

**Universidad de las Ciencias Informáticas
Facultad 3**



Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI

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

Autores:

Marilé Lemus Martínez

Camilo Fonseca Camejo

Tutores:

Ing. Bárbara Almarales Lara

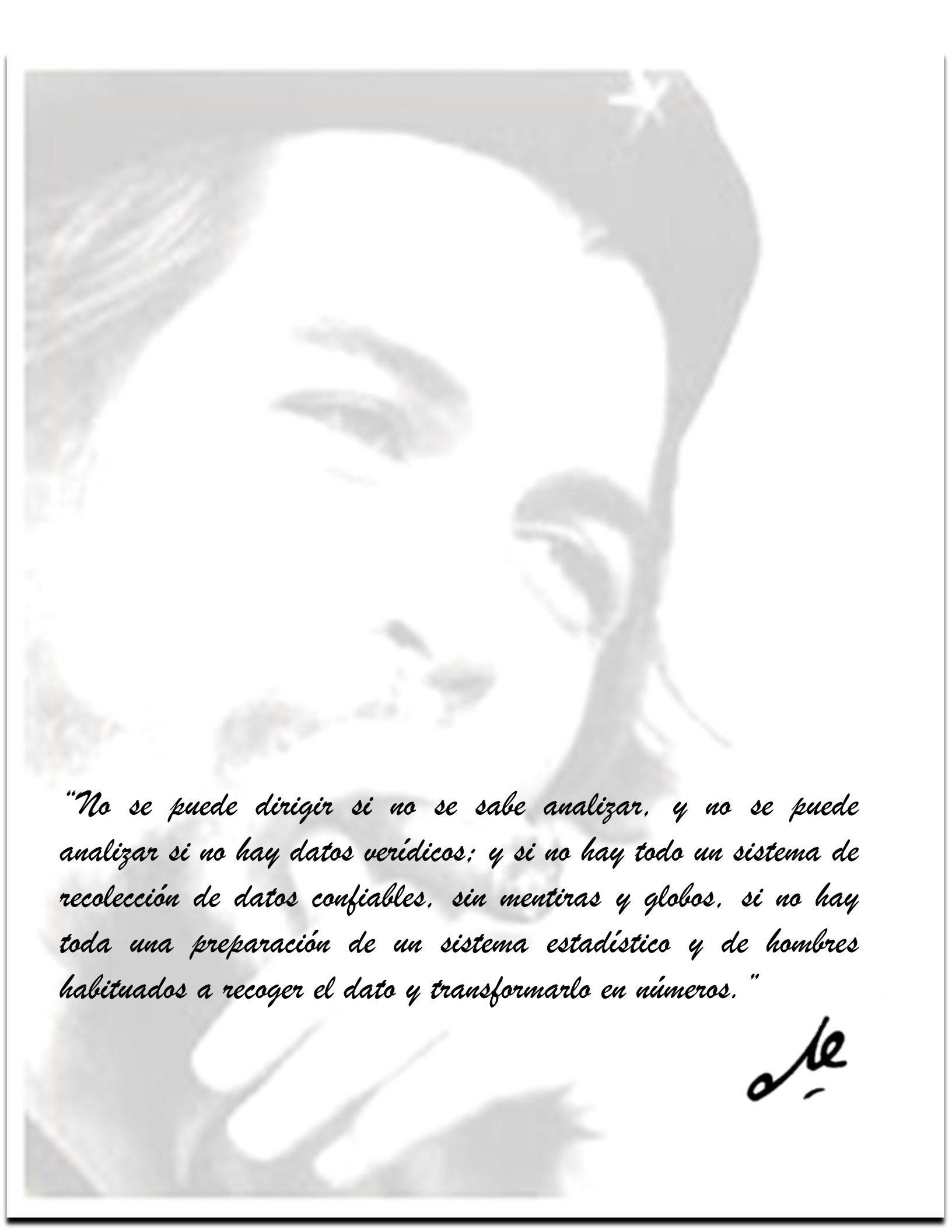
Ing. Yordanis García Leiva.

Cotutor:

Ing. Andy José Rivas Franco

Junio, 2014

“Año 56 del Triunfo de la Revolución”



"No se puede dirigir si no se sabe analizar, y no se puede analizar si no hay datos verídicos; y si no hay todo un sistema de recolección de datos confiables, sin mentiras y globos, si no hay toda una preparación de un sistema estadístico y de hombres habituados a recoger el dato y transformarlo en números."

de

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Marilé Lemus Martínez

Camilo Fonseca Camejo

Ing. Bárbara Almarales Lara

Ing. Yordanis García Leiva

Ing. Andy José Rivas Franco

AGRADECIMIENTOS

A mis padres por su entrega, cariño y comprensión, por hacer de mí una mejor persona cada día. A ellos por ser el regalo más valioso que me ha dado la vida. Los amo.

A mi hermana por ser mi amiga y hacerme reír con sus cuentos.

A toda mi familia (abuelos, tíos, primos), que a pesar de la distancia siempre me han apoyado y brindado lo mejor de sí.

A mi novio Alejandro (Ale), por apoyarme en cada momento, brindarme su amor incondicional y ser especial en mi vida.

A mis suegros por sus sabios consejos y por acogerme como una hija más de la familia.

A Rosi por quererme como una hermana y por su intento de ayudar en la tesis, a pesar de no contar con los conocimientos necesarios para ello.

A mi compañero de tesis Camilo (kmy), junto al cual compartí gran parte de mi carrera. A él por darme ánimo a seguir adelante, por su esfuerzo y dedicación.

A mis tutores Bárbara (baby) y Yordanis por emplear parte de su tiempo en la revisión del trabajo. Gracias por sus comentarios.

A Yaima, Elizabeth, Yanet, Ailin, Dianayis, Osiel, Javier, Raciél y Radel, por preocuparse por mí y mostrarme el valor de la amistad.

A mis profesores, en especial Alberto (PGI en la vocacional) y a mis compañeros de aula, los cuales influyeron en mi formación.

A Nani, Robin, Andy y Rainer por sus sugerencias en las exposiciones.

A todos los que de una forma u otra contribuyeron al desarrollo de este trabajo.

A todos, muchas gracias.

Mary

A mis tutores por todas las horas de paciencia y dedicación, a Mary mi compañera de tesis por haber sido además una gran amiga durante estos cinco años.

A todos y cada uno de los profesores que he tenido que me han aportado su conocimiento y valores.

A todos los amigos que de una forma u otra me han tendido los brazos.

A Claudi por haberme brindado su apoyo incondicional y todo su amor.

A mis abuelitos, hermanos, tíos, primos y en especial a mis padres.

A Yamira, Manuel y a Pepín por todas las cosas lindas que han hecho por mí, las cuales no tengo cómo agradecer.

A la familia de mi novia por haberme adoptado como un miembro más y hacerme partícipe de todos sus momentos.

Camilo

DEDICATORIA

A mis padres por ser ejemplos a seguir y por iluminar siempre mi camino.

A mi hermana y mi familia por apoyarme y estar presentes en todo momento.

A mi novio Ale por ser un verdadero amigo y hacerme tan feliz.

A mis suegros por quererme como a una hija.

A Rosí por ser como una hermana para mí.

A mis amigos y compañeros de aula.

Mary

A mis padres...eternos responsables de mi existir, por tanto amor, ternura y sacrificio.

A mi novia Claudí...mi amor, mi razón de ser y mi fuerza.

A toda mi maravillosa familia, incluyendo la nueva familia que me ha abierto los brazos.

A mi tío Leandro por haberme enseñado a estudiar...

A mis hermanos y amigos.

Camilo

RESUMEN

La Oficina Nacional de Estadística e Información tiene entre sus principales funciones, centralizar y emitir la estadística oficial de la Isla, para ello se vale de mecanismos como los Anuarios Estadísticos, los que constituyen publicaciones anuales formadas por capítulos, donde se brinda información relacionada con la situación demográfica, económica y social de Cuba. Actualmente esta entidad presenta varias deficiencias que dificultan la gestión de los Anuarios Estadísticos y los diferentes capítulos asociados a estos. Es por ello que el presente trabajo se plantea como objetivo general desarrollar una aplicación web que permita la gestión de los Anuarios Estadísticos en la ONEI, elevando la integridad de los datos, la usabilidad del servicio y la agilidad en el desarrollo del proceso.

Entre las herramientas utilizadas en el desarrollo del trabajo se encuentran Symfony 2 como marco de trabajo, el sistema gestor de bases de datos MySQL 5.5.16 y el entorno de desarrollo integrado NetBeans 7.4. Además se empleó SXP como metodología de desarrollo de software.

Como resultado de la investigación se obtuvo una aplicación web que facilita la gestión de los Anuarios Estadísticos en las diferentes dependencias de la Oficina Nacional de Estadística e Información.

PALABRAS CLAVES: anuarios estadísticos, aplicación web, oficina nacional de estadística e información

ÍNDICE DE CONTENIDOS

Introducción 1

Capítulo 1: Fundamentación Teórica 7

 1.1. Introducción..... 7

 1.2. Conceptos fundamentales asociados al dominio del problema..... 7

 1.3. Tendencias actuales..... 8

 1.3.1. Estudio de aplicaciones..... 8

 1.4. Metodología de desarrollo de software..... 9

 1.5. Lenguaje de Modelado Unificado 2.0..... 12

 1.6. Herramienta para el modelado de prototipos web..... 13

 1.7. Herramientas de Ingeniería de Software Asistida por Computadora..... 14

 1.8. Lenguajes..... 14

 1.8.1. Preprocesador de Hipertexto 5.3..... 15

 1.8.2. Lenguaje de Marcado de Hipertexto 5..... 15

 1.8.3. Hojas de Estilo en Cascada..... 15

 1.8.4. JavaScript..... 16

 1.9. Librerías para la conversión de los formatos Excel a HTML y HTML a PDF..... 16

 1.10. Marco de trabajo..... 17

 1.10.1. Mapeo Objeto-Relacional..... 22

 1.10.2. Motor de plantillas..... 23

 1.11. Sistema Gestor de Base de Datos..... 24

 1.12. Servidor web Apache..... 25

 1.13. Entorno de Desarrollo Integrado..... 26

 1.14. Conclusiones parciales..... 27

Capítulo 2: Descripción y diseño del sistema..... 28

 2.1. Introducción..... 28

 2.2. Descripción del sistema..... 28

 2.3. Requisitos del software..... 30

 2.3.1 Requisitos funcionales..... 30

 2.3.2 Requisitos no funcionales..... 34

 2.4. Historias de usuario..... 34

 2.5. Arquitectura..... 36

 2.6. Modelo de diseño..... 37

2.7. Patrones de diseño.....	37
2.7.1 Patrones Generales de Software para Asignar Responsabilidades.....	38
2.8. Implementación.....	40
2.8.1. Plan de <i>releases</i>	40
2.8.2. Estándar de codificación empleado.....	41
2.9. Conclusiones parciales.....	42
Capítulo 3: Validación	43
3.1. Introducción.....	43
3.2. Técnicas de validación de requisitos.....	43
3.3. Métrica aplicada a los requisitos.....	44
3.4. Validación del diseño del sistema.....	44
3.5. Pruebas.....	46
3.5.1. Pruebas de Caja Negra.....	47
3.5.2. Pruebas de Caja Blanca.....	48
3.6. Mejoras del SIGAE.....	48
3.7. Conclusiones parciales.....	49
Conclusiones Generales.....	51
Recomendaciones	52
Bibliografía Referenciada	53
Bibliografía Consultada	56

ÍNDICE DE FIGURAS

Figura 1: Top de uso de los marcos de trabajo para PHP según PHP Frameworks. 19

Figura 2: SGBD más utilizados en el mundo según DB-Engines (por puntaje)..... 24

Figura 3: Pantalla tipo para la administración del sistema..... 29

Figura 4: Pantalla tipo para la vista del usuario..... 29

Figura 5: Diagrama de módulos del sistema. 31

Figura 6: Esquema de funcionamiento del requisito Modificar provincia..... 36

Figura 7: Diagrama de paquetes del módulo Anuario Estadístico. 37

Figura 8: Clase experta en la información con respecto al tipo de estado que se le puede asignar a un capítulo en el módulo Anuario Estadístico. 38

Figura 9: Clase controladora perteneciente al módulo Anuario Estadístico, encargada de la creación de objetos de tipo Anuario Estadístico. 39

Figura 10: Existencia del patrón Bajo acoplamiento en el módulo Anuario Estadístico. .. 39

Figura 11: Presencia del patrón Alta cohesión en el módulo Anuario Estadístico..... 40

Figura 12: Representación en por ciento (%) del nivel de responsabilidad, reutilización y complejidad de implementación de las clases..... 45

Figura 13: Representación en por ciento (%) del nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases..... 46

Figura 14: No conformidades detectadas en el SIGAE a través de las pruebas de Caja Negra..... 47

Figura 15: Resultados obtenidos en la aplicación de las pruebas de Caja Blanca. 48

ÍNDICE DE TABLAS

Tabla 1: Comparación de las aplicaciones web investigadas. 9

Tabla 2: Lista de artefactos definidos en SXP. 12

Tabla 3: Especificación del requisito funcional “Añadir Anuario Estadístico”. 32

Tabla 4: Especificación del requisito funcional “Modificar capítulo del Anuario Estadístico”.
..... 33

Tabla 5: Historia de usuario “Gestionar Anuario Estadístico”. 34

Tabla 6: Historia de usuario “Gestionar tablas” 35

Tabla 7: Plan de releases del SIGAE. 40

INTRODUCCIÓN

El mundo en la actualidad experimenta una serie de transformaciones económicas, sociales y culturales, en las cuales juegan un papel importante las Tecnologías de la Información y las Comunicaciones (TIC). El crecimiento de internet y la rápida propagación de la información a través de la red de redes, hacen conveniente para cualquier empresa o entidad disponer de sitios o portales web con el objetivo de brindar información y servicios a millones y millones de internautas en el mundo, así como poseer sistemas automatizados.

Las instituciones cubanas no están ajenas a todo este proceso de reforma que han traído consigo las TIC, haciéndose cada vez mayor el número de servicios y procesos que se informatizan con el fin de aumentar la capacidad de competencia y uso de la información. Un ejemplo de ello lo constituye la Oficina Nacional de Estadística e Información de la República de Cuba (ONEI), “entidad creada para proponer, organizar y ejecutar, según corresponda, la aplicación de la política estatal en materia de estadística del país” (ONEI, 2010). “Dentro de sus principales funciones se encuentra centralizar y emitir la estadística oficial de la Isla” (ONEI, 2010), para ello se vale de diferentes mecanismos como los Anuarios Estadísticos, los que constituyen publicaciones anuales formadas por capítulos, donde se brinda información relacionada con los aspectos más significativos de la situación demográfica, económica y social de Cuba. Estos se clasifican en municipales, provinciales y de Cuba, teniendo en cuenta que son confeccionados en las distintas dependencias de la ONEI presentes en cada uno de los municipios y las provincias del país, además de la sede central de dicha entidad, respectivamente. Una vez revisados son publicados en el portal web de la institución e impresas una determinada cantidad de copias.

Actualmente, la información de cada uno de los Anuarios se muestra a través de un conjunto de archivos que contienen tablas, gráficos e informaciones estadísticas del país o la región a la que pertenecen, agrupados por capítulos.

Sin embargo, a partir de entrevistas realizadas a los especialistas de la Dirección Nacional de la ONEI, se identificó que en dicha entidad no se cuenta con un proceso que gestione de forma simple esta publicación, haciéndose extenso el tiempo de confección, edición o visualización de la misma debido a que, para su creación, se deben copiar uno

a uno los archivos que contienen la información de cada capítulo en un directorio FTP¹ hasta que concluya el año y quede lista para su divulgación. Asimismo, para la visualización del Anuario o algunos de sus capítulos, un usuario debe proceder primero a la descarga de dichos archivos para luego poder consultar su contenido, tarea que puede resultar tediosa en la mayoría de los casos, pues no existe una vía para obtener todas las tablas estadísticas² a la vez.

Además, la ONEI no ofrece un mecanismo de búsqueda que les brinde a los usuarios que consultan la información de los Anuarios, una vía rápida de llegar a los contenidos deseados debido a que, en ocasiones, estos desconocen la estructura y/o los datos que en él se recogen.

Por otra parte, la forma de notificar los errores detectados durante el proceso de revisión de los Anuarios Estadísticos se realiza mediante llamadas telefónicas a cada una de las dependencias implicadas, con el objetivo de informar las faltas cometidas y estas den solución a las mismas.

Además de lo planteado anteriormente, la ONEI no cuenta con una base de datos que le permita guardar la información referente a los Anuarios Estadísticos, por lo que solo tiene a su disposición, en un directorio FTP, los Anuarios digitales correspondientes a los últimos tres años, imposibilitando atender, de cierta forma, las demandas de los usuarios cuando solicitan una información publicada en fechas anteriores.

La entidad también tiene la necesidad de realizar, de forma rápida y sencilla, reportes estadísticos a partir de los datos brindados por los Anuarios Estadísticos correspondientes a cada una de las dependencias de la ONEI, que muestren al usuario cómo se comportan, a nivel de país o en territorios específicos, determinados indicadores previamente definidos, así como establecer niveles de seguridad para el proceso de gestión de Anuarios, a partir del tipo de dependencia (país, provincia o municipio) con el objetivo que solo tenga acceso a la información el personal autorizado.

A partir de la problemática antes descrita se genera la necesidad de desarrollar una investigación que dé respuesta al siguiente **problema a resolver**:

¹ FTP: Protocolo de Transferencia de Archivos, por sus siglas en inglés.

² Tablas estadísticas: ficheros de tipo Excel asociados al capítulo de un Anuario Estadístico.

La forma en que se gestionan los Anuarios Estadísticos en la ONEI no contribuye a la integridad de los datos, la usabilidad del servicio y la agilidad en el desarrollo del proceso.

Tomando en cuenta el problema antes propuesto se define como **objeto de estudio:** Aplicaciones web para sistemas de gestión de información.

Determinándose como **objetivo general:** Desarrollar una aplicación web que permita la gestión de los Anuarios Estadísticos en la ONEI elevando la integridad de los datos, la usabilidad del servicio y la agilidad en el desarrollo del proceso.

Para ello se identifica como **campo de acción:** Aplicaciones web para la gestión de información estadística.

Definiéndose como **idea a defender:** El desarrollo de una aplicación web permitirá gestionar los Anuarios Estadísticos en la ONEI, elevando la integridad de los datos, la usabilidad del servicio y la agilidad en el desarrollo del proceso.

Se desglosan del objetivo general los siguientes **objetivos específicos:**

- Elaborar el marco teórico de la investigación mediante un estudio del estado del arte sobre el desarrollo de aplicaciones web para la gestión de Anuarios Estadísticos.
- Realizar el levantamiento de requisitos para el desarrollo de la aplicación web de forma que contribuya a elevar la integridad de los datos, la usabilidad del servicio y la agilidad en el desarrollo del proceso.
- Realizar el análisis y diseño de la aplicación web.
- Realizar la implementación de la aplicación web.
- Validar el correcto funcionamiento de la aplicación y que la solución propuesta dé cumplimiento al objetivo general de la investigación.

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

1. Elaboración del marco teórico del tema de investigación.
2. Realización del estudio de las herramientas para el desarrollo de aplicaciones web.

3. Identificación de los requisitos funcionales a tener en cuenta para el desarrollo de los módulos División político administrativa, Traductor de ficheros, Búsqueda, Administración, Anuario Estadístico y Reportes estadísticos.
4. Validación de los requisitos funcionales a tener en cuenta para el desarrollo de los módulos División político administrativa, Traductor de ficheros, Búsqueda, Administración, Anuario Estadístico y Reportes estadísticos.
5. Identificación de los requisitos no funcionales a tener en cuenta para el desarrollo de la aplicación web.
6. Diseño de los prototipos de IU.
7. Validación de los prototipos de IU.
8. Confección de las historias de usuarios para los módulos División político administrativa, Traductor de ficheros, Búsqueda, Administración, Anuario Estadístico y Reportes estadísticos.
9. Realización de la matriz de trazabilidad.
10. Creación de la plantilla de la aplicación a partir de las especificaciones del diseño.
11. Definición de la arquitectura a utilizar para el desarrollo de la aplicación web.
12. Realización del diseño de la base de datos de la aplicación.
13. Implementación de los módulos División político administrativa, Traductor de ficheros, Búsqueda, Administración, Anuario Estadístico y Reportes estadísticos.
14. Realización de las pruebas de validación del producto.
15. Realización del informe de la investigación.

La realización de dichas tareas se sustenta en los siguientes métodos de investigación:

Métodos Teóricos:

Histórico-Lógico: permitió realizar un estudio sobre las principales tendencias de la web con respecto a las herramientas existentes, metodologías de desarrollo y la forma en que

se brindan determinados servicios en la web, sobre todo los relacionados con la gestión y publicación de datos estadísticos, tanto en el mundo, como en el marco productivo de la universidad, con el fin de seleccionar las más apropiadas para el desarrollo del sistema.

Analítico-Sintético: posibilitó la realización del estudio teórico de la investigación haciendo más sencillo el proceso de análisis de los documentos y la extracción de los elementos fundamentales a tener en cuenta para desarrollar el sistema.

Modelación: se empleó para la confección de los prototipos funcionales, la representación de los requisitos del sistema y el diseño de la base de datos.

Métodos Empíricos:

Observación: se utilizó a través del estudio realizado a diferentes aplicaciones web que existen en el mundo que generan Anuarios Estadísticos, para determinar los elementos más comunes que están presentes en las mismas.

Medición: permitió medir la calidad de la especificación de los requisitos y el grado de ambigüedad de estos, además de obtener una medida de la calidad del diseño para su validación.

Entrevista: fue utilizado para comprender las necesidades del cliente, capturar los requisitos correspondientes al sistema y obtener información que apoye la realización de la investigación.

El contenido de este trabajo consta de tres capítulos, definidos de la siguiente forma:

Capítulo 1: Fundamentación Teórica. En este capítulo se describen una serie de conceptos relacionados con la problemática antes mencionada, así como el estudio de las características fundamentales de las aplicaciones web que generan Anuarios Estadísticos en diversas regiones del mundo incluyendo a Cuba. Además se describe la metodología que guiará el proceso de desarrollo junto al lenguaje de modelado que usará la misma, los lenguajes de programación, entre otras herramientas que serán empleadas como: Entornos de Desarrollo Integrado (IDE según sus siglas en inglés), sistemas gestores de base de datos, servidores de aplicación y las herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

Capítulo 2: Descripción y diseño del sistema. En este capítulo se realiza una descripción de las principales características del Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI, así como del diseño del mismo, partiendo del estudio del estado del arte realizado en el capítulo anterior, centrándose fundamentalmente en los requisitos funcionales y no funcionales capturados y en otros elementos importantes dentro del proceso de desarrollo, tales como: historias de usuario, arquitectura, diseño del sistema, patrones empleados e implementación.

Capítulo 3: Validación. Este capítulo se centra en las técnicas empleadas para validar los requisitos del Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI, así como en la métrica aplicada para medir la calidad de los mismos. Además, se muestran los resultados de la validación del diseño del sistema y de las pruebas de Caja Negra y Caja Blanca realizadas a esta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

En el presente capítulo se realiza un estudio de las tendencias actuales referente a la forma de gestionar los Anuarios Estadísticos en distintas instituciones del mundo. Además se describen una serie de conceptos asociados al dominio del problema, así como la metodología de desarrollo de software a utilizar, en conjunto con otros elementos tales como: el lenguaje de modelado, los lenguajes de programación, el marco de trabajo, la herramienta para modelado de prototipos web, la herramienta CASE, los sistemas gestores de base de datos, el servidor web apache y el entorno de desarrollo integrado.

1.2. Conceptos fundamentales asociados al dominio del problema.

Aplicación web: software cliente/servidor que interactúa con usuarios y sistemas utilizando el protocolo HTTP³ (*Daniel Benchimol, 2011*). Esta interacción permite realizar consultas a bases de datos, así como brindar servicios de manera efectiva.

Sitio web: conjunto de páginas web que guardan una correlación entre sí, pertenecientes a un mismo dominio, al cual se accede lógicamente con conexión a internet, empleando un navegador que permite utilizar el protocolo HTTP.

Portal web: “un portal brinda de una manera integrada contenidos y aplicaciones, con el agregado de un espacio de trabajo unificado y colaborativo, con el objeto de proveer al usuario de toda la información relevante que necesita para poder tomar decisiones de manera acertada, acorde a sus necesidades y responsabilidades” (*Ing. Javier A. Voos, 2003*). El nombre portal se asocia a la idea de que es una puerta grande a múltiples servicios y oportunidades para el usuario.

Anuario Estadístico: consiste en “recopilaciones de datos numéricos sobre una gran variedad de temas elaborados y publicados por organismos internacionales e instituciones oficiales de carácter nacional, regional o local” (*HispaNetwork Publicidad y Servicios, 2009*).

³ HTTP:Protocolo de Transferencia de Hipertexto, por sus siglas en inglés.

Integridad: consiste en que “los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada” (*Ruth Vega Cutiño, 2012*).

Usabilidad: es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema. Se puede definir además como la facilidad y simplicidad de uso de un artículo u objeto (*Unidad de Modernización y Gobierno Digital, 2014*).

1.3. Tendencias actuales.

1.3.1. Estudio de aplicaciones.

Para el desarrollo de este trabajo se llevó a cabo una investigación en aquellas aplicaciones que gestionan Anuarios Estadísticos, que pudieran constituir una solución a las necesidades de gestión de la entidad. Como resultado del estudio, se identificó que la mayoría de las soluciones analizadas, solo divulgan y mantienen actualizados los datos estadísticos. Por tal motivo, se decide estudiar la forma en que se publica y visualiza la información en las mismas, tomando como referencia los siguientes portales web: el de la Universidad de Jaén perteneciente a España, el de la Comisión Económica para América Latina (CEPAL), en Santiago de Chile, el de la Comisión Económica y Social para Asia y el Pacífico (ESCAP según sus siglas en inglés), en Tailandia, el del Instituto Nacional de Estadística y Geografía (INEGI), en México y el del Instituto Nacional de Estadística (INE), en Uruguay.

Para el análisis de los portales web estudiados a partir de los criterios definidos (forma de publicación y visualización de la información), se tuvieron en cuenta las características que cumple cada portal, tales como: si la información es publicada a través de gráficos, cuadros estadísticos y textos, y si se visualiza en el mismo portal o es necesario realizar primero su descarga.

Seguidamente, en la Tabla 1, se muestran los resultados del estudio realizado, quedando identificadas con una X las características presentes en cada portal:

Tabla 1: Comparación de las aplicaciones web investigadas.

Aplicaciones investigadas	Criterios				
	Publicación de la información			Visualización de la información	
	Gráficos	Cuadros estadísticos	Textos	En el portal	A través de la descarga
Portal web de la Universidad de Jaén.	X	X	X	X	X
Portal web de la CEPAL.		X			X
Portal web de la ESCAP.	X		X	X	X
Portal web del INEGI.	X	X	X		X
Portal web del INE.		X	X		X

Fuente: Elaboración propia a partir del estudio de los portales web pertenecientes a: la Universidad de Jaén, CEPAL, ESCAP, INEGI, INE.

De manera general, se identificó que la mayoría de los portales web analizados cumplen con al menos 3 de las características definidas, por tal motivo se decide tener en cuenta el uso de estas en el desarrollo del Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI (SIGAE).

1.4. Metodología de desarrollo de software.

Una metodología de desarrollo de software se define como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software (Roger S. Pressman, 2001).

Actualmente existen un conjunto de metodologías, las cuales son clasificadas en tradicionales y ágiles. Por un lado, las tradicionales presentan grandes grupos de trabajo y posiblemente distribuidos, conllevando a la existencia de una mayor cantidad de roles, además se caracteriza por la interacción del cliente con el equipo de desarrollo a través de reuniones y evidenciarse la resistencia al cambio; por otra parte, en las metodologías ágiles, la cantidad de roles es menor debido a que los grupos de trabajo son pequeños

(menos de diez integrantes), aspecto que posibilita una mayor interrelación entre los implicados, además, el equipo de desarrollo está preparado para cambios durante el proyecto, siendo el cliente un miembro importante dentro del mismo.

A raíz de lo investigado anteriormente y el estudio de la metodología que utilizan los centros y proyectos de desarrollo de software en la Universidad de las Ciencias Informáticas, se decide utilizar en el desarrollo del SIGAE, una metodología ágil, ya que se cuenta con un grupo de trabajo pequeño y el cliente (ONEI) es parte del equipo.

Dentro de las metodologías ágiles se destacan Scrum, XP (*Extreme Programming* o Programación Extrema), ambas con gran éxito en el mundo de desarrollo del software y SXP, siendo esta última una mezcla de las dos antes mencionadas y “desarrollada en el 2008 por UNICORNIOS: un grupo de proyectos de investigación y desarrollo que formó parte del polo productivo de Software Libre (SWL) de la UCI” (*Lisandra Isabel Leyva Samada, 2009*). Seguidamente se explica en qué consiste cada una de ellas:

Scrum es una metodología ágil en la cual se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto, es por ello que está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos y la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (*ProyectosAgiles.org Team, 2014*). Esta metodología se caracteriza además por la duración de una iteración, que puede ser desde 15 hasta 30 días, proporcionando un incremento del producto final que sea susceptible de ser entregado cuando el cliente lo solicite y por contar con fases tales como: conceptualización, negocio, requerimientos, diseño e implementación y prueba.

XP, por su parte, es una metodología ágil centrada en potenciar las relaciones interpersonales como punto clave para el éxito en el desarrollo de software, “promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo” (*Infored, 2014*). La misma se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, así como una comunicación fluida entre todos los participantes; también presenta “simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico” (*Infored, 2014*). Además de ello, esta metodología es

propicia cuando se trata de proyectos a corto plazo, en donde cada iteración puede durar cerca de 2 meses. La programación extrema también se caracteriza por tener presentes aspectos como la refactorización⁴ y la programación en parejas⁵ y por contar con fases tales como: exploración, planificación de la entrega, iteraciones, producción, mantenimiento y muerte del proyecto.

SXP es un híbrido cubano de metodologías ágiles que tiene como base a Scrum y XP, centrándose en la primera para la gestión eficiente de proyectos y en la segunda para la ingeniería de software. La misma toma características de Scrum como: la duración de una iteración (desde 15 hasta 30 días) y que el equipo de desarrollo no permite cambios durante la misma; mientras que de XP posee elementos tales como: la programación en parejas, la refactorización y que el cliente sea quien determine la prioridad de las tareas. “Consta de 4 fases principales: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos y actividades que generan artefactos” (*Gladys Marsi Peñalver Romero, y otros, 2010*).

A partir de los elementos citados sobre las metodologías ágiles analizadas, los autores de la presente investigación deciden emplear como metodología de desarrollo de software SXP, ya que toma los mejores elementos de SCRUM y XP, tales como: la gestión eficiente de proyectos y las tareas de ingeniería respectivamente. La integración de estos aspectos permite alcanzar mayores resultados y tributan a la obtención de un producto de mayor calidad que satisfaga las necesidades del cliente.

En sentido general, esta metodología ayuda a fortalecer el trabajo en equipo, logrando la unidireccionalidad en el mismo y claridad en los objetivos. Por otra parte, permite seguir el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando así el nivel de interés del equipo (*Gladys Marsi Peñalver Romero, y otros, 2010*).

Seguidamente, se muestran más detalladas cada una de las fases de SXP antes mencionadas y en la Tabla 2 los artefactos que genera dicha metodología:

4 Refactorización: consiste en mejorar el código previamente escrito, cambiando su estructura interna sin modificar su comportamiento externo.

5 Programación en parejas: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Fases de SXP:

- **Planificación-Definición:** se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** se pone en marcha el producto desarrollado y se hace la entrega al cliente.
- **Mantenimiento:** se realiza el soporte para el cliente.

Tabla 2: Lista de artefactos definidos en SXP.

Artefactos definidos en SXP	
1	Concepción del sistema
2	Modelo de historia de usuario del negocio
3	Lista de reserva del producto
4	Historia de usuario
5	Lista de riesgos
6	Modelo de diseño
7	Tareas de ingeniería
8	Plan de <i>releases</i>
9	Estándar de programación
10	Caso de prueba de aceptación
11	Manual de usuario
12	Manual de identidad
13	Manual de desarrollo
14	Gestión de cambios

Fuente: Tomado de (Gladys Marsi Peñalver Romero, y otros, 2010).

1.5. Lenguaje de Modelado Unificado 2.0.

El Lenguaje de Modelado Unificado (*Unified Modeling Language (UML)*), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, respaldado el mismo por el OMG (*Object Management Group*). UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas

de clases y objetos, hasta la implementación y configuración con los diagramas de despliegue (*Enrique Hernández Orallo, 2012*).

La versión 2.0 de este lenguaje cuenta con muchas novedades y cambios que, fundamentalmente, se centran en resolver carencias prácticas. UML 2.0 posee 13 tipos de diagramas, los cuales se clasifican en los siguientes grupos para una mejor comprensión de los mismos: diagramas de estructura (diagrama de clases, componentes, objetos, estructura compuesta, despliegue y paquetes) y los diagramas de comportamiento (diagrama de actividades, casos de uso, estados, interacción, secuencia, comunicación, tiempos).

A pesar de la existencia de otras notaciones como BPMN (Notación para el Modelado de Procesos de Negocio) enfocadas al modelado de procesos, la gestión de Anuarios Estadísticos en la ONEI es un proceso sencillo y fácil de comprender, resultando conveniente usar UML para la modelación del negocio y la generación de artefactos de la metodología. Además este lenguaje facilita la representación de conceptos de la programación orientada a objetos.

1.6. Herramienta para el modelado de prototipos web.

Con el objetivo de validar los requisitos funcionales del sistema, son elaborados una serie de prototipos funcionales. Estos se centran en la representación de aquellos aspectos del sistema que serán visibles para el personal de la ONEI, los cuales al ser evaluados permitirán la retroalimentación del equipo de desarrollo. Para la confección de dichos prototipos existe una variada colección de herramientas, dentro de las cuales se destaca Axure RP Pro.

Axure RP Pro es una herramienta de especificación de software, orientada al diseño de aplicaciones web y de escritorio, que permite hacer prototipos funcionales con un alto grado de similitud al producto final. Además, es muy útil para validar su usabilidad con el cliente, posibilitando la exportación del trabajo como archivo HTML para esa validación, por lo que puede ser visto en cualquier navegador. Por otra parte, la herramienta facilita la reutilización de elementos como: los encabezados, los pies de página y las plantillas. Debido a la gran aceptación a nivel mundial que tiene Axure RP Pro, se pueden encontrar en internet diversas librerías que le pueden aportar muchos elementos al sistema.

Por las características planteadas anteriormente, se decide utilizar esta herramienta en su versión 5.5 para el desarrollo del SIGAE.

1.7. Herramientas de Ingeniería de Software Asistida por Computadora.

Se puede definir a las herramientas de Ingeniería de Software Asistida por Computadora (CASE) como “el nombre que se le da al software que se utiliza para ayudar a las actividades del proceso del software como la ingeniería de requerimientos, el diseño, el desarrollo de programas y las pruebas. Por lo tanto, las herramientas CASE incluyen editores de diseño, diccionarios de datos, compiladores, depuradores, herramientas de construcción de sistemas” (*Ian Sommerville, 2005*), permitiendo al software mejorar en cuanto a calidad y productividad. Actualmente existen diversas herramientas CASE, ejemplo de ello lo constituye: Erwin, EasyCASE, Oracle Designer y Visual Paradigm, siendo esta última una de las más utilizadas para el desarrollo de sistemas en los proyectos productivos de la UCI.

“Visual Paradigm es una herramienta CASE de modelado visual que facilita la construcción de artefactos en un proceso de desarrollo de software mediante UML. Soporta una amplia gestión de casos de uso y diseño de base de datos relacionales y proporciona medidas más eficaces en el análisis y diseño de sistemas. Algunas de las funcionalidades que brinda la herramienta son la generación de código y de base de datos a partir de los diagramas UML realizados, la realización de ingeniería inversa, generación automática de informes en formato PDF, Word o HTML; generación de máquinas de estados, integración con Visio y *Rational Rose*” (*Visual Paradigm Team, 2014*). Para el desarrollo del presente trabajo se selecciona dicha herramienta en su versión 8.0.

1.8. Lenguajes.

Un lenguaje de programación es un “conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias” (*Jose Luis Lira Turriza, 2007*).

A continuación se describen los lenguajes que serán utilizados en el desarrollo del SIGAE:

1.8.1. Preprocesador de Hipertexto 5.3.

Sus siglas responden a un acrónimo recursivo (PHP: *Hypertext Preprocessor*), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. Además es un lenguaje ideal para desarrollar aplicaciones web complejas, añadiendo a todo eso la ventaja de que su intérprete, sus diversos módulos y la gran cantidad de librerías desarrolladas para el mismo, son de código libre, por lo que el programador de PHP dispone de un impresionante arsenal de herramientas de este tipo para desarrollar aplicaciones (*Carles Mateu, 2004*). Las características más potentes y destacables de PHP son: su soporte para una gran cantidad de bases de datos y la posibilidad de hacer uso de otros servicios que empleen protocolos como: IMAP⁶, SNMP⁷, NNTP⁸, POP3⁹, HTTP y derivados. También se pueden abrir sockets¹⁰ de red directos (*raw sockets*) e interactuar con otros protocolos (*Bakken, y otros, 2001*).

1.8.2. Lenguaje de Marcado de Hipertexto 5.

Las siglas HTML¹¹, hacen referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenidos de una página web, como textos e imágenes. Es un estándar a cargo de la W3C¹², organización dedicada a la estandarización de la mayoría de las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web (*Sergio Luján Mora, 2001*).

1.8.3. Hojas de Estilo en Cascada.

“Las Hojas de Estilo en Cascada (CSS) fueron creadas para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML¹³. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear

⁶ IMAP: Protocolo de Acceso a Mensajes Electrónicos, por sus siglas en inglés.

⁷ SNMP: Protocolo Simple de Administración de Red, por sus siglas en inglés.

⁸ NNTP: Protocolo para la Transmisión de Noticias en Red, por sus siglas en inglés.

⁹ POP3: Protocolo de Oficina de Correo u Oficina Postal.

¹⁰ Sockets: es un método para la comunicación entre un programa del cliente y un programa del servidor en una red.

¹¹ HTML: Lenguaje de Marcado de Hipertexto o HyperText Markup Language.

¹² W3C: El World Wide Web Consortium (W3C) es un consorcio internacional que desarrolla estándares que aseguran el crecimiento de la web a largo plazo.

¹³ XHTML: Lenguaje de Marcado de Hipertexto Extensible, por sus siglas en inglés.

páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes" (*Javier Eguiluz Pérez, 2008*).

1.8.4. JavaScript.

JavaScript es "un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios" (*Javier Eguiluz, 2014*). "Es ampliamente utilizado para la validación básica de formularios, ya que es más práctico validar un formulario en el lado del cliente que hacer constantes peticiones al servidor" (*Web Developers Notes Team, 2014*).

"Las dos principales características de JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones), y por otro, JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza. Esto implica que gran parte de la programación en JavaScript se centra en describir objetos (con sus variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos" (*Víctor M. Rivas Santos, 2014*).

1.9. Librerías para la conversión de los formatos Excel a HTML y HTML a PDF.

Con el objetivo de conocer si existen herramientas que permitan obtener los datos de un fichero Excel y mostrarlos en formato HTML, así como exportar la información extraída de ficheros Excel a un archivo PDF¹⁴, se realiza un estudio sobre diferentes librerías que permiten realizar estas operaciones. Seguidamente se describen algunas de las más utilizadas para convertir ficheros en formato Excel a HTML:

¹⁴ PDF: sigla del inglés Portable Document Format, Formato de Documento Portátil.

- **PHPExcel:** es una librería escrita en PHP¹⁵, integrada por un conjunto de clases que permiten escribir y leer diferentes formatos de archivos de hojas de cálculo como .xls, .xlsx, .csv, .ods, PDF, HTML (*Microsoft Corporation, 2014*).
- **JExcel:** es una librería escrita en Java que permite trabajar con hojas de cálculo realizadas en Excel. Además proporciona la Clase *JWorkbook* que permite incluir un libro Excel dentro de la aplicación realizada en Java (*Dice Holdings, Inc, 2014*).
- **ClosedXML:** es una magnífica librería para .NET basada en el estándar Open XML que permite la generación de archivos Excel con formato, fórmulas, rangos, filtros y otros (*Microsoft Corporation, 2014*).

Para la conformación del PDF a partir del contenido extraído de varias tablas en HTML se encontraron librerías tales como:

- **TCPDF:** es una librería escrita en PHP que permite crear ficheros PDF dinámicamente. Dos de las cualidades más apreciadas de esta clase son: su simplicidad a la hora de crear archivos PDF y la capacidad de interpretar código HTML (*Nicola Asuni, 2013*).
- **Itext:** es una librería para crear y manipular archivos PDF, RTF y HTML en Java. El mismo documento puede ser exportado en múltiples formatos o múltiples instancias del mismo formato. Los datos pueden ser escritos a un fichero (*iText Group NV, 2014*).

A partir de la selección de PHP como lenguaje de desarrollo, se definen para la conversión de ficheros en el SIGAE, las librerías PHPExcel y TCPDF.

1.10. Marco de trabajo.

“Un marco de trabajo es un conjunto de componentes físicos y lógicos estructurados de tal forma que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de Información” (*Carlos Andrés Guerrero Alarcón, y otros, 2009*). Además, simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, proporcionando estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener (*Fabien Potencier, y otros, 2008*).

A continuación se describen algunas de las características fundamentales de los marcos de trabajo, las cuales hacen de los mismos una potente herramienta de desarrollo.

¹⁵ PHP: Preprocesador de Hipertexto o Hypertext Preprocessor por sus siglas en inglés.

- **Reducción de costos en licencias:** la mayoría de los componentes reutilizables desarrollados para un marco de trabajo son de código abierto.
- **Facilitan la creación de formularios:** cuentan con herramientas con el fin de gestionar formularios, permitiendo validar los objetos que son manipulados por los mismos.
- **Incluyen la Implementación de controladores:** generalmente incluyen una serie de controladores implementados, los cuales pueden ser perfectamente adaptables a las necesidades del negocio.
- **Incluye mecanismos para la autenticación y el control de acceso:** facilitan el trabajo con los mecanismos para la identificación de usuarios, con el objetivo de garantizar que los mismos solo accedan a los componentes del sistema a los que está autorizado su rol.
- **No es necesario manipular las URL¹⁶, ni las sesiones:** permiten abstraerse de trabajar con ambos elementos.
- **Gestionan el acceso a datos:** contienen las herramientas y conectores adecuados para la integración con los gestores de base de datos más reconocidos.

Para la selección del marco de trabajo a utilizar en el desarrollo del SIGAE, se tuvo en cuenta, entre otros factores, que el lenguaje de programación del lado del servidor seleccionado es PHP, por lo tanto se hace necesario optar por el uso de un marco de trabajo acorde al mismo.

Según las encuestas del sitio web *PHP Frameworks*, dedicado específicamente al estudio de este tipo de tecnología, los marcos de trabajo para PHP más reconocidos por la comunidad de desarrolladores en la actualidad son: Yii Framework, CodeIgniter, Zend Framework, CakePHP y Symfony, información que se muestra en la Figura 1.

¹⁶ URL: Localizador de Recursos Uniforme, por sus siglas en inglés.

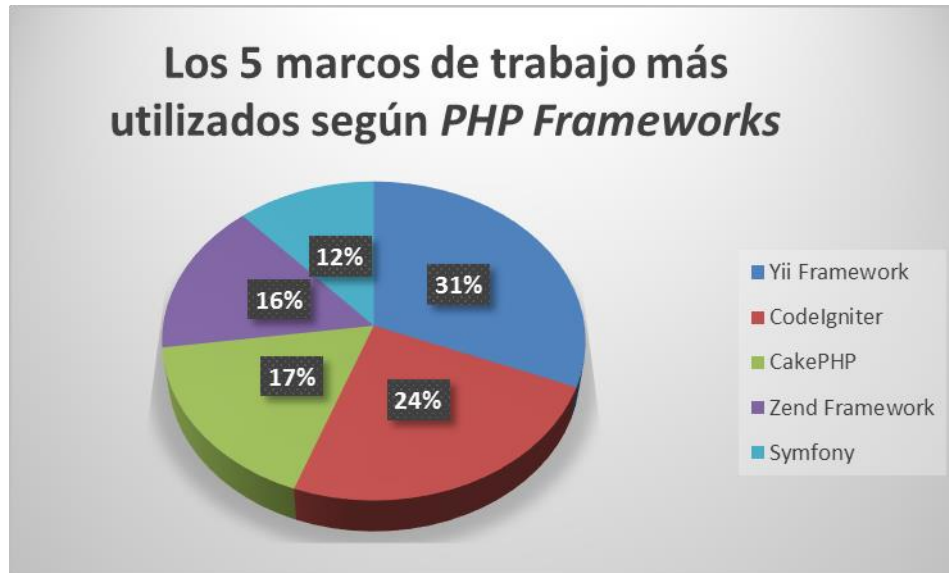


Figura 1: Top de uso de los marcos de trabajo para PHP según *PHP Frameworks*.

Fuente: Elaboración propia a partir de los datos recopilados en (*PHP Frameworks Team, 2013*).

Otros sitios web de diversas comunidades de desarrolladores muestran también resultados similares en sus estudios de tendencias. Tal es el caso de *Webcoderpro*, donde se encuentra publicado el artículo: *TOP 6 most popular PHP frameworks of 2013*, realizado a partir de las estadísticas de *Google Trends*¹⁷. En otro apartado, algunos analistas de la web como Bruno Skvorc prevén un excelente año desde ya para tres marcos de trabajo, fundamentalmente: Laravel, Phalcon y Symfony 2, sin dejar de esperar buenos resultados de Yii y CodeIgniter.

A pesar de la divergencia de opiniones sobre cuál marco de trabajo es más práctico o más flexible, o sobre cuál seleccionar para el desarrollo de un sistema determinado, lo cierto es que cada uno de ellos cuenta con crecientes comunidades que respaldan el desarrollo de soluciones, por lo que se garantiza el constante flujo informativo y de documentación generado por las mismas. No obstante, todo marco de trabajo necesita el tránsito por una curva de aprendizaje por parte del desarrollador, por lo que se hace necesario primero invertir tiempo y esfuerzo en conocer y aprender a desarrollar, antes de poder utilizar cualquiera de estas herramientas.

En varios de los centros productivos de la Universidad de las Ciencias Informáticas se emplean marcos de trabajo para PHP, ejemplo de ello lo constituyen la plataforma

¹⁷ Google Trends es una herramienta de Google Labs que muestra los términos de búsqueda más populares del pasado reciente.

educativa ZERA junto a Croda, Roda y la colección El Navegante del centro FORTES. También se encuentra el caso de los proyectos para la Informatización de los Tribunales Populares Cubanos y la Fiscalía General de la República de Cuba del centro CEGEL, predominando en estos sistemas el uso del marco de trabajo Symfony.

Symfony es un completo marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (*Fabien Potencier, 2008*).

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix y Linux) como en plataformas Windows (*Fabien Potencier, 2008*).

Por otra parte, es el que más ideas incorpora del resto de los marcos de trabajo, incluso de aquellos que no están programados con PHP, ejemplo de ello lo constituyen: Ruby On Rails, Django o Spring, los cuales presentan muchas similitudes con algunos de los componentes de Symfony (*Javier Eguiluz, 2011*).

Además permite la automatización de un grupo de características para proyectos web que lo hace ser una buena elección para el desarrollo ágil de sistemas, como por ejemplo:

- La capa de internacionalización que incluye, permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas que pueden ser creadas por diseñadores HTML sin ningún tipo de conocimiento del marco de trabajo.
- Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.

- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores (*Fabien Potencier, 2008*).

De acuerdo a los aspectos antes mencionados y teniendo en cuenta su amplia repercusión tanto a nivel internacional como en los proyectos productivos de la UCI, además de los excelentes materiales bibliográficos existentes, los cuales son actualizados constantemente y puestos a disposición de la comunidad de forma gratuita, así como el hecho de ser una herramienta libre para la cual se desarrollan cada día numerosos componentes, se decide utilizar en el desarrollo del SIGAE, el marco de trabajo Symfony, específicamente en su versión 2.3 por su flexibilidad, además de incluir un nuevo grupo de conceptos y tecnologías como: Motor de Plantillas y el uso de *Bundles*¹⁸, lo cual hace que esta versión sea más potente.

Por otra parte, este marco de trabajo emplea patrones de diseños pertenecientes al “Grupo de los Cuatro” (*Gang of Four (GoF)*, por su traducción al inglés), los cuales se encargan del proceso de creación de clases y objetos, la composición de los mismos, así como la forma en que interactúan y distribuyen sus responsabilidades. En general, estos patrones contribuyen a que el sistema sea independiente de cómo sus objetos son creados, integrados y representados, permitiendo reducir el acoplamiento entre los objetos y garantizando que la modificación en los requisitos del sistema ocasione la menor cantidad posible de cambios en las relaciones entre los objetos existentes. Seguidamente se muestran algunos de los patrones GoF que implementa Symfony 2, los cuales facilitarán la implementación del SIGAE:

Método de fabricación: parte del principio de separar la clase que crea los objetos, de la jerarquía de objetos a instanciar. Su utilización hace al diseño ganar en flexibilidad pues, es mucho más sencillo crear los objetos de la forma que plantea este patrón, que haciéndolo directamente. Se evidencia en la posibilidad que ofrece el marco de trabajo de crear objetos haciendo uso del contenedor de servicios, sin necesitar crear directamente una instancia del mismo.

¹⁸ El concepto de *Bundle* se asocia con un conjunto de funcionalidades que pueden ser instalados en cualquier proyecto que se desarrolle con Symfony y funcionar de forma independiente.

Constructor: plantea abstraerse de la construcción de un objeto complejo de forma tal que el mismo proceso de construcción pueda crear diferentes instancias. Su uso fomenta la reutilización del diseño, su flexibilidad y potencia el encapsulamiento. Se puede observar en la creación de formularios de Symfony 2 al utilizar la clase *FormBuilder*.

Además de los antes mencionados, Symfony 2 implementa otros patrones de diseño tales como:

Controlador frontal: consiste en tener un punto de entrada centralizado para el manejo de todas las solicitudes que son realizadas al sistema. El uso de este patrón aumenta la reusabilidad del código y mejora la gestión de la seguridad del sistema. En Symfony 2 dicha tarea la realizan las clases *App* y *App_dev*.

Inyección de dependencias: se manifiesta en el hecho de suministrar objetos a una clase en lugar de ser la propia clase quien cree los objetos. Symfony 2 lo utiliza a través del “contenedor de inyección de dependencias”, un objeto que tiene la información necesaria para crear objetos del sistema, para lo cual prepara todas las dependencias entre clases. Su utilización potencia en gran medida la flexibilidad y el encapsulamiento. En la clase *appDevDebugProjectContainer* están contenidos todos los servicios y parámetros de configuración del contenedor de inyección de dependencias del marco de trabajo.

1.10.1. Mapeo Objeto-Relacional.

“El Mapeo Objeto-Relacional (más conocido por su nombre en inglés, *Object-Relational Mapping*, o sus siglas ORM) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional, utilizando un motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la programación orientada a objetos” (*Hibernate Team, 2014*).

El uso de ORM incluye una serie de ventajas para el desarrollo de sistemas. Es importante mencionar que brinda: rapidez en el desarrollo ante la posibilidad de abstraerse de la base de datos relacional, seguridad frente a inyecciones SQL¹⁹, un

¹⁹ Inyecciones SQL: es una técnica donde un atacante crea o altera comandos SQL existentes para manipular la información de la base de datos de un sistema.

lenguaje propio para la realización de consultas, así como facilidades para el mantenimiento del código.

Para el desarrollo del SIGAE será utilizado **Doctrine 2.1** por ser la librería ORM soportada por Symfony 2.3. “Una de las características clave de Doctrine 2 es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos llamado Lenguaje de Consulta Doctrine (DQL por *Doctrine Query Language*), inspirado en Hibernates HQL²⁰. Además DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible” (*Doctrine Team, 2014*).

1.10.2. Motor de plantillas.

El uso de un Motor de plantillas le permite a los desarrolladores separar el código PHP del código HTML, lo cual hace más sencillo el desarrollo de la interfaz del sistema. Para el lenguaje PHP existen múltiples motores de plantillas, algunos de los más populares son Smarty, Template Sigma y Twig.

Para el desarrollo del SIGAE, será utilizado Twig como motor de plantillas para PHP, pues junto al hecho de ser rápido y eficiente, es soportado por Symfony. Además se le atribuye la condición de haber sido creado por el propio Fabien Potencier, autor y máximo responsable de Symfony, quien recomienda a Twig para el uso del marco de trabajo.

Twig se caracteriza fundamentalmente por ser:

- **Rápido:** compila las plantillas hasta código PHP regular optimizado. El costo general en comparación con código PHP regular se ha reducido al mínimo.
- **Seguro:** tiene un modo de recinto de seguridad para evaluar el código de plantilla que no es confiable.
- **Flexible:** es alimentado por flexibles analizadores léxico y sintáctico. Esto permite al desarrollador definir sus propias etiquetas y filtros personalizados, y crear su propio DSL²¹ (*Fabien Potencier, 2011*).

²⁰ *Hibernates HQL*: es el lenguaje propio para realizar consultas del ORM *Hibernate*.

²¹ DSL: Lenguaje Específico del Dominio (en inglés *domain-specific language*, DSL).

1.11. Sistema Gestor de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) se puede definir como el conjunto de programas, procedimientos y lenguajes que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, manteniendo su integridad, confidencialidad y seguridad (*Hernan Alberto Silva Rodríguez, 2006*).

Para el desarrollo de aplicaciones se utilizan disímiles SGBD. Según el *Ranking de DB-Engines*, espacio en la web dedicado a la recopilación de información acerca de los SGBD, los más utilizados son: Oracle, MySQL, Microsoft SQL Server y PostgreSQL, tal como se puede apreciar en la Figura 2.

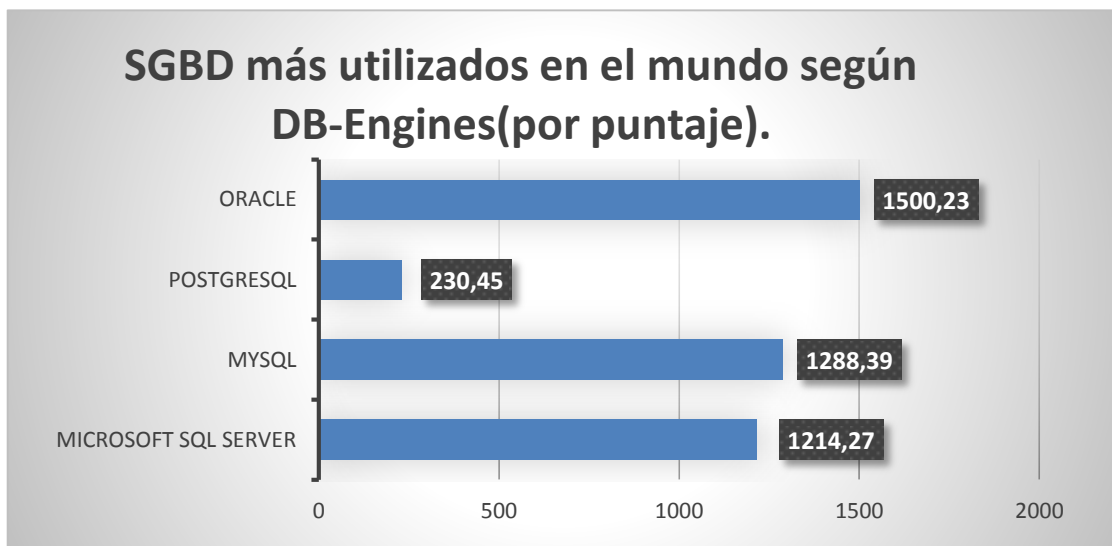


Figura 2: SGBD más utilizados en el mundo según *DB-Engines* (por puntaje).

Fuente: Elaboración propia a partir de los datos recopilados en (*solid IT, 2014*).

A pesar de la gran potencialidad de los SGBD Oracle y Microsoft SQL Server, ambos son privativos y poseen costosas licencias, por lo que su uso atenta contra la Política de Soberanía Tecnológica que lleva a cabo el país, no siendo así el caso de PostgreSQL y MySQL. En cuanto a este último, vale destacar que está enfocado a la optimización del rendimiento, característica que se le atribuye a partir de su diseño multihilo, además presenta una excelente integración con PHP y un eficiente control de acceso. Por su parte, PostgreSQL a pesar de consumir muchos más recursos y ser menos intuitivo para los desarrolladores, incluye características de la orientación a objetos como: la herencia,

tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es importante mencionar que tanto MySQL como PostgreSQL son soportados por el marco de trabajo Symfony.

Teniendo en cuenta los aspectos mencionados con anterioridad, su sencillez de uso, así como los numerosos proyectos que han sido realizados en la UCI con el uso de este gestor de base de datos, incluyendo el propio portal de la Oficina Nacional de Estadística e Información, se selecciona como SGBD a MySQL en su versión 5.5.16, para el desarrollo del SIGAE.

Para el diseño de la base de datos, MySQL permite emplear un conjunto de patrones que contribuyen a que la misma sea más fortalecida y consistente. Seguidamente se describen algunos de los que pudiesen constituir una alternativa para ello:

Árbol simple: permite representar una estructura de datos en forma de árbol donde los elementos que serán almacenados son de un mismo tipo.

Llaves subrogadas: es ampliamente utilizado pues facilita en gran medida la interacción con la base de datos y reduce el costo de las búsquedas. El mismo plantea la utilización de una llave única para cada entidad, en vez de usar un atributo identificador en el contexto dado.

Árboles fuertemente codificados: este patrón facilita la representación de jerarquías donde es bien conocida la estructura y cobra importancia la correspondencia existente entre las relaciones, para lo cual a cada nivel del árbol se le asocia una entidad. Generalmente las relaciones entre dichas entidades son de uno a muchos.

1.12. Servidor web Apache.

Un servidor web es un “programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS²²” (*Carles Mateu, 2004*).

“Apache es un servidor web de código libre, robusto, cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo (*Apache Group*) que, usando Internet y la web para comunicarse,

²² HTTPS: versión segura, cifrada y autenticada de HTTP.

planifican y desarrollan el servidor y la documentación relacionada” (*Carles Mateu, 2004*). Además se caracteriza por ser un servidor web que posee gran rapidez, goza de mucha eficiencia y flexibilidad y se adapta a los nuevos protocolos HTTP. Todo ello se une al hecho de ser multiplataforma, adaptable a diferentes entornos y necesidades y extensible a partir de su condición de modularidad.

Por las razones antes mencionadas se selecciona Apache en su versión 2.0, como servidor para el desarrollo del SIGAE.

1.13. Entorno de Desarrollo Integrado.

Un Entorno de Desarrollo Integrado, traducido del inglés *Integrated Development Environment* (IDE), es un “programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Los componentes de cualquier entorno de desarrollo integrado son un editor de texto, un compilador, un intérprete, un depurador y un constructor de interfaces gráficas de usuario (GUI)” (*Programación Desarrollo Team, 2014*).

Para el desarrollo de aplicaciones web con el lenguaje PHP son utilizados diversos IDE. Algunos de ellos permiten la integración con el marco de trabajo Symfony, destacándose en ese caso NetBeans y PHPStorm. El último de estos, a pesar de sus grandes potencialidades como IDE, es una solución privativa, desventaja que presenta ante su similar NetBeans que sí es un “producto libre sin restricciones de uso” (*NetBeans Team, 2012*).

NetBeans es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans (*Oracle Corporation, 2013*). A ello se le suma también el ser multiplataforma, poseer un excelente auto-completamiento de código y un buen rendimiento.

Teniendo en cuenta los elementos antes mencionados, así como su frecuente uso en los proyectos productivos de la UCI, se selecciona NetBeans como IDE en su versión 7.4 para el desarrollo del SIGAE.

1.14. Conclusiones parciales.

- El análisis realizado a los conceptos fundamentales asociados al dominio del problema, permitió una mejor interpretación del marco teórico de la investigación.
- El estudio de aplicaciones web relacionadas con la gestión de Anuarios Estadísticos, contribuyó a identificar la forma en que se publica y visualiza la información en las mismas.
- La metodología seleccionada permite proveer de eficiencia y rapidez al proceso de desarrollo del SIGAE.
- El estudio y selección de las herramientas utilizadas permiten obtener un sistema en correspondencia con las necesidades del cliente.

CAPÍTULO 2: DESCRIPCIÓN Y DISEÑO DEL SISTEMA

2.1. Introducción.

En este capítulo se exponen los requisitos funcionales y no funcionales, arquitectura, patrones y aspectos del diseño, tenidos en cuenta en la implementación del Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI.

2.2. Descripción del sistema.

El Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI estará enfocado a reducir la complejidad de confección de dicha publicación, facilitando la gestión de tablas, capítulos y separatas referentes a los Anuarios Estadísticos de las dependencias de la ONEI, en el cual se definirán niveles de usuarios para el acceso al mismo. Permitirá además la revisión de las tablas estadísticas y la posibilidad de enviar los errores detectados mediante notificaciones. Asimismo, posibilitará gestionar la división política administrativa del país, haciendo flexible el sistema ante cambios en la estructura de la Isla.

Además, el SIGAE estará encaminado a mejorar la experiencia de usuario al proporcionarle un conjunto de opciones tales como: visualizar el contenido de las tablas estadísticas directamente en el sistema sin necesidad de descargarlas, facilitar la descarga de una tabla, un capítulo y/o el Anuario completo, la existencia de un mecanismo de búsqueda de información y la realización de comparaciones entre diferentes Anuarios teniendo en cuenta los indicadores previamente seleccionados para ello.

Por otra parte, con el objetivo de restringir el acceso a las opciones de gestión que presenta el sistema, se definen los siguientes niveles de usuarios:

- **Administrador:** posee control total del sistema, por lo que puede realizar cualquier acción sobre los módulos que presenta la misma.
- **Usuario de tipo país:** puede gestionar los Anuarios Estadísticos del país, así como revisar los Anuarios de las provincias y municipios correspondientes al mismo.
- **Usuario de tipo provincia:** puede gestionar Anuarios Estadísticos de su provincia, así como revisar los Anuarios de los municipios correspondientes a la misma.
- **Usuario de tipo municipio:** puede gestionar Anuarios Estadísticos de su municipio.

En la Figura 3 se muestra la estructura que tendrán todas las interfaces de la administración del sistema, la cual se encarga de la confección de los Anuarios Estadísticos, mientras que en la Figura 4, se representa la pantalla tipo para la vista del usuario, donde se mostrará el contenido de los Anuarios Estadísticos.

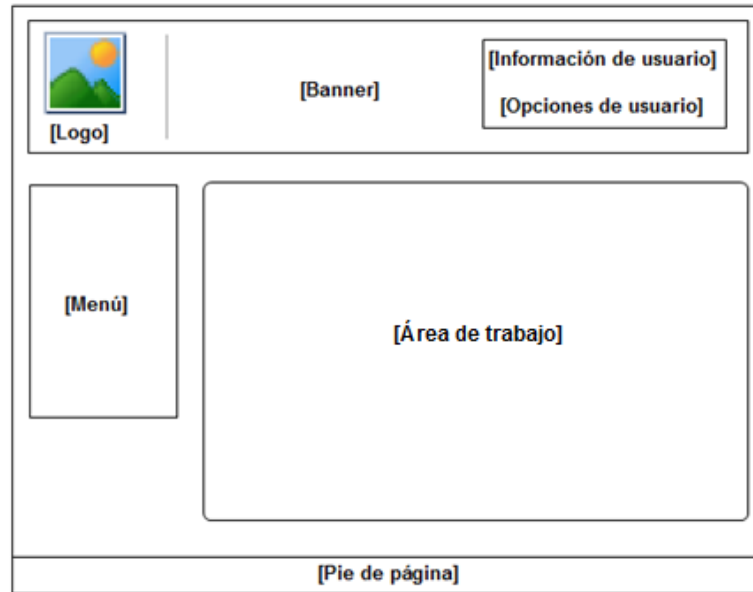


Figura 3: Pantalla tipo para la administración del sistema.

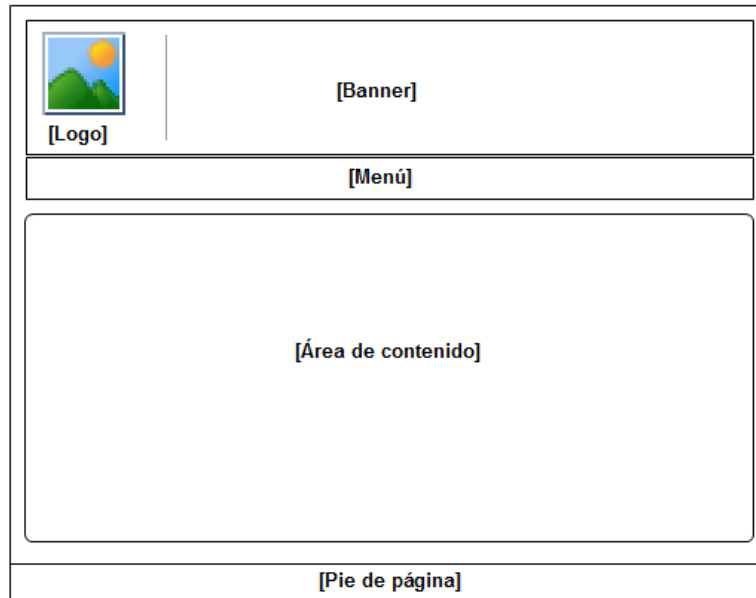


Figura 4: Pantalla tipo para la vista del usuario.

2.3. Requisitos del software.

2.3.1 Requisitos funcionales.

Los requisitos funcionales (RF) son aquellos que describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento; estos definen qué debe hacer el sistema para satisfacer las necesidades de la entidad (*Universidad de Granada, 2014*).

A través de entrevistas con el cliente, se obtuvieron un total de 62 RF (ver Tabla 8 del Anexo 1) para el desarrollo del sistema, los cuales fueron agrupados en los siguientes módulos con el objetivo de lograr mejor organización y comprensión del trabajo:

- **División político administrativa:** permite al administrador del sistema añadir, modificar, filtrar, listar y eliminar países, provincias y municipios, así como ver las provincias de un país y los municipios pertenecientes a una provincia.
- **Traductor de ficheros:** permite al usuario visualizar el contenido de los Anuarios Estadísticos en la propia aplicación, además de posibilitarle la descarga del propio Anuario, de un capítulo y/o de un fichero determinado asociado al mismo.
- **Búsqueda:** permite al usuario hacer búsquedas dentro de un Anuario Estadístico específico, empleando para ello las palabras claves que presentan las tablas y los capítulos asociados al mismo.
- **Reportes estadísticos:** muestra un reporte de los Anuarios Estadísticos con los indicadores seleccionados por el usuario, que le van a permitir a este ver el comportamiento de esos indicadores en los diferentes territorios (ya sean todos o si se especifican) de Cuba.
- **Anuario Estadístico:** permite a los usuarios añadir, modificar, filtrar, listar, eliminar, importar, exportar, visualizar, revisar y descargar los Anuarios Estadísticos, así como los capítulos y las tablas asociadas al mismo. Además facilita añadir, modificar, listar, eliminar y descargar separatas²³.
- **Administración:** permite al administrador del sistema gestionar usuarios (añadir, modificar, listar, eliminar, asignar permisos, configuración del perfil e iniciar sesión) y gestionar notificaciones (redactar, listar, eliminar, enviar, leer notificación y activar/desactivar notificación).

²³ Separata: tirada aparte de un capítulo o artículo de una publicación. Publicación complementaria del Anuario Estadístico.

Los módulos descritos anteriormente serán implementados a través del uso de las herramientas citadas en el Capítulo 1. Seguidamente, en la Figura 5, se muestra la interacción entre cada uno de ellos.

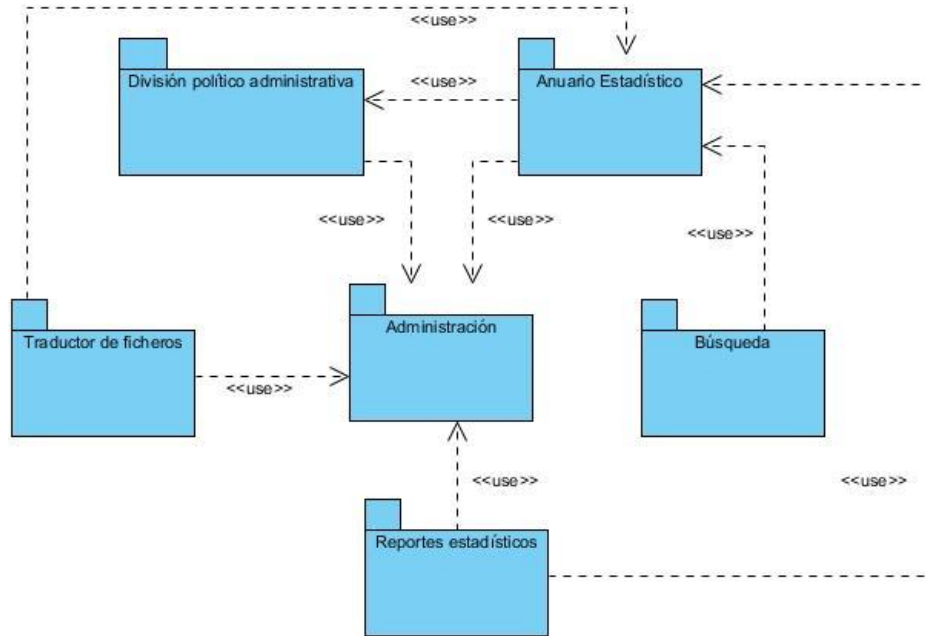


Figura 5: Diagrama de módulos del sistema.

Entre los módulos del sistema, Anuario Estadístico es el más crítico y complejo del negocio, al contener la mayor cantidad de requisitos funcionales y ser de alta prioridad para el cliente. Por tanto, será utilizado como objeto de estudio a lo largo del capítulo.

Este módulo agrupa los siguientes RF:

1. Añadir Anuario Estadístico.
2. Modificar Anuario Estadístico.
3. Filtrar Anuario Estadístico.
4. Listar Anuarios Estadísticos.
5. Eliminar Anuario Estadístico.
6. Importar Anuario Estadístico.
7. Exportar Anuario Estadístico.
8. Descargar Anuario Estadístico.
9. Mostrar vista previa del Anuario Estadístico.
10. Revisar Anuario Estadístico.
11. Ver capítulos del Anuario Estadístico.
12. Ver separatas del Anuario Estadístico.
13. Añadir capítulo al Anuario Estadístico.
14. Modificar capítulo del Anuario Estadístico.

15. Listar capítulos del Anuario Estadístico.
16. Eliminar capítulo del Anuario Estadístico.
17. Importar capítulo al Anuario Estadístico.
18. Exportar capítulo del Anuario Estadístico.
19. Descargar capítulo del Anuario Estadístico.
20. Añadir tabla al capítulo.
21. Modificar palabras claves de la tabla del capítulo.
22. Listar tablas del capítulo.
23. Eliminar tabla del capítulo.
24. Descargar tabla del capítulo.
25. Visualizar tabla del capítulo.
26. Añadir separata al Anuario Estadístico.
27. Modificar separata del Anuario Estadístico.
28. Listar separatas del Anuario Estadístico.
29. Eliminar separata del Anuario Estadístico.
30. Descargar separata del Anuario Estadístico.

Seguidamente, en la Tabla 3, se muestra la especificación del requisito Añadir Anuario Estadístico y en la Tabla 4, la especificación del requisito Modificar capítulo del Anuario Estadístico.

Tabla 3: Especificación del requisito funcional “Añadir Anuario Estadístico”.

#	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF1	Añadir Anuario Estadístico	Permite añadir un Anuario Estadístico al sistema.	Media	Alta
Prototipo				
Campos		Tipos de datos	Reglas o restricciones	

			Puede ser: <ul style="list-style-type: none"> • En construcción • Listo para observación • Listo para corrección • Listo para publicar • Publicado
	Palabras claves	Letra	No procede
	Introducción	Alfanumérico	No procede
	Notas metodológicas	Alfanumérico	No procede
Observaciones			

2.3.2 Requisitos no funcionales.

Los requisitos no funcionales (RNF) se definen como todas las características que debe presentar el sistema para que sea seguro, rápido y posea una interfaz legible, simple de usar, atractiva e interactiva. Para el desarrollo de la presente investigación se definieron un total de 43 RNF, los cuales se clasifican en RNF de usabilidad, confiabilidad, eficiencia, soporte, restricciones del diseño, interfaz, de licencia, estándares aplicables y seguridad. Estos se muestran más detallados en el Anexo 2.

2.4. Historias de usuario.

Las historias de usuario constituyen una forma rápida de administrar los requisitos del cliente y definen lo que se debe construir en el proyecto de software. En el desarrollo del SIGAE se confeccionaron un total de 12 historias de usuarios. A continuación, en la Tabla 5, se muestra la historia de usuario Gestionar Anuario Estadístico y en la Tabla 6, la historia de usuario Gestionar tablas.

Tabla 5: Historia de usuario “Gestionar Anuario Estadístico”.

Historia de usuario	
Número: HU#1	Nombre Historia de Usuario: Gestionar Anuario Estadístico
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Camilo Fonseca Camejo	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1,43 semanas (10 días)
Riesgo en desarrollo: Alto	Puntos reales: 1,43 semanas (10 días)
Descripción: Permite añadir, modificar, filtrar, listar, eliminar, importar, exportar, descargar, visualizar (Vista previa) y revisar los Anuarios Estadísticos de las distintas dependencias de la ONEI, así como ver los capítulos y las separatas asociadas al mismo.	

Observaciones: Para acceder a cada una de las funcionalidades para gestionar el Anuario Estadístico deseado, se debe mostrar inicialmente un listado con los Anuarios existentes en el sistema, donde las opciones capítulos, exportar, modificar, eliminar y revisar/vista previa deben estar presentes en cada uno de los elementos de la lista. La funcionalidad descargar solo se muestra al usuario de internet (usuario anónimo).

Prototipo de Interfaz: Ver en el documento Especificación de Requisitos de Software (ERS) del Proyecto Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI, los siguientes requisitos funcionales:

- **RF1** Añadir Anuario Estadístico (p.8).
- **RF2** Modificar Anuario Estadístico (p.9).
- **RF3** Filtrar Anuario Estadístico (p.11).
- **RF4** Listar Anuarios Estadísticos (p.11).
- **RF5** Eliminar Anuario Estadístico (p.12).
- **RF6** Importar Anuario Estadístico (p.12).
- **RF7** Exportar Anuario Estadístico (p.13).
- **RF8** Descargar Anuario Estadístico (p.13).
- **RF9** Mostrar vista previa del Anuario Estadístico (p.14).
- **RF10** Revisar Anuario Estadístico (p.15).
- **RF11** Ver capítulos del Anuario Estadístico (p. 16).
- **RF12** Ver separatas del Anuario Estadístico (p. 17).

Tabla 6: Historia de usuario “Gestionar tablas”.

Historia de usuario	
Número: HU#3	Nombre Historia de Usuario: Gestionar tablas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Camilo Fonseca Camejo	Iteración asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1,43 semanas (10 días)
Riesgo en Desarrollo: Alto	Puntos Reales: 1,43 semanas (10 días)
Descripción: Permite añadir, modificar, listar, eliminar, descargar y visualizar las distintas tablas asociadas a un capítulo determinado del Anuario Estadístico.	
Observaciones: Para acceder a cada una de las funcionalidades para gestionar la tabla deseada, se debe mostrar inicialmente un listado con las tablas existentes en el sistema, donde las opciones modificar, eliminar y visualizar deben estar presentes en cada uno de los elementos de la lista. La funcionalidad descargar solo se muestra al usuario de internet (usuario anónimo).	
Prototipo de Interfaz: Ver en el documento Especificación de Requisitos de Software (ERS) del Proyecto Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI, los siguientes requisitos funcionales:	
<ul style="list-style-type: none"> • RF20 Añadir tabla al capítulo (p.23). 	

- **RF21** Modificar palabras claves de la tabla del capítulo (p.24).
- **RF22** Listar tablas del capítulo (p.25).
- **RF23** Eliminar tabla del capítulo (p.25).
- **RF24** Descargar tabla del capítulo (p.26).
- **RF25** Visualizar tabla del capítulo (p.27).

2.5. Arquitectura.

La arquitectura propuesta para el desarrollo del SIGAE se basa en la arquitectura del marco de trabajo Symfony 2, seleccionado en el capítulo anterior. Esta se sustenta en los principios fundamentales de la arquitectura Modelo-Vista-Controlador, la cual plantea la separación de los datos y la lógica de negocio de la interfaz de usuario como tres componentes distintos.

En la Figura 6 se evidencia la utilización del patrón mencionado anteriormente a través del esquema de funcionamiento del requisito Modificar provincia.

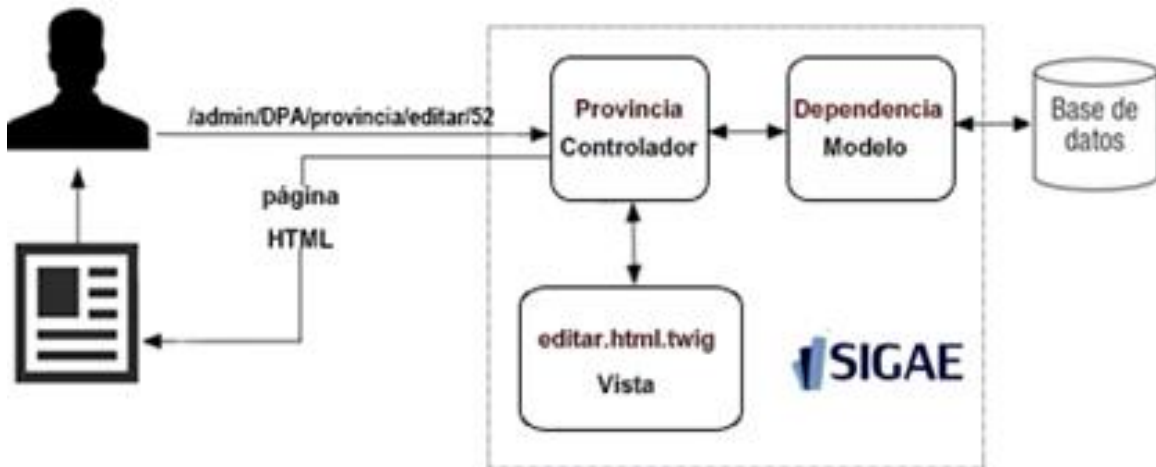


Figura 6: Esquema de funcionamiento del requisito Modificar provincia.

Fuente: Elaboración propia.

El sistema de enrutamiento de Symfony 2 determina que el controlador que debe ejecutar el sistema es “Provincia”, entregándole junto con la petición una variable que contiene el identificador de la provincia que el usuario desea modificar (52). Posteriormente el controlador le solicita al modelo los datos registrados en la base de datos de la provincia y se los entrega a la vista para que cree una página a partir del motor de plantillas Twig. Por último, el controlador retorna dicha página, mostrándole al usuario la petición deseada. La

utilización de esta arquitectura trae como resultado un código altamente extensible, escalable y de fácil mantenimiento.

2.6. Modelo de diseño.

El modelo de diseño es uno de los artefactos generados por la metodología SXP. En este se plantea un esbozo inicial del diseño del sistema, sin entrar en especificaciones ni detalles, solo lo que el diseñador necesita para hacer un primer entregable del sistema (*Gladys Peñalver Romero, y otros, 2008*). Este artefacto contiene el diagrama de clases, la descripción de las mismas y el diagrama de paquetes.

En la Figura 7 se muestra el diagrama de paquetes del módulo Anuario Estadístico.

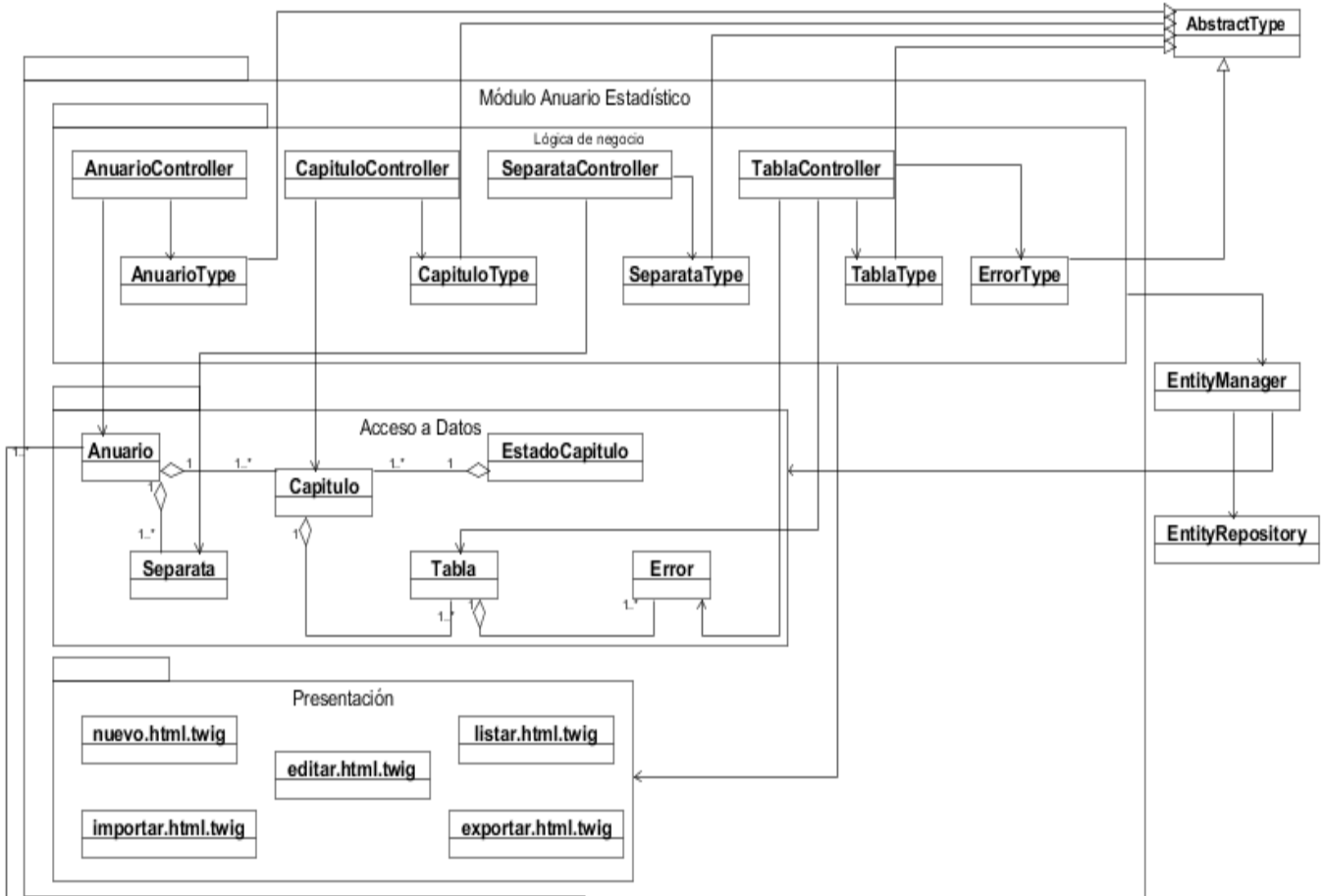


Figura 7: Diagrama de paquetes del módulo Anuario Estadístico.

2.7. Patrones de diseño.

Los patrones de diseño describen un problema particular en un contexto específico, para el cual presentan un esquema genérico probado para su solución, además influyen en la creación de un diseño para el sistema, dotado de flexibilidad y reusabilidad (*Buschmann,*

1996). A continuación se describen los que fueron utilizados en el desarrollo del SIGAE para la asignación de responsabilidades:

2.7.1 Patrones Generales de Software para Asignar Responsabilidades.

En el diseño orientado a objetos, es muy importante asignar las responsabilidades de forma correcta. Para ello se utilizan los patrones GRASP, los cuales promueven buenos principios para la realización de dicha tarea. Para el diseño del SIGAE fueron empleados los siguientes:

Experto: se emplea en las clases del sistema debido a que cada una de las mismas es experta, pues contienen la información necesaria para cumplir con su responsabilidad. El uso de este patrón trae consigo que se conserve el encapsulamiento debido a que las clases se valen de su propia información para cumplir con las tareas asignadas, soportando además un bajo acoplamiento. Este se evidencia en la clase *EstadoCapítulo* como se muestra en la Figura 8.

EstadoCapitulo
-id : int -denominacion : string
+getId() : int +setId(id : int) : void +getDenominacion() : string +setDenominacion(denominacion : string) : void +__toString() : string

Figura 8: Clase experta en la información con respecto al tipo de estado que se le puede asignar a un capítulo en el módulo Anuario Estadístico.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Se pone de manifiesto en las clases controladoras del sistema, las cuales tienen dicha responsabilidad a su cuenta, además brinda soporte a un bajo acoplamiento y mejora la reutilización. Este patrón se observa en la Figura 9 con la clase *AnuarioController*.

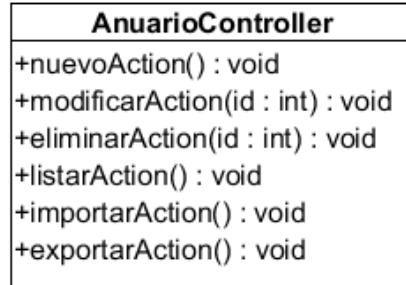


Figura 9: Clase controladora perteneciente al módulo Anuario Estadístico, encargada de la creación de objetos de tipo Anuario Estadístico.

Bajo acoplamiento: el diseño del sistema consta de este patrón pues cada clase depende lo menos posible de otra, de manera tal que una de ellas solo recurre a otra en caso de que exista referencia dentro de sus atributos, lo cual permite que el sistema sea mucho más robusto y de fácil mantenimiento. En la Figura 10 se evidencia el uso de este patrón en la relación existente entre las clases *AnuarioController*, *EntityManager* y *EntityRepository*.

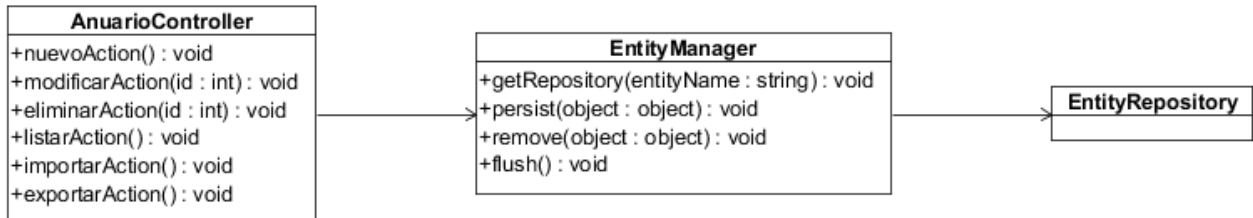


Figura 10: Existencia del patrón Bajo acoplamiento en el módulo Anuario Estadístico.

Alta cohesión: el sistema está diseñado de forma tal que las clases tienen responsabilidades estrechamente relacionadas y no realizan un trabajo excesivo, lo cual permite simplificar el mantenimiento, generar el bajo acoplamiento y aumentar la capacidad de reutilización de las mismas. La Figura 11 muestra el uso de este patrón en la relación existente entre las clases *EstadoCapitulo* y *Capitulo*.

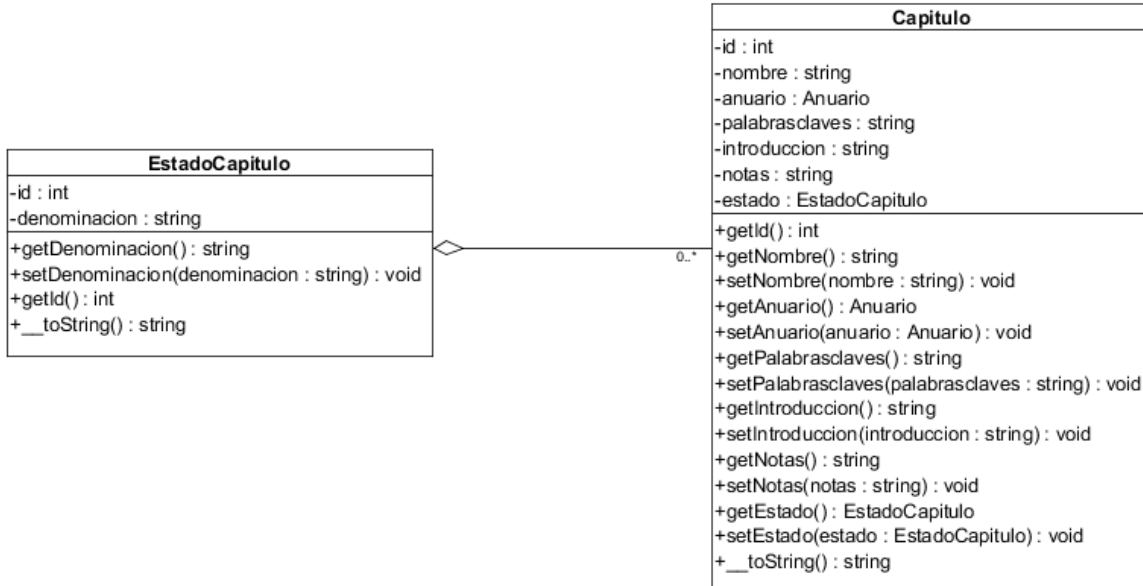


Figura 11: Presencia del patrón Alta cohesión en el módulo Anuario Estadístico.

2.8. Implementación.

La implementación, como fase del proceso de desarrollo de software, se nutre de los resultados obtenidos en el diseño. En correspondencia con los mismos se procede a la implementación del SIGAE con el objetivo de desarrollar un sistema acorde a las necesidades del cliente.

2.8.1. Plan de *releases*.

Una iteración permite separar la complejidad del proyecto gracias al desarrollo por parte de cada bloque. Para ello, es necesario realizar una reunión de planificación de iteración con el objetivo de determinar la funcionalidad del producto en la que trabajará el equipo y el tiempo que demorará en dar cumplimiento a la misma.

A continuación, en la Tabla 7, se muestra el Plan de *releases* del SIGAE.

Tabla 7: Plan de *releases* del SIGAE.

Release	Descripción de la iteración	Orden de la Historia de Usuario a implementar	Duración total
1	Implementación del módulo Anuario Estadístico.	HU#1 Gestionar Anuario Estadístico. HU#2 Gestionar capítulo. HU#3 Gestionar tablas.	40 días

		HU#4 Gestionar separata.	
2	Implementación del módulo División político administrativa.	HU#5 Gestionar país.	12 días
		HU#6 Gestionar provincia.	
		HU#7 Gestionar municipio.	
3	Implementación del módulo Administración.	HU#8 Gestionar usuario.	20 días
		HU#9 Gestionar notificación.	
4	Implementación del módulo Búsqueda.	HU#10 Búsqueda.	15 días
5	Implementación del módulo Traductor de ficheros.	HU#11 Traductor de ficheros.	25 días
6	Implementación del módulo Reportes estadísticos.	HU#12 Reportes estadísticos.	15 días

2.8.2. Estándar de codificación empleado.

Para lograr estandarización en el código del sistema se emplea, de forma general, el estándar de codificación *CamelCase* en su variante *lowerCamelCase* y *UpperCamelCase* para el caso de las clases, lo que hará más sencillo el análisis del código y el mantenimiento del sistema.

Convenciones de nomenclatura

General:

- En caso de existir palabras con la letra ñ, se empleará el término en inglés.
- Para las variables, clases o métodos que contengan tildes serán utilizadas las propias palabras pero sin la acentuación.
- Serán utilizados nombres relacionados con el contexto en que se desarrolla.

Identación:

- El código del sistema será identado por *tabs* en lugar de hacer uso de espacios en blanco.

Clases:

- El nombre de las clases siempre comenzará con mayúscula. En caso de ser una palabra compuesta, cada una de las palabras comenzarán también de la misma forma.

Nombres de variables:

- Los nombres de las variables no serán ambiguos.

2.9. Conclusiones parciales.

- La confección de diversos artefactos pertenecientes a la metodología utilizada como: las Historias de Usuario, el Modelo de diseño y el Plan de releases, permitieron agilizar y organizar el proceso de desarrollo del sistema.
- El uso de patrones y el empleo del marco de trabajo Symfony 2, permitió obtener un sistema mantenible y escalable.
- El empleo del estándar de codificación CamelCase y el uso de un conjunto de convenciones de nomenclatura, contribuyen al entendimiento y estandarización del código del sistema, facilitando su comprensión en futuros mantenimientos.

CAPÍTULO 3: VALIDACIÓN

3.1. Introducción.

En el presente capítulo se muestran los resultados obtenidos luego de aplicar las técnicas de validación tanto a los requisitos como al diseño del sistema y las métricas empleadas para medir la calidad de los mismos. Además, se documentan los resultados alcanzados una vez realizadas las pruebas de Caja Blanca y Caja Negra. También se exponen las mejoras obtenidas con el SIGAE, teniendo en cuenta las variables definidas en el problema a resolver.

3.2. Técnicas de validación de requisitos.

La aplicación de técnicas de validación tiene como objetivo demostrar que los requisitos previamente definidos cumplen con las expectativas del cliente. En este caso, para el desarrollo del SIGAE, se emplearon las siguientes:

- **Revisiones formales de los requisitos:** se realizaron revisiones formales a cada uno de los requisitos por parte del cliente y del equipo de desarrollo, obteniéndose un total de 11 no conformidades de tipo técnicas y de ortografía, las que fueron corregidas satisfactoriamente en tiempo. Con el cliente se llevaron a cabo 3 revisiones. En la primera, 6 de los requisitos funcionales fueron modificados con el objetivo de mejorar la interpretación de los mismos. En el segundo encuentro fueron modificados otros 3 con el mismo propósito de la primera revisión. En un tercer encuentro, el cliente quedó satisfecho con el trabajo efectuado por el equipo de desarrollo y se aprobaron finalmente la totalidad de los requisitos, generándose de este encuentro un Acta de Aceptación por parte del cliente.
- **Construcción de prototipos:** se confeccionaron prototipos funcionales mediante los cuales se mostró al cliente un modelo ejecutable del sistema, que permitió tener una visión preliminar del mismo y a través de la interacción con estos se comprobó su grado de satisfacción. Con el uso de esta técnica cada uno de los prototipos definidos fueron aprobados por el cliente.
- **Generación de casos de prueba:** como parte del proceso de validación de los requisitos funcionales del SIGAE, se diseñaron un conjunto de casos de pruebas a partir de cada historia de usuario definida. En el Anexo 3 se muestran los casos de pruebas de las funcionalidades Añadir provincia y Modificar provincia, asociados a la historia de usuario Gestionar provincia.

3.3. Métrica aplicada a los requisitos.

Con el propósito de medir la calidad de la especificación de los requisitos de software se aplicó la métrica Calidad de la Especificación (CE).

Para obtener el nivel de entendimiento y precisión de los requisitos se debe calcular, en primer lugar, el total de requisitos de software como se muestra a continuación:

Nr: total de requisitos de software.

Nf: cantidad de requisitos funcionales.

Nnf: cantidad de requisitos no funcionales.

$$Nr = Nf + Nnf$$

Sustituyendo los valores en la ecuación, capturados para el desarrollo del SIGAE, se obtiene:

$$Nr = 62 + 42$$

$$Nr = 104$$

Finalmente, para calcular la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

$$ER = Nui/Nr$$

Nui: número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas.

Teniendo en cuenta que de los requisitos obtenidos para el desarrollo del SIGAE ninguno produjo contradicción en las interpretaciones y que mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad, se procede a sustituir las variables en la ecuación:

$$ER = \frac{104}{104}$$

$$ER = 1$$

Como resultado final se obtiene 1, indicando que no existe ambigüedad en los requisitos capturados ya que el 100% de ellos es entendible.

3.4. Validación del diseño del sistema.

Para comprobar la calidad del diseño del SIGAE se emplearon las métricas Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC).

A través de la métrica TOC se mide la responsabilidad, la complejidad de implementación y la reutilización de las clases. En esta métrica, la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que a menor responsabilidad y complejidad de implementación de una clase, mayor será su nivel de reutilización.

En la Figura 12 se muestran los resultados obtenidos de la aplicación de la métrica TOC:

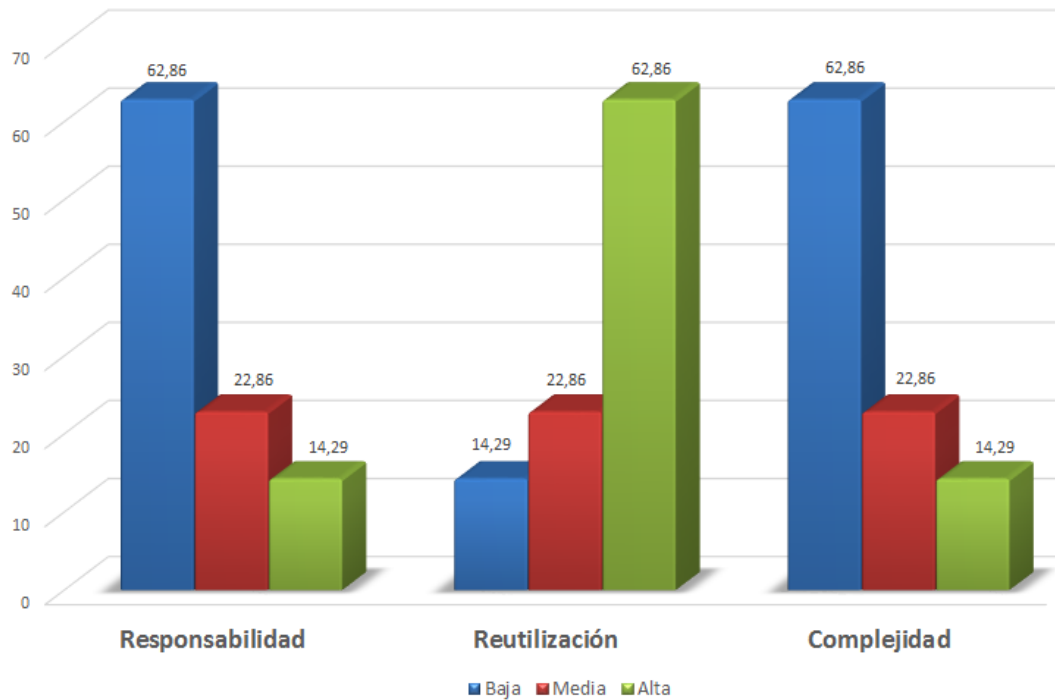


Figura 12: Representación en por ciento (%) del nivel de responsabilidad, reutilización y complejidad de implementación de las clases.

Una vez aplicada la métrica, se observa que las clases del SIGAE presentan bajo nivel de responsabilidad y complejidad, favoreciendo en gran medida la reutilización de las mismas.

Mediante el empleo de la métrica RC se evalúa el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar cada clase, teniendo en cuenta las relaciones existentes entre ellas.

En la Figura 13 se muestran los resultados obtenidos de la aplicación de la métrica RC:

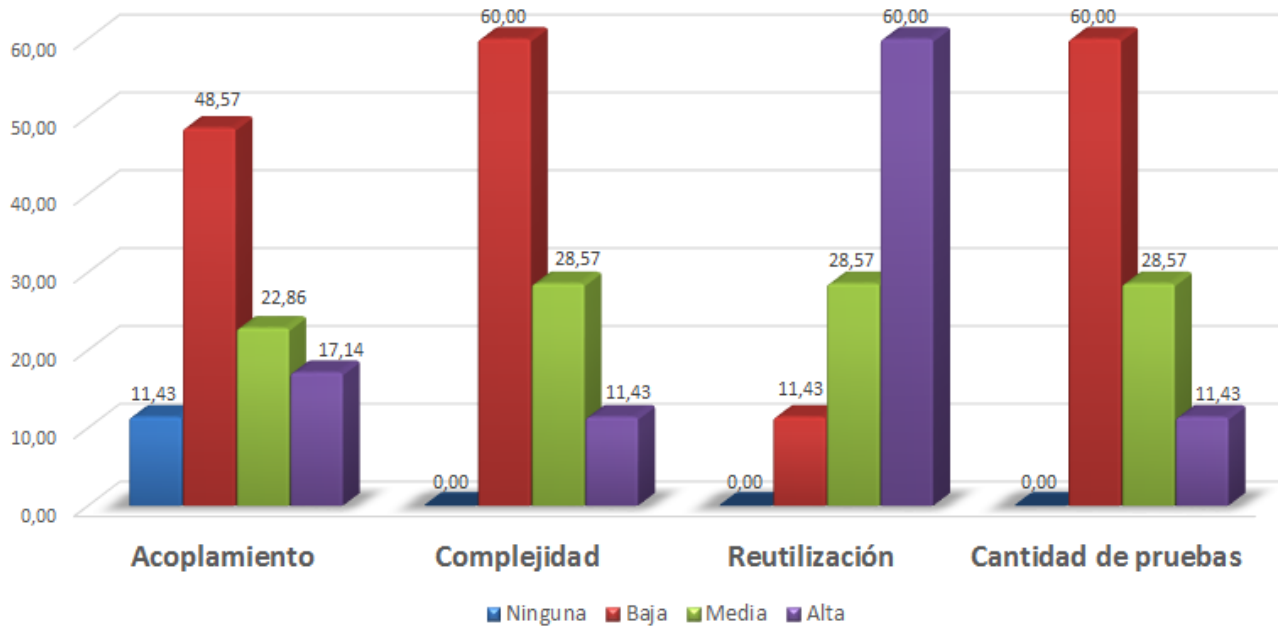


Figura 13: Representación en por ciento (%) del nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases.

Luego de aplicada la métrica, se obtiene como resultado que las clases del diseño del SIGAE presentan bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas son bajas y en consecuencia el grado de reutilización es alto.

De manera general, con los resultados obtenidos después de aplicar las métricas TOC y RC y teniendo en cuenta el umbral definido para la validación del diseño (Bien [0.1; 0.3], Regular [0.4; 0.7] y Mal [0.8; 1.0]) se demuestra que las clases del SIGAE presentan bajo acoplamiento y poseen un alto grado de reutilización.

3.5. Pruebas.

Las pruebas de software son el proceso de ejercitar el software con la intención de encontrar (y por último corregir) los errores (*Roger S. Pressman, 2002*). Por su parte, la metodología de desarrollo SXP, asegura el éxito del mismo al realizar dicho proceso de pruebas de manera continua, facilitando la obtención de un sistema de mayor calidad debido a que los errores son detectados en un corto período de tiempo y pueden ser corregidos de manera sencilla.

Con el objetivo de verificar la calidad del SIGAE, se realizaron pruebas de unidad, pruebas formales que permiten determinar si un módulo del programa está listo y

correctamente terminado. De ellas se aplicaron las técnicas de Caja Negra y Caja Blanca. En cuanto a la primera, permitió comprobar el funcionamiento de los componentes del sistema, analizando las entradas y salidas de los datos y verificando que el resultado es el esperado; mientras que la segunda, solo se centró en la estructura interna del programa.

3.8.1. Pruebas de Caja Negra.

Para comprobar que las funcionalidades del sistema se realizan correctamente y responden a las necesidades del cliente, se realizaron un total de 3 iteraciones de pruebas funcionales (Caja Negra), teniendo en cuenta los siguientes parámetros: ortografía, redacción, validación, funcionalidad y recomendación, como se muestra en la Figura 14.

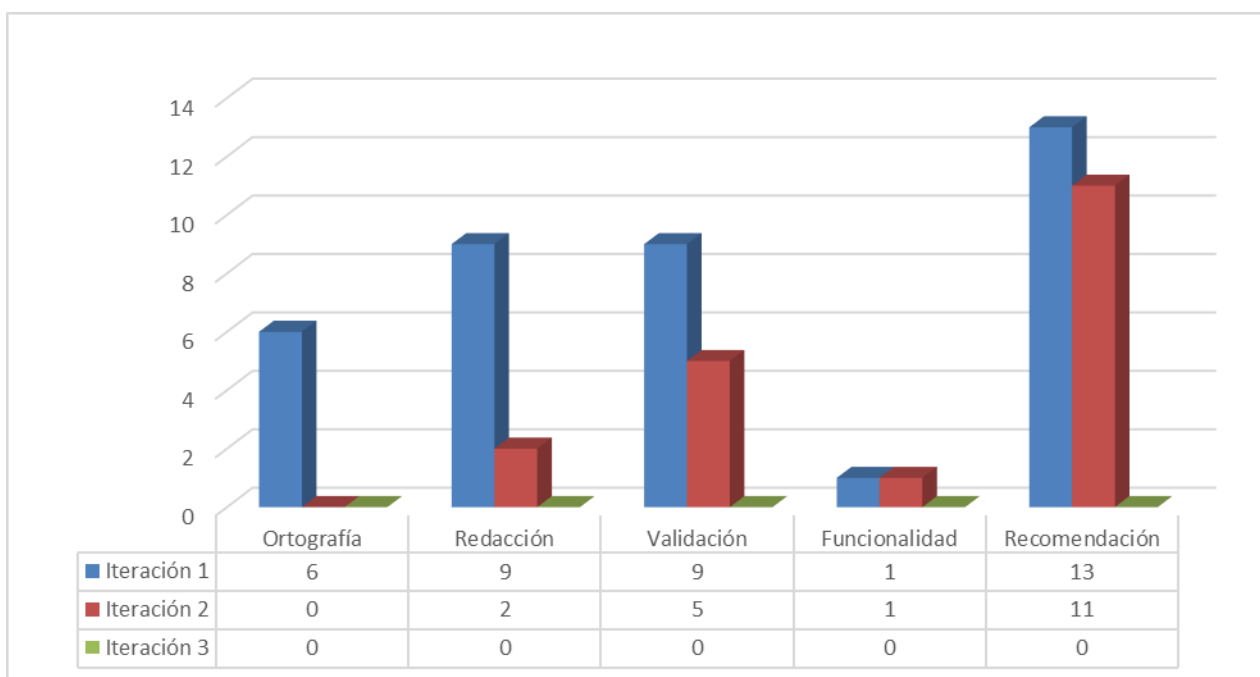


Figura 14: No conformidades detectadas en el SIGAE a través de las pruebas de Caja Negra.

En una primera iteración se detectaron un total de 38 no conformidades entre los seis módulos revisados, predominando entre ellas las de tipo recomendación. En la segunda iteración se obtuvieron 19 no conformidades donde aún existían errores de validación, realizándose además varias recomendaciones. Finalmente, en una tercera iteración se obtuvieron resultados satisfactorios debido a que no se encontraron no conformidades.

3.8.2. Pruebas de Caja Blanca.

Para comprobar que las funciones internas del SIGAE se ejecutan correctamente se le realizaron pruebas al código de las principales funcionalidades de cada uno de los módulos del sistema. Para ello se empleó la librería PHPUnit, la cual el marco de trabajo Symfony 2 utiliza en la realización de pruebas unitarias, facilitando la creación de pequeños scripts que ayudan a testear las aplicaciones y analizar los resultados.

De forma general se le realizaron las pruebas de Caja Blanca a 23 funcionalidades en dos iteraciones. Los resultados obtenidos se muestran en la Figura 15:

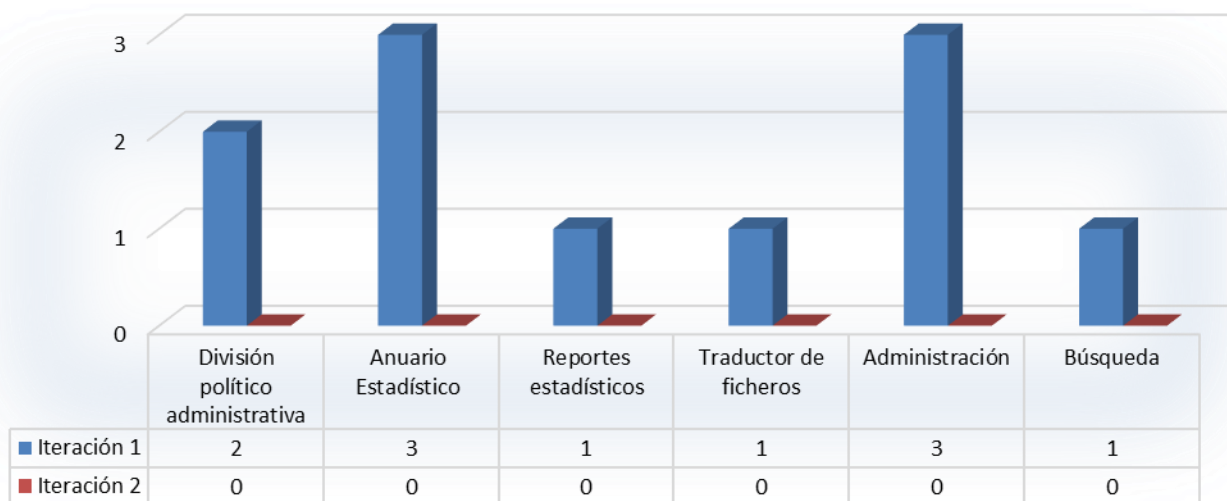


Figura 15: Resultados obtenidos en la aplicación de las pruebas de Caja Blanca.

En una primera iteración se detectaron once no conformidades, las cuales fueron corregidas en su totalidad. Por otro lado, la segunda iteración arrojó resultados satisfactorios con cero no conformidades.

Con los resultados obtenidos luego de realizar las pruebas de Caja Negra y Caja Blanca se demuestra que el SIGAE es estable y que su funcionamiento interno y externo es el correcto.

3.6. Mejoras del SIGAE.

El sistema desarrollado contribuye a la confidencialidad e integridad de los datos pues el acceso a este se realiza mediante un mecanismo de autenticación, controlándose la asignación de responsabilidades a los usuarios a partir de su rol, lo que permite que solo ejecuten las funcionalidades que les corresponden. Por otra parte, la utilización del ORM

Doctrine 2 le proporciona mecanismos de defensa contra ataques de inyección de código SQL.

En cuanto a la usabilidad del servicio, el SIGAE brinda un conjunto de facilidades al usuario tales como: un sistema de búsqueda que les permite encontrar de forma rápida contenidos específicos dentro los Anuarios Estadísticos, la posibilidad de visualizar el contenido de los mismos en el propio sistema, la descarga de un capítulo y/o el Anuario completo, así como mostrar reportes estadísticos a partir del contenido de los Anuarios y teniendo en cuenta los indicadores previamente seleccionados por el usuario. Dentro de sus valores agregados también se encuentran estrategias de navegabilidad como la existencia de migas de pan, la estandarización en la iconografía empleada para el acceso a diversas funcionalidades del sistema y la accesibilidad a través del logo a la página de inicio del mismo.

A partir de entrevistas con el cliente se evaluó su satisfacción subjetiva con respecto al uso del SIGAE, arrojando como resultado que el sistema satisface las necesidades de gestión de Anuarios Estadísticos en la entidad. Además se analizó la complejidad para la confección de dicha publicación y del mecanismo de notificación de errores resultantes del proceso, teniendo en cuenta la forma en que se realizan actualmente en la ONEI y el modo en que se gestionan en el SIGAE, concluyendo que este último posee un nivel de complejidad inferior en ambos casos, generándose de estos encuentros el Acta de Aceptación del Producto que refleja la conformidad del cliente (ver Figura 19 del Anexo 6).

A partir de lo planteado anteriormente se puede afirmar que con el desarrollo del SIGAE se eleva la integridad de los datos, la usabilidad del servicio y se agiliza el proceso de gestión de los Anuarios Estadísticos en la ONEI (ver Tabla 12 del Anexo 5).

3.7. Conclusiones parciales.

- La aplicación de la métrica CE, así como las técnicas de validación de los requisitos definidos para el SIGAE, demostraron la calidad de la especificación realizada, obteniendo que el 100% de los mismos fueron entendibles por el personal de la ONEI.
- Con la validación del diseño del sistema mediante las métricas TOC y RC, se obtuvo como resultado que las clases presentan bajo acoplamiento y poseen un alto grado de reutilización.

- La realización de pruebas unitarias al sistema permitió comprobar el correcto funcionamiento de este y la ausencia de no conformidades al concluir la última iteración de las mismas.

CONCLUSIONES GENERALES

- El estudio y aplicación de patrones de diseño, el uso de la arquitectura seleccionada y el empleo de estándares de codificación, contribuyeron al logro de una solución mantenible y escalable.
- Las pruebas funcionales y de aceptación con el cliente realizadas al sistema, arrojaron resultados satisfactorios que demuestran la obtención de un producto acorde a las necesidades de la ONEI.
- El sistema obtenido contribuyó a elevar la integridad de los datos, la usabilidad del servicio y la agilización en el desarrollo del proceso de gestión de Anuarios Estadísticos en las diferentes dependencias de la ONEI.

RECOMENDACIONES

- Añadir nuevas funcionalidades al sistema que permitan la confección de las tablas estadísticas sin hacer uso de Microsoft Excel para ello.
- Añadir nuevas funcionalidades al módulo Reportes estadísticos, permitiendo, de esta forma, graficar la comparación establecida entre diferentes Anuarios.

BIBLIOGRAFÍA REFERENCIADA

BAKKEN, Alexander Aulbach Stig Sæther, EGON SCHMID, Jim Winstead, LARS TORBEN WILSON, Rasmus Lerdorf, ZEEV SURASKI, Andrei Zmievski y JOUNI AHTO, 2001, *Manual de PHP*.

BUSCHMANN, Et Al, 1996, *A System of Patterns*.

CARLES MATEU, 2004, *Desarrollo de aplicaciones web*.

CARLOS ANDRÉS GUERRERO ALARCÓN y HERNANDO RECAMAN CHAUX, 2009, Estudio comparativo de marcos de trabajo para el desarrollo de aplicaciones web de código abierto. 4 Octubre 2009.

DANIEL BENCHIMOL, 2011, *Hacking desde cero: Manuales Users (Spanish Edition)*. Argentina: Creative Andina Corp.

DICE HOLDINGS, INC, 2014, JExcel. 2014. Disponible en: <http://jexcelapi.sourceforge.net/>

DOCTRINE TEAM, 2014, Doctrine 2 ORM v2.1 documentation. 2014. Disponible en: <http://sf2-es.net16.net/doctrine/orm-documentation/reference/introduction.html>

ENRIQUE HERNÁNDEZ ORALLO, 2012, *El Lenguaje Unificado de Modelado (UML)*. 2012.

FABIEN POTENCIER y FRANÇOIS ZANINOTTO, 2008, *Symfony, la guía definitiva*.

FABIEN POTENCIER, 2011, Twig The flexible, fast, and secure template language for PHP. In: *Zend/PHP Conference*. San Jose, California. 19 Noviembre 2011.

GLADYS MARSÍ PEÑALVER ROMERO, SERGIO JESÚS GARCÍA DE LA PUENTE y ABEL MENESES ABAD, 2010, SXP, metodología de desarrollo de software | Peñalver Romero | Serie Científica. 2010. Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/view/429/482>

HERNAN ALBERTO SILVA RODRÍGUEZ, 2006, Sistema Manejador de Base de Datos. 24 Febrero 2006. Disponible en: <http://www.mailxmail.com/curso-procesamiento-datos-oracle/sistema-manejador-base-datos>

HIBERNATE TEAM, 2014, What is Object/Relational Mapping? 2014. Disponible en: <http://hibernate.org/orm/what-is-an-orm/>

HISPANETWORK PUBLICIDAD Y SERVICIOS, 2009, Definición de Anuario estadístico. 26 Abril 2009. Disponible en: <http://cultura.glosario.net/terminos-bibliotecarios/anuario-estad%EDstico-12231.html>

IAN SOMMERVILLE, 2005, *Ingeniería del Software*. Séptima Edición. Pearson Educación S.A. ISBN 84-7829-074-5.

INFORED, 2014, XP - Extreme Programming Ingeniería de Software. 2014. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programming.html

ING. JAVIER A. VOOS, ING. EDUARDO GONZALEZ y ING. FERNANDO CAGNOLO, 2003, *Portal de Aplicaciones Médicas*. Argentina: Universidad Tecnológica Nacional.

ITEXT GROUP NV, 2014, iText. 2014. Disponible en: <http://itextpdf.com/>

JAVIER EGUILUZ PÉREZ, 2008, *Introducción a CSS*.

JAVIER EGUILUZ, 2011, *Desarrollo web ágil con Symfony 2*. Primera edición.

JAVIER EGUILUZ, 2014, *Introducción a JavaScript*. 2014. Disponible en: http://librosweb.es/javascript/capitulo_1.html

JOSE LUIS LIRA TURRIZA, 2007, *Algoritmos y lenguajes de programación*. Campeche : Instituto Tecnológico Superior De Calkiní.

LISANDRA ISABEL LEYVA SAMADA, 2009, *Flujo de Investigación para la Metodología Ágil SXP*. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2435_09. TD_2435_09

MICROSOFT CORPORATION, 2014a, PHPEXcel. 2014. Disponible en: <https://phpexcel.codeplex.com/>

MICROSOFT CORPORATION, 2014b, ClosedXML. 2014. Disponible en: <http://closedxml.codeplex.com/site/search/openings>

NETBEANS TEAM, 2012, *FaqLicense - NetBeans*. 2012. Disponible en: <http://wiki.netbeans.org/FaqLicense>

NICOLA ASUNI, 2013, TCPDF. 2013. Disponible en: <http://www.tcpdf.org/>

ONEI, 2010, *¿Qué es la ONE?* 2010. Disponible en: <http://www.onei.cu/publicaciones/08informacion/Que%20es%20la%20ONE.pdf>

ORACLE CORPORATION, 2013, *Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto*. 2013. Disponible en: https://netbeans.org/index_es.html

PROGRAMACIÓN DESARROLLO TEAM, 2014, *Qué es un entorno de desarrollo integrado, IDE*. 2014. Disponible en: <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>

PROYECTOSAGILES.ORG TEAM, 2014, *Qué es SCRUM | proyectos Ágiles*. 2014. Disponible en: <http://www.proyectosagiles.org/que-es-scrum>

ROGER S. PRESSMAN, 2001, *Ingeniería del Software. Un enfoque práctico. Quinta Edición*. España: McGraw Hill.

ROGER S. PRESSMAN, 2002, *Temas avanzados en Ingeniería del Software*. Disponible

en:

http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/Ediciones_del_Pressman/Pressman_5ta_edicion/Pressman_Parte_V_Temas_avanzados_en_ingenieria_del_software.pdf

RUTH VEGA CUTIÑO, 2012, Conceptos básicos de Seguridad Informática.2012.

Disponible en:

http://eva.uci.cu/file.php/92/Tema_1_Introduccion/Materiales/Apoyo/Conceptos_basicos_v2.0.pdf

SERGIO LUJÁN MORA, 2001, *Programación en Internet: Clientes Web*. 1ra Edición. Editorial Club Universitario.

SOLID IT, 2014, DB-Engines. 2014. Disponible en: <http://db-engines.com/en/>

UNIDAD DE MODERNIZACIÓN Y GOBIERNO DIGITAL, 2014, ¿Qué es la Usabilidad? 2014. Disponible en: <http://www.guiadigital.gob.cl/articulo/que-es-la-usabilidad>

UNIVERSIDAD DE GRANADA, 2014, Especificación de requerimientos. Diseño de bases de datos. 2014. Disponible en: <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>

VÍCTOR M. RIVAS SANTOS, 2014, Curso de JavaScript. 2014. Disponible en: http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html

VISUAL PARADIGM TEAM, 2014, All-in-One, End-to-End Information Technology System Modeling Tool. 2014. Disponible en: <http://www.visual-paradigm.com/product/vpuml/editions/enterprise.jsp>

WEB DEVELOPERS NOTES TEAM, 2014, Programming languages used on the Internet and the World Wide Web (WWW). 2014. Disponible en: http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3

BIBLIOGRAFÍA CONSULTADA

Área de Tecnologías de la Información y las Comunicaciones Aplicadas de la Universidad de Murcia (ATICA) - ANURI - Anuario Estadístico de Unidades Administrativas, 2014. Disponible en: <http://www.um.es/atica/anuri>

BRUNO SKVORC, 2014, Best PHP Frameworks for 2014 - SitePoint. 2014. Disponible en: <http://www.sitepoint.com/best-php-frameworks-2014>

CEPAL - Anuario Estadístico de América Latina y el Caribe 2013, 2014. Disponible en: http://www.cepal.org/cgi-bin/getProd.asp?xml=/publicaciones/xml/6/51946/P51946.xml&xsl=/publicaciones/ficha.xml&base=/publicaciones/top_publicaciones.xml#

PHP Frameworks Team, 2013, PHP Frameworks. 2013. Disponible en: <http://www.phpframeworks.com>

Povilas Korop, 2013, TOP 6 most popular PHP frameworks of 2013 | WebCoderPro. 20 Noviembre 2013. Disponible en: <http://webcoderpro.com/blog/top-6-most-popular-php-frameworks-of-2013/>

United Nations Economic and Social Commission for Asia and the Pacific, 2014. Disponible en: <http://www.unescap.org/publications/detail.asp?id=1515>