



**Universidad Nacional Autónoma de México**

---

**Facultad de Contaduría y Administración**

***DEFINICIÓN E IMPLANTACIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOLUCIONES  
ORIENTADAS A SERVICIOS PARA LA DISF DE BANCO DE MÉXICO A TRAVÉS DE UN  
PROYECTO PILOTO***

**Diseño de un Sistema o Proyecto**

**Jorge de Jesús Sánchez Aguilar**



México, D.F.

2014



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



**Universidad Nacional Autónoma de México**

---

**Facultad de Contaduría y Administración**

***DEFINICIÓN E IMPLANTACIÓN DE UNA METODOLOGÍA DE DESARROLLO DE SOLUCIONES  
ORIENTADAS A SERVICIOS PARA LA DISF DE BANCO DE MÉXICO A TRAVÉS DE UN  
PROYECTO PILOTO***

**Diseño de un Sistema o Proyecto**

**Que para obtener el título de:  
Licenciado en informática**

**Presenta:**

**Jorge de Jesús Sánchez Aguilar**

**Asesor:**

**L.I. Gabriel Guevara Gutiérrez**



**México, D.F.**

**2014**

## Contenido

Introducción	5
Capítulo Primero: Marco Referencial	7
<i>Descripción de la empresa</i>	7
<i>Misión</i>	7
<i>Visión</i>	7
<i>Estructura organizacional</i>	7
<i>Organigrama a nivel de la dirección</i>	8
<i>Objetivos estratégicos</i>	9
<i>Descripción de procesos principales</i>	9
<i>Diagrama de procesos</i>	10
<i>Proceso de acopio</i>	11
<i>Proceso de validación</i>	11
<i>Proceso de transformación</i>	11
<i>Proceso de diseminación</i>	11
Capítulo Segundo: Descripción del Problema	12
<i>Problemática actual</i>	12
<i>Necesidades de la DISF</i>	13
Capítulo Tercero: Marco Teórico	15
SOA	15
<i>Modelos de servicios SOA</i>	16
<i>Los ocho principios de SOA</i>	17
<i>El ciclo de vida de los servicios y la gobernabilidad SOA</i>	28
<i>Justificación del proceso de desarrollo de software</i>	32
RUP	32
OpenUP	34
RUP con SOA	37
Capítulo 4: Solución propuesta	43

<i>Objetivo del proyecto</i>	43
<i>Fase de inepción</i>	44
<i>Visión del sistema</i>	44
<i>Matriz de requerimientos</i>	50
<i>Plan de desarrollo de software (PDS)</i>	60
<i>Modelo de casos de uso</i>	81
<i>Diagramas de actividad "CU gestionar perfiles de servicios"</i>	84
<i>Blue print de servicios candidatos</i>	86
<i>Fase de elaboración</i>	87
<i>Diagrama físico de base de datos</i>	88
<i>Diccionario de datos</i>	90
<i>Especificaciones de caso de uso</i>	95
<i>Diagramas de secuencia</i>	107
<i>Prototipo no funcional</i>	110
<i>Perfiles de servicios y capacidades:</i>	115
<i>Plan de pruebas</i>	120
<i>Contratos de Servicios</i>	131
<i>Fase de Construcción</i>	138
<i>Especificaciones de casos de uso</i>	138
<i>Diagramas de actividad</i>	149
<i>Diagramas de secuencia</i>	153
<i>Perfiles de servicios y capacidades</i>	157
<i>Blue print de servicios</i>	161
<i>Contratos de servicios</i>	162
<i>Reporte de pruebas</i>	168
<i>Conclusiones</i>	175
<i>Glosario</i>	177
<i>Bibliografía</i>	179

## **Introducción**

El presente trabajo, tiene como propósito desarrollar una metodología para la Dirección de Información del Sistema Financiero (DISF), conformada por *Service Oriented Architecture(SOA)* y *Rational Unified Process(RUP)*; la cual se prueba por medio del desarrollo de una aplicación orientada a servicios que administra los servicios web de la DISF.

El capítulo de marco referencial describe a la DISF, organización a la cual se le presenta la problemática, se mencionan: su visión, misión, objetivos y procesos principales, a fin de conocer su contexto.

El capítulo de descripción del problema, enuncia la situación de la DISF, se divide en dos partes: la primera consiste en que no hay un proceso definido para realizar soluciones orientadas a servicios; y la segunda en que no hay un control que le permita llevar el inventario de servicios de manera eficiente.

El capítulo de marco teórico define los conceptos en los que se basa el desarrollo de la solución propuesta, compuestos por SOA y RUP; para el primero se describen sus principios y principales temas, y para el segundo se justifica su uso, realizando una comparación con otro proceso de desarrollo de software. Al final del capítulo se realiza la integración de SOA con RUP.

En el capítulo cuarto se presentan los resultados de la integración de SOA con RUP a través del análisis y diseño de una solución orientada a servicios. Con la aplicación de dicha metodología se eliminaron dos brechas de SOA y de RUP:

- La falta de una documentación en el análisis y diseño de soluciones orientadas a servicios.
- La falta de un enfoque de desarrollo de soluciones orientadas a servicios por parte de RUP.

## **Capítulo Primero: Marco Referencial**

En este capítulo se describe la dirección de información del sistema financiero del banco de México, su visión, misión, procesos principales y de soporte, así como su estructura organizacional.

### ***Descripción de la empresa***

La Dirección de Información del Sistema Financiero (DISF) es una dirección del Banco de México y se encarga de captar, validar, transformar y presentar la información de las distintas instituciones financieras (bancos, casas de bolsa, casas de cambio, etc.). Dichas instituciones son las encargadas de reportarle a esta dirección toda la información con respecto a sus finanzas a lo largo del día o en diversos periodos, de acuerdo con el tipo de información que requiera el Banco de México y con la cual pueda asegurar la estabilidad del sistema financiero. (Gerencia, 2012)

### ***Misión***

Proveer información financiera confiable y oportuna, a las unidades administrativas del Banco, a otras Autoridades Financieras y al Público en general, para contribuir al sano desarrollo del sistema financiero y a la estabilidad de precios. (Gerencia, 2012)

### ***Visión***

Consolidar un modelo de información del sistema financiero que se distinga a nivel internacional por la elevada calidad y precisión de sus contenidos, la adopción de las mejores prácticas y la satisfacción de las necesidades de sus usuarios. (Gerencia, 2012)

### ***Estructura organizacional***

La dirección de Información del sistema financiero cuenta con tres áreas cada una representada por una gerencia:

- Gerencia de Información del sistema financiero: ésta es el área más apegada a negocio, la encargada de analizar la información que se genera a través de los procesos principales.

- Gerencia de sistemas de información del sistema financiero: dicha área atiende los requerimientos de la Gerencia de información mediante la aplicación de las tecnologías de información.
- Gerencia de Arquitectura del sistema financiero: ésta gerencia se encarga de la identificación y diseño, planificación, implementación y gobierno de una arquitectura empresarial para la dirección mediante la aplicación de *The Open Group Architecture Framework (TOGAF)*.

**Organigrama a nivel de la dirección**



Ilustración 1 Organigrama DISF (banxico)

### ***Objetivos estratégicos***

- Establecer y apegarse a estándares elevados de calidad de la información que se reciba, genere y disemine.
- Garantizar la oportunidad de los entregables de información a los usuarios.
- Mejorar el acceso, disponibilidad, documentación y presentación del modelo de información que utilizan los usuarios.
- Construir un marco de gobierno de la información con los proveedores, usuarios y al interior de la DISF. (Gerencia, 2012)

### ***Descripción de procesos principales***

A continuación se describen los procesos principales (acopio, validación, transformación y diseminación) que componen a la DISF, primero gráficamente a través de un modelo de determinación de procesos definido por la Gerencia de Arquitectura del sistema Financiero y después en un texto a manera de resumen emitido por la misma Gerencia.

**Diagrama de procesos**

**Nivel 2. Modelo de Determinación de Procesos**

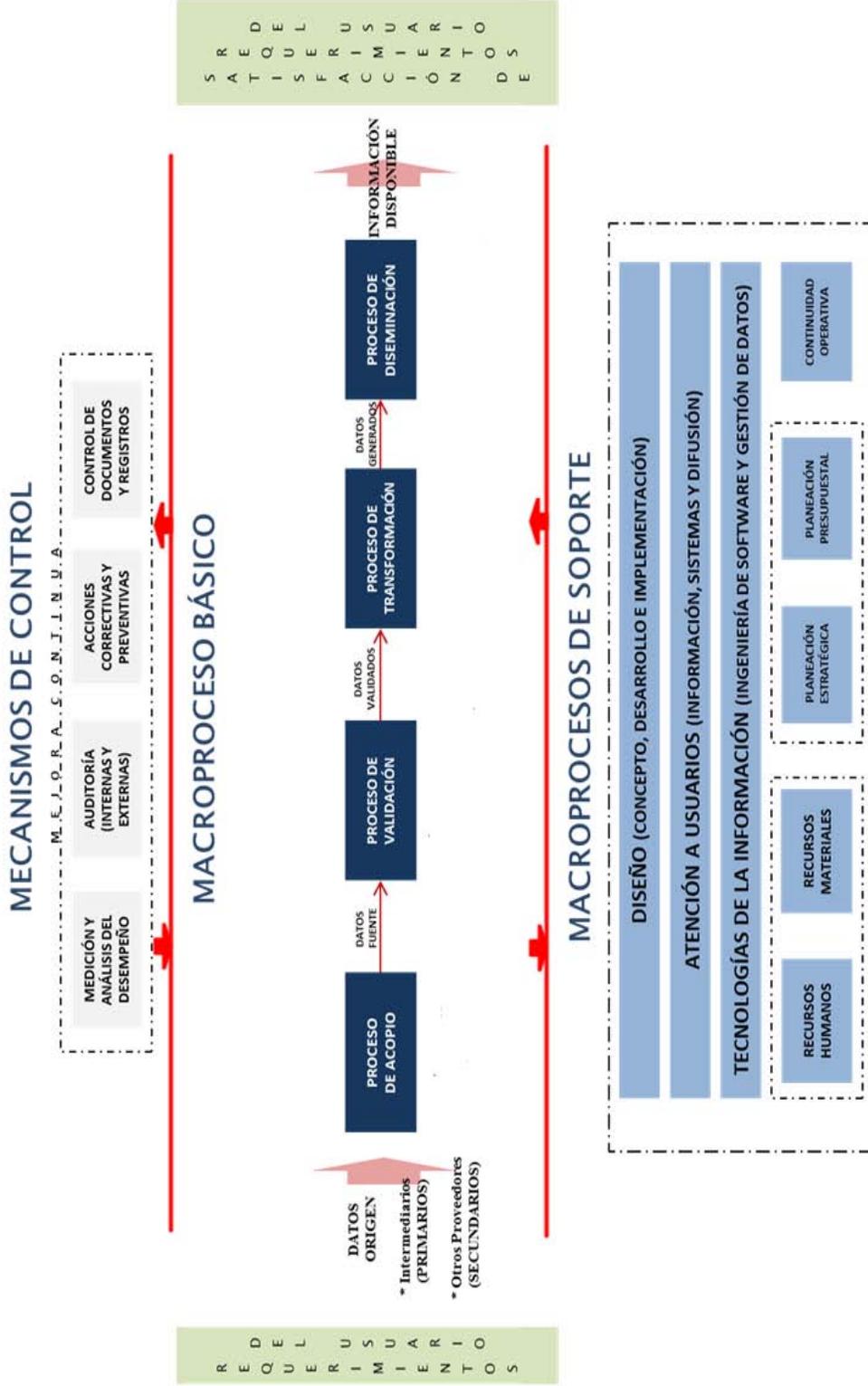


Ilustración 2 diagrama de procesos DISF (Gerencia, 2012)

### ***Proceso de acopio***

Es el proceso en el cual se incorporan los datos por parte de un proveedor, ya sea un intermediario financiero o un proveedor de datos como Valuación Operativa y Referencias de Mercado (Valmer). Los datos son validados en su estructura y características básicas antes de ser aceptados. (Gerencia, 2012)

### ***Proceso de validación***

Se refiere a la verificación de datos fuente a través de reglas de negocio con el fin de asegurar cierta calidad. Podemos identificar varios tipos de validación entre las que se encuentran las estadísticas, las de mercado, las regulatorias y las financieras. (Gerencia, 2012)

### ***Proceso de transformación***

En esta etapa se genera la información de valor agregado que soporta la revisión o que va a ser entregada a otros usuarios. (Gerencia, 2012)

### ***Proceso de diseminación***

Es el proceso a través del cual se distribuye la información ya sea mediante replicaciones físicas a otros equipos, a través de Internet o de consultas de bases de datos transaccionales o de *Data Warehouse*. (Gerencia, 2012)

## **Capítulo Segundo: Descripción del Problema**

La DISF decidió utilizar la Arquitectura Orientada a Servicios a principios del 2009 con la finalidad de estar a la vanguardia en las tecnologías de la información y debido a que se identificó que en varios de los proyectos de software existía lógica redundante, desde esa fecha a la actualidad ha estado en un proceso de capacitación para lograr la adopción de SOA, en este periodo se han llevado a cabo varios proyectos de soluciones orientadas a servicios, que han contemplado los procesos principales de la organización (acopio, validación, transformación y disseminación).

### ***Problemática actual***

Actualmente, se realizan proyectos orientados a servicios para atender las necesidades de la organización. En cada proyecto, se forma un equipo de siete personas como máximo donde se incluye un líder de proyecto. El equipo aborda de manera diferente el proyecto con base en la experiencia del líder de proyecto y a el ciclo de vida de servicios definido por SOA. Algunos líderes no documentan sus proyectos, y quienes si lo hacen, sólo documentan los requerimientos y realizan algunos diagramas que al final no concuerdan con la versión final de la solución orientada a servicios.

Lo anterior provoca que se tenga poca o nula documentación de los proyectos y del desarrollo de servicios. La documentación que se genera está en diferentes formatos y los diagramas que se emplean son de diversas notaciones lo que dificulta el entendimiento del proyecto por parte de cualquier integrante de los equipos y el mantenimiento de los servicios. Aunado a esto, la DISF no cumple con la documentación mínima que solicita la contraloría interna del Banco de México para las áreas que desarrollan software.

Por otro lado, no se tiene un control de los servicios web que se han venido realizando producto de los proyectos de soluciones orientados a servicios, esto genera problemas de redundancia en los servicios web debido a la poca documentación de los mismos provocando un difícil mantenimiento.

Otro problema, es que el acceso a la documentación existente es a través de un repositorio de documentos interno que no se mantiene actualizado y que no cuenta con un esquema de privilegios. La búsqueda de los servicios es lenta debido a que se debe revisar documento por documento, lo que dificulta el descubrimiento de los servicios web y su posible reuso, incumpliendo con los principios definidos por SOA.

### ***Necesidades de la DISF***

Con base en lo anterior se identifican las necesidades siguientes:

1. Definir un marco de trabajo para los equipos que realicen soluciones orientadas a servicios, aportando calidad a los proyectos mediante procesos bien definidos con entregables que faciliten tanto el mantenimiento como el entendimiento de los interesados y que cumpla con la siguiente documentación exigida por la contraloría del Banco de México:
  - Análisis de requerimientos:
    - Documentación de requerimientos y análisis con descripción de la funcionalidad prevista para el sistema.
    - Documentación de requerimientos y análisis con aceptación formal del líder funcional del proyecto.
    - Documentación de requerimientos y análisis con definición del proceso sujeto de automatización.
    - Documentación de requerimientos y análisis con la descripción de las necesidades del usuario.

- Documentación del plan de proyecto a seguir para el cumplimiento de los requerimientos del usuario.
  - **Diseño:**
    - Documentación del diseño con especificaciones de la organización de la información (modelo de datos) del sistema.
    - Diccionario de datos del sistema.
    - Documentación del diseño con descripción técnica de los principales procesos o métodos del sistema.
    - Documentación del diseño con descripción de las interfaces con el usuario, entre módulos y con otros sistemas.
  - **Entrega del sistema:**
    - Formalización de la entrega del sistema con constancia de ejecución de pruebas.
    - Formalización de la entrega del sistema con verificación de que el sistema funciona conforme a sus especificaciones.
    - Guía de operación para usuarios con descripción del conjunto de acciones que el usuario del sistema debe llevar a cabo para operarlo.
  - **Plan de pruebas:**
    - Plan de pruebas con descripción de cómo realizar la verificación de que la funcionalidad garantiza que la operación del sistema cumple con los requerimientos establecidos.
    - Plan de pruebas con especificaciones de como ejecutarlo y con constancia de su aplicación.
- 2.** Realizar un sistema informático que administre los servicios web y permita centralizar sus metadatos, con la finalidad de descubrirlos evitando redundancia en el desarrollo de servicios.

## Capítulo Tercero: Marco Teórico

En este capítulo se tratan los conceptos SOA y RUP, que se ocupan para el desarrollo de la solución orientada a servicios propuesta. El primero de ellos es *Service Oriented Architecture* (SOA) que es la arquitectura de software que actualmente está implementada en la DISF, y el segundo es *Rational Unified Process* (RUP) que es el proceso de desarrollo de software elegido con un enfoque iterativo e incremental, usado principalmente para desarrollar software orientado a objetos.

### SOA

Arquitectura Orientada a Servicios (SOA) : Es un modelo de arquitectura de tecnología para soluciones orientadas a servicios con distintas características que soportan la orientación a servicios y los objetivos estratégicos asociados con la computación orientada a servicios. (Inc., ServiceOrientation.com, 2005- 2014)

Para entender SOA podemos establecer su relación con otros conceptos:

- Hay un conjunto de objetivos estratégicos relacionados con la computación orientada a servicios.
- Estos objetivos representan un estado objetivo.
- La orientación a servicios es el paradigma que provee un método probado para alcanzar ese estado deseado.
- Cuando aplicamos la orientación a servicios para el diseño de software, construimos unidades lógicas llamadas "servicios".
- Las soluciones orientadas a servicios se componen de uno o más servicios.
- Para construir soluciones orientadas a servicios exitosas, necesitamos una arquitectura tecnológica distribuida con características específicas.
- Esas características distinguen a la arquitectura tecnológica como orientada a servicios. Esto es SOA. (Inc., ServiceOrientation.com, 2005- 2014)

SOA define entregables, fases, roles y responsabilidades por ciclo de vida de un servicio web, pero existe una brecha entre el ciclo de vida de un servicio web y el ciclo de vida de una solución orientada a servicios compuesta por más de un servicio o composiciones de servicios, ésta atiende a una problemática de la empresa, en resumen, es la realización de un sistema con la peculiaridad de orientarlo a servicios, y es ahí donde entra un proceso de desarrollo de software (complementando SOA), en este caso RUP.

El objetivo de SOA es mejorar la eficiencia, agilidad y productividad de la empresa haciendo uso de los servicios como el principal medio a través del cual se realizan las soluciones orientadas a servicios asociadas con los objetivos estratégicos de la empresa.

### ***Modelos de servicios SOA***

Existen tres tipos de servicios que define la metodología SOA y que van de acuerdo a la lógica encapsulada, a potencial de reuso o a su relación con alguna entidad dentro del negocio. Estos tipos son los siguientes:

- ❖ **Servicios de Entidad:** Servicios que reflejan entidades de negocio, tales como empleado, factura, etc. Son servicios centrados en la empresa y las entidades que representan definen su contexto funcional. Este tipo de servicios son llamados agnósticos ya que pueden ser usados por la mayoría de los procesos en los que participe ésta entidad.
  
- ❖ **Servicios de Tarea:** Este tipo de servicio está asociado directamente con una tarea o proceso de negocio lo que disminuye su potencial de reuso, generalmente se ocupa como un controlador en una composición de servicios, encargado de coordinar y envolver varios servicios que en conjunto reflejan al proceso o tarea de negocio.

- ❖ **Servicios de Utilería:** Este modelo de servicio está centrado en la tecnología más que en el negocio, característica que le da un alto potencial de reuso para cualquier proceso o tarea empresarial. Se encarga de proveer funcionalidad como notificaciones o manejo de excepciones, que puede ser reutilizable por cualquier otro servicio.

### Los ocho principios de SOA

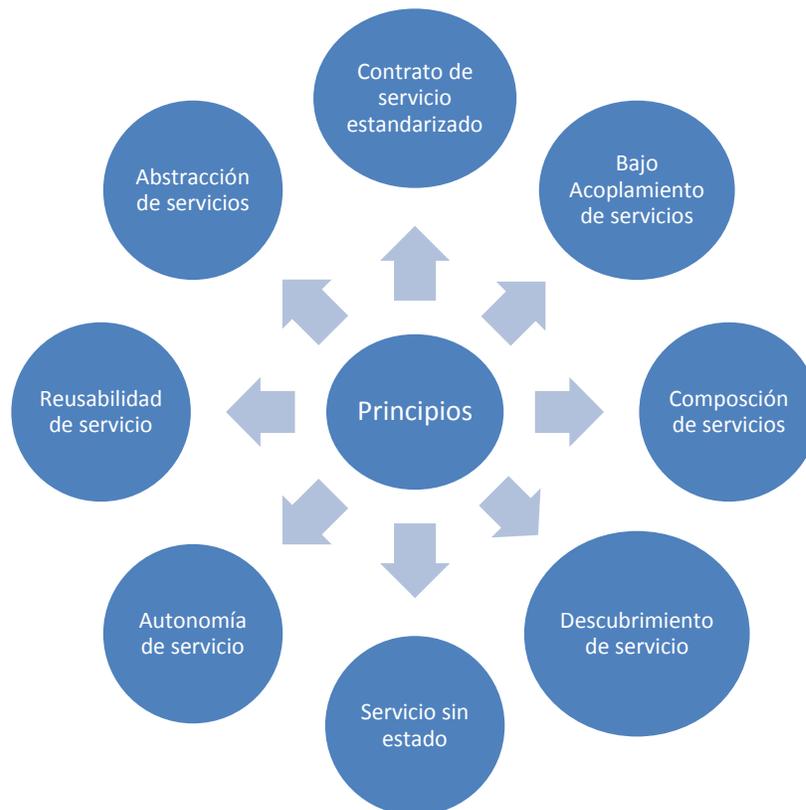


Ilustración 3 modelos de servicios

#### 1. Contrato de servicios estandarizado

*“Services within the same service inventory are in compliance with the same contract design standards.” (Earl, SOA Principles of Service Design , 2008)*

El contrato de los servicios es el medio para lograr la comunicación entre sí, por lo que este principio es una parte fundamental para que SOA se aplique de manera exitosa.

Estandarizar notaciones, lenguajes y tipos de datos de los contratos es importante para lograr la interoperabilidad transparente de los servicios. Un contrato se compone de tres tipos de documentos técnicos y uno de alto nivel. (Curbera, Christensen, Meredith, & Weerawarana, 2001)

Documentos técnicos:

- *WSDL (Web Services Description Language)*: Es una especificación escrita en lenguaje de marcado XML que describe un servicio web. Se definen tanto las operaciones como las entradas y salidas para cada operación haciendo uso de tipos de datos que a menudo son representados en “esquemas XML”.
- *Esquemas XML*: son especificaciones XML, que se utilizan para definir tipos de datos compuestos, por lo general representan entidades de negocio (cliente, usuario, venta, etc.). Los esquemas pueden ser parte de entradas y salidas de las operaciones en un contrato de servicios “WSDL”.
- *Políticas WS (web services)*: Es una especificación escrita en lenguaje XML que sirve para definir un conjunto de reglas que los consumidores del servicio deben cumplir para acceder al mismo o que el servicio debe definir para poder ser accedido por algún consumidor en particular.

Documento de alto nivel:

- *SLA (Service Level Agreement)*: El acuerdo a nivel de servicio es un documento de alto nivel que describe los requerimientos de calidad que un servicio está comprometido a brindar a sus consumidores.

Cada parte que compone a un contrato de servicios debe estar estandarizada (los tipos de dato utilizados en los esquemas XML, la notación ocupada en los tres documentos técnicos, los espacios de nombres, etc.) dentro de un mismo inventario de servicios.

## 2. Bajo acoplamiento de servicios

*“Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.”* (Inc., ServiceOrientation.com, 2005- 2014)

El bajo acoplamiento de los servicios a su entorno permite un mayor potencial de reúso. El contrato del servicio debe estar desacoplado de un proceso de negocio en específico para que los potenciales consumidores puedan hacer uso de él sin la necesidad de realizar grandes modificaciones. Esto hace al inventario de servicios flexible y adaptable a los cambios en el negocio, pudiendo crear nuevas composiciones de servicios o extendiendo las existentes que atienden a los procesos de negocio actuales, teniendo siempre la posibilidad de eliminar relaciones entre servicios con un impacto mínimo sobre los mismos.

El acoplamiento entonces, debe ser bajo, dentro del ambiente del servicio y entre los servicios.

SOA define los siguientes tipos de acoplamiento:

- **Acoplamiento de la lógica al contrato.**- Es el único acoplamiento permitido por SOA y se da por naturaleza si se aplica un enfoque *top-down*, donde lo primero que se realiza es el contrato por lo que la lógica se acopla al mismo.
- **Acoplamiento del contrato a la lógica.**- Este acoplamiento es típico del enfoque *bottom-up* donde a partir de la lógica ya existente se autogenera el contrato de servicio web, con alguna herramienta. En este acoplamiento el contrato depende completamente de la aplicación que se quiere exponer como servicio web.
- **Acoplamiento del contrato a la tecnología.**- Cuando el contrato de un servicio expresa detalles propios de la tecnología con la que se está implementando, tales como el protocolo de comunicación o del lenguaje, se tiene un acoplamiento del contrato a la tecnología.

- **Acoplamiento del contrato a la implementación.**- Este acoplamiento se da cuando el contrato refleja detalles como se resolvió la lógica del servicio, esto puede incluir rutas de red o atributos que utilizan componentes específicos como *frameworks* de vista o bibliotecas conversoras de archivos.

### 3. Abstracción de servicio

*"Service contracts only contain essential information and information about services is limited to what is published in service contracts"* (Inc., ServiceOrientation.com, 2005-2014)

La información que se publica del servicio debe ser solo la indispensable para que los posibles consumidores puedan acceder a él.

Se debe evitar el acceso innecesario a detalles del servicio tales como los de implementación, ya que entre más conozca un posible consumidor, puede acoplarse profundamente al servicio u hacer suposiciones y juicios acerca de su implementación o dependencias tecnológicas del mismo, lo que provoca un acoplamiento negativo y crea dependencias que después serán difíciles de eliminar.

### 4. Reusabilidad de servicio

*"Services contain and express agnostic logic and can be positioned as reusable enterprise resources"*. (Inc., ServiceOrientation.com, 2005- 2014)

Este principio es una parte fundamental de la orientación a servicios, ya que la reutilización es su principal ventaja en comparación con otros paradigmas de computación.

La idea de la reutilización es desarrollar software que agregue valor no sólo para un propósito en específico, sino que agregue un valor repetido, en otras palabras “realizar servicios multipropósito”, lo que a largo plazo resultará una mejor inversión.

El reuso de servicios permite responder de una manera más rápida y eficiente a las necesidades de la empresa, mediante la composición o extensión de servicios existentes, esto le da mucha agilidad a la empresa aumentando a la vez el ROI.

Para garantizar el cumplimiento de este principio, se necesita una infraestructura que permita una alta o extrema concurrencia de servicios, ya que conforme madura el inventario, las composiciones van aumentando cada vez más, al final se pretende con este principio que un servicio sirva para automatizar varias tareas de negocio.

SOA define las siguientes medidas de reusó:

- Reusabilidad Táctica: se refiere a desarrollar sólo las capacidades que atiendan las necesidades inmediatas, pero dejando el servicio flexible para que se pueda extender en el futuro.
- Reusabilidad selectiva: Este tipo de reusabilidad se da cuando un equipo de desarrollo tiene la oportunidad de entregar un servicio con la funcionalidad que va más allá de lo necesario. En este caso se realiza un análisis orientado a servicios y sólo se implementan las capacidades que cumplen con las necesidades inmediatas y las que tienen un alto potencial de reuso.
- Reusabilidad completa: En este tipo de reuso se realiza un análisis minucioso para dotar al servicio de la mayor funcionalidad posible, dentro de un inventario con un alcance definido, se realiza la implementación de todas sus capacidades, teniendo en cuenta que se cuenta con un plan de inventario de servicios.

El tipo de reuso seleccionado depende de las necesidades de la empresa, pero asegurar que los servicios sean altamente reutilizables aumenta el grado de éxito de la implementación de SOA.

## **5. Autonomía de servicios**

La autonomía representa el grado en que se es capaz de actuar independientemente, se da en la medida en la que un servicio tiene la capacidad de autogobernarse. La finalidad de lograr autonomía en los servicios es aumentar su fiabilidad en tiempo de ejecución, ya que entre más autónomo sea el servicio resultará menos afectado por ambientes externos, lo que ayuda a predecir su rendimiento y comportamiento, algo de suma importancia cuando se pretenda reutilizar.

Debido a que SOA se basa en composiciones de servicios que a su vez pueden estar en más composiciones, los servicios tienen a ser naturalmente no autónomos, pero se minimiza esta pérdida de autonomía si los servicios con gran potencial de reuso por sí solos tienen un alto nivel de autonomía.

Es importante que el servicio comunique su nivel de autonomía, para ayudar a los posibles consumidores a tomar mejores decisiones en tiempo de diseño. SOA define los siguientes niveles de autonomía:

- Autonomía de contrato de servicios.- este nivel representa que las capacidades del servicio no se traslapan con ninguna otra capacidad de los servicios existentes en el inventario, es decir, la lógica expresada en el contrato es única y no coincide con ningún otro servicio. Es importante mencionar que el concepto de normalización se debe aplicar en el inventario para ayudar a cumplir con este principio, y eliminar lógica redundante en un inventario de servicios.

- Autonomía compartida: en este nivel prácticamente la autonomía no existe. Se refiere al caso en el que no todos los servicios son implementados con un enfoque *top down*, algunos se realizan con un enfoque *bottom up* donde los sistemas existentes se convierten a servicios con ayuda de diversas herramientas, por lo que al final no resultan ser autónomos, ya que dependen de la lógica embebida en los sistemas también llamados “legados”. El término “compartida”, sólo se refiere al hecho de que al final, otras partes de la empresa pueden acceder a esa lógica de procesamiento que se encuentra dentro de los límites de esos servicios.
- Autonomía de la lógica del servicio: Este nivel de autonomía es el más común cuando se comienzan a desarrollar servicios. Indica que los servicios son autónomos en su lógica pero que comparten componentes tales como la base de datos. Compartir componentes como la base de datos ocasiona problemas cuando alguno de los servicios ejecuta una consulta muy tardada, o cuando varios servicios intentan acceder a los mismos registros, ya que se reduce el tiempo de respuesta o el comportamiento de los servicios se vuelve difícil de predecir.
- Autonomía pura: Es el nivel de autonomía ideal y se logra cuando un servicio es totalmente autogobernado en tiempo de ejecución, por lo general se llega a este nivel usando un enfoque *top-down*. Se pretende aislar el servicio de su entorno para tener el total control del mismo.

El servicio ofrece una mayor garantía de rendimiento en tiempo de ejecución, ya que su comportamiento al no depender de agentes externos es completamente predecible.

Es importante mencionar que los servicios son al final un conjunto de capacidades, cada una de las cuales puede tener diferentes niveles de autonomía, como el servicio se debe ver como un “todo”, SOA lo califica como autonomía mixta.

## **6. Servicio sin estado**

Cada que se van creando nuevos servicios las composiciones van creciendo, y la concurrencia a los servicios agnósticos también, por lo que en ocasiones es importante para los consumidores conocer el estado en el que se encuentra un servicio en tiempo de ejecución. (Inc., ServiceOrientation.com, 2005- 2014)

Anteriormente en las soluciones cliente servidor la interfaz de usuario cliente retenía en memoria grandes cantidades de datos referidos a la actividad en tiempo de ejecución durante periodos prologados, lo que al final no era problema ya que cada programa cliente se desplegaba en un equipo dedicado.

En la computación distribuida actual se requiere un programa del lado de servidor que atienda la interacción con los diversos programas clientes, cada uno con requisitos propios de ejecución e información de estado.

En los escenarios anteriores de computación distribuida, un servidor atiende a distintos clientes, cada uno con requerimientos especiales, pero en SOA con la reutilización de los servicios, guardar estado se vuelve un poco más complicado, importante y preocupante en algunos casos, ya que un servicio no sólo es un programa que está creado para atender varias peticiones de clientes relacionados a una tarea de negocio específica, un servicio debe estar preparado para recibir muchas más peticiones de clientes relacionados con varias tareas de negocio.

De acuerdo con lo anterior, lo que busca este principio es racionalizar la información del estado más a nivel de capacidad que a nivel de servicio, se debe reducir el consumo de recursos referidos a la administración de estado de una capacidad de servicio cuando sea necesario.

## 7. Descubrimiento de Servicios

El descubrimiento de servicios implica más que encontrar los mismos. Este principio se enfoca en que la información de los servicios existentes en el inventario sea fácilmente accesible, entendible, precisa y coherente, lo que maximiza la reutilización de los servicios y minimiza el riesgo de lógica redundante en el inventario de servicios. (Inc., ServiceOrientation.com, 2005- 2014)

Lo que busca SOA con este principio es responder a la pregunta: ¿Ya existe la funcionalidad que necesito o tengo que crearla?, para responder correctamente esta pregunta, tenemos que conocer entre otras cosas, el propósito de los servicios, de las capacidades y sus limitaciones.

SOA define los siguientes tipos de descubrimiento:

- *Descubrimiento en tiempo de diseño y tiempo de ejecución.*- En tiempo de diseño el descubrimiento es responsabilidad del humano, que es comúnmente ayudado por un repositorio central de información de los servicios existentes en el inventario, llamado “registro de servicios”. Otras tecnologías como UDDI proporcionan interfaces de programación que nos permiten construir programas y servicios capaces de hacer consultas de descubrimiento dinámicas o en tiempo de ejecución, lo que puede iniciar un automáticamente un proceso de descubrimiento que busque hasta consumir el servicio descubierto en tiempo de ejecución.
- *Descubrimiento de meta información.*- De acuerdo con el principio de abstracción debemos ocultar la información tanto de tecnología como de implementación de un servicio, pero tomando en cuenta este principio, se debe publicar la información referente a la funcionalidad (que hace) y calidad (como se comporta) del mismo para asegurar su fácil descubrimiento.

- El cumplimiento de este principio ayuda a potencializar un gran beneficio de SOA: la reutilización de servicios en varios procesos de negocio.

## **8. Composición de servicios**

Componer servicios en procesos de negocio es uno de los ideales de SOA. La composición nace a partir de descomponer algo, y es básicamente la aplicación de divide y vencerás, pero con la diferencia que cuando divides tienes que esforzarte un poco más en generalizar las piezas individuales para hacerlas generales. (Inc., ServiceOrientation.com, 2005- 2014)

Varios de los principios mencionados anteriormente son la base de este principio, pero están enfocados a características de diseño individuales, por lo que el principio de composición, intenta asegurar que el servicio, aplique de una manera adecuada las características de diseño colectivas que se requieren para soportar una composición efectiva.

Una de las metas de SOA es representar la empresa a través de un inventario de servicios altamente reutilizables, que proporcionen los medios necesarios para cumplir con futuras necesidades de automatización de la empresa, esto se puede lograr solamente con composiciones de los servicios existentes en el inventario.

Las características de diseño para las capacidades miembros de composición necesitan tanto que su ambiente de ejecución sea eficiente para funcionar individualmente cómo de un contrato flexible que soporte el intercambio de diferentes tipos de datos para funciones similares.

Por lo regular, el modelo de servicios de tipo “tarea”, asume las composiciones de servicios de acuerdo a una tarea de negocio en específico, por lo que a este modelo no es posible aplicarle de una manera estricta el principio de reutilización.

#### Tipos de composiciones

- Composiciones primitivas: consisten básicamente en un servicio iniciador, uno o dos servicios intermedios que realizan lógica de procesamiento adicional, como un filtro de datos del servicio iniciador, y un receptor final.
- Composiciones complejas: van de la mano con la evolución del inventario de servicios, en donde, lo que al principio eran pocas y primitivas composiciones, que se realizaron para atender procesos críticos de negocio a partir de sistemas legados, se convierten en un inventario de servicios, en su mayoría son agnósticos, a partir de los cuales se puede automatizar cualquier proceso de negocio convirtiéndose así en composiciones complejas.

***El ciclo de vida de los servicios y la gobernabilidad SOA***

SOA define fases para el ciclo de vida de los servicios, dichas fases tienen objetivos y roles que los hacen cumplir. La gobernabilidad SOA es básicamente un conjunto de preceptos y procesos que se deben hacer cumplir para maximizar el éxito de la implementación de SOA y garantizar todos sus beneficios.

El siguiente cuadro muestra un resumen de las fases del ciclo de vida de los servicios, así como los preceptos que se deben cumplir por cada fase y los procesos que ayudarán al cumplimiento de esos preceptos.

Fase	Descripción	Preceptos	Procesos
Análisis del inventario de servicios.	Es la fase donde se define el alcance del inventario de servicios para identificar los procesos y sistemas de información impactados.	Definición del alcance de inventario de servicios.	Priorización de requerimientos de negocio.
Análisis orientado a servicios	Es la fase donde se ejecuta el modelamiento de servicios para identificar y clasificar servicios de tipo tarea, entidad y utilidad.	Estándares de nomenclatura de capacidades y servicios candidatos. Normalización de Servicios. Estándares de control de versiones de servicios candidatos.	Revisión de servicios candidatos.

Fase	Descripción	Preceptos	Procesos
Diseño del contrato orientado a servicios.	Es la fase donde se diseña el contrato WSDL, XSD, WS-Policy y SLA de los servicios.	Estándares de diseño de esquemas Estándares de diseño del contrato del servicio Estándares de diseño del contrato de la orientación a servicios Plantilla SLA.	Revisión del diseño del contrato. Registro del Servicio.
Diseño de la lógica del servicio.	Es la fase donde se diseña la arquitectura de los servicios.	Estándares de diseño de la lógica del servicio Estándares de diseño de la arquitectura orientada a servicios.	Control de acceso al servicio Revisión del diseño de la lógica del servicio Auditoría de datos legales.
Desarrollo del servicio.	Es la fase donde se desarrolla la lógica del servicio a partir del contrato WSDL y XSD adoptando una estrategia Top-Down.	Estándares de programación de la lógica del servicio Estándares de tecnologías de desarrollo personalizados.	Revisión del código de la lógica del servicio.

Fase	Descripción	Preceptos	Procesos
Pruebas del servicio.	Es la fase donde se realizan las pruebas de unidad, funcional, seguridad, regresión, rendimiento e integración de los servicios y composiciones.	<p>Estándares de herramientas para pruebas.</p> <p>Estándares de parámetros para pruebas.</p> <p>Estándares de pruebas del servicio.</p> <p>Guías de uso de los datos de pruebas.</p>	Revisión del resultado de la prueba de servicio.
Despliegue y mantenimiento del servicio.	Es la fase donde se despliegan los servicios y composiciones en el ambiente de producción. También se hacen tareas de mantenimiento a la infraestructura.	Estándares de despliegue y mantenimiento en producción.	<p>Revisión de certificación del servicio.</p> <p>Revisión de mantenimiento del servicio.</p>
Uso y monitoreo del servicio.	Es la fase donde se realiza el monitoreo del uso de los servicios a nivel de consumo, datos, excepciones y composiciones que garanticen su correcta operación, escalabilidad, fiabilidad y consistencia.	<p>Umbrales de Uso del Servicio en Tiempo de Ejecución.</p> <p>Disparadores de vitalidad del servicio.</p>	Revisión de vitalidad del servicio.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Fase	Descripción	Preceptos	Procesos
Descubrimiento del servicio	Es la fase donde se registran los servicios para su descubrimiento y reutilización en otras composiciones.	Registro de servicios centralizado.	Control de acceso al registro del servicio. Revisión de los datos del registro de servicios. Descubrimiento del servicio. Solicitud de uso del servicio compartido. Solicitud de Modificación del Servicio Compartido.
Control de versiones y jubilación del servicio	Es la fase donde se versionan los servicios de acuerdo a su evolución y cambios.	Estrategia de control de versiones del servicio. Reglas de control de versiones del sla. Notificación de jubilación del servicio.	Control de versiones del servicio. Jubilación del servicio.

Tabla 1 Gobernabilidad SOA (Earl, SOA Governance Governing Shared Services On-Premise and in the Cloud., 2011)

### ***Justificación del proceso de desarrollo de software***

A continuación se explican brevemente los principales procesos de desarrollo de software existentes que se tomaron en cuenta para su adaptación con SOA, adecuándolo a las necesidades de la DISF. Los procesos ágiles como SCRUM o FDD no se tomaron en cuenta debido a que no cumplían con la mayoría de la documentación requerida por la contraloría de BANXICO:

#### ***RUP***

**Definición:** El “*Rational Unified Process*” es un proceso de ingeniería de software, provee un enfoque disciplinado de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios finales dentro de un presupuesto y calendario predecible. (Kruchten, 2003)

**Tipo de proyectos:** RUP está enfocado para proyectos grandes, con muchos recursos, aunque nos da la posibilidad de adecuarlo a nuestras necesidades.

<b>Disciplina</b>	<b>Artefactos Principales</b>	<b>Roles</b>
Administración de proyectos	Plan de desarrollo de software Caso de negocio Planes de Iteración	Administrador de proyectos
Modelado de negocio	Documento de visión de negocio Modelo de casos de uso de negocio Modelo de Análisis de negocio Glosario de negocio	Analista de procesos de negocio Diseñador de negocio

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Disciplina</b>	<b>Artefactos Principales</b>	<b>Roles</b>
Requerimientos	Plan de administración de requerimientos. Requerimientos de software. Documento de visión.	Analista de sistemas. Especialista en requerimientos.
Análisis y diseño	Modelado de diseño. Documento de arquitectura de software.	Arquitecto de software. Diseñador de base de datos. Revisor de arquitectura y diseño.
Implementación	Plan de integración. Implementación de subsistemas.	Revisor de código Desarrollador
Pruebas	Plan de pruebas. Reporte de pruebas. Bitácora de pruebas. Estrategias de prueba.	Administrador de pruebas. Analista de pruebas. Diseñador de pruebas. Probador.
Configuración y administración del cambio	Plan de administrador de la configuración. Petición de cambio.	Administrador de la configuración.
Ambiente	Caso de desarrollo. Plantillas específicas del proyecto. Guías específicas del proyecto.	Ingeniero de procesos. Especialista en herramientas. Administrador de sistemas
Despliegue	Plan de despliegue.	Administrador de despliegue

Tabla 2 Descripción RUP (Kruchten, 2003)

### **OpenUP**

**Definición:** Proceso de desarrollo de software basado en el manifiesto Ágil.

**Tipo de proyectos:** Está pensado para proyectos pequeños, con equipos de igual tamaño que trabajan juntos en el mismo lugar. La comunicación cara a cara así como la presencia de los involucrados como parte activa de todo el proyecto maximizan una implantación exitosa del proceso. (Mangan, 2014)

<b>Disciplina</b>	<b>Artefactos Principales</b>	<b>Roles</b>
<i>Arquitectura</i>	<i>Architecture Notebook</i>	<i>Arquitecto</i>
<i>Despliegue</i>	<i>Product Documentation</i> <i>User Documentation</i> <i>Training Materials</i> <i>Backout Plan</i> <i>Deployment Plan</i> <i>Infrastructure</i> <i>Release Communications</i> <i>Release Controls</i> <i>Release</i>	<i>Ingeniero de implementación.</i> <i>Administrador de despliegue.</i> <i>Dueño del producto.</i> <i>Escritor técnico.</i> <i>Capacitador.</i>
<i>Desarrollo</i>	<i>Implementation</i> <i>Build</i> <i>Developer Test</i> <i>Design</i>	<i>Desarrollador</i>
<i>Ambiente</i>	<i>Project Defined Process</i> <i>Tools</i>	<i>Ingeniero de procesos.</i> <i>Especialista en herramientas.</i>

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Disciplina</b>	<b>Artefactos Principales</b>	<b>Roles</b>
Administración de proyectos	<i>Risk List</i> <i>Work Items List</i> <i>Iteration Plan</i> <i>Project Plan</i>	Administrador de proyectos
Requerimientos	<i>Glossary</i> <i>Vision</i> <i>System-Wide Requirements</i> <i>Use-Case Model</i> <i>Use Case</i>	Analista
Pruebas	<i>Test Case</i> <i>Test Script</i> <i>Test Log</i>	Probador

Tabla 3 Descripción OpenUP (Hristo Toshkov Hristov, 2012)

	<b>RUP</b>	<b>OpenUP</b>
Ventajas	<ul style="list-style-type: none"> <li>➤ Pensado para proyectos grandes, adaptable para cualquier otro tamaño.</li> <li>➤ Existen muchas referencias basadas en la experiencia de su implementación.</li> <li>➤ SOA dedica un apéndice de uno de sus libros para referirse a un mapeo básico de SOA-RUP.</li> <li>➤ Cumple con la documentación necesaria solicitada por la contraloría del Banco de México.</li> </ul>	<ul style="list-style-type: none"> <li>➤ La carga de trabajo en la documentación es mínima.</li> <li>➤ Pensado para proyectos pequeños.</li> <li>➤ Pocas referencias de su implementación en comparación con RUP.</li> <li>➤ Facilidad de aprendizaje.</li> </ul>
Desventajas	<ul style="list-style-type: none"> <li>➤ Gran cantidad de documentación.</li> <li>➤ Tedioso y difícil de aprender por su complejidad para alguien que no cuenta con las bases suficientes.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Omite contenido que puede ser de interés para algún interesado en el proyecto.</li> <li>➤ No cumple con toda la documentación solicitada por la contraloría del Banco de México.</li> </ul>

**Tabla 4 comparacion RUP vs OpenUP**

### ***Proceso de desarrollo elegido***

Después de realizar a comparación de los diferentes procesos de desarrollo de software se decidió por Rational Unified Process, ya que cuenta con la documentación suficiente para cumplir con lo dictado por la contraloría. Otra de las razones es porque siendo un proceso tan complejo, existen referencias de su mapeo con la Arquitectura Orientada a Servicios.

RUP: El *Rational Unified Process* es un método de ingeniería de software. Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización. Su objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales dentro de un calendario y presupuesto predecible. RUP capta muchas de las mejores prácticas en el desarrollo de software moderno de manera que se adecúe a una amplia gama de proyectos y organizaciones, las mejores prácticas son: Desarrollar software de forma iterativa, gestionar requerimientos, uso de arquitecturas basadas en componentes, modelo de software visual, verificar continuamente la calidad de software, controlar los cambios de software. (Kruchten, 2003)

### ***RUP con SOA***

La siguiente es la propuesta de adaptación de RUP con SOA para llevar a cabo las soluciones orientadas a servicios. Parte de esta adaptación se refleja en algunos entregables tales como los diagramas de secuencia, en donde se combina notación UML, con notación de SOA.

### ***Fases***

Se propone conservar las fases que indica RUP, en cada fase se incluyen actividades y artefactos de las fases que propone SOA, y se complementan con algunos artefactos de RUP cumpliendo así con las necesidades de la DISF.

## **Incepción**

### Objetivos:

- Se establece el caso de negocio que atenderá el sistema.
- Se define el alcance inicial de la solución orientada a servicios.
- Se identifican los casos de uso críticos y los servicios web que cumplan con los casos de uso críticos.
- Proporcionar estimaciones para la fase de elaboración.

### Actividades:

- Formular el alcance del proyecto, que dará pie al alcance de la solución orientada a servicios, esto es, captar el contexto, las necesidades y las limitaciones más importantes para que se puedan derivar en criterios de aceptación para el producto final.
- Sintetizar una arquitectura candidata, evaluar las ventajas y desventajas en el diseño, y valorar si hacer, comprar o reusar para que el costo, la calendarización y los recursos puedan ser estimados. (Kruchten, 2003)
- Analizar los servicios y esquemas existentes para tomar en cuenta un posible reúso de los mismos y evaluar un posible impacto.
- Identificar y clasificar los servicios web identificados.

### Artefactos:

- Documento de visión: Un documento de visión, es decir, una visión general de los requisitos del proyecto básico, sus principales funciones y sus principales limitaciones.
- Matriz de requerimientos: Se registran todos los requerimientos identificados hasta esta fase.
- Modelo de casos de uso: que enumera todos los actores y casos de uso que pueden ser identificados en esta etapa.
- Glosario: un glosario inicial del proyecto, con palabras clave que ayuden a los

*stakeholders* a entender el proyecto.

- Plan de desarrollo de software: En el cuál se muestra lo que habrá que realizar en las fases e iteraciones.
- EDT (Estructura de descomposición de trabajo): Se definen los artefactos del proyecto e iteraciones en forma jerárquica para su fácil visualización (este es el único artefacto tomado del PMBOK). (Project Management Institute, 2008)
- *Blue print* de los servicios candidatos: Se realiza el análisis del inventario de servicios, se identifican los servicios que habrán de cumplir con el caso de uso más complejo.

Hitos:

- Aprobación del *blue print* de servicios.

## **Elaboración**

Objetivos:

- Analizar el dominio del problema, su alcance, requerimientos funcionales, no funcionales y de rendimiento.
- Ésta fase es la más importante de las cuatro, para el final de ésta fase se puede decir que la ingeniería dura está terminada y se enfrentará la decisión de si seguir con las fases de construcción y transición (Kruchten, 2003).
- Realizar un prototipo que incluye el desarrollo de los servicios web que cumplen con los casos de uso críticos detectados en la fase de inyección.

Actividades:

- Elaborar la arquitectura de los servicios web seleccionados en la fase de inyección y que cumplan con los casos de uso más complejos.
- Se diseña el contrato de los servicios web elegidos y se tiene todo para estimar la fase de construcción.

- Se ajustan los artefactos realizados en la fase de inyección.
- Se registran los servicios web en el Sistema de Inventario de Servicios SOA (ISSOA).
- Se desarrolla la lógica a partir de los contratos de los servicios web elegidos.
- Se prueba un prototipo funcional.

Artefactos:

- Especificaciones de casos de uso: Se representa de manera escrita el flujo de los casos de uso a un alto nivel.
- Perfiles de servicios y capacidades: Se realiza el diseño del contrato de los servicios web elegidos, se plasman por escrito su propósito, sus entradas, salidas y esquemas a utilizar.
- Plan de pruebas: Se analizan los posibles tipos de pruebas a los que serán sometidos los servicios web.
- Diagramas de actividad: Diagramas de alto nivel (nivel de sistema) que representarán el flujo de los casos de uso identificados, sólo el camino feliz.
- Diagrama de entidad relación físico: se realiza el diccionario de datos y un diagrama físico de E-R.
- Diagramas de secuencia (nivel sistema): Estos diagramas representan los servicios y capacidades (en vez de objetos y clases) involucradas en los flujos de secuencia de los casos de uso, sólo se representa el camino feliz y se ocupa notación definida por SOA para identificar los servicios.
- Contratos de los servicios web. A partir de los perfiles de servicios y capacidades definidos se generan los contratos mediante el uso del *Web Services Description Language (WSDL)*.
- Prototipo funcional con los servicios web implementados que cumplen con los casos de uso elegidos en fase de inyección.
- *Blue print* de Inventario de servicios actualizado.

Hitos:

- Aprobación del prototipo funcional.

**Construcción**

Objetivos:

- Minimizar los costos de desarrollo mediante la optimización de los recursos.
- Lograr la calidad adecuada tan rápidamente como sea práctico.
- Lograr versiones útiles (alfa, beta, y otras versiones de prueba) tan rápidamente como sea práctico.
- Durante la fase de construcción, se desarrollan todos los servicios web restantes y se integran en el producto. Los servicios web son probados a fondo según el plan de pruebas realizado en la fase anterior.
- Poner el producto en manos de los usuarios.
- Hacer pruebas beta frente a las expectativas de los usuarios.

Actividades:

- La gestión de recursos y optimización de procesos (plan de desarrollo de software).
- Diseño y desarrollo de servicios web restantes y la ejecución de las pruebas con los criterios de evaluación definidos en el plan de pruebas.

Artefactos:

- SLA de todos los servicios web.
- El producto listo para ponerlo en las manos de los usuarios.
- Reporte de pruebas.

Hitos:

- Aprobación de la versión beta.

- Al término de ésta fase de debería preguntar lo siguiente:
- ¿Es esta versión del producto estable y lo suficientemente maduro como para ser desplegado para los usuarios?
- ¿Están todos los interesados listos para la transición a los usuarios?
- ¿Son los gastos reales de recursos frente a los previstos todavía aceptables?

## **Transición**

### Objetivos:

- En esta fase es lograr el consentimiento de los *stakeholders*, de que la implementación concuerda con la visión que se tenía del sistema.

### Actividades:

- La capacitación del personal.
- Actividades de ajuste, incluida la corrección de errores y la mejora del rendimiento en los servicios web y la facilidad de uso del sistema.

### Artefactos:

- En esta fase se tendrían que tener actualizados todos los artefactos más:
  - Manual de usuario.

### Hitos:

- Estrenar el producto

## **Capítulo 4: Solución propuesta**

En este capítulo se presentan los resultados de aplicar la metodología SOA-RUP propuesta para la DISF en un proyecto enfocado al análisis y diseño de un sistema que administre los servicios web existentes, cumpliendo así con las necesidades de la DISF.

### ***Objetivo del proyecto***

Desarrollar una solución orientada a servicios para automatizar los procesos de administración de servicios web, esquemas y capacidades de servicios, permitiéndole al custodio del inventario ahorrar tiempo en sus actividades de gobernanza de servicios y mostrando información del inventario de servicios de acuerdo a las necesidades de los interesados.

El sistema deberá ser entregado el 7 de noviembre del 2014.

## ***Fase de inceptión***

Los siguientes entregables son los realizados en la fase de inicio.

- Documento de visión.
- Matriz de requerimientos.
- Modelo de casos de uso.
- Plan de desarrollo de software.
- EDT (Estructura de descomposición de trabajo).
- *Blue print* de los servicios candidatos.
- Diagramas de actividad del Caso de Uso gestionar servicios.

A continuación se muestran los entregables más importantes realizados en esta fase para la solución orientada a servicios ISSOA.

### ***Visión del sistema***

#### *Propósito Del Documento De Visión*

Este documento tiene como objetivo describir los requerimientos de alto nivel de la DISF para la construcción de un sistema informático Web orientado a servicios y así automatizar los procesos de control de inventario. El sistema se denomina ISSOA (INVENTARIO DE SERVICIOS SOA).

#### *Alcance Del Documento De Visión*

El documento está dirigido al arquitecto de software para el diseño y a los programadores para la construcción del ISSOA. El documento será actualizado permanentemente.

<b>ISSOA</b>	
<b>Descripción del Problema</b>	
<i>El problema de:</i>	<p>Tener un inventario de servicios en una wiki interna que a pesar de que puede ser accedida sólo por personal autorizado, una vez estando dentro todos pueden editar tanto el contenido desde el código HTML, como los documentos, sin llevar un control de cambios eficiente. De igual manera la búsqueda, se limita al código HTML, dejando de lado los documentos en los cuales se guarda la información importante acerca de los servicios, lo que a la larga provocará problemas de acceso a la información así como servicios o capacidades duplicadas.</p>
<i>Afecta a:</i>	<p>Usuarios técnicos: debido a que necesitan acceder a la información referente a los servicios de manera rápida y precisa, y la forma en que se presenta dicha información así como la navegación en la misma wiki carece de usabilidad y afecta el desempeño del usuario.</p> <p>Usuarios de negocio: hasta el momento, este tipo de usuario accede de manera ocasional a la documentación de la wiki, y en general carece de conocimiento técnico para explotar la manera en que la información es presentada. Sin embargo, si la información se presentara clasificada de acuerdo al segmento del negocio donde es utilizada, podría ser de gran utilidad para un analista de negocio, promoviendo además, el alcance de uno de los objetivos estratégicos SOA que es la alineación entre el negocio y el área de TI.</p> <p>SOA: La implementación de ésta arquitectura se basa en el</p>

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

	<p>desarrollo y manejo adecuado de diversos servicios que interactúan entre sí y que cumpliendo con ciertos principios logran una solución de gran alcance para el negocio. Si dichos servicios no son administrados de manera eficiente, hay un riesgo elevado de romper con algunos principios que marca la misma metodología tales como: descubrimiento y reusabilidad, mucho de esto debido a la implementación redundante de servicios que se pudiera llegar a manejar. Otro aspecto importante que puede verse afectado es la gobernabilidad de los servicios.</p>
<p><i>El impacto asociados:</i></p>	<p>Manejo poco eficiente del inventario de servicios.</p> <p>Riesgo potencial de redundancia de código.</p> <p>Cada nueva aplicación desplegada con código redundante, significa un incremento en la carga de TI (utilización de espacio de almacenamiento que podría ser evitado, mayor consumo de recursos de infraestructura, mayor esfuerzo de mantenimiento, etc.).</p>

**Tabla 5 Descripción del problema document vision.**

<b>Descripción del Producto</b>	
<i>Para:</i>	<ul style="list-style-type: none"> <li>● Custodio del inventario de servicios.</li> <li>● Usuarios técnicos (gerente, subgerente del área, analistas).</li> <li>● Usuarios de negocio.</li> </ul>
<i>Quienes:</i>	<ul style="list-style-type: none"> <li>● Requieren controlar el inventario de servicios.</li> <li>● Requieren disminuir el tiempo en la búsqueda de servicios del inventario SOA.</li> <li>● Requieren tener un fácil acceso a la información de los servicios.</li> <li>● Requieren que la información sea presentada de acuerdo a sus necesidades.</li> <li>● Requieren alinear el negocio con las TI.</li> </ul>
<i>Nombre del Producto:</i>	ISSOA (INVENTARIO DE SERVICIOS SOA).
<i>Que:</i>	Debe permitir la gestión, consulta y control de los datos de: los perfiles de los servicios SOA, los perfiles de las capacidades de los servicios SOA, las entidades de negocio, los procesos de negocio, así como el acceso a una vista técnica y una vista de negocio.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<i>No como:</i>	La wiki interna actual que no agiliza la gestión, consulta y control del inventario de servicios y no muestra de forma adecuada la información de acuerdo con las necesidades de los usuarios. Los documentos como perfiles de servicios y capacidades, que no reflejan la situación actual del inventario de servicios.
-----------------	--

Tabla 6 Descripción del problema ISSOA document de vision.

*Lista de stakeholders*

<b>Puesto</b>	<b>Rol</b>	<b>Responsabilidades</b>
Gerente	Sponsor	Patrocinar el proyecto.
Subgerente de sistemas	Cliente	Aprobar los avances del proyecto y proveer requerimientos del sistema.
Analista 1	Analista, Diseñador, Ingeniero de procesos, Especialista en gobernanza	Identificar requerimientos. Identificar servicios y capacidades. Generar arquitectura de la solución orientada a servicios. Hacer cumplir preceptos y procesos. Personalizar la metodología propuesta para el proyecto.
Analista 2	Analista, Desarrollador, Diseñador de interfaz,	Identificar requerimientos. Identificar servicios y capacidades. Implementar la IGU.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Analista 3	Administrador de base de datos, Analista BD.	Identificar requerimientos de base de datos. Identificar entidades de base de datos. Generar objetos y documentación de base de datos.
Analista 4	Cliente de negocio	Proveer procesos principales y secundarios de negocio para su mapeo con los servicios
Analista 5	Cliente de sistemas	Proveer requerimientos del sistema.

Tabla 7 lista de interesados y responsabilidades, documento de visión

**Matriz de requerimientos**

ID	Descripción	Estado	Fecha	Categoría	Quién registra/modifica
<b>Administrar perfiles de servicio</b>					
1.	<p>El sistema debe permitir al usuario agregar servicios con los siguientes datos de su perfil:</p> <p>DATOS DEL SERVICIO:</p> <ul style="list-style-type: none"> <li>• *Nombre.</li> <li>• *Propósito.</li> <li>• *Custodio.</li> <li>• *liga del SLA.</li> <li>• *modelo de servicio.</li> <li>• Estado.</li> <li>• Versión.</li> <li>• Dependencias.</li> <li>• *Tipo de implementación.</li> </ul>	<p>Aprobado</p> <p>Pendiente</p> <p>Aprobado</p>	<p>14/04/14</p> <p>16/04/14</p> <p>18/04/14</p>	<p>Funcional</p>	<p>JJSA</p> <p>ANALISTA</p> <p>JJSA</p>
2.	El sistema no debe borrar los servicios, únicamente se cambiará el estado a “jubilado” de los mismos.	Aprobado	14/04/14	Funcional	JJSA
3.	El custodio del inventario debe agregar sólo los servicios que cumplan con los estándares que marca la GSIF.	Aprobado	14/04/14	No funcional	JJSA

4.	El sistema debe permitir sólo al custodio del inventario y al administrador del sistema jubilar los servicios.	Aprobado	14/04/14	funcional	JJSA
5.	El sistema debe mostrar una lista con todos los servicios dados de alta para su fácil acceso con las siguientes columnas: <ul style="list-style-type: none"> <li>• Custodio.</li> <li>• Nombre.</li> <li>• Propósito.</li> <li>• Estado.</li> </ul> (Vista del Inventario).	Aprobado	14/04/14	funcional	JJSA
		Pendiente	18/04/14		JJSA
		Aprobado	18/04/14		JJSA
6.	El sistema debe mostrar en el inventario completo una parte donde se distingan los servicios populares clasificados de acuerdo al número de clientes que tengan. Mostrando el nombre del servicio, sus capacidades y clientes y proveedores de esas capacidades. Sólo se deben tomar en cuenta los servicios con estado de producción.	Aprobado	14/04/14	funcional	JJSA
		Pendiente	16/04/14		JJSA
		Aprobado	18/04/14		ANALISTA

7.	El sistema debe mostrar filtros en las columnas (Custodio, Nombre, Estado) de los servicios mostrados, facilitando la búsqueda de los mismos.	Aprobado	14/04/14	funcional	JJSA
8.	El sistema debe mostrar los mismos datos del servicio que se solicitaron en el registro al momento de modificar su información.	Aprobado	14/04/14	funcional	JJSA
9.	El sistema debe mostrar una interfaz amigable e intuitiva.	Aprobado	14/04/14	No funcional	JJSA
10.	El sistema debe mostrar las capacidades del servicio disponibles con la opción de consultarla detalladamente (perfil de la capacidad).	Aprobado	14/04/14	funcional	JJSA
11.	Sólo el custodio del inventario y el administrador del sistema tienen derecho a modificar los perfiles del servicio.	Aprobado Pendiente Aprobado	14/04/14 16/04/14 16/04/14	Funcional	JJSA JAFD JJSA

12.	El sistema debe permitir modificar los datos del servicio cuando se muestren los detalles del mismo. Ésta opción sólo está disponible para custodio de inventario y administrador de sistema.	Aprobado	14/04/14	Funcional	JJSA
13.	El sistema debe notificar vía correo electrónico cada que se agregue un servicio a todos los analistas de informática.	Aprobado	14/04/14	Funcional	ANALISTA
<b>Administrar Perfil de las capacidades del servicio</b>					
14.	El perfil del servicio debe ser agregado con al menos el perfil de una capacidad.	Aprobado	14/04/14	Funcional	JJSA
15.	El usuario podrá registrar el perfil de una capacidad con los siguientes datos: <ul style="list-style-type: none"> <li>• *Nombre.</li> <li>• *Propósito.</li> <li>• Clientes SOA.</li> <li>• Clientes no SOA.</li> <li>• Versión.</li> <li>• *Entrada.</li> <li>• *Estado</li> <li>• *Salida.</li> <li>• *Aplicación.</li> <li>• *Palabras clave (tags).</li> </ul>	Aprobado	14/04/14	Funcional	JJSA
		Pendiente	16/04/14		JJSA
		Aprobado	18/04/14		ANALISTA

16.	Sólo el administrador del sistema y el custodio del inventario pueden modificar los perfiles de la capacidad.	Aprobado	14/04/14	Funcional	JJSA
17.	El sistema debe permitir editar los datos de la capacidad al momento de mostrarlos.	Aprobado	14/04/14	Funcional	JJSA
18.	En caso de que el usuario no encuentre el tipo de dato complejo para asignar una entrada o salida de una capacidad el sistema debe permitir crearlo.	Aprobado	14/04/14	Funcional	JJSA
19.	El sistema debe permitir la búsqueda de capacidades para el descubrimiento de servicios bajo los siguientes criterios: nombre, estado, palabra clave o aplicación.	Aprobado	14/04/14	Funcional	JJSA
20.	El sistema debe permitir asignar a las entradas o salidas de la capacidad tipos de datos simples (entero, cadena, etc.).	Aprobado	14/04/14	Funcional	JJSA

<b>Generar vista de negocio</b>					
21.	El sistema debe generar una vista de negocio donde se mostrarán los procesos, asociados con actividades que a su vez se asocian con aplicaciones atendidas por capacidades SOA y que pertenecen a un servicio.	Aprobado	14/04/14	Funcional	JJSA
22.	La vista de negocio puede ser accedida por los siguientes usuarios:  Analista de negocio, custodio del inventario, administrador del sistema, analista de dominio.	Aprobado Propuesta Aprobado	14/04/14 14/04/14 16/04/14	Funcional	JJSA JJSA ANALISTA
23.	El sistema debe mostrar en círculos los servicios con su nombre para una fácil identificación.	Aprobado	14/04/14	Funcional	JJSA
24.	El sistema debe permitir acceder al perfil del servicio desde la vista de negocio al momento de elegir un servicio.	Aprobado	14/04/14	Funcional	JJSA

<b>Generar vista técnica.</b>					
25.	El sistema debe generar una vista para usuarios técnicos que contemple las siguientes tres capas: <ul style="list-style-type: none"> <li>•Servicios entidad.</li> <li>•Servicios de utilería.</li> <li>•Servicios de tarea.</li> </ul>	Aprobado	14/04/14	Funcional	JJSA
26.	El usuario puede acceder a los perfiles de servicios desde la vista técnica.	Aprobado	14/04/14	Funcional	JJSA
27.	El sistema debe presentar las capas con colores diferentes para una fácil visualización.	Aprobado	14/04/14	Funcional	JJSA
<b>Gestionar roles.</b>					
28.	Dentro del sistema deben existir los siguientes roles: Analista de informática, analista de negocio, custodio del inventario, administrador del sistema y analista de dominio.	Propuesta Aprobado	14/04/14 16/04/14	Funcional	JJSA ANALISTA

29.	El analista de negocio puede consultar lo siguiente: <ul style="list-style-type: none"> <li>• Vista de negocio.</li> <li>• Inventario de servicios. (Servicios y capacidades).</li> </ul>	Propuesta Aprobado	14/04/14 18/04/14	Funcional	JJSA ANALISTA
30.	El analista de informática puede consultar lo siguiente: <ul style="list-style-type: none"> <li>• Vista Técnica</li> <li>• Inventario de servicios.</li> </ul>	Propuesta Aprobado	14/04/14 18/04/14	Funcional	JJSA ANALISTA
31.	El custodio del inventario puede crear, consultar, modificar y eliminar lo siguiente: <ul style="list-style-type: none"> <li>• Vista de negocio.</li> <li>• Vista técnica.</li> <li>• Inventario.</li> </ul>	Propuesta Aprobado	14/04/14 18/04/14	Funcional	JJSA ANALISTA
32.	El administrador del sistema puede consultar, modificar y eliminar lo siguiente: <ul style="list-style-type: none"> <li>• Vista de negocio.</li> <li>• Vista técnica.</li> <li>• Inventario.</li> </ul>	Aprobado	14/04/14	Funcional	JJSA
33.	El analista de dominio puede consultar lo siguiente: <ul style="list-style-type: none"> <li>• Vista de negocio.</li> <li>• Vista técnica.</li> <li>• Inventario.</li> </ul>	Aprobado	14/04/14	Funcional	JJSA

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

34.	El sistema debe permitir al administrador del sistema agregar nuevos usuarios del inventario.	Aprobado Pendiente Aprobado	14/04/14 15/04/14 18/04/14	Funcional	JJSA JJSA ANALISTA
35.	El sistema debe permitir al administrador del sistema consultar los custodios existentes para su modificación.	Aprobado	14/04/14	Funcional	JJSA
<b>Bitácora y notificación.</b>					
36.	El sistema debe enviar notificaciones y registrar en la bitácora en los siguientes casos: <ul style="list-style-type: none"> <li>• Actualizar Capacidad.</li> <li>• Actualizar Servicio.</li> <li>• Crear Capacidad.</li> <li>• Crear Servicio.</li> </ul>	Aprobado	14/04/14	Funcional	JJSA

<b>Generar Reportes</b>					
37.	El sistema debe generar un reporte que contenga todos los servicios existentes en el inventario de servicios, separándolos de acuerdo a su estado.	Aprobado	14/04/14	Funcional	JJSA
38.	El sistema debe generar un reporte de los servicios populares, con una gráfica que represente los mismos, de acuerdo al número de clientes que tengan.	Aprobado	14/04/14	Funcional	JJSA
39.	El sistema debe generar un reporte basado en la vista técnica, que contenga gráficas donde se representen las aplicaciones y los servicios que atienden a los procesos.	Aprobado	14/04/14	Funcional	JJSA
40.	El sistema debe generar un reporte de la vista técnica con una gráfica que represente el número de servicios por modelo.	Aprobado	14/04/14	Funcional	JJSA
41.	El sistema debe generar un reporte del perfil de cada servicio existente en el inventario con los perfiles de sus capacidades.	Aprobado	14/04/14	Funcional	JJSA

### ***Plan de desarrollo de software (PDS)***

En este plan de desarrollo de software se plasma parte del análisis del sistema ISSOA y se definen hitos y calendarización del proyecto, así como el número y la duración de las iteraciones.

El proyecto está enfocado a SOA con una adaptación de RUP definida especialmente para la DISF y personalizada para este proyecto.

#### *Propósito del PDS*

Este documento provee información necesaria para controlar el proyecto a lo largo de sus fases e iteraciones. Los usuarios del Plan de Desarrollo del Software son:

- El líder del proyecto: lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- Equipo de desarrollo: lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

#### *Alcance*

El plan de desarrollo de software es una guía para el desarrollo del sistema en su primera versión y no pretende establecer una forma rígida y calendarizada de desarrollo, tanto las fechas y los diferentes artefactos de SOA y de RUP se irán adecuando según las necesidades.

### *Vista General del Proyecto*

Propósito, alcance y objetivos del proyecto:

El producto se trata de un sistema web construido a través de servicios web, que debe ayudar a lograr mayor gobernanza SOA. El sistema debe funcionar como un *registry* combinado con un repositorio facilitando la búsqueda de los servicios de la DISF y aumentando la reutilización de los mismos, también se deben generar dos vistas de los servicios, la primera de negocio en donde se mapean los servicios con los procesos de negocio y la segunda enfocada a los informáticos, donde se dividen los servicios de acuerdo a la clasificación que SOA propone (entidad, utilería y tarea).

### *Suposiciones y Restricciones*

Las suposiciones y restricciones impuestas al proyecto son las siguientes:

- Los procesos de la vista de negocio son identificados por el área de negocio.
- Un primer prototipo debe estar listo para el mes de agosto.
- El personal dedica tiempo parcial al proyecto.
- Hasta el momento no se cuenta con restricciones de presupuesto.

### *Entregables del Proyecto*

- Documento de visión.
- Matriz de requerimientos.
- Glosario.
- Plan de desarrollo de software.
- Estructura de descomposición de trabajo.
- Plan de pruebas.
- Reporte de pruebas.
- Especificaciones de casos de uso.

- Arquitectura del sistema
  - Modelo de casos de uso
  - Diagramas de secuencia por cada caso de uso.
  - Modelo de datos
  - Diagramas de actividad.
  - Blue print de servicios candidatos, capacidades de servicios candidatos y composiciones de servicio candidatas.
  - Blue print de inventario de servicios, capacidades de servicios y composiciones de servicio.
- Perfiles de servicio.
- Perfiles de capacidades.
- Manual de usuario.
- Prototipo inicial.
- Prototipo con funcionalidad del caso de uso más complejo.
- Versión 1.0 del sistema completo.

### Organización del Proyecto

Estructura organizacional del proyecto:

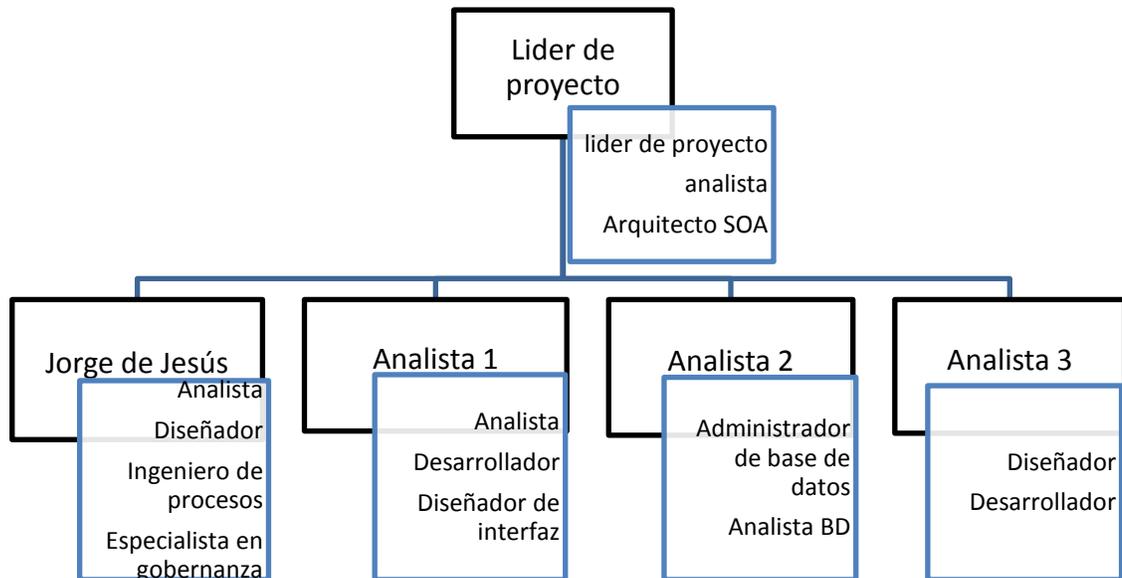


Ilustración 4 estructura organizacional proyecto.

### Roles y Responsabilidades

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos.
Analista de Sistemas	<p>Captura, especificación y validación de requerimientos, interactuando con el cliente y los usuarios mediante entrevistas.</p> <p>Define del alcance del sistema y requerimientos de alto nivel.</p> <p>Identifica servicios y capacidades.</p>

<b>Puesto</b>	<b>Responsabilidad</b>
Diseñador	Desarrolla del modelo de casos de uso, desarrollo de diagramas de actividad y diagramas de secuencia.
Diseñador de interfaz	Desarrollo de la interfaz gráfica de usuario.
Desarrollador	Implementación de servicios web y de la interacción con la interfaz gráfica de usuario.
Ingeniero de procesos	Personalización de metodología para el desarrollo de soluciones orientadas a servicios, hace cumplir la metodología así como las mejores prácticas de la administración de proyectos.
Administrador de base de datos	Realiza el diagrama de base de datos, genera el script y se encarga de las actualizaciones de la misma, así como la realización de todos los objetos de las base de datos.
Especialista en gobernanza.	Verifica que se cumplan preceptos y procesos dentro de las fases del ciclo de vida de servicios.
Arquitecto de servicios.	Identifica posibles patrones de diseño a utilizar y la implementación correcta de los mismos.

**Tabla 8 roles y responsabilidades proyecto**

*Plan del Proyecto*

Plan de fases RUP y ciclo de vida de servicios SOA:

<b>Fase</b>	<b>No. Iteraciones</b>
Fase de Inicio( <i>service-oriented analysis, service inventory analysis</i> )	1
Fase de Elaboración( <i>service oriented design, service logic design service development, service testing</i> )	3
Fase de Construcción ( <i>service development, service testing, service oriented design, service logic design</i> )	3
Fase de Transición	1

**Tabla 9: Fases RUP y fases ciclo de vida de servicios**

Fase	Descripción (Hitos)
Fase de Inicio	<p>En esta fase desarrollan los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión y posteriormente en el artefacto “matriz de requerimientos”. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Software. La aprobación del artefacto Visión el Plan de Desarrollo y el <i>blue print</i> de servicios marcan el final de esta fase.</p>
Fase de Elaboración	<p>En esta fase se analizan los requerimientos y se desarrolla un prototipo que incluye el desarrollo los servicios web que cumplen con el caso de uso más complejo, relevante o crítico del sistema.</p> <p>Se realiza un <i>Blue print</i> de servicios candidatos, capacidades de servicios candidatos y composiciones de servicio candidatas. La revisión y aceptación del prototipo del sistema y del blue print de servicios marcan el final de esta fase.</p> <p>En esta fase también se hace una revisión general del estado de los artefactos elaborados hasta este punto para ajustarlos, en caso de ser necesario.</p>

Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando la arquitectura del sistema y el blue print de servicios. El producto se construye en base a 3 iteraciones, cada una produciendo una versión estable a la cual se le aplican pruebas. El hito que marca el fin de esta fase es la versión 1.0 del sistema final, lista para ser entregada a los usuarios para realizar pruebas beta.
Fase de Transición	En esta fase se migra el sistema a los diferentes servidores de producción y se termina la documentación para el usuario.  El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales y todo el material de apoyo al usuario, la finalización de la capacitación a usuarios y la entrega final del producto.

Tabla 10: Plan por fase

### Objetivos de la Iteración

Fase	No de Iteración	Objetivos
Incepción	1	<p>Definir el alcance del sistema, en esta iteración se realizan los siguientes artefactos(en su primera versión):</p> <ul style="list-style-type: none"><li>• Documento de visión.</li><li>• Matriz de requerimientos.</li><li>• Plan de desarrollo de software.</li><li>• Modelo de casos de uso.</li><li>• Blue print de los servicios candidatos.</li><li>• Diagramas de actividad de los CU más complejos.</li></ul> <p>Al final de esta iteración se busca la aprobación total del proyecto.</p>

Elaboración	1	<p>Se identifican los casos de uso más complejos.</p> <p>Se realiza un blue print de los servicios candidatos y una primera versión de la arquitectura del sistema.</p> <p>Se realizan los siguientes artefactos para los servicios web que cumplan con los casos de Uso más complejos:</p> <p>Diagramas de secuencia.</p> <p>Especificaciones de los CU.</p> <p>Plan de pruebas.</p> <p>Se actualiza el blue print de los servicios candidatos. Se realiza el diagrama físico de base de datos.</p> <p>Se crea un prototipo no funcional.</p> <p>Se hace el registro de los servicios web candidatos (perfiles de servicios y capacidades).</p>
Elaboración	2	<p>Se actualizan los artefactos afectados.</p> <p>Se generan los perfiles de servicios y capacidades que cumplan con los casos de uso más complejos.</p> <p>Se generan los contratos para los servicios que cumplan con los casos de uso elegidos.</p> <p>Se realizan los objetos de base de datos que necesiten los servicios web elegidos.</p> <p>Se genera el plan de pruebas.</p>

Elaboración	3	<p>Se implementa los servicios web necesarios para cumplir con el/los casos de uso elegidos.</p> <p>Y se complementan o actualizan los artefactos que así lo requieran.</p> <p>Se realiza el contrato, el perfil de servicio, los perfiles de capacidades y se implementa el servicio controlador que se comunica con la interfaz de usuario.</p> <p>Se integra el prototipo no funcional con el servicio controlador.</p> <p>Se realizan las pruebas de acuerdo al plan y se genera el reporte.</p>
Construcción	1	<p>Se analizan y diseñan los casos de uso gestionar esquemas, generar vista de negocio y generar vista técnica</p>
Construcción	2	<p>Se define y se implementa el servicio web ayudantelInventario que cumplirá con los casos de uso gestionar esquemas, generar vista de negocio y generar vista técnica. También se analiza y diseña el CU generar bitácora.</p>
Construcción	3	<p>Se realiza la integración del prototipo funcional con los servicios web implementados, se analiza y diseña el caso de uso gestionar usuarios y se ejecutan las pruebas de acuerdo el plan.</p>
Transición	1	<p>Se actualizan los artefactos necesarios.</p> <p>Se genera el manual de usuario.</p> <p>Se lleva a cabo una capacitación a usuarios.</p>

**Tabla 11: Objetivos por iteración**

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### *Estructura de descomposición de trabajo*

Nivel Fase

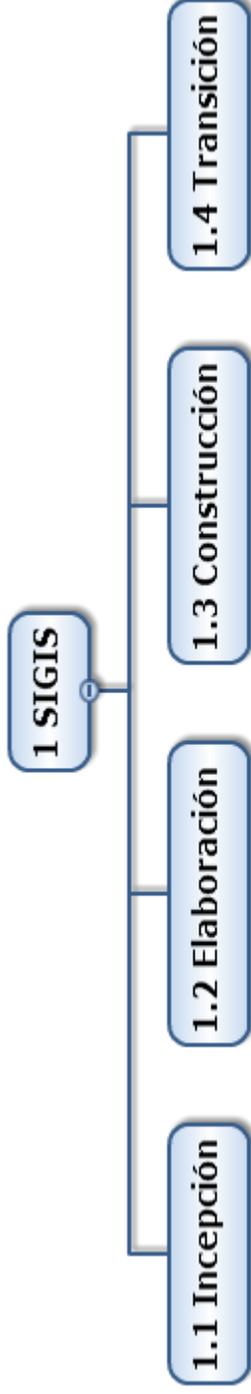


Ilustración 9: EDT nivel fase

## Fase Incepción

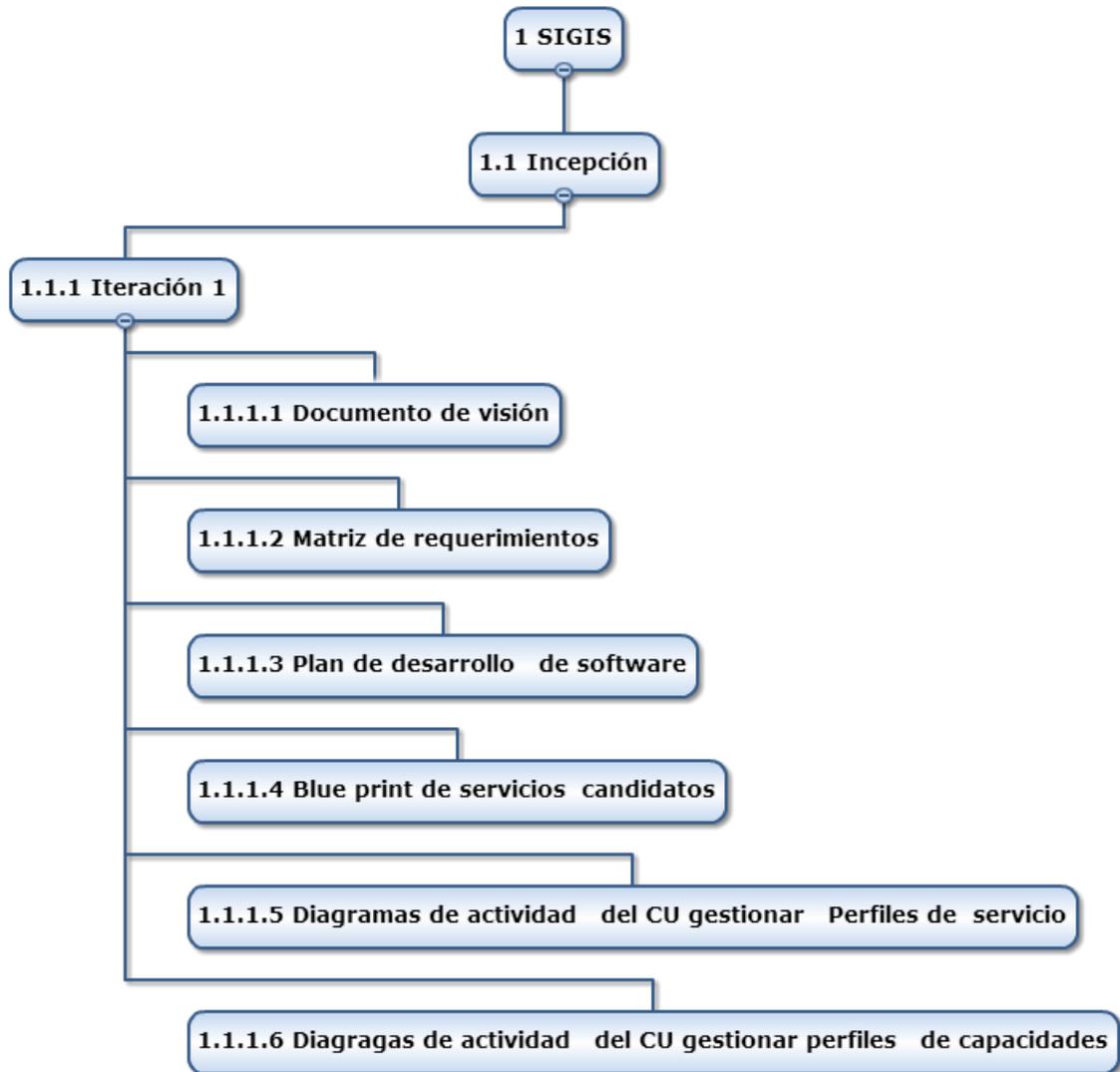


Ilustración 10 EDT fase inicio

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Fase de Elaboración

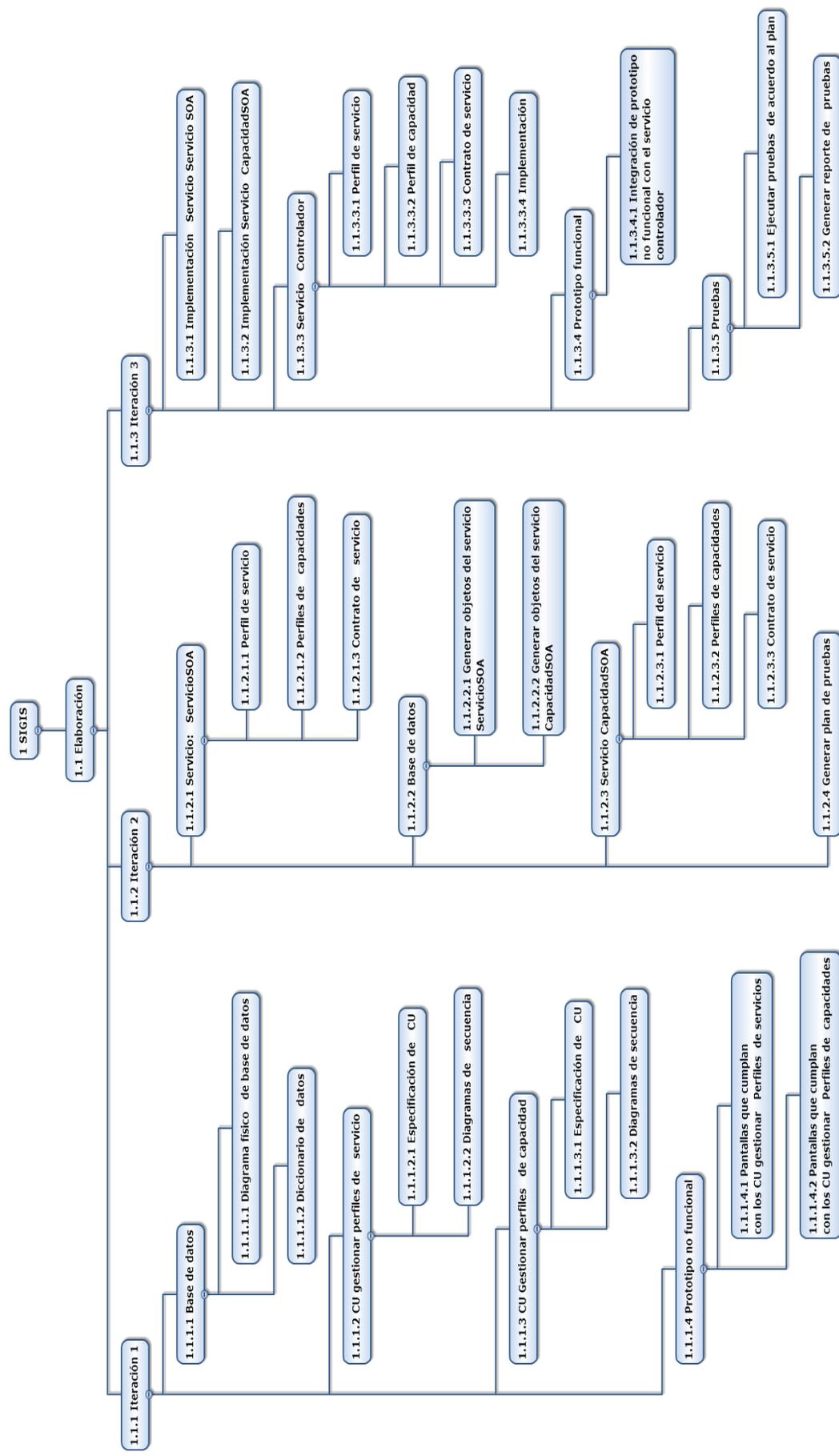


Ilustración 11: EDT fase de elaboración

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Fase de Construcción

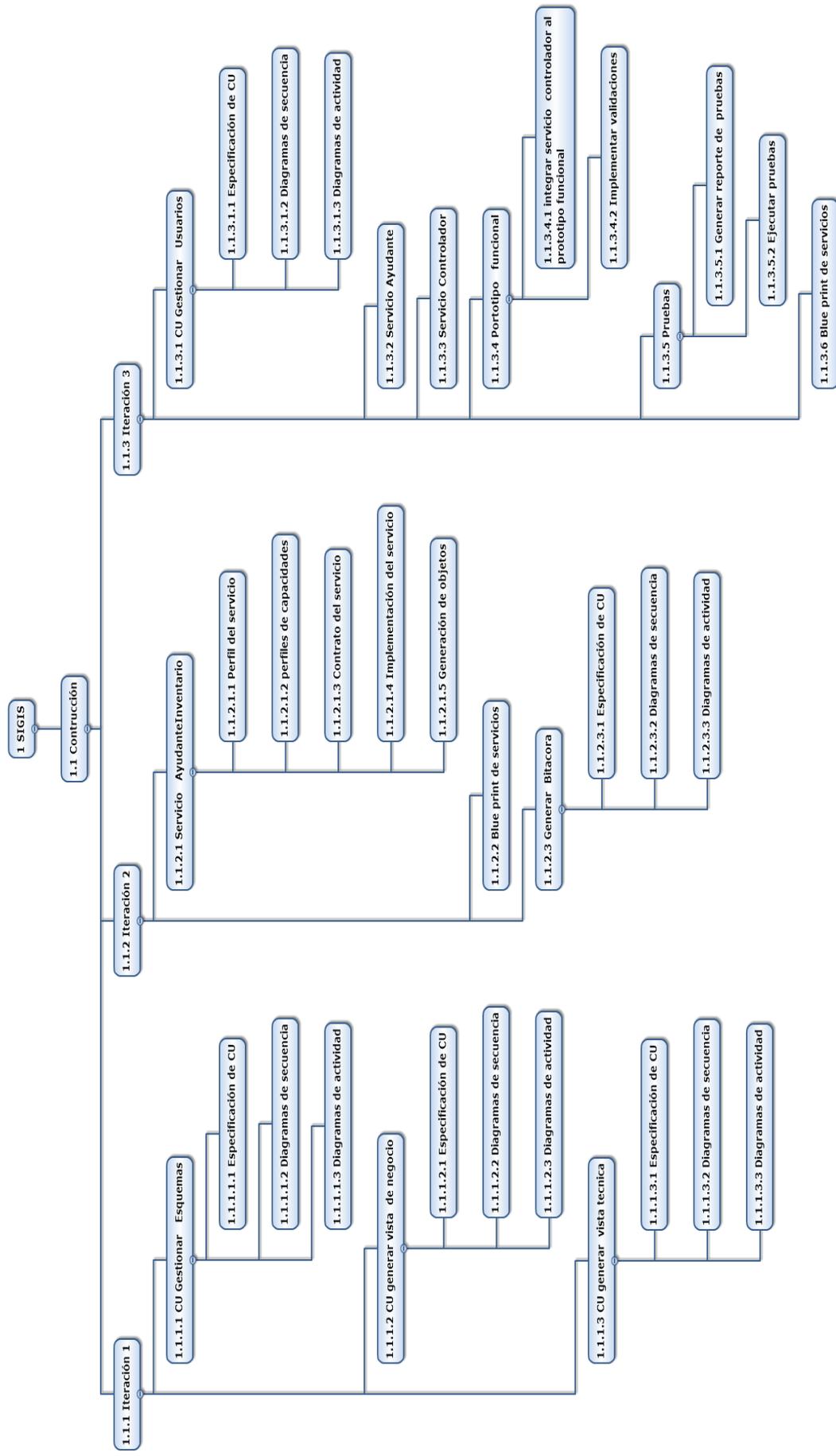


Ilustración 12: EDT fase de construcción.

## Fase de Transición

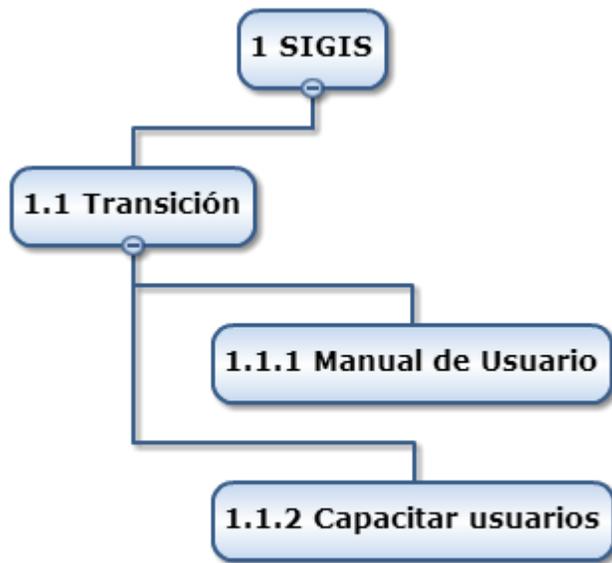


Ilustración 13: EDT fase de transición

### Estimación de Tiempo

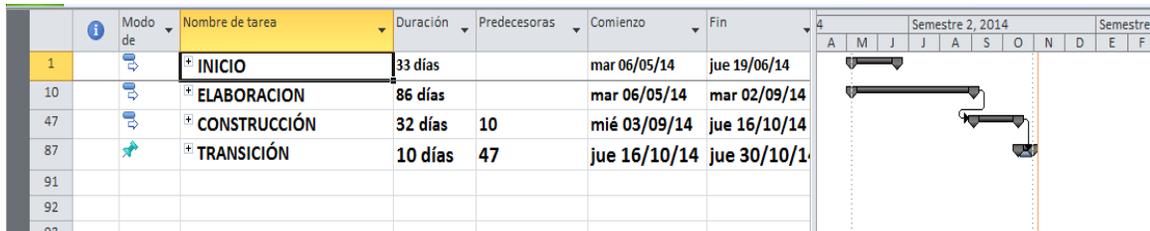


Ilustración 14: Gantt nivel fase

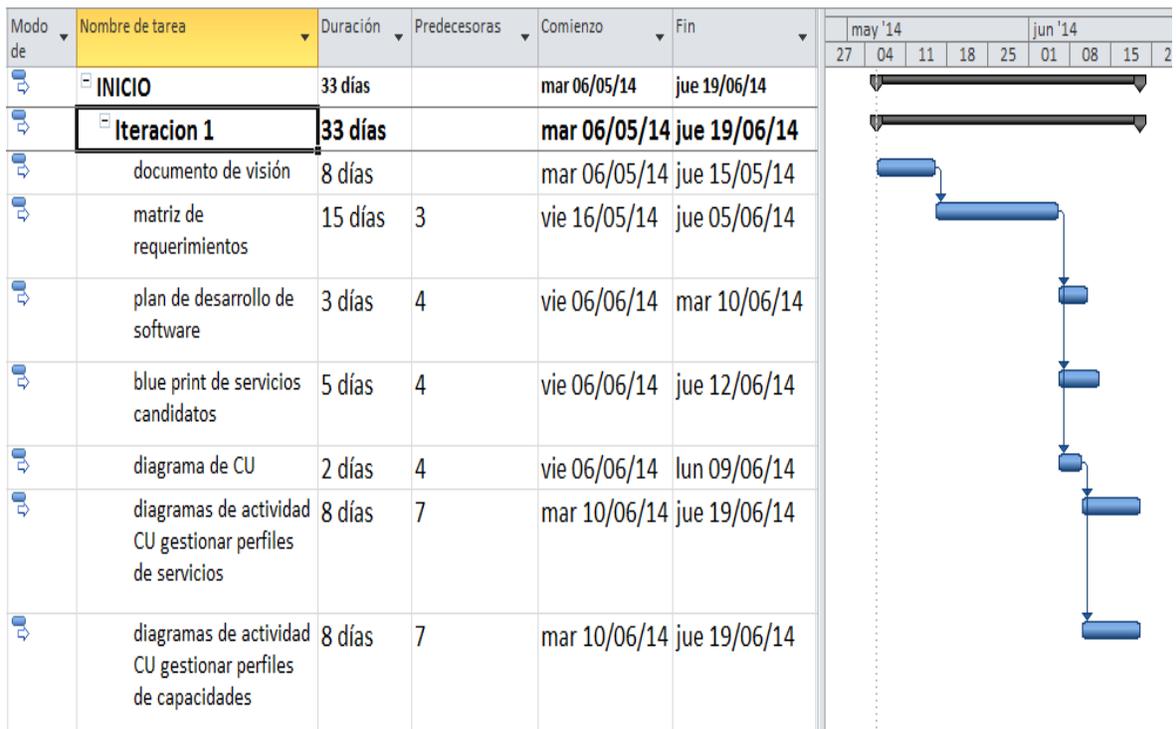


Ilustración 15: Gantt fase de inyección

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

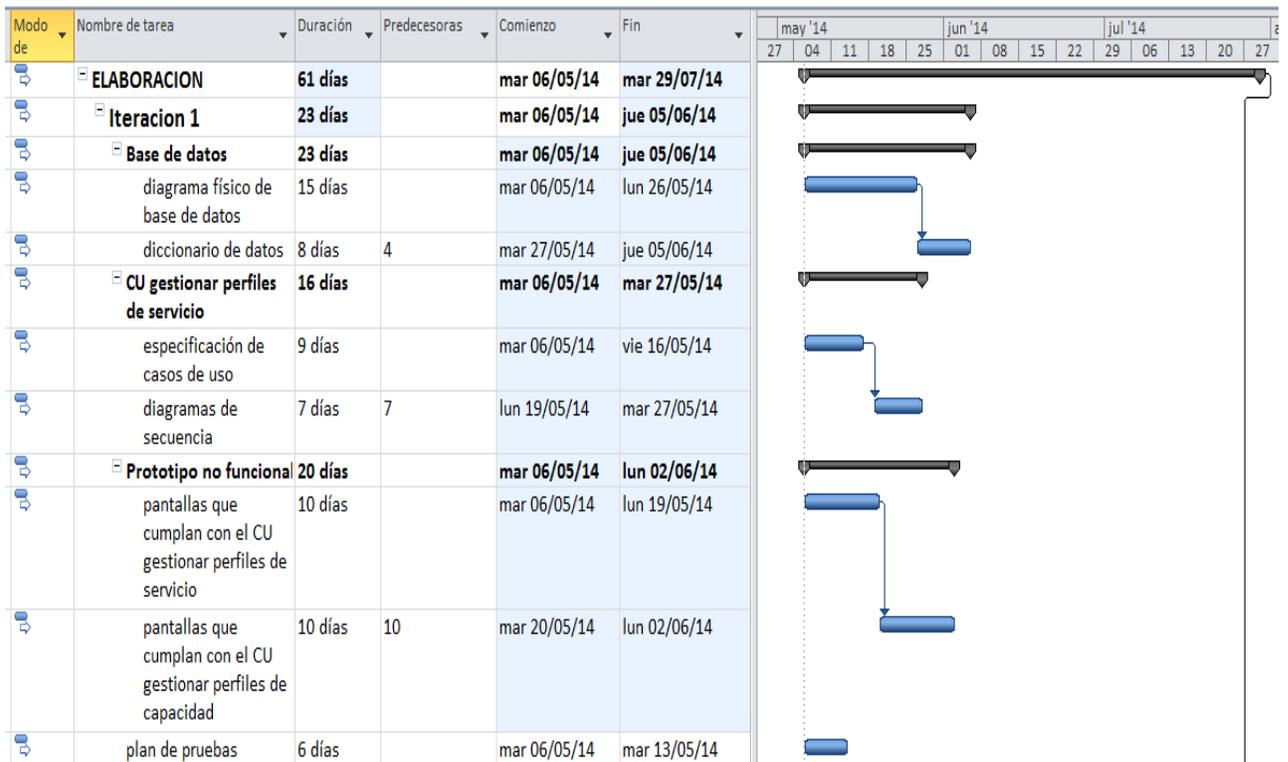


Ilustración 16: Gantt fase elaboración 1

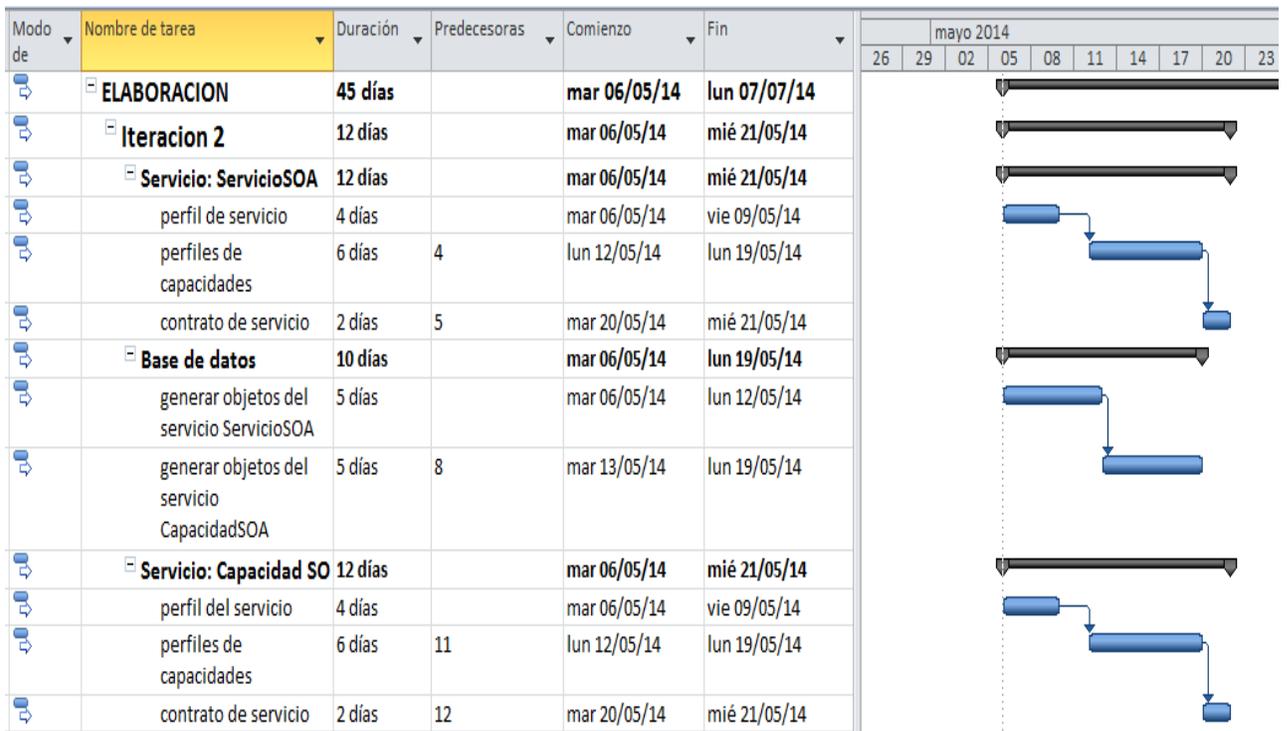


Ilustración 17: Gantt fase elaboración 2

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

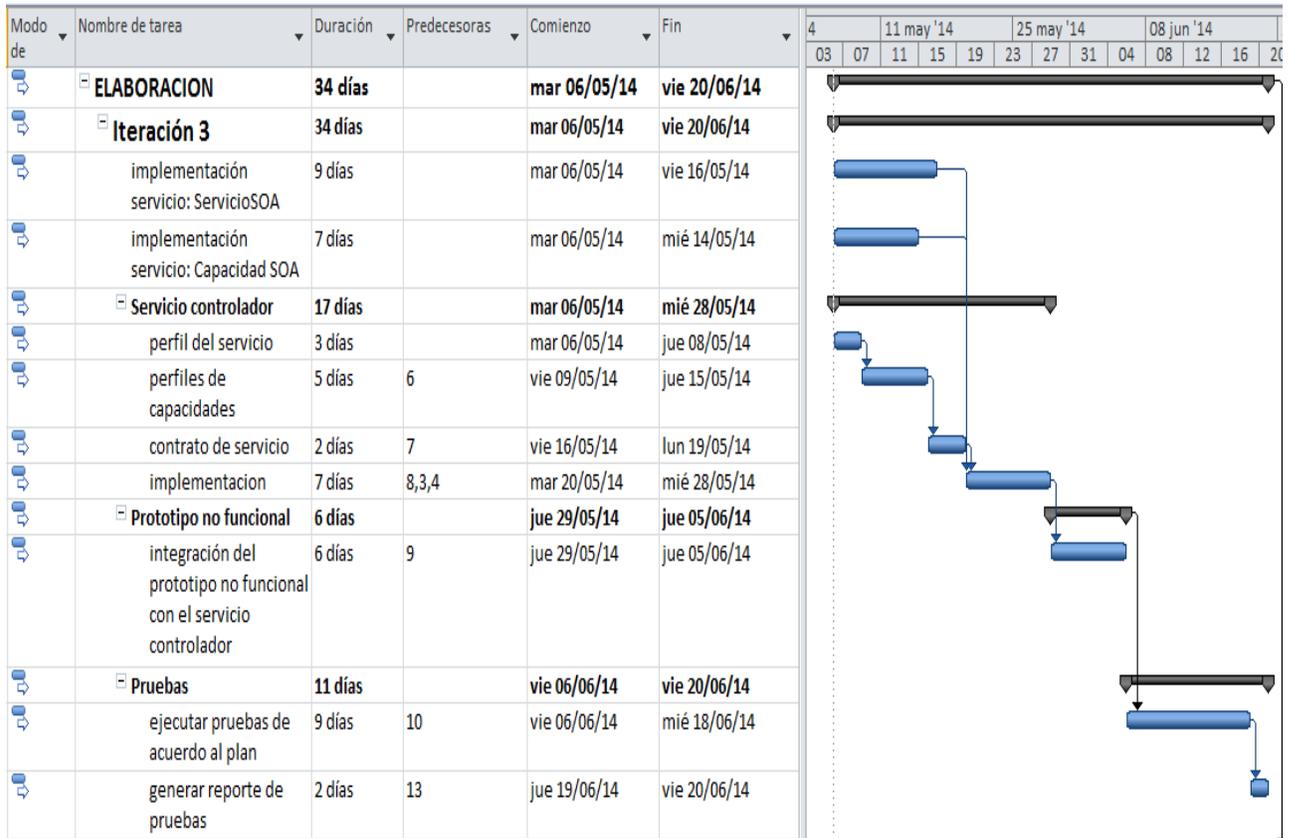


Ilustración 18: Gantt fase elaboración 3

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

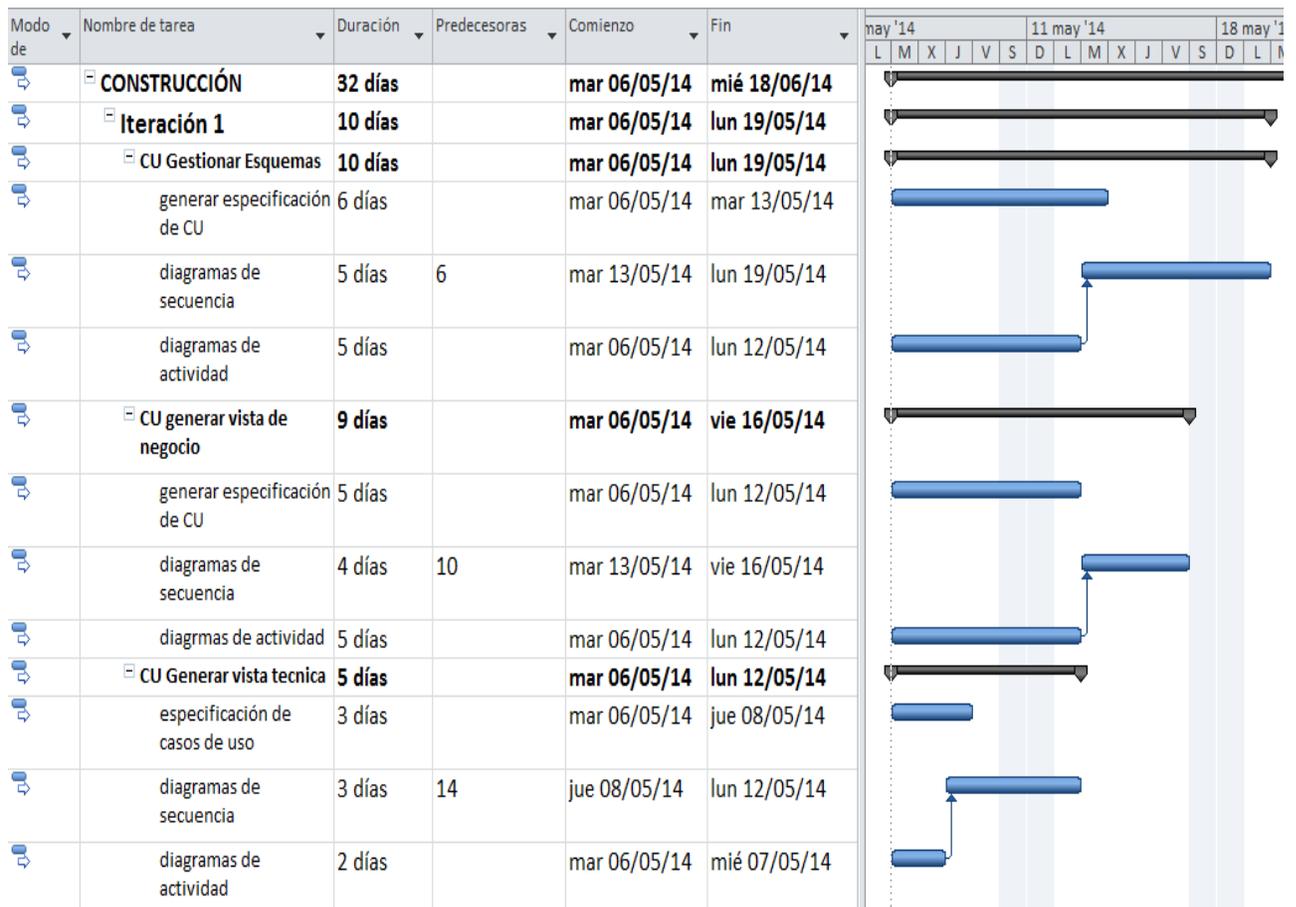


Ilustración 19: Gantt fase construcción 1

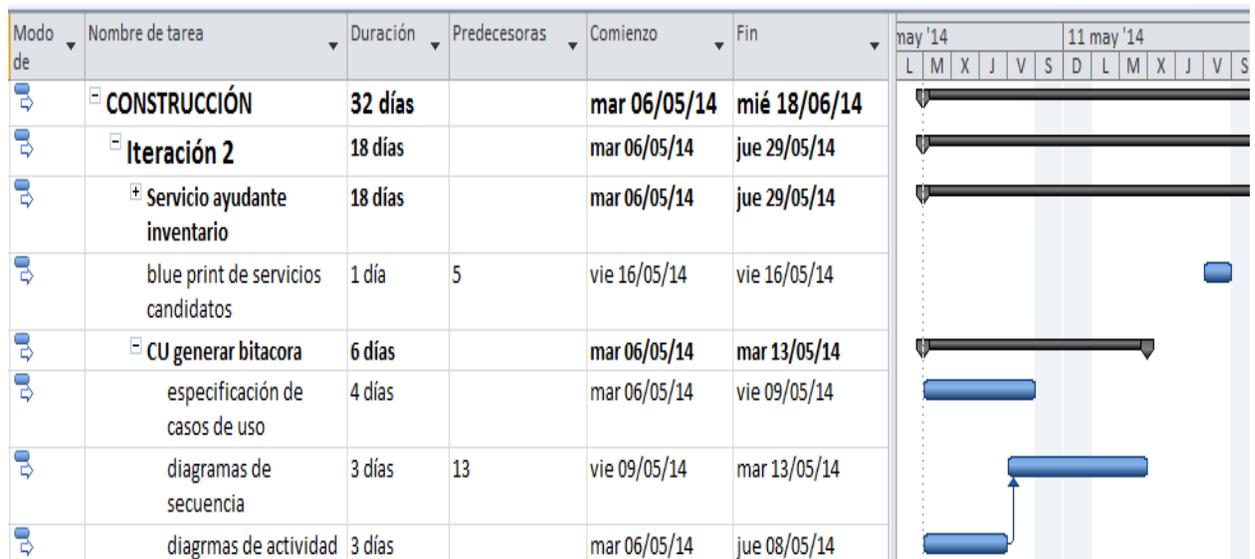


Ilustración 20: Gantt fase construcción 2

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

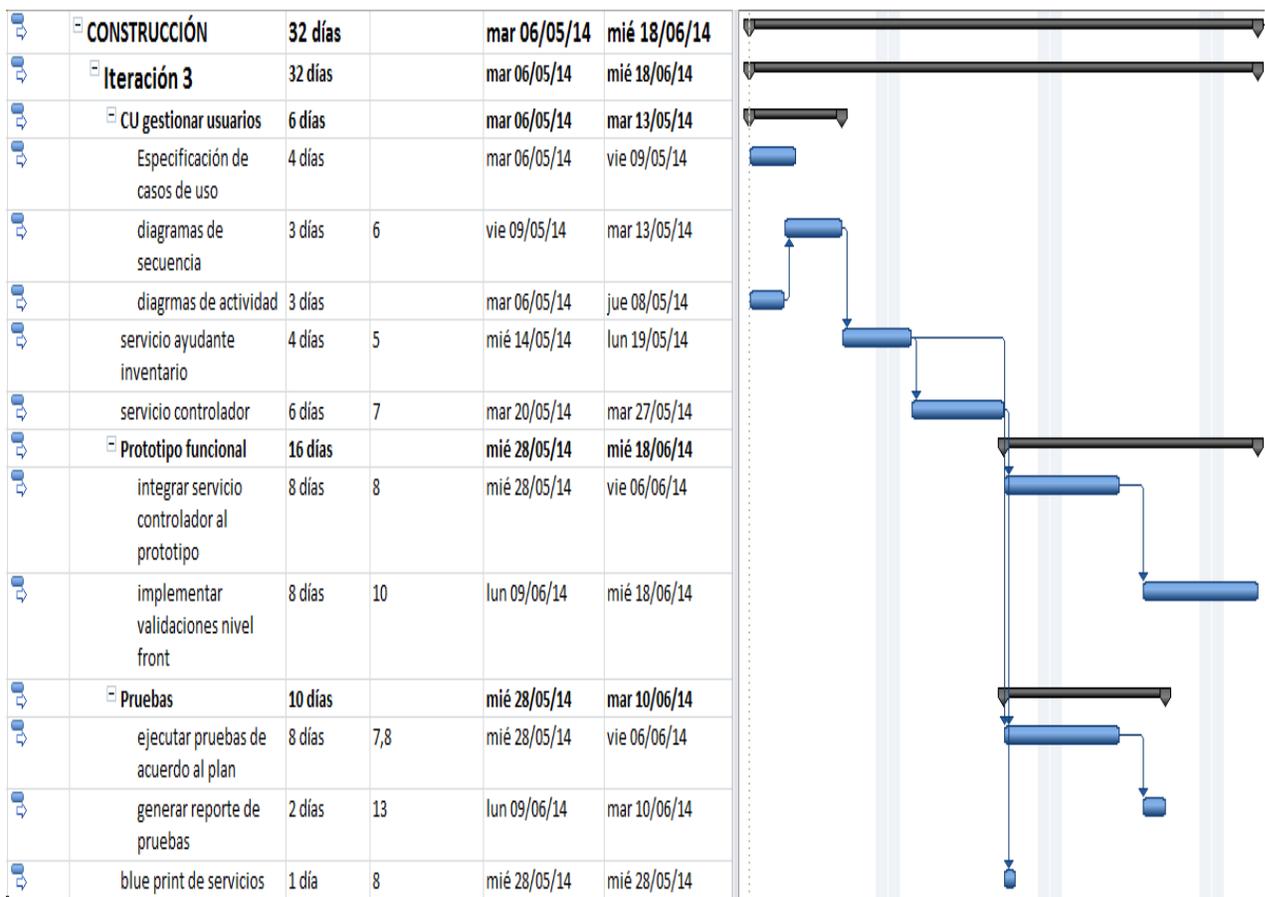


Ilustración 21: Gantt fase construcción 3

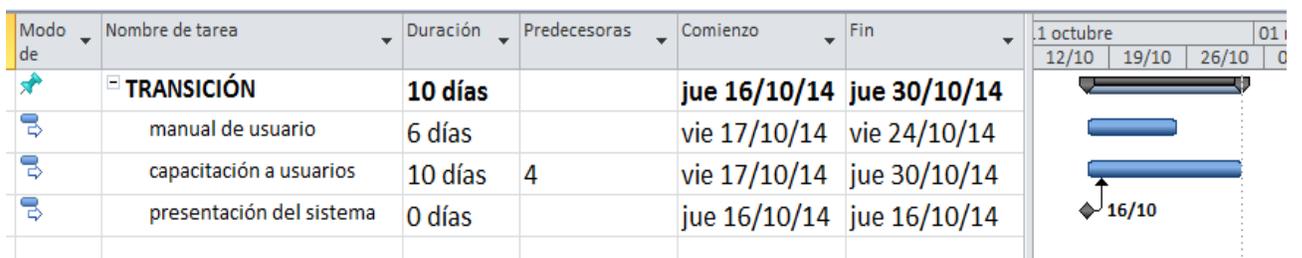


Ilustración 21: Gantt fase transición

Plan de reportes

Se le entregarán cada dos semanas reportes al líder de proyecto con los avances de las actividades planeadas. De igual manera habrá reuniones cada semana con el líder de proyecto con la finalidad de dar a conocer dudas y avances del mismo

### **Modelo de casos de uso**

Las siguientes imágenes representan el modelo de casos de uso del sistema.

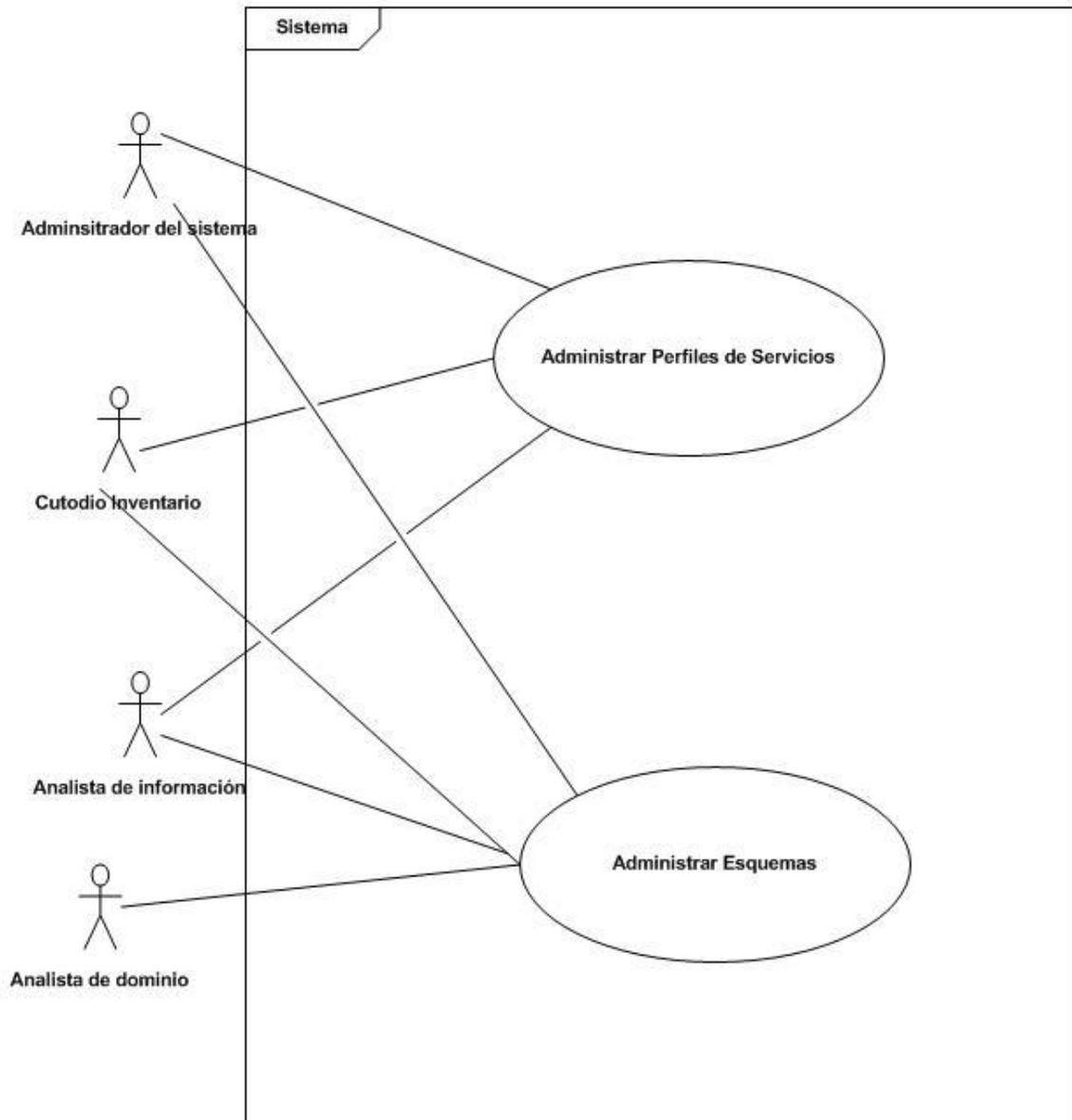


Diagrama de CU 1

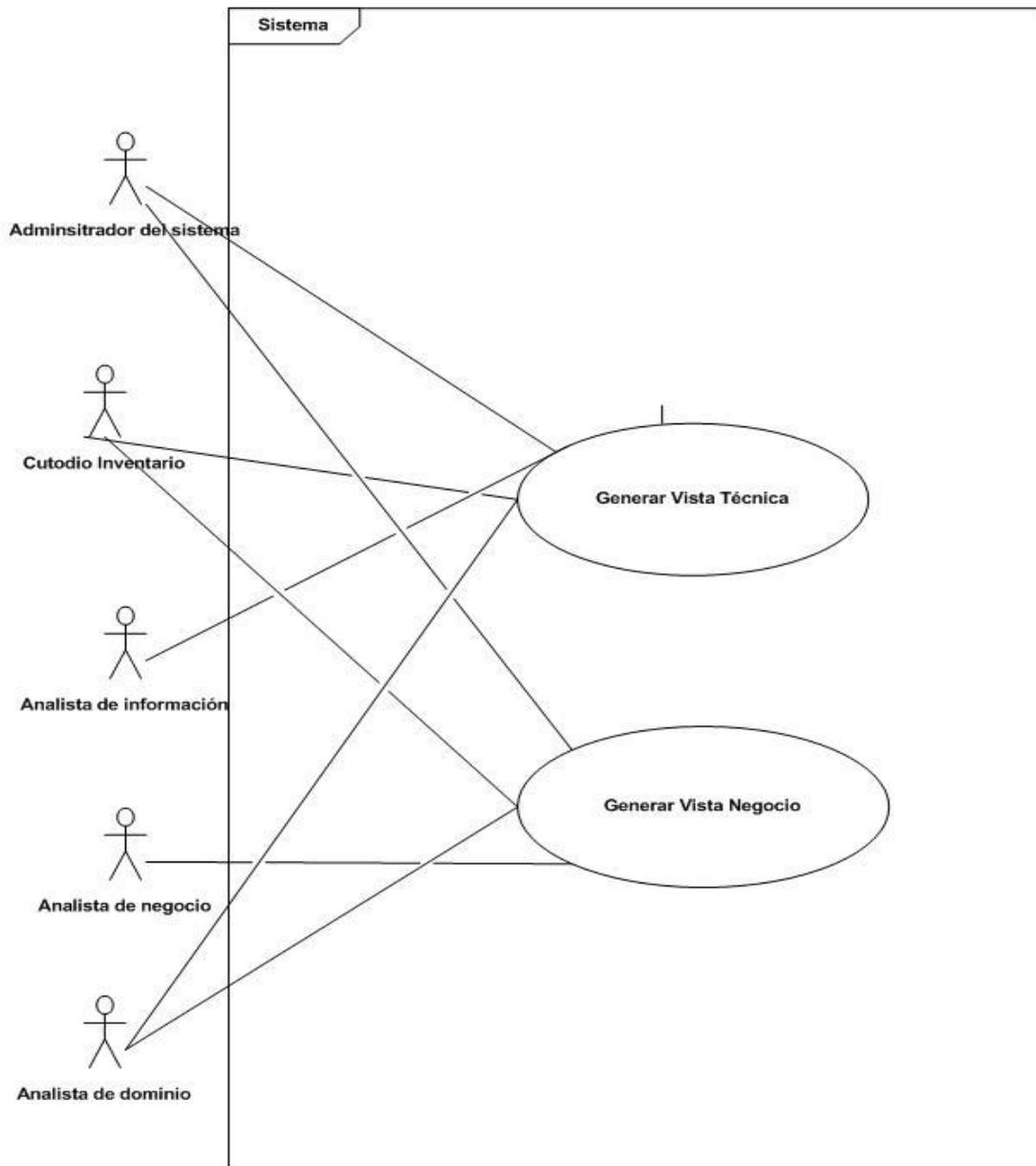


Diagrama de CU 2

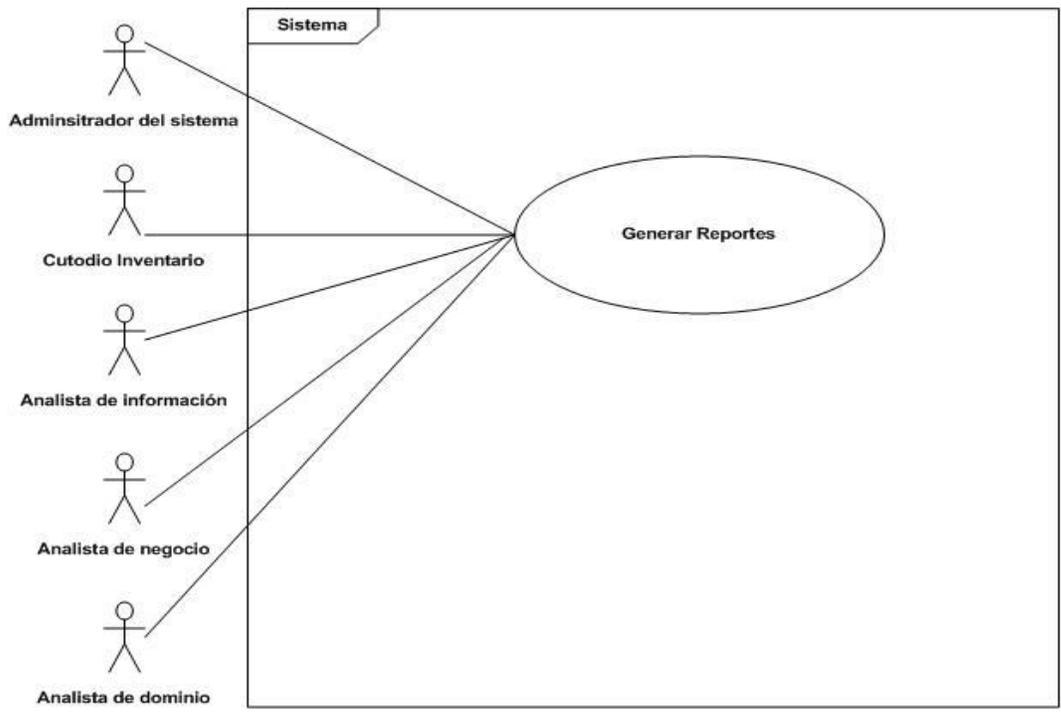


Diagrama de CU 3

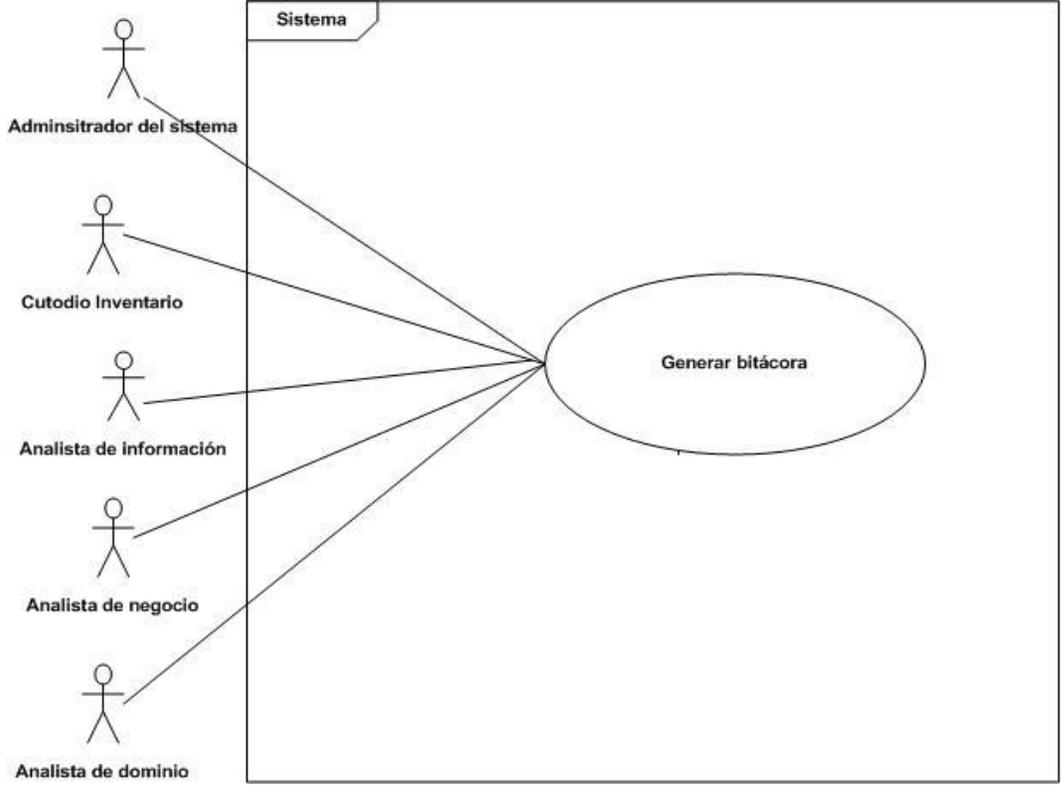


Diagrama de CU 4

### Diagramas de actividad "CU gestionar perfiles de servicios"

#### Crear Servicios

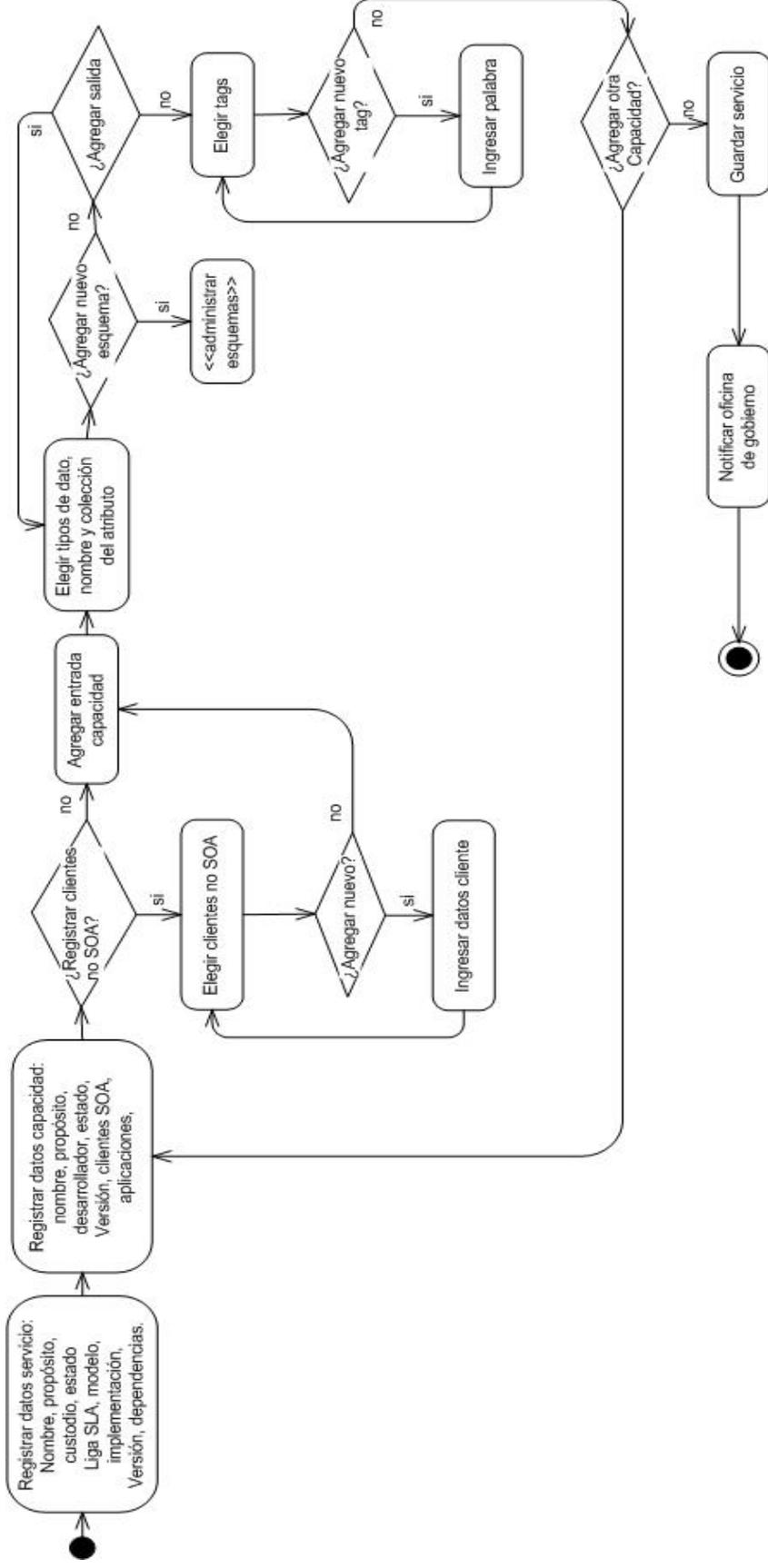


Diagrama de actividad 1: Crear servicios

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Actualizar servicios

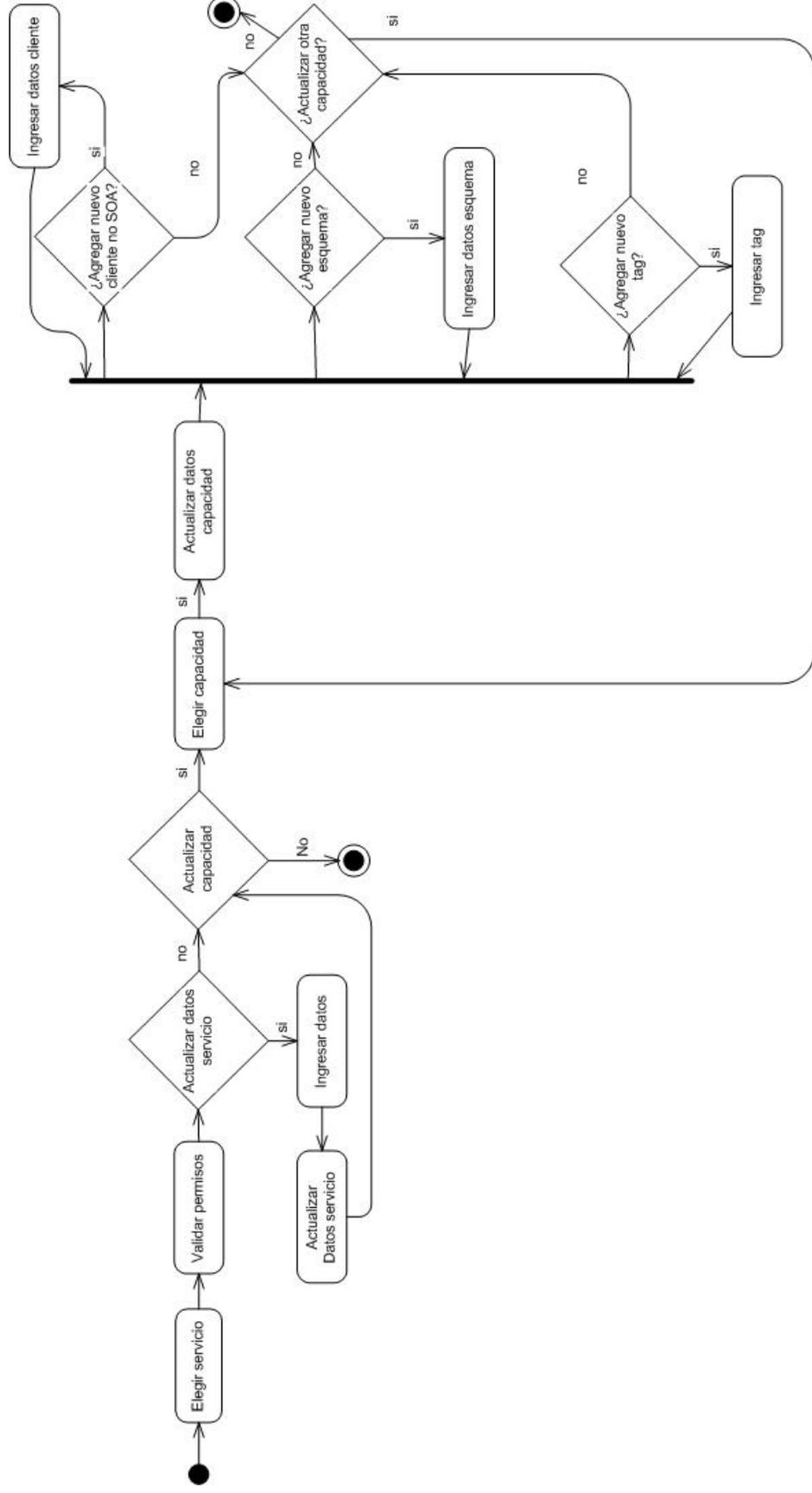


Diagrama de actividad 2: Actualizar servicios

**Blue print de servicios candidatos**



**Ciclo de Vida**

- Análisis
- Diseño
- Implementación
- Pruebas
- Producción

06/02/2014

**Ilustración 22: blue print de servicios**

### ***Fase de elaboración***

Los siguientes entregables son los realizados en la fase de inicio.

- Diagrama entidad relación físico de base de datos.
- Diccionario de datos.
- Especificaciones de caso de uso de los casos de uso: gestionar servicios y gestionar esquemas.
- Diagramas de secuencia de los casos de uso: gestionar servicios y gestionar esquemas.
- Prototipo no funcional de los casos de uso: gestionar servicios y gestionar esquemas.
- *Perfiles de servicios y capacidades* de los servicios web de los casos de uso: gestionar servicios y gestionar esquemas.
- Plan de pruebas.
- Contratos de servicios de los servicios web de los casos de uso: gestionar servicios y gestionar esquemas.

A continuación se muestran los entregables más importantes realizados en esta fase.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Diagrama físico de base de datos

### Primera parte

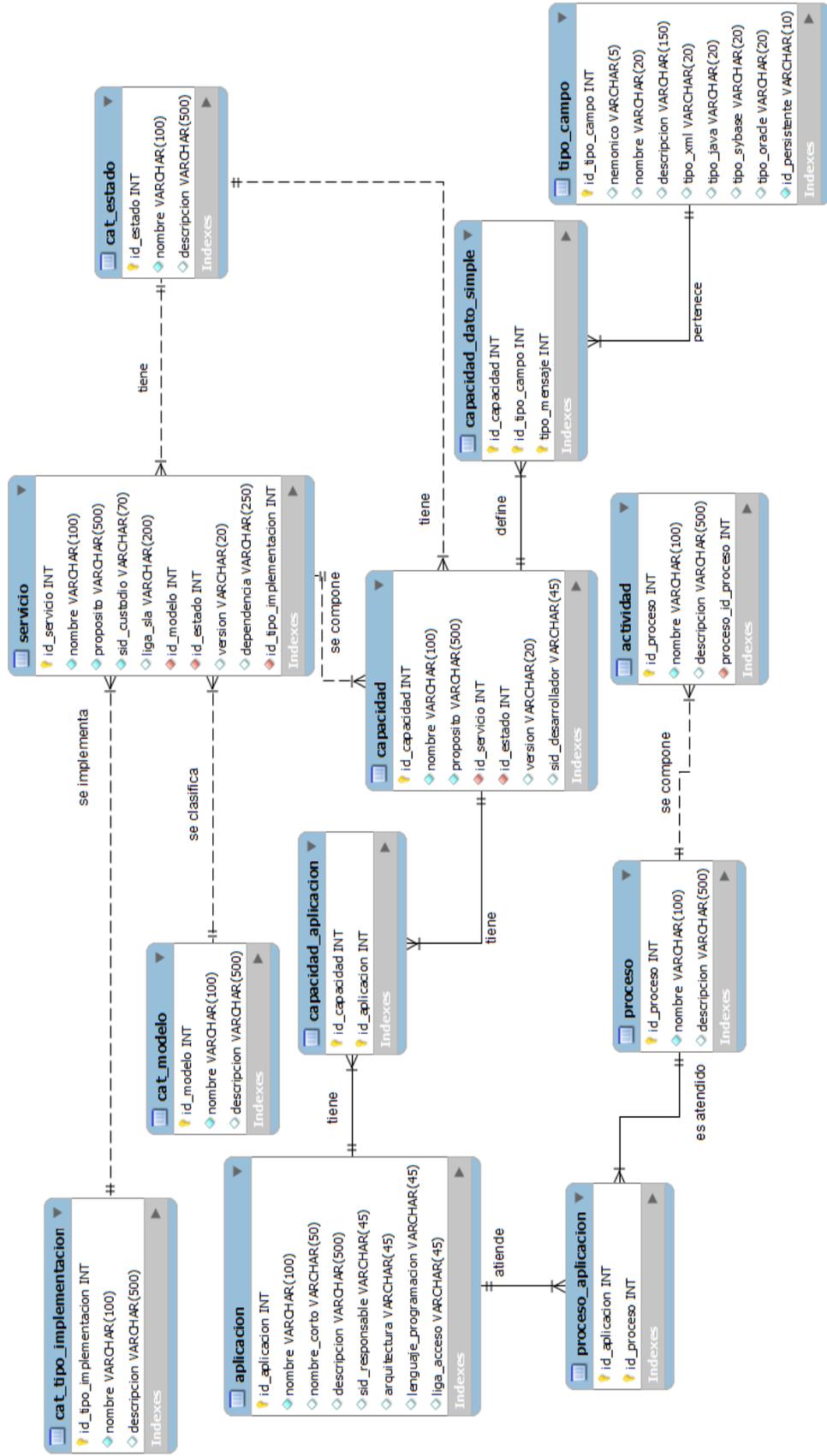


Diagrama bd 1

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Segunda parte

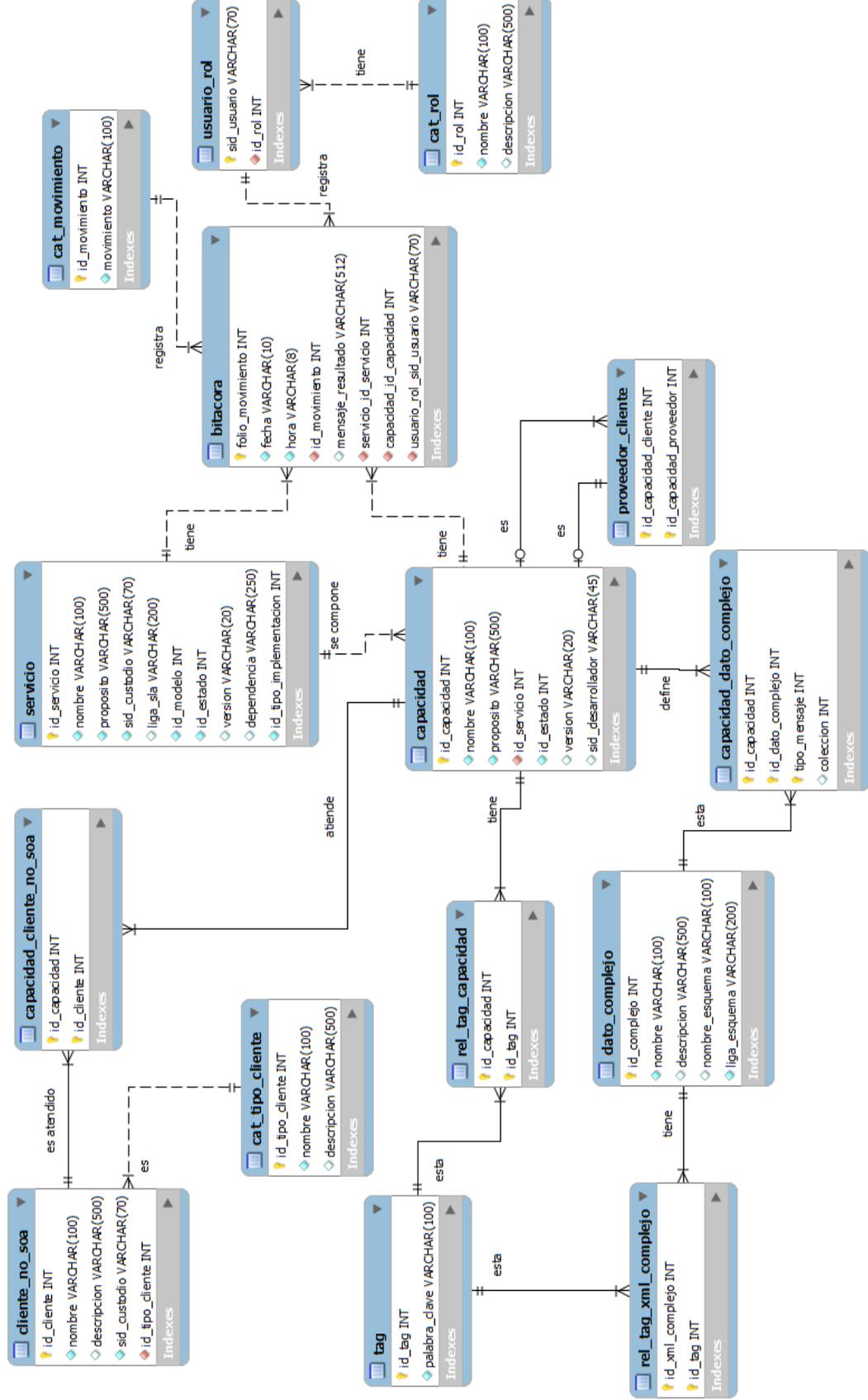


Diagrama bd 2.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Diccionario de datos

Se especifican las tablas más relevantes del sistema.

Nombre Tabla	Descripción de la Tabla	Nombre Campo	Descripción del campo	Tipo Dato Sybase	Tabla Referencia
bitácora	Tabla de tipo datos, que contiene todos los eventos que se registran al crear servicios, capacidades, asignar esquemas, clientes o cuando existe un error en alguna operación del ISSOA	folio_movimiento	Foliador de la bitácora	INT	
		sid_usuario	Sid usuario que provoca el movimiento para registrarse en bitácora	VARCHAR (70)	
		fecha	Fecha en que se registra en bitácora un movimiento	VARCHAR (10)	
		hora	Hora en la que se registra en bitácora un movimiento	VARCHAR (8)	
		id_servicio	Identificador único del servicio que provoco un movimiento que se registra en Bitácora	INT	
		id_capacidad	Identificador único de la capacidad que provoco un movimiento que se registra en bitácora	INT	
		id_movimiento	Identificador del movimiento que se registra	INT	<a href="#">cat_movimiento</a>
		mensaje_movimiento	Detalle del resultado del movimiento registrado en bitácora.	VARCHAR (512)	<a href="#">nto</a>

Definición e implementación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Nombre Tabla	Descripción de la Tabla	Nombre Campo	Descripción del campo	Tipo Dato Sybase	Tabla Referencia
proceso	Tabla de tipo entidad, que contiene todos los procesos que se llevan a cabo dentro de la DGEF.	id_proceso	Identificador único que el sistema asigna a un proceso de la DGEF	INT IDENTITY	
		nombre	Nombre de un proceso	VARCHAR (100)	
		descripcion	Descripción del proceso	VARCHAR (500)	
servicio	Tabla de tipo entidad, que contiene todos los servicios SOA que utilizan las aplicaciones de la DGEF o que están en etapa de análisis o diseño para futuras aplicaciones o mejoras.	id_servicio	Identificador único que el sistema asigna a un servicio utilizado en la DGEF.	INT IDENTITY	
		nombre	Nombre de un Servicio registrado en el ISSOA	VARCHAR (100)	
		propósito	Propósito del Servicio	VARCHAR (500)	
		sid_custodio	Sid del usuario custodio del Servicio	VARCHAR (70)	
		liga_sla	URL del "Portal wiki para arquitectura de servicios de la DGEF" donde se encuentra la documentación del acuerdo de nivel de servicio, contiene las Condiciones de comportamiento.	VARCHAR (200)	

Definición e implementación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Nombre Tabla	Descripción de la Tabla	Nombre Campo	Descripción del campo	Tipo Dato Sybase	Tabla Referencia
servicio	Tabla de tipo entidad, que contiene todos los servicios SOA que utilizan las aplicaciones de la DGEF o que están en etapa de análisis o diseño para futuras aplicaciones o mejoras.	id_modelo	Identificador del tipo de servicio al que pertenece el servicio.	INT	<a href="#">cat_modelo</a>
		id_estado	Identificador del estado en el que se encuentra el servicio.	INT	<a href="#">cat_estado</a>
		versión	Numero de adaptación del servicio	VARCHAR (20)	-
		dependencias	Indica los componentes, librerías, bibliotecas de los que el servicio es dependiente como son: Log4j, manejaArchivos.jar, saabm.jar, etc.	VARCHAR (500)	-
		id_tipo_implementacion	Identificador del método de implementación que se usó en el servicio	INT	<a href="#">cat_tipo_implementacion</a>
capacidad	Tabla de tipo entidad, que contiene todas las capacidades proporcionadas por los servicios del ISSOA.	id_capacidad	Identificador único que el sistema asigna a una capacidad que pertenece a un servicio de la DGEF	INT IDENTITY	
		nombre	Nombre de la capacidad	VARCHAR (100)	
		propósito	Propósito de la capacidad.	VARCHAR (500)	
		id_servicio	Identificador del servicio al que pertenece la capacidad.	INT	<a href="#">servicio</a>
		Id_estado	Identificador del estado en el que se encuentra la capacidad.	INT	<a href="#">cat_estado</a>

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Nombre Tabla	Descripción de la Tabla	Nombre Campo	Descripción del campo	Tipo Dato Sybase	Tabla Referencia
capacidad	Tabla de tipo entidad, que contiene todas las capacidades proporcionadas por los servicios del ISSOA.	version	Numero de Adaptación de la capacidad	VARCHAR (20)	
		sid_desarrollador	Sid del usuario que desarrolla la capacidad	VARCHAR (70)	
tipo_dato_c omplejo	Tabla de tipo entidad, que contiene los tipos de datos complejos o compuestos de un XML que pertenece a un esquema	id_tipo_dato_complejo	Identificador único que el sistema asigna a un tipo de dato complejo o compuesto	INT IDENTITY	
		nombre	Nombre del tipo de dato complejo	VARCHAR (100)	
		descripcion	Descripción de lo que almacena el tipo de dato complejo	VARCHAR (500)	
		nombre_esquema	Nombre del esquema que contiene el tipo de dato complejo.	VARCHAR (100)	
		liga_esquema	URL donde se encuentra el esquema que contiene el tipo de dato complejo	VARCHAR (200)	
		id_cliente	Identificador único que el sistema asigna a un cliente no SOA.	INT IDENTITY	
cliente_no_ soa	Tabla de que contiene los clientes que no están orientados bajo el enfoque de servicios.	nombre	Nombre del Cliente no SOA	VARCHAR (100)	
		descripcion	Descripción del Cliente no SOA	VARCHAR (500)	

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

Nombre Tabla	Descripción de la Tabla	Nombre Campo	Descripción del campo	Tipo Dato Sybase	Tabla Referencia
cliente_no_soa	Tabla de que contiene los clientes que no están orientados bajo el enfoque de servicios.	sid_custodio	Sid del usuario custodio del Cliente no SOA	VARCHAR (70)	
usuario_rol	Tabla de tipo entidad, que contiene los usuarios que están permitidos acceder al ISSOA, así como el rol que juegan dentro de este.	id_tipo_cliente	Identificador del tipo de cliente al que pertenece el Cliente no SOA.	INT	<a href="#">cat_tipo_cliente</a>
		sid_usuario	Sid del usuario a registrar en el sistema	VARCHAR (70)	
		id_rol	Identificador del rol que tendrá el usuario en el sistema.	INT	<a href="#">cat_rol</a>

Tabla 12: diccionario de datos

## ***Especificaciones de caso de uso***

### ***1. Especificación del caso de uso: administrar serviciosSOA***

#### **1.1 Descripción**

Este caso de uso describe las acciones a realizar para la creación, consulta y actualización de los serviciosSOA de la DISF.

#### **2. Actores Involucrados**

**Custodio del inventario:** Este tipo de usuario podrá realizar todas las operaciones del sistema, en el caso de los serviciosSOA, podrá consultarlos, modificarlos y cambiarlos de estado. **Custodio de servicio:** Puede consultar todos los serviciosSOA y está limitado a modificar sólo los serviciosSOA que están bajo su resguardo.

**Analista de negocio:** puede consultar los serviciosSOA.

**Analista de informática:** puede consultar los serviciosSOA.

#### **3. Precondiciones de Sistema.**

<< Autenticar usuario >>

#### **3. Flujo de Eventos**

##### **3.1 Flujo Básico**

El caso de uso inicia cuando el usuario quiere agregar o consultar algún servicioSOA.

1.- El sistema muestra las siguientes opciones:

- Agregar serviciosSOA [1]. (sólo en caso de que el usuario haya sido autenticado como custodio del inventario).
- Consultar Inventario [2].

### 3.2 Flujos Alternativos

#### 1. Agregar servicioSOA.

1.1. El sistema muestra un formulario solicitando los siguientes datos así como las opciones [1] Agregar nueva capacidad [2]crear servicio y [2]Cancelar:(E1)

- Nombre\*.
- Propósito\*.
- Custodio\*.
- Estado\*
- Liga SLA\*.
- Modelo
- Implementación.
- Versión
- Dependencias.

1.2. En el caso de los custodios el sistema mostrará una lista de los analistas de informática.

1.3. [1 Agregar nueva capacidad] El usuario selecciona la opción de agregar capacidad

1.3.1. El sistema muestra un formulario para ingresar la capacidad SOA asociada con el servicio que se está creando, solicita los siguientes datos y muestra las opciones de [1] crear y [2]cancelar:

- \*Nombre.
- \*Propósito.
- \*desarrollador.
- \*Estado.
- Versión.
- Agregar Clientes SOA.
- Agregar Clientes no SOA.

- Agregar Entrada.
- Agregar salida.
- Aplicaciones.
- Tags.

1.3.1.1. El usuario elije agregar clientes SOA

1.3.1.1.1. El sistema muestra una lista de las capacidades existentes y la opción de cancelar.

1.3.1.1.2. El usuario elije las capacidades deseadas.

1.3.1.1.3. El sistema redirige al flujo alternativo de agregar capacidad de este caso de uso.

1.3.1.1.4. El usuario elige la opción de cancelar.

1.3.1.1.4.1. El sistema redirige al flujo alternativo de agregar capacidad de este caso de uso.

1.3.1.2. Para agregar los tags el sistema muestra una lista de los existentes así como la opción de agregar uno nuevo.

1.3.1.3. El usuario elige agregar un nuevo tag.

1.3.1.3.1. El sistema muestra solicita la palabra a agregar y la opción de cancelar.

1.3.1.3.1.1. El usuario ingresa la palabra a agregar.

1.3.1.3.1.1.1. El sistema valida la palabra (E2).

1.3.1.3.1.2. El usuario elije la opción de cancelar.

1.3.1.3.1.2.1. El sistema regresa al flujo alternativo de agregar capacidad de este caso de uso.

1.3.1.4. El usuario elige la opción de agregar clientes no SOA.

1.3.1.4.1. El sistema muestra los clientes existentes y la opción de crear uno nuevo.

1.3.1.4.2. El usuario selecciona crear nuevo Cliente no SOA.

1.3.1.4.2.1. El sistema muestra un formulario solicitando los siguientes datos y muestra las opciones de crear y cancelar:

- \*Nombre.
- \*Descripción.
- Custodio.
- Tipo de cliente.

1.3.1.4.2.2. El usuario ingresa los datos y elige la opción de crear (E1).

1.3.1.4.2.2.1. El sistema valida los datos (E2).

1.3.1.4.2.3. El usuario elige la opción de cancelar.

1.3.1.4.2.3.1. El sistema regresa al flujo alternativo de agregar capacidad de este caso de uso.

1.3.1.5. El usuario selecciona la opción de agregar entrada.

1.3.1.5.1. El sistema muestra las opciones de agregar tipos de dato simple, tipos de dato complejos (esquemas) y cancelar.

1.3.1.5.2. El usuario elige la opción de tipo de dato simple.

1.3.1.5.2.1. El sistema solicita los siguientes datos del atributo de entrada y muestra las opciones de agregar y cancelar :

- \*Nombre.
- \*tipo.
- Es colección.

1.3.1.5.2.2. El usuario ingresa los datos y selecciona la opción de agregar (E1).

1.3.1.5.2.2.1. El sistema valida los datos (E2).

1.3.1.5.2.2.2. El sistema regresa al flujo alternativo de agregar capacidad.

1.3.1.5.2.3. El usuario selecciona la opción de cancelar.

1.3.1.5.2.3.1. El sistema regresa al flujo de agregar capacidad de este caso de uso.

1.3.1.5.3. El usuario selecciona la opción de agregar un tipo de dato complejo.

1.3.1.5.3.1. El sistema solicita los siguientes datos y muestra las opciones de agregar nuevo tipo de dato complejo y cancelar:

- Nombre\*.
- Tipo\*
- Es colección\*.

1.3.1.5.3.2. El usuario elige la opción de agregar nuevo tipo de dato complejo (E1).

1.3.1.5.3.2.1. <Administrar Esquemas>

1.3.1.5.3.3. El usuario elige la opción de cancelar.

1.3.1.5.3.3.1. El sistema regresa al flujo alternativo de agregar entrada.

1.3.1.5.4. El usuario selecciona la opción de cancelar.

1.3.1.5.4.1. El sistema regresa al flujo alternativo de agregar capacidad.

1.3.1.6. El usuario selecciona agregar salida.

1.3.1.6.1. El sistema lo redirige al flujo alternativo de agregar entrada de este caso de uso.

1.3.2. El usuario ingresa los datos requeridos y selecciona la opción de agregar capacidad.

1.3.2.1. El sistema valida los datos. (E2).

1.3.2.2. El sistema despliega un mensaje “Los datos han sido registrados correctamente” que confirma que la información fue almacenada.

1.3.3. El Usuario selecciona la opción de Cancelar.

1.3.3.1. El sistema regresa al flujo Alternativo de agregar servicio de este caso de uso.

1.4. [2 crear servicio] El Usuario selecciona la opción de crear Servicio (E1).

1.4.1. El sistema valida los datos (E2).

1.5. [3 cancelar] El Usuario selecciona la opción de cancelar.

1.5.1. El sistema lo redirige al flujo básico de este caso de uso.

2. Consultar servicioSOA.

2.1. El usuario selecciona la opción “inventario”.

2.2. El sistema muestra una lista de los serviciosSOA con los siguientes datos y la opción de seleccionar un servicioSOA para verlo detalladamente [1].

- Nombre
- Propósito
- Estado
- Nombre custodio

[1] El usuario selecciona un servicioSOA.

2.2.1.1. El sistema muestra los siguientes datos del servicioSOA seleccionado y las opciones de [1] actualizar metadatos (Sólo para custodios de inventario) [2] ver detalladamente una capacidad y [3] agregar capacidad (Sólo para custodios de inventario) y [4]guardar:

- Nombre.
- Propósito.
- Custodio.
- Estado
- Liga del SLA.
- Modelo.
- Implementación.
- Versión.
- Dependencias.
- Capacidades.

2.2.1.2. En el caso de las capacidades el sistema muestra una lista con los siguientes datos y la opción de ver una capacidad detalladamente:

- Nombre.
- Estado.
- Propósito.

2.2.1.3. El usuario actualiza los datos del servicio (sólo si está autenticado como custodio del inventario) y elige la opción de actualizar metadatos

2.2.1.3.1. El sistema valida los datos. (E2)

2.2.1.3.2. El sistema despliega un mensaje “Los datos han sido actualizados correctamente” que confirma que la

información fue almacenada y envía un correo electrónico con la información del servicioSOA modificada a los analistas de informática.

2.2.1.3.3. El sistema regresa al flujo alternativo de consultar servicio de este caso de uso.

2.2.1.4. [2] El usuario selecciona una capacidad y elige la opción de verla detalladamente.

2.2.1.4.1. El usuario es autenticado como custodio del inventario.

2.2.1.4.1.1. El sistema muestra las opciones de [1] actualizar clientes SOA, [2] Actualizar clientes no SOA, [3] Actualizar entradas [4] Actualizar salidas [5] Actualizar aplicaciones [6] Actualizar tags, [7]guardar, [8] cancelar y los siguientes datos:

- Nombre.
- Propósito.
- Estado.
- Versión
- Clientes SOA.
- Clientes no SOA.
- Entradas.
- Salidas.
- Aplicaciones.
- Tags.

2.2.1.4.1.2. [1]El usuario selecciona la opción de actualizar Clientes SOA.

2.2.1.4.1.2.1. El sistema muestra los clientes SOA existentes y los propios de la capacidad y la opción de aceptar.

- 2.2.1.4.1.2.2. El usuario selecciona los nuevos clientes o elimina los clientes para la capacidad seleccionada y elige la opción de aceptar.
- 2.2.1.4.1.2.3. El sistema regresa al flujo alternativo de ver detalladamente una capacidad.
- 2.2.1.4.1.3. [2] El usuario selecciona la opción de actualizar clientes no SOA.
  - 2.2.1.4.1.3.1. El sistema muestra todos los clientes no SOA distinguiendo los que pertenecen a la capacidad elegida y las opciones de agregar nuevo cliente no SOA y aceptar.
  - 2.2.1.4.1.3.2. El usuario modifica los clientes no SOA de la capacidad y selecciona la opción de aceptar.
    - 2.2.1.4.1.3.2.1. El sistema regresa al flujo alternativo de ver detalladamente una capacidad.
  - 2.2.1.4.1.3.3. El usuario selecciona la opción de agregar nuevo cliente no SOA.
    - 2.2.1.4.1.3.3.1. El sistema lo redirige al flujo alternativo de agregar clientes no SOA de este caso de uso.
- 2.2.1.4.1.4. [3] El usuario selecciona la opción de actualizar entradas.
  - 2.2.1.4.1.4.1. El sistema redirige al flujo alternativo de agregar entradas de este caso de uso.
- 2.2.1.4.1.5. [4] El usuario selecciona la opción de actualizar salida
  - 2.2.1.4.1.5.1. El sistema redirige al flujo alternativo de agregar entradas de este caso de uso.

2.2.1.4.1.6. [5] El usuario selecciona la opción de Actualizar aplicaciones.

2.2.1.4.1.6.1. El sistema muestra todas las aplicaciones existentes, distinguiendo las aplicaciones relacionadas con la capacidad elegida.

2.2.1.4.1.6.2. El usuario agrega o elimina aplicaciones de la capacidad elegida y selecciona la opción de aceptar.

2.2.1.4.1.6.3. El sistema regresa al flujo alternativo de ver detalladamente una capacidad.

2.2.1.4.1.7. [6] El usuario selecciona la opción de actualizar tags.

2.2.1.4.1.7.1. EL sistema muestra todos los tags distinguiendo los asignados para la capacidad elegida.

2.2.1.4.1.7.2. El usuario agrega o elimina tags de la capacidad elegida y selecciona la opción de aceptar.

2.2.1.4.1.7.3. El sistema regresa al flujo alternativo de ver detalladamente una capacidad.

2.2.1.4.1.8. [7] El usuario selecciona la opción de guardar.

2.2.1.4.1.8.1. El sistema valida los datos (E1) (E2).

2.2.1.4.1.9. [8] El usuario selecciona la opción de cancelar.

2.2.1.4.1.9.1. El sistema redirige al flujo básico de este caso de uso.

2.2.1.4.2. El usuario no es autenticado como custodio del inventario.

2.2.1.4.2.1. El sistema muestra los siguientes datos de la capacidad y la opción de aceptar:

- Nombre.
- Propósito.
- Estado.
- Versión
- Clientes SOA.
- Clientes no SOA.
- Entradas.
- Salidas.
- Aplicaciones.
- Tags.

2.2.1.4.2.2. El usuario consulta los datos necesarios y selecciona la opción de aceptar.

2.2.1.4.2.2.1. El sistema redirige al flujo básico de este caso de uso.

## 6. Poscondiciones

Sin poscondiciones.

## 7. Excepciones

E1. El sistema detecta que los campos señalados como obligatorios, marcados con un asterisco no han sido completados.

E1.1. El sistema muestra un mensaje de error señalando que los campos marcados como obligatorios no han sido completados.

E1.2. El sistema solicita que se ingresen los datos faltantes.

E2. El sistema detecta que los datos que se ingresan son incorrectos.

E2.1. El sistema muestra un mensaje de error señalando los datos que son incorrectos.

## 8. Requerimientos Especiales

Sin requerimientos especiales por el momento.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Diagramas de secuencia

### Agregar servicios

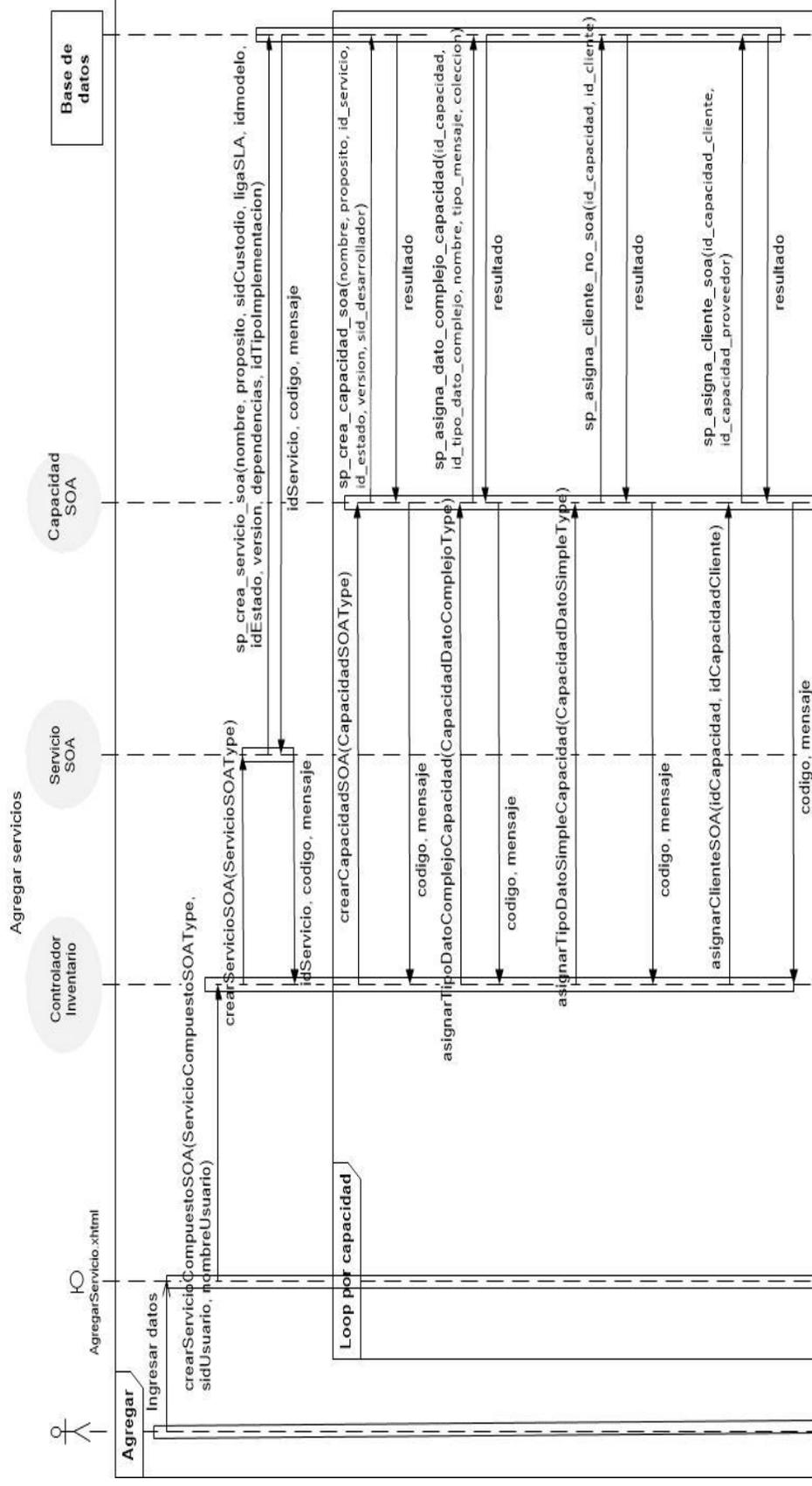
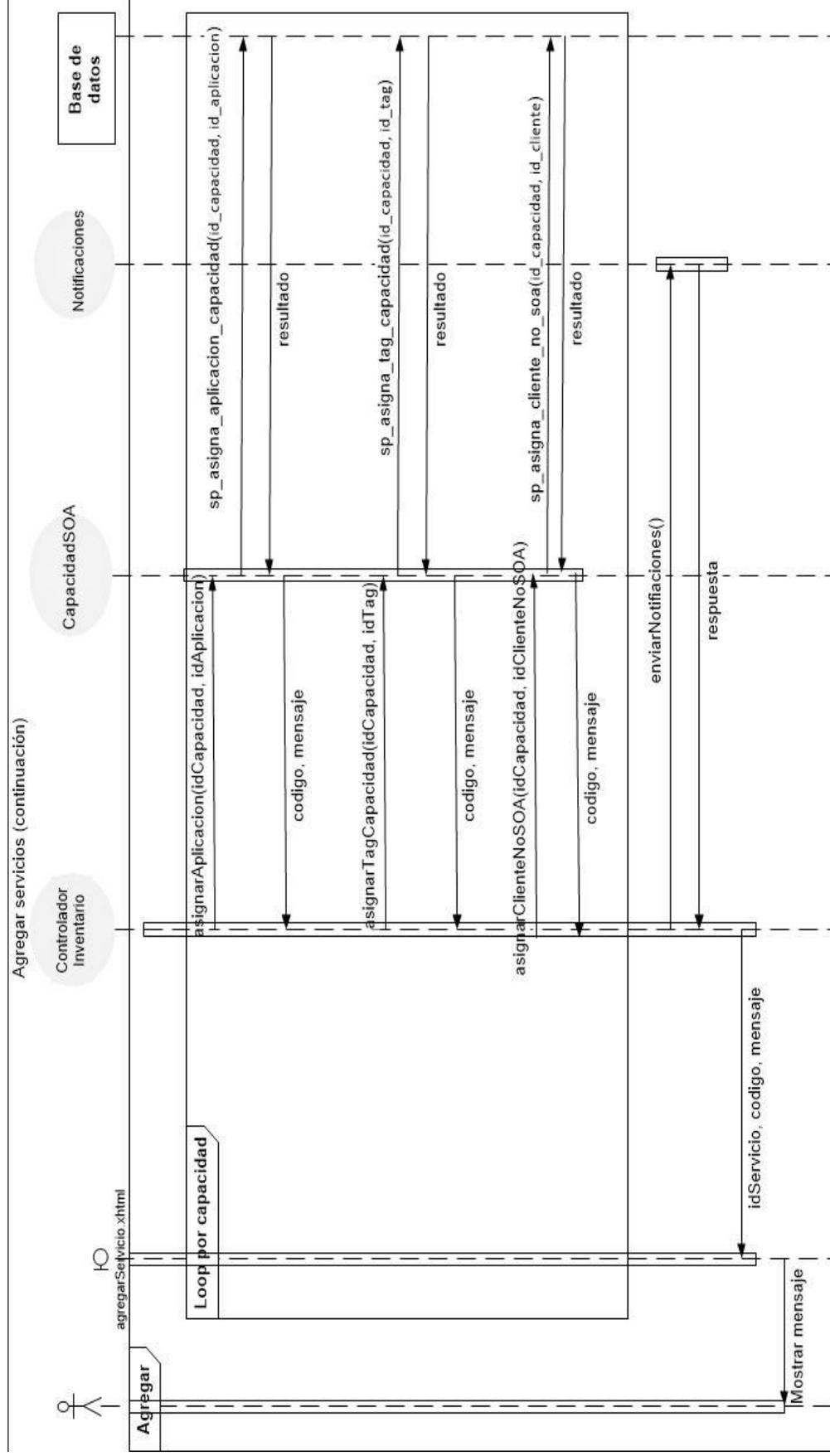


Diagrama de secuencia 1: Agregar servicios 1

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

**Agregar servicios (continuación).**



**Diagrama de secuencia 2 Agregar servicios 2**



Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Prototipo no funcional

A continuación se muestran las pantallas del prototipo no funcional del CU gestionar perfiles de servicios.

### Agregar Servicio

The screenshot shows a web application interface for adding a service. At the top, there are navigation links: 'Agregar Servicio', 'Inventario', 'Vista de negocio', 'Vista técnica', and 'Esquemas'. Below these are search and filter buttons: 'Buscar capacidad:', 'Buscar', 'Búsqueda Avanzada', and 'Listar Todas'. The main form is titled 'Nuevo servicio' and contains the following fields:

- Nombre:** A text input field.
- Propósito:** A large text area.
- Custodio:** A dropdown menu with 'Seleccionar'.
- Estado:** A dropdown menu with 'Seleccionar'.
- Liga del SLA:** A text input field.
- Modelo:** A dropdown menu with 'Seleccionar'.
- Implementación:** A dropdown menu with 'Seleccionar'.
- Versión:** A text input field.
- Dependencias:** A large text area.
- Capacidades:** A table with columns 'Propósito', 'Nombre', 'Estado', and 'Acción'. The table content shows 'No records found.'

At the bottom of the form are 'Cancelar' and 'Crear' buttons. A 'Nueva' button is located at the top right of the form area.

Ilustración 23: pantalla agregar servicios

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Agregar capacidad de servicio.

**Nueva Capacidad**

Nombre:  **Debe ingresar el nombre de la capacidad**

Propósito:

Desarrollador:  500 caracteres restantes. **Seleccionar**

Estado:  **Seleccionar**

Version:

Cientes SOA: 

Nombre
No records found.

**Agregar/Quitar**

Cientes No SOA: 

Nombre
No records found.

**Agregar/Quitar**

Entradas: 

Nombre	Tipo	Acción
No records found.		

**Agregar** **Debe agregar por lo menos una entrada**

Salidas: 

Nombre	Tipo	Acción
No records found.		

**Agregar**

Aplicaciones: **Debe agregar por lo menos una aplicación** **Agregar/Quitar**

Tags: **Debe agregar por lo menos un tag** **Agregar/Quitar**

**Crear**

Ilustración 24: pantalla agregar capacidad

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

**Actualizar capacidad de servicio:**

**Modificar Capacidad**

**Nombre:**  **Agregar/Quitar**

**Propósito:**  **Agregar/Quitar**

**Desarrollador:**  **Agregar/Quitar**

**Estado:**  **Agregar/Quitar**

**Versión:**  **Agregar/Quitar**

**Clientes SOA:**

Nombre	Tipo	Acción
No records found.		

**Agregar/Quitar**

**Clientes No SOA:**

Nombre	Tipo	Acción
No records found.		

**Agregar/Quitar**

**Entradas:**

Nombre	Tipo	Acción
IdEnvio	entero	Eliminar
IdSeccion	entero	Eliminar
version	entero	Eliminar

**Agregar**

**Salidas:**

Nombre	Tipo	Acción
codigoResultado	entero	Eliminar
mensajeResultado	cadena	Eliminar

**Agregar**

**Aplicaciones:**

Nombre	Nombre Corto
Validador de información SOA	ValidadorSOA

**Agregar/Quitar**

**Tags:**

Nombre
bitacora acopio
borra evento

**Agregar/Quitar**

**Actualizar** **Cancelar**

Ilustración 25: pantalla actualizar capacidad

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Actualizar Servicio:

[Agregar Servicio](#)
[Inventario](#)
[Vista de negocio](#)
[Vista técnica](#)
[Esquemas](#)

[Buscar](#)
[Búsqueda Avanzada](#)
[Listar Todas](#)

---

**Modificar servicio**

**BitacoraAcopio**

Nombre:

Propósito:

Custodio:

Estado:

Liga del SLA:

Modelo:

Implementación:

Versión:

Dependencias:

Capacidades:

Nombre	Estado	Propósito
borrarEventoBitacoraAcopio		Borrar el registro de bitácora que se genera durante la ejecución de la capacidad registrarEventoBitacoraAcopio ya que solo es un registro temporal. Por ahora solo se guarda mientras dura la atención del evento de acopio en cuestión. Después es borrado para indicar que ya fue atendido.
registrarEventoBitacoraAcopio		Registra en la bitácora de Acopio los detalles del envío realizado desde el sistema EspifmeI. Los atributos que componen al registro de bitácora son los siguientes: id de la Sección, id Envío, versión, llave Archivo, fecha Datos, institución, nombre del archivo, tipo archivo, y tipo transmisión.

Ilustración 26: pantalla actualizar servicio

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

**Mostrar Servicios:**

SERVICIOS				
Nombre	Propósito	Estado	Estado	Custodio
AdministradorCatalogo	Controlar los pasos necesarios para la administración de un catálogo, el cual incluye: 1. La creación y modificación de categorías, 2. La creación y modificación de catálogos, 3. La creación y modificación de post procesos, 4. La administración de usuarios y 5. Las asociaciones entre categorías, catálogos, post procesos, tablas y usuarios.	Producción	Todos	Adrián Ivé Martínez Araiza
AyudameSequenciador	Realizar operaciones necesarias para la ejecución de la secuencia de nodos pre configurada en la base de datos. Dichas operaciones no son parte esencial de la lógica del negocio, sin embargo, son fundamentales para el correcto funcionamiento de la secuencia de pasos a seguir para poder realizar el acopio de la información.	Implementación		Alan Baez Enriquez
BitacoraAcopio	Administrar todo lo referente a la bitácora del Servicio de Acopio. Específicamente se refiere a las actividades de registrar, consultar, borrar y actualizar registros de la bitácora de acopio.	Producción		José Armando Flores Díaz
BitacoraCatalogo	Almacenar y proporcionar información sobre la ejecución de movimientos (inserciones, actualizaciones, eliminaciones, respaldos, sustituciones completa o agregado de grupo de registros) en el AdeC SOA.	Producción		Adrián Ivé Martínez Araiza
BitacoraIntegración	Registrar en la bitácora del integrador, los detalles de los eventos de integración que atendió.	Producción		José Armando Flores Díaz
BitacoraPostProceso	Registrar y consultar los PostProcesos ejecutados en el AdeC, almacenando información útil para el diagnóstico. Los PostProcesos ejecutados corresponden a realizar una Réplica de Catálogo a Base de Datos, Réplica parcial de Catálogo a Base de Datos, Réplica de Catálogo a un Archivo, Ejecución de un Stored Procedure sobre un Catálogo, Ejecución de una Capacidad de un Servicio sobre un Catálogo, Generación de catálogos para el EspinNet.	Producción		Adrián Ivé Martínez Araiza

**Ilustración 27 : pantalla mostrar servicios**

***Perfiles de servicios y capacidades:***

En esta fase se desarrollaron los siguientes perfiles de servicios:

- Servicio SOA
- Capacidad SOA
- Controlador Inventario.

De igual manera se realizaron los siguientes perfiles de capacidades:

- actualizarCapacidadSOA
- asignarAplicacion
- asignarClienteNoSOA
- asignarClienteSOA
- asignarTagCapacidad
- asignarTipoDatoComplejoCapacidad
- asignarTipoDatoSimpleCapacidad
- obtenerCapacidadesxServicio
- crearCapacidadSOA
- obtenerCapacidadesSOA
- actualizarServicioSOA
- crearServicioSOA
- obtenerServicioSOAxId
- obtenerServiciosISSOA
- crearCapacidadCompuestaSOA
- crearServicioCompuestoSOA
- obtenerCapacidadCompuestaId
- obtenerCapacidadxId
- obtenerServiciosISSOA

A fin de ejemplificar los perfiles de servicios y capacidades se seleccionaron los más complejos que a continuación se muestran.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Perfil del Servicio servicioSOA</b>	
Nombre del Servicio	<b>servicioSOA</b>
Propósito	Proporciona operaciones de creación, consulta, actualización, y eliminación sobre los servicios SOA de la DISF.
Modelo del servicio	Entidad
Capacidades.	<ul style="list-style-type: none"> <li>- crearServicioSOA</li> <li>- actualizarServicioSOA</li> <li>- obtenerServicioSOAxId</li> <li>- obtenerServiciosSOAxModelo</li> <li>- obtenerServiciosISSOA</li> <li>- obtenerServiciosSOApopulares</li> <li>- obtenerServiciosSOAxAplicacion</li> <li>- obtenerClientesServiciosPopulares</li> </ul>
Palabras clave.	Servicios web, SOA, servicios DISF,
Versión.	V 1.0
Custodio.	Jorge de Jesús Sánchez Aguilar
Tipo de Implementación.	Servicio web.
Dependencias.	Servicio web.

**Tabla 13: perfil de servicio SOA**

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Perfil de la Capacidad crearServicioSOA</b>									
Nombre de la Capacidad*	crearServicioSOA								
Propósito *	Registra los datos de un servicio (perfil del servicio) en la base de datos del sistema ISSOA. Devuelve un número negativo si ocurrió un error grave al momento de la ejecución, una excepción si se trata de un servicio ya existente o un número positivo cuando se realizó correctamente la operación.								
Entradas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Servicio</td> <td>ServicioSOAType</td> </tr> </tbody> </table>	Nombre	Tipo	Servicio	ServicioSOAType				
Nombre	Tipo								
Servicio	ServicioSOAType								
Salidas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>IdServicio</td> <td>Entero</td> </tr> <tr> <td>codigoResultado</td> <td>Entero</td> </tr> <tr> <td>mensajeResultado</td> <td>Cadena</td> </tr> </tbody> </table>	Nombre	Tipo	IdServicio	Entero	codigoResultado	Entero	mensajeResultado	Cadena
Nombre	Tipo								
IdServicio	Entero								
codigoResultado	Entero								
mensajeResultado	Cadena								
Nombre del Desarrollador*	<i>Jorge de Jesús Sánchez Aguilar</i>								
Clientes SOA	<i>CrearServicioCompuesto del servicio ControladorInventario</i>								
Clientes no SOA	<i>Ninguno</i>								
Tags*	Servicio SOA, crear servicio, inventario.								
Versión*	V 1.0								
Aplicación*	ISSOA								

Tabla 14: Perfil de la capacidad crearServicioSOA

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Perfil de la Capacidad actualizarServicioSOA</b>							
Nombre de la Capacidad*	actualizarServicioSOA						
Propósito *	Actualizar los datos de un servicio (perfil del servicio) en la base de datos del sistema ISSOA y devolver como respuesta si se pudo realizar la actualización o no. Un servicio se define como: una unidad de solución lógica a la que se le ha aplicado la orientación a servicios en un grado significativo.						
Entradas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Servicio</td> <td>ServicioSOAType</td> </tr> </tbody> </table>	Nombre	Tipo	Servicio	ServicioSOAType		
Nombre	Tipo						
Servicio	ServicioSOAType						
Salidas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>codigoResultado</td> <td>Entero</td> </tr> <tr> <td>mensajeResultado</td> <td>cadena</td> </tr> </tbody> </table>	Nombre	Tipo	codigoResultado	Entero	mensajeResultado	cadena
Nombre	Tipo						
codigoResultado	Entero						
mensajeResultado	cadena						
Nombre del Desarrollador*	Jorge de Jesús Sánchez Aguilar						
Clientes SOA	<b>ActualizarServicioSOA de ControladorInventario.</b>						
Clientes no SOA	<b>ninguno</b>						
Tags*	Actualizar servicio, ServicioSOA, inventario						
Versión*	V 1.0						
Aplicación*	ISSOA						

Tabla 15: Perfil capacidad actualizarServicioSOA

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Esquemas utilizados

<b>Nombre TDC:</b>	<b>ServicioSOAType</b>
<b>Descripción:</b>	Tipo de dato que describe un servicio que cumple con los estándares de SOA
<b>Nombre del esquema:</b>	ServicioSOA.xsd
<b>Tags:</b>	servicioSOA, servicio.
<b>Atributos:</b>	
<b>Nombre</b>	<b>Tipo</b>
<b>idServicio</b>	entero
<b>nombre</b>	Cadena
<b>proposito</b>	Cadena
<b>sidCustodio</b>	Cadena
<b>ligaSLA</b>	Cadena
<b>idEstado</b>	Entero
<b>idModelo</b>	Cadena
<b>versión</b>	Cadena
<b>dependencias</b>	Cadena
<b>idTipoImplementacion</b>	Entero

Tabla 16: Esquema ServicioSOA

## ***Plan de pruebas***

### *Propósito del documento*

El propósito del plan de pruebas es reunir la información necesaria para planificar, controlar y ejecutar las pruebas del sistema Inventario de Servicios SOA (ISSOA).

El plan de pruebas para dicho sistema tiene los siguientes objetivos:

- Identificar los elementos que deben ser objeto de las pruebas.
- Describir los métodos de prueba que se utilizarán.
- Listar los entregables de las pruebas.

### *Alcance*

El presente plan muestra las pruebas aplicadas al sistema Inventario de Servicios SOA, estas son realizadas una vez que se termina su desarrollo y son documentadas respectivamente. Las pruebas que serán abordadas son las siguientes:

- Pruebas de función.
- Pruebas de uso.
- Pruebas de integridad de base de datos.
- Pruebas de seguridad.
- Pruebas de control de acceso.
- Pruebas de integración.

### *Audiencia a la que va dirigido*

El documento va dirigido a las siguientes personas:

- Equipo Inventario de Servicios SOA: requieren información de retroalimentación a la planeación y ejecución de las pruebas.
- Probadores: se encargarán de la ejecución de las pruebas.

### *Propósito de las Pruebas*

De acuerdo a los requerimientos del sistema, el propósito de las pruebas es verificar que se cumplan dichos lineamientos establecidos en el análisis del sistema.

Cumplir con la documentación necesaria que marca la normatividad de TI.

### *Propósito de la evaluación*

A continuación se listan una serie de objetivos de la evaluación de pruebas:

- Encontrar la mayor cantidad de fallas como sea posible.
- Verificar que todos los requerimientos se cumplan.

### *Aspectos en los que se enfocan las pruebas*

Las pruebas serán enfocadas en los elementos que constituyen el software creado y estarán basadas en los requerimientos del sistema.

### *Delimitación de las pruebas incluidas*

A continuación se muestra una lista de las pruebas a realizar:

- Pruebas de función: se hacen con la finalidad de verificar que el sistema cumpla con los requerimientos definidos.
- Pruebas de usabilidad: Las pruebas de usabilidad se realizaron con el fin de identificar posibles confucioses que se le pueden presentar al usuario, durante la manipulación de la herramienta.
- Pruebas de seguridad: Las pruebas de seguridad son llevadas a cabo por la DGTI mediante un proceso de escaneo de la aplicación. El resultado de dicho proceso es un reporte que contiene todas las vulnerabilidades detectadas así como recomendaciones de solución.

### *Delimitación de las pruebas no incluidas*

A continuación se muestra una lista de las pruebas no incluidas:

- Perfil de rendimiento: Debido a que no se tienen requerimientos de rendimiento, estas pruebas no fueron contempladas.
- Impacto a la infraestructura: Debido a la naturaleza del sistema las pruebas de volumen y de concurrencia no se requirieron.
- La viabilidad técnica con prototipo no aplica debido a que no se utilizará tecnología nueva.

*Tipos y técnicas de pruebas*

Pruebas de Función

Pruebas orientadas a las funcionalidades requeridas del sistema:

<i>Objetivo de la técnica</i>	Comprobar que el sistema cumpla con los requerimientos recabados.
<i>Técnica:</i>	<p><i>Checklist:</i> Se crea una lista de verificación a partir de la última versión de la lista de requerimientos.</p> <p>El registro de los incidentes se lleva a cabo en un archivo Excel, que está constituido de la siguiente manera:</p> <p>Número de requerimiento, nombre, su correspondiente evaluación y observaciones si es necesario.</p> <p>Para evaluar la prueba del requerimiento colocar “Si” en la casilla de “Cumplió” y “No” en caso de que no se cumpla el requerimiento.</p> <p>Los resultados de las pruebas se indican en el documento de Reporte de pruebas.</p>
<i>Herramientas requeridas:</i>	<p>Documentos de apoyo:</p> <p>Matriz de requerimientos.</p> <p>Lista de verificación de requerimientos.</p> <p>Datos de prueba en la base de datos.</p> <p>Primera versión a revisar en el ambiente de pruebas.</p>
<i>Criterios de Éxito:</i>	Si el 100% del listado de requerimientos fueron valorados positivamente, entonces se puede decir que las pruebas de función fueron superadas.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<i>Consideraciones Especiales:</i>	Ninguna.
------------------------------------	----------

**Tabla 17: pruebas de funcion**

### Pruebas de Usabilidad

Pruebas orientadas a la medición de la facilidad de uso que el usuario final tiene sobre el sistema:

<i>Objetivo de la técnica</i>	Verificar la facilidad de uso, la interacción del usuario con el software y la conformidad del usuario.
<i>Técnica:</i>	<p>Mediante un reporte de pruebas de usabilidad indicar incidencias de acuerdo a la versión de codificación. Verificar que los diversos componentes de la interfaz sean adecuados.</p> <p>El registro de las observaciones se realiza en un archivo Excel y tiene la siguiente estructura:</p> <p>Número de acción, Nombre de flujo, Descripción, Observaciones y pantalla (imagen) del incidente.</p> <p>Al final los resultados son mostrados en el documento: “reporte de pruebas”, en la sección que corresponda a este tipo de prueba.</p>
<i>Herramientas requeridas:</i>	<p>Documentos de apoyo:</p> <p>Listado de requerimientos.</p> <p>Lista de flujos.</p> <p>Primera versión a revisar en el ambiente de pruebas.</p> <p>Datos de prueba en la base de datos.</p>

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<i>Criterios de Éxito:</i>	El usuario final es capaz de intuir las funciones y uso del sistema gracias a su interfaz.
<i>Consideraciones Especiales:</i>	De acuerdo a los permisos otorgados a cada usuario se podrá hacer uso del sistema.

Tabla 18: pruebas de usabilidad

### Pruebas de Seguridad

Pruebas orientadas a detectar si la aplicación tiene vulnerabilidades de seguridad y si ésta cuenta con las medidas necesarias para su correcto funcionamiento.

<i>Objetivo de la técnica</i>	Detectar posibles vulnerabilidades en la aplicación.
<i>Técnica:</i>	<p>Mediante un análisis de vulnerabilidades con la herramienta IBM Rational AppScan 8.5.0.1 se pretende detectar incidencias relacionadas con la seguridad de la aplicación.</p> <p>El reporte generado muestra la aplicación escaneada y las recomendaciones pertinentes en caso de encontrar problemas en la aplicación. Cada problema es categorizado en bajo medio y alto según la severidad del problema.</p> <p>Al final los resultados son mostrados en el documento: “reporte de pruebas”, en la sección que corresponda a este tipo de prueba.</p> <p>Todos los archivos generados en estas pruebas se encontrarán anexos en el reporte de pruebas.</p>

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<i>Herramientas requeridas:</i>	Documentos y herramientas de apoyo:  Reporte de Pruebas.  Dirección URL de la aplicación y servicios.  Herramienta: IBM Rational AppScan 8.5.0.
<i>Criterios de Éxito:</i>	Que el reporte generado tenga el menor número posible de incidencias y la severidad de los problemas encontrados sea baja.
<i>Consideraciones Especiales:</i>	Ninguna.

Tabla 19: pruebas de seguridad

#### *Criterios de entrada y salida*

##### Criterios de entrada

Los servicios 100% implementados y funcionando en el esquema de base de datos y servidor de aplicaciones de prueba.

##### Criterios de salida

Una vez realizado todo el conjunto de pruebas satisfactoriamente, el documento de “reporte de pruebas” estará completo y se podrá realizar una evaluación cuantitativa de los resultados, si dicho resultado es de 95% o más sobre un total de 100%, entonces el sistema cumple con los objetivos de las pruebas.

### Criterios de una terminación anormal del plan

Si se descubre alguna falla que no permita la continuidad de las pruebas, serán suspendidas hasta que sea atendida la falla.

Si alguna prueba no cumple con el propósito para el cual fue creada, podrá ser eliminada, previa evaluación y, siempre y cuando, se justifique en el reporte de pruebas.

### *Entregables*

#### Reporte de pruebas

Al final del periodo de pruebas se entregará un reporte de las mismas, que abarcará un resumen de cada tipo de prueba realizado, así como su resultado y un global con los incidentes que se resolvieron y los que no.

### *Necesidades de ambiente*

#### Hardware

Recurso	Cantidad	Nombre y tipo
Servidor de base de datos	1	
—subred	1	
—Nombre del servidor		JDESAINT
—Nombre de base de datos		admon_issao_net
Servidor de aplicaciones	1	Glassfish 3.1.2
—Subred		Puerto 8040
—Nombre del servidor		Servap pruebas2
Pc's de los desarrolladores para pruebas	1	Equipo propio

**Tabla 20: necesidades de hardware**

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Software

<b>Nombre del elemento de software</b>	<b>Versión</b>	<b>Tipo y otros detalles.</b>
Windows 7	7	Sistema operativo./máquina de pruebas
Internet Explorer	9	Navegador de internet.
Java VM	7	Máquina virtual de java
Glassfish	3.1.2	Servidor de aplicaciones
IBM Rational AppScan	8.5.0.1	Herramienta que mide la seguridad de la aplicación
Soapui	3.5.1.	Herramienta que simula clientes de servicios web

Tabla 21: necesidades de software

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

*Responsabilidades, personal y necesidades de entrenamiento*

Roles y personal

<b>Rol</b>	<b>Recursos mínimos recomendados (número de recursos asignados)</b>	<b>Responsabilidades específicas o comentarios</b>
Probador	1	<ul style="list-style-type: none"> <li>• Ejecutar las pruebas de acuerdo al plan.</li> <li>• Reportar los resultados obtenidos.</li> <li>• Generar los entregables correspondientes.</li> </ul>
Líder de proyecto	1	<ul style="list-style-type: none"> <li>• Monitorea la ejecución de las pruebas.</li> <li>• Valida el plan de pruebas y el reporte de pruebas.</li> <li>• Recaba los comentarios.</li> <li>• Retroalimenta las observaciones y coordina las adecuaciones de las pruebas.</li> </ul>

**Tabla 22: roles y personal para las pruebas**

### *Riesgos*

- a) Las pruebas pueden ser interrumpidas debido al mantenimiento del servidor de pruebas. El servidor no está a cargo del equipo de trabajo.
- b) Las pruebas también pueden ser retrasadas debido al mantenimiento de red.

### *Reportes de problemas y resolución de conflictos*

Cuando exista algún problema al momento de ejecutar las pruebas, será discutido por el probador y el líder de proyecto, y se llegará a un acuerdo de suspender o no la ejecución de las pruebas. En caso de que sean suspendidas, se pondrá una justificación en el reporte de pruebas, dicha justificación deberá ir subrayada, también se mencionará una fecha estimada para la reanudación de las pruebas, o bien, algún evento que al cumplirse generará la reanudación de las pruebas.

### **Contratos de Servicios**

En esta fase se desarrollaron los contratos de los servicios:

- ServicioSOA.
- CapacidadSOA.
- ControladorInventario.

#### *ServicioSOA*

A fin de ejemplificar se muestra el contrato del servicio ServicioSOA.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="ServicioSOA"
    targetNamespace="http://banxico.org.mx/contrato/dgef/es/ServicioSOA/wsdl/"
    xmlns=http://schemas.xmlsoap.org/wsdl/
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://banxico.org.mx/contrato/dgef/es/ServicioSOA/wsdl/"
    xmlns:sss="http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/v1/"
    xmlns:com=http://banxico.org.mx/schema/dgef/us/Comun/v1/
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <documentation>
    Este es el servicio de entidad ServicioSOA, encargado de realizar diversas
    tareas relacionadas con El servicio ServicioSOA como crear, actualizar, y
    obtener la entidad Servicio de acuerdo con diversos parámetros.

    Autor Sánchez Aguilar Jorge de Jesús.

    Version 1.0

    GSISF
  </documentation>
```

```
<types>

  <xsd:schema targetNamespace=
    "http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/">

    <xsd:import

      namespace                                =
      http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/v1/
      schemaLocation="../schemas/ServicioServicioSOA.xsd" />

    <xsd:import                                namespace                                =
      "http://banxico.org.mx/schema/dgef/us/Comun/v1/"
      schemaLocation="../schemas/comun/Comun.xsd" />

  </xsd:schema>

</types>

<message name="ExcepcionNegocioMsg">

  <part          name="ResponseParameter"      element      =
    "com:ExcepcionNegocioType">

  </part>

</message>

<message name="ActualizarServicioSOAEntradaMsg">

  <part          name="RequestParameter"      element      =
    "sss:ActualizarServicioSOAEntradaType" />

</message>

<message name="ActualizarServicioSOASalidaMsg">

  <part          name="ResponseParameter"      element      =
    "sss:ActualizarServicioSOASalidaType" />

</message>
```

```
<message name="CrearServicioSOAEntradaMsg">
    <part name="RequestParameter" element="
        "sss:CrearServicioSOAEntradaType" />
</message>

<message name="CrearServicioSOASalidaMsg">
    <part name="ResponseParameter" element="
        "sss:CrearServicioSOASalidaType" />
</message>

<message name="ObtenerServicioSOAxIdEntradaMsg">
    <part name="RequestParameter" element="
        "sss:ObtenerServicioSOAxIdEntradaType" />
</message>

<message name="ObtenerServicioSOAxIdSalidaMsg">
    <part name="ResponseParameter" element="
        "sss:ObtenerServicioSOAxIdSalidaType" />
</message>

<message name="ObtenerServiciosISSOAEntradaMsg">
    <part name="RequestParameter" element="
        "sss:ObtenerServiciosISSOAEntradaType" />
</message>

<message name="ObtenerServiciosISSOASalidaMsg">
    <part name="ResponseParameter" element="
        "sss:ObtenerServiciosISSOASalidaType" />
</message>
```

```
<portType name="ServicioSOAInterface">
  <operation name="ActualizarServicioSOA">
    <input message="tns:ActualizarServicioSOAEntradaMsg" />
    <output message="tns:ActualizarServicioSOASalidaMsg" />
    <fault name="businessException" message="tns:ExcepcionNegocioMsg" />
  </operation>
  <operation name="CrearServicioSOA">
    <input message="tns:CrearServicioSOAEntradaMsg" />
    <output message="tns:CrearServicioSOASalidaMsg" />
    <fault name="businessException" message="tns:ExcepcionNegocioMsg" />
  </operation>
  <operation name="ObtenerServicioSOAId">
    <input message="tns:ObtenerServicioSOAIdEntradaMsg" />
    <output message="tns:ObtenerServicioSOAIdSalidaMsg" />
    <fault name="businessException" message="tns:ExcepcionNegocioMsg" />
  </operation>
</portType>
```

```
<binding name="ServicioSOABinding" type="tns:ServicioSOAInterface">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="ActualizarServicioSOA">
    <soap:operation
      soapAction="http://banxico.org.mx/dgef/es/ServicioSOA/Actualizar
      ServicioSOA" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
    <fault name="businessException">
      <soap:fault name="businessException" use="literal" />
    </fault>
  </operation>
  <operation name="CrearServicioSOA">
    <soap:operation
      soapAction="http://banxico.org.mx/dgef/es/ServicioSOA/
      CrearServicioSOA" />
    <input>
      <soap:body use="literal" />
    </input>
```

```
<output>
    <soap:body use="literal" />
</output>
<fault name="businessException">
    <soap:fault name="businessException" use="literal" />
</fault>
</operation>
<operation name="ObtenerServicioSOAxId">
    <soap:operation
    soapAction="http://banxico.org.mx/dgef/es/ServicioSOA/
    ObtenerServicioSOAxId" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="businessException">
        <soap:fault name="businessException" use="literal" />
    </fault>
</operation>
```

```
<operation name="ObtenerServiciosISSOA">
    <soap:operation
        soapAction="http://banxico.org.mx/dgef/es/ServicioSOA/
        ObtenerServiciosISSOA" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="businessException">
        <soap:fault name="businessException" use="literal" />
    </fault>
</operation>
</binding>
<service name="ServicioSOAService">
    <port name="ServicioSOAPort" binding="tns:ServicioSOABinding">
        <soap:address location="http://banxico.org.mx/dgef/es/ServicioSOA"/>
    </port>
</service>
</definitions>
```

## ***Fase de Construcción***

Los siguientes son los entregables realizados en la fase de construcción.

### ***Especificaciones de casos de uso***

#### 1. Especificación del caso de uso: *Administrar Esquemas*

##### 1.1 Descripción

Este caso de uso describe las acciones a realizar para la creación, consulta y actualización de los esquemas SOA utilizados en entradas y salidas de servicios.

##### 2. Actores Involucrados

Custodio del inventario: Este tipo de usuario podrá realizar todas las operaciones del sistema, en el caso de los esquemas podrá agregarlos desde la entrada de las capacidades o desde la sección de esquemas.

Analista de informática: Puede consultar todos los esquemas pero no puede modificarlos.

Analista de negocio: Puede consultar los esquemas pero no modificarlos.

##### 3. Precondiciones de Sistema

<< Autenticar usuario >>

##### 3. Flujo de Eventos

###### 3.1 Flujo Básico

El caso de uso inicia cuando el usuario quiere agregar, consultar, modificar o eliminar esquemas

1.- El sistema muestra las siguientes opciones:

- Esquemas [1].
- Agregar un esquema [2].

###### 3.2 Flujos Alternativos

3. Esquemas.

3.1. El sistema muestra los siguientes datos de los esquemas existentes.

- Nombre del tipo de dato.
- Descripción.
- Nombre del esquema\*.
- Tags.

3.2. El sistema permite abrir el esquema al dar click en el nombre del tipo de dato.

3.2.1. El sistema muestra el esquema en un navegador de internet.

3.3. El usuario es autenticado cómo custodio del inventario.

3.3.1. El usuario elije un esquema.

3.3.1.1. El sistema muestra los siguientes datos del esquema a manera de edición y las opciones de guardar[1] y cancelar[2]:

- Nombre del tipo de dato.\*
- Descripción.\*
- Nombre del esquema.\*
- Liga del esquema.\*
- Tags.\*

3.3.1.2. El usuario realiza las modificaciones correspondientes y selecciona la opción de guardar.

3.3.1.2.1. El sistema valida los datos (E2).

3.3.1.2.2. El sistema muestra el mensaje: “los datos se han actualizado correctamente” y se redirige a la sección de esquemas.

3.3.1.3. El usuario elije la opción de cancelar.

3.3.1.3.1. El sistema redirige a la sección de esquemas.

### 3.4. El usuario no es autenticado como custodio del inventario.

#### 3.4.1. El Sistema muestra los siguientes datos del esquema y la opción de aceptar.

- Nombre del tipo de dato.\*
- Descripción.\*
- Nombre del esquema.\*
- Liga del esquema.\*
- Tags.

#### 3.4.2. El usuario elige la opción de aceptar.

##### 3.4.2.1. El sistema redirige al flujo alternativo de ver esquemas de este caso de uso.

### 4. El usuario elige la opción de agregar nuevo esquema (esta opción también está disponible desde el caso de uso administrar servicios al momento de agregar las entradas y salidas de las capacidades).

#### 4.1. El sistema muestra un formulario para ingresar los esquemas, solicitando los siguientes datos y mostrando las opciones de agregar esquema, cancelar.

- \*Nombre del tipo de dato.
- \*Descripción.
- Nombre del esquema.
- Liga del esquema.
- Tags.

##### 4.1.1. [1] El usuario selecciona la opción de agregar esquema.

###### 4.1.1.1. El sistema valida los datos (E1).

###### 4.1.1.2. El sistema muestra el mensaje: “el esquema se ha agregado correctamente” y redirige a la sección de esquemas.

##### 4.1.2. [2] El usuario ingresa elige la opción de cancelar

###### 4.1.2.1. El sistema redirige a la sección de esquemas.

## 6. Poscondiciones

Sin poscondiciones.

## 7. Excepciones

E1. El sistema detecta que los campos señalados como obligatorios, marcados con un asterisco no han sido completados.

E1.1. El sistema muestra un mensaje de error señalando que los campos marcados como obligatorios no han sido completados.

E1.2. El sistema solicita que se ingresen los datos faltantes

E2. El sistema detecta que los datos que se ingresan son incorrectos.

E2.1. El sistema muestra un mensaje de error señalando los datos que son incorrectos.

## 8. Requerimientos Especiales

Sin requerimientos especiales por el momento.

## 1. Especificación del caso de uso: Gestionar Usuarios

### 1.1 Descripción

Este caso de uso describe las acciones a realizar para la administración de los usuarios, donde el administrador del sistema podrá dar de alta usuarios, eliminarlos y asignar roles.

### 2. Actores Involucrados

Administrador del sistema: se encargará de dar de alta a los usuarios en el sistema, así como de asignarles su rol correspondiente.

### 3. Precondiciones de Sistema

<< Autenticar usuario >>

### 3. Flujo de Eventos

#### 3.1 Flujo Básico

El caso de uso inicia cuando el administrador del sistema requiere dar de alta un usuario, actualizar sus roles, ver los usuarios dados de alta, o eliminar alguno del sistema.

- 1.- El sistema muestra la opción de agregar usuario.
- 2.- El sistema muestra la opción de consultar usuarios.

#### 3.2 Flujos Alternativos

1. Agregar usuarios.
  - 1.1. El sistema muestra una lista de todos los usuarios disponibles con la opción de filtrarlos por su nombre o su clave.
  - 1.2. El usuario elige al usuario deseado.
  - 1.3. El sistema solicita los roles para el usuario seleccionado y muestra las opciones de agregar y cancelar.
  - 1.4. El usuario elige los roles y selecciona la opción de agregar.
    - 1.4.1. El sistema valida los datos y muestra un mensaje de confirmación con la siguiente leyenda: “el usuario se agregó correctamente”.
  - 1.5. El usuario selecciona la opción de cancelar.
    - 1.5.1. El sistema lo redirige al flujo básico de este caso de uso.
2. El usuario selecciona la opción de consultar usuarios

- 2.1. El sistema muestra el nombre y los roles de todos los usuarios del sistema
- 2.2. El usuario selecciona un usuario.
- 2.3. El sistema muestra las opciones de actualizar roles, eliminar usuario o cancelar.
  - 2.3.1. El usuario selecciona la opción de actualizar roles.
    - 2.3.1.1. El sistema muestra el nombre del usuario con los roles asignados permitiendo agregar nuevos o eliminar los existentes y mostrando las opciones de actualizar y cancelar.
    - 2.3.1.2. El usuario edita los datos y selecciona la opción de actualizar.
      - 2.3.1.2.1. El sistema valida los datos y muestra el siguiente mensaje de confirmación “los roles se han asignado correctamente”.
    - 2.3.1.3. El usuario selecciona la opción de cancelar.
      - 2.3.1.3.1. El sistema regresa al flujo básico de este caso de uso.
  - 2.3.2. El usuario selecciona la opción de eliminar usuario.
    - 2.3.2.1. El sistema muestra un mensaje de confirmación preguntando si realmente se desea eliminar el usuario y las opciones de confirmar y cancelar.
      - 2.3.2.1.1. El usuario elige la opción de confirmar.
        - 2.3.2.1.1.1. El sistema muestra el siguiente mensaje de confirmación “el usuario se ha eliminado correctamente” y redirige al flujo básico de este caso de uso.
        - 2.3.2.1.2. El usuario selecciona la opción de cancelar.
          - 2.3.2.1.2.1. El sistema redirige al flujo básico de este caso de uso.
    - 2.3.3. El usuario selecciona la opción de cancelar.
    - 2.3.4. El sistema redirige al flujo básico de este caso de uso.

6. Poscondiciones

Sin poscondiciones.

7. Excepciones

Sin excepciones.

8. Requerimientos Especiales

Sin requerimientos especiales por el momento.

## 1. Especificación del caso de uso: Generar vista de negocio

### 1.1 Descripción

Este caso de uso describe las acciones a realizar para generación de una vista de negocio, a la cual podrán acceder los analistas de negocio.

### 2. Actores Involucrados

Analista de informática: Este tipo de usuario podrá acceder a la vista técnica.

Custodio del inventario: Este tipo de usuario podrá acceder a la vista técnica.

Administrador del sistema: Este tipo de usuario podrá acceder a la vista técnica.

### 3. Precondiciones de Sistema

<< Autenticar usuario >>

### 3. Flujo de Eventos

#### 3.1 Flujo Básico

El caso de uso inicia cuando el usuario quiere acceder a la vista técnica del sistema

1.- El sistema muestra la opción de Vista técnica.

#### 3.2 Flujos Alternativos

3. Vista técnica.

3.1. El sistema muestra los modelos de servicios existentes (tarea, utilería, entidad) así como la opción de consultar los servicios de cada modelo.

3.1.1. El usuario elige servicios de tarea.

3.1.1.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color azul.

3.1.1.1.1. El usuario elige un servicio.

3.1.1.1.2. El sistema muestra los datos del servicio

<<Administrar servicios >>

3.1.2. El usuario elige los servicios de tipo utilería.

3.1.2.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color amarillo.

3.1.2.1.1. El usuario elige un servicio.

3.1.2.1.2. El sistema muestra los datos del servicio  
<<Administrar servicios>>

3.1.3. El usuario elige los servicios de tipo entidad.

3.1.3.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color gris.

3.1.3.1.1. El usuario elige un servicio.

3.1.3.1.2. El sistema muestra los datos del servicio  
<<Administrar servicios>>

## 6. Poscondiciones

Sin poscondiciones.

## 7. Excepciones

Sin excepciones.

## 8. Requerimientos Especiales

Sin requerimientos especiales por el momento.

## 1. Especificación del caso de uso: Generar vista técnica

### 1.1 Descripción

Este caso de uso describe las acciones a realizar para generación de una vista técnica, a la cual podrán acceder los analistas de informática.

### 2. Actores Involucrados

Analista de informática: Este tipo de usuario podrá acceder a la vista técnica.

Custodio del inventario: Este tipo de usuario podrá acceder a la vista técnica.

Administrador del sistema: Este tipo de usuario podrá acceder a la vista técnica.

### 3. Precondiciones de Sistema

<< Autenticar usuario >>

### 3. Flujo de Eventos

#### 3.1 Flujo Básico

El caso de uso inicia cuando el usuario quiere acceder a la vista técnica del sistema

1.- El sistema muestra la opción de Vista técnica.

#### 3.2 Flujos Alternativos

##### 1. Vista técnica.

1.1. El sistema muestra los modelos de servicios existentes (tarea, utilería, entidad) así como la opción de consultar los servicios de cada modelo.

1.1.1. El usuario elige servicios de tarea.

1.1.1.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color azul.

1.1.1.1.1. El usuario elige un servicio.

1.1.1.1.2. El sistema muestra los datos del servicio

<<Administrar servicios >>

1.1.2. El usuario elige los servicios de tipo utilería.

1.1.2.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color amarillo.

1.1.2.1.1. El usuario elige un servicio.

1.1.2.1.2. El sistema muestra los datos del servicio .<<Administrar servicios>>

1.1.3. El usuario elige los servicios de tipo entidad.

1.1.3.1. El sistema muestra los servicios de tipo tarea en círculos dentro de una zona de color gris.

1.1.3.1.1. El usuario elige un servicio.

1.1.3.1.2. El sistema muestra los datos del servicio <<Administrar servicios>>

## 6. Poscondiciones

Sin poscondiciones.

## 7. Excepciones

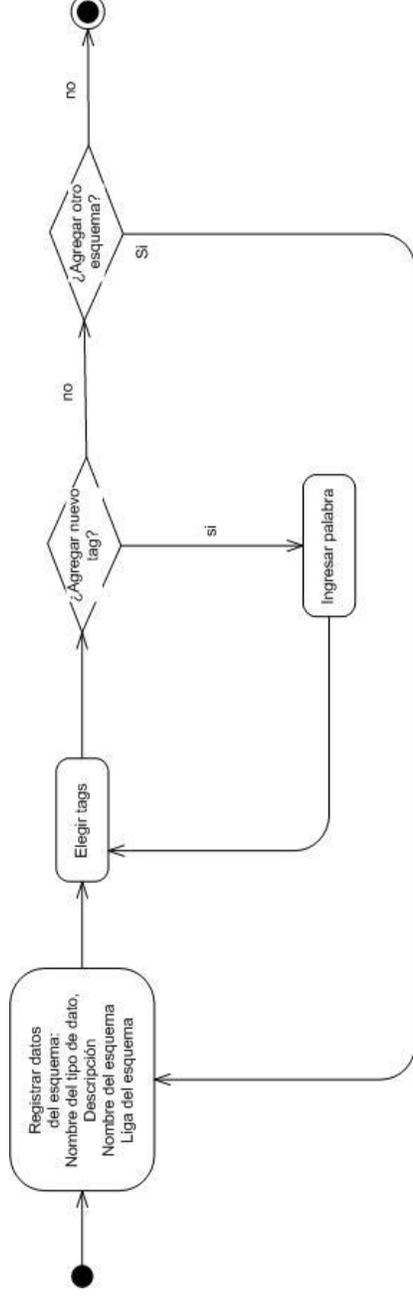
Sin excepciones.

## 8. Requerimientos Especiales

Sin requerimientos especiales por el momento.

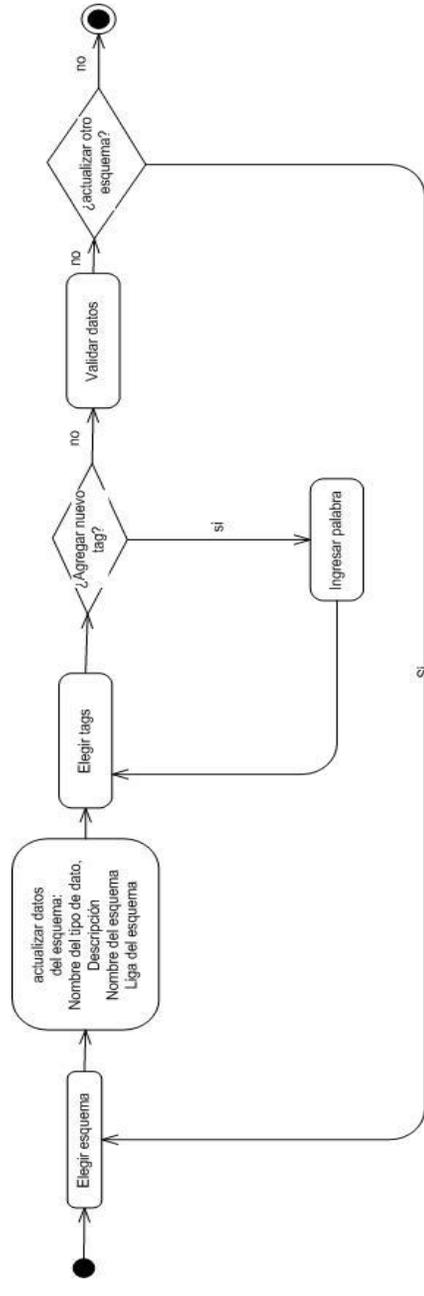
**Diagramas de actividad**

**Agregar esquemas**



**Diagrama de actividad 3: agregar esquemas**

**Actualizar esquemas**



**Diagrama de actividad 4: actualizar esquemas**

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Generar vista de negocio

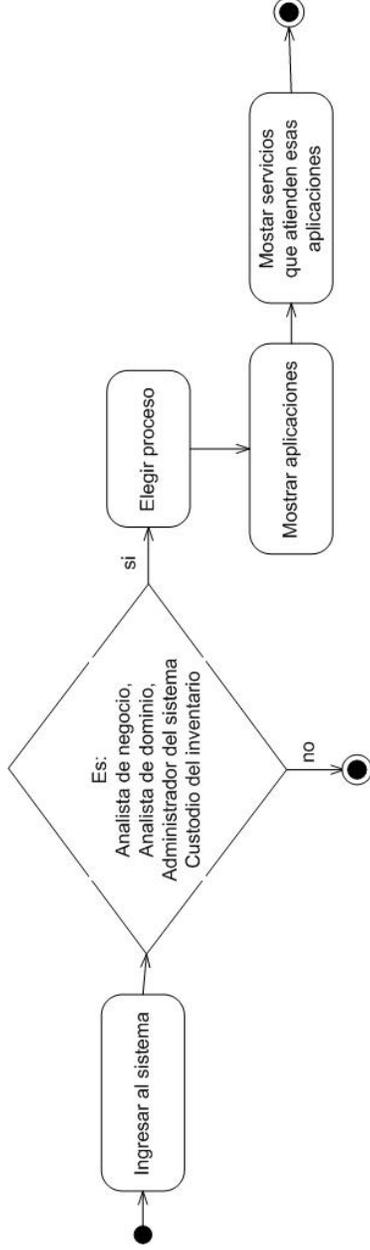


Diagrama de actividad 5: generar vista de negocio

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Generar vista técnica

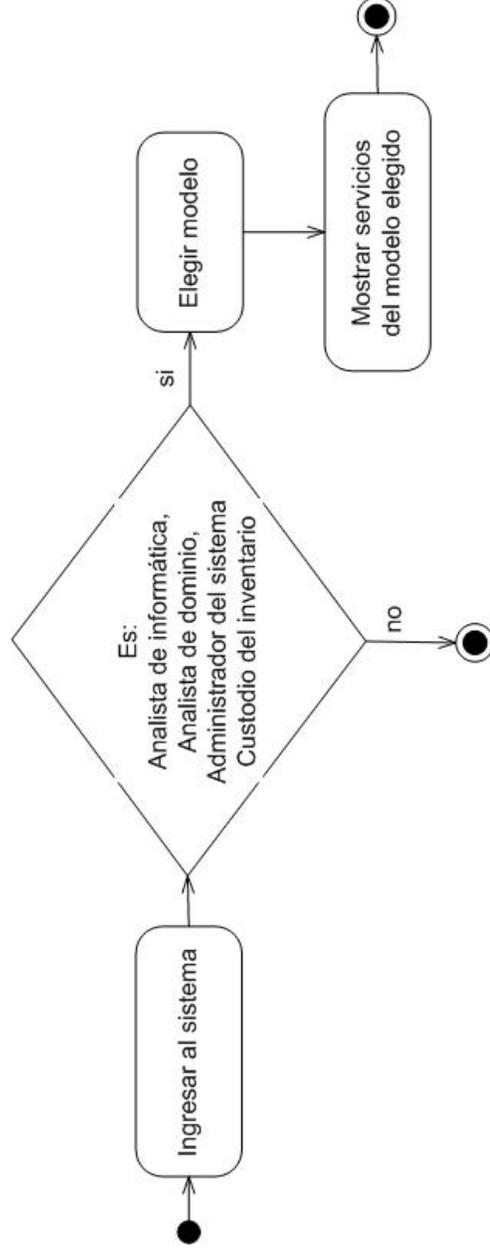


Diagrama de actividad 6: generar vista técnica

### Agregar Usuarios

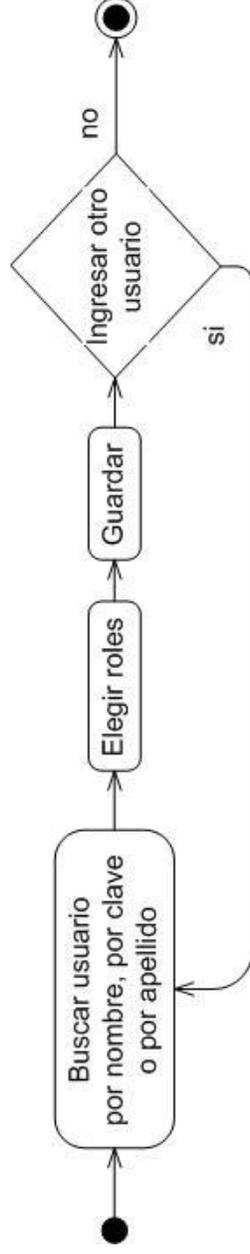


Diagrama de actividad 7: agregar usuarios

Definición e implementación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Actualizar y eliminar usuarios

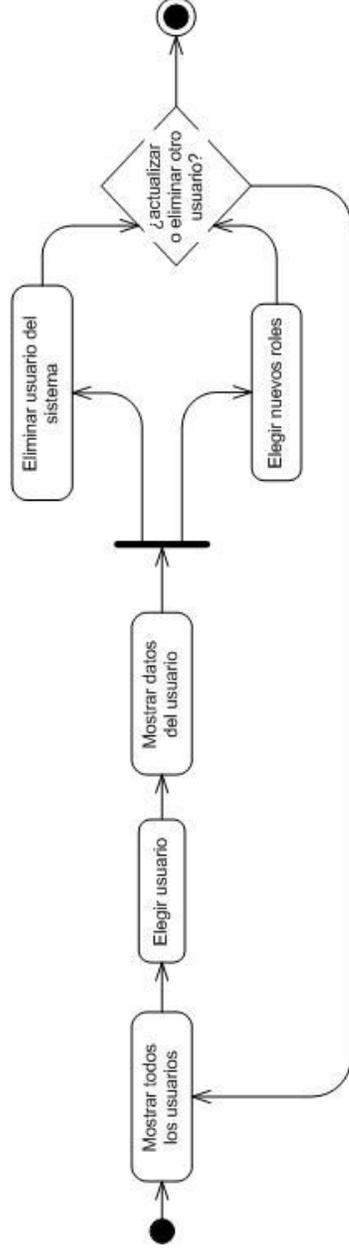


Diagrama de actividad 8: Actualizar, eliminar usuarios

### Generar bitácora

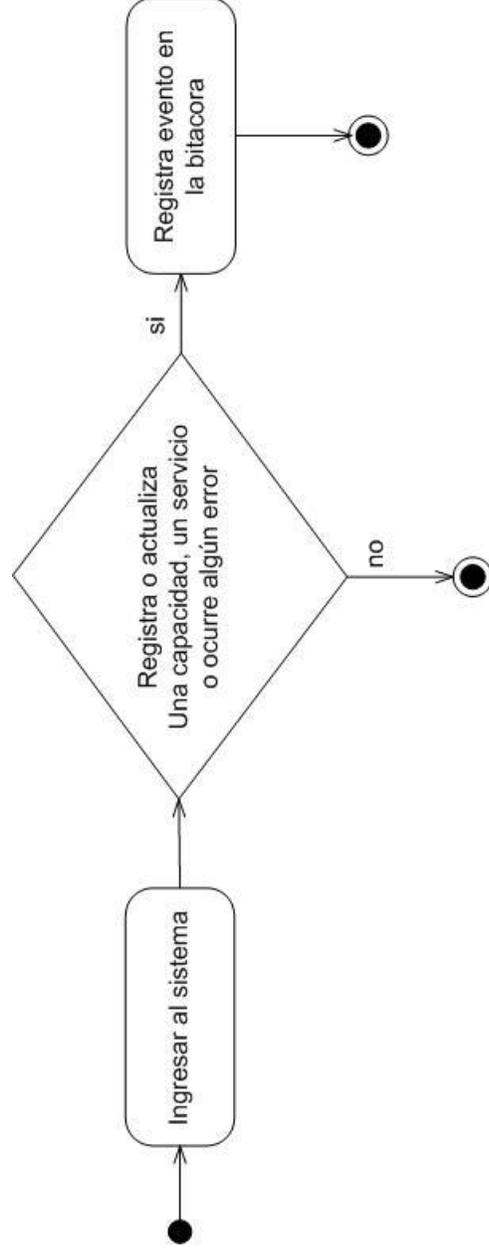


Diagrama de actividad 9: generar bitácora

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

## Diagramas de secuencia

### Gestionar Esquemas

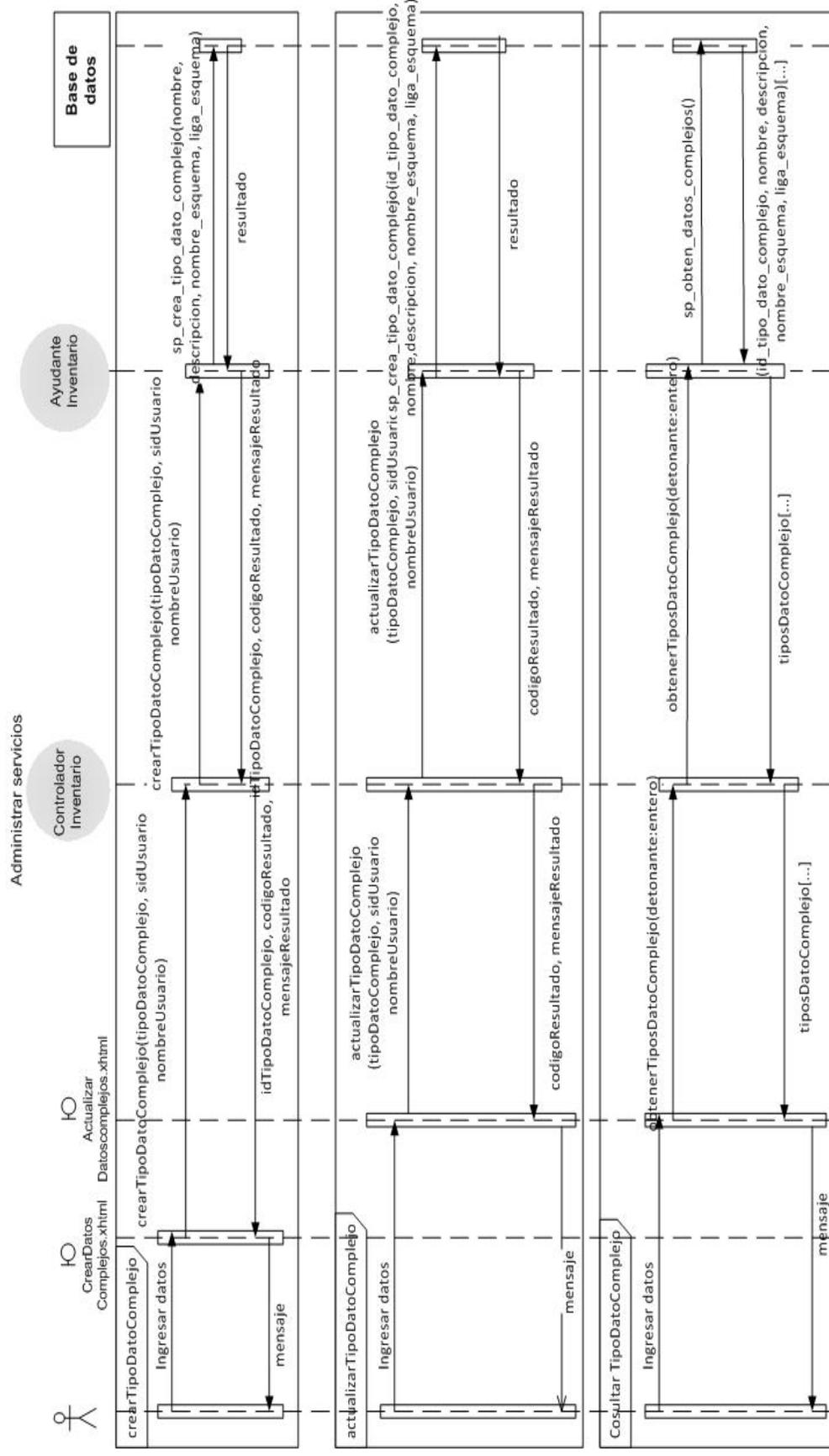


Diagrama de secuencia 4: gestionar esquemas

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Generar bitácora

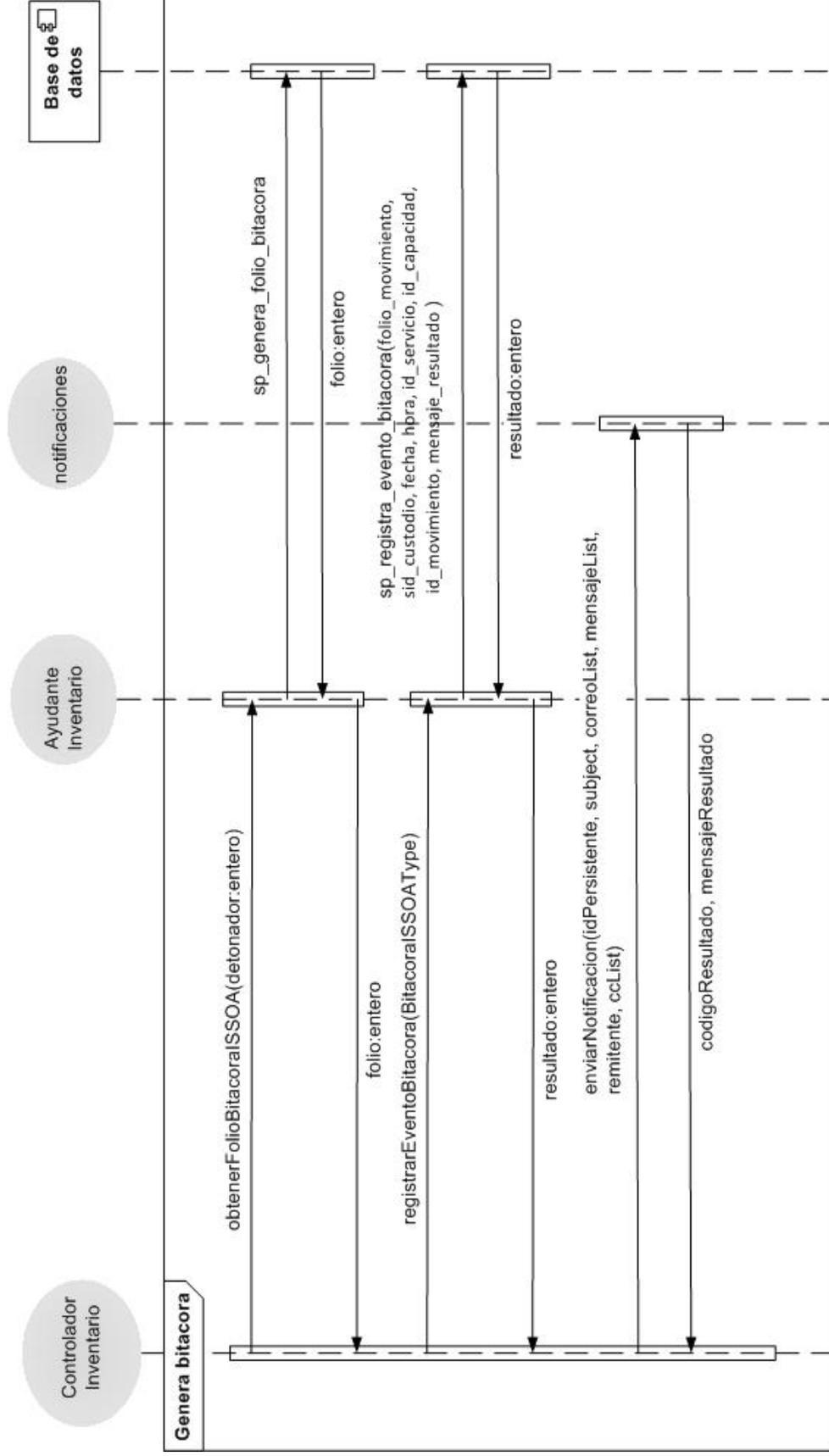


Diagrama de secuencia 5: generar bitácora ISSOA

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Generar vista de negocio

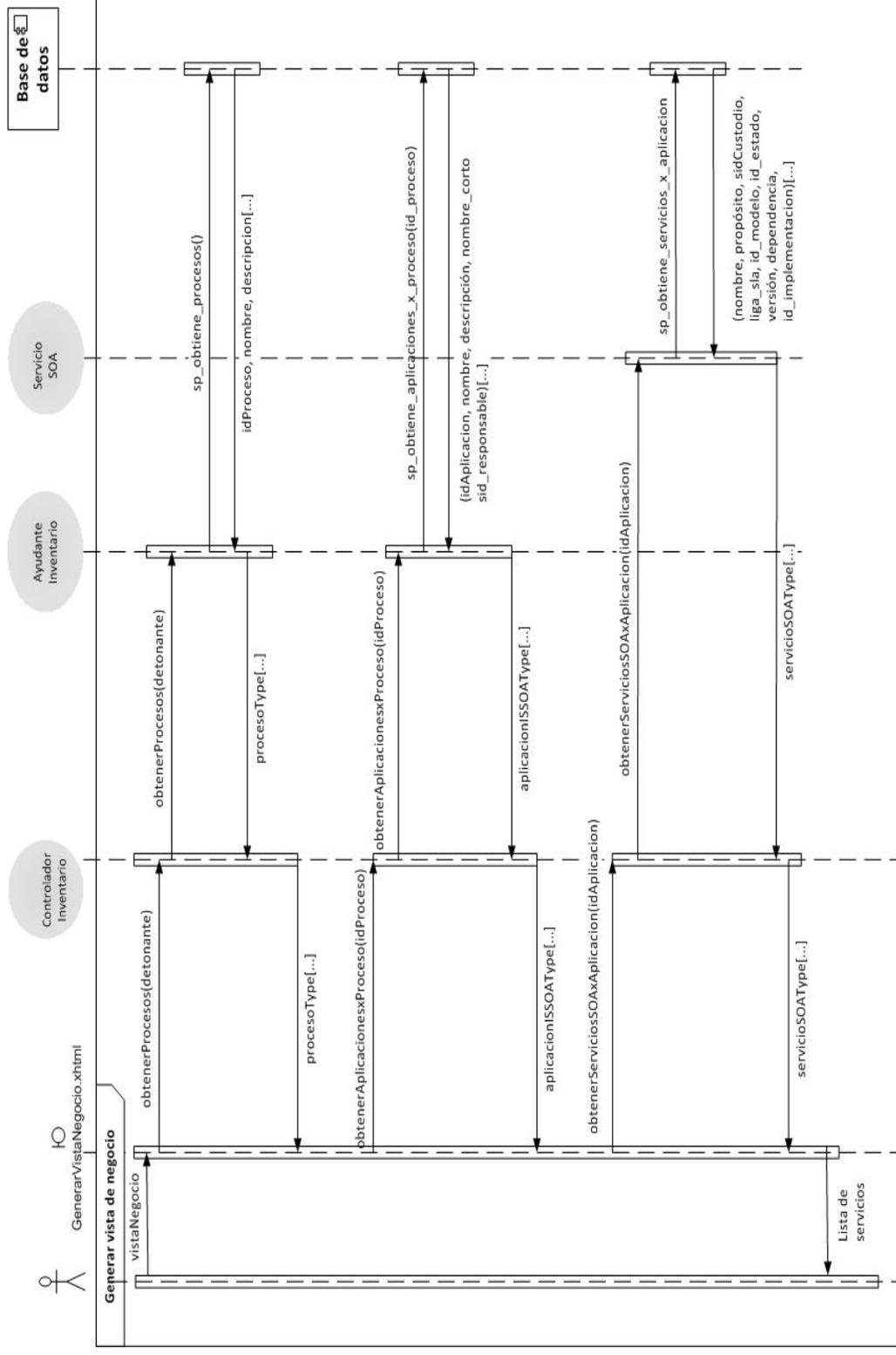


Diagrama de secuencia 6: generar vista de negocio

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

### Generar vista técnica

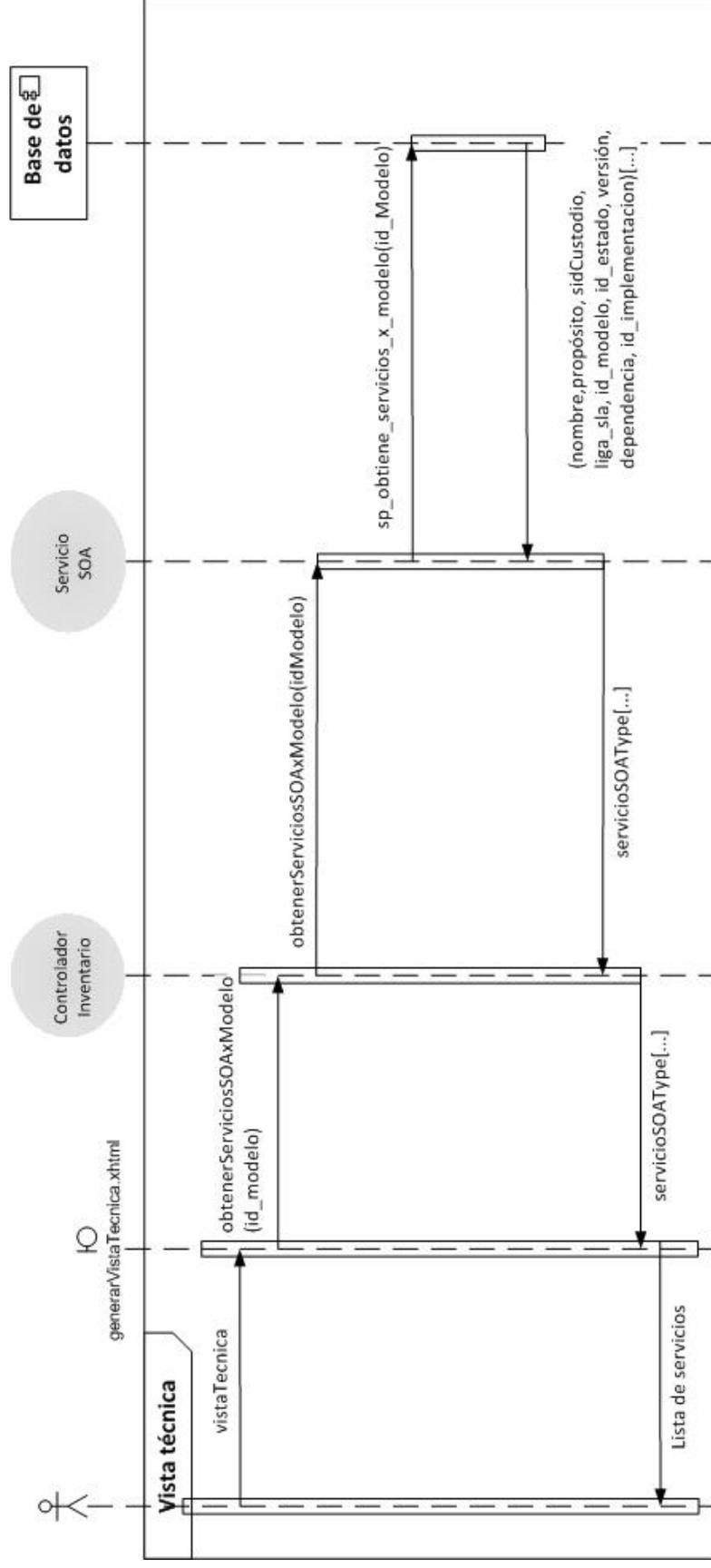


Diagrama de secuencia 7: Generar vista técnica

### ***Perfiles de servicios y capacidades***

En esta fase se desarrollaron los siguientes perfiles de servicios:

- AyudanteInventario

Y se actualizaron:

- Servicio SOA
- Capacidad SOA
- Controlador Inventario

De igual manera se realizaron los siguientes perfiles de capacidades:

- actualizarTipoDatoComplejo
- actualizarUsuarioISSOA
- asignarTagTipoDatoComplejo
- borrarTagsTipoDatoComplejo
- crearTag
- crearTipoDatoComplejo
- crearUsuarioISSOA
- obtenerActividadesxProceso
- obtenerAplicacionesISSOA
- obtenerAplicacionesxAct
- obtenerTiposDatoSimplexCapacidad
- obtenerUsuariosISSOA
- obtenerServiciosSOAxAplicacion
- obtenerAplicacionesXCapacidad
- obtenerFolioBitacoraISSOA
- obtenerProcesosISSOA
- obtenerRolxSID
- obtenerTagsISSOA
- obtenerTagsxCapacidad
- obtenerTagsxTipoDatoComplejo
- obtenerTipoDatoComplejoId
- obtenerTipoDatoSimplexId
- obtenerTiposDatoComplejo
- obtenerTiposDatoComplejoxCapacidad
- registrarEventoBitacoraISSOA
- obtenerServiciosSOAxModelo

A fin de ejemplificar los perfiles de servicios y capacidades se seleccionaron los más complejos que a continuación se muestran.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

<b>Perfil del Servicio AyudanteInventario</b>	
<i>Nombre del Servicio</i>	<b>AyudanteInventario</b>
<i>Propósito</i>	Servicio encargado de actividades que ayudan a la correcta operación del sistema ISSOA, actualizar y crear los clientes que no son del tipo SOA, aumentar folios, registrar eventos en bitácora etc.
<i>Modelo del servicio</i>	Utileria
<i>Capacidades.</i>	actualizarTipoDatoComplejo actualizarUsuarioISSOA asignarTagTipoDatoComplejo borrarTagsTipoDatoComplejo crearTag crearTipoDatoComplejo crearUsuarioISSOA obtenerTipoDatoSimplexId obtenerTiposDatoComplejo obtenerTiposDatoComplejoxCapacidad obtenerTiposDatoSimplexCapacidad obtenerUsuariosISSOA registrarEventoBitacoraISSOA.
<i>Palabras clave.</i>	Ayudante, utilería, esquemas, datos complejos.
<i>Versión.</i>	V 1.0
<i>Custodio.</i>	Jorge de Jesús Sánchez Aguilar
<i>Tipo de Implementación.</i>	Servicio web.
<i>Dependencias.</i>	ninguna.

Tabla 23: perfil del servicio Ayudante

<b>Perfil de la Capacidad crearTipoDatoComplejo</b>									
Nombre de la Capacidad*	crearTipoDatoComplejo								
Propósito *	Permite crear los tipos de dato complejo, también llamados esquemas utilizados por los servicios del ISSOA, tanto para entradas como salidas.								
Entradas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>tipoDatoComplejo</td> <td>TipoDatoComplejoType</td> </tr> </tbody> </table>	Nombre	Tipo	tipoDatoComplejo	TipoDatoComplejoType				
Nombre	Tipo								
tipoDatoComplejo	TipoDatoComplejoType								
Salidas*	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>idTipoDatoComplejo</td> <td>Entero</td> </tr> <tr> <td>codigoResultado</td> <td>Entero</td> </tr> <tr> <td>mensajeResultado</td> <td>Cadena</td> </tr> </tbody> </table>	Nombre	Tipo	idTipoDatoComplejo	Entero	codigoResultado	Entero	mensajeResultado	Cadena
Nombre	Tipo								
idTipoDatoComplejo	Entero								
codigoResultado	Entero								
mensajeResultado	Cadena								
Nombre del Desarrollador*	Jorge de Jesús Sánchez Aguilar								
Clientes SOA	<b>crearTipoDatoComplejo</b> <i>de</i> <b>ControladorInventario.</b>								
Clientes no SOA	<b>ninguno</b>								
Tags*	Esquema, tipo complejo, ISSOA, entradas, salidas.								
Versión*	V 1.0								
Aplicación*	ISSOA								

Tabla 24: perfil de la capacidad crear tipo complejo

## Esquemas utilizados

<b>Nombre TDC:</b>	<b>TipoDatoComplejoType</b>
<b>Descripción:</b>	Tipo de dato que describe un tipo de dato complejo representado a través de un esquema.
<b>Nombre del esquema:</b>	TipoDatoComplejo.xsd
<b>Tags:</b>	Tipo complejo, esquema, servicio.
<b>Atributos:</b>	
<b>Nombre</b>	<b>Tipo</b>
idTipoDatoComplejo	entero
nombre	Cadena
descripcion	Cadena
nombre_esquema	Cadena
liga_esquema	Cadena

Tabla 25: esquema tipo dato complejo

### **Blue print de servicios**

A continuación se presenta el blue print actualizado en esta fase.



**Ilustración 28: blue print servicios actualizado**

### **Contratos de servicios**

En esta fase se actualizan los contratos de los servicios:

- ServicioSOA.
- CapacidadSOA.
- ControladorInventario.

De igual manera se desarrolla el contrato del servicio: AyudanteInventario.

#### *ServicioSOA*

A fin de ejemplificar se muestran las actualizaciones del contrato del servicio ServicioSOA, mostrado en la fase de elaboración.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<definitions name="ServicioSOA"
```

```
    targetNamespace="http://banxico.org.mx/contrato/dgef/es/ServicioSOA/wsdl/"
```

```
    xmlns=http://schemas.xmlsoap.org/wsdl/
```

```
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
    xmlns:tns="http://banxico.org.mx/contrato/dgef/es/ServicioSOA/wsdl/"
```

```
    xmlns:sss="http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/v1/"
```

```
    xmlns:com=http://banxico.org.mx/schema/dgef/us/Comun/v1/
```

```
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <documentation>
```

Este es el servicio de entidad ServicioSOA encargado de realizar diversas tareas relacionadas con, el servicio ServicioSOA como crear, actualizar, y obtener la entidad Servicio de acuerdo con diversos parámetros.

Autor Sánchez Aguilar Jorge de Jesús.

Version 1.0

GSISF

```
</documentation>
```

```
<types>

  <xsd:schema

    targetNamespace                                =
      "http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/"

    <xsd:import

      namespace = "http://banxico.org.mx/schema/dgef/es/ServicioServicioSOA/v1/"
      schemaLocation = "../schemas/ServicioServicioSOA.xsd" />

    <xsd:import namespace = "http://banxico.org.mx/schema/dgef/us/Comun/v1/"
      schemaLocation="../schemas/comun/Comun.xsd" />

    </xsd:schema>

  </types>

  <message name="ExcepcionNegocioMsg">

    <part name="ResponseParameter" element="com:ExcepcionNegocioType">

    </part>

  </message>

  <message name="ObtenerServiciosSOAxAplicacionEntradaMsg">

    <part name = "RequestParameter"
      element="sss:ObtenerServiciosSOAxAplicacionEntradaType" />

  </message>

  <message name="ObtenerServiciosSOAxAplicacionSalidaMsg">

    <part name="ResponseParameter"
      element="sss:ObtenerServiciosSOAxAplicacionSalidaType" />

  </message>
```

```
<message name="ObtenerServiciosSOAxModeloEntradaMsg">
    <part name="RequestParameter"
        element="sss:ObtenerServiciosSOAxModeloEntradaType" />
</message>

<message name="ObtenerServiciosSOAxModeloSalidaMsg">
    <part name="ResponseParameter"
        element="sss:ObtenerServiciosSOAxModeloSalidaType" />
</message>

<message name="ObtenerServiciosISSOAEntradaMsg">
    <part name="RequestParameter"
        element="sss:ObtenerServiciosISSOAEntradaType" />
</message>

<message name="ObtenerServiciosISSOASalidaMsg">
    <part name="ResponseParameter"
        element="sss:ObtenerServiciosISSOASalidaType" />
</message>

<portType name="ServicioSOAInterface">
    <operation name="ObtenerServiciosSOAxAplicacion">
        <input message="tns:ObtenerServiciosSOAxAplicacionEntradaMsg" />
        <output message="tns:ObtenerServiciosSOAxAplicacionSalidaMsg" />
        <fault name="businessException" message="tns:ExcepcionNegocioMsg">
        </fault>
    </operation>
```

```
<operation name="ObtenerServiciosSOAxModelo">
    <input message="tns:ObtenerServiciosSOAxModeloEntradaMsg" />
    <output message="tns:ObtenerServiciosSOAxModeloSalidaMsg" />
    <fault name="businessException" message="tns:ExcepcionNegocioMsg">
    </fault>
</operation>
<operation name="ObtenerServiciosISSOA">
    <input message="tns:ObtenerServiciosISSOAEntradaMsg" />
    <output message="tns:ObtenerServiciosISSOASalidaMsg" />
    <fault name="businessException" message="tns:ExcepcionNegocioMsg">
    </fault>
</operation>
</portType>

<binding name="ServicioSOABinding" type="tns:ServicioSOAInterface">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="ObtenerServiciosSOAxAplicacion">
        <soap:operation
            soapAction=
                "http://banxico.org.mx/dgef/es/ServicioSOA/ObtenerServiciosSOAxAplicacion" />
        <input>
            <soap:body use="literal" />
        </input>
    </operation>
</binding>
```

```
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="businessException">
    <soap:fault name="businessException" use="literal" />
</fault>
</operation>

<operation name="ObtenerServiciosSOAxModelo">
    <soap:operation
        soapAction=
            "http://banxico.org.mx/dgef/es/ServicioSOA/ObtenerServiciosSOAxModelo" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="businessException">
        <soap:fault name="businessException" use="literal" />
    </fault>
</operation>
```

```
<operation name="ObtenerServiciosISSOA">
  <soap:operation
    soapAction=
      "http://banxico.org.mx/dgef/es/ServicioSOA/ObtenerServiciosISSOA" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="businessException">
    <soap:fault name="businessException" use="literal" />
  </fault>
</operation>
</binding>
<service name="ServicioSOAService">
  <port name="ServicioSOAPort" binding="tns:ServicioSOABinding">
    <soap:address location="http://banxico.org.mx/dgef/es/ServicioSOA"/>
  </port>
</service>
</definitions>
```

### ***Reporte de pruebas***

El siguiente, es el reporte de pruebas, generado en la fase de construcción.

#### *Resume de pruebas*

Nombre del proyecto: Inventario de Servicios SOA.

Nombre del sistema: Inventario de Servicios SOA.

Versión: 1.0

Comentarios adicionales: Este documento muestra un reporte de las pruebas a las que fue sometido el sistema de Inventario de Servicios SOA, las pruebas realizadas se enlistan a continuación:

- Función.
- Usabilidad.
- Integración.
- Seguridad.

#### *Pruebas de Función*

Responsables de las pruebas (probador): probador.

Resultado de las pruebas: El sistema cubrió satisfactoriamente todos los requerimientos funcionales.

Comentarios adicionales: Las pruebas se realizaron exitosamente, no hay incidencias que hayan interrumpido el proceso de pruebas.

#### *Pruebas de Integración*

Responsables de las pruebas (probador): probador.

Resultado de las pruebas: Las pruebas de integración se generaron correctamente, se encontró un error pero el equipo de trabajo lo resolvió de manera exitosa.

Comentarios adicionales: Los perfiles de las capacidades, después de las pruebas, quedaron actualizados y en orden con cada capacidad.

### *Pruebas de Usabilidad*

Responsables de las pruebas (probador): probador.

Resultado de las pruebas: Se encontraron algunos problemas pero se resolvieron correctamente de manera general.

Comentarios adicionales: Las pruebas se realizaron satisfactoriamente, no hay incidencias que hayan interrumpido el proceso de pruebas.

### *Pruebas de Seguridad*

Responsables de las pruebas (probador): Dirección General de Tecnologías de la Información (DGTI).

Resultado de las pruebas: Se sometieron a prueba cuatro aplicaciones de las cuales tres arrojaron resultados positivos, es decir, no se encontraron vulnerabilidades.

Respecto a la cuarta prueba se registró una vulnerabilidad de carácter crítico: “Solicitud de inicio de sesión, no cifrado”, pero debido a que la aplicación será de uso interno, el problema no causan ningún impacto en las pruebas de Seguridad.

### *Evaluación de las pruebas*

Las pruebas fueron documentadas de acuerdo al plan de pruebas. Los archivos generados se llenaron de manera correcta verificando que se cumplan con los requerimientos e indicando las observaciones necesarias en caso de incidencias.

*Resultados de la pruebas*

Las pruebas efectuadas tanto las que generaron entregables como las que no lo hicieron, tuvieron resultados favorables por lo que se concluye que el sistema cumple con los objetivos establecidos.

*Pruebas de función*

Probador	Resultado (aprobado, no aprobado).	Severidad de defectos encontrados	Resumen de los defectos encontrados.	Resuelto antes del lanzamiento	Comentarios.
probador	Aprobado	Baja	Hace falta crear el flujo de modificar tipo de dato en capacidad.	Sí	Se creó correctamente el flujo de Modificar tipo de dato complejo.

**Tabla 26: reporte de pruebas de función**

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

*Pruebas de integración*

Probador	Resultado (aprobado, no aprobado).	Severidad de defectos encontrados.	Resumen de los defectos encontrados.	Resuelto antes del lanzamiento	Comentarios.
Porbador	Aprobado	Media	Se permite tener como cliente de una capacidad la misma capacidad.	Sí	Se puso una restricción a nivel de base de datos para que no se pueda realizar esa asignación y a nivel de vista se eliminó la capacidad que se modifica de la lista de clientes.

**Tabla 27: reporte de pruebas de integración**

*Pruebas de usabilidad*

Probador	Resultado (aprobado, no aprobado).	Severidad de defectos encontrados.	Resumen de los defectos encontrados.	Resuelto antes del lanzamiento	Comentarios.
probador	Aprobado	Baja	Al crear una nueva capacidad, el sistema muestra los mensajes de error de la capacidad anterior.	Si	Se resolvió el problema creando una restricción.
probador	Aprobado	Baja	Los campos de la aplicación se encuentran en desproporción .	Si	Se modificaron los campos.
probador	Aprobado	Media	Al crear o actualizar una capacidad se mandan muchos correos.	Si	Sólo se actualizó el controlador y se envía sólo un correo.

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

probador	Aprobado	Baja	La información de consulta de capacidad está incompleta.	Si	Se agregaron nuevas columnas para mejorar la consulta.
probador	Aprobado	Baja	No hay estándar para las versiones de capacidad y servicio.	Si	Se estableció un estándar para las versiones.

Tabla 28: reporte pruebas de usabilidad

Definición e implantación de una metodología de desarrollo de soluciones orientadas a servicios para la DISF de Banco de México a través de un proyecto piloto

*Pruebas de seguridad*

Probador	Resultado (aprobado, no aprobado).	Severidad de defectos encontrados.	Resumen de los defectos encontrados.	Resuelto antes del lanzamiento	Comentarios.
DGTI	Aprobado	Alta	El login de acceso no se envía cifrado.	Si	El problema encontrado no tiene impacto en el sistema debido a que es una aplicación interna.

Tabla 29: reporte de pruebas de seguridad

## **Conclusiones**

De acuerdo al propósito del proyecto, se diseñó una solución orientada a servicios que administra los servicios web de la Dirección de Información del Sistema Financiero, esto mediante la aplicación de una metodología que integra *Service Oriented Architecture (SOA)* y el proceso de desarrollo *Rational Unified Process (RUP)*.

El proyecto realizado cumplió en alcance y tiempo, gracias a la metodología diseñada, respetando la implementación de SOA en la DISF e incorporando las buenas prácticas de RUP.

Cuando se realizó el mapeo de SOA con RUP se encontraron artículos de investigación( (Ahmad & Ahmad, 2010), (Valdivia, 2010), (Santas, 2011) )que fueron de gran valor para este trabajo.

Se integró la notación SOA con la notación UML propuesta por RUP, donde se tomó como eje rector esta última. Gracias esto se pudo cumplir con la actividad de análisis y diseño de la solución.

Uno de los problemas que surgieron durante el proyecto, fue que el equipo estaba poco familiarizado con la aplicación de una metodología, por lo que, la curva de aprendizaje fue alta, se tuvieron que estudiar buenas prácticas de RUP, notaciones UML y diversos entregables de análisis y diseño, entre otros. Esto se resolvió con capacitación y reuniones semanales. En estos eventos se plantearon problemas y dudas con respecto a las actividades de cada miembro del equipo y se procedieron a resolver.

Al terminar la fase de construcción, se entregó una versión beta del sistema “Inventario de Servicios SOA” (ISSOA), que pasó las pruebas realizadas, cumpliendo con el objetivo y satisfaciendo las necesidades de los interesados del proyecto.

De igual manera, el “ISSOA” ayudó a fortalecer la implementación de SOA en la DISF, ya que ahora se tiene información oportuna y precisa del inventario de servicios; uno de los interesados denominado especialista en gobernabilidad SOA, por ejemplo, tiene una mejor gestión del inventario de servicios con lo que puede verificar el cumplimiento de preceptos y principios SOA, tales como: el descubrimiento de servicios, la autonomía y composición de los mismos y su bajo acoplamiento.

Otro beneficio del ISSOA es que ayudó a la alineación de TI con el negocio, ya que con una de las vistas realizadas, cualquier analista de negocio puede acceder e identificar los servicios SOA que atienden a los procesos de negocio, con ellos se obtiene información que ayuda a identificar fallas u oportunidades de mejora en la automatización de su proceso.

Actualmente, la metodología sigue madurando, y se aplica en el desarrollo de dos soluciones orientadas a servicios en la DISF, una para el mantenimiento de servidores de archivos y otra enfocada al proceso de acopio de información<sup>1</sup>. La integración de SOA con RUP realizada puede ser referencia para aquellas organizaciones que, requieran desarrollar soluciones orientadas a servicios.

---

<sup>1</sup> Para más información consultar el capítulo primero en la sección de “descripción de procesos principales”.

## **Glosario**

**Artefacto:** Una pieza de información que es producida, modificada o usada por un proceso, define un área de responsabilidad y está sujeta al control de versiones. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento. Un documento puede incluir otros documentos. (Kruchten, 2003)

**Capacidad:** Un servicio puede ser visto como un contenedor para una colección de funciones relacionadas. Estas funciones se denominan capacidades de servicio y son expuestas a través de un contrato de servicios que establece una API básica mediante la cual el servicio puede ser invocado. Una vez que se sepa si existe un servicio como un servicio Web o como un componente, los términos "método de servicio" o "la operación del servicio" se pueden utilizar en su lugar. (Inc., ServiceOrientation.com, 2005- 2014)

**Contrato:** Conjunto de documentos publicados que expresan la información técnica de un servicio. (Inc., ServiceOrientation.com, 2005- 2014)

**Custodio:** Un custodio es la persona que posee la gestión y las responsabilidades de gobernanza de uno o más servicios específicos. Estas funciones no sólo giran en torno a la extensión, expansión y mantenimiento de la lógica de servicio, también incluyen tener que proteger la integridad del contexto del servicio y su límite funcional asociado. (Inc., ServiceOrientation.com, 2005- 2014)

**Esquemas:** Documentos que proporcionan un medio para definir la estructura, contenido y semántica de los documentos XML. (Curbera, Christensen, Meredith, & Weerawarana, 2001)

**Iteración:** Una secuencia clara de las actividades con un plan y criterios de evaluación que se traducen en un comunicado (interno o externo). (Kruchten, 2003)

**Políticas:** Las políticas proveen parámetros y guías de como las tareas y responsabilidades necesitan ser llevadas a cabo. Algunos de estos requerimientos son obligados por organizaciones externas, mientras que otros son impuestos por entidades internas en la organización. (Earl, SOA Governance Governing Shared Services On-Premise and in the Cloud., 2011)

**Precepto:** Un precepto es una regla autoritaria de acción. Los preceptos son la esencia de la gobernabilidad, ya que determinan quién tiene autoridad para tomar decisiones, establecen restricciones para esas decisiones, y prescriben consecuencias en caso de incumplimiento. A pesar de que un precepto puede contener estándares, algunos preceptos se consideran estándares. (Earl, SOA Governance Governing Shared Services On-Premise and in the Cloud., 2011)

**Servicio:** Un servicio es una unidad de solución lógica a la que se le ha aplicado la orientación a servicios en un grado significativo. Esto consiste en la aplicación de principios de diseño orientados a servicios que distinguen a una unidad de lógica como un servicio en comparación con las unidades lógicas que pueden existir como objetos o componentes. (Inc., ServiceOrientation.com, 2005- 2014)

**Vista de negocio:** Es una presentación del inventario de servicios SOA especialmente diseñada para el área de negocio y se basa principalmente en los procesos de negocio.

**Vista técnica:** Es una presentación del inventario de servicios SOA especialmente diseñada para el área de informática, basada en la clasificación de los servicios propuesta por SOA (servicios entidad, servicios tarea, servicios de utilería).

**Composición de servicios:** Es un conjunto coordinado de servicios. Una composición de servicios es comparable a una aplicación tradicional en que su alcance funcional se asocia generalmente con la automatización de un proceso de negocio de los padres. (Earl, SOA Principles of Service Design , 2008)

**Inventario de servicios:** Es una colección independientemente estandarizada y gobernada de servicios complementarios dentro de un marco definido que representa a una empresa o a un segmento significativo de ella. (Earl, SOA Governance Governing Shared Services On-Premise and in the Cloud., 2011)

## ***Bibliografía***

- Ahmad, B., & Ahmad, S. (1 de enero de 2010). Mapping of SOA and RUP: DOA as Case. *banxico*. (s.f.). *Banco de México*. Recuperado el 21 de agosto de 2014, de <http://www.banxico.org.mx/acerca-del-banco-de-mexico/organigrama.html>
- Curbera, F., Christensen, E., Meredith, G., & Weerawarana, S. (15 de marzo de 2001). *W3C*. Recuperado el 22 de agosto de 2014, de <http://www.w3.org/TR/wsd1>
- Earl, T. (2008). *SOA Principles of Service Design*. Boston, MA: prentice hall.
- Earl, T. (2011). *SOA Governance Governing Shared Services On-Premise and in the Cloud*. United States: Prentice hal.
- Gerencia, A. d. (junio de 2012). *Arquitectura objetivo del Macroproceso Básico*. Mexico, Distrito Federal.
- Grady, B., James, R., & Ivar, J. (2006). *El lenguaje unificado de modelado guía del usuario* (segunda ed.). Madrid, España: Addison Wesley.
- Hristo Toshkov Hristov, t. J. (2012). *OpenUP*. Recuperado el 22 de septiembre de 2014, de <http://epf.eclipse.org/wikis/openup/index.htm>
- Inc., A. E. (2005- 2014). *ServiceOrientation.com*. Recuperado el 15 de abril de 2014, de <http://serviceorientation.com/soaglossary/index>
- Inc., A. E. (2005- 2014). *ServiceOrientation.com*. Recuperado el 20 de abril de 2014, de [http://serviceorientation.com/static/pdf/SOA\\_Principles\\_Poster.pdf](http://serviceorientation.com/static/pdf/SOA_Principles_Poster.pdf)
- Información(Contraloría), O. d. (2011). *Lista de comprobación de cumplimiento de obligaciones mediante documentos*. México DF.
- Kruchten, P. (2003). *The rational unified process an introduction* (3a ed.). United States: Addison Wesley.
- Mangan, M. (04 de 08 de 2014). *Roadmap: OpenUP Roadmap*. Recuperado el 04 de 08 de 2014, de OpenUP: <http://epf.eclipse.org/wikis/openup/index.htm>
- Project Management Institute, I. (2008). *Guía de los fundamentos para la dirección de proyectos(guía del PMBOK) cuarta edición; 2008; (4a ed.)*. Newtown Square, Pennsylvania 19073-3299 , EE.UU: Project Management Institute.
- Santas, F. (2011). Mitigating Service-Oriented Risks with RUP. *SERVICE TECHNOLOGY MAGAZINE*, 12.
- Valdivia, M. M. (4 de junio de 2010). *Análisis y Diseño de Servicios en la Adopción de una Arquitectura Orientada a Servicios*. La habana, cuba.