



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA
UDELAR

Un enfoque dirigido por modelos para el procesamiento de eventos complejos con soporte geoespacial aplicado al cumplimiento de regulaciones ambientales

Ing. Federico Herrera

Tesis presentada en cumplimiento parcial de los requisitos para el grado de **Magíster en Informática**
Universidad de la República - Pedeciba Informática

Montevideo, Uruguay, 28 de septiembre de 2024

Directores de Tesis: Dr. Ing. Daniel Calegari
Universidad de la República
Dra. Ing. Laura González
Universidad de la República

Resumen

Las políticas medioambientales permiten sustentar el desarrollo productivo y social de un país. El marco regulatorio que se vincula al manejo de los recursos naturales y el medio ambiente suele contar con requerimientos espaciales, es decir, restringiendo a una zona geográfica la aplicación de una normativa. El cumplimiento de este marco regulatorio depende fuertemente de la existencia de mecanismos de seguimiento y control de los distintos fenómenos medioambientales que ocurren.

Con la explosión de las redes sociales y la Internet de las Cosas han aparecido sensores para medir calidad del aire, contaminación en el agua, entre otros, así como también aplicaciones que permiten compartir, reportar y brindar información diversa. Las tecnologías de Procesamiento de Eventos Complejos (Complex Event Processing, CEP) permiten procesar, analizar y relacionar grandes cantidades de datos heterogéneos, provenientes de diversas fuentes en forma de eventos para detectar, en tiempo real, situaciones críticas o relevantes. Si bien se trata de una tecnología que puede ser utilizada para monitorear eventos vinculados con el cumplimiento de regulaciones ambientales, generalmente no brindan soporte nativo para eventos que entre sus datos cuenten con una componente geoespacial.

El objetivo general de la tesis es desarrollar mecanismos basados en CEP para la verificación de conformidad de leyes medioambientales con requerimientos geoespaciales. Para ello, se propone una solución basada en Ingeniería Dirigida por Modelos que consiste en la extensión de un lenguaje existente para la especificación de CEP con la posibilidad de especificar eventos complejos que consideren la componente geoespacial: eventos con una ubicación y operadores geoespaciales para especificar eventos complejos. Se extiende el lenguaje original y se propone la generación automática de código desde una especificación geoCEP a un motor CEP que permite el monitoreo y control de las regulaciones ambientales. Finalmente, la propuesta se evalúa a través de un caso de estudio basado en normativas medioambientales de Uruguay modelado con la extensión geoespacial propuesta.

Palabras clave: Conformidad de leyes medioambientales, Procesamiento de Eventos Complejos, requerimientos geoespaciales, Ingeniería Dirigida por Modelos

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	3
1.3	Contribuciones	4
1.4	Estructura del documento	5
2	Conocimiento Existente	7
2.1	Conceptos Generales	7
2.1.1	Cumplimiento de Requerimientos de Conformidad	7
2.1.2	Datos Abiertos de Gobierno	8
2.1.3	Service Oriented Architecture	9
2.1.4	Arquitecturas Dirigidas por Eventos	9
2.2	Leyes Medioambientales	10
2.2.1	Estándares Internacionales	11
2.2.2	Legislación en Uruguay	12
2.2.3	Requerimientos de Conformidad y Leyes Medio Ambientales	12
2.3	Estándares de Información Geoespacial	13
2.4	Desarrollo de Software Dirigido por Modelos	18
2.5	Procesamiento de Eventos Complejos	20
2.5.1	Metamodelo para CEP	21
2.6	Trabajo Relacionado	25
2.6.1	Cumplimiento de Leyes	25
2.6.2	Enfoques tecnológicos para asegurar el cumplimiento	26
2.6.3	Reglas geoespaciales en OpenRules	27
2.6.4	CEP con expresividad geoespacial	28
3	Análisis del Problema	31
3.1	Descripción General	31
3.2	Regulación del Agua en Uruguay	32
3.3	Escenario: Monitoreo de la Calidad del Agua	33
3.4	SIMONA: Solución ad-hoc basada en CEP	35
3.5	Resumen	39
4	Extensión Geoespacial CEP para el Control de Leyes Medioambientales	41
4.1	Descripción General	41

4.1.1	Modelo Conceptual de Leyes	42
4.1.2	Plataforma Original:MEdit4CEP	42
4.1.3	Plataforma Extendida	46
4.2	Modelado de Regulaciones Ambientales	49
4.3	Extensión Geoespacial para CEP	51
5	Detalles de implementación	57
5.1	Principales tecnologías utilizadas	58
5.2	Extensión del Editor Gráfico	59
5.2.1	Editor Gráfico extendido	61
5.3	Generación del Código GeoEPL	61
5.4	Librería Java para Esper	63
6	Caso de Estudio	67
6.1	Descripción General	67
6.2	Modelado del Caso de Estudio	69
6.2.1	Detalles de Modelado	71
6.2.2	Modelado Gráfico en el Editor	73
6.3	Control de normativas en ejecución	74
7	Conclusiones y Trabajo Futuro	77
7.1	Resumen y Conclusiones	77
7.2	Problemáticas y Desafíos Identificados	78
7.3	Trabajo a Futuro	79
	Referencias	81

1

Introducción

Esta Tesis aborda el cumplimiento de regulaciones medioambientales y se apoya en múltiples tecnologías y conceptos, como lo son el procesamiento de eventos complejos, los sistemas de información geográfica y su aplicación en el cumplimiento de regulaciones. Este capítulo introductorio presenta la motivación del trabajo, así como los objetivos planteados y contribuciones del mismo.

1.1. Motivación

El manejo adecuado de los recursos naturales y el medio ambiente requiere de la constitución de marcos legales a través de leyes y normativas que van cambiando. Los gobiernos están cada vez más interesados en asegurar el cumplimiento de estas leyes [Kis07], para lo cual es necesario el monitoreo de diversos factores que las mismas involucran (p. ej. nivel de contaminación del aire).

En Uruguay la protección del Ambiente está declarada de interés general por la Constitución¹ en su artículo número 47², en el cual se establecen sanciones para quienes afecten el medio ambiente y en especial se menciona el recurso del agua potable como objeto de protección. Además del compromiso establecido en la Constitución del Uruguay, en las últimas décadas se han generado más normativas que ponen foco en la problemática, destacando la Ley número 17283³ denominada Ley General del Medio Ambiente, del 28 de noviembre del 2000, que busca garantizar la protección medioambiental.

A los efectos de asegurar el cumplimiento de las normativas, se vuelve necesario monitorear

¹La Constitución es la ley fundamental de un país que establece los principios básicos, derechos y estructura del gobierno, y sirve como la norma suprema que guía todas las demás leyes y regulaciones dentro de la nación.

²<https://www.impo.com.uy/bases/constitucion/1967-1967/47>

³<https://www.impo.com.uy/bases/leyes/17283-2000>

los factores que las mismas involucran como podría ser el nivel de contaminación del agua, para lo cual se debería poder establecer si se están violando las normativas establecidas, por ejemplo para el caso del agua por lo establecido en la Ley número 18610 (Ley de Política Nacional de Aguas)⁴. El marco regulatorio medioambiental puede contar con reglas que involucren zonas geográficas de aplicación específicas (p. ej. río, país, departamento, municipio). En Uruguay, por ejemplo, existe el Plan de Acción para la Protección del Agua en la Cuenca del Río Santa Lucía⁵ que cuenta de varias resoluciones ministeriales que aplican a una zona en cuestión y parámetros de elementos presentes en el agua, como Nitrógeno y Fósforo que pueden afectar la potabilidad. Es por esta característica de las leyes medioambientales que la componente geoespacial de los datos que se utilicen para monitorear el cumplimiento de estas leyes resulta fundamental.

La explosión de las redes sociales, la Internet de las Cosas [WF15](Internet of Things, IoT) y la creciente cantidad de sensores disponibles a lo largo de todo el territorio ha llevado a que existan grandes cantidades de datos relativos a los recursos naturales y al medioambiente que pueden ser aprovechados si son procesados correctamente. Por ejemplo, muchas organizaciones utilizan sensores para medir la calidad del aire y la contaminación del agua, entre otros usos. En Uruguay el Observatorio Ambiental Nacional proporciona datos abiertos del propio organismo y de terceros, entre los cuales se encuentran: datos abiertos de la calidad del agua, de la calidad del aire y denuncias medioambientales recibidas a través del Sistema de Denuncias Ambientales⁶. Asimismo, es común encontrar aplicaciones que permiten a ciudadanos comunes compartir, reportar y brindar información diversa. Por ejemplo, en Uruguay existe la aplicación de Vigilancia Ambiental⁷ que permite que los ciudadanos reporten en un mapa distintos tipos de eventos categorizados según el problema a reportar: Agua, Biodiversidad, Enfermedades transmitidas por vectores, Agroquímicos, Gestión de residuos, Aire, Suelo. También existen otras aplicaciones que tienen objetivos menos específicos pero en las cuales se pueden reportar problemas ciudadanos entre los que están la existencia de basureros ilegales en una zona (p. ej. a través de la aplicación PorMiBarrio⁸). Los datos reportados en estas aplicaciones constituyen lo que se conoce como Información Geográfica Voluntaria (Volunteered Geographic Information, VGI) [Goo07].

Las tecnologías de Procesamiento de Eventos Complejos (Complex Event Processing, CEP) permiten procesar, analizar y relacionar grandes cantidades de datos heterogéneos, provenientes de diversas fuentes en forma de eventos para detectar, en tiempo real, situaciones críticas o relevantes [Eck11]. Si bien CEP, al ser ampliamente utilizado en el procesamiento de eventos en tiempo real, es pertinente para monitorear eventos dentro de un marco regulatorio medioambiental, estas tecnologías generalmente no brindan soporte nativo para geoeventos, es decir, eventos que entre sus datos cuentan con una componente geoespacial, por ejemplo, el punto o coordenada en el cual se produjo el evento. Además, estas tecnologías carecen de la

⁴<https://www.impo.com.uy/bases/leyes/18610-2009>

⁵<https://www.gub.uy/ministerio-ambiente/politicas-y-gestion/plan-accion-para-proteccion-del-agua-cuenca-del-santa-lucia>

⁶<https://www.ambiente.gub.uy/oan/datos-abiertos/>

⁷<https://vigilancia.ambiental.uy/>

⁸<https://pormibarrío.uy/>

posibilidad de manejar de forma nativa estos geoeventos, ni proveen operadores (o funciones) específicos para correlacionar eventos en base a su ubicación geográfica, como la distancia entre dos puntos o la intersección de dos zonas.

En el trabajo "Model4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns" [Bou15] se presenta un enfoque para el modelado de sistemas CEP. El trabajo sigue un enfoque de Ingeniería Dirigida por Modelos (MDE) [Ken02], que es un paradigma de Ingeniería de Software que se enfoca en el uso de modelos como el principal artefacto de desarrollo. En MDE, los modelos son utilizados como una abstracción de los sistemas o procesos que se desean construir y, a partir de ellos, se generan de manera automatizada artefactos de software y documentación. La idea central es que los modelos deben ser el punto central del proceso de desarrollo, permitiendo una mejor comprensión del sistema a construir y un mayor nivel de automatización en la generación de código y documentación [Fra03].

Apoyándose en MDE, MEdit4CEP [Bou15] propone un metamodelo para especificar eventos y una transformación de modelos a código para permitir que lo modelado y generado luego pueda ser ejecutado por una tecnología particular de CEP llamada Esper⁹. Al igual que otras propuestas CEP, MEdit4CEP no provee soporte a eventos con una componente geoespacial. Para lograr este soporte, es posible pensar en extender la propuesta, por ejemplo considerando los lineamientos del OGC que define un modelo de datos para objetos espaciales conocido como Simple Features Standard (SFS) [OGCb].

1.2. Objetivos

El objetivo general de la tesis es desarrollar mecanismos basados en CEP para la verificación de conformidad de leyes medioambientales con requerimientos geoespaciales. Para lograr el mismo se definieron los siguientes objetivos específicos:

1. Analizar la problemática del monitoreo del cumplimiento de leyes medioambientales y las soluciones existentes para ello. Para lo que se debe relevar conocimiento existente en Leyes Medioambientales, CEP, Desarrollo Dirigido por Modelos y Sistemas de Información Geográfica.
2. Proponer una solución basada en un enfoque dirigido por modelos para el procesamiento de eventos complejos con soporte geoespacial para monitorear el cumplimiento de las leyes medioambientales
3. Evaluar la solución mediante un escenario de aplicación basado en reglamentación ambiental existente en el Uruguay.

⁹<http://www.espertech.com/products/esper.php>

1.3. Contribuciones

Se desarrolla una extensión geoespacial a un metamodelo de eventos complejos existente [Bou15] que permite la especificación de geoeventos con foco en facilitar el monitoreo y control del cumplimiento de leyes medioambientales. La extensión permite:

- (a) definir eventos que tengan una propiedad de tipo Geometría (p. ej. línea, punto, polígono), a través de la cual es posible especificar la ubicación geográfica del evento;
- (b) especificar eventos complejos utilizando operadores geoespaciales (p. ej. intersección), los cuales permiten correlacionar eventos teniendo en cuenta su ubicación geográfica;
- (c) representar eventos complejos de forma agnóstica con expresividad de eventos geoespaciales.

A nivel tecnológico se extiende el editor gráfico de Medit4CEP [Bou15] con los elementos que permiten representar geoeventos y se incluye la generación del código EPL correspondiente a los mismos. Además, se extiende el motor Esper con una librería para permitir el manejo de geoeventos.

La propuesta se evalúa a través de un caso de estudio basado en normativas medioambientales de Uruguay modelado con la extensión geoespacial propuesta.

Este desarrollo tecnológico permite evaluar la factibilidad técnica de la solución, en particular, a través de su aplicación al caso de estudio planteado.

Finalmente, esta tesis se desarrolla en parte en el contexto del proyecto de investigación «Cumplimiento de Requerimientos de Conformidad en Plataformas de Integración de Servicios Inter-organizacionales en el Marco de la Administración Pública del Uruguay», financiado por la Comisión Sectorial de Investigación Científica (CSIC) de la Universidad de la República. La misma realiza contribuciones en las áreas de interés del proyecto y genera también dos publicaciones:

- [Her16]Federico Herrera, Laura González y Daniel Calegari. «Complex event processing with geospatial support for monitoring and controlling compliance with environmental regulations». En: *XLII Latin American Computing Conference, CLEI 2016, Valparaíso, Chile, October 10-14, 2016*. IEEE, 2016, págs. 1-10. DOI: 10.1109/CLEI.2016.7833414. URL: <https://doi.org/10.1109/CLEI.2016.7833414>
- [Her17]Federico Herrera, Laura González, Daniel Calegari y Bruno Rienzi. «Compliance with Environmental Regulations through Complex Geo-Event Processing». En: *CLEI Electron. J.* 20.2 (2017). DOI: 10.19153/cleiej.20.2.2. URL: <https://doi.org/10.19153/cleiej.20.2.2>

1.4. Estructura del documento

El resto del documento se organiza de la siguiente manera.

En el Capítulo 2 se presentan los conceptos, tecnologías y estándares relevantes para el trabajo.

En el Capítulo 3 se analiza la problemática del cumplimiento de leyes medioambientales que motiva la tesis.

En el Capítulo 4 se propone una extensión a CEP para aplicar al control de las leyes.

Luego, en el Capítulo 5 se detalla la implementación de la solución propuesta y las tecnologías usadas para ese fin.

En el Capítulo 6 se presenta un caso de estudio en el que se aplica la solución a un artículo de una ley de la normativa medioambiental del Uruguay.

Finalmente, en el Capítulo 7 se presentan conclusiones y trabajo a futuro.

2

Conocimiento Existente

En esta sección se presenta el estudio del conocimiento existente relevante para esta tesis. Se realiza un relevamiento de diversos temas contextuales como las Leyes Medioambientales, el Cumplimiento de Requerimientos de Conformidad, y los Datos Abiertos. Además, se abordan los estándares de Información Geográfica, el Procesamiento de Eventos Complejos, y el Desarrollo de Software Dirigido por Modelos.

2.1. Conceptos Generales

2.1.1. Cumplimiento de Requerimientos de Conformidad

En la actualidad las organizaciones afrontan un número creciente de requerimientos de conformidad que se originan de distintas fuentes, como órganos legislativos (p. ej. leyes de protección de datos personales), estándares y códigos de buenas prácticas, organismos reguladores y contratos específicos (p. ej. acuerdos de nivel de servicio), entre otros [Pap11].

La necesidad de cumplir con estos requerimientos hace que las organizaciones tengan que establecer controles internos para garantizar que estos requerimientos se cumplan, ya que de no ser así las mismas se pueden ver enfrentadas a riesgos de litigios y sanciones, incluso penales [Tur11].

El control del cumplimiento de requerimientos de conformidad en tiempo de ejecución monitorea continuamente la conformidad de los sistemas, servicios y procesos mientras que éstos se están ejecutando [RW12]. En estos casos, cuando se detecta que un requerimiento no se cumple, se podría tener situaciones en las cuales es posible reponerse al “no cumplimiento”, por ejemplo, realizando una acción compensatoria. Sin embargo, hay otros casos donde esto no es posible. Cuando se trata con mecanismos que deben dar respuesta en tiempo real, el

manejo de las fuentes de datos es fundamental ya que no solo es importante la procedencia de la misma (p. ej. la web, redes sociales, aplicaciones, sensores), sino también otros atributos como son: los formatos, accesibilidad y disponibilidad de los mismos.

A nivel académico existen varias propuestas orientadas a brindar mecanismos que permitan el monitoreo y control de requerimientos de conformidad que surgen de distintos tipos de normativas. Entre estas propuestas, se destaca el proyecto COMPAS [Dan09] (Compliance-driven Models, Languages, and Architectures for Services) y el proyecto C3PRO [Sem14] (Enabling Change and Compliance For Collaborative Processes) enfocado a procesos colaborativos y entre organizaciones [Knu13].

Por otro lado, el proyecto INSPIRE (Infrastructure for Spatial Information in Europe) [Eur] es un proyecto activo actualmente que aborda temas de Infraestructuras de Datos Espaciales. Entre sus subproyectos existen algunos dedicados a estudiar el monitoreo de variables medioambientales para lo cual generan Lenguajes Específicos de Dominio (Domain Specific Languages, DSL) para representarlos. En particular tienen una representación para los distintos factores que son de interés monitorear por las leyes medioambientales, por ejemplo: agua, aire, suelo, etc., y también permite relacionarlos con leyes que los afectan. Este proyecto se enfoca únicamente en la definición de DSLs, pero no propone mecanismos para monitorear las variables medioambientales en tiempo de ejecución.

Por otro lado, en Uruguay se trabajó en un proyecto CSIC¹ de I+D denominado «Cumplimiento de Requerimientos de Conformidad en Plataformas de Integración de Servicios Inter-organizacionales en el Marco de la Administración Pública del Uruguay»². Este proyecto buscó proponer soluciones en distintos planos para ser incorporadas en una Plataforma de Integración Inter-organizacional con el objetivo de controlar requerimientos de conformidad con leyes, normas técnicas, especificaciones, entre otros, de los sistemas y servicios que la plataforma integra. Usando soluciones que se apoyan en tecnologías emergentes, se buscó también abordar contextos particulares de interés para la Administración Pública en Uruguay, como lo son la salud, el medioambiente y servicios geoespaciales.

2.1.2. Datos Abiertos de Gobierno

Según el W3C³ los datos son abiertos si cualquier persona es libre de utilizarlos, reutilizarlos y redistribuirlos. Las características principales de los datos abiertos incluyen su disponibilidad y acceso, la reutilización y redistribución de los mismos, y fundamentalmente, la ausencia de restricción tecnológica, lo que implica también el ser publicados en formatos abiertos.

El grado de apertura de los datos suele clasificarse utilizando el esquema de «cinco estrellas» propuesto por Tim Berners-Lee en el año 2010 [HB11] que se muestra en la Figura 2.1.

¹CSIC - Comisión Sectorial de Investigación Científica: <https://www.csic.edu.uy/>

²<https://www.csic.edu.uy/content/cumplimiento-de-requerimientos-de-conformidad-en-plataformas-de-integración-de-servicios>

³<https://www.w3.org/>

Estrellas	Tecnología y Entrega de Datos
1	Datos disponibles en la web, en cualquier formato, pero con una licencia abierta (Open Data)
2	Datos estructurados, e.g. Excel en lugar de escaneo de imagen de una tabla
3	Formato no propietario, e.g. CSV en lugar de Excel
4	Incluye estándares (W3C), e.g. RDF, y SPARQL, se utiliza URI (Uniform Resource Identifier)
5	Datos vinculados a otros datos para proporcionar un contexto

Figura 2.1: 5 estrellas de Tim Berners-Lee

En los últimos años, varias iniciativas de Datos Abiertos de Gobierno (Open Government Data, OGD) han surgido en el mundo con el objetivo de lograr que los datos públicos sean de libre acceso para todas las personas [Kal11]. En particular, la iniciativa de Datos Abiertos de Uruguay⁴ ha dejado a disposición un Catálogo de Datos Abiertos⁵. De esta forma, la información pública puede reutilizarse y combinarse con otro contenido digital para crear servicios y productos con un valor agregado.

2.1.3. Service Oriented Architecture

La Arquitectura Orientada a Servicios (SOA) no es un producto ni una tecnología, es un estilo arquitectónico que define la utilización de servicios como unidades básicas de desarrollo de aplicaciones. Bajo el paradigma de SOA las funcionalidades son expuestas como servicios independientes, con interfaces estándares para poder ser invocadas. El objetivo de estas arquitecturas es el de mantener el bajo acoplamiento entre los componentes (proveedores y consumidores de servicios), por esto son soluciones eficientes cuando se cuenta con sistemas modulares que se requiere comunicar generando sistemas distribuidos compuestos de componentes reutilizables y compartibles (servicios). Las interfaces de los servicios deben estar bien definidas y agnósticas a su implementación para facilitar la integración, escalabilidad y adaptación [Moh17].

2.1.4. Arquitecturas Dirigidas por Eventos

Un evento es un cambio en el estado de algo, algo que ocurre o se espera que ocurra o una condición detectable que puede lanzar una notificación [Soc10]. Un evento complejo implica la ocurrencia de varios eventos simples, por ejemplo en la Figura 2.2 se puede observar que para ser sospechoso de tener covid-19 deben ocurrir ciertos síntomas como tos seca y fiebre.

⁴<https://datos.gub.uy/>

⁵<https://catalogodatos.gub.uy/>

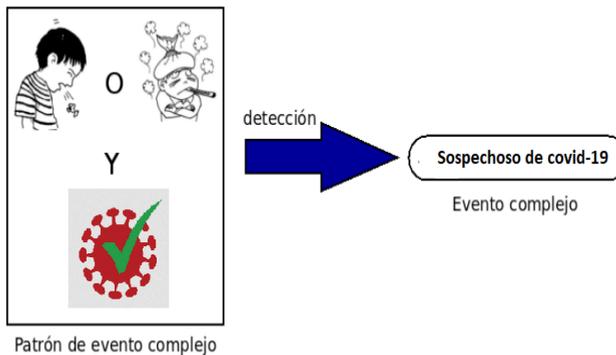


Figura 2.2: Evento Complejo

Una arquitectura dirigida por eventos (EDA) es un patrón de arquitectura de software en el que algunos de los componentes de un sistema son dirigidos por eventos y tienen bajo acoplamiento entre el productor y consumidor de los mismos.

En [WR10] se define que cualquier aplicación de negocio que implemente EDA debe cumplir los siguientes principios:

- Notificación de eventos: se debe informar en cuanto acontezca un nuevo evento en el sistema.
- Envío de notificaciones: el productor de eventos será el que decida cuando enviar las notificaciones de los eventos al consumidor de eventos.
- Respuesta inmediata: el consumidor de eventos reaccionará llevando a cabo alguna acción, tan pronto como reconozca un evento.
- Comunicación unidireccional: el productor de eventos envía las notificaciones de los eventos al consumidor de eventos, sin esperar ningún tipo de respuesta por parte del consumidor.
- Notificaciones libre de peticiones: las notificaciones de eventos son noticias en las que no se detallará ninguna de las acciones que deban ser ejecutadas por el consumidor de eventos, tras su recepción.

2.2. Leyes Medioambientales

En el ámbito del derecho, las leyes, normas, decretos y regulaciones actúan como instrumentos legales que permiten regular las acciones de los ciudadanos y las empresas dentro de un

territorio determinado. Estos textos legales establecen mandatos o prohibiciones, fundamentados en principios de justicia y en el bienestar de la sociedad en su conjunto. El medioambiente, siendo un tema de interés público, es susceptible de regulación a diversos niveles: regional, nacional e incluso internacional [uta][Oec]

2.2.1. Estándares Internacionales

Existen estándares internacionales en lo que respecta al cuidado del medioambiente. Los estándares ISO 14000 [ISO22] fueron concebidos con el fin de establecer procesos productivos más efectivos y ambientalmente responsables. Involucran distintos aspectos como especificaciones exactas, establecimiento de compromisos, procedimientos e instrucciones precisas, desarrollo de procesos, reducción de desechos, niveles de desempeño ambiental y la protección del entorno.

Las normas y su aplicación son el instrumento con el cual se busca que las organizaciones logren cumplir con los compromisos ambientales establecidos. Las normas son globales, están dirigidas a cualquier tipo de organización (son adaptables) y aplican a sus actividades, productos y servicios. Las principales normas de la serie son:

- **Sistemas de Manejo Ambiental. (ISO 14001 - 14004):** La norma ISO 14001 presenta los requerimientos básicos que debe seguir el sistema de gestión ambiental de una empresa dada, de manera que el mismo pueda ser objeto de una certificación. Tiene vínculos con la norma ISO 9000 [ISO15] que refiere a los Sistemas de Gestión de Calidad (SGC) y está diseñada para ayudar a las organizaciones a garantizar que cumplan con las necesidades de los clientes y otras partes interesadas, mientras cumplen con los requisitos legales y reglamentarios relacionados con el producto o servicio. Por otro lado, la norma ISO 14004 sirve de apoyo a la ISO 14001 e incluye principalmente recomendaciones e información sobre mejores prácticas.
- **Auditoría Medioambiental (ISO 14010 -14015):** Estas normas se encargan de proporcionar los parámetros necesarios para la ejecución de auditorías ambientales. En particular establece cuáles son sus principios, procedimientos de investigación y de evaluación del grado de cumplimiento.
- **Eco-etiquetado ambiental (ISO 14020-14024):** Proporcionan los parámetros relativos a procesos de mercadeo y propaganda con conceptos ambientales.
- **Evaluaciones de desempeño ambiental (ISO 14031):** Busca establecer las pautas que permitan construir un proceso para medir, analizar y determinar el desempeño ambiental de una organización. El principal aporte de esta norma está en que incorpora los indicadores de desempeño ambiental como herramienta de evaluación para evaluar el desempeño
- **Análisis del Ciclo de Vida (ISO 14041 - 14044):** Son normas que regulan el ciclo de vida del producto. Desde la elección y uso de la materia prima hasta el producto final,

desechos y distribución.

2.2.2. Legislación en Uruguay

La Constitución de la República declara en el artículo 47 que es de interés general la protección del medioambiente. Especifica que: “Las personas deberán abstenerse de cualquier acto que cause depredación, destrucción o contaminación graves al medioambiente.” También refiere al agua como recurso natural esencial para la vida y la importancia de legislar sobre el acceso y saneamiento. La ley vigente que regula en la materia es la 17.283 promulgada el 28/11/2000 y Reglamentada por el decreto N° 152/013 del 21/05/2013 [IMPb][IMPa].

Ley 17.283 - Protección del Medioambiente

La ley está dividida en cuatro capítulos. En el capítulo I, Disposiciones introductorias, establece las obligaciones de los habitantes de la República en lo que respecta al cuidado del medioambiente, así como también de personas físicas y jurídicas tanto públicas como privadas. Es deber del estado el controlar y penalizar a quien no cumpla con las disposiciones establecidas.

En el capítulo II: Disposiciones Generales, la ley establece los principios de la política medioambiental. En particular el Artículo 6, Inciso F, indica que: “La gestión ambiental debe basarse en un adecuado manejo de la información ambiental, con la finalidad de asegurar su disponibilidad y accesibilidad por parte de cualquier interesado”. Es claro que en este aspecto los sistemas informáticos se vuelven fundamentales para el apoyo en el cumplimiento de la misma.

En el capítulo III: Disposiciones Especiales, se abordan distintos problemas y disposiciones a cumplir sobre los mismos, estos son: calidad del aire, capa de ozono, cambio climático, sustancias químicas, residuos, diversidad biológica y bioseguridad.

Finalmente la ley destina un capítulo de otras disposiciones en el cual se aborda el tema del agua, en particular el “Inventario hídrico” y las costas. También se incluyen temas punitivos por el no cumplimiento de las disposiciones mencionadas.

2.2.3. Requerimientos de Conformidad y Leyes Medio Ambientales

Las regulaciones ambientales comprenden pautas, especificaciones o leyes diseñadas para administrar los recursos ambientales con el fin de proteger el medioambiente natural y mitigar los efectos de la contaminación en la salud humana y los ecosistemas [Lei09] [Fri03].

El desarrollo de regulaciones ambientales se ha convertido en una actividad importante para

la mayoría de los gobiernos. Por ejemplo, la Agencia de Protección Ambiental (EPA)⁶ de los Estados Unidos de América tiene la misión de proteger la salud humana y el medioambiente. Con este fin, la agencia desarrolla y hace cumplir las regulaciones ambientales⁷ que abarcan muchos temas ambientales, desde la reducción de la lluvia ácida hasta la restauración de humedales. Además, la Dirección General de Medio Ambiente es el departamento de la Comisión Europea a cargo de la política de la Unión Europea (UE) sobre medioambiente. Este departamento "tiene como objetivo proteger, preservar y mejorar el medioambiente para las generaciones presentes y futuras, proponiendo e implementando políticas que garanticen un alto nivel de protección ambiental y preserven la calidad de vida de los ciudadanos de la UE"⁸.

El cumplimiento ambiental se puede definir como el estado de estar de acuerdo con las regulaciones ambientales [Lei09; Fri03]. A medida que aumenta el número y la complejidad de estas regulaciones, lograr el cumplimiento ambiental se convierte en una tarea cada vez más ardua para los gobiernos. En este contexto, el monitoreo ambiental es fundamental para garantizar la seguridad pública y proporcionar información para los sistemas de apoyo a las decisiones [Res09]. El monitoreo ambiental involucra actividades para observar continuamente y medir regularmente los parámetros ambientales de áreas específicas con el fin de identificar cambios ambientales y ayudar a la toma de decisiones en una extensión geoespacial determinada [Kru10]. De hecho, esta actividad es crucial para controlar el cumplimiento ambiental dado que proporciona los valores de diferentes parámetros (por ejemplo, pH) que pueden incluirse en las regulaciones (por ejemplo, regulaciones de agua).

2.3. Estándares de Información Geoespacial

El Open Geospatial Consortium (OGC)⁹ es una organización internacional integrada por empresas privadas, agencias gubernamentales, centros de investigación y universidades para la promoción de la información geoespacial abierta disponible en el mundo. Con este fin, OGC desarrolla especificaciones que están disponibles públicamente. Estas especificaciones, conocidas como estándares OGC, proporcionan un marco para que los desarrolladores creen software que permita a los usuarios acceder, visualizar, analizar y procesar información geoespacial de una variedad de fuentes a través de una interfaz informática genérica, utilizando servicios web.

La arquitectura basada en web services definida por OGC que se muestra en la Figura 2.3 se basa en los siguientes principios[OGCa]:

- **Los componentes del servicio están organizados en varios niveles.** Los niveles son: el nivel de Cliente, el nivel de Servicios de Aplicación, el nivel de Servicios de Proce-

⁶<https://www.epa.gov/>

⁷<https://www.epa.gov/laws-regulations>

⁸<http://ec.europa.eu/dgs/environment>

⁹<http://www.opengeospatial.org/>

samiento y el nivel de Servicios de Gestión de la Información. La jerarquía de niveles es flexible en el sentido de que un determinado nivel puede omitir el nivel que está directamente debajo. Un servicio en el nivel de procesamiento también puede usar otro servicio de su mismo nivel.

- **Los servicios son autodescriptivos y encadenables:** Todos los servicios se pueden descubrir y consumir mediante una interacción publicar-buscar-vincular típica de las arquitecturas orientadas a servicios (SOA). Los servicios se pueden encadenar con otros servicios.
- **Los servicios usan estándares de Internet abiertos:** Para implementar estos servicios, solo se requieren estándares web ampliamente utilizados, como HTTP GET, HTTP POST, URL, tipos MIME y XML. SOAP y WSDL son opcionales.
- **Las interfaces de servicio ofrecen algunas operaciones estáticas.** Las interfaces de servicios web proporcionan solo unas pocas operaciones por especificación de servicio, ya sean obligatorias u opcionales. Dado que las operaciones se definen en una especificación de interfaz estándar, los servicios se pueden implementar mediante software comercial estándar (COTS), como los servidores de mapas de Internet.
- **Los servicios no cuentan con estado (stateless).** Los métodos de los servicios no conservan el estado (a menos que se especifique lo contrario) y no mantienen datos de sesión entre invocaciones.

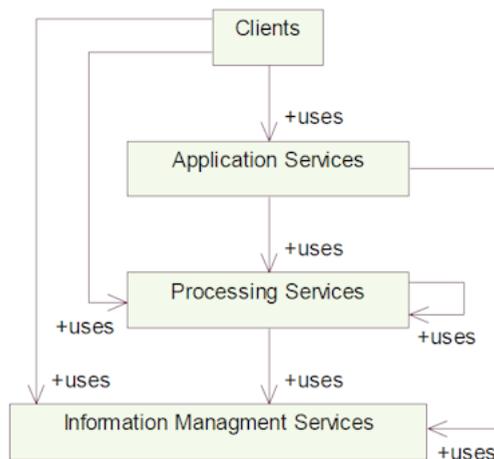


Figura 2.3: Arquitectura - OGC Web Services[OGCa]

Cada nivel de servicios incluye varios tipos específicos de servicios. La mayoría de esos tipos de servicios tienen un estándar de servicio web OGC asociado (por ejemplo, Web Map Service) como se describe a continuación. Vale la pena señalar que muchas normas OGC

también se han convertido en normas ISO.

- **Application Services.** Los servicios de aplicación están destinados a admitir clientes básicos, especialmente clientes ligeros como los navegadores web, llevando algunas tareas pesadas del lado del cliente al lado del servidor. Estos servicios permiten a los usuarios interactuar con los servicios de los niveles subyacentes. Algunos ejemplos son los servicios de portal web, los servicios de descubrimiento de datos geográficos, los servicios de control de acceso, entre otros. Los servicios en este nivel no siguen necesariamente una especificación estándar de OGC, por lo que podrían implementarse como un servicio web SOAP o REST específico de la aplicación. No obstante, también podrían implementarse como un servicio de procesamiento web OGC (WPS) ya que es el más genérico de todos los servicios web OGC.
- **Processing Services.** Este nivel contiene servicios diseñados para procesar datos. El servicio de procesamiento Web ¹⁰ (WPS) y el servicio de transformación de coordenadas¹¹ (CT) (que proporciona una forma estándar para que el software especifique y acceda a los servicios de transformación de coordenadas para su uso en datos espaciales especificados).
- **Information Management Services.** Los servicios de gestión de información están diseñados para almacenar datos o metadatos y proporcionar acceso a ellos, abarcando frecuentemente múltiples conjuntos de datos. El Web Map Service¹² (WMS), Web Feature Service¹³ (WFS), Sensor Observation Service¹⁴ (SOS) y Catalog Service¹⁵ (CSW) son algunos ejemplos de estos servicios.

En particular, OGC define un modelo de datos para objetos geoespaciales conocido como Simple Features Standard (SFS) [OGC]. SFS especifica un modelo de objetos en el cual se encuentra la clase Geometry en la base de la jerarquía y permite representar geometrías planas (hasta 2 dimensiones). Una Geometría está vinculada a un Sistema de Referencia Espacial (SRS) que da significado a las coordenadas de la geometría en referencia a la superficie de la Tierra, especificando el sistema de coordenadas, elipsoide, proyección, datum y otros parámetros geodésicos relevantes.

¹⁰<http://http://www.opengeospatial.org/standards/wps>

¹¹<http://http://www.opengeospatial.org/standards/ct>

¹²<http://http://www.opengeospatial.org/standards/wms>

¹³<http://http://www.opengeospatial.org/standards/wfs>

¹⁴<http://http://www.opengeospatial.org/standards/sos>

¹⁵<http://http://www.opengeospatial.org/standards/cat>

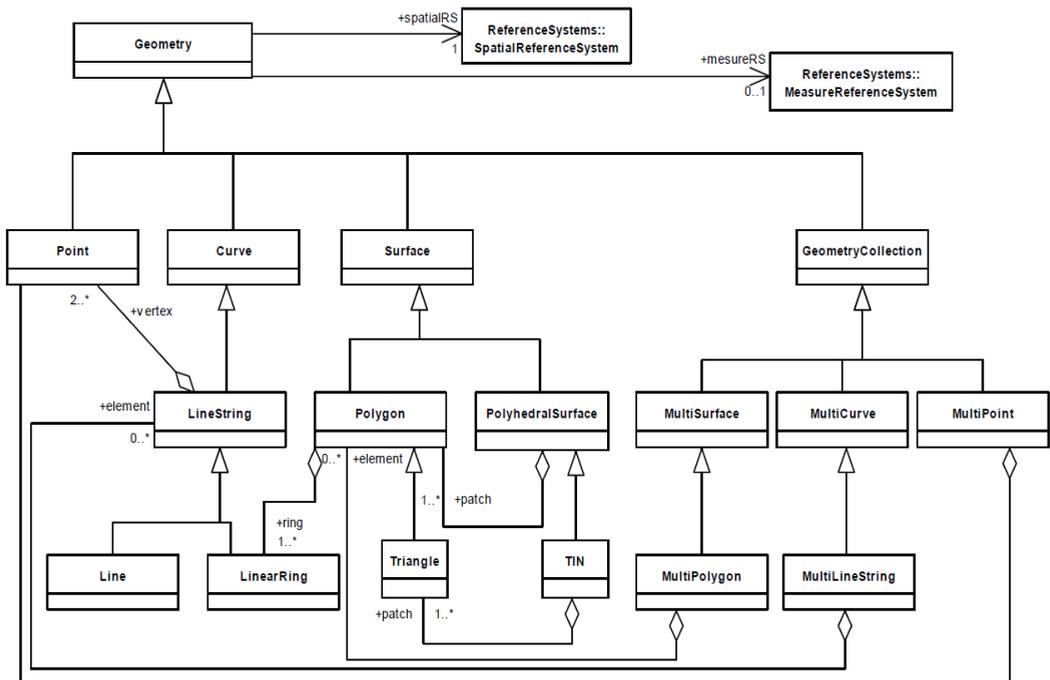


Figura 2.4: Jerarquía de Clases SFS [OGCa]

Como se muestra en la Figura 2.4, el objeto **Geometry** es del cual heredan los demás objetos de la jerarquía definida por OGC. A su vez los métodos del objeto pueden ser observados en la Figura 2.5 y se clasifican según el tipo, en:

- Métodos básicos que proveen información sobre el objeto (dimensión, tipo de geometría, sistema de referencia (SRID)).
- Métodos para comprobar relaciones espaciales entre objetos geométricos (operadores como: `Crosses`, `Contains` o `Intesects`).
- Métodos que efectúan algún tipo de análisis (operadores como: `Union`, `Distance` o `Area`).

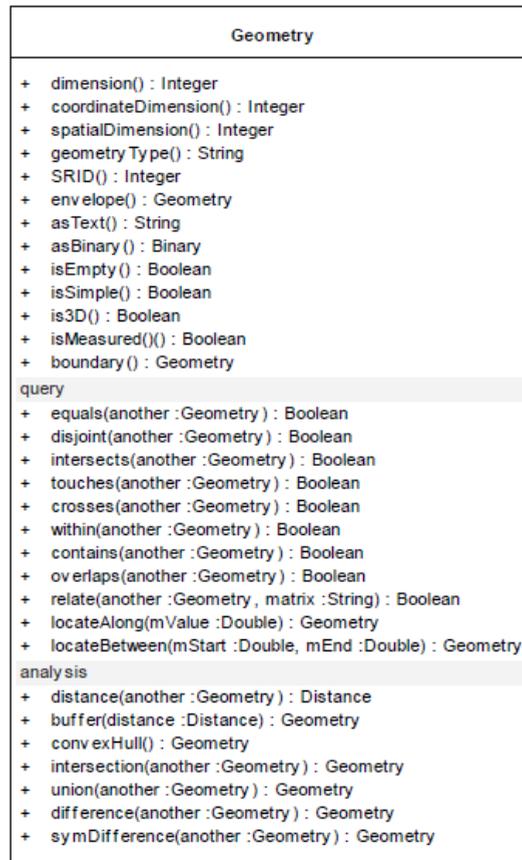


Figura 2.5: Propiedades y Métodos de la Clase Geometry [OGCb]

Algunas de las subclases concretas de Geometry más utilizadas son Point, LineString y Polygon. Los siguientes ejemplos muestran cómo se puede representar y almacenar cada uno de ellas utilizando la representación de texto bien conocido (WKT) [OGCa] en un espacio bidimensional.

- 'POINT(x y)': es cualquier Punto.
- 'LINESTRING(x1 y1, x2 y2, x3 y3)': es un LineString de tres puntos.
- 'POLYGON((x1 y1, x2 y2, x3 y3, x4 y4, x1 y1))': es un Polígono con un anillo exterior de cuatro puntos.

Dado que WKT no proporciona información del SRS, el identificador del SRS (SRID) debe proporcionarse en otro lugar. Por ejemplo, si se utilizan coordenadas geográficas (latitud y

longitud) con el SRS WGS84¹⁶, el SRID será 4326.

El estándar SFS se implementa en muchas interfaces de programación de aplicaciones (APIs), como Java Topology Suite¹⁷ (JTS) para el lenguaje de programación Java, y en muchas bases de datos como PostgreSQL¹⁸ con la extensión PostGIS¹⁹.

2.4. Desarrollo de Software Dirigido por Modelos

El desarrollo de software dirigido por modelos (MDD) es una técnica de ingeniería de software que utiliza modelos como artefactos principales en todo el proceso de desarrollo. Esta técnica se basa en el paradigma más amplio de la ingeniería dirigida por modelos (MDE), que se refiere a la aplicación sistemática de modelos en todo el ciclo de vida del software, incluyendo no solo el desarrollo sino también la verificación, validación y mantenimiento [Küh08] [Sel03] [Béz05].

MDE

La ingeniería dirigida por modelos (MDE) es un enfoque sistemático para el desarrollo de software que utiliza modelos como artefactos principales en todo el ciclo de vida del software. El objetivo principal del MDE es mejorar la calidad y eficiencia del proceso de desarrollo mediante el uso de modelos.

Etapas del MDE

El MDE se compone principalmente de tres etapas:

1. Modelado: En esta etapa se crean los modelos que representan diferentes aspectos del sistema. Estos modelos pueden ser creados utilizando lenguajes específicos para modelado, como UML o SysML [Obj17b].
2. Transformación: En esta etapa se realizan transformaciones entre los diferentes modelos creados en la etapa anterior. Estas transformaciones pueden ser manuales o automáticas y permiten generar automáticamente código a partir de los modelos [Obj14].
3. Validación: En esta etapa se verifica que los modelos y las transformaciones sean correctas y cumplan con los requisitos del sistema [Obj17a].

Metamodelos

Un metamodelo es un modelo que describe las características y restricciones de un lenguaje de modelado. En otras palabras, un metamodelo define las reglas y la sintaxis que se deben

¹⁶<http://gisgeography.com/wgs84-world-geodetic-system/>

¹⁷<http://tsusiatsoftware.net/jts/main.html>

¹⁸<https://www.postgresql.org/>

¹⁹<http://www.postgis.net/>

seguir al crear modelos en un lenguaje de modelado específico.

Los metamodelos son importantes en MDE porque permiten definir y estandarizar los lenguajes de modelado utilizados en el proceso de desarrollo. Además, los metamodelos también se utilizan para generar automáticamente herramientas de software que permiten crear, editar y validar modelos [Ste08].

MDD y los metamodelos

El MDD identifica distintos tipos de modelos con alto nivel de abstracción. Según [Bal06] son modelos que elevan el nivel de desarrollo del software y reducen las diferencias entre los dominios de la tecnología; es decir modelos independientes de cualquier metodología computacional, llamados CIM(Computational Independent Model), los modelos independientes de cualquier tecnología de implementación llamados PIM (Platform Independent Model), modelos que especifican el sistema en términos de construcciones de implementación disponibles en alguna tecnología específica, conocidos como PSM(PlatformSpecific Model), y los Modelos que representan el código fuente.

MDD propone transformaciones entre modelos automatizadas. Las etapas principales del proceso de desarrollo de MDD se pueden ver en la Figura 2.6.

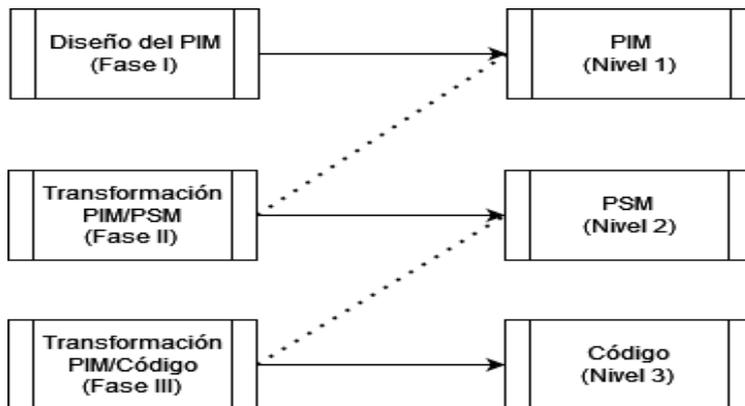


Figura 2.6: Proceso MDD [Bal06]

Diseño del PIM (Platform Independent Model): En esta fase, se crea un modelo independiente de la plataforma que describe el sistema y sus requisitos. Este modelo se enfoca en los aspectos funcionales y no en los detalles de implementación.

Transformación de PIM a PSM (Platform-Specific Model): En esta fase, se transforma el modelo independiente de la plataforma en un modelo específico de la plataforma que describe las construcciones de implementación disponibles para una tecnología específica.

Transformación del código: En esta fase, se genera el código fuente a partir del modelo

específico de la plataforma. Esta transformación puede ser realizada manualmente o mediante herramientas automatizadas.

En resumen, un metamodelo es una representación que permite definir formalmente lenguajes de modelado sin ambigüedades para que una herramienta de transformación pueda leer, escribir y entender los modelos, el metamodelo en sí es un modelo, y se utiliza para precisar cómo se representan los conceptos de la realidad en términos de elementos de modelado. El metamodelo describe la sintaxis abstracta del lenguaje.

Por otro lado, las transformaciones son el proceso en donde se toma un modelo de entrada y se produce otro modelo como salida. Estas deben especificar como generar modelos de destino en función de un conjunto de modelos de origen.

2.5. Procesamiento de Eventos Complejos

El Procesamiento de Eventos Complejos (Complex Event Processing, CEP) refiere a métodos, técnicas y herramientas para procesar eventos mientras estos ocurren y para derivar eventos de más alto nivel a partir de una combinación de eventos de más bajo nivel, de forma permanente y oportuna [Eck11]. Para esto, se monitorea continuamente el flujo de eventos entrantes mediante consultas de eventos (event queries) las cuales especifican situaciones como una combinación de eventos que ocurren, o no ocurren, a lo largo del tiempo. Las plataformas para CEP proveen soporte para varios tipos de patrones de eventos (p. ej. lógicos, temporales) que facilitan la especificación de las consultas de eventos [EN10].

Existen múltiples enfoques y clasificaciones de las soluciones CEP. Respecto a cómo modelar el dominio en particular se identifican dos corrientes: aquellas que usan lenguajes ontológicos buscando dotar de semántica a la información para así modelar el dominio [Cal16]; y aquellas que emplean técnicas de Model Driven Development (MDD) con el cometido de definir los dominios como modelos.

Actualmente, hay disponible diversos productos que dan soporte a la tecnología CEP. En particular, Esper²⁰ es un motor CEP de la compañía EsperTech Inc, de código abierto (licencia GPL) y disponible en Java y .NET. Esper permite ejecutar acciones personalizadas cuando se cumplen ciertas condiciones sobre los flujos de eventos. Estas consultas o patrones manejados por Esper se escriben en un lenguaje llamado Event Processing Language (EPL).

EPL es un lenguaje de código abierto basado en SQL. El mismo permite crear patrones de eventos para especificar eventos complejos que permitan controlar aquellas situaciones que se quiera monitorear.

A continuación, se presenta una regla EPL que tiene en cuenta dos variables: tiempo_registro y ph_agua.

²⁰<http://www.espertech.com/products/esper.php>

```
select x.tiempo_registro, x.ph_agua
from pattern
[every x = EventRegulaAgua (ph_agua>10)]
```

Esta sentencia indica que, para que se active el patrón, es necesario que dentro de la clase EventRegulaAgua, la variable ph_agua exceda el valor de 10.

2.5.1. Metamodelo para CEP

En [Bou15] el autor presenta una propuesta que permite modelar eventos complejos independientemente del motor CEP que se utilice. La propuesta se basa en un metamodelo genérico de eventos que permite representar cualquier tipo de evento, así como definir operadores sobre los mismos. Cuenta con dos modelos principales, uno para representar el dominio y el otro para representar los patrones CEP. Los modelos son extensibles pero no manejan de forma nativa el componente geoespacial del evento.

Asimismo, la propuesta provee una herramienta gráfica que, basándose en el metamodelo, permite modelar eventos complejos en forma simple y sin conocimientos avanzados de programación. A nivel de implementación, la herramienta traduce los eventos modelados en patrones EPL que son enviados a evaluar por un motor Esper.

Metamodelo de Dominio para CEP

En la Figura 2.7 se muestra el metamodelo y sus entidades principales:

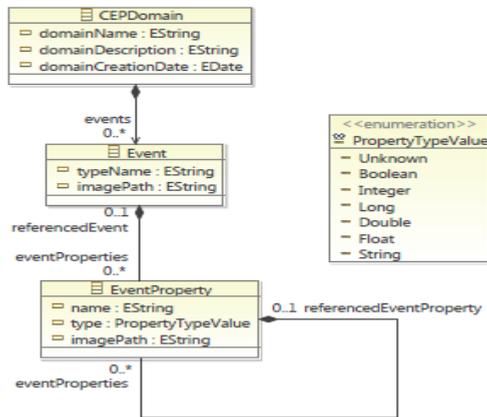


Figura 2.7: Metamodelo de Dominio CEP [Bou15]

- CEPDomain: se trata de la metaclass principal del metamodelo. Cuando la misma es instanciada se tiene un dominio CEP concreto el cual puede estar compuesto por al menos un tipo de evento (Event).
- Event: describe un evento para un dominio CEP en particular. Cada evento pertenece a un tipo concreto (typeName).
- EventProperty: representa la propiedad o característica de un evento. Cada propiedad debe tener un nombre (name) y uno de los siguientes tipos (type): Unknown, Boolean, Integer, Long, Double, Float o String.

Metamodelo de Patrones para CEP

Una vez definido el dominio concreto CEP es necesario elaborar las reglas o patrones de detección de los eventos. Para representar los mismos la herramienta utiliza un metamodelo que puede verse de forma simplificada en la Figura 2.8.

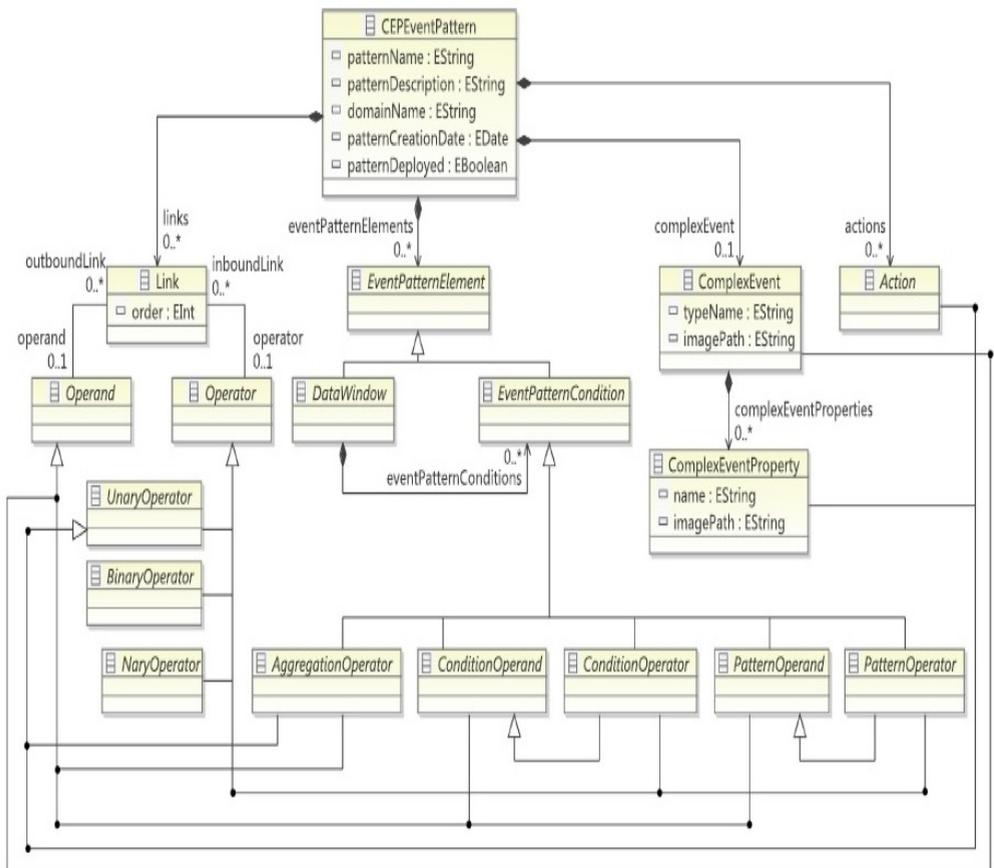


Figura 2.8: Metamodelo de Patrones CEP [Bou15]

Entre las entidades principales del metamodelo se destacan:

- ComplexEvent:** describe el tipo de evento complejo a crear cada vez que se detecte el patrón. El evento complejo tiene un tipo concreto (`typeName`). Cada evento complejo estará formado por una o más propiedades de evento complejo (`ComplexEventProperty`). El nombre del tipo (`typeName`) será igual al nombre del patrón de eventos (`patternName` en `CEPEventPattern`) y con esto el nombre del evento complejo determinará cuál es el patrón que lo ha detectado.
- CEPEventPattern:** es la metaclass principal del metamodelo. Un patrón de eventos es una instancia de la misma. El patrón puede estar formado por: enlaces (`Link`) que vinculen el resto de los componentes, elementos (`EventPatternElement`) para definir las

condiciones de detección del patrón, un evento complejo (ComplexEvent) que representa el tipo de evento generado al detectarse el patrón, y acciones (Action) las cuales son disparadas cuando se activa el patrón.

- Link: define la relación y aridad entre operandos (Operand) y operadores (Operator)
- Operand: describe uno de los datos necesarios para llevar a cabo la operación con la que esté enlazado. Se definen dos tipos de operandos:
 - ConditionOperand: que pueden ser vinculados a los operadores de condición
 - PatternOperand: que pueden ser vinculados con los operadores de patrón.
- Operator: representa una operación específica que puede llevarse a cabo con uno o más operandos dependiendo de la aridad del operador. Existen 3 tipos de operadores:
 - ConditionOperator: que pueden ser enlazados con los operandos de condición
 - PatternOperator: que pueden ser enlazados con los operandos de patrón
 - AggregationOperator: refiere a las funciones de agregación que pueden aplicarse sobre ciertos operandos.

2.6. Trabajo Relacionado

En esta sección se exploran trabajos y desarrollos que se centran en el cumplimiento de leyes, las tecnologías que respaldan su implementación y las capacidades geoespaciales aplicadas en el procesamiento de eventos. Se discuten desde enfoques sobre cómo las organizaciones aseguran el cumplimiento normativo, hasta la integración de componentes geoespaciales en sistemas de procesamiento de eventos.

2.6.1. Cumplimiento de Leyes

En la literatura existen varios artículos que abordan el tema del cumplimiento de la conformidad regulatoria (regulatory compliance) en el contexto de los procesos de negocio, ya sean intra-organizacionales [Dan09] o inter-organizacionales [Sem14][Knu13]. Estos trabajos se centran en el diseño y la operación de sistemas y procesos que cumplen con las regulaciones y normativas vigentes.

El proyecto INSPIRE (Infrastructure for Spatial Information in Europe) [Eur] es una iniciativa activa que aborda temas relacionados con Infraestructuras de Datos Espaciales. Entre sus subproyectos, algunos se dedican a estudiar el monitoreo de variables medioambientales, generando Lenguajes Específicos de Dominio (DSL) para representarlos. Estos DSLs se centran en factores medioambientales como agua, aire y suelo, y permiten relacionarlos con leyes que los afectan. Aunque INSPIRE se enfoca en la definición de DSLs, no propone mecanismos para monitorear las variables medioambientales en tiempo real. Sin embargo, los DSLs desarrollados podrían ser integrados en sistemas más amplios que incluyan monitoreo en tiempo real.

Existen diversos trabajos en el ámbito de los sistemas de gestión de procesos de negocio (BPMS) que exploran cómo estos sistemas pueden integrar consideraciones de cumplimiento normativo. Se han desarrollado varias metodologías que se enfocan en el modelado de los requerimientos de conformidad regulatoria durante la etapa de diseño de los procesos de negocio. No obstante, estos trabajos se centran principalmente en el cumplimiento normativo durante la fase de diseño, es decir, en el momento de la creación de un proceso de negocio. Un ejemplo notable es el trabajo de Cabanillas et al. [Cab20], donde se propone un marco basado en mashups para la verificación de cumplimiento de procesos empresariales. Este marco se centra en la variabilidad de las reglas y su interpretación, ofreciendo un enfoque innovador para la verificación de cumplimiento en diferentes fases del ciclo de vida de la gestión de procesos de negocio.

Respecto al cumplimiento de las leyes ambientales, se han realizado estudios sobre el papel que desempeñan las tecnologías de la información en la auditoría y el cumplimiento de las regulaciones ambientales. Si bien estos trabajos son relevantes para el campo, a menudo no abordan el monitoreo basado en eventos con información geoespacial [LC20].

Contrastando con nuestro escenario, estos proyectos se enfocan en requerimientos de conformidad asociados a procesos de negocio en el contexto de una única organización o en la interacción entre varias organizaciones. No obstante, estas iniciativas no consideran normativas que pueden ser monitoreadas basándose en datos con información geográfica obtenidos de sensores o de Volunteered Geographic Information (VGI). Nuestro enfoque se diferencia de estos trabajos en que se concentra en el control de normativas en tiempo real durante la operación de los procesos de negocio, utilizando de manera extensiva datos geoespaciales para dicho fin.

2.6.2. Enfoques tecnológicos para asegurar el cumplimiento

Las propuestas tecnológicas orientadas a garantizar el cumplimiento de regulaciones y normativas son variadas y evolucionan con rapidez. Se abordan desde soluciones que integran el Internet de las Cosas (IoT) para el monitoreo ambiental hasta enfoques basados en el análisis de imágenes.

En el ámbito de las ciudades inteligentes, se ha investigado cómo la integración de componentes geoespaciales en el Procesamiento de Eventos Complejos puede ser utilizada para monitorear y responder a eventos en tiempo real basados en la ubicación. Una arquitectura propuesta combina CEP con información geoespacial para detectar patrones específicos de eventos en flujos de datos en tiempo real. Esta combinación permite una respuesta rápida a situaciones que requieren intervención inmediata, como congestiones de tráfico o emergencias. Se destaca el uso de sistemas GIS para el procesamiento de datos geoespaciales en el contexto de las ciudades inteligentes [Kha21].

Wells et al. [Wel19] presentan un enfoque para monitorear el cumplimiento de los flujos ambientales utilizando tecnologías de Internet de las Cosas (IoT). Su trabajo se centra en la adaptación del ciclo de Evaluación e Interpretación (EI) para la estimación del cumplimiento de eflows²¹. Utilizando datos del país de Estonia, desarrollaron un sistema denominado EFCES (Environmental Flows Compliance Evaluation System) que transforma un sistema tradicionalmente desconectado en un sistema inteligente de IoT. Este sistema permite alertas en tiempo real, mejora la generación de informes y proporciona modelado de escenarios, predicción y análisis histórico. Aunque el sistema fue diseñado específicamente para monitorear el cumplimiento de los flujos ambientales en Estonia, su arquitectura modular sugiere que podría adaptarse para otros contextos. Sin embargo, es importante destacar que EFCES es una solución *ad hoc*, diseñada específicamente para el contexto de monitoreo de flujos ambientales en ríos estonios, y su aplicabilidad en otros contextos podría requerir adaptaciones significativas.

Existen otros trabajos con foco en el análisis de imágenes para permitir la detección temprana de violaciones ambientales para la prevención y mitigación de daños ecológicos, como el de

²¹Los «eflows» o environmental flows se refieren a la cantidad, calidad y temporalidad del agua que fluye a través de ríos, humedales y zonas costeras, necesaria para mantener ecosistemas acuáticos saludables y proporcionar servicios a las personas.

Chugg et al. [Chu20] que introduce una nueva configuración para el uso de la visión por computadora en la sostenibilidad ambiental. El estudio proporciona un nuevo conjunto de datos de imágenes de satélites para detectar violaciones ambientales y desarrolla un sistema para realizar inferencias en tiempo casi real. Esta investigación subraya la importancia de las tecnologías de imágenes satelitales en la identificación y respuesta rápida a violaciones ambientales.

Estos enfoques son interesantes pero no dejan de ser soluciones ad-hoc para problemáticas concretas que no cumplen con el objetivo de poder ser personalizados a distintos contextos.

2.6.3. Reglas geoespaciales en OpenRules

Karman [Kar23] aborda la necesidad de incorporar datos geoespaciales en la toma de decisiones empresariales. En muchos dominios, desde la planificación urbana hasta la conservación ambiental, las decisiones no solo se basan en criterios numéricos o categóricos, sino también en relaciones espaciales. La solución propuesta por Karman se centra en la integración de estas reglas espaciales en sistemas de reglas empresariales.

La base de esta integración es OpenRules, una herramienta que permite a los usuarios definir y gestionar reglas empresariales en hojas de cálculo. Al combinar OpenRules con el JTS Topology Suite, se introduce una capacidad geoespacial robusta. JTS es conocido por su habilidad para manejar y evaluar relaciones espaciales, y su integración con OpenRules permite que estas capacidades se utilicen directamente en las reglas empresariales.

Uno de los puntos destacados de esta solución es su capacidad para soportar las Relaciones Egenhofer ²², que son fundamentales para determinar relaciones espaciales entre objetos como puntos, líneas y polígonos. Además, la solución puede realizar cálculos esenciales como la distancia entre dos puntos o el área de un polígono. Estas capacidades son necesarias para muchas aplicaciones, desde determinar la proximidad de una actividad industrial a una zona protegida hasta calcular el área afectada por un derrame químico.

La solución también facilita la integración de datos geoespaciales, permitiendo la carga de datos desde archivos GML, un formato ampliamente utilizado para representar información geoespacial. Esta facilidad de integración es esencial para garantizar que los sistemas de reglas empresariales puedan acceder y utilizar datos geoespaciales actualizados.

Modelización de Decisiones Empresariales con DMN y OpenRules

Feldman [Fel17] aborda la modelización de decisiones empresariales utilizando DMN y OpenRules. En su trabajo, destaca la relevancia de los sistemas de gestión de reglas y decisiones empresariales en la arquitectura empresarial moderna. Feldman introduce que DMN,

²²Conjunto de relaciones topológicas binarias que describen las posibles interacciones espaciales entre dos objetos geométricos en un espacio bidimensional.

un estándar desde 2014, puede ser utilizado junto con OpenRules para modelar decisiones empresariales en diversos contextos. Al igual que Karman, Feldman resalta la importancia de las reglas empresariales en la toma de decisiones, pero se centra en cómo DMN y OpenRules pueden integrarse para crear soluciones de modelado de decisiones empresariales.

Uso en el contexto de leyes medioambientales

La combinación de DMN, OpenRules y JTS ofrece un enfoque para abordar el monitoreo y cumplimiento de leyes medioambientales. DMN permite modelar decisiones, mientras que OpenRules proporciona una interfaz para definir y gestionar reglas empresariales. JTS, por su parte, se encarga de las relaciones geoespaciales.

Teniendo en cuenta lo anterior, las leyes medioambientales podrían ser modeladas utilizando tablas de decisión DMN. OpenRules facilita la codificación de estas reglas, permitiendo que expertos en medioambiente gestionen y adapten las reglas según las necesidades cambiantes.

Además con JTS, es posible evaluar relaciones espaciales, como determinar si actividades específicas ocurren dentro de zonas protegidas. Al integrar JTS con OpenRules, estas verificaciones geoespaciales se incorporan en el proceso de toma de decisiones.

Por otro lado, la integración con tecnologías de IoT permite la recopilación de datos en tiempo real, facilitando el monitoreo constante de actividades con impacto medioambiental. Al detectarse una posible violación de las normativas, se pueden generar alertas para las autoridades.

En resumen, este enfoque cuenta con la capacidad de modelar diferentes escenarios y realizar simulaciones proporcionando una herramienta para anticipar el impacto de actividades o cambios en las leyes medioambientales. Esto facilita la toma de decisiones y permite una planificación para garantizar la protección del medioambiente.

Estas herramientas podrían constituir un camino alternativo para cumplir con el objetivo de esta tesis, con la excepción de que esta solución no brinda una herramienta gráfica de modelado con operadores geoespaciales nativos.

2.6.4. CEP con expresividad geoespacial

Las soluciones de CEP permiten analizar patrones entre eventos que ocurren en un sistema y responder en tiempo real [Luc02]. Aunque CEP tiene un gran potencial, muchas de sus soluciones todavía no ofrecen un soporte completo para eventos con componentes geoespaciales.

En términos de trabajos académicos, el modelo geoespacial para el procesamiento de eventos de Cugola y Margara [CM12] presenta una aproximación al procesamiento de eventos geoespaciales, centrando su trabajo en el análisis de eventos que ocurren en una ubicación geográfica específica. Sin embargo, el modelo está limitado en términos de representación

de relaciones espaciales complejas y detección de patrones de eventos a lo largo del tiempo, principalmente debido a que considera tanto los eventos primitivos como los complejos como independientes.

En una línea similar, Wanner et al. [Wan20] se centran en identificar y localizar secuencias anómalas en fábricas inteligentes del Internet de las Cosas mediante CEP. Aunque su sistema permite el análisis en tiempo real de eventos complejos, presenta limitantes, principalmente porque su configuración requiere un conocimiento profundo del dominio y es en gran medida declarativo y reactivo. A pesar de reconocer la potencia del CEP, sugieren que combinarlo con el aprendizaje automático podría eludir algunas de estas limitaciones tecnológicas.

Más allá de estos enfoques académicos, es importante destacar que la mayoría de las herramientas de CEP disponibles en el mercado actual no ofrecen un modelo genérico de modelado de geoeventos, y su uso está limitado a casos específicos. Algunas herramientas han intentado integrar capacidades geoespaciales, pero la mayoría han sido discontinuadas o no han tenido actualizaciones significativas en los últimos años. La Tabla 2.1 muestra una breve comparación de algunos productos analizados.

Tabla 2.1: Herramientas CEP

Herramienta	Características
RULE CORE ²³	La herramienta está desarrollada en Python y permite el disparo de acciones detectando patrones de eventos complejos. Si bien es referenciado en la bibliografía, el proyecto no tiene mantenimiento. Último acceso: 2022.
SPATIAL RULES ²⁴	Permite definir reglas según áreas y es compatible con el estándar geoespacial de OGC. Fue desarrollado por la empresa ObjectFX y no tiene mantenimiento. Último acceso: 2022.
GCEP ²⁵	Es un proyecto que buscó realizar una extensión geoespacial de Esper utilizando Java pero que no fue finalizado. Última actualización del proyecto: 2010. Último acceso: 2023
WSO2 ²⁶	WSO2 Complex Event Processor (CEP) es una herramienta de procesamiento de eventos complejos que permite a las organizaciones identificar patrones en la información que fluye a través de sus sistemas y responder en tiempo real. En el contexto geoespacial, WSO2 CEP puede manejar y analizar datos con información de ubicación, lo que lo hace adecuado para aplicaciones como la gestión de flotas, donde se requiere monitorear y responder a eventos basados en la ubicación de vehículos en tiempo real. WSO2 CEP puede integrarse con otras herramientas y plataformas para proporcionar soluciones completas de gestión de flotas y otras aplicaciones geoespaciales. Es un proyecto activo y con una cantidad considerable de documentación. Último acceso: 2022.

3

Análisis del Problema

En este capítulo se analiza la problemática del cumplimiento de leyes medioambientales. Se realiza primero una descripción general de lo que motiva el trabajo. A continuación, se especifica la reglamentación sobre las leyes medioambientales del Uruguay que regulan la calidad del agua, para luego plantear el escenario de aplicación que se utiliza. A partir del escenario definido, se analizan las ventajas y desventajas de utilizar soluciones ad-hoc y finalmente se plantean las características de una solución para dar respuesta a la problemática planteada.

3.1. Descripción General

El medioambiente ha sido tema de preocupación en la sociedad a lo largo de la historia. En las últimas décadas se ha incrementado la inquietud por parte de los individuos, organizaciones y países sobre como gestionar los problemas medioambientales tanto a nivel local como global surgiendo múltiples normativas con foco en esta problemática.

En el ámbito del derecho, las leyes, normas, decretos y otras regulaciones son textos que sirven de instrumentos legales que permiten regular las acciones de los ciudadanos y las empresas dentro de un territorio determinado. Este texto especifica lo que puede o no realizarse pensando en el bien de la sociedad y penalizando según corresponda a quien infrinja lo dispuesto. En este sentido, el medioambiente es sujeto de interés de la sociedad y por lo tanto susceptible a ser regulado, tanto de forma regional, a nivel de país o inclusive de forma internacional [CEP; Urub; Urua].

En cuanto a la legislación nacional, el artículo 47 de la Constitución de la República [Uruc] establece que la protección del medioambiente es de interés general e indica que las personas deberán abstenerse de cualquier acto que cause depredación, destrucción o contaminación grave al mismo.

Si bien las normativas establecen los criterios para determinar si se está atentando contra el medioambiente, estas carecen de referencias en lo que respecta a los mecanismos para monitorear su cumplimiento. En el presente trabajo, se plantea el caso de la normativa uruguaya, el cual se enfoca en un factor crítico con impacto directo en la calidad de vida de las personas: el agua.

3.2. Regulación del Agua en Uruguay

En Uruguay el Código de Aguas (DL. Nro. 14.859) es la base del régimen jurídico sobre el recurso del agua que establece las competencias de los distintos organismos y mecanismos diversos para su gestión. A su vez la Ley Nro. 14.440 reglamentada por el Decreto 216/76, establece normas sobre el vertido de aguas residuales. Por otro lado, la Política Nacional de Aguas, definida en la Ley Nro. 18.610 y en su D.R. 78/10, comprende la gestión de los recursos hídricos así como los servicios y usos vinculados al agua. El control de la contaminación de las aguas queda regulado mediante el Decreto 253/79¹ y demás normas que lo modifican. Este decreto establece los parámetros a evaluar de las aguas en Uruguay así como sus valores de control. Estos límites dependen del uso para el que está destinado el agua y establece una categorización en clases distintas.

- Clase 1: Aguas destinadas o que puedan ser destinadas al abastecimiento de agua potable.
- Clase 2: (a) Aguas destinadas al riego de plantas con el sistema de mojado y destinados al consumo humano en su forma natural; (b) Aguas destinadas a recreación por contacto directo con el cuerpo humano.
- Clase 3: Aguas destinadas a la preservación de integrantes de la flora y fauna hídrica, o también aguas destinadas al riego de cultivos cuyo producto no se consume en forma natural o en aquellos que apliquen sistemas de riego que no provocan el mojado del producto.
- Clase 4: Aguas corrientes en general que deben mantener armonía con el ambiente y aguas destinadas al riego de cultivos cuyos productos no son consumidos por humanos.

La Tabla 3.1 presenta los valores² aceptables que fija la reglamentación existente según la clase.

¹<http://www.impo.com.uy/bases/decretos/253-1979>

²Se toma un subconjunto de los valores establecidos en el Decreto 253/79

Tabla 3.1: Estándar de Calidad del Agua

Parámetro	Clase 1	Clase 2a	Clase 2b	Clase 3	Clase 4
PH (min-max)	6,5 - 8,5	6,5 - 9,0	6,5 - 8,5	6,5 - 9,0	6,0 - 9,0
Cromo Total (max)(mg/l)	0,05	0,005	0,05	0,05	0,5
Cianuro (max)(mg/l)	0,005	0,005	0,005	0,005	0,05
Plomo (max)(mg/l)	0,03	0,03	0,03	0,03	0,05
Cadmio (max)(mg/l)	0,001	0,001	0,005	0,001	0,01
Mercurio (max)(mg/l)	0,0002	0,0002	0,0002	0,0002	0,0002

3.3. Escenario: Monitoreo de la Calidad del Agua

El escenario de aplicación está basado en la regulación de la calidad del agua en Uruguay y plantea la situación en la que el municipio CFA de Sauce está interesado en monitorear la calidad del agua del río Santa Lucía en el tramo que corresponde a su jurisdicción. El objetivo del organismo es poder determinar si el agua es apta para consumo de su población para lo cual debe cumplir con los estándares establecidos en el decreto perteneciente al Código de Aguas del Uruguay. El agua destinada para este fin según la regulación del Uruguay debe cumplir con lo establecido en la Clase 1 de la Tabla 3.1.

Para evaluar estos parámetros el municipio planea utilizar datos abiertos recolectados a través de sensores ubicados a lo largo del río que podrían proporcionar distintos organismos, por ejemplo, la DINAMA³. Estos datos incluirían la información de las mediciones de los distintos parámetros y a su vez la ubicación del sensor donde fue captada. También se podrían utilizar datos de mediciones proporcionados por particulares (compartidas a través de redes sociales) y validadas, por ejemplo, por el MTOP⁴ que podría ofrecerlas además a través de servicios web.

En la Figura 3.1 se muestra la representación de una posible distribución de los sensores a lo largo del río Santa Lucía. Los sensores representados por iconos triangulares son los que se encuentran en la zona a estudiar la cual se resalta con un círculo, los sensores representados con cuadrados son los que siguen en proximidad y los representados con pequeños círculos los más alejados.

El municipio CFA necesita monitorear estos valores y en caso que los mismos salgan de los

³DINAMA: Dirección Nacional de Medio Ambiente del Uruguay

⁴MTOP: Ministerio de Transporte y Obras Públicas del Uruguay

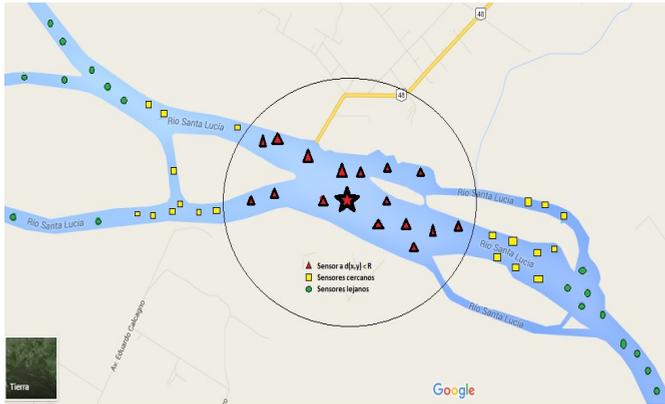


Figura 3.1: Posible ubicación de sensores en el río Santa Lucía

rangos permitidos es necesario generar alertas para tomar las acciones correspondientes. En una primera etapa se realizará el control sobre los valores del pH y el Cromo Total en el agua, para luego extender el mismo a otros parámetros de interés.

En este escenario es vital el poder detectar las desviaciones y posibles violaciones de las regulaciones medioambientales en tiempo real analizando las distintas fuentes de datos disponibles. La cantidad de información a procesar puede ser de gran volumen, heterogénea y proveniente de múltiples fuentes.

Una forma de abordar la solución en este escenario es mediante la utilización de eventos. En este sentido, los datos resultantes de las mediciones de los sensores o reportes pueden ser modelados como eventos. Un ejemplo de evento complejo podría ser el que se muestra en la Figura 3.2, que indica que se genera una sospecha de contaminación (evento complejo) cuando se da una combinación de otros eventos:

```
IF (PH < 6 OR Cromo > 0.05 ) AND (Vertido de aguas)
THEN Sospecha de Contaminacion
```

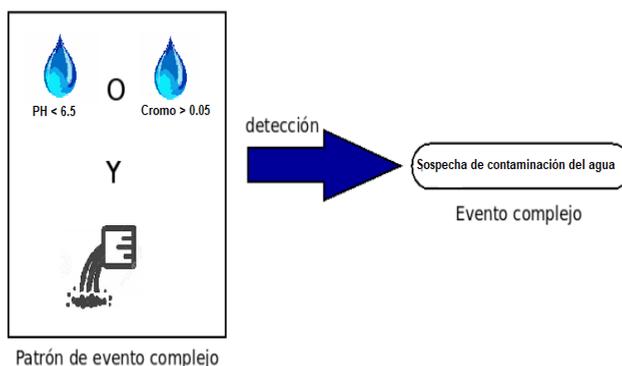


Figura 3.2: Evento Complejo

Las tecnologías para el procesamiento de eventos complejos (CEP) permiten procesar y analizar grandes cantidades de eventos, así como correlacionarlos para detectar y responder en tiempo real a situaciones críticas del contexto. Para ello, las condiciones que describen las situaciones que se pretenden detectar se especifican usando unas plantillas especiales, denominadas patrones de eventos.

Por lo antes mencionado, CEP es adecuado para procesar la información medioambiental disponible y utilizarla para validar los requisitos impuestos por la normativa respecto a los parámetros tolerados del factor a medir. Sin embargo, las regulaciones aplican en un marco geoespacial determinado y para esto las tecnologías CEP estudiadas en el Capítulo 2 no cuentan con soporte nativo. Una alternativa puede ser realizar una solución ad-hoc como la que se describe en la sección siguiente.

3.4. SIMONA: Solución ad-hoc basada en CEP

El proyecto SIMONA [Gon17] implementó una solución particular al problema planteado por el caso motivador de la Sección 3.3, con una interfaz web orientada a usuarios finales.

SIMONA utilizó patrones estáticos para la detección de eventos complejos escritos en el lenguaje EPL y basados en el trabajo [Her16]. Los patrones utilizados involucran el operador distancia y un conjunto de parámetros fijos cuyos valores se pueden ingresar interactuando con la interfaz. A su vez es posible seleccionar la zona (utilizando la geometría multipolígono) de los datos que son tenidos en cuenta para la detección de los eventos.

La herramienta implementada se muestra en la Figura 3.3 y permite gestionar el alta de sensores, fuentes de datos, eventos complejos a detectar y alarmas a generar a partir de la detección de los eventos. En la figura se muestra la pantalla de definición de un evento complejo que

tiene una alerta por distancia. En el mapa que se muestra a la derecha en la figura es posible delimitar la zona que es representada en este caso con un círculo para el cual se define su centro indicando latitud y longitud, y su radio en Kilómetros. A su vez, el evento complejo se especifica seleccionando el elemento y propiedad a evaluar entre los posibles (datos fijos) y estableciendo el umbral en el que deben estar estos valores en concordancia con el criterio que el usuario final quiera evaluar. En este caso el Cromo Total en al Agua debe ser menor a 0.05 y el pH debe estar entre 6.5 y 8.5.

Alta alerta por distancia

Nombre: 253/79 - Santa Lucía
 Descripción: Alerta para monitorear el río Santa Lucía
 Elemento: Todos

Elemento	Propiedad
<input checked="" type="checkbox"/>	Agua - Cromo Total
<input type="checkbox"/>	Agua - Densidad
<input type="checkbox"/>	Agua - Fósforo
<input type="checkbox"/>	Agua - NO2
<input type="checkbox"/>	Agua - O2
<input checked="" type="checkbox"/>	Agua - Ph
<input type="checkbox"/>	Agua - Temperatura

Agregar

Elemento	Propiedad	Rango
<input type="checkbox"/>	Agua - Cromo Total	Menor a 0.05
<input type="checkbox"/>	Agua - Ph	Rango inclusivo Desde 6.5 Hasta 8.5

Quitar

Tipo de Geometría: Círculo

Latitud: -34.57725635481406

Longitud: -56.49040207621382

Rango distancia (KM): 1.1849569710901493

Alta alerta

Figura 3.3: Prototipo de herramienta para manejo de sensores y alertas [Gon17]

También es posible definir un evento complejo con alerta por zona, como se muestra en la Figura 3.4. En ella, se puede delinear un polígono que demarca la zona geográfica sobre el mapa a la derecha, establecer los umbrales de los parámetros a monitorear e incluir información sobre la normativa aplicable, como en el caso del ejemplo, el Decreto 253/079.

Alta alerta por zona

Nombre:

Descripción:

Permiso:

Tipo:

Ley:

Elemento:

Elemento	Propiedad	Rango
<input type="checkbox"/> Agua	Cromo	<input type="text" value="Mayor"/> Mayor a <input type="text" value="0.01"/> <input type="button" value="Nuevo rango"/>
<input type="checkbox"/> Agua	PH	<input type="text" value="Mayor"/> Mayor a <input type="text" value="8.5"/> <input type="text" value="Menor"/> Menor a <input type="text" value="6.5"/> <input type="button" value="Nuevo rango"/>



Figura 3.4: Alerta por Zona [Gon17]

Además, cuando se detecta una situación de alerta, esta se muestra instantáneamente en la pantalla de inicio de la aplicación, como se puede ver en la Figura 3.5.

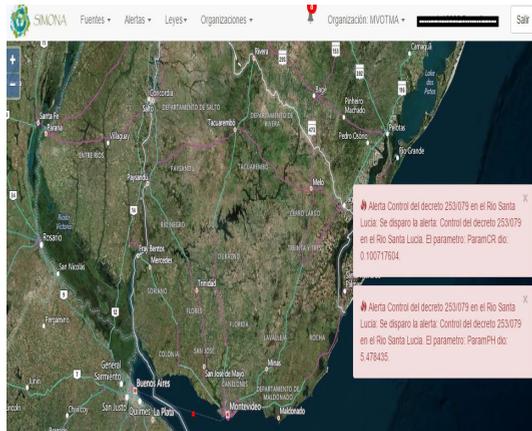


Figura 3.5: Notificación de Incumplimiento [Gon17]

Si bien SIMONA implementa una aplicación a medida dirigida a resolver parte de la problemática presentada en el caso, la misma tiene algunas limitantes subyacentes a su naturaleza descriptas a continuación:

- **Dependencia tecnológica:** Soluciona el problema específico utilizando un conjunto de tecnologías y lenguajes de programación (Java, EPL, ESB, entre otros). Se dificulta portar la solución conceptual a otras tecnologías (lenguajes CEP, implementaciones de ESBs, etc.).
- **Sin flexibilidad en la generación de patrones CEP:** SIMONA cuenta con un conjunto limitado de patrones CEP a ser utilizados. Si bien permite al usuario establecer los valores y algunas relaciones lógicas entre los parámetros a medir, la generación de patrones está limitada a lo que la interfaz permite ingresar.
- **Limitación en sus capacidades geospaciales:** Solo permite seleccionar datos por zona utilizando un multipolígono pero no permite realizar operaciones con las geometrías definidas, por ejemplo podría ser necesario conocer eventos en la intersección de dos zonas y con esta solución no es posible.
- **Limitaciones en extensibilidad:** Debido a las dos limitantes anteriores, la solución no permite agregar nuevos patrones CEP con facilidad. Esto limita el conjunto de leyes que potencialmente se podrían monitorear.
- **Solución dependiente del contexto:** Dada su especificidad tecnológica, la solución está enfocada solo en el monitoreo de leyes medioambientales y no puede ser portada a otro escenario.

3.5. Resumen

Si bien es posible resolver la problemática que implica monitorear el cumplimiento de leyes medioambientales utilizando soluciones ad-hoc basadas en CEP, las mismas tienen limitantes que impiden su escalabilidad como solución que se adapte al contexto. Además, el control del cumplimiento de regulaciones podría no limitarse a leyes medioambientales y en este sentido las tecnologías de CEP permiten ser usadas en otros escenarios.

Con el cometido de superar las restricciones que las soluciones de este tipo pueden tener es que se busca una solución con las siguientes características:

- **Capacidad de modelado de leyes:** Las leyes medioambientales son especificadas en lenguaje natural por lo que es deseable tener un modelo conceptual de referencia que permita identificar los elementos presentes en una ley con el objetivo de facilitar a los expertos del dominio el modelado de las reglas a controlar.
- **Independencia del lenguaje de CEP:** La solución debe implementarse en un lenguaje de procesado de eventos particular (EPL) para Esper pero debe poder ser extendida a otros.
- **Generación de nuevos patrones:** La herramienta implementada debe generar patrones de eventos complejos.
- **Generación de nuevos patrones con componente geoespacial:** Se debe contar con la capacidad de definir patrones de geoeventos que incluyan operaciones sobre las geometrías asociadas a los mismos.
- **Solución agnóstica:** La solución a nivel de modelos debe aplicar a las leyes medioambientales, pero también permitir ser aplicada en otros escenarios.

4

Extensión Geoespacial CEP para el Control de Leyes Medioambientales

Este capítulo presenta la extensión geoespacial para el modelado de eventos complejos diseñada con el objetivo de monitorear y controlar leyes medioambientales. Esta extensión encuentra sus fundamentos en el metamodelo de eventos complejos presentado en la Sección 2.5.1, y su implementación ha sido alcanzada por medio de una estrategia dirigida por modelos.

El capítulo comienza por la Sección 4.1 que expone una descripción general de la propuesta, mostrando cómo las leyes medioambientales pueden ser representadas a través de eventos complejos. Esta sección presenta el proceso de diseño de la solución, derivada del metamodelo que se introdujo en el Capítulo 2, al que se le incorpora soporte geoespacial e instancia para ser aplicado específicamente en el control de leyes medioambientales.

En la sección 4.2 se propone un modelo conceptual para la representación y control de leyes medioambientales mediante el uso de la extensión geoespacial propuesta.

Por último, en la Sección 4.3 se profundiza en la extensión geoespacial para el modelado de eventos complejos. Se detalla cómo se integran los aspectos geoespaciales en el metamodelo de eventos complejos y se presentan los componentes gráficos que facilitan su representación.

4.1. Descripción General

En el Capítulo 3 se identifica la necesidad de encontrar una solución que permita generar diferentes tipos de patrones geoespaciales, con un modelo independiente del lenguaje CEP utilizado y agnóstico al contexto. Para lograr estas características, se adopta un enfoque dirigido por modelos que permite modelar y abordar un problema con un nivel de abstracción que trasciende las implementaciones particulares. En este sentido, se utilizó como base el trabajo presentado en la Sección 2.5.1, el cual se extendió a nivel de modelos, conservando así

las características de independencia tecnológica y contextual que posee la solución inicial.

4.1.1. Modelo Conceptual de Leyes

El Modelo Conceptual de Leyes es un artefacto que permite la representación del contexto específico que se desea abordar mediante la plataforma extendida. El mismo debe ser instanciado por un experto en el dominio, quien posea un profundo conocimiento de las leyes y restricciones que se desean controlar en el cumplimiento normativo.

Este artefacto es una pieza fundamental en el proceso de control de las regulaciones medioambientales ya que proporciona una representación estructurada de las leyes y restricciones que se aplican en el ámbito específico del cumplimiento normativo.

El Modelo tiene como objetivo ser una base para la definición de eventos y patrones en la plataforma extendida. Proporciona una estructura homogénea que permite la identificación y especificación de las condiciones y acciones necesarias para el cumplimiento normativo.

La elaboración de este modelo debe considerar cada aspecto relevante de las regulaciones medioambientales. Además, su diseño debe ser lo suficientemente flexible y adaptable para abordar escenarios cambiantes y nuevas exigencias normativas.

En resumen, el Modelo Conceptual de Leyes constituye un componente esencial de la plataforma extendida, brindando un marco para el análisis y control del cumplimiento de las regulaciones medioambientales. Este modelo se explica con detalle en la Sección 4.2.

4.1.2. Plataforma Original: MEdit4CEP

La Figura 4.1 presenta una representación gráfica a alto nivel de la solución propuesta en [Bou15], la cual consta de tres componentes principales: el **CEP Editor**, el **Transformer M2T** y el **Executor Component**. El objetivo principal de esta solución es abordar la definición y ejecución de patrones de eventos complejos en sistemas distribuidos.

El editor gráfico (CEP Editor) desempeña un papel fundamental, al permitir a los usuarios definir patrones de eventos utilizando modelos de dominio para CEP. Esta capacidad facilita la creación y modificación de patrones por parte del usuario final, ya que se puede trabajar con una interfaz gráfica.

Por otro lado, el Transformer M2T transforma los modelos CEP, definidos en el editor, en código ejecutable por el componente ejecutor. Esta traducción es necesaria para llevar los patrones definidos en el entorno gráfico a nivel de ejecución real.

Finalmente, el componente ejecutor (Executor Component) se encarga de la ejecución del código generado a partir del modelo CEP traducido por el Transformer M2T. Su tarea principal es detectar y reconocer patrones complejos en tiempo real a medida que los eventos ocurren,

lo que brinda la capacidad de procesamiento y análisis en tiempo real que se requiere en aplicaciones CEP.

En conjunto, estos tres componentes trabajan de manera sincronizada para permitir la definición, traducción y ejecución efectiva de patrones de eventos complejos en sistemas distribuidos. Esto ofrece a los usuarios la capacidad de aprovechar el potencial de CEP en entornos en tiempo real y distribuidos.

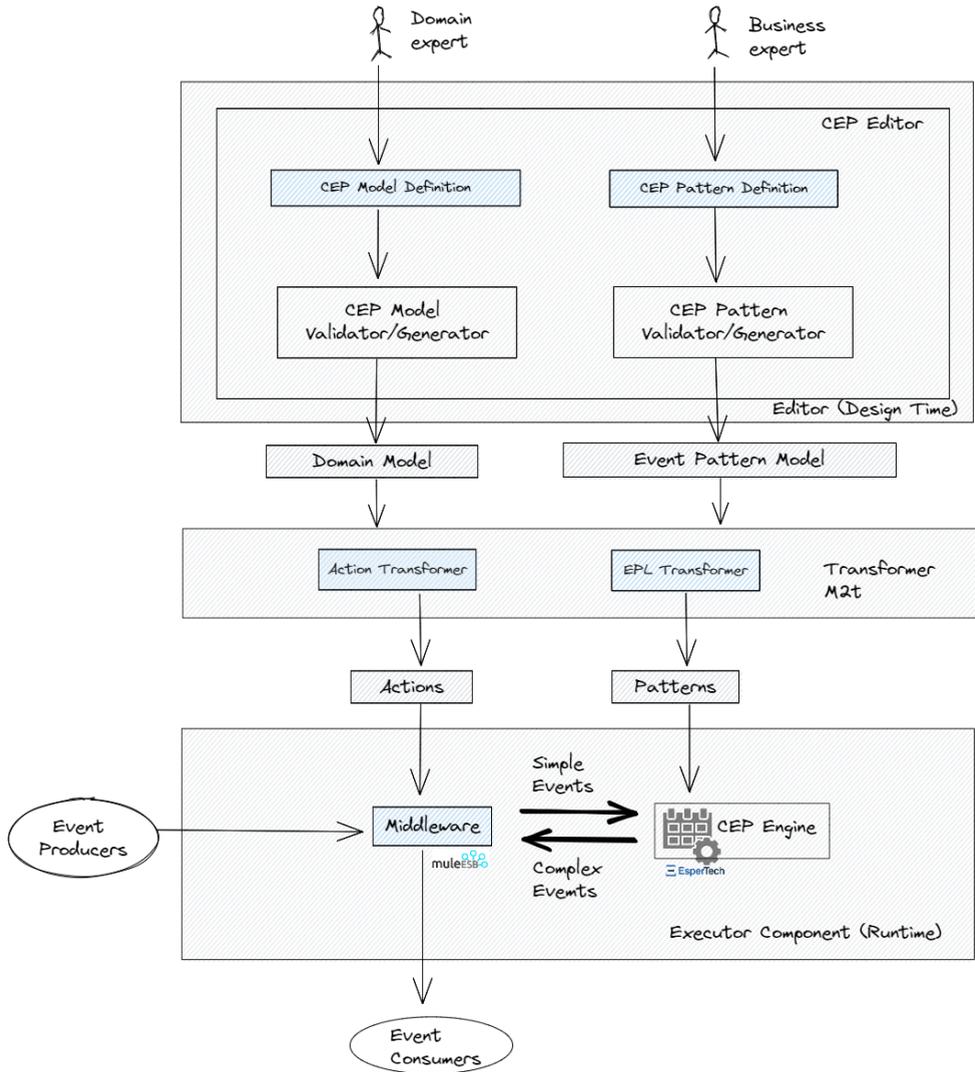


Figura 4.1: Plataforma Original

A continuación se describe con más detalle cada uno de estos componentes y su rol dentro de la plataforma.

CEP Editor

El CEP Editor es un componente construido utilizando la tecnología Eclipse RCP (Rich Client Platform). Esta característica le confiere la capacidad de ser ejecutado fuera de Eclipse y en múltiples plataformas, como Windows, Linux y Mac. Este componente permite a los usuarios definir eventos y patrones utilizando una interfaz gráfica.

El editor gráfico cuenta con dos componentes principales que pueden observarse en la Figura 4.1: el CEP Model Definition y el CEP Pattern Definition. El CEP Model Definition es el componente encargado de definir los modelos de dominio CEP que se utilizan para definir los patrones de eventos. Estos modelos de dominio definen las entidades y relaciones relevantes para el problema que se está abordando, y son utilizados por el CEP Pattern Definition para definir los patrones de eventos. Por otro lado, el CEP Pattern Definition es el componente encargado de definir los patrones de eventos utilizando los modelos de dominio previamente definidos. Este componente propone un lenguaje de modelado para la definición de patrones de eventos, así como un editor gráfico que permite al usuario definir fácilmente estos patrones.

También forman parte del editor las herramientas CEP Model Validator y Generator, que se encarga de validar y generar modelos de dominio CEP, y CEP Pattern Validator y Generator, que valida y genera patrones de eventos, respectivamente.

La interacción de los componentes del editor ocurre en **tiempo de diseño** (Design Time). En esta etapa, el usuario final puede definir modelos de dominio CEP y patrones de eventos utilizando tanto un lenguaje de modelado como un editor gráfico específico para este fin. Estos modelos permiten la detección de situaciones de interés a partir del flujo continuo de eventos.

Una vez validados, estos modelos pueden ser utilizados para generar código que implemente los patrones definidos en sistemas CEP. Esto significa que el usuario final puede diseñar y validar sus modelos de dominio CEP y patrones de eventos utilizando el editor gráfico, sin necesidad de tener conocimientos técnicos profundos, generando una representación agnóstica a la tecnología particular a utilizar. Luego, estos modelos pueden ser transformados en código que en la plataforma propuesta por [Bou15] se despliega en tiempo de ejecución tanto en un motor CEP (CEP Engine) como en un ESB¹ que hace de Middleware.

¹Un ESB (Enterprise Service Bus) es una herramienta de integración de software diseñada para conectar y coordinar la interacción entre diferentes aplicaciones en un entorno empresarial. Actúa como un intermediario que permite a las aplicaciones comunicarse entre sí sin tener que interactuar directamente. Facilita la reutilización de servicios, la orquestación de procesos de negocio, la transformación de datos y la gestión de comunicaciones en sistemas distribuidos. Es esencial para lograr la agilidad y flexibilidad en arquitecturas de TI complejas.

Componente Transformer M2T

Para convertir los modelos definidos en código ejecutable y evaluable durante la ejecución en tiempo real, es necesario realizar transformaciones que se adapten a una tecnología específica, para lo que existen dos componentes:

- **EPL Transformer:** Este componente se encarga de transformar los patrones definidos en el editor gráfico en código EPL (Event Processing Language), que puede ser interpretado por el motor Esper durante la ejecución en tiempo real.
- **Action Transformer (XML):** Este componente se encarga de transformar las acciones asociadas a los patrones definidos en el editor gráfico en código XML, el cual puede ser interpretado por el Middleware utilizado en la plataforma. En el caso del trabajo base, se emplea el Enterprise Service Bus (ESB) MuleESB como Middleware.

Estos componentes son módulos que utilizan EGL, un lenguaje de transformación de modelo a texto proporcionado por Epsilon y basado en plantillas.

Las transformaciones realizadas por estos componentes son críticas para convertir los modelos definidos en el editor gráfico en código ejecutable, capaz de ser evaluado y procesado durante la ejecución en tiempo real. El componente EPL Transformer asegura la adaptación de los patrones definidos al lenguaje de procesamiento de eventos (EPL) requerido por el motor Esper. De manera similar, el Action Transformer se encarga de convertir las acciones asociadas a los patrones en un formato XML compatible con el Middleware utilizado en la plataforma, facilitando la correcta ejecución y procesamiento de dichas acciones.

Estas transformaciones son imprescindibles para garantizar una integración efectiva entre el modelado realizado en el editor gráfico y el código que se ejecuta en tiempo real. Al asegurar que los patrones definidos y las acciones asociadas se adapten a las tecnologías utilizadas en la plataforma, se facilita el procesamiento de los eventos complejos en el entorno distribuido.

Componente Ejecutor

En tiempo de ejecución, la plataforma es dependiente de la tecnología utilizada, ya que requiere de implementaciones específicas tanto del motor CEP como del middleware encargado de la ejecución de acciones. Los componentes principales en este contexto son:

- **Motor CEP (Esper):** Es el motor CEP utilizado para detectar situaciones de interés en tiempo real en la plataforma de este trabajo. Esper interpreta el código EPL generado por el EPL Transformer y realiza una evaluación continua de los eventos entrantes para detectar si se ha producido alguna situación de interés.
- **Middleware (MuleESB):** Es un bus de servicios empresariales empleado como middleware en la plataforma propuesta en [Bou15] para orquestar las acciones necesarias ante la detección de situaciones de interés. MuleESB interpreta el código XML ge-

nerado por el Action Transformer y ejecuta las acciones correspondientes, las cuales pueden incluir desde el envío de correos electrónicos hasta la escritura de datos en una base de datos, entre otras posibles respuestas.

Estos componentes son fundamentales para garantizar el funcionamiento del sistema en tiempo real. La colaboración e interdependencia entre estos componentes asegura que la plataforma detecte a tiempo los eventos relevantes y responda de manera adecuada, lo que permite el monitoreo y la emisión de alertas ante la detección de incumplimientos en tiempo real.

4.1.3. Plataforma Extendida

La extensión de la plataforma MEdit4CEP incorpora la capacidad de interactuar con geo-eventos. La misma se ilustra en la Figura 4.2 y requirió modificaciones tanto a nivel del editor gráfico, la transformación y generación de código EPL y de las capacidades del motor CEP.

GeoCEP Editor

El Editor Gráfico permite al usuario definir los modelos de dominio CEP necesarios para la detección de situaciones de interés en un contexto específico. En el caso de esta tesis, el contexto se relaciona con el cumplimiento de las leyes medioambientales. Además, el Editor Gráfico permite la definición de patrones de eventos utilizando estos modelos. Los modelos resultantes pueden ser validados y generados automáticamente por las herramientas CEP Model Validator y Generator, así como por el CEP Pattern Validator y Generator.

En el marco de este trabajo, se llevó a cabo una extensión del CEP Model Definition para incorporar la representación de eventos geoespaciales. Asimismo, se amplió el CEP Pattern Definition con operadores que permiten la definición de patrones para trabajar con componentes geoespaciales. Esta mejora brinda una mayor flexibilidad y capacidad para la detección de situaciones de interés en entornos que involucran información geoespacial.

Geo EPL Transformer M2T

El EPL Transformer original, implementado utilizando EGL y Java, fue extendido para permitir la transformación de eventos y patrones geoespaciales. Como resultado de esta transformación, se obtiene un nuevo lenguaje que extiende el EPL de Esper, añadiendo expresividad geoespacial al mismo (al que nos referimos como geoEPL).

Una característica destacada de esta solución es su capacidad de ser extensible, ya que fue diseñada teniendo en cuenta la posibilidad de agregar nuevos operadores en el futuro.

GeoEsper

El Geo EPL Transformer permite generar EPLs con capacidad de expresión geoespacial (geo-EPL). Estas expresiones no pueden ser interpretadas por el motor de Esper nativo, por lo que fue necesario crear el componente GeoEsper para permitir al motor de procesamiento de eventos Esper procesar los geoeventos. Esto puede verse representado dentro del Ejecutor Componente de la Figura 4.2.

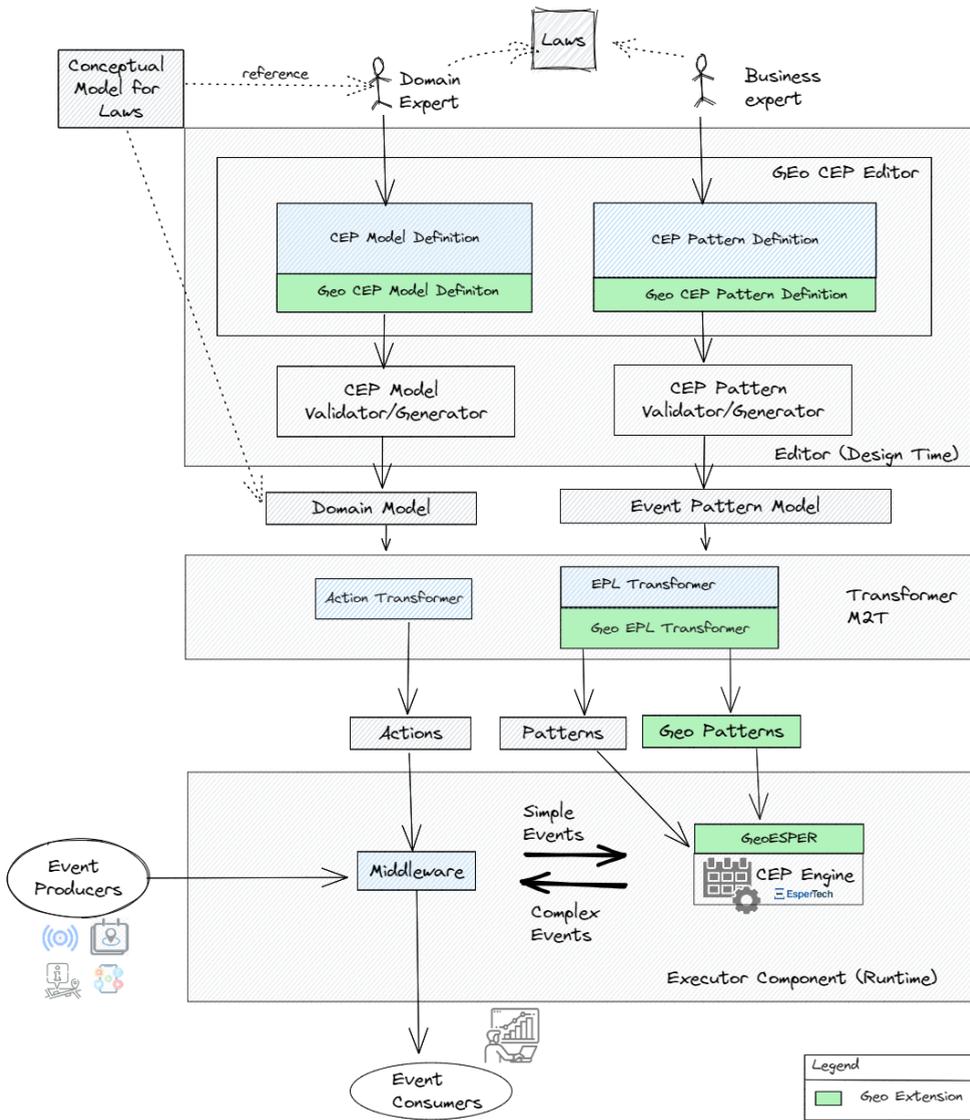


Figura 4.2: Plataforma Extendida

En resumen la extensión de la plataforma MEdit4CEP [Bou15] propuesta permite la incorporación y el procesamiento de eventos geoespaciales, ampliando las capacidades de detección de situaciones de interés en entornos que involucran información geoespacial.

4.2. Modelado de Regulaciones Ambientales

Para abordar la problemática discutida en el Capítulo 3, se ha elaborado un Modelo Conceptual. Este modelo emplea el sistema de representación de eventos detallado en la Sección 2.5.1 y considera las especificidades inherentes a las leyes.

Las leyes y regulaciones se organizan en «Secciones». Por ejemplo, el Decreto 253/97 consta de 33 artículos. Cada sección puede imponer restricciones sobre elementos o «Factores», que en el caso de las leyes medioambientales, son específicos como el *Agua* en el Decreto 253/97. A su vez, las secciones de las leyes establecen «Restricciones» que regulan estos factores. Por ejemplo, el artículo 5 sección b del Decreto 253/97 establece límites de concentración para el arsénico y el boro en el agua destinada a riego. Así, las leyes se pueden representar conceptualmente como un conjunto de «Documentos», compuestos por «Secciones», las cuales pueden incluir «Restricciones» a «Factores» regulados. Los tipos de Factores en las leyes medioambientales son los elementos naturales como aire, agua, suelo, etc. Sin embargo, en otros contextos, los Factores podrían ser distintos, como salario, edad u otro aspecto sujeto a la norma.

Además, es esencial representar el mecanismo de control de las «Restricciones». Si los datos sobre el estado de los elementos ambientales se reciben como eventos, el cumplimiento de las restricciones impuestas por las regulaciones puede monitorearse especificando patrones de eventos para dichos sucesos. En este contexto, el modelo de representación de eventos complejos de la Sección 2.5.1 resulta útil. Por ejemplo, se puede definir un patrón para detectar si el nivel de cromo en el agua ha estado aumentando continuamente durante la última hora y si está cerca de superar el umbral de 0,05 mg/l. Así, esta información puede ser utilizada para tomar medidas antes de que se alcance el límite establecido por la regulación.

La Figura 4.3 formaliza el análisis realizado a través de la especificación de un metamodelo para regulaciones medioambientales. Este modelo permite especificar las leyes analizadas, así como los factores y restricciones que determinan.

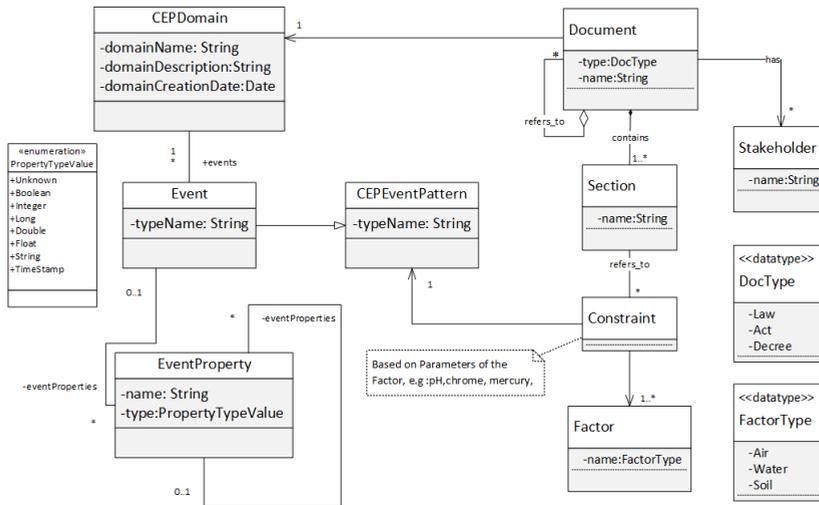


Figura 4.3: Modelo Conceptual para el Monitoreo de Regulaciones Ambientales [Her17]

El metamodelo define los siguientes conceptos para representar regulaciones medioambientales:

- **Document:** Representa el documento principal a considerar, que puede ser una *Ley*, un *Decreto* o cualquier otro tipo de regulación. Un documento está relacionado con un dominio concreto de CEP, que definirá eventos y patrones para el seguimiento de la normativa.
- **DocType:** Sirve para identificar los diferentes tipos de documentos legales, por ejemplo: decreto, ley, entre otros.
- **Section:** Forma parte del documento principal. Por ejemplo, un *Artículo* puede ser una sección si el documento principal es una Ley.
- **Factor:** Representa el elemento que la ley regula. En el ámbito medioambiental, algunos ejemplos de estos elementos son: aire, agua y residuos.
- **FactorType:** Se utiliza para identificar los diferentes tipos de factores. En el caso del Decreto 253/97 el factor es el agua.
- **Constraint:** Representa los requisitos establecidos en una Sección de un Documento Principal para un Factor. Las restricciones permiten fijar rangos o valores límite para los parámetros considerados en la ley.

El metamodelo propuesto ofrece una manera estandarizada y sistemática de representar una ley, lo que facilita la generación de artefactos de software que pueden ser utilizados para supervisar la conformidad con las restricciones especificadas a través del Procesamiento de

lógicos y aritméticos. Este trabajo amplía dicho modelo al incluir una dimensión geoespacial. Así, se propone un metamodelo donde la Geometría, definida de acuerdo al modelo de referencia de OGC SFS, se convierte en una propiedad inherente del evento, tal como se ilustra en la Figura 4.5. Esta figura extiende el metamodelo que se muestra en la Figura 2.7.

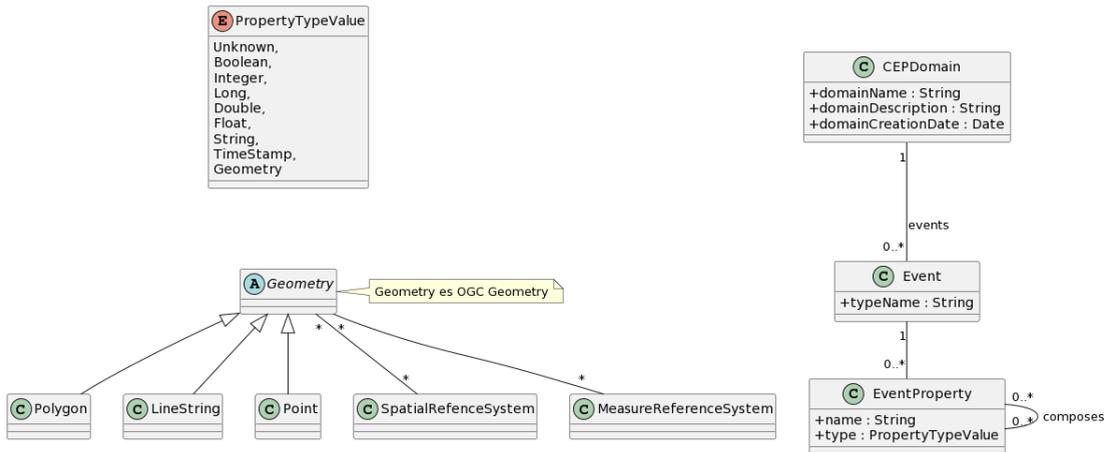


Figura 4.5: Modelo de Dominio de GeoEventos

Operadores

El modelo original, representado en la Figura 2.8, define diversos operadores lógicos (And, Or, Not), aritméticos (Addition, Subtraction) y de comparación (Equals, LessThan, GreaterThan). Sin embargo, para poder manejar geometrías, se hizo necesario expandir los operadores existentes para soportar operaciones geoespaciales.

En términos de geometrías, se pueden realizar diversas operaciones que se clasifican en cuatro categorías principales:

- **Teoría de conjuntos:** Incluye operaciones como la unión, la intersección y la diferencia. Por ejemplo, la intersección de dos polígonos puede resultar en otro polígono, una línea, un punto o puede no haber intersección.
- **Operaciones topológicas:** Incluyen relaciones como toca y superpone. Un caso ilustrativo es cuando la frontera de Uruguay toca la frontera de Brasil, asumiendo que ambos países se representan como polígonos.
- **Operaciones métricas:** Incluyen cálculos como el área y la distancia. Un ejemplo sería calcular el área de Montevideo, que es de 200,7 km².
- **Operaciones direccionales:** Incluyen direcciones como norte, sur y sureste. Por ejem-

plo, se puede afirmar que la ciudad de La Paz está al norte de Montevideo.

La extensión propuesta adopta algunos de estos operadores y los incorpora al modelo, tal como se muestra en la Figura 4.6. Se crean nuevas categorías de operadores, basándose en si el resultado es una geometría (operador aritmético) o un predicado que retorna verdadero o falso (operador booleano):

- **GeoArithmetic:** Estos operadores toman n-operandos del tipo geometría y retornan otra geometría. Incluyen operaciones como Union, Intersection y Difference.
- **GeoBoolean:** Estos operadores toman n-operandos y retornan un valor booleano. Incluyen operadores clásicos como Equals, Distance, Intersects y Contains.

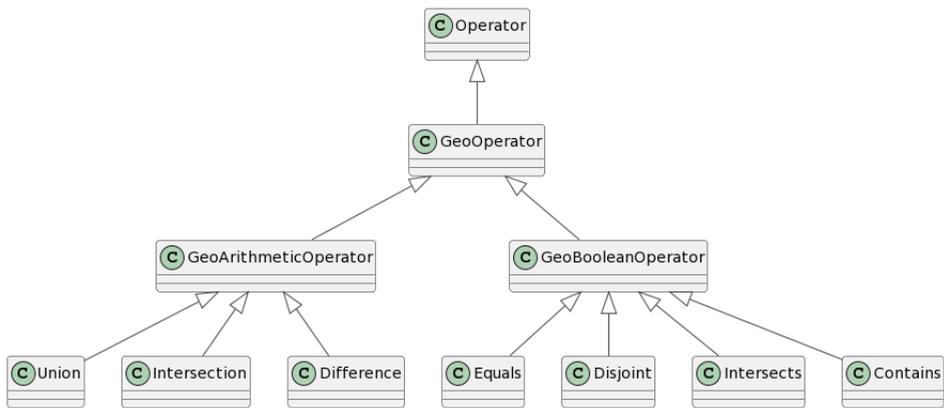


Figura 4.6: Operadores de GeoEventos

El detalle de estos operadores, el cual se basa en las especificaciones de OGC vistas en la Sección 2.3, se expone en la Tabla 4.1.

Tabla 4.1: Especificación de Geo Operadores

Operador	Descripción	Tipo Resultado
Union	Se define $a \cup b = c$, donde a , b , y c son geometrías. c es el resultado de la unión de todos los puntos de a y b .	Aritmético
Intersection	Se define $a \cap b = c$, donde a , b , y c son geometrías. c es la geometría de los puntos en común de a y b .	Aritmético
Difference	Se define $a - b = c$, donde a , b , y c son geometrías. c es la geometría resultante de la diferencia entre a y b .	Aritmético
Equals	Sean a y b geometrías, $a.Equals(b)$ es verdadero si se contienen mutuamente; de lo contrario, es falso.	Booleano
Disjoint	Sean a y b geometrías, $a.Disjoint(b)$ es verdadero si su intersección es vacía, es decir, no tienen ningún punto en común; de lo contrario, es falso.	Booleano
Intersects	Sean a y b geometrías, $a.Intersects(b)$ es verdadero si $a.Disjoint(b)$ es falso.	Booleano
Contains	Sean a y b geometrías, $a.Contains(b)$ es verdadero si b está contenido en a .	Booleano
Distance	Sean $p1$ y $p2$ geometrías en R^2 y d un número, $p1.Distance(p2, d)$ es verdadero si $p1$ se encuentra dentro de la región formada por todos los puntos que están a una distancia d o menos de $p2$.	Booleano

Cabe señalar que en el estándar SFS [OGCb], la OGC define *distance* como un operador nativo de la clase *Geometry*, de la forma: `distance(another: Geometry): Distance`. No obstante, en este trabajo se utiliza una variante adaptada específicamente para dos dimensiones, diseñada para aplicarse en escenarios donde es necesario detectar si un evento ocurre dentro de una distancia determinada.

Los modelos ilustrados en las Figuras 4.5 y 4.6 se pueden vincular con el propuesto en [Bou15], tal como se muestra en la Figura 4.7

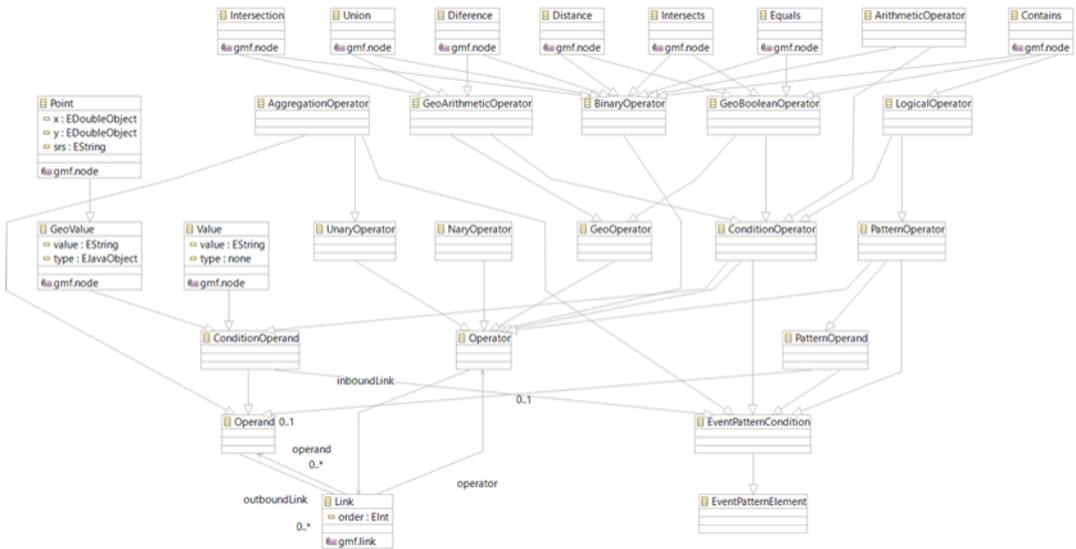


Figura 4.7: Operadores de Medit4CEP con capacidades Geo

En resumen, el modelo propuesto posibilita la representación de geoeventos y geoperadores de una forma general, sin estar atado a una plataforma CEP en particular. Sin embargo, para llevar a cabo la evaluación de estos eventos en un motor CEP específico, se requiere un mecanismo de traducción que convierta los conceptos del metamodelo al lenguaje de programación propio de ese motor. Los detalles de este mecanismo de traducción se abordarán detalladamente en el Capítulo 5.

5

Detalles de implementación

En el Capítulo 4 se proporcionó una descripción detallada de la solución propuesta desde una perspectiva conceptual, enfocándose en modelos y representaciones a nivel de metamodelos a través de una sintaxis abstracta, de la extensión CEP para geoeventos y geooperadores.

En este capítulo, se presenta la implementación concreta de dicha solución, detallando cómo se extendió el Editor Gráfico empleado en [Bou15]. Aquí, se discute la propuesta de una sintaxis concreta que posibilita la modelización gráfica de geoeventos, y además, la extensión del generador de código EPL para dar soporte a la expresividad geoespacial. Como culminación del proceso, se desarrolló una librería que utiliza y extiende Esper para permitir que el motor CEP evalúe los geoeventos y geopatronos generados.

La implementación se enfocó específicamente en lo siguiente:

- **Implementación del metamodelo:** Se implementó el metamodelo definido en el Capítulo 4
- **Extensión del Editor Gráfico:** Se crearon los elementos gráficos necesarios para representar los geoeventos y especificar patrones con geooperadores a nivel visual (geopatronos). Esta tarea implicó crear iconos y referenciarlos en el código para que sean visibles en el editor gráfico.
- **Generación de Código EPL:** Se extendió el código de la herramienta de referencia para permitir mapear los nuevos operadores geoespaciales al código EPL.
- **Librería Java para Esper:** Se implementó una librería Java para permitir operar con geoeventos y geopatronos.

5.1. Principales tecnologías utilizadas

Para implementar el modelo propuesto, se empleó un conjunto de tecnologías y herramientas que facilitaron la extensión del metamodelo de referencia, el editor gráfico y la generación del código EPL. En esta sección, se describen las principales tecnologías involucradas en la realización:

- **Eclipse Modeling Framework (EMF):** EMF¹ es un marco de trabajo que se usa para construir herramientas y aplicaciones basadas en un modelo de datos estructurado. En este proyecto, EMF se empleó en la extensión gráfica del editor (Geo CEP Editor) para extender el modelo de dominio y de patrones de referencia con las capacidades geoespaciales modeladas en la Sección 4.3.
- **Emfatic:** Es un lenguaje textual para la definición de modelos EMF. En el contexto de este trabajo, Emfatic² facilitó la extensión del metamodelo mencionado en el punto anterior.
- **Ecore:** Ecore³ es una implementación del lenguaje EMOF (Essential MOF), un subconjunto del lenguaje MOF (Meta-Object Facility) definido por la OMG (Object Management Group) para la construcción de metamodelos. En este trabajo, Ecore sirvió como base para la definición del metamodelo de dominios CEP, estableciendo la estructura y relaciones entre los diferentes elementos del dominio.
- **Graphical Modeling Framework (GMF):** GMF⁴ permite la generación basada en modelos de editores gráficos de diagramas a partir de modelos EMF. En este proyecto, GMF fue esencial para la creación del editor gráfico.
- **Eugenia:** Esta herramienta facilita la generación de editores gráficos para EMF utilizando GMF. En el proyecto, Eugenia⁵ se empleó para la creación de iconos y su integración en el editor gráfico.
- **Epsilon:** Epsilon⁶ es una familia de lenguajes de programación específicos del dominio (DSL) utilizados para tareas de modelado y metamodelado, como la transformación, validación, comparación y generación de modelos. Es parte del proyecto Eclipse y está diseñado para trabajar con modelos de cualquier metamodelo, siendo EMF el más común. Sus lenguajes incluyen EOL (Epsilon Object Language), ETL (Epsilon Transformation Language) y EVL (Epsilon Validation Language) entre otros.
- **EGL (Epsilon Generation Language):** EGL se utiliza para generar código a partir de modelos (M2T). Permite la generación de código utilizando plantillas y puede trabajar

¹EMF: <https://eclipse.dev/modeling/emf/>

²Emfatic: <https://eclipse.dev/emfatic/>

³Ecore: https://wiki.eclipse.org/Ecore_Tools

⁴GMF: <https://eclipse.dev/modeling/gmf/>

⁵Eugenia: <https://eclipse.dev/epsilon/doc/eugenia/>

⁶Epsilon: <https://eclipse.dev/epsilon/>

con modelos que se adhieren a cualquier metamodelo, siendo EMF el más comúnmente utilizado. En el trabajo se modificó el EGL existente para permitir generar los nuevos elementos modelados, constituyéndose en el Geo EPL Transformer.

- **EVL:** EVL se utiliza para la validación de modelos. Proporciona una sintaxis declarativa para definir restricciones sobre modelos y para proporcionar mensajes de error y reparaciones automáticas para violaciones de restricciones. Esto se utiliza para definir restricciones del modelo de dominio y patrones.
- **Java:** Java se usó para la implementación de la generación de código EPL y la creación de una librería para Esper. El código Java interactúa con el modelo de datos definido con EMF y genera el código EPL correspondiente.
- **JTS (Java Topology Suite):** JTS es una biblioteca de código abierto en Java que proporciona funciones para manipular objetos geométricos. Se usó en este proyecto para implementar geoeventos y las operaciones Geo de OGC.
- **Esper:** Esper es un motor de procesamiento de eventos complejos (CEP) y de análisis de eventos que se utiliza para identificar patrones significativos en los flujos de eventos. Esper se empleó en este proyecto para evaluar los geoeventos y ejecutar las operaciones definidas con los geoperadores.

5.2. Extensión del Editor Gráfico

En el nuevo metamodelo, una geometría definida por el modelo de referencia OGC SFS 2.5 constituye una propiedad nativa del evento. Esto, a nivel del Editor Gráfico, repercute en la capacidad de los eventos que se pueden incluir en el dominio modelado (geoeventos). Para lograr esto, se ha extendido el CEP Model Definition, modificando la sintaxis concreta existente.

Además, de forma análoga, para que las operaciones geoespaciales sean accesibles en el editor, se ha modificado el CEP Pattern Definition, añadiendo a la sintaxis concreta existente la capacidad de representar patrones que incluyan operadores geoespaciales.

Sintaxis Concreta

La sintaxis concreta especifica como se representan los diferentes artefactos en el editor incorporados para ser usados por el usuario que interactúa con el editor.

En el caso del dominio fue necesario definir los elementos definidos en la Figura 5.1

Type	Operator	Graphic Representation
Value		
	Point	
	GeoValue	

Figura 5.1: Sintaxis concreta para Geo Values

En el caso de los patrones fue necesario definir los elementos gráficos definidos en la Figura 5.2

Type	Operator	Graphic Representation
GeoBoolean		
	Contains	
	Intersects	
	Equals	
	Distance	
GeoArithmetic		
	Diference	
	Union	
	Intersection	

Figura 5.2: Sintaxis concreta para los Geo Patrones

Restricciones del Modelo

Las restricciones especifican condiciones invariantes que deben cumplirse para que el sistema sea modelado. En el caso de los nuevos elementos que se introducen al modelo EMF tienen algunas restricciones que se pueden representar usando EVL. Estas restricciones implican limitantes a como se puede usar el editor.

5.2.1. Editor Gráfico extendido

Como resultado de la implementación de la nueva sintaxis concreta se obtuvo un editor gráfico con capacidades geoespaciales, tal como se puede observar en la Figura 5.3, que permite a los usuarios escribir y gestionar las reglas de control utilizando operadores y valores geoespaciales. Este enfoque basado en gráficos proporciona una interfaz gráfica, facilitando la creación y manipulación de reglas de control geoespacial.

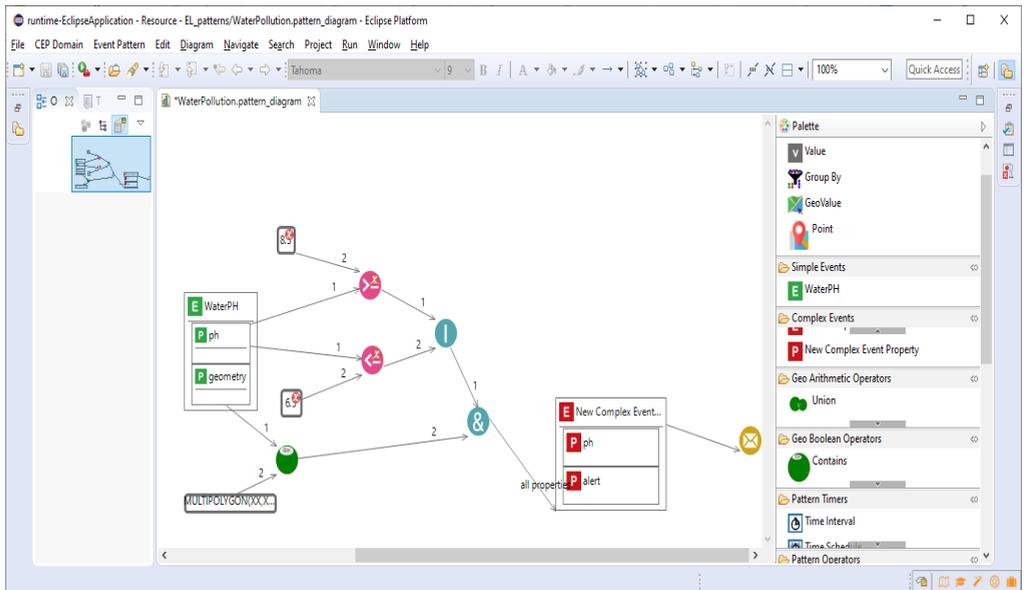


Figura 5.3: Layout de herramienta para manejo de sensores y alertas

En el Editor, se puede ver en la Paleta de Herramientas las nuevas capacidades geoespaciales. Por un lado, se puede utilizar un GeoValue o un Point como valor; y por el otro, es posible utilizar los nuevos operadores geoespaciales que están disponibles agrupados según su naturaleza: los booleanos (Geo Boolean Operators) y los aritméticos (Geo Arithmetic Operators).

5.3. Generación del Código GeoEPL

Si bien a nivel de modelos la solución es independiente del lenguaje CEP a generar, para la extensión se utilizó Esper EPL ya que la herramienta original así lo realiza. Esto conlleva a que el editor gráfico deba poder ejecutar la generación del EPL con capacidades geoespaciales (geoEPL).

Transformación Modelo a Código (Texto)

Para transformar del modelo a código interpretable por un motor CEP, se usa el lenguaje EGL. Este lenguaje permite definir plantillas para mapear los objetos del modelo de texto, específicamente al EPL en este contexto.

En este trabajo se extendió el código EGL y Java ya existente para que el Editor pueda generar un geoEPL. Esta ampliación implicó añadir a la traducción original la capacidad de interpretar los nuevos operadores, geoeventos y geovalues.

La Figura 5.4 muestra un fragmento del código EGL utilizado para la transformación de operadores, poniendo como ejemplo el operador «contains».

```
operation traversal(r) : String {
    ...
    if (r.isTypeOf(Event)) {
        ...
    }
    if (r.isTypeOf(EventProperty)) {
        ...
    }
    ...

    if(r.isTypeOf(Contains)) {
        if (r.inboundLink.at(0).order == 1) {
            return 'contains('+ traversal(getOperand(r,0)) + ', '
                + traversal(getOperand(r,1)) + ')';
        }
        else {
            return 'contains('+ traversal(getOperand(r,1)) + ', '
                + traversal(getOperand(r,0)) + ')';
        }
    }
    ...
}
```

Figura 5.4: Extracto del EGL extendido

Este fragmento de EGL ilustra cómo la operación «traversal» procesa un objeto «r» y devuelve una representación en cadena de ese objeto. La lógica interna verifica el tipo de objeto y produce la representación geoEPL correspondiente.

El operador «contains» se genera comprobando el orden de los operandos y luego generando la cadena correspondiente. Si el primer operando es el contenedor y el segundo es el contenido, se genera «contains(operando1, operando2)». Si es al revés, se invierte el orden en la cadena generada.

Como resultado de esta transformación, se puede obtener un geoEPL como el del ejemplo de la Figura 5.5. En el mismo, se define un patrón llamado «PollutionWaterPattern» que se aplica a eventos de tipo «WaterPH». Este patrón busca detectar eventos donde la propiedad «pH» exceda el valor: «3» y que ocurra dentro de una zona delimitada por el polígono especificado.

```
@Name("PollutionWaterPattern")
@Description("Evaluation of Water PH")
@Tag(name="domainName", value="Sensor")
insert into PollutionWaterPattern
select a1.ph as ph
from pattern [a1 = WaterPH((a1.ph > 3 and contains(a1.geometry,
MULTIPOLYGON((( -6326845.70522061 -4094578.73118032,
-6326845.70522061 -4170404.26323922,
-6171525.66374513 -4185691.66889625,
-6143396.83733619 -4081737.31042841,
-6326845.70522061 -4094578.73118032))))))] ]
```

Figura 5.5: EPL que valida la polución en el agua

5.4. Librería Java para Esper

Para la evaluación de sentencias del lenguaje de procesamiento de eventos (EPL) con eventos y operaciones geoespaciales, se implementó una librería Java diseñada para integrarse con el motor de procesamiento de eventos complejos Esper.

La arquitectura de la librería se basa en una estructura orientada a objetos que facilita la manipulación y el procesamiento de eventos geoespaciales y operaciones asociadas, tal como se ilustra en la Figura 5.6.

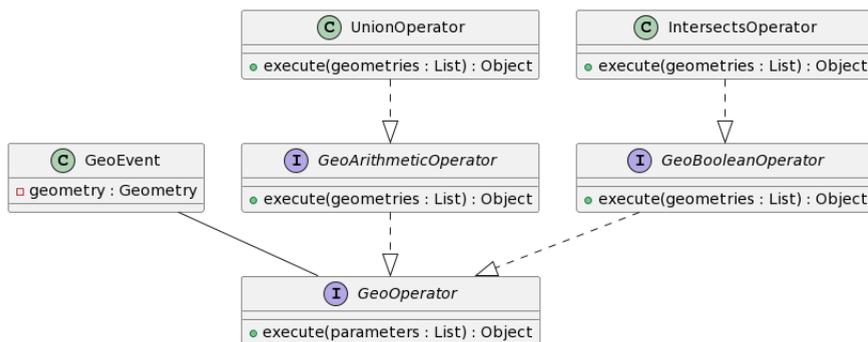


Figura 5.6: Diseño de Clases de la Librería GeoEsper

La librería se compone de varios componentes clave, descritos a continuación:

- **GeoEvent:** Esta clase representa un evento geoespacial. Cada instancia de GeoEvent contiene un atributo "geometry" de tipo "Geometry", proporcionado por la librería Java Topology Suite (JTS). Este atributo refleja la ubicación espacial del evento y puede tomar varias formas, como puntos, líneas, polígonos, entre otros.
- **GeoOperator:** Es una interfaz que establece una estructura común para todas las operaciones geoespaciales. Define un método «execute» que acepta una lista de parámetros y devuelve un objeto, permitiendo la realización de la operación geoespacial deseada.
- **IntersectsOperator, UnionOperator, ContainsOperator, DistanceOperator:** Estas son clases que implementan la interfaz GeoOperator. Cada una representa una operación geoespacial específica: intersección, unión, contiene y cálculo de distancia, respectivamente. Cada una de estas clases utiliza el método *execute()* para realizar la operación correspondiente con dos geometrías y devolver el resultado.

En esencia, con la librería se proporciona un marco de trabajo para la manipulación de eventos geoespaciales y operaciones relacionadas en el contexto de Esper, permitiendo una integración más fluida de los datos geoespaciales en el procesamiento de eventos complejos. Es importante destacar que esta solución es extensible y se puede ampliar para incorporar más operaciones o características en el futuro.

Ejemplo de Uso

En primer lugar es necesario configurar el motor Esper para que reconozca el evento que se quiere monitorear, lo cual se logra con la sentencia `config.addEventType()`. Adicionalmente, es necesario registrar las operaciones geo para que cuando se evalúen los EPL las mismas puedan ser invocadas, esto puede realizarse utilizando `config.addPluginSingleRowFunction()`. Un ejemplo de configuración inicial puede verse en la Figura 5.7 en la cual se establece el evento GeoEvent y las operaciones contains e intersects.

```
// Registro el Evento a escuchar
config.addEventType("GeoEvent", GeoEvent.class.getName());
// Configuro las funciones Geo
config.addPluginSingleRowFunction("contains",
    ContainsOperator.class.getName(), "execute");
config.addPluginSingleRowFunction("intersects",
    IntersectsOperator.class.getName(), "execute");
```

Figura 5.7: Configuración de funciones y eventos geo

Para realizar pruebas con un EPL enviando eventos manualmente, se puede desarrollar un método *main* empleando las funcionalidades que Esper ofrece de forma nativa:

- **createEPL:** para crear el EPL a utilizar.

- **addListener**: para agregar un listener y escuchar eventos.
- **sendEvent**: para lanzar los eventos.

La Figura 5.8 ilustra un uso típico de estas funciones.

```
String epl = "@Name('ParametroNotOK') " +
    "@Description('ParametroNotOK') " +
    "@Tag(name='domainName', value='GeoEvent') " +
    "insert into ParametroNotOK " +
    "select a1.parametro as parametro " +
    "from GeoEvent as a1 " +
    "where (contains(a1.geometry, 'POINT(3 5)') " +
    "or intersects(a1.geometry, 'POINT(5 8)')) " +
    "and a1.parametro > 5";

EPStatement stmt = epService.getEPAdministrator().createEPL(epl);

stmt.addListener(new UpdateListener() {
    public void update(EventBean[] newEvents, EventBean[]
        oldEvents) {
        if (newEvents != null) {
            for (EventBean event : newEvents) {
                System.out.println("ParametroNotOK event
                    received with parametro: " + event.get("
                        parametro"));
            }
        }
    }
});

WKTReader reader = new WKTReader();

// Send some GeoEvent events to the engine using the sendEvent
// method
epService.getEPRuntime().sendEvent(new GeoEvent(reader.read("POINT
    (1 2)"), 1));
epService.getEPRuntime().sendEvent(new GeoEvent(reader.read("
    POLYGON((0 0, 0 10, 10 10, 10 0, 0 0)"), 6));
epService.getEPRuntime().sendEvent(new GeoEvent(reader.read("POINT
    (5 8)"), 2));
epService.getEPRuntime().sendEvent(new GeoEvent(reader.read("POINT
    (7 9)"), 3));
```

Figura 5.8: Ejecución de Eventos de Prueba

Finalmente, la ejecución de este ejemplo conduce a la detección de un evento específico que cumple con todas las condiciones impuestas, generando la salida siguiente:

```
ParametroNotOK event received with parametro: 6
```

6

Caso de Estudio

En este capítulo se presenta un Caso de Estudio ilustrativo enmarcado en el contexto del cumplimiento de leyes medioambientales, el cual es resuelto utilizando los elementos propuestos en este trabajo.

6.1. Descripción General

El municipio CFA de Sauce está interesado en monitorear la calidad del agua del río Santa Lucía en el tramo que corresponde a su jurisdicción. El objetivo del organismo es poder determinar si el agua es apta para consumo de su población para lo cual debe cumplir con los estándares establecidos en el decreto perteneciente al Código de Aguas del Uruguay en el cual se establece que el agua destinada para este fin debe cumplir con los valores de calidad establecidos en la Clase 1 de la Tabla 6.1.

Tabla 6.1: Estándar de Calidad del Agua

Parámetro	Clase 1	Clase 2a	Clase 2b	Clase 3	Clase 4
PH (min-max)	6,5 - 8,5	6,5 - 9,0	6,5 - 8,5	6,5 - 9,0	6,0 - 9,0
Cromo Total (max)(mg/l)	0,05	0,005	0,05	0,05	0,5
Cianuro (max)(mg/l)	0,005	0,005	0,005	0,005	0,05
Plomo (max)(mg/l)	0,03	0,03	0,03	0,03	0,05
Cadmio (max)(mg/l)	0,001	0,001	0,005	0,001	0,01
Mercurio (max)(mg/l)	0,0002	0,0002	0,0002	0,0002	0,0002

Para evaluar estos parámetros el municipio planea utilizar datos abiertos recolectados a través de sensores ubicados a lo largo del río que podrían proporcionar organismos como por ejemplo DINAMA¹. Estos datos incluirían la información de las mediciones de los distintos parámetros y a su vez la ubicación del sensor donde fue captada. También se podrían utilizar datos de mediciones proporcionados por particulares, compartidos a través de redes sociales y validados, por ejemplo, por el MTOP², que además podría ofrecerlos a través de servicios web. A este tipo de datos se les conoce como información geográfica voluntaria (VGI).

En la Figura 6.1 se pueden ver resaltados, y dentro del radio marcado, los sensores cuyas lecturas son las que son de interés de acuerdo a la normativa a monitorear. Los que le siguen en proximidad se diferencian de los más lejanos también.

¹DINAMA: Dirección Nacional de Medio Ambiente del Uruguay

²MTOP: Ministerio de Transporte y Obras Públicas del Uruguay

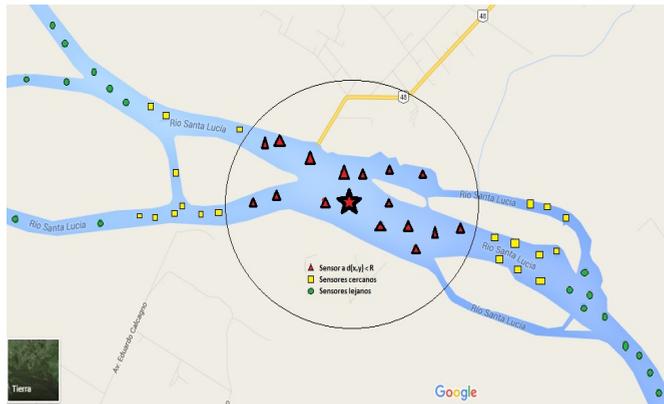


Figura 6.1: Caso de Estudio: Calidad del Agua

El municipio CFA necesita monitorear estos valores y en caso que los mismos salgan de los rangos permitidos es necesario generar una alerta para tomar las acciones correspondientes. En particular en una primera etapa se realizará el control de que las lecturas reportadas en el área de interés no superen el valor esperado de pH en las mediciones.

En una etapa futura podrían tomarse nuevos parámetros o incluso establecer reglas sobre los sensores adyacentes al área de interés que permitan monitorear tendencias en vez de violaciones normativas estáticas.

Este caso de estudio se abordará realizando primero un modelado de la realidad utilizando la herramienta especificada en el trabajo, luego se realizará con la misma la generación del código geoEPL para luego realizar una simulación de la prueba en tiempo de ejecución.

6.2. Modelado del Caso de Estudio

Para el caso de estudio se evalúa la Clase 1 especificada en el artículo 5 del Decreto N° 253/979 que es la que establece los parámetros para el agua potable destinada a poblaciones para su consumo. Entre los parámetros a evaluar dentro de esta clase se considera la presencia de olor, color, pH alterado, sodio, nitratos, cianuro, coliformes, plomo, cromo, entre otros.

Para modelar el dominio se considera que se quiere controlar el pH, pero la solución es aplicable a los demás parámetros de calidad del agua.

En la Tabla 6.2 se presenta el resumen del caso abordado y su modelado:

Tabla 6.2: Calidad del agua - CLASE 1

	Contaminación Agua CLASE 1
Ley/Decreto/Normativa	Ley 14859, Decreto 253/79
Sección de interés	Artículo 5
Organismos competentes	Ministerio de Transporte y Obras Públicas (MTOB) y Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente (MVOTMA)
Factor Monitoreado	Agua
Parámetros a monitorear	PH
Elementos de medición	Los parámetros pueden ser monitoreados por medio de sensores específicos para cada fin.
Criterio de Alerta	Se da una alarma cuando el PH se sale de su rango de control y la medición es detectada dentro de la zona de interés.
Evento	El PH sale del límite de control
Acción a tomar	Alarma

Teniendo en cuenta la realidad, se utilizó para modelarla el modelo conceptual que se presentó en la Sección 2.7 y en este sentido se define un evento de nombre **WaterPH** asociado a la fuente que proporciona datos del pH del agua, a partir de esto se puede plantear la solución utilizando el Editor Gráfico.

6.2.1. Detalles de Modelado

La Figura 6.2 ilustra un modelo detallado que representa la instanciación del modelo conceptual propuesto en la Sección 4.2. Este modelo se ha desarrollado con el propósito de capturar y representar de manera gráfica las interacciones y relaciones entre los diferentes componentes del dominio.

El evento base instanciado es «WaterPH», que es la representación del geoevento a monitorear. Este evento además de las propiedades intrínsecas de los eventos, cuenta con los atributos **pH** para capturar los valores de este elemento que influye en la calidad del agua y por otro lado el atributo **geometry** que refiere a la zona donde se quiere hacer cumplir la regulación (en este caso en el río Santa Lucía).

Las restricciones que definen los criterios de alerta están vinculadas directamente al Artículo 5 del Decreto 253/79, que a su vez hace referencia a la Ley 14859. En el modelo, estas restricciones se asocian a través del objeto «Constraint». Esta restricción establece que si el pH se sale de su límite de control y la medición es detectada fuera de la zona de interés, se debe generar una alarma. Esta restricción es la que determina el patrón de evento complejo «CEPEventPattern» que se quiere detectar.

Cada elemento del modelo tiene una correspondencia directa con uno de la realidad. Por ejemplo, el objeto «Decree» en el modelo representa el Decreto 253/79 mencionado en la tabla y es una instancia de la entidad «Document». De manera similar, el objeto «Section» que tiene en la propiedad **name** el valor «art5» es una instancia de «Section» y refiere a el artículo 5 anteriormente mencionado.

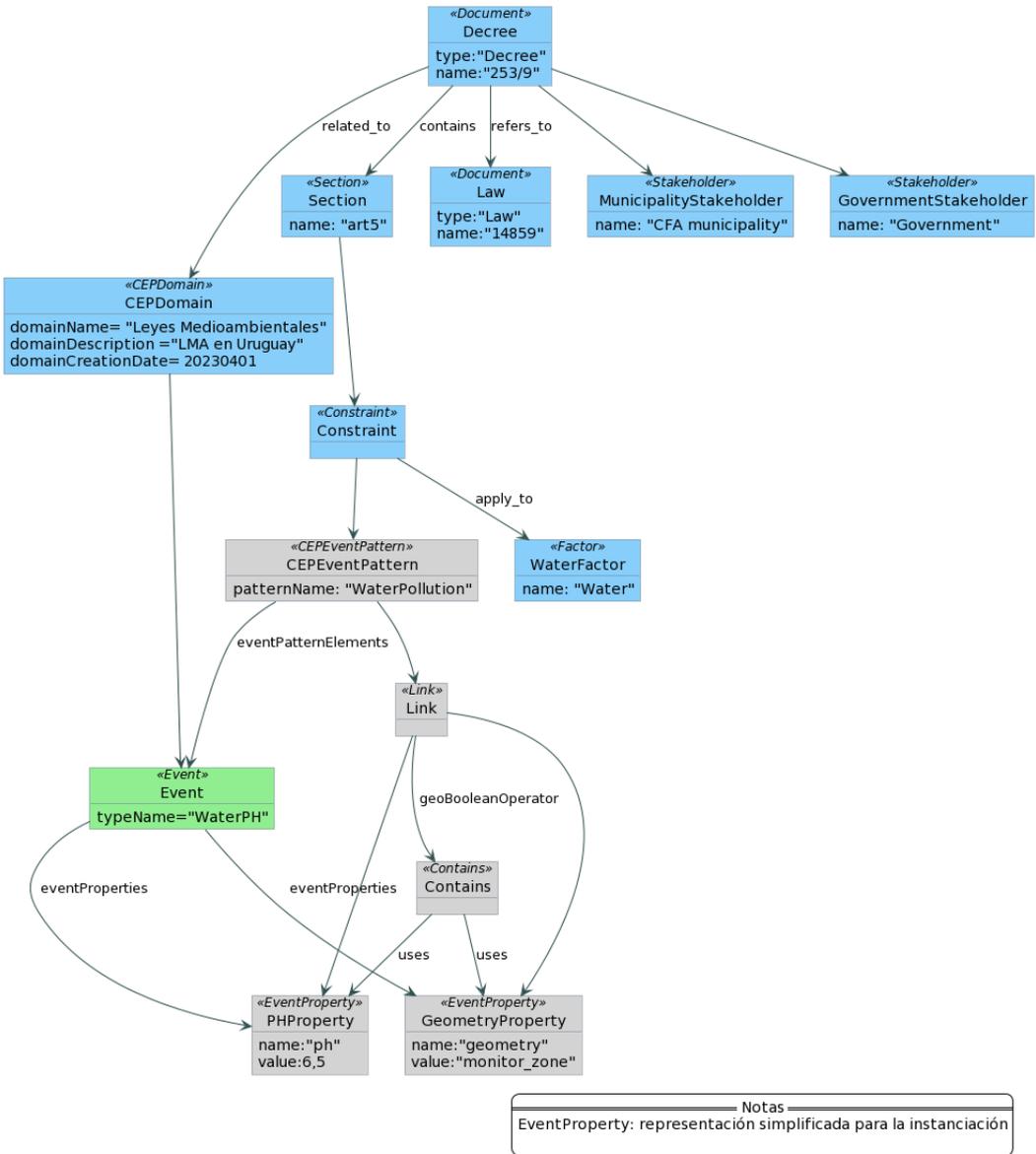


Figura 6.2: Modelo de Dominio de Eventos

Dada la naturaleza geoespacial del caso de estudio, es crucial no solo monitorear los valores en sí, sino también asegurarse de que provengan de sensores ubicados dentro de áreas geográficas específicas de interés. Por esta razón, hemos optado por utilizar para la representación el operador «Contains» tal como se ve en la instanciación de la Figura 6.2. Este operador permite determinar si la geometría de un geoevento ocurre dentro de un área geoespacial definida. Este operador puede recibir cualquier geoevento como operando. En nuestro caso, se ha instanciado específicamente para eventos puntuales que tienen asociada la Geometry:Point. El modelo detallado del componente puede verse en la Figura 6.3.

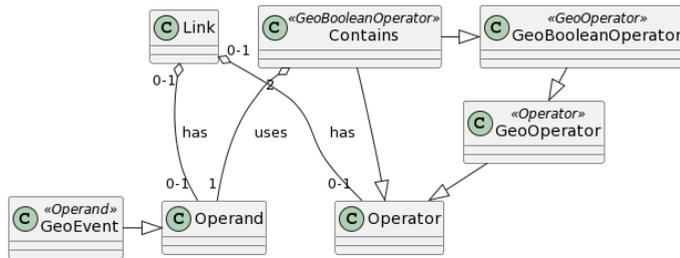


Figura 6.3: Geo Operador contains

Al margen de la instanciación particular realizada, el modelo nos permite especificar otros tipos de eventos que ayuden a predecir tendencias en las mediciones. Por ejemplo, podría definirse un evento complejo que genere una alarma cuando los sensores próximos a la zona principal detecten valores excesivos por un intervalo dado de tiempo, ya que esto podría ser un indicador de que en breve los sensores cercanos al lugar donde se necesita que el agua cumpla con los estándares fijados en la Clase 1 se están por alterar.

6.2.2. Modelado Gráfico en el Editor

El Editor Gráfico extendido (Geo CEP Editor) permite realizar el modelado de los eventos y las reglas que se quieren detectar de forma gráfica, tal como se ilustra en la Figura 6.4. Allí, utilizando como guía el modelo conceptual instanciado de la Figura 6.2, se especifica tanto el modelo de dominio como los patrones a utilizar.

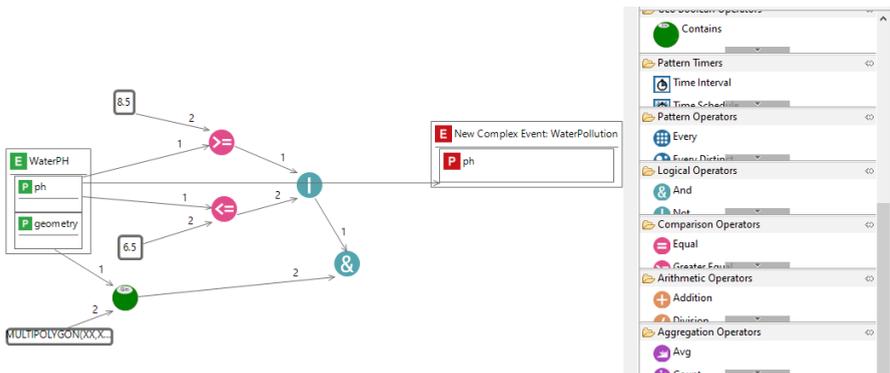


Figura 6.4: Diseño del Geo Patrón WaterPollution en el Editor

Luego de tener la definición del dominio y del patrón, se valida el patrón y genera el código geoEPL a ser evaluado por el motor CEP extendido y que se puede ver en la Figura 6.5.

```
@Name ("WaterPollution")
@Description ("WaterPollution")
@Tag (name="domainName", value="Leyes Medioambientales")
insert into WaterPollution
select a1.ph as ph
from pattern [a1 = WaterPH(((a1.ph >= 8.5 or a1.ph <= 6.5)
and contains(a1.geometry,
MULTIPOLYGON((( -6326845.70522061 -4094578.7,
-6326845.7 -4170404.2,
-6171525.6-4185691.6,
-6143396.8 -4081737.3,
-6326845.7 -4094578.7)))))))]
```

Figura 6.5: GeoEPL del caso de estudio.

Este código es el que debe ser interpretado utilizando la librería implementada en este trabajo y detallada en la Sección 5.4.

6.3. Control de normativas en ejecución

En la Figura 6.6 se muestra el flujo que siguen los datos a través del prototipo para la detección de los patrones de eventos complejos y la generación de alarmas correspondientes.

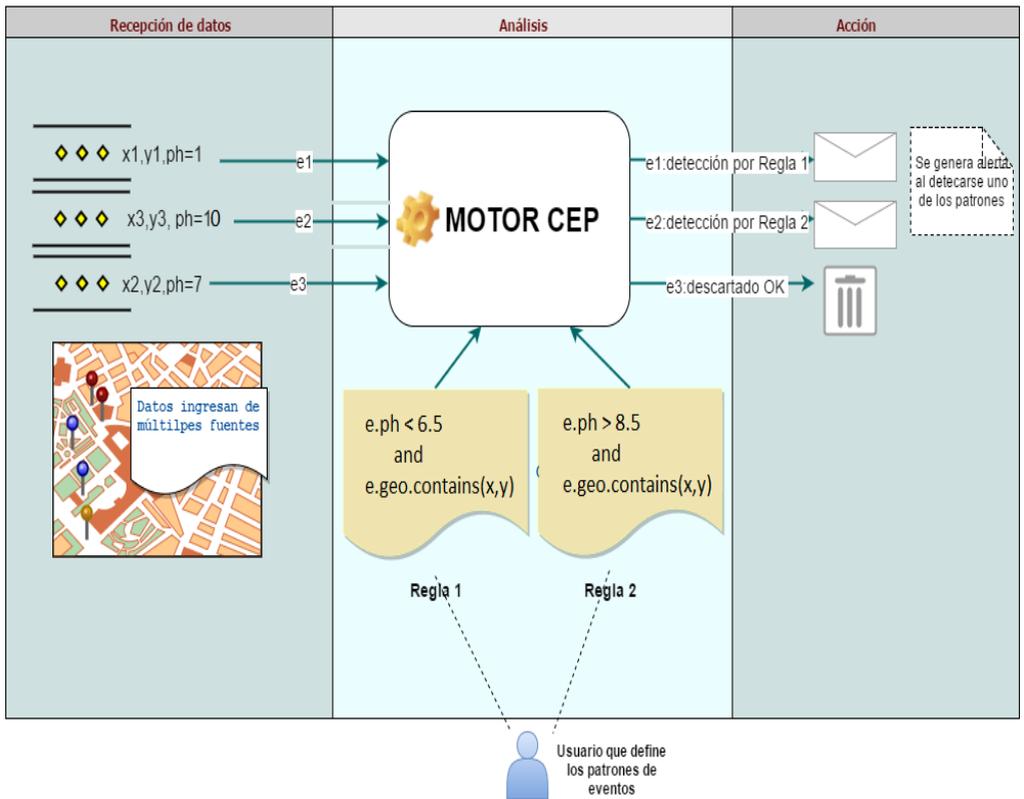


Figura 6.6: Flujo de datos y generación de Alertas

Una vez que el usuario define los patrones o reglas a detectar, el funcionamiento es el siguiente:

- **Recepción de datos:** Los eventos provienen desde diferentes fuentes de datos y sensores en un formato común y son captados por el prototipo. En el ejemplo, se capturan los eventos de mediciones del pH del agua, junto con la coordenada desde donde se realizaron esas mediciones.
- **Análisis:** El motor CEP con capacidades extendidas evalúa las reglas que hay definidas y si alguno de los eventos cumple con el patrón en ellos presente, entonces es detectado. En el ejemplo se pueden observar el patrón «WaterPollution» definido y cuyo geoEPL fue generado.
- **Acción:** Cuando se detecta un patrón puede realizarse una acción. En este caso se genera una alarma a través del envío de un mail indicando que existe una medición que está fuera del rango. Los eventos que no aplican a ninguna regla son descartados, como

lo es el evento 3 (e3) del ejemplo.

Esta interacción se representa en términos de componentes en el diagrama de secuencia de la Figura 6.7, en la cual se representa como interactúan los componentes de la plataforma y cuáles son sus distintas responsabilidades:

- **EventProducers:** Es la fuente de datos de eventos.
- **Middleware:** Recibe los eventos y utiliza la librería implementada para extender Esper.
- **GeoEsper:** Interactúa con el motor Esper evaluando los eventos con los distintos geo-patrones registrados.
- **EventConsumers:** Son los consumidores de los eventos complejos y alertas generadas.

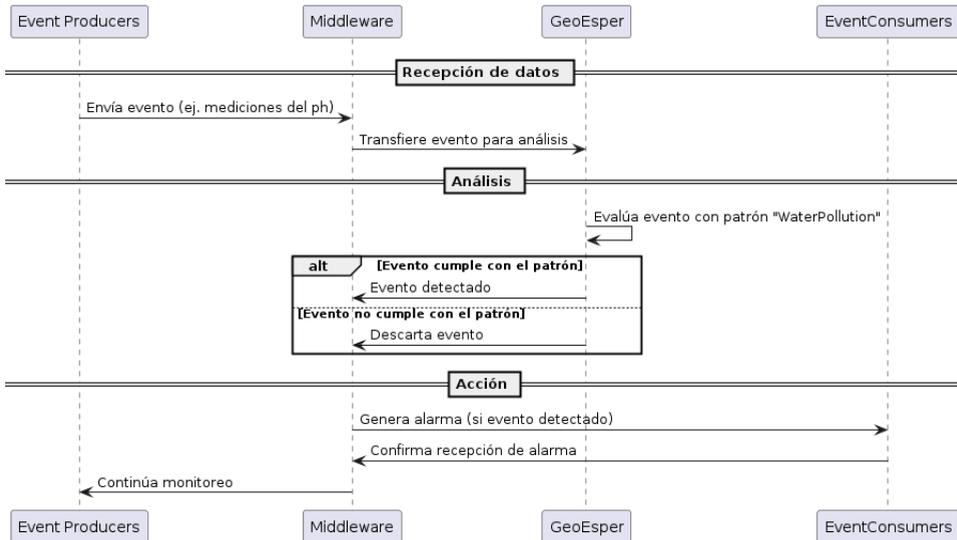


Figura 6.7: Diagrama de Secuencia de la ejecución del Caso de Uso

7

Conclusiones y Trabajo Futuro

En este capítulo se presentan las conclusiones de la tesis. Primero se presenta el resumen del trabajo realizado y sus principales contribuciones. Luego, se describen problemáticas y desafíos identificados. Por último, se comentan posibles trabajos a futuro.

7.1. Resumen y Conclusiones

La tesis propuso desarrollar mecanismos basados en CEP para la verificación de conformidad de leyes medioambientales con requerimientos geoespaciales. Para ello, se desarrolló una extensión geoespacial a una solución de CEP ya existente que permite modelar el problema del cumplimiento de leyes utilizando eventos que cuentan con un componente geoespacial. Además, se pueden especificar patrones de eventos complejos utilizando operadores geoespaciales, los cuales permiten correlacionar eventos en base a su ubicación. Este modelo es independiente de una plataforma CEP específica, independizando la solución propuesta de una implementación particular.

También se desarrolló un prototipo que sirve como prueba de concepto para la implementación de una herramienta de monitoreo de leyes utilizando una tecnología de CEP específica. Esta implementación se basa en la propuesta presentada en [Bou15], donde se introduce un editor gráfico para representar eventos y generar código destinado a la evaluación de eventos en tiempo real. Se extendió este editor para incluir la capacidad de representar geoeventos y patrones geoespaciales. Además, se añadió la funcionalidad de generar código geoEPL mediante la transformación del modelo geoespacial a código, permitiendo así la evaluación de geoeventos en un motor CEP.

Finalmente, se presentó un caso de estudio que aplica a la legislación del agua en Uruguay. El mismo fue modelado con la extensión geoespacial a los efectos de evaluar la propuesta y ejecutado con el prototipo de herramienta extendida implementada.

Las principales contribuciones de esta tesis fueron:

- **Modelo de Eventos con Ubicación Geográfica:** Se introduce una representación a nivel de modelo que permite representar eventos asociados a una ubicación mediante el uso de una propiedad de tipo Geometría (p. ej. línea, punto, polígono). A través de esta propiedad es posible especificar la ubicación geográfica del evento y definir eventos complejos utilizando operadores geoespaciales que permiten correlacionar eventos teniendo en cuenta su ubicación.
- **Representación Conceptual de Leyes Medioambientales:** Permite definir una representación conceptual de las leyes medioambientales para ser usada como lenguaje común de especificación. Brinda una forma unificada de representar los conceptos de la realidad en función de abstracciones que están presentes en las leyes.
- **Herramienta Gráfica para GeoEventos y GeoPatrones:** Se brinda una herramienta gráfica que permite definir geoeventos y geopatrones mediante una interfaz gráfica. Esto facilita la escritura de las restricciones medioambientales establecidas en las leyes como geoeventos complejos.
- **Generación de Lenguaje con Expresividad Geoespacial:** Se permite la generación de un lenguaje basado en Esper EPL que cuenta con expresividad geoespacial (geo-EPL). Adicionalmente, se crea una librería que permite interpretarlo extendiendo el motor CEP de Esper.

7.2. Problemáticas y Desafíos Identificados

Durante el trabajo realizado se identificaron varias problemáticas y desafíos, algunos abordados en su totalidad y otros que pueden abordarse en trabajo a futuro:

- **Diversidad de las leyes medioambientales:** Uno de los desafíos identificados fue la existencia de diversas leyes y normativas medioambientales, cada una con textos distintos y objeto de regulación (factor) distinto. Este punto fue abordado abstrayendo la realidad y definiendo un modelo de representación de las leyes.
- **Solución Integral:** Para cumplir con todos los aspectos del proceso de definición de reglas, generación, ejecución y monitoreo, es necesario implementar una plataforma integral. En esta tesis se ha realizado una Prueba de Concepto pero no una implementación completa.
- **Abstracción e Independencia Tecnológica:** Otro desafío que enfrentamos fue el diseño de una solución abstracta que pudiera implementarse de forma independiente a una tecnología específica. Aunque la solución propuesta cuenta con varios niveles de abstracción y su modelado de la realidad, junto con la representación a través de eventos y patrones geoespaciales, es independiente de tecnologías concretas, aún existe una

dependencia a nivel del Editor durante la fase de diseño y del lenguaje EPL específico que debe ser interpretado por el motor en tiempo de ejecución.

- **Pruebas en Tiempo Real:** El realizar una prueba en tiempo real implica el poder simular la información de los sensores procesarla como eventos y evaluar los patrones definidos en la ley. Esta validación completa con datos reales quedó fuera del alcance del trabajo.

7.3. Trabajo a Futuro

Se identificaron varias líneas de trabajo que quedaron fuera del alcance y pueden ser tomadas en cuenta a futuro para extender el mismo:

- **Validación del Modelo Conceptual con leyes internacionales:** Si bien para elaborar el modelo conceptual se analizó el marco normativo internacional más importante, en este trabajo se puso foco en las leyes medioambientales del Uruguay. Por lo tanto, pueden existir particularidades en otros marcos normativos que no hayan sido tenidos en cuenta en la abstracción planteada y que podrían originar cambios en el modelo propuesto en la Sección 4.2.
- **Extensión de los operadores:** En la solución propuesta se implementaron algunos de los operadores geoespaciales especificados por OGC, los cuales podrían ser complementados en caso de ser necesario. Para esto debería por una lado extenderse el Editor Gráfico con los elementos visuales correspondientes, así como la lógica para interpretar los mismos y generar patrones considerando estos nuevos operadores. Por otro lado, debería extenderse la librería de Esper implementada para poder interpretar el nuevo operador geoEPL.
- **Soporte a otros lenguajes:** El editor propuesto genera código en EPL extendido (asociado con Esper), aunque es posible usar otros lenguajes CEP distintos al de Esper. Esto sería posible modificando el código EGL para realizar transformaciones distintas de modelo a código.
- **Restricciones en el Editor:** La solución tiene algunas mejoras a incorporar, en particular quedó pendiente la definición de restricciones en el editor que permita bloquear acciones incorrectas.
- **Pruebas exhaustivas:** La solución planteada en el trabajo contempla todo el ciclo de vida desde la definición de los eventos y reglas, validación, generación de código y evaluación de eventos. Estos componentes implementados fueron probados con casos puntuales pero sería interesante el realizar pruebas en diferentes escenarios para validar y perfeccionar la solución propuesta.
- **Talleres con usuarios:** La solución proporciona una forma gráfica de modelar los even-

tos y patrones. A futuro se podría realizar un taller con usuarios para evaluar la usabilidad del editor.

- **Integración en Plataforma Real:** En el trabajo se construye una Prueba de Concepto completa para validar el flujo, pero para poder ser aplicada en situaciones reales se debería iterar la solución para mejorar sus características. Por ejemplo, se puede trabajar en la automatización de la interpretación de los patrones generados por el editor, o sea que los mismos puedan ser tomados automáticamente por el componente que reciba los datos y se comunique con el motor CEP. Al ser una plataforma real también se debe procurar que cuente con ciertas características deseables en soluciones productivas como lo son: alta disponibilidad de la plataforma, control de los tiempos de respuesta, herramientas de gestión de alertas y acciones en caso de detectar violaciones de las leyes, entre otras.

Este punto puede implicar el explorar la posibilidad de integrar la solución con otras herramientas de monitoreo y análisis.

- **Complementación tecnológica de CEP con Inteligencia Artificial (IA):** La combinación de CEP con IA [Zha18] podría permitir no solo la detección de patrones en eventos en tiempo real, sino también la predicción y anticipación de posibles eventos futuros basados en datos históricos y tendencias emergentes [Fül12]. Este enriquecimiento tecnológico ofrecería un valor adicional al monitoreo de leyes medioambientales, permitiendo a las organizaciones prever y prepararse para posibles incumplimientos antes de que ocurran. Además, la integración de técnicas de aprendizaje automático puede mejorar la precisión y adaptabilidad del sistema, ajustándose dinámicamente a nuevos patrones y comportamientos en los datos.
- **Pruebas de Rendimiento:** En un escenario real, la cantidad de datos recibidos por una plataforma de evaluación de eventos puede ser muy grande. Por lo tanto, es recomendable llevar a cabo pruebas de rendimiento para evaluar cómo se comporta el sistema en condiciones de carga, estrés y rendimiento. Estas pruebas permitirán determinar la capacidad y la eficiencia del sistema cuando se enfrenta a volúmenes significativos de datos y tráfico.

Referencias

- [Bal06] Krishnakumar Balasubramanian, Aniruddha S. Gokhale, Gabor Karsai, Janos Sztipanovits y Sandeep Neema. «Developing Applications Using Model-Driven Design Environments». En: *Computer* 39.2 (2006), págs. 33-40. DOI: 10.1109/MC.2006.54. URL: <https://doi.org/10.1109/MC.2006.54>.
- [Béz05] Jean Bézivin. «On the Unification Power of Models». En: *Software and Systems Modeling* 4.2 (2005), págs. 171-188.
- [Bou15] Juan Boubeta-Puig, Guadalupe Ortiz e Inmaculada Medina-Bulo. «MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0». En: *Knowledge-Based Systems* 89 (nov. de 2015), págs. 97-112. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2015.06.021.
- [Cab20] C. Cabanillas, Manuel Resinas y Antonio Ruiz-Cortés. «A Mashup-Based Framework for Business Process Compliance Checking». En: *IEEE Transactions on Services Computing* (2020). DOI: 10.1109/TSC.2020.3001292. URL: <https://dx.doi.org/10.1109/TSC.2020.3001292>.
- [Cal16] Francois-élie Calvier, Abderrahmen Kammoun, Antoine Zimmermann, Kamal Singh y Jacques Fayolle. «Ontology driven complex event pattern definition». En: oct. de 2016. URL: https://dx.doi.org/10.1007/978-3-319-48472-3_31.
- [CEP] CEPAL. *Informe de la Cumbre Mundial sobre el Desarrollo Sostenible*. URL: http://www.cepal.org/rio20/noticias/paginas/6/43766/WSSD_Informe.ESP.pdf (visitado 20-12-2017).
- [Chu20] Ben Chugg, Nicolas Rothbacher, Alex Feng, Xiaoyi Long y Daniel E. Ho. «Detecting Environmental Violations with Satellite Imagery in Near Real Time: Land Application under the Clean Water Act». En: *arXiv preprint arXiv:2208.08919* (2020).
- [CM12] Gianpaolo Cugola y Alessandro Margara. «Processing Flows of Information: From Data Stream to Complex Event Processing». En: *ACM Computing Surveys* 44 (jun. de 2012). DOI: 10.1145/2187671.2187677.

- [Dan09] Florian Daniel, Fabio Casati, Vincenzo D'Andrea, Emmanuel Mulo, Uwe Zdun, Schahram Dustdar, Steve Strauch, David Schumm, Frank Leymann, Samir Sebahi, Fabien De Marchi y Mohand-Said Hacid. «Business Compliance Governance in Service-Oriented Architectures». En: *The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA)*. 2009, págs. 113-120. DOI: 10.1109/AINA.2009.112. URL: <http://dx.doi.org/10.1109/AINA.2009.112>.
- [Eck11] Michael Eckert, François Bry, Simon Brodt, Olga Poppe y Steffen Hausmann. «A CEP Babelfish: Languages for Complex Event Processing and Querying Surveyed». En: *Reasoning in Event-Based Distributed Systems*. Studies in Computational Intelligence 347. Springer, 2011, págs. 47-70. ISBN: 978-3-642-19723-9 978-3-642-19724-6. DOI: 10.1007/978-3-642-19724-6_3. URL: http://dx.doi.org/10.1007/978-3-642-19724-6_3 (visitado 05-04-2014).
- [EN10] Opher Etzion y Peter Niblett. *Event Processing in Action*. Manning Publications Company, 2010. ISBN: 978-1-935182-21-4.
- [Eur] European Union. *INSPIRE (Infrastructure for spatial information in Europe) project*. URL: <http://inspire.ec.europa.eu/>.
- [Fel17] Jacob Feldman. *Business Decision Modeling with DMN and OpenRules*. Available at: <https://www.openrules.com/>. 2017. URL: <https://www.openrules.com/>.
- [Fra03] D. S. Frankel. «Model-Driven Architecture: Applying MDA to Enterprise Computing». En: (2003).
- [Fri03] Lana Friesen. «Targeting enforcement to improve compliance with environmental regulations». En: *Journal of Environmental Economics and Management* 46.1 (2003), págs. 72-85. ISSN: 0095-0696. DOI: 10.1016/S0095-0696(02)00033-5. URL: [http://dx.doi.org/10.1016/S0095-0696\(02\)00033-5](http://dx.doi.org/10.1016/S0095-0696(02)00033-5).
- [Fül12] Lajos Jenő Fülöp, Árpád Beszédes, Gabriella Tóth, Hunor Demeter, László Vidács y Lóránt Farkas. «Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics». En: *Balkan Conference in Informatics, 2012, BCI '12, Novi Sad, Serbia, September 16-20, 2012*. Ed. por Mirjana Ivanovic, Zoran Budimac y Milos Radovanovic. ACM, 2012, págs. 26-31. DOI: 10.1145/2371316.2371323. URL: <https://doi.org/10.1145/2371316.2371323>.
- [Gon17] Marcelo Piñeiro Gonzalo Antúnez. «Sistema de Monitoreo Ambiental (SIMONA)». Grado. Montevideo, Uruguay: UDELAR, jun. de 2017.
- [Goo07] Michael F. Goodchild. «Citizens as sensors: the world of volunteered geography». En: *GeoJournal* 69.4 (2007), págs. 211-221. ISSN: 1572-9893. DOI: 10.1007/s10708-007-9111-y. URL: <http://dx.doi.org/10.1007/s10708-007-9111-y>.

- [HB11] Tom Heath y Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. 1st. Morgan & Claypool, 2011. ISBN: 978-1-6084-543-10. URL: <http://linkeddatabook.com/>.
- [Her16] Federico Herrera, Laura González y Daniel Calegari. «Complex event processing with geospatial support for monitoring and controlling compliance with environmental regulations». En: *XLII Latin American Computing Conference, CLEI 2016, Valparaíso, Chile, October 10-14, 2016*. IEEE, 2016, págs. 1-10. DOI: 10.1109/CLEI.2016.7833414. URL: <https://doi.org/10.1109/CLEI.2016.7833414>.
- [Her17] Federico Herrera, Laura González, Daniel Calegari y Bruno Rienzi. «Compliance with Environmental Regulations through Complex Geo-Event Processing». En: *CLEI Electron. J.* 20.2 (2017). DOI: 10.19153/cleiej.20.2.2. URL: <https://doi.org/10.19153/cleiej.20.2.2>.
- [IMPa] IMPO. *Decreto N° 152/013*. URL: <http://www.impo.com.uy/bases/decretos/152-2013>.
- [IMPb] IMPO. *LEY DE PROTECCION DEL MEDIO AMBIENTE*. URL: <http://www.impo.com.uy/bases/leyes/17283-2000>.
- [ISO15] ISO. *ISO 9000: Sistemas de Gestión de Calidad - Fundamentos y Vocabulario*. Último acceso: Octubre 2023. Organización Internacional de Normalización (ISO), 2015. URL: <https://www.iso.org/standard/45481.html>.
- [ISO22] ISO. *Normativa Medioambiental ISO 14000*. Último acceso: Diciembre 2022. Organización Internacional de Normalización (ISO), 2022. URL: <http://www.iso.org/iso/iso14000>.
- [Kal11] Evangelos Kalampokis, Efthimios Tambouris y Konstantinos Tarabanis. «Electronic Government: 10th IFIP WG 8.5 International Conference, EGOV 2011, Delft, The Netherlands, August 28 – September 2, 2011. Proceedings». En: ed. por Marijn Janssen, Hans J. Scholl, Maria A. Wimmer y Yao-hua Tan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. Cap. Open Government Data: A Stage Model, págs. 235-246. ISBN: 978-3-642-22878-0. DOI: 10.1007/978-3-642-22878-0_20.
- [Kar23] Alex Karman. *The Decision Table Template For Geospatial Business Rules*. Available at: <https://www.rev-mac.com/>. 2023. URL: <https://www.rev-mac.com/>.
- [Ken02] S. Kent. «Model Driven Engineering». En: *Integrated Formal Methods. IFM 2002. Lecture Notes in Computer Science*. Ed. por M. Butler, L. Petre y K. Sere. Vol. 2335. Berlin, Heidelberg: Springer, 2002. DOI: https://doi.org/10.1007/3-540-47884-1_16.
- [Kha21] Behnam Khazael, H. Malazi y S. Clarke. «Complex Event Processing in Smart City Monitoring Applications». En: *IEEE Access* (2021). URL: <https://dx.doi.org/10.1109/access.2021.3119975>.

- [Kis07] Alexandre Kiss. *Guide to International Environmental Law*. Library of Congress Cataloging-in-Publication Data, 2007. URL: https://ds.amu.edu.et/xmlui/bitstream/handle/123456789/9110/Guide_to_International_Environmental_Law.pdf?sequence=1&isAllowed=y.
- [Knu13] David Knuplesch, Manfred Reichert, Rüdiger Pryss, Walid Fdhila y Stefanie Rinderle-Ma. «Ensuring compliance of distributed and collaborative workflows». En: *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. 2013, págs. 133-142. DOI: 10.4108/icst.collaboratecom.2013.254095.
- [Kru10] Ernst Kruijff, Erick Mendez, Eduardo Veas y Thomas Gruenewald. «On-site monitoring of environmental processes using mobile augmented reality (HYDROSYS)». En: *Proceedings of the Environmental Information Systems and Services—Infrastructures and Platforms Workshop (ENVIP 2010)*. Citeseer. 2010, págs. 6-8.
- [Küh08] Thomas Kühne. *Model-driven software development: Technology, engineering, management*. John Wiley & Sons, 2008.
- [LC20] Tengfu Liu y Lingyao Cheng. «Research on Key Information Processing Technology of Environmental Law Enforcement Based on Big Data». En: *Journal of Physics: Conference Series* 1648.3 (oct. de 2020), pág. 032055. DOI: 10.1088/1742-6596/1648/3/032055. URL: <https://dx.doi.org/10.1088/1742-6596/1648/3/032055>.
- [Lei09] James K. Lein. «Implementing remote sensing strategies to support environmental compliance assessment: A neural network application». En: *Environmental Science and Policy* 12.7 (2009), págs. 948-958. ISSN: 1462-9011. DOI: 10.1016/j.envsci.2009.08.001. URL: <http://dx.doi.org/10.1016/j.envsci.2009.08.001>.
- [Luc02] David Luckham. «The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems». En: *Addison-Wesley* (2002).
- [Moh17] Mohsen Mohammadi. «SERVICE ORIENTED ARCHITECTURE : A REVIEW AND EVALUATION OF REFERENCE MODELS». En: 2017.
- [Obj14] Object Management Group. *Query/View/Transformation Relations (QVT), Version 1.4*. 2014.
- [Obj17a] Object Management Group. *MetaObject Facility (MOF), Version 2.5.1*. 2017.
- [Obj17b] Object Management Group. *OMG Unified Modeling Language Specification (OMG UML), Version 2.5.1*. 2017.
- [Oec] Oecd.org. URL: <http://www.oecd.org/env/outreach/37867511.pdf>.
- [OGCa] OGC. *OpenGIS Web Services Architecture Description*. URL: http://portal.opengeospatial.org/files/?artifact%7B%5C_%7Ddid=13140 (visitado 19-11-2016).

- [OGCb] OGC. *Simple Feature Access - Part 2: SQL Option* | OGC. URL: <http://www.opengeospatial.org/standards/sfs> (visitado 29-04-2016).
- [Pap11] M.P. Papazoglou. «Making Business Processes Compliant to Standards and Regulations». En: *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*. Ago. de 2011, págs. 3-13. DOI: 10.1109/EDOC.2011.37.
- [Res09] B. Resch, M. Mittlboeck, F. Girardin, R. Britter y C. Ratti. «Real-Time Geo-awareness Sensor Data Integration for Environmental Monitoring in the City». En: *2009 International Conference on Advanced Geographic Information Systems Web Services*. 2009, págs. 92-97. DOI: 10.1109/GEOWS.2009.31. URL: <http://dx.doi.org/10.1109/GEOWS.2009.31>.
- [RW12] Manfred Reichert y Barbara Weber. «Business Process Compliance». en. En: *Enabling Flexibility in Process-Aware Information Systems*. Springer Berlin Heidelberg, ene. de 2012, págs. 297-320. ISBN: 978-3-642-30408-8 978-3-642-30409-5. (Visitado 23-04-2014).
- [Sel03] Bran Selic. «The Pragmatics of Model-Driven Development». En: *IEEE Software* 20.5 (2003), págs. 19-25.
- [Sem14] Franziska Semmelrodt, David Knuplesch y Manfred Reichert. «Enterprise, Business-Process and Information Systems Modeling: 15th International Conference, BPMDS 2014, 19th International Conference, EMMSAD 2014, Held at CAiSE 2014, Thessaloniki, Greece, June 16-17, 2014. Proceedings». En: ed. por Ilia Bider, Khaled Gaaloul, John Krogstie, Selmin Nurcan, Henderik A. Proper, Rainer Schmidt y Pnina Soffer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, págs. 48-63. ISBN: 978-3-662-43745-2. URL: http://dx.doi.org/10.1007/978-3-662-43745-2_4.
- [Soc10] Event Processing Technical Society. *Event processing glossary - version 2.0*. 2010. URL: http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf.
- [Ste08] Dave Steinberg, Frank Budinsky, Ed Merks y Marko Paternostro. *EMF: Eclipse modeling framework*. Pearson Education, 2008.
- [Tur11] Oktay Turetken, Amal Elgammal, Willem-Jan van den Heuvel y Mike Papazoglou. «Enforcing Compliance on Business Processes through the use of Patterns». En: (2011). URL: <http://aisel.aisnet.org/ecis2011/5/> (visitado 22-10-2013).
- [Urua] Parlamento del Uruguay. 15.239. URL: <https://www.impo.com.uy/bases/decretos-ley/15239-1981> (visitado 14-11-2021).
- [Urub] Parlamento del Uruguay. 17.283. URL: <https://www.impo.com.uy/bases/leyes/17283-2000> (visitado 14-11-2021).
- [Uruc] Parlamento del Uruguay. *Constitución de la República*. URL: <https://parlamento.gub.uy/documentosyleyes/constitucion> (visitado 14-11-2021).

- [uta] utas.edu.au. *LEGISLATION, REGULATORY MODELS AND APPROACHES TO COMPLIANCE AND ENFORCEMENT*. URL: http://www.utas.edu.au/__data/assets/pdf_file/0006/278007/Briefing_Paper_6_-_Laws_Regulation_Enforcement.pdf.
- [W R10] K. Mani Chandy y W. Roy Schulte. *Event Processing: Designing IT Systems for Agile Companies*. McGraw-Hill, 2010. ISBN: 9780071633505.
- [Wan20] Jonas Wanner, Christopher Wissuchek y Christian Janiesch. «Machine Learning and Complex Event Processing. A Review of Real-time Data Analytics for the Industrial Internet of Things». En: *EMISA 15.1* (2020). DOI: 10.18417/EMISA.15.1. URL: <https://dx.doi.org/10.18417/EMISA.15.1>.
- [Wel19] Joshua J. Wells, C. Parr y Stephen J. Yerka. «Automated Environmental Compliance Monitoring with IoT and Open Government Data». En: *arXiv preprint arXiv:2010.11945* (2019).
- [WF15] Felix Wortmann y Kristina Flüchter. «Internet of Things». En: Berlin, Heidelberg: Springer, 2015. DOI: 10.1007/s12599-015-0383-3. URL: <https://dx.doi.org/10.1007/s12599-015-0383-3>.
- [Zha18] Xingjian Zhang, Junjie Zhao y Yann LeCun. «Deep learning for event-driven stock prediction». En: *Expert Systems with Applications* 41.18 (2018), págs. 8132-8141.