



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

MIGRACIÓN DE LA CAPA DE ACCESO A DATOS

DEL MARCO DE TRABAJO SAUXE

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

Autores:

Inoelkis Velázquez Osorio

Lisandra Cordero Estrada

Tutores:

Ing. René Bauta Camejo

Ing. Leandro Pérez–Borroto Vivero

La Habana, Junio de 2013

“Año 55 de la Revolución”



“No hay enigmas..., si un problema puede plantearse, es que puede resolverse”

Ludwig Wittgenstein



DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los ____ días del mes de Junio del año 2013.

Firma del Autor 1

Firma del Autor 2

Firma del Tutor 1

Firma del Tutor 2



DATOS DE CONTACTO

Tutor: Ing. René Bauta Camejo

Edad: 28 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría docente: Instructor

E-mail: rrbauta@uci.cu

Ingeniero en Ciencias Informáticas, graduado en 2009 en la Universidad de las Ciencias Informáticas. Instructor. Cuatro años de experiencia en el desarrollo de software. Cuatro años de graduado.

Tutor: Ing. Leandro Pérez–Borroto Vivero

Edad: 27 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría docente: Instructor

E-mail: lvivero@uci.cu

Ingeniero en Ciencias Informáticas, graduado en 2009 en la Universidad de las Ciencias Informáticas. Instructor. Siete años de experiencia en el desarrollo de software. Cuatro años de graduado.



Agradecimientos:

De Inoelkís:

A Míma y Papi que depositaron en mí toda la confianza y no escatimaron en esfuerzo y empeño para darme toda la educación, el cariño y lo necesario para llegar hasta aquí.

A mis dos hermanos que me apoyaron en todo momento.

A mi tía Joaquina que siempre estuvo a mi lado dándome apoyo y cariño.

A mis abuelos que están y los que no están por darme dos padres inigualables.

A Odélnis que ha sido mi hermano además de primo toda mi vida.

A ella que la quiero mucho y ocupa una parte importante de mi vida.

A todos mis tíos, primos y familia de manera general que siempre han estado pendiente de mí.

A Rogelio que ha sido mi hermano durante estos 5 años y nunca me defraudó.

A Maricet, Tallys, Anabel y Roíder que fueron mis primeros amigos aquí y me ayudaron durante los 5 años.

A mis tutores, en especial a René por soportarme en todo momento y estar ahí cuando lo necesitaba.

A mi compañera de tesis Lisandra por soportarme y ayudar todo lo posible.

A El Crackem que ha demostrado su apoyo incondicional durante todo el proceso de tesis.

A todos mis compañeros de aula y de proyecto que durante estos 5 años estuvieron ahí.

A todas mis amistades del barrio que depositaron su confianza en mí y nunca dudaron.

A los maestros y profesores excepcionales que he tenido a lo largo de mi vida estudiantil y que han contribuido con mi formación personal y profesional.

A todos los que me apoyaron y confiaron en mí; mis más sinceros agradecimientos.



De Lisandra:

*A mi **Díos**, mi rey, mi salvador, mi consejero, mi pastor, mi restaurador, mi consolador, él es y será siempre el merecedor de todos mis agradecimientos, a él que restaura cada día mi vida, que me ama con amor indescriptible, que destila miel en cada palabra que me dice, que escucha atento a mis palabras y se apresura a socorrerme, a **Jesús** que dio su vida por salvarme y no escatimó nada, al **Espíritu Santo de Díos**, a él sea toda la gloria, la honra, la majestad, el poder.*

*A Yuniel, mi esposo, por estar a mi lado, amarme siempre, perseverar junto a mí estos 5 años; y darme siempre la palabra de **Díos** que necesitaba, por levantarme, por consolarme. **Díos** ha sido en gran manera bueno por permitirme vivir la vida a tu lado.*

*Al pueblo de **Díos** que está y estuvo en la UCI, mis hermanos y mis amigos, gracias a todos por sus oraciones, por sus palabras, por sus consejos, por alentarme y ayudarme a caminar, gracias.*

*A mi mamá, gracias por nunca dudar, gracias por confiar, gracias por tus palabras, gracias por decirme: "Lisandra estudia". Gracias a **Díos** porque cuando lo conocí me entregó un amor inmenso por ella y hoy puedo decir que tengo la mejor madre del mundo, gracias a ella por su preocupación y por tratarme como su niñita.*

A mi papá, por todos sus consejos, por sus regaños, por sentirse orgulloso de mí, por su confianza, por su cariño.

A mis hermanas, mis tres princesitas, a Liannet aunque aparentas ser fría, es dulce y amorosa, a Lirisleidis mi chocolatico, gracias por su sonrisa, cuando sonrío puedo sentir, como el ambiente cambia, a Lía por su alegría.

A mi abuela gracias por tu preocupación, tu atención, a ella por tantas noches sin dormir por mí. Gracias porque de ti nunca me faltó el cariño y el amor.

A mis tías y a mis tíos por su apoyo en estos 5 años en especial a Leidis.



De Inoelkís:

A mis padres que se merecen todo en este mundo y que haré lo que este en mis manos para dárselo.

A mis dos hermanos que son mis hermanos y no pediría otros.

A mi tía Joaquína que se lo merece por su confianza en mí.

A todos los que confían en mí y me brindan su apoyo día a día.

De Lisandra:

Al único y verdadero Dios, Jehová de los Ejércitos porque si hoy me estoy graduando es gracias a él.

A mi esposo, por ser ejemplo para mí y amarme siempre.

A mi mamá porque este es un sueño cumplido para ti.

A mis hermanas que son mis princesas para que nunca duden que con Dios todo es posible.

A Yasmery (Kiri), por su cariño, por todos los estragos que nos hizo, te quiero mucho mi niña, eres muy especial.

A Jose y a Diana, porque son muy especiales, doy gracias a Dios por su amistad y por cada momento que hemos compartido.

A Aímara, a Rosana y a Marilaura por encaminarme y enseñarme a perseverar en la fe.

A Tania, a Yunexis, a Elizabeth María, a Yanelis, a Marian por escucharme y por cada momento que pasamos justas, aprendí mucho con ellas.

A mis tutores, en especial a René Rodrigo por estar siempre ahí apoyándonos y guiándonos.

A mi compañero de tesis, a Inoelkís porque he llegado a quererlo como a uno de mis hermanos, porque creo que Dios tiene un propósito contigo aunque todavía no lo ha entendido.

A mis compañeros de proyecto y a los del aula por todos los momentos que he compartido con ellos, en especial a Miri y a Meli.

A todos los que de una manera u otra ayudaron a la realización de esta tesis.



RESUMEN

La utilización de herramientas que proporcionen facilidades en la creación y desarrollo de software han tomado gran auge; y como ejemplo de esto se tiene la utilización de marcos de trabajos. Estos garantizan que los equipos de desarrollo se centren en la implementación de los escenarios propios de su negocio logrando de esta forma rapidez en el proceso, productividad y profesionalidad en cuanto a la elaboración de aplicaciones.

Sauxe es un marco de trabajo desarrollado en el Centro para la Informatización de la Gestión de Entidades que contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor agilidad en el proceso de desarrollo. Está basado en el marco de trabajo Zend, Doctrine, PHP y ExtJS, empleando como gestor de base de datos a PostgreSQL. Actualmente Sauxe emplea la versión 1.2.2 del Doctrine lo que ha provocado problemas de rendimiento y en el manejo de los datos.

El presente trabajo constituye una guía de migración para lograr la actualización de la capa de acceso a datos de este marco de trabajo a la versión 2 de Doctrine, definiendo con claridad las etapas y actividades a desarrollar en aras de lograr una exitosa migración.

Palabras claves: Doctrine, Marco de Trabajo, Rendimiento, Sauxe.



ÍNDICE

INTRODUCCIÓN 12

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 16

 1.1. Introducción 16

 1.2. Marco de Trabajo de Persistencia de Objetos Relacionales 16

 1.2.1. Hibernate 16

 1.2.2. Propel 17

 1.2.3. Django 18

 1.2.4. SubSonic 19

 1.2.5. Doctrine 1.2.2 20

 1.2.6. Doctrine 2.2.0 21

 1.3. Tecnologías y herramientas para el desarrollo 22

 1.3.1. Subversion 22

 1.3.2. RapidSVN v0.12.0 23

 1.3.3. PostgreSQL 23

 1.3.4. Netbeans 23

 1.3.5. Apache 24

 1.3.6. Sauxe 24

 1.3.7. Marco de Trabajo Zend 24

 1.3.8. PHP 24

 1.4. Validación de la solución 25

 1.4.1. Método Delphi 25

 1.4.2. Proceso de selección de expertos 26

 1.4.3. Desarrollo práctico y explotación de los resultados 26

 1.5. Conclusiones parciales 27

CAPÍTULO 2: DESARROLLO DE LA MIGRACIÓN 28

 2.1. Introducción 28

 2.2. Contenido de la migración 28

 2.3. Etapa Inicial 28

 2.3.1. Estudio de la arquitectura de Sauxe 28



TABLA DE CONTENIDOS

2.3.2.	Estudio de la estructura de carpetas de los componentes de Sauxe	29
2.3.3.	Integración de Doctrine 2 con el MT Zend	30
2.3.4.	Estudio de las librerías seleccionadas	30
2.3.5.	Estudio de las clases seleccionadas	30
2.3.6.	Pruebas de Integración	34
2.4.	Etapa de Implementación	35
2.4.1.	Redefinir la Capa de Acceso a Datos	35
2.4.2.	Estudiar la nueva estructura de carpetas de los componentes de Sauxe	35
2.4.3.	Estudiar equivalencia entre la estructuras de carpetas	36
2.4.4.	Estudiar Formas de mapeo	37
2.4.5.	Consultas en las clases Repository	40
2.4.6.	Funcionalidades en las clases Model	41
2.4.7.	Redefinir la Capa del Negocio	42
2.4.8.	Probar la funcionalidad de los componentes de MT.	42
2.4.9.	Actualizar estado de los elementos de configuración	43
2.5.	Etapa de Pruebas	43
2.5.1.	Preparar el entorno de pruebas	43
2.5.2.	Realizar pruebas de rendimiento	43
2.5.3.	Resolver las no conformidades	44
2.6.	Conclusiones parciales.	44
CAPÍTULO 3: VALIDACIÓN DE LA MIGRACIÓN		45
3.1	Introducción	45
3.2.1.	Elaboración del cuestionario para la validación de la propuesta	46
3.2.2.	Establecimiento de la concordancia entre los expertos	47
3.2.3.	Desarrollo práctico y explotación de los resultados	49
3.3.	Pruebas de Rendimiento	53
3.4.	Conclusiones Parciales	54
CONCLUSIONES GENERALES		55
RECOMENDACIONES		56
REFERENCIAS BIBLIOGRÁFICAS		57



ÍNDICE DE TABLAS

TABLA I. COMPARACIÓN DE RENDIMIENTO EN LAS VERSIONES DE DOCTRINE [14].21
TABLA II. CÁLCULO DEL COEFICIENTE DE KENDALL.48
TABLA III. FRECUENCIA ABSOLUTA.50
TABLA IV. FRECUENCIA ABSOLUTA ACUMULADA.51
TABLA V. FRECUENCIA RELATIVA ACUMULADA.51
TABLA VI. PUNTOS DE CORTE.52
TABLA VII. RANGOS DE LAS CATEGORÍAS OBTENIDOS A PARTIR DE LOS PUNTOS DE CORTE.....53

ÍNDICE DE FIGURAS

FIGURA 1. ESTRUCTURA DE CARPETAS DEL COMPONENTE SEGURIDAD.29
FIGURA 2. UBICACIÓN DE LAS NUEVAS LIBRERÍAS DE DOCTRINE.30
FIGURA 3. UBICACIÓN DE LA CLASE DOCTRINE.PHP.31
FIGURA 4. UBICACIÓN DEL FICHERO DE CONFIGURACIÓN.....32
FIGURA 5. UBICACIÓN DE LA CLASE TRANSACTIONMANAGER.....33
FIGURA 6. MÉTODO OPENCONNECTIONS() DE LA CLASE TRANSACTIONMANAGER.34
FIGURA 7. NUEVA ESTRUCTURA DE CARPETAS DE DOCTRINE 2.35
FIGURA 8. COMPARACIÓN DE LAS ESTRUCTURAS DE CARPETAS.36
FIGURA 9. PORCENTAJE DEL COEFICIENTE DE COMPETENCIA DE LOS EXPERTOS.46
FIGURA 10. PORCENTAJE DE LA FRECUENCIA ABSOLUTA.....50
FIGURA 11. GRÁFICA DE COMPARACIÓN DE RENDIMIENTO EN DOCTRINE.....54

ÍNDICE DE ANEXOS

ANEXO 1. ENCUESTA PARA DETERMINAR EL COEFICIENTE DE COMPETENCIA60
ANEXO 2. TABLA PARA MARCAR EL COEFICIENTE DE CONOCIMIENTO.....61
ANEXO 3. TABLA PARA MARCAR EL COEFICIENTE DE ARGUMENTACIÓN.....61
ANEXO 4. PATRÓN DE VALORES PARA EL COEFICIENTE DE ARGUMENTACIÓN61
ANEXO 5. RESULTADO DEL COEFICIENTE DE LOS EXPERTOS.....62
ANEXO 6. ENCUESTA PARA LA VALIDACIÓN DEL TRABAJO62
ANEXO 7. TABLAS DE DISTRIBUCIÓN CHI CUADRADA O X2.....64



INTRODUCCIÓN

La utilización de herramientas que proporcionen facilidades en la creación y desarrollo de sistemas de software ha tomado gran auge en la actualidad; ejemplo de esto es la utilización de Marcos de Trabajo (MT). Estos sistemas tienen como objetivo mejorar la rapidez, productividad y profesionalidad en cuanto a la elaboración de sistemas web dinámicos o no y sistemas complejos. Además, tienden a promover una estructura estándar para las aplicaciones, así como el uso de las buenas prácticas de la programación [1].

Cuba cuenta con diversas instituciones que se dedican al desarrollo de software, una de ellas es la Universidad de las Ciencias Informáticas (UCI) en la cual existen numerosos proyectos productivos para el desarrollo de software. Varios de estos utilizan los MT para el desarrollo de aplicaciones ya sean web o de escritorio. La utilización de MT conlleva al uso de diversos Mapeadores de Objetos Relacionales (del inglés ORM) seleccionados de acuerdo con la compatibilidad, lógica del negocio y software definidos por dichos proyectos.

En la Facultad 3 de la UCI, el Centro de Informatización de la Gestión de Entidades (CEIGE) cuenta con varios proyectos que hacen uso del MT Sauxe. Dicho MT está basado en el MT Zend y utiliza en la capa de acceso a datos el Lenguaje de Consulta de Datos (DQL) que implementa Doctrine¹. Además, utiliza ExtJS² en la capa de presentación, por la amplia gama de componentes que se pueden reutilizar y para mostrarle al usuario una interfaz más amigable [2].

El MT Sauxe utiliza actualmente la versión 1.2.2 del marco de persistencia de datos de Doctrine. El uso de este provoca que a la hora de ejecutar consultas que necesiten manejar gran cantidad de información o que necesiten hacer referencia a varias tablas, se obtenga un rendimiento reducido. Por ejemplo, al realizarse una consulta al sistema gestor de base de datos, el sistema tendrá que convertir la consulta al lenguaje de

¹ **Doctrine:** es un marco de persistencia de datos que trabaja con lenguaje de programación PHP 5.2.3 o superior, situándose sobre su poderosa capa de abstracción de la base de datos.

² **ExtJS:** es una biblioteca Javascript ligera y de alto rendimiento, compatible con la mayoría de navegadores para crear páginas web y aplicaciones dinámicas.



consulta estructurado o SQL (acrónimo del inglés Structured Query Language) del proveedor usado; enviar la consulta, leer los registros, limpiarlos y convertirlos a objetos.

Pruebas realizadas sobre el Sistema de Planificación de Recursos Empresariales Xedro (ERP), el cual fue desarrollado empleando el MT Sauxe, demostraron que se obtenía un rendimiento reducido de la aplicación y que los tiempos de ejecución se elevaban al hacer uso del marco de persistencia de datos de Doctrine. A partir de los resultados obtenidos se evidenció que el rendimiento era mayor cuando no se hacía uso del marco de persistencia de datos de Doctrine [3]. Además, al ejecutar funciones como cargar los elementos del menú, donde no se maneja gran cantidad de datos y la relación entre tablas es mínima, el tiempo de ejecución estaba por encima de los 1.3 segundos.

Igualmente, debido a que en el MT Sauxe existen varios componentes y que la base de datos trabaja con esquemas independientes relacionados a cada uno de estos, al realizar llamadas a funciones entre varios componentes el marco de persistencia de datos de Doctrine tiene que abrir una nueva conexión, realizar la consulta y cerrar la conexión por cada llamada que se realice, siendo este un proceso repetitivo. Este proceso, ligado a la forma que se maneja la herencia, donde se debe hacer referencia a numerosas tablas para obtener los datos, provoca que la aplicación tenga un rendimiento reducido.

Otro inconveniente que presenta el uso de esta versión de Doctrine es que a partir del 1 de junio de 2011 se dejó de dar soporte, provocando que no se corrijan vulnerabilidades en el código que puedan atentar contra el rendimiento [4].

A partir de la problemática antes descrita se define como **problema a resolver**: ¿Cómo mejorar el funcionamiento de la capa de acceso a datos en el MT Sauxe para aumentar el rendimiento de las soluciones desarrolladas sobre él?

Con el fin de resolver el problema identificado se propone como **objetivo general**: obtener la versión 2 de la capa de acceso a datos del marco de trabajo Sauxe para aumentar el rendimiento de las soluciones desarrolladas sobre él.

Se define como **objeto de estudio**: marco de trabajo para la persistencia de datos y como **campo de acción**: capa de acceso a datos del marco de trabajo Sauxe.



Para darle cumplimiento al objetivo general se desglosó el mismo en los siguientes **objetivos específicos**:

- ✓ Construir el Marco Teórico de la investigación relacionada con los marcos de trabajo para la persistencia de datos existentes para identificar buenas prácticas y posibles puntos de reutilización.
- ✓ Realizar la migración de la capa de acceso a datos del MT Sauxe para aumentar el rendimiento de las soluciones desarrolladas sobre él.
- ✓ Validar la solución.

La investigación parte de la siguiente **idea a defender**: si se obtiene la versión 2 de la capa de acceso a datos en el MT Sauxe, se aumentará el rendimiento de las soluciones desarrolladas sobre él.

En el desarrollo de la investigación se han utilizado los siguientes **métodos científicos**:

Métodos Teóricos

Los Métodos Teóricos son factibles en el estudio de las características poco observables del objeto de investigación. Dentro de este grupo se utilizan:

- **Histórico – Lógico**: Permite realizar un estudio del estado del arte de los principales ORM existentes que se utilizan en la actualidad. Además, facilita el análisis evolutivo de dichas herramientas en materias de diseño y su tendencia actual.
- **Inducción – Deducción**: Al aplicar este método se estudian algunos de los ORM más utilizados actualmente, con el fin de definir sus características particulares y beneficios que brinda su uso.

Métodos Empíricos

Los métodos empíricos tienen gran importancia pues permiten efectuar el análisis preliminar de la información, así como verificar y comprobar las concepciones teóricas. Dentro de este grupo se utiliza:

- ✓ **Observación**: Con su utilización se centralizó toda la atención en la situación actual existente en el MT Sauxe y en los inconvenientes de la utilización de la versión 1.2.2 de Doctrine.
- ✓ **Entrevista**: Este método brinda apoyo a la incorporación de conocimientos mediante las entrevistas planificadas efectuadas a los especialistas de áreas de CEIGE, así como en la selección de los expertos en las pruebas que se realizan para validar la migración.



El presente trabajo se encuentra estructurado en tres capítulos, resumidos de la siguiente forma:

- ✓ En el **Capítulo 1:** “Fundamentación Teórica” se realiza el estudio del estado del arte, así como el uso de los marcos de persistencia de datos y de las herramientas que se utilizan en la migración.
- ✓ En el **Capítulo 2:** “Desarrollo de la migración” se realiza una descripción detallada de la migración propuesta, identificando las etapas y las actividades a realizar.
- ✓ En el **Capítulo 3:** “Validación de la Migración” se describen los resultados obtenidos en la validación de la propuesta de solución mediante el método de expertos y aplicación de pruebas de rendimiento.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se reflejan los temas fundamentales que sustentan la investigación. En él se abordan varios conceptos asociados a los marcos de persistencia de datos, específicamente Doctrine. Debido a que la solución a desarrollar está enmarcada a las soluciones desarrolladas sobre el MT Sauxe, se seleccionan las tecnologías y herramientas informáticas a utilizar. Además, se hace un estudio del método Delphi para su aplicación a la hora de validar la migración.

1.2. Marco de Trabajo de Persistencia de Objetos Relacionales

Con el propósito de conocer posibles puntos de reutilización y buenas prácticas en el uso de los marcos de persistencia de datos se hace un estudio de los mismos.

Marco de trabajo de persistencia de datos

Es una estructura de software orientada a mejorar la productividad a través de la reutilización de código. Es un conjunto de clases creadas para apoyar la escritura de código en un contexto determinado. *“Un marco de trabajo de persistencia es una biblioteca de clases que facilita la tarea del programador al permitirle guardar objetos en bases de datos relacionales de manera lógica y eficiente, de otra manera se debería hacer manualmente, y este es un proceso tedioso, repetitivo y propenso a errores”* [5].

1.2.1. Hibernate

Hibernate es un conjunto de proyectos relacionados, que permiten a los desarrolladores utilizar estilo POJO³, que no son más que modelos de dominio en sus aplicaciones de manera que extienden mucho más allá del mapeo objeto relacional. Por lo tanto, es una herramienta que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML⁴) o anotaciones en las entidades que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU/LGPL⁵. Ofrece un lenguaje de consulta de datos llamado HQL (del inglés Hibernate Query Language) [6].

³ **POJO:** del inglés (Plain Old Java Object), utilizado por programadores que implementan con Java para enfatizar el uso de clases simples y que no dependen de un MT en especial.

⁴ **XML:** por sus siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible).

⁵ **GNU/LGPL:** Licencia Pública General Reducida de GNU, o más conocida por su nombre en inglés GNU Lesser General Public License.



Rendimiento:

Hibernate trata con la reflexión de Java y el aumento de clases en tiempo de ejecución utilizando una biblioteca de generación de código Java muy poderosa y de alto rendimiento llamada CGLIB. Este se utiliza para extender clases Java e implementar interfaces Java en tiempo de ejecución [7].

Compatibilidad con el gestor de base de datos PostgreSQL:

Hibernate no es compatible con el gestor de base de datos PostgreSQL, es una herramienta para la plataforma Java y disponible también para .Net⁶ con el nombre de NHibernate.

Compatibilidad con el MT Sauxe:

Debido a que el servidor de base de datos que utiliza el MT Sauxe es PostgreSQL no se puede utilizar Hibernate como marco de persistencia en la capa de acceso a datos.

Uso del Lenguaje PHP:

Hibernate utiliza como lenguaje de programación Java.

Comunidad que respalde su soporte:

Hibernate cuenta con una amplia comunidad y con cientos de colaboradores a lo largo de todo el mundo por lo que su soporte está garantizado.

Buenas prácticas:

- ✓ Por lo general, no es aconsejable crear una conexión cada vez que se interaccione con la base de datos. Para evitar esta situación, las aplicaciones Java, suelen utilizar una piscina (del inglés pool) de conexiones. Cada subproceso de la aplicación que realiza solicitudes sobre la base de datos, solicita una conexión a la piscina, devolviéndola cuando todas las operaciones SQL han sido ejecutadas.
- ✓ Cuando ocurra una excepción, se debe deshacer la operación y cerrar la sesión.
- ✓ Mapear cada clase en su propio fichero, evitando usar un solo documento para mapear todas las clases.

1.2.2. Propel

Propel es un ORM de código abierto para PHP5. Permite el acceso a la base de datos utilizando un conjunto de objetos, proporcionando una API⁷ simple para almacenar y recuperar datos. Este le brinda al desarrollador de aplicaciones web las herramientas para trabajar con bases de datos, de la misma manera

⁶ **.Net:** es un MT de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware.

⁷ **API:** Interfaz de programación de aplicaciones (IPA) o *API* (del *inglés* Application Programming Interface).



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

que se trabaja con otras clases y objetos en PHP. Le da a la base de datos una API bien definida utilizando los estándares PHP5, carga automática e iteradores [8].

Rendimiento:

Lo más importante es que Propel utiliza *PDO* (del inglés PHP Data Objects) en lugar de Creole como DBAL (Capa de Abstracción de Base de Datos), ofreciendo una significativa mejora de rendimiento [9].

Compatibilidad con el gestor de base de datos PostgreSQL:

Este soporta MySQL, PostgreSQL, SQLite, MSSQL y Oracle [8].

Uso del Lenguaje PHP:

Utiliza PHP5 o una versión superior.

Compatibilidad con el MT Sauxe:

Puede adaptarse perfectamente al MT Sauxe debido a que está basado en PHP como lenguaje de programación y soporta el gestor de base de datos PostgreSQL.

Comunidad que respalde su soporte:

Propel posee una amplia comunidad y con un prestigio elevado mundialmente por sus potencialidades.

Buenas prácticas:

- ✓ A cada URL⁸ se le debe crear la ruta asociada. Y esto es obligatorio si quitas las reglas de enrutamiento por defecto.
- ✓ En caso de que se elimine un método generado se debe eliminar la plantilla generada asociada a ese método.

1.2.3. Django

Django es un entorno de desarrollo web escrito en Python⁹ que fomenta el desarrollo rápido y el diseño limpio y pragmático. Django es un MT de código abierto que permite construir aplicaciones web más rápido y con menos código. Está compuesto por un mapeo objeto relacional para convertir los modelos de datos, un apoyo en configuración de URL, un sistema completo de plantillas, sistemas para realizar procesos de

⁸ **URL:** es un localizador uniforme de recursos, denominado URL (sigla en inglés de uniform resource locator).

⁹ **Python:** es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

solicitud y administración de base de datos relacional, lo que permite un óptimo desarrollo de aplicaciones [10].

Rendimiento:

Django es muy potente, porque permite obtener provecho de las ventajas del lenguaje para el cual está diseñado (Python).

Compatibilidad con el gestor de base de datos PostgreSQL:

A pesar de que Django es compatible con el gestor de base de datos PostgreSQL se recomienda que para su uso se instale el paquete *psycopg2*; el uso de PostgreSQL en Django logra un delicado equilibrio entre costo, características, velocidad y estabilidad [11].

Uso del Lenguaje PHP:

Django está creado sobre el lenguaje Python por lo que usa PHP.

Compatibilidad con el MT Sauxe:

La compatibilidad de Django con el MT Sauxe se ve limitada debido a que los dos utilizan lenguajes diferentes.

Comunidad que respalde su soporte:

Django cuenta con una comunidad en crecimiento y ya se ha extendido a varios idiomas como son: español, inglés y portugués.

Buenas prácticas:

- ✓ Sigue el principio DRY (conocido también como Una vez y sólo una).
- ✓ Utiliza una modificación de la arquitectura Modelo-Vista-Controlador (MVC) esta forma de trabajar permite que sea pragmático.

1.2.4. SubSonic

SubSonic es un marco de persistencia que trabaja en la capa de acceso a datos usando Linq¹⁰ 3.0, es Open Source y está creado para la tecnología ASP.Net. Provee un lenguaje de consultas que permite la extracción de datos de la base de datos de manera muy simple sin menospreciar la potencia. Además de lo

¹⁰ **Linq**: Language Integrated Query (LINQ) es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes .Net.



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

mencionado, ofrece en la estructura de clases generadas una separación clara que permite implementar el patrón MVC en la aplicación [12].

Rendimiento:

El rendimiento en Subsonic no está garantizado del todo debido a que se deja en manos del usuario la elección de donde se va a guardar el resultado en caché ya sea a nivel de aplicación o a nivel de propio Subsonic [13].

Compatibilidad con el gestor de base de datos PostgreSQL:

Subsonic está creado específicamente para el gestor de base de datos PostgreSQL por lo que su compatibilidad está garantizada [12].

Uso del Lenguaje PHP:

Al estar creado sobre y para tecnología ASP.Net el lenguaje es Java.

Compatibilidad con el MT Sauxe:

La compatibilidad de Django con el MT Sauxe se ve limitada debido a que los dos utilizan lenguajes diferentes.

Comunidad que respalde su soporte:

Django cuenta con una comunidad poco abundante y su reputación está limitada sobretodo porque es bastante joven.

Buenas prácticas [4]:

- ✓ Los objetos de negocio deben comunicarse con los objetos de acceso a datos a través de interfaces, es decir, siempre depende de abstracciones.
- ✓ Los objetos de acceso a datos deben implementar interfaces definidas por el "cliente" en la capa de lógica de negocio.

1.2.5. Doctrine 1.2.2

Los estudios concernientes al rendimiento realizados sobre Xedro ERP evidenciaron que, el uso de la versión 1.2.2 de Doctrine provocaba un rendimiento reducido del sistema [3], arrojando tiempos de respuesta superior a los 10 segundos al insertar y/o consultar grandes volúmenes de datos o al realizar consultas sobre múltiples tablas en la base de datos.



1.2.6. Doctrine 2.2.0

Doctrine 2 es la versión superior de este marco de persistencia de datos para PHP 5.3.0+ que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos [14].

Doctrine 2 implementa diseño Mapa de Datos (del inglés DataMapper) proponiendo que los objetos de negocio sean POCO¹¹ (POJO en Java), o sea, que no tengan nada de código que los acople a una tecnología de persistencia, ya que no es su función manejarla. Los objetos de negocio solo deben preocuparse de cumplir con el modelo del dominio del diagrama de clases con las asociaciones pertinentes.

Rendimiento [14]:

- ✓ El uso de PHP 5.3 brinda un gran rendimiento.
- ✓ Mejor hidratación y algoritmo optimizado.
- ✓ Nuevas implementaciones de consulta y el resultado caché.
- ✓ El código más explícito y menos mágico, como resultados: el código mejor y más rápido.

El rendimiento en Doctrine 2 es muy superior a Doctrine 1. En la **Tabla I** se muestra un estudio realizado a un sistema haciendo uso de dos versiones diferentes de Doctrine (Doctrine 1.2.2 y Doctrine 2.2.0), la comparación se hizo teniendo en cuenta el tiempo de ejecución a la hora de insertar de 5000 y 10000 registros respectivamente en una base de datos.

Tabla I. Comparación de rendimiento en las versiones de Doctrine [14].

	Registros	Tiempo	Registros	Tiempo
Doctrine 1	5000	4.3 seg.	10000	7 seg.
Doctrine 2	5000	1.4 seg.	10000	3.5 seg.

Compatibilidad con el gestor de base de datos PostgreSQL:

Doctrine 2 soporta múltiples gestores de base de datos (MySQL, PostgreSQL, SQLite, MSSQL y Oracle).

Uso del Lenguaje PHP:

Este está escrito en PHP pero solo compatible con las versiones superiores a 5.3.0.

¹¹ **POCO**: Componentes Portables (en inglés POrtable COmponents).



Compatibilidad con el MT Sauxe:

La compatibilidad con el MT Sauxe está garantizada debido a que este utiliza en su capa de acceso a datos la versión previa de este marco de persistencia de datos.

Comunidad que respalde su soporte:

Cuenta con una amplia comunidad y colaboradores alrededor de todo el mundo, es uno de los marcos de persistencias más usados y con mayor prestigio a escala mundial.

Buenas prácticas:

- ✓ No utilizar propiedades públicas en las entidades.
- ✓ Es importante limitar las relaciones tanto como sea posible.
- ✓ Se deben evitar las asociaciones bidireccionales, si es posible.
- ✓ Eliminar las asociaciones que no sean esenciales.

Resumen del estudio de los marcos de persistencia de datos:

El estudio de los marcos de persistencia de datos demostró las potencialidades que representa el uso de Doctrine 2 debido a que este brinda un alto rendimiento y que en esta versión se encuentran reflejadas las potencialidades que implementan los demás marcos de persistencia de datos.

1.3. Tecnologías y herramientas para el desarrollo

La palabra tecnología hace referencia al conjunto de nociones técnicas, aplicadas al diseño y construcción de productos y servicios de acuerdo a las necesidades humanas [15].

1.3.1. Subversion

Subversion es un sistema de control de versiones libre (código abierto). Es decir, maneja ficheros y directorios a través del tiempo; un árbol de ficheros en un repositorio central y el repositorio es como un servidor de ficheros ordinario, exceptuando que guarda todos los cambios hechos a sus ficheros y directorios. Esto permite recuperar versiones antiguas de los datos, o examinar la historia de cómo cambiaron sus datos. En este sentido, muchas personas piensan en un sistema de control de versiones como una especie de "máquina del tiempo" [16].

Subversion puede acceder al repositorio a través de redes, lo que permite que pueda ser utilizado por personas en diferentes equipos. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto a través del cual todas las modificaciones que se producen. Y debido a que el trabajo se versiona, la calidad es la compensación por la pérdida de ese



conducto, si algún cambio que se haga en los datos es incorrecto, simplemente se deshace ese cambio [16].

1.3.2. RapidSVN v0.12.0

RapidSVN es una interfaz gráfica de usuario multiplataforma para el sistema de revisión de Subversion escrito en C++. Con Subversion se quiere construir un mejor cliente visual que utilice las mejores características de los clientes de otras arquitecturas de control de revisiones. Si bien es bastante fácil para los nuevos usuarios de Subversion para trabajar con él, también, debe ser lo suficientemente potente como para hacer que los usuarios experimentados sean aún más productivos [17].

1.3.3. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto relacional, distribuido bajo licencia BSD (del inglés Berkeley Software Distribution) para bases de datos y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones ha logrado incorporar las potencialidades de otros sistemas gestores de bases de datos incluso comerciales. PostgreSQL utiliza un modelo cliente-servidor y es multiprocesos en vez de multihilos para lograr la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [18].

Sus características técnicas lo hacen uno de los más potentes y robustos del mercado. Durante su desarrollo las características que más se evidencian son: estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Este presenta un rendimiento elevado con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema [18].

1.3.4. Netbeans

La plataforma de NetBeans es principalmente un MT de aplicación genérica para las aplicaciones de escritorio en Java aunque también puede usarse para aplicaciones web reconociendo lenguajes como PHP, JavaScript, C y C++. El beneficio principal de esta plataforma es su arquitectura modular predefinida. Los beneficios secundarios son las soluciones reusables en combinación con las herramientas proporcionadas por su SDK¹², NetBeans IDE¹³ [19].

¹² **SDK**: Kit de desarrollo de software o SDK (siglas en inglés de Software Development Kit).

¹³ **IDE**: Un entorno de desarrollo integrado (de su acrónimo en inglés de Integrated Development Environment).



1.3.5. Apache

Apache es un servidor web gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Es válido destacar las siguientes características: es multiplataforma (aunque idealmente está preparado para funcionar bajo Linux), muy sencillo de configurar, es open-source (código abierto), muy útil para proveedores de servicios de internet que requieran miles de sitios pequeños con páginas estáticas, amplias librerías de PHP y Perl a disposición de los programadores, posee diversos módulos que permiten incorporarle nuevas funcionalidades (estos son muy simples de cargar) y es capaz de utilizar lenguajes como PHP, TCL, Python, entre otros [20].

1.3.6. Sauxe

Contiene un conjunto de componentes reutilizables que proveen la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor agilidad en el proceso de desarrollo y las aplicaciones de gestión. Sauxe está basado en el MT Zend, además utiliza en la capa de acceso a datos DQL. Utiliza ExtJS en la capa de presentación, por la amplia gama de componentes que se pueden reutilizar y para mostrarle al usuario una interfaz más amigable [2].

1.3.7. Marco de Trabajo Zend

Zend es un MT de código abierto para desarrollar aplicaciones y servicios web con PHP5. Este es una implementación que usa un código totalmente orientado a objetos. Su estructura de componentes es única; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada, permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como “uso a voluntad” (del inglés “use at will”) [21].

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend conforman un potente y extensible MT de aplicaciones web al combinarse. Este ofrece un gran rendimiento y una robusta implementación del MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML¹⁴, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos [22].

1.3.8. PHP

PHP cuyas siglas responden a un acrónimo recursivo (PHP: Hypertext Preprocessor) es un lenguaje de programación interpretado, originalmente diseñado para la creación de páginas web dinámicas. Sigue un

¹⁴ HTML: siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto).



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

estilo clásico ya que es un lenguaje de programación con variables, sentencias condicionales, bucles y funciones. Está más cercano a JavaScript o al lenguaje C, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor [15].

Resumen del estudio de las herramientas:

Las herramientas antes mencionadas son las definidas en el documento “*Vista de Entorno de Desarrollo Tecnológico del proyecto Sauxe_v2.2*” se hace un estudio para destacar algunas de sus potencialidades.

1.4. Validación de la solución

Debido a que en el área donde se va a aplicar la migración se cuenta varias especialistas con amplios conocimientos en cuanto a los marcos de persistencias de datos se hace un estudio del método Delphi para validar la solución a través de criterio expertos.

1.4.1. Método Delphi

Según Linstone y Turoff el método Delphi es un método de estructuración de un proceso de comunicación grupal, que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo. Este método pretende extraer y maximizar las ventajas que presentan los métodos basados en grupos de expertos y minimizar sus inconvenientes. Para llevar a cabo esta técnica se selecciona un grupo de expertos que deben dar respuesta a un conjunto de preguntas que forman parte de los cuestionarios a realizar. La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los expertos a consultar [23].

Para aplicar el método se siguen tres etapas fundamentales [24].

- ✓ Elección de expertos.
- ✓ Elaboración y lanzamiento de los cuestionarios.
- ✓ Desarrollo práctico y explotación de resultados.

Delphi presenta tres características fundamentales [23]:

- ✓ **Anonimato:** No debe existir contacto entre los participantes, pero el encargado de la encuesta sí puede identificar a cada participante y sus respuestas.
- ✓ **Iteración y realimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Como se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

- ✓ **Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

1.4.2. Proceso de selección de expertos

Uno de los aspectos fundamentales a considerar metodológicamente para la aplicación práctica del método, es la selección del grupo de expertos a encuestar. Se considera como experto el individuo en sí, grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

De esta definición se deduce, entre otros, como requisitos básicos para la selección de un experto los que se plantean a continuación [25]:

- ✓ Experiencia en el tema a consultar, dado por sus años de trabajo.
- ✓ Conocimientos teóricos adquiridos a través de las distintas formas de superación.
- ✓ Grado académico o científico alcanzado en relación al tema.

La selección de los expertos, así como la cantidad para la realización del Delphi, es una tarea fundamental, aunque no hay forma de determinar el número óptimo de expertos para participar en una encuesta. Estudios realizados por investigadores de la Rand Corporation [25] señalan que es necesario un mínimo de siete expertos, teniendo en cuenta que el error disminuye notablemente por cada experto añadido hasta llegar a los siete, pero no es aconsejable recurrir a más de 30 expertos, pues la mejora en la suposición es muy pequeña y normalmente el incremento en coste y trabajo de investigación no compensa dicha mejora.

1.4.3. Desarrollo práctico y explotación de los resultados

Para la valoración de consensos para los expertos, existen distintas técnicas y modelos, pero es más conveniente aplicar el modelo matemático Torgerson, el cual es muy útil cuando las escalas empleadas en los instrumentos aplicados a los expertos son ordinales. Además de lo antes planteado, una de las desventajas del método Delphi radica en la subjetividad de los criterios emitidos, y para resolver este problema se opta por emplear este modelo matemático que permite asignar un valor de escala a cada indicador y determinar límites entre cada categoría, y así, obtener los límites reales entre las categorías ordinales y sus correspondientes a escala de intervalo, y entre cada uno de los rangos que componen los criterios evaluativos dados por los expertos.

De esta forma se puede conocer hasta qué valores reales se puede considerar que la variable es: muy adecuado, bastante adecuado, adecuado, poco adecuado y no adecuado.



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Resumen del estudio del método Delphi:

El estudio del método Delphi permitió identificar sus principales ventajas así como obtener los conocimientos necesarios para su correcta aplicación a partir del criterio de expertos.

1.5. Conclusiones parciales

El estudio realizado sobre los marcos de trabajo para la persistencia de datos arrojó como resultado que estos presentaban características comunes y que Doctrine 2 presenta las principales potencialidades de los marcos estudiados.

El análisis realizado sobre el funcionamiento de la capa de acceso a datos de Xedro ERP, permitió identificar debilidades relacionadas con el rendimiento de las aplicaciones desarrolladas sobre él en el acceso a los datos.

Se realizaron estudios de las herramientas y métodos para sentar las bases de la investigación.



CAPÍTULO 2: DESARROLLO DE LA MIGRACIÓN

2.1. Introducción

En este capítulo se propone una guía para facilitar la migración de la capa de acceso a datos del MT Sauxe y de las soluciones desarrolladas sobre él. Se definen las directrices mediante la cual los proyectos implementados con Sauxe o que lo utilizan para su solución puedan realizar la migración; se muestra una forma intuitiva, sencilla y detallada a seguir para su realización.

2.2. Contenido de la migración

La migración propuesta ofrece un avance en cuanto al rendimiento de la capa de acceso a datos para las soluciones desarrolladas sobre Sauxe. La misma cuenta con tres etapas fundamentales las cuales están relacionadas entre sí, mostrando una secuencia lógica y organizada para su ejecución.

2.3. Etapa Inicial

Inicialmente se lleva a cabo el estudio de la arquitectura del MT Sauxe, donde se valora el contenido de cada una de las capas que contiene. Se realiza la integración donde el propósito de esta actividad es la integración de Doctrine 2 con el MT Sauxe y como consecuencia se hace necesaria la creación de directorios donde se añaden las clases y funcionalidades para la integración. Además se adicionan los parámetros de configuración y las nuevas librerías.

2.3.1. Estudio de la arquitectura de Sauxe

La arquitectura de Sauxe está basada en capas aunque en una de sus capas implementa el patrón Modelo Vista Controlador. Está compuesta básicamente por cinco niveles o capas [26]:

1. **Capa de Presentación:** En esta capa se emplea las facilidades que brinda el MT ExtJS para la construcción de interfaces amigables a la vista de los usuarios. ExtJS centra su desarrollo en tres componentes fundamentales JS-File, CSS-File y Client-page y Server-Page.
2. **Capa de Control o Negocio:** En esta capa se emplea el patrón de arquitectura MVC. De forma vertical al modelo descrito hasta este momento, estarán los aspectos fundamentales del sistema así como el encargado del tratamiento de la seguridad a nivel de aplicación Web.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

3. **Capa de Acceso a Datos:** En esta capa estará presente el marco de persistencia de datos de Doctrine, para la comunicación con el servidor de datos mediante el protocolo PDO¹⁵, también estará un Persistidor de Configuración que es el encargado de comunicarse vía XML con los ficheros de configuración del sistema denominado FastResponse.

4. **Capa de Datos:** En esta capa estará ubicado como servidor de base de datos PostgreSQL y un conjunto de Ficheros de Configuración de la arquitectura tecnológica.

5. **Capa de Servicio:** En esta última capa se encuentran todos los subsistemas que prestan y consumen servicios entre sí.

Esta gran estructura descrita, se comunica con unas series de servicios Web mediante protocolos SOAP¹⁶, e IoC¹⁷, que interactúan con el sistema proporcionándole un conjunto de funcionalidades tanto de seguridad como de negocio [26].

2.3.2. Estudio de la estructura de carpetas de los componentes de Sauxe

Cada uno de los componentes del MT Sauxe contiene la misma estructura de carpetas en la capa de datos, en la **Figura 1** se muestra el componente seguridad con su estructura de carpetas. Es importante estudiar esta estructura debido a que durante migración sufre modificaciones.

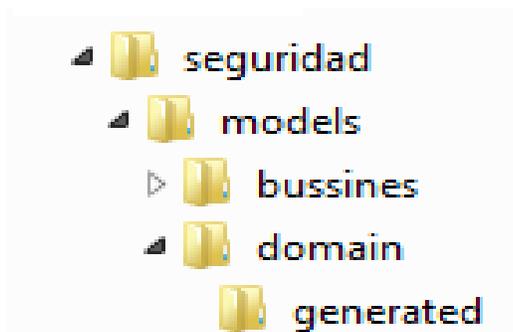


Figura 1. Estructura de carpetas del componente seguridad.

¹⁵ **PDO:** acrónimo del inglés PHP Data Objects, es una extensión que provee una capa de abstracción de acceso a datos para PHP 5.

¹⁶ **SOAP:** (siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

¹⁷ **IoC:** (siglas de *Inversion of Control*) es el mecanismo de integración de los componentes de Sauxe.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

2.3.3. Integración de Doctrine 2 con el MT Zend

La actividad de Integración es la base para realizar la implementación de la migración. Es imprescindible integrar la nueva versión del marco de persistencia de datos de Doctrine con el MT Zend en su versión 1.11 para recuperar la información de la base de datos. En la misma se definen cada una de las librerías que se utilizan y las clases que se modifican en la integración, además, se crean cada una de las funcionalidades y clases necesarias.

Para realizar la integración, inicialmente se:

- ✓ Seleccionan las clases necesarias para comenzar a implementar la actividad.
- ✓ Seleccionan las nuevas librerías a utilizar.
- ✓ Crean los directorios para ubicar las nuevas clases.
- ✓ Precisan los nuevos parámetros de configuración.

2.3.4. Estudio de las librerías seleccionadas

Las librerías escogidas son las propuestas por Doctrine 2 (Common, DBAL y ORM); se puede apreciar su ubicación en la **Figura 2** Es de suma importancia que el desarrollador realice un estudio de cada una de las librerías, para conocer cuándo, cómo y para qué se utilizan.

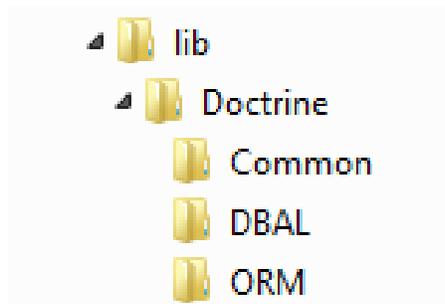


Figura 2. Ubicación de las nuevas librerías de Doctrine.

2.3.5. Estudio de las clases seleccionadas

Para lograr la integración de Doctrine 2 con el MT Sauxe es necesaria la creación del fichero **Doctrine.php**. Para la ubicación de dicho fichero se crea en **lib/ZendExt/App/** la carpeta **Resource** y dentro de esta se implementa la clase **ZendExt_App_Resource_Doctrine** como muestra la **Figura 3**.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

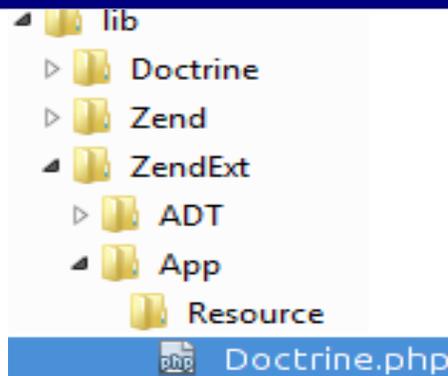


Figura 3. Ubicación de la clase Doctrine.php.

Para que la clase creada sea reutilizable en todas las aplicaciones dentro del MT Sauxe se debe crear como un recurso del MT Zend es por eso que la misma debe extender de **Zend_Application_Resource_ResourceAbstract**. La función principal de este recurso será la configuración de los parámetros necesarios para crear una instancia del **EntityManager**¹⁸ logrando el acceso a las funcionalidades del marco de persistencia de datos de Doctrine.

```
class ZendExt_App_Resource_Doctrine
extends Zend_Application_Resource_ResourceAbstract {

    public function init() {

        $options = Zend_Registry::get('configdoct');

        $classLoader = new \Doctrine\Common\ClassLoader('Doctrine');
        $classLoader->register();

        $classLoader = new \Doctrine\Common\ClassLoader(
            $options['entitiesPathNamespace'],
            $options['entitiesPath']
        );
        $classLoader->register();

        $classLoader = new \Doctrine\Common\ClassLoader(
            $options['proxiesPathNamespace'],
            $options['proxiesPath']
        );
        $classLoader->register();
        $classLoader = new \Doctrine\Common\ClassLoader(
            $options['repositoriesPathNamespace'],
            $options['repositoriesPath']
        );
        $classLoader->register();
    }
}
```

¹⁸ **EntityManager**: punto de acceso a la funcionalidad del marco de persistencia de datos de Doctrine 2.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

```
$cacheClass = 'Doctrine\Common\Cache\ArrayCache';

$cache = new $cacheClass();
$config = new \Doctrine\ORM\Configuration();
$config->setMetadataCacheImpl($cache);
$driverImpl = $config->newDefaultAnnotationDriver($options['module']);
$config->setMetadataDriverImpl($driverImpl);
$config->setQueryCacheImpl($cache);

$config->setProxyDir($options['proxiesPath']);
$config->setProxyNamespace($options['proxiesPathNamespace']);
$config->setAutoGenerateProxyClasses($options['autoGenerateProxyClasses']);

$em = \Doctrine\ORM\EntityManager::create(
    $options['conection'], $config
);

Zend_Registry::set('em', $em);
return $em;
}
```

Para la correcta configuración de Doctrine 2 será necesario ubicar en el fichero **apps/config.php** los parámetros de configuración y guardarlos para su posterior uso.

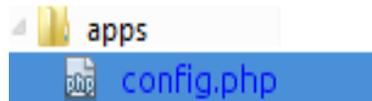


Figura 4. Ubicación del fichero de configuración.

Estos parámetros serán usados por Doctrine 2 para configurar los directorios de las entidades y los namespaces¹⁹.

```
$config['configdoct']['driver'] = "pdo_pgsql";
$config['configdoct']['entitiesPath'] = $_SERVER['DOCUMENT_ROOT'] . '/../apps/';
$config['configdoct']['entitiesPathNamespace'] = "";
$config['configdoct']['proxiesPath'] = $_SERVER['DOCUMENT_ROOT'] . '/../apps/';
$config['configdoct']['proxiesPathNamespace'] = "";
$config['configdoct']['repositoriesPath'] = $_SERVER['DOCUMENT_ROOT'] . '/../apps";
$config['configdoct']['repositoriesPathNamespace'] = "";
$config['configdoct']['cacheClass'] = "Doctrine\Common\Cache\ApcCache";
$config['configdoct']['autoGenerateProxyClasses'] = true;
$config['configdoct']['Path'] = $_SERVER['DOCUMENT_ROOT'] . '/../lib/ZendExt/";
```

Una vez creado el recurso y ubicados los parámetros de configuración de Doctrine 2 será necesario modificar la funcionalidad **OpenConnections()** de la clase **ZendExt_Aspect_TransactionManager.php** la cual es

¹⁹ **Namespaces:** espacios de nombre utilizados por Doctrine 2 para el manejo de los ficheros del mapeo.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

encargada de configurar y manejar las conexiones del marco de persistencia de datos de Doctrine en el MT Sauxe.

En la **Figura 5** se muestra una imagen con la ubicación de dicha clase.

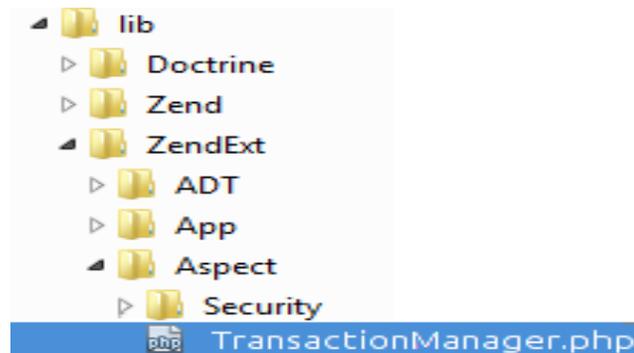


Figura 5. Ubicación de la clase *TransactionManager*.

Para lograr la configuración de Doctrine 2 a través de la funcionalidad **openConnections()** será necesario hacer uso de los parámetros anteriormente creados en el fichero de configuración **config.php**.

A continuación se hace mención a los nuevos parámetros de configuración que se utilizarán:

- ✓ **driver:** tipo de controlador que implementan los PDO de interfaz para permitir el acceso de PHP a las bases de datos. Debido a que el MT Sauxe utiliza PostgreSQL como gestor de base de datos se debe usar *pdo_pgsql*.
- ✓ **entitiesPath:** ruta en la que se encontrarán las entidades del mapeo.
- ✓ **entitiesPathNamespace:** nombre del *namespace* que deben asociado a la ruta de las entidades del mapeo.
- ✓ **proxiesPath:** ruta en la que se encontrarán las clases *proxy* pertenecientes a las entidades del mapeo.
- ✓ **proxiesPathNamespace:** nombre del *namespace* de las clases *proxy*.
- ✓ **repositoriesPath:** ruta en las que se encontrarán las clases *repository* pertenecientes a las entidades del mapeo.
- ✓ **repositoriesPathNamespace:** nombre del *namespace* de las clases *repository*.
- ✓ **cacheClass:** tipo de cache que utilizará el ORM Doctrine, en este caso: *Doctrine\Common\Cache\ApcCache*.
- ✓ **autoGenerateProxyClasses:** parámetro de tipo boolean para indicar si se generan o no de manera automática las clases *proxy*.

Los demás parámetros son los referentes a la conexión a la Base de Datos.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

De manera general es válido argumentar que es imprescindible el uso de los namespaces debido a que estos son usados por Doctrine 2 para evitar que las entidades extiendan de una clase base como en Doctrine 1.

Después de crear y configurar los parámetros necesarios se inicializa la funcionalidad de Doctrine haciendo una llamada al recurso anteriormente creado (**ZendExt_App_Resource_Doctrine::init()**). Seguidamente se podrá acceder a las funcionalidades del marco de persistencia de datos de Doctrine.

En la **Figura 6** se muestra la funcionalidad **OpenConexions** una vez adaptada a la nueva estructura de doctrine:

```
public function openConexions($module = null, $current = false) {
    try {
        $config = Zend_Registry::getInstance()->config->configdoct;
        $config->entitiesPathNamespace = $module . "\models";
        $config->proxiesPath = $config->proxiesPath . $module . "/models/Proxies";
        $config->proxiesPathNamespace = $module . "\models\Proxies";
        $config->repositoriesPathNamespace = $module . "\models\Repositories";
        if ($module == "Trace") {
            $config->proxiesPath = $config->Path . $module . "/models/Proxies";
            $config->entitiesPath = $config->Path;
            $config->repositoriesPath = $config->Path;
        }
        $configdoct = $config->toArray();
        $modulesconfig = ZendExt_FastResponse::getXML('modulesconfig');
        $bdconfig = $modulesconfig->conn;
        $configdoct['connection']['driver'] = 'pdo' . $bdconfig['gestor'];
        $configdoct['connection']['user'] = $bdconfig['usuario'];
        $RSA = new ZendExt_RSA();
        $configdoct['connection']['password'] = $RSA->decrypt($bdconfig['password'],
            '85550694285145230823', '99809143352650341179');

        $configdoct['connection']['host'] = $bdconfig['host'];
        $configdoct['connection']['port'] = $bdconfig['port'];
        $configdoct['connection']['dbname'] = $bdconfig['bd'];
        $configdoct['module'] = $config->entitiesPath . "$module/models";
        Zend_Registry::set('configdoct', $configdoct);
        ZendExt_App_Resource_Doctrine::init();
        $this->connection = Zend_Registry::get('em');
    } catch (Exception $e) {
        throw new ZendExt_Exception('E014', $e, null);
    }
    return $this->connection;
}
```

Figura 6. Método *OpenConexions()* de la clase *TransactionManager*.

2.3.6. Pruebas de Integración

Las pruebas realizadas en la migración tienen gran importancia, su propósito consiste en verificar si la misma está siendo implementada correctamente. En esta prueba se verifica la conectividad a la Base de Datos, la correcta configuración de los namespaces y la utilización de los nuevos parámetros.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

-Verificar la conectividad a la Base de Datos: En esta prueba se verifica la correcta conexión a la base de datos introduciéndole los parámetros de forma correcta e incorrecta.

-Verificar la correcta configuración de los namespaces: En esta prueba se acredita la correcta configuración de los namespaces mediante la llamada a entidades con namespaces existentes e inexistentes.

2.4. Etapa de Implementación

En la etapa de Implementación se rediseña la arquitectura base del sistema donde se redefine la nueva estructura de carpetas y archivos para la capa de acceso a datos del MT Sauxe. También se realiza la redefinición de la capa de acceso a datos en la cual se mapean las tablas y se definen las relaciones de la base de datos para la versión 2.2.0 de Doctrine. Además se re implementan los métodos y consultas de esta capa, ejecutándose la restructuración de la capa del negocio en caso de ser necesario.

2.4.1. Redefinir la Capa de Acceso a Datos

Debido a las diferencias existentes entre las versiones 1 y 2 de Doctrine se hace necesario realizar cambios en las capas. En la capa de acceso a datos se genera una nueva estructura de carpetas y se re implementa los métodos y las consultas. Es necesario realizar el mapeo de las tablas adaptándolas a la nueva estructura de Doctrine 2 y la migración manual de cada una de las clases ya mapeadas con Doctrine 1, esta es la acción que más tiempo conlleva no por su complejidad sino por el monto de objetos que se tienen que mapear y re implementar.

2.4.2. Estudiar la nueva estructura de carpetas de los componentes de Sauxe

La nueva estructura de carpetas de Doctrine 2 para las clases del modelo propone la creación de 2 nuevas carpetas (Proxy y Repositories), con el fin de organizar los ficheros generados por su tipo o función.

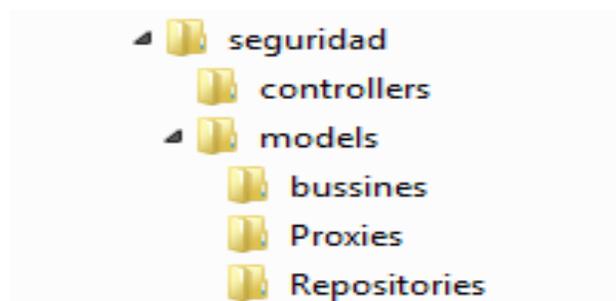


Figura 7. Nueva estructura de carpetas de Doctrine 2.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

2.4.3. Estudiar equivalencia entre la estructuras de carpetas

La **Figura 8** muestra la relación que existe entre la estructura de carpeta pertenecientes a las versiones 1.x y 2.x de Doctrine. Visualizando la nueva ubicación de los ficheros del modelo.

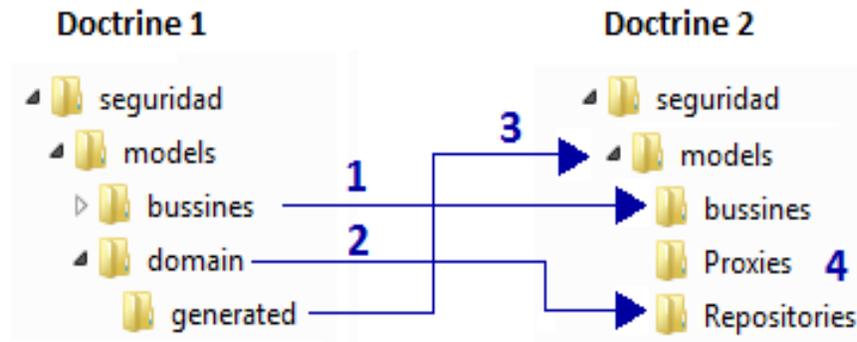


Figura 8. Comparación de las estructuras de carpetas.

1- Los ficheros localizados en la carpeta *bussines* que son los encargados de manejar el negocio de la capa de acceso a datos se mantienen en este directorio.

2- Las funcionalidades con las consultas que se encuentran en los ficheros localizados en la carpeta *domain* la deben realizar los ficheros ubicados dentro de la carpeta *Repositories* y además se deben modificar de acuerdo a la nueva estructura de Doctrine 2.

En la carpeta *Repositories* se encuentran los ficheros *Repository* generados como resultado del mapeo. Por cada tabla en la base de datos se genera un fichero *Repository*, su nomenclatura está compuesta por el nombre del fichero de mapeo de la tabla y *Repository* al final. Estos ficheros son clases que extienden de la clase *Doctrine\ORM\EntityRepository*.

3- Los ficheros que representan las tablas mapeadas ubicados en *generated* son generados con su nueva estructura en el directorio raíz de la carpeta *model*.

4- En la carpeta *Proxies* se ubican los ficheros generados como resultado del mapeo que contienen las clases *Proxy* usadas por Doctrine 2. Estas clases no son más que objetos que se ponen en el lugar del objeto real, y se utilizan para realizar varias funciones, pero principalmente, para la transparencia de carga diferida. Los objetos proxy con sus instalaciones de carga diferida ayudan a mantener el subconjunto de objetos que ya están en la memoria conectada con el resto de los objetos.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

2.4.4. Estudiar Formas de mapeo

Las formas de mapeo entre Doctrine 1 y Doctrine 2 son bastante diferentes. En Doctrine 1 se generan entidades que extienden de una clase base haciendo uso para la definición de tablas y columnas de funciones predefinidas dentro del marco de persistencia de datos de Doctrine. Sin embargo Doctrine 2 mapea la información a partir de anotaciones docblock²⁰ en el código. Doctrine 2 define su propio conjunto de anotaciones para suministrar asignación de metadatos objeto relacional.

En resumen y de una forma más sencilla de entender las anotaciones docblock, es verlas como comentarios que se muestran en las clases PHP.

Para definir el mapeo de una tabla en Doctrine 1.x se genera una clase que extiende de una clase base llamada Doctrine_Record y una funcionalidad llamada **setTableDefinition**. Por su parte Doctrine 2 define el mapeo de una tabla a través de la anotación *@Table* y se debe indicar además que esta es una entidad *@Entity*.

Definir Tablas

Doctrine 1

```
abstract class BaseSegRol extends Doctrine_Record {  
  
    public function setTableDefinition() {  
        $this->setTableName('mod_seguridad.seg_rol');  
    }  
}
```

Doctrine 2

```
<?php
```

```
namespace seguridad\models;
```

```
/**  
 * SegRol  
 *  
 * @Table(name="mod_seguridad.seg_rol")  
 * @Entity(repositoryClass="seguridad\models\Repositories\SegRolRepository")  
 */  
class SegRol {
```

²⁰ **Docblock**: es un tipo especial de comentario en el código que puede proporcionar información detallada acerca de un elemento.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

Variantes para definir columnas

A la hora de representar una columna en Doctrine 1 se llama a la función **hasColumn** y se definen cada uno de los parámetros definidos. Doctrine 2 a través de las anotaciones hace algo similar.

Doctrine 1

```
$this->hasColumn('idrol', 'decimal', null, array('ntype' => 'numeric', 'alltypes'  
=> array(0 => 'decimal',), 'nonnull' => true, 'primary' => true, 'sequence' => 'sec_segrol',  
));
```

Doctrine 2

```
/**  
 * @var float  
 *  
 * @Column(name="idrol", type="decimal", nullable=false)  
 * @Id  
 * @GeneratedValue(strategy="SEQUENCE")  
 * @SequenceGenerator(sequenceName="mod_seguridad.seq_rol_idrol_seq", allocationSize=1, initialValue=1)  
 */  
private $idrol;
```

Formas de representación de las relaciones entre tablas

La representación de relaciones entre tablas es otra de las cuestiones básicas a la hora de realizar una representación en entidades de mapeo.

En la versión 1 de Doctrine se generan 2 ficheros de Mapeo, el fichero *Base* que extiende de *Doctrine_Record* y otro fichero que extiende del fichero *Base* en este fichero se definen a través de la funcionalidad **setUp** las relaciones existentes entre tablas.

Doctrine 1

```
parent::setUp();  
$this->hasOne('NomDesktop', array('local' => 'iddesktop', 'foreign' => 'iddesktop'));
```



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

En este caso se está especificando que un usuario tiene una única instancia del objeto mapeado por la entidad `NomDesktop`.

Doctrine 2

```
/**
 * @var \NomDesktop $iddesktop
 *
 * @ManyToOne(targetEntity="NomDesktop")
 * @JoinColumn({
 *   @JoinColumn(name="iddesktop", referencedColumnName="iddesktop")
 * })
 */
private $iddesktop;
```

La anotación `@ManyToOne` indica que se tiene una única instancia del objeto mapeado por la entidad `NomDesktop` pero que un `NomDesktop` puede ser usado por muchos a la vez.

Doctrine 1

Como se explicaba anteriormente las relaciones entre tablas se mapean en el fichero que extiende de la clase base. En este caso se hace referencia a la clase que se genera del mapeo a la tabla que une la relación, es decir la que tiene ambas llaves primarias.

```
public function setUp() {
    parent::setUp();
    $this->hasMany('DatEntidadSegUsuarioSegRol', array('local' => 'idrol', 'foreign' => 'idrol'));
}
```

Doctrine 2

La particularidad del mapeo en Doctrine 2 está dada debido a que no se genera un fichero por la tabla en base de datos que guarda las llaves primarias de las tablas a unir, en este caso se hace referencia a la entidad de mapeo con que se posee la relación y mediante la anotación `@JoinTable` se le dice cuál es la tabla a través de la cual se relacionan. Otro aspecto que se debe tener en cuenta es que en estos casos se genera el constructor y dentro se inicializa la variable de la relación `ManyToOne` como un `ArrayCollection` definido por Doctrine.



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

```
/**
 * @var \Doctrine\Common\Collections\Collection
 *
 * @ManyToOne(targetEntity="DatSistema", inversedBy="idrol")
 * @JoinTable(name="mod_seguridad.dat_sistema_seg_rol",
 *   joinColumns={
 *     @JoinColumn(name="idrol", referencedColumnName="idrol")
 *   },
 *   inverseJoinColumns={
 *     @JoinColumn(name="idsistema", referencedColumnName="idsistema")
 *   }
 * )
 */
private $idsistema;

/**
 * Constructor
 */
public function __construct() {
    $this->idsistema = new \Doctrine\Common\Collections\ArrayCollection();
}
```

De manera general se ha representado la estrategia que adopta Doctrine para la representación de tablas mediante el mapeo objeto relacional en las dos versiones.

2.4.5. Consultas en las clases Repository

Una vez mapeadas las tablas de la base de datos se procede a modificar las funcionalidades y consultas que se estaban ubicadas en los ficheros de la carpeta *domain*; estas funcionalidades se deben ubicar en los ficheros dentro de la carpeta *Repositories*. En la versión 1 de Doctrine las funcionalidades quedaban de la manera que se evidencia a continuación:

```
public function cargarsistemahijos($idsistema, $idrol) {
    $query = Doctrine_Query::create();
    $result = $query->select('s.denominacion text, s.idsistema id,
        s.idsistema,s.denominacion,s.descripcion,s.icono,s.externa')
        ->from('DatSistema s')
        ->innerjoin('s.DatSistemaSegRol sr
            ON s.idsistema = sr.idsistema')
        ->where("sr.idrol = ? AND s.idpadre = ?
            AND s.idsistema <> s.idpadre", array($idrol, $idsistema))
        ->setHydrationMode(Doctrine :: HYDRATE_ARRAY)
        ->execute();
    return $result;
}
```



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

En Doctrine 2 las funcionalidades deben tener como parámetro además de los necesarios para la ejecución de las consultas, una variable que posea una instancia de la entidad *EntityManager* (*\$_em*).

```
public function cargarsistemahijos($idsistema, $idrol, $_em) {
    $result = $_em->createQuery('select s.denominacion text, s.idsisistema id,
                                s.idsisistema,s.denominacion,s.descripcion,
                                s.icono,s.externa
                                from seguridad\models\DatSistema s
                                JOIN s.idrol sr
                                where sr.idrol = ?1 AND s.idpadre = ?2
                                AND s.idsisistema <> s.idpadre')
        ->setParameter(1, $idrol)
        ->setParameter(2, $idsistema)
        ->getResult();
    return $result;
}
```

2.4.6. Funcionalidades en las clases Model

Las clases Model son las encargadas de manejar el negocio de la capa de acceso a datos.

```
class SegUsuarioModel extends ZendExt_Model {

    public function SegUsuarioModel() {
        parent::ZendExt_Model('seguridad');
    }
}
```

A la hora de hacer la llamada a las funcionalidades de las clases Repository se debe crear una instancia de la clase *TransactionManager* y hacer una llamada al método *openConnection* para obtener la instancia del *EntityManager* una vez hecho esto se llama la funcionalidad *getRepository* pasándole como parámetro en *namespace* de la clase que se quiere obtener el repositorio y finalmente se llama la funcionalidad.

```
public function cargarsistemahijos($idsistema, $idrol) {
    $mg = ZendExt_Aspect_TransactionManager::getInstance();
    $_em = $mg->getConnection('seguridad');
    $result = $_em
        ->getRepository("seguridad\models\DatSistema")
        ->cargarsistemahijos($idsistema, $idrol, $_em);
    return $result;
}
```



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

En este caso se está llamando la funcionalidad `cargarsistemahijos($idsistema, $idrol, $_em)` de la clase que tiene como *namespace* `seguridad\models\DatSistema'`

2.4.7. Redefinir la Capa del Negocio

En esta actividad no se redefine en su totalidad la capa de negocio, pero los cambios que se realizan son necesarios para la migración. Aquí se debe realizar la re implementación de las funcionalidades adaptándolas a la nueva estructura de la capa de acceso a datos y la creación o definición de directorios y ficheros.

Debido a que todo el negocio de la capa de acceso a datos se debe encontrar en las clases *Model* se debe redefinir la forma en que se llaman las funcionalidades. En Doctrine 1 las funcionalidades se encontraban en los ficheros de la carpeta *domain* es por eso que las llamadas a estos desde el negocio pueden variar o no respecto a la versión actual.

Doctrine 1

Se hacían llamadas a las funcionalidades (estáticas o no) que se encontraban en los ficheros de mapeo generados por Doctrine:

```
public function authenticateUser($user, $password) {  
    $observ = SegUsuarioDatSerautenticacion::verificarpass($user);
```

Doctrine 2

Se hacen llamadas a las funcionalidades (estáticas o no) que se encuentran en las clases *Model* dentro del directorio *bussines*. En este caso las llamadas a las funcionalidades se pueden realizar de la misma manera que con la versión anterior o se puede crear una instancia de la clase a la que se hace referencia y luego se llama la funcionalidad:

```
public function authenticateUser($user, $password) {  
    $userinstance = new SegUsuarioModel();  
    $verificar = $userinstance->verificarpass($user);
```

2.4.8. Probar la funcionalidad de los componentes de MT.

El MT está estructurado por componentes, al culminar con el mapeo y con la transformación de cada una de las clases y funcionalidades, se realizan pruebas a los componentes para garantizar su pleno funcionamiento operativo, estas pruebas son individuales a cada uno y a las integraciones de estos. En caso de que se



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

encuentren errores o inconsistencias en el código, en esta actividad se llevaría a cabo su refinamiento y corrección inmediatamente.

2.4.9. Actualizar estado de los elementos de configuración

Para actualizar el estado de los elementos de la configuración inicialmente se debe realizar una actualización del repositorio, aplicar cambios a los elementos de configuración, revisar actualizaciones e integridad de los elementos de configuración. Finalmente se deben publicar los elementos actualizados.

2.5. Etapa de Pruebas

En esta etapa se desarrollan las pruebas de rendimiento, que si bien en la actualidad existe una ausencia de su aplicación de manera satisfactoria en aplicaciones que así lo requieren, el resultado de ellas brinda el estado actual en cuanto al rendimiento de la aplicación a la cual se le realiza la prueba. Al utilizar una herramienta para la automatización de este proceso se logra disminuir los costos en recursos y en tiempo destinados a su realización que es la razón principal por lo cual no se realizan eventualmente las mismas [27].

En la migración propuesta se propone utilizar JMeter, teniendo en cuenta que la herramienta es ajustable a aplicaciones web para la comprobación de los recursos del sistema, además, supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. JMeter como herramienta de prueba dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la ventaja de simular para cada uno de esos escenarios miles de usuarios [27].

2.5.1. Preparar el entorno de pruebas

En esta actividad se debe preparar el sistema para realizarle las pruebas que se estimen conveniente. En la misma se crean los casos de pruebas y se pueden realizar de diferentes condiciones en dependencia de la importancia del sistema.

2.5.2. Realizar pruebas de rendimiento

En esta actividad, inicialmente se selecciona la herramienta para las pruebas, teniendo en cuenta que la herramienta seleccionada debe arrojar como resultado si se logra o no mejorar el rendimiento.

Para la realizar las pruebas en esta investigación se seleccionó la herramienta Apache JMeter la cual es una herramienta de carga para llevar acabo simulaciones sobre cualquier recurso de Software. Inicialmente diseñada para pruebas de estrés en aplicaciones web, hoy en día, su arquitectura ha evolucionado no sólo



CAPÍTULO II: DESARROLLO DE LA MIGRACIÓN

para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además en bases de datos, programas en Perl, requisiciones FTP y prácticamente cualquier otro medio [27].

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción. En este sentido, simula todas las funcionalidades de un Navegador ("Browser"), o de cualquier otro cliente, siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia. El componente principal de JMeter es denominado Plan de Prueba o Test Plan, en él se definen todos los aspectos relacionados con una prueba de carga, como: parámetros empleados por requisición, tipo de reportes a generarse con los resultados obtenidos, la posible reutilización de requisiciones compuestas por usuarios, entre otros aspectos [27].

2.5.3. Resolver las no conformidades

En la actividad se resuelven las no conformidades encontradas durante la ejecución de las pruebas en el tiempo determinado, teniendo en cuenta que el tiempo en esta actividad es muy importante, para la culminación de la migración, se debe tener el compromiso de cada uno de los involucrados en la migración para resolver las no conformidades en un tiempo record.

2.6. Conclusiones parciales.

En este capítulo se realizó la migración para la capa de acceso a datos del MT Sauxe permitiendo establecer una guía para la migración de las soluciones desarrolladas sobre él.

La migración se dividió en tres etapas: Inicial, Implementación y Pruebas, reuniéndose las tareas específicas en cada una de ellas, para el cumplimiento de los objetivos de la migración, definiéndose una serie de acciones a realizar, que constituyen una fortaleza para la propuesta en su conjunto.



CAPÍTULO 3: VALIDACIÓN DE LA MIGRACIÓN

3.1 Introducción

La validación de la migración se realizará mediante la aplicación del método Delphi, utilizando como fuente de información un grupo de personas que deben presentar un conocimiento elevado del tema a tratar. Este grupo estará conformado por especialistas que poseen experiencias en temas relacionados con el presente trabajo. Se realizarán preguntas a los expertos con la ayuda de cuestionarios sucesivos, a fin de poner de manifiesto convergencias de opiniones y deducir eventuales consensos.

Para llevar a cabo la aplicación de este método se realizó la selección un grupo expertos del CEIGE y de otros centros de la universidad teniendo en cuenta su disposición a participar en la encuesta y apreciación en el tema investigado. Luego de hacer la selección, se aplica la metodología usada en el método Delphi la cual resulta completa y sencilla para determinar la competencia de los expertos. Para realizar las evaluaciones se realizó una encuesta (ver **Anexo 1**) la cual fue resuelta por los posibles expertos.

La metodología establece [25] que lo primero que el posible experto debe hacer es marcar con una x en una escala creciente de 0 a 10, ver la tabla en el **Anexo 2**, el valor que se corresponde con el grado de conocimiento o información que tienen sobre el tema de estudio. Con estos datos se calcula el coeficiente de conocimiento K_c para cada uno de los expertos donde:

K_c = El valor que marcó el experto en la tabla por (0.1).

Lo segundo es pedirles a los expertos su autoevaluación de sus niveles de argumentación o fundamentación sobre el tema de estudio, para esto debe llenar con una x en la tabla de coeficiente de argumentación que se muestra en el **Anexo 3**. A partir de la autoevaluación realizada y el patrón de valores para el coeficiente de argumentación, que se muestra en el **Anexo 4** se calcula el coeficiente de argumentación K_a . Dónde:

K_a = Σ de los valores que se obtienen de sustituir las cruces del **Anexo 3** por los valores correspondientes en su posición de la tabla de valores para el coeficiente de argumentación.

Luego de calculado los valores de K_c y K_a , se calcula el coeficiente de competencia K mediante la fórmula:

$$K = \frac{K_c + K_a}{2}$$

Una vez calculado el valor de K



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

Si $0,8 < K < 1$ coeficiente de competencia alto.

Si $0,5 < K < 0,8$ coeficiente de competencia medio.

Si $K < 0,5$ coeficiente de competencia bajo.

Para obtener mejores resultados en la aplicación del Delphi es conveniente utilizar los expertos cuyo coeficiente de competencia sea alto o en algunos casos medio.

Luego de realizada la encuesta de autoevaluación a los posibles expertos, se seleccionaron 7 para formar parte del grupo de validación de la propuesta, eligiéndolos de acuerdo los resultados obtenidos, los cuales arrojaron que presentan un coeficiente de competencia alto o medio. Los resultados del análisis del coeficiente de competencia se muestran en el **Anexo 5**.

En la **Figura 9** que se muestra a continuación se representa el resultado obtenido del grupo resultante para la validación de la propuesta, donde los expertos con coeficiente alto representan un 57% del total y un 43% los que poseen coeficiente medio.

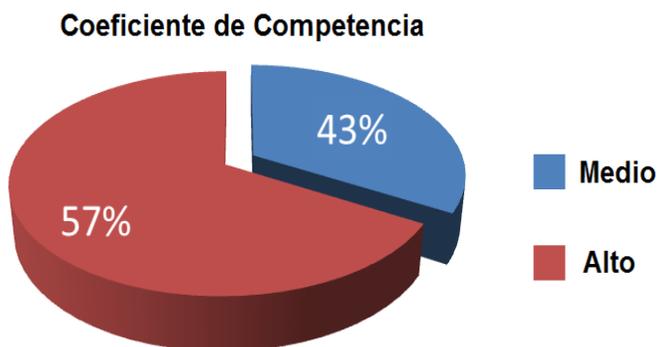


Figura 9. Porcentaje del coeficiente de competencia de los expertos.

3.2.1. Elaboración del cuestionario para la validación de la propuesta

Para validar la solución propuesta se utilizó un cuestionario conformado por 13 preguntas, el cual se muestra en el **Anexo 7** orientadas a evaluar aspectos críticos de la solución, entre ellos la variable a solucionar con el trabajo (el aumento del rendimiento).

El cuestionario fue realizado por los 7 expertos seleccionados pidiéndosele que se evaluaran en las categorías ordinales de Muy Adecuado (5), Bastante Adecuado (4), Adecuado (3), Poco Adecuado (2), y No Adecuado (1). Con las respuestas dadas por los expertos se podrá obtener la concordancia entre ellos y la aceptación y validez de la propuesta.



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

3.2.2. Establecimiento de la concordancia entre los expertos

Cuando se tiene datos de tipo ordinal el coeficiente de Kendall, el cual toma en consideración el orden, es usualmente un instrumento estadístico muy apropiado para indicar el grado de asociación de las evaluaciones ordinales hechas por evaluadores múltiples cuando se evalúa la misma muestra. Los valores del coeficiente deben oscilar entre 0 y 1. Entre mayor sea el valor Kendall, más fuerte será la asociación [28]. Un coeficiente Kendall significativo implica que los evaluadores están aplicando esencialmente el mismo estándar cuando evalúan la muestra.

Después de haber realizado el cuestionario, y con las respuestas de los expertos, se verificará si existe concordancia entre ellos mediante el coeficiente de Kendall, el cual se calcula mediante la siguiente fórmula [25]:

$$W = \frac{S}{12k^2(N^3+N) - kT_i} \quad (1)$$

Donde:

w: Coeficiente de concordancia.

S es la suma de los cuadrados de las desviaciones observadas de la media de S_j (rangos); y se calcula mediante la expresión:

$$S = \sum_{j=1}^n (S_j - \bar{S})^2 \quad (2) \text{ Donde } \bar{S} = \frac{\sum_{j=1}^n S_j}{N} \quad (3)$$

\bar{S} : Suma de los rangos dividido entre la cantidad de preguntas.

N: Cantidad de preguntas.

k: Cantidad de expertos.

T_i : Es el factor de corrección cuando existen rangos iguales, el cual es 0 cuando las evaluaciones realizadas por los expertos i son diferentes, y se obtiene aplicando la siguiente fórmula.

$$T_i = \frac{\sum_{i=1}^l (t^3 - t)}{12} \quad (4) \text{ Siendo:}$$

l: Número de grupos con rangos iguales para el experto i .



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

t: Número de observaciones dentro de cada uno de los grupos para el experto i.

Una vez calculado el coeficiente de Kendall (w) es necesario conocer su probabilidad de significancia, para lo cual se debe transformar éste valor en Chi cuadrado de Pearson mediante la fórmula siguiente:

$$X^2_{cal} = k(N - 1)w \quad (5)$$

Para buscar el Chi cuadrado tabulado en la tabla de distribución, mostrada en el **Anexo 7** se calcula el grado de libertad (gl) donde: $gl = N - 1$

$X^2_{tab} = X^2_{\alpha, gl}$ α es el nivel de significación utilizado para calcular el nivel de confianza. El nivel de confianza es igual a $100(1 - \alpha) \%$, es decir, un α de 0,05 indica un nivel de confianza de 95%.

Una vez obtenido los valores se comparan X^2_{tab} y X^2_{cal} , si $X^2_{tab} > X^2_{cal}$ entonces existe concordancia de criterios entre los expertos.

Para calcular el rango (Sj), cuando se tienen varios expertos, se le asignan valores a las categorías, Muy Adecuado (5), Bastante Adecuado (4), Adecuado (3), Poco Adecuado (2), No Adecuado (1). Dependiendo de la evaluación que el experto de a cada pregunta, será el valor asociado que se pondrá en la tabla, a continuación se muestran los resultados.

Tabla II. Cálculo del Coeficiente de Kendall.

Expertos								Sj
Preguntas	1	2	3	4	5	6	7	
1	4	3	5	4	3	4	5	28
2	5	4	5	5	4	5	4	32
3	5	4	5	5	4	4	5	32
4	4	5	5	5	5	5	5	34
5	4	5	5	4	5	5	4	32
6	5	4	4	3	3	5	4	28
7	4	4	4	5	3	5	4	29
8	4	4	4	5	4	5	4	30
9	5	3	5	3	4	5	1	26
10	5	3	5	4	4	5	3	29



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

11	5	4	5	4	4	4	4	30
12	4	5	5	5	4	4	5	32
13	4	4	5	4	3	4	5	29

Calculo del grado de concordancia entre los expertos w

Sustituyendo los valores en la ecuación 3.

$$\bar{S} = \frac{28+32+32+34+32+28+29+30+26+29+30+32+29}{13} = 30.7$$

Sustituyendo y calculando en la ecuación 2.

$$S = 58,92$$

Debido a que los expertos concuerdan en muchas de las evaluaciones realizadas es necesario calcular el factor de corrección T_i , para lo cual se sustituye en la ecuación 4.

$$T_i = 228,66$$

Una vez calculadas las variables necesarias se procede a calcular el coeficiente de Kendall, sustituyendo en la fórmula 1.

$$w = 58,9 / (12 \cdot 7^2 (13^3 + 13) - 7 \cdot 228,66) = 0,00009$$

$$X^2_{cal} = k(N - 1)w$$

Después de calculada la concordancia entre los expertos se procede a calcular el valor de Chi cuadrado sustituyendo en la ecuación 5.

$X^2_{cal} = 7(13-1) \cdot 0,00009 = 0,00756$, luego se compara el Chi cuadrado calculado con el Chi cuadrado de la tabla para un $\alpha = 0,05$ y $gl = 11$; $X^2_{cal} = 0,00693$ y $X^2_{tab} = X^2_{(0,05; 11)} = 19,675$ como $X^2_{cal} < X^2_{tab}$, se puede decir que hay concordancia entre los expertos.

3.2.3. Desarrollo práctico y explotación de los resultados

A partir de las respuestas dada por cada experto se analizarán los resultados, los cuales se recogen en la siguiente tabla.



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

Tabla III. Frecuencia Absoluta.

Tabla de frecuencia absoluta						
Preguntas	MA	BA	A	PA	NA	Total
1	2	3	2	0	0	7
2	4	3	0	0	0	7
3	4	3	0	0	0	7
4	6	1	0	0	0	7
5	4	3	0	0	0	7
6	2	3	2	0	0	7
7	2	4	1	0	0	7
8	5	2	0	0	0	7
9	3	1	2	0	1	7
10	3	2	1	0	0	7
11	2	5	0	0	0	7
12	2	4	1	0	0	7
13	2	4	1	0	0	7

Los resultados recogidos en la tabla anterior quedan representados en la siguiente figura mostrando el grado de aceptación de cada uno de los aspectos y preguntas evaluadas por los expertos.

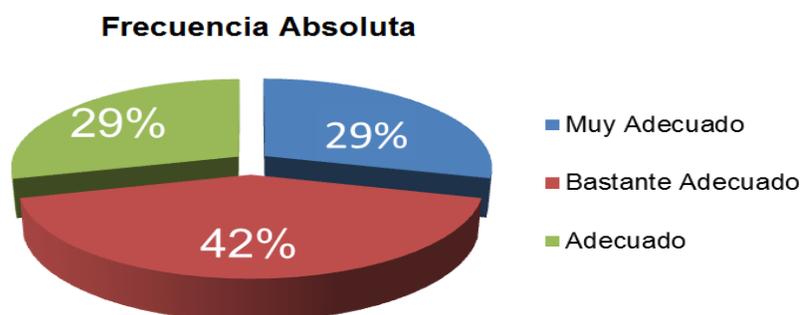


Figura 10. Porcentaje de la frecuencia absoluta.

Para obtener resultados satisfactorios, una vez calculados los datos, se realizan los siguientes pasos:

Paso 1: Se construye una tabla de frecuencia absoluta acumulada donde cada número en la fila se obtiene sumándole el anterior, excepto el primero que se mantiene igual.



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

Tabla IV. Frecuencia absoluta acumulada.

Tabla de frecuencia absoluta acumulada					
Preguntas	MA	BA	A	PA	NA
1	2	5	7	7	7
2	4	7	7	7	7
3	4	7	7	7	7
4	6	7	7	7	7
5	4	7	7	7	7
6	2	5	7	7	7
7	2	6	7	7	7
8	5	7	7	7	7
9	3	4	6	6	7
10	3	5	7	7	7
11	2	7	7	7	7
12	2	6	7	7	7
13	2	6	7	7	7

Paso 2: Se copia la tabla anterior y se borran los dígitos. En esta nueva tabla se construye la tabla de frecuencia relativa acumulada, la cual se logra dividiendo por 7, que es el número total de expertos, cada uno de los números de la tabla anterior. Dado que con cuatro puntos se obtienen cinco intervalos y en la tabla hay cinco categorías, se elimina una columna, pues no es necesaria.

Tabla V. Frecuencia relativa acumulada.

Preguntas	MA	BA	A	PA
1	0,29	0,71	1	1
2	0,57	1	1	1
3	0,57	1	1	1
4	0,86	1	1	1
5	0,57	1	1	1
6	0,29	0,71	1	1
7	0,29	0,86	1	1



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

8	0,71	1	1	1
9	0,43	0,57	0,86	0,86
10	0,43	0,71	1	1
11	0,29	1	1	1
12	0,29	0,86	1	1
13	0,29	0,86	1	1

Paso 3: Se buscan las imágenes de los elementos de la tabla anterior, y la nueva tabla que se obtiene cuenta con cuatro elementos nuevos.

- ✓ **Suma:** Es la suma de las columnas y de las filas.
- ✓ **P:** Promedio de las filas
- ✓ **N:** Promedio de los valores de los puntos de cortes, que no son más que los promedios de cada categoría (columna), por lo que se obtiene el promedio de promedio, llamado valor límite.
- ✓ **N-P:** Valor promedio que otorgan los expertos consultados a cada pregunta propuesta, y se utiliza para situar las categorías en el rango de adecuación.

Tabla VI. Puntos de corte.

Puntos de corte								N=2,23
Preguntas	MA	BA	A	PA	Suma	P	N-P	Grado de Adecuación
1	-1,08	0,55	3,5	3,5	6,47	1,62	0,61	Bastante Adecuado
2	0,18	3,5	3,5	3,5	10,68	2,46	-0,23	Muy Adecuado
3	0,18	3,5	3,5	3,5	10,68	2,46	-0,23	Muy Adecuado
4	1,08	3,5	3,5	3,5	11,58	2,90	-0,67	Muy Adecuado
5	0,18	3,5	3,5	3,5	10,68	2,46	-0,23	Muy Adecuado
6	-1,08	0,55	3,5	3,5	6,47	1,62	1,66	Bastante Adecuado
7	-1,08	1,08	3,5	3,5	7	1,75	0,56	Bastante Adecuado
8	0,55	3,5	3,5	3,5	11,05	2,90	-0,59	Muy Adecuado
9	-0,18	0,18	1,08	1,08	4,22	1,05	1,26	Bastante Adecuado



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

10	-0,18	0,55	3,5	3,5	7,37	1,84	0,47	Bastante Adecuado
11	-1,08	3,5	3,5	3,5	9,42	2,36	-0,05	Muy Adecuado
12	-1,08	1,08	3,5	3,5	7	1,75	0,56	Bastante Adecuado
13	-1,08	1,08	3,5	3,5	7	1,75	0,56	Bastante Adecuado
Suma	3,75	25,98	43,08	43,08				
Puntos de corte	0,29	2	3,31	3,31				

Cuando la probabilidad acumulada sea igual a 1, le corresponde el valor estándar inverso de 3,5, y para trabajar con ambos extremos iguales, si la probabilidad acumulada es igual a 0, entonces el valor estándar inverso se asume igual a -3,5 [25].

El promedio de cada columna da como resultado los puntos de corte, los cuales se utilizan para determinar el grado de adecuación de cada criterio según la opinión de los expertos consultados.

Tabla VII. Rangos de las categorías obtenidos a partir de los puntos de corte.

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
Menor que 0,29	0,29 a 2	2 a 3,31	Mayor que 3,31	Mayor que 3,31

De acuerdo a los resultados mostrados en la **Tabla VI** se puede resumir que la solución propuesta resulta muy adecuada, debido a que de 13 aspectos a evaluar 7 fueron valorados de bastante adecuado y 6 de muy adecuado, por lo que se concluye que con la obtención de la versión 2 de la capa de acceso a datos del MT se contribuye a aumentar su rendimiento y el de las soluciones desarrolladas sobre él.

3.3. Pruebas de Rendimiento

Una vez obtenida la versión 2 de la capa de acceso a datos del MT Sauxe se procede a realizar pruebas de rendimiento con el objetivo de verificar si se logró obtener un mayor rendimiento.



CAPÍTULO III: VALIDACIÓN DE LA MIGRACIÓN

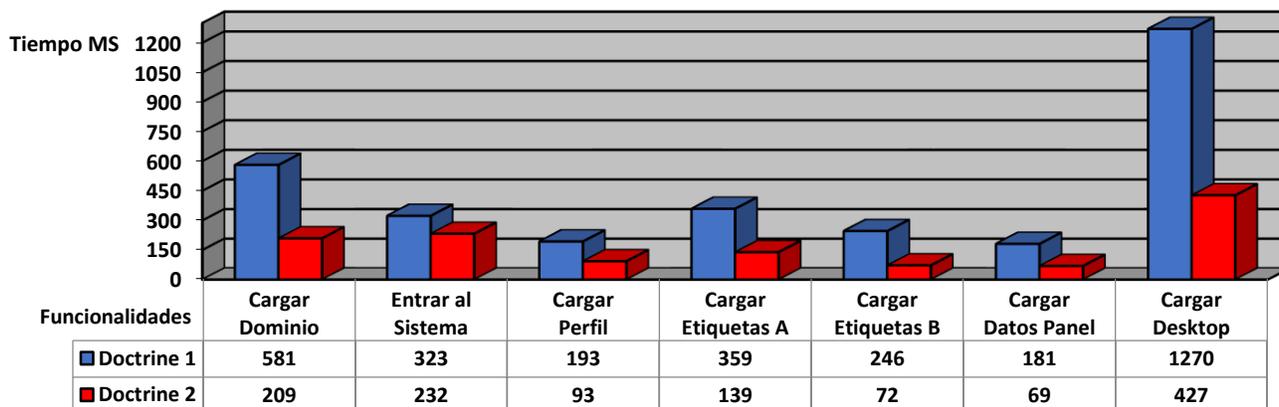


Figura 11. Gráfica de comparación de rendimiento en Doctrine.

En la **Figura 11** se muestra un gráfico con los valores obtenidos de las pruebas de rendimiento realizadas a siete funcionalidades utilizando las versiones 1 y 2 de la capa de acceso a datos MT Sauxe. Se puede observar que el rendimiento obtenido con el uso de la versión 2 de la capa de acceso a datos es muy superior al rendimiento de la versión anterior quedando demostrado que con la obtención de la versión 2 de la capa de acceso a datos en el MT Sauxe, se aumenta el rendimiento de las soluciones desarrolladas sobre él.

3.4. Conclusiones Parciales

La validación de la propuesta de solución definida, empleando el método Delphi arrojó como resultado que:

- ✓ Se seleccionaron 7 expertos de los cuáles uno tiene el grado científico de Doctor y dos tienen el grado de Máster.
- ✓ De los 13 aspectos evaluados 6 fueron categorizados por los expertos como muy adecuado y 7 de bastante adecuado.
- ✓ Se obtuvieron resultados satisfactorios ya que según la opinión de los expertos encuestados la solución propuesta fue evaluada de muy adecuada, quedando de esta manera demostrada su correcta realización.

Una vez terminada la versión 2 de la capa de acceso a datos del MT Sauxe y realizar pruebas sobre 7 funcionalidades se obtuvo una reducción de 1.912 segundos lo que representa un aumento de aproximadamente el 165% del rendimiento.



CONCLUSIONES GENERALES

El estudio realizado sobre los principales marcos de persistencia de datos arrojó como resultado que estos presentaban características comunes y que Doctrine 2 presenta las principales potencialidades de los marcos estudiados.

Se realizó la migración de la capa de acceso a datos del MT Sauxe permitiendo establecer una guía para facilitar dicha migración en las soluciones desarrolladas sobre él, garantizando la organización de la misma.

Mediante la validación de la migración definida empleando el método Delphi, se obtuvieron resultados satisfactorios, ya que según la opinión de los expertos encuestados fue evaluado de bastante adecuado y muy adecuado, quedando de esta manera demostrada su correcta definición.

Una vez terminada la versión 2 de la capa de acceso a datos del MT Sauxe y realizar pruebas sobre 7 funcionalidades se obtuvo una reducción de 1.912 segundos lo que representa un aumento de aproximadamente el 165% del rendimiento.



RECOMENDACIONES

Actualizar el Mapeador de Doctrine para facilitar la migración de la capa de acceso a datos del MT Sauxe y de las soluciones desarrolladas sobre él.

Realizar la migración a todas las soluciones desarrolladas sobre el MT Sauxe para aumentar el rendimiento de las mismas.



REFERENCIAS BIBLIOGRÁFICAS

1. Minetto, E.L., *Frameworks para Desenvolvimento em PHP*. 2007, Livraria Martin Fontes: Sao Paulo.
2. Baryolo, O.G., *SOLUCIÓN INFORMÁTICA DE AUTORIZACIÓN EN ENTORNOS MULTIENTIDAD Y MULTISISTEMA*. 2001, Universidad de las Ciencias Informáticas: La Habana.
3. Arquitectura, D.d., *Pruebas de rendimiento sobre Cedrux*. 2012: Universidad de las Ciencias Informáticas.
4. McCafferty, B. *Best Practices with ASP.NET, 1.2nd Ed*. 2008 [cited 2008 11 Jun 2008]; Available from: <http://www.codeproject.com/Articles/13390/NHibernate-Best-Practices-with-ASP-NET-1-2nd-Ed#SUMMARY>.
5. Cadavid, A.N. *Sistema de investigación de la Universidad Icesi*. 2008 [cited 2013 26 de Enero de 2013]; Available from: <http://www.icesi.edu.co/investigaciones/proyecto.do?id=38>
6. teamCommunity, J. *Hibernate*. 2013 [cited 2012 24 de Noviembre de 2012]; Available from: <http://www.hibernate.org/about/why-hibernate>.
7. Community, H. *Hibernate Community*. 2012 [cited 2012 24 de Noviembre de 2012]; Available from: <http://docs.jboss.org/hibernate/orm/4.2/devguide/en-US/html/ch02.html#d5e618>.
8. Propel, S., easy object persistence. *Propel, Smart, easy object persistence*. 2013 [cited 2013 26 de Enero de 2013]; Available from: <http://propelorm.org/>.
9. Propel?, C.u. *¿Cómo utilizar Propel?* 2011; Available from: http://symfony.com/legacy/doc/cookbook/1_2/es/propel_13.
10. García, L. *Scribd, Django ORM*. 2011 03 de Marzo de 2011 [cited 2013 27 de Enero de 2013]; Available from: <http://es.scribd.com/doc/63901225/Django-ORM>.
11. Adrian Holovaty, J.K.-M. *The Django Book*. 2012 [cited 2012 Noviembre 2012]; Available from: <http://www.djangobook.com/en/2.0/chapter02.html>.
12. SubSonic. *SubSonic*. 2013 [cited 2013 27 de Enero de 2013]; Available from: <http://subsonicproject.com/>.
13. Subsonic. *Subsonic*. 2012; Available from: <http://subsonicproject.com/docs/Comparisons/#Performance>.



REFERENCIAS BIBLIOGRÁFICAS

14. Documentation, D.O. *Doctrine 2 ORM Documentation*. 2012 04 de Diciembre de 2012 [cited 2012 04 de Diciembre de 2012]; Documentación del ORM Doctrine 2]. Available from: <http://docs.doctrine-project.org/es/latest/>.
15. México, U.N.A.d. *Universidad Nacional Autónoma de México*. 2012 28 de enero 2013 [cited 2012 04-12-2012]; Administración para la toma de decisiones]. Available from: <http://fcasua.contad.unam.mx/apuntes/interiores/docs/2005/administracion/5/1553.pdf>.
16. Ben Collins-Sussman, B.W.F.y.C.M.P., *Control de Versiones con Subversion*. 2012.
17. RapidSVN. *RapidSVN*. 2012 04 de Diciembre de 2012; Available from: <http://www.rapidsvn.org/>.
18. Martínez, R., *Portal de Postgres en Español*. 2012.
19. NetBeans, *NetBeans quick start*. 2012.
20. Foundation, T.A.S. *The Apache Software Foundation*. 2012 [cited 2012 20 de Diciembre de 2012]; Available from: <http://www.apache.org>
21. Inc, Z.T., *Manual Zend Framework Español*.
22. Inc, Z.T. *Manual Zend Framework Español*. 2005 01 de Diciembre de 2012 [cited 2012 01 de Diciembre de 2012]; Available from: <http://manual.zfdes.com/es/introduction.overview.html>.
23. Murray Turoff, L.H.A. *The Delphi Method. Techniques and Applications*. 2013 2002 [cited 2013 25 de Febrero de 2013]; Available from: <http://is.njit.edu/pubs/delphibook/delphibook.pdf>.
24. Eneko, A. *METODO DELPHI - Prospectiva*. 2001 [cited 2013 30 de abril de 2013]; Available from: http://www.prospectiva.eu/zaharra/03_Delphi_ESTE.pdf.
25. Martínez, M.Á.M., *Métodos Y Técnicas De Investigación Para Las Ciencias Sociales*. Revista, 2013. 4(8).
26. Pupo, Y.C. (2010) *Libro de Ayuda del Marco de Trabajo Sauxe*. Libro de Ayuda del Marco de Trabajo Sauxe.
27. JMeter, A. *Apache JMeter*. 2013 [cited 2013 12 de Abril de 2013]; Available from: <http://jmeter.apache.org/>.
28. Alvarado, F.P. *Análisis de concordancia de atributos*. 2008 [cited 2013 03 de Mayo de 2013]; Available from: http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial_tecnologica/Revista_Tecnologia_Marcha/pdf/tecnologia_marcha_21-4/cap%203.pdf.
29. Sampieri, R.H., et al., *Metodología de la investigación*. 1998: McGraw-Hill México.



REFERENCIAS BIBLIOGRÁFICAS

30. Díaz, J., et al. *Usando Jmeter para pruebas de rendimiento*. in XIV Congreso Argentino de Ciencias de la Computación. 2008.
31. Pablos Ruiz, E., *Migración de una base de datos Oracle 9i a Oracle 11g*. 2012.
32. Hurtado de Mendoza, S., *Criterio de expertos. Su procesamiento a través del método Delphi*. 2009.
33. Barnes, D.J. and M. Kölling, *Planificación y automatización de pruebas de software*. Índice.
34. Cardona Velásquez, C.d.J., *Propuesta metodológica para la realización de pruebas de software en ambientes productivos*. 2009, Universidad Nacional de Colombia.



Anexo 1. Encuesta para determinar el coeficiente de competencia

Compañero (a):

Usted ha sido seleccionado para colaborar con la investigación realizada a la valoración de la definición de la *Migración de la Capa de Acceso a Datos del MT Sauxe*. Para ello necesitamos conocer el grado de dominio que usted posee sobre Doctrine y el funcionamiento en general del MT Sauxe.

Nombre: _____

Rol que desempeña: _____

Años de experiencia: _____

Especialidad: _____

Categoría docente: _____

Categoría científica: _____

Se necesita, como parte del método de investigación de consulta de expertos, determinar su coeficiente de competencia en el tema, por lo que se le pide que responda con la mayor objetividad posible.

1. Marque con una (x) en la tabla siguiente, el valor que corresponda con el grado de conocimiento que usted posee sobre el tema. Tenga en cuenta que la tabla que se le presenta cuenta con una escala creciente de 1 a 10, donde la máxima competencia se corresponde con el 10.

1	2	3	4	5	6	7	8	9	10

2. Se le solicita que realice una autoevaluación de los niveles de influencia que ha tenido en sus conocimientos sobre el tema antes referido cada una de las fuentes de argumentación que se le presentan. Para esto marque con una (x) Alto, Medio o Bajo, según corresponda.

Fuentes de argumentación	Grado de influencia		
	Bajo	Medio	Alto
Análisis teóricos realizados por Ud.			
Experiencia obtenida con el uso de Doctrine.			
Uso de Doctrine 1.x en cualquier contexto.			



Uso de Doctrine 2.x en cualquier contexto.			
Su propio conocimiento del estado del problema.			
Su intuición.			

Anexo 2.Tabla para marcar el coeficiente de conocimiento

1	2	3	4	5	6	7	8	9	10

Anexo 3.Tabla para marcar el coeficiente de argumentación

Fuentes de argumentación	Grado de influencia		
	Bajo	Medio	Alto
Análisis teóricos realizados por Ud.			
Experiencia obtenida con el uso de Doctrine.			
Uso de Doctrine 1.x en cualquier contexto.			
Uso de Doctrine 2.x en cualquier contexto.			
Su propio conocimiento del estado del problema.			
Su intuición.			

Anexo 4. Patrón de valores para el Coeficiente de Argumentación

Fuentes de argumentación	Grado de influencia		
	Bajo	Medio	Alto
Análisis teóricos realizados por Ud.	0,13	0,22	0,3
Su experiencia obtenida.	0,22	0,32	0,4
Estudio de trabajos de autores nacionales.	0,025	0,04	0,05
Estudio de trabajos de autores extranjeros.	0,025	0,04	0,05
Su propio conocimiento del estado del problema	0,05	0,09	0,1



Su intuición.	0,05	0,09	0,1
Total	0.5	0.8	1

Anexo 5. Resultado del coeficiente de los expertos

Experto	Kc	Ka	K	Competencia
1	0,8	0,9	0,85	Alto
2	0,8	0,86	0,83	Alto
3	0,5	0,78	0,64	Medio
4	0,7	0,79	0,74	Medio
5	0,4	0,69	0,55	Medio
6	0,9	0,81	0,85	Alto
7	0,9	0,92	0,91	Alto

Anexo 6. Encuesta para la validación del trabajo

Encuesta para validar la Migración de la Capa de Acceso a Datos del MT Sauxe.

Usted ha sido seleccionado para validar el trabajo de diploma “*Migración de la Capa de Acceso a Datos del MT Sauxe*”, por sus conocimientos y experiencia en el tema de investigación.

Usted debe evaluar las siguientes preguntas asignando un valor cuantitativo comprendido del 0 al 5. Para ello marque con una x según considere.

No.	Preguntas	0	1	2	3	4	5
1	¿Considera usted que la propuesta tiene un valor científico apropiado?						
2	¿Considera usted que tiene lógica la secuencia de pasos a seguir para la migración?						
3	¿Considera usted que tiene lógica la sucesión de las tareas a realizar?						
4	¿Considera usted que la migración contribuye a mejorar el rendimiento de las soluciones desarrolladas sobre Sauxe?						



5	¿Considera usted que la solución propuesta contribuye a la migración de la capa de acceso a datos de Sauxe y de cada uno de sus componentes?						
6	¿Considera usted que la solución propuesta cuenta con las actividades necesarias para realizar correctamente la migración de la Capa de Acceso a Datos del MT Sauxe?						
7	¿Considera usted que los roles definidos por los proyectos son suficientes para llevar a cabo de manera efectiva la migración?						
8	¿Considera usted que las responsabilidades para los roles definidos por los proyectos son suficientes para llevar a cabo su trabajo de manera efectiva en la migración?						
9	¿Considera usted que se proporciona la información necesaria para realizar correctamente las pruebas de rendimiento en los sistemas y aplicaciones desarrollados en el CEIGE?						
10	¿Considera usted que los parámetros de configuración propuestos cumplen algún objetivo específico para la migración?						
11	¿Considera usted que la taxonomía de carpetas propuesta permite la migración de la Capa de Acceso a Datos del MT Sauxe?						
12	¿Considera usted que las clases y funcionalidades propuestas permiten la integración de Doctrine 2 con el MT Sauxe?						
13	¿Considera usted que se explica de manera clara en qué lugar debe ir cada clase para la nueva estructura de Doctrine 2?						



Anexo 7. Tablas de distribución Chi cuadrada o X².

		gl -->									
1/alfa	alfa	1	2	3	4	5	6	7	8	9	10
10	0.1	2.7055	4.6052	6.2514	7.7794	9.2363	10.645	12.017	13.362	14.684	15.987
20	0.05	3.8415	5.9915	7.8147	9.4877	11.07	12.592	14.067	15.507	16.919	18.307
40	0.025	5.0239	7.3778	9.3484	11.143	12.832	14.449	16.013	17.535	19.023	20.483
50	0.02	5.4119	7.8241	9.8374	11.668	13.388	15.033	16.622	18.168	19.679	21.161
100	0.01	6.6349	9.2104	11.345	13.277	15.086	16.812	18.475	20.09	21.666	23.209
200	0.005	7.8794	10.597	12.838	14.86	16.75	18.548	20.278	21.955	23.589	25.188
1000	0.001	10.827	13.815	16.266	18.466	20.515	22.457	24.321	26.124	27.877	29.588
2000	0.0005	12.115	15.201	17.731	19.998	22.106	24.102	26.018	27.867	29.667	31.419
10000	0.0001	15.134	18.425	21.104	23.506	25.751	27.853	29.881	31.827	33.725	35.557
20000	0.00005	16.458	19.802	22.544	25.002	27.282	29.444	31.52	33.493	35.44	37.32
100000	0.00001	19.504	23.079	25.884	28.415	30.799	33.089	35.204	37.286	39.278	41.265
200000	5E-06	20.847	24.366	27.312	29.939	32.462	34.661	36.725	38.916	41.088	43.023
1000000	1E-06	24.366	28.229	30.482	33.379	35.612	37.965	40.138	42.31	45.162	46.766
2000000	5E-07	24.366	28.229	33.379	34.344	37.241	40.138	42.31	43.94	46.078	48.828

		gl -->									
1/alfa	alfa	11	12	13	14	15	16	17	18	19	20
10	0.1	17.275	18.549	19.812	21.064	22.307	23.542	24.769	25.989	27.204	28.412
20	0.05	19.675	21.026	22.362	23.685	24.996	26.296	27.587	28.869	30.144	31.41
40	0.025	21.92	23.337	24.736	26.119	27.488	28.845	30.191	31.526	32.852	34.17
50	0.02	22.618	24.054	25.471	26.873	28.259	29.633	30.995	32.346	33.687	35.02
100	0.01	24.725	26.217	27.688	29.141	30.578	32	33.409	34.805	36.191	37.566
200	0.005	26.757	28.3	29.819	31.319	32.801	34.267	35.718	37.156	38.582	39.997
1000	0.001	31.264	32.909	34.527	36.124	37.698	39.252	40.791	42.312	43.819	45.314
2000	0.0005	33.138	34.821	36.477	38.109	39.717	41.308	42.881	44.434	45.974	47.498
10000	0.0001	37.365	39.131	40.873	42.575	44.26	45.926	47.559	49.185	50.787	52.383
20000	0.00005	39.149	40.959	42.718	44.444	46.154	47.866	49.528	51.164	52.803	54.442
100000	0.00001	43.252	45.04	46.895	48.673	50.438	52.29	53.949	55.639	57.296	58.976
200000	5E-06	44.872	46.766	48.828	50.393	52.29	54.121	55.811	57.468	59.157	60.987
1000000	1E-06	48.828	51.117	52.834	54.121	56.354	57.983	59.7	61.953	63.22	64.85
2000000	5E-07	51.117	52.834	54.121	57.983	57.983	59.7	60.987	64.85	64.85	66.137