Proceso para la encuesta:

- Fijar objetivos específicos y medibles

- Planificar la encuesta
- Asegurarse que los recursos necesarios están disponibles
- Diseño de la encuesta
- Preparar el instrumento para la recogida de datos
- Validar
- Seleccionar participantes
- Administrar y apuntar el instrumento
- Análisis de datos
- Informe sobre resultados.

Formula:

$$n = \frac{k^2 \cdot p \cdot q \cdot N}{(e^2 \cdot (N-1)) + k^2 \cdot p \cdot q}$$

“N: es el tamaño de la población o universo (número total de posibles encuestados).

K: es una constante que depende del nivel de confianza que asignemos. El nivel de confianza indica la probabilidad de que los resultados de nuestra investigación sean ciertos: un 95,5 % de confianza es lo mismo que decir que nos podemos equivocar con una probabilidad del 4,5%.

Los valores k más utilizados y sus niveles de confianza son:

K	1,15	1,28	1,44	1,65	1,96	2	2,58
Nivel de confianza	75%	80%	85%	90%	95%	95,5%	99%

e: es el error muestral deseado. El error muestral es la diferencia que puede haber entre el resultado que obtenemos preguntando a una muestra de la población y el que obtendríamos si preguntáramos al total de ella” (Díaz Contreras, 2011, pág. 25).

### **Tabulación 006E de la encuesta.**

#### **Encuesta**

**Fecha:** 13/06/2019

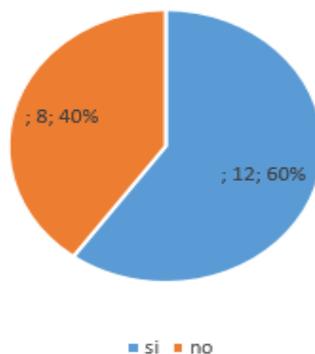
**Dirigida a:** Departamento Administrativo/ Dirección de Control de Bienes.

**Objetivo:** El objetivo de la entrevista es conocer el proceso actual de registro bienes a nivel solo de matriz principal, y entender la problemática de la empresa a lo largo de todos los campos asociados.

#### **Preguntas y respuestas.**

**Pregunta 1:** ¿Considera eficiente el proceso del cual se obtiene registro de bienes a escala de matriz?

- Si
- No



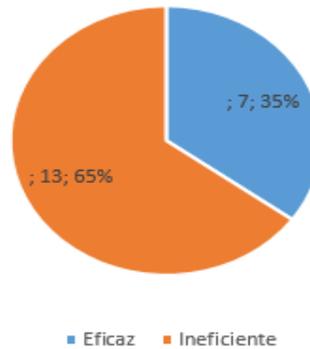
#### **Análisis Respuesta 1**

**Respuesta 1(SI):** El 60% de los encuestado considero eficiente al proceso, pero desearía en un futuro cambiar a ser un sistema en cuanto a todos los anexos a la empresa.

**Respuesta 2(NO):** El 40% de los encuestados considero eficiente al proceso por lo cual no se está obteniendo esta información de manera automática de todos los campos anexos.

**Pregunta 2:** ¿Cómo piensa usted que están manejando los reportes de la información de bienes de acuerdo al Estado en el sistema de custodio bienes?

- Eficaz
- Ineficiente



### **Análisis Respuesta 2**

**Respuesta 1(Eficaz):** El 35% de los encuestados considero eficiente al proceso, pero desearía en un futuro cambiar en algunos procesos.

**Respuesta 2(Ineficiente):** El 65% de los encuestados respondió en tema de procesamiento de la información es un poco demoroso la obtención de los reportes.

**Pregunta 3:** ¿Qué indicadores de información en el área Bienes Custodio necesita conocer reportes para el área o departamento por?

- Ubicación Física del Bien por departamento.
- Bien estado Activo O dado Baja por Anexos.
- No necesita cambios.

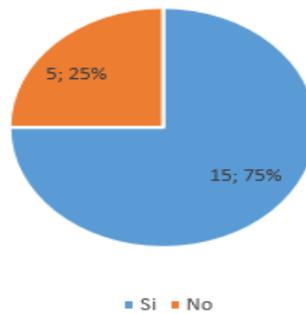


### **Análisis Respuesta 3**

**Respuesta 3:** En la encuesta realizada el 45% de los encuestados afirman en su mayoría algunos cambios en reportes y añadir nuevos reportes de procesos, el 33% requiere reportes por Ubicación Física del Bien por departamento y el 22% No necesita cambios.

**Pregunta 4:** ¿Requiere capacitación del sistema en la actualidad?

- Si.
- No.

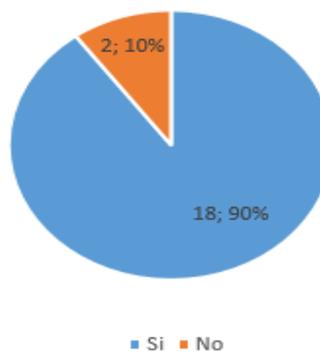


#### **Análisis Respuesta 4**

**Respuesta 4:** El 75% de la encuesta realizada la mayoría requiere capacitación del sistema ya que cuenta con desconocimientos por otro lado el 25% conoce en su totalidad el manejo del sistema.

**Pregunta 5:** ¿Qué le parece la posibilidad de manejar el sistema Bienes EPMMOP en todos los campos anexos para ser una información única y registrada en la intranet?

- Si
- No



#### **Análisis Respuesta 5**

**Respuesta 5:** El 90% de los encuestados le parece interesante el manejar en la intranet el sistema que permita presentar información actualizada y precisa que nos ayude a monitorear el estado de bienes de la empresa.

**Respuesta 5:** El 10% piensa que no por no saber manejar la intranet Portal Interno.

#### **Análisis de Encuestas**

De las encuestas realizadas en la empresa EPMMOP a la población seleccionada un gran porcentaje tiene la aceptación de un sistema coordinado de Custodia Bienes y desean una capacitación de ser necesario.

# CAPÍTULO 3. PROPUESTA

## 3.1 Diagramas de procesos

Proceso actual mediante sistema Oracle Forms Custodia Bienes

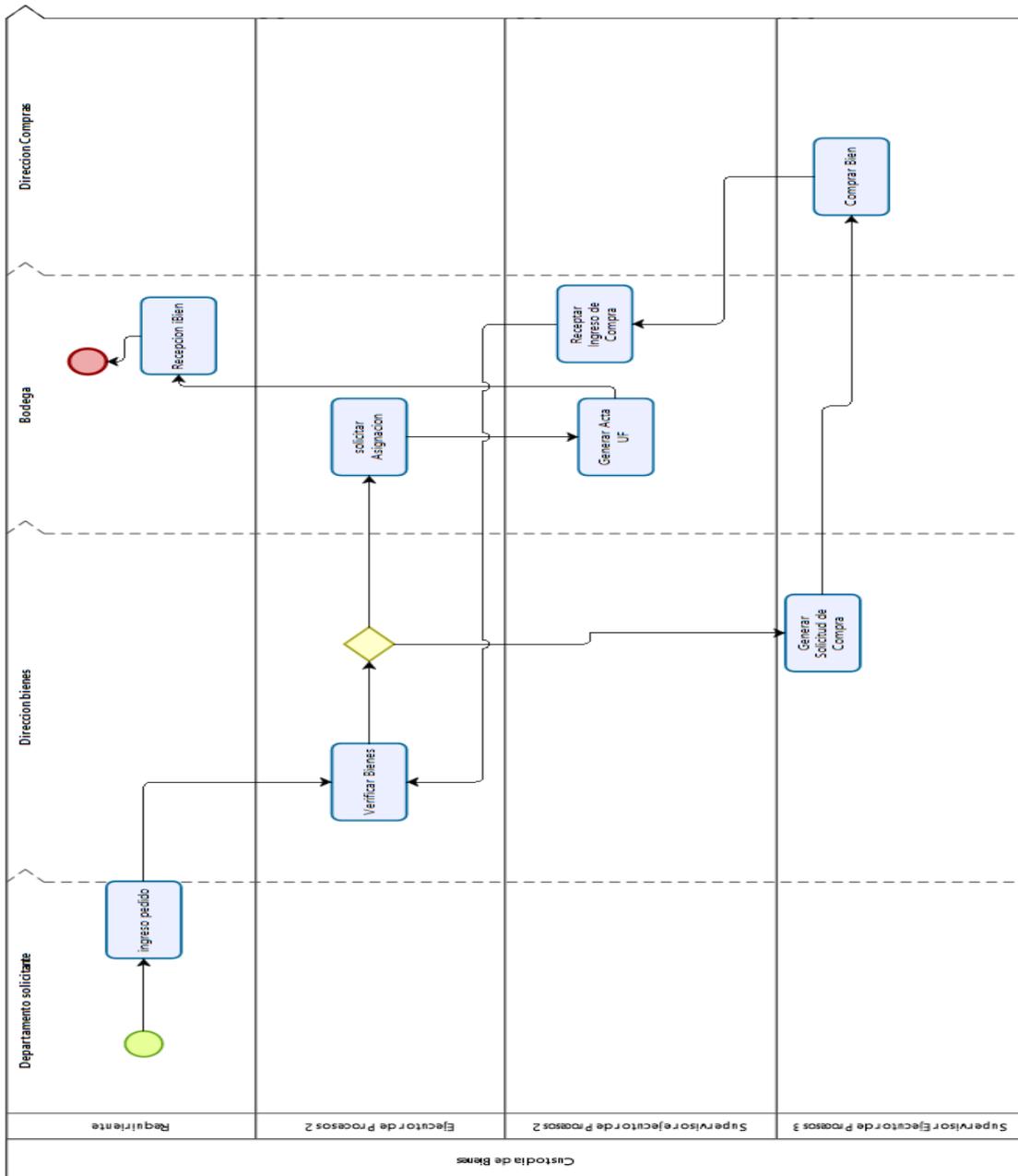


Figura 2. Diagramas de Procesos (limitados tecnológicamente)

Proceso Migrado a Java EE de Sistema Custodia Bienes en Portal Institucional.

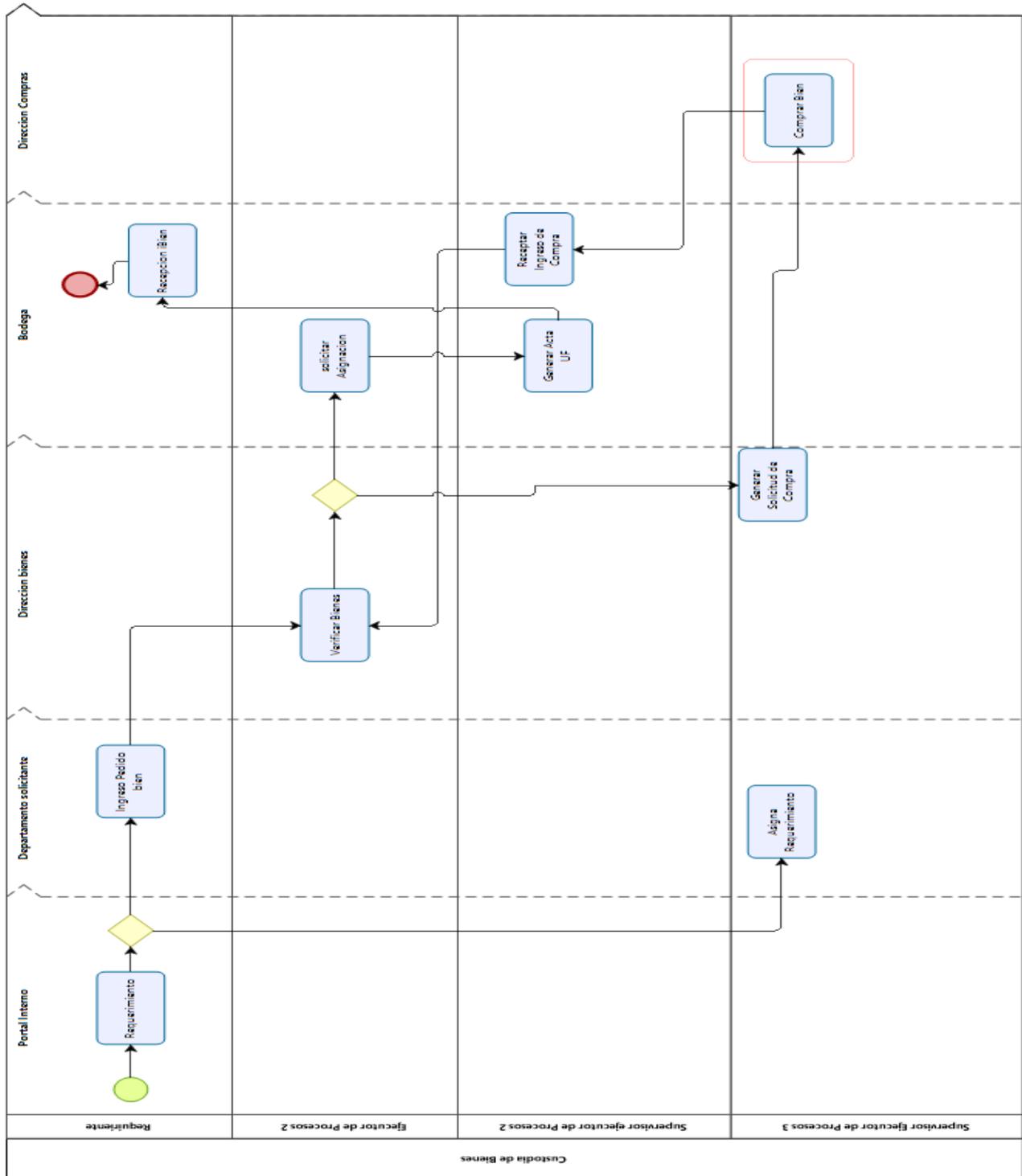


Figura 3. Diagrama de proceso automatizado

### **3.2 Factibilidad técnica**

La empresa EPMMOP cuenta con los recursos necesarios para la migración del sistema:

- Servidores que maneja el equipo de TI área infraestructura
- Computadores alto nivel que maneja el área de desarrollo de programación en TI.
  1. Sistema Operativo Windows 10
  2. Antivirus kaspersky
  3. Framework de desarrollo Eclipse Luna for Java Developers
  4. Oracle PL/SQL Developer
- Un espacio físico de trabajo suficiente para el área de desarrollo en TI.

### **3.3 Factibilidad operacional**

Se cuenta con recursos humanos capacitados para el proyecto de migrar sistemas y competentes de acuerdo al área de desarrollo con Ingenieros/Analistas, además cuenta con recurso humano externo profesional el que será capaz de manejar el portal interno con facilidad luego de su capacitación y entrega de manuales técnicos.

### **3.4 Factibilidad económica-financiera**

La empresa EPMMOP-Q dispone en el ámbito financiero del sector público generado un presupuesto anual mediante el sector del gobierno designado de paso.

### **3.5 Especificación de Requerimientos**

#### **Propósito**

Con la migración del sistema de control de bienes beneficiará a toda la institución ya que controlará y reportará el bien institucional de acuerdo a su estado como puede ser

adquisición o bajas mediante el sistema permitirá registrar su distinta información de acuerdo al tipo de bien.

### **3.5.1 Ámbito del software**

MIGRACIÓN DE SISTEMA “CONTROL DE BIENES” DE ORACLE FORMS A UNA ARQUITECTURA (JSF) JAVA PARA LA EMPRESA MUNICIPAL DE MOVILIDAD Y OBRAS PÚBLICAS (EMMOP-Q).

Beneficiará al personal Administrativo que maneja el custodio de todos los bienes que dispone en distintos campamentos anexos que registran estado del mismo y novedades a su vez mantendrá un reporte del cual se podrá manejar en el momento que así disponga su beneficio ya que contarán con el servicio en el portal interno de la empresa así manejar una misma información ordenada y sin mucha documentación será un sistema independiente.

- El sistema no permitirá el ingreso a personas que no laboren en el área administrativa custodia bienes, con esto dar seguridad al bien físico.
- No podrá enlazarse con otros sistemas de la web interna ya que contará con información y diseño solo de control de bienes.
- No podrá registrar personal nuevo que ingrese a la institución ya registra en un sistema de nómina distinto.

### **3.5.2 Funciones del producto**

#### **Antecedentes**

La Dirección de Tecnología de la Información de la EPMMOP posee 4 bases para la arquitectura de servicios, plasmadas en el PETI (Plan Estratégico de Tecnologías de la Información) y que son:

1. Arquitectura de Información
2. Arquitectura de Negocio

3. Arquitectura Tecnológica
4. Arquitectura de Aplicaciones

Dentro de la arquitectura de aplicaciones, que es la encargada de prestar el servicio eficiente, oportuno y operativo de las aplicaciones informáticas a los usuarios internos de la Empresa, posee varios desarrollos de software en algunas plataformas y bases de datos.

Durante el ciclo de vida de un software, puede surgir la necesidad de modificarlo para ejecutarse en un entorno diferente, se debe determinar las acciones necesarias para lograr dicha migración, y luego desarrollar y documentar los pasos requeridos para efectuar la migración.

Específicamente, se puede definir la migración de aplicaciones como el conjunto de actividades para actualizar, modificar o eliminar equipos informáticos y recursos relacionados con la tecnología y aplicaciones de información.

La migración, “tiene por finalidad el traslado del sistema a un nuevo ambiente operativo, conservando su funcionalidad y datos originales. En todos los casos se persigue el posibilitar el mantenimiento y posterior adecuación a nuevos requerimientos” (Ciolli, 2007, pág. 12).

### **Justificación**

Entre las causas por las cuales se realizará la migración de aplicaciones se destacan las principales:

- La inexistencia de soporte técnico para la herramienta de desarrollo de algunas aplicaciones actuales en producción.
- La no existencia de soporte técnico en algunas de las bases de datos que posee la EPMOP.
- Normativas vigentes, respecto de utilización de software libre.

## PROCEDIMIENTO PROPUESTO

La migración es un proceso de cambio que puede efectuarse tanto en los elementos del software como del hardware. Todas las migraciones van a incluir una serie de pasos a seguir:

- Determinación de la causa de la migración.
- Recolección de datos claves del elemento a migrar
- Consulta a los usuarios y explicación de las razones por las que se va a llevar a cabo la migración y cómo les afectará.
- Planificar el momento y procedimiento de migración.
- Evaluar los resultados de la migración.

“Debe reconocerse que un trabajo de migración es normalmente un proyecto de ingeniería de sistemas, que por su importancia merece el calificativo de crítico. Esto es así tanto por la relevancia de los entornos migrados (datos y aplicaciones), que deberán ofrecer finalmente la misma eficiencia y operatividad que ofrecían en el entorno anterior, así como, por la necesidad de hacer mínimo el impacto en todos los niveles de la organización” (Ciolli, 2007, pág. 12).

Hay que hacer referencia al objetivo de enfrentar un cambio de cultura tecnológica, para el que habrá que prever recursos técnicos y humanos, y que deberá ser acompañado del necesario entrenamiento del personal y usuarios.

## ESTRATEGIA DE MIGRACIÓN

El plan de migración se apoyará en tres pilares:

- Metodología.
- Herramientas
- Métodos de pruebas y personalización.

“La **metodología** garantiza:

En primer lugar, un procedimiento sistemático que asegura que el trabajo realizado sea controlable y sus resultados predecibles.

En segundo lugar, que se dispone de un repositorio con toda la información necesaria para abordar la migración: cadenas de programas, programas fuente, estructura de bases de datos, librerías de funciones y demás elementos informáticos necesarios.

En tercer lugar, contempla la obtención del modelo de negocio a migrar, a partir de la información contenida en el repositorio, y considera además la realización de los planes de prueba de las aplicaciones migradas.

Por último, define las reglas de generación del código migrado, conforme a los estándares establecidos, las librerías de funciones usadas y cualquier otra consideración de interés.

Las **herramientas** de migración permiten obtener un modelo del negocio a migrar, que lo hace independiente de los lenguajes de las aplicaciones, con lo cual el modelo obtenido resultará válido en caso de ser necesarias futuras migraciones a otras tecnologías.

Las **técnicas de pruebas** y personalización incorporan las reglas de generación introducidas por la metodología a los fines de obtener aplicaciones funcionales y operativamente fiables y las optimizan para su funcionamiento en el entorno informático existente en la empresa.” (Ciolli, 2007, pág. 13).

- Parametrización como de colores.
- Parametrización de marcas.
- Parametrización de custodios.
- Parametrización de custodios administrativos.
- Parametrización de motivo movimientos.
- Parametrización de tipo interviniente.
- Parametrización de procedencia del bien.
- Parametrización periodo constatación física.

- Registrar ingresos de nuevos Bienes Muebles de acuerdo a la categoría (tipo de bien) por parte de los proveedores.
- Registro individual de la persona con datos de lugar de trabajo, dependencia y gerencia a la cual pertenece dentro de la empresa y ejerce sus labores con el bien mueble a su cargo (custodio).
- Registro masivo a la persona que verificara la supervisión del lugar de trabajo y como jefe inmediato se registraran los bienes de sus colaboradores bajo su dependencia como Custodio Administrativo.
- Registro de cambio de custodios mediante búsqueda por persona (custodio) o por numeración del bien (niop).
- Registro de cambio de custodio administrativo y nuevo ingresos de custodios administrativos.
- Registro de movimientos de bienes muebles del lugar de trabajo o dependencia.
- Habilitar custodios que se encuentren desactivados.
- Habilitar bienes dados de baja (desactivados).
- Generación de acta de usuario final.
- Modificar el estado de bienes en estado activo.
- Certificado de No poseer bienes.
- Consulta de bienes por usuario que inicio la sesión.
- Modificar estados de bienes que poseen los custodios administrativos.

### **3.5.3 Características de los usuarios del sistema**

Los usuarios a disponer el sistema podrán ser empleados administrativos que manejaran sistemas de gestión dentro del portal interno web institucional la asignación de sus roles son aceptados de acuerdo a su función y registrados bajo un sistema de registro de usuario por acceso.

Tabla 1. *Perfiles de usuario*

Nombre de Usuario	Tipo de Usuario	Área Funcional	Actividad
<b>Administrador</b>	Administrador del Sistema	Administración	Administrar el sistema. Administrar cuentas. Realizar búsquedas. Control de Materiales. Obtener reportes. Realizar reportes.
<b>Usuario 1</b>	Ejecutor de Procesos 2	Constatación y Asignación	Control entrada y salida de bienes Realizar Búsquedas
<b>Usuario 2</b>	Supervisor Ejecutor de Procesos 2	Verificación Almacenamiento Adquisición	Obtención de reportes Control Nuevas Adquisiciones
<b>Usuario 3</b>	Supervisor Ejecutor Procesos 3	Confirmación Solicitud Adquisición Asignación Recursos	Administrar Recursos por direcciones o departamentos Aceptación de adquisiciones ingresos

### 3.5.4 Restricciones de desarrollo

- Se utilizará exclusivamente el sistema gestor de base de datos PL/SQL Developer debido a que la empresa posee dicho gestor con sus respectivas licencias.
- El sistema no permitirá crear usuarios.
- Se manejará un esquema de normas internas de desarrollo.
- El sistema no permitirá generar los perfiles de usuario.
- El sistema no permitirá acceder a datos personales exclusivos registrados en otro sistema.

- El desarrollo del sistema se lo realizará exclusivamente dentro de las instalaciones de la empresa acogiéndose a los horarios de trabajo pre-establecidos por la misma.
- El sistema será solo de uso institucional dentro de la plataforma interna.

### **3.5.5 Requisitos**

#### ***Funcionales.***

- El sistema permitirá crear, modificar y eliminar en las páginas tipo catálogos.
- En la creación de cada página tipo se genera un identificador único para cada clase.
- El sistema permitirá registrar bienes siempre y cuando se asigne por su primera vez el ingreso a un custodio administrativo.
- El sistema asignará el bien contando con una dependencia y lugar físico donde será designado para su uso con el custodio final.
- El sistema permitirá el cambio de custodio por custodio manejándose con un niop único por bien
- El sistema permitirá verificar de forma visual los bienes y que persona esta responsable así verificar mediante un reporte de acta usuario final.
- El sistema realizara un certificado de no tener bienes para uso empresarial.

#### ***No funcionales.***

##### **Eficiencia**

- Los datos modificados serán actualizados por los usuarios que se encuentren con acceso al sistema en tiempos cortos.
- El sistema tendrá la capacidad de operar con los datos almacenados que se encuentren en el momento y así controlar tiempos reales.

### **Usabilidad**

- El sistema contara con manuales de usuario y técnicos estructurados adecuadamente para conocimientos tanto de usuarios como técnicos de desarrollo.
- El sistema debe poseer una interfaz de acuerdo al esquema empresarial manejado por la institución.

### **Disponibilidad**

- El sistema contara con manuales de usuario y técnicos estructurados adecuadamente para conocimientos tanto de usuarios como técnicos de desarrollo.
- El sistema debe poseer una interfaz de acuerdo al esquema empresarial manejado por la institución.

### **Seguridad**

- El sistema tendrá un acceso restringido solo para personal de la institución con acceso personal.
- El sistema se manejará en una red privada de la institución.

### **Mantenibilidad**

- El sistema cuenta con técnicos de soporte para cada sistema destinado de acuerdo a sus funciones.
- El sistema contara con mantenimiento de infraestructura para las conexiones y depuración de bases de datos.

### **Metodología XP**

Se desarrollará con este modelo puesto que es de una forma ágil y flexible.

“Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

### **Roles**

- Cliente: responsable de definir y conducir el proyecto, así como sus objetivos.
- Programadores: estiman tiempos de desarrollo de cada actividad y programan el proyecto.
- Tester: Encargado de Pruebas.
- Tracker: Encargado de Seguimiento.
- Coach: Entrenador. Su papel es guiar y orientar al equipo.
- Big Boss: Gestor del proyecto, gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado” (Metodología XP Programación Extrema (Metodología ágil), 2018).

Tabla 2. *Roles Metodología XP*

<b>Roles</b>	<b>Asignado A:</b>
Programador	Edison Pumasunta, Luis Simbaña, Fabricio Pacheco.
Cliente	EPMMOP.
Encargado de pruebas (Tester)	Johanna Palacios.
Encargado de Seguimiento (Tracker)	Iván Terán.
Entrenador (Coach)	Luis Simbaña
Consultor	Paulina Silva.
Gestor (Big Boss)	Iván Terán.

## CAPÍTULO 4. IMPLEMENTACIÓN

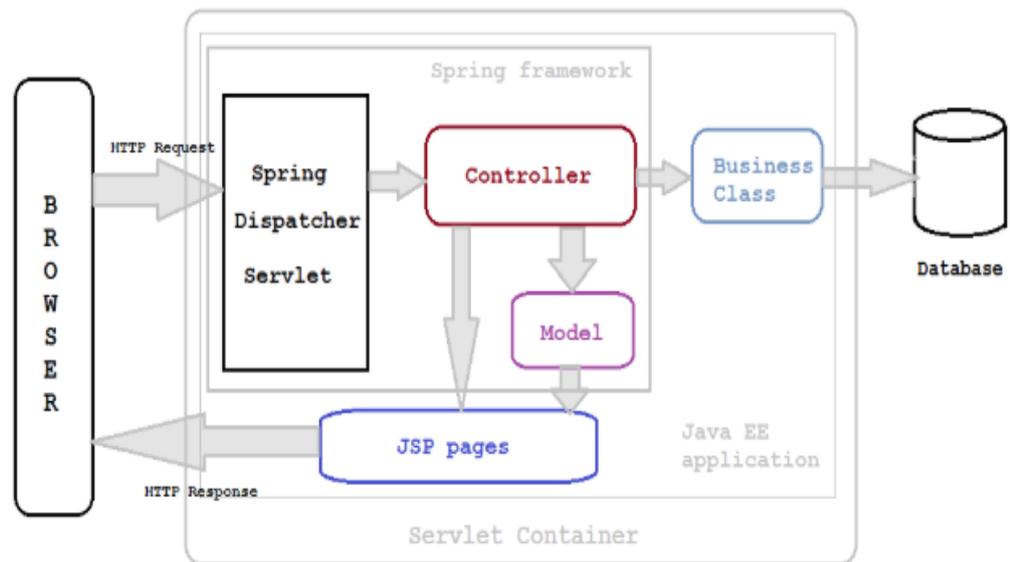
### 4.1 Diseño

### 4.2 Esquema de la base de datos

Anexos.

*Figura 4.1* Modelo de Base de datos del sistema Control Bienes.

### 4.3 Diagrama de la arquitectura del sistema



*Figura 4.* Diagrama de la arquitectura del sistema Custodia Bienes. Tomado de: <https://www.codejava.net/frameworks/spring/how-to-enable-spring-mvc-in-webxml>

## 4.4 Diseño de interfaces

The screenshot shows the 'Sistema de Control de Bienes' web interface. At the top, there is a header with the 'QUITO' logo and 'EPMMOP' text. Below the header, the system title 'Sistema de Control de Bienes' is displayed. The interface includes a navigation menu ('Menú Selecciones'), a system identification area ('Identificación Sistema'), and a page title ('Titulo Página'). The main content area is divided into two sections: 'Área contenido Datos' and 'Área interacción'. The 'Área interacción' section contains a table with the following data:

Secuencial	Descripcion
56	AZUL
41	BEIGE
89	BLANDO
45	CAQUI
13	CARAMELOS
42	DORADO
73	EDISON
71	EEE
49	HAYA
38	MARRON

The footer of the page contains the text '© COPYRIGHT 2014 - EPMMOP-QUITO' and 'Pie de Página'.

## 4.5 Estándares de programación utilizados

### Nombrar Paquetes

Por defecto todos los paquetes se escribirán en minúsculas y sin utilizar caracteres especiales.

Los paquetes deberán empezar con:

`ec.gob.epmmop.nombreDeAplicacion.nombrePaquete`

`ec.gob.epmmop.bienesEpmmpop.controladores`

## Nombrar Clases

Los nombres de clases empiezan con letra mayúscula. Si es una palabra compuesta se debe utilizar estilo de escritura CamelCase. Intentar mantener los nombres de clases simples y descriptivos. Si la clase cumpliera algún patrón determinado o tuviera una funcionalidad específica es recomendable definirlo en el nombre. Por ejemplo, para los controladores se debe agregar el sufijo Controlador al final del nombre, para los servicios se agrega la palabra Servicio etc. Ejemplo:

- ConsultaProveedoresInscritosControlador (Controlador)
- AdjudicaciónICServicio (Servicio)
- Adjudicación (Modelo o entidad)

## Nombrar Métodos

Los métodos deben ser verbos en infinitivo, deben empezar con letra minúscula y si es una palabra compuesta se debe utilizar lowerCamelCase. No se permiten caracteres especiales.

- verificarDepenUsuario();
- consultar();
- inicializar();

## Nombrar Variables

Deben empezar con minúsculas y se debe utilizar lowerCamelCase. Los nombres deben ser descriptivos y deben ser en singular, no deben ser verbos. En el caso de listas debe ir el prefijo " lista" o ir en plural. Se debe respetar el encapsulamiento de cada una de las clases, es decir todas las variables deben ser privadas.

- Private List<PlanificacionVehicular>listaPlanificacion;
- Private List<Dependencia>dependencia;

## Constantes

Las constantes deben ir escritas en Mayúsculas, todas las letras. Deben ser estáticas y final. Si son palabras compuestas, van separadas por guion bajo. Se deben utilizar constantes en cada clase que requiera alguna parametrización o manejo de estados.

- `Public static final String NOTIFICADOR_EQUIPOS="E";`

## Comentarios

Se debe utilizar el estándar de JEE

Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase, no se debe exagerar en su uso.

Como norma es obligatorio proporcionar un comentario de documentación por cada clase/interface, método, propiedad o constante creado.

Para cada clase se debe redactar un comentario general de la clase, que hace la clase o para que fue creada.

## Tomar en cuenta

- Se debe escribir en tercera persona.
- Las variables dentro de la clase son private y deben tener métodos de acceso (get & set).
- Por cada aplicación se debe exportar un @javadoc.

## 4.6 Implementación

Analizadas las metodologías de migración enfocadas en una nueva arquitectura de desarrollo y respuesta rápida, siendo eficiente durante el proceso de migración del proyecto cada una contando con su respectiva información a realizar por fases, así como

metodología de investigación en base la recolección y minería de datos, siendo parte de una metodología de migración de software información que proviene de un antecedente del Sistema Heredado.

Para el presente proyecto se procede a seleccionar la metodología de migración MADIISH que abarca en gran ámbito todos los procesos de trabajo a desarrollar contando con un plan guía enfocado en etapas contempladas en la propuesta.

Contemplada dentro de la metodología de migración a Sistemas Heredados en una de sus etapas de desarrollo durante la planificación se utilizará para el presente proyecto Extreme Programing (XP) como método de desarrollo ágil.



**Figura 5.** Estructura general de la metodología MADIISH. Tomado de: Silva, D. T., Galván, J. D. O., Gómez, H. A., & López, R. R. Una Propuesta de Metodología para la Migración de Sistemas Heredados.

Durante el ciclo de vida de un software, puede surgir la necesidad de modificarlo para ejecutarse en un entorno diferente, se debe determinar las acciones necesarias para lograr dicha migración, y luego desarrollar y documentar los pasos requeridos para efectuar la migración. Este documento es un detalle para la elaboración del plan de factibilidad de migración, para lo cual se debe formar un equipo de trabajo multidisciplinario orientado a medir el impacto, establecer planes de acción, para: usuarios, administradores y sistema a migrar.

## GUÍA PARA EL PLAN DE MIGRACIÓN

Tabla 3. *Guía de Migración*

---

### DESCRIPCIÓN

---

La Dirección de Tecnología de la Información de la EPMMOP posee 4 bases para la arquitectura de servicios, plasmadas en el PETI (Plan Estratégico de Tecnologías de la Información) beneficiando a un desarrollo tecnológico y que estas son:

- Arquitectura de Información.
- Arquitectura de Negocio.
- Arquitectura Tecnológica.
- Arquitectura de Aplicaciones.

Dentro de la arquitectura de aplicaciones, que es la encargada de prestar el servicio eficiente, oportuno y operativo de las aplicaciones informáticas a los usuarios internos de la Empresa, posee varios desarrollos de software en algunas plataformas y bases de datos.

---

Tabla 4. *Descripción de las hojas de la presente guía*

Roles	Asignado A:
<b>Datos Generales</b>	En la hoja definida como datos generales, se solicita información del proyecto aprobado por Gobierno Electrónico ha superado los 180 días de su ejecución, por lo tanto debe presentar el plan de factibilidad de migración, como también datos del sistema al cual se va a migrar.
<b>Guía de elaboración del plan</b>	Las actividades e hitos que se presentan en esta hoja son una guía de referencia para la elaboración del plan de factibilidad de migración. El análisis debe contemplar; para usuarios y administradores, los factores de análisis propuesto son: brecha de conocimientos; expectativa al cambio, necesidades especiales; y para tecnología: compatibilidad de hardware/software; compatibilidad de documentos; infraestructura de gestión; costos requeridos para la migración.

---

Tabla 4. *Descripción de las hojas de la presente guía*

<b>Roles</b>	<b>Asignado A:</b>
<b>Matriz de análisis</b>	La presente hoja se divide en dos partes: matriz de análisis, permite evidenciar el impacto y las acciones establecidas por la institución para poder llevar a cabo la migración, estos resultados son elementos base para la construcción del plan de factibilidad de migración. En el segundo, grupo de trabajo deberá registrar los datos de los funcionarios que desarrollarán el análisis, el plan y la ejecución de la migración.
<b>Plan</b>	La presente hoja contiene el formato del cronograma para el Plan de Factibilidad de Migración, la cual contempla 4 grandes fases que debe ser considerada por la institución el momento de su desarrollo, las fases son: planeación inicial, plan de difusión, plan de capacitación, plan de migración tecnológica. La institución podrá establecer los hitos y actividades que requiera de acuerdo a su realidad.

Tabla 5. *Datos generales del proyecto*

<b>DATOS GENERALES DEL PROYECTO</b>	
<b>Nombre de la Institución:</b>	Empresa Pública Metropolitana de Movilidad y Obras Públicas.
<b>Nombre del proyecto:</b>	Sistema Control de Bienes.

Tabla 6. *Datos sistema/software a migrar*

<b>DATOS DEL SISTEMA /SOFTWARE A MIGRAR</b>	
<b>Nombre versión del sistema /software a migrar:</b>	Oracle Forms: Sistema Control de Bienes (versión 1).
<b>Alternativa de software de código abierto a implementar:</b>	IDE Eclipse Luna: Java.

De metodologías mencionadas anteriormente se describe con la seleccionada para el desarrollo durante todo el proceso de migración del sistema de control de bienes,

descrito en un alcance por módulos el cual se iniciará con el módulo principal para posterior en un futuro proseguir con los submódulos de control.

La migración debe cumplir las siguientes fases **MADISSH**:

**Fase C:** Se define de forma abstracta la conformación del supra sistema en sus diferentes sistemas, así como la de sus sistemas en sub sistemas, lo cual permite establecer la relación entre cada sub sistema para facilitar la determinación de la jerarquía de migración y la interoperabilidad de dicho subsistema o sistema con otros. Esta fase se denomina fase cero debido a que durante la migración se realiza una vez, al inicio de la migración. A su vez, permite el reconocimiento de las necesidades del cliente y se plantean las estrategias necesarias para poder llevar a cabo el proceso de migración.

Tabla 7. *Madiish fase C*

Fase	Actividades
Recolección de información	<p data-bbox="624 1113 1428 1178">Esta fase brindará los datos necesarios para empezar el proceso de migración:</p> <ul data-bbox="675 1216 1428 1491" style="list-style-type: none"> <li data-bbox="675 1216 1007 1247">• Inventario del Software.</li> <li data-bbox="675 1252 1428 1317">• Definición de orden de migración por precedencia de aplicaciones</li> <li data-bbox="675 1321 1166 1352">• Definición de prioridad de migración.</li> <li data-bbox="675 1357 1428 1422">• Determinación de unidad propietaria del sistema y de los datos que maneja.</li> <li data-bbox="675 1426 1326 1458">• Detección de usuario final responsable del sistema.</li> <li data-bbox="675 1462 1370 1494">• Determinación de responsable técnico de cada sistema.</li> </ul>
Prioridad de migración	<p data-bbox="624 1520 1059 1552">Se establece una escala de prioridad:</p> <p data-bbox="624 1556 983 1588">0: No es urgente su migración</p> <p data-bbox="624 1592 711 1624">1: Baja</p> <p data-bbox="624 1628 730 1659">2: media</p> <p data-bbox="624 1664 708 1695">3: Alta</p>

**Fase 1:** Se elige un sistema a migrar tomando en cuenta los siguientes criterios: importancia, sencillez, conveniencia, complejidad, contingencia y selectividad.

A su vez, se realiza un análisis de entradas y salidas de datos para determinar la dependencia de otros subsistemas. De igual forma se definen los nuevos requerimientos de usuario y se comprende mediante el análisis de procesos de negocio la operación del sistema, esquematizando el conocimiento sobre un diagrama de entradas y salidas de datos, permitiendo la eficiencia de los nuevos procesos.

Tabla 8. *Madiish fase 1*

Fase	Actividades
<b>Recolección de información</b>	<p data-bbox="624 685 1426 752">Esta fase brindará los datos necesarios para empezar el proceso de migración:</p> <ul data-bbox="751 790 1294 891" style="list-style-type: none"> <li data-bbox="751 790 1294 822">• Categorización por Front End y Back End</li> <li data-bbox="751 824 1267 855">• Definición de base de datos a ser usada.</li> <li data-bbox="751 857 1267 889">• Definición de Front End a ser utilizado.</li> </ul>
<b>Capacitación</b>	<p data-bbox="624 960 1426 1126">Uno de los puntos claves en el proceso de migración es el entrenamiento que se le debe proporcionar a los usuarios, el mismo debe contribuir a que el factor de resistencia al cambio sea lo más bajo posible y las metodologías de aprendizaje a utilizar deben incentivar a la autoformación e investigación.</p> <ul data-bbox="675 1164 1418 1196" style="list-style-type: none"> <li data-bbox="675 1164 1418 1196">• Capacitación del personal técnico involucrado por sistema.</li> </ul>

**Fase 2:** Se realiza el modelado la base de datos o actualización de la misma en caso de existir una previa de algún subsistema migrado anteriormente, el desarrollo de interfaces de usuario, la programación para la comunicación entre interfaces con BDD y la generación de una versión prototipo para realizar pruebas y mostrar al usuario.

Como primer modelo el prototipo esté dispuesto cambio de acuerdo a la necesidad de la organización, a su vez el subsistema migrado como nuevos módulos anexos podrán incluir registro de datos en versión pruebas para identificar el proceso (forma formulario web) cumpla sus requerimientos.

Tabla 9. *Madiish fase 2*

Fase	Actividades
<b>Migración total</b>	<p data-bbox="608 367 1426 461">La Migración Total contempla el cambio total del sistema. Se migran todos los servicios y se diseñan herramientas Open Source: Se debe hacer en fases por sistema, se determina:</p> <ul data-bbox="655 506 1050 640" style="list-style-type: none"> <li data-bbox="655 506 1050 528">• Pantallas de Parametrización.</li> <li data-bbox="655 539 963 562">• Pantallas de Procesos.</li> <li data-bbox="655 573 975 595">• Pantallas de Consultas.</li> <li data-bbox="655 607 963 629">• Pantallas de Reportes.</li> </ul> <p data-bbox="608 674 1426 707">Adicional se debe contemplar, reprogramación en base de datos de:</p> <ul data-bbox="655 752 1050 842" style="list-style-type: none"> <li data-bbox="655 752 772 775">• Vistas</li> <li data-bbox="655 786 1050 808">• Procedimientos almacenados</li> <li data-bbox="655 819 820 842">• Funciones</li> </ul> <p data-bbox="608 887 1426 940">Todo lo realizado debe guardarse en un solo repositorio de diccionario de datos.</p>

**Fase 3:** Se implementa un ambiente de pruebas lo más semejante al ambiente de producción en donde se pueda ejecutar un prototipo con información actualizada, si son los resultados esperados se podrá continuar con la siguiente fase, de no serlo se deberá regresar al análisis y esquematización de entradas y salidas de datos. Es aquí otra de las características de MADIISH, permite regresar dentro del ciclo de migración a un punto en el que se pueda determinar las fallas presentadas según el avance obtenido.

Tabla 10. *Madiish fase 3*

Fase	Actividades
<b>Documentación de la migración</b>	<ul data-bbox="608 1648 1426 1883" style="list-style-type: none"> <li data-bbox="608 1648 1426 1738">• Se deberá documentar todo el proceso paso a paso resaltando las experiencias que se consideren relevantes y que puedan ser de utilidad durante las migraciones.</li> <li data-bbox="608 1749 1426 1883">• Se deberá documentar todas las pruebas realizadas en el laboratorio (pruebas de hardware y pruebas de software), de manera que pueda ser utilizada como material de apoyo y/o referencia.</li> </ul>

Tabla 10. *Madiish fase 3*

Fase	Actividades
<b>Documentación de la migración</b>	<ul style="list-style-type: none"> <li>• La documentación de estas pruebas permitirá elaborar un manual de procedimientos y/o protocolos de pruebas, para usar en el laboratorio.</li> <li>• La documentación deberá realizarse en un formato estándar.</li> </ul>
<b>Capacitación Usuarios</b>	<ul style="list-style-type: none"> <li>• Uno de los puntos claves en el proceso de migración es el entrenamiento que se le debe proporcionar a los usuarios, el mismo debe contribuir a que el factor de resistencia al cambio sea lo más bajo posible y las metodologías de aprendizaje a utilizar deben incentivar a la autoformación e investigación.</li> <li>• Capacitación del usuario final.</li> </ul>

**Fase 4:** Una vez superadas las fases anteriores es necesario implementar el nuevo sistema en el ambiente de producción por lo que se debe anular el funcionamiento del subsistema heredado migrado, actualizar la base de datos del nuevo sistema e implementar finalmente el nuevo sistema.

**Fase 5:** Se realiza una documentación del sub sistema o sistema migrado según sea el caso en la cual se describen los procesos de negocios, la documentación del código fuente y la descripción gráfica del sistema.

Tabla 11. *Madiish fase 5*

Fase	Actividades
<b>Documentación de migración</b>	<ul style="list-style-type: none"> <li>• Diccionario de datos de la aplicación</li> <li>• Listado de objetos que usa con su respectiva descripción.</li> </ul>

**Fase T:** Elaboración de la documentación final, consiste en unir toda la documentación y retroalimentar los manuales o procesos en los que se encuentre ambiguo su contenido.

Tabla 12. *Madiish fase T*

Fase	Actividades
<b>Documentación de la migración</b>	<ul style="list-style-type: none"> <li>Manual técnico del sistema.</li> <li>Manual de usuario del sistema</li> </ul>

### Cronograma de trabajo.

### Catálogos – Parametrización

 															
Tarea Pantallas Parametrización	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			
												Semana1	Semana 2	Semana3	Semana4
Administrar Color	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Marca	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Tipo Interviniente	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Motivo Cambio del bien	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Custodios	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Custodios Administrativos/Registro	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Custodios Administrativos/Asignar Ubicación física	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Origen de bienes	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Firmantes Actos	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Periodo constatación física	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		
Administrar Ubicación física organigrama	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Corrección	Test 2	Pruebas Pre-Producción	Pruebas Implementación		

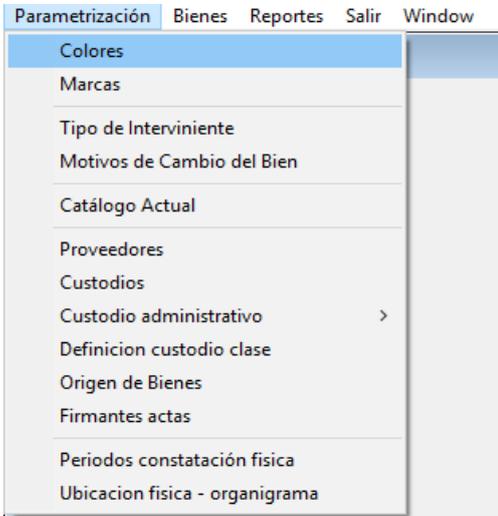
### Procesos

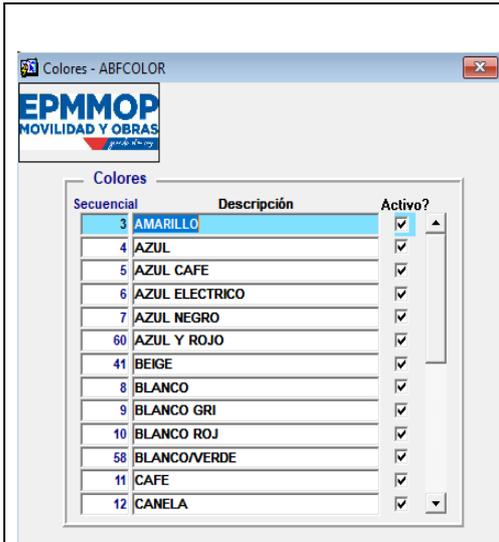
 															
Tarea Pantallas de Procesos	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			
												Semana1	Semana 2	Semana3	Semana4
Registro de Bienes	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
Acta de bienes ingresos/ Acta Entrega Recepción	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
Acta Usuario Final	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
CA./Acta Inicial	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
CA./Modificar estados bienes en actas	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
CA./Eliminar bienes en actas	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
CA./Traslado de bienes	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
Cambio custodia/ por custodia	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		
Cambio custodia/ por Nlpo	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio	Capa Controlador y Página					Test	Correcciones	Test	Pruebas Pre-Producción	Pruebas Implementación		

**Consultas**

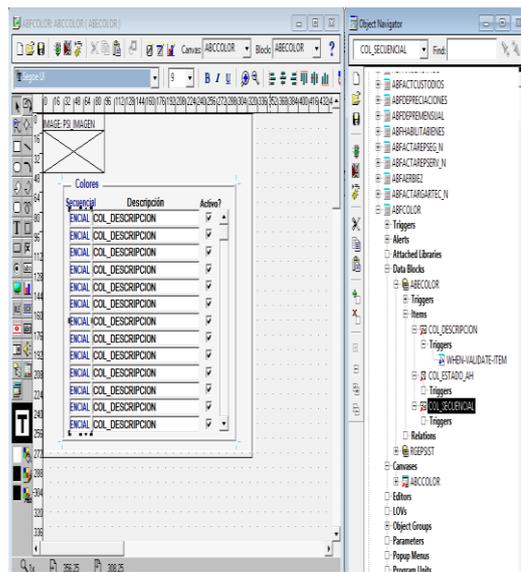
 <span style="font-size: 2em; font-weight: bold;">CRONOGRAMA DE MIGRACIÓN</span> 															
Tarea Pantallas de Procesos	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			
												Semana1	Semana 2	Semana3	Semana4
Consulta bienes custodio	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio				Capa Controlador y Página		Test	Correcciones	Test	Pruebas Pre-Producción		Pruebas Implementación	
Certificado No tener bienes	Capa Modelo	Capa Acceso Datos	Capa Acceso Servicio				Capa Controlador y Página		Test	Correcciones	Test	Pruebas Pre-Producción		Pruebas Implementación	

**Arquitectura y forma.**

Oracle Forms	Java
<p>Forma Color</p> 	<p>Forma Color</p> 



Arquitectura Modelo

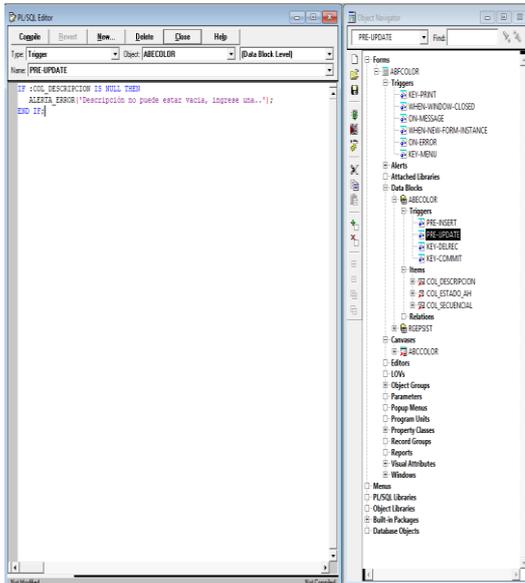


Arquitectura modelo página





### Pre-update



### Acceso Datos

```

import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import ec.gob.egempop.generales.modelo.Color;
import ec.gob.egempop.utilitarios.ManejadorDatosGenerico;

@Stateless
public class ColorAccesoDatos extends ManejadorDatosGenerico<Color> {
    @PersistenceContext(unitName = "bd_traccia_01_jpa")
    EntityManager entityManager;

    public ColorAccesoDatos() {
        super(Color.class);
    }

    @Override
    protected EntityManager getEntityManager() {
        return entityManager;
    }

    public void crearRegistro(Color color) throws Exception {
        create(color);
    }

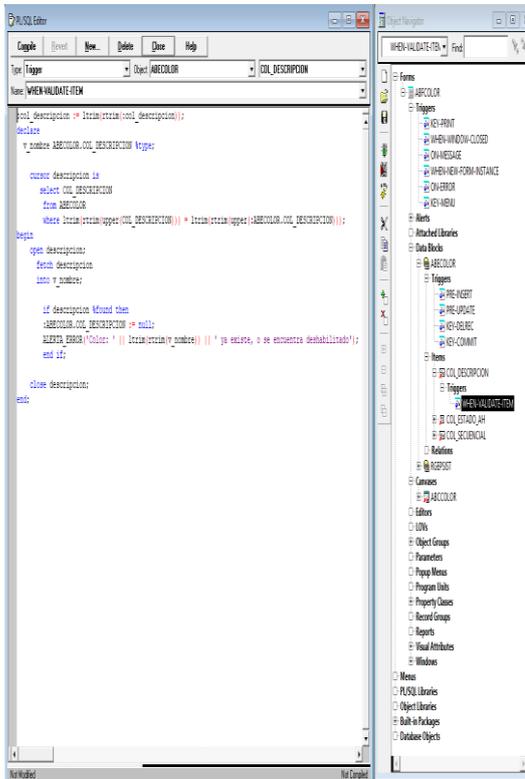
    public void modificarRegistro(Color color) throws Exception {
        edit(color);
    }

    public void eliminarRegistro(Color color) throws Exception {
        color.setCodigo("0");
        edit(color);
    }

    public List<Color> listarColor() throws Exception {
        String consulta = "select c from Color c where c.codigo != '0' order by c.descripcion";
        Query query = this.getEntityManager().createQuery(consulta);
        if (query.getResultList() != null && !query.getResultList().isEmpty()) {
            return query.getResultList();
        }
        return null;
    }

    public Color obtenerColorByCodigo(Integer p_Cod) throws Exception {
        String consulta = "select c from Color c where c.codigo = " +
            p_Cod;
        Query query = this.getEntityManager().createQuery(consulta);
        if (query.getResultList() != null && !query.getResultList().isEmpty()) {
            return (Color) query.getResultList().get(0);
        }
    }
    
```

### Validacion



### Servicio

```

package ec.gob.egempop.generales.servicios;

import java.util.List;

@EJB
private ColorAccesoDatos colorAccesoDatos;

public void crearRegistro(Color p_registro) throws Exception {
    colorAccesoDatos.crearRegistro(p_registro);
}

public void modificarRegistro(Color p_registro) throws Exception {
    colorAccesoDatos.modificarRegistro(p_registro);
}

public void eliminarRegistro(Color p_registro) throws Exception {
    colorAccesoDatos.eliminarRegistro(p_registro);
}

public List<Color> listarColor() throws Exception {
    return colorAccesoDatos.listarColor();
}

public Color obtenerColorByCodigo(Integer p_Cod) throws Exception {
    return colorAccesoDatos.obtenerColorByCodigo(p_Cod);
}

public Long obtenerNumRegistros(String condicion) throws Exception {
    return colorAccesoDatos.obtenerNumRegistros(condicion);
}

public Color obtenerNumRegistrosSecoActiva(String condicion)
    throws Exception {
    return colorAccesoDatos.obtenerNumRegistrosSecoActiva(condicion);
}
    
```



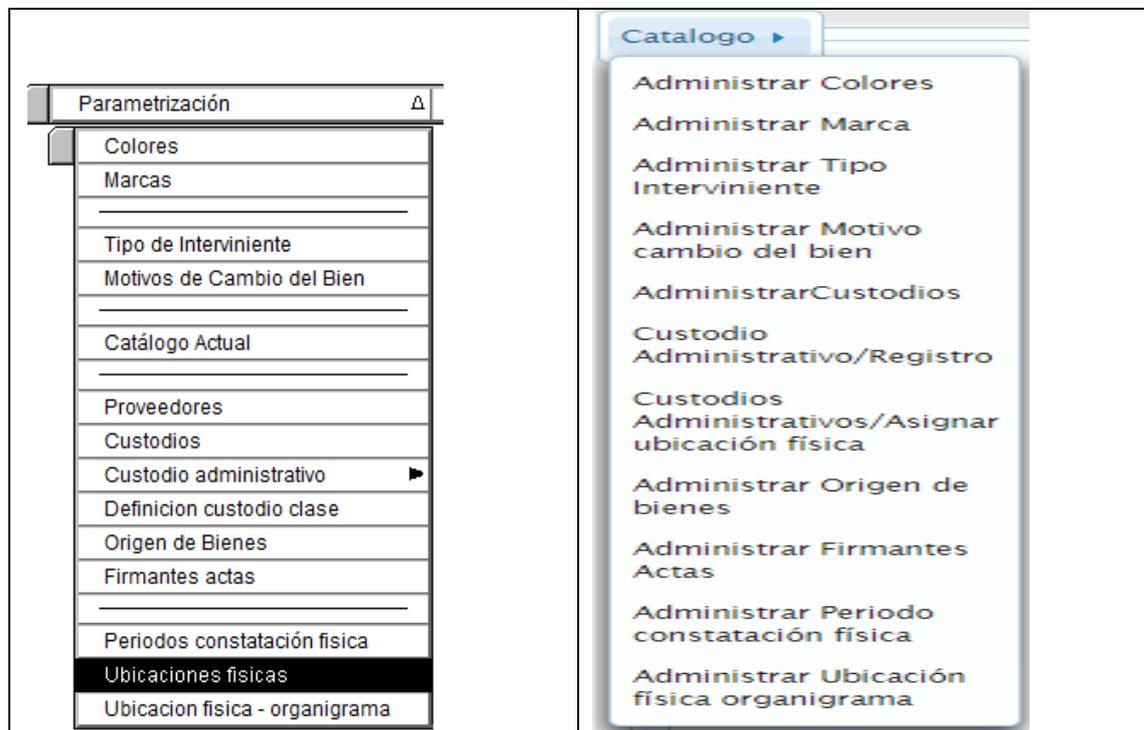


Realizado Por: Ing. Johana Palacios  
 Ambiente de revisión: TEST

Detalles de la aplicación  
 Nombre: Control de Bienes - Bienes  
 Plataforma: Java  
 Base de datos: Oracle  
 Desarrollado Por: Edison Pumasunta

Opciones de revisión

### CATÁLOGOS



### Opción Administrar Colores

Insertar	Correcto
Actualizar	Correcto
Eliminar	Correcto

#### Visualización:

El campo secuencial está muy ancho, ponerle del ancho del texto de la cabecera.

El secuencial debe ir a la derecha (Todo numérico a la derecha).

#### Validación:

Al momento de guardar, si existe un registro ya ingresado y este se encuentra inactivo (“H”), debe preguntar si desea activarle nuevamente y hacerlo si es el caso.

La alerta de Registro existente, debe salir únicamente cuando se trate de registros vigentes.

#### Funcionalidad:

Luego de hacer un filtro desde la columna “Descripción”, se desea agregar un nuevo registro, al momento de guardar, no permite hacerlo, no sale la ventana de confirmación.

\* Descripción: prueba

Activo:

**Guardar** Nuevo

Lista Colores

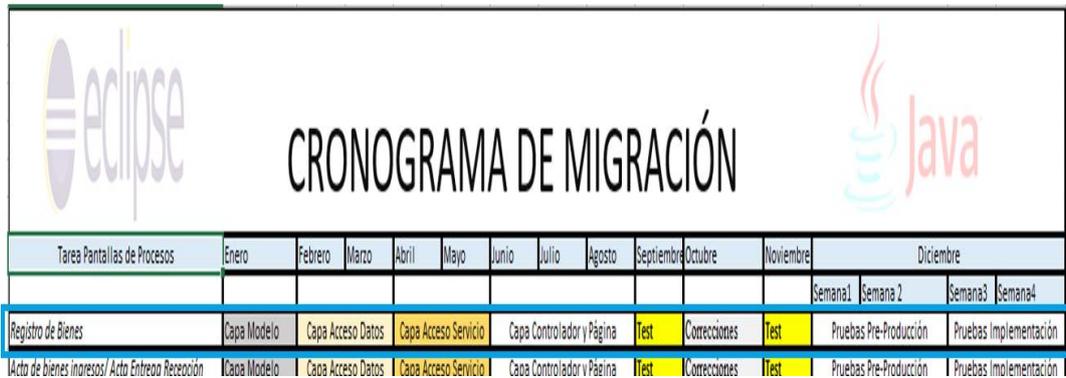
Secuencial	Descripción
111	EDISON
73	EDISON
110	EDISONIIII

### Reportes:

En la parte derecha de la opción debe contener un botón (con ícono de PDF) que permita realizar la impresión del listado vigente, como se visualiza en la imagen siguiente:

Secuencial	Descripción
3	AMARILLO
4	AZUL
5	AZUL CAFE
6	AZUL ELECTRICO
7	AZUL NEGRO
60	AZUL Y ROJO
41	BEIGE
8	BLANCO
9	BLANCO GRI
10	BLANCO ROJ
58	BLANCOVERDE
11	CAFE
12	CANELA
45	CAQUI
13	CARAMELO
14	CELESTE
15	CREMA
16	CROMADO
42	DORADO
17	GRAFITO
18	GRIS
19	HABANO
49	HAYA
57	LADRILLO
20	LILA
21	LILA VERDE

Tomado una forma Registro Bien como muestra del Test plan pruebas



Realizado Por: Ing. Johana Palacios  
Ambiente de revisión: TEST

Detalles de la aplicación  
Nombre: Control de Bienes - Bienes  
Plataforma: Java  
Base de datos: Oracle  
Desarrollado Por: Edison Pumasunta

Opciones de revisión

## PROCESOS

Bienes	Procesos
<ul style="list-style-type: none"> <li>Registro de Bienes</li> <li>Carga Masiva de bienes</li> <li>Modificación de Datos Bienes</li> <li>Custodio administrativo</li> <li>Actualización de Inventario C.A</li> <li>Actas de Bienes</li> <li>Actas Entrega Recepción</li> <li>Actas Usuario Final</li> <li>Baja Masiva de Bienes</li> <li>Habilitar Bienes dados de baja</li> <li>Cambio de Custodios</li> <li>Habilitación de Custodios</li> <li>Depreciación</li> <li>Generación de Bienes en Excel</li> <li>Generación de Vehículos en Excel</li> <li>Certificado de No tener Bienes</li> <li>Constatacion fisica</li> <li>Constatacion vehicular</li> <li>Tramites de Seguro</li> </ul>	<ul style="list-style-type: none"> <li>Registro de Bienes</li> <li>Acta de Bienes Ingresos / Acta Entrega Recepción</li> <li>Acta de Usuario Final</li> <li>Custodios Administrativos/Acta Inicial</li> <li>Custodios Administrativos/Modifica estados bienes en actas</li> <li>Custodios Administrativos/Eliminar bienes en actas</li> <li>Custodios Administrativos/Traslado de bienes</li> <li>Cambio custodio / Por custodio</li> <li>Cambio custodio / Por niop</li> </ul>

## Opción Registro de bienes

Insertar	Correcto
Actualizar	No debe permitir modificar datos de los bienes (Correcto)
Eliminar	No debe hacerlo en esta opción (Correcto)

### Visualización:

Bloque Lista de Bienes:

En el bloque superior, incluir los campos:

Categoría

Clase

Serie

Dependencia

Quitar la columna “Secuencial”

Bloque Datos registro del bien:

Poner un <br/> salto de línea en el medio de los dos paneles

Los campos secuencial y NIM poner más a la izquierda, alinear con los demás campos.

Poner solo dos columnas

El campo Niop debe ser un poco más grande

El campo Nim debe ser un poco más pequeño

Cambiar la etiqueta “Catálogo/Clase” por “Catálogo”

Cambiar la etiqueta “Clase/Tipo Autom” por “Clase”

Las etiquetas de los combos deben estar en español

Fecha de compra más pequeño

Quitar el asterisco (\*) de la base imponible



Colocar bien, el orden de los campos de la base imponible

Quitar el enable (false) de los campos, porque no se visualiza bien la información; si no se digitan debe ser readonly = true.

Colocar los campos del panel, como se visualiza en la imagen:

**DATOS REGISTRO BIEN MUEBLE**

**Datos del Bien**

Secuencial: 37777 NIM: Niop: 111

\* Catalogo: TUMBADORA NIM: 101040050072

Descripcion: TUMBADORA

\* Clase: Liviano \* Categoria: (Activo Fijo)

Caracteristica:

Modelo: Serie:

\* Color: CAMELOS \* Marca: SINCOLOR

\* Estado: Select One \* Situacion: Bueno

**Datos de la compra**

\* Motivo: CARGA INICIAL

\* Origen: EMSAT

\* Proveedor: ABRACOL

\* Fecha compra: 29/05/2019

\* Base Imponible: 12 %: 12 Valor Compra: 1,44

Valor Seguro: Valor a Depreciar: 0,108

Dep Acumulada: Valor Util: 12

\* Garantia(años): 12 \* Vida Util inicial(meses): 12

Vida Util Actual: 12 Valor Actual: 12

**Datos adicionales**

Codigo: Unidad: AREA DE DISEÑO

Custodio: CustodioAdm:

Ubica. fisica: TEXT\_LUGHS\_ORG Depend. UF: TEXT\_DEPEN\_UBIFIS

Observacion:

**Datos de acta**

No. de Acta: TEXT\_NUMACTA

Factura: TEXT\_FACTU

Fecha: TEXT\_FECHASIG

Observaciones de acta:

Quitar los colores rojos y azules de las etiquetas de los campos.

**Validación:**

Niop no puede duplicarse, debe salir un mensaje, que ya existe y con qué descripción.

Descripción debe ser obligatorio

En los combos categoría, clase y estado no debe escogerse “Select One”.

Los siguientes campos deben ser obligatorios:

Característica

Modelo

Serie

Valor de seguro

Porcentaje de IVA debe ser obligatoria

Vida útil debe ser mayor a cero

Base imponible y valor de compra debe ser mayor a cero

**Funcionalidad:**

Valores Iniciales al momento de guardar:

El campo bie\_estado debe iniciarse con el valor de “D”.

El campo bie\_situacion debe iniciarse con el valor de “B”.

El origen tiene que ser por defecto EPMMOP, pero si debe permitir modificarlo.

El porcentaje de IVA debe visualizarse según el select a la tabla RGEPSIST, pero debe ser modificable.

El motivo de compra debe inicializarse según el motivo de bien (mcb\_secuencial = 13), pero debe ser modificable.

El bie\_valdepacum, debe ingresarse con el valor de cero.

**Al momento de guardar:**

El campo “Descripción” si debe ser modificado por el usuario.

Cuando se escoge la marca, color, proveedor, se está borrando lo que se ingresó en los campos “característica”, “modelo” y “serie”.

Cuando se escoge el origen, unidad, motivo, se está borrando lo que se ingresó en los campos “característica”, “modelo”, “serie” y “observación”.

Está calculando mal el valor a depreciar, ese valor se calcula con la siguiente fórmula:

$(\text{Valor de compra} - \text{Valor residual}) / \text{Vida útil}$

Donde:

$\text{Valor residual} = (\text{Valor de compra} * 10) / 100$

Valor de vida útil subir parte del padre a cargar en meses vida útil inicial

No se está registrando los siguientes campos:

Bie\_valcom\_ini (Con el mismo valor de la compra)

Bie\_valdepre\_ini (Con el mismo valor del valor a depreciar)

El campo Categoría, no debe ser un combo, debe ser solo texto, es decir informativo cuando se selecciona un bien.

Cuando se escoge el catálogo y se filtra mediante la casilla de la columna se queda procesando.



El estado también debe ser solo informativo

Los campos vida útil actual y valor actual, son solo informativos, no se debe digitar. El valor útil de la abevaldepre (val\_valutilcal) debe colocarse con el mismo valor de compra.

El campo val\_origreg de la abevaldepre debe registrarse con el valor de “B”

El campo val\_valresid\_orig debe registrarse con el valor residual (val\_valresid).

El campo val\_valdacu\_orig debe registrarse con el valor de depreciación acumulada (val\_valdacu).

El campo val\_valdepre\_orig debe registrarse con el valor de depreciación (val\_valdepre).

El campo val\_valutilcal\_orig debe registrarse con el valor útil (val\_valutilcal).

Cuando la clase es tipo Liviano (L), Pesado (P), Maquinaria (M), Equipo Auxiliar (A), el registro debe guardarse en la tabla AVEVEHICULO.

Los datos de acta, deben visualizarse cuando se consulte un bien.

El nombre de custodio, debe visualizarse cuando se consulte un bien.

El valor de depreciación acumulada visualizada (Dep Acumulada), debe ser un campo transitorio, es decir no de base, debe visualizarse solo al seleccionar un bien. Debe calcularse con la siguiente función:

$VALDEPACUM=ABUVALDEPACUM(bie\_secuencial)$

El valor actual visualizado, debe ser un campo transitorio, es decir no de base, debe visualizarse solo al seleccionar un bien. Debe calcularse con la siguiente condición:

```
if bie_tipo = 'AF' then
    valor actual = bie_val_compra - valdepacum;
else
    valor actual = bie_valor
end if;
```

### Reportes:

En la parte derecha de la opción debe contener un botón (con ícono de PDF) que permita realizar la impresión del bien, como se visualiza en la imagen siguiente:

VOLQUETA		MARCA		NCF	BBB
DESCRIPCIÓN DEL BIEN		INTERNACIONAL		14	62110
Características:	01800 499 68301 14750R2R3R4R780	Tipo:	ACTIVO FUO		
Código Catálogo:	201050290001	Fecha Compra:	31/10/2009		
Catálogo:	VOLQUETA	Color:	BLANCO		
Modelo:	4900 4X2	Valor de Compra:	40,335.30		
Serie:		Estado:	Entregado		
Vida Útil:	0 Meses	Factura:	No existe registrada factura		
Proveedor:	TRANSFERENCIA M D M Q	Acte:	No existe registrada acte		
Dependencia:	Fiscalización Zona Sur	Valor Actual:	4,933.92		
Situación:	Buena	Valor Seguro:	91,463.66		
Origen:	EMMOP-Q	Motivo de Reg:	TRANSFERENCIA		
	HUIZ HUIZ SEGUNDO MELCHOR	RESPONSABLE DEL BIEN	OPERADOR DE DISTRIBUIDORA DE ASPALTO		
	Responsable:	170973391	Cédula		
			Cargo		

(\* ) Bien en Trámite de Seguros

## 4.7.2 Pruebas de rendimiento (Aceptación técnica)

Ingreso al sistema en tipo color: se procede a guardar nuevo color blanco

Catálogo > Procesos > TIPOS DE COLORES

Datos Motivo Movimiento

\* Descripción:

Activo:

Lista Colores (1 of 7)

Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
89	BLANDO
159	BLUE
151	BLUE
45	CAQUI
13	CARAMELOS
160	CARPIO

Tiempo de respuesta en procedimiento es milisimas de segundos iniciando en 972 y culminando en 995 como prueba se verifica.

```

Console Progress Servers Search SVN Repository Browser Debug
JBoss Enterprise Application Platform 6.1+ [JBoss Application Server Startup Configuration] C:\Program Files\Java\jre7\bin\javaw.exe (3/1/2020 9:02:19)
12:00:35,972 INFO [stdout] (http-localhost/127.0.0.1:8080-6) ingresa a guardar
12:00:35,995 INFO [stdout] (http-localhost/127.0.0.1:8080-6) Registro creado==> 164

```

Catalogo > Procesos >

Registro Guardado Correctamente

ADMINISTRACIÓN DE TIPOS DE COLORES

Datos Motivo Movimiento

\* Descripción :

Activo:

Nuevo Guardar 

Lista Colores

(1 of 7) 1 2 3 4 5 6 7 10

Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
164	BLANCO

FORMATO PLAN DE PRUEBAS DE USUARIO			
<b>Identificador</b>	FACAP01	<b>Versión</b>	V01
<b>Responsable</b>	María Belén Flores Brandt		
<b>Nombre del caso de prueba</b>	Administrar Color		
<b>Módulo</b>	Administrar Color		<b>Submódulo</b>
<b>Formulario</b>	bienesEpmop/Catalogos/administrarColor.xhtml		
<b>Descripción de la Prueba</b>	Ingresar descripción del color		
<b>Resultados esperados</b>	Color registrado		
<b>Resultados reales</b>	Registro guardado correctamente		
<b>Error</b>			
<b>Imagen</b>			

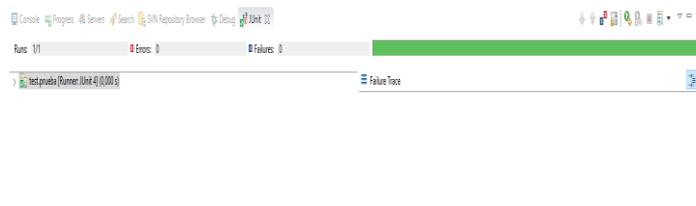
<b>FORMATO PLAN DE PRUEBAS DE USUARIO</b>			
<b>Identificador</b>	FACCON01	<b>Versión</b>	V01
<b>Responsable</b>	María Belén Flores Brandt		
<b>Nombre del caso de prueba</b>	Registro de Bienes		
<b>Módulo</b>			<b>Submódulo</b>
Registro de Bienes			
<b>Formulario</b>			
bienesEpmop/procesos/registroBienes.xhtml			
<b>Descripción de la Prueba</b>			
Datos del Bien: Ingresar Niop Ingresar catalogo bien Ingresar descripción Ingresar característica Ingresar modelo Ingresar serie Ingresar color Ingresar marca Datos de la Compra: Ingresar proveedor Ingresar fecha compra Ingresar base imponible Ingresar garantía en años Datos Adicionales: Ingresar unidad			
<b>Resultados esperados</b>			
Bien registrado			
<b>Resultados reales</b>			
Registro guardado correctamente			
<b>Error</b>			
<b>Imagen</b>			
			

### 4.7.3 Pruebas de carga y estrés (Aceptación técnica)

“JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios, con énfasis en aplicaciones web” (Sandoval, 2014, pág. 24).

```
1 package test;
2
3+ import static org.junit.Assert.*;
10
11 public class prueba {
12     Color color = new Color();
13     @Before
14     public void init() {
15
16         color.setColDescripcion("NEGRO MATE");
17         color.setColEstadoAh("A");
18         color.setColSecuencial(166);
19     }
20
21     @Test
22     public void test() {
23
24         int a=new AdministrarColorControlador().guardarcolor(color);
25         assertEquals(0, a);
26     }
27 }
28
```

## Resultados



### 4.7.4 Plan de implementación

Detallar el plan de implementación con tiempos, herramientas, participantes y otros recursos utilizados, debe incluir lo siguiente, si es que aplica:

Generar el esquema (final) de la BDD

Generar el esquema (final) de la Arquitectura del Sistema

Migración de datos

Gestión de la Infraestructura requerida

### **4.7.5 Requerimientos de implementación**

#### **Software para servidor**

- SGBD con su respectiva versión, esto es un ejemplo.
- Servidor Web, esto es un ejemplo.
- Servidor de aplicaciones, esto es un ejemplo.
- Aplicaciones de terceros (Ms-office, Adobe, WinRar).

#### **Software para usuario**

- Lector de PDF, esto es un ejemplo.
- Navegador web (Chrome, Firefox)

#### **Hardware para servidor**

- RAM al menos de 8Gb, esto es un ejemplo.
- Procesador Core I7, esto es un ejemplo.
- Tarjeta de Red 1Gbps, esto es un ejemplo.

#### **Hardware para usuario**

- RAM al menos de 2Gb, esto es un ejemplo.

### **4.7.6 Manual de usuario**

ANEXO

### **4.7.7 Manual técnico**

ANEXO

### **4.7.8 Plan de capacitación**

Se coloca los recursos, personas involucradas, fechas, resultados de la capacitación, se hace referencia a los manuales o ayudas entregadas.

Implementación- guía.

## CONCLUSIONES

Luego de analizar detalladamente se llegó a comprender una parte del proceso de negocio, la cual permite ver un enfoque basado en el crecimiento de la institución al desarrollo en gran volumen de dependencias en vigilancia de Bienes Muebles, para lo cual se vio necesario la intervención de una migración planificada del sistema Control de Bienes con el fin de llegar a un acceso de control y alcance de todas las dependencias con las que cuenta la EPMMOP.

Mediante un estudio de metodologías de migración analizadas seleccionamos la Metodología MADIISH, la cual contempla modelos de desarrollo ágil, para la cual se tomó en referencia Metodología de desarrollo ágil XP disminuyendo el volumen de documentación, referenciado en la fase tres de la metodología de migración. Evidenciando un resultado favorable en tiempo y logrando un desarrollo oportuno.

La metodología propuesta permitió la consecución del objetivo principal, de disponer el sistema “Control de bienes” migrado en su núcleo.

El proyecto presentado es muestra de desarrollo estructurado en java; Modelo Spring MVC modelo, vista y controlador; con el cual se llevó un aporte tecnológico a la EPMMOP, de tal manera se pretende actualizar todos los sistemas corporativos en un entorno WEB.

Con la migración de los procedimientos y funciones del Sistema Heredado se mejora la portabilidad de la BDD, permitiendo que la nueva aplicación tenga una mejor organización.

## **RECOMENDACIONES**

Para futuras migraciones en módulos anexos y sistemas de la institución se recomienda el documento generado ya que es una metodología basada en un Sistema Heredado enfocado hacia un software abierto en una nueva tendencia tecnológica.

Se recomienda migrar todos los procedimientos y funciones de base de datos de módulos, sistemas de la institución para que la aplicación sea independiente y flexible al cambio o migración de BDD en tiempos futuros.

Se recomienda la generación de una interfaz responsive la cual permitirá adaptarse a cualquier dispositivo donde se pueda visualizar, este aspecto será ancho de pantallas y contenido.

Para futuras migraciones se recomienda utilizar Spring Boot un tipo modelo de software orientada a microservicios, ocupando menos recursos haciendo más ligeras. SOA una arquitectura que integra todos los distintos aplicativos que forman la infraestructura.

## REFERENCIAS BIBLIOGRÁFICAS

*¿Qué es el método inductivo?* (s.f.). Obtenido de concepto.de:  
<https://concepto.de/metodo-inductivo/>

*¿Qué es la investigación cuantitativa?* (s.f.). Obtenido de sisinternational:  
<https://www.sisinternational.com/investigacion-cuantitativa/>

Ciulli. (2007). *esting de migración de aplicaciones distribuidas a entornos Web(Doctoral dissertation, Universidad Nacional de La Plata)*. Plata.

*Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.* (s.f.). Obtenido de Ministerio de Telecomunicaciones y de la Sociedad de la Información:  
<http://software.gobiernoelectronico.gob.ec/slides/slide/codigo-organico-de-la-economia-social-de-los-conocimientos-creatividad-e-innovacion-4>

Díaz Contreras, S. F. (2011). *Eficacia de la aplicación de las normas de seguridad social en la construcción de vivienda de interés social en obras públicas y privadas en el municipio de San Jose de Cúcuta*. Cúcuta.

*Eclipse, entorno de desarrollo integrado.* (s.f.). Obtenido de EcuRed:  
[https://www.ecured.cu/Eclipse,\\_entorno\\_de\\_desarrollo\\_integrado](https://www.ecured.cu/Eclipse,_entorno_de_desarrollo_integrado)

*ENCUESTA.* (s.f.). Obtenido de ELEMENTOS DE UNA REVISTA:  
<http://ubenceg135.blogspot.com/>

León, V. F. (2019). *Carrera de Comunicación Social Investigación periodística sobre migración en proceso de refugio en la ciudad de Cuenca: historias de vida de*

*“Vanesa” y “Juan”(nombres protegidos), inmigrantes venezolanos.* Cuenca-Ecuador.

Madera Villada, A. Y. (2012). *Diseño de componente para el framework zathura code que permita la migración de código fuente de formularios con tecnología oracle forms 6i a componentes con tecnología java enterprise edition.* Santiago de Cali.

*Metodología XP Programación Extrema (Metodología ágil).* (7 de abril de 2018). Obtenido de DIEGO CALVO: <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

Pech-May, F. G.-R.-J. (2010). *Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces.* Mexico.

PLANA, E. I. (2018). *Gestor de publicidad.* Madrid.

Sandoval, A. M. (2014). *Sistema web para la reserva de citas y administración de fichas dentales para la Clínica Dental El Roble, Chillán.* Chillán.

Silva, D. T. (2018). *Una Propuesta de Metodología para la Migración de Sistemas Heredados.* Mexico.

*Sistemas Manejadores de Base de Datos.* (14 de Septiembre de 2009). Obtenido de SlideShare: <https://www.slideshare.net/alinacarrion/smbd-2-1995964>

Torres Gonzales, B. R. (2017). *Aplicación Web para la Gestión de Ventas en el área de marketing en la empresa Zam Marketing Consultora SAC.* Lima-Perú.

Zalazar, A. S. (2014). *Migración de sistemas heredados a cloud computing.* Buenos Aires.

## **ANEXOS**



# ACTA ENTREGA RECEPCION

ADEPROVEEDOR : 1

PRV_SECUENCIAL	NO8	<M>
PRV_NOMBRE	VA80	<M>
PRV_RUC	VA13	
PRV_REPRE	VA40	
PRV_DIRECCION	VA80	
PRV_TELEFONO	VA80	
PRV_FAX	VA30	
PRV_CASILLA	VA40	
PRV_CALIFICACION	A1	<M>
PRV_DIRELECTRO	VA50	
PRV_OBSERVACION	VA200	
PRV_ESTADO_AH	A1	
PRV_PAGWEB	VA120	
PRV_CALIFSN	A1	
PRV_FECREG	DT	
PRV_IMPRIME	A1	
PRV_USUARIO	VA20	
PRV_CLAVE	VA20	
PRV_TIPO	A1	
PRV_USER	A1	
PRV_PROCEDE	VA2	
PRV_RUP	VA15	
PRV_ESTADO	A1	
PRV_FEC_ACT	DT	
PRV_INSCRITO_WEB	A1	
PRV_EMAIL_ALTERNO	VA40	
PRV_BENEFICIARIO	VA50	
PRV_CELULAR	VA30	
PRV_USUMOD	VA30	
PRV_FECHMOD	DT	
PRV_FECHA_INSCRIP	DT	
PRV_NOMBRE_COMER	VA80	

ABECUSTODIO : 1

CUS_SECUENCIAL	NO6	<M>
CUS_CEDULA	A15	<M>
CUS_ESTADO_AH	A1	
CUS_DENDINIA	VA80	
CUS_TIPO	A1	<M>
CUS_CARORIG	N6	
CUS_DEPORIG	N6	

ABEABIENM : 1

IAB_SECUENCIAL	NO6	<M>
IAB_OBSERVACION	VA400	<M>
IAB_ESTADO_AH	A1	<M>
IAB_DEPENDENCIA	VA120	
IAB_CARGO	VA250	
IAB_EXTERNO	VA400	
IAB_BENEFICIARIO	VA120	

ROEDEPEN : 1

DEP_SECUENCIAL	NO6	<M>
DEP_DESCRIPCION	VA80	<M>
DEP_SIGLAS	VA10	<M>
DEP_FECHACRE	DT	<M>
DEP_NUMPLE	I	<M>
DEP_TOTALEMPL	I	
DEP_NVACAN	I	<M>
DEP_TOTALVAC	I	
DEP_GRAFICO	VA80	
DEP_NIVELGEST	A1	<M>
DEP_DIRECCION	VA40	
DEP_TELEFONO	VA20	
DEP_CODANT	VA10	
DEP_FECHAELI	DT	
DEP_CONSIV	A1	
DEP_NIVELIER	I	
DEP_DESPLIEGUE	N3	
DEP_ESTADO_AH	A1	<M>
DEP_TOT_PUESTOS	I	
DEP_TOT_VACANTES	I	
DEP_USUREG	VA20	
DEP_FECHAREG	DT	
DEP_USUMOD	VA20	
DEP_FECHAMOD	DT	
DEP_PARTIDA_RH	VA20	
DEP_VISION	VA200	
DEP_MISION	VA200	
DEP_OBJSIGRAL	VA200	
DEP_CODPARABENES	VA15	
DEP_TIPO	A1	

ROECATCA : 1

CAT_SECUEN	NO6	<M>
CAT_DESCRI	VA80	<M>
CAT_IDCOMP	A1	<M>
CAT_NATURA	VA200	
CAT_ESTADO_AH	A1	
CAT_DESCOR	VA40	
CAT_CARACTE	VA200	
CAT_PUNTAJE	I	<M>
CAT_USUREG	VA20	
CAT_FECHAREG	DT	
CAT_USUMOD	VA20	
CAT_FECHAMOD	DT	
CAT_COD_RH	VA10	
CAT_C_DISTRIB	N4	

FTEENFIN

EFL_SECUENCIAL	N6	<M>
EFL_NOMBRE	VA40	<M>
EFL_TELEFONO	VA12	<M>
EFL_DIRECCION	VA80	<M>
EFL_CALIFICA	A1	
EFL_ESTADO_AH	A1	
EFL_TELEFON2	VA12	
EFL_FAX1	VA12	
EFL_FAX2	VA12	
EFL_WEB	VA80	
EFL_MAIL	VA80	
EFL_CELULAR	VA12	
EFL_CONTACTO	VA80	
EFL_TUNEL	A1	
EFL_CODSUPER	VA12	
EFL_LIQUIDACION	A1	
EFL_CODENT	N4	

ABEAERBIENM : 1

AER_SECUENCIAL	NO6	<M>
AER_NUMERO	VA20	<M>
AER_TIPO	A1	<M>
AER_FECHA	DT	<M>
AER_OBSERVACION	VA400	<M>
AER_FACTURA	VA80	
AER_APROBAR	A1	<M>
AER_MONTO	DC15.2	
AER_TOTALBEN	N6	
AER_ESTADO_AH	A1	<M>
AER_CTRLROPA	A1	
AER_FECCTRLROPA	DT	
AER_USUCTRLROPA	VA20	
AER_DIGITAL	PRC20	
AER_AFECTAINVEN	A1	
AER_NUMADJUDICA	VA20	
AER_REPRE_EXTERNO	VA120	
AER_CARGO_MEMO	VA120	
AER_UNIDAD_MEMO	VA120	
AER_CONTABILIZADA	A1	<M>
AER_FECHACONTAB	DT	
AER_USUSCONTAB	VA20	

ROECATCARP : 2

CRP_SECUENCIAL	NO6	<M>
CRP_DESCRIPCION	VA120	<M>
CRP_ESTADO_AH	A1	

ABETINTERV : 2

TIN_SECUENCIAL	NO2	<M>
TIN_DESCRIPCION	VA40	<M>
TIN_SIGLAS	VA10	<M>
TIN_ESTADO_AH	A1	<M>
TIN_ORDPRINT	N2	

AREDACTA : 2

DAT_SECUENCIAL	NO6	<M>
DAT_TALLA	N3	<M>
DAT_ENTR	N2	<M>
DAT_ESTADO_AH	A1	
DAT_ANO	N4	

ABECODIFACTAS

COD_SECUENCIAL	NO2	<M>
COD_NOMBRE_ACTA	VA80	<M>
COD_DESCRIPCION	VA400	<M>
COD_INICIAL	A1	<M>
COD_CODADM	VA2	<M>
COD_ORDEN	N2	
COD_ESTADO_AH	A1	
COD_TIPOMOV	A1	
COD_GEN_UF	A1	

ABEMCBIE : 1

MCB_SECUENCIAL	NO6	<M>
MCB_DESCRIPCION	VA80	<M>
MCB_TIPO	A1	<M>
MCB_ESTADO_AH	A1	<M>
MCB_ES_FACTURA	A1	<M>

01 DE JUNIO

ESTA TABLA SE CREA PARA CUMPLIR CON LA LISTA DE ACTAS A CREAR , NO TIENE RELACION PORQUE SE CONECTARA CON LA TABLA ABEAERBIEN EN EL CAMPO AER\_TIPO IGUAL A LA INICIAL





## 5.2 Validación y verificación metodológica XP

Esta metodología ágil también se basa en iteraciones (Spring).

Actividades	Spring 1(Análisis y recolección)		Spring 2(Clase)			Spring 3(Acceso)		Spring 4(Servicio)		Spring 5(Controlador y Pagina)						
		Test	Test	Test	Test	Test	Test	Test	Test	Test	Test					
		Porcentaje %			Porcentaje %		Porcentaje %		Porcentaje %	test(%)		Porcentaje %				
• Inventario del Software	X	100			100		100									
• Categorización por Front End y Back End	X	X	100		100		100									
• Definición de base de datos a ser usada.	X	X	100		100		100									
• Definición de orden de migración por precedencia de aplicaciones	X	X	100		100		100									
• Desarrollo de capa modelo y relaciones		0	X	X	X	90	X	100								
• Desarrollo de capa acceso datos para Crud y consultas		0		X	X	20	X	80	X	100						
• Desarrollo de capa Servicio		0		X	X	10	X	50	X	100						
• Desarrollo capa controlador		0				0		0	X	10	X	50	X	X	100	
• Desarrollo de pagina		0				0		0	X	10	X	50	X	X	100	
• Desarrollo de procedimientos(BASE-JAVA)de acuerdo a forma		X	10	X	X	X	11	X	20	X	30	X	70	X	X	100

Spring (2)



# Parametrización(Catalogos y consultas)

Actividades	Spring 1(Análisis y recolección)		Spring 2(Clase)			Spring 3(Acceso)		Spring 4(Servicio)		Spring 5(Controlador y Pagina)			Test			
			Porcentaje %				Porcentaje %				test(%)		Porcentaje %			
• Inventario del Software	X		100				100									
• Categorización por Front End y Back End	X	X	100				100									
• Definición de base de datos a ser usada.	X	X	100				100									
• Definición de orden de migración por precedencia de aplicaciones	X	X	100				100									
• Desarrollo de capa modelo y relaciones			0	X	X	X	90	X	100							
• Desarrollo de capa acceso datos para Crud y consultas			0		X	X	20	X	80	X	100					
• Desarrollo de capa Servicio			0		X	X	10	X	50	X	100					
• Desarrollo de capa modelo Dto(Vista)			0	X	X	X	90	X	100							
• Desarrollo de capa acceso datos consultas (Vista)			0		X	X	20	X	80	X	100					
• Desarrollo de capa Servicio (Vista)			0		X	X	10	X	50	X	100					
• Desarrollo capa controlador			0				0		0	X	10	X	50	X	X	100
• Desarrollo Logica de programación Procesos y Pagina			0				0		0	X	10	X	50	X	X	100
• Desarrollo de pagina			0				0		0	X	10	X	50	X	X	100
• Desarrollo de procedimientos(BASE-JAVA)de acuerdo a forma		X	10	X	X	X	11	X	20	X	30	X	70	X	X	100

## Control de Bienes

**OBJETIVO:** *Obtener un correcto funcionamiento de todas las opciones de la aplicación*

### DESCRIPCIÓN DEL SISTEMA:

*Registro de Bien Mueble que quieran obtener como bien de trabajo o herramienta para el labor diario de los servidores dentro de la EPMMOP y posterior a su verificación por parte de departamento de Bienes (Talento Humano).*

### MÓDULOS DEL SISTEMA

#### CATÁLOGOS

- Administrar Color
- Administrar Marca
- Administrar Tipo Interviniente
- Administrar Motivo Cambio del bien
- Administrar Custodios
- Custodios Administrativos/Registro
- Custodios Administrativos/Asignar Ubicación física
- Administrar Origen de bienes
- Administrar Firmantes Actas
- Administrar Periodo constatación física
- Administrar Ubicación física organigrama

#### PROCESOS

- Registro de Bienes
- Acta de bienes ingresos/ Acta Entrega Recepción
- Acta Usuario Final
- C.A./Acta Inicial
- C.A./Modificar estados bienes en actas
- C.A./eliminar bienes en actas
- C.A./Traslado de bienes
- Cambio custodio/ por custodio
- Cambio custodio/ por Niop

## CONSULTAS

Consulta bienes custodio  
Certificado No tener bienes

## METODOLOGÍA PARA LA APLICACIÓN XP

La metodología general de pruebas a implementar es una metodología de pruebas de interfaz de usuario con pruebas de caja negra, en la cual se verificará el correcto funcionamiento sistema.

Las no conformidades generadas por el proceso de pruebas serán enviadas al DTI, para sus respectivas correcciones. Este formato presenta la información necesaria para que el desarrollador identifique exactamente en donde debe realizar la corrección.

### 5.3 Manual de usuario

#### ADMINISTRAR COLORES

**Objetivo.** – Registrar colores.

**Composición de la pantalla.** – Se compone de dos bloques:

- **Administración:** En este bloque se ingresa la información para ser guardada.

The screenshot shows a form titled "Datos" with a blue header. It contains a text input field for "\* Descripción :", a checkbox for "Activo:" which is checked, and two buttons labeled "Nuevo" and "Guardar". To the right of the buttons is a PDF icon.

- **Lista:** Se visualiza una lista de los registros ingresados, únicamente vigentes.

The screenshot shows a table titled "Lista Colores" with a blue header. The table has two columns: "Secuencial" and "Descripción". The "Descripción" column has a search filter input. The table contains 10 rows of data. Navigation controls are visible at the top and bottom of the table, including page numbers and a dropdown menu.

Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
164	BLANCO
89	BLANDO
159	BLUE
151	BLUE
45	CAQUI
13	CARAMELOS

**Condiciones de registro.** – Existen las siguientes condiciones:

***Pasos para registro:***

- Ingresar la descripción y presionar el botón Guardar.



Datos

\* Descripción : BLANCO HUESO

Activo:

Nuevo Guardar PDF

- Al guardar la información, ésta se visualizará en la lista inferior.

***Modificar un registro:***

- En el bloque inferior, donde se encuentra la lista de registros, hacer clic sobre la fila que contenga el registro a modificar. Al hacerlo la fila se pintará de color amarillo.



Lista Colores

(1 of 7) 1 2 3 4 5 6 7 10

Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
164	BLANCO
165	BLANCO HUESO
89	BLANDO
159	BLUE
151	BLUE
45	CAQUI

(1 of 7) 1 2 3 4 5 6 7 10

- Al hacer clic sobre la fila, en el bloque superior se visualizarán todos los datos que pertenece a ese registro.

Lista Colores	
(1 of 7) [1] [2] [3] [4] [5] [6] [7] [10]	
Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
164	BLANCO
165	BLANCO HUESO
89	BLANDO
159	BLUE
151	BLUE
45	CAQUI

- Modificar los datos deseados y presionar el botón “Guardar”.

**Datos**

\* Descripción :

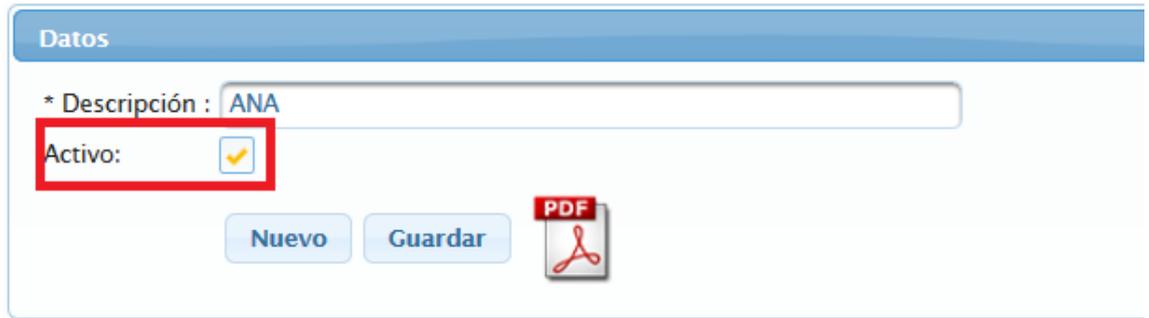
Activo:

***Eliminar un registro:***

- En el bloque inferior, donde se encuentra la lista de registros, hacer clic sobre la fila que contenga el registro a modificar. Al hacerlo la fila se pintará de color amarillo.

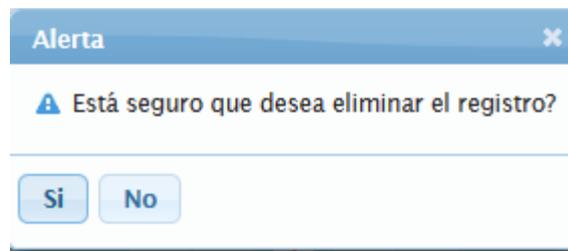
Lista Colores	
(1 of 7) [1] [2] [3] [4] [5] [6] [7] [10]	
Secuencial	Descripción
149	ANA
162	ANADOS
158	ANALUISA
161	ANAUNO
164	BLANCO
165	BLANCO HUESO
89	BLANDO
159	BLUE
151	BLUE
45	CAQUI

- Al hacer clic sobre la fila, en el bloque superior se activará el campo Activo.



The screenshot shows a form titled "Datos" with a blue header. Below the header is a text input field labeled "\* Descripción :" containing the text "ANA". Below this is a checkbox labeled "Activo:" which is checked and highlighted with a red rectangular border. At the bottom of the form are three buttons: "Nuevo", "Guardar", and a PDF icon.

- Al quitar el check del campo "Activo", se visualizará una ventana de confirmación, luego presionar Si, si desea eliminar el registro.



The screenshot shows a confirmation dialog box titled "Alerta" with a close button (X) in the top right corner. The main text inside the dialog is "Está seguro que desea eliminar el registro?". At the bottom of the dialog are two buttons: "Si" and "No".

## 5.4 Manual Técnico

### Tabla

Tabla 1. Estructura de la tabla ABECOLOR en la base de datos ORACLE				
<b>Descripción:</b> <i>Contiene los colores, los mismo están almacenados en alfresco.</i> <b>Esquema:</b> emop_oracle				
CAMPO	TYPE	NULO	DEFAULT	COMENTARIOS
COL_SECUENCIAL	NUMBER(5)	N		
COL_DESCRIPCION	NUMBER(10)	Y		
COL_ESTADO_AH	VARCHAR2(20)	N		

### Mapa de navegación

Datos

\* Descripción :

Activo:



Guardar: permite registrar el nuevo registro o modificar el registro existente con los cambios realizados.

### Descripción Gráfica del mapa de navegación

Opción 1 Administrar Personal Talento Humano		
<b>Descripción:</b> <i>Registrar la lista de analistas que pueden solicitar un expediente.</i>		
Campos	Obligatorio	Funcionalidad
<i>Descripción</i>	<i>SI</i>	<i>Descripción del color</i>
<i>Estado</i>	<i>Predeterminado</i>	<i>Si es activo</i>
<i>Guardar</i>		<i>Permite guardar la información ingresada</i>

## AdministrarColorControlador

No	Método	Descripción	Utilizado en
1	Inicializar()	Inicialización de la pantalla	Ejecución de la pantalla

```
public void inicializar() {
    try {
        administrarSesionControlador = (AdministrarSesionControlador) obtenerAtributoSesi;
        color.setEstado(true);
        colorList = colorServicio.listarColor();

    } catch (Exception ex) {
        Logger.getLogger(AdministrarColorControlador.class.getName()).log(
            Level.SEVERE, null, ex);
        ponerMensajeError("OCURRIO UN ERROR AL HACER LA CONSULTA");
    }
}
```

2	Guardar()	Guarda la información de un registro	Guarda información
---	-----------	--------------------------------------	--------------------

```
public void guardar() {
    try {
        System.out.println("ingresa a guardar");
        sebuscadoActivar = 0;
        if (!validarInformacion()) {
            return;
        } else {

            if (color.getColSecuencial() == null
                || color.getColSecuencial() == 0) {
                color.setEstado(true);
                color.setColDescripcion(color.getColDescripcion()
                    .toUpperCase().trim());

                if (colorServicio
                    .obtenerNumRegistros(" where p.colDescripcion = '"
                        + color.getColDescripcion() + "'"
                        + "and p.colEstadoAh='A'" ) > 0) {
                    ponerMensajeAlerta("Registro existente con la misma descripcion estado Activo");
                    sebuscadoActivar = 1;
                } else if (colorServicio
                    .obtenerNumRegistros(" where p.colDescripcion = '"
                        + color.getColDescripcion() + "'"
                        + "and p.colEstadoAh='H'" ) > 0) {

                    Color colorbuscadoActivar = new Color();
                    colorbuscadoActivar = colorServicio
                        .obtenerNumRegistrosSecActivar(color
                            .getColDescripcion());
                    sebuscadoActivar = colorbuscadoActivar
                        .getColSecuencial();
                    System.out
                        .println("Secuencial => " + sebuscadoActivar);

                    RequestContext rc = RequestContext.getCurrentInstance();
                    rc.execute("confirmacion1.show()");
                }
            }
            if (sebuscadoActivar == 0 || sebuscadoActivar == null) {

                colorServicio.crearRegistro(color);
                ponerMensajeInfo(MENSAJEGUARDADO);

                System.out.println("Registro creado==> "
                    + color.getColSecuencial());
            }
        }
    }
}
```

3	Modificar()	Modifica información de un registro	Modificar información
---	-------------	-------------------------------------	-----------------------

```

    } else {
        color.setEstado(true);
        color.setColDescripcion(color.getColDescripcion()
            .toUpperCase().trim());

        if (colorServicio
            .obtenerNumRegistros(" where p.colDescripcion = '"
                + color.getColDescripcion() + "'"
                + "and p.colEstado=1") > 0) {
            ponerMensajeAlerta("Registro existente con la misma descripcion estado Activo para modificar");
            sebuscadoActivar = 1;
        } else {

            color.setColDescripcion(color.getColDescripcion()
                .toUpperCase().trim());
            colorServicio.modificarRegistro(color);
            ponerMensajeInfo(MENSAJEMODIFICADO);
        }
    }

    nuevo();
}
} catch (Exception e) {
    Logger.getLogger(AdministrarColorControlador.class.getName()).log(
        Level.SEVERE, null, e);
}
}
}

```

- 4 **SeleccionarColor()** Carga la información de los colores. Ejecuta la lista de los colores.

```

public void seleccionar(SelectEvent event) {
    color = colorSeleccionado;
}

```

- 5 **Eliminar()** Cambia de estado un registro Cambia estado registro

```

public void eliminar() {
    try {
        colorServicio.eliminarRegistro(colorSeleccionado);
        ponerMensajeInfo(MENSAJEELIMINADO);
        nuevo();
    } catch (Exception ex) {
        Logger.getLogger(AdministrarColorControlador.class.getName()).log(
            Level.SEVERE, null, ex);
    }
}

```

- 6 **Validar Información()** Valida Registro no ingreso Campo nulo Guardar Registros

```

public boolean validarInformacion() {
    boolean validado = true;

    if (color.getColDescripcion().equals(null)
        || color.getColDescripcion() == "") {
        ponerMensajeAlerta("!!! ingreso descripcion del color");
        validado = false;
    }
    return validado;
}

```

- 7 **Activar Color ()** Permite realizar la activación del color que se encontraba en estado inactivo. En guardar nuevo registro si ya existe.

```

public void ActivarColor() {
    try {
        Color coloractivar = new Color();
        coloractivar = colorServicio
            .obtenerColorByCodigo(secbuscadoActivar);

        coloractivar.setColEstadoAh("A");
        colorServicio.modificarRegistro(coloractivar);
        ponerMensajeInfo(MENSAJEMODIFICADO);

        colorList = colorServicio.listarColor();

    } catch (Exception e) {
        Logger.getLogger(AdministrarColorControlador.class.getName()).log(
            Level.SEVERE, null, e);
    }
}

```

8 Imprimir () Documento de Muestra colores  
 colores estado ingresados  
 activo

```

public void imprimir() {
    try {
        System.out.println("solicitud imprimir Colores ");

        reporteControlador = new ReportesBienesControlador();
        reporteControlador.imprimirColores();
    } catch (Exception ex) {
        ponerMensajeError("Ocurrio un error al imprimir el reporte");
        Logger.getLogger(AdministrarColorControlador.class.getName()).log(
            Level.SEVERE, null, ex);
    }
}

```

