**TABLA DE CONTENIDO**

1. **OBJETO**
2. **ALCANCE**
3. **RESPONSABLE**
4. **DOCUMENTOS DE REFERENCIA**
5. **DEFINICIONES**
6. **CONDICIONES GENERALES**
7. **ESTANDARES PARA EL DESARROLLO DE SOFTWARE EN TRANSMILENIO S.A**
8. **TABLA DE FORMATOS**

**MODIFICACIONES**

| **VERSIÓN** | **FECHA** | **CAMBIO** | **SOLICITÓ** |
| --- | --- | --- | --- |
| 0 | 2017-06-07 | Primera versión oficial del documento | N.A. |

1. **OBJETO**

Este documento tiene como objetivo establecer los estándares de desarrollo de software en TRANSMILENIO S.A., que se deben incluir como obligación en los estudios técnicos y económicos cuando las diferentes áreas de la empresa requieran contratar empresas de desarrollo de software o ingenieros de diferentes disciplinas para la construcción de herramientas de software que necesiten.

1. **ALCANCE**

El alcance de este protocolo es de conocimiento y aplicación de todas las Dependencias de TRANSMILENIO S.A., que en cumplimiento de la misión de la entidad, demanden desarrollo de software o funcionalidades a la medida.

1. **RESPONSABLE**

El responsable por la elaboración, implementación, mantenimiento de este procedimiento es el Profesional Especializado Grado 06. Por su cumplimiento es responsable el (la) Director(a) Técnico(a) de TIC’s.

La revisión y/o actualización de este documento deberá realizarse cuando se considere pertinente por parte de los responsables de su aplicación y cumplimiento.

1. **DOCUMENTOS DE REFERENCIA**

* Estándar SQL ANSI 92
* Metodología de Immon
* MVC Patrón de Modelo-Controlador-Vista
* GoF The Gang of Four, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides autores del libro Design Patterns: Elements of Reusable Object-Oriented Software

1. **DEFINICIONES**

**Diseño del GoF:** Patrones de Diseño que describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos.

**ETL (Extract, Transform and Load** - **extraer, transformar y cargar**)**:** proceso que permite mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

**Filtro:** Es una expresión lógica que restringe el conjunto de datos a trabajar

**Hard Code:** término que hace referencia a una mala práctica en el desarrollo de software que consiste en incrustar datos directamente (a fuego) en el código fuente del programa, en lugar de obtener esos datos de una fuente externa como un fichero de configuración o parámetros de la línea de comandos, o un archivo de recursos.

**JIRA:** Herramienta que permite gestionar todo el proceso de desarrollo de software, desde la planificación hasta el mantenimiento post-venta.

**La base:** Es el conjunto de unidades de análisis que se procesa para construir un indicador; la base es el resultado de aplicar las expresiones lógicas de filtro y repetición al universo de casos en el banco de datos.

**MVC: E**stilo de arquitectura de software, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. La Vista, o interfaz de usuario, **que** compone la información **que** se envía al cliente y los mecanismos interacción con éste.

**OLAP:** Es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing) y consiste en una solución utilizada en Inteligencia de negocios (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

**Patrón Caché:** Patrón de diseño que aporta soluciones a problemas típicos y recurrentes a la hora de desarrollar software. Consiste en cargar datos bajo demanda en un cache desde una fuente de almacenamiento y ayuda a mantener la consistencia entre los datos y el caché y la fuente original.

**Patrón de Inyección:** Patrón para implementar partes de los desarrollos de software. Los patrones proponen soluciones estándares a los problemas comunes de diseño de los sistemas de información.

**Repeticiones:** Son expresiones lógicas que se definen en conjunto y que también restringen el conjunto de datos que conforma la base; es normal pensar que una repetición se exprese como “para cada X”, por ejemplo, para cada estación, para cada ruta, para cada concesionario.

**SLACK: P**lataforma de trabajo; un área de trabajo digital que centraliza las comunicaciones de la organización y junto con otros recursos, humanos y técnicos, permite llevar a cabo los proyectos.

**Tecnología Java EE, JPA, JSF, JMS:** plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java.

**Tercera Forma Normal:** Modelo de organización del diseño de la Base de Datos Relacional en el cual se han eliminado las redundancias de información evitando las anomalías de actualización, inserción y borrado.

**TOGAF:** Herramienta para asistir en la aceptación, producción, uso y mantenimiento de arquitecturas empresariales basándose en un modelo de proceso iterativo soportado por buenas prácticas y un conjunto reusable de activos arquitecturales existentes.

**UML (lenguaje unificado de modelado**)**:** Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

**Unidad de análisis o unidad de observación:** Es la unidad de información que contiene las variables que se utilizarán para agregar o contar en la producción de cada indicador. La producción de un indicador puede requerir del procesamiento de múltiples tipos de unidades de análisis.

1. **CONDICIONES GENERALES.**

El desarrollo y el soporte de aplicaciones computarizadas es una labor compleja que involucra la participación activa de distintas áreas del negocio, incluyendo usuarios, personal técnico de sistemas y contratistas.

Hoy la empresa no tiene en su Dirección de TICs un equipo de planta para el desarrollo y mantenimiento de software; por esta razón las diferentes áreas de la institución han tenido que contratar la realización de desarrollos, ya sea con empresas de desarrollo, o con ingenieros de diferentes disciplinas para la construcción de las herramientas que requieren.

Paralelamente la Dirección de Tics adelanta el desarrollo del plan estratégico de sistemas el cual sigue la metodología TOGAF de arquitectura empresarial, a través del cual se han identificado la necesidad de desarrollar un conjunto importante de sistemas con fin automatizar e integrar los procesos de la institución.

Dada la variedad de sistemas y de equipos de trabajo se impone el establecimiento de estándares que deberán ser utilizados por los desarrolladores y líderes de proyectos.

El presente documento no pretende mostrar un producto terminado ya que a medida que evoluciona la tecnología de sistemas, aparecerán nuevas herramientas que apoyarán en mayor o menor grado las labores relacionadas con el desarrollo y el soporte de aplicaciones. Para cada una de estas nuevas herramientas se deberá evaluar su impacto en las actividades de administración y desarrollo de proyectos antes de ser utilizadas.

**6.1 PROBLEMAS DE NO ESTANDARIZAR**

* El trabajo sin estándares genera múltiples problemas
* El software se vuelve artístico, esto es, es imposible entender y mantener el código si no se conoce a su autor.
* El mantenimiento del conjunto de sistemas se vuelve imposible para mantener por un solo grupo de ingenieros pues la excesiva variedad de disciplinas dificulta enormemente el reclutamiento y mantenimiento de los ingenieros de soporte.
* El descanso o salida de un ingeniero hace imposible el mantenimiento del software a su cargo.
* La arquitectura de software se dispersa, existiendo diseños distintos para cada problema.

**6.2. OPORTUNIDADES DE LA ESTANDARIZACIÓN**

* Cuando se estandariza el método y herramientas de construcción de software se obtienen múltiples ventajas.
* El software se vuelve impersonal. Cualquier ingeniero que comparta los métodos y herramientas de software puede leer y en corto tiempo mantener las aplicaciones.
* El software evoluciona con facilidad. La madurez de las arquitecturas y métodos estandarizados permite evolucionar más fácilmente las aplicaciones pues todas aprovechan las mejoras de los frameworks y técnicas de trabajo compartidas.
* Puede pensarse en que un solo grupo cohesivo de ingenieros realicen el mantenimiento de las aplicaciones de la institución.
* El software sobrevive al ingeniero. El ascenso, descanso, y salida de un ingeniero no acaban con las aplicaciones que construyó pues su código es entendible y mantenible. El ingeniero que lo remplace tendrá una curva de aprendizaje de menor duración que la que tendría si no se utilizaran estándares.
* Se puede compartir código. Existen problemas comunes para cada aplicación, por ejemplo, el manejo de la autorización, la construcción de logs, la creación de unidades de análisis, el manejo de tablas básicas y tablas de parámetros, etc. Al estandarizar el código, esta funcionalidad solo se programa una vez, y las siguientes aplicaciones pueden hacer uso de esa funcionalidad sin tener que volver a programar y probar.

**6.3 ESTABLECIMIENTO DE ESTÁNDARES**

El concepto de estándar implica una restricción en las posibilidades disponibles para la escogencia.

Los estándares se establecen con múltiples criterios y para su establecimiento en TRANSMILENIO S.A. se han utilizado los siguientes criterios:

1. Mantenimiento del software producido

2. Adaptación al problema a resolver

3. Seguridad del software y de su código fuente

El criterio de mantenimiento establece que para mantener el software construido y entregado a TRANSMILENIO para su operación debe ser posible identificar sin dificultad el recurso humano y tecnológico requerido para realizar los cambios que requiera el software durante su vida útil.

El criterio de adaptación al problema establece que las herramientas de desarrollo que se utilicen para la construcción de un software deben ser adecuadas para resolver el tipo de problema que requiere atención.

El criterio de seguridad del software y de su código fuente establece que los desarrollos que se realicen y que involucren la custodia de cualquier tipo de activo de la institución deben tener protección de seguridad para asegurar la confidencialidad, integridad, y durabilidad de la información.

1. **ESTANDARES PARA EL DESARROLLO DE SOFTWARE EN TRANSMILENIO S.A.**

**7.1 ESTÁNDARES GENERALES APLICABLES A TODO TIPO DE SISTEMAS USO DEL REPOSITORIO DE VERSIONES**

* Todo desarrollo debe registrarse en el repositorio de versiones de código institucional
* Toda actualización al repositorio (update/ commit) debe ir precedida de una operación de actualización del área de trabajo local (pull/ update).
* No debe realizarse nunca una actualización al repositorio que no haya sido probada de antemano.
* Toda línea nueva de desarrollo realizada sobre código existente debe realizarse como una nueva rama del proyecto en el repositorio.
* Se debe incluir, al momento de actualizar algún artefacto en el repositorio, un breve comentario con la justificación de dicha actualización.
  + 1. **COMUNICACIÓN**
* La comunicación en los equipos de trabajo debe ser constante para solucionar las dudas o problemas de manera rápida y eficaz. Para esto puede utilizarse la herramienta Slack.
* El control de ejecución de tareas debe ser gestionado a través de un software de control de desarrollo. Para esto puede utilizarse la herramienta JIRA que facilita además el seguimiento de errores e incidentes y tener un control operativo sobre el software.

**7.2 ESTÁNDARES PARA LA CONSTRUCCIÓN DE SISTEMAS TRANSACCIONALES**

Se define como sistema transaccional todo aquel sistema orientado a apoyar procesos que

* Implican flujo de información,
* Mantienen del estado del sistema en una base de datos,
* Son utilizados simultáneamente por múltiples usuarios.
* Sus casos de uso de especificación reflejan una interacción entre el usuario y el sistema para lograr el objetivo deseado

**7.2.1 ESPECIFICACIÓN FUNCIONAL**

Debe construirse un documento de especificación que documente

* El inventario de casos de uso del sistema,
* El producto de la ejecución de cada caso de uso
* La interacción entre el usuario y el sistema.

El documento de especificación debe ser suficiente para que el ingeniero que diseñe la solución realice su trabajo sin conocer más información que la de dicho documento.

**7.2.2 DISEÑO**

El diseño debe estar orientado a maximizar la cohesión y minimizar el acoplamiento del código que se construya

* Debe utilizarse el lenguaje UML
* Debe utilizarse una arquitectura de software de múltiples niveles.
* Considerar la utilización del patrón MVC cuando sea procedente
* Deben utilizarse los patrones básicos de diseño del GoF
* Debe utilizarse el patrón de inyección
* Debe evitarse en lo posible la utilización del patrón caché

**7.2.3 DESARROLLO**

* Deben utilizarse las tecnologías asociadas al lenguaje Java
* Tener en cuenta los tiempos de respuesta de la ejecución de los casos de uso de sistemas de ejecución en línea no debe ser superior a tres (3) segundos.

**7.2.4 TÉCNICAS DE PROGRAMACIÓN**

* El código debe segmentarse de forma que todo método quepa en una sola pantalla
* Todo método público debe verificar sus parámetros de entrada
* Debe utilizarse la tecnología Java EE, JPA, JSF, JMS
* Debe utilizarse una programación orientada a servicios
* Debe utilizarse una herramienta para la definición y ejecución de las pruebas unitarias.
* Debe utilizarse una herramienta para la definición y ejecución de las pruebas de carga.
* Los valores de configuración del sistema o valores específicos asociados a reglas de negocio no deben quedar estáticos en el código fuente. En otras palabras, no debe programarse con “hard code”.

**7.2.5 BASE DE DATOS**

* Debe utilizarse un motor de base de datos relacional que cumpla el estándar SQL ANSI 92.
* Una tabla no puede ser accedida desde múltiples módulos del sistema. Para esto la tabla debe ser poseída por un solo módulo, y este módulo debe proveer servicios públicos que implanten las operaciones de negocio que requieran la actualización o consulta de la tabla. Solo el módulo dueño conoce la estructura de la tabla; los demás módulos solo conocen los servicios provistos por dicho módulo.
* La estructura de la base de datos debe cumplir con la tercera forma normal de Edgar F. Codd.
* Las tablas y las columnas de dichas tablas deben tener el respectivo script de comentarios.
* Lo nombres de las llaves foráneas, llaves primarias e índices de cualquier tipo que se requieran deben ser legibles y no autogenerados por el motor de base de datos.

**7.3 ESTÁNDARES PARA LA CONSTRUCCIÓN DE SISTEMAS ESTADÍSTICOS**

Se define un sistema estadístico como aquel sistema orientado a

* Acumular información resultado de la ejecución de procesos en el tiempo
* Clasificar la información de acuerdo con criterios espaciales, temporales, y temáticos
* Procesar la información histórica y producir indicadores estadísticos
* Servir como base para estimar modelos de estadística descriptiva y predictiva
* Servir como base para ejecutar procesos de control estadístico de calidad

**7.3.1 ESPECIFICACIÓN FUNCIONAL**

Debe construirse un documento de especificación que documente

* El inventario de fuentes de información del sistema y la información que se requiere de cada fuente
* El inventario de indicadores estadísticos a producir
* Las unidades de análisis requeridas para producir los indicadores.
* El documento de especificación debe ser suficiente para que el ingeniero que diseñe la solución realice su trabajo sin conocer más información que la de dicho documento. Al analizar los indicadores y documentarlos, considerar la organización de los indicadores en forma jerárquica de acuerdo con los temas que se desee estudiar o procesos a apoyar.

**7.3.2 DISEÑO**

* Debe utilizarse la metodología de Immon para el diseño de la bodega de datos que almacenará la información del sistema. Esta metodología establece que se debe tener un área de trabajo (staging area) en donde se conforme una base de datos relacional en tercera forma normal, en donde se almacene la información todas las unidades de análisis a observar.
* De esta base de datos previa se selecciona la información requerida para los cubos requeridos para consulta estadística por medio de un OLAP.
* De esta base de datos se seleccionan los subconjuntos de información requerida para correr modelos estadísticos disponibles en sistemas estadísticos diferentes a un OLAP.
* Adicionalmente, es necesario representar visualmente el proceso de ETL a través de un diagrama de tal manera que quede claro dicho proceso.

**7.3.3 DESARROLLO**

Considérese las etapas estándar en la construcción del sistema estadístico

**7.3.3.1 Extracción, transformación, y carga (ETL)**

* Los procesos de extracción y transformación normalmente requieren de la implementación de múltiples árboles de decisión, y de la consideración casos excepcionales en el rango de las variables a trabajar.

Por esta razón, en cuanto sea posible, utilícese la herramienta de ETL que acompaña a la herramienta estándar de bodega de datos y OLAP seleccionada como estándar. Esta herramienta ya apoya la implementación de árboles de decisión y la construcción de rangos de las variables con casos excepcionales.

Cuando no sea posible utilizar la herramienta de ETL, utilice el lenguaje Java para realizar la tarea de ETL.

* En todo caso, cree métodos de ETL pequeños que puedan ser verificados fácilmente.

**7.3.3.2 Conformación del banco de explotación**

* El proceso de ETL debe alimentar la base de datos relacional en tercera forma normal prescrita por la metodología de Immon. Coloque a las tablas de la base nombres descriptivos del concepto que representan. Coloque a las columnas de las tablas nombres descriptivos de la información.
* Conozca y documente el rango de variación y los valores excepcionales de cada variable.
* Tomando como entradas la base de datos en tercera forma normal, construya los cubos de explotación que requiera creando los hechos y dimensiones que requiera según el modelo de bodega que considere apropiado (estrella, copos de nieve, etc). Utilice para esto las herramientas de inteligencia de negocio de Qlik.
* Cuando su problema sea construir y estimar modelos estadísticos no provistos por el OLAP, utilice la base de datos en tercera forma normal para seleccionar, aparear, y construir las unidades de análisis que requiera para la estimación de los modelos con otra herramienta, tal como SAS, SPSS, STATA, R.

**7.3.3.3 Explotación del banco de información**

* Defina claramente cuáles son los filtros y repeticiones que conforman la base de producción de cada indicador.
* Construir los indicadores que requiera para resolver el problema bajo análisis.
* Guardar en el repositorio el código fuente de los programas de ETL con nomenclatura y comentarios descriptivos de su función.
* Guardar en el repositorio el código fuente de los programas de producción de indicadores con nomenclatura y comentarios descriptivos de su información de resultado.
* Guardar en el repositorio el código fuente de todos los programas de estimación de modelos realizados.

**7.4 ESTÁNDARES PARA LA CONSTRUCCIÓN DE SISTEMAS DOCUMENTALES**

Se define un sistema documental como aquel sistema que documenta las actuaciones de la institución y de los terceros de la institución con respecto a los asuntos de competencia de la institución.

La unidad de trabajo de un sistema documental es el documento definido como un mensaje fijado a un medio.

La institución en el ejercicio de su actividad genera documentos en muchos frentes. Por ejemplo, documentos para pago de facturas, documentos para ejecución de presupuesto, documentos para planeación, actas, documentos de inspección de conductores y vehículos, documentos de interventoría de operaciones.

Los documentos generados deben ser clasificados y organizados en expedientes que den testimonio de todas las actuaciones relacionadas con un asunto. Existen normas expedidas por el Archivo General de la Nación con respecto a la organización de los documentos en archivos de Gestión, Central, e Histórico y a las acciones que se deben realizar para garantizar su utilización y preservación.

La enorme mayoría de los documentos tiene restricciones de acceso a sus usuarios.

Al implementar sistemas de naturaleza documental deben seguirse los siguientes estándares en la:

**7.4.1 PRODUCCIÓN DE LOS DOCUMENTOS**

* Cuando se diseñe un nuevo documento debe involucrarse al área de Planeación de la entidad para asociar los nuevos documentos con los las oficinas productoras y procesos que los generan o tramitan.
* Debe involucrarse al área de Archivo en el diseño de nuevos documentos para asegurar que contienen todos los metadatos necesarios.
* Solo se deben construir sistemas de naturaleza documental en la fase de archivo de gestión. La fase de archivo central y de histórico deben ser manejadas por el sistema de gestión documental de la institución.

**7.4.2 CONSERVACIÓN DE LOS DOCUMENTOS**

* Todo documento debe ser clasificado tanto con los clasificadores normativos (según tabla de retención documental), como con los clasificadores de proceso.
* Todo documento debe ser incluido en un expediente de acuerdo con el asunto al que pertenece
* Todo documento debe ser recuperable
* Los índices de expedientes deben registrar todas las operaciones realizadas sobre el expediente.
* La integridad de todo índice de expedientes debe estar protegida contra alteración no autorizada.
* Todo sistema de naturaleza documental que se construya debe proveer búsquedas según texto libre, búsqueda según clasificadores, búsqueda según palabras clave, y búsqueda de expedientes.

**7.4.3 EVALUACIÓN Y DISPOSICIÓN DE LOS DOCUMENTOS**

* Todo documento de un sistema documental debe tener su clasificación según las tablas de retención documental aprobadas por la institución.
* Todo sistema de naturaleza documental debe implementar alertas sobre cuándo se cumplen los tiempos de retención del documento en la fase de archivo en que se encuentra.

**7.4.4 INTEGRACIÓN CON EL SISTEMA DE GESTIÓN DOCUMENTAL DE LA ENTIDAD**

Todo sistema de naturaleza documental que se construya debe tener una interfaz automática con el sistema de gestión documental de la institución para transferir al sistema los expedientes que ya cumplieron su etapa de gestión y deben pasar a la etapa de archivo central (intermedio).

**7.4.5 DISEÑO DE SISTEMAS DE NATURALEZA DOCUMENTAL**

* Utilizar la notación BPMN2 para describir los procesos requeridos por las áreas de negocio de la entidad que generan los documentos.
* Desarrollo de sistemas de naturaleza documental
  + 1. **CONSTRUCCIÓN DE SISTEMAS DE NATURALEZA DOCUMENTAL**
* En lo posible utilizar la herramienta de gestión documental y de gestión de procesos adquirida por TRANSMILENIO.
* Para las demás tareas utilizar tecnologías Java JEE
* Tener en cuenta los tiempos de respuesta tanto para las consultas de documentos entre periodos de tiempo y para las operaciones de gestión documental masivas, ya el tiempo no debe ser superior a tres (3) segundos.

**7.5 ESTÁNDARES PARA LA CONSTRUCCIÓN DE SISTEMAS ESPACIALES**

Se define un sistema espacial como aquel sistema que contiene componentes asociados con coordenadas geográficas.

Se distinguen dos tipos de sistemas

* Sistemas cuya salida es geográfica
* Sistemas cuyo estado es geográfico

**7.5.1 SISTEMAS CUYA SALIDA ES GEOGRÁFICA**

Son aquellos sistemas en donde se desea presentar sus salidas en forma de mapas geográficos, por ejemplo para visualización. Sin embargo el sistema puede corresponder a cualquier otro tipo de sistemas. Ejemplos de estos sistemas son la visualización geográfica de estadísticas, la visualización de eventos, etc.

La empresa está trabajando al presente en la estandarización de las herramientas de presentación geográfica que se utilizan, pero aún no existe una decisión sobre una herramienta estándar a utilizar.

Para la construcción de los sistemas debe utilizarse los estándares correspondientes a la naturaleza de los sistemas que generan la información. Por ejemplo, si el sistema que mantiene el estado y genera la información es transaccional, deben respetarse los estándares de correspondientes a sistemas transaccionales. Si por el contrario, los sistemas son de naturaleza estadística, deben respetarse los estándares correspondientes a los sistemas estadísticos.

**7.5.2 SISTEMAS CUYO ESTADO ES GEOGRÁFICO**

Son los sistemas conocidos como sistemas de información geográfica (SIG), compuestos por diferentes capas y cada capa se comporta como un subsistema aparte, con componentes diferentes y reglas diferentes.

Por ejemplo, una capa puede tener solo cartografía básica, otra capa contiene vías, otra contiene diseño de rutas, etc.

Cada capa puede tener reglas de negocio diferentes implantadas como subsistemas diferentes. Por ejemplo, una capa puede contener infraestructura con componentes de paraderos y estaciones, y reglas de negocio mantenimiento de planta, aseo, seguridad de acceso y negocios instalados. Otra capa puede tener rutas, con componentes de vías, paraderos, portales, con reglas de negocio de horarios, inventario de flota, etc.

El sistema estándar para implantar los sistemas cuyo estado es geográfico es el sistema ArcGis, licenciado por la institución.

Para la programación de las reglas de negocio asociadas con cada capa deben respetarse los estándares definidos según la naturaleza de la información; por ejemplo, si las reglas de negocio corresponden a un sistema transaccional, deben respetarse los estándares de un sistema transaccional. Si las reglas de negocio son de naturaleza estadística, deben respetarse los estándares de sistemas estadísticos, etc.

Es imposible prever todas las direcciones en que el código evoluciona. Sin embargo, debe pensarse que siempre todos los sistemas evolucionan, y los sistemas espaciales también evolucionan. Considere al momento de diseño si debe implementar la carga de mapas en forma histórica, y cómo debiera ser la administración de dicho sistema.