

Universidad de las Ciencias Informáticas

Facultad 6



**Título: “Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística”**

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

**Autores: Alex Brito de las Cuevas**

**Emilio Ruiz Rodríguez**

**Tutores: Ing. Clara Elena Brizuela Figueredo**

**Ing. Janier José Ramírez Landaburo**

**Ciudad de la Habana, Junio 2015**

**“Año del 55 Aniversario del Triunfo de la Revolución”**



*“Un hombre sólo tiene derecho a mirar a otro hacia abajo,  
cuando ha de ayudarlo a levantarse.”*

*Gabriel García Márquez.*

## Declaración de Auditoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2015.

Alex Brito de las Cuevas

Emilio Ruiz Rodríguez

---

Firma de Autor

---

Firma de Autor

Clara Elena Brizuela Figueredo

Janier José Ramírez Landaburo

---

Firma de Tutora

---

Firma de Tutor

## Datos de Contacto

**Tutora:** Ing. Clara Elena Brizuela Figueredo

Universidad de las Ciencias Informáticas, La Habana, Cuba

Especialidad de graduación: Ingeniería en Ciencias Informáticas

Correo Electrónico: [cebrizuela@uci.cu](mailto:cebrizuela@uci.cu)

**Tutor:** Ing. Janier José Ramírez Landaburo

Universidad de las Ciencias Informáticas, La Habana, Cuba

Especialidad de graduación: Ingeniería en Ciencias Informáticas

Correo Electrónico: [jjramirez@uci.cu](mailto:jjramirez@uci.cu)

**Autor:** Alex Brito de las Cuevas

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo Electrónico: [adelascuevas@estudiantes.uci.cu](mailto:adelascuevas@estudiantes.uci.cu)

**Autor:** Emilio Ruiz Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo Electrónico: [erodriguez@estudiantes.uci.cu](mailto:erodriguez@estudiantes.uci.cu)

## Agradecimientos

### De Alex:

A mi madre por ser mi luz y mi ejemplo a seguir.

A mi padre por ser mi guía y aconsejarme en todo momento.

A mi abuela Herlinda por consentirme tanto, pero por ser sobre todo la mejor abuela del mundo.

A mi abuelo Héctor por tenerme siempre a su lado y convertirme en su mejor creación.

A mis tutores Clara y Janier, porque sin ellos no estaría hoy donde estoy.

A Alejandro por ser un amigo, pero sobre todo un hermano.

A Anabel, Bidot y Andy por ser mis amigos más cercanos y más especiales.

A Emilio por saber ser el compañero y dúo de tesis más comprensivo y elocuente que se pueda pedir.

A todos mis compañeros de grupo, viejos y nuevos, gracias por hacer mejor mi estancia en la universidad.

A mis compañeros de apartamentos, Yoan, Alain, Daldis, Rolando, Frank, Yainel, Yasel, Ernesto, Bernardo, Luis Enrique, Fabiel; que siempre estuvieron en cada fiesta, pero también en cada bache.

Al piquete grande del café, Ernesto, Gabriel, Roberto, Albuerne y otros tantos; que me alegraron cada noche con su música y su compañía.

Al piquete del dominó, que tanta diversión compartimos en tardes y noches.

A todos los que están y a los que lamentablemente partieron, gracias por todo.

### **De Emilio:**

A mis padres Zoraida y Emilio por todo el amor y sacrificio que me han dedicado.

A mis hermanos Radier y Rolian que han sido siempre mi ejemplo a seguir y con quienes siempre podré contar.

A mi novia Maylen por darme su amor incondicional.

A mi cuñada Yely por todo el esfuerzo y cariño que me dedicó desde mis primeros años.

A mis amigos de crianza Jorgito y Luis Ángel que más que amigos son hermanos.

A mis tutores Clara y Janier pues sin su ayuda no estaríamos aquí.

A mis profesores.

A mis amigos y compañeros de aula en especial a Alejandro, Bernardo, Bidot, Fabiel y Yader

A mis hermanitas Tahimi, Laura y Anita.

A mi compañero de tesis Alex con el que he pasado buenos y malos momentos este año.

Gracias por todo.

## Dedicatoria

De Alex:

A mis padres, por su educación y su entrega incondicional.

A mis abuelos, por hacer posible cada día que me convierta en una mejor persona.

De Emilio:

A mis padres a los que les debo lo que soy.

Y en especial a mi papá que siempre creyó en mí y que sé le hubiera encantado estar aquí para verme convertido en ingeniero.

## **Resumen**

La Oficina Nacional de Estadística e Información (ONEI), realiza la gestión de los datos estadísticos en Cuba a través del Sistema Integrado de Gestión Estadística (SIGE) creado por la Universidad de las Ciencias Informáticas (UCI). El sistema brinda facilidades para el trabajo estadístico de la ONEI, no obstante la disciplina estadística, uno de los procesos fundamentales que realiza la ONEI en la actualidad no se encuentra informatizado. Esto provoca que se cometan errores a la hora de trabajar con los datos estadísticos además de elevarse cada vez más la gestión documental y el tiempo de realización del proceso. El objetivo de la investigación es desarrollar un módulo para SIGE que permita el control de la disciplina estadística en la ONEI. Como resultado se obtuvo una disminución del tiempo a la hora de realizar el proceso de disciplina estadística, reducción de la gestión documental y un mayor control del cumplimiento de los convenios informativos.

## **Palabras Claves**

Disciplina Estadística, Oficina Nacional de Estadística e Información, Sistema Integrado de Gestión Estadística.

## **Abstract**

The National Bureau of Statistics and Information (NBSI), makes the management of statistical data in Cuba through the Integrated Management System Statistics (IMSS) developed by the University of Informatics Science (UIS). The system provides facilities for statistical work of the ONEI, nonetheless the statistical discipline, one of the fundamental processes that performs the NBSI today is not computerized. Causing mistakes when working with statistical data in addition to soar ever more document management and the time of realization of the process. The objective of the research is to develop a module for IMSS which allows control of the statistical discipline at the NBSI. As a result a decrease in time to when performing statistical process discipline, reduction of document management and greater control over the compliance of informational conventions.

## **Keywords**

Integrated Management System Statistics, National Bureau of Statistics and Information, Statistical Discipline.

## Índice

Declaración de Auditoría.....	I
Datos de Contacto .....	II
Agradecimientos .....	III
Dedicatoria.....	V
Resumen .....	VI
Abstract .....	VII
Índice .....	VIII
Índice de Figura .....	XI
Índice de Tablas .....	XII
Introducción .....	1
Capítulo 1: Fundamentación Teórica .....	6
Introducción.....	6
1.1 Conceptos relacionados con la disciplina estadística .....	6
1.2 Sistemas informáticos existentes que sustentan la investigación .....	7
1.3 Metodología, tecnologías y herramientas a emplear en la solución .....	10
1.3.1 Metodología de desarrollo de software.....	10
1.3.2 Lenguaje de modelado.....	11
1.3.3 Herramienta CASE.....	11
1.3.4 Lenguajes de programación.....	12
1.3.5 Marco de trabajo .....	13
1.3.6 Entorno de desarrollo integrado .....	14
1.3.7 Sistema Gestor de Base de Datos (SGBD).....	14
1.3.8 Servidor de aplicaciones .....	15
1.4 Conclusiones del capítulo .....	16
Capítulo 2: Análisis y Diseño del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística .....	17
Introducción.....	17
2.1 Propuesta de solución .....	17

---

2.2 Modelo del dominio .....	17
2.2.1 Diagrama conceptual del Modelo de Dominio .....	18
2.2.2 Definición de clases del modelo del dominio .....	18
2.3. Especificación de los requisitos del sistema .....	19
2.3.1. Requisitos funcionales .....	19
2.3.2 Requisitos no funcionales .....	23
2.4. Modelo de Casos de Uso del Sistema .....	24
2.4.1 Diagrama de Casos de Uso del Sistema .....	25
2.4.2 Patrones de Casos de Uso utilizados .....	25
2.4.3 Descripción textual de los Casos de Uso del Sistema .....	26
2.5. Diseño del sistema .....	35
2.5.1 Modelo de diseño .....	36
2.5.2 Patrones de diseño utilizados en la solución .....	36
2.5.3 Diagrama de clases de diseño .....	40
2.5.4 Diagrama de secuencia .....	41
2.5.5 Modelo de datos .....	43
2.5.6 Modelo de despliegue .....	43
2.6 Conclusiones del capítulo .....	45
Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística .....	46
Introducción .....	46
3.1 Modelo de implementación .....	46
3.1.1 Diagrama de componentes .....	46
3.2 Código fuente .....	47
3.2.1 Estándares de codificación .....	47
3.2.2 Estándares de codificación utilizados .....	48
3.2.3 Ejemplo de código fuente .....	49
3.3 Pruebas de software .....	49
3.3.1 Prueba de integración .....	50
3.3.2 Prueba de validación .....	51

3.3.3 Prueba de aceptación .....	56
3.4 Conclusiones del capítulo.....	56
Conclusiones Generales .....	58
Recomendaciones .....	59
Referencias Bibliográficas.....	60
Anexo .....	64

## Índice de Figura

Fig. 1: Diagrama del Modelo de Dominio .....	18
Fig. 2: Diagrama de caso de uso del sistema.....	25
Fig. 3: Patrón Experto .....	37
Fig. 4: Patrón Creador (Fragmento de inicialización de un objeto) .....	38
Fig. 5: Patrón Controlador.....	39
Fig. 6: Patrón Fábrica .....	40
Fig. 7: Diagrama de clases de diseño del CU Gestionar Convenio Informativo .....	41
Fig. 8: Diagrama de Secuencia correspondiente al escenario Adicionar Convenio Informativo.....	42
Fig. 9: Diagrama Entidad-Relación .....	43
Fig. 10: Diagrama de Despliegue.....	44
Fig. 11: Diagrama de componentes del CU Gestionar Convenio .....	47
Fig. 12: Ejemplo de Código Fuente de la clase Importar Convenio .....	49
Fig. 13: Ejemplo de código fuente del método Crear Convenio.....	49
Fig. 14: Gráfica de iteraciones de pruebas funcionales .....	56

## Índice de Tablas

Tabla. 1: Descripción de usuario del sistema. ....	24
Tabla. 2: Descripción del Caso de Uso Gestionar Convenios Informativos. ....	26
Tabla. 3: Secciones de prueba para el caso de uso Gestionar Convenios. ....	52
Tabla. 4: Descripción de las variables del diseño de casos de prueba del CU Gestionar Convenios. ....	52
Tabla. 5: Valores válidos de variables del diseño de casos de prueba del CU Gestionar Convenios. ....	53
Tabla. 6: Caso de prueba Adicionar Convenio del CU Gestionar Convenios. ....	53
Tabla. 7: Caso de prueba Editar Convenio del CU Gestionar Convenios. ....	54

## **Introducción**

Desde sus inicios el significado de estadística estaba restringido a la información sobre los estados. Con el paso de los años se amplió para incluir a todas las colecciones de información de todo tipo, más tarde se amplió para incluir el análisis y la interpretación de estos datos. Hoy en día, los datos se recogen, las estadísticas se calculan, y son ampliamente distribuidas en el gobierno, los negocios, la mayoría de las ciencias, los deportes, e incluso para muchos pasatiempos. Con la ayuda de los ordenadores electrónicos se han acelerado los elaborados cálculos estadísticos, facilitando la recogida y agregación de datos. (University, 2015)

En Cuba para llevar a cabo este proceso se crea la Oficina Nacional de Estadísticas e Información (ONEI). La ONEI es la entidad creada para la aplicación de la política estatal en materia de estadística. (Rosada, 2013) Su misión fundamental es: “Garantizar la producción de estadísticas de calidad a través del Sistema Estadístico Nacional, ejerciendo una adecuada dirección, ejecución y control de la captación de las cifras económicas y sociales. Así como su difusión de acuerdo con los requerimientos de la economía y las demás necesidades del país en información estadística”. (ONEI, 2006)

Esta misión se fundamenta sobre los objetivos principales de la ONEI entre los cuales se encuentran aumentar la disponibilidad de recursos financieros para hacer frente a las necesidades vitales del trabajo, las tareas de desarrollo y alcanzar una ejecución más eficiente de los gastos. Así como garantizar la disciplina estadística y el cumplimiento de las regulaciones establecidas para la captación de las cifras por parte de las fuentes informantes que conforman el Sistema Estadístico Nacional y el Sistema de Estadística velando por la veracidad y calidad de la información. (ONEI, 2009)

La ONEI está estructurada en varios niveles, estos son la propia entidad, las provincias, los municipios y las unidades de observación en este orden. Las unidades de observación constituyen la principal fuente de información de la ONEI, las cuales están divididas entre unidades presupuestadas, empresas mixtas y cooperativas. A partir de los datos brindados por dichas entidades la ONEI, a través de sus diferentes niveles, realiza los procesos de captación, validación y consolidación que posibilitarán, como resultado, emitir la información oficial de apoyo al proceso de toma de decisiones gubernamentales.

Para garantizar la calidad, oportunidad, cantidad y relevancia de la información la ONEI establece convenios informativos con dichas unidades de observación, de forma tal que éstas se comprometen a brindar los datos correspondientes en el tiempo establecido. Estos convenios constan de un documento legal donde cada unidad de observación se compromete a entregar sus datos en tiempo. Además contiene los formularios establecidos por la ONEI para cada unidad de observación con sus indicadores estadísticos, y una ficha donde se representaran los indicadores que se van a reportar.

Cada unidad de observación es responsable de la entrega en tiempo de los indicadores estadísticos que le correspondan a la oficina territorial encargada de recibir los mismos. Este proceso se realiza siempre antes de la fecha pactada en el convenio informativo. Una vez cumplido el plazo de entrega la ONEI decide si es necesario que una unidad de observación arrastre los indicadores no entregados al siguiente mes por su importancia a la hora de la toma de decisiones. Es importante que se garantice que una vez que las unidades de observación han cumplido con su convenio los mecanismos del flujo interno de la ONEI funcionen para lograr el resultado informativo deseado. El éxito de estos dos procesos garantiza la disciplina estadística de dicha institución.

La ONEI en conjunto con la Universidad de las Ciencias Informáticas (UCI) llegó al consenso de crear un sistema informático para facilitar el trabajo con los datos estadísticos. A partir de este acuerdo surgió el Sistema Integrado de Gestión Estadística (SIGE), el cual brinda en gran medida facilidades para el trabajo estadístico como son la captación de información estadística a través de formularios matriciales y encuestas. Sin embargo no permite la gestión y control del cumplimiento de los convenios informativos que establece la ONEI con las unidades de observación, proceso denominado por la propia entidad como disciplina estadística.

Actualmente estos procesos se realizan de forma manual, provocando que se cometan errores al trabajar con los datos estadísticos, así como el aumento de la gestión documental y el tiempo de realización del proceso. Además no existe un modo de controlar en qué nivel de la estructura de la organización se incumplió a la hora de la entrega de los indicadores estadísticos; ya que puede existir retraso en algún nivel afectando al siguiente, el cual se detectaría una vez que la información llega a manos de la entidad.

A raíz de la problemática planteada, surge el siguiente **problema de la investigación** ¿Cómo contribuir al cumplimiento de la disciplina estadística en la Oficina Nacional de Estadística e Información?

Donde el **objeto de estudio** está dirigido al desarrollo de sistemas informáticos para el control de la disciplina estadística, enmarcado en el **campo de acción**: Control de la disciplina estadística en el Sistema Integrado de Gestión Estadística.

Para dar solución al problema de investigación planteado se define como **objetivo general**: Desarrollar un módulo para el Sistema Integrado de Gestión Estadística que permita el control de la disciplina estadística en la Oficina Nacional de Estadística e Información.

A partir del objetivo general se desglosan los siguientes **objetivos específicos**:

- Analizar los conceptos necesarios relacionados con los sistemas informáticos para el control de la disciplina estadística y las nuevas tecnologías.
- Realizar el análisis y diseño del módulo para guiar la implementación.
- Implementar el módulo de Control de la Disciplina Estadística para el Sistema Integrado de Gestión Estadística.
- Realizar pruebas al módulo de Control de la Disciplina Estadística para el Sistema Integrado de Gestión Estadística.

Para dar cumplimiento a cada uno de los objetivos específicos propuestos se definen las siguientes **tareas de investigación**:

- Identificación de los elementos teóricos fundamentales relacionados con los sistemas informáticos que permitan el control de la disciplina estadística para elaborar el marco teórico de la investigación.
- Análisis del procedimiento utilizado por la ONEI en el control de la disciplina estadística para comprender su funcionamiento.
- Selección de las herramientas y metodología a utilizar para realizar la implementación del módulo para el control de la disciplina estadística.
- Definición de los requisitos funcionales y no funcionales para el correcto funcionamiento del módulo

de control de la disciplina estadística.

- Diseño del módulo de control de la disciplina estadística para guiar el proceso de implementación.
- Implementación del módulo de control de la disciplina estadística para satisfacer necesidades funcionales identificadas.
- Diseño de los casos de pruebas para determinar el correcto funcionamiento de los requisitos.
- Realización de pruebas funcionales para comprobar el correcto funcionamiento del módulo de control de la disciplina estadística.

Dichas tareas de investigación se derivaron de las siguientes **preguntas de investigación**:

- ¿Cuáles son los fundamentos teóricos asociados al control de la disciplina estadística en la Oficina Nacional de Estadística e Información?
- ¿Qué funcionalidades debe tener el módulo para que permita el control de la disciplina estadística en la Oficina Nacional de Estadística e Información?
- ¿Qué beneficios aporta la informatización del control de la disciplina estadística en la Oficina Nacional de Estadística e Información?

Para el cumplimiento del objetivo general planteado se utilizarán los siguientes **métodos de investigación**:

1. **Analítico-Sintético:** se utiliza para realizar un estudio por separado de la información y la tecnología existente permitiendo la extracción de los elementos más importantes que se relacionan con el control y seguimiento de convenios, con el fin de elaborar una idea clara de lo que se quiere realizar para obtener una solución a la petición realizada por la ONEI. (Barchini, 2005)
2. **Modelación:** se crean abstracciones con el propósito de explicar la realidad. El modelo es el sustituto del objeto de investigación. (Barchini, 2005) Se realizar los diagramas correspondientes a los modelos de análisis, diseño e implementación del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística.
3. **La técnica de entrevista:** se realiza a miembros de la ONEI, con el objetivo de obtener información actualizada de los procesos que están informatizados y de las nuevas funcionalidades que se desean incorporar y mejorar; así como para definir los requisitos funcionales y no funcionales del mismo. Se

utilizó la entrevista no estructurada debido a que no se basa en un cuestionamiento rígido y es aplicado a personas que son especialistas en el tema del cual se está preguntando.

El Trabajo de Diploma está estructurado de la siguiente manera: Introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y bibliografía.

➤ Capítulo 1: Fundamentación Teórica del Sistema Integrado de Gestión Estadística.

En este capítulo se presenta la definición del marco teórico de la investigación. Se abordan varios conceptos ligados a la disciplina estadística. Características de algunos sistemas para la informatización y control de convenios, así como sistemas de gestión estadística existentes tanto en Cuba como en el resto del mundo. Se aborda el estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del módulo.

➤ Capítulo 2: Análisis y Diseño del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística.

En el capítulo se describe la propuesta de solución para el problema de la investigación. Mediante el modelo de dominio se describe la estructura que se desea representar en el módulo a desarrollar. Describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de Análisis y Diseño del módulo propuesto. Se establecen los requisitos funcionales y no funcionales para la obtención del diagrama de casos de uso. Se identifican los actores, casos de uso y las relaciones existentes entre ellos.

➤ Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística.

En el capítulo se aborda sobre las actividades que se llevan a cabo durante la fase de implementación y pruebas. Se analizará el Modelo de Implementación así como una descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes. Además se diseña y aplican las pruebas para comprobar el correcto funcionamiento del módulo.

## Capítulo 1: Fundamentación Teórica

### Introducción

En este capítulo se presenta la definición del marco teórico de la investigación. Se abordan varios conceptos ligados a la disciplina estadística. Se exponen características de algunos sistemas para la informatización y control de convenios, así como sistemas de gestión estadística existentes tanto en Cuba como en el resto del mundo. Se aborda el estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del módulo.

### 1.1 Conceptos relacionados con la disciplina estadística

**Estadística:** la estadística, en general, es la ciencia que trata de la recopilación, organización, presentación, análisis e interpretación de datos numéricos con el fin de realizar una toma de decisión más efectiva. (Muñoz, 2008) La ONEI realiza el trabajo con los datos estadísticos a través de SIGE, creado por la Universidad de las Ciencias Informáticas. Para realizar la captación de información la ONEI debe establecer convenios informativos con las unidades de observación asegurando la entrega de los datos requeridos.

**Convenio informativo:** un convenio informativo es el acuerdo de voluntades entre dos o más personas sobre cualquier punto en discusión o cuestión pendiente de resolver. (ABC, 2014) Estos convenios no son más que un documento legal en el cual la ONEI establece los formularios e indicadores que deben captar las unidades de observación, así como el período de tiempo para la entrega de estos. Una vez firmado el convenio es necesario que las unidades de observación cumplan lo establecido en este. De forma tal que se estaría efectuando una correcta disciplina en la organización.

**Disciplina:** es la capacidad que puede ser desarrollada por cualquier ser humano y que implica, para toda circunstancia u ocasión, la puesta en práctica de una actuación ordenada y perseverante, en orden a obtener un bien o fin determinado. (ABC, 2014) En este caso la disciplina no es más que la entrega en tiempo de todos los indicadores requeridos por la ONEI.

**Disciplina estadística:** proceso mediante el cual se controla la entrega en tiempo de los indicadores estadísticos que reportan las unidades de observación antes de una fecha indicada, teniéndose en cuenta que cada unidad de observación es responsable de la entrega de los indicadores; con los que se llevará a cabo la toma de decisiones a nivel nacional y se realizarán reportes mensuales que contengan la

información necesaria para el estudio estadístico y auditorías por parte de las distintas oficinas territoriales de la ONEI. (García, 2014)

## **1.2 Sistemas informáticos existentes que sustentan la investigación**

### **Sistema de monitoreo de convenios**

La Superintendencia de Informática del Centro Ejecutivo da Itaipu Binacional desarrolló un sistema de definición de panel de monitoreo que permite tener una visión integral, al instante, de todos los convenios que Itaipu lleva adelante. El mencionado sistema emite indicadores sobre el estado de cada uno de los convenios, posibilitando la detección de eventuales problemas que se registren en los mismos, propiciando su solución.

Con este sistema, los directivos de la entidad cuentan hoy con una herramienta que les permite tener en línea todas las informaciones sobre los convenios. Esto permite conversar con las áreas demandantes de los convenios, llámese gobernaciones, municipios y otras instituciones, y al tener las informaciones “en línea” a las que pueden acceder a través de tabletas o teléfonos inteligentes, les será más fácil definir con los mismos las nuevas acciones a ser tomadas.

El sistema permite identificar informaciones incorrectas que se corrigen inmediatamente, es decir, permite también ir detectando informaciones que fueron mal cargadas, las cuales se actualizan y corrigen para que la información que se presenta a los directores sea la más transparente y real posible. (Bogarín, 2014)

### **Sistema informático para el control estatal de la producción porcina no especializada en Cuba (SConv)**

El sistema informático SConv se ha creado para automatizar el control y la gestión estatal de la producción porcina, convenida entre el sector productivo especializado y el no especializado en Cuba. El sistema fue programado en Borland Delphi, con base de datos en MSAccess<sup>1</sup> para microcomputadora IBM<sup>2</sup> compatibles. SConv permite almacenar y procesar toda la información recopilada por medio de los servicios técnicos territoriales en el país. Esta se refiere a la producción contratada entre ambos sectores.

---

<sup>1</sup> Sistema de gestión de bases de datos incluido en el paquete de programas de Microsoft Office.

<sup>2</sup> International Business Machines Corporation.

El programa permite el seguimiento de cada uno de los convenios, de acuerdo con los parámetros establecidos para cada uno de ellos. Los informes de salida posibilitan hacer un resumen individual por convenios de los pesos y la carne en proceso, señalizan aquellos que no han cumplido y las causas del incumplimiento.

SConv brinda una panorámica general de este tipo de producción en todo el país o en grupos de convenios, escogidos según las necesidades de los usuarios (región, entidad, tipo, fecha de terminación, cumplimiento), además ofrecen las estadísticas generales de los mismos. El sistema constituye una herramienta en el manejo de la gestión de los convenios, lo que ayuda a tomar decisiones, a determinar las deficiencias en el proceso de producción y a crear acciones correctivas, con el propósito de elevar la eficiencia productiva y contribuir a la labor educativa del sector no especializado en Cuba. (SConv: Sistema informático para el control estatal de la producción porcina no especializada en Cuba, 2006)

SConv constituye una herramienta para los especialistas dedicados a esta actividad en el país. Se concluye que SConv es un sistema informático de gestión que contribuye a mejorar el control de los convenios entre el sector estatal y privado en Cuba. El sistema se validó en el instituto de investigaciones porcinas por el grupo que atiende los servicios técnicos territoriales. En estos momentos se encuentra en explotación, a nivel nacional, y a disposición de todas las empresas porcinas del país.

### **Sistema de información de convenios de cooperación educativa de la Universidad Politécnica de Catalunya (UPC)**

El sistema de información de convenios de cooperación educativa de la UPC es una aplicación web que se encarga de la gestión de las prácticas en empresas. Estas prácticas se llevan a cabo tanto desde los diferentes centros que realizan los convenios como por parte de la Oficina de Convenios del Servicio de Gestión Académica (SGA) que es quien gestiona la facturación.

Los beneficios que obtiene el cliente del desarrollo de este proyecto son sobre todo dos. Por un lado, la carga automática de convenios, de forma que se ahorran la introducción manual de los datos de los convenios. Y por otro lado, la apertura de la aplicación a los centros de la UPC permitirá a estos estar al día de las facturaciones vinculadas a sus convenios. (Girona, 2013)

La UPC, mediante su SGA, encargó al inLab<sup>3</sup> el desarrollo de un sistema informático para la gestión de los Convenios de Cooperación Educativa de la UPC. Por lo que este sistema actualmente ha sido desarrollado para el uso exclusivo de la UPC.

### **Sistema Integrado de Gestión Estadística (SIGE)**

SIGE es una aplicación de suma importancia para la Gestión de la Información Estadística de cualquier empresa o institución teniendo en cuenta que muestra el cumplimiento de las metas establecidas lo que contribuye a la toma de decisiones. Su capacidad de integración con otros sistemas, lo convierte en componente indispensable de cualquier sistema de información o software de gestión. Forma parte del Sistema de Información del Gobierno como herramienta principal en la automatización de los procesos de gestión de información.

Su objetivo fundamental es la gestión de información realizada mediante la captación de información estadística a nivel empresarial a través de formularios matriciales y encuestas. Los formularios son construcciones matriciales que relacionan indicadores. Los formularios matriciales se definen a partir de indicadores los cuales constituyen medidas verificables de cambio o resultado diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a metas establecidas.

SIGE fue desarrollado de conjunto con la ONEI de forma tal que contempla el marco metodológico y operativo de esta. Como sistema está integrado por módulos altamente especializados que permiten la gestión de las unidades de observación y la captación de la información estadística. Este sistema está compuesto por los siguientes módulos: Gestión de Configuración, Gestor de Plantillas, Entrada de Datos, Administración, Herramientas, Seguridad y Generador de Reportes, este último componente es externo al sistema y permite la visualización de la información almacenada.

Actualmente SIGE se encuentra en explotación en la ONEI así como en todas sus entidades a nivel nacional. El sistema permite la captura de información de forma digital y directa desde las unidades de observación así como también que estas validen la información de todos los formularios correspondientes al Sistema de Información de Estadística Nacional (SIEN). (Ing. Frank González Fernández, 2014)

---

<sup>3</sup>Laboratorio de innovación e investigación de la Facultad de Informática de Barcelona de la Universidad Politécnica de Catalunya - Barcelona Tech.

## **Conclusiones del estudio de los sistemas informáticos**

A raíz del análisis realizado sobre el control del cumplimiento de convenios se ha demostrado que existen varios sistemas informáticos que se adecuan a la solución requerida por la ONEI. Las soluciones como el Sistema de monitoreo de convenios no resulta viable para la ONEI pues no se cuenta con una infraestructura tecnológica que permita servicios de comunicación “en línea” para todos sus niveles. Por otra parte el Sistema de información de Convenios de Cooperación Educativa de la Universidad Politécnica de Catalunya fue desarrollado para ser usado solo por la UPC imposibilitando el acceso de la ONEI al mismo. En el caso del Sistema informático para el control estatal de la producción porcina no especializada en Cuba el problema que se identifica es que su negocio no se adecúa al que se realiza en la ONEI. Teniendo en cuenta que actualmente la ONEI realiza todas sus acciones sobre SIGE se ha decidido desarrollar un módulo para el control de la disciplina estadística en este sistema. El cual permitirá la gestión de convenios, monitorearlos y administrar los incumplimientos ocurridos.

## **1.3 Metodología, tecnologías y herramientas a emplear en la solución**

Es indispensable disponer de metodologías y herramientas que permitan valorar las mejoras introducidas y comparar el “estado de situación” antes y después de la introducción de los cambios. (Foundation, 2011) Para el desarrollo del módulo propuesto se utilizarán las tecnologías y herramientas definidas en el Sistema Integrado de Gestión Estadística. La selección del ambiente de desarrollo se ajusta a la Resolución No. 15 del 2011, donde se establece la migración a Código Abierto como objetivo principal para lograr la independencia y soberanía tecnológica de la nación cubana.

### **1.3.1 Metodología de desarrollo de software**

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Suele estar documentada y promovida por algún tipo de organización ya sea esta pública o privada. (Romero, 2009)

#### **OpenUP**

El Proceso Unificado Abierto u OpenUP (acrónimo en inglés de Open Unified Process) es una metodología de Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. Utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración para el desarrollo de software,

basada en RUP (Rational Unified Process), que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad. (Hernandez Barrios, 2014)

Se escoge esta metodología por ser apropiada para proyectos pequeños y de bajos recursos lo cual permite disminuir las probabilidades de fracaso. También por detectar errores tempranos a través de un ciclo iterativo y por ser una metodología ágil con un enfoque centrado al cliente y con iteraciones cortas.

### **1.3.2 Lenguaje de modelado**

El modelado es el diseño de las aplicaciones de software antes de codificar. El modelado es una parte esencial de los grandes proyectos de software, y útil para proyectos medianos e incluso pequeños. Un modelo juega el papel análogo en el desarrollo de software. (OMG, 2005)

### **UML 2.1**

El Lenguaje de Modelado Unificado o UML (acrónimo en inglés de Unified Modeling Language) es un lenguaje de modelado visual estándar destinado a ser utilizado para el modelado de procesos de negocio y similares; y para el análisis, diseño e implementación de sistemas basados en software. (uml-diagrams.org, 2009)

UML es un lenguaje común para los analistas, arquitectos y desarrolladores de software, utilizado para describir, especificar, diseñar nuevos procesos de negocio. (uml-diagrams.org, 2009) UML proporciona valiosa ayuda a los profesionales visualizando, comunicando y aplicando sus diseños. A lo largo de los años se ha visto evolucionar al UML logrando convertirse en una valiosa herramienta de prestigio y que ahorra mucho tiempo, llevándola a crecer y evolucionar a su edición 2.1.

Se utiliza este lenguaje de modelado por proporcionar orientación en cuanto al orden de las actividades de un equipo, dirigir las tareas de los desarrolladores individuales y del equipo en su conjunto y por ofrecer criterios para el seguimiento y la medición de los productos y las actividades de un proyecto.

### **1.3.3 Herramienta CASE**

Son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. (Alonso, 2011)

### **Visual Paradigm 8.0**

Visual Paradigm for UML Standard Edition (VP-UML SE) es una plataforma de modelado diseñado para apoyar a los arquitectos de sistemas, desarrolladores y diseñadores de UML. Además acelera el proceso de análisis y diseño de aplicaciones empresariales complejas que facilita la visualización UML en diferentes tipos de diagramas, así como diagramas entidad relación, diagramas de requisitos y el análisis textual. (softwaresea, 2009 )

Se escoge la herramienta Visual Paradigm por soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas.

### **1.3.4 Lenguajes de programación**

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. (ALEGSA, 2010)

#### **PHP 5**

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting'<sup>4</sup> de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas. (Achour, 2011)

#### **JavaScript**

JavaScript es un lenguaje de script multiplataforma orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos. (MDN, 2014)

El lenguaje PHP5 es seleccionado por ser simple para el desarrollo de aplicaciones web, pero que a su vez ofrece muchas características avanzadas para programadores expertos. Por otra parte se escoge el lenguaje JavaScript por su adaptabilidad y uso tanto desde el lado del cliente como del lado del servidor.

---

<sup>4</sup> Tipo de lenguaje de programación que es generalmente interpretado.

### **1.3.5 Marco de trabajo**

Un marco de trabajo o *framework* es una colección organizada de clases que constituyen un diseño reutilizable para un dominio específico de software. Contiene un conjunto de librerías, componentes de software y directrices arquitectónicas que ofrece al desarrollador un equipo de herramientas completo para construir una aplicación de principio a fin, siempre teniendo en cuenta que es necesario adaptarlo a cada aplicación en particular. (Multiplataforma, 2012)

#### **Symfony 1.1.7**

Symfony es un marco de trabajo de desarrollo web para PHP. Permite a los desarrolladores construir y mantener más fácil sitios web con PHP. (GITHUB, 2014) Está basado en la arquitectura Modelo-Vista-Controlador (MVC), surgió como un proyecto de software libre el cual se ha convertido en uno de los *frameworks* más populares de PHP que existe en la actualidad.

Symfony utiliza por defecto al Mapeo Objeto-Relacional (ORM) Propel que se encarga por defecto de gestionar el modelo. En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la base de datos. (Fabien Potencier, 2008)

El uso de Symfony en este proyecto garantiza rapidez y confiabilidad a la hora de la creación de productos usando PHP, gracias a su rápida respuesta y fácil manejo por parte del desarrollador. Es compatible con la mayoría de los gestores de bases de datos, como: PostgreSQL, Oracle, SQL Server de Microsoft y MySQL. Además las aplicaciones desarrolladas con él pueden ejecutarse en disímiles plataformas, como son Windows y sistemas Unix/Linux.

#### **ExtJS 3.4**

ExtJS es una librería Javascript que permite construir aplicaciones complejas en Internet. Esta librería incluye componentes de Interfaz de Usuario (UI) del alto performance y personalizables, modelo de componentes extensibles, una Interfaz de Programación de Aplicaciones (API) fácil de usar y licencias Open source y comerciales.

Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de *layouts* (composición) similar al que provee Java

Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari).

Además la ventana flotante que provee ExtJS al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros. (Rosas, 2008) El uso de este marco de trabajo permite que exista un balance entre cliente-servidor, una comunicación asíncrona y una mayor eficiencia en la red.

Como marcos de trabajo se utilizarán Symfony 1.1.7 y ExtJS 3.4 teniendo en cuenta que la programación del lado del servidor en SIGE se llevó a cabo con Symfony 1.1.7 debido a su destacado uso del patrón arquitectónico Modelo-Vista-Controlador que facilita la separación de estos elementos. En el caso de ExtJS 3.4 debido a que posee una base de componentes amplia que permite la reutilización y extensión de los mismos, así como también una vasta documentación y comunidad.

### **1.3.6 Entorno de desarrollo integrado**

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. (Biithahh, 2011)

#### **NetBeans 8.0**

NetBeans es un proyecto de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito para Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. (NETBEANS, 2011)

Este IDE de desarrollo se usa fundamentalmente por ser multilenguaje, pero además por ser de código abierto y gratuito. Por lo cual es la herramienta idónea para el desarrollo del módulo.

### **1.3.7 Sistema Gestor de Base de Datos (SGBD)**

Un sistema de gestión de base de datos o en inglés Database Management System (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos. (ALEGSA, 2010)

#### **PostgreSQL 9.1**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>5</sup> y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Martinez, 2010)

### **PgAdmin III**

PgAdmin cuenta con una rica administración en código abierto y desarrollo de plataforma para PostgreSQL. La aplicación puede utilizarse en sistemas operativos como Linux, FreeBSD, Solaris, Mac OSX y las plataformas de Windows para administrar PostgreSQL, así como las versiones comerciales y derivados de PostgreSQL como Postgres Plus Advanced Server y base de datos Greenplum.

Se decide utilizar el gestor de bases de datos PostgreSQL ya que es estable y permite un número ilimitado de bases de datos. Mientras que la herramienta PgAdmin es seleccionada por permitir el trabajo con las bases de datos de forma sencilla, brindando facilidades en la escritura de consultas y el manejo de las bases de datos.

#### **1.3.8 Servidor de aplicaciones**

Permite el procesamiento de datos de una aplicación cliente. Las principales ventajas de la tecnología de los servidores de aplicación es la centralización y la disminución de la complejidad del desarrollo de aplicaciones, dado que estas no necesitan ser programadas; en su lugar, estas son ensambladas desde bloques provistos por el servidor de aplicación. (ALEGSA, 2010)

### **Apache 2.2**

Apache, también conocido como Apache HTTP Server, es una norma establecida en la distribución en línea de servicios de sitios web. Se trata de una plataforma de servidor web de código abierto, lo que garantiza la disponibilidad en línea de la mayoría de los sitios web activos hoy día. El servidor está dirigido a servir a una gran cantidad de plataformas web que trabajan sobre sistemas operativos como Unix, Windows, Linux, Solaris, Novell NetWare, FreeBSD, Mac OS X, Microsoft Windows, OS / 2. (Hosting, 2013)

---

<sup>5</sup> Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution)

Se decide utilizar el servidor de aplicaciones Apache por ser un servidor web confiable que ofrece nuevos y más flexibles módulos de autenticación de usuario, el almacenamiento en caché del proxy, soporte para archivos superiores a 2 GB, así como el apoyo de SQL.

### **1.4 Conclusiones del capítulo**

En el capítulo que finaliza se analizaron algunas herramientas existentes en la actualidad que realizan trabajos con datos estadísticos y auditorios, las cuales son la base de apoyo para la realización del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística. Se seleccionó como metodología de desarrollo de software OpenUP, la cual guiará todo el proceso de creación del sistema propuesto. Se hizo un estudio y selección de las herramientas y tecnologías que serán usadas durante el proceso de desarrollo del módulo. Seleccionando UML 2.1 como lenguaje de modelado y Visual Paradigm 8.0 como herramienta CASE. Como lenguajes de programación PHP 5 y JavaScript, Symfony 1.1.7 y ExtJS 3.4 como *frameworks* de desarrollo, NetBeans 8.0 como IDE de desarrollo. Para el tratamiento de las bases de datos PostgreSQL 9.1 y pgAdmin III, y como servidor de aplicaciones Apache 2.2. Cada una de estas herramientas y tecnologías posee características que las hacen idóneas para el trabajo con ellas durante el proceso de desarrollo del módulo.

## **Capítulo 2: Análisis y Diseño del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística**

### **Introducción**

En el capítulo se describe la propuesta de solución para el problema de la investigación. Mediante el modelo de dominio se describe la estructura que se desea representar en el módulo a desarrollar. Describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de Análisis y Diseño del módulo propuesto. Se establecen los requisitos funcionales y no funcionales para la obtención del diagrama de casos de uso. Se identifican los actores, casos de uso y las relaciones existentes entre ellos.

### **2.1 Propuesta de solución**

Se propone el desarrollo de un Módulo para el Control de la Disciplina Estadística propuesto por la ONEI. El Módulo permitirá el control del cumplimiento de los convenios informativos de las unidades de observación con la ONEI. Al igual que el correcto flujo informativo entre los diferentes niveles de la institución a partir de la automatización de los procesos de gestión y control de los convenios informativos, así como el seguimiento de la entrega de información a los diferentes niveles. Con la creación del Módulo se logrará una reducción de tiempo al establecer convenios informativos, disminución de la gestión documental derivada del proceso de disciplina estadística y un mejor análisis de las responsabilidades de las unidades de observación. Además de un mejor entendimiento de los incumplimientos por parte de las unidades de observación y de las oficinas municipales y provinciales de la ONEI, en cuanto a la entrega en tiempo de los indicadores establecidos en los convenios informativos.

### **2.2 Modelo del dominio**

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo.

Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (synergix, 2008)

### 2.2.1 Diagrama conceptual del Modelo de Dominio

El siguiente modelo de dominio tiene como objetivo contribuir a la comprensión del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística.

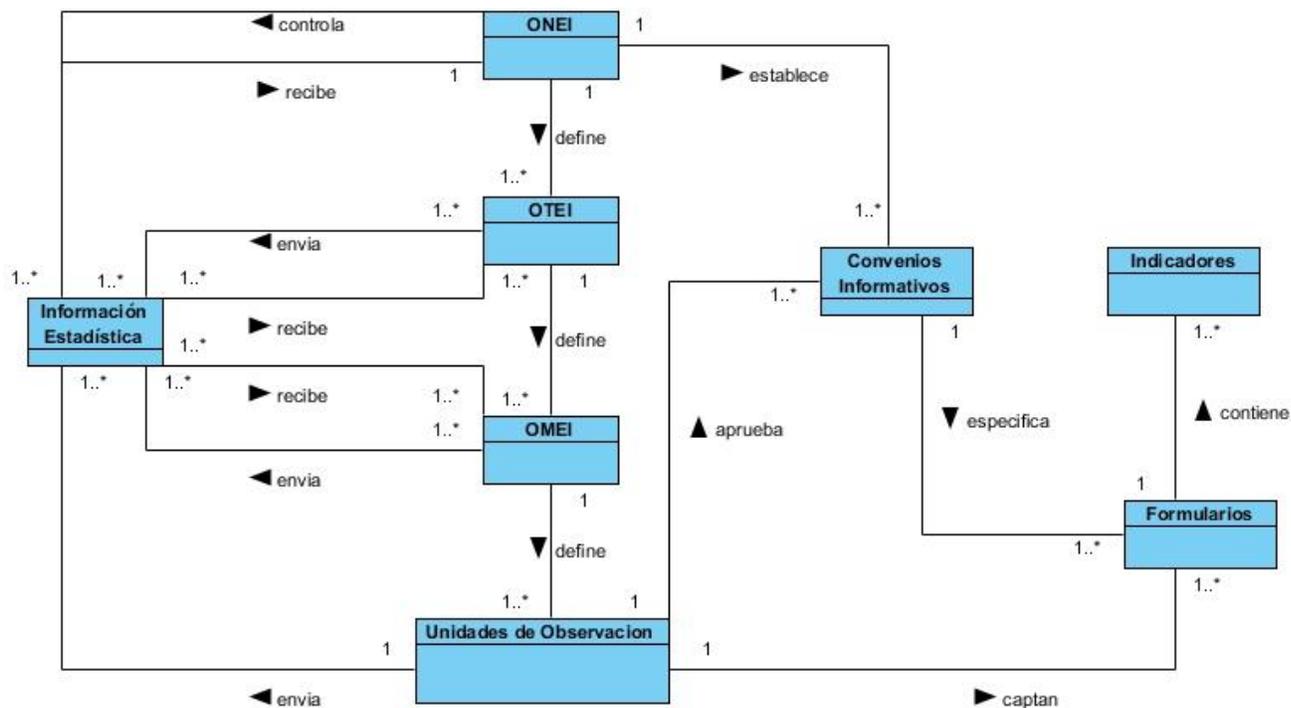


Fig. 1: Diagrama del Modelo de Dominio

### 2.2.2 Definición de clases del modelo del dominio

**ONEI:** Oficina Nacional de Estadística e Información. A través de sus diferentes niveles realiza los procesos de captación, validación y consolidación que posibilitarán, como resultado, emitir la información oficial de apoyo al proceso de toma de decisiones gubernamentales. Además lleva a cabo el control de la información estadística recibida asegurando que esta se entregue en tiempo y sin errores.

**OTEI:** Oficina Territorial de Estadística e Información. Estructura que organiza el ámbito de la Oficina Nacional de Estadísticas en cuanto a la captación de la información.

**OMEI:** Oficina Municipal de Estadística e Información. Estructura que organiza el ámbito de la Oficina Territorial de Estadística en cuanto a la captación de la información.

**Unidades de Observación:** constituyen la principal fuente de información de la ONEI, las cuales suman más de 6000 en todo el país entre unidades presupuestadas, empresas mixtas y cooperativas.

**Convenios informativos:** documento legal donde cada unidad de observación se compromete a entregar sus datos en tiempo. Además contiene los formularios establecidos por la ONEI para cada unidad de observación con sus indicadores estadísticos.

**Formularios:** los formularios constituyen la principal forma de captura de datos estadísticos numéricos utilizada por la ONEI. Son construcciones matriciales que relacionan indicadores y aspectos.

**Indicadores:** medidas verificables de cambio o resultado diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a metas establecidas.

**Información estadística:** análisis provenientes de una muestra representativa de datos, utilizada para la toma de decisiones en áreas de negocios o instituciones gubernamentales.

### 2.3. Especificación de los requisitos del sistema

Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación. Se pueden clasificar en: funcionales y no funcionales. Partiendo de la descripción de las clases representadas en el Modelo de Dominio y las necesidades del cliente, fueron determinados los requisitos funcionales y no funcionales del sistema.

#### 2.3.1. Requisitos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (scribd, 2015) Se identificaron los siguientes requisitos funcionales:

##### **RF1: Adicionar convenio**

**Descripción:** el sistema debe permitir adicionar un nuevo convenio informativo.

**Entrada:** se especifican los detalles del nuevo convenio informativo que se va a adicionar (Código del convenio, Representante, Cargo que ocupa, Provincia, Municipio, Dirección, Resolución No, Año de

Resolución, Correo, Fecha de Creación, Fecha de Entrega, Nombre del Centro Informante, Código del Centro Informante, Dirección del Centro Informante, Representante del Centro Informante, Resolución No, Año de Resolución, Correo).

### **RF2: Editar convenio**

**Descripción:** el sistema debe permitir editar un convenio informativo previamente seleccionado.

**Entrada:** se modifican los detalles del convenio informativo seleccionado (Código del convenio, Representante, Cargo que ocupa, Provincia, Municipio, Dirección, Resolución No, Año de Resolución, Correo, Fecha de Creación, Fecha de Entrega, Nombre del Centro Informante, Código del Centro Informante, Dirección del Centro Informante, Representante del Centro Informante, Resolución No, Año de Resolución, Correo).

### **RF3: Eliminar convenio**

**Descripción:** el sistema debe permitir eliminar un convenio informativo previamente seleccionado.

**Entrada:** selección del convenio que se desea eliminar.

### **RF4: Buscar convenio**

**Descripción:** el sistema debe permitir buscar un convenio informativo existente a partir de un criterio de búsqueda.

**Entrada:** los criterios de búsqueda pueden ser: Código del convenio, Representante, Cargo que ocupa, Provincia, Municipio, Dirección, Resolución No, Año de Resolución, Correo, Fecha de Creación, Fecha de Entrega, Nombre del Centro Informante, Código del Centro Informante, Dirección del Centro Informante, Representante del Centro Informante, Resolución No, Año de Resolución, Correo.

**Salida:** listado de convenios informativos que cumplan con el parámetro especificado.

### **RF5: Exportar convenio a XML**

**Descripción:** el sistema debe permitir la exportación del convenio informativo al formato XML.

**Entrada:** selección del convenio que se desea exportar.

**Salida:** archivo XML.

### **RF6: Importar convenio**

**Descripción:** el sistema debe permitir la importación de convenios informativos en formato XML.

**Entrada:** selección del archivo XML.

### **RF7: Listar convenios**

**Descripción:** el sistema debe permitir listar los convenios informativos.

**Salida:** listado de los convenios informativos.

### **RF8: Adicionar plantilla de formulario**

**Descripción:** el sistema debe permitir seleccionar los formularios que se establecerán en el convenio así como los indicadores que se necesiten del mismo. La selección de estos formularios e indicadores es opcional en caso de que las requiera el convenio informativo.

**Entrada:** las selecciones realizadas por el usuario.

### **RF9: Mostrar formato de convenio**

**Descripción:** muestra toda la información que contendrá el convenio informativo al usuario.

**Entrada:** selección del convenio del cual el usuario desea ver su información.

**Salida:** información contenida en el convenio seleccionado por el usuario.

### **RF10: Exportar convenio a PDF**

**Descripción:** una vez mostrado el formato del convenio el sistema debe permitir la exportación del convenio informativo al formato PDF.

**Salida:** archivo PDF.

### **RF11: Denegar permiso**

**Descripción:** el sistema debe permitir denegar el permiso de verificar cumplimiento del convenio a una UO, una vez que se muestre el mensaje de que la UO ha incumplido con el convenio informativo.

**Entrada:** selección de la unidad de observación a la cual se le denegará el permiso.

### **RF12: Aprobar permiso**

**Descripción:** el sistema debe permitir aprobar el permiso a una UO de verificar el cumplimiento cuando se ha excedido con el período de tiempo establecido en el convenio informativo.

**Entrada:** selección de la unidad de observación a la cual se le aprobará el permiso.

### **RF13: Mostrar mensaje**

**Descripción:** el sistema debe mostrar un mensaje informando sobre el estado en que se encuentra el convenio, así como los días restantes que le quedan a la unidad de observación para la entrega de la información solicitada en el convenio informativo.

**Entrada:** selección de la unidad de observación de la que se desea conocer su estado.

**Salida:** mensaje.

### **RF14: Verificar cumplimiento**

**Descripción:** el sistema debe comprobar que la información recibida corresponda con lo establecido en el convenio informativo. Para esto se debe mostrar una ventana dividida en dos por una parte los formularios

e indicadores establecidos en el convenio informativo y por la otra los datos enviados por las unidades de observación de manera que exista una correspondencia entre estos.

**Entrada:** selección de la unidad de observación de la cual se verificará su cumplimiento. Importación del modelo.

### **RF15: Enviar correo**

**Descripción:** el sistema debe permitir el envío de un correo en caso de que ocurra el incumplimiento de un convenio informativo, en el que se exigirá una respuesta del por qué se ha incumplido.

**Entrada:** se especifican los detalles del correo (Asunto, Cuerpo del correo).

### **RF16: Generar gráfica**

**Descripción:** el sistema debe permitir la generación de un gráfico de pastel que represente el cumplimiento de los convenios informativos en el sistema.

**Salida:** generación del grafico de pastel.

### **RF17: Actualizar estado**

**Descripción:** el sistema debe permitir modificar el estado en que se encuentra el convenio informativo.

**Entrada:** selección del tipo de estado en que se encuentra el convenio informativo.

### **RF18: Editar incumplimiento**

**Descripción:** el sistema debe permitir editar un incumplimiento previamente seleccionado.

**Entrada:** se modifica la selección del tipo de incumplimiento llevado a cabo en el convenio informativo así como la descripción de sus causas.

### **RF19: Eliminar incumplimiento**

**Descripción:** el sistema debe permitir eliminar un incumplimiento previamente seleccionado.

**Entrada:** Selección del incumplimiento que se desea eliminar.

### **RF20: Buscar incumplimiento**

**Descripción:** el sistema debe permitir buscar un convenio informativo existente a partir de un criterio de búsqueda.

**Entrada:** los criterios de búsqueda serian: Código del convenio, Nombre del Centro Informante, Tipo de incumplimiento.

**Salida:** listado de convenios informativos con sus respectivos incumplimientos que cumplan con el parámetro especificado.

### 2.3.2 Requisitos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. (scribd, 2015) Se identificaron los siguientes requisitos no funcionales:

#### Restricciones del diseño:

**RNF1:** lenguaje y marco de trabajo (*framework*) para el desarrollo del sistema del lado del servidor.

El módulo deberá ser implementado en el lenguaje de programación PHP en su versión 5 o superior. Como *framework* de desarrollo se usará Symfony 1.1.7 el cual propone una arquitectura modular en tres capas: el modelo, la vista y el controlador.

**RNF2:** lenguaje y marco de trabajo para el desarrollo del sistema del lado del cliente.

Una de las bibliotecas fundamentales en el desarrollo de la herramienta es la de ExtJS 3.4 la cual permite el diseño de interfaces visuales interactivas usando metodologías como AJAX y permiten crear aplicaciones WEB con la apariencia de escritorio.

#### Interfaz:

**RNF3:** interfaces de usuario.

Las interfaces de usuario serán diseñadas a modo de aplicaciones RIA (Rich Internet Application) lo que permite a los usuarios contar con aplicaciones web muy similares a las aplicaciones de escritorios. Para lograr este objetivo se usará la librería Javascript, ExtJS, la cual conjuga una serie de componentes visuales que proveen funcionalidades que ayudan al diseño de este tipo de aplicaciones WEB con apariencia de escritorio.

**RNF4:** interfaces de Hardware.

Cliente

- Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- Memoria RAM mínimo 256MB.

Servidor

- Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- Memoria RAM mínimo 1GB.

**RNF5:** interfaces de Software.

Un servidor en el que se instale únicamente SIGE con los siguientes elementos.

- Sistema operativo GNU/Linux Ubuntu 10.10 o versión superior.
- Servidor de aplicaciones Apache, versión 2.
- Lenguaje de programación PHP y librerías versión 5.
- Sistemas Gestor de Bases de Datos PostgreSQL, versión 8.4.
- En los clientes Mozilla Firefox 3.4 o superior.

### **Confiabilidad:**

**RNF6:** la aplicación debe permanecer en línea en los horarios de trabajo establecidos por las entidades, excepto cuando sea necesario reiniciarlo o detenerlo por tareas de mantenimiento o cambio de configuración.

## **2.4. Modelo de Casos de Uso del Sistema**

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema, donde cada caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. (Sparx, 2007) El modelo de casos de uso está compuesto por actores, casos de uso y la relación que existe entre ellos.

### **Casos de Uso**

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. (Software, 2014)

### **Actores del sistema**

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema. (Software, 2014)

**Tabla. 1: Descripción de usuario del sistema.**

Actor	Descripción
-------	-------------

Usuario	El usuario es cualquier persona autorizada a utilizar el sistema para trabajar en el módulo. Varía en dependencia de los niveles en los que está estructurada la ONEI.
---------	---

### 2.4.1 Diagrama de Casos de Uso del Sistema

Los CU son tareas específicas que se realizan a raíz de la orden de un actor. Un diagrama de casos de uso del sistema representa gráficamente a las funciones del sistema y su interacción con los actores. (Software, 2014)

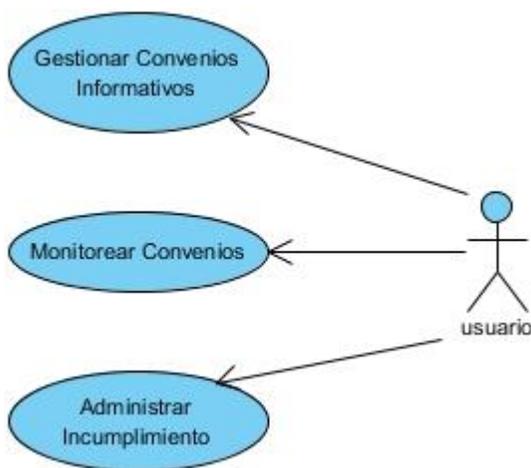


Fig. 2: Diagrama de caso de uso del sistema

### 2.4.2 Patrones de Casos de Uso utilizados

Al desarrollar un software es muy útil apoyarse en la experiencia anterior, ya sea de uno mismo o de otros. De esta manera se sabe que la solución va a funcionar, y tendremos identificados los problemas potenciales, así como soluciones para éstos. Estas soluciones a problemas comunes se conocen como patrones, y se utilizan en diversos aspectos del desarrollo de software. Los Patrones de Casos de Uso ayudan a describir el uso del sistema y su interacción con los usuarios. (Rodríguez, 2014)

#### CRUD Completo

Su nombre es un acrónimo de las palabras en inglés Create, Read, Update, Delete. El patrón CRUD Completo consiste en un caso de uso para administrar la información (CRUD Información), nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer,

cambiar y dar de baja. Este patrón es usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples. (Rodríguez, 2014) Ejemplo de ello se evidencia en el caso de uso Gestionar Convenios Informativos representado en el diagrama de caso de uso del sistema.

### CRUD Parcial

Este patrón es usado cuando alguna de las alternativas del caso de uso es más significativo, muy largo, o mucho más complejo que el patrón completo. (Rodríguez, 2014) Ejemplo de ello se evidencia en el caso de uso Administrar Incumplimientos representado en el diagrama de caso de uso del sistema.

### 2.4.3 Descripción textual de los Casos de Uso del Sistema

Un caso de uso cuenta una historia estilizada de la manera que un usuario final interactúa con el sistema en un conjunto específico de circunstancias. A continuación se muestra un ejemplo:

Tabla. 2: Descripción del Caso de Uso Gestionar Convenios Informativos.

<b>Objetivo</b>	Manipular los convenios informativos
<b>Actores</b>	Usuario (Inicia)
<b>Resumen</b>	El CU se inicia cuando el Usuario va a realizar algunas de las siguientes operaciones: Crear nuevo convenio, Modificar, Eliminar convenio, Buscar convenio, Importar convenio, Exportar a XML, Mostrar formato de convenio, Exportar a PDF, Adicionar Plantilla de Formulario.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	Que el Usuario esté autenticado en el sistema.
<b>Postcondiciones</b>	En dependencia de la acción del Usuario:  Se adiciona un nuevo convenio.  Se modifica un convenio existente.  Se elimina un convenio existente.

	<p>Se busca un convenio existente según un criterio de búsqueda.</p> <p>Se importa un convenio.</p> <p>Se exporta un convenio existente a formato XML.</p> <p>Se muestra el formato de un convenio existente.</p> <p>Se exporta un convenio existente a formato PDF.</p> <p>Se adiciona una plantilla de formulario a un convenio existente.</p>	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso inicia cuando el Usuario selecciona la pestaña Disciplina Estadística y accede a la opción Actualizar Convenios del menú Inicio.	
2.		<p>El sistema muestra el listado de los convenios informativos y activa las opciones correspondientes a la gestión de convenios informativos. El Usuario puede:</p> <p>a. Adicionar nuevo convenio: ir a la sección “Adicionar”</p> <p>b. Modificar datos de convenio: ir a la sección “Editar”</p> <p>c. Eliminar convenio: ir a la sección “Eliminar”</p> <p>d. Buscar convenio: ir a la sección “Buscar”</p> <p>e. Importar convenio: ir a la sección “Importar”</p> <p>f. Exportar convenio: ir a la sección “Exportar a XML”</p>

		<p>g. Mostrar formato de convenio: ir a la sección “Mostrar formato de convenio”</p> <p>h. Exportar a PDF: ir a la sección “Exportar a PDF”</p> <p>i. Adicionar plantilla de formulario: ir a la sección “Adicionar Plantilla de Formulario”</p>
<b>Sección 1: “Adicionar Convenio”</b>		
<b>Flujo básico &lt; Adicionar Convenio &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El Usuario selecciona la opción Adicionar.	
2.		<p>El sistema muestra la interfaz para crear un nuevo convenio informativo, solicitando los siguientes datos:</p> <p>Código del convenio</p> <p>Representante</p> <p>Cargo que ocupa</p> <p>Provincia</p> <p>Municipio</p> <p>Dirección</p> <p>Resolución No</p> <p>Año de Resolución</p> <p>Correo</p> <p>Fecha de Creación</p>

		<p>Fecha de Entrega</p> <p>Nombre del Centro Informante</p> <p>Código del Centro Informante</p> <p>Dirección</p> <p>Representante</p> <p>Resolución No</p> <p>Año de Resolución</p> <p>Correo</p>
3.	El Usuario introduce los datos correspondientes.	
4.		El sistema valida los datos introducidos.
5.	El Usuario selecciona la opción aceptar.	
6.		El sistema registra el nuevo convenio, recarga el listado de convenios y muestra el mensaje convenio adicionado correctamente, finaliza el CU.
<b>Flujos alternos</b>		
<b>Nº 4.a &lt;Validación&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		El sistema muestra los campos con errores de validación subrayados en rojo.
2.	El Usuario se posiciona sobre el campo que tiene errores en los datos.	
3.		El sistema muestra el mensaje de error de acuerdo al campo donde se posicione el Usuario.

<b>Flujos alternos</b>		
<b>Nº 5.a &lt;Cancelar&gt;</b>		
1.	Si el Usuario indica cancelar, se cancela la operación.	
2.		El sistema cierra la interfaz correspondiente y finaliza el CU.
<b>Sección 2: “Modificar Convenio”</b>		
<b>Flujo básico &lt; Modificar Convenio &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El Usuario selecciona un convenio del listado de convenios que se muestra y selecciona la opción Editar.	
2.		El sistema muestra una interfaz con los datos del convenio seleccionado previamente.
3.	El Usuario modifica los datos del convenio que desea actualizar de la interfaz mostrada previamente por el sistema.	
4.		El sistema valida los campos editados.
5.	El Usuario selecciona la opción aceptar.	
6.		El sistema actualiza el convenio, recarga el listado para mostrar la actualización realizada y muestra el mensaje convenio actualizado correctamente, finalizando el CU.
<b>Flujos alternos</b>		
<b>Nº 4.a &lt;Validación&gt;</b>		

## Capítulo 2: Análisis y Diseño del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística

---

1.		El sistema muestra los campos con errores de validación subrayados en rojo.
2.	El Usuario se posiciona sobre el campo que tiene errores en los datos.	
3.		El sistema muestra el mensaje de error de acuerdo al campo donde se posicione el Usuario.

### Flujos alternos

#### Nº 5.a < Cancelar >

1.	Si el Usuario indica cancelar los cambios, se cancela la operación y finaliza el CU.	
2.		El sistema cierra la interfaz correspondiente y finaliza el CU.

### Sección 3: “Eliminar Convenio”

#### Flujo básico < Eliminar Convenio >

	Actor	Sistema
1.	El Usuario selecciona un convenio del listado de convenios que se muestra y selecciona la opción Eliminar.	
2.		El sistema brinda una alerta: “¿Esta seguro que desea eliminar el Convenio Informativo seleccionado?”.
3.	El Usuario selecciona la opción Sí.	
4.		El sistema elimina el convenio, recarga el listado de convenios y muestra el mensaje convenio eliminado correctamente, finalizando el CU.

<b>Flujos alternos</b>		
<b>Nº 3.a &lt;Selección No&gt;</b>		
1.	Si el Usuario selecciona la opción No se cancela la operación.	
2.		El sistema cierra la alerta y finaliza el CU.
<b>Sección 4: “Buscar Convenio”</b>		
<b>Flujo básico &lt; Buscar Convenio &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		El sistema muestra una interfaz para introducir el criterio de búsqueda.
2.	El Usuario escribe un criterio de búsqueda y selecciona la opción Buscar.	
3.		El sistema muestra el o los convenios que se relacionan con el criterio introducido por el Usuario, finalizando el CU.
<b>Flujos alternos</b>		
<b>Nº 3.a &lt;No existe convenio que coincida con el criterio de búsqueda&gt;</b>		
		El sistema muestra un mensaje “No existe convenio con el criterio especificado”
<b>Sección 5: “Exportar Convenio a XML”</b>		
<b>Flujo básico &lt; Exportar Convenio a XML &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El Usuario selecciona un convenio del listado de convenios que se muestra y selecciona la opción Exportar a XML.	

## Capítulo 2: Análisis y Diseño del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística

---

2.		El sistema brinda la posibilidad de guardar el convenio a XML.
3.	El Usuario selecciona Aceptar.	
4.		El sistema exporta el convenio, finalizando el CU.

### Flujos alternos

#### Nº 3.a <Selección Cancelar>

1.	Si el Usuario indica cancelar se cancela la operación.	
2.		El sistema cierra la interfaz correspondiente y finaliza el CU.

### Sección 6: “Importar Convenio”

#### Flujo básico < Importar Convenio >

	Actor	Sistema
1.	El Usuario selecciona la opción Importar.	
2.		El sistema muestra una interfaz posibilitando que el Usuario seleccione el convenio que desea importar.
3.	El Usuario selecciona el convenio.	
4.	El Usuario selecciona aceptar.	
5.		El sistema importa el convenio, recarga el listado de convenios y muestra el mensaje convenio importado correctamente, finalizando el CU.

### Flujos alternos

#### Nº 4.a <Selección cancelar>

1.	Si el Usuario indica cancelar se cancela la operación.	
----	--	--

2.		El sistema cierra la interfaz correspondiente y finaliza el CU.
----	--	---

**Sección 7: “Mostrar formato de Convenio”**

**Flujo básico < Mostrar formato de Convenio >**

	<b>Actor</b>	<b>Sistema</b>
1.	El Usuario selecciona un convenio del listado de convenios que se muestra y selecciona la opción Mostrar formato de Convenio.	
2.		El sistema muestra una interfaz mostrando el formato del convenio seleccionado, finaliza el CU.

**Sección 8: “Exportar Convenio a PDF”**

**Flujo básico < Exportar Convenio a PDF >**

	<b>Actor</b>	<b>Sistema</b>
1.		Una vez que el sistema muestra la interfaz mostrando el formato de un convenio permite realizar la operación de Exportar a PDF.
2.	El Usuario selecciona la opción Exportar a PDF.	
3.		El sistema exporta el convenio, finalizando el CU.

**Sección 9: “Adicionar Plantilla de Formulario”**

**Flujo básico < Adicionar Plantilla de Formulario >**

	<b>Actor</b>	<b>Sistema</b>
1.	El Usuario selecciona un convenio del listado de convenios que se muestra y selecciona la opción Adicionar Plantilla de Formulario.	

2.		El sistema muestra la interfaz para seleccionar los formularios e indicadores que requiere el convenio informativo.
3.	El Usuario selecciona los formularios e indicadores.	
4.	El Usuario selecciona aceptar.	
5.		El sistema registra los formularios e indicadores en el convenio y muestra el mensaje formularios adicionados correctamente, finaliza el CU.
<b>Flujos alternos</b>		
<b>Nº 4.a &lt;Selección cancelar&gt;</b>		
1.	Si el Usuario indica cancelar se cancela la operación.	
2.		El sistema cierra la interfaz correspondiente y finaliza el CU.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede
	<b>CU Extendidos</b>	No procede
<b>Requisitos no funcionales</b>	<b>RNF: Seguridad:</b> Al sistema se accederá utilizando un protocolo seguro (https).	
<b>Asuntos pendientes</b>	No procede	

## 2.5. Diseño del sistema

El diseño del sistema es el arte de definir la arquitectura del hardware y software, componentes, módulos y datos de un sistema de cómputo para satisfacer ciertos requerimientos. Donde estos requerimientos son el conjunto de actividades encaminadas a obtener las características necesarias que deberá poseer el nuevo sistema, para comprender cómo trabaja y dónde es necesario efectuar mejoras o cambios considerables. (Corporation, 2015)

### 2.5.1 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. El modelo de diseño es esencialmente el diseño mismo. Es el puente entre los requerimientos y la implementación del sistema.

### 2.5.2 Patrones de diseño utilizados en la solución

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. Aunque nuestra aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas. En lugar de reinventar la rueda, podemos solucionar problemas utilizando algún patrón, ya que son soluciones probadas y documentadas por multitud de programadores. (Genbeta, 2014)

#### Patrón arquitectónico

El módulo se desarrollará en Symfony lo que implica que se utilice el patrón arquitectónico Modelo-Vista-Controlador (MVC). Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

**El Modelo** representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

**La Vista** transforma el modelo en una página web que permite al usuario interactuar con ella.

**El Controlador** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Potencier, 2015)

#### Patrones GRASP

Los patrones generales de software para asignar responsabilidades o GRASP (acrónimo en inglés de General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

### Experto:

Asignar una responsabilidad al experto en información, la clase que tiene la información necesaria para realizar la responsabilidad. Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades. Propel como ORM crea cuatro clases: dos clases de acceso a datos que trabajan directamente con la base de datos y dos de abstracción de datos que son las que tienen los atributos necesarios para realizar dicha función.

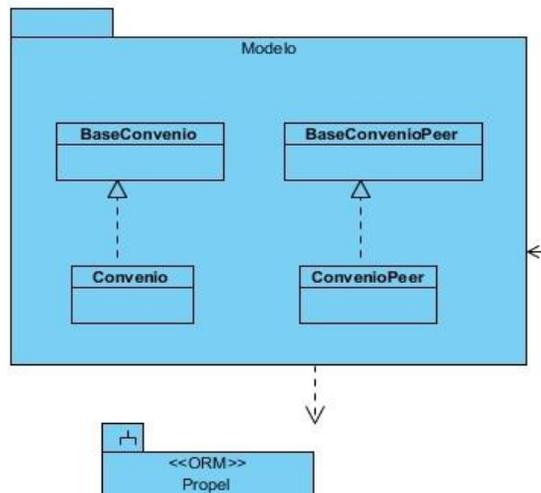


Fig. 3: Patrón Experto

### Creador:

Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. En la clase Actions se encuentran las acciones definidas, en las cuales se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

```
class saveComplianceAction extends MySIGAction {  
    public function execute($request) {  
        try {  
            $codconvenioinformativo = $request->getParameter('codconvenioinformativo');  
            $codestado = $request->getParameter('codestado');  
            $codincumplimiento = $request->getParameter('codincumplimiento');  
            $causaincumplimiento = $request->getParameter('causaincumplimiento');  
  
            $convenio = ConvenioinformativoPeer::retrieveByPK($codconvenioinformativo);  
            $request->hasParameter('option') && $request->getParameter('option') != "" ? $option = $request->getP  
  
            switch ($option) {  
                case 'edit':  
                    if (!(is_null($convenio))) {  
                        ConvenioinformativoPeer::saveCompliance($codconvenioinformativo, $codincumplimiento,  
                            $causaincumplimiento, $convenio);  
                    } else {  
                        return $this->renderText("{\"success:false, errors:{message: 'Imposible realizar la operaci  
                    }  
                    break;  
                case 'delete':  
                    if (!(is_null($convenio))) {  
                        ConvenioinformativoPeer::deleteCompliance($codconvenioinformativo, $codestado,  
                            $codincumplimiento, $convenio);  
                    } else {  
                        return $this->renderText("{\"success:false, errors:{message: 'Imposible realizar la operaci  
                    }  
                    break;  
            }  
        }  
    }  
}
```

Fig. 4: Patrón Creador (Fragmento de inicialización de un objeto)

### Bajo Acoplamiento:

Acoplamiento es la medida de cuánto una clase está conectada (tiene conocimiento) de otras clases. Es un patrón evaluativo: un bajo acoplamiento permite que el diseño de clases sea más independiente. La clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

### Controlador:

Es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Un Controlador define el método para la operación del sistema. Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. En este caso el "index.php" escucha las peticiones que vienen desde una URL, luego se encarga de llamar al controlador específico, el cual maneja la acción requerida para satisfacer la petición realizada.

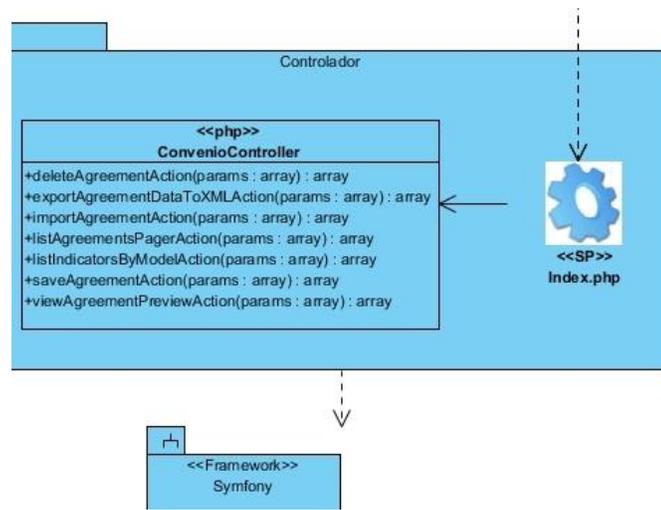


Fig. 5: Patrón Controlador

### Patrones GoF

Los patrones GoF (Gang of Four) se utilizan en situaciones frecuentes. Debido a que se basan en la experiencia acumulada al resolver problemas reiterativos. Ayudan a construir software basado en la reutilización, a construir clases reutilizables.

### Fábrica:

Es un objeto que maneja la creación de otros objetos. La fábrica permite la reutilización de código, así como la facilidad a la hora de programar. En este caso se define la clase Factory en la cual se establecen los métodos a usar para darle solución a las funcionalidades de los CU. Esto permite que en caso de tener que realizar cambios a alguna funcionalidad en vez de cambiar todas las clases en las que intervenga solo sería necesario modificar la clase Factory.

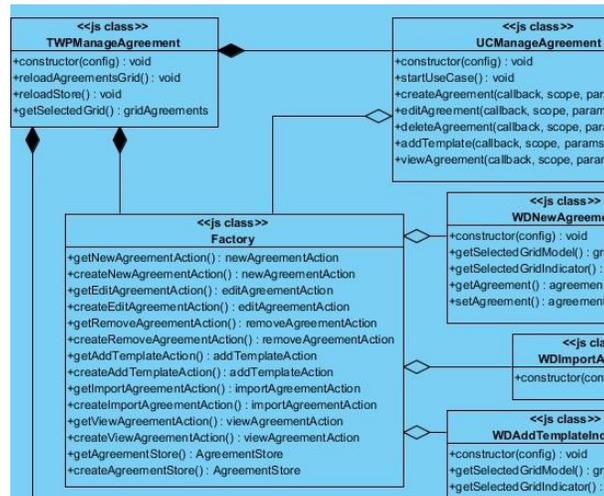
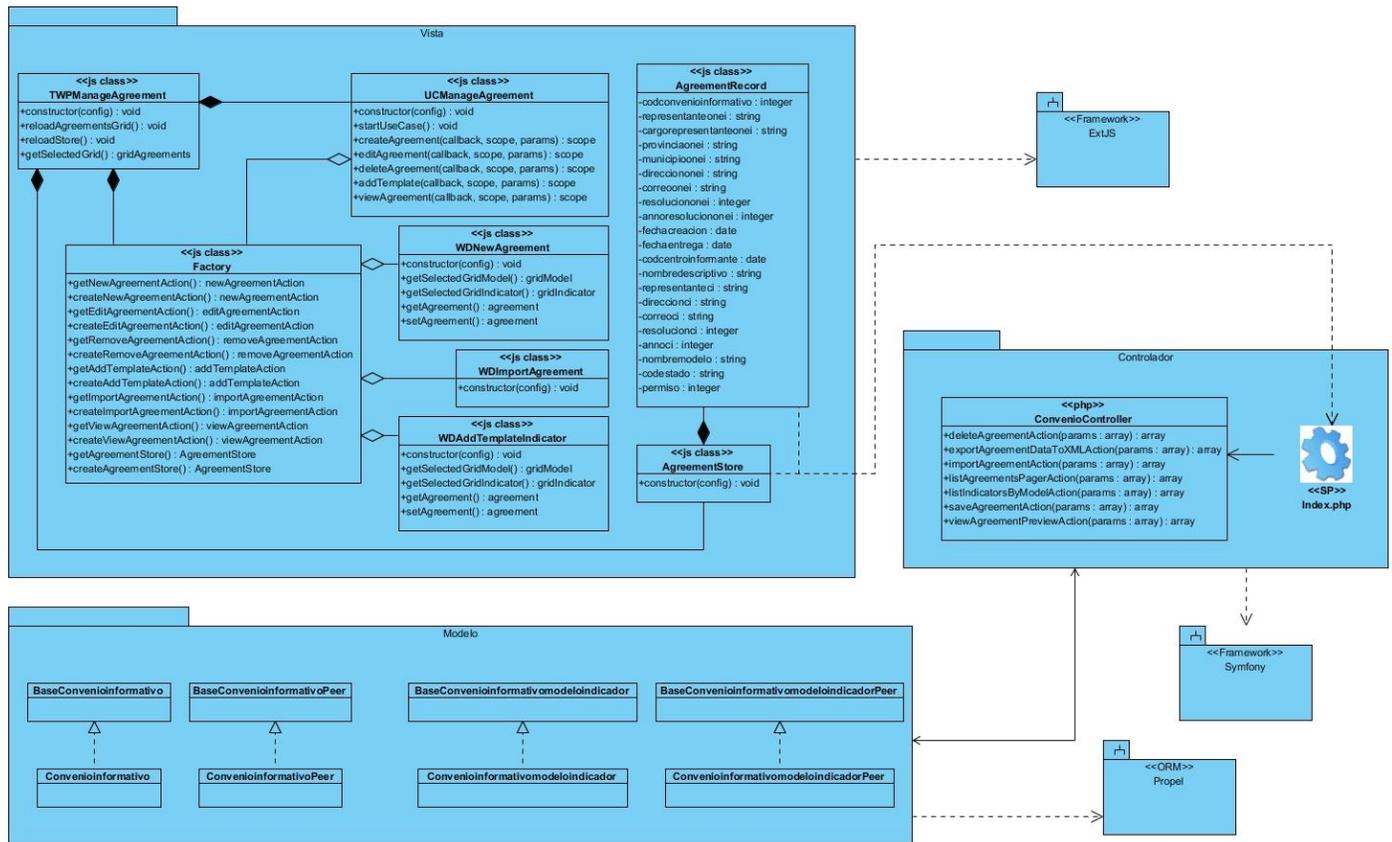


Fig. 6: Patrón Fábrica

### 2.5.3 Diagrama de clases de diseño

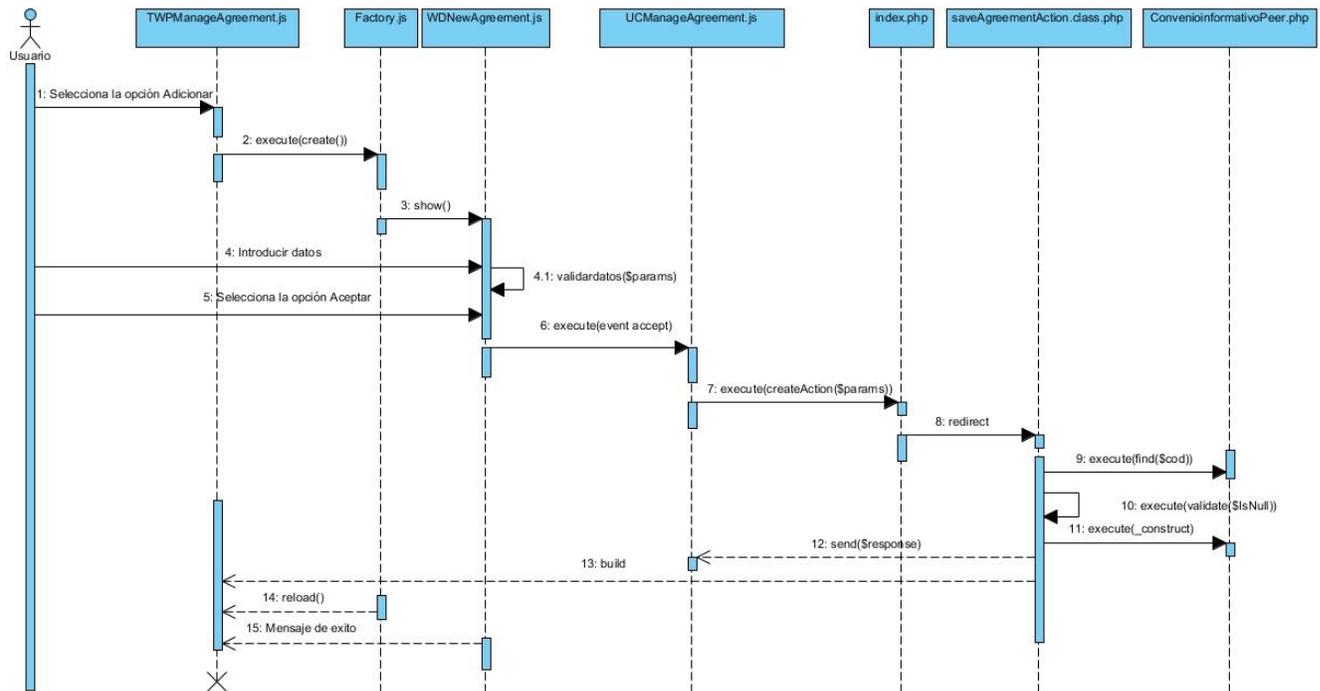
El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las Clases de Software y las Interfaces en una aplicación. Es una representación concreta de lo que se debe implementar. Estos diagramas representan la parte estática del sistema a través de la representación de las clases y sus relaciones. Los componentes de Symfony que intervienen en la construcción de la aplicación, al ser transparentes al programador se decidió modelarlos como un subsistema que recibió el nombre Symfony; además de la existencia de otro componente fundamental de Symfony, el cual se encarga de realizar el mapeo relacional de los objetos, el cual se denomina Propel. A continuación se muestra un ejemplo:



**Fig. 7: Diagrama de clases de diseño del CU Gestionar Convenio Informativo**

Los diagramas de clases de diseño representan todas las clases con los métodos y atributos correspondientes a los CU. Para simplificar el desarrollo del módulo se utilizó la automatización de uno de los patrones más utilizados en el *framework* Symfony: el patrón MVC. En este diagrama, los elementos del modelo corresponden a las clases generadas por el ORM Propel. El mismo crea cuatro clases por cada tabla de la base de datos. Las acciones, elementos que construyen las respuestas del servidor, están implementadas en el lenguaje PHP, con la dependencia del subsistema Symfony y comprobadas por el controlador frontal "index.php". Los elementos del lado del cliente corresponden a componentes específicos del negocio, en lenguaje JavaScript utilizando los componentes del subsistema ExtJS.

### 2.5.4 Diagrama de secuencia



**Fig. 8: Diagrama de Secuencia correspondiente al escenario Adicionar Convenio Informativo**

En este diagrama de secuencia el actor Usuario selecciona la opción Adicionar en la pantalla del sistema proporcionada por la clase TWPMManageAgreement. Este ejecuta la acción create de la clase Factory, el cual realiza una llamada para que se muestre el WDNNewAgreement que le permite al usuario introducir los datos que se adicionarán. El usuario introduce los datos en el WDNNewAgreement, estos se validan y el usuario selecciona la opción Aceptar. Una vez aceptado se avisa al UCManageAgreement del evento ocurrido. Este a su vez utiliza el record entregado por el WDNNewAgreement para realizar una petición sobre el controlador frontal Index.php, el cual redirecciona la acción al controlador saveAgreementAction. El controlador realiza una búsqueda en el modelo para ver los códigos de convenios, luego ejecuta el método isNull para verificar que no exista un convenio con el mismo código y una vez comprobado crea un objeto de tipo ConvenioinformativoPeer. Al terminar la acción el controlador saveAgreementAction envía una respuesta al UCManageAgreement. Este construye el nuevo convenio. Luego la clase Factory realiza una llamada de recarga al TWPMManageAgreement actualizando de este modo el listado de convenios mostrados en este. Y por último el WDNNewAgreement envía un mensaje de éxito al TWPMManageAgreement.

### 2.5.5 Modelo de datos

Un Modelo de Datos es un catálogo de datos en forma de entidades que son conjuntos de datos formados por atributos que son datos atómicos o indivisibles. A partir de estas entidades se pueden crear estructuras complejas de datos mediante definición de relaciones entre estas entidades.

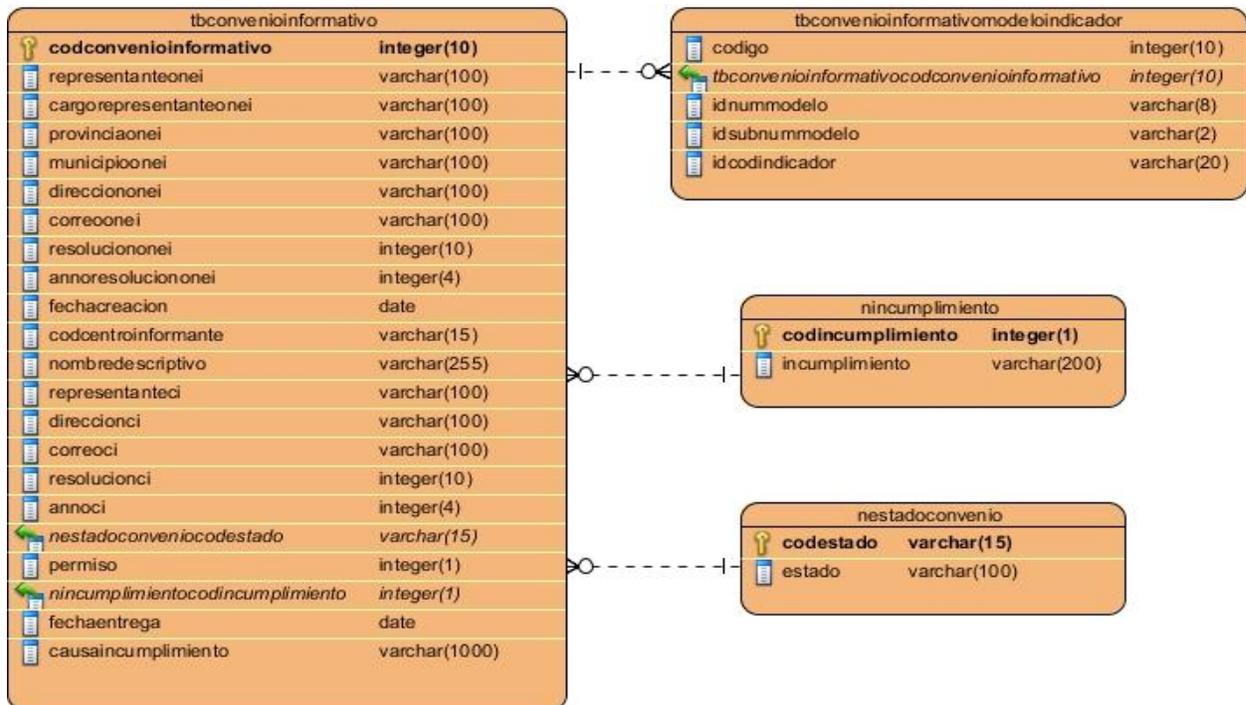


Fig. 9: Diagrama Entidad-Relación

### 2.5.6 Modelo de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos.

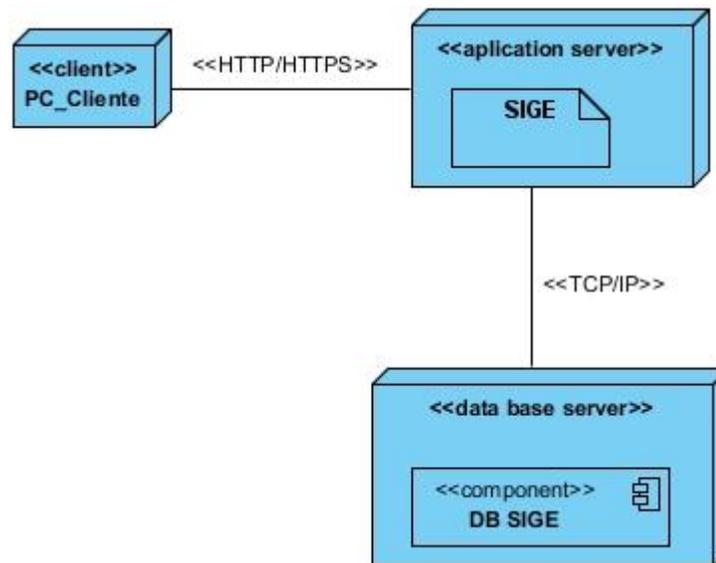


Fig. 10: Diagrama de Despliegue

**Pc Cliente:** se refiere a las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web y transcribir sus datos.

**Servidor de Aplicación:** servidor de aplicación utilizado para la publicación de la aplicación; y para lograr la conexión del sistema con la PC Cliente se utiliza HTTP (Hypertext Transfer Protocol) y HTTPS (Hypertext Transfer Protocol Secure) como protocolo de comunicación. Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor. Es el responsable de ejecutar el código de las páginas servidor. Se utiliza el servidor de aplicación.

**Conexión HTTP/HTTPS:** es el protocolo utilizado entre los browser de los clientes y el servidor Web. Este elemento de la arquitectura representa un tipo de comunicación no orientado a la conexión entre clientes y servidor.

**Conexión TCP/IP:** es la base del Internet que sirve para enlazar computadoras. El protocolo TCP/IP es utilizado para establecer la conexión entre el servidor de aplicación y el servidor de base de datos.

**Servidor de Base de Datos:** se refiere a un servidor que radica en cada nodo regional en el cual el cliente define que sean guardados los datos. En el servidor central estarán almacenados todos los datos recopilados por todos los nodos regionales. Los servidores de bases de datos elegidos son PostgreSQL, el cual está disponible para Linux y Windows.

## **2.6 Conclusiones del capítulo**

En el capítulo que finaliza se obtuvo un mayor entendimiento del negocio, donde se confeccionó el Modelo de Dominio. Se definieron los requisitos funcionales y no funcionales a tener en cuenta obteniendo una idea general de las funcionalidades que debe cumplir el sistema, las cuales darán respuesta a las necesidades del problema. Se identificaron 20 requisitos funcionales agrupados en 3 casos de usos del sistema, los cuales fueron relacionados mediante un diagrama de casos de uso del sistema y se les realizó una descripción detallada logrando así un mayor acercamiento a lo que el sistema deberá cumplir. Se determinó el patrón arquitectónico a seguir, así como los patrones de diseños utilizados en la realización de los diagramas de clases del diseño. Se generaron los diagramas de clases del diseño los cuales representan la estructura estática del sistema. Así como la generación de diagramas de secuencias para brindar una visión de cómo el usuario interactúa con la aplicación. A partir del diseño de clases persistentes se obtuvo el modelo de datos, donde este describe la representación lógica y física de los datos persistentes y la realización del diagrama de despliegue propició una visión de cómo está distribuido el sistema físicamente.

## **Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística**

### **Introducción**

En el capítulo se aborda sobre las actividades que se llevan a cabo durante la fase de implementación y pruebas. Se analizará el Modelo de Implementación así como una descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes. Además se diseña y aplican las pruebas para comprobar el correcto funcionamiento del módulo.

### **3.1 Modelo de implementación**

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

#### **3.1.1 Diagrama de componentes**

El Modelo de Implementación es una réplica del Modelo de Diseño, donde los paquetes de este tendrán sus correspondientes subsistemas de implementación, que contendrán uno o varios directorios y archivos que se necesitan para implementar los elementos de diseño correspondientes. De esta manera se agrupan las páginas clientes y servidoras que no sean independientes en un sólo componente, una vez creados estos componentes pues se crean las relaciones entre estos, teniendo en cuenta las establecidas en el modelo de diseño. Así nuestro equipo confeccionó un diagrama de componente por cada caso de uso que tiene nuestra problemática.

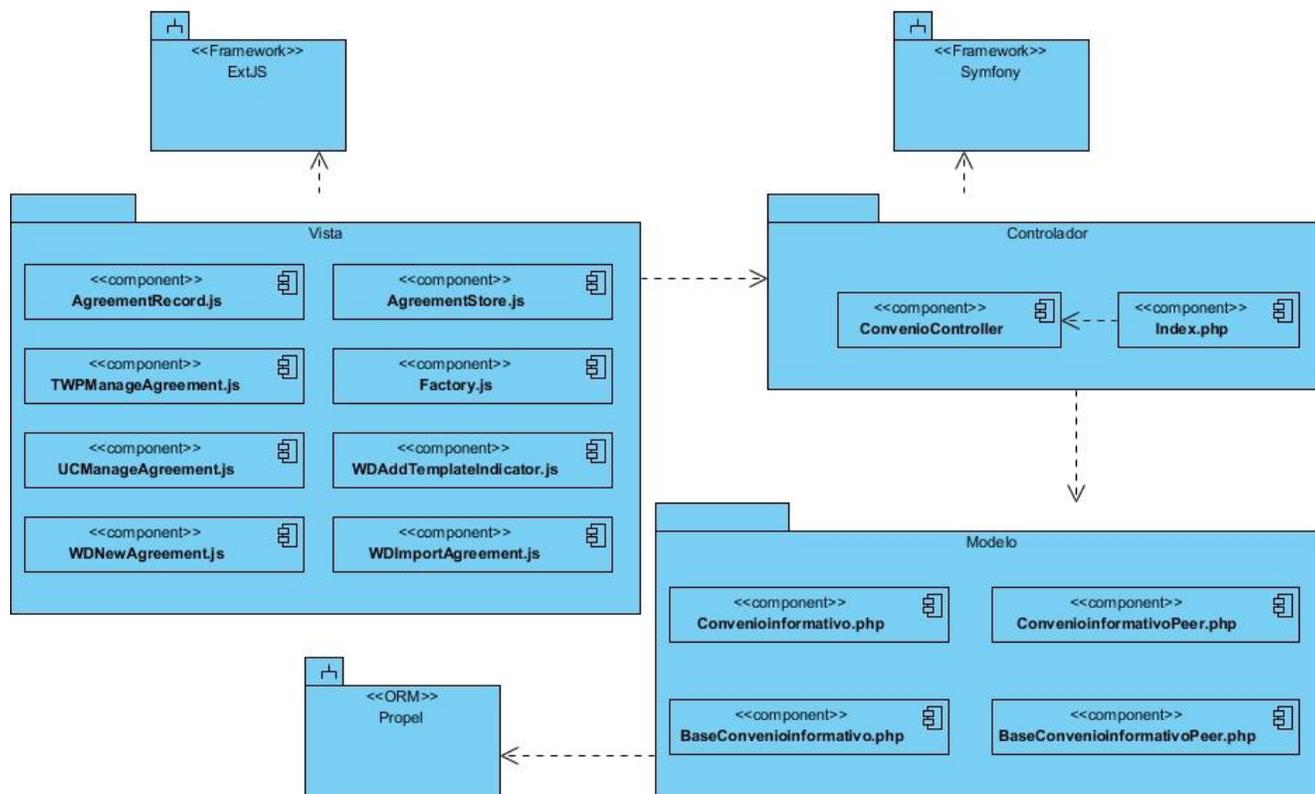


Fig. 11: Diagrama de componentes del CU Gestionar Convenio

## 3.2 Código fuente

El código fuente es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está escrito por completo su funcionamiento. Estas líneas de texto están escritas en un lenguaje de programación específico y que puede ser leído por un programador. Debe traducirse a lenguaje máquina para que pueda ser ejecutado por la computadora o a bytecode para que pueda ser ejecutado por un intérprete. Este proceso se denomina compilación.

### 3.2.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Lo cual permite

que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia presente mantenibilidad.

### 3.2.2 Estándares de codificación utilizados

- Todas las etiquetas php deben ser completas (<?php?>)... no reducidas (<? ?>).
- Los bloques de código siempre deben estar encerrados por llaves, si consta de una línea no es necesario utilizar llaves.
- Los nombres de clases pueden contener sólo caracteres alfanuméricos.
- Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos.
- Los nombres de funciones y variables deben empezar siempre con una letra minúscula utilizando la forma "camelCase".
- Se recomienda en sentido general la elocuencia, los nombres de las funciones deben ser elocuentes como para describir su propósito y comportamiento.
- Los ficheros tienen que concluir con la extensión .js, y no deben incluir signos de puntuación excepto – (guión medio) o \_ (guión bajo).
- Los métodos de los objetos deben ser nombrados utilizando la forma 'camelCase'.
- Usar paréntesis en expresiones que implican distintos operadores para evitar problemas con el orden de precedencia de los operadores. Incluso si parece claro el orden de precedencia de los operadores, podría no ser así para otros, no se debe asumir que otros programadores conozcan el orden de precedencia.
- Las cadenas tienen que ser definidas utilizando comillas simples siempre que sea posible, para obtener un mejor rendimiento.

### 3.2.3 Ejemplo de código fuente

```
class importAgreementAction extends sfAction {  
  
    public function execute($request) {  
        try {  
            $nombreArchivo = $this->getRequest()->getFileName('file');  
            $directorioSubidas = sfConfig::get();  
            $this->getRequest()->moveFile('file', $directorioSubidas.'/var/www/html/trunk/data/convenios/'.$nombreArchivo);  
  
            $convenio = new CustomAgreements();  
            $convenio->importTemplateAgreementFromXML($nombreArchivo);  
            return $this->renderText("{success:true}");  
  
        } catch (Exception $e) {  
  
            $error = new Error("{success:false, errors:{message:'Imposible importar la plantilla de convenio.'}}", 3, $e);  
            return $this->renderText($error->getMessage());  
  
        }  
    }  
}
```

Fig. 12: Ejemplo de Código Fuente de la clase Importar Convenio

```
createNewAgreementAction: function(usecase) {  
    if (newAgreementAction)  
        return newAgreementAction;  
  
    newAgreementAction = new Ext.Action({  
        text: patdsi.discipline.manage_agreement.Factory.I18N.LABEL1,  
        iconCls: 'addGrid',  
        scope: this,  
        handler: function() {  
            var wdNewAgreement = new patdsi.discipline.manage_agreement.WDNewAgreement({  
                services: usecase,  
                iscreate: true  
            });  
            wdNewAgreement.show();  
            wdNewAgreement.on('agreementcreated', function() {  
                if (newAgreementAction.fnAgreementCreated)  
                    newAgreementAction.fnAgreementCreated();  
            }, this);  
        }  
    });  
  
    return newAgreementAction;  
}
```

Fig. 13: Ejemplo de código fuente del método Crear Convenio

## 3.3 Pruebas de software

Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto. Esta actividad forma parte del proceso de control de calidad global. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software

y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del proceso de desarrollo. (Fiestas, 2014)

Para asegurar el correcto funcionamiento de las aplicaciones en cada tipo de terminal y sistema operativo, el equipo de pruebas debe realizar diversas pruebas a diferentes niveles. Los diferentes niveles corresponden a las siguientes clasificaciones que son: pruebas de unidad, integración, validación, sistema y aceptación. A partir del hecho de que las pruebas pueden ejecutarse en cualquier punto del proceso de desarrollo de software, es entonces que los niveles de prueba nos permiten entender con claridad los diferentes puntos o etapas en donde pueden ejecutarse ciertos tipos de prueba. Para verificar que el desarrollo del módulo cumple con los requisitos previamente identificados se realizarán las pruebas a nivel de validación y sistema.

En cada uno de estos niveles de prueba, se podrán ejecutar diferentes tipos de prueba tales como: función, regresión, integración y pruebas relacionadas con el desempeño del sistema. Los tipos de pruebas buscan asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.

### **3.3.1 Prueba de integración**

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. En este se verifica que módulos individuales son combinados y probados como un grupo. En una interfaz es posible perder datos, un módulo podría tener un efecto adverso e inadvertido sobre otro.

Se decide aplicar como tipo de prueba las pruebas de integración ascendente, como su nombre lo indica, empieza la construcción y la prueba con módulos atómicos (es decir, componentes de los niveles más bajos de la estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, siempre está disponible el procesamiento requerido para los componentes subordinados a un determinado nivel y se elimina la necesidad de resguardarlos. (Pressman, 2009)

### **Pasos para realizar la integración**

- Creación de la app discipline por consola, así como los módulos manage\_agreemenet, discipline\_management y manage\_compliance. Ej:

```
php symfony generate:app discipline
```

php symfony generate:module discipline manage\_agreement

- Creación de los correspondientes TWP y UC por cada uno de los módulos.
- En la base de datos con la que se trabaja se adiciona en la tabla napp el nombre de la clase que se creó y el nombre que aparecerá en la aplicación.
- En la tabla nmodule se ponen los nombres de los módulos creados, a que clase pertenecen y el nombre que tendrán en la aplicación.
- Se ejecuta el comando php symfony propel:build-model para que propel cree en el sistema las clases correspondientes a las tablas creadas en la BD.
- En la clase indexTestSuccess.php se agregan las rutas de los módulos.
- En la clase patdsi-base.js se especifican las rutas de las clases .js pertenecientes al módulo.
- Una vez realizados los pasos anteriores el módulo aparece en el sistema y lo que queda es trabajar en base a este.

### **3.3.2 Prueba de validación**

Proporciona un aseguramiento final de que el software cumple con todos los requisitos funcionales, de comportamiento y desempeño. La prueba se concentra en las acciones visibles para el usuario y en la salida que este puede reconocer. La validación se alcanza cuando el software funciona de tal manera que satisface las expectativas razonables (especificación de requisitos de software) del cliente. La validación se logra mediante una serie de pruebas que demuestran que se cumplen con los requisitos.

Se decide aplicar como tipo de prueba las pruebas funcionales ya que asegura el apropiado trabajo de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Las pruebas Funcionales deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los CU y las reglas del negocio. Las metas de estas pruebas son verificar la apropiada aceptación de datos y verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio. Este tipo de pruebas están basadas en técnicas de caja negra, que es, verificar la aplicación (y sus procesos internos) mediante la interacción con la aplicación y analizar la salida (resultados). (Londoño, 2005)

Caja Negra este método se centra en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (Pressman, 2009)

La partición equivalente es una técnica de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. El diseño de casos de prueba para partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. (Pressman, 2009)

**Tabla. 3: Secciones de prueba para el caso de uso Gestionar Convenios**

Nombre de la sección	Descripción de la funcionalidad
SC1 Adicionar Convenio	El usuario decide insertar un nuevo convenio, el sistema envía un mensaje indicando que la petición se realizó exitosamente, terminando así el CU.
SC2 Editar Convenio	El usuario decide editar un convenio existente, el sistema envía un mensaje indicando que la petición se realizó exitosamente, terminando así el CU.

**Tabla. 4: Descripción de las variables del diseño de casos de prueba del CU Gestionar Convenios**

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Código convenio	Campo de texto	No	Código con que se identificará el convenio. Numérico. Admite 0-9
2	Representante	Campo de texto	No	Nombre del representante de la OMEI. Cadena. Admite A-Z, a-z
3	Provincia	Campo de selección	No	Será una provincia almacenada en la base de datos
4	Municipio	Campo de selección	No	Será un municipio almacenad en la base de datos

### Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística

5	Nombre del centro informante	Campo de texto	No	Nombre del centro informante. Cadena. Admite A-Z, a-z
6	Representante CI	Campo de texto	No	Nombre del representante del centro informante. Cadena. Admite A-Z, a-z

**Tabla. 5: Valores válidos de variables del diseño de casos de prueba del CU Gestionar Convenios**

Código convenio	Representante	Provincia	Municipio	Nombre del centro informante	Representante CI
2	(Alex Brito de las Cuevas)	(La Habana)	(Boyeros)	(MINISTERIO DE LA INDUSTRIA SIDERO MECANICA)	(Emilio Ruiz Rodríguez)

**Tabla. 6: Caso de prueba Adicionar Convenio del CU Gestionar Convenios**

Escenario	Descripción	Variables						Respuesta del sistema	Flujo central
		1	2	3	4	5	6		
EC 1.1 Adicionar convenio	En este escenario se realiza la adición de un nuevo convenio al sistema correctamente.	V	V	V	V	V	V	Se adiciona correctamente el convenio y el sistema permite la inserción de uno nuevo. Muestra un mensaje confirmando el éxito de la operación.	1. Módulo Disciplina Estadística/Actualizar Convenios/Adicionar (El botón Adicionar se encuentra: Menú superior dentro del panel Actualizar Convenios, en la esquina superior derecha). 2. Se llenan los campos correctamente. 3. Se presiona el botón Aceptar.

### Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística

EC 1.2 Adicionar convenio con datos incorrectos	En este escenario se realiza la adición de un nuevo convenio al sistema con datos incorrectos.	I	V	V	V	V	V	El sistema valida que los campos estén correctamente y muestra un mensaje informando que existen datos no válidos.	1. Módulo Disciplina Estadística/Actualizar Convenios/Adicionar (El botón Adicionar se encuentra: Menú superior dentro del panel Actualizar Convenios, en la esquina superior derecha). 2. Se llenan los campos con datos no válidos.
		V	I	V	V	V	V		
		V	V	I	V	V	V		
		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		
EC 1.3 Cancelar operación	En este escenario se cancela la operación de adicionar un nuevo convenio							Se cancela la operación de adicionar un nuevo convenio.	1. Módulo Disciplina Estadística/Actualizar Convenios/Adicionar (El botón Adicionar se encuentra: Menú superior dentro del panel Actualizar Convenios, en la esquina superior derecha). 2. Se completan los datos. 3. Se presiona el botón Cancelar.

**Tabla. 7: Caso de prueba Editar Convenio del CU Gestionar Convenios**

Escenario	Descripción	Variables						Respuesta del sistema	Flujo central
		1	2	3	4	5	6		
EC 2.1 Editar convenio	En este escenario se realiza la edición de un convenio existente en el	V	V	V	V	V	V	Se edita correctamente el convenio seleccionado. El sistema muestra	1. Módulo Disciplina Estadística/Actualizar Convenios/Editar (El botón Editar se encuentra: dentro del panel Actualizar Convenios,

### Capítulo 3: Implementación y Pruebas del Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística

	sistema correctamente.							un mensaje confirmando el éxito de la operación.	específicamente dentro del grid de los convenios e la parte derecha). 2. Se editan los campos correctamente. 3. Se presiona el botón Aceptar.
EC 2.2	En este escenario se realiza la edición de un convenio existente en el sistema con datos incorrectos.	I	V	V	V	V	V	El sistema valida que los campos estén correctamente y muestra un mensaje informando que existen datos no válidos.	1. Módulo Disciplina Estadística/Actualizar Convenios/Editar (El botón Editar se encuentra: dentro del panel Actualizar Convenios, específicamente dentro del grid de los convenios e la parte derecha). 2. Se editan los campos con datos no válidos.
		V	I	V	V	V	V		
		V	V	I	V	V	V		
		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		
EC 1.3	En este escenario se cancela la operación de editar un convenio existente							Se cancela la operación de adicionar un nuevo convenio.	1. Módulo Disciplina Estadística/Actualizar Convenios/Editar (El botón Editar se encuentra: dentro del panel Actualizar Convenios, específicamente dentro del grid de los convenios e la parte derecha). 2. Se completan los datos. 3. Se presiona el botón Cancelar.

Tras aplicar las pruebas funcionales al módulo desarrollado se comprobó la correcta validación de los campos, verificando que cada cual aceptará exclusivamente los caracteres válidos. En el proceso de pruebas se detectaron en la primera iteración 7 No Conformidades (NC) clasificadas como significativas. En la segunda iteración se detectaron 4 NC significativas. En la tercera iteración se comprobó que las NC fueran corregidas y gestionadas correctamente.

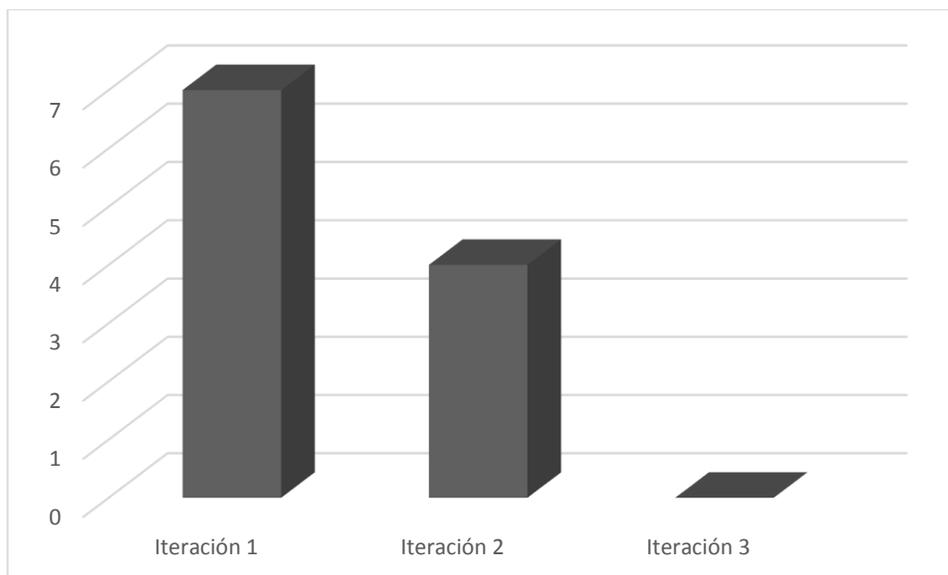


Fig. 14: Gráfica de iteraciones de pruebas funcionales

### 3.3.3 Prueba de aceptación

Las pruebas de aceptación son aquellas que son diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados en la fase de análisis para cubrir todo ese espectro, y ejecutadas por el propio usuario final, no por todos evidentemente, pero sí por una cantidad de usuarios finales significativa que dé validez y conformidad al producto que se les está entregado en base a lo que se acordó inicialmente. De manera general las pruebas de aceptación pueden afrontarse mediante dos tipos de procedimiento para realizarlas: pruebas alfa y pruebas beta. (José Ponce González, 2014)

Se decide aplicar las pruebas alfa ya que en ellas se le entrega a un usuario final todo el producto terminado, junto a su documentación correspondiente para que éste, en presencia del desarrollador y en entornos previamente preparados para el proceso de dichas pruebas, vaya informando de las inconsistencias y errores que detecte. En este caso el usuario final será el líder del proyecto el cual en conjunto con los desarrolladores realizaran las pruebas de aceptación, esta prueba se evidencia en los anexos.

## 3.4 Conclusiones del capítulo

En el desarrollo del presente capítulo se realizó el modelo de implementación del módulo con el propósito de mostrar los componentes del sistema y sus relaciones, a través del diagrama de componentes. Se especificó el uso de los estándares de codificación para lograr un estilo claro y organizado del código durante

la fase de implementación. Para evaluar la calidad de la aplicación se validó la completitud de los requisitos con las pruebas funcionales al software. Se identificaron un total de 11 NC las cuales fueron resueltas paulatinamente, obteniéndose un producto libre de errores y listo para su ejecución. Se comprobó que el módulo de Control de la Disciplina Estadística obtenido satisface las necesidades de la ONEI.

## **Conclusiones Generales**

Una vez culminado el trabajo es posible afirmar que se les dio cumplimiento a los objetivos trazados para el mismo:

- Se realizó el estudio sobre los principales conceptos relacionados al control y seguimiento de convenios, y las nuevas tecnologías permitiendo sentar las bases para el desarrollo del módulo.
- El componente fue implementado con el objetivo de cumplir todas las funcionalidades que fueron identificadas durante la etapa de análisis y diseño. Además se utilizaron herramientas, lenguajes y tecnologías propuestas por el grupo de arquitectura del departamento del Centro, en su mayoría distribuidas bajo licencias de software libre en correspondencia con las políticas de la Universidad y del país.
- Se aplicaron pruebas de integración, funcionales y de aceptación al módulo corrigiéndose los errores encontrados para comprobar su adecuado funcionamiento. Se diseñaron casos de prueba basados en los casos de uso del sistema, se realizó el método de caja negra para verificar la eficacia de la aplicación y se llevó a cabo la prueba de aceptación por parte del cliente.

## Recomendaciones

Luego de haber alcanzado los objetivos trazados en la investigación se recomienda:

- Agregar al módulo la funcionalidad de realizar auditorías a partir de los resultados obtenidos en el mismo.
- Agregar a SigeLite las funcionalidades del módulo que le permitan conocer el estado en que se encuentra el cumplimiento del convenio, así como el tiempo restante para su entrega.

---

## Referencias Bibliográficas

1. **ABC, Definición. 2014.** Definición ABC. [En línea] 2014. [Citado el: 6 de Diciembre de 2014.] [www.definicionabc.com](http://www.definicionabc.com).
2. **Achour, Mehdi. 2011.** Manual de PHP. [En línea] 2011. [Citado el: 4 de Febrero de 2015.] <http://php.net/manual/es/preface.php>.
3. **ALEGSA. 2010.** Definición de lenguaje de programación. [En línea] 2010. [Citado el: 4 de Febrero de 2015.] <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php>.
4. —. **2010.** Definición de Servidor de Aplicaciones. [En línea] 2010. <http://www.alegsa.com.ar/Dic/servidor%20de%20aplicaciones.php>.
5. —. **2010.** Definición de SGBD. [En línea] 2010. <http://www.alegsa.com.ar/Dic/sgbd.php>.
6. **Alonso, Evelyn Menéndez. 2011.** Herramientas CASE para el proceso de desarrollo de Software. [En línea] 2011. [Citado el: 4 de Febrero de 2015.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software.shtml>.
7. **Angel. 2008.** Angel Software. [En línea] 30 de Noviembre de 2008. <http://angelmolsoftware.blogspot.com>.
8. **Barchini, Graciela Elisa. 2005.** Entorno Visual de aprendizaje. [En línea] 2005. [Citado el: 19 de Noviembre de 2014.] [eva.uci.cu/filephp/104/Tema 2/Recursos bibliograficos/Metodos/2-Metodos I D en la Informatica.pdf](http://eva.uci.cu/filephp/104/Tema%202/Recursos%20bibliograficos/Metodos/2-Metodos%20I%20D%20en%20la%20Informatica.pdf).
9. **Biithahh, Gha. 2011.** slideshare . *Entornos de desarrollo integrados*. [En línea] 2011. [Citado el: 4 de Febrero de 2015.] <http://es.slideshare.net/GhaBiithahh/entornos-de-desarrollo-integrados>.
10. **Bogarín, José. 2014.** Itaipu Binacional. *Sistema de monitoreo de convenios*. [En línea] 8 de Agosto de 2014. [Citado el: 20 de Abril de 2015.] <http://www.itaipu.gov.py/es/sala-de-prensa/noticia/sistema-de-monitoreo-de-convenios>.
11. **Corporation, LinkedIn. 2015.** slideshare. [En línea] 2015. [Citado el: 6 de Febrero de 2015.] <http://es.slideshare.net/JUANESTEFAdiseo-de-sistemas>.
12. **Fabien Potencier, François Zaninotto. 2008.** *Symfony la guía definitiva*. 2008.
13. **Fiestas, Jhonattan. 2014.** ElevenPaths. [En línea] 2014. [Citado el: 9 de Abril de 2015.] <http://blog.elevenpaths.com/2014/09/qa-pruebas-para-asegurar-la-calidad-del.html>.

14. **Foundation, ITIL®. 2011.** *Herramientas y metodologías*. [En línea] 2011. [Citado el: 4 de Febrero de 2015.] [http://itilv3.osiatis.es/proceso\\_mejora\\_continua\\_servicios\\_TI/herramientas\\_metodologias.php](http://itilv3.osiatis.es/proceso_mejora_continua_servicios_TI/herramientas_metodologias.php).
15. **García, Elena Fernández. 2014.** *disciplina estadística*. La Habana, 16 de Noviembre de 2014.
16. **Genbeta. 2014.** Genbeta:dev. [En línea] 14 de Julio de 2014. [Citado el: 6 de Febrero de 2015.] <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>.
17. **Girona, Jordi. 2013.** inLab FIB. [En línea] Noviembre de 2013. [Citado el: 6 de Diciembre de 2014.] <http://inlab.fib.upc.edu/es/sistema-de-informacion-de-convenios-de-cooperacion-educativa-de-la-upc>.
18. **GITHUB. 2014.** What is Symfony? [En línea] 2014. [Citado el: 12 de Diciembre de 2014.] <https://github.com/symfony/symfony>.
19. **Hernandez Barrios, Carmelo Jose. 2014.** slideshare. [En línea] 2014. [Citado el: 4 de Febrero de 2015.] <http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348>.
20. **Hosting, Module For. 2013.** What is Apache HTTP Server? [En línea] 2013. [Citado el: 12 de Diciembre de 2014.] <http://www.modulehosting.com/apache.html>.
21. **Ing. Frank González Fernández, Ing. Héctor Luis Reyes, Lic. Elena Leonila Fernández García, Claudia García Suárez del Villar, Ing. Virgilio Suárez Bello, Ing. Alejandro González Sánchez, Ing. Adrian Rosales Cruz, Ing. Clara Elena Brizuela Fig. 2014.** *Sistema Integrado de Gestión Estadística*. La Habana, Cuba : UCIENCIA, 2014.
22. **José Ponce González, Francisco José Dominguez Mayo, Javier Jesús Gutiérrez Rodríguez, Maria José Escalona Cuaresma. 2014.** *Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión*. Sevilla : s.n., 2014.
23. **Londoño, Jorge Hernan Abad. 2005.** Ingeniería de Software. [En línea] 6 de Abril de 2005. [Citado el: 2 de Abril de 2015.] <http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>.
24. **Martinez, Rafael. 2010.** PostgreSQL-es. *Sobre PostgreSQL*. [En línea] 2 de Octubre de 2010. [Citado el: 12 de Diciembre de 2014.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
25. **MDN. 2014.** JavaScript. [En línea] 2014. [Citado el: 12 de Diciembre de 2014.] <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.
26. **Multiplataforma, Desarrollo Móvil. 2012.** ¿Que denominamos Framework en Informática? [En línea] 2012. [Citado el: 12 de Diciembre de 2014.] <http://desarrollomovilmultiplataforma.blogspot.com/2012/08/aspectos-teoricos-framework.html>.

27. **Muñoz, David Ruiz. 2008.** *Manual de Estadística*. 2008.
28. **NETBEANS. 2011.** ¿Qué es NetBeans? [En línea] 2011. [Citado el: 12 de Diciembre de 2014.] [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).
29. **OMG, Object Management Group. 2005.** Introduction to OMG's Unified Modeling Language™ (UML®). [En línea] Julio de 2005. [Citado el: 4 de Febrero de 2014.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
30. **ONEI. 2009.** Objetivos, Funciones y Atribuciones de la Oficina Nacional de Estadística. *Acuerdo No.6688*. La Habana : s.n., 2009.
31. —. **2006.** Oficina Nacional de Estadística e Información. [En línea] 2006. [Citado el: 17 de Noviembre de 2014.] [www.one.cu](http://www.one.cu).
32. **Potencier, Fabien. 2015.** librosweb. [En línea] 2015. [Citado el: 6 de Febrero de 2015.] [http://librosweb.es/libro/symfony\\_1\\_1/capitulo\\_2/el\\_patron\\_mvc.html](http://librosweb.es/libro/symfony_1_1/capitulo_2/el_patron_mvc.html).
33. **Pressman. 2009.** Cap\_13\_Estrategia\_Prueba. 2009.
34. **Rodríguez, Saúl Cuesta. 2014.** SG Buzz. [En línea] 2014. [Citado el: 12 de Febrero de 2015.] <http://sg.com.mx/>.
35. **Romero, Hermenegildo. 2009.** slideshare. [En línea] 2009. [Citado el: 4 de Febrero de 2015.] <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>.
36. **Rosada, Alexander Rodríguez. 2013.** RESOLUCIÓN No. 157/2013. s.l. : ONEI, 2013.
37. **Rosas, Juan Eladio Sánchez. 2008.** Desarrollo en Web. *ExtJS lo bueno, lo malo y lo feo*. [En línea] 22 de Octubre de 2008. [Citado el: 12 de Diciembre de 2014.] <http://blogs.antartec.com/desarrolloweb/page/5/>.
38. *SConv: Sistema informático para el control estatal de la producción porcina no especializada en Cuba*. **Beatriz L. García, R. Pirez, G. Hernández, Edelkys Antunez. 2006.** 1, s.l. : Revista Cubana de Ciencia Agrícola, 2006, Vol. 40.
39. **scribd. 2015.** Scribd Inc. [En línea] 2015. [Citado el: 16 de Enero de 2015.] <http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales#scribd>.
40. **Software, Departamento Ingeniería y Gestión de. 2014.** *Modelando el Sistema Con Escenario CU*. 2014.
41. **softwaresea. 2009 .** Visual Paradigm for UML estándar. [En línea] 2009 . [Citado el: 4 de Febrero de 2015.] <http://es.softwaresea.com/download-Visual-Paradigm-for-UML-est%E1ndar-10136498.htm>.

42. **Sparx. 2007.** Sparx Systems. *El Modelo de Caso de Uso*. [En línea] 2007. [Citado el: 16 de Enero de 2014.] [http://www.sparxsystems.com.ar/resources/tutorial/use\\_case\\_model.html](http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html).
43. **synergix. 2008.** Tecnología y Synergix. *Modelo de Dominio*. [En línea] 10 de Julio de 2008. [Citado el: 16 de Diciembre de 2014.] <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.
44. **uml-diagrams.org. 2009.** The Unified Modeling Language. [En línea] 2009. [Citado el: 4 de Febrero de 2015.] <http://www.uml-diagrams.org/>.
45. **University, William Carey. 2015.** CourseHero. *History of statistics*. [En línea] 2015. [Citado el: 23 de Enero de 2015.] <https://www.coursehero.com/file/9720046/History-of-Statistics/>.

## **Anexo**

### **Guía para entrevista no estructurada**

#### **Objetivo**

El objetivo de la presente entrevista es obtener información sobre las actividades que se realizan en la organización en sus diferentes niveles de acceso, el personal que interviene en estas actividades en cada nivel, así como las características de la estructura organizativa de los organismos de la ONEI y requisitos que se necesitan para desarrollar el Módulo de Control de la disciplina estadística para el Sistema Integrado de Gestión Estadística. Esta entrevista va dirigida a la Directora de la ONEI, la cual maneja dicha información en la empresa y es la más indicada para tratar los temas que intervienen en la entrevista.

#### Preguntas dirigidas a la Directora de la ONEI

1. ¿Qué se entiende por disciplina estadística?
2. ¿Cómo se realizan este proceso y qué problemas existen hoy en el mismo?
3. ¿Qué se entiende por convenio informativo?
4. ¿Qué mecanismos se utilizan para la obtención en tiempo de los indicadores?

**Centro de Tecnologías de Gestión de Datos**



**ACTA DE ACEPTACIÓN**

Por medio del presente documento se hace constar que el Trabajo de Diploma fue probado y cumple con cada uno de los requisitos definidos durante la fase de Análisis y Diseño. De igual manera el producto en sentido general satisface las necesidades del cliente.

**Entrega:** Tesistas

**Recibe:** Proyecto SIGE

**Nombre y apellidos:**

**Nombre y apellidos:** Ing. Alejandro González Sánchez

Emilio Ruiz Rodríguez

Alex Brito De las Cuevas

**Tesis:** Módulo de control de la disciplina estadística para SIGE.

**Cargo:** Líder del proyecto SIGE

**Firma:**

**Firma:**

**Representante Parte Suministradora**

**Nombre y Apellidos:** Ing. Glennis Tamayo Morales

**Cargo:** Jefa de Departamento Componentes Informáticos

**Firma:**

**Fecha:** 29/05/2015