



Universidad de las Ciencias Informáticas
Facultad 1

**Trabajo de Diplomas para optar por el título de
INGENIERO EN CIENCIAS INFORMÁTICAS**

Título: Módulo de premios y publicaciones del sistema de Ciencia,
Tecnología e Innovación en la Universidad de las Ciencias
Informáticas

Autores: Ibrael Espinosa Moscoso

Suley Martínez Pérez

Tutores: Yunaysy Ortiz Batista

Yeslidier López Reinoso

Ciudad de la Habana, Junio 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Módulo de premios y publicaciones del sistema de Ciencia, Tecnología e Innovación en la Universidad de las Ciencias Informáticas y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 23 días del mes de junio del año 2009.

Ibrael Espinosa Moscoso

Suley Martínez Pérez

Firma del Autor

Firma del Autor

Ing. Yeslidier López Reinoso.

Ing. Yunaysy Ortiz Batista

Firma del Tutor

Firma del Tutor

Resumen

La gestión de la información relacionada con los premios curriculares y las publicaciones científicas dentro del sistema de CTI es un proceso de vital importancia para el adecuado funcionamiento de la Universidad. En la actualidad, el procedimiento a través del cual se gestiona esta información es lento, inexacto y primitivo en relación al contexto en que se desarrolla la UCI. En esta investigación científica se presenta la concepción de un sistema informático, que se propone como solución del problema planteado, teniendo en cuenta los procesos de premios y publicaciones dentro del sistema de CTI.

Para desarrollar este sistema se eligió *Symfony* como marco de trabajo, *Doctrine* como capa de acceso a datos para el Mapeo Objeto-Relacional y *Rational Unified Process* como metodología de desarrollo.

La propuesta de solución consiste en una aplicación que, integrada en un sistema único y como parte del grupo de aplicaciones del Sistema de Gestión Universitaria, tramite y controle toda la información relacionada con los procesos de premios curriculares y publicaciones científicas dentro del sistema de CTI. Con la puesta en marcha de este sistema, se podrá gestionar de manera eficiente toda la información generada producto de la actividad científica que se realiza en la Universidad. Sus potencialidades y flexibilidad permiten extender esta metodología a otros centros de educación superior y de investigación en el país.

Índice de contenido

Introducción	- 1 -
Capítulo 1: Fundamentación teórica	- 5 -
1.1. Introducción	- 5 -
1.2. Investigación científica	- 5 -
1.3. Publicaciones científicas	- 5 -
1.4. Premios curriculares	- 6 -
1.5. Sistemas de Control de Publicaciones	- 7 -
1.5.1 Internacionales	- 7 -
1.5.2 Nacionales	- 9 -
1.5.3 Sistemas usados en la UCI	- 10 -
1.6. Control de premios curriculares	- 10 -
1.7. Metodología de desarrollo	- 11 -
1.7.1 RUP	- 11 -
1.8. Arquitectura del software	- 12 -
1.9. Arquitectura cliente-servidor	- 12 -
1.9.1 Modelo-Vista-Controlador (MVC)	- 13 -
1.10. Tecnologías y lenguajes a utilizar	- 14 -
1.10.1 Tecnologías y lenguajes del lado del cliente	- 14 -
1.10.2 Tecnologías y lenguajes del lado del servidor	- 16 -
1.11. Herramientas	- 19 -
1.12. Conclusiones del capítulo	- 21 -
Capítulo 2: Propuesta de solución	- 22 -
2.1 Introducción	- 22 -
2.2 Actores, trabajadores y casos de uso	- 22 -

2.3	Procesos del negocio	- 22 -
2.4	Modelado del negocio	- 23 -
2.4.1	Reglas del negocio.....	- 23 -
2.4.2	Diagrama de casos de usos del negocio	- 24 -
2.4.3	Descripción de actores y trabajadores del negocio	- 24 -
2.4.4	Descripción de casos de usos del negocio.....	- 25 -
2.5	Modelado del Sistema.....	- 26 -
2.5.1	Requerimientos funcionales del sistema.....	- 26 -
2.5.2	Requerimientos no funcionales del sistema	- 28 -
2.5.3	Actores del sistema:.....	- 29 -
2.5.4	Descripción de los casos de uso del sistema.....	- 30 -
2.6	Conclusiones.....	- 36 -
Capítulo 3: Diseño de la solución propuesta.		- 37 -
3.1	Introducción.....	- 37 -
3.2	Modelo de Diseño	- 37 -
3.2.1	Estructura de Symfony	- 37 -
3.2.2	Diagrama de clases del diseño.....	- 39 -
3.2.3	Diagramas de iteración: diagrama de secuencia.	- 39 -
3.3	Patrones de diseño.....	- 41 -
3.3.1	Patrones GRASP (Patrones generales de software para asignar responsabilidades)-	41 -
3.3.2	Patrones GOP	- 42 -
3.4	Diagrama de clases persistentes	- 42 -
3.4.1	Descripción de las tablas.....	- 44 -
3.5	Conclusiones.....	- 55 -
Capítulo 4: Construcción y pruebas de la solución.....		- 56 -

Módulo de gestión de Publicaciones y Premios

4.1	Introducción.....	- 56 -
4.2	Diagrama de componentes.....	- 56 -
4.3	Diagrama de despliegue	- 57 -
4.4	Pruebas.....	- 58 -
4.5	Conclusiones.....	- 62 -
	Conclusiones	- 63 -
	Recomendaciones	- 65 -
	Bibliografía referenciada.....	- 66 -
	Bibliografía consultada	- 67 -
	Glosario de términos	- 69 -

Índice de tabla

Tabla 1 Descripción de actores y trabajadores del negocio	- 24 -
Tabla 2 Descripción del caso de uso Registrar premios	- 25 -
Tabla 3 Descripción del caso de uso Registrar publicaciones.....	- 26 -
Tabla 4 Descripción de los actores del sistema	- 30 -
Tabla 5 Descripción del CU Visualizar revista.....	- 30 -
Tabla 6 Descripción del CU Gestionar revista.....	- 31 -
Tabla 7 Descripción del CU Visualizar repositorios	- 31 -
Tabla 8 Descripción del CU Gestionar repositorios	- 32 -
Tabla 9 Descripción del CU Gestionar memorias de eventos	- 32 -
Tabla 10 Descripción del CU Gestionar publicaciones.....	- 32 -
Tabla 11 Descripción del CU Gestionar premios curriculares	- 33 -
Tabla 12 Descripción del CU Visualizar premio curricular	- 33 -
Tabla 13 Descripción del CU Gestionar convocatoria de premio curricular	- 34 -
Tabla 14 Visualizar convocatoria de premio curricular	- 34 -
Tabla 15 Descripción del CU Gestionar propuesta para optar por premio curricular	- 35 -
Tabla 16 Descripción del CU Registrar premiado.....	- 35 -
Tabla 17 Descripción del CU Visualizar premiado.....	- 36 -
Tabla 18 Descripción de la tabla tb_nalcance.....	- 44 -
Tabla 19 Descripción de la tabla tb_npatrocinador.....	- 44 -
Tabla 20 Descripción de la tabla tb_nentrega	- 45 -
Tabla 21 Descripción de la tabla tb_nfrecuencia	- 45 -
Tabla 22 Descripción de la tabla tb_ncondicion.....	- 45 -
Tabla 23 Descripción de la tabla tb_rpremio_condicion.....	- 45 -

Módulo de gestión de Publicaciones y Premios

Tabla 24 Descripción de la tabla tb_dpremio	- 46 -
Tabla 25 Descripción de la tabla tb_area_conocimiento.....	- 46 -
Tabla 26 Descripción de la tabla tb_rcampo_premio	- 46 -
Tabla 27 Descripción de la tabla tb_rcategoria_campo	- 47 -
Tabla 28 Descripción de la tabla tb_dcategoria	- 47 -
Tabla 29 Descripción de la tabla tb_dedicion_premio	- 47 -
Tabla 30 Descripción de la tabla tb_daval.....	- 48 -
Tabla 31 Descripción de la tabla tb_dotorgado.....	- 48 -
Tabla 32 Descripción de la tabla tb_dpropuesta_individual.....	- 48 -
Tabla 33 Descripción de la tabla tb_dpropuesta_colectivo	- 48 -
Tabla 34 Descripción de la tabla tb_dpropuesta_resultado	- 49 -
Tabla 35 Descripción de la tabla tb_dpropuesta_publicacion.....	- 49 -
Tabla 36 Descripción de la tabla tb_rmiembro_colectivo.....	- 49 -
Tabla 37 Descripción de la tabla tb_rautor_resultado	- 49 -
Tabla 38 Descripción de la tabla tb_drevista.....	- 50 -
Tabla 39 Descripción de la tabla tb_dpublicacion.....	- 51 -
Tabla 40 Descripción de la tabla tb_dautor_publicacion	- 51 -
Tabla 41 Descripción de la tabla tb_rrevista_referencia	- 51 -
Tabla 42 Descripción de la tabla tb_rrevista_idioma	- 51 -
Tabla 43 Descripción de la tabla tb_rrevista_idioma	- 52 -
Tabla 44 Descripción de la tabla tb_rrevista_editorial	- 52 -
Tabla 45 Descripción de la tabla tb_rrevista_patrocinadores	- 52 -
Tabla 46 Descripción de la tabla tb_rrevista_publicacion	- 52 -
Tabla 47 Descripción de la tabla tb_rpublicacion_repositorio.....	- 53 -
Tabla 48 Descripción de la tabla tb_rrepositorio_tema	- 53 -

Módulo de gestión de Publicaciones y Premios

Tabla 49 Descripción de la tabla tb_dreferencia	- 53 -
Tabla 50 Descripción de la tabla tb_npais	- 54 -
Tabla 51 Descripción de la tabla tb_neditorial.....	- 54 -
Tabla 52 Descripción de la tabla tb_norganizacion	- 54 -
Tabla 53 Descripción de la tabla tb_ntema	- 54 -
Tabla 54 Descripción de la tabla tb_drepositorio	- 54 -
Tabla 55 Descripción de escenarios del CU Gestionar revista.....	- 58 -
Tabla 56 Descripción de los casos de prueba del CU Gestionar revista	- 59 -
Tabla 57 Descripción de los casos de prueba del CU Gestionar revista (con datos).....	- 60 -

Índice de figura

Fig. 1 Ejemplo de Arquitectura Cliente-Servidor	- 13 -
Fig. 2 Patrón MVC.....	- 14 -
Fig. 3 Diagrama de casos de uso del negocio	- 24 -
Fig. 4 Diagrama de actividades del caso de uso Registrar premios	- 25 -
Fig. 5 Diagrama de actividades del caso de uso Registrar publicaciones.....	- 26 -
Fig. 6 Funcionamiento de Symfony de acuerdo al MVC	- 38 -
Fig. 7 Diagrama de clases del diseño del CU Gestionar revista	- 39 -
Fig. 8 Diagrama de secuencia del CU “Visualizar revista”	- 40 -
Fig. 9 Diagrama de secuencia CU Gestionar revista. Escenario “Registrar revista”	- 40 -
Fig. 10 Diagrama de secuencia CU Gestionar revista. Escenario “Editar revista”.....	- 41 -
Fig. 11 Diagrama de secuencia CU Gestionar revista. Escenario “Eliminar revista”.....	- 41 -
Fig. 12 Diagrama de clases persistentes del módulo Premios	- 43 -
Fig. 13 Diagrama de clases persistentes del módulo Publicaciones	- 44 -
Fig. 14 Diagrama de componentes	- 57 -
Fig. 15 Diagrama de despliegue.....	- 57 -
Fig. 16 Fragmento de código de las pruebas funcionales del CU Gestionar revista.....	- 60 -
Fig. 17 Fragmento de código de prueba funcional del método save () de la clase PropuestaResultado	- 61 -
Fig. 18 Fragmento del resultado de las pruebas funcionales del CU Gestionar revista.....	- 62 -
Fig. 19 Resultado de la prueba del método save () de la clase PropuestaResultado	- 62 -

Introducción

La gestión deviene un elemento esencial para el buen funcionamiento de cualquier entidad, debido a que es responsable del manejo de toda la información que fluye dentro de esta. Este proceso es muy abarcador, por lo que resulta indispensable la división en líneas o grupos de trabajo para mejorar la organización a la hora de llevar a cabo el mismo. Dependiendo de la estructura de la institución, estas líneas pueden ser varias, teniendo en cuenta el volumen de información que maneja cada una.

La Universidad de las Ciencias Informáticas(UCI) es una entidad muy grande, que se desarrolla a pasos agigantados en la mayor parte de sus esferas, y para tener un indicador que permita medir su desarrollo, asocia estas a direcciones de trabajo, entre las cuales se encuentran la Dirección de Investigaciones (DI), encargada de las diferentes líneas de investigación. Cada una de estas direcciones se ubica como una línea de procesos dentro del Sistema de Gestión Universitaria, que es el nombre que recibe el proyecto encargado de realizar un grupo de aplicaciones informáticas con el objetivo de automatizar los flujos de trabajo más importantes dentro de la Universidad.

La DI es el resultado de agrupar las diferentes líneas investigativas de la Universidad, teniendo en cuenta el carácter creador del modo de producción de la UCI, que implica generar conocimiento para poder desarrollar mejores productos de software y profesionales más capaces. Desde sus inicios, trabaja incansablemente por la organización y el buen desempeño de la actividad científica del centro, y sus esfuerzos están encaminados a guiar y orientar metodológicamente la actividad del Sistema de Ciencia, Tecnología e Innovación (CTI) en la Universidad y estimular la participación de estudiantes y trabajadores, establecer alianzas con diferentes instituciones científicas e introducir el centro dentro del sistema científico nacional. Como una esfera de importancia estratégica para el desarrollo de la universidad, a esta dirección corresponde la línea de investigaciones dentro del Sistema de Gestión Universitaria.

Internamente, el sistema de CTI se conforma por varios procesos, entre los cuales se pueden mencionar Eventos Científicos, el cual es responsable de la gestión de los sucesos científico-académicos que se realizan dentro y fuera de la entidad, Premios Curriculares y Publicaciones Científicas, que se encarga de la gestión tanto de premios académico-científicos que entrega la Universidad como del control de los premios otorgados por otras instituciones científicas a personal del centro y de las publicaciones académico-científicas que se generan en la Universidad.

Módulo de gestión de Publicaciones y Premios

Se entienden como publicaciones científicas: artículos, tesis, ponencias y ensayos de carácter científico-académico que se encuentran en fuentes tales como revistas, libros e Internet. Los premios gestionados, se dividen según su alcance: UCI (Premios del Rector), nacionales e internacionales. Entre los nacionales se pueden mencionar los premios CITMA otorgados por el Ministerio de Ciencia, Tecnología y Medio Ambiente, el premio Nacional Anual para Estudiantes Investigadores, entre otros. Dentro de los premios internacionales se pueden mencionar el TWAS, patrocinado por la Academia de Ciencias para el Mundo en Desarrollo y el Consejo Nacional para Investigaciones Científicas y Tecnológicas (CONICIT) de Costa Rica, el premio LÓREAL para mujeres científicas, y el premio Kalinga para la popularización de las ciencias patrocinado por la UNESCO/India.

La gestión de las publicaciones y los premios en la Universidad genera gran cantidad de información, por lo que el fácil y rápido acceso a la misma implica el buen funcionamiento del centro, a la vez que agilizan los procesos de control en la Universidad y en particular en la DI.

Situación problemática

A pesar de que la UCI es una universidad muy joven, ya cuenta entre su comunidad con gran cantidad de profesionales con un expediente curricular extenso, que participan en eventos científicos y académicos, publican gran cantidad de conocimiento, y que como resultado de su trabajo obtienen premios que reconocen su quehacer científico, los cuales se registran en las diferentes estructuras de CTI de la Universidad.

Todo este proceso de registro de premios curriculares en la Universidad se realiza de forma manual y la información resultante se almacena actualmente en tablas Excel o páginas Word, lo que implica que la información no se encuentra debidamente organizada y estructurada, dificultando así el procesamiento de la misma por parte del personal encargado en la dirección.

Por otra parte, el proceso de registrar una publicación resulta un tanto engorroso, debido a que los datos de la publicación y el autor se envían por correo electrónico al encargado de realizar el registro en la Editorial. Dependiendo de cómo y dónde se realizó la publicación, se registran los datos de la misma. En caso de que la publicación se haya realizado en un evento científico, el autor necesita llevar directamente al local donde radica el grupo editorial el dispositivo que almacena las memorias del evento para la confirmación de autoría, mientras que si la fuente es una revista o libro, el autor debe llevar un ejemplar del mismo o enviar la dirección web en caso de que estuviese publicada en internet. Una vez recogidos todos estos datos, el grupo editorial se encarga de validar el registro. Los datos se almacenan en una base de datos bastante limitada y pequeña, con un tiempo de respuesta bastante grande. El sistema para acceder a ella es obsoleto y con una interfaz poco amigable, lo que representa una dificultad para los usuarios de la aplicación a la hora de trabajar con ella.

Módulo de gestión de Publicaciones y Premios

Dada esta situación, este trabajo plantea como **problema científico**: ¿Cómo facilitar el proceso de control de premios y publicaciones científicas en la Universidad de las Ciencias Informáticas?

Para dar solución a este problema, se tomará como **objeto de estudio** los procesos del Sistema de Ciencia, Tecnología e Innovación en la UCI y como **campo de acción** los procesos de gestión de publicaciones y premios de la Universidad de las Ciencias Informáticas.

Teniendo en cuenta lo planteado anteriormente, el **objetivo general** de este trabajo es desarrollar un sistema que, integrado al resto de los procesos de Ciencia Tecnología e Innovación, facilite el proceso de control de premios curriculares y publicaciones científicas en la Universidad de las Ciencias Informáticas.

Para cumplir con el objetivo general expuesto, se han trazado varios **objetivos específicos** que facilitan y organizan el trabajo:

1. Analizar el estado del arte y las condiciones actuales en que se realiza el proceso de control de publicaciones y premios en la universidad.
2. Realizar diseño de la aplicación.
3. Construir la aplicación.
4. Realizar las pruebas a la aplicación.

Tras definir los objetivos y analizar el problema, se obtiene la siguiente **hipótesis**: la implementación del Módulo de Gestión de Publicaciones y Premios facilitará el proceso de gestión de publicaciones y premios dentro del Sistema de Ciencia, Tecnología e Innovación en la Universidad de las Ciencias Informáticas.

Para hacer efectivo el cumplimiento de cada uno de los objetivos mencionados, se orienta un conjunto de **tareas** que aseguran el éxito de cada objetivo:

1. Valoración de las técnicas de programación, lenguaje y marco de trabajo para el desarrollo de la aplicación.
2. Análisis del estado del arte en relación con sistemas para el control de premios curriculares y publicaciones científicas.
3. Modelación de las condiciones actuales en que se desarrolla el control de premios y publicaciones en la UCI.
4. Identificación de las funcionalidades que tendrá el sistema.

5. Realización del diseño de la aplicación.
6. Implementación del sistema.
7. Realización de las pruebas de integración.

Estructura del trabajo

Este documento está estructurado por capítulos, acápites y secciones, de forma que facilita la lectura y la búsqueda de información por el índice.

En el **capítulo 1** se realiza una fundamentación teórica sobre los sistemas de control de publicaciones y premios curriculares, su evolución y situación actual en la Universidad, las herramientas más utilizadas así como las ventajas y desventajas de usar cada una de ellas. Además, se hace una breve explicación de lo que significan los sistemas de control de publicaciones y premios curriculares para las entidades de tipo científico-académico.

El **capítulo 2** aborda todo el modelado del negocio del módulo. Ahí se explica cómo funciona este, se exponen las descripciones completas de los requisitos funcionales y no funcionales con que contará el sistema, se definen los usuarios, los trabajadores, la descripción de cada caso de uso del negocio así como los diagramas de actividades correspondientes a cada uno.

En el **capítulo 3** explica todo lo referido al diseño del sistema. Se muestran los diagramas de clases así como los diagramas de interacción correspondientes, la descripción de las clases con que contará el sistema para lograr su funcionalidad, el modelo entidad-relación así como el diseño de la base de datos.

El **capítulo 4** muestra todo lo relacionado con la implementación del sistema a partir del diagrama de componentes y la obtención del diagrama de despliegue, así como la descripción de las pruebas realizadas al sistema.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En este capítulo se abordan una serie de aspectos que fundamentan teóricamente esta investigación, de modo que resulta palpable su importancia. Para esto, se ha dividido el mismo en varios temas que de forma general incluyen antecedentes históricos y conceptos principales relacionados con esta investigación.

1.2. Investigación científica

La investigación científica es el procedimiento utilizado por una entidad ligada de algún modo a la ciencia que puede ser una persona, institución, proyecto, o simplemente un grupo de interesados en algún campo específico, para comprobar una hipótesis, encontrar respuestas a una interrogante, dar solución a un problema, formular teorías, entre otras (1).

No existe una investigación científica que sea común para todas las ciencias, debido a que el método, los objetivos y las formas en que se realiza la indagación están estrechamente relacionados con el fenómeno que se estudia (1).

La generación de conocimientos, de nuevos productos y servicios, es la base de la investigación científica, que se traduce en el uso de la innovación tecnológica como factor clave para el crecimiento y avance de la ciencia tanto en las instituciones de educación superior como en el progreso y bienestar de la sociedad en general. Como consecuencia, el progreso de la ciencia depende, en gran medida, de la difusión y divulgación del conocimiento, por medio de las revistas científicas impresas y electrónicas. Es decir, que la comunicación científica es la culminación del proceso de investigación, y el inicio del intercambio, la discusión y aceptación nacional e internacional entre los pares y el resto de la comunidad de usuarios del conocimiento científico (1).

1.3. Publicaciones científicas

Las publicaciones científicas constituyen uno de los resultados finales de cualquier investigación científica, aumentando la posibilidad de que ocurra de acuerdo a la envergadura de la exploración a la cual hace referencia. Estas tienen la misión de transmitir el conocimiento generado durante todo el proceso de búsqueda por lo que constituyen uno de los últimos pasos dentro de las actividades investigativas.

La definición de "publicación científica" abarca las revistas periódicas de cualquier disciplina cuya actividad central es la difusión de informes, avances o proyectos de investigaciones, así como artículos o ensayos de investigadores universitarios o de centros de investigación privados o estatales, docentes, alumnos o graduados. Estos artículos suelen estar sometidos a referato. Cuando esto se hace, es con el objetivo declarado de corroborar tanto la validez del trabajo como su interés y oportunidad. En algunos casos, la revisión puede dar lugar a sugerencias al autor del trabajo, para que corrija algún aspecto de éste antes de su publicación, o puede ser la base o referencia para otros trabajos relacionados con el campo de la investigación (2).

Entre los tipos de publicaciones científicas más comunes, se pueden nombrar las reseñas bibliográficas, los artículos y ensayos.

1.4. Premios curriculares

Un premio, según la definición de la Real Academia de la Lengua Española, es un lote o recompensa que se le otorga a una entidad como reconocimiento a una obra, una actividad o una cualidad (7). De aquí se puede inferir que un premio curricular se traduce en un reconocimiento a un currículo con determinadas características. Estos pueden ser otorgados a una publicación científica, una persona o grupo, o incluso al resultado de una actividad científica.

Los premios en general, y los curriculares en particular, tienen varias características, entre las cuales podemos citar que poseen cierta periodicidad, un patrocinador que es el encargado de hacer entrega del mismo, el alcance que puede ser local, nacional, o internacional y poseen cierto valor que puede ser material o no.

Teniendo en cuenta que la comunidad universitaria está vinculada a la investigación y el desarrollo constante de nuevos productos y tecnologías, y que produce grandes volúmenes de información científica agrupada en publicaciones de cierta relevancia, productos de software y de capital humano, así como resultados de impacto social, cultural, científico y económico, se puede afirmar que la universidad es una cantera de potenciales ganadores de premios curriculares, y que la cantidad de premios, así como su relevancia, son indicadores de eficiencia y desarrollo que definen la madurez con que evoluciona la institución en una rama determinada.

1.5. Sistemas de Control de Publicaciones

El control de publicaciones es el nombre con que se le conoce al proceso de revisar, registrar y finalmente sacar a la luz cierta publicación científica. Dado que hay instituciones, como las editoriales, que publican grandes volúmenes de información científica en diversos tipos de publicaciones, llevar un control estricto de las mismas constituye una necesidad vital.

Las herramientas más primitivas para esta actividad son los llamados catálogos que almacenan los datos generales de las publicaciones y autores, mientras que las más modernas constituyen catálogos informáticos que simplifican notablemente la gestión de publicaciones. En lo adelante, cuando se hable de Sistemas de Control de Publicaciones (SCP) se hará referencia a los catálogos modernos o informáticos, por ser el interés directo y estricto de este trabajo.

1.5.1 Internacionales

Existen varios sistemas de control de publicaciones a nivel internacional de los cuales se nombrarán algunos de los más destacados y se caracterizarán de acuerdo a sus ventajas y desventajas.

C2C

Es un software de altas prestaciones desarrollado por la empresa C2CSoluciones para crear gestores de publicaciones en internet. (3)

Ventajas

Entre las ventajas con que cuenta este sistema se puede mencionar que, al ser de tipo web y estar online, brinda una alta disponibilidad de la información que se desea consultar. Contiene una interfaz relativamente amigable para el usuario y es fácil de usar por usuarios no informáticos. Posee un gestor de administración que permite administrar el gestor de publicaciones desde el rol de administrador.

Desventajas

Este software es de tipo propietario, lo que significa que su uso implica pagar una licencia, además del soporte de forma general. Además, este software está orientado a la gestión de publicaciones en lugar del control de las mismas.

Latindex

Latindex -Sistema Regional de Información en Línea para Revistas Científicas de América Latina, el Caribe, España y Portugal- es producto de la cooperación de una red de instituciones que funcionan de manera coordinada para reunir y disseminar información bibliográfica sobre las publicaciones científicas seriadas producidas en la región.(4)

Ventajas

Módulo de gestión de Publicaciones y Premios

Este software posee tres aspectos fundamentales: un directorio, un catálogo de información bibliográfica y proporciona enlaces a revistas electrónicas. Estos servicios proporcionan una alta disponibilidad de la información a partir de aumentar el rigor de las búsquedas de bibliografías. También es un sistema de tipo web, por lo que hace disponibles sus recursos desde cualquier lugar.

Desventajas

Este sistema está hecho con el objetivo de publicar a autores latinoamericanos y exclusivamente publicaciones seriadas, lo que limita el caudal de información que almacena en él. La interfaz que se muestra al usuario es poco atractiva por lo que los usuarios prefieren otros sistemas incluso con menos prestaciones. Carece de un diseño original y genera sensación de poca actividad.

INE

El Instituto Nacional de Ecología (INE) de México tiene como objetivo generar, integrar y difundir conocimiento e información a través de la investigación científica aplicada y el fortalecimiento de capacidades, para apoyar la formulación de política ambiental y la toma de decisiones que promuevan el desarrollo sustentable. Este instituto es objeto de estudio de esta investigación debido a que posee un sistema que permite la gestión de publicaciones científicas sobre ecología en general. (5)

Ventajas

Este sistema tiene una gran cantidad de prestaciones, sobre todo en el manejo de metadatos sobre las publicaciones, que es una de las carencias de los sistemas gestores de contenidos y otros dedicados al manejo de información. Contiene un catálogo de información sobre publicaciones en línea y en formato duro.

Desventajas

Este sistema limita su ámbito a publicaciones científicas de corte ecológico por lo que limita el proceso de gestión a autores e investigadores del área de la ecología. Además, el sistema es propietario y exige pago por cuestiones de licencia.

Open Journal System (OJS)

Open Journal Systems (OJS) es un sistema de gestión de publicaciones periódicas, que ha sido desarrollado por el *Public Knowledge Project* (PKP) a través de sus esfuerzos con fondos federales para ampliar y mejorar el acceso a la investigación. Este software es capaz de gestionar publicaciones tales como revistas, anuarios y diarios. OJS asiste en cada etapa del proceso de publicación con arbitraje, de las comunicaciones a través de la publicación en línea y la indexación. A través de sus sistemas de gestión, su indexación de nivel muy detallado de la investigación, y el contexto que proporciona para la investigación, OJS busca mejorar la calidad académica y pública de la

Módulo de gestión de Publicaciones y Premios

investigación con referato (6). Es el más complejo y completo de todos los antecedentes que se han estudiado hasta el momento en esta investigación.

Ventajas

- OJS se instala y se administra localmente.
- Contiene editores de los requisitos de configuración, secciones, proceso de revisión, edición, entre otros.
- Módulo de suscripción con retraso de las opciones de acceso abierto.
- La indexación completa de contenidos de la parte del sistema global.
- Herramientas de lectura de los contenidos, basados en el terreno y la elección de los editores.
- Uso de roles bien definidos con sus respectivos permisos.
- Notificación por correo electrónico y la capacidad de comentar para los lectores.
- Presenta un contexto completo de ayuda en línea provista por la comunidad del PKP.
- Es un sistema no propietario, que puede ser utilizado o modificado libremente.

Desventajas

El OJS es un sistema limitado debido a que está diseñado para ser usado en la gestión de publicaciones de carácter periódico, por lo que resulta difícil adaptarlo a las características de una empresa determinada con variados tipos de publicaciones.

1.5.2 Nacionales

Aunque existen muchos sitios que gestionan publicaciones de varios tipos, estos usan software de otros propietarios, de modo que no hay un antecedente directo en nuestro país de sistemas de gestión de publicaciones. Sí existen algunos sitios que, primitivamente, hacen cierta aproximación a la gestión de publicaciones como es el caso del Instituto de Información Científica y Tecnológica (IDICT), pero no se tiene conocimiento de lo que se denomina un software de control de publicaciones que haya sido construido en nuestro país o por alguna entidad cubana.

1.5.3 Sistemas usados en la UCI

En la Universidad actualmente existe la Serie Científica, revista en la cual se almacena cierto número de publicaciones, y se gestiona por medio de un OJS. Este sistema guarda los trabajos publicados, y otros en proceso de publicación, edición y revisión, pero aun así no es suficiente puesto que no es capaz de controlar todas las publicaciones que se generan en la comunidad universitaria, solamente las que se registran en ella.

Estos antecedentes son interesantes desde el punto de vista de su orientación a la gestión de publicaciones. Sin embargo, no se puede contar con ellos debido al número de limitaciones importantes que frenan el desarrollo en una sola herramienta informática de toda la actividad de control de la Dirección de Investigaciones.

1.6. Control de premios curriculares

El control de premios curriculares es el proceso para registrar y consultar las entregas de reconocimientos en un ámbito determinado. Como resultado de esta actividad, siempre debe quedar registrada la entrega del premio con todos los datos relacionados a esta. Aunque este proceso ocurre en muchas instituciones, especialmente en las de orden científico, no se usa un sistema informático estándar publicado que se encargue de automatizarlo.

En la actualidad, el registro de premios curriculares se lleva a cabo de forma manual por los encargados de esta tarea. No podemos afirmar con un nivel de seguridad total que no existan sistemas que se encarguen de este proceso de control, pero sí podemos afirmar que no está publicado en los catálogos de software y, al no contar con información alguna referente a ellos, no se pueden considerar un antecedente directo.

Por otra parte, el control de premios curriculares en la Universidad responde a una nueva necesidad, derivada de la política llevada a cabo por la dirección de la misma que se propone elevar el nivel científico de la UCI, a partir de reconocer y registrar los premios otorgados a resultados, colectivos, publicaciones y personal que más se destacan en la actividad científica.

Las limitaciones de los sistemas citados anteriormente, sumadas al hecho de que no existe ninguna herramienta que gestione y controle en una misma aplicación los premios curriculares y las publicaciones científicas dentro del sistema de CTI, hacen que el desarrollo de esta investigación cobre una especial importancia.

1.7. Metodología de desarrollo.

Al enfrentarse al desarrollo de un producto de software se hace indispensable tener básicamente una metodología. En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

Resumiendo, una metodología proporciona un esquema de roles que cumplen ciertas actividades en un momento determinado del desarrollo del producto de software, por lo que el rendimiento y el tiempo de duración del desarrollo es mucho menor en comparación con proyectos que no definen un método para la construcción del producto.

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente, desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería (8).

En el mundo del desarrollo de software, entre las metodologías más conocidas se encuentran el Proceso Unificado de Desarrollo (RUP), y la llamada Programación Extrema (XP), ambos por sus siglas en inglés. Otras metodologías menos conocidas pero igualmente interesantes son Método de Desarrollo de Sistemas Dinámicos (DSDM): proporciona un método simple de desarrollo, originalmente basada en el desarrollo rápido de aplicaciones web; *Microsoft Solutions Framework* (MSF): flexible e interrelacionada con una serie de conceptos, modelos, y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Existen otras como *Scrum* y *Crystal*, pero de interés para este trabajo resulta RUP, porque es la que se aplicará en el proceso de desarrollo debido a varias características.

1.7.1 RUP

RUP es el resultado de varios años de uso práctico en el que se han unificado técnicas de desarrollo a través de UML, y trabajo de muchas metodologías utilizadas por los clientes. Existe gran cantidad de documentación de proyectos construidos mediante esta metodología, de forma que resulta ideal a la hora de desarrollar con ella (9).

Entre sus principales características se destacan:

- **Centrado en la arquitectura:** muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. Se desarrolla mediante iteraciones, comenzando por los casos de uso (CU) relevantes desde el

punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

- **Iterativo e Incremental:** los flujos de trabajo se desarrollan en cascada. Propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.
- **Dirigido por casos de uso:** los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Por todas estas características, se ha definido esta metodología como rectora del proceso de desarrollo de la solución de software de esta investigación.

1.8. Arquitectura del software

Una arquitectura es un ensamblaje de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permiten integrar una amplia gama de productos y servicios informáticos, de forma tal que pueden ser utilizados eficazmente dentro de la organización (10). Define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación ente ellos.

Toda arquitectura de software debe cumplir con el principio de ser construida en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea. Hay que señalar que para seleccionar el modelo de una arquitectura, se debe partir del contexto tecnológico y organizativo del momento. La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

1.9. Arquitectura cliente-servidor

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar

Módulo de gestión de Publicaciones y Premios

información, servicios o recursos (11). Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, que denominaremos rol activo, y servidor al proceso que responde a las solicitudes (rol pasivo). En términos generales, esta arquitectura se forma a partir de la agrupación de conjuntos de elementos que efectúan procesos distribuidos y cómputo cooperativo.

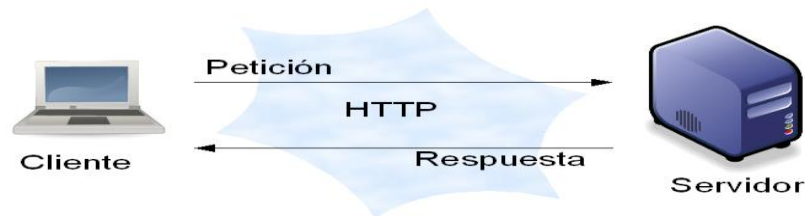


Fig. 1 Ejemplo de Arquitectura Cliente-Servidor

Entre otras ventajas, este modelo arquitectónico permite centralizar el control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos. Es de fácil mantenimiento al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes. Existen tecnologías suficientemente desarrolladas, diseñadas para el paradigma de Cliente-Servidor(C/S) que garantizan la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

1.9.1 Modelo-Vista-Controlador (MVC)

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web. (12)

El **modelo** es la representación específica del dominio de la información sobre la cual funciona la aplicación. Es otra forma de llamar a la capa de dominio y representa información persistente del mundo real con la cual trabaja la aplicación.

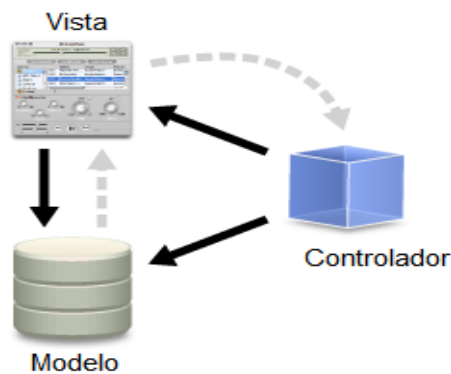


Fig. 2 Patrón MVC

La **vista** presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario. Toda la información obtenida del modelo se le muestra al usuario de la aplicación con elementos de diseño que la hacen amigable e interactiva.

El **controlador** responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos (gestiona las entradas del usuario).

Este patrón es implementado intrínsecamente por el marco de trabajo Symfony, que se abordará más adelante, de manera que la aplicación utilizará una versión mejorada de un patrón MVC convencional para aplicaciones web.

1.10. Tecnologías y lenguajes a utilizar

La tecnología a utilizar constituye una de las tres capas del desarrollo de software. De la elección cuidadosa, de su buen uso y del nivel de conocimiento que tiene el equipo de desarrollo sobre la tecnología en general y de las herramientas en particular depende en gran medida la duración del proyecto, la plena utilización de su potencial y por último la calidad con que transcurre el proceso de desarrollo y el producto final.

1.10.1 Tecnologías y lenguajes del lado del cliente

Se denominan tecnologías del lado del cliente a aquellas que participan en un proceso o transacción solamente desde el lado del cliente. Generalmente, proporcionan una interfaz que permite el intercambio entre el usuario y el sistema de una forma mucho más amigable. Estas tecnologías

participan en la interacción de forma independiente a lo que ocurre del lado del servidor, de manera que solo se ocupan de aspectos relacionados directamente con el cliente de la aplicación.

Hay que señalar además que estas tecnologías y lenguajes han sido definidos por el grupo de trabajo del Sistema de Gestión Universitaria como línea base para el desarrollo.

JavaScript

JavaScript es un lenguaje de scripting basado en objetos. Se utiliza principalmente integrado en un navegador web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Es un dialecto de ECMAScript y se caracteriza por tener entradas dinámicas y funciones de primera clase. Ha tenido influencias de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no son especialistas.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar y mejorar la calidad de una página web, JavaScript contiene embebida una implementación del Modelo de Objetos del Documento (DOM). Al utilizarlo podemos crear diferentes efectos e interactuar con nuestros usuarios de una forma mucho más acabada y moderna.

Lenguaje de marcado de Hipertexto Extensible

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende de HTML 4.0 combinando la sintaxis del mismo y diseñado para mostrar datos, mientras que con la de XML fue diseñado para describir los datos (13). Este lenguaje presenta ventajas claras sobre el HTML, entre ellas:

- Un navegador no necesita implementar heurísticas para detectar qué quiso poner el autor, por lo que la conversión de lenguajes puede ser mucho más sencillo.
- Como es también una implementación de XML se pueden utilizar fácilmente herramientas creadas para procesamiento de documentos XML (editores, XSLT, etc.).

Ajax

Acrónimo de asíncrono JavaScript And XML (JavaScript asíncrono + XML), es una tecnología de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona. JavaScript es el lenguaje en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante una petición en segundo plano que viene con formato XML. El objeto encargado de realiza la misma está disponible en los navegadores actuales. (14)

Ajax no es solo una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. Ajax incorpora:

- Presentación basada en estándares usando XHTML y CSS.
- Exhibición e interacción dinámicas usando el DOM.
- Intercambio y manipulación de datos usando XML and XSLT.
- Recuperación de datos asíncrona usando peticiones en segundo plano.
- JavaScript.

1.10.2 Tecnologías y lenguajes del lado del servidor

Este término se emplea para agrupar las tecnologías que participan en un proceso o transacción desde el servidor, de forma que son transparentes al cliente o usuario final de las aplicaciones que manejan arquitectura cliente-servidor. En el caso de las aplicaciones web, estas tecnologías generalmente están compuestas de un lenguaje de programación que se encarga de toda la lógica de implementación por parte del servidor, y en el caso de las aplicaciones dinámicas, un gestor de bases de datos.

Procesador de Hipertexto

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Su nombre es un acrónimo recursivo que significa procesador de hipertexto (inicialmente PHP *Tools*). Es actualmente un lenguaje de propósito general ampliamente utilizado y diseñado especialmente para desarrollo web. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Entre sus mejores características podemos señalar que:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

Módulo de gestión de Publicaciones y Premios

- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

Symfony

Es un completo marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (16)

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

Symfony se diseñó para que se ajustara a los siguientes requisitos:

1. Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y **nix* estándares).
2. Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Doctrine, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
3. Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
4. Sencillo de usar en la mayoría de casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.
5. Aunque utiliza MVC, tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Módulo de gestión de Publicaciones y Premios

6. Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
7. Una potente línea de comandos que facilita la generación de código, ahorrando tiempo de trabajo.

PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. No es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo (17).

PostgreSQL provee soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas).
- Direcciones IP (IPv4 e IPv6).
- Disparadores (triggers):
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte para transacciones distribuidas.

Es un sistema libre, gratuito y con una amplia comunidad internacional dedicada por completo a su desarrollo. Es el más potente de los sistemas gestores de bases de datos gratuitos que existen y también uno de los más documentados. Por todas estas razones, este sistema resulta ideal para desarrollar la solución de esta investigación.

Lenguaje de Modelado. UML

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad (18). Se utiliza para visualizar, especificar, construir y documentar un sistema y ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento.

1.11. Herramientas

Las herramientas en el desarrollo de software constituyen la tercera y no menos importante capa porque facilitan, automatizan y agilizan el proceso de desarrollo.

Servidor Apache

Es un servidor web de código abierto multiplataforma, que implementa el protocolo HTTP/1.1 e incorpora la noción de sitios virtuales. Inicialmente consistía solo en un conjunto de parches a aplicar al servidor de NCSA.

Apache es altamente configurable y permite bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración (19).

Entre sus más interesantes ventajas podemos mencionar su carácter modular, lo cual permite incrementar o quitar los módulos que posee en función de las necesidades específicas del usuario. Es un sistema de código abierto, lo que significa que se puede modificar para adaptarlo a los requerimientos. Por último, es el servidor web más popular, ya que es muy fácil de conseguir y con una gran comunidad que provee ayuda y soporte (19).

NetBeans

Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de cien socios en todo el mundo. *Sun Microsystems* fundó este proyecto en junio 2000 y continúa siendo el patrocinador principal. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos por lo que pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans 6.8

La nueva versión que trae NetBeans IDE es la 6.8 ME. Lo más interesante a juicio de este trabajo es la integración total que trae con el lenguaje de programación PHP y más específicamente con el framework *Symfony*. Esta característica permite el desarrollo ágil debido a un gran auto-

completamiento de código por parte del entorno de trabajo, de forma que ahorra tiempo a la hora de programar y evita errores comunes de tipografía a los desarrolladores. Por otra parte, esta nueva versión de NetBeans permite integrarse con el servidor web Apache durante el *debug* una aplicación web alojada en este servidor.

Visual Paradigm para UML

Visual Paradigm es una herramienta CASE profesional que incluye UML y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de mejor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundante información sobre UML, demostraciones interactivas y proyectos UML (20).

Entre las características de la versión 2.6 que se usará para el modelado durante el desarrollo de la solución podemos citar:

- Diagramas de Procesos del Negocio.
- Modelado colaborativo con Subversion.
- Ingeniería inversa.
- Editor de Detalles de Casos de Uso.
- Generación de bases de datos.
- Módulo ajustable para NetBeans.

Por último, aunque Visual Paradigm es una herramienta de software propietario, la Universidad cuenta con una licencia de esta versión, por lo que desarrollar con ella resulta factible desde el punto de vista económico.

Subversion

Es un software de sistema de control de versiones, libre bajo una licencia de tipo Apache/BSD y se le conoce también como *svn* por ser el nombre de la herramienta de la línea de comandos. Una característica importante de Subversion es que todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo de desarrollo. Este sistema puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. Entre sus características se hace notar que contiene un cliente modular ajustable a la plataforma NetBeans.

Módulo de gestión de Publicaciones y Premios

Al analizar estas herramientas, se debe destacar, sobre todo, la integración con el entorno de desarrollo NetBeans que provee al desarrollador de un ambiente de trabajo cómodo y completamente funcional que le permite ahorrar tiempo durante todo el proceso de creación, especialmente durante la fase de construcción y pruebas de la solución.

1.12. Conclusiones del capítulo

La ausencia de un sistema de control integrado de publicaciones científicas y premios curriculares en la UCI, hace que el desarrollo de este módulo cobre especial importancia por el aporte que significa para el desarrollo de la comunidad científica y particularmente dentro del sistema de CTI de la Universidad.

En esta primera fase se ha hecho un estudio profundo, fundamentando teóricamente el uso de la metodología, lenguajes, tecnologías y herramientas en este proyecto con sus principales características y ventajas, demostrando que al utilizarlas se ahorra tiempo, recursos y se minimizan los errores durante el proceso de desarrollo.

Capítulo 2: Propuesta de solución

2.1 Introducción

En este capítulo se realiza un estudio detallado del modelo de negocio, abordando la descripción de los actores, trabajadores y casos de usos, se muestran los diagramas de casos de uso, de actividades y el modelo de objetos del negocio. Además, se comienzan a definir los principales requerimientos funcionales y no funcionales que deberá tener el sistema.

2.2 Actores, trabajadores y casos de uso

Actor: es una entidad externa al sistema que guarda relación con este y que le demanda una funcionalidad. Pueden ser operadores humanos, sistemas externos ó entidades abstractas como el tiempo.

Trabajador: define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. (1)

Caso de uso: es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. (1)

Requisito: característica del sistema que es una condición para su aceptación por el cliente. (1)

2.3 Procesos del negocio

Publicaciones científicas

Para el proceso de emisión de un certificado de publicación, el autor debe presentarse en el local donde radica la editorial con los datos la fuente donde se encuentra la publicación, ya sea la dirección web, el formato duro o el dispositivo de almacenamiento con las memorias del evento. El especialista de la editorial valida los datos de la fuente e informa al autor de la conformidad o no de la misma. En caso de ser aceptada, se procede a emitir el certificado de publicación y se notifica al autor que debe pasar a recoger el mismo por el local de la editorial.

Premios curriculares

El proceso de registro de un premio curricular comienza cuando la institución promotora del mismo abre una convocatoria. La DI se encarga entonces de circular los datos de la convocatoria por las diferentes áreas. La dirección del área se encarga de enviar los expedientes de las propuestas a la Dirección de Investigaciones donde se revisa que el expediente esté completo y correctamente conformado. La DI presenta las propuestas en el Consejo Científico de la Universidad donde se realiza una valoración y se decide si se aprueban o no las propuestas, emitiéndose un aval de aprobación en caso positivo. Posteriormente la DI se encarga de enviar las propuestas a las instituciones promotoras del premio.

Una vez que los promotores tienen los resultados, la Dirección es la encargada de darlos a conocer en las diferentes áreas.

2.4 Modelado del negocio

2.4.1 Reglas del negocio

Publicaciones científicas:

- Para emitirse un certificado de publicación deben presentarse todos los datos de la fuente de publicación en el grupo editorial.
- La no aprobación de una publicación en una fuente anula todo el proceso de certificación de la misma.
- Si los datos de la fuente no son congruentes o están incompletos, no se realizará el proceso de certificación y se notifica al autor las causas de la anulación del certificado.
- Cuando se produce la aprobación de una publicación, se debe notificar al autor para que recoja el certificado.

Premios curriculares:

- Para emitir un expediente de una propuesta a optar por un premio curricular, debe haberse lanzado la convocatoria del premio.
- Las propuestas y conformación de los expedientes correspondientes, debe ser realizado por la dirección del área.

Módulo de gestión de Publicaciones y Premios

- Las propuestas se deben revisar en el Consejo Científico de la Universidad.
- La no aprobación del expediente de una propuesta anula el proceso de selección de la misma por parte de la DI para enviar a la institución promotora del premio.

2.4.2 Diagrama de casos de usos del negocio

El diagrama de casos de uso del negocio es una representación de la relación que existe entre los actores y casos de uso del negocio, de forma tal que constituye una guía de cómo ocurren los procesos de negocio.

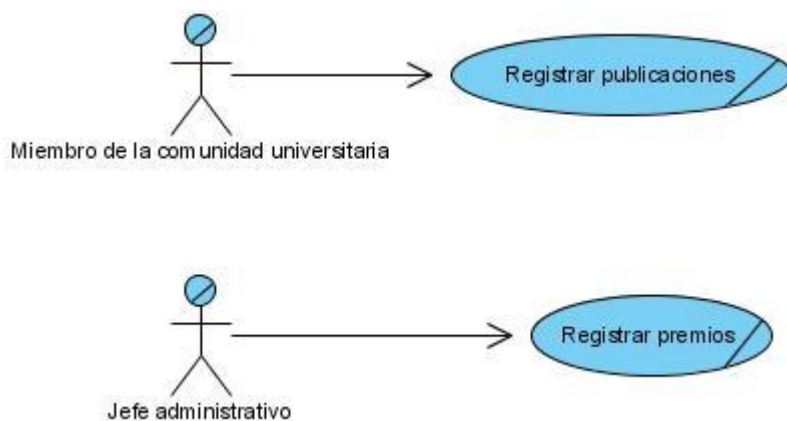


Fig. 3 Diagrama de casos de uso del negocio

2.4.3 Descripción de actores y trabajadores del negocio

Actor	Descripción
Jefe administrativo	Es la persona que propone el premio a otorgar. Pueden ser decanos, directores, vicerrectores y cualquier otro jefe administrativo con facultad para ello.
Miembro de la comunidad universitaria	Puede ser cualquier persona de la comunidad universitaria (profesores, trabajadores, estudiantes, etc.).
Trabajador	Descripción
DI	Es el especialista de la Dirección facultado para desarrollar determinados procesos.
Especialista de la editorial	Es el especialista del grupo editorial facultado para gestionar los certificados de publicaciones de la comunidad universitaria.

Tabla 1 Descripción de actores y trabajadores del negocio

2.4.4 Descripción de casos de usos del negocio

Registrar premios

Caso de uso del negocio	Registrar premios.
Actores	Jefe administrativo.
Resumen	El caso de uso del negocio comienza cuando algún jefe administrativo presenta en la DI los expedientes curriculares de las personas propuestas para optar por algún premio curricular. Esta selección debe hacerse en las áreas teniendo en cuenta los currículos de los mismos. La DI debe enviar a las instituciones promotoras de los premios dichos expedientes y esta a su vez, enviar a la DI los resultados, finalizando así el caso de uso.
Casos de uso asociados	

Tabla 2 Descripción del caso de uso Registrar premios

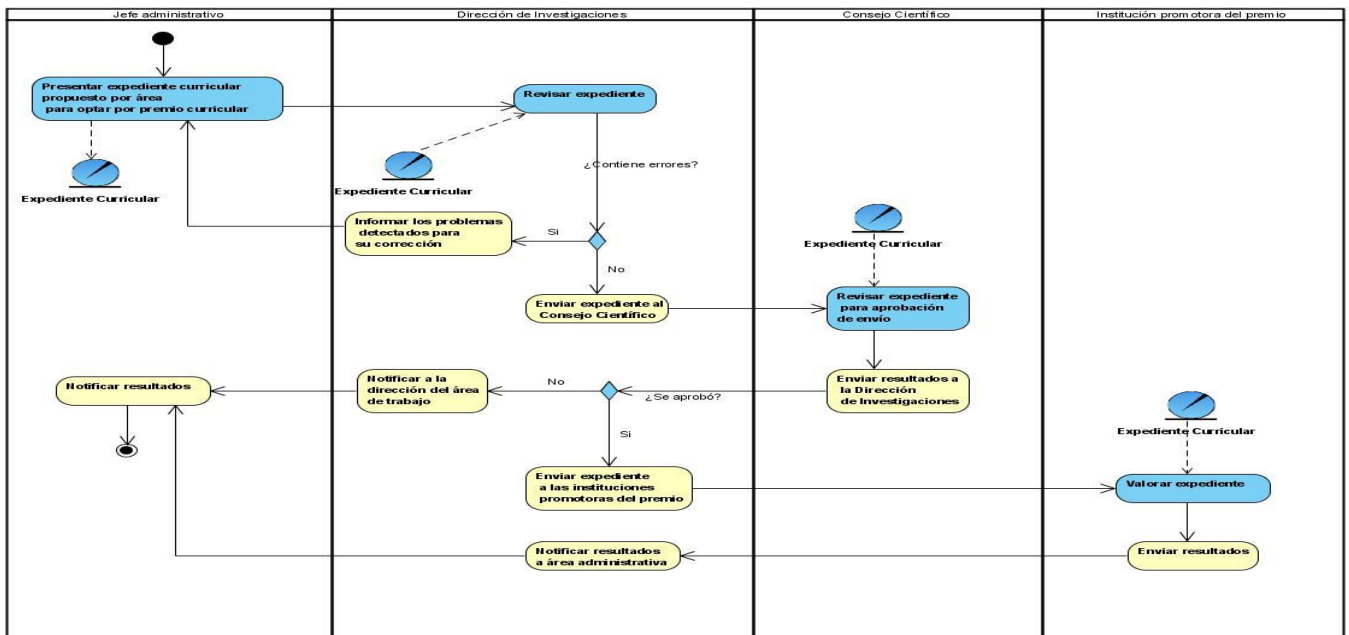


Fig. 4 Diagrama de actividades del caso de uso Registrar premios

Registrar publicaciones

Caso de uso del negocio	Registrar publicaciones.
--------------------------------	---------------------------------

Módulo de gestión de Publicaciones y Premios

Actores	Miembro de la comunidad universitaria.
Resumen	El caso de uso del negocio comienza cuando cualquier miembro de la comunidad universitaria solicita al grupo editorial -por correo electrónico o personalmente- el certificado de alguna publicación que haya realizado. El especialista de la editorial comprueba dicha solicitud en la fuente de publicación que le haya proporcionado el solicitante. El caso de uso del negocio termina cuando el especialista certifica o no la publicación.
Casos de uso asociados	

Tabla 3 Descripción del caso de uso Registrar publicaciones

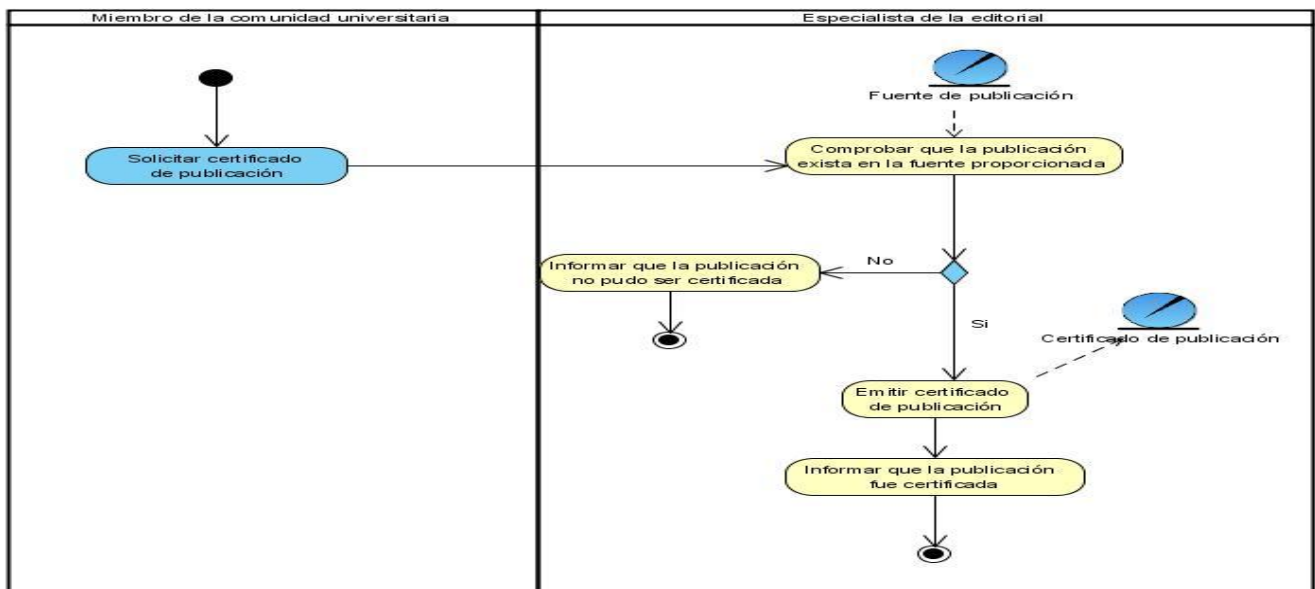


Fig. 5 Diagrama de actividades del caso de uso Registrar publicaciones

2.5 Modelado del Sistema

2.5.1 Requerimientos funcionales del sistema

RF1- Registrar revistas.

RF2- Mostrar revistas registradas.

RF3- Mostrar datos de una revista.

RF4- Modificar datos de una revista.

Módulo de gestión de Publicaciones y Premios

RF5- Eliminar revista.

RF7- Mostrar repositorios registrados.

RF8- Mostrar datos de un repositorio.

RF6- Registrar repositorio.

RF9- Modificar datos de un repositorio.

RF10- Eliminar repositorio.

RF11- Mostrar eventos que lleven memorias, tengan o no registrado su ISBN.

RF12- Registrar ISBN de memorias de eventos.

RF13- Modificar ISBN.

RF14- Registrar publicación nueva.

RF15- Registrar publicación en otras fuentes.

RF16- Mostrar publicaciones realizadas.

RF17- Mostrar datos de la publicación registrada.

RF18- Modificar publicación.

RF19- Eliminar publicación.

RF20- Emitir certificado de publicación.

RF1- Registrar características de premios curriculares.

RF2- Mostrar premios curriculares registrados.

RF3- Mostrar datos de un premio curricular.

RF4- Modificar datos de un premio curricular.

RF5- Eliminar premio curricular.

RF6- Registrar convocatoria de premio curricular.

RF7- Mostrar datos de una convocatoria de premio curricular.

Módulo de gestión de Publicaciones y Premios

RF8- Modificar los datos de una convocatoria de premio curricular.

RF9- Cerrar convocatoria de premio curricular (no permitir subir más propuestas).

RF10- Cerrar premio (se hará una vez que se registren los premios otorgados).

RF11- Eliminar una convocatoria de premio curricular.

RF12- Registrar propuesta de premio curricular.

RF13- Mostrar propuestas de premios curriculares.

RF14- Mostrar los datos de una propuesta.

RF15- Mostrar otros premios, presentación en eventos y publicaciones que haya tenido la propuesta en otras ocasiones.

RF16- Modificar los datos de una propuesta de premio curricular.

RF17- Eliminar una propuesta de premio curricular.

RF18- Modificar el estado de una propuesta de premio curricular en diferentes niveles (registrar premiado).

RF19- Mostrar premios curriculares obtenidos.

RF20- Mostrar premios obtenidos de participación en eventos.

2.5.2 Requerimientos no funcionales del sistema

Interfaz:

- Interfaz interna: estará determinada por los desarrolladores, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.

Usabilidad:

- Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

Módulo de gestión de Publicaciones y Premios

- Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.

Seguridad:

- Seguridad de la base de datos: la base de datos deberá estar fraccionada en esquemas que permitan un mejor uso de la información y la división de forma lógica de las funcionalidades del sistema, garantizando la protección de la información al ocurrir un incidente sobre una parte de la base de datos.
- Registro sistemático de incidencias: el sistema debe ser capaz de registrar el accionar del usuario, así como permitir auditorías y exámenes de las trazas tanto en tiempo real como en históricos. Se precisa un monitor de incidencia para la visualización y tratamiento de las mismas.
- Alta protección de los datos: al estar trabajando con información sensible, se hace necesario una alta protección de los datos a nivel de aplicación y de tráfico de red, para tal fin se ha definido la seguridad en varios niveles dentro de la aplicación: nivel de interfaz, nivel de acceso a datos y nivel de base de datos.

Eficiencia:

- El sistema debe soportar un tiempo de respuesta menor o igual a cinco segundos.
- El sistema debe soportar una conexión simultánea de más de diez mil usuarios.

Soporte:

- Grupo de soporte y asesoría: el sistema contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo.

Software:

- Servidor Web con soporte para PHP 5.3 y framework Symfony versión 1.4 o superior.

2.5.3 Actores del sistema:

Actor	Descripción
DI.	Es el usuario encargado de la configuración y administración del sistema. Tiene acceso a todas las funcionalidades del mismo.

Módulo de gestión de Publicaciones y Premios

Usuario simple.	Son todos los usuarios de la comunidad universitaria, que solo podrán visualizar determinada información del sistema, además de poder inscribir sus publicaciones.
Especialista de la editorial	Es el encargado de aprobar las publicaciones y fuentes que registren los responsables de las diferentes áreas, así como emitir los certificados de publicación.
Responsable del área	Son las personas que podrán registrar fuentes de publicación y las publicaciones correspondientes. Entre ellas se pueden encontrar el asesor de investigación, vicedecano de producción-investigación, líderes científicos, de centros y otros directivos.
Consejo Científico UCI	Es el grupo encargado de valorar la aprobación o no de determinados proyectos a nivel UCI.

Tabla 4 Descripción de los actores del sistema

2.5.4 Descripción de los casos de uso del sistema

Caso de Uso:	Visualizar revistas.
Actores:	Usuario simple.
Resumen:	El caso de uso se inicia cuando el usuario accede a las fuentes de publicación para visualizar las revistas que hayan registradas. El caso de uso finaliza cuando se muestra la página con los detalles de la revista.
Precondiciones:	
Referencias	RF2, RF3
Prioridad	Media.

Tabla 5 Descripción del CU Visualizar revista

Caso de Uso:	Gestionar revista.
Actores:	Responsable de área.
Resumen:	El caso de uso se inicia cuando el usuario accede a las fuentes de publicación para registrar, modificar o eliminar los datos de una revista determinada. El caso de uso finaliza cuando se muestra la página de detalles de una revista o se muestra el listado

Módulo de gestión de Publicaciones y Premios

	de revistas después de eliminar. Este usuario solo tendrá acceso a modificar y eliminar las revistas que hayan sido registradas por él y mientras el estado de la revista esté como "propuesto".
Precondiciones:	Para modificar y eliminar, debe primeramente haber listado las revistas y haber accedido a los detalles de la revista.
Referencias	RF1, RF4, RF5
Prioridad	Alta.

Tabla 6 Descripción del CU Gestionar revista

Caso de Uso:	Visualizar repositorios.
Actores:	Usuario simple.
Resumen:	El caso de uso se inicia cuando el usuario accede a las fuentes de publicación para visualizar los repositorios que hayan registrados. El caso de uso finaliza cuando se muestran los detalles del repositorio.
Precondiciones:	
Referencias	RF7, RF8
Prioridad	Media.

Tabla 7 Descripción del CU Visualizar repositorios

Caso de Uso:	Gestionar repositorios.
Actores:	Responsable de área.
Resumen:	El caso de uso se inicia cuando el usuario accede a las fuentes de publicación para registrar, modificar o eliminar los datos de un repositorio determinado. El caso de uso finaliza cuando se muestra la página de detalles del repositorio o la página del listado en caso de que se haya eliminado. Este usuario solo tendrá acceso a modificar y eliminar los repositorios que hayan sido registrados por él y mientras el estado del repositorio esté como "propuesto".
Precondiciones:	Para modificar y eliminar, debe primeramente haber listado los repositorios y haber accedido a los detalles del repositorio.

Módulo de gestión de Publicaciones y Premios

Referencias	RF6, RF9, RF10
Prioridad	Alta.

Tabla 8 Descripción del CU Gestionar repositorios

Caso de Uso:	Gestionar memorias de eventos.
Actores:	Especialista de la editorial.
Resumen:	El caso de uso se inicia cuando el usuario accede a las fuentes de publicación para registrar, visualizar o modificar el ISBN de un evento que lleve memorias. El caso de uso finaliza cuando se muestra la página de detalles de la memoria o la página del listado en caso de que se haya eliminado.
Precondiciones:	Cuando se registró la edición del evento se debió poner que incluía memorias.
Referencias	RF11, RF12, RF13
Prioridad	Alta.

Tabla 9 Descripción del CU Gestionar memorias de eventos

Caso de Uso:	Gestionar publicación.
Actores:	Especialista de la Editorial.
Resumen:	<p>El caso de uso se inicia cuando el usuario accede a publicaciones para registrar, visualizar, modificar o eliminar los datos de una publicación, así como para registrar una publicación en otras fuentes y emitir el certificado de publicación. El caso de uso finaliza cuando se muestra la página de detalles de la publicación o la página del listado en caso de que se haya eliminado.</p> <p>Dentro de los datos a modificar también saldrá un campo para cambiar el estado de la publicación registrada (propuesto, aprobado, rechazado). Cuando se registre la publicación, saldrá por defecto "propuesto".</p>
Precondiciones:	La fuente de publicación deberá estar registrada.
Referencias	RF14, RF15, RF17, RF18, RF19, RF20
Prioridad	Alta.

Tabla 10 Descripción del CU Gestionar publicaciones

Módulo de gestión de Publicaciones y Premios

Caso de Uso:	Gestionar premios curriculares.
Actores:	DI.
Resumen:	El caso de uso se inicia cuando el usuario accede a los premios para registrar, visualizar, modificar o eliminar los datos de un premio curricular. El caso de uso finaliza cuando se accede a otra opción.
Precondiciones:	
Referencias	RF1, RF2, RF3, RF4, RF5
Prioridad	Alta.

Tabla 11 Descripción del CU Gestionar premios curriculares

Caso de Uso:	Visualizar premio curricular.
Actores:	Usuario simple.
Resumen:	El caso de uso se inicia cuando el usuario accede a visualizar los tipos de premios curriculares. Se podrá también visualizar información del tipo de premio. El caso de uso finaliza cuando se muestran los datos.
Precondiciones:	
Referencias	RF2, RF3.
Prioridad	Alta.

Tabla 12 Descripción del CU Visualizar premio curricular

Caso de Uso:	Gestionar convocatoria de premio curricular.
Actores:	DI.
Resumen:	El caso de uso se inicia cuando el usuario accede a los premios curriculares para registrar, visualizar, modificar o eliminar los datos de una convocatoria de premio curricular, así como para cerrar la convocatoria o cerrar el premio. El caso de uso finaliza cuando se muestra la página de detalles de la convocatoria o la página del listado en caso de que se haya eliminado.
Precondiciones:	<ul style="list-style-type: none">• Para realizar cualquier opción con las convocatorias de premios curriculares, debe primero listar todos los premios curriculares.

Módulo de gestión de Publicaciones y Premios

	<ul style="list-style-type: none"> Para registrar una convocatoria, el premio no debe tener ninguna convocatoria abierta y se debe pinchar encima del premio. <p>Para modificar, eliminar, cerrar convocatoria y cerrar premio, se debe acceder a la convocatoria en caso de que tenga.</p>
Referencias	RF6, RF8, RF9, RF10, RF11.
Prioridad	Alta.

Tabla 13 Descripción del CU Gestionar convocatoria de premio curricular

Caso de Uso:	Visualizar convocatoria de premio curricular
Actores:	Usuario simple.
Resumen:	El caso de uso se inicia cuando el usuario accede a la convocatoria de un premio curricular para visualizar su información. El caso de uso finaliza cuando se muestra la página de detalles de la convocatoria.
Precondiciones:	Debe estar la convocatoria abierta o al menos el premio. Esto último significa que ya no se podrán registrar más propuestas, pero se continúan mostrando los datos de la convocatoria hasta tanto no se cierre el premio y se registren los resultados del mismo.
Referencias	RF7.
Prioridad	Alta.

Tabla 14 Visualizar convocatoria de premio curricular

Caso de Uso:	Gestionar propuestas para optar por premio curricular.
Actores:	Responsable de área.
Resumen:	El caso de uso se inicia cuando el usuario accede a una convocatoria de premio curricular para registrar, visualizar, modificar o eliminar los datos de una propuesta para optar por un premio. El caso de uso finaliza cuando se muestra la página de detalles de la propuesta o la página del listado en caso de que se haya eliminado.
Precondiciones:	<ul style="list-style-type: none"> Para realizar cualquier acción, se debe acceder a la convocatoria del premio. <p>Para modificar y eliminar una propuesta, además, se debe acceder a los datos en detalle de esa propuesta.</p>

Módulo de gestión de Publicaciones y Premios

Referencias	RF12, RF13, RF14, RF15, RF16, RF17.
Prioridad	Alta.

Tabla 15 Descripción del CU Gestionar propuesta para optar por premio curricular

Caso de Uso:	Registrar premiado.
Actores:	Dirección de Investigaciones.
Resumen:	<p>El caso de uso se inicia cuando el usuario accede a las propuestas para registrar el/los premiado(s) que otorga la academia que lanza la convocatoria del premio curricular. El caso de uso finaliza cuando se muestra la página de detalles de la propuesta premiada o la página del listado en caso de que se haya eliminado.</p> <p>Este caso de uso es similar al “Valorar propuesta para optar por premio curricular”, solo que en este caso en “Estado” saldrá la opción “Aceptado por la Academia”, que significará haber otorgado el premio a esa propuesta.</p>
Precondiciones:	<ul style="list-style-type: none"> • Debe acceder a la convocatoria del premio. • Debe listar las propuestas.
Referencias	RF18.
Prioridad	Alta.

Tabla 16 Descripción del CU Registrar premiado

Caso de Uso:	Visualizar premiados.
Actores:	Usuario simple.
Resumen:	El caso de uso se inicia cuando el usuario accede a ver las personas que han resultado premiadas, tanto en participación en eventos, como en premios curriculares. El caso de uso termina cuando se muestra la página de detalles de la propuesta premiada.
Precondiciones:	<ul style="list-style-type: none"> • Para visualizar las personas que han obtenido premios curriculares, debe haber listado los premios curriculares que estén registrados. • Para visualizar los premios de participación en eventos, debe haber listado los eventos en los que se hayan obtenido premios.
Referencias	RF19, RF20.

Prioridad	Alta.
------------------	-------

Tabla 17 Descripción del CU Visualizar premiado

2.6 Conclusiones

En este capítulo, después de haber realizado un análisis profundo del negocio y de la forma en que se desarrollan los procesos en la DI con respecto a publicaciones científicas y premios curriculares, han quedado definidos los casos de uso con los que contará el sistema, así como las reglas del negocio y los requerimientos funcionales y no funcionales de la aplicación.

Capítulo 3: Diseño de la solución propuesta.

3.1 Introducción

El diseño es un refinamiento del análisis que tiene como objetivo formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución, que prepara para la implementación, pruebas del sistema y que pretende crear un plano del modelo de implementación, por lo que se define como propósitos del diseño:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y finalmente tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas de forma temprana en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

3.2 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Para lograr una mejor comprensión en cuanto al diseño se realiza un breve resumen donde se explica en qué se basa y cómo funciona el flujo de trabajo de Symfony.

3.2.1 Estructura de Symfony

Symfony está basado en el patrón arquitectónico MVC, pero este implementa todas las ventajas de dicha arquitectura, separa la vista (interfaz) y el modelo (base de datos) mediante el controlador que es el encargado de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista, pero además Symfony hace otra división en el modelo.

Módulo de gestión de Publicaciones y Premios

Modelo: Symfony divide el modelo en una capa de acceso a datos y otra de abstracción de datos. La abstracción indica que quiere de la base de datos, y la capa de acceso hace las consultas necesarias para obtener esa información, de esta forma si se cambia de base de datos, solamente se cambiaría la capa de acceso, y la capa de abstracción podría seguir haciendo las mismas operaciones.

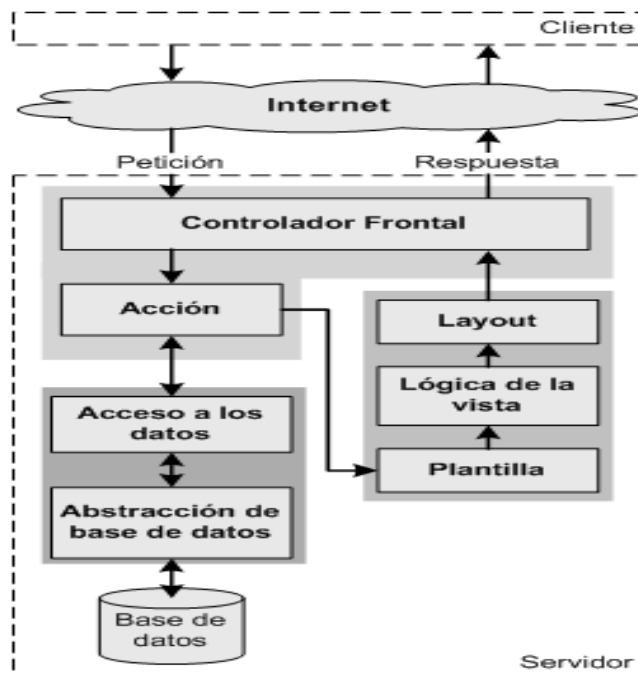


Fig. 6 Funcionamiento de Symfony de acuerdo al MVC

Vista: En la presentación de la mayoría de las páginas existen varios elementos comunes como son: la cabecera, la navegación, el pie de página y la plantilla global conocido como *layout*, cambiando tan solo interior o contenido de la página. Así estos tres elementos quedan separados.

Controlador: El trabajo del controlador se repite para muchas acciones. Symfony crea un controlador frontal, único en la aplicación, que está encargado de realizar labores comunes, como son: el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Además es el único punto de entrada a la aplicación.

Symfony utiliza un ORM para gestionar el acceso a la base de datos, por lo que puede guardar, obtener, modificar y borrar información en una base de datos sin crear sentencias SQL a mano y sin tener que descender hasta los detalles más técnicos de cada base de datos. Otra gran ventaja del ORM es que se puede cambiar de una base de datos otra simplemente cambiando una opción en un archivo de configuración.

3.2.2 Diagrama de clases del diseño

Los diagramas de clases del diseño son utilizados con el objetivo de representar las relaciones que existen entre los distintos tipos de clases. Para la realización de los siguientes diagramas se tuvo en cuenta el marco de trabajo de Symfony, por lo que el mismo queda estructurado de la siguiente forma:

Modelo: contiene las clases que modelan los datos persistentes en la base de datos.

Vista: contiene las páginas clientes dentro de cada submódulo.

Controlador: compuesto por las clases Actions por su nombre en inglés.

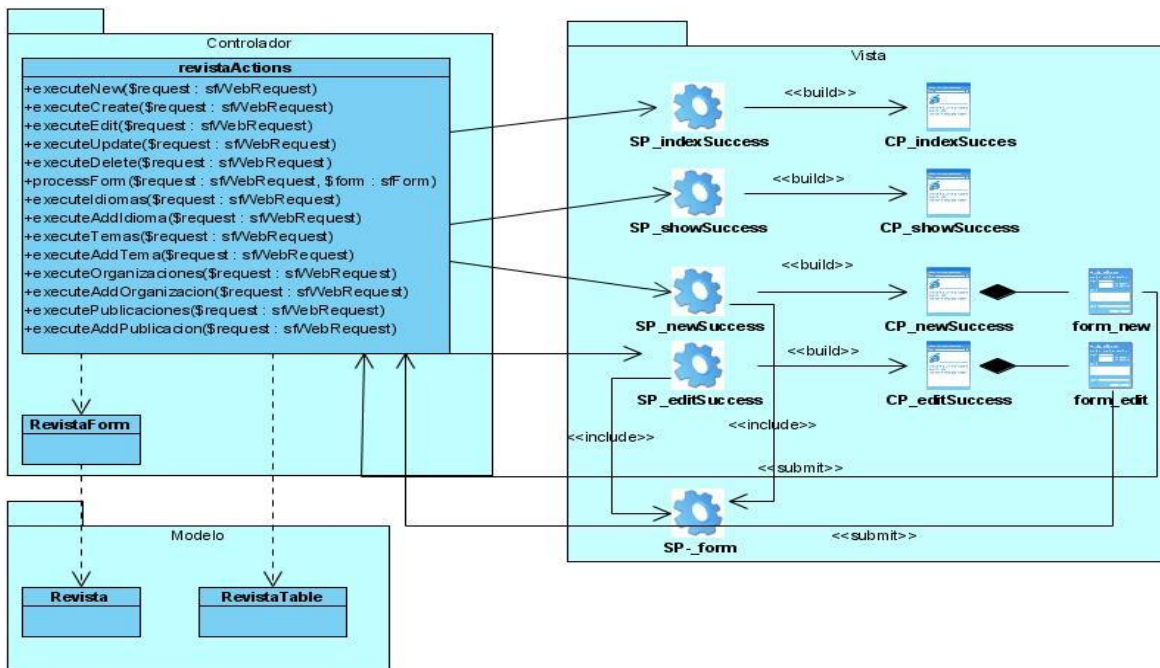


Fig. 7 Diagrama de clases del diseño del CU Gestionar revista

3.2.3 Diagramas de iteración: diagrama de secuencia.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el escenario que ilustra un comportamiento determinado. En el contexto de las clases se describe la forma en que grupos de objetos colaboran para proveer un comportamiento. Mientras que un diagrama de casos de uso presenta una visión externa del sistema, la funcionalidad de dichos casos de uso se recoge como un flujo de eventos utilizando para ello interacciones entre sociedades de objetos.

Módulo de gestión de Publicaciones y Premios

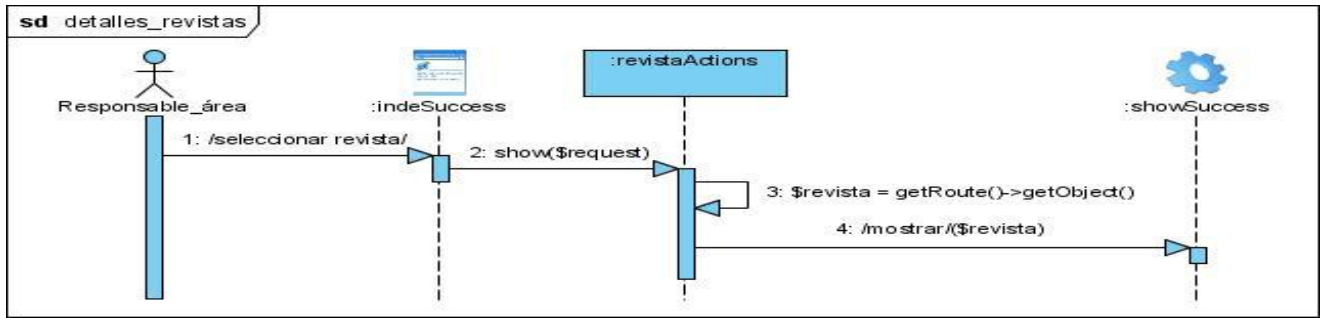


Fig. 8 Diagrama de secuencia del CU “Visualizar revista”

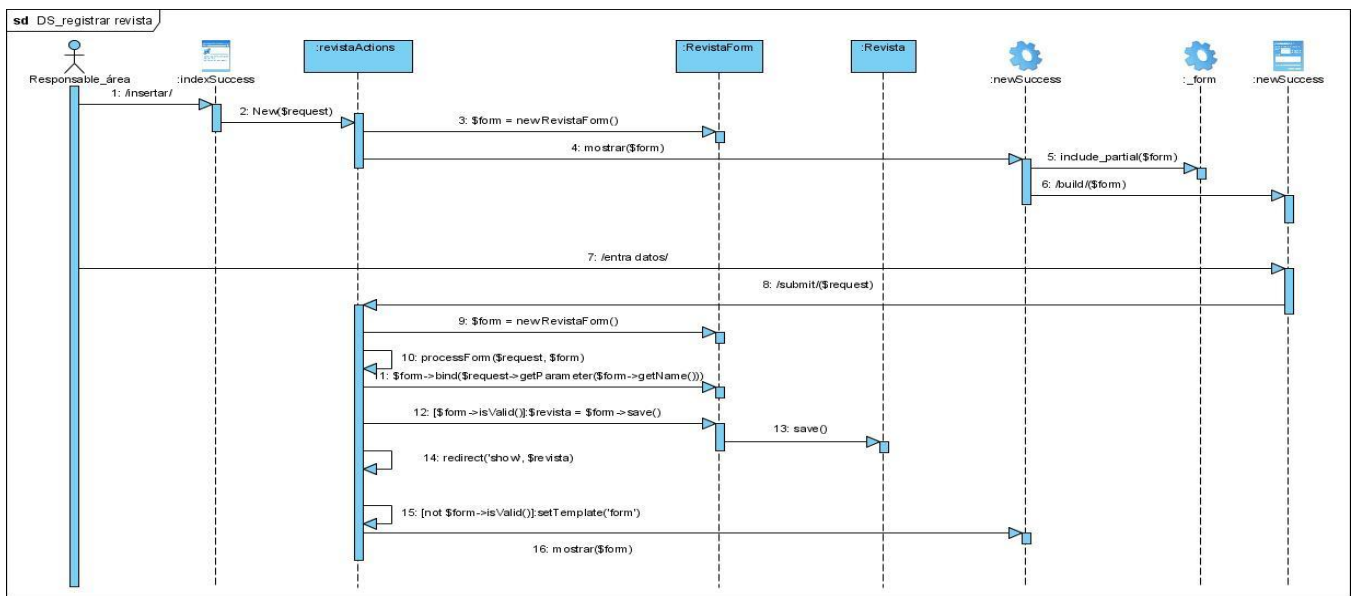


Fig. 9 Diagrama de secuencia CU Gestionar revista. Escenario “Registrar revista”

Módulo de gestión de Publicaciones y Premios

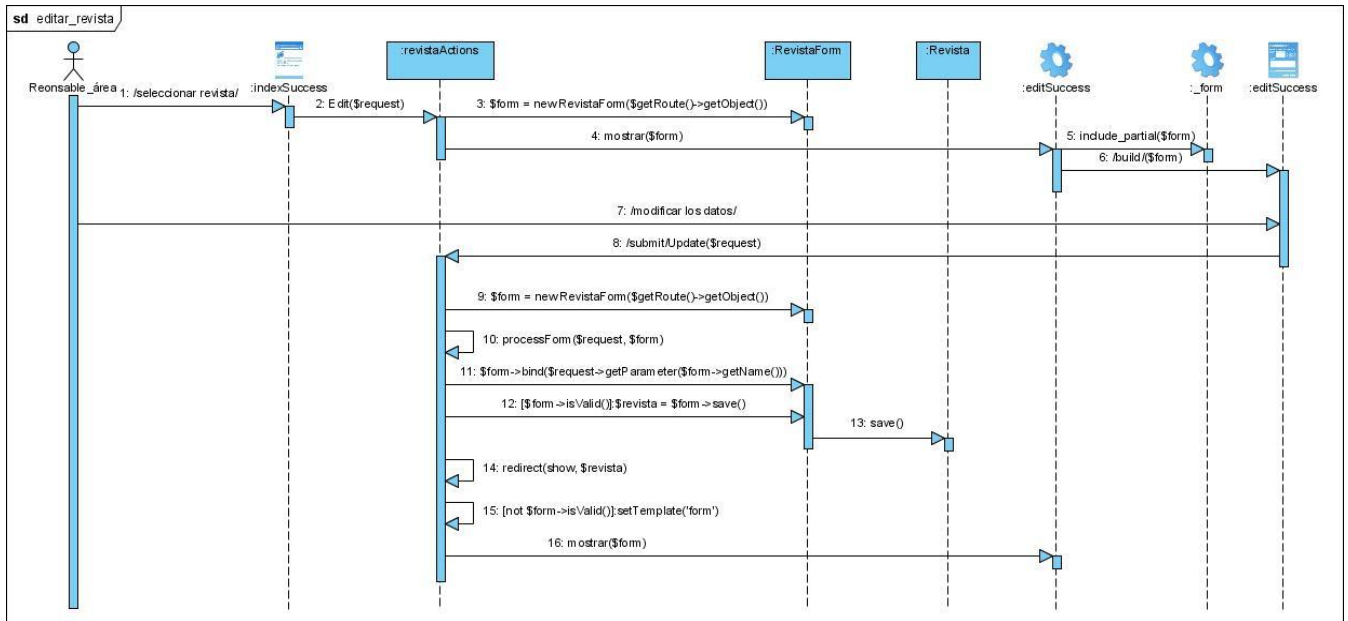


Fig. 10 Diagrama de secuencia CU Gestionar revista. Escenario “Editar revista”

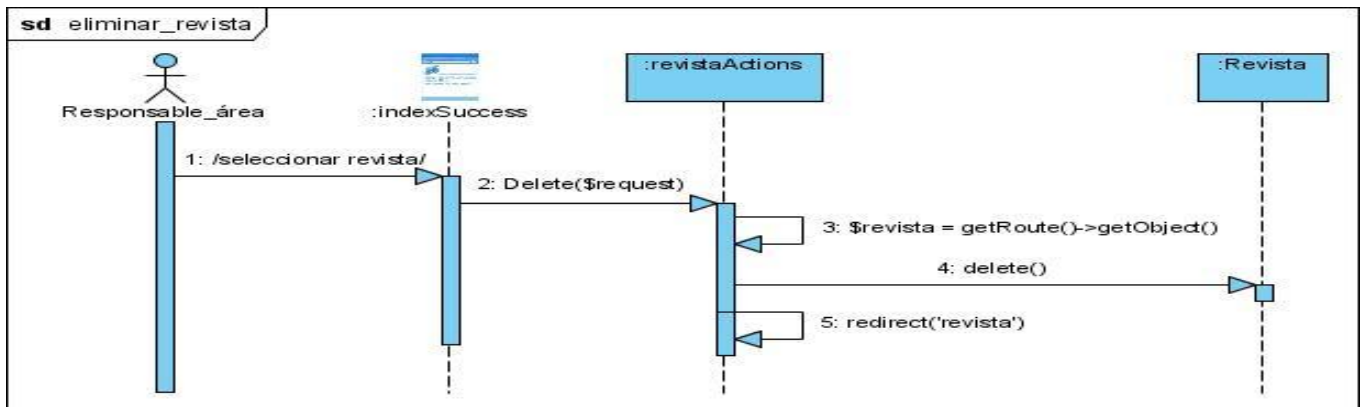


Fig. 11 Diagrama de secuencia CU Gestionar revista. Escenario “Eliminar revista”

3.3 Patrones de diseño

Los llamados patrones de diseño constituyen una solución a un problema relacionado con el diseño. Se consideran procedimientos para solucionar problemas del mismo tipo repetidas veces y son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Los desarrolladores lo usan como una forma de reutilizar la experiencia, clasificando las soluciones con términos de común denominación.

3.3.1 Patrones GRASP (Patrones generales de software para asignar responsabilidades)

Alta Cohesión: Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras

Módulo de gestión de Publicaciones y Premios

para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Creador: En la clase Actions de cada uno de los módulos se encuentran definidas las acciones del sistema. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

Controlador: Todas las peticiones son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Bajo Acoplamiento: Este patrón se manifiesta en cada uno de los módulos del sistema, la clase Actions hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

3.3.2 Patrones GOP

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el plantilla global, o si se mira desde otro punto de vista, la plantilla global decora la plantilla.

Patrón Abstract Factory (Factoría abstracta): este patrón proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. Este permite configurar en tiempo de ejecución un sistema con una familia u otra de objetos. Además garantiza que un conjunto de clases se usen a la vez.

3.4 Diagrama de clases persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases persistentes por lo general tienen como origen las clases entidad que modelan la información y el comportamiento asociado de algún fenómeno o concepto, como un objeto del mundo real o un suceso.

El diagrama de clases persistentes cuenta con todas las clases del modelo de datos incluyendo sus atributos, métodos y las relaciones que se establecen entre ellos. A continuación se muestra el diagrama de clases persistentes correspondientes a cada módulo.

Módulo de gestión de Publicaciones y Premios

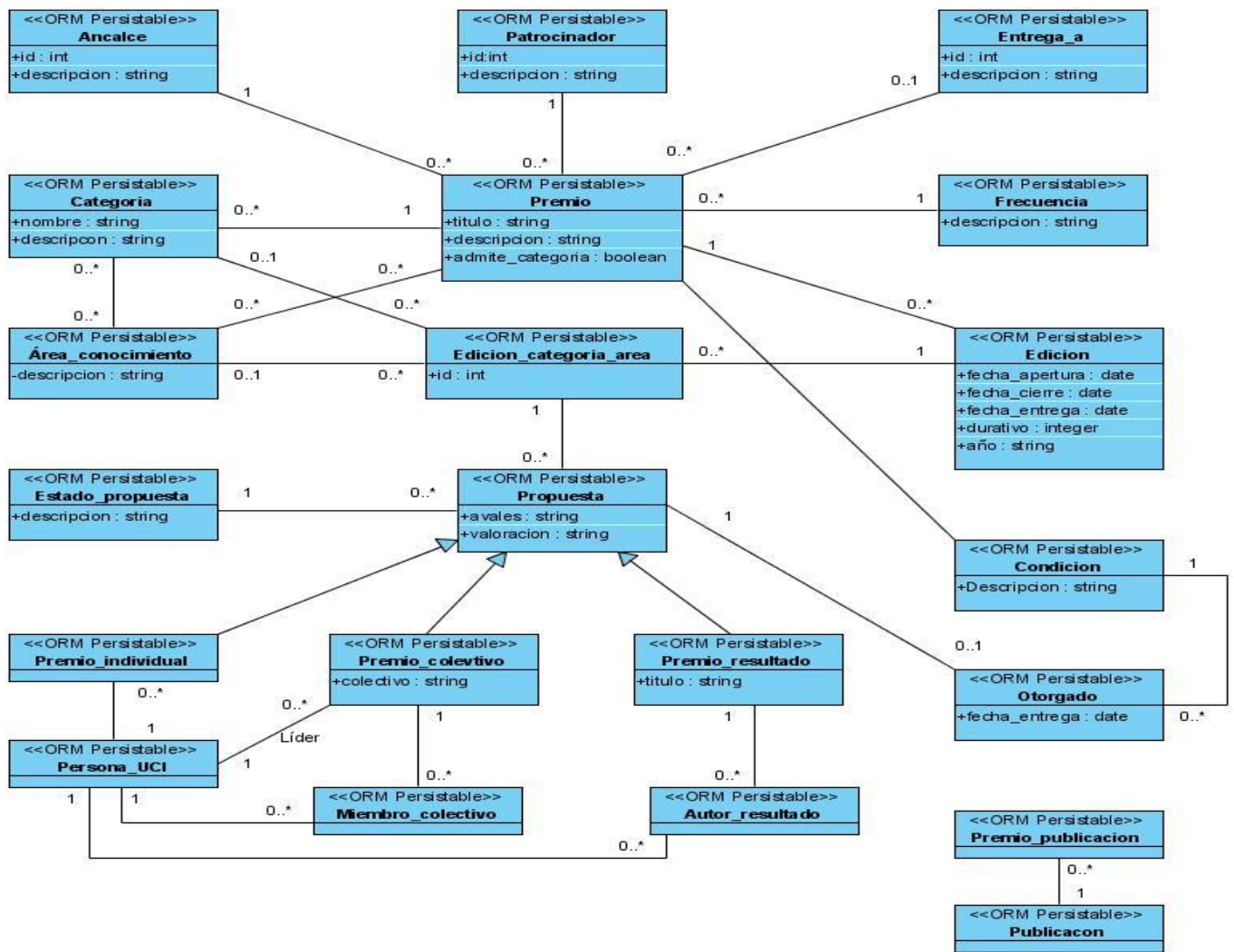


Fig. 12 Diagrama de clases persistentes del módulo Premios

Módulo de gestión de Publicaciones y Premios

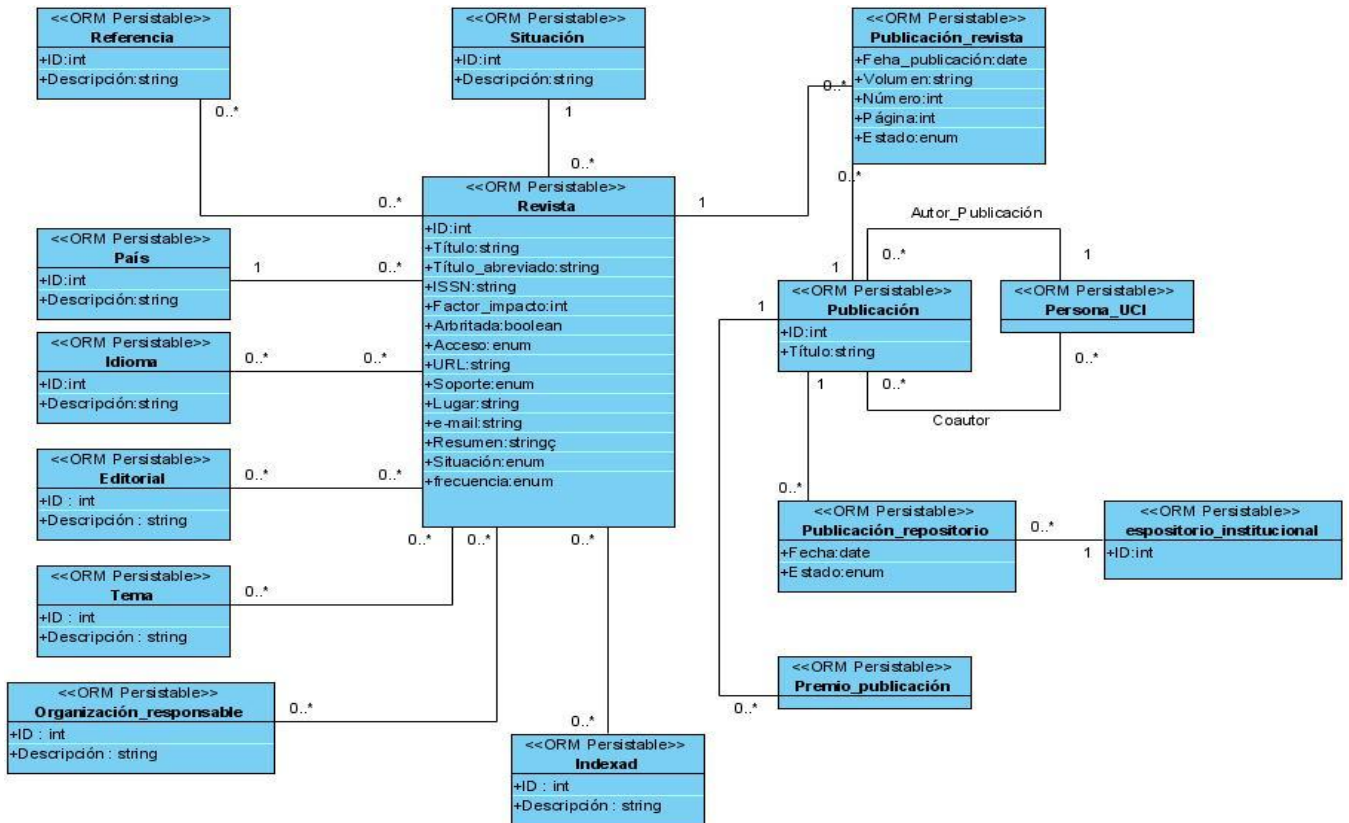


Fig. 13 Diagrama de clases persistentes del módulo Publicaciones

3.4.1 Descripción de las tablas

Módulo: Premios

Nombre: tb_nalcanace		
Descripción: Guarda el alcance del premio.		
Atributo	Tipo	Descripción
descripcion.	string	Breve descripción del alcance.

Tabla 18 Descripción de la tabla tb_nalcanace

Nombre: tb_npatrocinador		
Descripción: almacena el nombre del patrocinador.		
Atributo	Tipo	Descripción
nombre	string	Nombre del patrocinador.

Tabla 19 Descripción de la tabla tb_npatrocinador

Módulo de gestión de Publicaciones y Premios

Nombre: tb_nentrega		
Descripción: almacena información acerca de a quien se le entrega el premio.		
Atributo	Tipo	Descripción
se_entrega_a	string	Objetivo del premio. Puede ser un resultado, una publicación, un colectivo o un individuo.

Tabla 20 Descripción de la tabla tb_nentrega

Nombre: tb_nfrecuencia		
Descripción: define la frecuencia con que ocurre un evento determinado		
Atributo	Tipo	Descripción
descripción	string	Breve descripción de la frecuencia.

Tabla 21 Descripción de la tabla tb_nfrecuencia

Nombre: tb_ncondicion		
Descripción: Define las condiciones de entrega de premios o el contexto en que se desarrolla un evento		
Atributo	Tipo	Descripción
condición	string	Condición bajo la cual se desarrolla un evento determinado o se entrega un premio.

Tabla 22 Descripción de la tabla tb_ncondicion

Nombre: tb_rpremio_condicion		
Descripción: Relación entre el premio y las condiciones.		
Atributo	Tipo	Descripción
premio_id	int	Identificador del premio.
condicion_id	int	Identificador de condición.

Tabla 23 Descripción de la tabla tb_rpremio_condicion

Nombre: tb_dpremio		
--------------------	--	--

Módulo de gestión de Publicaciones y Premios

Descripción: Contiene información acerca del premio.		
Atributo	Tipo	Descripción
titulo	string	Titulo del premio.
descripción	string	Breve descripción del premio.
admite_categoria	booeLAN	Categoría del premio.
frecuencia_id	int	Identificador de la frecuencia.
alcance_id	int	Identificador del alcance.
entrega_id	int	Identificador de la entrega.
patrocinador_id	int	Identificador del patrocinador.

Tabla 24 Descripción de la tabla tb_dpremio

Nombre: tb_darea_conocimiento		
Descripción: contiene el área de conocimiento.		
Atributo	Tipo	Descripción
descripción	string	Breve descripción del área de conocimiento.

Tabla 25 Descripción de la tabla tb_area_conocimiento

Nombre: tb_rcampo_premio		
Descripción: Relación entre el campo y el premio.		
Atributo	Tipo	Descripción
premio_id	int	Identificador del premio.
campo_id	int	Identificador del campo.

Tabla 26 Descripción de la tabla tb_rcampo_premio

Nombre: tb_rcategoria_campo		
Descripción: Relación entre el campo y las categoría.		

Módulo de gestión de Publicaciones y Premios

Atributo	Tipo	Descripción
campo_id	int	Identificador del campo.
categoria_id	int	Identificador de la categoría.

Tabla 27 Descripción de la tabla tb_rcategoria_campo

Nombre: tb_dcategoria		
Descripción: Contiene información referente a la categoría.		
Atributo	Tipo	Descripción
nombre	string	Nombre de la categoría.
descripción	string	Breve descripción de la categoría.
premio_id	int	Identificador del premio.

Tabla 28 Descripción de la tabla tb_dcategoria

Nombre: tb_dedicion_premio		
Descripción: Ediciones de los premios		
Atributo	Tipo	Descripción
apertura	time	Tiempo de apertura.
cierre	time	Tiempo de cierre.
entrega_propuestas	time	Tiempo de entrega.
duración	int	Duración de la edición.
abierta	boolean	Estado de la edición.
premio_id	int	Identificador del premio.

Tabla 29 Descripción de la tabla tb_dedicion_premio

Nombre: tb_daval		
Descripción: Contiene la información correspondiente al premio que se entrega.		

Módulo de gestión de Publicaciones y Premios

Atributo	Tipo	Descripción
propuesta_id	int	Identificador de la propuesta.
nombre	string	Nombre del aval.
url	string	Dirección donde se encuentra.

Tabla 30 Descripción de la tabla tb_daval

Nombre: tb_dotorgado		
Descripción: Define las propuestas a las que se le ha otorgado algún premio		
Atributo	Tipo	Descripción
propuesta_id	int	Identificador de la propuesta.
fecha_entrega	time	Fecha de entrega.

Tabla 31 Descripción de la tabla tb_dotorgado

Nombre: tb_dpropuesta_individual		
Descripción: Propuesta de tipo individual.		
Atributo	Tipo	Descripción
autor_id	int	Identificador del autor.

Tabla 32 Descripción de la tabla tb_dpropuesta_individual

Nombre: tb_dpropuesta_colectivo		
Descripción: Propuestas de tipo colectivo.		
Atributo	Tipo	Descripción
colectivo	string	Nombre del colectivo.
autor_id	int	Identificador del autor.

Tabla 33 Descripción de la tabla tb_dpropuesta_colectivo

Nombre: tb_dpropuesta_resultado		
Descripción: Propuestas de resultados.		

Módulo de gestión de Publicaciones y Premios

Atributo	Tipo	Descripción
autor_id	int	Identificador del autor.
resultado	string	Resultado que se propone al premio.

Tabla 34 Descripción de la tabla tb_dpropuesta_resultado

Nombre: tb_dpropuesta_publicacion		
Descripción: Propuesta de publicaciones.		
Atributo	Tipo	Descripción
publicacion_id	int	Identificador de la publicación.

Tabla 35 Descripción de la tabla tb_dpropuesta_publicacion

Nombre: tb_rmiembro_colectivo		
Descripción: Relación entre los miembros y el colectivo.		
Atributo	Tipo	Descripción
propuesta_id	int	Identificador de la propuesta.
autor_id	int	Identificador del autor.

Tabla 36 Descripción de la tabla tb_rmiembro_colectivo

Nombre: tb_rautor_resultado		
Descripción: Relación entre personas y resultados.		
Atributo	Tipo	Descripción
propuesta_id	int	Identificador de la propuesta.
autor_id	int	Identificador del autor.

Tabla 37 Descripción de la tabla tb_rautor_resultado

Módulo de publicaciones:

Nombre: tb_drevista		
---------------------	--	--

Módulo de gestión de Publicaciones y Premios

Descripción: Contiene los datos de la revista.		
Atributo	Tipo	Descripción
titulo	string	Título de la revista.
titulo_abreviado	string	Título abreviado de la revista.
ISSN	string	Identificador de la fuente de publicación en el sistema bibliográfico.
factor_impacto	int	Factor de impacto otorgado por la editorial.
acceso	enum	Define si el acceso es gratis o mediante pago.
url	string	Dirección de la revista.
soporte	enum	Formato en que se edita la revista.
lugar	string	Lugar en el que se publica la revista.
email	string	Dirección de contacto de la revista.
resumen	string	Resumen de la revista donde se explican los detalles de la misma.
activa	boolean	Define si la revista aun se edita.
arbitrada	boolean	Define si la revista cuenta con un equipo de árbitros.
frecuencia_id	int	Frecuencia con que se publica la revista.
estado	enum	Estado de la revista en el sistema.
pais_id	int	País donde se publica la revista.

Tabla 38 Descripción de la tabla **tb_drevista**

Nombre: tb_dpublicacion
Descripción: contiene los datos de la publicación.

Módulo de gestión de Publicaciones y Premios

Atributo	Tipo	Descripción
titulo	string	Título de la publicación.
estado	enum	Estado de la publicación en el sistema.

Tabla 39 Descripción de la tabla tb_publicacion

Nombre : tb_dautor_publicacion		
Descripción: Relación entre los autores y las publicaciones.		
Atributo	Tipo	Descripción
nombre	string	Nombre del autor de la publicación.

Tabla 40 Descripción de la tabla tb_dautor_publicacion

Nombre: tb_rrevista_referencia		
Descripción: Relación entre las revistas y los sitios donde está referenciada.		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.

Tabla 41 Descripción de la tabla tb_rrevista_referencia

Nombre: tb_rrevista_idioma		
Descripción: Relación entre la revista y los idiomas en que se publica.		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.
idioma_id	int	Identificador del idioma.

Tabla 42 Descripción de la tabla tb_rrevista_idioma

Nombre: tb_rrevista_tema		
Descripción: Relación entre la revista y los temas que aborda		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.

Módulo de gestión de Publicaciones y Premios

tema_id	int	Identificador del tema.
---------	-----	-------------------------

Tabla 43 Descripción de la tabla tb_rrevista_idioma

Nombre: tb_rrevista_editorial		
Descripción: Relación entre la revista y las editoriales que la publican.		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.
editorial_id	int	Identificador de la editorial.

Tabla 44 Descripción de la tabla tb_rrevista_editorial

Nombre: tb_rrevista_patrocinadores		
Descripción: Relación entre la revista y los patrocinadores.		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.
organizacion_id	int	Identificador de la organización.

Tabla 45 Descripción de la tabla tb_rrevista_patrocinadores

Nombre: tb_rrevista_publicacion		
Descripción: Relación entre la revista y las publicaciones que contiene.		
Atributo	Tipo	Descripción
revista_id	int	Identificador de la revista.
publicacion_id	int	Identificador de la publicación.
volumen	string	Volumen en que fue publicada.
numero	int	Número de la revista.
pagina	int	Número de la página.
estado	enum	Estado de la publicación en la revista.

Tabla 46 Descripción de la tabla tb_rrevista_publicacion

Módulo de gestión de Publicaciones y Premios

Nombre: tb_rpublicacion_repositorio		
Descripción: Relación entre el repositorio y las publicaciones que contiene.		
Atributo	Tipo	Descripción
publicacion_id	int	Identificador de la publicación.
repositorio_id	int	Identificador del repositorio.
fecha	time	Fecha en que fue publicada.
estado	enum	Estado de aprobación de la publicación en el repositorio.

Tabla 47 Descripción de la tabla tb_rpublicacion_repositorio

Nombre: tb_repositorio_tema		
Descripción: Relación entre el repositorio institucional y los temas que aborda.		
Atributo	Tipo	Descripción
repositorio_id	int	Identificador del repositorio.
tema_id	int	Identificador del tema.

Tabla 48 Descripción de la tabla tb_repositorio_tema

Nombre: tb_dreferencia		
Descripción: contiene los datos de la referencia.		
Atributo	Tipo	Descripción
descripción	string	Breve descripción de la referencia.

Tabla 49 Descripción de la tabla tb_dreferencia

Nombre: tb_npais		
Descripción: Tabla de países.		
Atributo	Tipo	Descripción
país	string	Nombre del país.

Módulo de gestión de Publicaciones y Premios

Tabla 50 Descripción de la tabla tb_npais

Nombre: tb_neditorial		
Descripción: Contiene los datos de las editoriales		
Atributo	Tipo	Descripción
nombre	string	Nombre de la editorial.

Tabla 51 Descripción de la tabla tb_neditorial

Nombre: tb_norganizacion		
Descripción: Contiene los datos de los patrocinadores.		
Atributo	Tipo	Descripción
descripción	string	Breve descripción de la organización.

Tabla 52 Descripción de la tabla tb_norganizacion

Nombre: tb_ntema		
Descripción: Contiene los datos de los temas que se abordan en las diferentes fuentes de publicación.		
Atributo	Tipo	Descripción
descripción	string	Breve descripción del tema.

Tabla 53 Descripción de la tabla tb_ntema

Nombre: tb_drepositorio		
Descripción: Almacena información referente al repositorio.		
Atributo	Tipo	Descripción
nombre	string	Nombre del repositorio.
url	string	Dirección del repositorio.
tipo	enum	Tipo de repositorio.

Tabla 54 Descripción de la tabla tb_drepositorio

3.5 Conclusiones

Para comenzar a construir la solución, y de acuerdo a la metodología RUP para el flujo de trabajo de Análisis y Diseño, se han creado los diagramas de interacción correspondientes por cada uno de los casos de uso, facilitando el entendimiento del sistema. El diseño de la solución propuesta implementa una versión personalizada del patrón MVC a través del marco de trabajo Symfony, la cual cumple con los patrones de diseño GRASP y GOP garantizando la organización de las clases dentro del sistema y de la estructura de trabajo. El modelo de clases persistentes se traduce en tablas de la base de datos de las cuales se ha hecho una descripción detallada.

Capítulo 4: Construcción y pruebas de la solución.

4.1 Introducción

La fase de construcción dentro del proceso de desarrollo de software es la que materializa un producto final. Para lograr esto, se basa en el uso de los artefactos generados en flujos e iteraciones precedentes. Este capítulo está dedicado a la implementación y pruebas de la solución propuesta a través de la explicación de cada uno de los componentes y artefactos generados en este flujo de trabajo.

4.2 Diagrama de componentes

Este artefacto representa la forma en que un sistema es dividido en partes que se denominan componentes, y que muestra la organización y dependencias lógicas entre elementos de software, sean estos código fuente, binarios o ejecutables. Pueden ser utilizados para modelar y documentar la arquitectura, la vista estática y el código fuente, versiones ejecutables y bases de datos físicas del sistema, entre otros.

En el diagrama realizado se muestra como están distribuidos los componentes según el patrón arquitectónico MVC que utiliza Symfony en su organización interna. En el paquete Vista se agrupan el elemento decorador y el parcial que se genera en la lógica de la vista. El paquete controlador contiene el controlador frontal y el secundario que define la acción que se realiza en una petición al servidor. Por último, el componente Modelo contiene los elementos que modelan los datos y el acceso a estos en la base de datos física. Cada uno de estos componentes se subdivide en otros de menor tamaño, con el objetivo de maximizar la reutilización de código, el bajo acoplamiento y la modularidad del sistema.

Módulo de gestión de Publicaciones y Premios

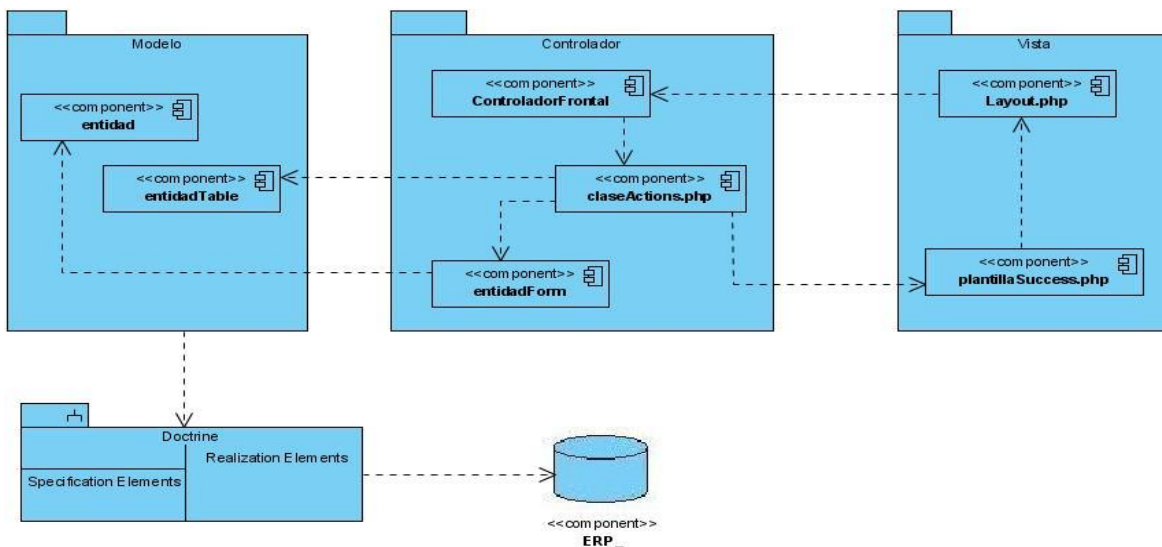


Fig. 14 Diagrama de componentes

4.3 Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama UML que se emplea para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. El modelo de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

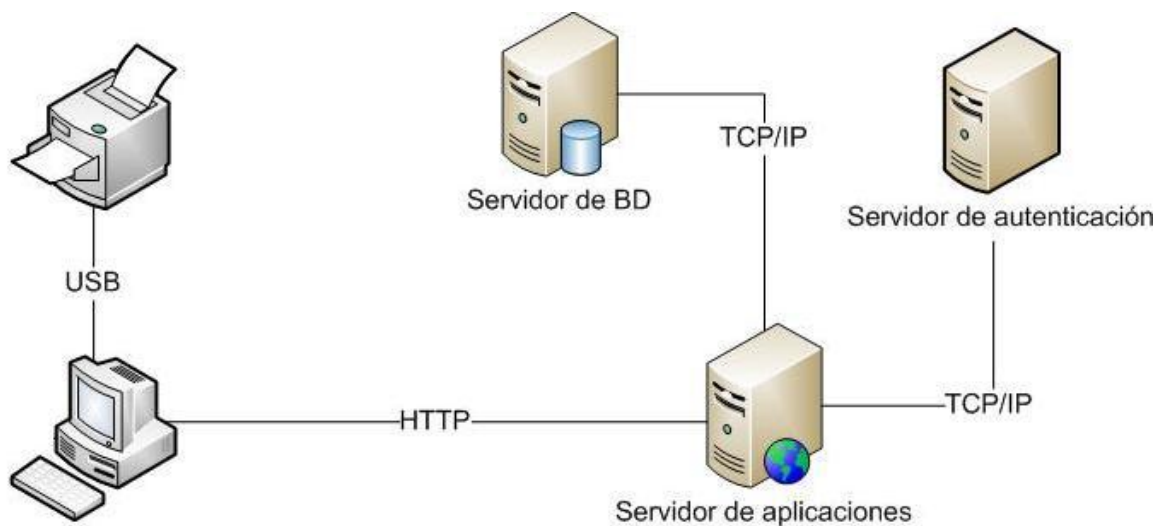


Fig. 15 Diagrama de despliegue

- **PC Cliente:** Estación de trabajo desde donde interactúa el usuario con la aplicación.
- **Servidor Web Apache + PHP:** Contiene la aplicación Web.
- **Servidor BD PostgreSQL:** Servidor donde se almacenan los datos de la aplicación.
- **Impresora:** Dispositivo que permite imprimir las informaciones generadas en la aplicación.

- **Servidor de Autenticación:** Servidor de autenticación.

4.4 Pruebas

El marco de trabajo Symfony permite realizar dos tipos de pruebas: unitarias y funcionales.

Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de prueba valida la forma en que las funciones trabajan en un determinado ambiente y se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su comportamiento varía en función del contexto.

Las pruebas funcionales son la mejor forma de probar una aplicación completa, desde la petición realizada por un navegador, hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas de la aplicación: el sistema de enrutamiento, el modelo, las acciones y las plantillas.

Las pruebas se guardan en el directorio *test/* del proyecto. Este contiene a su vez dos subdirectorios, uno para las pruebas unitarias (*test/unit/*) y otro para las pruebas funcionales (*test/functional/*).

Por su parte, RUP trae consigo su propia forma de describir los casos de prueba por cada caso de uso, de manera que el desarrollador de las pruebas debe llenar una tabla como la que se muestra a continuación: con los datos de conformidad de la prueba, así como la descripción detallada de la misma.

Caso de uso: Gestionar revistas.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Detalles de la revista.	Normal	
2	Registrar revista.	Normal	
3	Editar revista.	Normal	
4	Eliminar revista.	Normal	

Tabla 55 Descripción de escenarios del CU Gestionar revista

ID	Escenario	Datos	Resultado esperado
CP 1	1	V	El sistema debe mostrar la página de la revista especificada en el parámetro "id" de

Módulo de gestión de Publicaciones y Premios

			la petición.
CP 2	2	V	Si los datos son correctos tras la validación, se insertan los datos de la revista en la base de datos y se redirecciona hacia la página de detalles de la revista.
CP 3	3	V	Se escoge la revista que se desea editar y se muestra el formulario con los datos de la misma. Al enviar, si los datos son correctos, se actualizan los datos de la revista en la base de datos y se redirecciona hacia la página de detalles de la revista.
CP 4	4	V	Al hacer clic en el botón “eliminar” de la revista, se elimina esta de la base de datos y se actualiza el listado de las revistas.

Tabla 56 Descripción de los casos de prueba del CU Gestionar revista

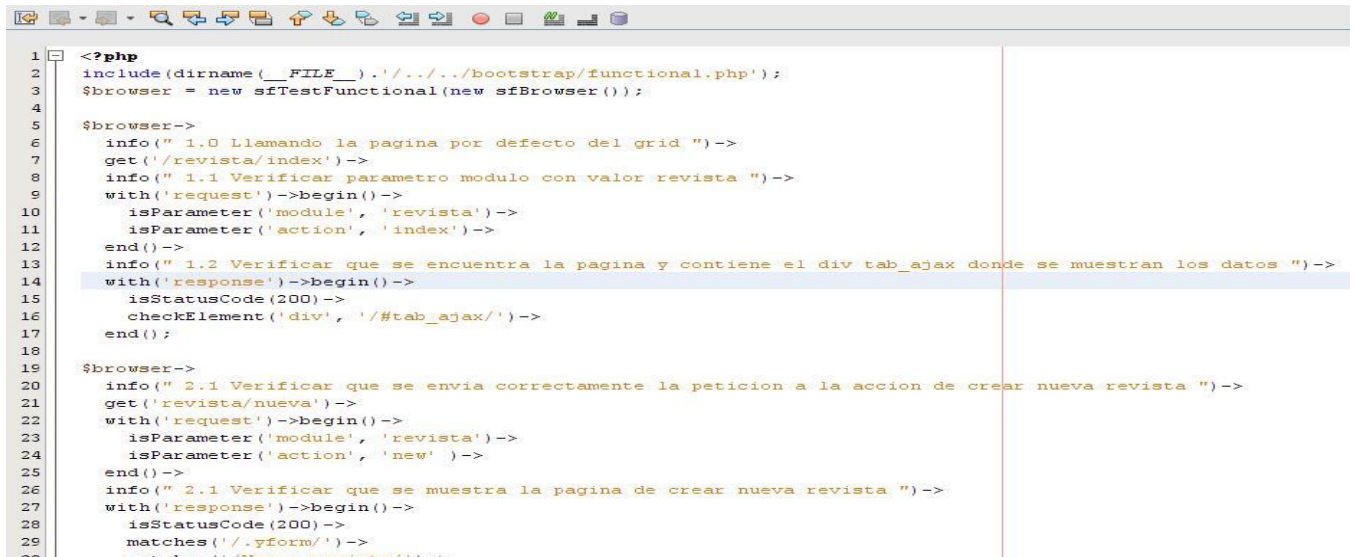
ID	Escenario	Datos	Resultado esperado
CP 1	1	Módulo: revista. Acción: show. Id: Identificador de la revista.	El sistema debe mostrar la pagina de la revista especificada en el parámetro “id” de la petición.
CP 2	2	Módulo: revista. Acción: nueva.	Si los datos son correctos tras la validación, se insertan los datos de la revista en la base de datos y se redirecciona hacia la página de detalles de la revista.
CP 3	3	Módulo: revista. Acción: editar. Id: Identificador de la revista.	Se escoge la revista que se desea editar y se muestra el formulario con los datos de la misma. Al enviar, si los datos son correctos, se actualizan los datos de la revista en la base de datos y se redirecciona hacia la página de detalles de la revista.
CP 4	4	Módulo: revista.	Al hacer clic en el botón “eliminar” de la

Módulo de gestión de Publicaciones y Premios

		Acción: eliminar. Id: Identificador de la revista.	revista, se elimina esta de la base de datos y se actualiza el listado de las revistas.
--	--	---	---

Tabla 57 Descripción de los casos de prueba del CU Gestionar revista (con datos)

Por su parte, Symfony trae consigo su propio marco de trabajo dedicado exclusivamente a las pruebas funcionales y unitarias llamado *Lime*, el cual es una herramienta que prueba el funcionamiento desde el lado del servidor, con una sintaxis de declaración bastante legible y sencilla. Este instrumento de Symfony no sustituye otras herramientas más avanzadas de desarrollo de pruebas como *Selenium* que permite pruebas que incluyen interacciones Ajax. Sin embargo, se puede integrar fácilmente con la descripción establecida por RUP. El código y resultado de las pruebas descritas anteriormente se puede observar en las Fig. 16, y 17, escritas con la sintaxis establecida por *Lime* para la escritura de pruebas funcionales y unitarias.



```
1 <?php
2 include(dirname(__FILE__) . '/../bootstrap/functional.php');
3 $browser = new sfTestFunctional(new sfBrowser());
4
5 $browser->
6   info(" 1.0 Llamando la pagina por defecto del grid ")->
7   get('/revista/index')->
8   info(" 1.1 Verificar parametro modulo con valor revista ")->
9   with('request')->begin()->
10    isParameter('module', 'revista')->
11    isParameter('action', 'index')->
12  end()->
13  info(" 1.2 Verificar que se encuentra la pagina y contiene el div tab_ajax donde se muestran los datos ")->
14  with('response')->begin()->
15    isStatusCode(200)->
16    checkElement('div', '/#tab_ajax/')->
17  end();
18
19 $browser->
20   info(" 2.1 Verificar que se envia correctamente la peticion a la accion de crear nueva revista ")->
21   get('/revista/nueva')->
22   with('request')->begin()->
23     isParameter('module', 'revista')->
24     isParameter('action', 'new')->
25   end()->
26   info(" 2.1 Verificar que se muestra la pagina de crear nueva revista ")->
27   with('response')->begin()->
28     isStatusCode(200)->
29     matches('/.yform/')->
30     matches('/#nueva_revista/')
```

Fig. 16 Fragmento de código de las pruebas funcionales del CU Gestionar revista

Módulo de gestión de Publicaciones y Premios

```
<?php
include(dirname(__FILE__).'/../bootstrap/Doctrine.php');

$t = new lime_test(1);
$t->comment('Pruebas unitarias de la clase PropuestaResultado');

$p = createPropuestaResultado();
$t->comment("Probando el metodo save()");
$p->save();

$t->is($p->Edicion->getId(), Doctrine::getTable('EdicionPremio')->findOneByAbierta(true)->getId(), "Probando

function createPropuestaResultado()
{
    $propuesta = new PropuestaResultado();
    $edicion = Doctrine::getTable('EdicionPremio')->findOneByAbierta(true);
    $categoria = Doctrine::getTable('Categoria')->findOneByPremioId($edicion->Premio->getId());
    $campo = Doctrine_Query::create()->from('Campo c')->fetchOne();
    $values = array(
        'edicion_premio_id' => $edicion->getId(),
        'campo_id' => $campo->getId(),
        'valoracion' => "Parrafo de valoracion",
        'resultado' => "Trabajo de mayor aporte economico a la uci",
        'Autores' => Doctrine::getTable('Autor')->findByGradocientificoId(1)
    );
    $propuesta->fromArray($values);

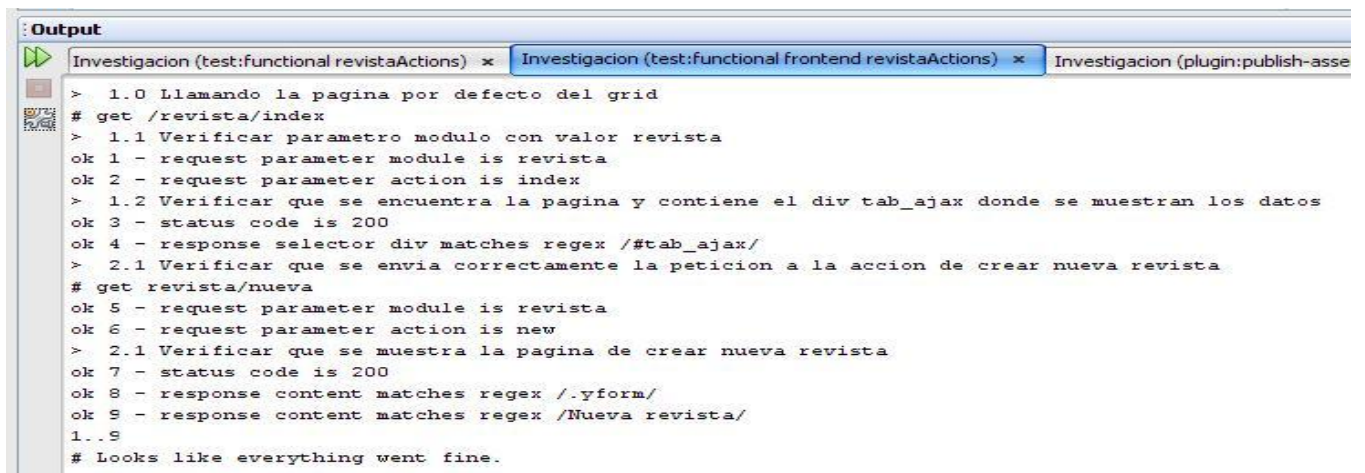
    return $propuesta;
}
```

Fig. 17 Fragmento de código de prueba funcional del método save () de la clase PropuestaResultado

Las pruebas de *Lime* se realizan a través de la consola de comandos del sistema operativo, y básicamente lo que hacen es simular un navegador virtual que envía peticiones al servidor y recibe respuestas de este. Se pueden utilizar varios probadores, entre ellos, los más utilizados son:

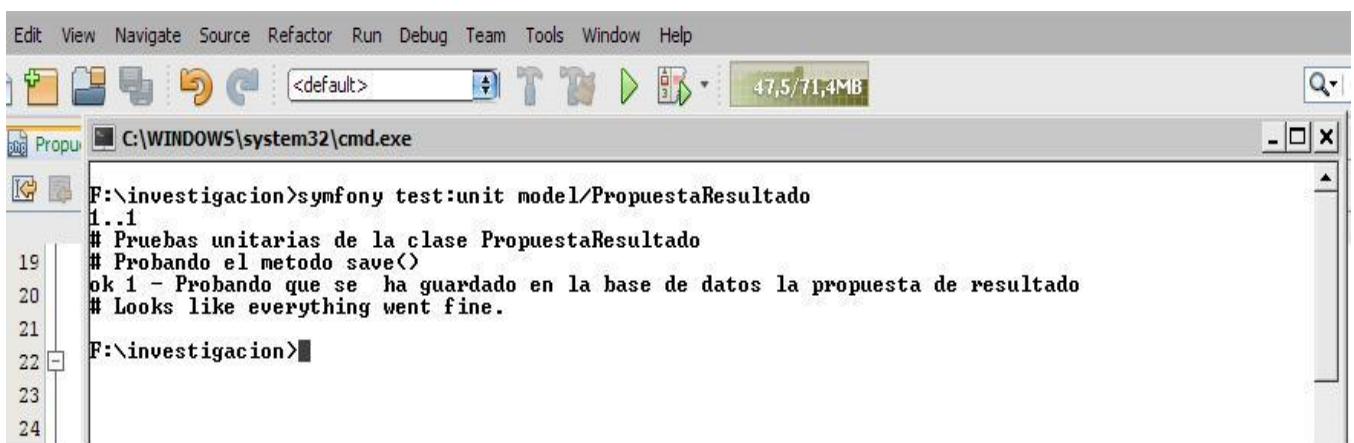
- *sfTestBrowser*: permite trabajar con la petición web y la respuesta.
- *sfTestDoctrine*: permite ejecutar pruebas con valores provenientes desde la base de datos y es proporcionado por el ORM.
- *sfTestUser*: permite probar los datos relacionados con la sesión de usuario, el manejo de *cookies* y otros.
- *sfTestForm*: permite realizar pruebas en los formularios.

La salida de las pruebas en la consola embebida de *NetBeans* luce de la siguiente forma:



```
Output
Investigacion (test:funcional revistaActions) x Investigacion (test:funcional frontend revistaActions) x Investigacion (plugin:publish-asse
> 1.0 Llamando la pagina por defecto del grid
# get /revista/index
> 1.1 Verificar parametro modulo con valor revista
ok 1 - request parameter module is revista
ok 2 - request parameter action is index
> 1.2 Verificar que se encuentra la pagina y contiene el div tab_ajax donde se muestran los datos
ok 3 - status code is 200
ok 4 - response selector div matches regex /#tab_ajax/
> 2.1 Verificar que se envia correctamente la peticion a la accion de crear nueva revista
# get revista/nueva
ok 5 - request parameter module is revista
ok 6 - request parameter action is new
> 2.1 Verificar que se muestra la pagina de crear nueva revista
ok 7 - status code is 200
ok 8 - response content matches regex /.yform/
ok 9 - response content matches regex /Nueva revista/
1..9
# Looks like everything went fine.
```

Fig. 18 Fragmento del resultado de las pruebas funcionales del CU Gestionar revista



```
Edit View Navigate Source Refactor Run Debug Team Tools Window Help
C:\WINDOWS\system32\cmd.exe
F:\investigacion>symfony test:unit model/PropuestaResultado
1..1
# Pruebas unitarias de la clase PropuestaResultado
# Probando el metodo save()
ok 1 - Probando que se ha guardado en la base de datos la propuesta de resultado
# Looks like everything went fine.
F:\investigacion>
```

Fig. 19 Resultado de la prueba del método save () de la clase PropuestaResultado

4.5 Conclusiones

En este capítulo se transitó a través de la fase de construcción de la solución, evidenciando la materialización de los artefactos del diseño en componentes y clases, ejecutables y documentación vitales para el funcionamiento de la solución. El uso de las pruebas unitarias y funcionales facilitó la corrección de errores, evitando la pérdida de tiempo en el desarrollo de la aplicación. Se creó el modelo de despliegue de forma que muestre cómo se distribuye físicamente el sistema sobre una arquitectura de hardware, mientras que el modelo de componentes muestra cómo interactúan los paquetes de clases y las diferentes capas de la aplicación dentro del marco de trabajo.

Conclusiones

Durante el transcurso de esta investigación se realizó un estudio profundo del estado del arte y las condiciones actuales en que se realiza el proceso de control de publicaciones científicas y premios curriculares en la universidad. A partir de este se pudieron conocer algunos de los sistemas más usados y sus características principales, que constituyen antecedentes del sistema desarrollado, y se conoció además que no existen sistemas registrados que se dediquen por completo a la gestión de premios curriculares, y por tanto, la no existencia de una aplicación que logre integrar la gestión de premios y publicaciones en una única herramienta.

Luego de haber analizado minuciosamente los procesos de negocio dentro de la DI, se concluyó que la manera en que se realizan actualmente los procesos relacionados con publicaciones científicas y premios curriculares no permite un adecuado control de la información generado por los mismos. Basándose en el primer prototipo no funcional, se realizó el diseño de la aplicación quedando estructurado completamente el sistema en forma de subsistemas y paquetes, que permitió lograr una mayor organización y reutilización del código fuente.

Completamente diseñado el sistema, se procedió a realizar la implementación y pruebas de la aplicación, utilizando la filosofía del marco de trabajo de probar antes de programar. Se demostró que el uso y escritura de pruebas sobre partes y funciones sensibles de la aplicación aseguraba la escritura de código sin errores, aumentando la consistencia de la implementación.

Teniendo en cuenta que el proceso de control de premios curriculares y publicaciones científicas dentro en la DI no contaba con una herramienta capaz de automatizar y unificar el trabajo, el primer gran aporte de esta investigación científica lo constituye la creación de un sistema completamente funcional capaz de informatizar el flujo de trabajo de la DI.

El desarrollo de esta aplicación se ha basado en el estándar y la arquitectura diseñada por el grupo de trabajo del Sistema de Gestión Universitaria, de manera que un gran aporte de esta investigación es su posible integración, como un complemento, al grupo de aplicaciones desarrolladas para el proceso de informatización de la universidad, y que se encarga de la gestión de procesos relacionados con los premios curriculares y las publicaciones científicas dentro de la DI.

El uso de este sistema informático ahorra tiempo de trabajo y al mismo tiempo contribuye a la disminución de errores en los procesos de registro y control de premios curriculares y publicaciones científicas por parte de los trabajadores de la DI, de manera que aumenta la confiabilidad y la eficiencia en el trabajo.

Módulo de gestión de Publicaciones y Premios

Esta herramienta se ha desarrollado como un complemento que aporta valor agregado a una aplicación más grande que abarca todos los procesos de la Dirección de Investigaciones. Sin embargo, la filosofía de creación de la misma permite usar este complemento como una aplicación enteramente funcional por sí sola, y adaptable a cualquier institución de carácter científico en la cual se manejen premios curriculares y publicaciones científicas.

Durante todo el proceso de creación de esta aplicación se ha generado una gran cantidad de documentación relacionada con los procesos de negocio, implementación y pruebas de la aplicación de manera que todo esto constituye una fuente de conocimiento importante, que puede servir de material de consulta en futuras investigaciones de carácter científico.

La investigación de otros marcos de trabajo, de otras metodologías y el uso de diferentes métodos de investigación científica han creado una experiencia en los autores y tutores, que se puede aprovechar para futuras exploraciones de carácter científico y desarrollador.

Recomendaciones

Después de haber conocido los aportes y resultados más relevantes de esta investigación, se recomienda la puesta en funcionamiento del sistema construido, de manera que pueda ser estudiado y probado en un ambiente de trabajo real.

Se recomienda además la puesta en marcha del proceso de integración total de la solución con el conjunto de aplicaciones que integran el Sistema de Gestión Universitaria, de manera que se adviertan de forma temprana los errores más comunes de este proceso y de esta manera acelerar su corrección.

Bibliografía referenciada

1. **Gilberto Vizcaíno Salazar.** Hacia un nuevo sistema equitativo de publicaciones científicas. [En línea] Junio de 2009. <http://www.scielo.org.ve> .
2. **Perrone, Ignacio.** Las publicaciones científicas e Internet. *Las publicaciones científicas y la noción de campo.* s.l. : Biblos.
3. C2CSoluciones. [En línea] C2C Soluciones, 2009. <http://www.c2csoluciones.com/publicacion/contenido.htm>.
4. Latindex. *Latindex.* [En línea] Universidad Autónoma de México. <http://www.latindex.unam.mx/latindex/>.
5. Sistema de publicaciones. [En línea] Instituto nacional de Ecología, 2010. <http://www.ine.gob.mx>.
6. *Oper Journal System.* [En línea] Public Knowledge Project, 2010. <http://pkp.sfu.ca>.
7. *RAE.* [En línea] Real Academia de la Lengua Española. <http://www.rae.es>.
8. Metodologías de desarrollo de Software. [En línea] Monografías.com. <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software2.shtml>.
9. **Kruchten, Phillipe.** *The Rational Unified Process: An Introduction.* s.l. : Addison-Wesley, 2004. 0-321-19770-4.
10. **Sommerville, Ian.** *Ingeniería de Software.* Madrid : Pearson Education, 2005. 84-7829-074-5.
11. **Arebas, Jesus Barranco de.** *Metodologías del análisis estructurado de sistemas.* Madrid : s.n. 84-8468-043-6.
12. **Ladd, Seth y Davidson, Darren.** *Expert Spring MVC and Web Flow.* s.l. : Appress, 2006.
13. Guía Breve de XHTML. *Guía Breve de XHTML.* [En línea] W3C. <http://www.w3c.es/Divulgacion/Guiasbreves/XHTML>.
14. **Pérez, Javier Eguíluz.** Introducción a Ajax. *LibrosWeb.es.* [En línea] LibrosWeb. <http://www.librosweb.es/ajax/index.html>.
15. Página oficial de PHP. *Sitio Oficial de PHP.* [En línea] <http://www.php.net>.

16. **Potencier, Fabien.** La Guía Definitiva. *Symfony Project*. [En línea] Sencio Labs, 2007. <http://www.symfony-project.org>.
17. PostgreSQL: The world's most advanced open-source database. *PostgreSQL*. [En línea] <http://www.postgresql.org/>.
18. UML. *UML*. [En línea] Object Management Group. <http://www.uml.org>.
19. Apache. *Apache*. [En línea] Apache Foundation. <http://www.apache.org>.
20. Visual Paradigm for UML. *Visual Paradigm*. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.

Bibliografía consultada

1. monografias. [En línea] www.monografias.com/trabajos14/linux.

2. E-LIS. [En línea] http://eprints.rclis.org/12992/1/Jornadas_GRUPO_SIOU.pdf.
3. TecnoRetales. [En línea] <http://www.tecnoretas.com/programacion/que-es-doctrine-orm/comment-page-1/>.
4. **Potencier, Fabien y Zaninotto, Francois.** librosweb. [En línea] www.librosweb.es .
5. Blog Cultura. *Blog Cultura*. [En línea] <http://blogcultura.com/java/netbeans-6-8-liberado/>.

Glosario de términos

CASE: Ingeniería en Computación de Software Asistida (*Computer Aided Software Engineering*).

CSS: siglas de Hojas de Estilo en Cascada, nombre original de las hojas de estilos que se usan en las páginas web. Es un lenguaje artificial usado para definir la presentación de un documento estructurado escrito en HTML o XML.

CUS: Caso de Uso del Sistema.

DOM: Modelo de Objetos del Documento para la representación de documentos es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

DS: Diagrama de Secuencia.

DCDW: Diagrama de Clases del Diseño Web.

ECMAScript: Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudoclasses.

Framework: Marco de trabajo. Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un marco de trabajo (*framework*) puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros tipos de software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GOF: *Gang of Four*, expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software.

GRASP: Patrones Generales de Software para Asignación de Responsabilidades.

Linux: Sistema operativo, multitarea, multiusuario, multiplataforma y multiprocesador.

Lime: utilidad para las pruebas unitarias, desarrollado por el equipo de *Symfony*.

MVC: Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web.

NCSA: Servidor web desarrollado originalmente en el Centro Nacional de Aplicaciones de Supercomputación.

Módulo de gestión de Publicaciones y Premios

ORM: Mapa Objeto-Relacional , técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad.

RUP: Proceso Unificado de Desarrollo, es un proceso de desarrollo de software, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SCP: Sistemas de Control de Publicaciones.

SGPs: Sistemas de Gestión de Premios.

UML: Lenguaje Unificado de Modelado, lenguaje visual para especificar, construir y documentar un software.

Unix: sistema operativo portable, multitarea y multiusuario

XML: Lenguaje de Marcado Extensible, es un metalenguaje extensible de etiquetas.