

**ADAPTACIÓN DE SERVICIOS WEB PARA LA GENERACIÓN DE ALERTAS
TEMPRANAS DE ENFERMEDADES EN CULTIVOS BASADO EN LA
INTEGRACIÓN DE PROCESAMIENTO DE EVENTOS COMPLEJOS Y BUS DE
SERVICIOS EMPRESARIALES**



Oscar Ricardo Valencia Aguilar
Tesis de Maestría en Ingeniería Telemática

Director:
Juan Carlos Corrales
Doctor en Ciencias de la Computación

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de investigación e-@mbiente
Popayán, febrero de 2018

OSCAR RICARDO VALENCIA AGUILAR

**ADAPTACIÓN DE SERVICIOS WEB PARA LA
GENERACIÓN DE ALERTAS TEMPRANAS DE
ENFERMEDADES EN CULTIVOS BASADO EN LA
INTEGRACIÓN DE PROCESAMIENTO DE EVENTOS
COMPLEJOS Y BUS DE SERVICIOS
EMPRESARIALES**

**Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de**

**Magíster en:
Ingeniería Telemática**

**Director:
PhD. Juan Carlos Corrales**

**Popayán
2018**

A Dios por regalarme la oportunidad de seguir avanzando académicamente, por darme salud y vida para afrontar las nuevas oportunidades que pone en mi camino. A mi familia por ser el soporte para continuar adelante y por toda la ayuda que siempre me han brindado, a mi tutor por la paciencia y su acompañamiento, a mis amigos por el apoyo en cada momento.

Resumen Estructurado

Antecedentes: Históricamente el café ha sido uno de los cultivos más significativos en la agricultura colombiana, pues se ha caracterizado por ser el tercer país con mayor producción a nivel mundial con una participación entre 2014 – 2015 de 8% de la producción mundial ubicándose debajo de Vietnam con el 20 y Brasil con el 33%. Por esta razón, es de gran importancia estudiar el principal problema fitosanitario de alto impacto para la caficultura que es la roya. Particularmente en Colombia llegó en los años 80's afectando los cultivos de las áreas bajas de 600 a 1000 msnm, perjudicando a más de un millón de familias que dependen de este cultivo, motivo por el cual se hace de vital importancia contar con medidas para minimizar las pérdidas que este puede generar sobre las cosechas. En este sentido, la adaptación de los servicios web para la generación de alertas tempranas de enfermedades de acuerdo a la ubicación y la información del cultivo brindados por el usuario desempeñan un papel importante para los agricultores que dependen de esta enfermedad.

Objetivos: Adaptar servicios web para la generación de alertas tempranas de enfermedades en cultivos mediante la integración del procesamiento de eventos complejos y el bus de servicios empresariales.

Métodos: Se propone integrar el procesamiento de eventos complejos y el bus de servicios empresariales con la intención de adaptar los servicios web que soportan la generación de alertas tempranas de enfermedades en cultivos. Esto con el fin de lograr un sistema de alerta temprana que además de identificar las condiciones climáticas adecuadas para la aparición de la roya en el cafeto, permita realizar una personalización de alertas tempranas a partir de información del usuario, por medio de la adaptación de los servicios encargados de generar la alerta.

Resultados: Modelo de integración entre bus de servicios empresariales y procesamiento de eventos complejos para la adaptación de servicios web, conjunto de patrones de eventos que permiten identificar los eventos complejos de interés relacionados con las condiciones climáticas y un mecanismo de adaptación de servicios web basado en la integración del bus de servicios empresariales y el procesamiento de eventos complejos.

Conclusiones: Mediante la integración del bus de servicios empresariales (*Enterprise Bus Services*, ESB) y el procesamiento de eventos complejos (*Complex Event Processing*, CEP) se logró adaptar la respuesta ofrecida al usuario, gracias a la adición de distintos servicios web que proporcionan información relevante sobre el manejo y control de una enfermedad, teniendo en cuenta los datos adicionales que el usuario pudiera brindar.

Palabras Clave: Bus de Servicios Empresariales, Procesamiento de Eventos Complejos, Sistemas de Alerta Temprana, Arquitecturas dirigidas por eventos, Arquitecturas Orientadas a Servicios.

Structured Abstract

Background: Historically, coffee has been one of the most significant crops in Colombian agriculture, as it has been characterized as the third country with the highest production worldwide with a participation between 2014 - 2015 to 8% of the world production located below Vietnam with 20% and Brazil with 33%. For that reason, is very important study the main high-impact phytosanitary problem for coffee production, which is rust. Particularly in Colombia, it arrived in the 1980s, affecting crops in the low areas of 600 to 1000 masl, harming more than one million families that depend on this crop, which is why it is vital to have measures to minimize the losses that this can generate on the crops. In this sense, adaptation of web services for the generation of early disease alerts according to the location and information on the crop provided by user play an important role for farmers who depend on this disease.

Goals: Adapt generation web services of early disease alerts in crops through integration of complex event processing and enterprise service bus

Methods: Enterprise service bus and complex event processing integration are proposed with the purpose that adapted web service that support generation of early disease alerts in crops. This in order that achieve an early warning system that detect weather conditions for coffee rust and allows personalization that early warning from user information through web service adaptation.

Results: Enterprise service bus integration model and complex event processing for web service adaptation, an event pattern set that allows detect complex event related to weather conditions and an adaptive mechanism of web services based on enterprise service bus integration and complex event processing.

Conclusions: Through ESB and CEP integration, was achieve offered adapted response to user, thanks to addition to different web services that provides relevant information about management and control to disease, keeping in mind get user additional data.

Keywords: Enterprise Services Bus, Complex Events Processing, Early Warning Systems, Event-driven Architectures, Service-Oriented Architectures

Contenido

Lista de Figuras	III
Lista de Tablas	IV
Capítulo 1 Introducción	1
Planteamiento del Problema	1
Escenario de Motivación	2
Objetivos	4
Objetivo general.....	4
Objetivos específicos.....	4
Contribuciones	5
Contenido de la Monografía	6
Capítulo 2 Estado actual del conocimiento	9
Conceptos generales	9
Arquitecturas Orientadas a Servicios y Dirigidas por Eventos.....	9
Servicios Web.....	10
Procesamiento de Eventos Complejos (CEP)	11
Bus de Servicios Empresariales (ESB).....	11
Adaptación de Servicios Web	13
Sistemas de Alerta Temprana	14
Trabajos Relacionados	15
Procesamiento de Eventos Complejos en Diferentes Dominios de Aplicación	
.....	15
Bus de Servicios Empresariales y Procesamiento de Eventos Complejos	16
Adaptación de Servicios Web y Bus de Servicios Empresariales	17
Sistemas de Alertas Tempranas para la Detección de Enfermedades	21
Brechas de Investigación	22
Resumen	23
Capítulo 3 Adaptación de Servicios Web a través de un ESB y CEP	25
Descripción General	25
Componentes	27
Flujo de ejecución.....	28
Criterios de adaptación	29
Directivas de adaptación	29
Patrones de adaptación	35
Resumen	40
Capítulo 4 Solución Propuesta	41
Descripción General	41
Arquitectura General	43
Generadores de Información.....	45
Detección de Situaciones.....	45
Interacción entre Componentes	46
Detección de Eventos.....	46
Interacción del Usuario	47
Desarrollo del Prototipo	49
Herramienta Utilizadas	49

Desarrollo de servicios web	50
Descripción de la implementación	52
Resumen	58
Capítulo 5 Caso de Estudio.....	61
Dominio de aplicación	61
Caso de estudio 1	64
Caso de estudio 2	67
Evaluación.....	70
Creación del escenario.....	70
Tiempo de respuesta de los servicios web	71
Uso de recursos computacionales	72
Efectividad de decisión	73
Resumen	73
Capítulo 6 Conclusiones y Trabajos Futuros	75
Conclusiones.....	75
Trabajos Futuros.....	76
Bibliografía	77
Anexo A	83
Artículo AACCC17	83
Artículo ICCSA 2017	106
Anexo B	125
Uso de recursos computacionales	125
Anexo C	135
Código fuente	135

Lista de Figuras

Figura 1. Arquitectura lógica plataforma ESB adaptativa, extraído de [47].....	19
Figura 2. Arquitectura General Event-Driven Integration Platform for Context-Aware Web Services, extraído de [50].....	20
Figura 3. Modelo CEP-ESB-Adaptación. Fuente propia	27
Figura 4. Evento simple. Fuente propia.....	37
Figura 5. Evento complejo intervalo 30 días. Fuente propia.....	37
Figura 6. Evento complejos generación de alerta tipo TI1. Fuente propia.....	38
Figura 7. Evento complejo alerta tipo TI2. Fuente propia	38
Figura 8. Evento complejo alerta tipo TI3. Fuente propia	39
Figura 9. Diagrama general propuesto CEP-ESB. Fuente propia	41
Figura 10. Arquitectura general, basada en [50].....	44
Figura 11. Diagrama de secuencia para la interacción entre CEP y ESB. Fuente propia	47
Figura 12. Diagrama de secuencia para la interacción entre el usuario y los componentes de adaptación. Fuente propia	48
Figura 13. Diagrama de componentes y tecnologías. Fuente propia.....	49
Figura 14. Flujo de invocación servicio web productos. Fuente propia.....	55
Figura 15. Flujo de invocación servicio web experto. Fuente propia.....	56
Figura 16. Parte I flujo principal. Fuente propia	57
Figura 17. Parte II flujo principal. Fuente propia	58
Figura 19. Factores para que se presente la roya en el café [3]	62
Figura 20. Zona cafetera colombiana 1. Fuente propia	65
Figura 21. Zona cafetera colombiana 2. Fuente propia	66
Figura 22. Zona cafetera colombiana 3. Fuente propia	67
Figura 23. Zona cafetera 1 y fenología del café basado en [3].....	68
Figura 24. Zona cafetera 2 y fenología del café basado en [3].....	69
Figura 25. Zona cafetera 3 y fenología del café basado en [3].....	69

Lista de Tablas

Tabla 1. Funcionalidades básicas de un ESB, extraído de [32]	12
Tabla 2. Brechas existentes. Fuente propia	22
Tabla 3. Directivas de adaptación. Fuente propia	30
Tabla 4. Patrones de generación de roya. Fuente propia.....	35
Tabla 5. Servicios web AgroCloud – Fuente propia.....	51
Tabla 6. Servicios web desarrollados - Fuente propia	51
Tabla 7. Variables climáticas escenario prueba - Fuente propia.....	70
Tabla 8. Tiempo de respuesta de servicios web. Fuente propia	72
Tabla 9. Uso de recursos computacionales. Fuente propia.....	72
Tabla 10. Resumen Adaptación. Fuente propia	73

Capítulo 1

Introducción

Planteamiento del Problema

En la agricultura, los productores deben lidiar con distintos problemas que se presentan durante la producción de los diferentes cultivos [1] [2] [3]. Uno de estos problemas son las contra las distintas enfermedades y plagas que afectan a los cultivos, lo cual impacta en la disminución de la calidad y cantidad de producción. Generalmente, el origen de las enfermedades en los cultivos se debe al ataque de hongos, bacterias o virus, entre otros. Lo cual depende del cultivo sembrado, las condiciones meteorológicas, las zonas de producción del país, entre otras.

A través del tiempo, distintas enfermedades han arrasado con cultivos y cosechas en todo el mundo, causando epidemias y pérdidas. Por ejemplo: Estados Unidos perdió grandes extensiones de maíz ya que fueron atacados por el tizón [4], Brasil presentó epidemias con la incidencia del café de exportación, asimismo, en Colombia la *Sigatoka negra* [5] ha ocasionado en varias oportunidades ataques severos en cultivos de banano. De la misma manera, a nivel mundial se reportan casos de grandes epidemias ocasionadas en cultivos como la caña de azúcar causado por el mosaico de la caña, la mancha amarilla en el trigo, entre otros [6]. Lo anterior implica que se deben tomar medidas eficientes con el objetivo de mitigar las pérdidas y mejorar la productividad mediante prácticas como la protección de cultivos, el uso de tecnologías limpias, entre otras.

Por esta razón, se hace importante usar técnicas y tecnologías que permitan realizar la detección de distintas enfermedades para su posterior tratamiento. Una alternativa para dicha detección es la implementación de sistemas de alerta temprana, estos permiten informar anticipadamente a la sociedad sobre las posibles amenazas de carácter natural, bióticas y físicas, con el propósito de prepararse y aplicar las medidas necesarias de acuerdo al nivel de riesgo. Estos sistemas son ampliamente utilizados en la vigilancia y control de enfermedades en cultivos para declarar medidas preventivas y de control en problemas relacionados con el diagnóstico de las enfermedades [7]. Algunos ejemplos de estos sistemas se encuentran en [8] [9] [10], los cuales plantean distintos sistemas de alerta temprana para la detección y posterior control de enfermedades como el tizón en el maíz, la roya en el café, el chinche de los

pastos, entre otros. Estos sistemas tienen como propósito advertir sobre la presencia de cualquiera de estas enfermedades, apoyados en el comportamiento del clima, modelos estadísticos, modelos de simulación, entre otros. De esta manera, los agricultores tienen una pauta para iniciar el tratamiento adecuado de acuerdo a los lineamientos previamente establecidos en cada enfermedad y cultivo.

En los sistemas de alerta temprana existen factores a considerar que permiten tomar decisiones apropiadas ante la presencia de una alerta, algunos de estos factores a considerar son los tiempos extensos de preaviso y las tasas bajas de falsas alarmas [11]. Estos factores deben actuar positivamente en la toma apropiada de decisiones y para lograrlo requieren analizar aspectos clave para la generación de la alerta como el correcto almacenamiento de los datos (tasas bajas en los errores), funcionamiento adecuado de los dispositivos usados para la recolección de datos, procesos efectivos de envío de datos, calibración adecuada de los dispositivos, entre otros.

Es así como los sistemas de alerta temprana habitualmente están diseñados para advertir sobre alguna situación de riesgo a la mayor cantidad de usuarios, es decir, a una gran población como, por ejemplo, un país o una región. En el caso particular de la agricultura, cuando se habla de alertas tempranas se debe tener en cuenta que estas alertas deben ser más específicas, de tal manera, que incluyan características propias del lugar de siembra y del cultivo. En este sentido, la presente propuesta de maestría plantea la siguiente pregunta de investigación: ¿Cómo integrar el procesamiento de eventos complejos y el bus de servicios empresariales con el propósito de adaptar los servicios web que soportan la generación de alertas tempranas de enfermedades en cultivos?

Escenario de Motivación

Históricamente el café ha sido uno de los cultivos más significativos en la agricultura colombiana, pues se ha caracterizado por ser el tercer país con mayor producción a nivel mundial con una participación entre 2014 – 2015 de 8% de la producción mundial ubicándose debajo de Vietnam con el 20% y Brasil con el 33% [12]. Sin embargo, últimamente Colombia ha presentado una reducción en las exportaciones pasando de exportar 624.420 toneladas de café en el periodo de enero a octubre de 2015 a 605.040 toneladas de café en el mismo periodo del 2016 lo que significó una variación del -3% en las exportaciones de café [12]. Esta reducción se presenta por diversas razones entre las cuales se encuentra la afectación de distintas enfermedades como la roya (*Hemileia vastatrix*), para lo

cual la Federación Nacional de Cafeteros presentó un plan de erradicación incentivando la siembra de una variedad con resistencia genética al patógeno [13]. No obstante, esta variedad y otras de las variedades de cafeto cultivados en Colombia son susceptibles al hongo, si no se aplican las recomendaciones mínimas para evitar la generación, claro está, que unas variedades pueden infectarse más fácilmente que otras y además pueden presentar un mayor o menor grado de la enfermedad [14].

En este sentido, se hace importante diseñar estrategias enfocadas en el sector cafetero, con el fin de mejorar la producción de café en Colombia y con ello la participación en el mercado mundial. Las estrategias se apoyan en diversas herramientas que reduzcan las pérdidas generadas por las enfermedades que atacan a los cultivos de café como los sistemas de alerta temprana, estos sistemas se han convertido en un instrumento importante para evitar o reducir el riesgo ante posibles desastres, enfermedades o cualquier evento de carácter predecible. Este tipo de sistemas permite que los usuarios puedan contar con información oportuna, clara y efectiva, lo que ayuda a tomar las medidas adecuadas para evitar, reducir o responder adecuadamente ante un riesgo [15].

Actualmente, los sistemas de alerta temprana se han convertido en una herramienta importante para evitar o reducir el riesgo ante posibles desastres, enfermedades o cualquier evento de carácter predecible [15], permitiendo que todos sus usuarios puedan contar con información oportuna, clara y efectiva, lo que ayuda a tomar las medidas adecuadas para evitar, reducir o responder adecuadamente ante un riesgo.

Un sistema de alerta tempranas eficaz y eficiente contiene cuatro elementos fundamentales:

- Conocimiento del riesgo, este componente proporciona información esencial para evaluar el riesgo actual y priorizar las estrategias de mitigación, prevención y respuesta ante el riesgo.
- Servicio de vigilancia y advertencia, este componente permite realizar un análisis de la información percibida para proporcionar estimaciones oportunas del riesgo potencial con el fin de generar avisos precisos de manera oportuna.
- Difusión y comunicación, este componente se encarga de gestionar todos los procedimientos necesarios para enviar mensajes de advertencia a los usuarios potencialmente afectados mediante diversos canales o servicios, a través de avisos confiables, simplificados y fáciles de entender para todo público.

- Capacidad de respuesta, este componente se encarga de los conocimientos del riesgo y de los mecanismos de respuesta por parte de la población, es decir, los planes de acción apropiados para reaccionar ante una situación de peligro, además de las herramientas disponibles para afrontar un fenómeno [15].

Normalmente, los sistemas de alerta temprana producen notificaciones de manera general, es decir, que abarcan la mayor cantidad de población como, por ejemplo, alertas para todo un país o una región. Debido a esto, muchas alertas son emitidas y enviadas a todos los usuarios que se encuentran dentro del territorio; pero no necesariamente la alerta emitida afecta a todos los usuarios notificados, dicho de otra manera, solo algunos de los usuarios podrían resultar afectados por el riesgo anunciado por la alerta. En los sistemas de alerta temprana para la agricultura se requieren especificidades, puesto que los cultivos están altamente relacionados con las condiciones particulares del lugar donde se encuentra sembrado el cultivo, por lo tanto, al hablar de alertas tempranas para los cultivos se deben tener en cuenta diversas características específicas del lugar donde se emiten, de tal manera que si se desea brindar alertas tempranas que sean relevantes para los agricultores estas deberían incluir información específica del agricultor y su cultivo, por ejemplo, la ubicación, características particulares del cultivo (tipo de cultivo, fecha de siembra, fecha de floración, etc.), entre otros, en caso contrario la alerta generada no lograría el propósito de evitar o reducir el riesgo, sino que se convertiría en información adicional y general para el agricultor. Por lo anterior, surge la necesidad de proponer un sistema de alerta temprana que además de identificar las condiciones climáticas adecuadas para la aparición de la roya en el cafeto, permita realizar una personalización de alertas tempranas a partir de información del usuario, por medio de la adaptación de los servicios encargados de generar la alerta.

Objetivos

Objetivo general

Adaptar servicios web para la generación de alertas tempranas de enfermedades en cultivos mediante la integración del procesamiento de eventos complejos y el bus de servicios empresariales.

Objetivos específicos

- Establecer criterios de adaptación a los contratos de los servicios a partir de un conjunto de patrones de eventos.

- Definir las directivas de adaptación de los servicios mediante el procesamiento de eventos complejos.
- Integrar la adaptación de servicios propuesta en el módulo de monitorización del ESB.
- Evaluar experimentalmente la propuesta de integración CEP-ESB en el dominio de las alertas tempranas de enfermedades en cultivos, tomando como caso de estudio la roya en el café.

Contribuciones

Las principales contribuciones de este trabajo de maestría son las siguientes.

- Modelo de integración entre ESB y CEP para la adaptación de servicios web. En él que se presentan los componentes a tener en cuenta, además del flujo de ejecución empleado para realizar dicha integración.
- Conjunto de patrones de eventos que permiten identificar los eventos complejos de interés para el trabajo de grado, los cuales se encuentran relacionados con las condiciones climáticas.
- Mecanismo de adaptación de servicios web basado en la integración de ESB y CEP. Dentro de este mecanismo se encuentran los diagramas, herramientas, demás elementos necesarios para realizar la adaptación de los servicios web.
- Un artículo expuesto en la Conferencia Internacional para la Adaptación de la Agricultura al Cambio Climático (AACC'17) realizados los días 22, 23 y 24 de noviembre de 2017 en Popayán – Colombia, el cual tuvo como principal objetivo desarrollar un modelo de integración entre ESB y CEP para la adaptación de servicios web para el entorno cafetero colombiano. Este artículo será publicado en *Advances in Intelligent Systems and Computing*. Clasificado como C en Publindex-Colciencias, ISSN: 2194-5357. (Ver ANEXO A).
- Un artículo expuesto en 17th Conferencia Internacional en Ciencias de la Computación y Aplicaciones (ICCSA, 2017) los días 3, 4, 5 y 6 de julio de 2017 en Trieste, Italia, el cual se presenta un sistema de soporte a la toma de decisiones para el manejo de la roya en el café basado en el conocimiento de expertos. Este artículo fue publicado en el libro

Computational Science and Its Applications Volumen 6 de julio de 2017.
Clasificado como B2 en Publindex-Colciencias, ISBN 978-3-319-62407-5.
(Ver ANEXO A).

Contenido de la Monografía

El presente trabajo de grado está compuesto por seis capítulos los cuales se describen a continuación.

- **Capítulo 2. Estado Actual del Conocimiento**

Presenta los conceptos teóricos correspondientes a los conceptos principales en los que se basa el trabajo de grado como lo son el Procesamiento de Eventos Complejos (CEP), el Bus de Servicios Empresariales (ESB), Servicios Web, y la Adaptación de Servicios Web.

- **Capítulo 3. Adaptación de Servicios Web Mediante la Integración ESB y CEP**

Expone la propuesta conceptual que permite realizar la adaptación de los servicios web de la generación de alertas para los cultivos, dentro de él se presenta una descripción de la propuesta, seguida de los criterios y directivas de adaptación y finalmente los elementos que permitieron identificar los patrones de eventos de interés a partir de los cuales se realiza el proceso de adaptación.

- **Capítulo 4. Solución Propuesta**

Describe los diferentes pasos correspondientes a la implementación de la solución propuesta. Para esto se ilustra la manera en la cual fueron abordados los temas de las directivas de adaptación, la detección de los patrones de eventos de interés y como estos componentes se desarrollaron dentro de MuleESB.

- **Capítulo 5. Caso de Estudio**

En este capítulo fueron presentados los casos de prueba a utilizar en la solución propuesta. Principalmente se desarrollaron dos posibilidades de información adicional que puede ser proporcionada por el usuario para que sea combinada con la información climática y a partir de esta sea realizado el proceso de adaptación.

- **Capítulo 6. Conclusiones y Trabajos Futuros**

En este capítulo se describen las principales conclusiones del presente trabajo de grado de maestría a las cuales se llegó durante su desarrollo, posteriormente presenta las recomendaciones, y finalmente propone los trabajos futuros que pueden generarse a partir de este estudio.

Capítulo 2

Estado actual del conocimiento

El presente trabajo de grado dispone de una amplia referencia documental, la cual permite formar una base para los núcleos de trabajo sobre los cuales se orienta: Procesamiento de eventos complejos (CEP), Bus de servicios empresariales (ESB) y Adaptación de servicios web, conceptos que se han combinado dentro de los Sistemas de alerta temprana y se encuentran expuestos dentro de los Conceptos Generales (sección 2.1). Para la sección 2.2 se realiza una revisión de un conjunto de trabajos correspondientes a cada de los núcleos identificados en el proyecto.

Conceptos generales

Arquitecturas Orientadas a Servicios y Dirigidas por Eventos

En esta sección se describen las arquitecturas software relevantes dentro de este trabajo de grado. Primero se explican las Arquitecturas Orientadas a Servicios (SOA), seguidamente, las Arquitecturas Dirigidas por Eventos (EDA).

Arquitecturas Orientadas a Servicios (SOA)

Una arquitectura orientada a servicios (SOA) es una forma lógica de diseñar sistemas software con el objetivo de proveer servicios a aplicaciones o incluso a otros servicios, a través de interfaces que son publicadas y pueden ser descubiertas de manera dinámica [16]. En SOA el término de servicio puede entenderse como una función bien definida, autónoma y que no depende del contexto o del estado de otros servicios [17].

Cuando se habla de SOA se piensa en una arquitectura que ofrece escalabilidad, reutilización, rendimiento e interoperabilidad entre las aplicaciones o funcionalidades que son desarrollados en distintas plataformas en cualquier dominio de aplicación, razón por la cual SOA se caracteriza por contar con las siguientes propiedades [18]:

- Vista lógica: Un servicio es una vista lógica abstracta de programas o bases de datos, definido en términos de lo que hace.
- Orientado a mensajes: Un servicio está definido formalmente en términos del mensaje intercambiado entre el proveedor del servicio y el

que solicita dicho servicio. La estructura interna de los agentes involucrados es abstracta, es decir, no es necesario conocer cómo está implementado un servicio.

- Orientado a su descripción: Un servicio es descrito por metadatos procesable por máquinas. La semántica del servicio debe estar documentada en esta descripción.
- Granularidad: Los servicios tienden a usar un número reducido de operaciones con mensajes relativamente largos y complejos.
- Orientado a redes: Los servicios están enfocados para que su uso sea a través de redes.
- Independientes de la plataforma: Los mensajes intercambiados son independientes de la plataforma y estandarizados.

Arquitecturas Dirigidas por Eventos (EDA)

Una arquitectura dirigida por eventos, o *Event Driven Architecture* (EDA), define una metodología para diseñar e implementar aplicaciones y sistemas en donde los eventos son transmitidos entre componentes y servicios desacoplados, mientras SOA es más apropiado para un intercambio solicitud/respuesta, EDA introduce capacidades de procesamiento asíncrono de larga duración [19].

EDA se basa en publicación y suscripción a eventos, donde el que publica dichos eventos no conoce al suscrito y viceversa, lo único que comparten es la semántica del mensaje [20].

SOA Dirigido por Eventos (EDA)

Event-Driven SOA (ED-SOA) o también llamado SOA 2.0 es una extensión de SOA que combina lo mejor de dicha arquitectura con las capacidades de manejo de eventos de las arquitecturas EDA. Surge con la necesidad de proveer mecanismos para publicar y consumir eventos en arquitecturas orientadas a servicios [21].

Servicios Web

Un servicio web puede definirse como un conjunto de aplicaciones y tecnologías para soportar la interoperabilidad en la interacción máquina-máquina a través de una red [22]. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando estos procedimientos a través de la red [18].

Los servicios web proporcionan mecanismos de comunicación estándar entre aplicaciones que interactúan entre sí para presentar información dinámica al usuario.

Procesamiento de Eventos Complejos (CEP)

Tecnología que proporciona un conjunto de técnicas para usar eficientemente EDA [23]. CEP permite procesar, analizar y correlacionar grandes cantidades de eventos, para detectar y responder en tiempo cercano al real ante situaciones relevantes o críticas de interés [24].

De acuerdo con *Luckman*, un evento puede ser definido como cualquier cosa que ocurra o deba ocurrir [25], de tal modo que un evento puede ser clasificado en tres categorías: eventos simples, eventos complejos y eventos derivados. Los eventos simples ocurren en un punto de tiempo y son indivisibles, los eventos complejos resumen, incorporan o expresan un conjunto de eventos, y los eventos derivados son generados cuando se aplica un método o proceso a otros eventos [26]. Los eventos pueden ser producto de otros eventos mediante la aplicación o coincidencia de patrones de eventos, los cuales son modelos que especifican las condiciones donde se describen las situaciones a detectar. Es así como, el procesamiento de eventos complejos hace referencia a todas las operaciones que se realizan sobre dichos eventos como lo son: crear, leer, transformar, entregar, descartar, abstraer eventos, entre otros [27]. El motor CEP es el *software* usado para encontrar las coincidencias con los patrones de eventos establecidos sobre flujos de eventos continuos y heterogéneos, advirtiendo de los eventos complejos generados al detectar estos patrones [24].

En definitiva, CEP permite detectar eventos significativos e inferir conocimiento importante para los usuarios finales en distintos dominios. Para este fin, el CEP usa métodos algorítmicos para dar sentido a los eventos simples o base (bajo nivel de conocimiento) para obtener eventos complejos (alto nivel de conocimiento) de manera oportuna y en cortos periodos de tiempo [28]. La principal ventaja de utilizar esta tecnología es la posibilidad de identificar y reportar estos eventos en tiempo cercano al real, reduciendo la tardanza en la toma de decisiones.

Bus de Servicios Empresariales (ESB)

En sus inicios el Bus de Servicios Empresariales o *Enterprise Service Bus* (ESB) fue descrito como “una nueva arquitectura que aprovecha las ventajas de los servicios web, tecnologías mediadoras (*middleware*), enrutamiento inteligente y transformaciones” [29]. Actualmente, las definiciones de ESB ya no hacen referencia a una arquitectura, sino a una infraestructura que constituye el pilar de las arquitecturas orientadas a servicios (SOA). Esta infraestructura se basa en

estándares abiertos y está diseñada para facilitar la implementación, despliegue y administración de soluciones SOA. Cuenta con las capacidades de las infraestructuras implementadas con tecnologías *middleware*, que facilitan la utilización de SOA y disminuyen los problemas de disparidades entre aplicaciones que se ejecutan en plataformas heterogéneas usando diversos formatos para los datos [30].

Resumiendo, un ESB es una plataforma de integración centrada en estándares abiertos basado en mensajes que provee servicios de enrutamiento, invocación y mediación para facilitar la interacción de las aplicaciones y servicios distribuidos heterogéneos de manera segura y confiable [31]. ESB usa diferentes patrones como los de conectividad y mediación los cuales respectivamente permiten especificar los estilos de integración y especificar las familias de operaciones de mediación que se pueden efectuar sobre los mensajes que viajan a través del bus. ESB se encarga de coordinar la interacción entre varios recursos y proporcionar el soporte transaccional con la finalidad de proveer mensajes y la integración de los mismos sin escribir código adicional. Para esto, ESB dispone de componentes genéricos configurables con el objetivo de realizar el escenario deseado. Además, ofrece un estándar para interconectar todos los servicios requeridos, logrando una solución donde se tienen en cuenta aspectos como la seguridad, rendimiento, escalabilidad y fiabilidad.

Las ventajas de usar ESB se perciben de manera clara en contextos donde se realice una integración de ambientes heterogéneos, con aplicaciones que trabajen en múltiples plataformas. Los ESB tienen la capacidad de resolver en gran parte los procesos de integración permitiendo tener un control adecuado de los mismos.

Funcionalidades básicas de un ESB

Debido a que actualmente pueden encontrarse gran variedad de implementaciones de plataformas ESB, es importante identificar cuáles son las funcionalidades básicas que debe tener un ESB para ser considerado como tal. De acuerdo al libro “*Patterns: Implementing an SOA Using an Enterprise Service Bus*” [32] las funcionalidades básicas esperadas en un ESB se resumen en la Tabla 1.

Tabla 1. Funcionalidades básicas de un ESB, extraído de [32]

Categoría	Funcionalidad	Motivo
Comunicación	<ul style="list-style-type: none"> • Ruteo • Direccionamiento • Al menos un estilo de mensajería 	Proveer transparencia de localización de servicios y soportar la sustitución de servicios

	(solicitud/respuesta, etc.) <ul style="list-style-type: none"> Al menos un protocolo de transporte 	
Integración	<ul style="list-style-type: none"> Múltiples estilos de integración y adaptadores Transformación de protocolos 	Soportar la integración en ambientes heterogéneos y sustituir servicios
Interacción de servicios	<ul style="list-style-type: none"> Definición de interfaz de servicios Modelo de servicios de mensajes Sustitución de la implementación de servicios 	Soportar los principios de SOA, abstrayendo el código de la aplicación de los protocolos de servicios y su implementación
Administración	<ul style="list-style-type: none"> Capacidad de administración 	Punto de control sobre el direccionamiento y nombrado de servicios

Adaptación de Servicios Web

La adaptación es la capacidad de un servicio web, una herramienta o cualquier otro ente para modificarse, de tal forma que pueda acomodarse a diferentes circunstancias o condiciones generadas, ya sea por las necesidades específicas del usuario o por el entorno en el que se está ejecutando [33].

Los sistemas adaptativos se basan en dos factores para identificar el marco de adaptación: el primero es la personalización, que busca brindar el mejor acceso a la información, haciendo uso de diferentes variables como lo son: físicas, cognitivas, psicológicas o ambientales del usuario. Y el segundo factor que utilizan es el entorno o contexto, es decir, todo aquello que permite abstraer el ambiente del usuario cuando está interactuando con una aplicación o consumiendo un servicio, teniendo en cuenta aspectos como el tiempo, dispositivo, acceso y la localización [33].

La adaptación en los servicios web puede ser estática o dinámica. La adaptación estática significa que se cambia la composición del servicio web antes de que se inicie, mientras la adaptación dinámica hace referencia a los cambios en la composición del servicio cuando este está funcionando, es decir sin ser detenido

o reiniciado [34]. Una adaptación dinámica se puede presentar debido a diversas situaciones que pueden ser clasificadas en cuatro categorías: correctivas, adaptativas, de extensión y de perfeccionamiento.

Una adaptación correctiva elimina fallos en el comportamiento del componente que se está ejecutando, reemplazándolo por una nueva versión que tiene la misma funcionalidad pero que funciona correctamente. Una adaptación adaptativa, es una adaptación de la aplicación en respuesta a cambios del entorno (hardware, sistema operativo, entre otros). Otra clase de adaptación es la adaptación por extensión la cual permite añadir nuevas funcionalidades al sistema mediante la adición de nuevos componentes. Por último, las adaptaciones de perfeccionamiento se realizan para mejorar la aplicación, reemplazando un componente por otro que ha sido optimizado [35].

Sistemas de Alerta Temprana

Los sistemas de alerta temprana conocidos como SAT, son un conjunto de procedimientos e instrumentos, a través de los cuales se monitoriza una amenaza o evento adverso de carácter previsible [36]. Para esto, se recolectan y procesan datos e información, ofreciendo pronósticos o predicciones temporales sobre su acción y posibles efectos. Millones de personas en todo el mundo salvan sus vidas y sus medios de subsistencia gracias a la implementación de estos sistemas. La importancia de un SAT, radica en que permite conocer anticipadamente y con cierto nivel de certeza, en que tiempo y espacio, una amenaza o evento adverso de tipo natural o generado por la actividad humana puede desencadenar situaciones potencialmente peligrosas [37]. Por lo cual las alertas deben difundirse con suficiente anticipación.

El objetivo primordial de un sistema de alerta es capacitar a las personas y a las comunidades para que respondan de manera oportuna y adecuada a los peligros, con el fin de reducir el riesgo de muerte y los daños materiales [37]. Los avisos deben difundir el mensaje y estimular a quienes se encuentren en riesgo para que tomen medidas.

Cabe señalar que los plazos de preaviso más largos se deberían considerar junto con la necesidad de reducir la tasa de falsas alarmas y se debería lograr un equilibrio entre los dos, para que las decisiones se pudieran basar en plazos de preaviso óptimos para las alertas.

Los sistemas de alerta temprana son aplicables tanto a eventos naturales, como a aquellos provocados por la actividad humana y por la interacción de ambos elementos, cuyas características permiten su vigilancia y monitorización. Entre las amenazas o eventos más comunes a los cuales se aplican SAT tenemos las inundaciones, deslizamientos de tierra, huracanes, volcanes, tsunamis, incendios

forestales, fenómeno del niño y la niña, detección de enfermedades, entre otros [38].

Trabajos Relacionados

El presente trabajo es desarrollado alrededor de la adaptación de los servicios web que generan las alertas tempranas de enfermedades en cultivos mediante la integración del procesamiento de eventos complejos y el bus de servicios empresariales.

En el área de los sistemas de alerta temprana el principal objetivo es advertir oportunamente sobre cualquier eventualidad que pueda ser causada por la aparición de alguna enfermedad, con el fin de tomar las medidas necesarias para contrarrestar o controlar dicha enfermedad dependiendo del nivel de riesgo presente en el momento de la generación de la alerta. Por este motivo, es importante contar con mecanismos que puedan reaccionar ante distintos cambios en los servicios web que son utilizados para la generación de alertas tempranas de enfermedades en cultivos, y de esta forma brindar al usuario repuestas más completas para manejar determinadas situaciones de riesgo para su cultivo. Por lo anterior, CEP es una solución alternativa ante este tipo de situaciones ya que permite la detección de diferentes situaciones que se pueden requerir algún tipo de adaptación de los servicios en tiempo de ejecución.

En esta sección se consideran algunas aproximaciones existentes, relacionados con el problema de investigación declarado. Las propuestas de estos trabajos están orientadas hacia la adaptación de servicios web por medio de la integración entre ESB y CEP.

Procesamiento de Eventos Complejos en Diferentes Dominios de Aplicación

En la actualidad CEP ha cobrado gran importancia ya que brinda solución a diversos problemas donde el factor diferenciador es el procesamiento de patrones, datos, consultas, entre otros, en tiempo cercano al real y en distintos dominios de aplicación. Algunos de los dominios en los cuales ha incursionado el CEP son los siguientes: en la detección temprana de enfermedades humanas [39], en el cual se realizan revisiones periódicas a los pacientes verificando un conjunto de síntomas con el fin de descubrir posibles brotes de una determinada enfermedad. En el caso específico presentado en [39] se emplea como caso de estudio la detección de la gripe aviar, obteniendo como resultado la detección de la enfermedad en un tiempo cercano al real. Esto, comparado con los sistemas convencionales existentes para la detección de la enfermedad logra una mejorar y

optimizar el tiempo de detección de una semana aproximadamente a unos pocos segundos.

Otro dominio de aplicación se encuentra relacionado con el internet de las cosas (*Internet of Things*). El trabajo [40] utiliza el internet de las cosas en el ámbito de la domótica para lo cual se emplean un conjunto de sensores para detectar automáticamente y en tiempo cercano al real situaciones sobresalientes para el propietario de la vivienda, como lo son, el despilfarro energético, detección de un posible incendio, entre otras. Para lograr este propósito se emplea CEP para filtrar, agregar, transformar, detectar y responder en tiempo cercano al real y de forma automática ante los patrones de eventos, generados a partir de los datos enviados por diversos sensores instalados en la casa. El trabajo concluye con una arquitectura que integra los sensores del internet de las cosas, CEP y ESB con el fin de notificar a las partes interesadas sobre los eventos detectados. El trabajo [41] presenta un enfoque en el cual se utiliza CEP para el apoyo a la toma de decisiones en los sistemas de manejo de tráfico. Para esto el trabajo propone una arquitectura de un sistema de soporte a la toma de decisiones dirigido por eventos, donde la idea general es establecer un grupo de sensores en determinadas rutas con el fin de notificar posibles problemas de tráfico vehicular. Esto permite el análisis y procesamiento de flujos complejos de eventos en tiempo cercano al real, los cuales apoyarán la toma de decisiones considerando los sistemas de control de tráfico.

En el campo de la seguridad informática, el trabajo [42] presenta una herramienta que tiene como objetivo detectar los posibles ataques a un equipo informático a través de la red utilizando el procesamiento de eventos complejos. Para lograrlo, monitoriza el tráfico de la red para detectar cualquier situación que pueda poner en riesgo la seguridad del equipo. De esta manera, se generan distintos ficheros en tiempo cercano al real que son procesados y convertidos en eventos que luego llegarán al CEP donde se contrastan con los patrones de eventos previamente definidos identificando los posibles ataques. Esta estructura utiliza una arquitectura que integra SOA, CEP y ESB con el fin de responder en tiempo cercano al real a los ataques detectados enviando un correo electrónico al encargado para que este tome una decisión con respecto al incidente reportado.

Bus de Servicios Empresariales y Procesamiento de Eventos Complejos

ESB es una plataforma de integración centrada en estándares abiertos basado en mensajes que provee servicios de enrutamiento, invocación y mediación para facilitar la interacción de aplicaciones. Por esta razón, es cada vez más usado para integrar las arquitecturas dirigidas por eventos usando la tecnología que hace posible el uso efectivo de estas arquitecturas, es decir, CEP [43]. Algunos

trabajos que hacen uso de ESB y CEP están enfocados principalmente en la adaptación, transformación o mediación de diferentes servicios de acuerdo al contexto en el que están siendo utilizados. Este es el caso de [44] el cual presenta un trabajo cuyo principal objetivo es permitir la adaptación dinámica en infraestructuras de servicios basados en ESB, con el fin de soportar la auto-adaptación en SOA. El trabajo propone un enfoque donde se emplea un ESB para realizar adaptaciones dinámicas en SOA. Este planteamiento se encuentra basado en los patrones ofrecidos por ESB y definiciones sobre adaptación en SOA.

El trabajo [43] relaciona ESB y CEP con el principal objetivo de proponer una infraestructura basada en ESB que aproveche las capacidades de mediación y CEP para construir servicios web sensibles al contexto, esto con el propósito de presentar una infraestructura con los componentes nombrados anteriormente (ESB y CEP) que permitan la adaptación de servicios web de acuerdo a la información capturada del contexto, generando una adaptación transparente para el usuario con la capacidad de procesar diversas fuentes de información del contexto gracias al uso del CEP.

El trabajo [45] propone una arquitectura basada en SOA haciendo uso de las arquitecturas dirigidas por eventos (SOA 2.0) [46], teniendo como elemento central el ESB para optimizar la percepción del contexto por parte de los servicios. El caso de estudio evaluado está orientado en brindar actividades de ocio junto con ofertas especiales dependiendo de la localización y las condiciones climáticas. Dichas actividades y ofertas son brindadas gracias al uso de un ESB que permite conectar las distintas entradas - salidas del sistema y el uso de CEP para procesar los diversos factores que inciden en las actividades ofrecidas.

Adaptación de Servicios Web y Bus de Servicios Empresariales

La adaptación de los servicios web parte de la premisa de que los sistemas deberán ser capaces de reaccionar automáticamente, ante cambios inesperados y seguir funcionando correctamente, por eso, algunas situaciones que pueden tenerse en cuenta cuando se habla de adaptación en servicios web surgen debido a la degradación en la calidad, la saturación y a cambios en los contratos de los servicios.

Degradación en la calidad de los servicios: En el contexto de la adaptación, la degradación en la calidad de servicio refiere a la degradación de las propiedades de tiempo de respuesta y porcentaje de respuestas exitosas [47]. La saturación de servicios: En el contexto de la adaptación, la saturación de servicio refiere a aquella situación en la cual un servicio recibe más invocaciones de las que puede procesar un determinado periodo de tiempo [47]. Y los cambios en los contratos

de los servicios: En el contexto de la adaptación, el cambio de contrato refiere a la modificación del lenguaje de descripción del servicio web o WSDL por sus siglas en inglés (*Web Services Description Language*) de un servicio [47].

De acuerdo a las adaptaciones nombradas anteriormente existen trabajos como en [44] que presenta un enfoque donde se aprovechan las características de mediación presentes en el ESB, para proporcionar una adaptación dinámica de los servicios. El trabajo [48] describe la implementación del ESB a partir de la adaptación de sistemas basados en servicios, se implementa utilizando JBossESB. El trabajo [49] propone una arquitectura para la adaptación de servicios de acuerdo al contexto, haciendo uso del CEP para percibir y analizar la información relacionada con el usuario. El trabajo [50] presenta una solución que aprovecha los patrones de mediación del ESB a fin de adaptar los servicios a un contexto, aprovechando el uso del CEP para analizar los eventos recibidos de recursos externos y detectar las situaciones relevantes y de acuerdo a ello realizar la adaptación requerida.

Plataforma ESB adaptativa para sistemas basados en servicios

Este proyecto de maestría propone una plataforma ESB adaptativa la cual, basándose en las capacidades de mediación de los ESB, puede responder ante necesidades de adaptación en SOA de forma dinámica, automática y en tiempo de ejecución [47].

Para esto la solución presentada intercambia mensajes que ingresan al ESB, determina si requiere realizar alguna acción de adaptación para el servicio que se invoca y en dicho caso procesar el mensaje por medio de los flujos de adaptación. Estos flujos contienen los pasos de mediación necesarios para realizar las acciones de adaptación requerida. Los flujos de adaptación son seleccionados automáticamente y dinámicamente, dependiendo de la necesidad de adaptación concreta, En específico este trabajo se enfoca en servicios implementados como servicios web y las necesidades de adaptación surgen a partir de la degradación de la calidad de los servicios, por la saturación de los mismos o por los cambios en los contratos de los servicios.

La plataforma propuesta en este trabajo se basa, en patrones de mensajería e integración comúnmente soportados en los productos ESB, por lo que brinda una solución genérica y factible de aplicarse en la mayoría de estos productos.

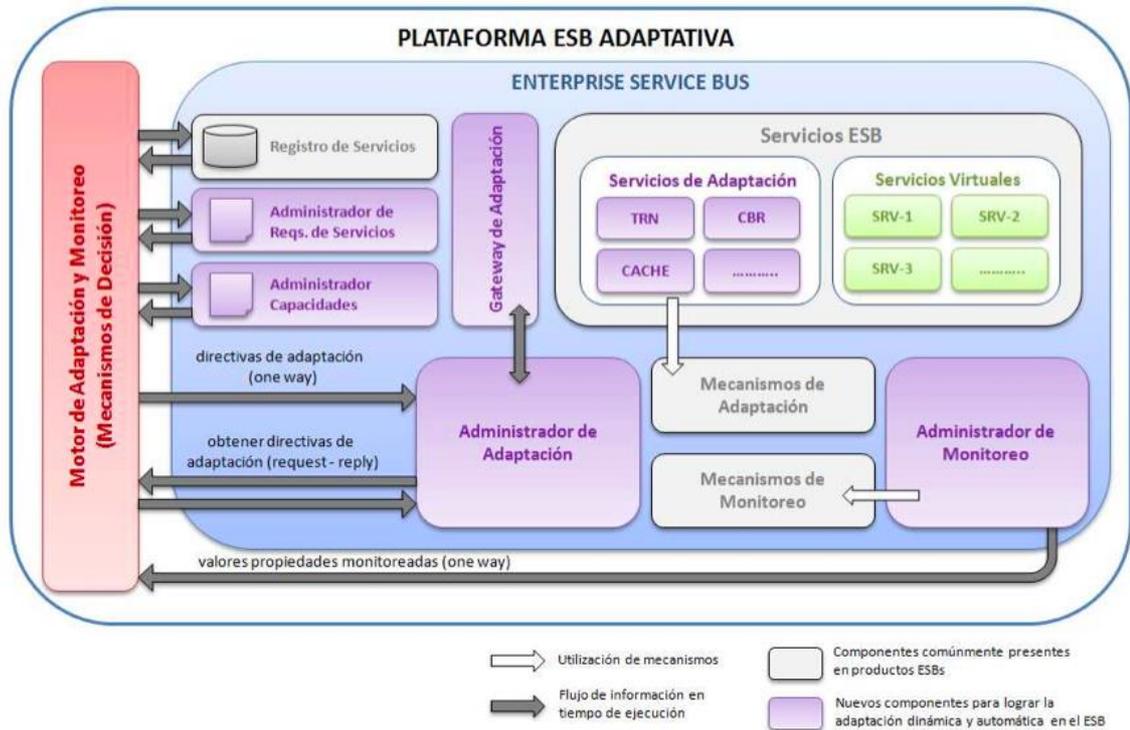


Figura 1. Arquitectura lógica plataforma ESB adaptativa, extraído de [47]

La Figura 1 presenta la arquitectura lógica de la solución presentada en este trabajo donde se encuentra el Motor de Adaptación y Monitoreo, los componentes internos del ESB (Gateway de adaptación y Monitoreo, Servicios de adaptación, Administrador de Adaptación, Administrador de Monitoreo, Administrador de Capacidades y Administrador de Requerimientos de Servicios) además del Registro de Servicios, Mecanismos de Monitoreo, Mecanismos de Adaptación y las principales interacciones y flujos de información entre ellos.

Event-Driven Integration Platform for Context-Adware Web Services

El trabajo [50] presenta una solución que aprovecha los patrones de mediación del ESB a fin de adaptar los servicios a un contexto, aprovechando el uso del CEP para analizar los eventos recibidos de recursos externos y detectar las situaciones relevantes y de acuerdo a ello realizar la adaptación requerida.

Para esto el trabajo muestra la construcción y presentación de servicios sensibles al contexto a través de ESB utilizando servicios virtuales, partiendo de servicios que no son sensibles al contexto. Para la construcción de estos servicios, el ESB aplica una lógica de adaptación utilizando sus capacidades de mediación. Las adaptaciones se generan automáticamente dependiendo de la situación en que se encuentra el usuario que invoca el servicio. Para determinar las situaciones, éstas deben ser definidas previamente en el motor CEP como eventos complejos, luego serán ejecutados dependiendo de la información del contexto que llega al ESB

desde distintas fuentes. La Figura 2 presenta el esquema general de la arquitectura propuesta.

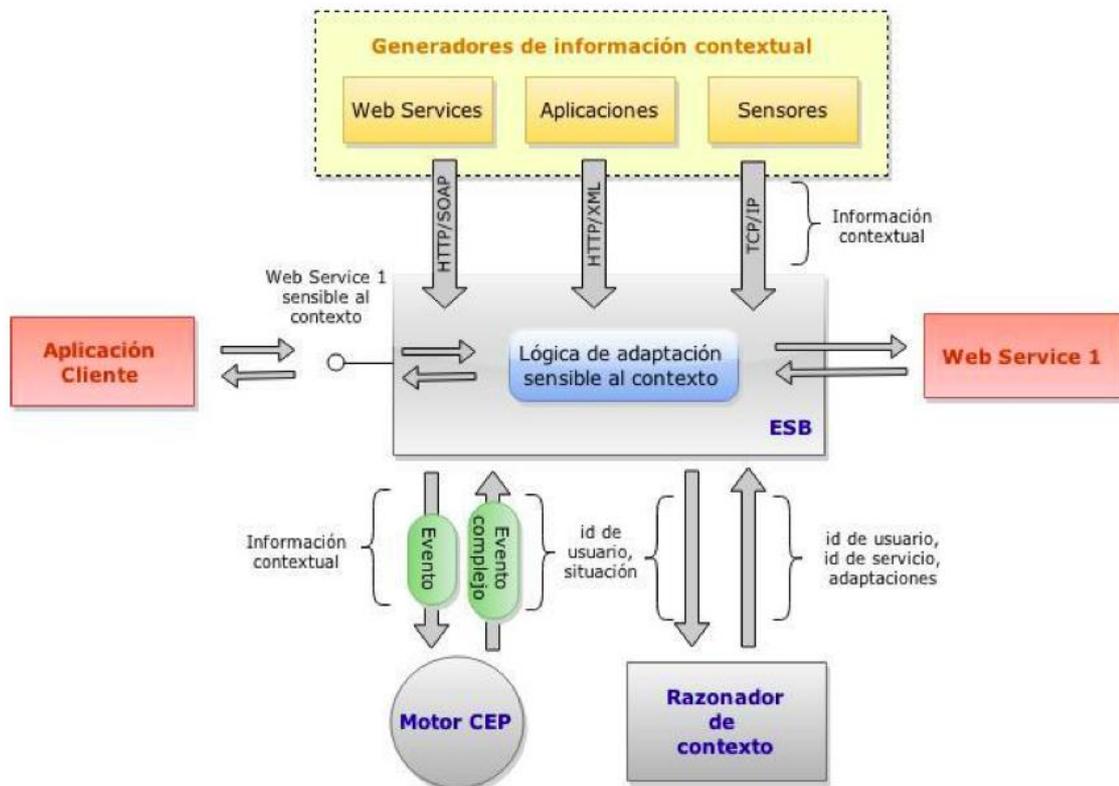


Figura 2. Arquitectura General Event-Driven Integration Platform for Context-Aware Web Services, extraído de [50]

Las fuentes de contexto proveen al ESB con distintos tipos de información contextual en forma de eventos (dispositivo móvil con GPS, sensores de ambiente, entre otros). El motor CEP recibe toda esta información a través del ESB y en base a las reglas definidas detecta situaciones complejas en las que los usuarios se encuentran.

El razonador de contexto recibe las situaciones en la que los usuarios se encuentran y en tiempo de ejecución genera las adaptaciones que sean necesarias para cada servicio que se encuentre configurado. Para esto el razonador de contexto debe tener establecidas las situaciones que afectan cada servicio y las adaptaciones que debe aplicar en cada caso.

Finalmente, el ESB recibe las adaptaciones requeridas para cada usuario y servicio, de tal manera que cuando se presente una invocación aplica las adaptaciones correspondientes mediante sus capacidades de mediación.

Sistemas de Alertas Tempranas para la Detección de Enfermedades

Los sistemas de alerta temprana son ampliamente utilizados en un sin fin de aplicaciones, una de ellas es la detección de enfermedades en seres humanos, animales o cultivos. La presente investigación está enfocada hacia las alertas relacionadas directamente con la detección de enfermedades en los cultivos, sin dejar de lado las metodologías, técnicas, estrategias y demás elementos utilizados para la detección de enfermedades en seres humanos, animales o en cualquier otro ámbito. Ocharan [51] afirma que los sistemas de alerta temprana incluyen tres elementos fundamentales: conocimiento y mapeo de amenazas, monitorización y pronóstico de eventos inminentes, procesos y difusión de alertas. La correcta combinación de estos tres elementos tiene como resultado un sistema de alerta temprana completo y confiable que brinde información oportuna y comprensible a las autoridades y la población. Ocharan destaca que los sistemas de alerta deben de estar centrados en la gente con el objetivo de permitir que las personas y las comunidades amenazadas tengan la posibilidad de actuar a tiempo y de forma apropiada, de manera que se reduzcan posibles pérdidas.

El trabajo [52] presenta un estudio de factibilidad con respecto a la utilización de datos climáticos recolectados automáticamente, con el fin de que estos sean procesados para lograr un sistema de alarma temprana contra el tizón tardío en los cultivos de papa. Para realizar este estudio se establecieron ocho parcelas experimentales donde se recolectaron los datos para el sistema, una vez analizados y procesados estos datos se obtuvo un calendario fijo con las fechas de alarma para la enfermedad, así como también, los tratamientos a realizar en los cultivos. El trabajo concluye con la factibilidad y aceptación de la implementación de los sistemas de alerta temprana para el control del tizón tardío en la papa. El trabajo [9] presenta un informe sobre la creación de un sistema de alerta temprana para la mitigación del impacto de la aparición del ojo de gallo en los cultivos de café. Este sistema se basa en modelos matemáticos que usan datos meteorológicos para predecir las condiciones favorables para el desarrollo de la enfermedad. Además, permite a los agricultores realizar aplicaciones de fungicidas en momentos oportunos, impidiendo con esto el desarrollo de la enfermedad.

Por otro lado, el trabajo [53] expone la importancia de los datos climáticos como insumo para obtener un exitoso sistema de alerta temprana para enfermedades en los cultivos. Concretamente, para que sirva de apoyo a la toma de decisiones con el fin de ayudar a los productores a determinar cuándo aplicar medidas de control para suprimir enfermedades en los cultivos. En este proceso se evalúan las ventajas y desventajas de los métodos más comunes para la recolección de

datos climáticos. Del mismo modo, se cuenta con sistemas de alerta temprana para enfermedades en seres humanos, este es el caso de [54], el cual expone un *software* utilizado para obtener y procesar datos satelitales, los cuales alimentan un sistema de alerta temprana, que toma como caso de estudio las enfermedades transmitidas por los mosquitos. Este trabajo que generó un *framework* que soporta la captura y procesamiento de múltiples recursos de datos haciendo al sistema flexible ante la disponibilidad de datos.

Brechas de Investigación

Finalmente, para obtener información detallada y concisa de cada trabajo que ha sido seleccionado y clasificado previamente, se elaboró una plantilla de extracción de datos por cada estudio, en esta plantilla se consignó la información principal como título, aportes y ventajas, así como las falencias que derivan al final en las brechas relacionadas con el problema de investigación.

En esta sección se abordan los principales vacíos y falencias de los trabajos que se presentaron anteriormente, de tal manera que se pueda establecer un punto de referencia para este trabajo investigativo.

La siguiente tabla expone las brechas identificadas de acuerdo a la clasificación de los trabajos relacionados.

Tabla 2. Brechas existentes. Fuente propia

Núcleo Temático	Enfoque	Brechas
Procesamiento de eventos complejos	[39] [40] [41] [42] [55] [56]	En los distintos trabajos revisados se evidencio que solo algunos de ellos integran CEP y ESB, además esta integración es realizada en dominios diferentes a la generación de alertas tempranas de enfermedades en los cultivos.
Bus de servicios empresariales y procesamiento de eventos complejos	[43] [44] [45] [48] [49] [50]	En su gran mayoría los trabajos revisados están enfocados en la adaptación de servicios utilizando distintos patrones de mediación del ESB. Sin embargo, no se evidencio el uso del CEP para la monitorización y adaptación de los servicios.
Sistemas de alerta	[9] [51] [52] [54]	Los trabajos revisados implementan sistemas de alertas tempranas en distintos entornos. A pesar de

temprana para la detección de enfermedades	[54] [57] [58]	ello, no contemplan la posibilidad de brindar información adicional al usuario a partir de las alertas tempranas de enfermedades generadas y los datos adicionales que el usuario pueda complementar.
--	----------------	---

Teniendo en cuenta lo anterior, este trabajo de investigación plantea abordar el problema de la adaptación de los servicios web de generación de alertas tempranas en cultivos mediante la integración del bus de servicios empresariales y el procesamiento de eventos complejos. Es decir, que se integrará el procesamiento de eventos complejos dentro del bus de servicios empresariales para detectar situaciones relevantes que indiquen los criterios bajo los cuales se debe realizar la adaptación de los servicios web. De esta manera, se propone personalizar los sistemas de alerta temprana de enfermedades en cultivos con el objetivo de que se ajusten a las características del usuario, generando las advertencias y adaptándolas mediante la integración del bus de servicios empresariales y el procesamiento de eventos complejos según la zona en la que se encuentre y la fenología del cultivo.

Resumen

En este capítulo se presentaron los conceptos teóricos en los que se basa el trabajo de grado como lo son el Procesamiento de Eventos Complejos (CEP), el Bus de Servicios Empresariales (ESB), Servicios Web, y la Adaptación de Servicios Web.

Posteriormente fueron expuestos los trabajos relacionados respecto al problema de investigación definido en el Capítulo 1, haciendo uso de una revisión de los trabajos que se consideraron más relevantes para el desarrollo de este trabajo, teniendo en cuenta los temas de interés, es decir el procesamiento de eventos complejos el bus de servicios empresariales y la adaptación de servicios web.

Capítulo 3

Adaptación de Servicios Web a través de un ESB y CEP

Tomando como base la revisión documental realizada en el capítulo anterior, este estudio plantea abordar el problema referente a la integración de CEP y ESB con el fin de adaptar los servicios web utilizados para la generación de alertas tempranas en los cultivos, es decir, si se cuenta con un conjunto de servicios web los cuales basados en la información climática recolectada generan distintas alertas de enfermedades en cultivos, a partir de estos servicios es posible adaptar la respuesta brindada a los usuarios, con el objetivo de incluir información relevante para ellos dependiendo de la situación en la que se encuentren y los datos complementarios que estos pudieren suministrar. Por ejemplo, si se presentan condiciones climáticas ideales para la generación de la roya en el café, es generada una alerta para esta enfermedad, la cual es emitida con un nivel o grado que puede ser bajo, medio o alto, de tal manera que el sistema ante esta situación debería adaptar la respuesta presentada al usuario con el objetivo de incluir datos adicionales importantes para él, como el proceso a seguir para manejar adecuadamente la enfermedad, los productos a utilizar, entre otros, de tal forma que el usuario pueda tomar decisiones ante la alerta que fue generada.

El desarrollo del proceso de construcción de la solución propuesta se basó en el planteamiento propuesto en los trabajos [47][50], en los cuales se realizan adaptación de servicios web utilizando las capacidades de mediación del ESB.

Descripción General

Dentro del análisis documental, los trabajos revisados orientan la adaptación de servicios web mediante las capacidades de mediación del ESB, pero dejan de lado el CEP, es por esto que la presente propuesta da solución a ese importante aspecto haciendo uso de la integración de ESB y CEP.

Los sistemas de alerta temprana normalmente producen notificaciones de manera general, es decir, que los avisos enviados para advertir sobre alguna situación de riesgo son emitidos a una población grande como a un país o una región. Por tal motivo, cuando se habla de alertas tempranas aplicadas al sector agrícola, se debe tener en cuenta que se necesita que estas alertas sean más específicas, de

tal manera, que incluyan características propias del lugar donde se encuentran sembradas, al igual que algunas características propias del cultivo. Por este motivo, la adaptación es un componente que puede contribuir dentro de los sistemas de alerta temprana ya que, gracias a su singularidad, permite tener en cuenta aspectos como el entorno o contexto del usuario, lo que hace posible que la alerta generada tenga en cuenta aspectos particulares relacionados con el lugar donde se encuentra sembrado el cultivo, además de las características fenológicas del mismo.

Por lo anterior, y con la idea de incluir la adaptación dentro de los sistemas de alerta temprana se plantea utilizar un elemento común para los dos, que es la monitorización; este elemento puede ser abordado por medio de la integración entre ESB y CEP debido a que estos componentes brindan; por un lado, un componente mediador por el cual viajan todos los mensajes y pueden ser transformados, enrutados, invocados, entre otros. Por otro lado, se encuentra el componente que se encargaría de analizar, identificar y correlacionar los mensajes que son intercambiados por los distintos componentes, con el objetivo de identificar eventos relevantes que sean de interés para los usuarios, proporcionando la adaptación de los sistemas de alerta temprana soportada en la integración ESB y CEP a partir de las necesidades de los usuarios.

Por consiguiente, contar con las capacidades de adaptación para responder adecuadamente ante las necesidades del usuario de manera automática, dinámica, en tiempo de ejecución y seguir funcionando correctamente, es un aspecto clave en los diversos sistemas basados en servicios y en especial para los sistemas de alerta temprana, ya que por lo general estos sistemas no cuentan con los componentes que proveen de forma completa y nativa soluciones para la adaptación automática dinámica en tiempo de ejecución [47], dado el rol del ESB y sus capacidades de mediación este componente resulta ideal para efectuar acciones de adaptación. Por lo tanto, siguiendo con el planteamiento propuesto en [48] donde la adaptación se realiza dentro del ESB usando las capacidades de mediación que generalmente están presentes en todos los ESB, se propone el modelo presentado en la figura 3.

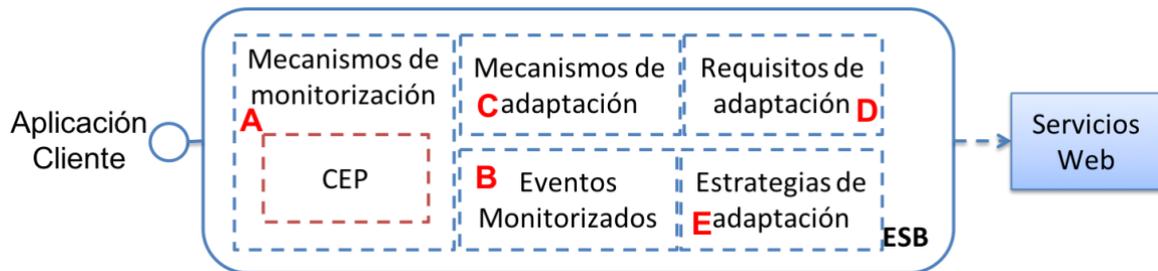


Figura 3. Modelo CEP-ESB-Adaptación. Fuente propia

En la Figura 3 se expone la propuesta de adaptación diseñada dentro de ESB-CEP, en ella se observan los elementos presentados en el marco de adaptación y monitorización propuesta por el proyecto S-Cube [59], el cual aborda principios, técnicas y metodologías para la adaptación y monitorización de sistemas basados en servicios, definiendo un marco conceptual de adaptación y monitorización que generaliza y extiende la adaptación SOA. Elementos que fueron ajustados a las características del ESB para incluir al CEP dentro del mismo, donde cada componente será especificado en la siguiente sección.

Componentes

A continuación, se describen los componentes que se presentaron en la Figura 3:

Mecanismos de monitorización (A)

Módulo que representa cualquier mecanismo que pueda utilizarse para comprobar si la situación actual corresponde a la esperada. Por ejemplo, un mecanismo que monitoriza los parámetros tenidos en cuenta en la invocación del servicio, adición de nuevas operaciones, entre otros. Se propone que este componente sea abordado mediante el procesamiento de eventos complejos (CEP) encargado de monitorizar los eventos de interés.

Eventos monitoreados (B)

Componente que representa los eventos que brindan información relevante sobre la ejecución, evolución o contexto y son detectados a través de los mecanismos de monitorización. Estos eventos representan el hecho de que existe una diferencia con respecto al estado, funcionalidad o ambiente esperado. Por ejemplo, los parámetros ingresados al invocar una operación son superiores a los esperados.

Mecanismos de adaptación (C)

Componente que contiene las técnicas y facilidades proporcionados por el ESB para realizar las estrategias correspondientes de acuerdo a los requisitos de adaptación.

Requisitos de adaptación (D)

Elemento que representa la necesidad de modificar un servicio o un sistema con el objetivo de eliminar la diferencia que existe entre el estado actual (previsto) y el esperado. Por ejemplo, incluir en las operaciones parámetros relacionados con la ubicación del usuario. Los requerimientos de adaptación pueden incluir requisitos de confiabilidad o exactitud funcional, optimización, interoperabilidad y usabilidad, entre otros.

Estrategias de adaptación (E)

Módulo encargado de definir las posibles maneras de satisfacer los requisitos de adaptación de acuerdo a la situación actual. Por ejemplo, invocar una operación equivalente que tenga en cuenta los nuevos parámetros. Puede ocurrir que se utilicen distintas estrategias para la misma situación, lo cual requiere que se cuente con mecanismos de decisión que permitan seleccionar la estrategia más adecuada.

Flujo de ejecución

Los componentes descritos anteriormente, participan en el proceso de adaptación de la siguiente manera:

1. Los mecanismos de monitorización se encargan de detectar las situaciones de interés, en él se incluyen los elementos que contribuyen a la identificación de los eventos importantes para la adaptación (eventos monitorizados).
2. Se identifican los eventos monitorizados, decir los eventos que son considerados de gran importancia ya que a partir de ellos se descubren la necesidad de adaptación.
3. Una vez identificada una necesidad de adaptación, los mecanismos de adaptación a partir de las características básicas de los ESB (ruteo, transformación, mediación, entre otros) brindan los posibles elementos que servirán para cumplir con la adaptación identificada.
4. Los requisitos de adaptación los cuales plasman la adaptación a partir de los eventos monitorizados y teniendo en cuenta los mecanismos de adaptación disponibles.
5. Finalmente, las estrategias de adaptación utilizan los mecanismos de adaptación a partir de las capacidades del ESB en cuanto al manejo de los mensajes.

En resumen, el proceso general propuesto en la Figura 3 se ejecuta de la siguiente manera:

- A. La información relevante se recolecta a través de mecanismos de monitorización.

- B. Se detectan eventos críticos.
- C. Se identifica la necesidad de adaptación.
- D. Se identifica una forma adecuada de realizar la adaptación (selección de estrategia de adaptación).
- E. Se ejecuta la adaptación utilizando los mecanismos de adaptación disponibles [59].

La propuesta descrita anteriormente, se encuentra representada de manera general, de tal manera que se podría aplicar a cualquier entorno, ya que hasta aquí no se han tenido en cuenta aspectos propios del caso a tratar. En este trabajo de grado la propuesta será aplicada en la generación de alertas tempranas para enfermedades en los cultivos, teniendo como caso particular la roya en el cafeto, caso que será explicado en el capítulo 5.

Criterios de adaptación

En el desarrollo del presente trabajo de grado se entiende como criterios de adaptación a las reglas que establecen el momento a partir del cual se comienza el proceso de adaptación por parte del ESB, en este caso particular este momento está dado por los eventos que son analizados por el motor de procesamiento de eventos complejos. Por ejemplo, cuando el motor CEP detecta un evento complejo relacionado con las condiciones climáticas que se encuentra afín con la generación de la roya en el café, se convierte en un criterio de adaptación para el ESB, de tal manera que una vez se presente un evento complejo de interés comience el proceso de adaptación a partir de los datos que han sido detectados.

Directivas de adaptación

En el desarrollo del presente trabajo de grado basado en [50] se entiende como una directiva de adaptación a las capacidades de medicación que deben ser utilizadas para lograr una adaptación, lo que significa que por cada una de las adaptaciones que el ESB pretenda realizar, debe existir una directiva que le permita al ESB alcanzar la adaptación deseada. Teniendo en cuenta que una directiva puede utilizar una o varias capacidades de mediación del ESB para lograr un objetivo, por ejemplo, se desea realizar una adaptación de los servicios web de alerta de la enfermedad de la roya cuando se encuentre en un nivel medio, de tal manera que cuando se presente este nivel de alerta se adapta la respuesta enviada al usuario adicionando información sobre los productos químicos a utilizar, tiendas de insumos disponibles en la zona, entre otros.

Para realizar la adaptación de los servicios web de acuerdo a la situación en la que se encuentra el usuario, se deben contar con las siguientes configuraciones:

- La situación que afecta cada servicio.
- Las adaptaciones que deben aplicarse en cada caso.
- El momento a ser aplicadas las adaptaciones, ya sea en la invocación o en la respuesta del servicio.
- La prioridad de las adaptaciones para cada servicio, en caso que haya más de una adaptación configurada para un servicio.

Cada adaptación debe contar con la capacidad de mediación a aplicar e información que utilizará, tal como se puede apreciar en la tabla 3.

Tabla 3. Directivas de adaptación. Fuente propia

Servicio	Criterio (OUT)	Directiva de adaptación	Momento
Coordenadas (latitud, longitud)	Zona	Convertir las coordenadas (latitud, longitud) en zonas cafeteras colombianas	Request
WS_Alerta	Grado = TI1	Enriquecer el mensaje añadiendo información de los productos y equipos a utilizar. Mensaje original: Enriquecer(WS_Alerta, Fecha, Zona) Mensajes adicionales: Enriquecer(WS_Recomendacion, Grado, Enfermedad, Ubicación) Enriquecer(WS_Productos, Fecha floración, Enfermedad) Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad)	Response
WS_Alerta	Grado = TI2	Enriquecer el mensaje añadiendo información de los productos, equipos de aspersión, tiendas cercanas y guías. Mensaje original: Enriquecer(WS_Alerta, Fecha,	Response

		<p>Zona)</p> <p>Mensajes adicionales:</p> <p>Enriquecer(WS_Recomendacion, Grado, Enfermedad, Ubicación)</p> <p>Enriquecer(WS_Productos, Fecha floración, Enfermedad)</p> <p>Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad)</p> <p>Enriquecer(WS_TiendasInsumos, Ubicación)</p> <p>Enriquecer(WS_GuiaEnfermedad, Enfermedad)</p>	
WS_Alerta	Grado = TI3	<p>Enriquecer el mensaje añadiendo información de los productos, equipos de aspersión, tiendas cercanas, guías, expertos y comparación de costos.</p> <p>Mensaje original:</p> <p>Enriquecer(WS_Alerta, Fecha, Zona)</p> <p>Mensajes adicionales:</p> <p>Enriquecer(WS_Recomendacion, Grado, Enfermedad, Ubicación)</p> <p>Enriquecer(WS_Productos, Fecha floración, Enfermedad)</p> <p>Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad)</p> <p>Enriquecer(WS_TiendasInsumos, Ubicación)</p> <p>Enriquecer(WS_GuiaEnfermedad, Enfermedad, Cultivo)</p> <p>Enriquecer(WSExperto, Ubicación, Enfermedad)</p> <p>Enriquecer(WSCmpCosto, Productos, Enfermedad)</p>	Response

WS_Experto	Lista de expertos (Expertos con ubicación y enfermedad)	Enriquecer el mensaje añadiendo información de productos, equipos de aspersión y las tiendas de insumos disponibles. Mensaje original: Enriquecer(WS_Experto, Ubicación, Enfermedad) Mensajes adicionales: Enriquecer(WS_Productos, Ubicación, Fecha de Floración, Enfermedad) Enriquecer(WS_EquiposAsp, Ubicación, Enfermedad) Enriquecer(WS_TiendasInsumos, Ubicación)	Response
WS_Experto	Lista de expertos roya (Expertos en ubicación específica)	Filtrar los mensajes que contengan expertos en ubicaciones diferentes a la ubicación actual. Filtrar(WS_Experto, Ubicación)	Response
WS_Experto	Lista de expertos en ubicación (Expertos en determinada ubicación)	Filtrar los mensajes que contengan expertos en una determinada enfermedad. Filtrar(WS_Experto, Enfermedad)	Response
WS_Productos	Lista de productos y precio	Enriquecer los mensajes añadiendo información de los equipos de aspersión disponibles y la comparación de costos. Mensaje original: Enriquecer(WS_Productos, Ubicación, Fecha Floración,	Response

		<p>Enfermedad)</p> <p>Mensajes adicionales: Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad) Enriquecer(WS_CmpCosto, Productos, Enfermedad)</p>	
WS_EquipoAsp	Lista de equipos	<p>Enriquecer los mensajes con información adicional de la comparación de costos y las tiendas disponibles.</p> <p>Mensaje original: Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad)</p> <p>Mensajes adicionales: Enriquecer(WS_TiendasInsumos, Ubicación) Enriquecer(WS_CmpCosto, Productos, Enfermedad)</p>	Response
WS_EquipoAsp (requiere contar con parámetro de presupuesto)	Lista de equipos	<p>Filtrar los mensajes cuyos precios superen el presupuesto estimado del agricultor</p> <p>Filtrar(WS_EquipoAsp, Ubicación, Presupuesto, Enfermedad)</p>	Response
WS_TiendasInsumos	Lista de tiendas	<p>Enriquecer los mensajes adicionando información de los productos y los expertos.</p> <p>Mensaje original: Enriquecer(WS_TiendasInsumos, Ubicación)</p> <p>Mensajes adicionales: Enriquecer(WS_Productos, Ubicación, Fecha Floración, Enfermedad) Enriquecer(WS_Experto, Ubicación, Enfermedad)</p>	Response

WS_GuiaEnfermedad	Guía para el manejo de la enfermedad (Consejos)	Filtrar los mensajes que contengan las guías dependiendo del grado de alerta Filtrar(WS_GuiaEnfermedad, Enfermedad, Cultivo)	Response
WS_CmpCosto	Lista de precios por producto	Enriquecer los mensajes con información adicional de las tiendas de insumos. Mensaje original: Enriquecer(WS_CmpCosto, Productos, Enfermedad) Mensajes adicionales: Enriquecer(WS_TiendasInsumos, Ubicación)	Response
WS_CmpCosto (requiere contar con parámetro de presupuesto)	Lista de precios por producto	Filtrar los mensajes que contengan la comparación de los productos con precios superiores a el presupuesto del usuario Filtrar(WS_CmpCosto, Productos, Enfermedad)	Response
WS_Recomendación	Recomendación	Enriquecer los mensajes con información adicional de productos, equipos de aspersión, tiendas y guías. Mensaje original: Enriquecer(WS_Recomendación, Grado, Ubicación) Mensajes adicionales: Enriquecer(WS_Productos, Ubicación, Fecha Floración, Enfermedad) Enriquecer(WS_EquipoAsp, Ubicación, Enfermedad) Enriquecer(WS_TiendasInsumos, Ubicación) Enriquecer(WS_GuiaEnfermedad, Enfermedad, Cultivo)	Response

Una vez realizadas las directivas de adaptación de cada servicio y situación presentadas en la Tabla 3, cuando el ESB recibe la información de los generadores de eventos, este realiza automáticamente las adaptaciones necesarias de acuerdo a la situación dada. Por ejemplo, si el ESB identifica que hay una alerta de roya con nivel medio, y adicionalmente el usuario ha proporcionado la ubicación en la que se encuentra, se podrían realizar recomendaciones sobre los productos químicos a utilizar en la zona donde se encuentra, además, brindar información sobre los expertos más cercanos, entre otros, suponiendo que anteriormente fueron configuradas las directivas de adaptación que permiten manejar la situación descrita.

El ESB debe aplicar las adaptaciones de acuerdo a los criterios de adaptación que fueron definidos previamente, de tal manera que la información que recibe el ESB debe ser tratada como directivas de adaptación a aplicarse en el momento en el que se realice una invocación al servicio por parte del usuario.

Patrones de adaptación

Luego de definir las directivas de adaptación que serán utilizadas por el ESB, se procede a identificar las situaciones en las cuales reaccionará el motor CEP, es decir los eventos complejos. Estos eventos partirán de eventos simples relacionados con algunas variables climáticas de interés, que luego se convertirán en eventos complejos. Cuando se presenten eventos simples en un intervalo de treinta días (30), junto con valores en determinadas variables climáticas que alcancen o superen los umbrales definidos se habrá identificado un evento complejo. Por ejemplo, un evento simple sería el valor de la temperatura, y un evento complejo sería que el promedio de temperatura sea superior a veintitrés grados centígrados (23°) en un periodo de treinta días (30).

A continuación, se listan los eventos complejos a partir de las condiciones climáticas que fueron tenidos en cuenta durante el desarrollo de este trabajo, estas condiciones fueron plateadas por Lasso en [60]:

Tabla 4. Patrones de generación de roya. Fuente propia

Patrón de generación	Variables climáticas	Descripción
T11	<ul style="list-style-type: none"> - Promedio de humedad relativa nocturna menor o igual a 6.35 - Temperatura máxima menor o 	Tasa de infección de roya número uno. Indica la tasa de infección de

	<p>igual a 22.76</p> <ul style="list-style-type: none"> - Temperatura mínima menor o igual a 14.48 	la roya en el café más baja que se puede presentar.
TI2	<ul style="list-style-type: none"> - Promedio de humedad relativa nocturna menor o igual a 6.35 - Promedio de temperatura en las horas de la noche menor o igual a 14.56 - Temperatura mínima menor o igual a 14.48 	Tasa de infección de roya número dos. Indica la tasa de infección de la roya en el café de un nivel intermedio.
TI3	<ul style="list-style-type: none"> - Sombra mayor que cero - Promedio de precipitación menor o igual a 0.57 - Promedio de humedad relativa nocturna mayor a 6.35 	Tasa de infección de roya número tres. Indica la tasa de infección de la roya en el café más alta que se puede presentar.

Para modelar los eventos complejos identificados en la Tabla 4 se utilizó la herramienta MEdit4CEP [61], la cual gráficamente permite la definición de los eventos simples y complejos, como se puede apreciar en las Figuras 4-8. En ellas se presenta un evento simple llamado clima que contiene las variables climáticas de interés (sombra, precipitación horas de humedad relativa diurnas, horas de humedad relativa nocturnas, temperatura). A partir de este evento se derivarán una serie de eventos complejos, los cuales teniendo en cuenta algunas variables presentes en los eventos simples, las operaciones que se pueden efectuar sobre ellas (promedio, máximos y mínimos, entre otros) además de un rango de tiempo generan un evento complejo como se puede apreciar en la Figura 5. Finalmente, partiendo de los eventos identificados anteriormente, se definen una serie de umbrales que deben de cumplir las variables climáticas de interés luego de aplicar sobre cada una de ellas alguna de las operaciones disponibles, con el objetivo de establecer algunos de los patrones que permitirán identificar el tipo de alerta que se está presentando como lo ilustran las Figuras 6, 7, 8. En este caso particular se han realizado tres patrones para la identificación de los tipos de alerta para la roya en el café que pueden: tasa de infección uno (“TI1”), tasa de infección dos (“TI2”) y tasa de infección tres (“TI3”), siendo “TI1” el nivel más bajo y “TI3” el más alto.

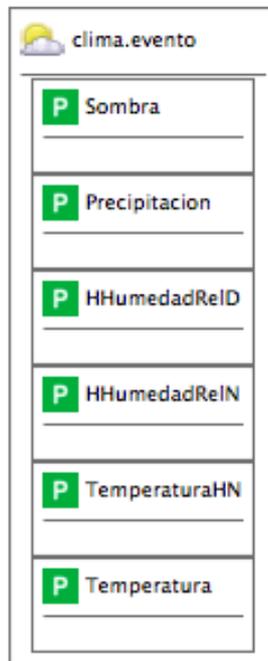


Figura 4. Evento simple. Fuente propia

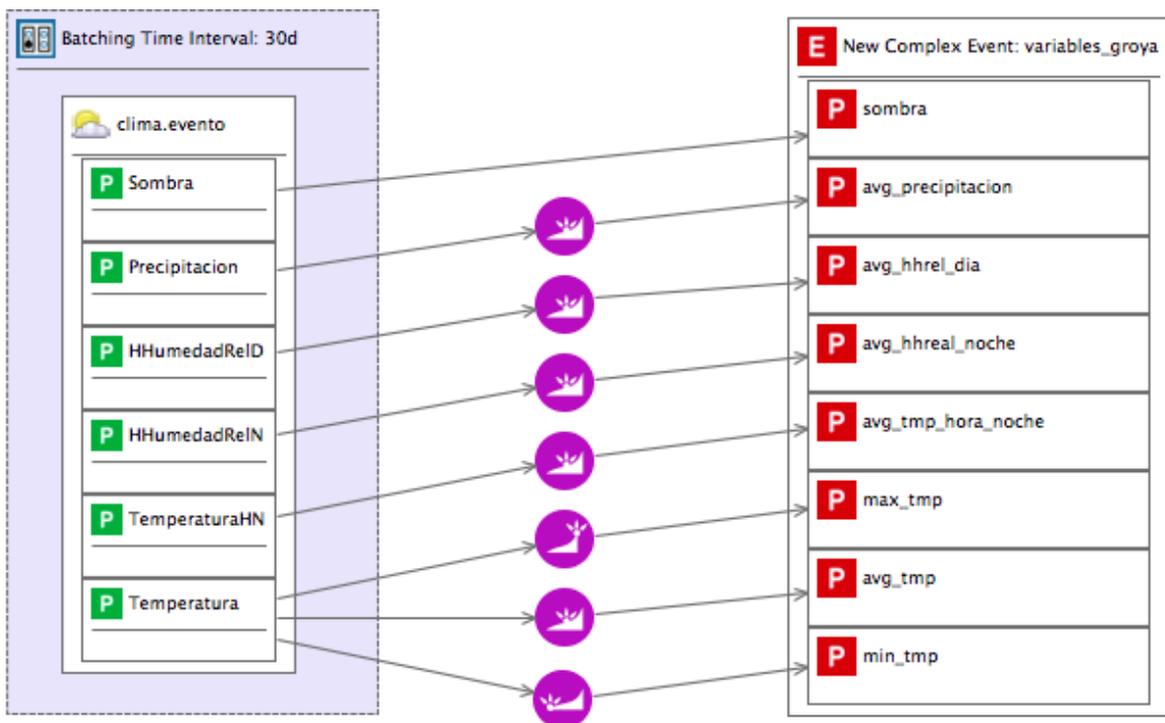


Figura 5. Evento complejo intervalo 30 días. Fuente propia

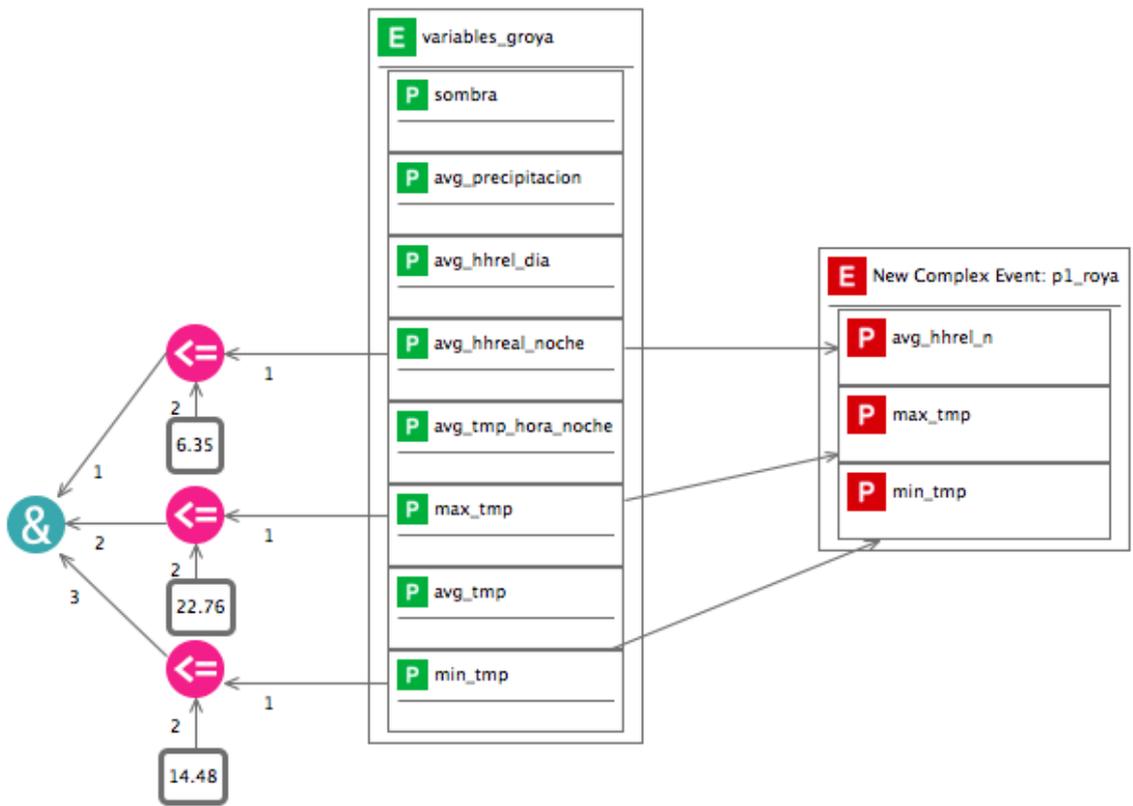


Figura 6. Evento complejos generación de alerta tipo T11. Fuente propia

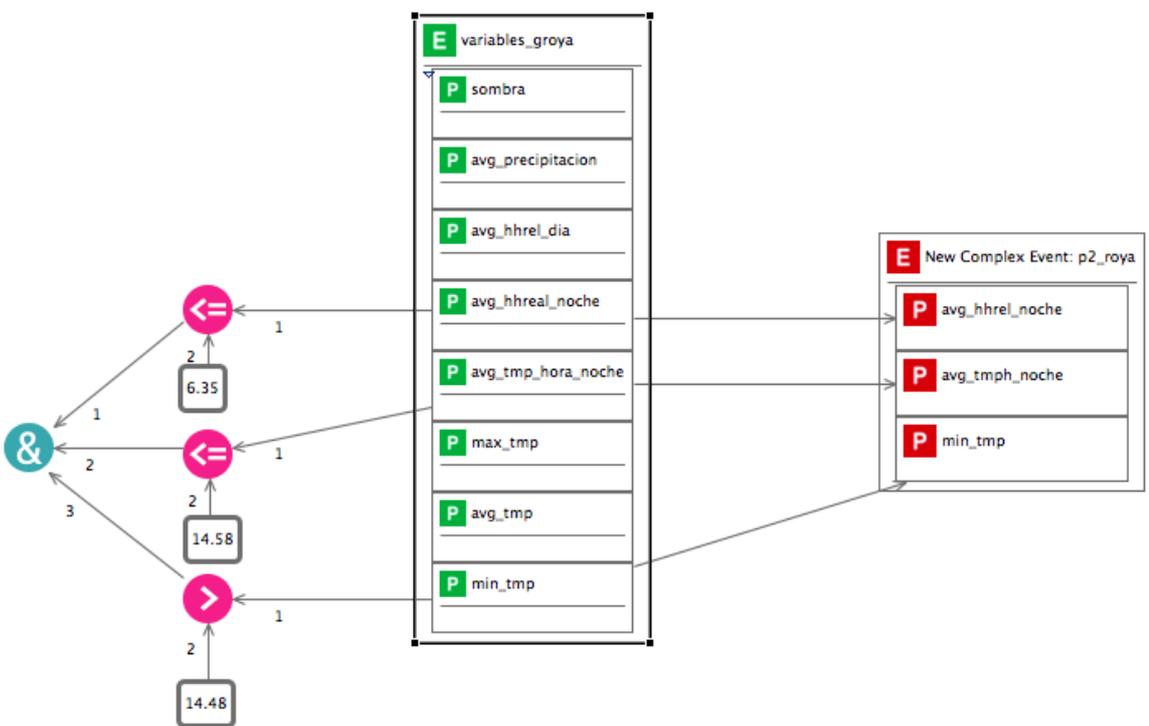


Figura 7. Evento complejo alerta tipo T12. Fuente propia

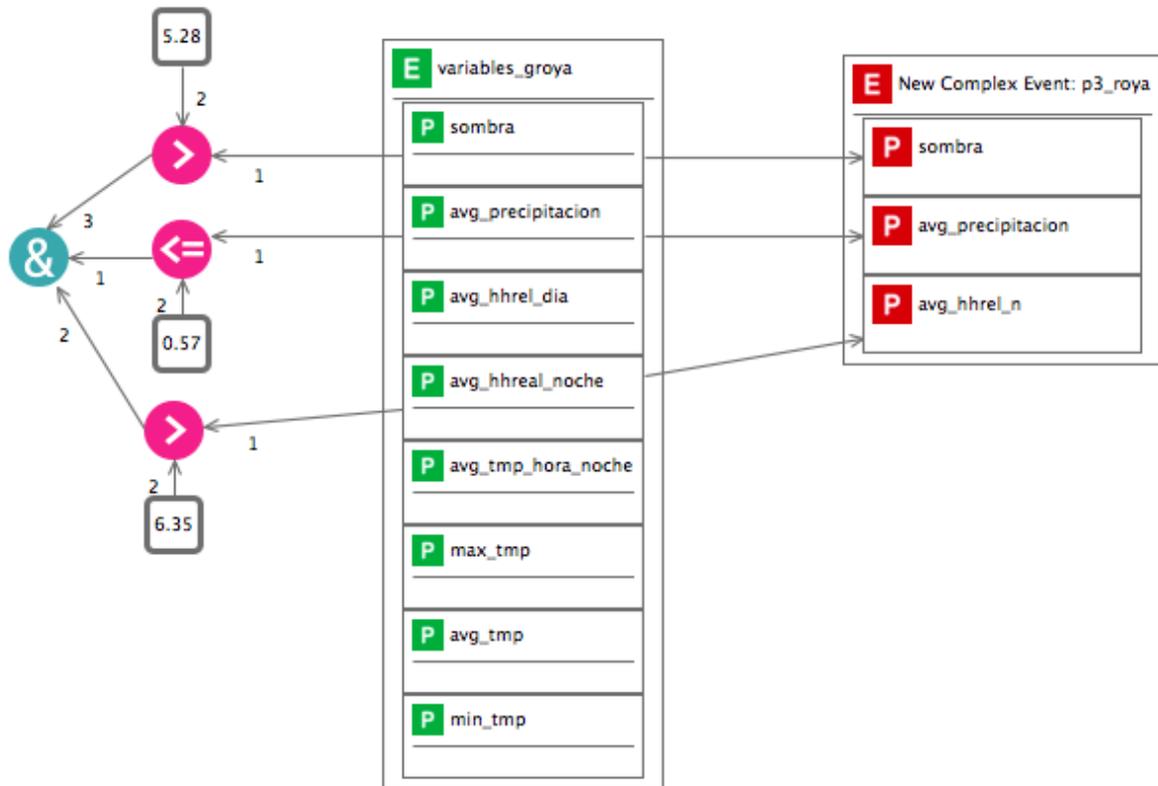


Figura 8. Evento complejo alerta tipo TI3. Fuente propia

En la Figura 6 se aprecia el patrón para la generación de alertas para la roya en el café tipo uno o “TI1”; en él se muestran las variables que inciden en la generación de la alerta junto con los umbrales en los que se presenta. Una vez disparado este tipo de alerta el ESB recibe esta información y de acuerdo a las directivas de adaptación que fueron definidas previamente se realiza una recomendación al usuario. Por ejemplo, el motor CEP detecta un evento complejo relacionado con una alerta de tipo 1 o “TI1” y lo envía al ESB, este siguiendo las directivas de adaptación definidas procede a realizar la adaptación requerida que puede ser un enriquecimiento de la información presentada al usuario o un filtrado de la misma según sea el caso, para finalmente entregarle un mensaje enriquecido al usuario con toda la información adicional como respuesta a la petición realizada. De la misma manera, en la Figura 7 se presenta el patrón para la generación de alertas para la roya en el café tipo dos “TI2”, en él se puede apreciar las variables que inciden en la generación de la alerta junto con los umbrales en los que se presenta, para realizar la recomendación al usuario se utiliza el proceso que fue descrito anteriormente. Finalmente, la Figura 8 presenta el último patrón para la generación de alertas para la roya en el café tipo tres o “TI3”, igual que en los casos anteriores se puede apreciar las variables que inciden en la generación de la alerta junto con los umbrales en los que se presenta.

Resumen

En este capítulo fue presentada la propuesta conceptual que permite realizar la adaptación de los servicios web de la generación de alertas para los cultivos, dentro de él se presentó una descripción de la propuesta, seguida de los criterios y directivas de adaptación y finalmente los elementos que permitieron identificar los patrones de eventos de interés a partir de los cuales se realiza el proceso de adaptación.

Con respecto a los criterios y directivas de adaptación se planteó la Tabla 3 en la cual se encuentran los casos puntuales en los cuales se pretende realizar un proceso de adaptación, los cuales están acompañados de la directiva planteada y el momento en el cual será invocado.

Para los patrones de eventos la Tabla 4 presenta las distintas situaciones que son de interés para la generación de la enfermedad de la roya en un cultivo de café, teniendo en cuenta que estas situaciones se encuentran relacionadas con el comportamiento del clima. Partiendo de estas se representan cada una de estas situaciones como modelos donde se muestra la relación entre eventos simples y como estos se convierten en eventos complejos.

Capítulo 4

Solución Propuesta

En el trabajo de grado fueron realizadas diferentes pruebas para determinar la adaptación de los servicios web para la generación de alertas tempranas de enfermedades. El proceso fue dividido en dos fases, el primero consistió en determinar los eventos simples y complejos junto con los patrones que serían detectados por el CEP a partir de las condiciones climáticas. En la segunda fase fue definido el mecanismo que se utilizará para integrar CEP y ESB de tal manera que una vez fuera detectado algún evento de interés el ESB realizará la adaptación correspondiente teniendo en cuenta las directivas que fueron definidas inicialmente para cada caso.

A continuación, se detalla el proceso de desarrollo del prototipo que implementa el mecanismo de adaptación propuesto.

Descripción General

En esta sección se presenta una descripción general de la solución propuesta en este trabajo de grado. Para empezar, se propone abordar la adaptación de los servicios web para la generación de alertas tempranas de enfermedades de acuerdo a la ubicación y la información del cultivo brindados por el usuario. Para esto, teniendo en cuenta que los servicios que han sido desarrollados para identificar las condiciones que favorecen la aparición de la roya en el café, dentro de los trabajos [62] [60], se propone el esquema propuesto en la Figura 9.

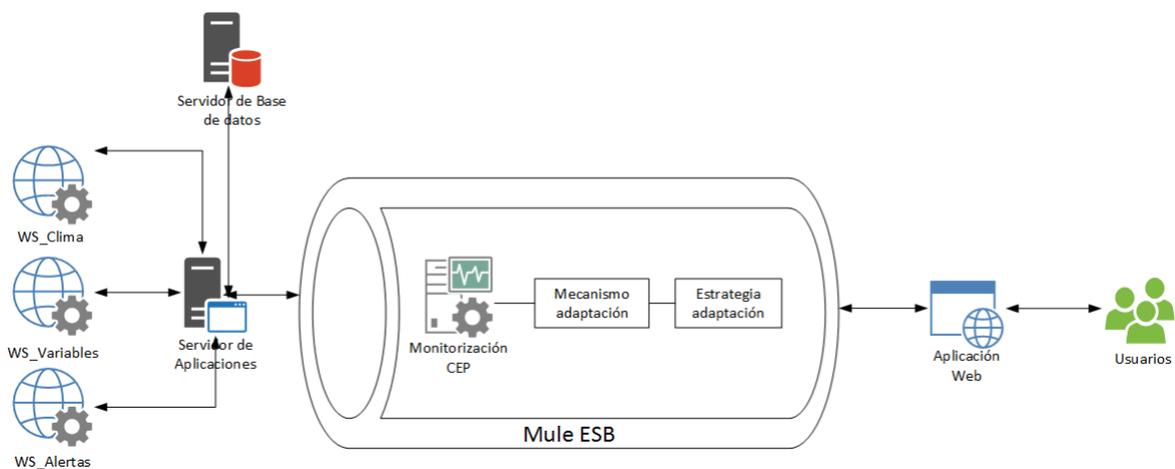


Figura 9. Diagrama general propuesto CEP-ESB. Fuente propia

En la Figura 9 se encuentran los principales elementos tenidos en cuenta para realizar la adaptación de los servicios de generación de alertas tempranas de enfermedades en cultivos mediante la integración del ESB y CEP, los elementos que componen la propuesta son:

- Servicios web encargados de los datos climáticos: estos servicios web son todos los que brindan información relacionada con el comportamiento del clima, donde se encuentra los servicios web que han sido desarrollados en el proyecto AgroCloud para detectar las condiciones climáticas favorables para la aparición de la roya en el cafeto, así como también los servicios web que se encargan de generar alertas de posibles brotes de roya a partir de la información climática (WS_DClima, WS_VClimaticas, WS_Alertas).
- Servidor de aplicaciones: en él se encuentran alojados todos los servicios web que han sido desarrollados, como, por ejemplo: los servicios utilizados para obtener información climática, gestión de alertas, base de datos, entre otros.
- Servidor de bases de datos: en él se encuentran alojadas las bases de datos encargadas de manejar toda la información del sistema como, por ejemplo, información climática, localizaciones, entre otras.
- MuleESB: bus de servicios empresariales (mediador) por el cual viajan todos los mensajes que van desde los distintos servicios hasta los servidores y aplicaciones. Dentro de este elemento se realizaría todo lo relacionado con la adaptación.
- CEP: motor de procesamiento de eventos complejos encargado de realizar la monitorización de las situaciones relevantes presentadas en los casos de estudio, con el fin de encontrar todos los eventos que sean significativos y en los cuales se requiera realizar una adaptación, de acuerdo a la información suministrada por el usuario y sus necesidades.
- Mecanismos de adaptación: relaciona las actividades de monitorización con las actividades de adaptación disponibles dentro del ESB. Es la manera de identificar que un evento monitorizado corresponde a una situación que requiere una adaptación.
- Estrategias de adaptación: definen los posibles procedimientos para satisfacer los requerimientos de adaptación de acuerdo a la situación actual. Por ejemplo, invocar una operación equivalente que tenga en cuenta los nuevos parámetros que han sido ingresados por el usuario.

Arquitectura General

En base a la solución propuesta en [50], el ESB puede ser utilizado como mecanismo de adaptación gracias a sus capacidades de mediación, de tal manera que las adaptaciones pueden ser realizadas dependiendo de la situación en la que se encuentre el usuario que invoca el servicio. Para detectar las situaciones donde se requiera realizar algún tipo de adaptación se utiliza el motor CEP, el cual requiere contar con una declaración previa de las situaciones de interés definidas como eventos complejos. Estas situaciones son disparadas de acuerdo a los datos entregados por los generadores de información que llegan al ESB desde distintas fuentes. La Figura 10 presenta un esquema general de la arquitectura propuesta.

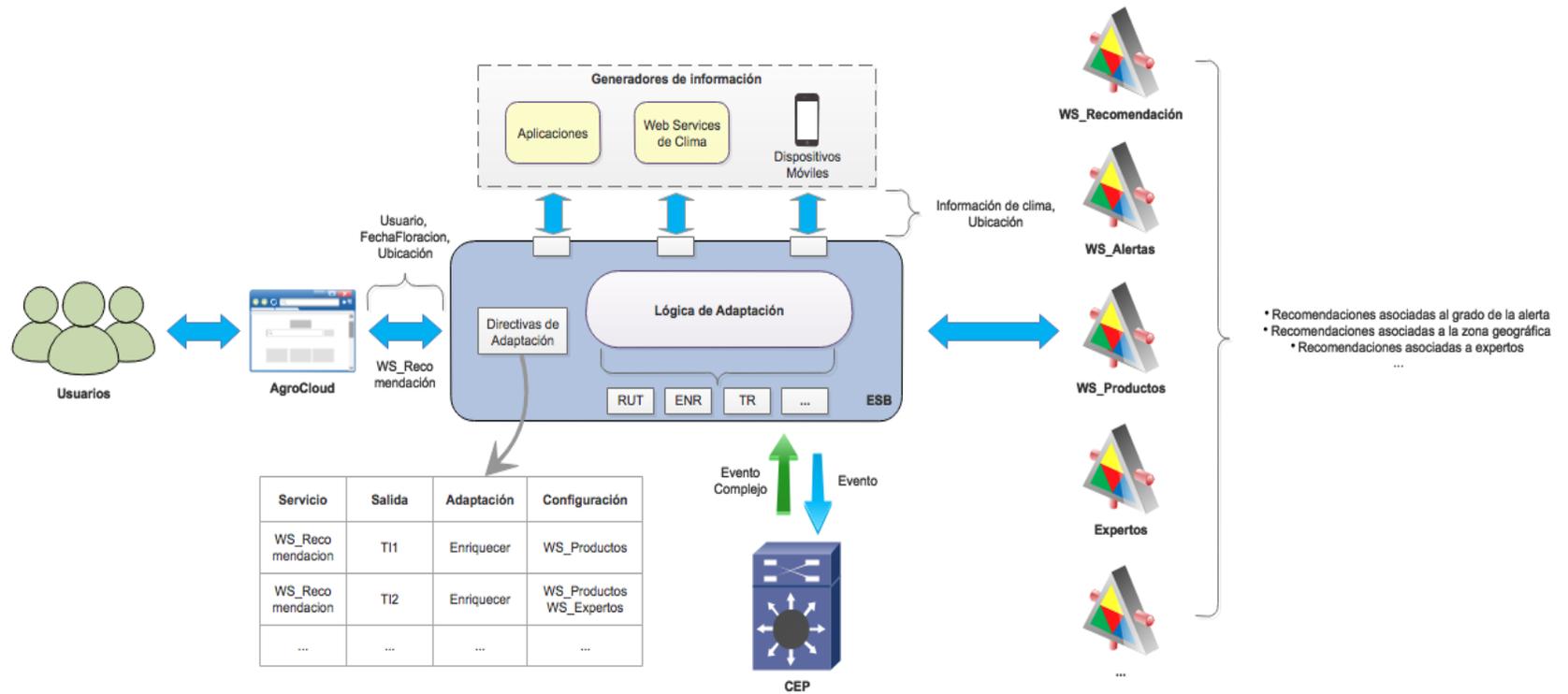


Figura 10. Arquitectura general, basada en [50]

Los generadores de información proveen al ESB con diferentes tipos de información en forma de eventos. Por ejemplo, datos climáticos, localización geográfica del usuario o distintos sensores, entre otros. El motor CEP recibe toda esta información a través del ESB y en base a las reglas que han sido definidas detecta las situaciones complejas (situaciones de interés relacionadas con la generación de la roya en el café) para los usuarios. Por ejemplo, basado en la información climática (temperatura, humedad relativa, pluviosidad, entre otros) presentadas en los últimos treinta días (30), el motor puede inferir el nivel de riesgo de generación de la enfermedad de la roya en el café (“TI1”, “TI2” o “TI3”).

Las directivas de adaptación reciben las situaciones en las que los usuarios se encuentran y en tiempo de ejecución realiza la adaptación necesaria para cada servicio que se encuentre disponible. Para esto, se debe establecer las situaciones que afectan a cada servicio mediante los criterios de adaptación, de tal manera que en él se encuentran las adaptaciones que deben ser aplicadas en cada caso. Por ejemplo, si un servicio retorna el nivel de alerta para la enfermedad de la roya, una adaptación puede ser retornar los expertos y los lugares donde puede abastecerse de los productos químicos para realizar el control de la enfermedad.

De esta manera, el ESB recibe las adaptaciones y servicios requeridos para cada caso, de tal manera que cuando llegue la invocación debe aplicar la o las adaptaciones correspondientes utilizando las capacidades de mediación. Continuando con el ejemplo anterior, se desea aplicar una adición para agregar a la respuesta entregada al usuario los distintos expertos en la enfermedad y el cultivo sembrado, así como también pueden ser adicionados productos químicos para el manejo y control de la enfermedad que en este caso particular sería la roya en el café.

Generadores de Información

La información que llega al ESB es obtenida a través de distintas fuentes como, por ejemplo, servicios web de clima, aplicaciones, dispositivos móviles, entre otros. Utilizando las capacidades de conectividad del ESB, esta información es recibida y enviada al motor CEP para que sea procesada.

Detección de Situaciones

Las situaciones son detectadas como eventos complejos por el motor CEP. Para poder especificar las situaciones de interés, es necesario definir las reglas en el motor CEP para detectar dichas situaciones. Las reglas están basadas en información entregada por los generadores de eventos, además pueden tener en

cuenta otras situaciones detectadas junto con funciones o servicios adicionales. Por ejemplo, si se cuenta con información de clima y un servicio adicional devuelve la ciudad a partir de las coordenadas, se puede detectar la zona cafetera en la que se encuentra y a partir de esta identificar las temporadas de riesgo de la roya en el café. Para esto se debe especificar que una regla reconozca dicha situación automáticamente cuando se reciba la localización del usuario utilizando un servicio adicional.

Interacción entre Componentes

Los diagramas de secuencia son utilizados para modelar la interacción entre objetos en una aplicación a través del tiempo, de tal manera que este contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos [63]. En esta sección se presenta la interacción de los componentes y subcomponentes que conforman la solución propuesta.

Detección de Eventos

Cuando se detecta una situación de interés para el usuario se desencadena una serie de interacción entre los componentes de la solución: Motor CEP, ESB y Directivas de adaptación. En la Figura 11 se muestra el diagrama de secuencia detallando dicha interacción.

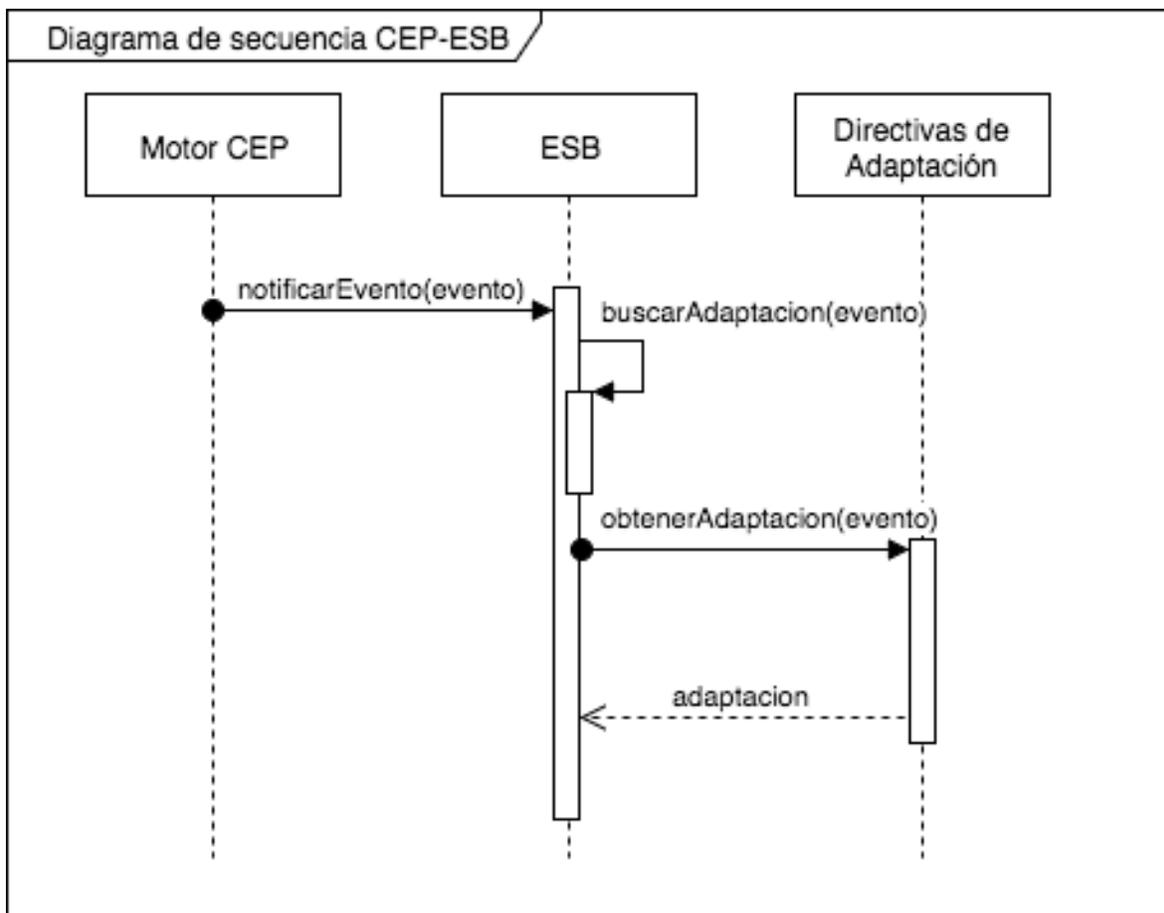


Figura 11. Diagrama de secuencia para la interacción entre CEP y ESB. Fuente propia

Como se presenta en la figura anterior, el CEP internamente contiene todo lo necesario para el manejo de los eventos simples y complejos de tal manera que pueden ser identificados tan pronto suceda alguno de ellos. Una vez identificados, el CEP se encarga de notificar la presencia de un evento complejo al ESB, para que este le dé el tratamiento adecuado según las directivas de adaptación presentadas anteriormente en la sección 3.3.

Interacción del Usuario

El usuario interactúa con la plataforma al solicitar información acerca de una recomendación para su cultivo, brindando información específica del mismo, a lo que la plataforma dará una respuesta a esta petición consumiendo un servicio web. En la Figura 12 se muestra la interacción del usuario con la plataforma desde que solicita la información a la plataforma hasta que le devuelve una respuesta.

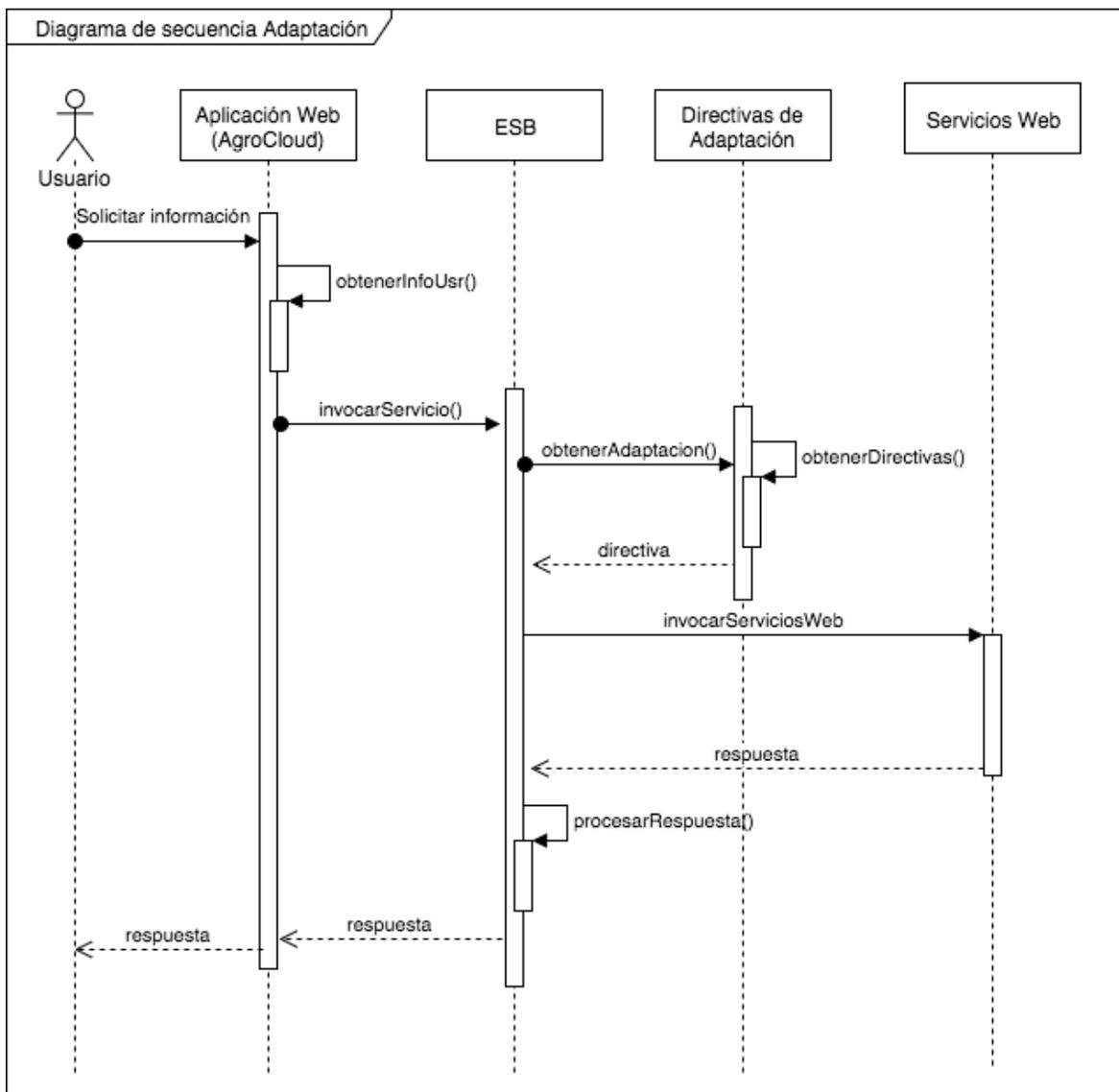


Figura 12. Diagrama de secuencia para la interacción entre el usuario y los componentes de adaptación. Fuente propia

Como se presenta en la figura anterior, el usuario del sistema se conecta a través de una aplicación web donde solicita información sobre una recomendación para su cultivo, la aplicación a su vez obtiene algunos datos de interés del usuario como su ubicación y fechas de importancia para el cultivo (fecha de floración). Con esta información comienza todo el proceso de adaptación de la respuesta que le será devuelta al usuario. Para esto la aplicación web se comunica con el ESB, el cual es el encargado de realizar el proceso de adaptación, teniendo en cuenta los eventos complejos que han sido identificados, las directivas de adaptación previamente

definidas y la invocación de los distintos servicios web necesarios para brindar una mejor respuesta al usuario.

Desarrollo del Prototipo

En esta sección se presentan los distintos aspectos de la implementación de la solución propuesta. En la primera sección se presenta un diagrama de componentes y tecnologías junto con una descripción de las herramientas utilizadas. En la sección 4.4.2 se muestra la descripción de la implementación de los distintos componentes de la plataforma.

Herramienta Utilizadas

En esta sección se presenta un diagrama de componentes y tecnologías junto a una breve descripción de las herramientas utilizadas. En la Figura 13 se presenta el diagrama de componentes y tecnologías de la solución.

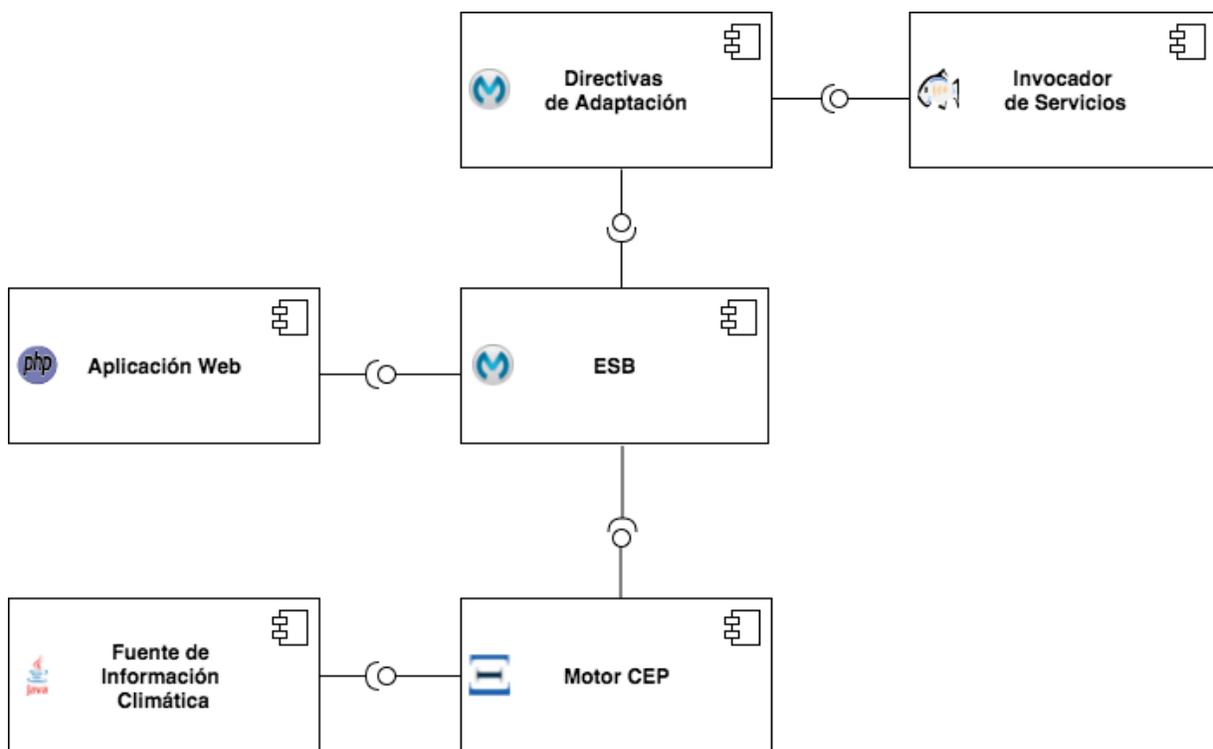


Figura 13. Diagrama de componentes y tecnologías. Fuente propia

A continuación, se describen las tecnologías mostradas en el diagrama de la Figura 13.

Mule ESB

Mule ESB es un *framework* de mensajería ligera basada en Java que permite la rápida conexión e intercambio de datos entre aplicaciones [64]. Mule permite a los desarrolladores integrar sus aplicaciones e intercambiar datos sin tener en cuenta las tecnologías y maneja todas las interacciones entre aplicaciones y componentes de manera transparente sin importar los protocolos de transporte utilizados (JMS, Web Services, JDBC, HTTP, FTP, SMTP). Permite la publicación y la invocación de servicios web de tipos REST y SOAP, haciendo uso del *framework* CXF [65].

Esper

Motor CEP para el análisis de series de eventos el cual se encuentra disponible en Java y .NET. Permiten el desarrollo de aplicaciones con grandes volúmenes de eventos entrantes independientes si, los eventos son de naturaleza histórica o en tiempo real, permitiendo filtrar, analizar los eventos de varias maneras y responder a las condiciones de interés. Para esto Esper proporciona un lenguaje de procesamiento de eventos (EPL) lo que le permite expresar condiciones de coincidencias complejas que incluyen ventanas temporales de tiempo, uniones de diferentes flujos de eventos, así como también filtrado, agregación y clasificación de eventos [66].

GlassFish

Es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que sigan esta especificación, es de código abierto derivado de Apache Tomcat.

GlassFish es la implementación de referencia de Java EE y, como tal, es compatible con Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, servlets, etc. Esto permite a los desarrolladores crear aplicaciones empresariales que sean portátiles, escalables y que se integren con tecnologías heredadas.

Desarrollo de servicios web

Dado que este trabajo de grado se encuentra en el marco del proyecto AgroCloud, que pertenece al programa de la red interinstitucional de cambio climático y seguridad alimentaria (RICCLISA), este proyecto ha desarrollado una plataforma web que tiene como objetivo establecer servicios de monitorización para advertir a los usuarios sobre situaciones de amenaza para los cultivos de café en la zona de interés del proyecto. Para esto, se desplegaron un conjunto de servicios presentados en la Tabla 5:

Tabla 5. Servicios web AgroCloud – Fuente propia

Servicio web	Descripción
Condiciones climáticas	Este servicio web se encarga de gestionar las condiciones climáticas actuales para una determinada ubicación a partir de los datos que se encuentran almacenados. Cuenta con operaciones para consultar datos máximos, mínimos, promedios, entre otros, para cada una de las variables climáticas que se encuentran disponibles.
Alertas generadas	Este servicio web se encarga de la gestión de las alertas que son almacenadas en la base de datos, permitiendo hacer consultas a partir de un rango de fechas, una ubicación específica, un tipo de alerta, entre otros.

Los servicios web presentados en la Tabla 5 se encuentran implementados dentro de la plataforma AgroCloud, pero de acuerdo a las necesidades de este trabajo de grado se hace necesaria la implementación de nuevos servicios web que brindan mayor información al usuario los cuales se presentan en la Tabla 6:

Tabla 6. Servicios web desarrollados - Fuente propia

Servicio web	Descripción
Productos	Este servicio web se encarga de consultar en la base de datos los distintos productos químicos que se encuentran registrados para cada enfermedad, teniendo en cuenta parámetros como la enfermedad donde se utiliza, el cultivo, el momento en el cual se debe aplicar, entre otros.
Tiendas de insumos	Este servicio web se encarga de la gestión de las tiendas de insumos en las cuales se comercializan los productos químicos para el control de una enfermedad. Cuenta con operaciones para consultar a partir de la ubicación, la enfermedad, el cultivo, entre otros.
Guías para enfermedades	Este servicio web se encarga de la gestión de las guías existentes para el manejo de una determinada enfermedad. Cuenta con operaciones para consultar a partir de la enfermedad, la etapa en la cual se

	encuentra la misma, entre otros.
Expertos	Este servicio web se encarga de consultar a los expertos que se encuentran registrados en la base de datos para cada enfermedad. Cuenta con operaciones para consultar a partir de la ubicación, la enfermedad, el cultivo, entre otros.
Comparación de costos	Este servicio web se encarga de la gestión de los costos de aplicación de los distintos productos químicos para el control de una enfermedad. Cuenta con operaciones para consultar a partir de los productos, la enfermedad, el cultivo, entre otros.

Los servicios web presentados en la Tabla 6 son los que brindan información adicional al usuario, los cuales serán adicionados a la respuesta que se entrega al mismo a partir de la alerta de la enfermedad que se haya generado, teniendo en cuenta los datos adicionales que el usuario pueda brindar como lo son la ubicación y las fechas de importancia para su cultivo.

Descripción de la implementación

Con el propósito de construir un prototipo que implemente la adaptación de los servicios web para la generación de alertas tempranas en cultivos, en primera instancia se desarrolló todo lo relacionado con el procesamiento de eventos complejos, para esto se utilizó el motor CEP de Esper [66] en su versión 4.11.0, la cual es una herramienta de distribución libre que funciona sobre java, utilizando el lenguaje de procesamiento de eventos EPL por sus siglas en inglés (*Event Processing Language*), este es un lenguaje declarativo para el análisis de eventos basados en el tiempo, lo que permite detectar situaciones a medida que ocurran. En segunda instancia se desarrolló lo relacionado con ESB, para lo cual se utilizó MuleESB [65] en su versión 3.8.1, empleando la herramienta Anypoint Studio [67] para el desarrollo de los flujos y sub-flujos necesarios para el proceso de adaptación. La implementación de los patrones de adaptación descritos anteriormente en la sección 3.4 se realizó utilizando el lenguaje de procesado de eventos EPL, empleando nuevamente la herramienta MEdit4CEP ya que cuenta con la capacidad de exportar los modelos gráficos de eventos creados previamente al lenguaje EPL, obteniendo como resultado los siguientes códigos.

EPL 1. Variables generación de Roya

```
1: @Name('variables_groya')
2: insert into variables_groya
3: select a1.Sombra as sombra,
4:     avg(a1.Precipitacion) as avg_precipitacion,
5:     avg(a1.HHumedadReID) as avg_hhrel_dia,
6:     avg(a1.HHumedadReIN) as avg_hhreal_noche,
7:     avg(a1.TemperaturaHN) as avg_tmp_hora_noche,
8:     max(a1.Temperatura) as max_tmp,
9:     avg(a1.Temperatura) as avg_tmp,
10:    min(a1.Temperatura) as min_tmp,
11:     a1.Zona as zona
12: from clima.evento.win:time_batch(30 day) a1
```

EPL 1. Lenguaje EPL para el evento variables generación roya (variables_groya)

EPL 1 permite identificar los eventos complejos que contengan las variables significativas para la generación de la roya en el café. Para esto aplica diferentes operaciones como el promedio, máximo o mínimo sobre los datos entregados por el evento simple llamado clima (Figura 4).

EPL 2. Patrón 1 de generación de Roya

```
1: @Name('p1_roya')
2: insert into p1_roya
3: select p1.avg_hhreal_noche as avg_hhrel_n,
4:     p1.max_tmp as max_tmp,
5:     p1.min_tmp as min_tmp,
6:     p1.zona as zona
7: from pattern [p1 = variables_groya((p1.avg_hhreal_noche <=
6.35 and p1.max_tmp <= 22.76 and p1.min_tmp <= 14.48))]
```

EPL 2. Lenguaje EPL para el patrón de roya 1 (p1_roya)

EPL 2 permite identificar el patrón número uno (“T11”) para la generación de la roya en el café. Este patrón requiere de cuatro variables del evento complejo de variables significativas para la generación de roya (variables_groya), junto con los umbrales que cada variable debe igualar o superar con el fin de que pueda ser reconocido como un evento perteneciente a este patrón.

EPL3. Patrón 2 de generación de Roya

```
1: @Name('p2_roya')
2: insert into p2_roya
3: select p2.avg_hhreal_noche as avg_hhrel_noche,
4:     p2.avg_tmp_hora_noche as avg_tmph_noche,
5:     p2.min_tmp as min_tmp,
6:     p2.zona as zona
7: from pattern [p2 = variables_groya((p2.avg_hhreal_noche <=
6.35 and p2.avg_tmp_hora_noche <= 14.58 and p2.min_tmp > 14.48))]
```

EPL 3. Lenguaje EPL para el patrón de roya 2

EPL 3 permite identificar el patrón número dos (“T11”) para la generación de la roya en el café. Este patrón requiere de cuatro variables del evento complejo de variables significativas para la generación de roya (variables_groya), junto con los umbrales que cada variable debe igualar o superar con el fin de que pueda ser reconocido como un evento perteneciente a este patrón.

Patrón 3 de generación de Roya

```
1: @Name('p3_roya')
2: insert into p3_roya
3: select p3.sombra as sombra,
4:     p3.avg_precipitacion as avg_precipitacion,
5:     p3.avg_hhreal_noche as avg_hhrel_n,
6:     p3.zona as zona
7: from pattern [p3 = variables_groya((p3.avg_precipitacion <=
0.57 and p3.avg_hhreal_noche > 6.35 and p3.sombra > 5.28))]
```

EPL 4. Lenguaje EPL para el patrón de roya 3

EPL 4 permite identificar el patrón número tres (“T13”) para la generación de la roya en el café. Este patrón requiere de cuatro variables del evento complejo de variables significativas para la generación de roya (variables_groya), junto con los umbrales que cada variable debe igualar o superar con el fin de que pueda ser reconocido como un evento perteneciente a este patrón.

Una vez generados los EPL son ingresados al motor CEP, de tal manera que cuando se presente alguna de las situaciones de interés logren ser detectadas por el motor CEP y de acuerdo a ellos realizar las distintas adaptaciones de los servicios web con el objetivo de entregar una recomendación al usuario según el caso.

Por otra parte, la implementación de las directivas de adaptación presentados en la sección 3.1 se realizará utilizando las distintas capacidades de medicación propias del ESB, como lo son: enrutamiento, filtrado, transformación, entre otros, teniendo en cuenta que se utilizó MuleESB, se procedió a realizar los distintos flujos de acuerdo a la Tabla 3. En esta tabla se encuentran los casos para los cuales el ESB realizará la o las adaptaciones correspondientes, tal y como se presenta en las siguientes figuras 14-17.

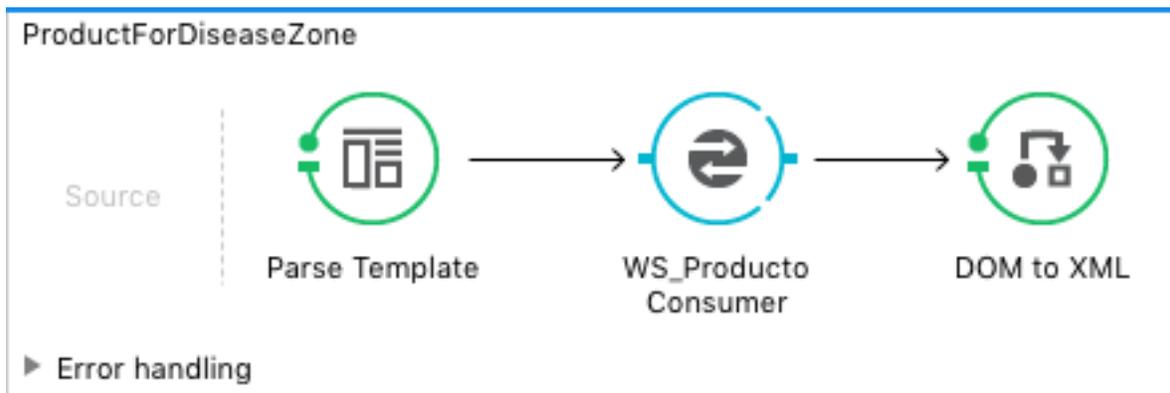


Figura 14. Flujo de invocación servicio web productos. Fuente propia

La Figura 14 presenta el flujo de invocación para el servicio web que entrega la información relacionada con los productos químicos para el control y manejo de una enfermedad dentro de una zona específica, este servicio es llamado por el flujo principal del ESB dependiendo de la adaptación que se le desee brindar al usuario. De la misma manera se realizaron las invocaciones a los distintos servicios web,

encargados de brindar información acerca de las tiendas de insumos, expertos en la enfermedad, entre otros, resaltando algunos casos puntuales como el presentado en la Figura 15. En ella se muestra un elemento enrutador que es el *Choice*, el cual selecciona una alternativa a seguir dependiendo del contenido del mensaje, lo que posibilita que se pueda utilizar una o más operaciones de un servicio web según sea el caso presentando.

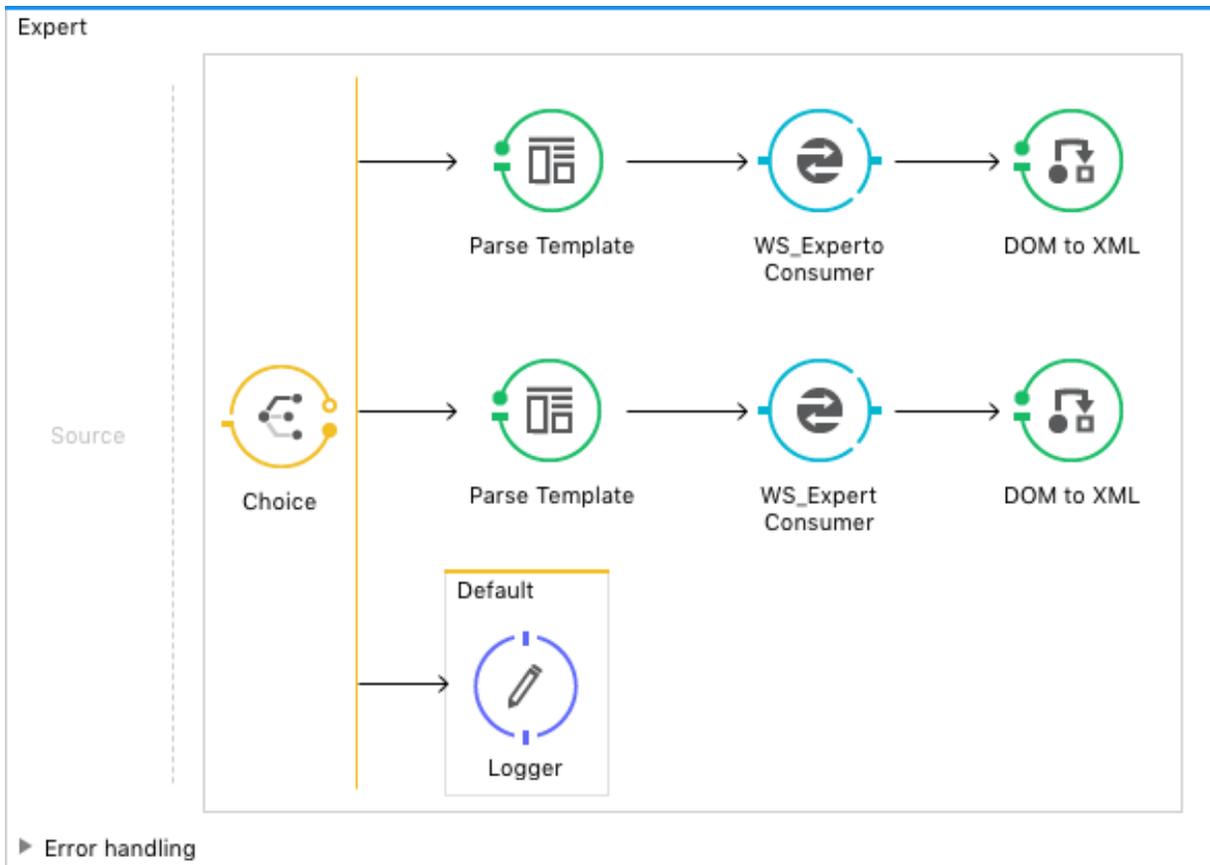


Figura 15. Flujo de invocación servicio web experto. Fuente propia

Luego de realizar los flujos de invocación para los distintos servicios web que son utilizados para la generación de alertas tempranas para enfermedades en cultivos, se procede a implementar el flujo principal del ESB que permitirá realizar las directivas de adaptación planteadas anteriormente el cual será presentado en dos partes. La primera parte es presentada en la Figura 16, la cual muestra el inicio del proceso de adaptación, proceso que se inicia a partir de una petición realizada por el usuario mediante el protocolo HTTP, seguidamente se encuentra la asignación de las variables locales que son obtenidas de esta petición, a partir de estas variables se

procede a identificar el proceso de adaptación a seguir, donde dichas variables y los eventos complejos detectados por la información climática permitirán entregar la mejor respuesta al usuario.

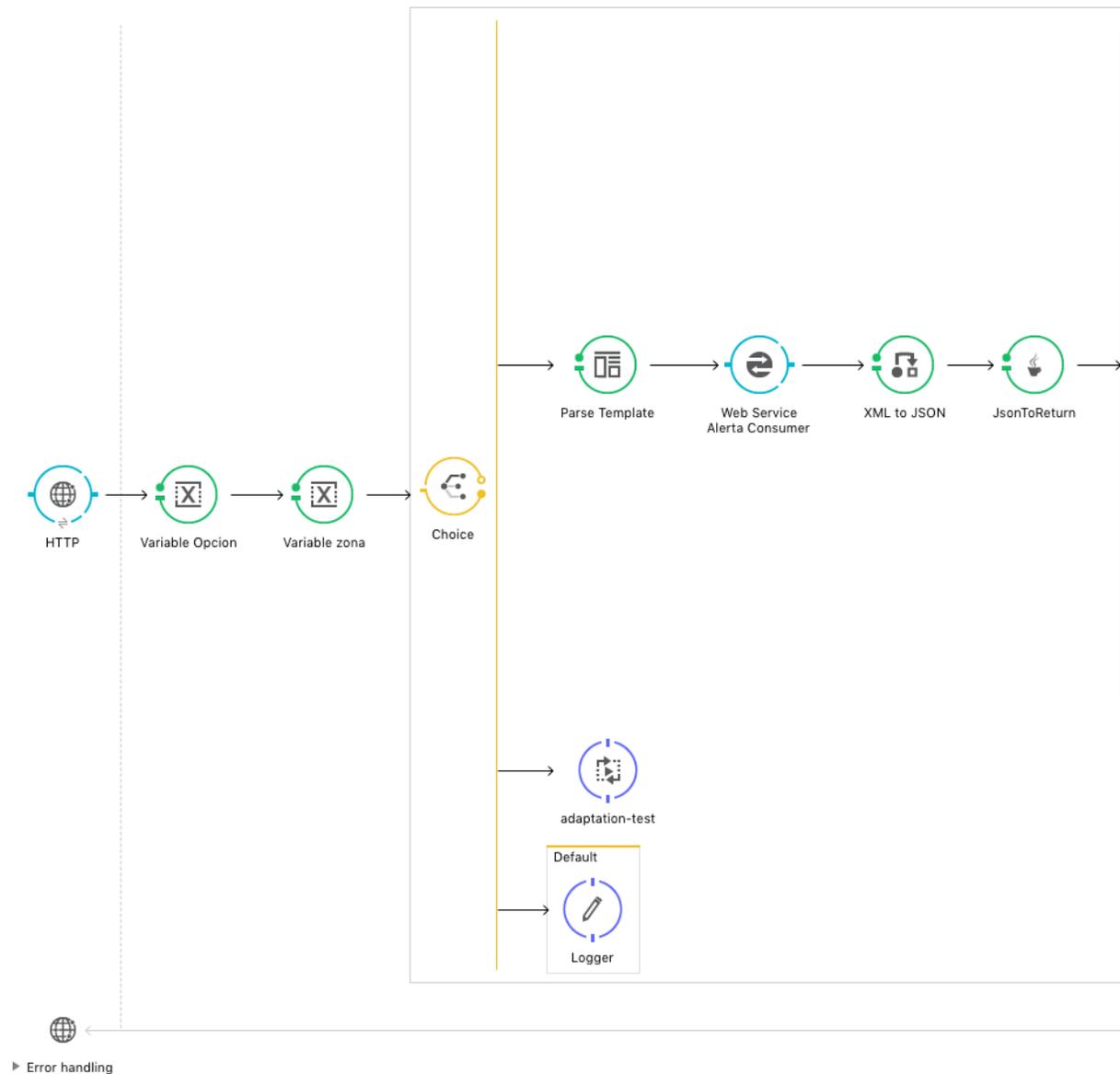


Figura 16. Parte I flujo principal. Fuente propia

La Figura 17 muestra la segunda parte del flujo principal del ESB, en él se presentan las distintas opciones disponibles a partir del tipo de alerta de la enfermedad que ha sido identificado, teniendo en cuenta las condiciones climáticas y los distintos datos ingresados por el usuario, con el objetivo de agregar información relevante como los

productos químicos disponibles para esa enfermedad, los expertos en ella, tiendas de insumos de la zona, entre otros.

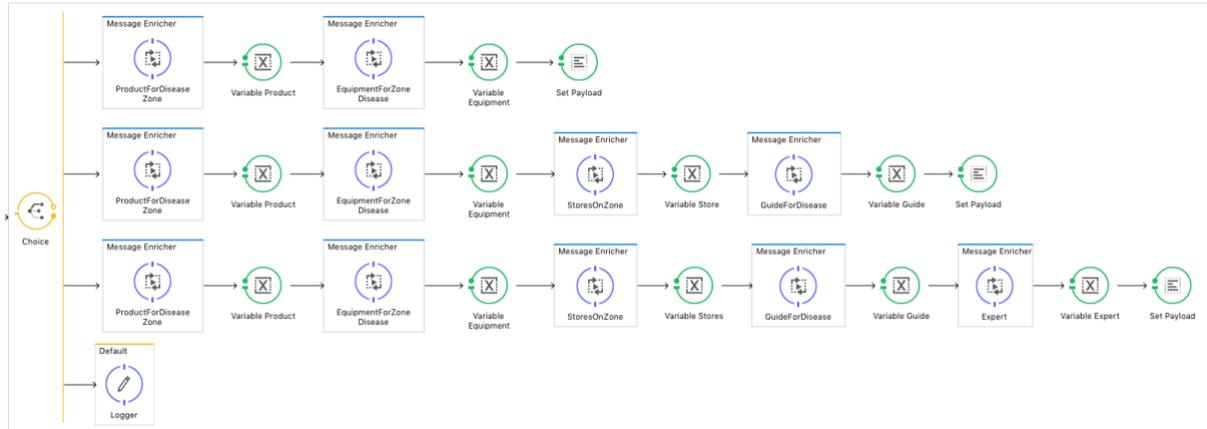


Figura 17. Parte II flujo principal. Fuente propia

En el anterior flujo se pueden apreciar varias de las características de mediación del ESB como lo son *Choice*, *Message Enricher*, entre otros, cada uno de estos elementos interactúa con el mensaje procedente de la petición enviada por el usuario como se mostró en el diagrama presentado en Figura 12 y a partir de este mensaje se va realizando la adaptación enriqueciendo el mensaje con información relevante para el usuario, adicionando recomendaciones relacionadas con productos, tiendas, expertos, entre otros.

Resumen

En este capítulo fueron presentadas los diferentes pasos correspondientes a la implementación de la solución propuesta. Para esto se ilustró la manera en la cual fueron abordados los temas de las directivas de adaptación, la detección de los patrones de eventos de interés y como estos componentes se desarrollaron dentro de MuleESB.

En el caso de los patrones de eventos estos fueron desarrollados utilizando el motor CEP Esper, haciendo uso del lenguaje EPL para la generación de los distintos eventos de interés, que luego fueron conectados con el ESB.

En el caso de las directivas de adaptación estas fueron desarrolladas dentro de MuleESB aprovechando las capacidades de mediación que son ofrecidas por la herramienta, donde se destaca el uso de mediadores (HTTP), transformadores (*Message Enricher*, *XML to JSON*, *Parse Template*), enrutadores (*Choice*), entre

otros. Los cuales trabajan en conjunto para brindar una respuesta adaptada al usuario.

Capítulo 5

Caso de Estudio

En esta sección se presenta el caso de estudio donde se aplicó la solución que fue propuesta anteriormente, el dominio de aplicación del problema a resolver y los casos de estudio planteados, además de la solución dada para el caso.

Dominio de aplicación

La roya es el principal problema fitosanitario de alto impacto para la caficultura [68], particularmente en Colombia llegó en los años 80's afectando los cultivos de las áreas bajas de 600 a 1000 msnm, perjudicando a más de un millón de familias que dependen de este cultivo, motivo por el cual se hace de vital importancia contar con medidas para minimizar las pérdidas que este puede generar sobre las cosechas. Las consecuencias que trae consigo esta enfermedad principalmente se encuentran relacionadas con pérdidas económicas para los agricultores junto con una disminución en la calidad y cantidad de café. La roya, es una enfermedad causada por el patógeno *Hemileia vastatrix*, un hongo que pertenece a la familia *Puccineaceae*, orden *Uredinales*, clase *Basidiomycetes*, los cuales se desarrollan únicamente en el tejido vivo de su planta hospedera, para este caso en las hojas del café.

Un brote de la roya en el cafeto y posterior progreso depende de la ocurrencia simultánea de cuatro factores como se puede distinguir en la Figura 19:

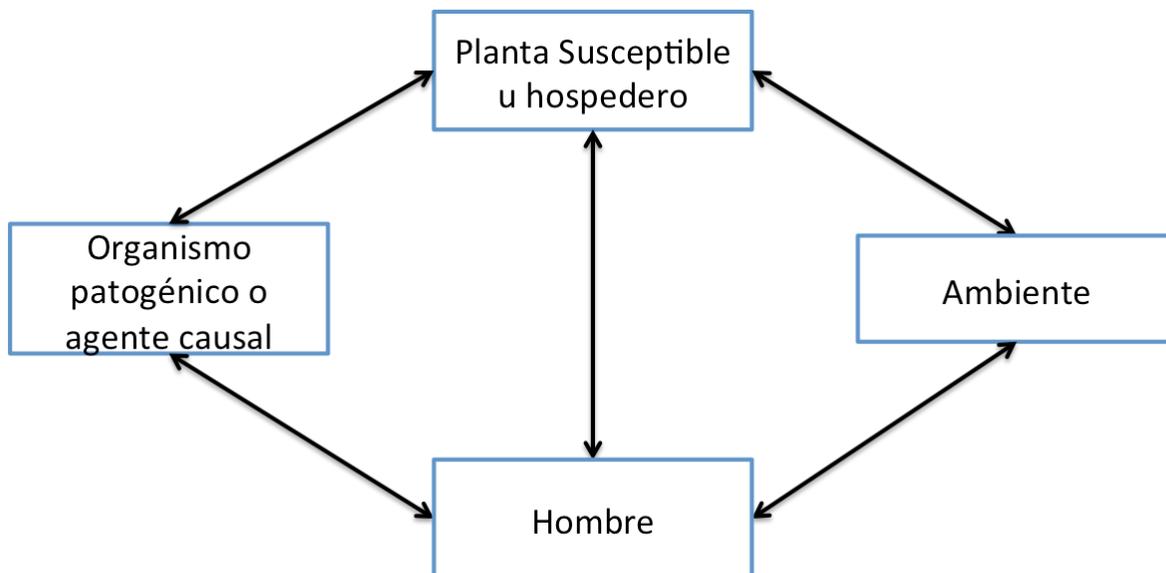


Figura 18. Factores para que se presente la roya en el café [3]

En la Figura 19 se indican los principales factores que inciden en la aparición de la roya en el café, a continuación, se describen sus elementos:

- **Planta susceptible u hospedero:** Hace referencia a la variedad de café que ha sido sembrada, teniendo en cuenta si es una variedad resistente a la roya o no. Además, son importantes aspectos como el estado fisiológico de la planta, el suelo, el exceso de agua, las deficiencias nutricionales, entre otras.
- **Organismo patogénico o agente causal:** El agente causal es un hongo que afecta las hojas del café de manera progresiva, que para el caso de la roya en el cafeto este hongo es conocido como *Hemileia vastatrix* de la familia uredinales.
- **Ambiente:** Se refiere a las condiciones climáticas propicias para la generación de la roya. El hongo *Hemileia vastatrix* necesita condiciones muy particulares, como lo es la salpicadura de la lluvia para iniciar su proceso de dispersión entre hojas y entre plantas, así como de la presencia de una capa de agua en el envés de las hojas para germinar, todo esto acompañado de temperaturas entre 16 y 28 °C, junto con condiciones de bajo brillo solar.
- **Hombre:** El hombre juega un papel importante en la aparición y desarrollo de la enfermedad, ya que de él depende todo lo relacionado con el manejo agronómico, como, por ejemplo: manejo de arvenses, fertilización, sombrero, entre otras.

Teniendo en cuenta los factores mencionados anteriormente, se han realizado algunas aproximaciones desde las ciencias de la computación relacionadas con la identificación automática de episodios de la enfermedad, a partir del uso de técnicas de aprendizaje automático. En estas investigaciones [69]–[75], los autores hacen uso de las distintas variables que intervienen en el desarrollo de la roya en un cultivo, las cuales han sido monitorizadas en los mismos. Con esto, es obtenido un conjunto de datos de entrenamiento que relaciona las condiciones de un cultivo con episodios de la enfermedad y es el insumo principal para la aplicación de técnicas como inducción de árboles de decisión, máquinas de vector de soporte, métodos de ensamble, entre otros. En el caso particular de [72], los autores hacen uso de la inducción de árboles de decisión para generar un modelo que predice una tasa de infección de roya a partir de datos de condiciones de cultivo. Esta tasa indica si el nivel de infección de la enfermedad tiende a mantenerse estable o reducirse, subir moderadamente o, por otra parte, subir de forma acelerada. En consecuencia, los resultados de esta investigación pueden ser usados como elemento principal para alertar a los productores de café sobre las condiciones propicias para la roya.

Estos trabajos se encuentran implementados dentro del proyecto AgroCloud, que pertenece al programa de la red interinstitucional de cambio climático y seguridad alimentaria¹ (RICCLISA), este proyecto ha desarrollado una plataforma web que tiene como objetivo establecer servicios de monitorización para advertir a los usuarios sobre situaciones de amenaza para los cultivos de café en la zona de interés del proyecto. Para esto, se desplegaron un conjunto de servicios de la siguiente manera: Servicios web de datos climáticos (*WS_DClima*), servicios están encargados de obtener todos los datos climáticos que son monitorizados en distintos puntos del país. Servicios web para al cálculo de las variables generadas a partir de los datos climáticos (*WS_VClimaticas*), servicios se encargan de calcular las distintas variables que son utilizadas por los trabajos antes mencionados, dichas variables dependen de las condiciones climáticas. Servicios web para la gestión de las alertas (*WS_Alertas*), servicios se encargan de realizar la gestión en base de datos de las alertas que han sido generadas contiene operaciones como inserción consulta, eliminación entre otros.

Sin embargo, estas aproximaciones carecen de una personalización de la alerta, donde se tengan en cuenta factores que determinan qué tan oportuna es su difusión

¹ www.ricclisa.org

en un momento determinado, dadas las distintas zonas cafeteras del país, las cuales presentan diferentes ciclos productivos y, por ende, fechas de floración y cosechas. Por lo anterior, se presentan dos casos de estudio en los que se pretende personalizar la generación de alertas tempranas mediante la adaptación de servicios web, el caso uno tiene en cuenta la ubicación como un factor adicional y el caso dos tiene en cuenta la floración como un factor adicional.

Caso de estudio 1

El primer caso cuenta con datos espaciales del clima, a los que se pretende adicionar información relacionada con la ubicación del usuario, con el objetivo de identificar la zona cafetera en la que se encuentra y a partir de esta información junto con la obtenida de las condiciones climáticas de la zona, generar alertas tempranas para la roya en el cafeto.

En Colombia las zonas cafeteras son tres y se encuentra divididas de acuerdo al semestre del año en el que se presenta la cosecha principal [3], la zonas cafeteras se encuentran distribuidas de la siguiente manera:

- Zona 1: Cosecha principal en el segundo semestre del año, comprende los departamentos de la Guajira, Magdalena, Cesar, Norte de Santander, Santander, Antioquia, Caldas, Risaralda y Meta (Figura 20).



Figura 19. Zona cafetera colombiana 1. Fuente propia

- Zona 2: Cosecha principal en el primer semestre del año comprende los departamentos de Cauca y Nariño (Figura 21).



Figura 20. Zona cafetera colombiana 2. Fuente propia

- Zona 3: Cosecha principal en los dos semestres del año comprende los departamentos de Valle, Huila, Tolima, Quindío, Cundinamarca, Boyacá (Figura 22).



Figura 21. Zona cafetera colombiana 3. Fuente propia

Caso de estudio 2

El segundo caso cuenta con datos espaciales de clima, a los que se pretende adicionar información brindada por el usuario acerca de la o las fechas de floración de su cultivo. La floración puede entenderse como un estado fenológico, asociado directamente a la producción del cultivo [76], por lo tanto, los períodos de floración en el café, se relacionan directamente con las épocas de brote de enfermedades como la roya, la broca, entre otros. En el caso de la roya, al estimar la fecha de floración del cultivo se consigue identificar el momento en el cual el cultivo es más propenso a la enfermedad, y a partir de este, establecer las fechas en las cuales es ideal iniciar el proceso de control. En resumen, este caso tiene como objetivo generar alertas tempranas para la roya en el cafeto, con la información concebida a partir la fecha de floración, junto con la información obtenida de las condiciones climáticas.

La floración en los cultivos de café es un factor determinante ya que a partir de esta se inicia el manejo de la roya. De acuerdo con [3], conociendo la cantidad y la concentración de la floración se puede establecer el período en el cual corresponde

realizar la aplicación de fungicidas. Las Figuras 23, 24 y 25 muestran el mapa de las zonas cafeteras de Colombia y el comportamiento fenológico del cultivo de café.

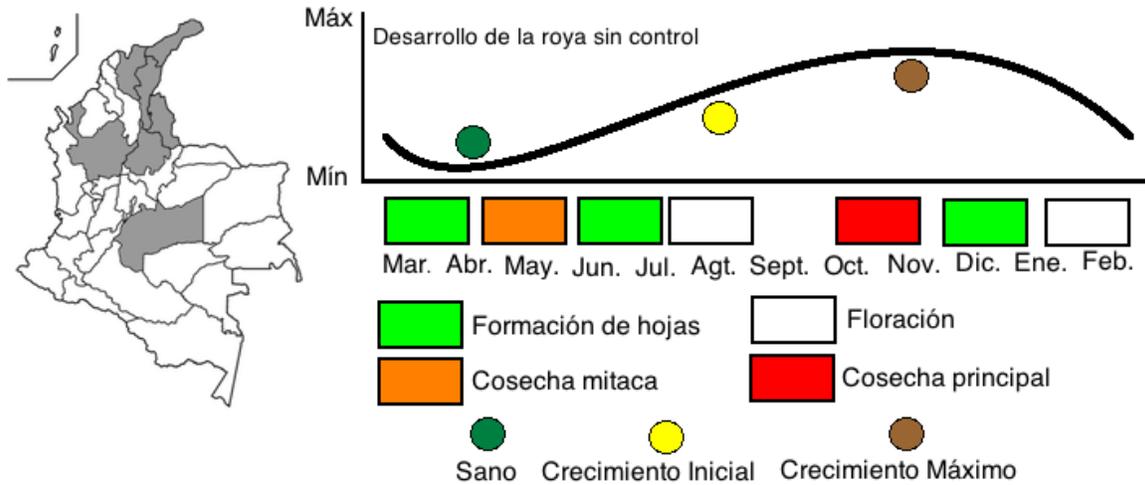


Figura 22. Zona cafetera 1 y fenología del café basado en [3]

En la Figura 23 se observa el desarrollo sin control de la roya en la zona cafetera 1 del país, alcanzando la cosecha principal durante el segundo semestre del año. La época estimada de floración antes de la cosecha principal para este caso se encuentra entre los meses de julio, agosto y septiembre, lo cual indica que a partir de estas fechas se inicia el proceso de desarrollo de la enfermedad. Lo que sugiere que después de la floración se debe iniciar el manejo de la roya en el cafeto, que dependiendo de la cantidad y la concentración puede iniciarse a los 60 días después de la fecha de floración, continuando con aplicaciones a los 90, 120 y 180 días o siguiendo el calendario fijo.

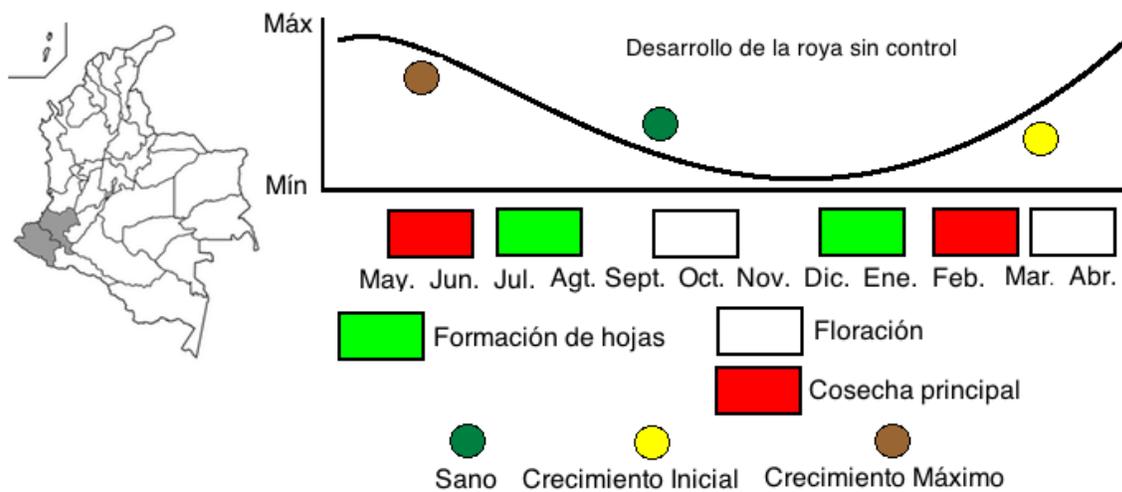


Figura 23. Zona cafetera 2 y fenología del café basado en [3]

En la Figura 24 se observa el desarrollo sin control de la roya en la zona cafetera 2 del país, alcanzando la cosecha principal durante el primer semestre del año. La época estimada de floración antes de la cosecha principal para este caso se encuentra entre los meses de septiembre, octubre, noviembre, marzo y abril, lo cual indica que a partir de estas fechas se inicia el proceso de desarrollo de la enfermedad. Lo que sugiere que después de la floración se debe iniciar el manejo de la roya en el cafeto, que dependiendo de la cantidad y la concentración puede iniciarse a los 60 días después de la fecha de floración, continuando con aplicaciones a los 90, 120 y 180 días o siguiendo el calendario fijo.

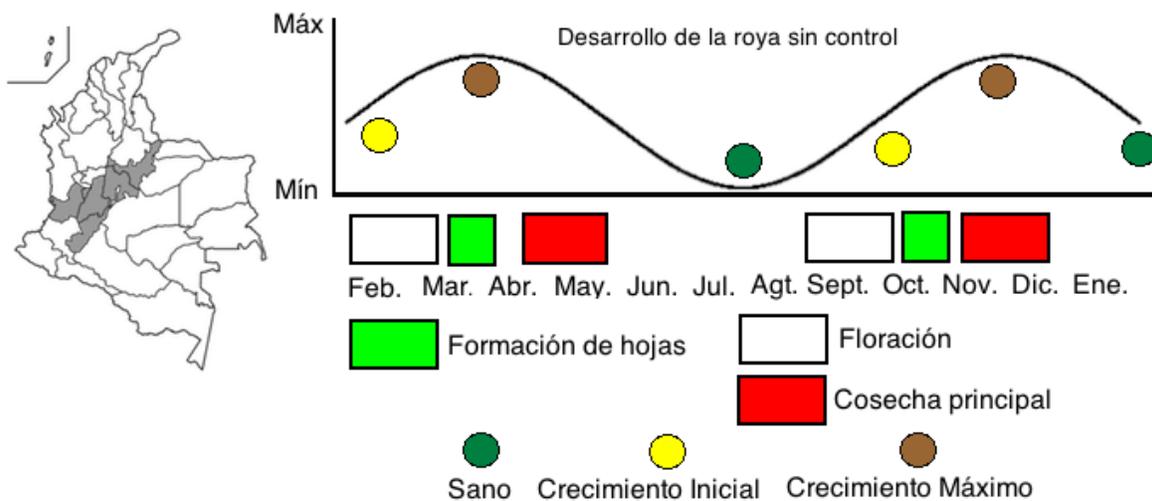


Figura 24. Zona cafetera 3 y fenología del café basado en [3]

En la Figura 25 se observa el desarrollo sin control de la roya en la zona cafetera 3 del país, alcanzando la cosecha principal durante los dos semestres del año. La época estimada de floración antes de la cosecha principal para este caso se encuentra entre los meses de febrero, marzo, septiembre y octubre, lo cual indica que a partir de estas fechas se inicia el proceso de desarrollo de la enfermedad. Lo que sugiere que después de la floración se debe iniciar el manejo de la roya en el cafeto, que dependiendo de la cantidad y la concentración puede iniciarse a los 60 días después de la fecha de floración, continuando con aplicaciones a los 90, 120 y 180 días o siguiendo el calendario fijo.

Evaluación

Esta sección presenta las pruebas realizadas a la solución propuesta. Para esto se tuvo en cuenta el tiempo de respuesta de los servicios web, el uso de recursos computacionales de los mismos y la eficiencia en la detección de los eventos de interés. Para cada una de las pruebas realizadas se consideraron situaciones en las cuales se aplicaba la adaptación propuesta y situaciones en las que no.

Creación del escenario

Para crear un escenario de prueba, primero se definen los datos climáticos que van a servir de punto de partida para el proceso de adaptación, seguidamente se realiza el despliegue de los distintos servicios web que estarán disponibles para el proceso de adaptación y finalmente se realiza la ejecución del proceso de adaptación mediante la integración de CEP y ESB.

Para la definición de los datos climáticos que serán utilizados como punto de partida del proceso de adaptación, se simuló el ingreso de los mismos mediante la generación aleatoria de datos para cada una de las variables climáticas de importancia para el desarrollo como se aprecia en la Tabla 7. El objetivo de generar valores para cada una de estas variables es que a partir de estos datos se puedan identificar eventos complejos para la generación de la roya en el café como se presentó en el capítulo 3.

Tabla 7. Variables climáticas escenario prueba - Fuente propia

Variable climática	Valor mínimo	Valor máximo	Escala
Sombra	0	100	%
Precipitación	0	200	mm

Horas con humedad relativa diurna superior a 90 (hhumedalreID)	0	12	Horas
Horas con humedad relativa nocturna superior a 90 (hhumedalreIN)	0	12	Horas
Temperatura Nocturna (tempHN)	-5	20	Horas
Temperatura	-5	45	° C

Una vez generados los datos climáticos, el CEP (Esper) cuenta con las reglas que fueron definidas en EPL1, EPL2, EPL3 y EPL4, de tal manera que cuando se presente alguno de los casos que fueron tenidos en cuenta se identificará un evento complejo de interés.

En cuanto al despliegue de los servicios web disponibles en el proceso de adaptación, se procedió a desarrollar en Java los servicios presentados en la Tabla 6, los cuales fueron realizados con datos no reales, es decir, datos de prueba que fueron almacenados en una base de datos construida en MySQL [77], con el propósito de satisfacer cada una de las operaciones ofrecidas por los distintos servicios web. Estos servicios fueron montados sobre el servidor de aplicaciones GlassFish.

Finalmente, la interacción del usuario con el proceso de adaptación se realizará mediante una solicitud HTTP ejecutada por medio de un navegador web, de tal manera que una vez recibida esta petición el ESB (MuleESB) recibe el mensaje y a partir de este comienza el proceso de adaptación, donde se identifican los datos adicionales que el usuario puede brindar, como lo son la ubicación y la fecha de floración de su cultivo, los cuales combinados con los eventos identificados por el CEP dan como resultado la adaptación de la respuesta brindada al usuario teniendo en cuenta la información específica ingresada y los eventos complejos identificados a partir de las condiciones climáticas.

Tiempo de respuesta de los servicios web

Con el objetivo de medir el tiempo de respuesta de los servicios web con y sin realizar el proceso de adaptación propuesto se utilizó la herramienta SoapUI v5.3.0, con el fin de invocar cada servicio utilizado. La Tabla 8 presenta los resultados obtenidos basados en 1200 invocaciones para cada uno de los diferentes servicios web.

Tabla 8. Tiempo de respuesta de servicios web. Fuente propia

Servicio Web	Tiempo de respuesta promedio (ms)
WS_Ubicación	9.97
WS_Floración	33.07
WS_Productos	7.8
WS_Tienda	7.4
WS_Expertos	8.0
WS_TasaRoya	8.77
WS_Adaptación Caso 1	39.17
WS_Adaptación Caso 2	14.66

De acuerdo con [78], los valores obtenidos se consideran aceptables, ya que los tiempos de respuesta promedio son inferiores a 200 milisegundos.

Uso de recursos computacionales

Para medir el uso de recursos computacionales (CPU y RAM) en la ejecución de los servicios web con y sin realizar el proceso de adaptación propuesto, se utilizó la herramienta VisualVM v1.3.9. Estos servicios se desplegaron en un servidor con las siguientes características: 16 GB de RAM, procesador Intel Xeon 1.8Ghz y sistema operativo Ubuntu 12.04. Los resultados se presentan en la Tabla 9, el uso de CPU y RAM es bajo y muy similar en todos los casos, lo que indica que la solución propuesta no genera una sobrecarga en los recursos computacionales, para mayor detalle ver ANEXO B.

Tabla 9. Uso de recursos computacionales. Fuente propia

Servicio Web	Porcentaje promedio de uso de CPU	Promedio de uso de RAM
WS_Ubicacion	1.58	53.20
WS_Flowering	1.72	45.28
WS_TasaRoya	1.55	50.43
WS_Productos	1.40	40.27
WS_Tiendas	1.50	37.3
WS_Expertos	1.38	37.29

Efectividad de decisión

En las pruebas realizadas al CEP para verificar la efectividad en la detección de eventos de interés, se consideraron diez casos. Primero, se generaron 100 ejemplos divididos en instancias verdaderas (eventos que requieren adaptación) e instancias negativas (eventos que no requieren adaptación). Posteriormente, se observó el número de instancias clasificadas correctamente por el CEP, determinando los verdaderos positivos, falsos negativos, verdaderos negativos y falsos positivos. Para los siguientes casos, el número de ejemplos se incrementó en 100 sucesivamente hasta obtener 100 instancias con una proporción aleatoria de instancias verdaderas y negativas en cada uno de ellos. Como resultado, el número de falsos negativos y falsos positivos fue cero en todos los casos, lo que significa que el CEP logró identificar la necesidad de adaptación en cada uno de los casos probados. Esto indica que el proceso de detección de eventos de interés llevado a cabo por el CEP es efectivo y adecuado para la solución propuesta dentro de cada uno de los casos de estudio.

Resumen

En este capítulo fueron presentados los casos de prueba a utilizar en la solución propuesta, principalmente se encuentran dos posibilidades de información adicional que puede ser proporcionada por el usuario (ubicación y fecha de floración del cultivo) para que sea combinada con la información climática, y a partir de esta sea realizado el proceso de adaptación por parte del ESB.

Los casos presentados anteriormente, son resumidos en la siguiente Tabla 10:

Tabla 10. Resumen Adaptación. Fuente propia

Caso	Adaptación
Caso 1	Inclusión de la ubicación del usuario con el propósito de identificar la zona cafetera en la que se encuentra.
Caso 2	Inclusión de la fecha de floración con el propósito de identificar los periodos críticos de aparición de la roya en el café.

Capítulo 6

Conclusiones y Trabajos Futuros

Este capítulo describe inicialmente las principales conclusiones de la presente tesis de maestría a las cuales se llegó durante su desarrollo, posteriormente presenta las recomendaciones, y finalmente propone los trabajos futuros que pueden generarse a partir de este estudio.

Conclusiones

A continuación, se presentan las principales conclusiones obtenidas con este trabajo.

- Para la identificación de las brechas de investigación que condujeron al desarrollo de la presente investigación, fue importante realizar una revisión de los trabajos relacionados con la adaptación de servicios web y el procesamiento de eventos complejos. Cabe resaltar que esta temática es bastante amplia con trabajos muy puntuales que ayudaron a conducir el trabajo realizado, lo que llevo a la identificación de la temática en la cual el trabajo tiene cabida.
- Mediante la integración de ESB y CEP se logró adaptar la respuesta ofrecida al usuario, gracias a la adición de distintos servicios web que proporcionan información relevante sobre el manejo y control de una enfermedad, teniendo en cuenta los datos adicionales que el usuario pudiera brindar.
- Experimentalmente fueron evaluados la detección de eventos complejos y la implementación de las directivas de adaptación, que fueron definidas basadas en la información adicional que pudiera ser brindada por el usuario. El CEP presentó un buen desempeño en la identificación de eventos simples y complejos, permitiendo identificar los eventos de interés de manera rápida y oportuna para que a partir de estos el ESB iniciara el proceso de adaptación. Las directivas de adaptación generadas pudieron ser implementadas en forma de flujos dentro del ESB lo que permitió realizar las adaptaciones a los servicios web requeridas a partir del caso identificado.
- El proceso de adaptación presentó un comportamiento aceptable respondiendo adecuadamente a los eventos presentados; teniendo en cuenta que se realizaron pruebas para los tres tipos de alertas que se podían

presentar, partiendo de las situaciones climáticas que fueron creadas de manera aleatoria.

Trabajos Futuros

Este estudio tuvo como principal objetivo proponer, desarrollar y evaluar la adaptación de servicios web para la generación de alertas tempranas de enfermedades, basado en la integración de procesamiento de eventos complejos y el bus de servicios empresariales. El principal enfoque de este trabajo es la integración del bus de servicios empresariales y el procesamiento de eventos complejos lo que permite adaptar los distintos servicios.

Teniendo en cuenta lo anterior, y con el objetivo de complementar el mecanismo de predicción, se proponen los siguientes trabajos futuros:

- Generación dinámica de las directivas de adaptación de servicios web al que corresponda un determinado caso de estudio, esto es, implementar un módulo que permita la generación automática de las directivas de adaptación de acuerdo al caso de estudio que se desee aplicar, con el fin de automatizar y mejorar la capacidad de adaptación de los servicios web.
- Detección de otro tipo de eventos complejos que permitan realizar otras adaptaciones, además la utilización de otros CEP con el objetivo de establecer un punto de comparación con los resultados obtenidos.
- Realizar los flujos de adaptación teniendo en cuenta otros ESB disponibles en el mercado, así como también estudiar la posibilidad de incluir otras características de los ESB que permitan realizar las distintas adaptaciones.
- Detección de otros tipos de enfermedades y plagas, así como también el estudio de otros posibles cultivos.
- Realizar un conjunto de pruebas con un grupo de usuarios finales para que brinden su opinión acerca de la relevancia de la información adicional que es entregada con las alertas generadas.

Bibliografía

- [1] G. C. Gómez, «Desarrollos científicos de Cenicafe en la última década», *Rev. Acad. Colombiana Cienc. Exactas Físicas Nat.*, vol. 1, n.º 30, pp. 89–100, 2005.
- [2] A. de CAMARGO y A. R. Pereira, «Agrometeorology of the coffee crop», *Geneva World Meteorol. Organ.*, 1994.
- [3] Carlos A. Rivillas Osorio, César A. Serna Giraldo, Marco A. Cristancho Ardila, y Alvaro L. Gaitán Bustamante, «La Roya del Cafeto en Colombia Impacto, manejo y costos del control», feb. 2011.
- [4] C. De León, «Enfermedades del maíz: Guía para su identificación en el campo», 1974.
- [5] F. Gauhl y others, «Multilocational evaluation of black Sigatoka resistance in banana and plantain. Evaluación, en varios lugares, de bananos y plátanos resistentes a la Sigatoka negra.», *IITA Res. Guide 47*, 1993.
- [6] Agrios, G.N, *Fitopatología*, Segunda Edición. México: UTEHA-Noriega.
- [7] A. García y Dulce Obín, «Sistemas de Alerta Temprana para Prevención de Enfermedades y Plagas», dic. 2013.
- [8] I. Acuña, R. Bravo, F. Cádiz, M. Vargas, y S. Mancilla, «CURVA DE LIBERACIÓN DE CONIDIAS DE ALTERNARIA Y SU RELACIÓN CON LA ACUMULACIÓN DE DÍAS FISIOLÓGICOS (P-Days) TENDIENTES A DESARROLLAR UN SISTEMA DE ALERTA TEMPRANA PARA AL CULTIVO DE PAPA EN LA ZONA SUR DE CHILE».
- [9] M. Barquero Miranda, «Sistema de alerta temprana para el ojo de gallo», *Rev. Inf. 2012*, 2012.
- [10] A. O. Rodríguez Roa y others, «Desarrollo de un sistema de alertas agroclimáticas tempranas para la chinche de los pastos, *Collaria scenica*, en la sabana de Bogotá», Universidad Nacional de Colombia, 2011.
- [11] D. Gunasekera, N. Plummer, T. Bannister, y L. Anderson-Berry, «Natural disaster mitigation: role and value of warnings», *Econ. Value Fire Weather Serv.*, p. 3, 2005.
- [12] J. J. G. Quintero, C. D. N. Mendez, y Z. V. Sánchez, «Análisis de buenas prácticas en el proceso de beneficio del café: experiencia de estudio en el municipio de Viotá (Cundinamarca, Colombia)», *Ing. Solidar.*, vol. 13, n.º 22, 2017.
- [13] Federación Nacional de Cafeteros de Colombia (Fedecafé), «MANEJO ADECUADO DE LA ROYA DEL CAFETO».
- [14] A. T. Mosquera *et al.*, «Evaluación de fertilización orgánica en cafeto (*Coffea arabica*) con pequeños productores de Santander, Colombia», *Temas Agrar.*, vol. 21, n.º 1, pp. 90–101, 2016.
- [15] A. Wiltshire, «Developing early warning systems: A checklist», en *Proc. 3rd Int. Conf. Early Warning (EWC)*, 2006.
- [16] D. Georgakopoulos, M. Papazoglou, y others, *Service-oriented computing*. 2009.
- [17] «Service-Oriented Architecture (SOA) Definition», *Service Architecture*. [En

- línea]. Disponible en: http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html. [Accedido: 03-feb-2017].
- [18] W. W. Group, W. W. Group, y others, *Web services architecture*. 2004.
- [19] J.-L. Maréchaux, «Combining service-oriented architecture and event-driven architecture using an enterprise service bus», *IBM Dev. Works*, pp. 1269–1275, 2006.
- [20] J. van Hoof, *How eda extends soa and why it is important*. unpublished, 2006.
- [21] B. Sosinsky, *Cloud computing bible*, vol. 762. John Wiley & Sons, 2010.
- [22] G. Alonso, F. Casati, H. Kuno, y V. Machiraju, «Web services», en *Web Services*, Springer, 2004, pp. 123–149.
- [23] B. M. Michelson, «Event-driven architecture overview», *Patricia Seybold Group*, vol. 2, 2006.
- [24] J. Boubeta-Puig, G. Ortiz, y I. Medina-Bulo, «A model-driven approach for facilitating user-friendly design of complex event patterns», *Expert Syst. Appl.*, vol. 41, n.º 2, pp. 445-456, feb. 2014.
- [25] D. C. Luckham, *Event processing for business: organizing the real-time enterprise*. John Wiley & Sons, 2011.
- [26] «Event Processing Glossary – Version 2.0 | Real Time Intelligence & Complex Event Processing».
- [27] O. Etzion y P. Niblett, *Event processing in action*. Manning Publications Co., 2010.
- [28] M. Eckert, F. Bry, S. Brodt, O. Poppe, y S. Hausmann, «A CEP babelfish: Languages for complex event processing and querying surveyed», en *Reasoning in Event-Based Distributed Systems*, Springer, 2011, pp. 47–70.
- [29] R. Schulte, «Predicts 2003: Enterprise service buses emerge», *Rep. Gart. Dec.*, 2002.
- [30] M. P. Papazoglou y W.-J. Van Den Heuvel, «Service oriented architectures: approaches, technologies and research issues», *VLDB J.*, vol. 16, n.º 3, pp. 389–415, 2007.
- [31] F. Menge, «Enterprise service bus», en *Free and open source software conference*, 2007, vol. 2, pp. 1–6.
- [32] M. Keen *et al.*, «Patterns: Implementing an SOA using an enterprise service bus», *IBM Redb.*, vol. 336, 2004.
- [33] V. García Gutiérrez, «Sistema para la adaptación de servicios a nivel de presentación y de navegación en portales web», 2013.
- [34] A. Erradi, P. Maheshwari, y V. Tasic, «Policy-driven middleware for self-adaptation of web services compositions», en *Middleware 2006*, Springer, 2006, pp. 62–80.
- [35] J. L. Pastrana, E. Pimentel, y M. Katrib, «Técnicas de Web Semántica para la Adaptación Dinámica de Componentes y Servicios», en *Memorias de la IX Conferencia Iberoamericana de Software Engineering (CIBSE 2006)*, La Plata, Argentina, Abril 24-28, 2006, 2006, pp. 23–36.
- [36] R. Basher, «Global early warning systems for natural hazards: systematic and people-centred», *Philos. Trans. R. Soc. Lond. Math. Phys. Eng. Sci.*, vol. 364, n.º 1845, pp. 2167–2182, 2006.
- [37] «Early Warning Systems A State of the Art Analysis and Future Directions», *Environ. Dev.*, vol. 4, pp. 136-171, oct. 2012.

- [38] U. Meissen y A. Voisard, «Increasing the effectiveness of early warning via context-aware alerting», en *Proceedings of the 5th International Conference, on Information Systems for Crisis Response and Management (ISCRAM)*, 2008, pp. 431–440.
- [39] J. Boubeta-Puig, G. Ortiz, y I. Medina-Bulo, «An Approach of Early Disease Detection using CEP and SOA», presentado en SERVICE COMPUTATION 2011, The Third International Conferences on Advanced Service Computing, 2011, pp. 143-148.
- [40] G. O. Juan Boubeta-Puig, «Approaching the Internet of Things through Integrating SOA and Complex Event Processing», pp. 304-323, 2014.
- [41] J. Dunkel, A. Fernández, R. Ortiz, y S. Ossowski, «Event-driven architecture for decision support in traffic management systems», *Expert Syst. Appl.*, vol. 38, n.º 6, pp. 6530-6539, jun. 2011.
- [42] J. A. Dorado-Cerón, «Detección de Amenazas de Seguridad mediante Procesamiento de Eventos Complejos», 2012.
- [43] L. González y G. Ortiz, «An ESB-Based Infrastructure for Event-Driven Context-Aware Web Services», en *Advances in Service-Oriented and Cloud Computing*, Springer, 2013, pp. 360–369.
- [44] L. González y R. Ruggia, «Towards dynamic adaptation within an ESB-based service infrastructure layer», en *Proceedings of the 3rd International workshop on Monitoring, Adaptation and Beyond*, 2010, pp. 40–47.
- [45] G. Ortiz, J. Boubeta-Puig, y others, «Context-Aware Leisure Service: A Case-Study based on a SOA 2.0 Infrastructure», en *ICIW 2014, The Ninth International Conference on Internet and Web Applications and Services*, 2014, pp. 25–30.
- [46] J. Boubeta-Puig, «Procesamiento de Eventos Complejos en Arquitecturas Orientadas a Servicios 2.0», 14-may-2013. [En línea]. Disponible en: <http://rodin.uca.es/xmlui/handle/10498/16756>. [Accedido: 12-nov-2014].
- [47] Laura González, «Plataforma ESB Adaptativa para Sistemas Basados en Servicios», Universidad de la República, Montevideo, Uruguay, 2011.
- [48] L. González, J. L. Laborde, M. Galnares, M. Fenoglio, y R. Ruggia, «An adaptive enterprise service bus infrastructure for service based systems», en *Service-Oriented Computing–ICSOC 2013 Workshops*, 2013, pp. 480–491.
- [49] G. Ortiz, J. Boubeta-Puig, A. García de Prado, y I. Medina-Bulo, «Towards event-driven context-aware web services», *Adapt. Web Serv. Modul. Reusable Softw. Dev. Tactics Solut.*, pp. 148–159, 2012.
- [50] L. González y G. Ortiz, «An Event-Driven Integration Platform for Context-Aware Web Services.», *J UCS*, vol. 20, n.º 8, pp. 1071–1088, 2014.
- [51] J. Ocharan, «Sistemas de Alerta Temprana. Fotografía actual y retos», *Cuad. Int. Tecnol. Para El Desarro. Hum.*, n.º 6, p. 2, 2007.
- [52] P. Flores N., J. Lerdon F., R. Bravo H., y I. Acuña, «FACTIBILIDAD DE IMPLEMENTAR PRONOSTICADORES AUTOMATIZADOS PARA CONTROLAR EL TIZÓN TARDIO DE LA PAPA EN EL SUR DE CHILE», *Agro Sur*, vol. 36, n.º 1, pp. 37-42, may 2008.
- [53] M. L. Gleason *et al.*, «Obtaining weather data for input to crop disease-warning systems: leaf wetness duration as a case study», *Sci. Agric.*, vol. 65, n.º SPE, pp. 76-87, dic. 2008.

- [54] Y. Liu, J. Hu, I. Snell-Feikema, M. S. VanBemmel, A. Lamsal, y M. C. Wimberly, «Software to facilitate remote sensing data access for disease early warning systems», *Environ. Model. Softw.*
- [55] A. M. D. Sottara, «A CEP-based SOA for the management of WasteWater Treatment Plants», pp. 58-65, 2009.
- [56] A. Adi, D. Botzer, G. Nechushtai, y G. Sharon, «Complex Event Processing for Financial Services», en *IEEE Services Computing Workshops, 2006. SCW '06*, 2006, pp. 7-12.
- [57] C. Funk y J. P. Verdin, «Real-time decision support systems: the famine early warning system network», en *Satellite rainfall applications for surface hydrology*, Springer, 2010, pp. 295–320.
- [58] P. E. S. DE CÓRDOBA, «SISTEMA DE ALERTA TEMPRANA DE INUNDACIONES», 2005.
- [59] R. Kazhimiakin, «Adaptation and Monitoring in S-Cube: Global Vision and Roadmap», en *Workshop on Service Monitoring, Adaptation and Beyond*, 2009, p. 67.
- [60] E. Lasso, T. T. Thamada, C. A. A. Meira, y J. C. Corrales, «Graph Patterns as Representation of Rules Extracted from Decision Trees for Coffee Rust Detection», en *Research Conference on Metadata and Semantics Research*, 2015, pp. 405–414.
- [61] J. Boubeta-Puig, G. Ortiz, y I. Medina-Bulo, «MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0», *Knowl.-Based Syst.*, vol. 89, pp. 97-112, nov. 2015.
- [62] D. C. Corrales, A. F. Casas, A. Ledezma, y J. C. Corrales, «Two-Level Classifier Ensembles for Coffee Rust Estimation in Colombian Crops», *Int. J. Agric. Environ. Inf. Syst. IJAEIS*, vol. 7, n.º 3, pp. 41–59, 2016.
- [63] M. Sliger y S. Broderick, *The software project manager's bridge to agility*. Addison-Wesley Professional, 2008.
- [64] C. I. Castro-Zamora y E. Flores-Valdés, «Capacidad de Orquestación de Servicios Web en las Herramientas MULE ESB y Oracle Service Bus», *Rev. Ing. Lámpsakos*, n.º 14, pp. 40–50, 2015.
- [65] D. Dossot, J. D'Emic, y V. Romero, *Mule in action*. Manning, 2014.
- [66] «EsperTech - Products». [En línea]. Disponible en: <http://www.espertech.com/products/>. [Accedido: 02-mar-2016].
- [67] «Anypoint Platform». [En línea]. Disponible en: <https://anypoint.mulesoft.com/login/#/signin>. [Accedido: 04-oct-2017].
- [68] A. GAITAN *et al.*, «Evento de La Niña en Colombia: Recomendaciones para la caficultura», 2016.
- [69] D. C. Corrales, A. Ledezma, A. J. Peña, J. Hoyos, A. Figueroa, y J. C. Corrales, «Un nuevo conjunto de datos para la detección de roya en cultivos de café Colombianos basado en clasificadores», *Sist. Telemática*, vol. 12, n.º 29, pp. 9–23, 2014.
- [70] D. C. Corrales, A. Figueroa, A. Ledezma, y J. C. Corrales, «An Empirical Multi-classifier for Coffee Rust Detection in Colombian Crops», en *Computational Science and Its Applications – ICCSA 2015; 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part I*, 2015, vol. 9155, pp. 60–74.
- [71] D. C. Corrales, A. F. Casas, A. Ledezma, y J. C. Corrales, «Two-Level Classifier Ensembles for Coffee Rust Estimation in Colombian Crops», *Int. J. Agric.*

Environ. Inf. Syst. IJAEIS, vol. 7, n.º 3, pp. 41–59, 2016.

[72] E. Lasso, T. T. Thamada, C. A. A. Meira, y J. C. Corrales, «Graph Patterns as Representation of Rules Extracted from Decision Trees for Coffee Rust Detection», en *Metadata and Semantics Research*, E. Garoufallou, R. J. Hartley, y P. Gaitanou, Eds. Springer International Publishing, 2015, pp. 405-414.

[73] C. A. Meira, L. H. Rodrigues, y S. A. Moraes, «Análise da epidemia da ferrugem do cafeeiro com árvore de decisão», *Trop. Plant Pathol.*, vol. 33, n.º 2, pp. 114–124, 2008.

[74] C. A. A. Meira, L. H. A. Rodrigues, y S. A. de Moraes, «Modelos de alerta para o controle da ferrugem-do-cafeeiro em lavouras com alta carga pendente», *Pesqui. Agropecuária Bras.*, vol. 44, pp. 233-242, 2009.

[75] M. E. Cintra, C. A. A. Meira, M. C. Monard, H. A. Camargo, y L. H. A. Rodrigues, «The use of fuzzy decision trees for coffee rust warning in Brazilian crops», en *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, 2011, pp. 1347–1352.

[76] V. RAMIREZ *et al.*, «Variabilidad climática y la floración del café en Colombia.», 2013.

[77] A. B. MySQL, *MySQL*. 2001.

[78] Z. Liang-Jie, *Web Services Research and Practices*. Idea Group Inc (IGI), 2008.

Anexo A

Artículo AACC17

Improving Early Warning Systems for Agriculture Based on Web Service Adaptation

Oscar Ricardo Valencia^(✉), Emmanuel Lasso , and Juan Carlos Corrales

Grupo de Ingeniería Telemática, Universidad del Cauca, Campus Tulcán, Popayán
Cauca, Colombia {ovalencia,egllasso,jcorral}@unicauca.edu.co

Abstract. Early warning systems are designed to inform the largest number of users, such as a country or a region, about a risky situation. However, in specific domains such as agriculture, it is commonly required that these alerts be more specific according to the crops location and their properties, consequently the web services of these systems must be adapted. On the other hand, the Enterprise Services Bus with its mediation capabilities (such as message transformation and routing) and Complex Event Processing with their monitoring characteristics can be integrated to meet the adaptation requirements of web services at runtime. This paper presents an improvement for Early Warning System for coffee production that, according to the area in which a crop is located and its phenology, manages the adaptation of alerts for coffee rust, based on the integration of an Enterprise Services Bus and a Complex Events Processing.

Keywords: Webserviceadaptation·Earlywarningsystems·Enterpriseservice bus · Complex event processing

Artículo ICCSA 2017

Decision Support System for Coffee Rust Control Based on Expert Knowledge and Value-Added Services

Emmanuel Lasso^(&), Óscar Valencia, and Juan Carlos Corrales

Grupo de Ingeniería Telemática, Universidad del Cauca, Campus Tulcán, Popayán,
Cauca, Colombia {eglasso,ovalencia,jcorral}@unicauca.edu.co

Abstract. In coffee production, the quality and quantity of harvests depends on the diseases treatment. One of the diseases with the most negative impact is the Coffee Rust. It causes losses around 30% in Colombian coffee crops. Several studies propose the use of computer sciences techniques for the automatic detection of conditions that trigger epidemics. On the other hand, the knowledge of experts in the control of the disease needs to be disseminated in a more agile way so that the farmers can make correct decisions to avoid great losses in the production. This paper presents a Decision Support System (DSS) for Coffee Rust Control based on expert knowledge. It recommends the best alternative for the application of fungicides and proposes some value-added services based on the integration of its functionalities with those offered in the services of an Early Warning System (EWS) for Coffee Rust. Our proposal represents a highly scalable and flexible solution for the disease management at farmer level.

Keywords: Decision support Service bus SOA Expert knowledge Coffee rust Agriculture Disease management

Anexo B

Uso de recursos computacionales

Este anexo presenta los datos adicionales acerca de la monitorización del uso de recursos computacionales que fueron obtenidos luego de realizar las pruebas con los distintos servicios web.

Servicio web Floración:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web de floración, se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la misma herramienta. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web de floración.

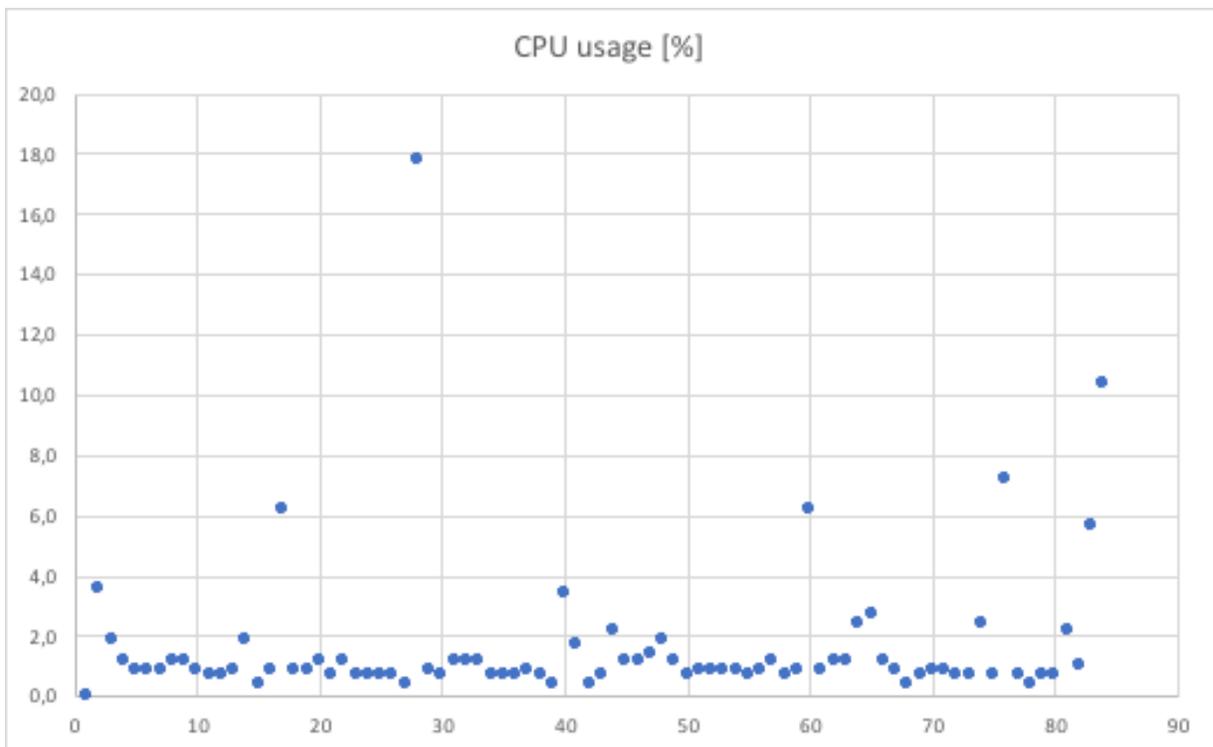


Figura 1. Uso de CPU - Servicio Web Floración

La Figura 1 presenta el porcentaje de uso de CPU cuando se realizan las distintas invocaciones al servicio web de floración.

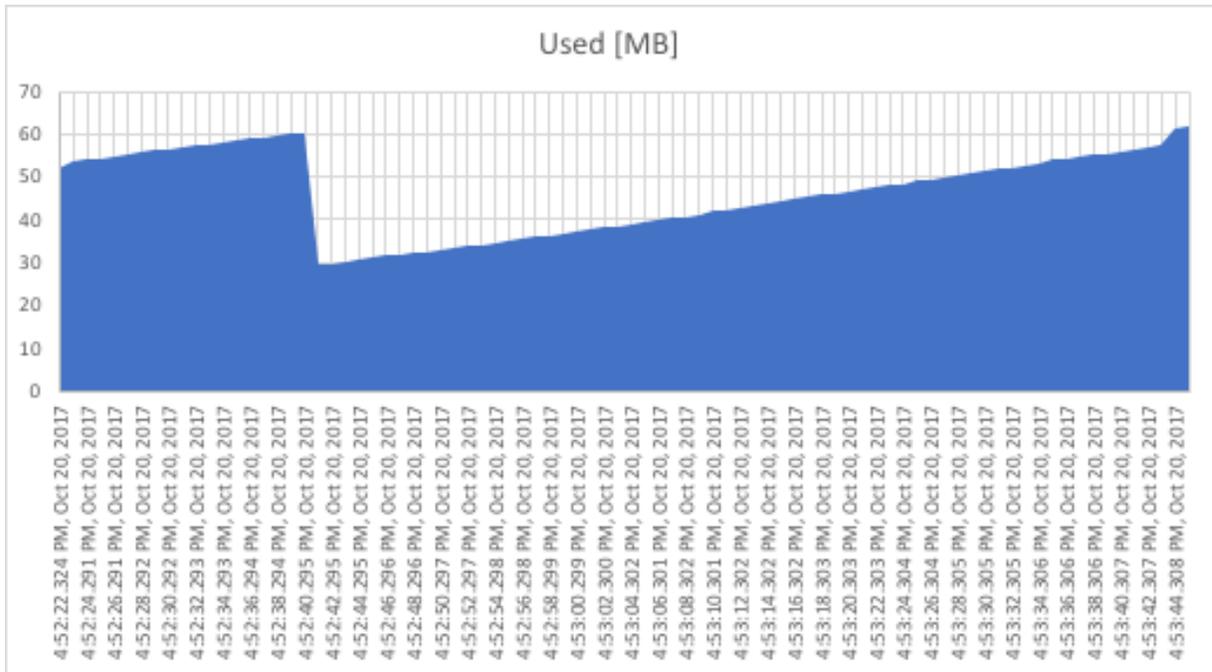


Figura 2. Uso de memoria RAM – Servicio Web Floración

La Figura 2 presenta el uso de memoria RAM comparado con el tiempo de duración de las distintas invocaciones al servicio web de floración.

Servicio web ubicación:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web de ubicación, se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la misma herramienta. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web de ubicación.

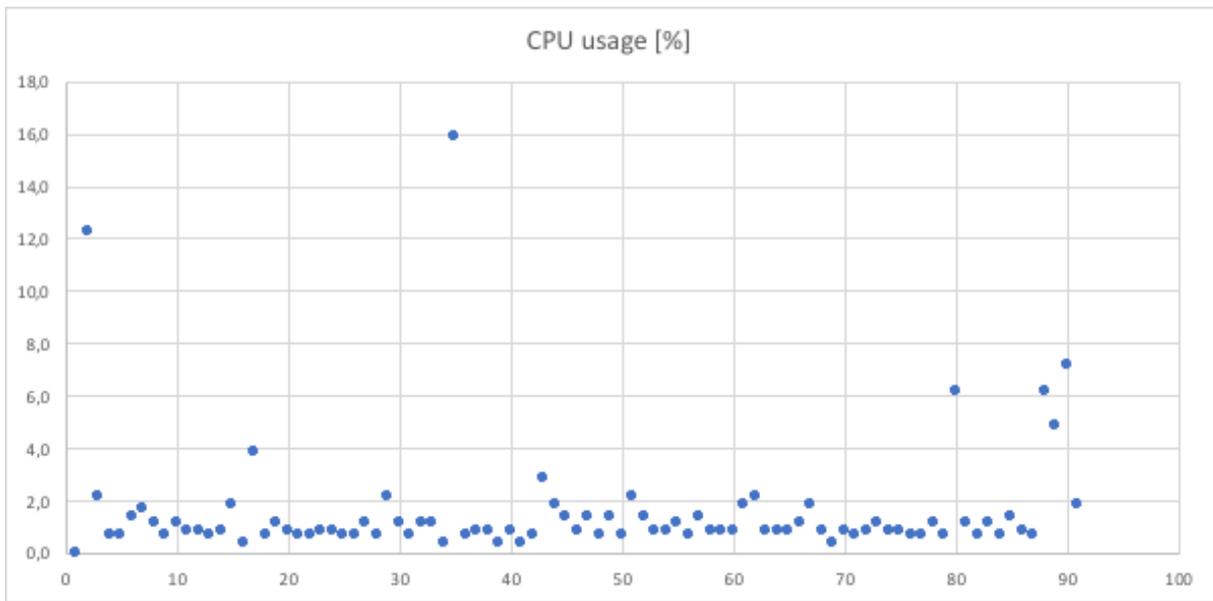


Figura 3. Uso de CPU – Servicio web ubicación

La Figura 3 presenta el porcentaje de uso de CPU cuando se realizan las distintas invocaciones al servicio web de ubicación.

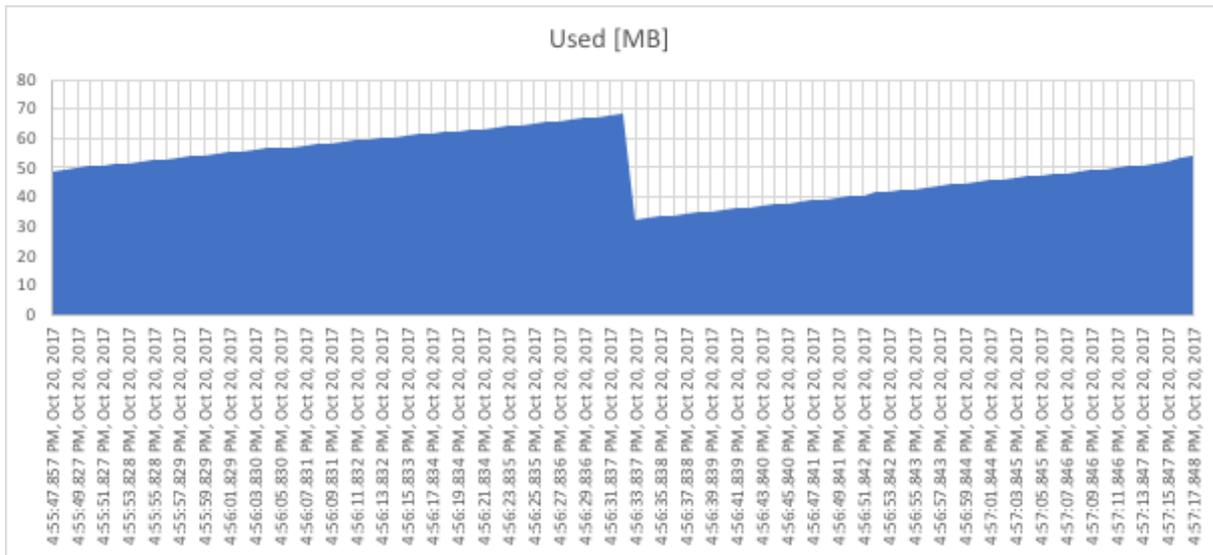


Figura 4. Uso de memoria RAM – Servicio web ubicación

La Figura 4 presenta el uso de memoria RAM comparado con el tiempo de duración de las distintas invocaciones al servicio web de ubicación.

Servicio web Tasa Roya:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web de tasa de roya, se procedió a realizar 1200

invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la misma herramienta. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web de tasa roya.

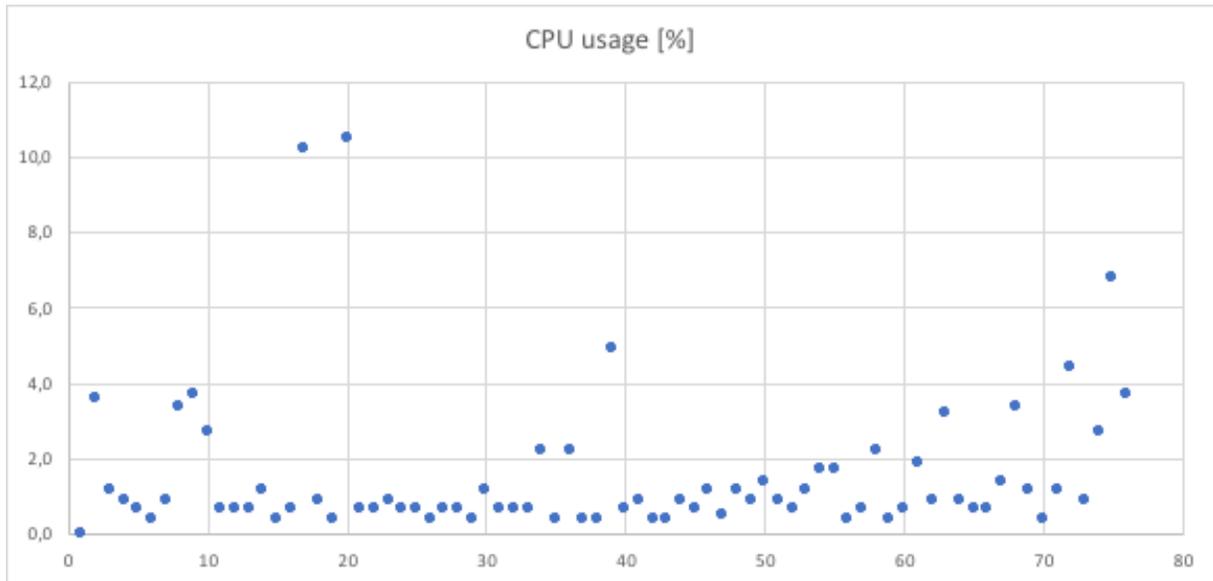


Figura 5. Uso de CPU – Servicio web tasa roya

La Figura 5 presenta el porcentaje de uso de CPU cuando se realizan las distintas invocaciones al servicio web de tasa roya.

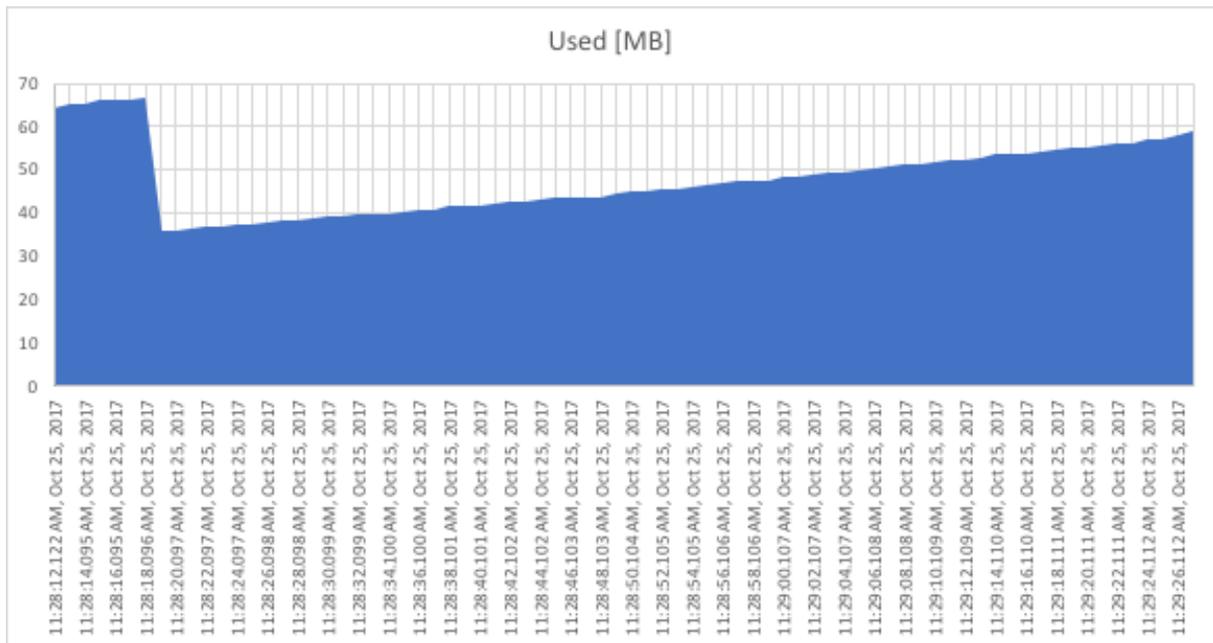


Figura 6. Uso de memoria – Servicio web tasa roya

La Figura 6 presenta el uso de memoria RAM comparado con el tiempo de duración de las distintas invocaciones al servicio web de tasa roya.

Servicio web productos:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web productos, se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la misma herramienta. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web productos.

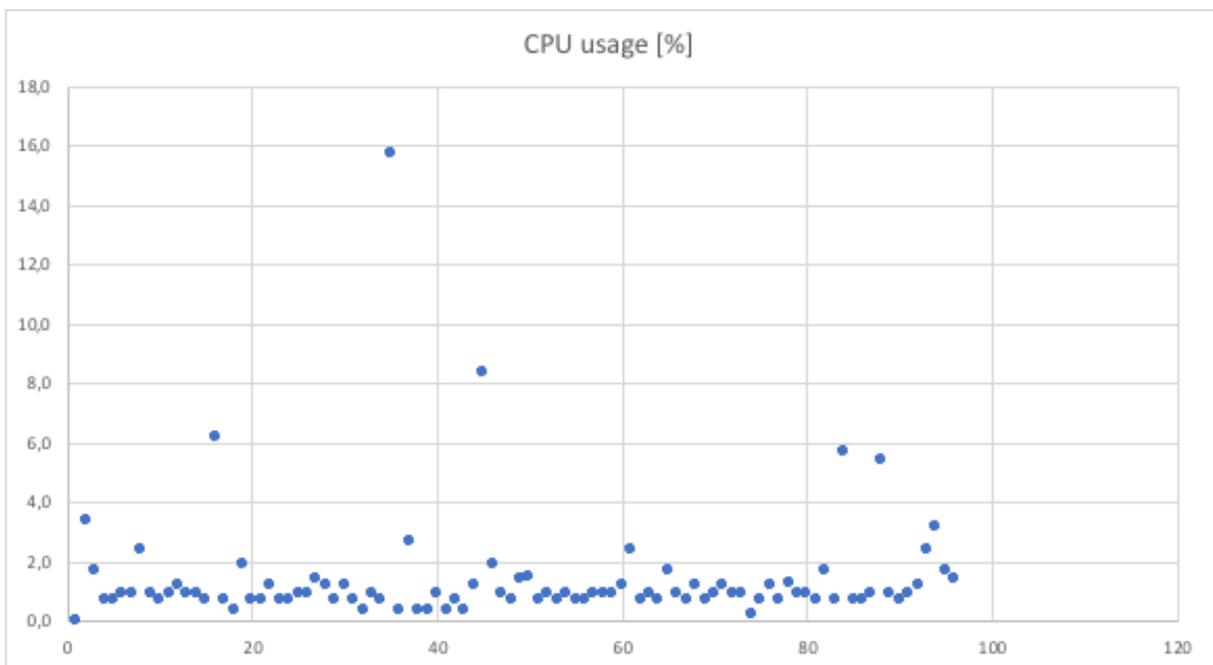


Figura 7. Uso de CPU – Servicio web productos

La Figura 7 presenta el porcentaje de uso de CPU cuando se realizan las distintas invocaciones al servicio web producto.

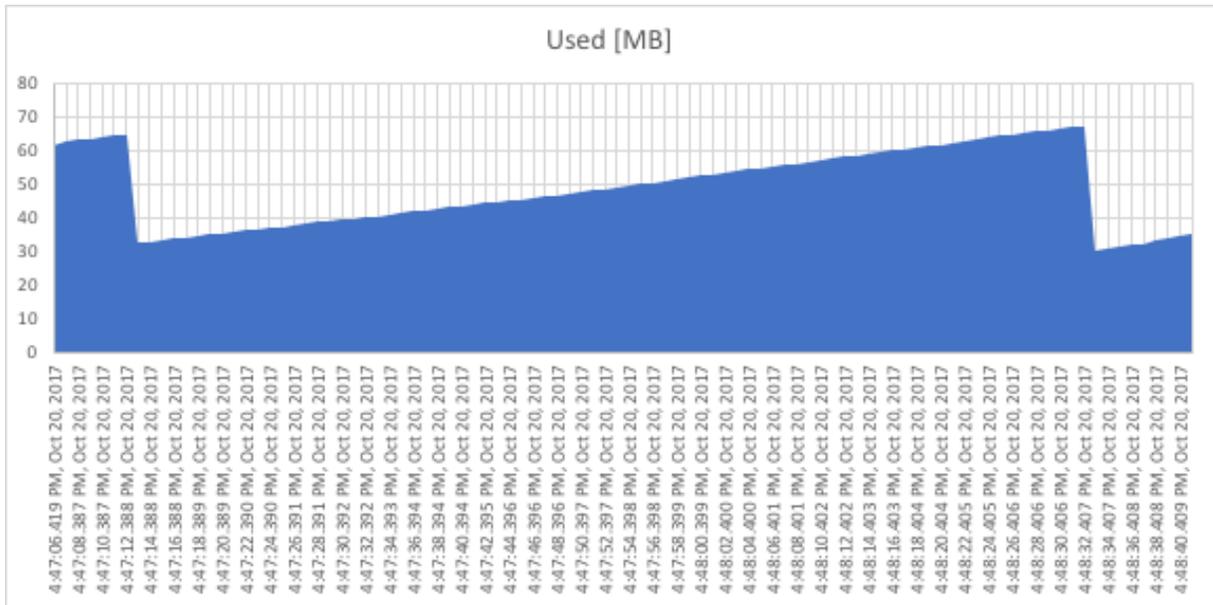


Figura 8. Uso de memoria RAM – Servicio web productos

La Figura 8 presenta el uso de memoria RAM comparado con el tiempo de duración de las distintas invocaciones al servicio web productos.

Servicio web expertos

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web expertos, se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la misma herramienta. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web expertos.

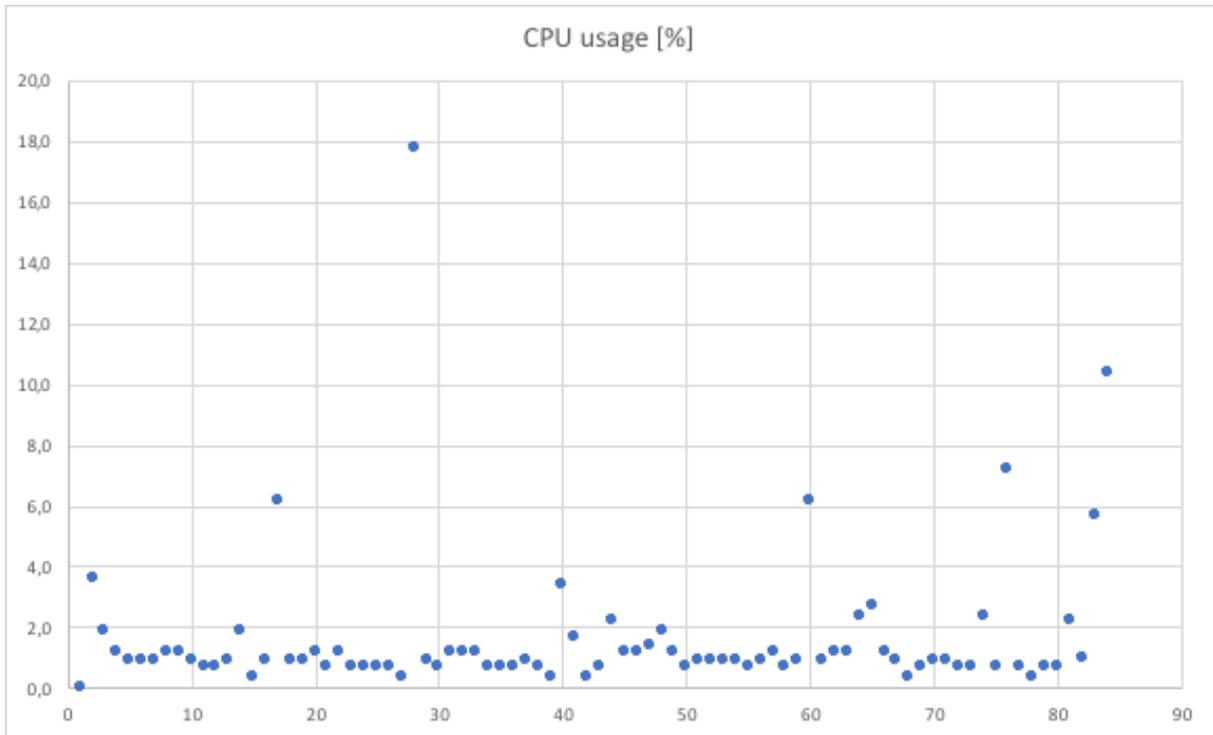


Figura 9. Uso de CPU – Servicio web expertos

La Figura 9 presenta el porcentaje de uso de CPU cuando se realizan las distintas invocaciones al servicio web expertos.

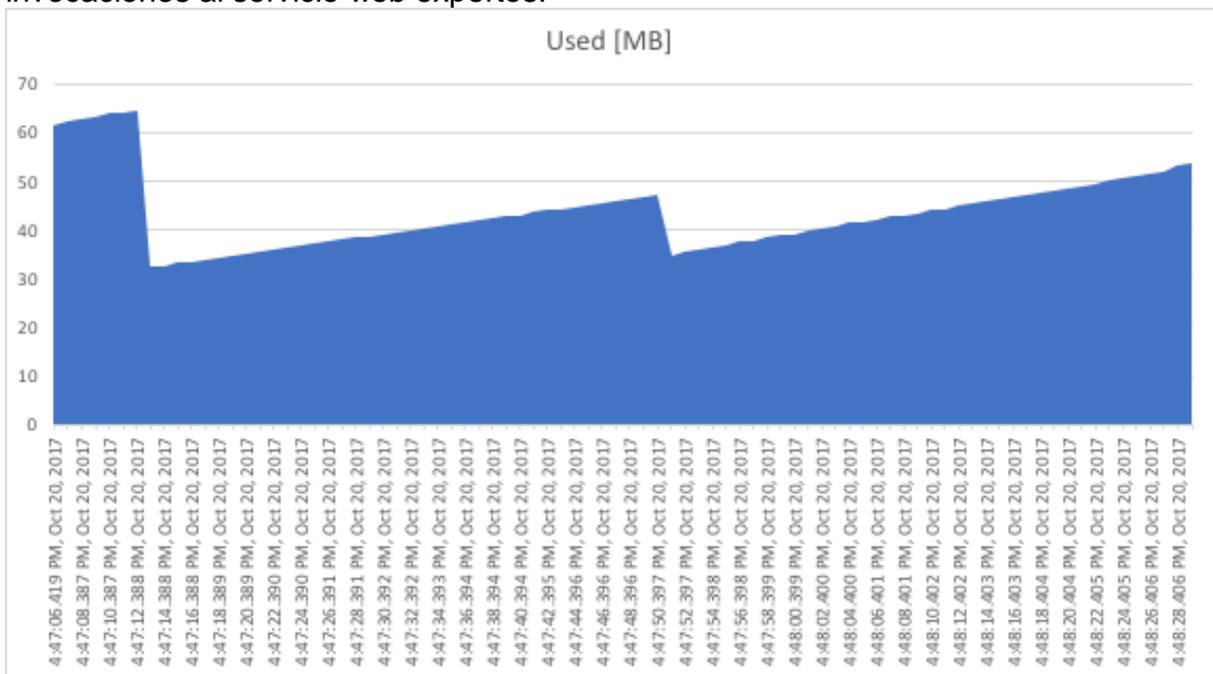


Figura 10. Uso de memoria RAM – Servicio web experto

La Figura 10 presenta el uso de memoria RAM comparado con el tiempo de duración de las distintas invocaciones al servicio web expertos.

Servicio web caso adaptado 1:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web para el caso adaptado 1 (zona cafetera), se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la herramienta VisualVM. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web para el caso adaptado 1.

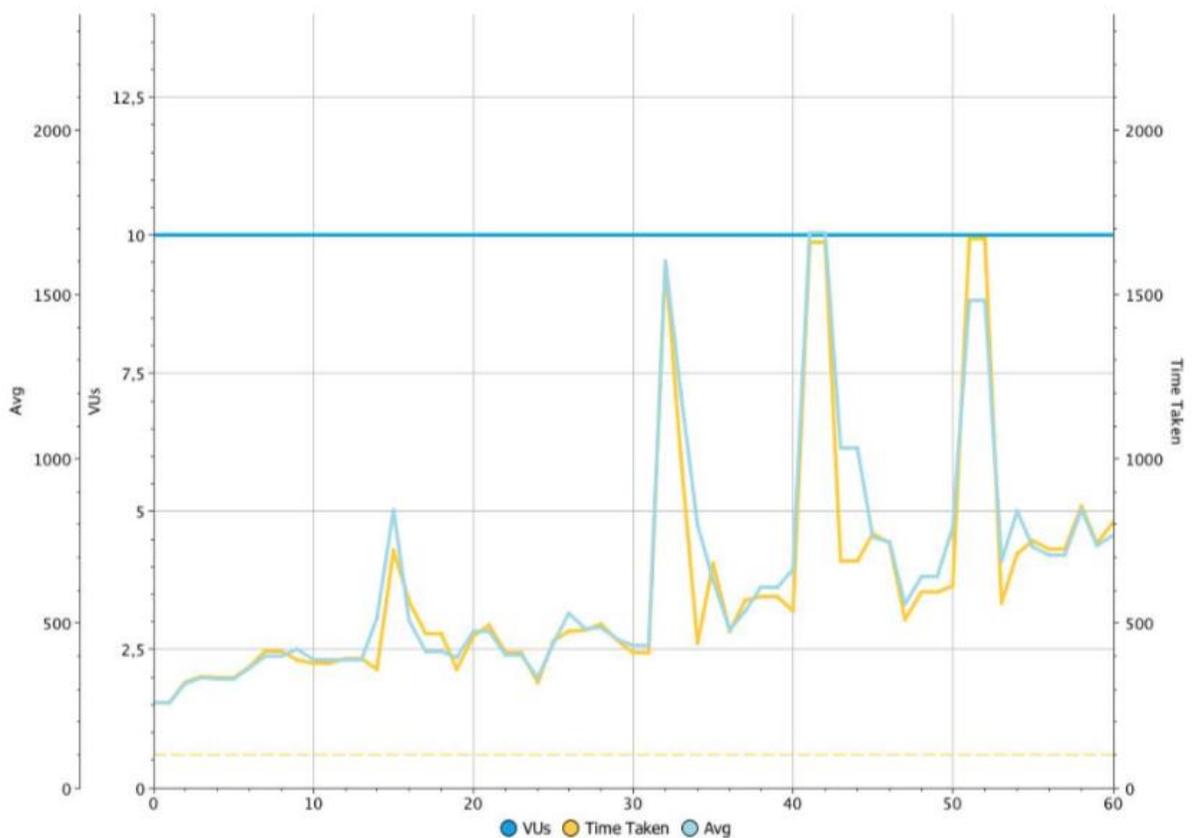


Figura 11. Uso de recursos computacionales – Servicios web caso adaptado 1

La Figura 11 presenta el número de usuarios simulados (VUs), el tiempo en milisegundos que tomado para terminar la prueba (time taken) y el promedio (Avg) del mismo cuando se realizan las distintas invocaciones al servicio web para el caso adaptado 1.

Servicio web caso adaptado 2:

Con el objetivo de medir el consumo de recursos computacionales (uso de CEP y memoria RAM) del servicio web para el caso adaptado 2 (periodos críticos para la generación de la roya), se procedió a realizar 1200 invocaciones al servicio con la herramienta SOAPUI y mientras se generaban estas peticiones se midió la variación en el comportamiento de la CPU y la memoria RAM con ayuda de la herramienta VisualVM. Con el objetivo de conocer el consumo de recursos computacionales por parte del servicio web para el caso adaptado 2.

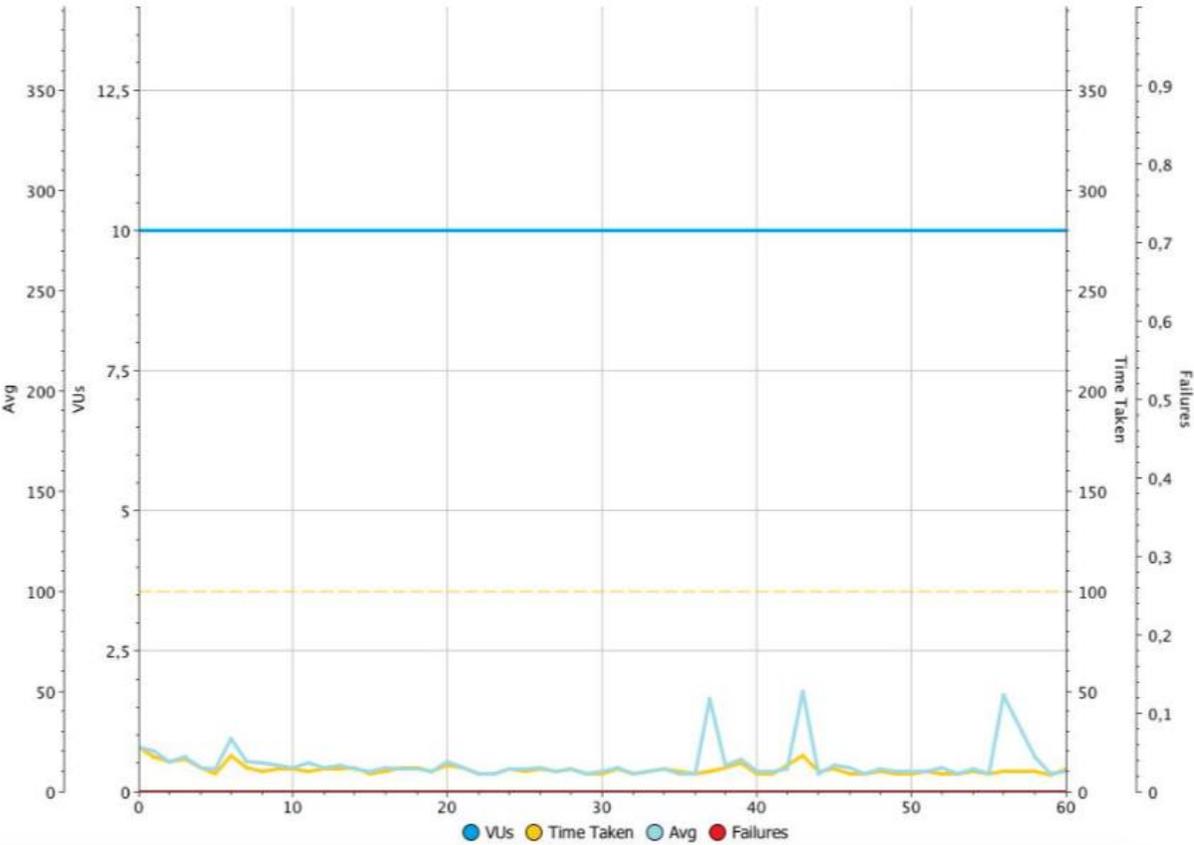


Figura 12. Uso de recursos computacionales – Servicios web caso adaptado 2

La Figura 12 presenta el número de usuarios simulados (VUs), el tiempo en milisegundos que tomado para terminar la prueba (time taken) y el promedio (Avg) del mismo cuando se realizan las distintas invocaciones al servicio web para el caso adaptado 2.

Anexo C

Código fuente

Este anexo presenta algunos de los códigos fuentes que fueron desarrollados dentro del prototipo realizado dentro de este trabajo.

Función que permite crear los eventos complejos:

```
1. public Object onCall(final MuleEventContext eventContext) throws Exception {
2.
3.     final Object payload = eventContext.getMessage().getPayload();
4.     final EPStatement pattern = EsperUtils.addNewEventPattern(payload.toString());
5.
6.     StatementAwareUpdateListener
7.     genericListener = new StatementAwareUpdateListener() {
8.         @Override
9.         public void update(EventBean[] newComplexEvents,
10.            EventBean[] oldComplexEvents,
11.            EPStatement detectedEventPattern, EPServiceProvider
12.            serviceProvider) {
13.
14.             if (newComplexEvents != null) {
15.                 try {
16.                     // newComplexEvents[0].getUnderlying() is the payload of the
17.                     // complex event detected by this listener.
18.                     System.out.println("====complex event
19. payload:" + newComplexEvents[0].getUnderlying());
20.
21.                     // detectedEventPattern.getEventType().getName() specifies the
22.                     // event pattern that has created this complex event.
23.                     String eventPatternName = detectedEventPattern.getEventType
24.                     ().getName();
25.                     System.out.println("====eventPatternName:
26. " + eventPatternName);
27.
28.                     // Create the detected complex event as a Java map
29.                     // (eventPatternName, event properties)
30.                     Map<String,
31.                     Object> complexEvent = new LinkedHashMap<String, Object>();
32.                     complexEvent.put(eventPatternName,
33.                     newComplexEvents[0].getUnderlying());
34.
35.                     // Create the Mule message containing the complex event to be
36.                     // sent to the ComplexEventReceptionAndDecisionMaking flow
37.                     MuleMessage
38.                     message = new DefaultMuleMessage(complexEvent, eventContext.getMuleContext());
39.
40.                     // Add messageProperties, a map containing eventPatternName,
41.                     // to the Mule message
```

```

28.         Map<String,
Object> messageProperties = new HashMap<String, Object>();
29.         messageProperties.put("eventPatternName",
eventPatternName);
30.         messageProperties.put("eventPatternZone",
newComplexEvents[0].get("zona").toString());
31.         message.addProperties(messageProperties,
PropertyScope.OUTBOUND);
32.         if (eventPatternName.equals("variables_groya")) {
33.             //Si el evento es de variables de roya crear evento y
evaluar si cumple con algun patron
34.             //System.out.println("=====MENSAJE EVENTO
COMPLEJO: " + newComplexEvents[0].getUnderlying());
35.             CmpToVarEventTransformer
eventVar = new CmpToVarEventTransformer();
36.             eventVar.sendVarEventToEsper(newComplexEvents[0].get
("min_tmp").toString(), newComplexEvents[0].get("avg_hhreal_noche").toString(),
newComplexEvents[0].get("avg_tmp_hora_noche").toString(),
newComplexEvents[0].get("max_tmp").toString(),
newComplexEvents[0].get("avg_tmp").toString(),
newComplexEvents[0].get("avg_hhrel_dia").toString(),
newComplexEvents[0].get("sombra").toString(),
newComplexEvents[0].get("avg_precipitacion").toString(),
newComplexEvents[0].get("zona").toString());
37.         } else {
38.             // Send the created Mule message to the
ComplexEventConsumerGlobalVM connector located in the
ComplexEventReceptionAndDecisionMaking flow
39.             eventContext.getMuleContext().getClient().dispatch("Compl
exEventConsumerGlobalVM", message);
40.         }
41.     } catch (MuleException e) {
42.         e.printStackTrace();
43.     }
44. }
45. }
46. };
47.
48.     pattern.addListener(genericListener);
49.     return payload.toString();
50. }

```

Función que permite enviar eventos al motor CEP:

```

1. public synchronized void sendEventToEsper(@Payload Map<String, Object> eventMap) {
2.
3.     String eventTypeName = (String) eventMap.keySet().toArray()[0];
4.     // System.out.println("===eventTypeName: " + eventTypeName);
5.
6.     Map<String, Object> eventPayload = new HashMap<String, Object>();
7.     eventPayload = (Map<String, Object>) eventMap.get(eventTypeName);
8.

```

```

9.  Map<String, Object> eventPayloadTypeMap = new HashMap<String, Object>();
10. eventPayloadTypeMap = getEventType(eventPayload);
11. // System.out.println("===eventPayloadTypeMap: " + eventPayloadTypeMap);
12.
13. if (!EsperUtils.eventTypeExists(eventName)) {
14.     EsperUtils.addNewEventType(eventName, eventPayloadTypeMap);
15.     System.out.println("****" + eventName + " event type added to Esper engine");
16. }
17.
18. EsperUtils.sendEvent(eventPayload, eventName);
19. System.out.println("\nSending event '" + eventName + "' to Esper: " + eventPayload);
20. }
21.
22.
23. private Map<String, Object> getEventType(Map<String, Object> eventMap) {
24.
25.     // LinkedHashMap will iterate in the order in which the entries were put into the map
26.     Map<String, Object> eventTypeMap = new LinkedHashMap<String, Object>();
27.
28.     for (String key : eventMap.keySet()) {
29.
30.         Object value = eventMap.get(key);
31.
32.         if (value instanceof Map) {
33.             @SuppressWarnings("unchecked")
34.             Map<String, Object> nestedEventProperty = getEventType((Map<String, Object>) value);
35.
36.             eventTypeMap.put(key, nestedEventProperty);
37.         } else {
38.             eventTypeMap.put(key, value.getClass());
39.         }
40.     }
41.
42.     return eventTypeMap;
43. }

```

Función que permite crear datos aleatorios de clima:

```

1.  @Override
2.  public synchronized Map<String, Object> transformMessage(MuleMessage
message, String outputEncoding)
3.      throws TransformerException {
4.
5.     // LinkedHashMap will iterate in the order in which the entries were put into the map
6.     Map<String, Object> eventMap = new LinkedHashMap<String, Object>();
7.     //ObjectMapper mapper = new ObjectMapper();
8.
9.     double sombra = rand(0, 100);
10.    double precipitacion = rand(0, 200);
11.    double hhumedalreID = rand(0, 12);
12.    double hhumedalreIN = rand(0, 12);
13.    double tempHN = rand(-8, 50);

```

```

14. double temperatura = rand(-8, 50);
15. //Colombia
16. //double lat = rand(-72, -68);
17. //double log = rand(4, 12);
18. //Zona cafetera
19. double lat = rand(-72, -78);
20. double log = rand(0, 12);
21. try {
22.
23.     //Find Zone
24.     //String zone = LocationToZone.consultarLocation(2.4574702, -76.6349538);
25.     String zone = LocationToZone.consultarLocation(log, lat);
26.     // Event payload is a nested map
27.     Map<String, Object> eventPayload = new LinkedHashMap<String, Object>();
28.     eventPayload.put("Sombra", new Double(df2.format(sombra)));
29.     eventPayload.put("Precipitacion", new Double(df2.format(precipitacion)));
30.     eventPayload.put("HHumedadReID", new Double(df2.format(hhumedalreID)));
31.     eventPayload.put("HHumedadReIN", new Double(df2.format(hhumedalreIN)));
32.     eventPayload.put("TemperaturaHN", new Double(df2.format(tempHN)));
33.     eventPayload.put("Temperatura", new Double(df2.format(temperatura)));
34.     eventPayload.put("Zona", zone);
35.
36.     eventMap.put("clima.evento", eventPayload);
37.
38. } catch (Exception e) {
39.     e.printStackTrace();
40. }
41.
42. System.out.println("===ClimaEvent created: " + eventMap);
43. return eventMap;
44. }
45.
46. /**
47.  * Returns a psuedo-random number between min and max, inclusive.
48.  * The difference between min and max can be at most
49.  * <code>Integer.MAX_VALUE - 1</code>.
50.  *
51.  * @param min Minimim value
52.  * @param max Maximim value. Must be greater than min.
53.  * @return Double between min and max, inclusive.
54.  * @see java.util.Random#nextDouble()
55.  */
56. public static double rand(int min, int max) {
57.     // Usually this can be a field rather than a method variable
58.     Random r = new Random();
59.     // nextInt is normally exclusive of the top value,
60.     // so add 1 to make it inclusive
61.     double randomValue = min + (max - min) * r.nextDouble();
62.     return randomValue;
63. }

```

Flujo principal muleESB en XML:

```

1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <mule xmlns:file="http://www.mulesoft.org/schema/mule/file" xmlns:quartz="http://www.mulesoft.org/schema/mule/quartz" xmlns:esper="http://www.mulesoft.org/schema/mule/esper" xmlns:smtmps="http://www.mulesoft.org/schema/mule/smtmps" xmlns:ws="http://www.mulesoft.org/schema/mule/ws" xmlns:vm="http://www.mulesoft.org/schema/mule/vm" xmlns:http="http://www.mulesoft.org/schema/mule/http" xmlns:mulexml="http://www.mulesoft.org/schema/mule/xml" xmlns="http://www.mulesoft.org/schema/mule/core" xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
4.   xmlns:spring="http://www.springframework.org/schema/beans"
5.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6.   xsi:schemaLocation="http://www.springframework.org/schema/beans
7.     http://www.springframework.org/schema/beans/spring-beans-current.xsd
8.     http://www.mulesoft.org/schema/mule/core
9.     http://www.mulesoft.org/schema/mule/core/current/mule.xsd
10.    http://www.mulesoft.org/schema/mule/smtmps
11.    http://www.mulesoft.org/schema/mule/smtmps/current/mule-smtmps.xsd
12.    http://www.mulesoft.org/schema/mule/vm
13.    http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd
14.    http://www.mulesoft.org/schema/mule/ws
15.    http://www.mulesoft.org/schema/mule/ws/current/mule-ws.xsd
16.    http://www.mulesoft.org/schema/mule/http
17.    http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
18.    http://www.mulesoft.org/schema/mule/xml
19.    http://www.mulesoft.org/schema/mule/xml/current/mule-xml.xsd
20.    http://www.mulesoft.org/schema/mule/esper
21.    http://www.mulesoft.org/schema/mule/esper/current/mule-esper.xsd
22.    http://www.mulesoft.org/schema/mule/quartz
23.    http://www.mulesoft.org/schema/mule/quartz/current/mule-quartz.xsd
24.    http://www.mulesoft.org/schema/mule/file
25.    http://www.mulesoft.org/schema/mule/file/current/mule-file.xsd">
26.   <vm:endpoint exchange-pattern="one-way" path="ComplexEventConsumerPath" name="ComplexEventConsumerVM" doc:name="VM"/>
27.   <http:request-
28.     config name="HTTP_Request_Configuration" host="api.thingspeak.com" port="80" doc:name="HTTP Request Configuration"/>
29.     <flow name="clima-adaptationFlow">
30.       <quartz:inbound-
31.         endpoint jobName="j1" repeatInterval="60000" responseTimeout="10000" doc:name="Quartz">
32.           <quartz:event-generator-job/>
33.         </quartz:inbound-endpoint>
34.         <scatter-gather doc:name="Scatter-Gather">
35.           <http:request config-
36.             ref="HTTP_Request_Configuration" path="/channels/67049/feed.json?results=1" method="GET" doc:name="Home 01"/>
37.           <http:request config-
38.             ref="HTTP_Request_Configuration" path="/channels/67064/feeds.json?results=1" method="GET" doc:name="Home 02"/>
39.           <http:request config-
40.             ref="HTTP_Request_Configuration" path="/channels/67065/feeds.json?results=1" method="GET" doc:name="Home 03"/>

```

```

26.     <http:request config-
ref="HTTP_Request_Configuration" path="/channels/67074/feeds.json?results=1" method="G
ET" doc:name="Home 04"/>
27.     </scatter-gather>
28.     <collection-splitter doc:name="Collection Splitter"/>
29.     <custom-
transformer class="transformer.JsonToHomeEventTransformer" doc:name="JsonToHomeEv
entTransformer"/>
30.     <scatter-gather doc:name="Scatter-Gather">
31.         <logger message="#[payload]" level="INFO" doc:name="Logger"/>
32.         <component doc:name="Send HomeEvent to Esper Engine">
33.             <singleton-object class="esper.SendEventToEsper"/>
34.         </component>
35.     </scatter-gather>
36. </flow>
37. <flow name="ComplexEventReceptionAndDecisionMaking">
38.     <vm:inbound-endpoint exchange-pattern="one-
way" ref="ComplexEventConsumerVM" doc:name="ComplexEventConsumerVM"/>
39.     <scatter-gather doc:name="Scatter-Gather">
40.         <processor-chain>
41.             <set-payload value="Detected Alert
#[message.inboundProperties['eventPatternName']]: #[payload]" doc:name="Set Alert
Payload"/>
42.             <smtps:outbound-
endpoint host="smtp.gmail.com" port="465" user="#{email.username}" password="#{email.pa
ssword}" to="#{emailto}" from="#{emailfrom}" subject="Detected
ALERT" responseTimeout="10000" doc:name="Send Alert by Email"/>
43.             </processor-chain>
44.             <logger message="***Complex Event: #[payload]" level="INFO" doc:name="Logger"/>
45.         </scatter-gather>
46.     </flow>
47.     <flow name="EventPatternAdditionToEsper">
48.         <file:inbound-endpoint path="new-eventpattern" moveToDirectory="deployed-
eventpattern" pollingFrequency="2000" fileAge="1000" responseTimeout="10000" doc:name="
New EPL EventPattern">
49.             <file:filename-regex-filter pattern=".+\.epl" caseSensitive="true"/>
50.         </file:inbound-endpoint>
51.         <file:file-to-string-transformer doc:name="File to String"/>
52.         <component doc:name=" Add EventPattern to Esper Engine">
53.             <singleton-object class="esper.AddEventPatternToEsperComponent"/>
54.         </component>
55.         <choice-exception-strategy doc:name="Choice Exception Strategy">
56.             <catch-exception-
strategy when="exception.causeMatches('com.espertech.esper.*')" doc:name="Syntax Error
in EventPattern Code">
57.                 <file:outbound-endpoint path="incorrect-
eventpattern" responseTimeout="10000" doc:name="Incorrect EventPattern Code"/>
58.             </catch-exception-strategy>
59.         </choice-exception-strategy>
60.     </flow>
61. </mule

```