

**UCLV**  
Universidad Central  
"Marta Abreu" de Las Villas



**MFC**  
Facultad de Matemática  
Física y Computación

**Departamento**

**Ingeniería Informática**

# **TRABAJO DE DIPLOMA**

## **BIMAS Smifor 3.0**

**Autor:** Daniel Rodríguez Fuentes

**Tutores:** DrC. Ana María García García

Ing. Dairo López Mollinedo

Santa Clara, Cuba, Noviembre 2021  
Copyright©UCLV

**UCLV**  
Universidad Central  
"Marta Abreu" de Las Villas



**MFC**  
Facultad de Matemática  
Física y Computación

**Department**

**Informatics Engineering**

**DIPLOMA THESIS**  
**BIMAS Smifor 3.0**

**Author:** Daniel Rodríguez Fuentes

**Mentors:** PhD. Ana María García García

Eng. Dairo López Mollinedo

Santa Clara, Cuba, noviembre 2021  
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

**Atribución- No Comercial- Compartir Igual**



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas.

Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-14190



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de **Ingeniería Informática**, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

---

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del tutor

---

Firma del tutor

---

Firma del jefe del Dpto

Esta tesis, con la que culmino mi carrera universitaria y fruto de un importante período de estudio, esfuerzo, dedicación y sacrificios se la dedico a la persona que ha estado a mi lado durante toda mi vida, la que me enseñó a luchar por mis ideas y sueños, la que dejó de ser de ella para ser nuestra, la que acompañó y apoyó cada uno de mis pasos.

*A ti, madre amada.*

*En un primer y muy importante lugar a mi abuelo materno, Elio: donde estés, gracias por tu ejemplo, tu fuerza, tu carácter mezclado con esa dulzura de abuelo y padre de familia. ¡Gracias!*

*Agradecer a mis padres, los que formaron en mí el amor al estudio, al trabajo y la superación.*

*A mi hermana, por ser el apoyo, confidente, amiga, madre y soporte en los buenos y malos tiempos.*

*A mi abuela, por su cariño, dedicación, su dulzura, apoyo y por ser la indiscutible madre de nuestra familia.*

*A mi novia y futura esposa; gracias por tu cariño, amor, por tu paciencia, por ser mi cómplice, mi amiga, mi estandarte en momentos de dificultad, mi compañera de vida.*

*A mi padrino Omar, por sus consejos, guía, por su tiempo.*

*A mis tutores.*

## **RESUMEN**

Smifor en su versión 2.0 es una herramienta para la extracción, transformación y carga de datos asociada al sistema de inteligencia de negocios BIMAS, ambos pertenecientes a la cartera de la empresa Desoft. Dicha herramienta presenta limitaciones en correspondencia con las necesidades actuales. En aras de solucionar dichas limitaciones el presente trabajo presenta una propuesta de solución en la cual se implementa una nueva forma de extracción de los datos que satisface las necesidades de la empresa, migrando el producto a la plataforma web y empleando las tecnologías requeridas para su uso como servicio, tanto en la nube como en premisa.

### **Palabras claves:**

Inteligencia de negocios, herramientas ETL, Servicios WEB, Smifor, BIMAS

**ABSTRACT**

Smifor version 2.0 is a tool for data extraction, transformation and loading associated with the BIMAS business intelligence system, both belonging to the Desoft company portfolio. Said tool has limitations in correspondence with current needs. In order to solve these limitations, this work presents a solution proposal in which a new form of data extraction is implemented that satisfies the needs of the company, migrating the product to the web platform and using the technologies required for its use. as a service, both in the cloud and on-premises.

**Keywords:**

Business intelligence, ETL tools, WEB services, Smifor, BIMAS



## Tabla de Contenidos

<b>INTRODUCCIÓN.....</b>	<b>1-2</b>
<b>1. MARCO TEÓRICO. HERRAMIENTAS ETL ASOCIADAS A SISTEMAS DE INTELIGENCIA DE NEGOCIOS.....</b>	<b>7</b>
1.1 <i>Inteligencia de negocios .....</i>	<i>7</i>
1.2 <i>Herramientas ETL .....</i>	<i>7</i>
1.3 <i>Algunos sistemas de Inteligencia de negocios y sus herramientas ETL.....</i>	<i>8</i>
1.3.1 <i>Oracle BI .....</i>	<i>8</i>
1.3.2 <i>Pentaho BI.....</i>	<i>9</i>
1.3.3 <i>SQL Server Management Studio .....</i>	<i>9</i>
1.3.4 <i>Spago BI .....</i>	<i>10</i>
1.4 <i>Entorno de desarrollo.....</i>	<i>11</i>
1.4.1 <i>Spring Framework y Springboot.....</i>	<i>11</i>
1.4.2 <i>Servicios Web.....</i>	<i>11</i>
1.4.3 <i>JSON.....</i>	<i>11</i>
1.4.4 <i>Bases de datos no SQL .....</i>	<i>12</i>
1.4.5 <i>Mongodb.....</i>	<i>13</i>
1.5 <i>Conclusiones Parciales.....</i>	<i>15</i>
<b>2. BIMAS SMIFOR: CARACTERÍSTICAS ACTUALES Y PERSPECTIVAS DESEADAS 17</b>	
2.1 <i>Bimas Smifor 2.0.....</i>	<i>17</i>
2.2 <i>Versión a desarrollar.....</i>	<i>18</i>
2.2.1 <i>Casos de Uso del Sistema .....</i>	<i>19</i>
2.3 <i>Mantenimiento de Software .....</i>	<i>19</i>
2.3.1 <i>Tipos de mantenimiento.....</i>	<i>20</i>
2.3.1.1 <i>Mantenimiento Correctivo .....</i>	<i>20</i>
2.3.1.2 <i>Mantenimiento Perfectivo .....</i>	<i>20</i>
2.3.1.3 <i>Mantenimiento Adaptativo.....</i>	<i>20</i>
2.3.1.4 <i>Mantenimiento a Bimas Smifor.....</i>	<i>21</i>
2.4 <i>Conclusiones parciales.....</i>	<i>21</i>
<b>3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN. METODOLOGÍA ÁGIL DE DESARROLLO Y PRUEBAS DE SOFTWARE.....</b>	<b>23</b>
3.1 <i>Metodología Ágil .....</i>	<i>23</i>
3.1.1 <i>Scrum.....</i>	<i>23</i>
3.1.2 <i>Artefactos de Scrum .....</i>	<i>23</i>
3.1.3 <i>Definición de roles del proyecto.....</i>	<i>24</i>
3.1.4 <i>Pila del Producto.....</i>	<i>25</i>
3.1.5 <i>Definición de los Sprint .....</i>	<i>26</i>
3.1.6 <i>Sprint 1 .....</i>	<i>28</i>
3.1.6.1 <i>Planificación Sprint 1 .....</i>	<i>29</i>
3.1.6.2 <i>Revisión 1.....</i>	<i>29</i>
3.1.6.3 <i>Revisión 2.....</i>	<i>31</i>
3.1.7 <i>Sprint 2 .....</i>	<i>32</i>
3.1.7.1 <i>Planificación Sprint 2 .....</i>	<i>33</i>
3.1.7.2 <i>Revisión 1.....</i>	<i>34</i>
3.1.7.3 <i>Revisión 2.....</i>	<i>36</i>
3.1.8 <i>Sprint 3 .....</i>	<i>37</i>
3.1.8.1 <i>Planificación Sprint 3 .....</i>	<i>38</i>
3.1.8.2 <i>Revisión 1.....</i>	<i>39</i>

3.1.8.3	Revisión 2.....	40
3.2	Arquitectura del Sistema.....	42
3.2.1	Diagrama de Clases .....	43
3.2.2	Descripción de Clases y métodos significativos .....	43
3.2.3	Diagrama de Componentes.....	45
3.2.4	Diagrama de Secuencia .....	45
3.2.5	Diagrama de Despliegue.....	46
3.2.6	Bases de Datos.....	47
3.2.7	Patrones de Diseño.....	47
3.3	Pruebas de Software .....	48
3.3.1	Pruebas de Integración .....	49
3.3.1.1	Añadir un nuevo servicio .....	50
3.3.1.2	Editar o actualizar un servicio.....	51
3.3.1.3	Obtener un servicio dado su “id” .....	51
3.3.1.4	Obtener todos los servicios .....	52
3.3.1.5	Eliminar un servicio.....	53
3.3.1.6	Servicio Web de prueba .....	53
3.4	Conclusiones parciales.....	54
<b>CONCLUSIONES.....</b>		<b>55</b>
<b>RECOMENDACIONES.....</b>		<b>56</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>57</b>
<b>ANEXOS .....</b>		<b>58</b>

## Lista de Figuras

Ilustración 1: Casos de Uso del Sistema.....	19
Ilustración 2: Diagrama de Clases .....	43
Ilustración 3: Diagrama de Componentes .....	45
Ilustración 4: Diagrama de Secuencia .....	46
Ilustración 5: Diagrama de Despliegue .....	46
Ilustración 6: Colección de Postman .....	50
Ilustración 7: Imagen de la prueba de Añadir un nuevo Servicio .....	50
Ilustración 8: Imagen de la prueba de Modificar un servicio existente.....	51
Ilustración 9: Imagen de prueba de Obtener un Servicio .....	52
Ilustración 10: Imagen de prueba de obtener todos los Servicios .....	52
Ilustración 11: Imagen de prueba de Eliminar un Servicio .....	53
Ilustración 12: Imagen de respuesta del servicio Web de prueba.....	54

**Lista de Tablas**

Tabla 2 Definición de Roles .....	24
Tabla 3 Pila del Producto .....	25
Tabla 4 Tiempo del Sprint.....	27
Tabla 5 Pila del Sprint 1 .....	28
Tabla 6 Planificación del Sprint 1 .....	29
Tabla 7 Revisión 1 Sprint 1 .....	30
Tabla 8 Revisión 2 Sprint 1 .....	31
Tabla 9 Pila del Sprint 2 .....	32
Tabla 10 Planificación Sprint 2 .....	33
Tabla 11 Revisión 1 Sprint 2 .....	34
Tabla 12 Revisión 2 Sprint 2 .....	36
Tabla 13 Pila del Sprint 3 .....	37
Tabla 14 Planificación Sprint 3 .....	38
Tabla 15 Revisión 1 Sprint 3 .....	39
Tabla 16 Revisión 2 Sprint 3 .....	41

## INTRODUCCIÓN

Desoft es una empresa de aplicaciones informáticas integrada al Grupo Empresarial de la Informática y las Comunicaciones (GEIC), perteneciente al Ministerio de Comunicaciones (MINCOM), considerada líder en la prestación de Software como Servicio (SaaS) en el país.

Desoft avanza a la par del desarrollo de las nuevas y modernas tecnologías buscando una concordancia entre los productos que ofrece y la actualidad tecnológica. Tiene como principal objetivo la satisfacción del cliente en la informatización de sus procesos internos y el cumplimiento de sus metas. Como parte de los productos que oferta esta empresa de tecnologías, encontramos el software **BIMAS Smifor**.

BIMAS Smifor está concebido como un instrumento para el intercambio de información, actualmente es una aplicación de escritorio funcional como herramienta Extract Transform and Load (siglas en inglés, ETL), que cuenta con disímiles tipos de fuentes de datos desde las cuales se extrae la información que será enviada a distintos destinos.

Muchos clientes manifiestan cierto grado de desconfianza a la hora de permitir un acceso directo a sus bases de datos como fuente principal utilizada por el software en cuestión, lo que hace necesario encontrar alternativas novedosas que permitan tanto el correcto funcionamiento del programa como la satisfacción de los clientes.

Es necesario implementar cambios en el software que faciliten su uso y actualizarlo, lograr que esté acorde a las exigencias actuales, ajustándolo de manera tal que garantice el intercambio de información de una manera segura y eficaz. Además, se considera necesario migrar el programa a la plataforma web para lograr insertarse como módulo dentro de una plataforma web de análisis de datos que es la nueva versión de BIMAS que se proyecta.

BIMAS se concibe como una solución para la Gestión del Desempeño Empresarial, que facilita la gestión de objetivos estratégicos y su seguimiento a través de indicadores de desempeño.

BIMAS Smifor es concebido como un instrumento para el intercambio de información de forma tal que se facilite el monitoreo y control entre los niveles Gerenciales Alto y Medio, Tácticos y Operativos entre organizaciones en esquemas jerárquicos para lograr una visión general\detallada oportuna, de lo que sucede en toda la organización.

Teniendo en cuenta las consideraciones descritas anteriormente se plantea el siguiente Problema de Investigación:

### **Problema de Investigación**

La necesidad de realizar cambios al software BIMAS Smifor en aras de adaptarlo a las necesidades y tecnologías requeridas para su uso en la nueva plataforma de análisis de datos, que se concibe como evolución del actual sistema BIMAS.

A partir del problema expuesto se plantea el Objetivo General al que se dará solución:

### **Objetivo general**

Lograr un conjunto de mejoras en funcionalidades del producto BIMAS Smifor, de la cartera de Desoft, para la extracción, transformación y carga de datos hacia almacenes.

Para contribuir con el desarrollo del propósito general se desglosa en los siguientes Objetivos Específicos:

### **Objetivos específicos**

1. Caracterizar diferentes herramientas existentes para la extracción, transformación y carga de datos asociadas a los sistemas de inteligencia de negocios y tratamiento de grandes volúmenes de información (“Big Data”).
2. Definir el alcance de la versión 3.0 de Bimas Smifor desarrollando los requisitos de esta versión en términos de especificaciones formales, identificando los cambios que resulta necesario aplicar al software.
3. Implementar dicha versión 3.0 sometiéndola a pruebas de verificación a escala de laboratorio.

### **Preguntas de investigación**

- ¿Qué características poseen las herramientas para la extracción, transformación y carga de datos en la actualidad?
- ¿Qué requisitos debe cumplir la nueva versión del sistema Bimas Smifor?
- ¿La nueva versión desarrollada cumple con lo previsto y presenta las características necesarias?

### **Hipótesis**

**H1:** La utilización de servicios web como fuente de datos en el sistema ETL a desarrollar contribuirá a aumentar la calidad del software BIMAS Smifor para así satisfacer las necesidades actuales de la empresa y sus clientes.

**H2:** La migración del sistema hacia un entorno web permitirá una correcta usabilidad de la aplicación y un diseño acorde a sus características.

**H3:** El empleo de una base de datos noSQL garantizará el correcto almacenamiento y la seguridad de los datos a utilizar por la plataforma de análisis de datos que se concibe como evolución de BIMAS.

### **Justificación**

Se lleva a cabo la presente investigación con el propósito de realizar cambios en el software BIMAS Smifor como aplicación de la empresa Desoft para su utilización y aplicación en la actualidad, atendiendo a las necesidades de dicha empresa y sus clientes y a las tecnologías requeridas. Smifor se concibe como el módulo de integración de datos de la herramienta de inteligencia de negocios BIMAS, ambos productos pertenecientes a la empresa Desoft. La necesidad de llevar a cabo el presente proyecto radica en que dicho proceso de integración de datos es la base de la inteligencia de negocios, por lo tanto, sin este resultaría imposible realizar cálculo de indicadores u otros procesos que pertenecen a la inteligencia de negocios.





# Marco Teórico. Herramientas ETL asociadas a Inteligencia de negocios

## **1. MARCO TEÓRICO. HERRAMIENTAS ETL ASOCIADAS A SISTEMAS DE INTELIGENCIA DE NEGOCIOS**

El presente capítulo describe los principales conceptos que se manejan en el desarrollo del trabajo como son la inteligencia de negocios y las operaciones de extracción, transformación y carga. Se mencionan ejemplos del uso de las mismas en la actualidad y sus características, y se abordan elementos de las estrategias y tecnologías empleadas para dar solución a la problemática planteada.

### ***1.1 Inteligencia de negocios***

El término inteligencia de negocios se refiere a un conjunto de tecnologías, herramientas y prácticas que tienen como objetivo la recolección, integración, análisis y presentación de grandes volúmenes de datos para favorecer la toma de decisiones.

Actualmente la arquitectura de la mayoría de sistemas de inteligencia de negocios consiste en almacenes de datos que sirven a una variedad de herramientas front-end para el análisis y reporte. En el back-end de la arquitectura se encuentran herramientas de integración de datos para poblar el almacén, que extraen la información desde fuentes operacionales que usualmente presentan una estructura heterogénea, realizan un proceso de limpieza o transformación de los datos y luego los cargan o insertan en el almacén.

Con el paso del tiempo la inteligencia de negocios ha demostrado su utilidad a la vez que ha presentado nuevos desafíos para los desarrolladores de este tipo de sistemas, ya que surgen necesidades como la disponibilidad de estos sistemas en todo momento, la inmediatez o una velocidad muy elevada en el flujo de integración de los datos e incluir nuevas fuentes que proporcionan información. Todo esto para mejorar el proceso de toma de decisiones en tiempo real.<sup>1</sup>

### ***1.2 Herramientas ETL***

Las siglas en inglés “ETL” corresponden a la definición *Extract, Transform and Load*, que son aquellas que se encargan de extraer datos desde variados tipos de fuentes, transformarlos y/o

---

<sup>1</sup> Data Integration Flows for Business Intelligence (2018)

limpiarlos y luego cargarlos en un sistema destino. Su función es la que completa la cohesión entre aplicaciones y sistemas. Son frecuentemente empleados para realizar integración con sistemas heredados.

Existen variaciones conceptuales relativas al proceso de ETL, de las que depende el rendimiento del proceso de manejo de los datos. Tal es el caso de los procesos ELT y ETLT, donde cambia el momento en que se realiza la transformación o limpieza de los datos. En los procesos ELT estas transformaciones ocurren luego de que los datos hayan sido cargados en el destino y en ETLT se realiza una primera limpieza de los datos luego de que estos han sido extraídos desde las fuentes y luego de cargados se realizan transformaciones más profundas.<sup>2</sup>

### ***1.3 Algunos sistemas de Inteligencia de negocios y sus herramientas ETL***

A continuación, se mencionan y caracterizan algunos de los sistemas de Inteligencia de Negocios más populares o usados y las herramientas de extracción, transformación y carga asociadas a estos.

#### **1.3.1 Oracle BI**

La Suite de Oracle Business Intelligence es una solución completa abierta e integrada para las necesidades de gestión empresarial como análisis de posibilidades, paneles de control, cuadros de mando, informes, consultas ad hoc y procesamiento OLAP.

Es un conjunto completo de infraestructuras y herramientas de inteligencia empresarial, incluido un servidor de análisis y consultas con alta escalabilidad y eficiencia, alertas e inteligencia proactivas, y un motor de generación de informes empresariales.

Las capacidades principales de Oracle BI son la recopilación de datos actualizados de la organización, la presentación de los mismos en formatos de fácil comprensión y su entrega de manera oportuna a los empleados de la organización.<sup>3</sup>

La herramienta Oracle Warehouse Builder es la encargada del ciclo de diseño y despliegue de los Data Warehouse y ofrece facilidades para la creación de operaciones ETL de manera visual con una generación de código del 95%.<sup>4</sup>

---

<sup>2</sup> Power Data Especialistas en Gestión de Datos (2019)

<sup>3</sup> Guía del Usuario de Oracle Business Intelligence Enterprise (2017)

<sup>4</sup> Introduction to Oracle Business Intelligence Standard Edition On (2018)

La Suite de Oracle Business Intelligence tiene altos costos de licencia.

### **1.3.2 Pentaho BI<sup>5</sup>**

Pentaho es una solución de BI implementada bajo los principios de Open Source para la gestión y toma de decisiones empresariales. Está compuesta de diferentes programas para la gestión y análisis de la información, que incluyen entorno Web, herramientas ETL, análisis multidimensional (OLAP), creación de cuadros de mando, minería de datos y generación de informes. Una de las principales ventajas de esta plataforma es que cuenta con una comunidad basada en el desarrollo, por lo que se realizan mejoras constantes y extensiones en la plataforma.

Pentaho Data Integration, también llamado Kettle o Spoon, es el módulo que ofrece la posibilidad de realizar operaciones de tipo ETL en Pentaho dentro de sus herramientas de diseño, y es también uno de sus puntos fuertes por los cuales muchos usuarios suelen elegir esta plataforma. Ofrece funciones como la migración de datos entre aplicaciones o bases de datos, la exportación de la información hacia archivos planos, cargas masivas de datos, limpieza de datos e integración de aplicaciones.

Pentaho Big Data es otra herramienta de integración de datos basada en Data Integration que ofrece la posibilidad de realizar trabajos de ETL dentro y fuera del entorno Big Data.

Pentaho, ahora mismo, es la gran alternativa a los sistemas de Business Intelligence tradicionales como Oracle, Business Objects, Cognos, Qlikview o Microstrategy; puesto que su precio es muchísimo más reducido.<sup>6</sup>

### **1.3.3 SQL Server Management Studio<sup>7</sup>**

SQL Server es un sistema gestor de bases de datos relacionales desarrollado por Microsoft y está construido para las funciones básicas de almacenar datos y extraerlos de acuerdo a las necesidades de otras aplicaciones. En su edición BI contiene todas las características de la versión estándar y añade las características de inteligencia de negocios.

---

<sup>5</sup> Business Intelligence en el ámbito Académico (2015)

<sup>6</sup> Pentaho software líder de Inteligencia de Negocio de código abierto (2020)

<sup>7</sup> MS SQL Server Tutorial (2020)

SQL Server Management Studio es una estación de trabajo o herramienta para el cliente que permite manejar SQL Server desde una interfaz gráfica en lugar de usar la línea de comandos.

Contiene una amplia variedad de servicios entre los que se encuentra Integration Services, usado para llevar a cabo tareas de extracción, transformación y carga de datos.

#### 1.3.4 Spago BI<sup>8</sup>

Es una plataforma de integración de inteligencia de negocios desarrollada con la filosofía de software libre. Esta cumple con los requisitos de inteligencia de negocios como la gestión de datos, capacidad de análisis, seguridad y administración. Ofrece soluciones para la generación de informes, minería de datos, análisis multidimensional (OLAP) y análisis de datos geográficos. También cuenta con herramientas para la extracción, transformación y carga de datos.

Entre los principales componentes de Spago BI se encuentran:

Reporting: Permite realizar informes como listas, tablas y graficos y exportarlos a una amplia variedad de formatos. Integra las herramientas BIRT, JasperReports y BusinessObjects.

OLAP: Permite realizar análisis multidimensionales OLAP a través de operaciones de tipo drill-down, drill-across, slice-and-dice y drill-through.

Chart: Es un módulo grafico basado en JFreeChart que permite desarrollar widgets gráficos con el objetivo de usarlos en informes tales como histogramas y gráficos de pastel.

Dashboard: Ofrece opciones de visualización grafica que permiten mostrar los Key Performance Indicators (KPI) para visualizaciones de rendimiento en tiempo real.

Geo/Sig: Ofrece utilidades geográficas que permiten establecer una conexión entre los datos geográficos y los empresariales.

Data Mining: Permite el análisis de datos avanzados mediante el proceso de minería de datos. Integra Weka como herramienta de código abierto de minería de datos.

ETL: Spago BI ETL permite extraer los datos desde varios tipos de fuentes y formas según la lógica de ETL. Integra el producto de código abierto Talend Open Studio (TOS).

---

<sup>8</sup>La plataforma SpagoBI Open Source Business Intelligence (2020)

## **1.4 Entorno de desarrollo**

### **1.4.1 Spring Framework y Springboot**

Spring Framework es un framework Open Source que facilita el desarrollo de aplicaciones en Java, Kotlin y Groovy. Es ampliamente utilizado en la actualidad, pero las tareas de configuración inicial y preparación de las aplicaciones para producción son bastante tediosas, ahí es donde radica la importancia de Springboot.

Spring Boot permite compilar aplicaciones Web como un archivo .jar que se puede ejecutar como una aplicación Java normal (como alternativa a un archivo .war, que se despliega en un servidor de aplicaciones como Tomcat).

Springboot facilita la creación de starters, que brindan la posibilidad de añadir dependencias tanto propias de Spring como de terceros, minimizando así la necesidad de configuración.

### **1.4.2 Servicios Web**

El término “servicios web” designa una tecnología que permite que las aplicaciones puedan comunicarse de una forma que no depende de la plataforma ni del lenguaje de programación. Un servicio web es un componente software que describe un conjunto de operaciones a las cuales se puede acceder por la red a través de mensajería XML estandarizada, mediante documentos WSDL, haciendo uso de protocolos basados en el lenguaje XML con el objetivo de describir una operación para ejecutar o datos para intercambiar con otro servicio web. Un grupo de servicios web que interactúa de esa forma define la aplicación de un servicio web específico en una arquitectura orientada a servicios (SOA).

### **1.4.3 JSON<sup>9</sup>**

Con la creciente popularidad de los servicios Web, XML se ha convertido prácticamente de facto en el estándar para transmisión de datos. Pero se necesita transmitir a través de Internet muchos más bytes de información para realizar una tarea que se podría llevar a cabo con un flujo de información mucho más pequeño.

---

<sup>9</sup> Curso Librerías Web 2.0 JSON (2008)

Así se han desarrollado nuevas formas de compresión XML e, incluso, nuevos formatos XML completos, tales como Binary XML (XML binario). Todas estas soluciones funcionan ampliando o añadiéndose a XML, conviniendo los aspectos de compatibilidad descendente en un asunto a tener en cuenta. Douglas Crockford, un experimentado ingeniero software, propuso un nuevo formato de datos construido sobre JavaScript llamado JSON, JavaScript Object Notation (notación de objetos JavaScript).

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML.

#### **1.4.4 Bases de datos no SQL**

Las bases de datos no SQL son estructuras que permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas, debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales, donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias. Además de lo comentado anteriormente, las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos, sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos.

Esta forma de almacenar la información ofrece ciertas ventajas sobre los modelos relacionales. Entre las ventajas más significativas podemos destacar:

- Se ejecutan en máquinas con pocos recursos: Estos sistemas, a diferencia de los sistemas basados en SQL, no requieren de apenas computación, por lo que se pueden montar en máquinas de un coste más reducido.
- Escalabilidad horizontal: Para mejorar el rendimiento de estos sistemas simplemente se consigue añadiendo más nodos, con la única operación de indicar al sistema cuáles son los nodos que están disponibles.

- Pueden manejar gran cantidad de datos: Esto es debido a que utiliza una estructura distribuida, en muchos casos mediante tablas Hash.
- No genera cuellos de botella: El principal problema de los sistemas SQL es que necesitan transcribir cada sentencia para poder ser ejecutada, y cada sentencia compleja requiere además de un nivel de ejecución aún más complejo, lo que constituye un punto de entrada en común, que ante muchas peticiones puede ralentizar el sistema.

Algunas de las diferencias más destacables que nos podemos encontrar entre los sistemas NoSQL y los sistemas SQL están:

- No utilizan SQL como lenguaje de consultas. La mayoría de las bases de datos NoSQL evitan utilizar este tipo de lenguaje o lo utilizan como un lenguaje de apoyo. Por poner algunos ejemplos, Cassandra utiliza el lenguaje CQL, MongoDB utiliza JSON y BigTable hace uso de GQL.
- No utilizan estructuras fijas como tablas para el almacenamiento de los datos. Permiten hacer uso de otros tipos de modelos de almacenamiento de información como sistemas de clave-valor, objetos o grafos.
- No suelen permitir operaciones JOIN. Al disponer de un volumen de datos tan extremadamente grande suele resultar deseable evitar los JOIN. Esto se debe a que, cuando la operación no es la búsqueda de una clave, la sobrecarga puede llegar a ser muy costosa. Las soluciones más directas consisten en desnormalizar los datos, o bien realizar el JOIN mediante software, en la capa de aplicación.
- La arquitectura es distribuida. Las bases de datos relacionales suelen estar centralizadas en una única máquina o bien en una estructura máster-esclavo, sin embargo, en los casos NoSQL la información puede estar compartida en varias máquinas mediante mecanismos de tablas Hash distribuidas.

#### **1.4.5 MongoDB**

Se trata de una base de datos del tipo orientada a documentos, de esquema libre, es decir, que cada entrada puede tener un esquema de datos diferente que nada tenga que ver con el resto de registros



almacenados. Es bastante rápido a la hora de ejecutar sus operaciones ya que está escrito en lenguaje C++.

Para el almacenamiento de la información, utiliza un sistema propio de documento conocido con el nombre BSON, que es una evolución del conocido JSON pero con la peculiaridad de que puede almacenar datos binarios. MongoDB se ha convertido en una de las bases de datos NoSQL favoritas por los desarrolladores.

### ***1.5 Conclusiones Parciales***

Se han caracterizado algunos de los principales sistemas de inteligencia de negocios y las herramientas de extracción, transformación y carga de datos asociadas a estos. A partir del estudio de las tecnologías, herramientas y lenguajes empleados para el desarrollo del sistema, se decide utilizar Spring como framework y Java como lenguaje de programación, así como se determina usar MongoDB como gestor de bases de datos.



## 2. BIMAS SMIFOR: CARACTERÍSTICAS ACTUALES Y PERSPECTIVAS DESEADAS

En este capítulo se estudian las características del software en cuestión para definir los cambios que se deben aplicar, definiendo el alcance de la versión a desarrollar. Como parte crucial del proceso de aplicar cambios a la aplicación se definen requisitos funcionales y no funcionales y se abordan elementos de mantenimiento de software.

### 2.1 Bimas Smifor 2.0

BIMAS Smifor es un sistema encargado de facilitar el intercambio de información, actualmente desplegado como aplicación de escritorio, cuyo concepto principal son los *Servicios*:

Un BSS o BIMAS Smifor Service no es más que un conjunto de propiedades tales como:

- Fuente de datos: Sistema o archivo del cual se extrae información con el objetivo de transformarla y cargarla en un destino.
- Un formato óptimo para el archivo en el cual se almacenan los datos que varía según el destino.
- Destino: Es un sistema o simplemente una ruta a la que se desea enviar la información extraída

Las fuentes de datos permitidas en Smifor 2.0 son: DBF, MS Access, SQL Server, PostgreSQL, MySQL y SQLite. Esta variedad de fuentes garantiza una vía de extracción de la información de los clientes, pero estos reflejan desconfianza ya que otorgar permisos de acceso físico a las bases de datos no es seguro. Por ello se necesita encontrar un tipo de fuente de datos capaz de contener solo la información que el cliente desea que Smifor maneje y cargue en el destino. Aunque más adelante se desee incorporar más funcionalidades, el presente trabajo se centra en BIMAS Server como destino de datos, ya que es el de mayor importancia tanto para la empresa como sus clientes. Además, una aplicación de escritorio presenta ciertas desventajas como que es el cliente el encargado de monitorear la ejecución de los BSS con la información de los servicios del Smifor instalado localmente, exigiendo que ellos cambien dichos servicios cuando se requiera, sin embargo, sería mucho más adecuado que los trabajadores que diseñen los servicios pudieran crearlos en un sistema capaz de almacenarlos y desde el cual sea posible facilitar su modificación o eliminación, de ser necesarias. Por ello es importante migrar el sistema hacia la plataforma web,

logrando un producto más fácil de usar y que se pueda seguir desarrollando, con tecnologías modernas y pensando siempre en la escalabilidad.

Todo lo anterior explica la necesidad de realizar mantenimiento al sistema BIMAS Smifor en aras de adecuarlo a las necesidades actuales.

## ***2.2 Versión a desarrollar***

### **Requisitos funcionales del sistema:**

Según lo expuesto se pueden identificar requisitos funcionales tales como:

RF1. Añadir un nuevo Servicio.

RF2 Modificar un servicio en específico

RF3 Obtener un servicio dado su id.

RF4 Listar todos los servicios.

RF5. Eliminar un servicio dado su id.

RF6. Diseñar un servicio web que retorne objetos de ejemplo con el objetivo de realizar pruebas a la aplicación, demostrando que se realiza de manera correcta la extracción de datos desde un servicio web.

### **Requisitos no funcionales del sistema:**

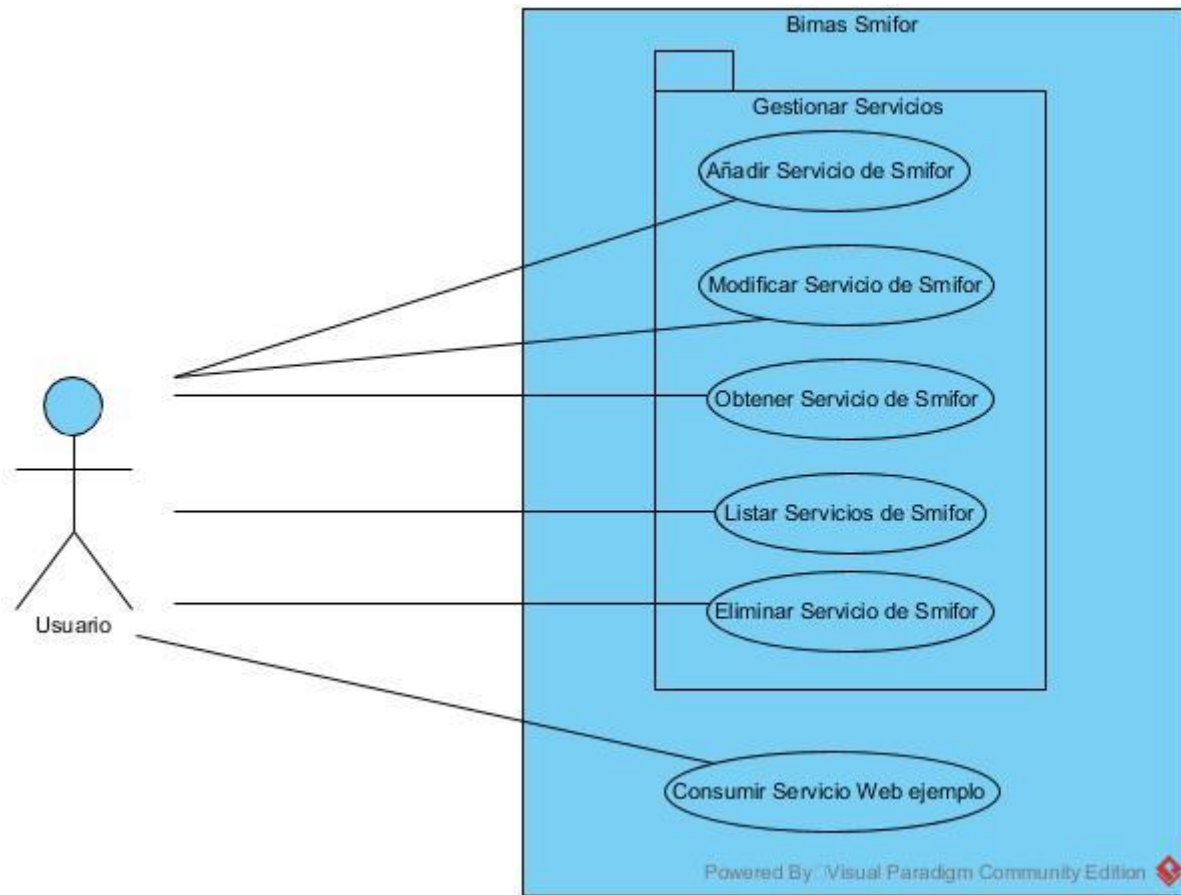
RNF1. Version 11 del jdk.

RNF2. Tomcat versión 11.

RNF3. MongoDB versión 3.2.

Esto se debe a que son las versiones disponibles en la infraestructura de los servidores de los que se dispone para el despliegue y explotación de las aplicaciones de Desoft en la nube.

### 2.2.1 Casos de Uso del Sistema



**Ilustración 1: Casos de Uso del Sistema**

En la imagen se pueden apreciar los casos de uso del sistema implementado, donde cada uno corresponde a un end point de la API.

### 2.3 *Mantenimiento de Software*

Es el proceso de modificación que se aplica a un producto de software después de su entrega. Las tareas de mantenimiento son las últimas que se realizan en el ciclo de vida del software. Se encarga de corregir defectos, mejorar el rendimiento o adaptarlo a un cambio de entorno.

Su objetivo fundamental es preservar la funcionalidad y operabilidad, a la par que aumentar la vida útil del software.

La necesidad de mantenimiento se establece para corregir fallas, mejorar el diseño, implementar mejoras, definir interfaces con otros sistemas, migrar software o adaptarlo a otros tipos de hardware, capacidad de las redes de telecomunicaciones o características del sistema.

### 2.3.1 Tipos de mantenimiento

#### 2.3.1.1 *Mantenimiento Correctivo*

Tiene por objetivo localizar y eliminar los posibles defectos de un programa dados por una característica del sistema con el potencial de causar un fallo. Estos ocurren cuando el comportamiento del sistema es diferente del esperado. Pueden darse fallos de procesamiento como las salidas incorrectas, de rendimiento como un alto tiempo de espera, de programación como los errores en el diseño o de documentación como las inconsistencias entre una funcionalidad y el manual de usuario de la aplicación.

#### 2.3.1.2 *Mantenimiento Perfectivo*

Normalmente debido a cambios en los requerimientos del producto, este tipo de mantenimiento se define como un conjunto de actividades que se realizan para mejorar o añadir nuevas funcionalidades al producto de software.

Según algunos autores se puede dividir en:

- Mantenimiento de Ampliación: Orientado a la inclusión o desarrollo de nuevas funcionalidades.
- Mantenimiento de Eficiencia: Orientado a lograr una mejoría en la eficiencia de la ejecución.

#### 2.3.1.3 *Mantenimiento Adaptativo*

Consiste en la modificación de un software promovida por cambios en el entorno en el cual se ejecuta, tales como el cambio de sistema operativo o marco de trabajo, la infraestructura informática o simplemente el uso de nuevas herramientas. La envergadura de estas modificaciones puede variar desde la sustitución o cambio de una línea de código hasta la reescritura de la totalidad del programa.

Los cambios que promueven este tipo de mantenimiento pueden darse en el entorno de los datos o de los procesos:

En el entorno de los datos cuando cambia la forma en la cual es necesario almacenar la información de interés para el cliente, y en el entorno de los procesos cuando se necesita que el software funcione en una nueva plataforma, sistema operativo o versión de estos, o simplemente existen nuevas necesidades que el software no satisface.

#### ***2.3.1.4 Mantenimiento a Bimas Smifor***

Luego de analizar las características y definiciones de los tipos de mantenimiento de software se identifica al proceso de migrar al sistema Bimas Smifor a la plataforma Web, así como añadirle nuevas funcionalidades como una combinación de mantenimiento adaptativo y perfectivo, donde a la par que se adapta al sistema a nuevas condiciones de hardware y software se desarrollan nuevas funcionalidades con las que se espera satisfacer los requisitos actuales.

### ***2.4 Conclusiones parciales***

A partir del análisis realizado se identifican los cambios que se deben aplicar al software, los casos de uso del sistema y se elaboró el diagrama correspondiente a los mismos. También se definen los requisitos funcionales y no funcionales de la aplicación y se abordan elementos de mantenimiento de software identificando el tipo de mantenimiento que se realiza a Smifor 2.0.



# 3

Descripción de la propuesta de solución. Metodología ágil de desarrollo  
y pruebas de Software

### **3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN. METODOLOGÍA ÁGIL DE DESARROLLO Y PRUEBAS DE SOFTWARE**

En este capítulo se muestra cómo se llevó a cabo el desarrollo del proyecto utilizando la metodología ágil Scrum y sus artefactos, se describe la propuesta de solución y el proceso de pruebas al que fue sometida para su validación.

#### **3.1 Metodología Ágil**

El proyecto posee características como la necesidad de lograr incrementos del producto en intervalos de tiempo relativamente breves y el equipo compuesto de una cantidad pequeña de integrantes. Lo anterior fundamenta la decisión de enfrentar el proyecto utilizando los principios de metodología ágil Scrum y sus artefactos.

##### **3.1.1 Scrum<sup>10</sup>**

Es una de las metodologías ágiles de desarrollo más populares, iterativa, rápida, flexible y eficaz. Garantiza el esfuerzo y la responsabilidad colectiva, así como la continuidad del progreso. Posee una alta compatibilidad con los productos y el desarrollo de servicios en todo tipo de industrias o proyectos, con independencia de su complejidad.

##### **3.1.2 Artefactos de Scrum<sup>11</sup>**

A continuación, se describen los artefactos necesarios para el desarrollo del proyecto utilizando la metodología mencionada:

- Pila del producto: (en inglés product backlog) Consiste en una lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo. Refleja todos los requisitos, funcionalidades, mejoras y características a realizar para lograr el incremento.

---

<sup>10</sup> Una guía para el CONOCIMIENTO DE SCRUM (2013)

<sup>11</sup> Scrum Manager (2016)

- Pila del Sprint: (en inglés sprint backlog) Listado de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- Incremento: Es el resultado de cada sprint.

### 3.1.3 Definición de roles del proyecto

En la tabla se muestra la definición de los roles asociados a las personas que los asumen:

**Tabla 1 Definición de Roles**

Scrum Master	Dairo López Mollinedo
Product Owner	Ana M. García
Equipo	Daniel Rodríguez Fuentes

Las personas que asumen los roles tienen las responsabilidades siguientes:

#### **Scrum Master:**

- Planificar las actividades del proyecto.
- Cerciorarse de que el proyecto se desarrolla de acuerdo a las prácticas,

valores y reglas de Scrum.

- Gestionar la Lista de Producto de manera efectiva

#### **Product Owner:**

- Es responsable de la creación de la pila del producto, o sea, define cuales son las funcionalidades del proyecto en desarrollo.
- Planifica el inicio de los Sprint así como la revisión del producto al final de estos para garantizar que se desarrollan las funcionalidades previstas.

#### **Equipo:**

- Es responsable de cumplir con el desarrollo de las funcionalidades previstas en el tiempo acordado y de entregar un incremento del producto al final de cada Sprint.

### 3.1.4 Pila del Producto

En la tabla se muestra la pila del producto, que contiene todas las funcionalidades o requisitos que debe cumplir el producto al momento de su entrega, la tabla presenta la siguiente estructura:

- Por cada fila un requisito o funcionalidad del producto.
- Columna Id: Número único que identifica el requerimiento o funcionalidad.
- Columna Nombre: Nombre que se le da al requerimiento y se redacta en forma de caso de uso.
- Columna Descripción: Ofrece una descripción del requerimiento para esclarecer lo que desea el usuario.
- Columna Importancia: Número que en orden descendente refleja el orden en el que se enfrentará el desarrollo de los requisitos.

**Tabla 2 Pila del Producto**

<b>Id</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Importancia</b>
1	Gestionar Servicios de Smifor	Insertar, modificar, eliminar, y listar los servicios de Smifor	100
2	Lograr persistencia de los datos pertenecientes a los servicios	Almacenar los distintos servicios en una base de datos para lograr que sus datos persistan	99
3	Consumir Servicio Web para obtener la información del cliente	Consumir un servicio Web dada su URL para obtener la información que provee el cliente	95

4	Almacenar datos del cliente	Consiste en almacenar la información previamente obtenida en una colección de mongo del sistema BIMAS para que este pueda utilizar dichos datos	90
5	Diseñar la ejecución de los Servicios	Lograr que los servicios se ejecuten en un momento dado logrando la extracción de los datos del Servicio Web y su posterior almacenamiento en la colección como ya se ha descrito	85

### 3.1.5 Definición de los Sprint

En la tabla se muestra el tiempo de trabajo empleado por el equipo en cada Sprint

**Tabla 3 Tiempo del Sprint**

<b>Equipo Scrum</b>	<b>Jornada laboral</b>	<b>Horas de trabajo por día</b>	<b>Horas de trabajo por semana</b>	<b>Semanas de trabajo al mes</b>	<b>Total de horas laborables por mes</b>	<b>Total de días laborables por mes</b>
Ana M. García Dairo López Daniel Rodríguez Fuentes	8 horas	5 horas	25 horas	4 semanas	100 horas	20 días

En el desarrollo del proyecto se espera que surjan impedimentos, obstáculos o distracciones que se contemplan en las estimaciones realizadas. Se establece así un factor de dedicación del 80%.

La velocidad estimada para el desarrollo de cada Sprint queda calculada de la siguiente manera:

$$16 = 20 \times 80\%$$

$$\text{Puntos de Historia} = \text{Días Hombre disponibles} * \text{Factor de Dedicación}$$

El desarrollo del proyecto se organiza en 3 Sprint, teniendo en cuenta la velocidad estimada para cada uno y la importancia asignada a cada historia de usuario. Se planifican revisiones al final de cada Sprint y se establece una frecuencia semanal para las reuniones al estar constituido el Equipo de desarrollo de una sola persona.

A continuación, se muestra el trabajo realizado en cada uno de los Sprint, tomando de la pila del producto las tareas de acuerdo con su importancia, organizándolo por semanas numeradas secuencialmente desde la 1 hasta las necesarias para la culminación del proyecto.

### 3.1.6 Sprint 1

Este Sprint se planifica en función de garantizar un correcto diseño del modelado correspondiente a los servicios de Smifor y su persistencia dentro del sistema, aspectos base y fundamentales para el correcto funcionamiento del software.

Fecha de Inicio: 1/7/2021

Fecha Final: 22/7/2021

**Tabla 4 Pila del Sprint 1**

<b>Id</b>	<b>Funcionalidad</b>	<b>Prioridad</b>	<b>Estimación de tiempo (días)</b>
1	Diseñar modelo Servicios de Smifor	100	4
2	Insertar Servicio de Smifor	98	2
3	Modificar Servicio de Smifor	97	2
4	Eliminar Servicio de Smifor	95	2
5	Listar Servicios de Smifor	94	2
6	Instalar y configurar el Gestor de Bases de Datos (Mongodb)	92	1
7	Diseñar e implementar la persistencia de la información en la base de datos	90	3

Total de días del Sprint: 16 días

### 3.1.6.1 Planificación Sprint 1

En la tabla siguiente se muestra la planificación del Sprint con las fechas establecidas para las revisiones

**Tabla 5 Planificación del Sprint 1**

Fecha de Inicio	1/7/2021
Fecha Final	22/7/2021
Revisión de los avances	Se realizarán revisiones cada 8 días laborables ya que los Sprint están planificados para 16 días Fecha revisión 1: 12/7/2021 Fecha revisión 2: 22/7/2021
Tareas a desarrollar	<ul style="list-style-type: none"> <li>➤ Diseñar modelo Servicios de Smifor</li> <li>➤ Insertar Servicio de Smifor</li> <li>➤ Modificar Servicio de Smifor</li> <li>➤ Eliminar Servicio de Smifor</li> <li>➤ Listar Servicios de Smifor</li> <li>➤ Instalar y configurar el Gestor de Bases de Datos(Mongodb)</li> <li>➤ Diseñar e implementar la persistencia de la información en la base de datos</li> </ul>

### 3.1.6.2 Revisión 1



Al momento de la revisión 1 perteneciente al Sprint actual, la pila del Sprint se encuentra de la siguiente manera:

**Tabla 6 Revisión 1 Sprint 1**

Funcionalidad	Terminada	En curso	Pendiente
Diseñar modelo Servicios de Smifor	X		
Insertar Servicio de Smifor	X		
Modificar Servicio de Smifor	X		
Eliminar Servicio de Smifor			X
Listar Servicios de Smifor			X
Instalar y configurar el Gestor de Bases de Datos(Mongodb)			X
Diseñar e implementar la persistencia de la información en la base de datos			X

Se puede observar como al momento de la revisión 1 quedan terminadas algunas funcionalidades y otras quedan pendientes.

### 3.1.6.3 Revisión 2

Al momento de la revisión 2 perteneciente al Sprint actual la pila del Sprint se encuentra de la siguiente manera:

**Tabla 7 Revisión 2 Sprint 1**

Funcionalidad	Terminada	En curso	Pendiente
Diseñar modelo Servicios de Smifor	X		
Insertar Servicio de Smifor	X		
Modificar Servicio de Smifor	X		
Eliminar Servicio de Smifor	X		
Listar Servicios de Smifor	X		
Instalar y configurar el Gestor de Bases de Datos (Mongodb)	X		
Diseñar e implementar la	X		

persistencia de la información en la base de datos			
--	--	--	--

Al momento de la segunda revisión se observa que quedan terminadas todas las funcionalidades pendientes en la revisión 1 cumpliendo así con la planificación del Sprint 1.

### 3.1.7 Sprint 2

Este Sprint se planifica con el objetivo de lograr que los servicios de Smifor sean capaces de consumir en el momento de su ejecución un Servicio Web publicado por el cliente para obtener información de este y almacenarla en una colección mongo específica en el sistema BIMAS.

**Tabla 8 Pila del Sprint 2**

<b>Id</b>	<b>Funcionalidad</b>	<b>Prioridad</b>	<b>Estimación de tiempo (días)</b>
8	Diseñar e implementar un método en los servicios de Smifor capaz de obtener información consumiendo un Servicio Web que previamente haya sido publicado por el cliente	100	7
9	Diseñar e implementar un Servicio Web en la API de Smifor que	98	3

	devuelva una serie de objetos de ejemplo con el fin de comprobar el correcto funcionamiento del proyecto		
10	Diseñar e implementar un método que cree una conexión con la base de datos mongodb de BIMAS e inserte la información obtenida del Servicio Web	97	6

Total de días del Sprint: 16

### 3.1.7.1 Planificación Sprint 2

En la tabla siguiente se muestra la planificación del Sprint con las fechas establecidas para las revisiones

**Tabla 9 Planificación Sprint 2**

Fecha de Inicio	23/7/2021
Fecha Final	13/8/2021
Revisión de los avances	Se realizarán revisiones cada 8 días ya que los Sprint están planificados para 16 días Fecha revisión 1: 3/8/2021 Fecha revisión 2: 13/8/2021

Tareas a desarrollar	<ul style="list-style-type: none"> <li>➤ Diseñar e implementar un método en los servicios de Smifor capaz de obtener información consumiendo un Servicio Web que previamente haya sido publicado por el cliente</li> <li>➤ Diseñar e implementar un Servicio Web en la API de Smifor que devuelva una serie de objetos de ejemplo con el fin de comprobar el correcto funcionamiento del proyecto</li> <li>➤ Diseñar e implementar un método que cree una conexión con la base de datos MongoDB de BIMAS e inserte la información obtenida del Servicio Web</li> </ul>
----------------------	--

### 3.1.7.2 Revisión 1

Al momento de la revisión 1 planificada para el Sprint actual la pila del Sprint se comporta de la siguiente manera:

**Tabla 10 Revisión 1 Sprint 2**

Funcionalidad	Terminada	En curso	Pendiente
Diseñar e implementar un método en los servicios de Smifor capaz de obtener	X		

información consumiendo un Servicio Web que previamente haya sido publicado por el cliente			
Diseñar e implementar un Servicio Web en la API de Smifor que devuelva una serie de objetos de ejemplo con el fin de comprobar el correcto funcionamiento del proyecto		X	
Diseñar e implementar un método que cree una conexión con la base de datos mongodb de BIMAS e inserte la información obtenida del Servicio Web			X

En la revisión 1 del presente Sprint se observa una tarea terminada, una en curso y una pendiente.

### 3.1.7.3 Revisión 2

Al momento de la revisión 2 la pila del Sprint se observa de la siguiente manera:

**Tabla 11 Revisión 2 Sprint 2**

Funcionalidad	Terminada	En curso	Pendiente
Diseñar e implementar un método en los servicios de Smifor capaz de obtener información consumiendo un Servicio Web que previamente haya sido publicado por el cliente	X		
Diseñar e implementar un Servicio Web en la API de Smifor que devuelva una serie de objetos de ejemplo con el fin de comprobar el correcto funcionamiento del proyecto	X		

Diseñar e implementar un método que cree una conexión con la base de datos mongodb de BIMAS e inserte la información obtenida del Servicio Web	X		
--	---	--	--

Es posible apreciar que tras la revisión 2 quedan terminadas todas las funcionalidades planificadas para el Sprint 2.

### 3.1.8 Sprint 3

Este Sprint se planifica con el objetivo de lograr que los servicios de Smifor se ejecuten de manera automática cada un intervalo de tiempo definido y realizar pruebas de integración a la API desarrollada.

**Tabla 12 Pila del Sprint 3**

<b>Id</b>	<b>Funcionalidad</b>	<b>Prioridad</b>	<b>Estimación de tiempo (días)</b>
11	Estudiar la forma de planificar la ejecución de métodos en específico que ofrece el framework utilizado para el desarrollo del proyecto	100	6



12	Implementar la funcionalidad de planificar la ejecución automática de los servicios de Smifor	98	4
13	Instalar el software Postman y estudiar su funcionamiento para su empleo a la hora de realizar pruebas de integración	97	2
14	Realizar pruebas de integración consumiendo los end point de la API desarrollada		4

Total de días del Sprint: 16

#### 3.1.8.1 Planificación Sprint 3

En la tabla siguiente se muestra la planificación del Sprint con las fechas establecidas para las revisiones

**Tabla 13 Planificación Sprint 3**

Fecha de Inicio	16/8/2021
Fecha Final	6/9/2021
Revisión de los avances	Se realizarán revisiones cada 8 días ya que los Sprint están planificados para 16 días

	<p>Fecha revisión 1: 25/8/2021</p> <p>Fecha revisión 2: 6/9/2021</p>
Tareas a desarrollar	<ul style="list-style-type: none"> <li>➤ Estudiar la forma de planificar la ejecución de métodos en específico que ofrece el framework utilizado para el desarrollo del proyecto</li> <li>➤ Implementar la funcionalidad de planificar la ejecución automática de los servicios de Smifor</li> <li>➤ Instalar el software Postman y estudiar su funcionamiento para su empleo a la hora de realizar pruebas de integración</li> <li>➤ Realizar pruebas de integración consumiendo los end point de la API desarrollada</li> </ul>

### 3.1.8.2 Revisión 1

En la tabla se muestra la pila del Sprint al momento de la revisión 1:

**Tabla 14 Revisión 1 Sprint 3**

Funcionalidad	Terminada	En curso	Pendiente
Estudiar la forma de planificar la ejecución de métodos en específico que ofrece el framework	X		

utilizado para el desarrollo del proyecto			
Implementar la funcionalidad de planificar la ejecución automática de los servicios de Smifor		X	
Instalar el software Postman y estudiar su funcionamiento para su empleo a la hora de realizar pruebas de integración			X
Realizar pruebas de integración consumiendo los end point de la API desarrollada			X

Se puede apreciar cómo se encuentra una tarea terminada, una en curso y dos pendientes

#### 3.1.8.3 Revisión 2

A continuación, se muestra cómo se comporta la pila del Sprint en el momento de la revisión 2:

**Tabla 15 Revisión 2 Sprint 3**

Funcionalidad	Terminada	En curso	Pendiente
Estudiar la forma de planificar la ejecución de métodos en específico que ofrece el framework utilizado para el desarrollo del proyecto	X		
Implementar la funcionalidad de planificar la ejecución automática de los servicios de Smifor	X		
Instalar el software Postman y estudiar su funcionamiento para su empleo a la hora de realizar pruebas de integración	X		
Realizar pruebas de integración consumiendo los	X		

end point de la API desarrollada			
-------------------------------------	--	--	--

Al momento de esta revisión se encuentran terminadas todas las funcionalidades planificadas para el Sprint.

### ***3.2 Arquitectura del Sistema***

En este epígrafe se reflejan los aspectos relacionados con la arquitectura del sistema, dando explicación a la solución desarrollada.

### 3.2.1 Diagrama de Clases

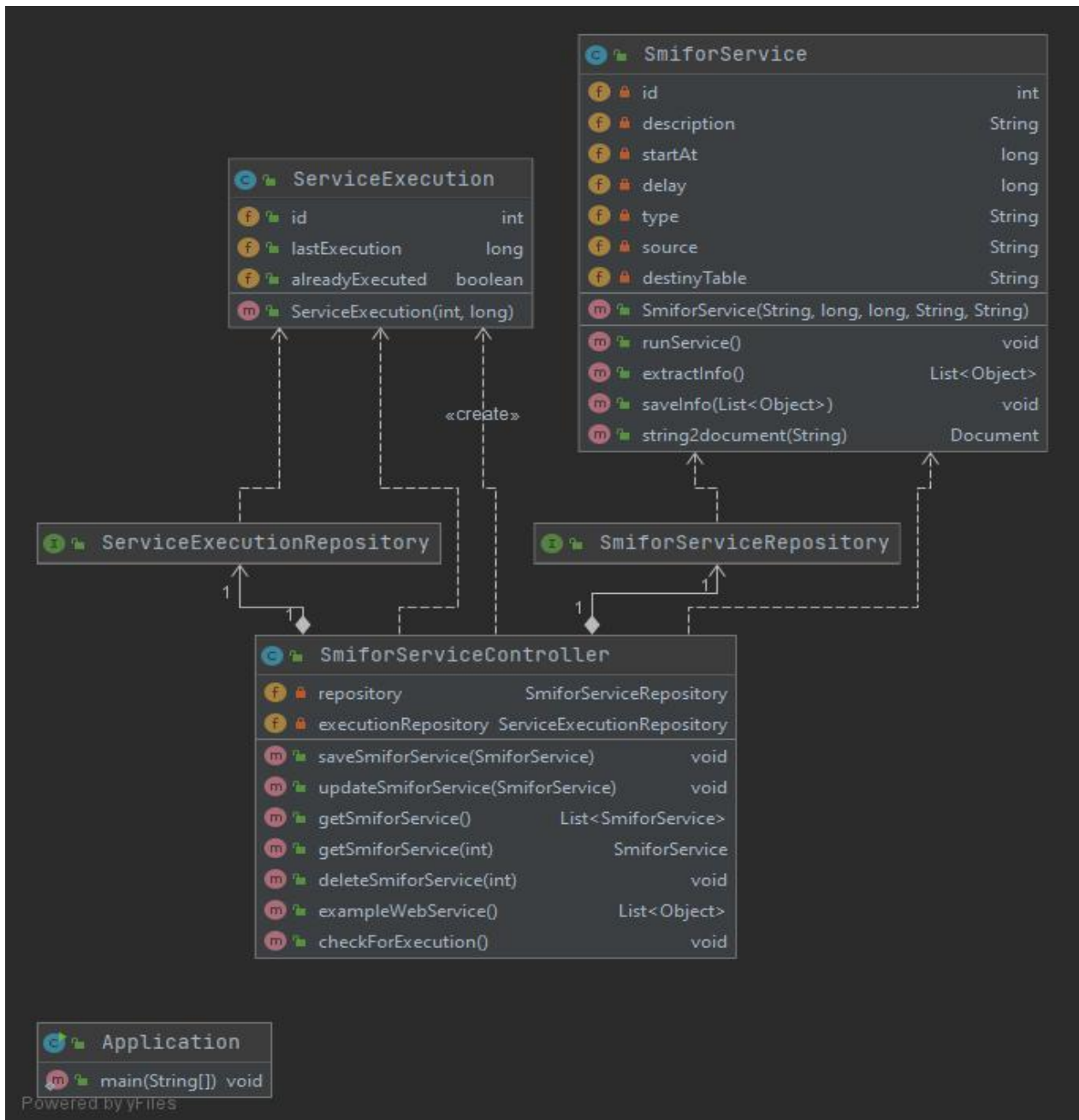


Ilustración 2: Diagrama de Clases

### 3.2.2 Descripción de Clases y métodos significativos

- **SmiforService:**

Es el modelo de los servicios de Smifor. Contiene los siguientes atributos:

**id:** Identificador único para cada servicio.

**description:** Una descripción textual para el servicio.

**startAt:** Representa el momento en el que se desea que el servicio se ejecute por primera vez.

**delay:** Representa la cantidad de tiempo que se desea que transcurra entre una y otra ejecución del servicio

**type:** Tipo de servicio de Smifor. Por el momento los servicios solo son de un tipo, pero se incluye en el modelo teniendo en cuenta que Smifor seguirá desarrollándose.

**source:** Fuente de la que se extraeran los datos (Hasta ahora la URL del Servicio Web publicado por el cliente).

**destinyTable:** Coleccion de BMAS en la cual se cargarán los datos obtenidos.

**Esta clase también contiene todos los métodos necesarios para que el servicio cumpla con su función, tales como extraer la información, insertarla en la colección pertinente y ejecutarse.**

- ServiceExecution:

Es el modelo de la ejecución de los servicios, contiene los siguientes atributos:

**Id:** Identificador único que representa el id de un servicio de Smifor

**lastExecution:** Momento en el que el servicio con el id correspondiente se ejecutó por última vez

**alreadyExecuted:** De tipo binario. Su función es representar si el servicio ya ha sido ejecutado por primera vez o no

- SmiforServiceRepository:

Interfaz que extiende de *MongoRepository*, es instanciada por SmiforServiceController para definir las operaciones que se realizaran sobre la base de datos que tengan que ver con los servicios de Smifor.

- ServiceExecutionRepository:

Interfaz que extiende de *MongoRepository*, es instanciada por *SmiforServiceController* para definir las operaciones que se realizarán sobre la base de datos que tengan que ver con las ejecuciones de servicios.

- *SmiforServiceController*:

Es el controlador principal de la aplicación. Contiene las instancias de los repositorios tanto de servicios como de sus ejecuciones. También define los distintos end point de la API y cómo se comportan.

### 3.2.3 Diagrama de Componentes

A continuación, se observa el Diagrama de componentes correspondiente al caso de uso Añadir Nuevo Servicio de Smifor.

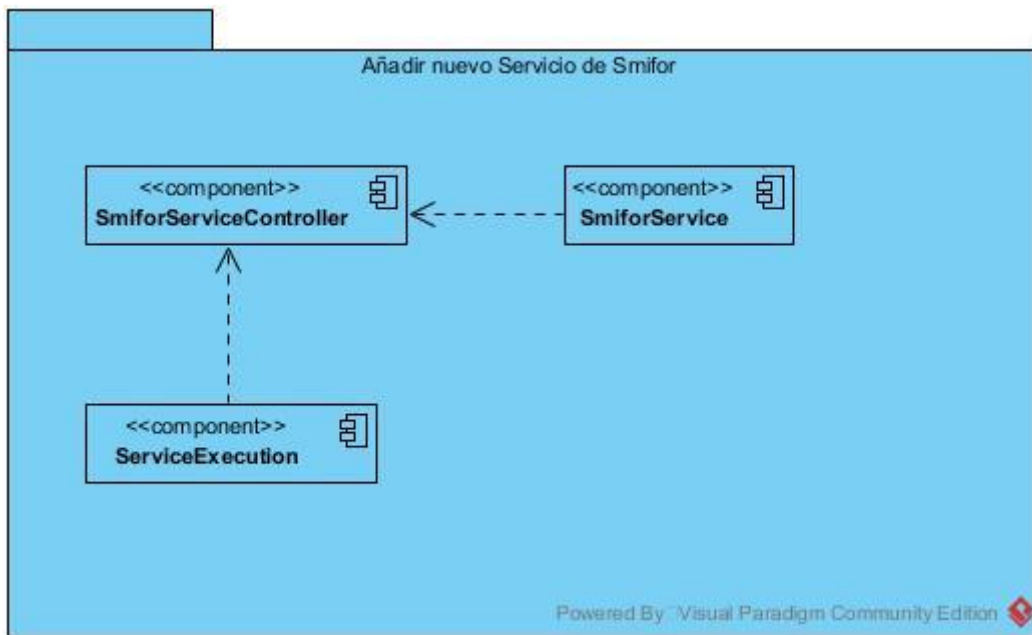


Ilustración 3: Diagrama de Componentes

### 3.2.4 Diagrama de Secuencia

En la siguiente imagen se muestra el Diagrama de Secuencia correspondiente al caso de uso Añadir Nuevo Servicio de Smifor.



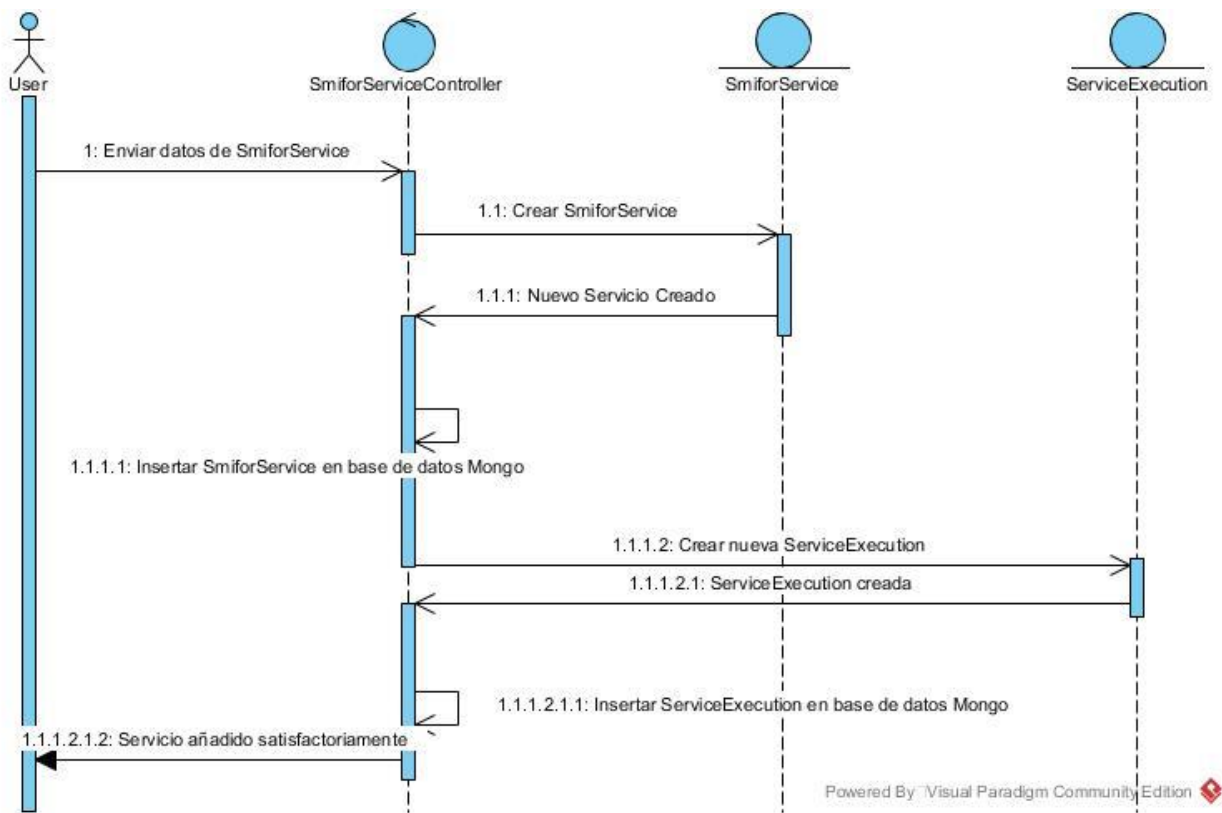


Ilustración 4: Diagrama de Secuencia

### 3.2.5 Diagrama de Despliegue

En la imagen que se muestra a continuación se presenta el Diagrama de despliegue del proyecto.

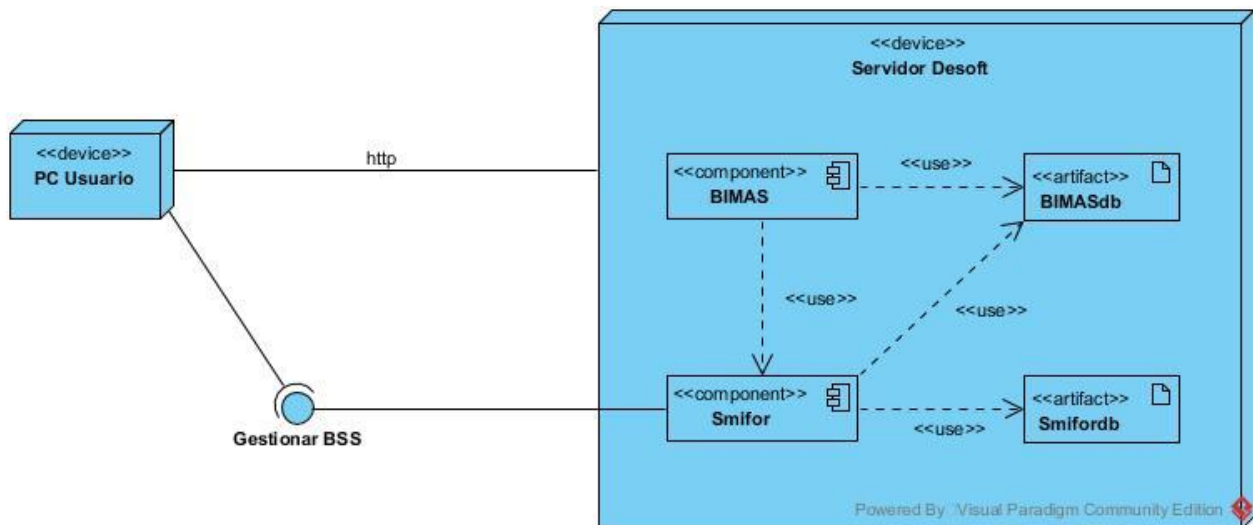


Ilustración 5: Diagrama de Despliegue

### 3.2.6 Bases de Datos

La aplicación usa dos bases de datos de MongoDB:

- smifordb:

Es la base de datos en la cual se almacena la información que usa la aplicación. En esta persistirán los servicios de Smifor y todo lo referente a sus ejecuciones y las fechas de las mismas.

- testdb:

Representa la base de datos del sistema BIMAS. Contiene las colecciones en las que se guardará la información que provee el cliente mediante la publicación de servicios web. Esta información será utilizada por BIMAS para realizar varias operaciones completando así la función de un sistema ELT.

Smifor extrae la información, la carga en el sistema BIMAS y este la transforma. Esto constituye una novedad.

Es importante recalcar que MongoDB es un gestor de bases de datos no relacional orientado a documentos, por lo que las colecciones que componen sus bases de datos no poseen un modelo estructurado ni un esquema de datos estricto, esto representa una ventaja para el proyecto ya que no se conoce la estructura de los datos que el cliente publica y necesita cargar en BIMAS. Además, contribuye a la escalabilidad de Smifor ya que si cambiara el modelo de los servicios de este no será necesario modificar ningún modelo de bases de datos para hacerlos persistir. También es importante el hecho de que se espera que las operaciones más usadas sobre estas bases de datos sean de tipo Read o Create, significando que se van a insertar datos u obtenerlos. Estas operaciones suelen ser más rápidas en bases de datos no relacionales porque se evitan las consultas anidadas.

### 3.2.7 Patrones de Diseño

En la aplicación desarrollada se utilizaron algunos patrones de diseño, típicos del trabajo con el framework Spring y se mencionan a continuación.

- Simple Factory:

La esencia de este patrón de diseño es que una clase Factory determina dinámicamente qué clase de producto debe crearse atendiendo a los parámetros pasados. Así sucede con el

BeanFactory de Spring, donde el objeto se obtiene pasando un identificador único, pero si se crea antes o después de pasar los parámetros depende de la situación.

- Singleton:

Asegura que se trabaja con una sola instancia de una clase. Los beans definidos en los archivos de configuración de Spring son singleton por defecto.

- Proxy:

Es ampliamente utilizado en AOP. Desde un punto de vista estructural es muy parecido al patrón Decorator, pero es más un control o una restricción de funciones.

- Controlador frontal:

Este patrón se utiliza para proporcionar un mecanismo centralizado que procesa las solicitudes y luego pasa la solicitud al controlador apropiado.

- Dependency Injection (DI) and Inversion of Control (IOC):

El contenedor de IOC de Spring se encarga de crear los objetos, vincularlos, configurarlos y manejar su ciclo de vida desde la creación hasta la destrucción.

La inyección de dependencias controla los componentes de la aplicación. Estos componentes son llamados Beans.

### ***3.3 Pruebas de Software***

En este epígrafe se presentan todos los aspectos referentes a las pruebas que se realizaron para validar la solución implementada, sus definiciones, y se reflejan los resultados de la realización de las mismas.

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de un software. Son la forma en la que se puede comprobar y garantizar la funcionalidad, el rendimiento del mismo y la experiencia del usuario.<sup>12</sup>

---

<sup>12</sup> Tipos de pruebas de software: diferencias y ejemplos (2020)

### 3.3.1 Pruebas de Integración

Consisten en probar diferentes módulos de una aplicación como grupo. Tienen como propósito validar la integración de dichos módulos con el fin de identificar los problemas y errores relacionados con ellos.<sup>13</sup>

Se valora si los componentes individuales trabajan en conjunto tal y como se espera de ellos, o lo que es lo mismo, se prueba el funcionamiento de los diferentes módulos del sistema una vez unidos o agrupados en elementos mayores, verificando el comportamiento de los mismos frente a las comunicaciones que se produzcan entre ellos. El objetivo es la localización de errores y comprobar el correcto funcionamiento conjunto de los componentes.

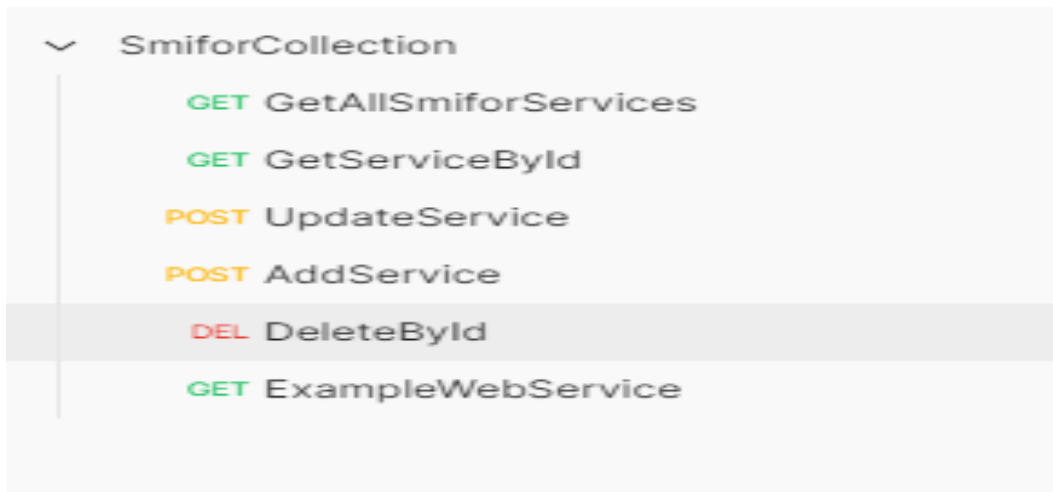
En las pruebas de integración es posible encontrarnos con los mismos errores que nos encontraríamos en las pruebas unitarias, pero su localización sería más complicada, de ahí que sea aconsejable realizar ambos tipos de pruebas en fases diferentes para poder localizar los errores de manera eficiente y garantizar la calidad del producto.<sup>14</sup>

Para realizar las pruebas de integración se utilizó la herramienta Postman, en la cual se creó una colección de requests, que corresponden a cada uno de los end point que conforman la API desarrollada. A continuación, se muestran los resultados obtenidos al realizar cada una de las pruebas:

---

<sup>13</sup> Ídem

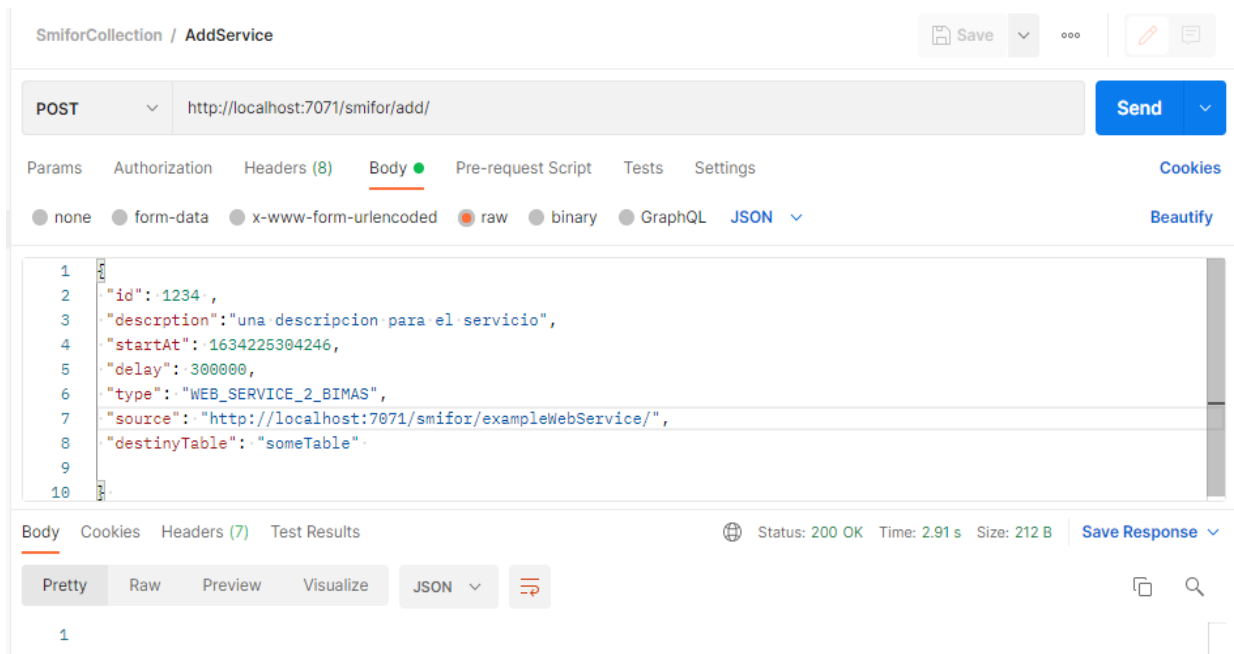
<sup>14</sup> Las Pruebas de Integración como Proceso de la Calidad del Software en el Ámbito de las Telecomunicaciones (2015)



**Ilustración 6: Colección de Postman**

### 3.3.1.1 Añadir un nuevo servicio

En la siguiente imagen se puede observar cómo se prueba la funcionalidad de añadir un nuevo servicio con sus datos correspondientes:

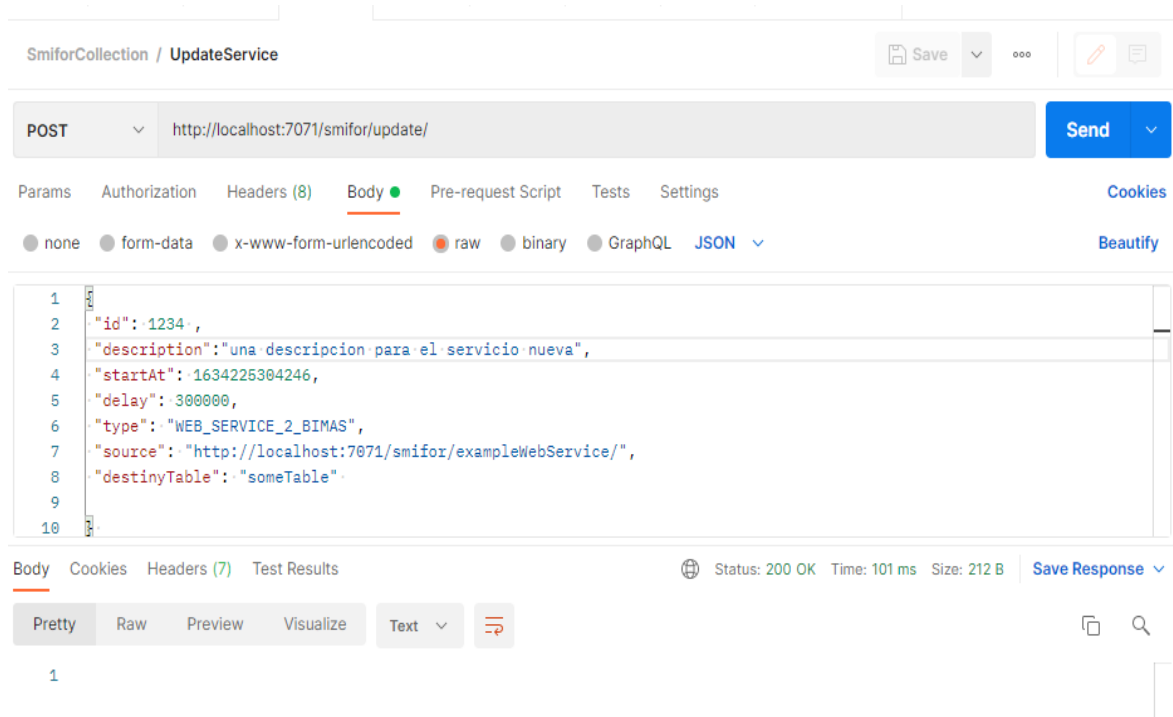


**Ilustración 7: Imagen de la prueba de Añadir un nuevo Servicio**

Como se puede observar, se prueba el end point sin problemas. La respuesta es nula ya que el método al que se llama no devuelve nada, pero se aprecia que no ocurre ningún error. Un nuevo servicio ha sido añadido con los datos que se aprecian en la imagen.

### 3.3.1.2 Editar o actualizar un servicio

En la siguiente imagen se realiza la prueba para la funcionalidad de actualizar o modificar los datos de un servicio en específico:

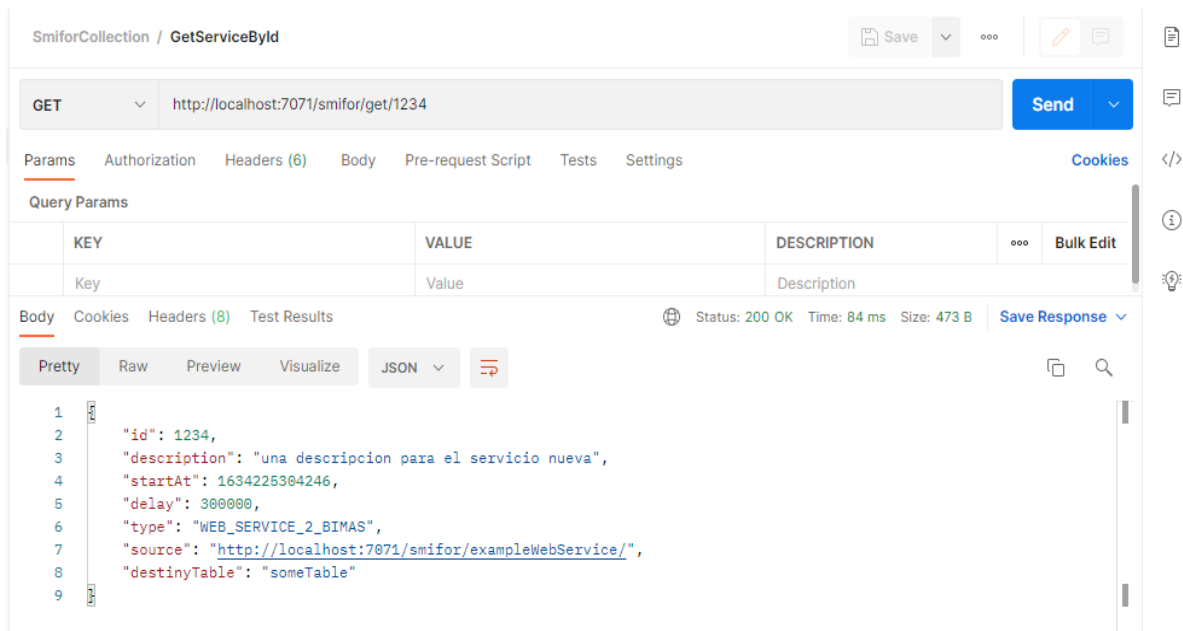


**Ilustración 8: Imagen de la prueba de Modificar un servicio existente**

Como se puede apreciar, no se obtienen errores como respuesta por lo que se ha modificado sin problemas la información correspondiente al servicio con el “id” escogido.

### 3.3.1.3 Obtener un servicio dado su “id”

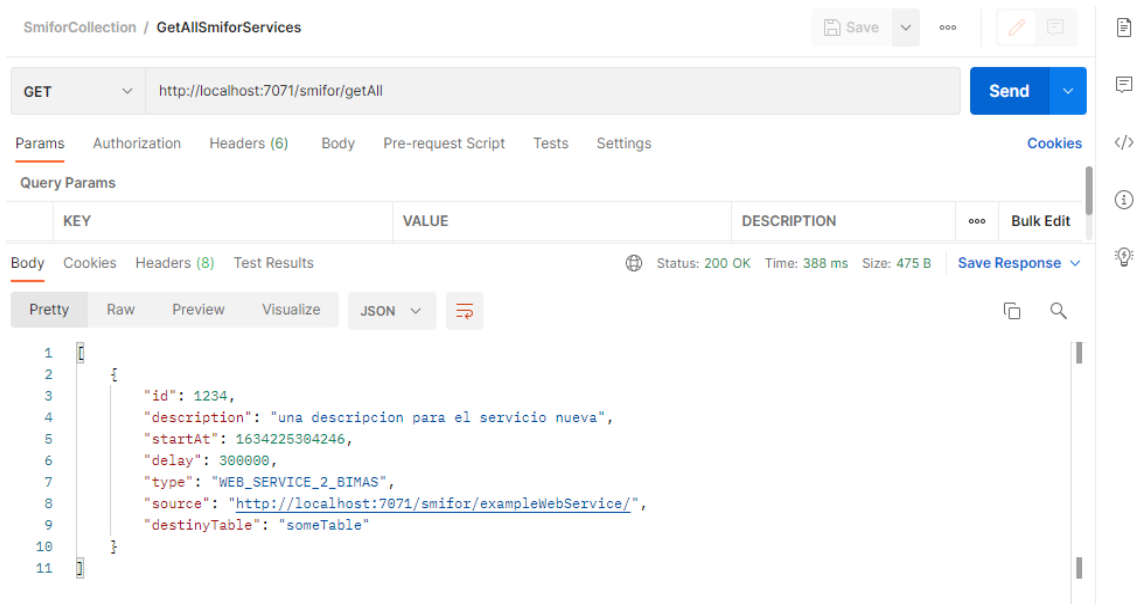
En este caso se va a trabajar con el caso de prueba correspondiente a obtener, de existir este, un servicio dado su “id”:

**Ilustración 9: Imagen de prueba de Obtener un Servicio**

Se obtiene como respuesta un objeto de tipo JSON que representa el servicio con el “id” especificado (es en este caso el mismo servicio que previamente fue insertado y luego modificado).

### 3.3.1.4 Obtener todos los servicios

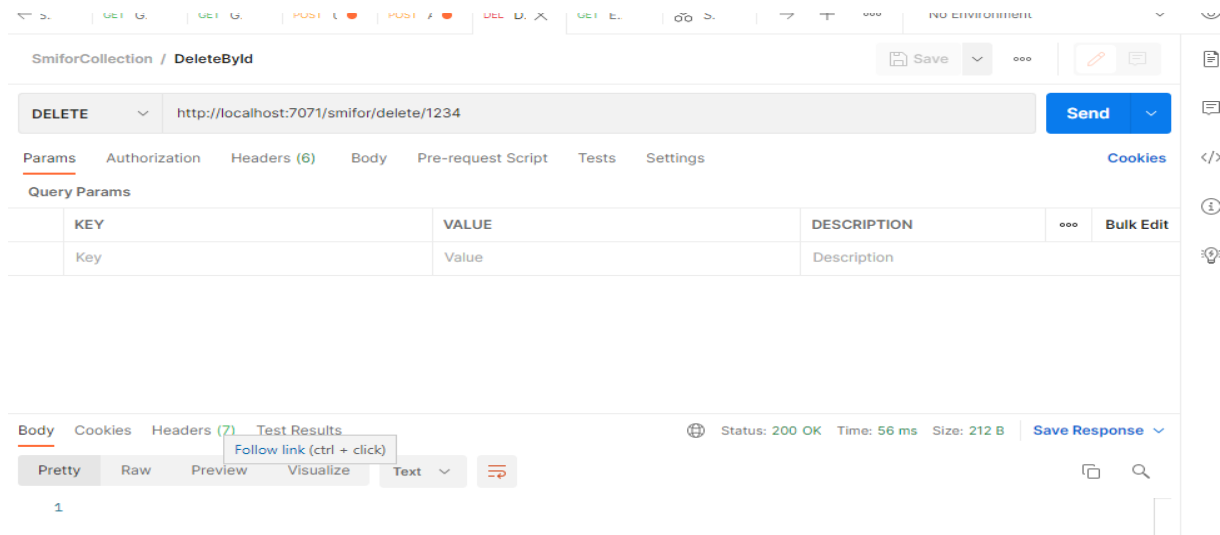
Ahora se intentará obtener todos los servicios que se han insertado:

**Ilustración 10: Imagen de prueba de obtener todos los Servicios**

Ya que solo se ha insertado un servicio obtenemos como respuesta el mismo con el que se ha ido trabajando hasta el momento.

### 3.3.1.5 Eliminar un servicio

A continuación, se probará el caso de eliminar un servicio dado su “id”:



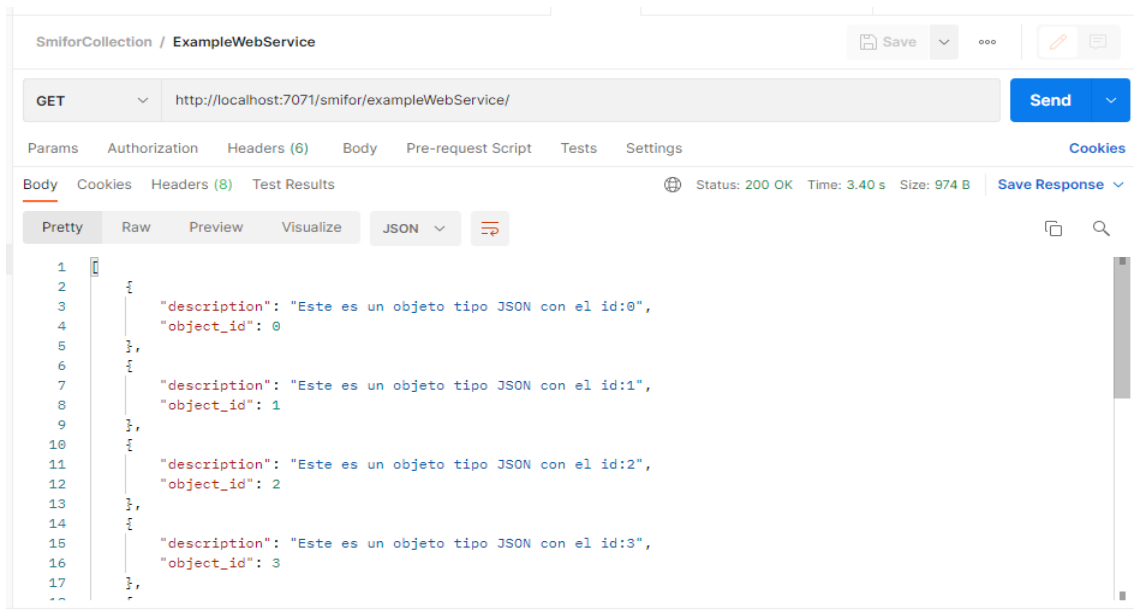
**Ilustración 11: Imagen de prueba de Eliminar un Servicio**

Nuevamente nos encontramos ante un caso donde no recibimos respuesta, pues el método de eliminar un servicio no devuelve nada.

### 3.3.1.6 Servicio Web de prueba

También se comprobará el funcionamiento de este servicio, pensado para devolver una cantidad de objetos en formato JSON:





**Ilustración 12:** Imagen de respuesta del servicio Web de prueba

Se observa cómo se recibe como respuesta la lista de objetos ya mencionada.

### 3.4 Conclusiones parciales

En este capítulo se mostró el desarrollo del proyecto siguiendo los preceptos de la metodología ágil Scrum y sus artefactos mostrando cada uno de los sprint. Se describe la propuesta de solución mostrando el diagrama de clases, se describen los patrones de diseño empleados producto del uso del framework Spring y se realizan pruebas de integración para validar la solución empleando la herramienta Postman para consumir los end point de la API desarrollada.

## CONCLUSIONES

La realización del presente trabajo permite arribar a las siguientes conclusiones:

1. Se estudiaron las características de distintos sistemas de inteligencia de negocios usados en la actualidad, así como las herramientas de extracción, transformación y carga de datos asociadas a los mismos.
2. Se identificaron los cambios a aplicar en la versión 2.0 de Smifor y se definió el alcance y los requisitos de la versión 3.0 implementados en este trabajo.
3. Se desarrolló dicha versión 3.0 usando las tecnologías y metodología adecuadas y se sometió la propuesta de solución a pruebas de integración para validar los resultados.

## RECOMENDACIONES

A partir de la realización del presente trabajo y el estudio realizado, se recomienda:

- Incorporar funcionalidades al sistema desarrollado, rescatando funcionalidades de la versión anterior y añadiendo nuevas para incrementar los tipos de fuentes de datos que pueden consumir los servicios de Smifor y los destinos a los que se envía información.
- Desarrollar el front end del sistema, proporcionando una interfaz gráfica amigable que permita el correcto uso de la aplicación.

## REFERENCIAS BIBLIOGRÁFICAS

- ALEXANDER MENZINSKY, G. L., JUAN PALACI 2016. SCRUM Manager Guía de Formación. *SCRUM Manager*, 94.
- BELLÓN, C. N. 2015. Business Intelligence en el ámbito Académico. 109.
- C.A., A. A. W. A. 2020. La plataforma SpagoBI Open Source Business Intelligence *SpagoBI Open Source Business Intelligence*, 19.
- CORPORATION, M. S. S. 2020. MS SQL Server Tutorial. *Tutorials Point Simply Easy Learning*, 92.
- CORPORATION, P. 2020. Pentaho software líder de Inteligencia de Negocio de código abierto 11.
- DATOS, P. D. E. E. G. D. 2019. Procesos ETL La base de la Inteligencia de Negocios. 26.
- DAYAL, U., CASTELLANOS, M., SIMITSIS, A. & WILKINSON, K. 2018. Data Integration Flows for Business Intelligence. 11.
- DONOHUE, M. 2018. Introduction to Oracle Business Intelligence Standard Edition On. *Business Intelligence Standard Edition One*, 40.
- LEE, G. 2020. Tipos de pruebas de software: diferencias y ejemplos. *LoadView*.
- RODRÍGUEZ, N. G. 2015. Las Pruebas de Integración como Proceso de la Calidad del Software en el Ámbito de las Telecomunicaciones. 199.
- SCRUMSTUDY 2013. Una guía para el CONOCIMIENTO DE SCRUM. *SCRUM Study Targeting Success*, 310.
- SMALLING, J. 2017. Guía del Usuario de Oracle Business Intelligence Enterprise. 488.
- UA, A. W. S. I. 2008. JSON. *Curso librerías Web 2.0.*, 12.

## **ANEXOS**

**Anexo 1.** Manual de Usuario BIMAS Smifor v3.0

**Anexo 2.** SmiforCollection.postman\_collection.json