



## **Universidad de las Ciencias Informáticas**

### **“Facultad 8”**

#### **“Desarrollo de un módulo para la validación de modelos estadísticos en el Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI).”**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.**



#### **Autores:**

**Oswaldo Ernesto Stable Vilches.  
Yandris Mata Cabrera.**

#### **Tutores:**

**Ing. Omar Ahmed García Pérez.  
Ing. Armando Robert Lobo.**

**Ciudad de la Habana, junio 2010**

**Frase:**



*“Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes.”*

## **Declaración de Autoría:**

Declaramos que somos los únicos autores del trabajo titulado: **“Desarrollo de un módulo para la validación de modelos estadísticos en el Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI)”**, y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio de 2010.

**Yandris Mata Cabrera**

**Oswaldo Ernesto Stable Vilches**

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Autor

**Ing. Omar Ahmed García Pérez**

**Ing. Armando Robert Lobo**

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

### Agradecimientos:

*Les agradezco a todas las personas que de una forma u otra han ayudado a realizar este importante objetivo, a todos aquellos que me han brindado su mano, a quienes me han enseñado, educado, instruido y mostrado siempre el camino correcto.*

*Agradecerle a mis maestras y profesores, a mis compañeros y amigos en especial a los amigos de la Universidad que siempre han estado presente, a mi compañero de tesis Yandris, Yusdel, Aldo, Giraldo, Aramis, el Yíyo, Elisabeth, Yeni, Cindy y a los charangueros por mantenernos unidos desde el principio y a mis tutores Omar y Lobo por ayudarnos y guiarnos en todo momento.*

**Oswaldo**

*A todos los maestros y profesores que he tenido a lo largo de mi trayectoria como estudiante ya que todos han contribuido a mi formación como profesional y como persona.*

*A todos los integrantes del centro DATEC que lejos de comportarse como simples compañeros de trabajo han sido como unos verdaderos amigos para mí.*

*A los tutores de este trabajo por toda la ayuda y el apoyo que nos han brindado durante el desarrollo del mismo y por haberse comportado realmente como grandes amigos.*

*A todos mis familiares por toda la ayuda y el amor que siempre me han dado, en especial a mis tíos Armando, Rosi, Fe, Cachita, Chicho, Esperanza, Clemente, Onelia, Vanque, Tola, Pedrito por siempre haberse comportado como unos padres para mí.*

*A mi compañero de tesis por todo el interés y la dedicación que puso en el desarrollo de este trabajo.*

*A todos mis amigos por toda la ayuda y los consejos que siempre me han dado contribuyendo de una forma u otra a hacerme mejor persona.*

*A este país y a la Revolución Cubana que sin ellos nada de esto fuese posible.*

**Yandris**

## Dedicatoria:

*A mi hermana Yalini y a mi hermano Jorge Luis por ser los mejores hermanos del mundo.*

*A mi abuela Eudisia, que aunque ya no esté presente, siempre la llevo en mi corazón como una de las cosas más hermosas que me ha pasado en la vida.*

*A mi abuela Aracelis, por todo el amor, comprensión y cariño que me ha dado toda la vida. Muchas gracias por ser tan especial.*

*A mi sobrinita Yoselin por llegar a este mundo y llenarnos a todos de amor y alegría.*

*A mi compañera Liasmeli por llegar a mi vida y hacerme la persona más feliz del mundo.*

*A todos mis familiares, amigos y vecinos por toda la confianza que siempre han depositado en mí.*

*Dedico este trabajo de forma muy especial a mis padres, Xiomara y Jorge, por haberme dado la vida y por todo el amor, el apoyo, la confianza y el esfuerzo que han dedicado para que este sueño se hiciese realidad.*

**Yandris**

*Le dedico este trabajo que conforma el punto cumbre de una larga trayectoria como estudiante a mi abuelo por ser el ángel de mi guarda, mi abuela por vivir para mí, mi madre por ser mi inspiración, mi padrastro por ser mi padre, mi tía Maite por ser mi ídolo, mi tía Toti por ser mi ayuda, a la UCI y a la Revolución por quedarme aún en deuda.*

**Oswaldo**

## **Resumen:**

La informatización como un proceso indetenible contribuye al desarrollo de toda sociedad. La automatización de procesos haciéndolos más eficaces y seguros deja constancia de ello. Todas las ramas de la sociedad actual se ven afectadas con este proceso y nuestro país está llamado a contribuir a ello para el beneficio de todos. Un pilar importante en ello lo constituye la Universidad de las Ciencias Informáticas (UCI), donde se desarrolla un producto denominado Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI) para la Oficina Nacional de Estadística (ONE). Este producto no cuenta con un módulo para la validación de datos en los modelos estadísticos aquí generados y este proceso es llevado a cabo dentro de la ONE de forma muy engorrosa.

A partir del análisis anterior se desarrolla la solución mediante del estudio de productos similares ya existentes, el proceso de selección de tecnologías, metodología, lenguajes y herramientas; la descripción de las características del sistema, el desarrollo del análisis, diseño, implementación y pruebas para la obtención del producto llamado “Módulo para la validación de modelos estadísticos en el Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI)”.

## Índice de Contenidos:

Índice de Figuras: .....	X
Índice de Tablas:.....	XI
Introducción: .....	1
Fundamentación Teórica.....	4
Introducción.....	4
<b>1.1 Métodos de validación de datos.....</b>	<b>4</b>
1.1.1 Cuadros de Validación .....	4
1.1.2 Funciones Matriciales .....	5
<b>1.2 Sistemas de validación de datos.....</b>	<b>5</b>
1.2.1 SPSS .....	5
1.2.2 MICROSET NT .....	6
1.2.3 SIGE 1.0 .....	6
<b>1.3 Herramientas y tecnologías a utilizar .....</b>	<b>7</b>
1.3.1 Metodología de desarrollo de software.....	7
1.3.1.1 RUP .....	7
1.3.1.2 Metodología OpenUp .....	8
1.3.2 Lenguaje de Modelado.....	9
1.3.3 Herramienta CASE para el modelado .....	9
1.3.3.1 Rational Rose .....	9
1.3.3.2 Visual Paradigm .....	9
1.3.4 Tecnologías Web.....	10
1.3.4.1 JavaScript.....	10
1.3.4.2 Extensible Markup Language (XML) .....	10
1.3.4.3 XML Schema .....	11
1.3.4.4 Extensible Stylesheet Language for Transformations (XSLT) .....	11
1.3.4.5 JavaScript Object Notation (JSON).....	11
1.3.4.6 JavaScript Asíncrono y XML (AJAX).....	12
1.3.4.7 Procesador de Hipertexto (PHP) .....	12
1.3.4.8 Lenguaje Etiquetado de Validación para Modelos Estadísticos (XVAME) .....	13
1.3.5 Framework.....	13
1.3.5.1 Symfony .....	14

1.3.5.2 ExtJS.....	15
1.3.6 Sistema Gestor de Base de Datos (SGBD).....	15
1.3.6.1 MySQL.....	16
1.3.6.2 PostgreSQL.....	16
1.3.7 Servidor Web.....	17
1.3.7.1 ZOPE.....	17
1.3.7.2 Apache.....	17
1.3.8 Entorno de Desarrollo Integrado (IDE).....	18
1.3.8.1 Eclipse.....	18
1.3.8.2 NetBeans.....	18
<b>Conclusiones.....</b>	<b>19</b>
<b>Características, Análisis y Diseño del Sistema.....</b>	<b>20</b>
<b>Introducción.....</b>	<b>20</b>
<b>2.1 Flujo actual del proceso de validación de datos.....</b>	<b>20</b>
<b>2.2 Información que se maneja.....</b>	<b>20</b>
2.2.1 Información de un modelo.....	20
2.2.2 Información para la validación.....	21
<b>2.3 Propuesta del módulo.....</b>	<b>21</b>
<b>2.5 Especificación de requerimientos del software.....</b>	<b>22</b>
2.5.1 Requisitos funcionales.....	22
<b>2.6 Tabla 1 Descripción de los actores del sistema.....</b>	<b>23</b>
<b>2.7 Descripción de los casos de uso.....</b>	<b>24</b>
2.7.1 Tabla 2 Caso de uso extendido: Gestionar validaciones.....	24
2.7.2 Tabla 3 Caso de uso: Validar Modelos.....	31
<b>2.8 Diagrama de casos de uso del sistema.....</b>	<b>33</b>
<b>2.9 Diagrama de clases del análisis.....</b>	<b>34</b>
<b>2.10 Diagramas de secuencia.....</b>	<b>35</b>
<b>2.11 Modelo del diseño.....</b>	<b>39</b>
2.11.1 Patrones de diseño.....	39
2.11.2 Diagrama de clases.....	40
<b>2.12 Interfaz de Usuario.....</b>	<b>50</b>
<b>Conclusiones.....</b>	<b>52</b>
<b>Implementación y Prueba.....</b>	<b>53</b>

<b>Introducción.....</b>	<b>53</b>
<b>3.1 Modelo de implementación.....</b>	<b>53</b>
<b>3.2 Diagrama de Componentes .....</b>	<b>53</b>
<b>3.3 Modelo de despliegue .....</b>	<b>56</b>
<b>3.4 Pruebas .....</b>	<b>57</b>
<b>Conclusiones.....</b>	<b>63</b>
<b>Conclusiones Generales: .....</b>	<b>64</b>
<b>Recomendaciones: .....</b>	<b>65</b>
<b>Bibliografía:.....</b>	<b>66</b>
<b>Glosario de Términos: .....</b>	<b>67</b>

## Índice de Figuras:

Fig. 1 Diagrama de casos de uso del sistema .....	33
Fig. 2 Diagrama de clases del análisis: Validar Modelos .....	34
Fig. 3 Diagrama de clases del análisis: Gestionar Validaciones .....	35
Fig. 4 Diagrama de secuencia: Validar Modelos.....	36
Fig. 5 Diagrama de secuencia: Gestionar Validaciones, sección "Adicionar Regla" .....	37
Fig. 6 Diagrama de secuencia: Gestionar Validaciones, sección "Modificar Regla" .....	38
Fig. 7 Diagrama de secuencia: Gestionar Validaciones, sección "Eliminar Regla" .....	39
Fig. 8 Clase Main .....	40
Fig. 9 Clase WPArbol .....	41
Fig. 10 Clase WPTabPanel.....	41
Fig. 11 Clase WPRule .....	42
Fig. 12 Clase WPXVame.....	43
Fig. 13 Clase WPExpLogica .....	43
Fig. 14 Clase WPLiteral.....	44
Fig. 15 Clase WPEscaque .....	45
Fig. 16 Clase WPFuncion.....	46
Fig. 17 Clase WPRango .....	46
Fig. 18 Clase WPAritmetica .....	47
Fig. 19 Clase TWPValidateModel.....	48
Fig. 20 Clase WPValidationModels .....	48
Fig. 21 Clase WPInformation .....	49
Fig. 22 Clase UCValidateModel.....	49
Fig. 23 Interfaz de Usuario: "Generar Regla" .....	50
Fig. 24 Interfaz de Usuario: "Generar Regla de Implicación" .....	51
Fig. 25 Interfaz de Usuario: "Mostrar Código XVAME" .....	51
Fig. 26 Interfaz de Usuario: "Validar Modelos" .....	52
Fig. 27 Diagrama de Componentes .....	54
Fig. 28 Diagrama de componentes: Gestionar Validaciones.....	55
Fig. 29 Diagrama de componentes: Validar Modelos.....	55
Fig. 30 Diagrama de componentes: Core.....	56
Fig. 31 Modelo de Despliegue .....	57

## Índice de Tablas:

Tabla 1 Descripción de los actores del sistema .....	23
Tabla 2 Caso de uso extendido: Gestionar validaciones .....	24
Tabla 3 Caso de uso: Validar Modelos .....	31
Tabla 4 Secciones a probar del caso de uso Gestionar Validaciones .....	58
Tabla 5 Descripción de variables .....	58
Tabla 6 Matriz de datos: Adicionar regla de validación .....	59
Tabla 7 Matriz de datos: Editar regla de validación .....	60
Tabla 8 Matriz de datos: Eliminar regla de validación .....	61
Tabla 9 Secciones a probar en al caso de uso Validar Modelos.....	62
Tabla 10 Matriz de datos: Validar Modelos .....	62

## **Introducción:**

La sociedad en que vivimos trae consigo un vertiginoso desarrollo marcado especialmente por el continuo avance de las tecnologías, donde los medios informáticos han constituido un factor indispensable, por lo que las computadoras y sus software se han insertado creando una enorme dependencia, constituyendo una herramienta sumamente necesaria para la informatización de numerosas actividades y procesos que anteriormente se realizaban de forma manual. Gracias a ello, actualmente se logra una mayor seguridad calidad, agilidad y eficiencia en el manejo de datos.

Nuestro país también se ha hecho sentir en este proceso y para ello se han dedicado esfuerzos al desarrollo de la informática en numerosas instituciones y la Universidad de las Ciencias Informáticas constituye un paso importantísimo. Esta universidad cuenta con una estructura de producción muy novedosa ayudando con ello al desarrollo de productos informáticos destinados a un gran número de esferas de la sociedad, tales como: salud, educación, deporte, turismo, biotecnología, telecomunicaciones, estadística, entre otras.

Uno de los proyectos que se llevan a cabo en esta institución es el desarrollo de un software para la Oficina Nacional de Estadística (ONE), el cual incluye una serie de herramientas que ayudarán a una mayor eficacia en la toma de decisiones. Dicho producto es denominado Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI).

Este sistema no cuenta con módulo para la validación de los datos de los modelos estadísticos en este generados con el objetivo de mitigar posibles errores, afectando con ello la completa automatización del proceso. Debido a que dicho sistema en su función procesa gran cantidad de datos estadísticos, estos pueden verse afectados y producir resultados con un mayor margen de error, provocando errores en la elaboración de los modelos estadísticos, lo cual puede ocasionar la generación de resultados incorrectos y empobrecimiento de la calidad de la información; además de lo complicado y engorroso que se le hace este proceso a los trabajadores de la Oficina Nacional de Estadística (ONE) con el uso de herramientas que no cuentan con las suficientes facilidades para el desarrollo rápido de estas tareas y los inconvenientes que traería la capacitación del personal que realiza este proceso.

Por todo lo antes mencionado se toma la siguiente interrogante como el **problema científico**: ¿Cómo gestionar y realizar de forma exitosa la validación de los modelos estadísticos de PATDSI?

Para desarrollar este trabajo se toma como **objeto de estudio** el proceso de validación de datos.

El **campo de acción** de esta investigación se enmarca en los sistemas automatizados que desarrollan procesos de validación de datos en los modelos estadísticos.

Para darle solución a los problemas antes expuestos se define como **objetivo general**: Desarrollar un módulo para la validación de los modelos estadísticos de PATDSI.

Como **objetivos específicos** se plantean:

- Realizar un estudio del estado del arte de la validación de datos en los modelos estadísticos así como el análisis de las herramientas actualmente utilizadas para el desarrollo de este proceso en la ONE.
- Realizar un estudio de las herramientas, tecnologías, metodologías y lenguajes que se utilizarán.
- Realizar el análisis y diseño de un módulo para la validación de datos en los modelos de PATDSI.
- Realizar la implementación y prueba de un módulo para la validación de datos en los modelos de PATDSI.

Para guiar el desarrollo de la investigación se define la siguiente **idea a defender**: Con la implementación de un módulo para la validación de datos en los modelos estadísticos de PATDSI se logrará mejorar el trabajo de procesamiento de la información y se facilitará el trabajo de validación de datos a los trabajadores de la Oficina Nacional de Estadísticas (ONE).

Como parte de la investigación que se debe llevar a cabo se utilizaron para su desarrollo los siguientes métodos científicos:

El **Analítico-Sintético** mediante el cual se extrae la información más relevante acerca de las metodologías, tecnologías y herramientas posibles a ser utilizadas en el desarrollo de la aplicación, esto permite realizar una mejor selección de las mismas, sintetizando sus características, y analizando las potencialidades de cada una y el **Análisis histórico – lógico** para analizar la evolución y desarrollo de los sistemas de gestión estadística, los principales conceptos que giran alrededor de los mismos, los sistemas existentes sobre ese tema y además para estudiar el proceso de validación de modelos estadísticos.

Para dar cumplimiento a los objetivos, se proponen las siguientes tareas a desarrollar:

- Valorar el enfoque de autores acerca de las herramientas para la validación de datos en los modelos estadísticos.
- Estudiar la metodología de desarrollo de software OpenUP para el modelado del sistema.
- Seleccionar las herramientas a usar en el análisis, diseño, implementación y prueba de un

módulo para la validación de datos en los modelos estadísticos de PATDSI.

- Realizar el análisis y el diseño del módulo a desarrollar.
- Realizar la implementación y las pruebas del módulo.

El presente trabajo de diploma consta de los siguientes **capítulos**:

**Capítulo 1.-** Fundamentación Teórica: este capítulo contiene un estudio de sistemas y herramientas que basan su funcionamiento o parte de este en la validación de datos, se recopilan todos los aspectos tenidos en cuenta para la selección de las herramientas, metodologías y tecnologías que serán utilizadas para el desarrollo del módulo.

**Capítulo 2.-** Características, Análisis y Diseño del Sistema: en este capítulo se describe de forma detallada y enfocada en el objeto de estudio, cómo deberá funcionar el módulo a desarrollar y para ello las especificaciones de cómo debe ser su implementación.

**Capítulo 3.-** Implementación y Prueba: en este capítulo se describe la implementación del módulo, así como las garantías de su funcionamiento de acuerdo con lo previsto.

# Capítulo 1

## Fundamentación Teórica

### Introducción

Este capítulo contiene la información acerca del estudio del estado del arte del trabajo, así como el análisis de las herramientas, metodologías y tecnologías posibles a utilizar para desarrollar el sistema y la definición de cuáles serán las seleccionadas para lograr tal propósito.

### 1.1 Métodos de validación de datos

Para el sistema que se desarrollará con el fin de la validación de datos en los modelos estadísticos, se lleva a cabo el análisis de dos métodos de validación de datos.

#### 1.1.1 Cuadros de Validación

Los cuadros de validación constituyen fórmulas matemáticas, que integran toda una serie de aspectos para la validación de datos contenidos en los modelos estadísticos, los cuales están identificados por un número y un subnúmero, además de presentar celdas referenciadas por sus correspondientes filas y columnas en las que se almacena la información a validar.

Los cuadros se clasifican como:

Cuadros Horizontales y Verticales: En este tipo de cuadros se consideran aquellos en que inicialmente se define un rango, de filas si el cuadro se aplica a determinadas columnas y de columnas si se aplica a determinadas filas.

Dentro de estos se tienen cuatro casos: Cuadros Normales, Bloqueados, Decimales y Condicionales.

Cuadros de Escaques<sup>1</sup>: Estos cuadros se definen a nivel de escaques dentro de la matriz de filas y columnas del modelo a procesar. Se considera por definición como escaque a un par de números separados por dos puntos (:), donde el primero representa una fila y el segundo una columna. Dentro de estos cuadros se encuentran dos casos: Cuadros normales y Cuadros condicionales.

Cuadros de filas en página: En este caso se definirán las filas que pertenecen a cada página según la definición del modelo, principalmente para aquellos que no son de talón abierto.

---

<sup>1</sup> Escaques: un par de números separados por dos puntos (:), donde el primero representa una fila y el segundo una columna.

Filas que deben existir necesariamente: Este cuadro se aplicará a aquellas filas del modelo que tienen que reportar valor obligatoriamente, o sea, debe existir la fila con al menos una de sus columnas con un valor mayor que cero.

Cuadros entre Variantes<sup>2</sup>: Estos cuadros son los que permiten comparar diferentes variantes para el caso de los modelos que las tengan.

Cuadros Inter-Modelo: En este tipo de cuadros se chequearán determinados cálculos entre el modelo base y otro u otros modelos.

Cuadros contra período anterior: En este paso se definirán las filas y columnas del modelo que serán chequeadas contra el mes, trimestre o semestre anterior.

### 1.1.2 Funciones Matriciales

Las funciones matriciales permiten efectuar cálculos muy interesantes y proporcionan en la hoja de cálculo una enorme potencia, permitiendo una mayor expansión para la comparación entre los datos ya que abarcan cualquier cantidad de celdas. Estas funciones son muy utilizadas en Microsoft Excel con el fin de validar datos mediante sumatorias, conteo y verificaciones. Sus funcionalidades están basadas en los diferentes tipos de cálculos proporcionados a través de las variables que representan pares ordenados (fila, columna).

## 1.2 Sistemas de validación de datos

Los sistemas de validación de datos surgen por la necesidad de verificar y a la vez revisar la concordancia y el correcto manejo de datos de acuerdo con las funcionalidades y finalidades de estos, utilizados básicamente en los trabajos estadísticos.

Actualmente existen una serie de sistemas informáticos que ayudan a la validación de estos, como son:

### 1.2.1 SPSS

El Statistical Package for the Social Sciences (SPSS) es un programa estadístico informático muy usado en las ciencias sociales y las empresas de investigación de mercado. El mismo cuenta con un módulo de validación de datos que permite al usuario realizar revisiones lógicas de la información y obtener reportes de los valores considerados extraños. Es similar al uso de sintaxis o scripts para realizar revisiones de los ficheros. De la misma forma que estos mecanismos, el proceso de validación es posterior a la digitalización de los datos.(1)

---

<sup>2</sup> Variante: término para diferenciar entre dos modelos con iguales vistas pero destinados a distintos marcos de una institución.

Presenta versiones para Windows y Linux pero siempre comportándose como software propietario, además de no cumplir con los requisitos requeridos por la ONE para el procesamiento estadístico en lo que a gestión se refiere.

### 1.2.2 MICROSET NT

El MICROSET NT fue desarrollado en los años 70 por informáticos de la Oficina Nacional de Estadística (ONE), es un sistema que permite procesar modelos de información sin que sea necesario realizar una programación específica para cada uno y que a la vez brinda grandes posibilidades tanto en la validación de los datos primarios, como en la edición de tablas, de acuerdo con las características particulares de cada uno.

Este sistema surge con el objetivo fundamental de lograr un sistema general e integral para el procesamiento de datos y la edición de tablas, facilitando mejoras en el procesamiento de la información de las estadísticas económicas de la ONE.

Este sistema consta de un proceso de validación de datos que se lleva a cabo a partir de un mecanismo denominado cuadros de validación, definidos en los parámetros del modelo solicitado, que aunque ayuda en la automatización de la validación de los datos en los modelos estadísticos no cuenta con las facilidades suficientes, debido a lo engorroso y complicado que llega a ser la implementación de estos cuadros.

### 1.2.3 SIGE 1.0

SIGE 1.0 es un sistema que automatiza los principales procesos que se desarrollan en la ONE y para ello se sustenta en las funcionalidades de cuatro módulos fundamentales: Módulo Generador de Modelos (SIGE-MGM), Módulo de Entrada de Datos (SIGE-MED), Módulo de Registros y Clasificadores (SIGE-MRC) y Módulo Generador de Reportes (SIGE-MGR).

En el MGM se lleva a cabo la validación de los modelos estadísticos aquí generados, utilizando para ello una herramienta que basa su funcionamiento básicamente en el mismo estilo de cuadros estadísticos originados en MICROSET manteniendo con ello las mismas complicaciones, además de desarrollarse sobre tecnologías propietarias con las necesidades que ello implica para su funcionamiento, debido a que SIGE 1.0 es una aplicación de escritorio que entre los requisitos de instalación exige Windows 98 o superior como sistema operativo para el cliente y Windows Server 2003 o superior como sistema operativo para el servidor.

### 1.3 Herramientas y tecnologías a utilizar

Para el desarrollo de este módulo se deben de tener en cuenta una serie de tecnologías y software imprescindibles para su realización. Para llevar a cabo esta selección se deben estudiar cada una de las existentes analizando sus características y ventajas además de mantener las políticas de desarrollo del centro y el camino hacia la soberanía tecnológica mediante la utilización de software libre.

#### 1.3.1 Metodología de desarrollo de software

El desarrollo de software resulta verdaderamente complicado. El mismo engloba una serie de bases, herramientas y métodos que necesitan imperativamente de una metodología, con el objetivo de encausar a cada uno de estos engranajes en lo que sería un proceso coordinado y acorde al fin que se persigue.

Hoy en día existen diferentes propuestas metodológicas con el fin de alcanzar una mayor planificación, mejor costo y ganancia de tiempo, dentro de estas se encuentran:

##### 1.3.1.1 RUP

Proceso Unificado de Desarrollo (RUP), el cual constituye una metodología pesada, dirigida a grandes proyectos, está centrado en la arquitectura, guiado por casos de uso y es iterativo e incremental.

Define como sus principales elementos:

- Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto con un equipo.
- Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de Actividades (“cuándo”): Secuencia de actividades realizadas por los trabajadores y que produce un resultado de valor observable.

Las actividades son agrupadas en grupos lógicos, donde se definen 9 flujos de trabajo fundamentales, de los cuales los 6 primeros son considerados flujos de ingeniería y los 3 restantes como flujos de apoyo. Estos flujos son: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Configuración y Administración de Cambios, Administración de Proyecto y Entorno.

Además, RUP define cuatro fases importantes en el desarrollo, las cuales son: Conceptualización (Concepción o Inicio), la cual describe el negocio y delimita el proyecto, se definen sus alcances con la identificación de los casos de uso del sistema; Elaboración, donde se define la arquitectura que tendrá el

sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen; Construcción, donde se obtiene un producto documentado y listo para su utilización, y por último la fase de Transición, en la cual el release queda listo para su instalación en condiciones reales.(2)

Esta metodología es colocada dentro de las mejores. Es considerada una metodología robusta y como tal basa su éxito en mantener una documentación ordenada a lo largo de todo el proceso de desarrollo, lo cual no ayuda en el desarrollo del presente proyecto debido a que es demasiado extensible en cuanto a documentación y tiempo requerido para el control de cambios. Esto la hace estar más dirigida a grandes proyectos de desarrollo de software.

### 1.3.1.2 Metodología OpenUp

OpenUp toma las mejores prácticas de RUP. Busca cubrir el mayor número de necesidades para los proyectos de desarrollo en un cierto plazo.

Es un proceso iterativo para un desarrollo de software que es: mínimo por solo incluir el contenido fundamental del proceso, completo pues manifiesta un proceso entero para desarrollar un sistema y es extensible ya que puede ser utilizado como base para agregar o adaptar más procesos.

Los beneficios de esta metodología se destacan de la siguiente forma: permite disminuir las probabilidades de fracaso en los proyectos pequeños, las detecciones tempranas de errores, evita la elaboración de documentación innecesaria y permite un enfoque centrado en el cliente y con iteraciones cortas.

Tiene como principios fundamentales la colaboración en busca de alinear intereses y compartir conocimientos, balancear las necesidades con el fin de maximizar las necesidades de los stakeholder, así como estar centrado en la arquitectura y llevar a cabo un desarrollo iterativo.

Por lo antes expuesto se considerada esta metodología como las más factible y ajustable al equipo de trabajo para el desarrollo de este proyecto, debido a que está dirigida a proyectos pequeños y logra: colocar a los individuos y su interacción, por encima de los procesos y las herramientas, el software que funciona, por encima de la documentación exhaustiva, la colaboración con el cliente, por encima de la negociación contractual y la respuesta al cambio, por encima del seguimiento de un plan, además de la versatilidad en el equipo de desarrollo y su capacidad de personalización y adaptación a nuevos requerimientos.(3)

### 1.3.2 Lenguaje de Modelado

Un lenguaje de modelado constituye la forma común para establecer una coordinada comunicación entre los miembros de un equipo de desarrollo de software. Este lenguaje sirve de herramienta para desarrollar una representación simplificada de la realidad a través de abstracciones que se plasman en notaciones gráficas y sirven de apoyo al análisis de problemas.

El Lenguaje Unificado de Modelado (UML) es considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Con el objetivo de cumplir sus funciones de ayuda a la construcción y documentación de sistemas ofrece una serie de ventajas tales como: un estándar para describir un “plano” del sistema (modelo), aspectos conceptuales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.(2)

UML no especifica qué metodología o proceso de desarrollo de software usar; cuenta con una serie de diagramas que muestran diferentes aspectos de las entidades representadas como son: los diagramas de estructura, de comportamiento y de interacción y constituye por ende el que será utilizado para el modelado del módulo a desarrollar.

### 1.3.3 Herramienta CASE para el modelado

Las herramientas CASE (Computer Aided Software Engineering) tienen el objetivo de hacer más eficiente en cuestión de tiempo y costo el desarrollo de software. Se encuentran disponibles en todo el proceso de desarrollo sirviendo de soporte para el diseño, compilación, detección de errores, generación de parte del código del producto en cuestión así como hacer más fácil la documentación del mismo.

#### 1.3.3.1 Rational Rose

Rational Rose permite completar los flujos fundamentales de RUP y se ha convertido en una de las mejores opciones por la notación estándar que brinda para especificar, visualizar y construir productos software y sistemas, por lo que está en la avanzada en cuanto al desarrollo de UML. Pero aun con todos estos beneficios presenta desventajas significativas ya que para su utilización requiere de una licencia, de alta capacidad de procesamiento y de un alto costo para su adquisición.(4)

#### 1.3.3.2 Visual Paradigm

Visual Paradigm es una herramienta de modelado que cuenta con una distribución gratuita, además cuenta con una interfaz gráfica muy cómoda para el usuario ayudándoles al modelado con el uso de estándares UML.

Este software cuenta además con una serie de ventajas que lo hacen ser el elegido para el modelado del presente trabajo, como son: presencia de un entorno de creación de diagramas para UML 2.0; un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad; utiliza un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación; presenta capacidades de ingeniería directa e inversa; modelo y código que permanece sincronizado en todo el ciclo de desarrollo y disponibilidad de múltiples versiones para cada necesidad; es multiplataforma.

Se caracteriza además por ser un producto de calidad, soportar aplicaciones web y varios idiomas, generación de código para Java y exportación como HTML, fácil de instalar, actualizar y presenta compatibilidad entre ediciones.

Teniendo en cuenta de que se está en presencia de un software con distribuciones gratuitas, característica que lo coloca por encima de Rational Rose y que cuenta con un excelente entorno de trabajo, modelador UML y soporte para plugins y plantillas, se decide tomarlo como herramienta de modelado para el desarrollo del presente proyecto.(5)

### **1.3.4 Tecnologías Web**

#### **1.3.4.1 JavaScript**

JavaScript es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente. Es un lenguaje orientado a eventos. Cuando un usuario da click sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos, además es un lenguaje orientado a objetos. El modelo de objetos de este lenguaje está reducido y simplificado, pero incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

#### **1.3.4.2 Extensible Markup Language (XML)**

Extensible Markup Language o lenguaje de marcas extensible (XML) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes. Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

El XML es un metalenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de "extensible"). Es además especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (6)

### 1.3.4.3 XML Schema

XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio XML.

Los documentos esquema (usualmente con extensión .xsd de XML Schema Definition) se crearon como una alternativa a las DTD (Document Type Definition), superando sus puntos débiles y buscando nuevas capacidades a la hora de definir estructuras para documentos XML. La principal aportación de XML Schema es el gran número de tipos de datos que incorpora. De esta manera, aumenta las posibilidades y funcionalidades de las aplicaciones de procesamiento de datos, incluyendo tipos de datos complejos como fechas, números y strings.

Al restringir el contenido de los ficheros XML es posible intercambiar información entre aplicaciones con gran seguridad. Disminuye el trabajo de comprobar la estructura de los ficheros y los tipos de datos.

### 1.3.4.4 Extensible Stylesheet Language for Transformations (XSLT)

XSLT es un lenguaje creado por World Wide Web Consortium (W3C) que tiene como objetivo la transformación de documentos XML a otros formatos aplicando las reglas definidas en dicho lenguaje. XSLT utiliza sintaxis de XML por lo que se puede considerar un lenguaje creado o definido a partir de XML.

La principal ventaja de XSLT es la posibilidad de separar el contenido del formato, con los beneficios que ello conlleva pudiéndose representar la misma información en diferentes lugares con apariencia distinta dependiendo de la hoja de estilo aplicada para la transformación del documento XML.

Actualmente, XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

### 1.3.4.5 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) es un formato de intercambio de datos basado en objetos javascript que es ampliamente utilizado por la facilidad para codificar y decodificar y por su poco tamaño.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX, ya que al resultar más ligero, el intercambio de datos entre el cliente y el servidor se puede efectuar a una mayor velocidad.

### 1.3.4.6 JavaScript Asíncrono y XML (AJAX)

JavaScript Asíncrono y XML (AJAX) es una técnica de desarrollo web para crear aplicaciones interactivas.

Estas aplicaciones se crean combinando:

- Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.
- XML, XSLT para intercambiar y manipular datos.
- CSS para definir el aspecto del documento.
- JSON pueden ser usado como alternativa a XML/XSLT
- XMLHttpRequest para recuperar datos de forma asincrónica.
- Javascript como nexo de unión de todas estas tecnologías.

Estas aplicaciones se ejecutan sobre el navegador, es decir, en el lado del cliente. Durante esta ejecución se mantiene la comunicación asíncrona con el servidor en un segundo plano. Esto permite realizar cambios en las páginas sin la necesidad de recargarlas, lo que posibilita aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que cuando se necesite cargar datos nuevos se envía una petición al servidor web el cual devuelve estos datos y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

### 1.3.4.7 Procesador de Hipertexto (PHP)

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe

la página a través de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

Dentro de las ventajas que supone el uso de PHP destacan las siguientes: es de fácil aprendizaje, es un lenguaje muy rápido, soporta la programación orientada a objetos y puede ser utilizado tanto sobre Linux como Windows. Tiene capacidad de conexión con la mayoría de los manejadores de bases de datos tales como: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras. Posee documentación en su página oficial la cual incluye la descripción y ejemplos de cada una de sus funciones. Es libre y no requiere definición de tipos de variables ni manejo detallado del bajo nivel.(7)

### **1.3.4.8 Lenguaje Etiquetado de Validación para Modelos Estadísticos (XVAME)**

Lenguaje Etiquetado de Validación para Modelos Estadísticos (XVAME): Esta tecnología está basada en la utilización del lenguaje XML durante el proceso de definición de las reglas de validación que sirven de entrada a un mecanismo de validación. Dicho mecanismo es el encargado de aplicar las reglas de validación a los modelos estadísticos, en formato XML también, y devolver un resultado; este último es satisfactorio en caso de verificarse la regla.

El mecanismo de validación funciona como un compilador que utiliza un archivo XSD (esquema XML) para validar la estructura del XML contenedor de las reglas, funcionando como analizador léxico, sintáctico y semántico. Si las reglas están correctamente definidas se realiza una transformación de estas al lenguaje PHP haciendo uso del lenguaje XSLT siguiendo un conjunto reglas específicamente definidas para esa transformación en un archivo XSL. Finalmente, el código PHP generado se evalúa para determinar si se cumplen o no las reglas de validación informando finalmente al usuario si alguna de estas no es verificada.

### **1.3.5 Framework**

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Los frameworks ayudan a la hora de realizar el mantenimiento del sistema gracias a la organización que se logra durante el desarrollo de la aplicación.

En general son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos.

### 1.3.5.1 Symfony

Symfony es uno de los frameworks más populares para PHP en la actualidad. Es un completo marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP5. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas unix (Unix, Linux, etc.) como en plataformas Windows. Además, utiliza al Modelo Vista Controlador como patrón de diseño web.

#### **Modelo Vista Controlador (MVC)**

El Modelo Vista Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

Dentro de las ventajas que supone el uso de este estilo arquitectónico se encuentran:

- Presenta soporte de vistas múltiples: dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- Adaptación al cambio: los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Como las bases de datos siguen una estructura relacional, y PHP5 y Symfony están orientados a objetos, es necesario acceder a la base de datos como si esta estuviera orientada a objetos. Para lograr esto, es necesaria la utilización del Mapeo Objeto-Relacional (ORM).

#### **Mapeo Objeto-Relacional (ORM)**

El Mapeo Objeto-Relacional es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientada a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Symfony utiliza por defecto al ORM Propel como motor generador de capa tipo ORM, el cual se encuentra integrado completamente en él. Propel es un proyecto de software libre el cual es una de las mejores capas de abstracción objetos/relacional disponibles en PHP5. (8)

### 1.3.5.2 ExtJS

ExtJS es una librería JavaScript ligera y de alto rendimiento compatible con la mayoría de los navegadores, que permite crear aplicaciones enriquecidas del lado del cliente usando tecnologías como AJAX, DHTML y DOM. Fue creado por Jack Slocum originalmente como extensión para Yahoo User Interface (YUI), en la actualidad puede usarse como extensión para las bibliotecas jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente.

Sus características principales son:

- Componentes de interfaz de usuario personalizables.
- Buenos diseños.
- Es intuitivo.
- API extensa y fácil de utilizar.
- Se distribuye bajo licencias Open Source y comerciales.
- Gran desempeño.(9)

### 1.3.6 Sistema Gestor de Base de Datos (SGBD)

Un Sistema Gestor de Base de Datos es un tipo de software que funciona como interfaz entre el usuario, la aplicación y la base de datos. Están compuestos por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Tiene como propósito el manejo sencillo de los datos y las entidades que los organizan con el fin de llegar más fácilmente a la información contenida dentro de la base de datos. Con la utilización del gestor de base de datos se logra: independencia de los datos y los programas de aplicación, minimización de la redundancia, integración y sincronización de las bases de datos, integridad, seguridad y protección de los datos, facilidad de manipulación de la información y control centralizado.

### 1.3.6.1 MySQL

MySQL tiene más de seis millones de instalaciones, es multihilo, multiusuario, rápido y bastante robusto y es propiedad de la empresa MySQL AB. Por un lado se ofrece bajo la GNU GPL (Licencia Pública General de GNU) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Es un SGBD muy utilizado en el mundo, y es muy popular por estar ampliamente ligado a PHP. Funciona sobre múltiples plataformas, incluyendo GNU/Linux, Mac OS X, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.

Se caracteriza por: tener verificación basada en el host y tráfico de contraseñas encriptado, ofrece un sistema de contraseñas y privilegios seguro, denotando cualidades de seguridad altamente confiables y es capaz de soportar gran cantidad de datos.

Pero aunque es favorable su uso por su destacada rapidez, el mismo no cuenta con una completa distribución de su código como la mayoría de los software libres debido a que no se encuentra desarrollado por una comunidad pública y el copyright del código está en poder de un autor individual.

### 1.3.6.2 PostgreSQL

PostgreSQL brinda las siguientes posibilidades:

- Instalación ilimitada: con PostgreSQL, nadie puede ser demandado por violar acuerdos de licencia, puesto que no hay un costo asociado a la licencia del software.
- Soporte: existe una gran comunidad de profesionales y empresas que ofrecen soporte a PostgreSQL, de la cual el Grupo Global de Desarrollo de PostgreSQL puede obtener beneficios y contribuir en su mejoramiento.
- Estabilidad y Confiabilidad Legendarias: es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- Extensible: el código fuente está disponible para todos sin costo. Si un equipo de trabajo necesita extender o personalizar PostgreSQL de alguna manera puede hacerlo con un mínimo esfuerzo y sin costos adicionales, lo cual es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden dicho gestor sistemáticamente.
- Multiplataforma: se encuentra disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.
- Herramientas gráficas de diseño y administración de bases de datos: existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer

diseño de bases de datos.

- Soporta funciones que pueden definirse para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido, además retornan filas donde las salidas pueden tratarse como un conjunto de valores retornados por una consulta.
- Soporta bloques de código que se ejecutan en el servidor, los cuales pueden ser escritos en diferentes lenguajes de programación con la potencia que presenta cada uno, desde sus operaciones básicas tales como la bifurcación y bucles hasta las complejidades de la programación orientada a objetos o la programación funcional. (10)

Indiscutiblemente se está en presencia de un excelente SGBD por lo que es considerado como el indicado para el presente trabajo, ya que este gestor es considerado como el SGBD software libre más avanzado del mundo; es un sistema de gestión de base de datos relacional, orientado a objetos y solo basa su costo en la capacidad personal de saber utilizarlo.

### 1.3.7 Servidor Web

Un servidor web es un software en un servidor que se encuentra en constante procesamiento al mantenerse en espera de solicitudes de ordenadores clientes a través de navegadores web y dándole repuesta a estas mediante páginas web que se muestran en dicho navegador. Las peticiones son realizadas con el uso del protocolo HTTP y respondidas con códigos HTML que el cliente es el encargado de descifrar en fuentes, colores y formas exhibidas en dichas páginas.

#### 1.3.7.1 ZOPE

Es un servidor de aplicaciones Web, pertenece a la gran familia de software libre, es desarrollado en Python. Entre las características que lo distinguen se encuentra su base de datos orientada a objetos denominada ZODB (Zope Object Database), la cual almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación difiere bastante de las bases de datos relacionales actuales. No obstante, Zope posee la ventaja de contar con múltiples conectores para las diferentes bases de datos relacionales ofreciendo sistemas básicos de conexión y consulta abstrayéndolos como objetos.

#### 1.3.7.2 Apache

El Servidor Apache también forma parte de los software libre. Su diseño le permite ser un servidor web potente y flexible, funcional en una amplia gama de plataformas y entornos y que permite ser adaptado a diferentes entornos y necesidades. Incentiva a la retroalimentación con los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos y es extensible debido a que al ser

modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Este servidor es funcional tanto para páginas estáticas como para páginas dinámicas a través de otras herramientas soportadas que facilitan la actualización de los contenidos usando bases de datos, ficheros u otras fuentes de información.

Aunque el servidor web primeramente mencionado constituye una buena opción a utilizar, Apache posee algunas ventajas que lo hacen realmente exclusivo para su selección en este proyecto, debido a su: estructuración en módulos, gran compatibilidad con el lenguaje de programación PHP y la experiencia acumulada sobre su uso tanto a nivel mundial como en la Universidad de las Ciencias Informáticas.

### **1.3.8 Entorno de Desarrollo Integrado (IDE)**

Un Entorno de Desarrollo Integrado es un software que consta de una serie de herramientas acopladas que tienen como fin fundamental la programación en uno o más de un lenguaje de programación.

#### **1.3.8.1 Eclipse**

Es un IDE de programación gratuito que lleva tiempo demostrando su hegemonía asociado comúnmente al lenguaje de programación Java. Constituye un almacén sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la instalación de los plugins adecuados. La arquitectura de plugins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces y ayuda en línea para librerías. Después de posicionarse como un software de desarrollo altamente competitivo, Eclipse está demostrando su calidad en el desarrollo de aplicaciones Web mediante el plugin para el lenguaje PHP. Dicho plugin permite a los desarrolladores en PHP un trabajo rápido, cómodo y sencillo pues tiene auto completamiento y colorea errores. Actualmente se encuentra disponible la versión 2.0 de las herramientas de desarrollo de PHP (PDT) de Eclipse. (5)

#### **1.3.8.2 NetBeans**

NetBeans un producto de código. Ayuda a la flexibilidad en el trabajo por sus posibilidades de modificación por parte de los desarrolladores. Ofrece además una serie de ventajas que lo hacen realmente codiciado como son: la creación de aplicaciones multiplataforma, permite que las aplicaciones creadas por el IDE se adhieran a los estándares de la industria, proporciona facilidades y garantías para la migración, facilidad de uso, cumplimiento de regulaciones, y flexibilidad entre plataformas.

Este IDE presenta además características adicionales y plugins de los paquetes de funciones que abarcan C/C++ y la Web. También maneja la complejidad de la Arquitectura Orientada a Servicio (SOA) y posee soporte para PHP5.

Teniendo en cuenta cada una de las condiciones que hacen que tanto NetBeans como el Eclipse posean verdaderas ventajas para la implementación de este proyecto se selecciona a NetBeans debido a que actualmente existe una versión con un aumento de facilidades para el completamiento de código y se tiene mayor experiencia por parte del equipo de trabajo.

### **Conclusiones**

Este capítulo constituye la base sobre la cual se construirá la solución del presente problema. El estudio de sistemas existentes en los cuales se desarrolla la validación de datos estadísticos así como los métodos utilizados en algunos de ellos permitieron capturar un cierto número de características y cualidades que no deben faltar en el módulo a desarrollar. También se realizó un estudio de las tecnologías, metodologías y herramientas que en la actualidad forman parte inseparable de la industria del software y se elaboró sobre ello una propuesta para la selección de las más adecuadas para utilizar en el presente proyecto.

# Capítulo 2

## Características, Análisis y Diseño del Sistema

### Introducción

En este capítulo se describe de forma más explícita la situación problemática, su entorno y objetivo de automatización. Se propone una solución al problema presentando la propuesta del módulo y la descripción de los artefactos generados durante el desarrollo de la ingeniería del software.

### 2.1 Flujo actual del proceso de validación de datos

El desarrollo de la validación de datos en los modelos estadísticos de la ONE fluye de la siguiente manera:

Primeramente se crea un modelo estadístico con cada uno de sus indicadores y aspectos definidos. Una vez creado el modelo se le introducen cada uno de los datos correspondientes de acuerdo con la institución a la que pertenece. Por último, se definen una serie de cuadros estadísticos para la validación de los datos del mismo con el objetivo de detectar los errores para que cada dato que conforma el modelo entre en correspondencia con las reglas solicitadas en dichos cuadros.

### 2.2 Información que se maneja

Para llevar a cabo el proceso de validación es necesaria la información proveniente de los modelos estadísticos así como la solicitada para la creación de las reglas de validación.

#### 2.2.1 Información de un modelo

- Identidad del modelo
- Número
- Subnúmero
- Clasificación de seguridad
- Periodicidad
- Variantes
- Centro informante

- Municipio
- Provincia
- Páginas:
  - Título
  - Indicadores
  - Aspectos
- Nota metodológica

### 2.2.2 Información para la validación

- Número del modelo
- Referencia de filas y columnas
- Constantes numéricas
- Operadores lógicos
- Operadores aritméticos
- Operadores relacionales
- Estructura condicional (implicación)

### 2.3 Propuesta del módulo

El módulo propuesto debe manejar la información correspondiente a las reglas de validación respecto a un modelo estadístico. Una vez creado el modelo estadístico con cada una de sus referencias definidas (aspectos e indicadores) el sistema brinda la posibilidad de acceder a las funcionalidades del módulo de validación para la creación de reglas que permitan la validación del modelo.

El sistema debe permitir generar las reglas de validación necesarias con la mayor facilidad posible. Debe brindar una interfaz lo suficientemente cómoda para garantizar la agilidad y eficiencia en el proceso de diseño de las reglas de validación.

El sistema debe permitir la ejecución de las reglas diseñadas sobre su modelo correspondiente, después de completar la información perteneciente a este, dando la respuesta a dicho proceso en el menor tiempo posible.

## 2.5 Especificación de requerimientos del software

### 2.5.1 Requisitos funcionales

#### RF 1.- Definir regla de validación

RF 1.1.- El sistema debe permitir crear regla de validación asociada a un modelo.

RF 1.1.2.- El sistema debe permitir crear regla de validación que admita operaciones matemáticas con rangos de datos de un modelo y constantes.

**Entradas:**

Operaciones con rangos (suma, conteo, multiplicación, división)

Rango vertical (Desde - Hasta por las filas y la columna)

Rango horizontal (Desde - Hasta por las columnas y la fila)

Escaque (fila y columna)

Constante (número)

RF 1.1.3.- El sistema debe permitir crear regla de validación que admita comparación.

**Entradas:**

Expresiones a comparar (expresiones aritméticas, operaciones con rangos, constantes, escaques)

Operados lógico (<, >, =, /, <=, >=)

RF 1.1.4.- El sistema debe permitir crear regla de validación que admita condicional.

**Entradas:**

Condición (expresión lógica)

Consecuencia (expresión lógica)

Operador de implicación (->)

RF 1.2.- El sistema debe permitir modificar la regla de validación previamente creada.

**Entrada:**

Datos de la nueva regla

RF 1.3.- El sistema debe permitir mostrar regla de validación.

**Salida:**

Regla de validación

RF 1.4.- El sistema debe permitir eliminar regla de validación.

**Entrada:**

Regla de validación (número)

**RF 2.- Validar Modelos**

RF 2.1.- El sistema debe permitir validar modelos de acuerdo con sus reglas de validación especificadas.

**Entrada:**

Modelo a validar

**Salida:**

Resultado de la validación

### 2.6 Tabla 1 Descripción de los actores del sistema

Actor	Descripción
Especialista de estadística	Responsable del diseño de un modelo que capte información estadística con un objetivo bien definido.
Validador	Responsable de validar los modelos que presentan toda su información con el objetivo de obtener una respuesta de la validación.

## 2.7 Descripción de los casos de uso

El módulo cuenta con dos casos de uso: uno correspondiente a la generación de modelos en PATDSI, en este caso Gestionar Validaciones y otro, Validar Modelos, en la entrada de datos a los modelos antes diseñados.

### 2.7.1 Tabla 2 Caso de uso extendido: Gestionar validaciones

<b>Caso de Uso:</b>	Gestionar validaciones
<b>Actores:</b>	Especialista de estadística
<b>Resumen:</b>	El caso de uso inicia cuando el especialista de estadística necesita generar una o varias reglas de validación para un modelo estadístico de acuerdo a diferentes criterios. Se obtiene el programa de validación de acuerdo con el criterio especificado terminando el caso de uso.
<b>Precondiciones:</b>	<ol style="list-style-type: none"> <li>1. Debe existir el modelo a validar.</li> <li>2. Deben estar definidos los indicadores y aspectos.</li> </ol>
<b>Referencias</b>	
<b>Prioridad</b>	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. El actor solicita definir validación del modelo.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema muestra una interfaz (ver figura 1) que brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Adicionar regla ver sección: "Adicionar regla de validación"</li> <li>• Editar regla ver sección: "Editar regla de validación"</li> <li>• Eliminar regla ver sección: "Eliminar regla de validación"</li> </ul> </li> </ol>

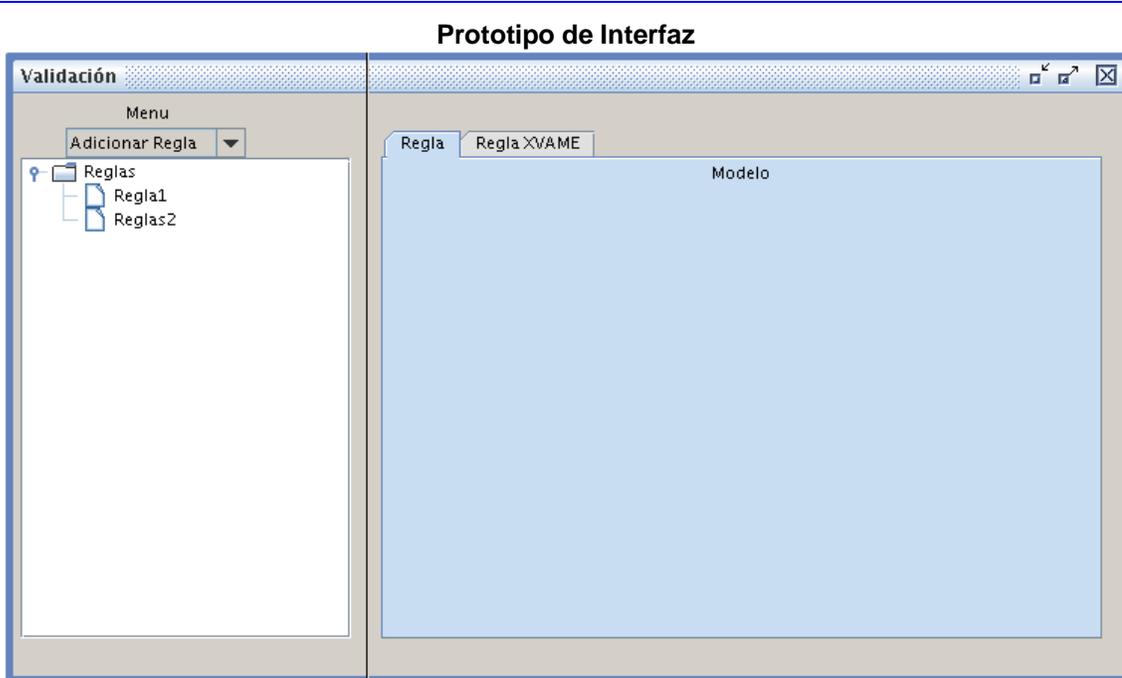


figura 1

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prototipo de Interfaz	
Poscondiciones	
Flujo Normal de Eventos	
Sección "Adicionar regla de validación"	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona adicionar regla.	2. El sistema brinda interfaz (ver figura 2) para introducir datos: <ul style="list-style-type: none"> <li>• Numero de la regla</li> <li>• Descripción</li> </ul>

	<ul style="list-style-type: none"> <li>• Mensaje de error</li> </ul> <p>3. Seleccionar la estructura de la regla (ver figura 3).</p>
<p>4. El actor introduce los datos:</p> <ul style="list-style-type: none"> <li>• Numero de la regla</li> <li>• Descripción</li> <li>• Mensaje de error.</li> <li>• Selecciona estructura principal de la regla.</li> </ul>	<p>5. El sistema comprueba que los datos obligatorios sean llenados, en caso contrario ir al <i>flujo alterno 5.1</i>.</p> <p>6. En caso de seleccionar :</p> <ul style="list-style-type: none"> <li>• Una expresión condicional obligatoriamente la condición y la consecuencia de la misma deben ser operaciones lógicas.</li> <li>• Una expresión de comparación puedes suceder entre operaciones aritméticas, funciones, constantes numéricas, escaques y todas las combinaciones posibles entre ellas.</li> </ul> <p>7. El sistema brinda según la opción seleccionada:</p> <ul style="list-style-type: none"> <li>• Interfaz para operación lógica (ver figura 3) seleccionar operador lógico y las expresiones de comparación.</li> <li>• Interfaz para operación aritmética (ver figura 4) seleccionar operador aritmético y los argumentos pueden ser: escaques, constantes numéricas o funciones.</li> <li>• Interfaz para constante numérica (ver figura 5) introducir constante.</li> <li>• Interfaz para escaque (ver figura 6) introducir fila y columna que referencie la celda.</li> </ul>

	<ul style="list-style-type: none"> <li>• Interfaz para funciones (ver figura 7)                      Seleccionar función e introducir rango horizontal o vertical (ver figura 8).</li> </ul>
<p>8. El actor selecciona opciones e introduce datos correspondientes:</p> <p>Para constante numérica:</p> <ul style="list-style-type: none"> <li>• Constante numérica.</li> </ul> <p>Para escaque:</p> <ul style="list-style-type: none"> <li>• Fila</li> <li>• Columna</li> </ul> <p>Para función:</p> <ul style="list-style-type: none"> <li>• Selecciona función</li> <li>• En caso de rango horizontal:                      Desde - Hasta por las columnas y fila.</li> <li>• En caso de rango vertical:                      Desde - Hasta por las filas y columna.</li> </ul>	<p>9. El sistema comprueba que los datos obligatorios sean llenados, en caso contrario ir al <i>flujo alterno</i> 9.1.</p>

Prototipo de Interfaz

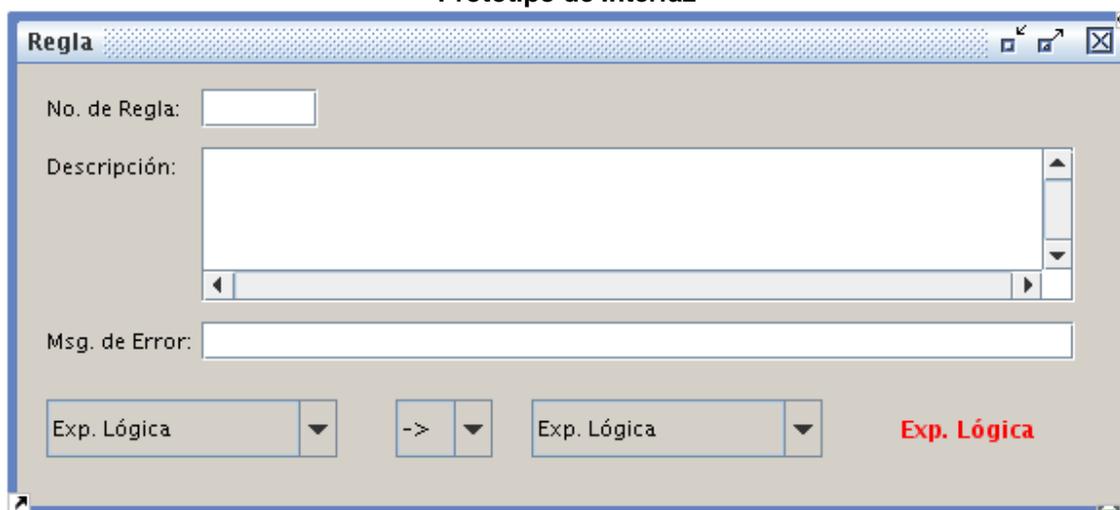


figura 2



figura 3



figura 4

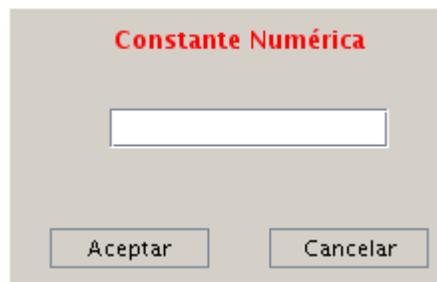


figura 5

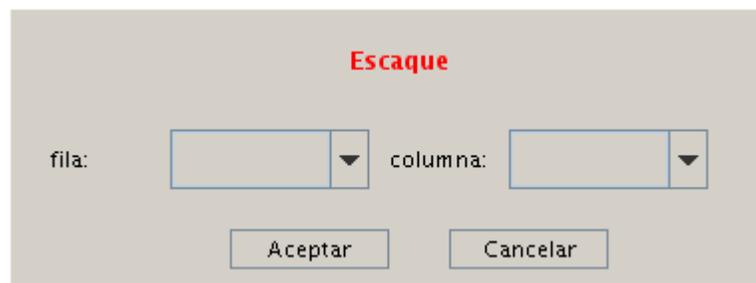


figura 6

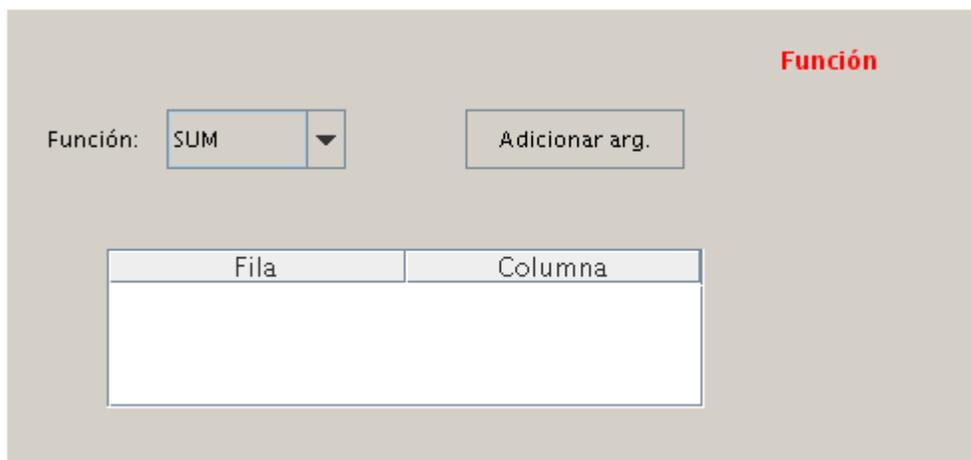


figura 7

figura 8

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>5.1. Si los datos obligatorios no son introducidos, el sistema muestra un mensaje de error.</p> <p>9.1. Si los datos obligatorios no son introducidos, el sistema muestra un mensaje de error</p>
Prototipo de Interfaz	
Poscondiciones	Se obtiene la regla especificada.
Flujo Normal de Eventos	
Sección "Editar regla de validación"	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona editar regla.	2. El sistema muestra un formulario con los valores predefinidos con la posibilidad de modificarlos.
3. El actor modifica los datos necesarios y confirma.	4. El sistema comprueba que los datos obligatorios sean llenados, en caso contrario ir al <i>flujo alternativo</i> 4.1.

	5. El sistema modifica los datos y muestra un mensaje satisfactorio.
<b>Prototipo de Interfaz</b>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 . Si los datos obligatorios no son introducidos, el sistema muestra un mensaje de error volviendo al paso 2.
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	Se obtiene la regla modificada.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar regla de validación”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor selecciona eliminar regla.	2. El sistema muestra mensaje de confirmación de la acción.
3. El actor confirma la acción.	4. El sistema elimina la regla y muestra un mensaje satisfactorio, en caso de no aceptar la acción, ir al <i>flujo alterno</i> 4.1.
<b>Prototipo de Interfaz</b>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1. Si el usuario no acepta el sistema brinda la posibilidad de volver al paso 1.
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	Se elimina la regla.

2.7.2 Tabla 3 Caso de uso: Validar Modelos

<b>Caso de Uso:</b>	Validar Modelos
<b>Actores:</b>	Validador
<b>Resumen:</b>	El caso de uso inicia cuando el validador necesita validar un modelo ya creado. Se valida el modelo deseado y se obtiene el resultado correspondiente a la validación de dicho modelo finalizando así el caso de uso.
<b>Precondiciones:</b>	1. Deben estar creadas las reglas de validación correspondientes al modelo a validar.
<b>Referencias</b>	
<b>Prioridad</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor solicita validar modelos	2. El sistema Muestra una interfaz (ver figura ) que brinda la posibilidad de seleccionar el modelo a validar.
3. El actor selecciona el modelo a validar y solicita validar.	4. El sistema muestra el resultado correspondiente a la validación. En caso contrario ir a <i>flujo alterno 4.1</i> .
<b>Prototipo de Interfaz</b>	

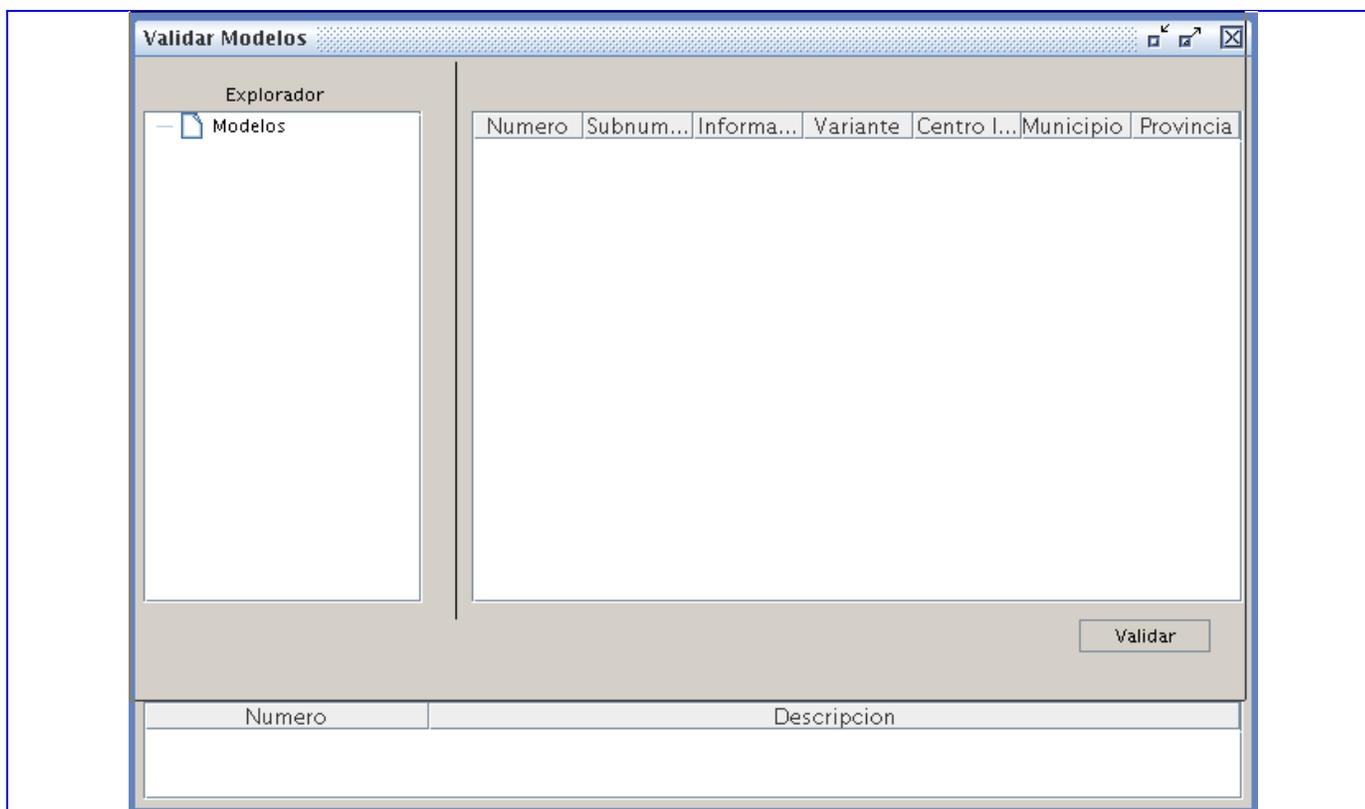


figura 1

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si el sistema no muestra resultado de validación el sistema muestra mensaje de error por no existir reglas de validación sobre el presente modelo.
Prototipo de Interfaz	
Poscondiciones	Se obtiene el resultado de la validación del modelo.

2.8 Diagrama de casos de uso del sistema

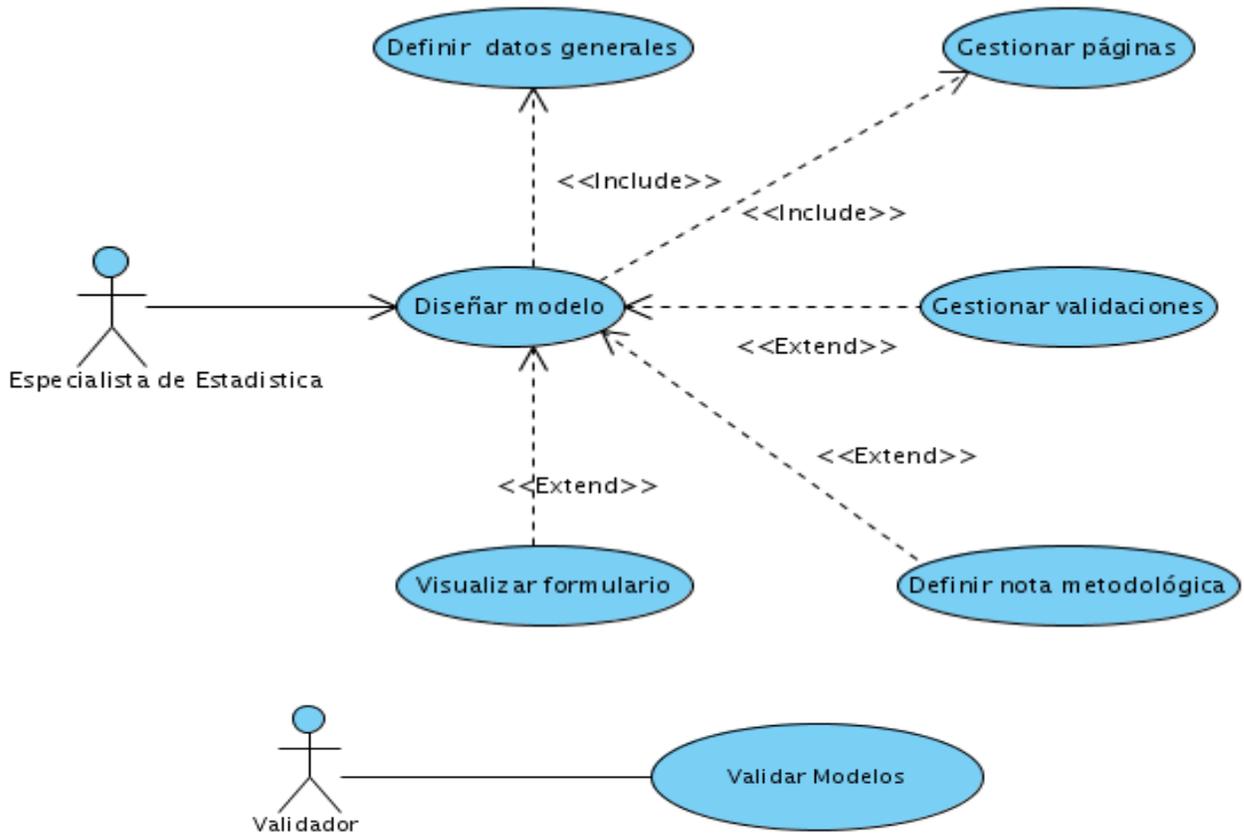


Fig. 1 Diagrama de casos de uso del sistema

## 2.9 Diagrama de clases del análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real y no de la implementación automatizada del mismo. Surge como parte del refinamiento de los requisitos obtenidos anteriormente. Se realiza un modelo esbozando cómo llevar a cabo la funcionalidad dentro del sistema.

### Validar Modelos

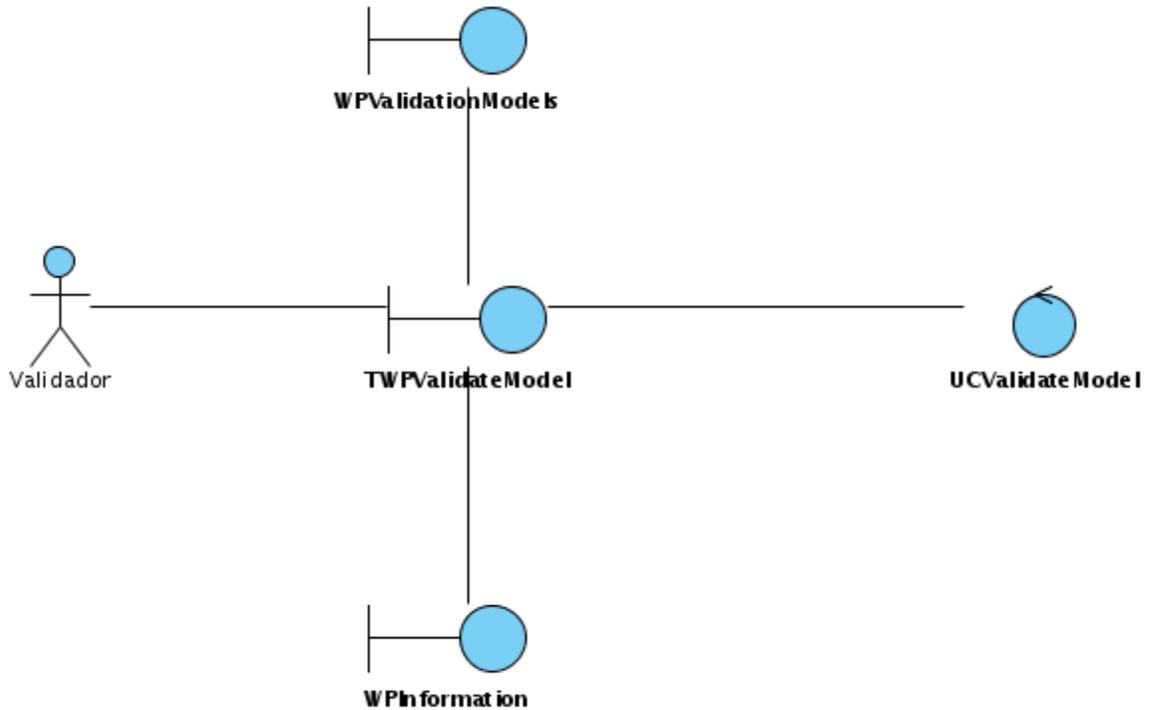


Fig. 2 Diagrama de clases del análisis: Validar Modelos

Gestionar Validaciones

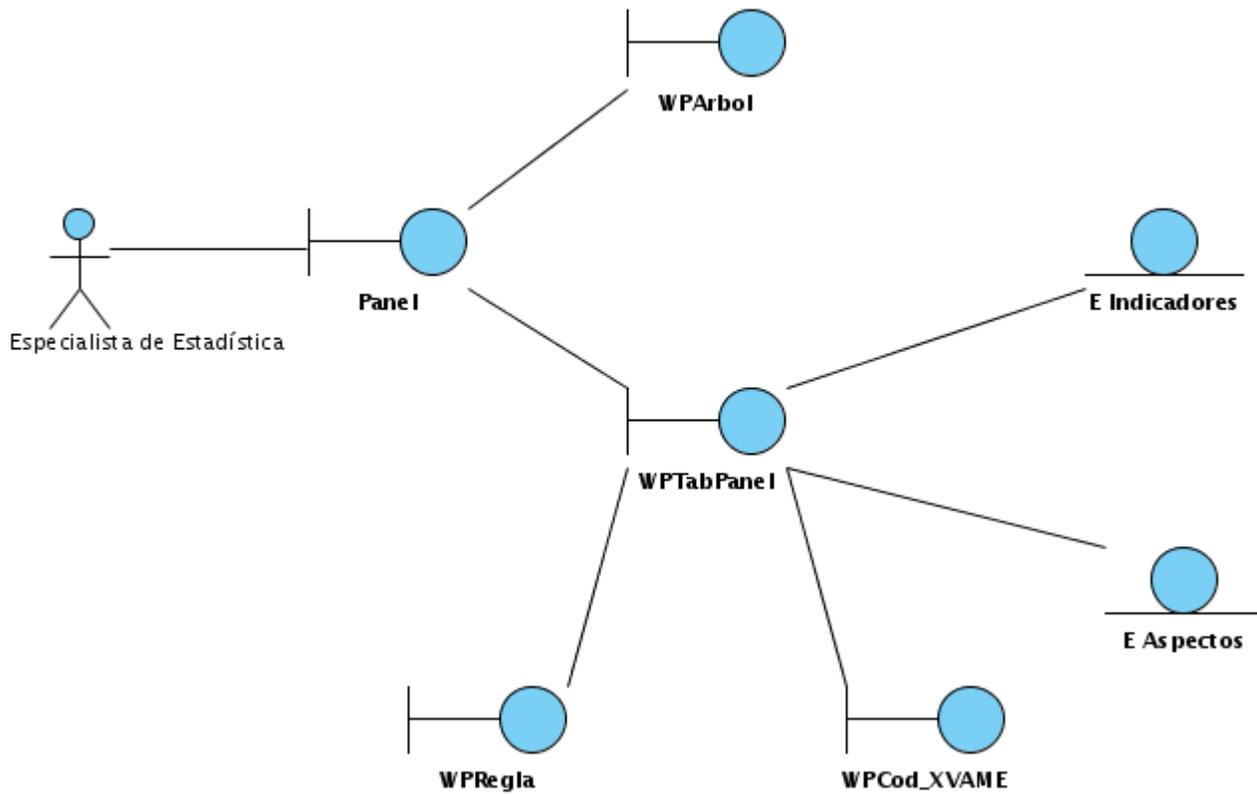


Fig. 3 Diagrama de clases del análisis: Gestionar Validaciones

2.10 Diagramas de secuencia

Los diagramas de secuencia muestran una interacción que está organizada como una secuencia temporal. En particular, muestran los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal.

Validar Modelos

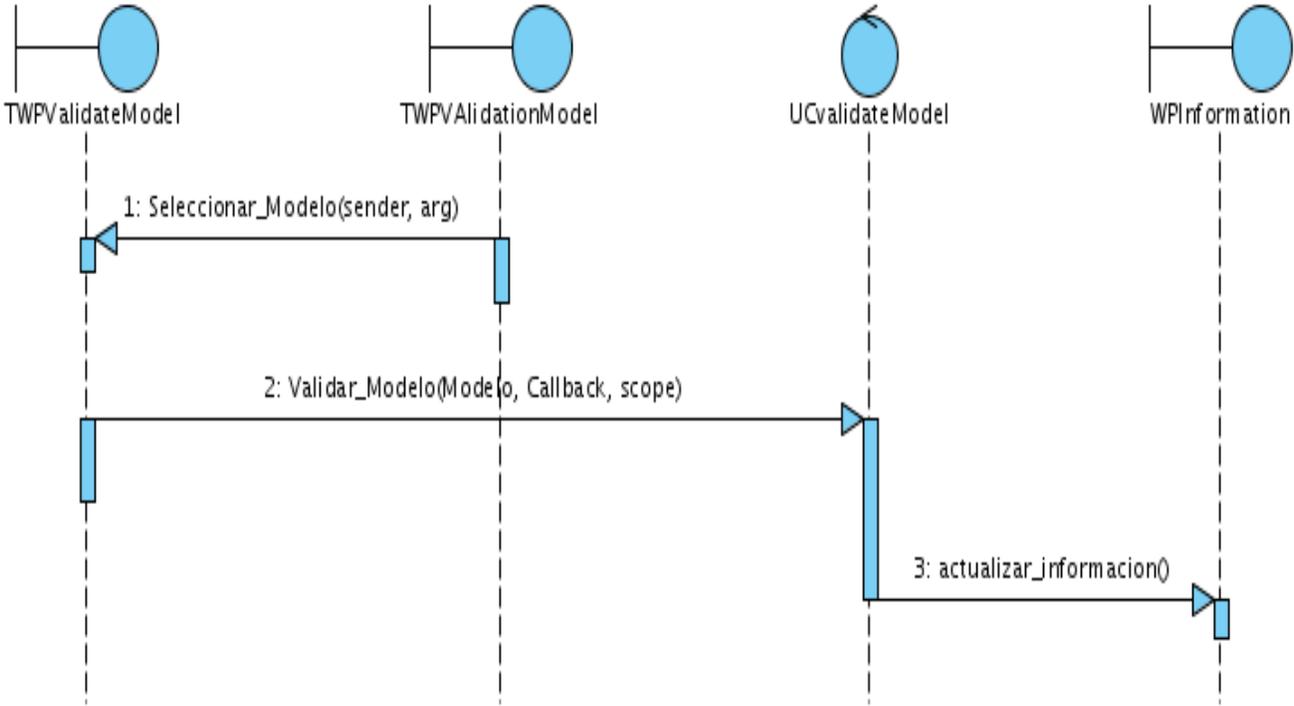


Fig. 4 Diagrama de secuencia: Validar Modelos

**Gestionar Validaciones**

**Sección: “Adicionar Regla”**

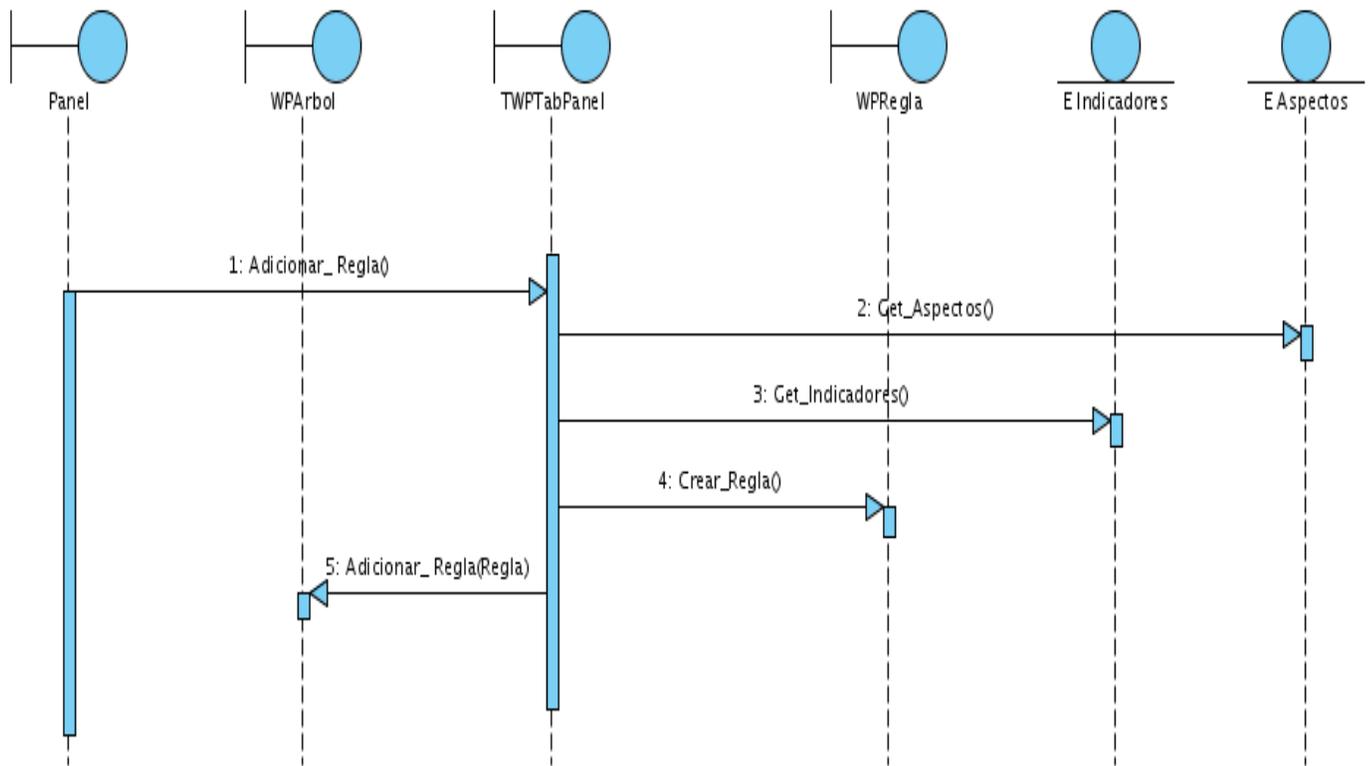


Fig. 5 Diagrama de secuencia: Gestionar Validaciones, sección “Adicionar Regla”

**Sección: “Modificar Regla”**

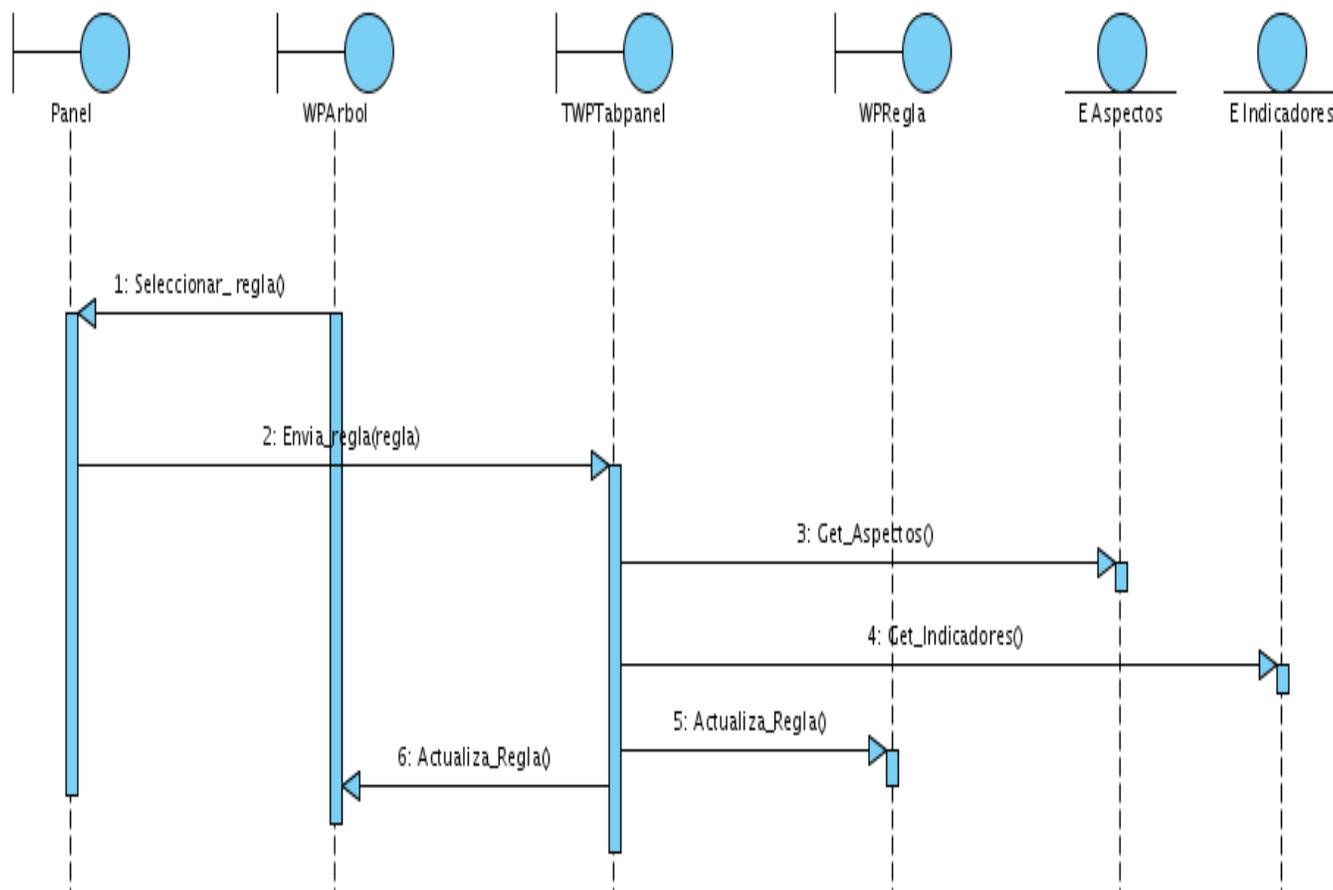


Fig. 6 Diagrama de secuencia: Gestionar Validaciones, sección “Modificar Regla”

### Sección: “Eliminar Regla”

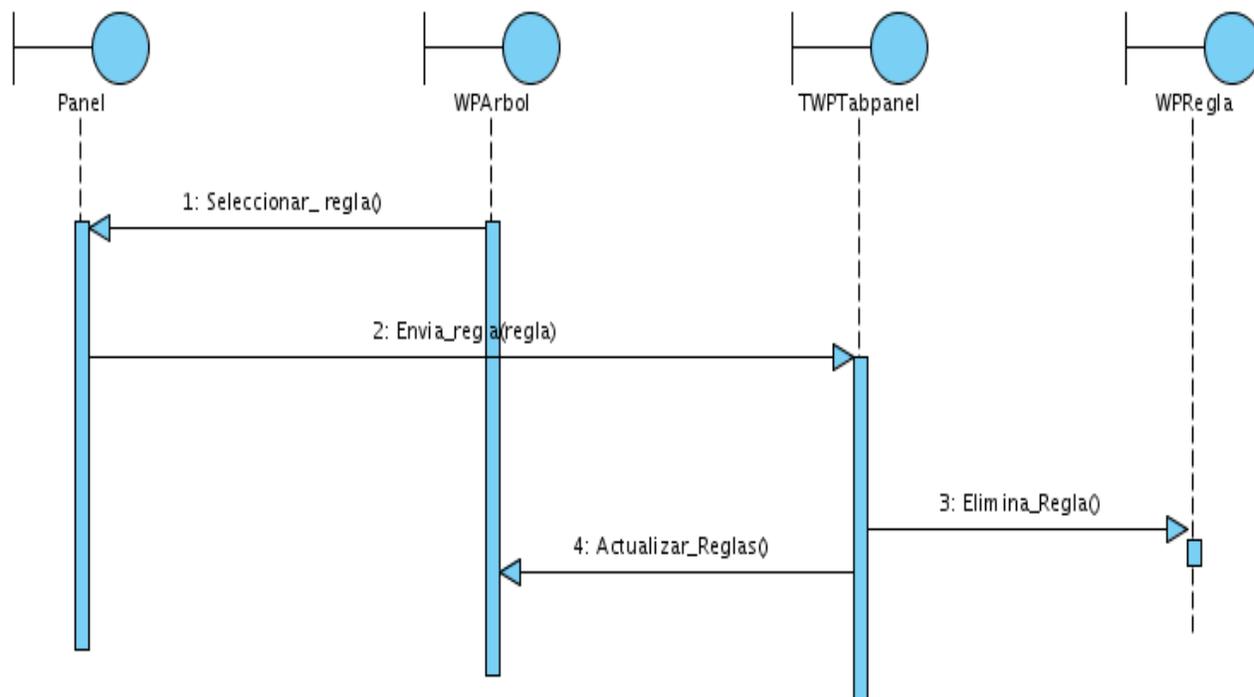


Fig. 7 Diagrama de secuencia: Gestionar Validaciones, sección “Eliminar Regla”

### 2.11 Modelo del diseño

En el diseño se modela el sistema y se encuentra su forma para que soporte todos sus requerimientos, incluyendo los no funcionales y sus restricciones. Debe ser una guía para los que construyan el código y los que prueban y mantienen el software.

#### 2.11.1 Patrones de diseño

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basadas en la experiencia y que se ha demostrado que funcionan.

#### Patrones creacionales

Prototype (Prototipo): Crea nuevos objetos clonándolos de una instancia ya existente. Con la creación de instancias utilizando el objeto *prototype* se pone en práctica este patrón.

#### Patrones de Comportamiento

Interpreter (Intérprete): Dado un lenguaje, define una gramática para dicho lenguaje, así como las herramientas necesarias para interpretarlo. Este patrón es utilizado en la definición del Lenguaje de Marcado para Validación de Modelos Estadísticos (XVAME).

**GRASP: Patrones para Asignar Responsabilidades**

Patrón Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. El bajo acoplamiento significa que una clase no depende de muchas clases. Se pone en práctica con las clases: Literal, Escaque, Rango y Función que no presentan ninguna dependencia.

**2.11.2 Diagrama de clases**

**Gestionar Validaciones**

<b>Clase: Main</b>
<b>Descripción:</b> es el componente inicial del módulo donde convergen la WPTabPanel y WParbol.

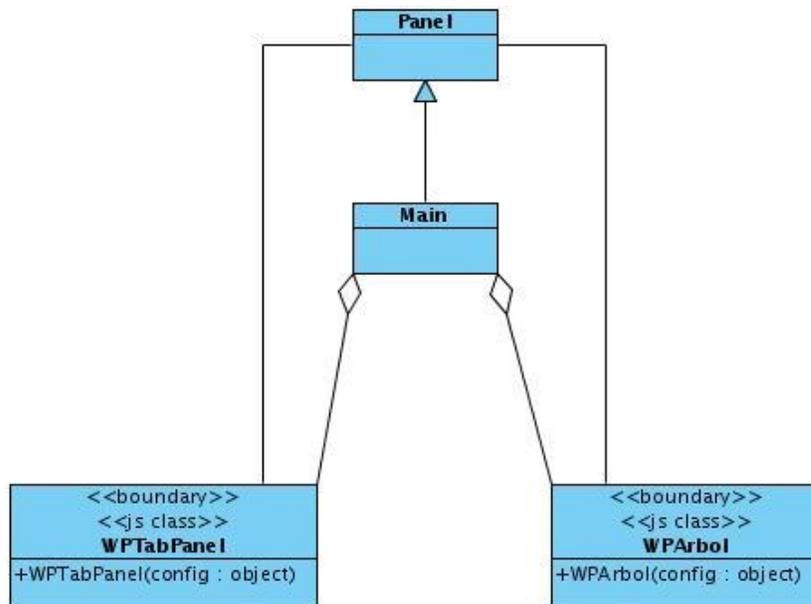


Fig. 8 Clase Main

<b>Clase: WParbol</b>
<b>Descripción:</b> en esta clase se muestran las reglas creadas cada una como un nodo de un árbol.

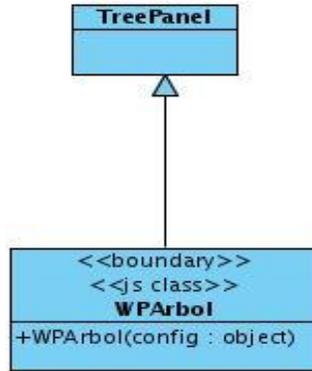


Fig. 9 Clase WPArbol

<b>Clase: WPTabPanel</b>
<b>Descripción:</b> es la encargada del tratamiento de la regla y del formato XVAME.

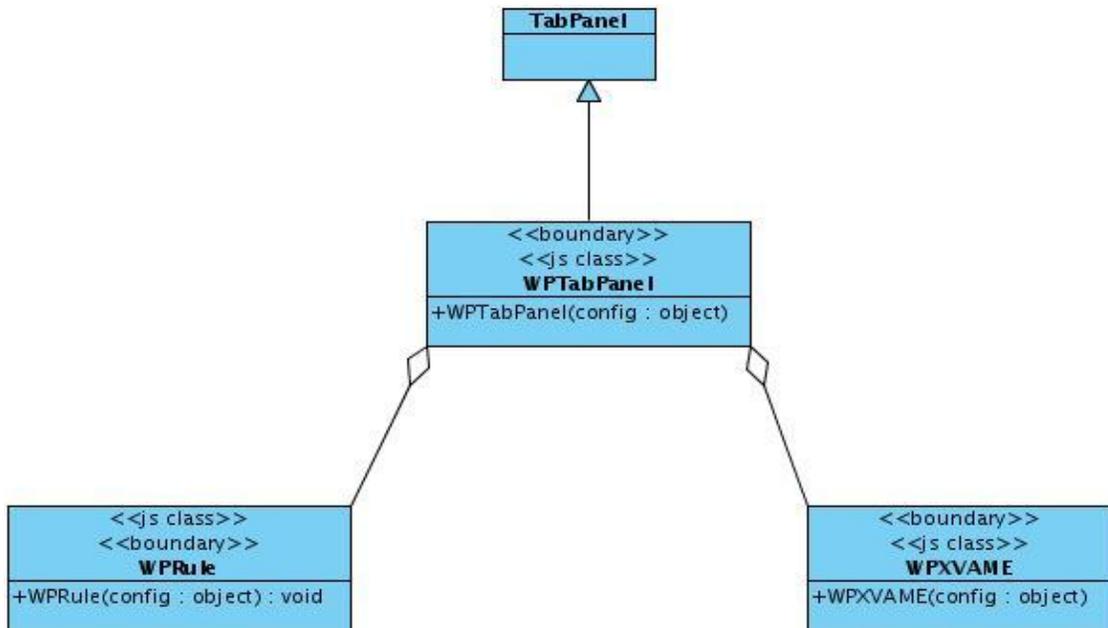


Fig. 10 Clase WPTabPanel

<b>Clase: WPRule</b>
<b>Descripción:</b> esta clase controla la regla para el manejo de la estructura que la conforma.

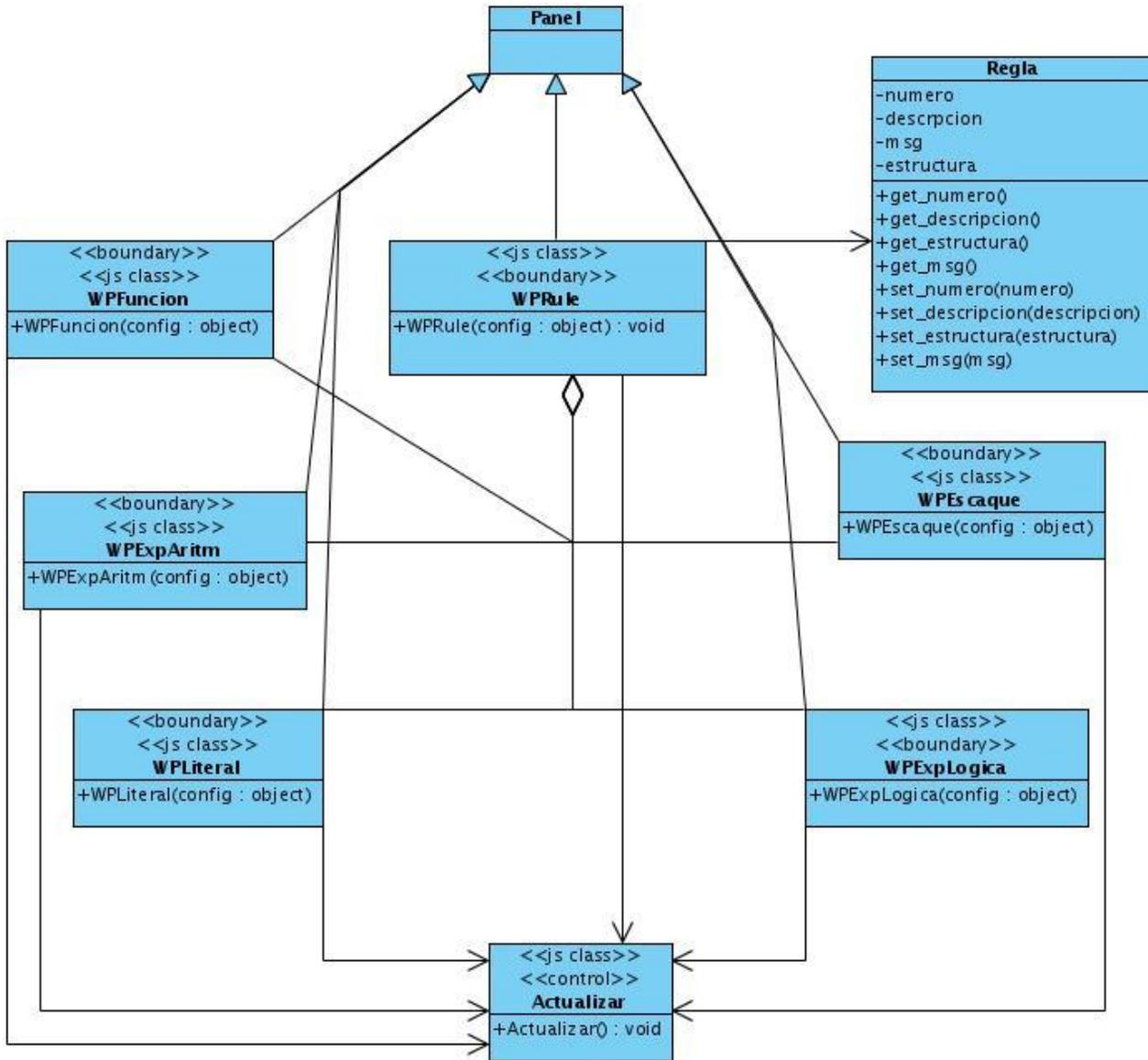


Fig. 11 Clase WPRule

<b>Clase:</b> WPCodXVAME
<b>Descripción:</b> encargada de mostrar el formato XVAME de una regla.

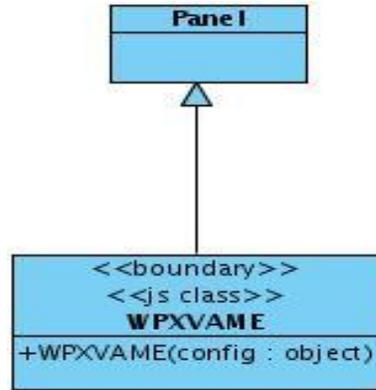


Fig. 12 Clase WPXVame

**Clase: WPExpLogica**

**Descripción:** en esta clase se controla la estructura para la creación de un formato de expresión lógica.

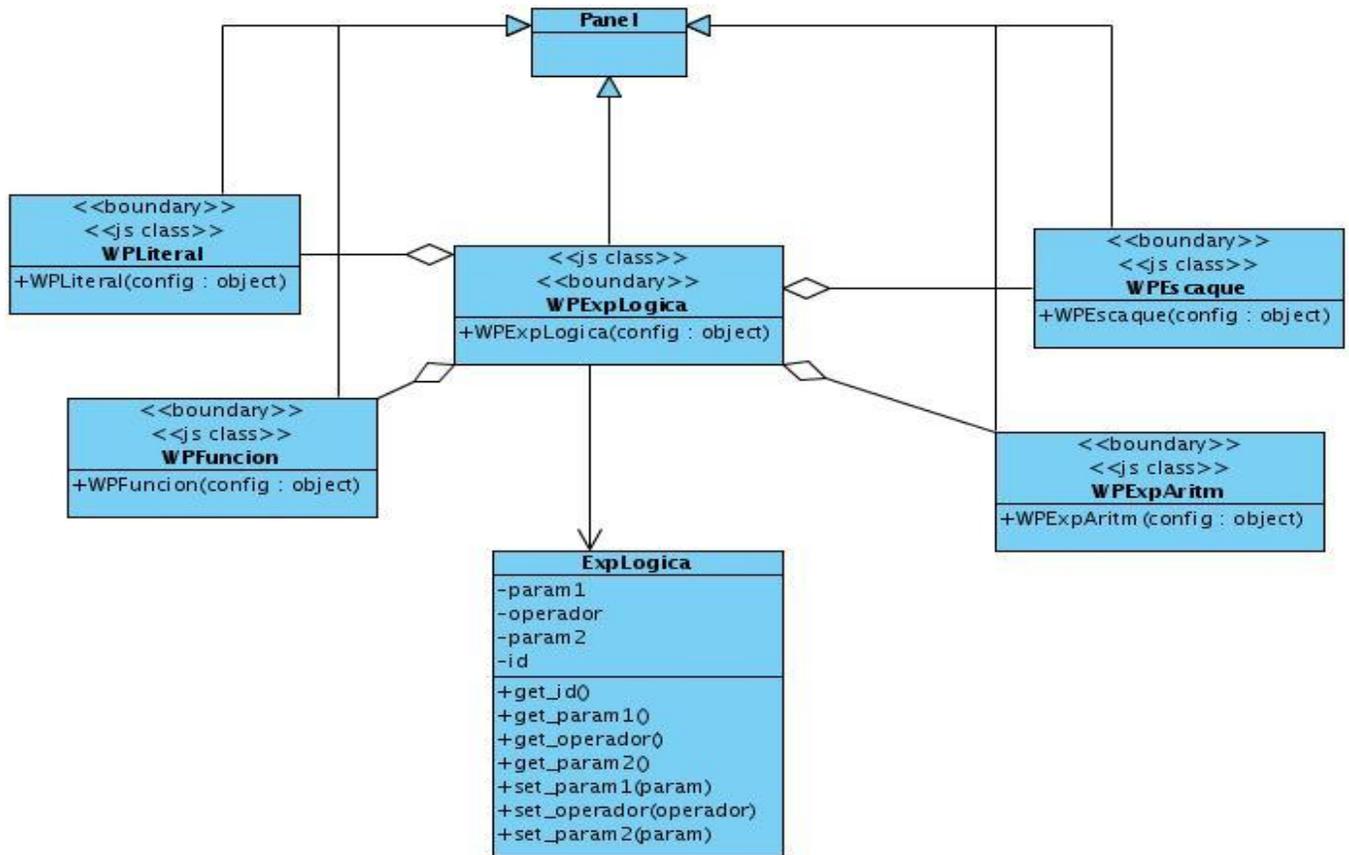


Fig. 13 Clase WPExpLogica

<b>Clase: WPLiteral</b>
<b>Descripción:</b> en esta clase se controla la estructura para la creación de una constante numérica.

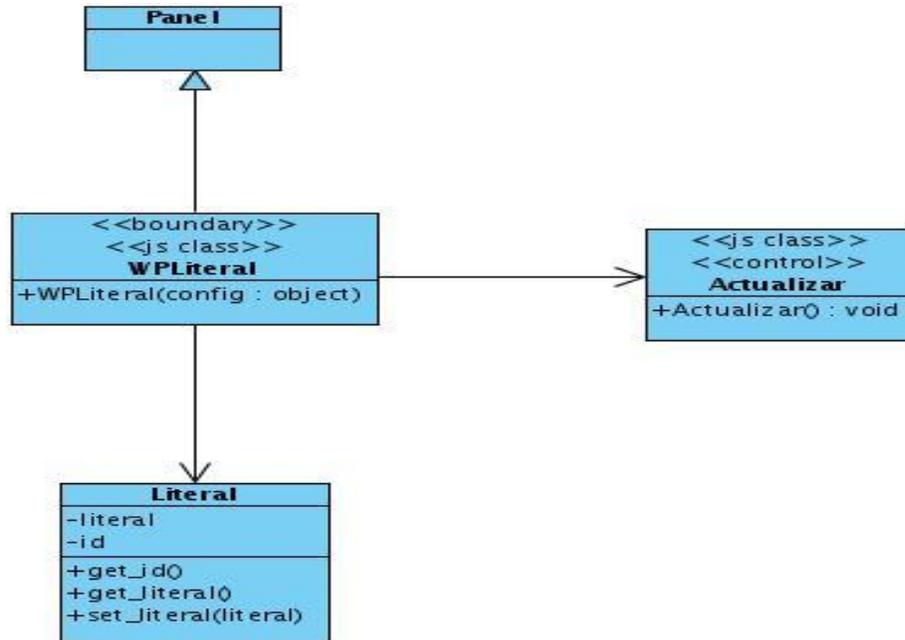


Fig. 14 Clase WPLiteral

<b>Clase: WPEscaque</b>
<b>Descripción:</b> en esta clase se controla la estructura para la creación de un escaque.

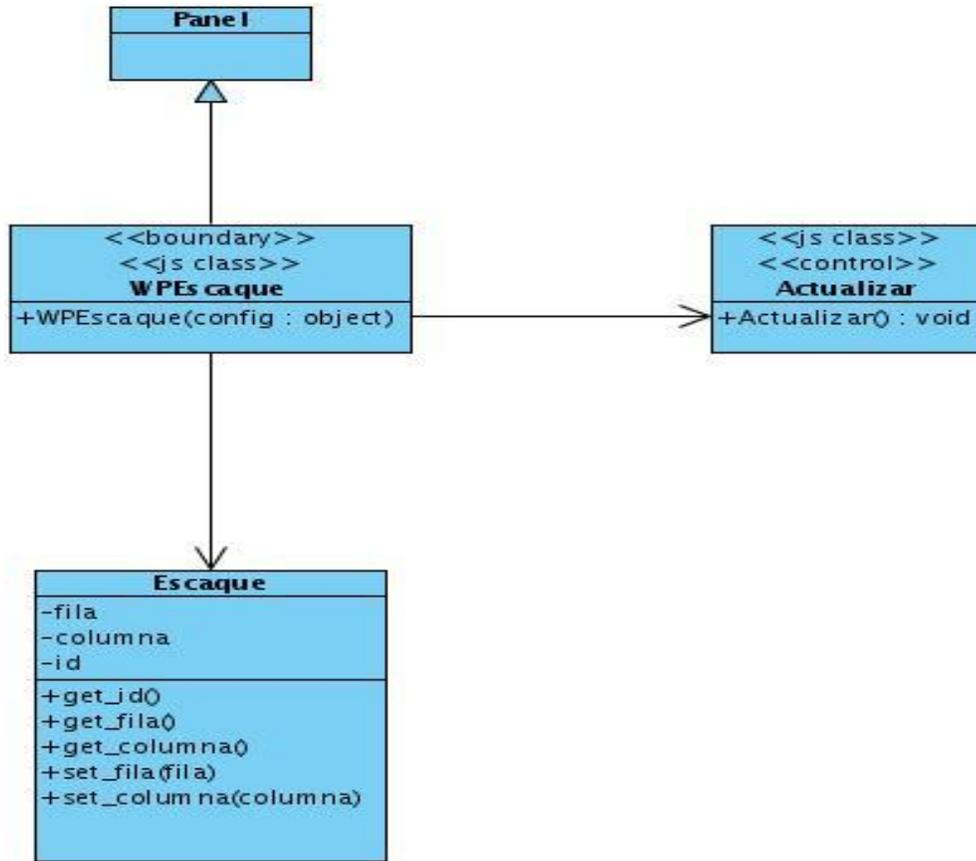


Fig. 15 Clase WPEscaque

**Clase:** WPFuncion

**Descripción:** en esta clase se controla la estructura para la creación de una función definida por XVAME.

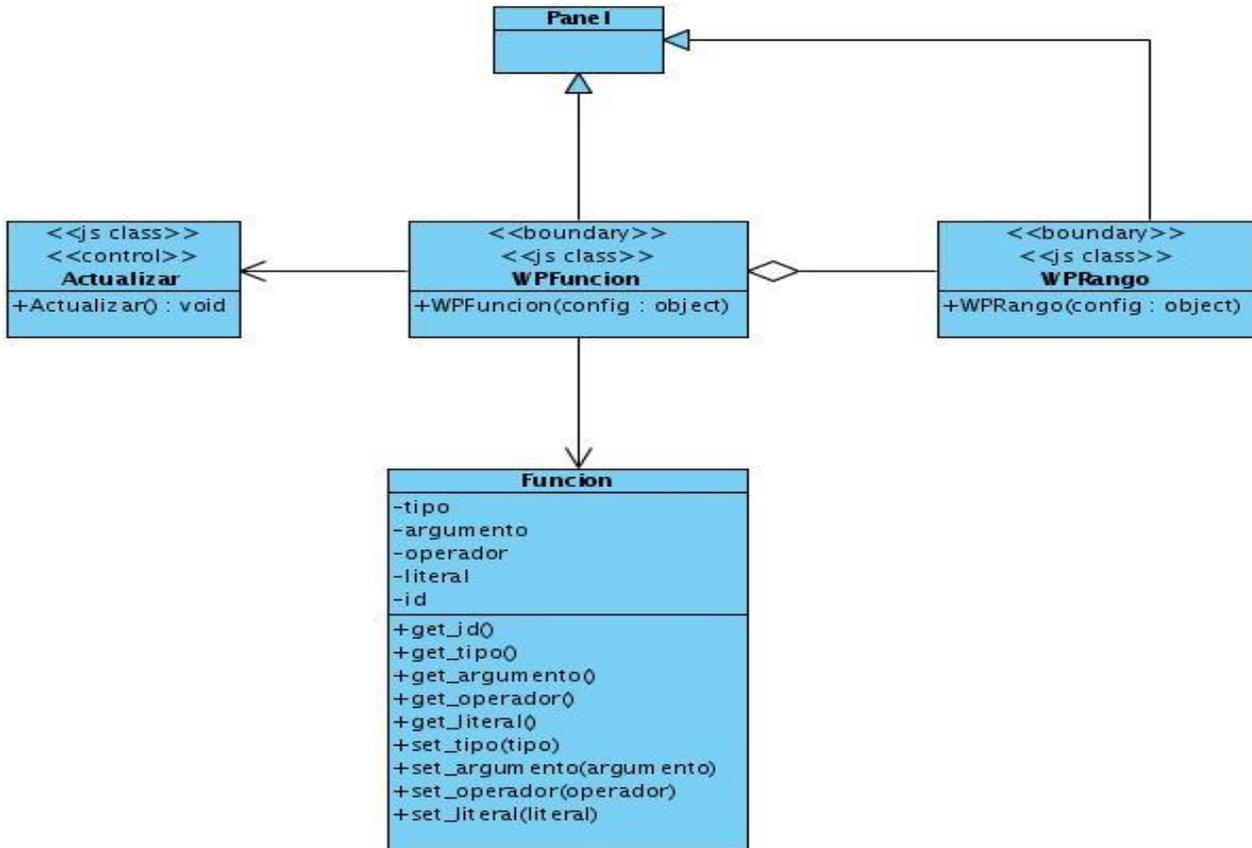


Fig. 16 Clase WPFuncion

<b>Clase: WPRango</b>
<b>Descripción:</b> en esta clase se controla la estructura para la creación de un rango.

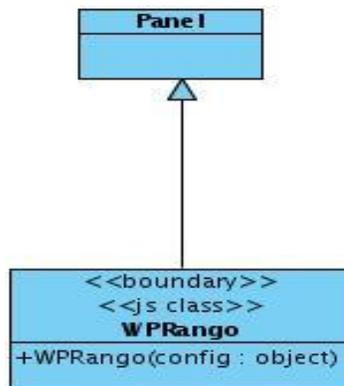


Fig. 17 Clase WPRango

<b>Clase: WPAritmetica</b>
<b>Descripción:</b> en esta clase se controla la estructura para la creación de una expresión aritmética.

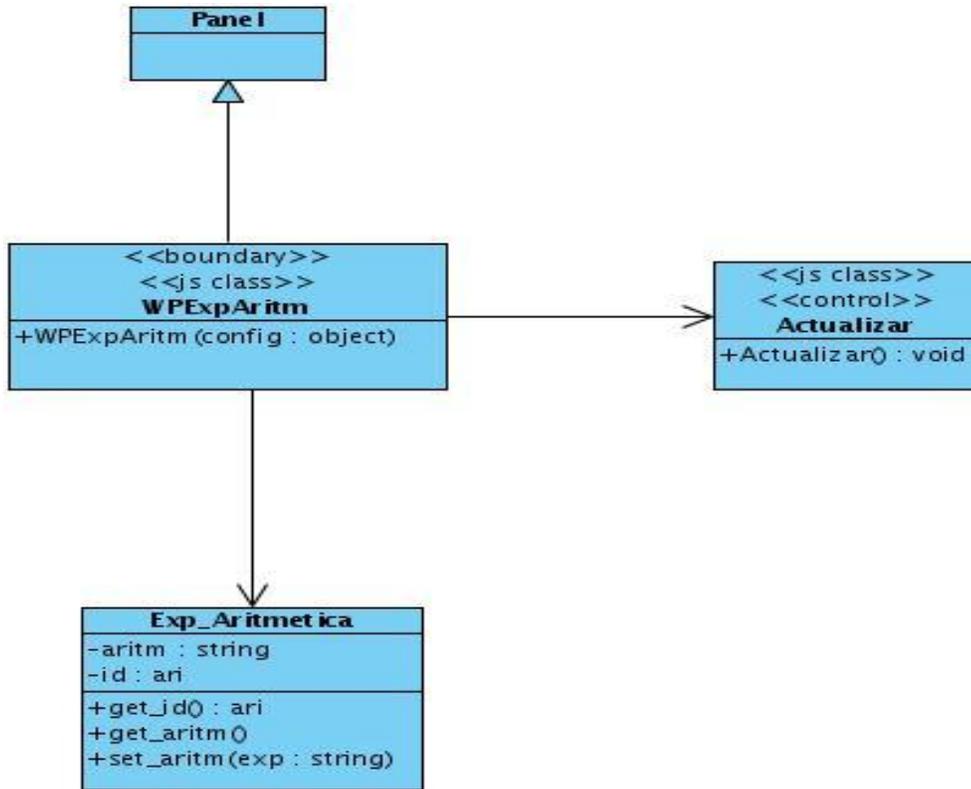


Fig. 18 Clase WPAritmetica

Las clases Literal, Escaque, Función, Aritmética y Regla constituyen unidades básicas que encapsulan toda la información referente a cada una de esas estructuras y la clase Actualizar es la encargada de actualizar la estructura de una regla durante su creación.

### Validar Modelos

<b>Clase: TWPValidateModel</b>
<b>Descripción:</b> esta clase controla la convergencia entre los modelos a validar y la información generada por cada uno de los modelos.

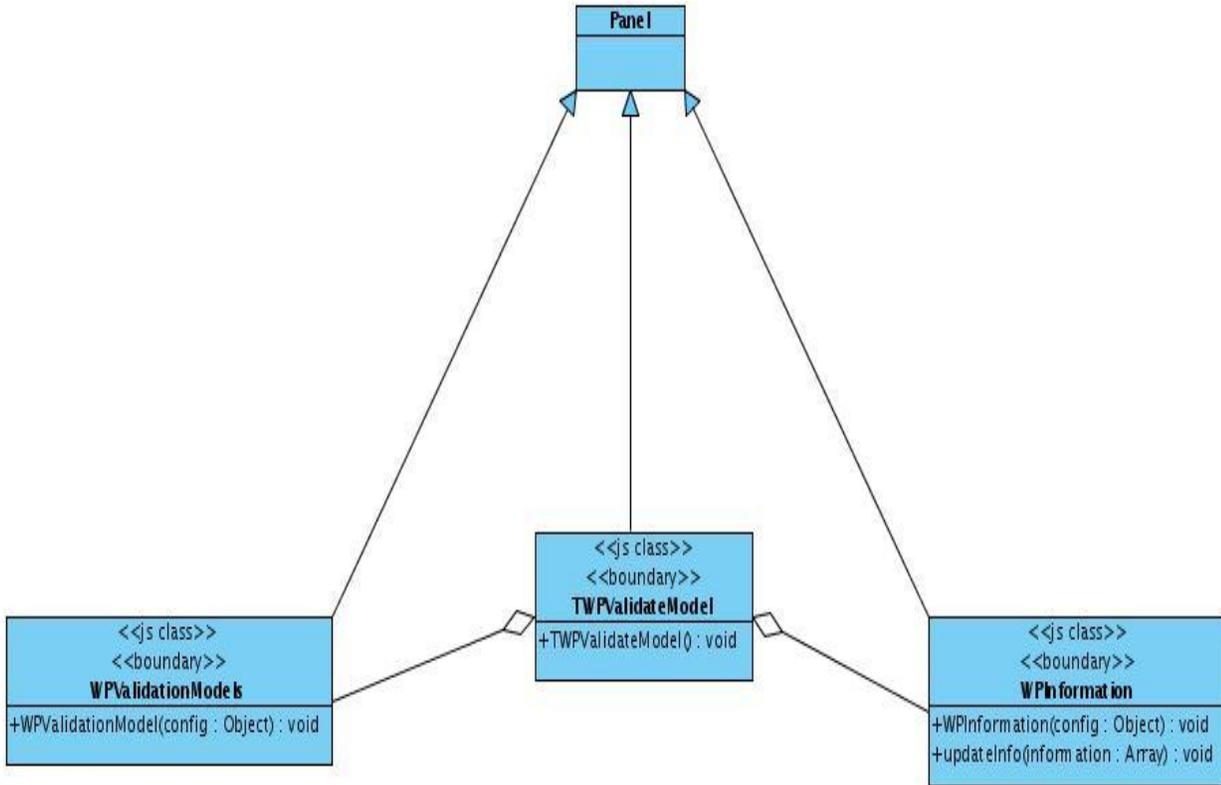


Fig. 19 Clase TWPValidateModel

<b>Class: WPValidationModels</b>
<b>Descripción:</b> Esta clase muestra todos los modelos a validar.

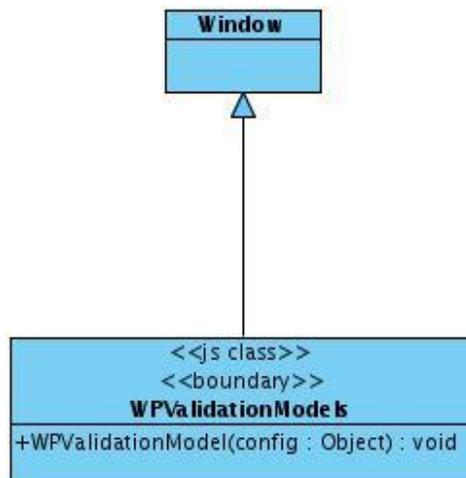


Fig. 20 Clase WPValidationModels

**Clase:** WPinformation

**Descripción:** Esta clase muestra la información de un modelo y de su validación.

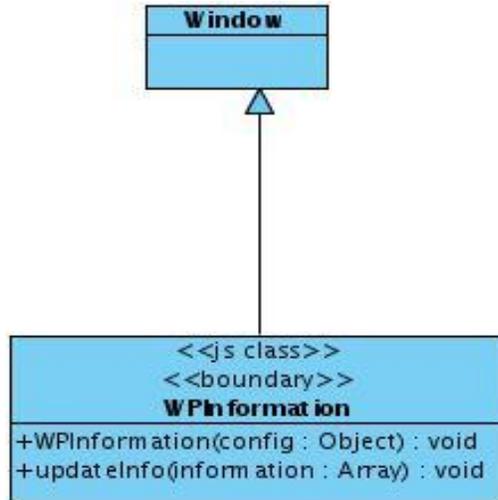


Fig. 21 Clase WPinformation

**Clase:** UCValidateModel

**Descripción:** es la encargada de controlar la validación de un modelo.

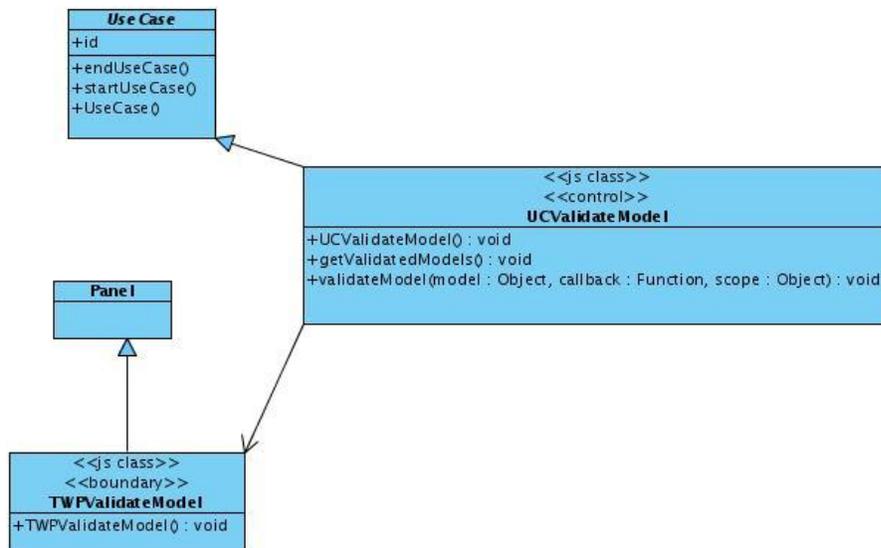


Fig. 22 Clase UCValidateModel

## 2.12 Interfaz de Usuario

### Gestionar Validaciones

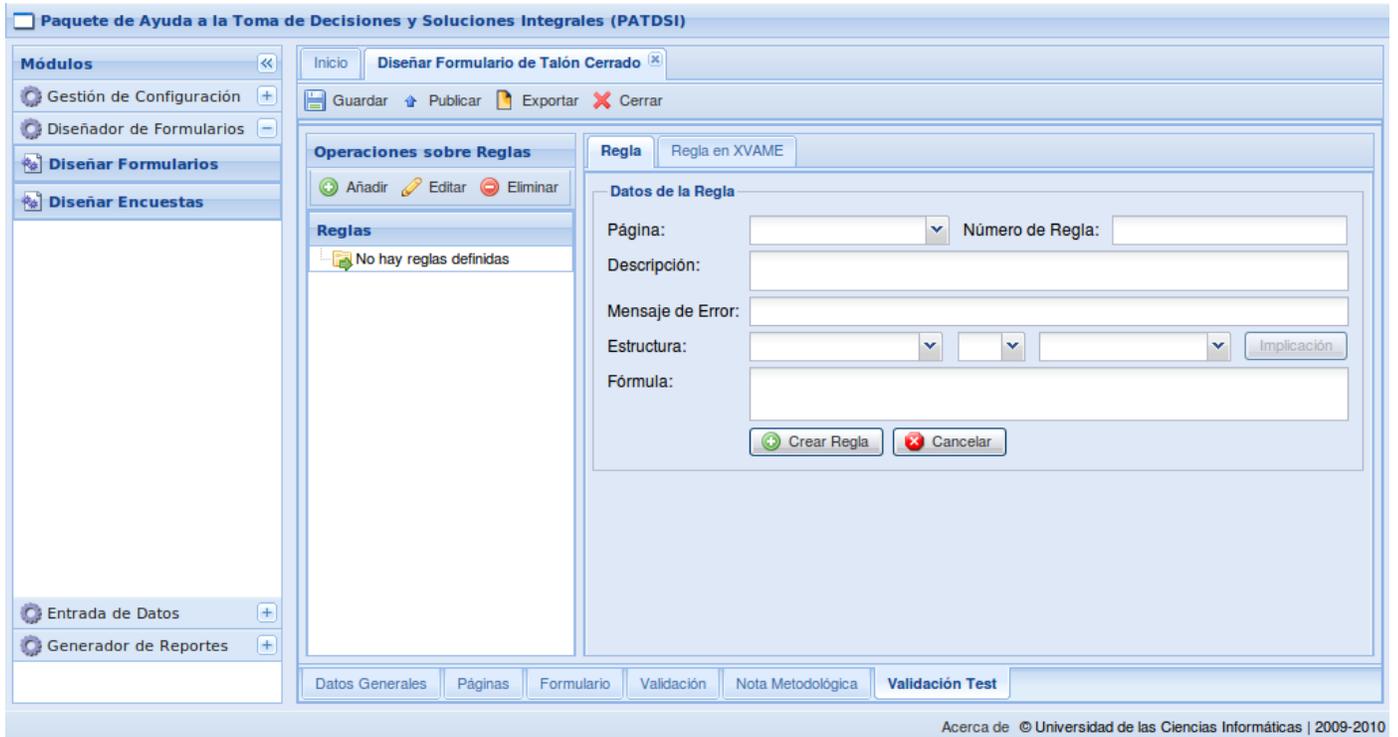


Fig. 23 Interfaz de Usuario: "Generar Regla"

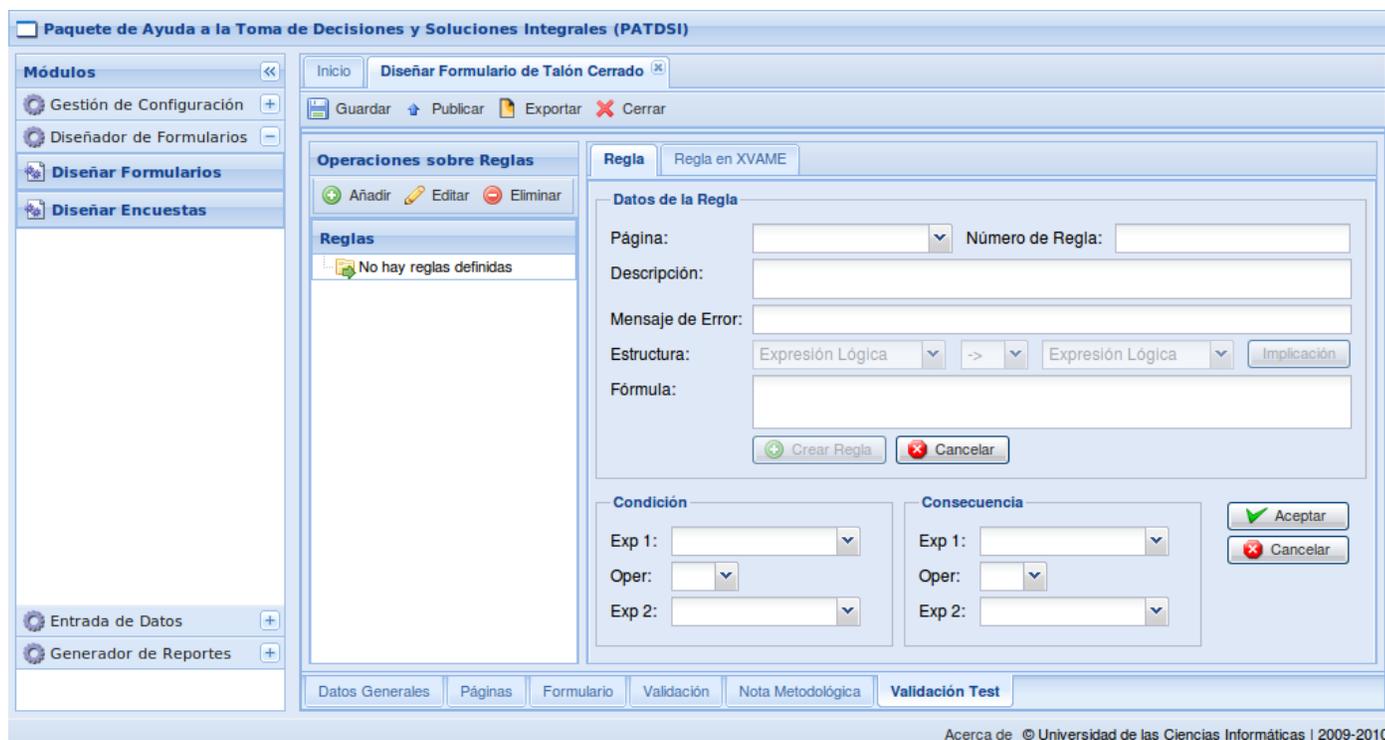


Fig. 24 Interfaz de Usuario: "Generar Regla de Implicación"

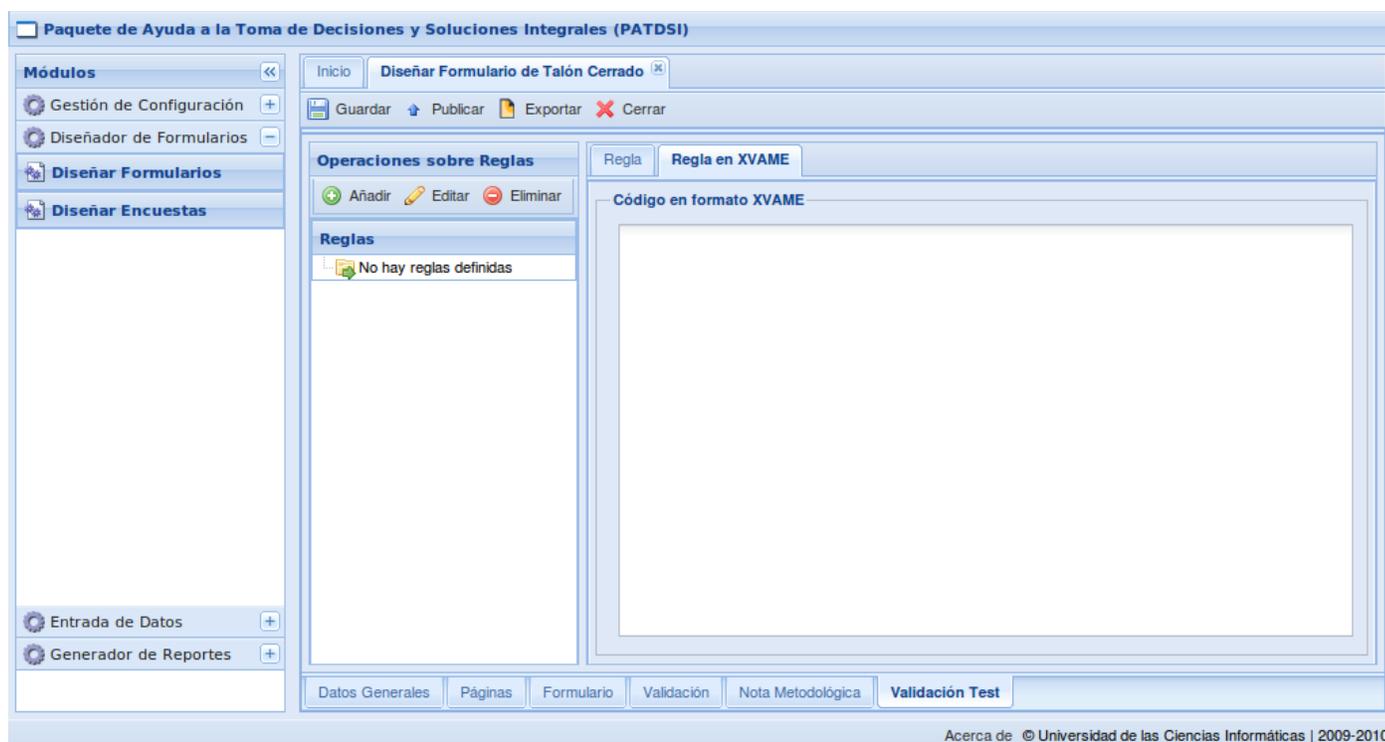


Fig. 25 Interfaz de Usuario: "Mostrar Código XVAME"

### Validar Modelos

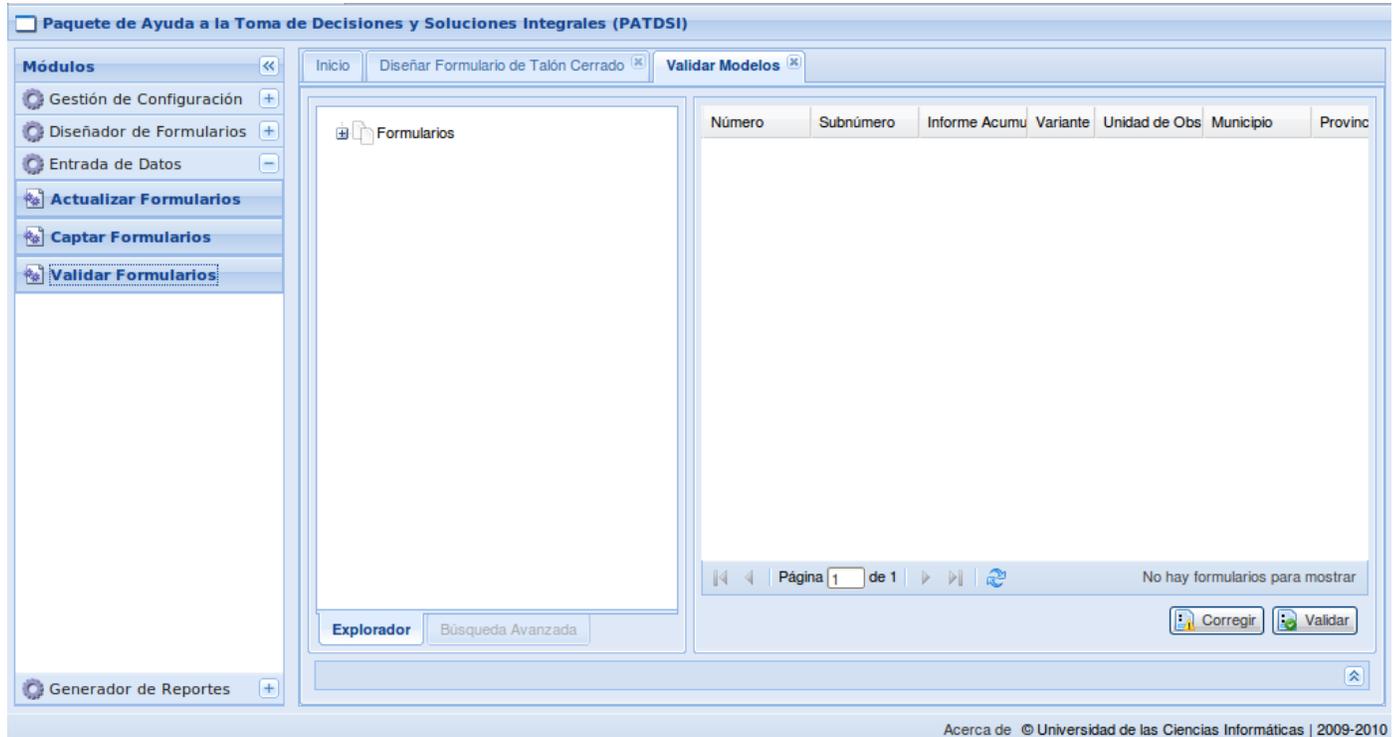


Fig. 26 Interfaz de Usuario: "Validar Modelos"

### Conclusiones

En este capítulo se trató la propuesta del módulo con el análisis de los modelos estadísticos como objeto de validación y la notación utilizada como el mecanismo de validación de estos modelos. Se detectaron los requisitos funcionales y los no funcionales que sirvieron de base, para la elaboración del modelo de casos de uso del sistema y con esto último se impulsó al desarrollo de las clases del análisis, los diagramas de secuencia y la elaboración de las clases del diseño. Como consecuencia se desarrolló un capítulo que es de suma importancia porque el resultado del mismo constituye la entrada al flujo de trabajo de implementación pues describe detalladamente las acciones a tomar durante la implementación.

# Capítulo 3

## Implementación y Prueba

### Introducción

En este capítulo se describe la implementación como objetivo final del diseño anteriormente desarrollado, la organización de los componentes creados durante la implementación, la estructura a utilizar para el despliegue del sistema en función de nodos y las pruebas que acreditan el correcto funcionamiento del producto.

### 3.1 Modelo de implementación

El Modelo de Implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para el despliegue del sistema. Este modelo se encuentra descrito en términos de dependencias a través del diagrama de componentes.

### 3.2 Diagrama de Componentes

El diagrama de componentes muestra las relaciones entre las partes físicas y reemplazables del sistema, por tanto, expresa las dependencias existentes entre estas estructuras denominadas componentes.

#### Diagrama de paquetes de componentes

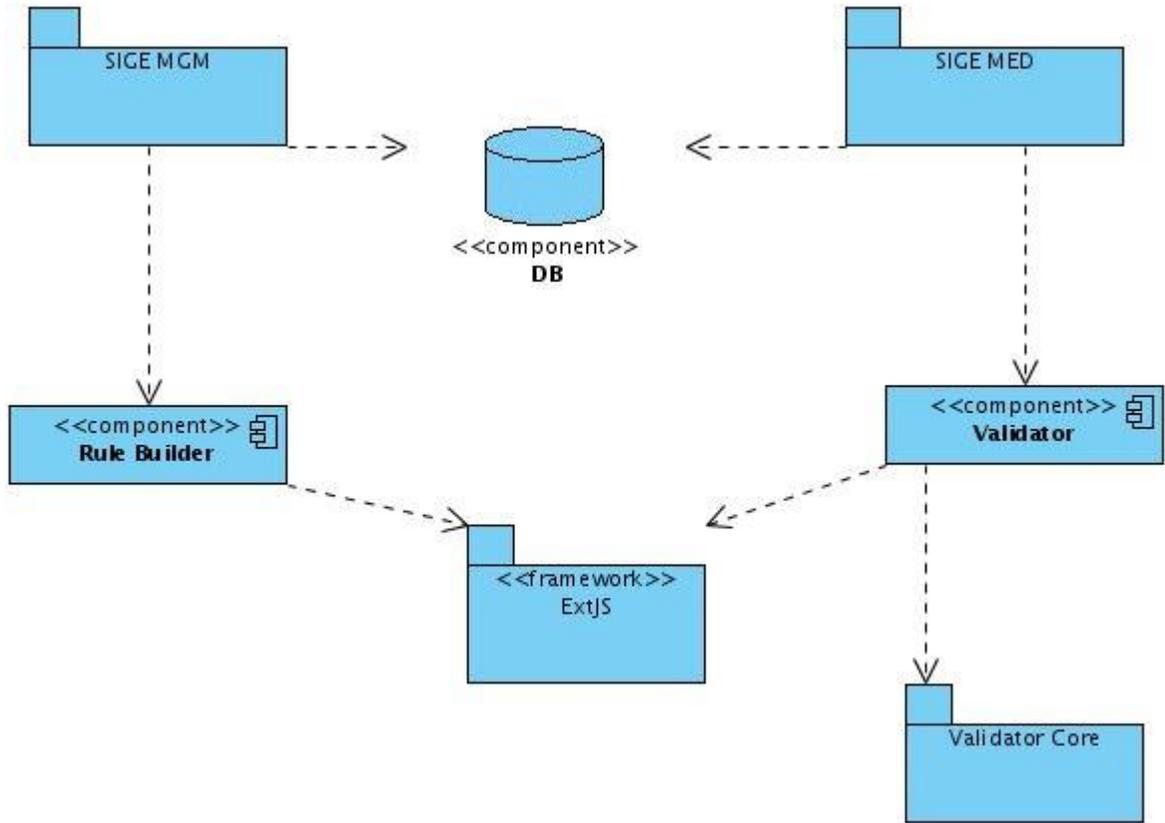


Fig. 27 Diagrama de Componentes

Diagrama de componentes: Gestionar Validaciones

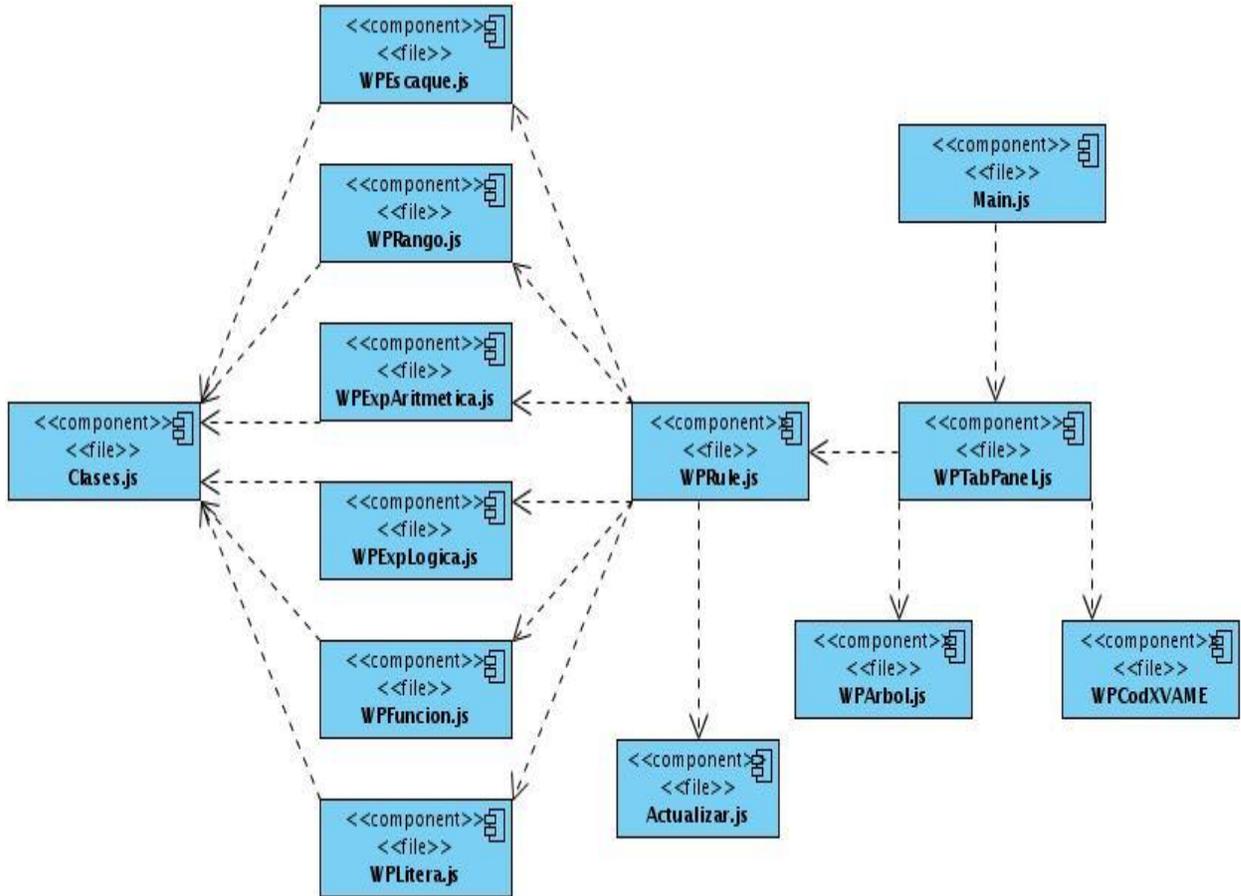


Fig. 28 Diagrama de componentes: Gestionar Validaciones

Diagrama de componentes: Validar Modelos

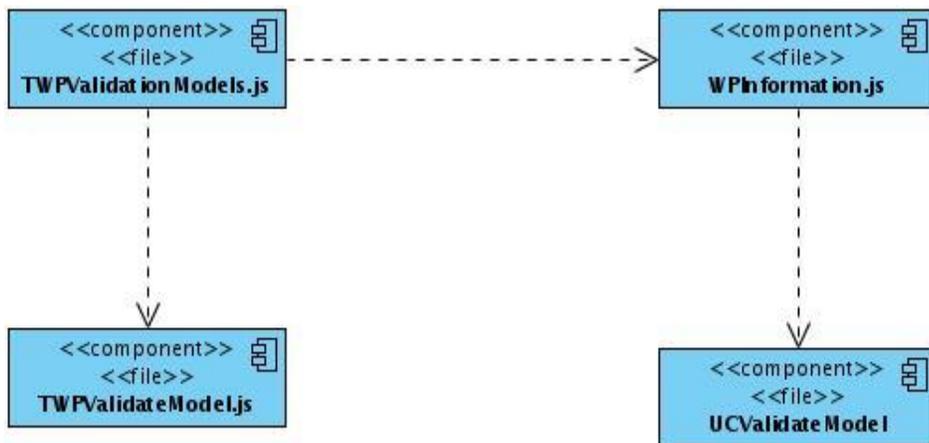


Fig. 29 Diagrama de componentes: Validar Modelos

Diagrama de componentes: Core

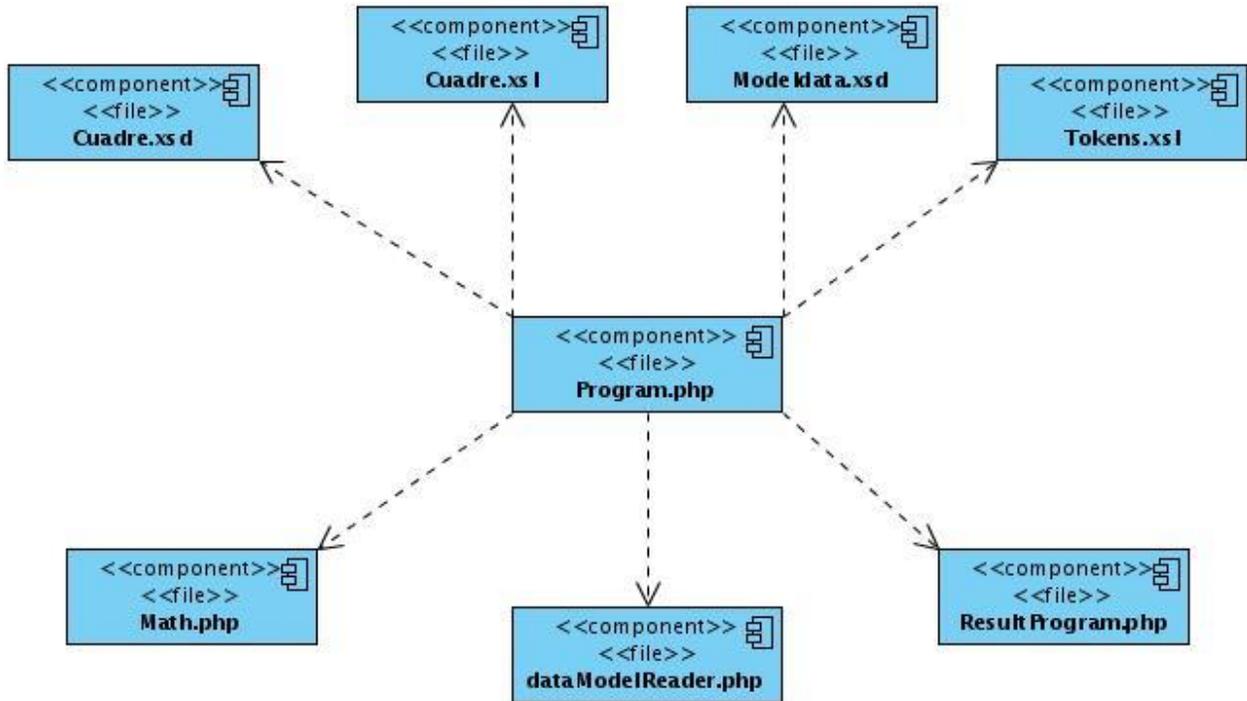


Fig. 30 Diagrama de componentes: Core

### 3.3 Modelo de despliegue

En el modelo de despliegue se representa la configuración de los tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes. En el mismo se muestra una computadora (PC) cliente desde la cual el usuario accederá al sistema. Una PC Servidor Web donde estará emplazada la aplicación, una PC Servidor de Base de Datos en la cual estará ubicada la base de datos que utilizará el sistema.

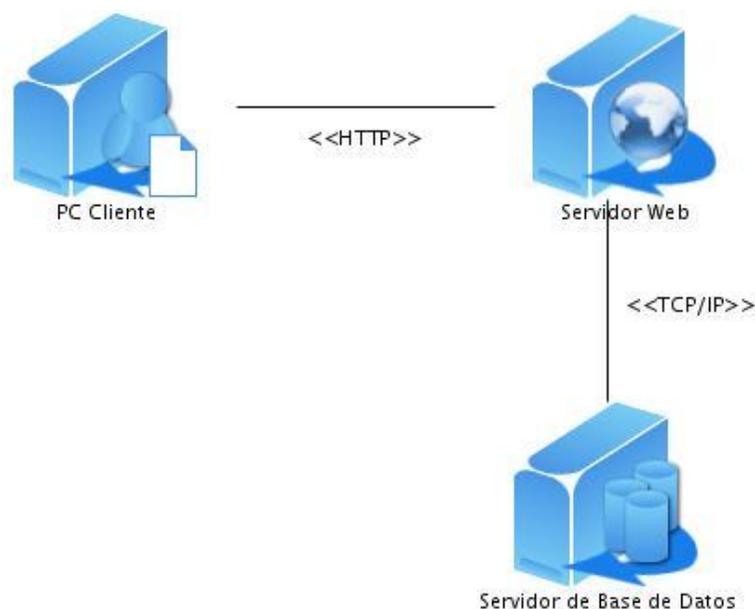


Fig. 31 Modelo de Despliegue

### 3.4 Pruebas

Las pruebas del software son el proceso que permite verificar y mostrar la calidad de un producto, a través de resultados registrables que proporcionan una evaluación.

Dentro de los diversos métodos de pruebas se encuentran las pruebas de caja negra, las cuales verifican las especificaciones funcionales sin tener en cuenta la estructura interna del programa y son realizadas sin el conocimiento interno del producto. Por ello se realizan sobre la interfaz del sistema controlando los datos de entrada y de salida.

Para la evaluación del presente producto se realizaron las siguientes pruebas estructuradas por casos de uso basadas en el método de caja negra anteriormente descrito.

#### Gestionar Validaciones

Descripción general.

El caso de uso inicia cuando el especialista de estadística necesita generar una o varias reglas de validación para un modelo estadístico de acuerdo con diferentes criterios. Se obtiene el programa de validación de acuerdo con el criterio especificado terminando el caso de uso.

Condiciones de ejecución.

- Debe existir el modelo a validar.
- Deben estar definidos los indicadores y aspectos.

**Tabla 4 Secciones a probar del caso de uso Gestionar Validaciones**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Adicionar regla de validación	EC 1.1: Flujo básico	Creación de una regla de validación, cumpliendo los pasos correspondientes.
	EC 1.2: Flujo alterno.	Expresa las inconformidades del sistema por campos no llenados o con datos incorrectos.
SC2: Editar regla de validación	EC 2.1: Flujo básico	Modificación de una regla de validación ya creada.
	EC 2.2: Flujo alterno.	Expresa las inconformidades del sistema por campos no llenados o con datos incorrectos.
SC3: Eliminar regla de validación	EC 3.1: Flujo básico	Elimina una regla de validación ya creada.

**Tabla 5 Descripción de variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Número de la regla.	Campo de texto	No	Un número entero
2	Descripción	Campo de texto	No	Un texto que defina la funcionalidad de la regla.

3	Mensaje de error	Campo de texto	No	Texto que el usuario define para que sea mostrado en caso de haber fallado la regla.
4	Estructura principal.	Tres listas desplegables	No	Estructura que compone la regla.

Tabla 6 Matriz de datos: Adicionar regla de validación

Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Flujo básico	1	Descripción de la regla 1	Error en la regla 1	Seleccionar estructura	Se muestra la regla especificada	Satisfactorio	Paso 1: insertar número de la regla.
		Descripción de la regla 2	Error en la regla 2	Seleccionar estructura	Muestra mensaje de error	Satisfactorio	Paso 2: insertar descripción de la regla.
	3		Error en la regla 3	Seleccionar estructura	Muestra mensaje de error	Satisfactorio	Paso 3: insertar mensaje de error.
	4	Descripción de la regla 4		Seleccionar estructura	Muestra mensaje de error	Satisfactorio	Paso 4: crear estructura.
	5	Descripción de la regla 5	Error en la regla 5		Muestra mensaje de error	Satisfactorio	

Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Flujo Alternativo	NA	NA	NA	NA	Muestra mensaje de error	Satisfactorio	Paso 1: insertar número de la regla. Paso 2: insertar descripción de la regla.
					Muestra mensaje de error	Satisfactorio	Paso 3: insertar Mensaje de error. Paso 4: crear estructura.

Tabla 7 Matriz de datos: Editar regla de validación

Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Flujo básico	6	Descripción de la regla 6	Error en la regla 6	Seleccionar estructura	Se muestra la regla especificada	Satisfactorio	Modificar el campo necesario
		Descripción de la regla 7	Error en la regla 7	Seleccionar estructura	Muestra mensaje de error	Satisfactorio	
	8		Error en la regla 8	Seleccionar estructura	Muestra mensaje de error	Satisfactorio	

Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
	9	Descripción de la regla 9		Seleccionar estructura	Muestra mensaje de error	Satisfactorio	
	10	Descripción de la regla 10	Error en la regla 10		Muestra mensaje de error	Satisfactorio	
Flujo Alternativo	NA	NA	NA	NA	Muestra mensaje de error	Satisfactorio	Modificar el campo necesario
					Muestra mensaje de error	Satisfactorio	

Tabla 8 Matriz de datos: Eliminar regla de validación

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central

Flujo básico	Se elimina la regla ya creada.	Satisfactorio	Clik derecho en la regla y eliminar
--------------	--------------------------------	---------------	-------------------------------------

### Validar Modelos

Descripción General.

El caso de uso inicia cuando el validador necesita validar un modelo ya creado. Se valida el modelo deseado y se obtiene el resultado correspondiente a la validación de dicho modelo finalizando así el caso de uso.

Condiciones de ejecución.

Deben estar creadas las reglas de validación correspondientes al modelo a validar.

**Tabla 9 Secciones a probar en al caso de uso Validar Modelos**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Validar Modelos	EC 1.1: Flujo básico	Valida el modelo seleccionado mediante las reglas anteriormente especificadas.
	EC 1.2: Flujo alterno.	Expresa inconformidad por no correspondencia entre el modelo a validar y las reglas especificadas para la validación de este.

**Tabla 10 Matriz de datos: Validar Modelos**

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Flujo básico	Valida el modelo	Satisfactorio	Seleccionar el modelo a validar y ejecutar botón validar.

### **Conclusiones**

En este capítulo se describió la implementación del sistema en términos de componentes, los cuales muestran sus dependencias a través de diagramas. Se describió el diagrama de despliegue como artefacto fundamental para la puesta en práctica del sistema, donde se especifica cada nodo necesario para el funcionamiento del mismo, y como parte del análisis para verificar el correcto funcionamiento del módulo desarrollado, se realizaron los casos de pruebas que cubren cada una de las funcionalidades especificadas.

## **Conclusiones Generales:**

Debido a la importancia que requiere la validación de los datos estadísticos generados en los modelos de PATDSI, y como parte de hacer más fácil y eficaz el desarrollo de esta tarea en la ONE, se hizo un análisis de la situación existente y se llegó a la conclusión de la necesidad del desarrollo de un módulo dentro de PATDSI que cubriera estas necesidades.

Como primeros pasos del desarrollo se analizaron las características del sistema, los métodos a utilizar, las discusiones necesarias para la selección de las herramientas, tecnologías, lenguajes y la metodología que se utilizaría, manteniendo las políticas del centro como base para esta selección.

Una vez realizadas estas selecciones se desarrolló el análisis y diseño del módulo donde a partir de los requisitos se realizaron los casos de uso que junto a una arquitectura adecuada, dieron lugar a una implementación satisfactoria y probada.

La utilización de este módulo contribuirá al mejor desarrollo del proceso de validación de los datos en los modelos estadísticos con la utilización de PATDSI en la ONE.

## **Recomendaciones:**

- Agregar funcionalidades para el trabajo con operaciones sobre rangos.
- Detallar los errores que se producen como resultado de la validación.

## Bibliografía:

1. **AulaFacil S.L.** AulaFacil.com. [En línea] 2010. [Citado el: 2 de febrero de 2010.] Disponible en: <http://www.aulafacil.com/investigacionsspss/Temario.htm>.
2. **Universidad de las Ciencias Informáticas.** Entorno Virtual de Aprendizaje. [En línea] 2009. [Citado el: 15 de febrero de 2010.] Disponible en: <http://eva.uci.cu/course/view.php?id=102>.
3. **epf.eclipse.org.** OpenUP/Basic. [En línea] 2006. [Citado el: 19 de enero de 2010.] Disponible en: <http://epf.eclipse.org/wikis/openupsp/>.
4. **Menéndez, Rosa.** Proyectos. [En línea] 2009. [Citado el: 26 de enero de 2010.] Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html#a>.
5. **Free Download Manager.** Free Download Manager. [En línea] 2010. [Citado el: 2 de febrero de 2010.] Disponible en: <http://www.freedownloadmanager.org/>.
6. **dat.etsit.upm.es.** dat.etsit.upm.es. [En línea] 2010. [Citado el: 2 de febrero de 2010.] <http://dat.etsit.upm.es/~abarbero/curso/xml/xmltutorial.html>.
7. **desarrolloweb.com.** desarrolloweb.com. [En línea] 2008. [Citado el: 26 de enero de 2010.] Disponible en: <http://www.desarrolloweb.com/articulos/pdt-eclipse-php.html>.
8. **Potencier, Fabien y Zaninotto, François.** *Symfony: La Guía definitiva.* 2008.
9. **Ext JS.** Ext JS. [En línea] 2010. [Citado el: 5 de marzo de 2010.] Disponible en: <http://www.extjs.com/>.
10. **PostgreSQL Global Development Group.** PostgreSQL. [En línea] 2010. [Citado el: 20 de enero de 2010.] Disponible en: <http://www.postgresql.org/>.

## **Glosario de Términos:**

**Regla:** objeto sobre el cual se define una expresión que especifica cómo debe ser el comportamiento de determinados elementos del modelo.

**Exp:** expresión matemática que puede utilizar operadores de suma, resta, multiplicación, división y resto de una división.

**Función:** responde a funciones definidas por el lenguaje que pueden ser: suma, producto, conteo, pero siempre sobre rangos específicos del modelo.

**Rango:** corresponde a una secuencia de datos del modelo y esta puede ser vista de forma vertical (desplazándose por las filas y manteniendo una columna) u horizontal (desplazándose por las columnas y manteniendo una fila).

**Literal:** responde a una constante numérica.

**Cond:** relación respecto a un literal y esta relación puede ser (mayor, menor, mayor o igual, menor o igual, igual o distinto).

**Fault:** mensaje de error que el usuario determina para que sea mostrado según la regla.