



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

**Módulo “Mérito Científico” para el Sistema de Certificación de
Roles en la Disciplina de Práctica Profesional**

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

Autor: Leonardo Torres Silega

Tutores: DrC. Yoan Martínez Márquez
Ing. Haydee Fernández Díaz
Ing. Felinda Rosabel León Mendoza
Ing. Eliodanis Maceo Rosales

La Habana, junio de 2016

“Año 58 de la Revolución”



“La Revolución no se lleva en los labios para vivir de ella, se lleva en el corazón para morir por ella”.

Ernesto Guevara de la Serna

DECLARACIÓN JURADA DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Leonardo Torres Silega

Autor

DrC. Yoan Martínez Márquez

Tutor

Ing. Haydee Fernández Díaz

Tutor

Ing. Felinda León Rosabel Mendoza

Tutor

Ing. Eliodanis Maceo Rosales

Tutor

DEDICATORIA

Quiero dedicar este trabajo a mi mamá que siempre ha soñado por el momento en que su hijo logre un título de profesional, a mis tías y mis tíos que siempre estuvieron preocupados por mí, a mi hermana que quiero tanto, a mi papá, a mis abuelos que siempre me tiene presente y rezan por mí, a todas mis amistades y mis compañeros de aula, a mis profesores, a las personas que de una forma u otra compartieron conmigo diferentes espacios en festivales de artistas aficionados, presentación de trabajos, en fin, a todas las personas que de una forma u otra aportaron un granito de arena a mi formación profesional.

Leonardo Torres Silega

En la vida te das cuenta de lo que en verdad es importante y de las personas que te rodean si son honestas o no lo son. Es por eso que le doy muchas gracias a Dios por haberme dado una familia estupenda que me acompañó día a día en este camino.

Agradezco a todas aquellas personas que de una forma u otra han contribuido a mi formación profesional en especial a:

Mi querida Madre: *Mujer incansable, padre, madre y amiga en todo momento, por siempre estar pendiente de todo lo que necesito, por ser una persona fuerte ante cualquier problema y aún con lágrimas en los ojos se las arregla para decir con una sonrisa “Estoy Bien”, y sé que en estos momentos está muy orgullosa de mí.*

A mí papá: *por darme tan buenos consejos, aunque no estuvo muchas veces cuando lo necesité. A mis queridos abuelitos María y José Silega, siempre diciéndome rodéate de buenas personas y sé una de ellas. A todos mis tíos, que para mí son los mejores, a mi tío Georkis, Grabiél, Cleidis, en general a toda mi familia que de una forma u otra han sabido guiarme por el camino correcto.*

A todos mis amigos, en especial a los de la universidad.

A mis tutores, que me ayudaron enormemente a dar un paso firme en el desarrollo de la tesis y siempre estaré eternamente agradecido.

A todos mis compañeros de aula a Yesenia, Yamila, Paúl, Juan Carlos, Luis Manuel, Manuel Arias, Alejandro, a todos mis profesores que les debo mucho más de lo que creo. Gracias por valorarme y dedicarme su más preciado tiempo para que en el mañana me convirtiese en un ingeniero, gracias por encaminarme por las buenas sendas de la vida, por ofrecerme sus mejores saberes. Ustedes son los que se merecen las mejores dedicatorias.

A varios de mis amigos los conocí en la universidad, gracias por hacer mis días en la universidad mucho más felices e inolvidables, a esas personas agradezco que se tomaron la molestia de gastar un poco de su tiempo para explicarme las cosas que no entendía, me ayudaron en mis trabajos a la hora de elaborarlos, dándome consejos de lo que podía cambiar. Hoy en día los amigos son algo muy raro y escaso porque todas las personas son interesadas, le agradezco a Dios por mandarme unos amigos honestos y sinceros que no buscan nada a cambio, solo buscan ayudar a los demás para poder superarse en la vida.

Sin nada más que decir les deseo suerte a todos y que Dios bendiga cada uno de nuestros caminos, nos llene de fortalezas y virtudes y si el destino lo quiere, nos reencontremos mucho más adelante

RESUMEN

El presente trabajo de diploma se basa en el desarrollo de un módulo, en el Sistema de Certificación de Roles 2.0 para el Mérito Científico en la Universidad de las Ciencias Informáticas (UCI). La resolución 79/2015 planteada por el Ministerio de Educación Superior (MES) mide los indicadores referentes al Mérito Científico en cuanto a publicaciones científicas, eventos científicos, exámenes de premio y certificación de rol. El Sistema para la Certificación de Roles, actualmente almacena la información de tres de los cuatros indicadores que se miden para el Mérito Científico. Sin embargo, no cuenta con un sistema informático que permita el control y análisis de las evidencias y bonificaciones alcanzada por los estudiantes. El objetivo de la siguiente investigación, es desarrollar el módulo “Mérito Científico” para la identificación de los estudiantes potenciales a obtener la distinción en el Sistema de Certificación de Roles 2.0. Luego de una revisión bibliográfica en aras de identificar los principales conceptos relacionados con la temática a tratar, se identificaron elementos importantes para la selección de la arquitectura del sistema, metodología, lenguajes, herramientas y un conjunto de artefactos usados como apoyo para darle solución al problema identificado. Para acelerar el proceso de desarrollo web, hasta la fase de implementación se utilizó un enfoque ágil, mostrando los resultados del diseño en cada una de las fases de la metodología seleccionada. Se realizan pruebas al sistema que verifican su correcto funcionamiento y se demuestra que la solución propuesta resuelve las necesidades del proceso para la gestión del premio al Mérito Científico.

Palabras claves: bonificaciones, eventos, gestión, resolución

RESUMEN	VI
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO “MÉRITO CIENTÍFICO”	6
Introducción	6
1.1. Conceptos fundamentales.....	6
1.2. Escenario actual del proceso	8
1.3. Metodología, herramientas y tecnologías de desarrollo.....	9
1.3.1. Metodología de Desarrollo de Software	9
1.3.2. Marco de trabajo	11
1.3.3. Lenguajes de programación.....	12
1.3.4. Herramientas de base de datos	14
1.3.5. Servidor web	15
1.4. Patrones.....	16
1.4.1. Patrón arquitectónico	16
1.4.2. Patrones de Diseño.....	17
1.5. Métricas de validación.....	17
1.5.1. Métricas para el diseño	17
1.6. Pruebas.....	20
1.6.1. Pruebas de Caja Blanca.....	20
1.6.2. Pruebas de Caja Negra.....	21
1.6.3. Pruebas de aceptación.....	22
1.7. Conclusiones parciales	23
CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	24
Introducción	24
2.1. Descripción de los procesos del negocio.....	24
2.2. Propuesta de solución.....	27
2.2.1. Actores del sistema.....	28

2.3.	Exploración	29
2.3.1.	Historias de usuario	29
2.4.	Planificación	30
2.4.1.	Estimación de esfuerzo por historia de usuario	30
2.4.2.	Lista de reserva del producto	32
2.5.	Desarrollo de iteraciones.....	33
2.6.	Diseño de la solución	35
2.6.1.	Arquitectura del sistema.....	35
2.6.2.	Tarjetas CRC	37
2.7.	Codificación de la solución.....	38
2.7.1.	Tareas de ingeniería	38
2.7.2.	Estándares de Codificación.....	39
2.8.	Patrones de diseño utilizados.....	41
2.8.1.	Patrones GRASP	41
2.8.2.	Patrones GoF.....	41
2.9.	Conclusiones parciales	42
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.....		43
Introducción		43
3.1.	Validación del diseño	43
3.1.1.	Relaciones entre clases (RC).....	43
3.1.2.	Tamaño operacional de clases TOC	46
3.2.	Verificación del sistema.....	48
3.2.1.	Nivel de unidad	49
3.2.2.	Pruebas de caja blanca.....	49
3.2.3.	Prueba de caja negra	52
3.3.	Validación del sistema.....	54
3.3.1.	Nivel de aceptación.....	54

3.4. Conclusiones parciales 57

CONCLUSIONES GENERALES 58

RECOMENDACIONES 59

REFERENCIAS..... 60

Tabla 1: Diferencias entre Metodologías ágiles y no ágiles Fuente: (Schenone, 2004)	10
Tabla 2: Clasificación de valores Fuente: (Fornaris, 2009)	18
Tabla 3: Rango de valores para la evaluación de los atributos de calidad con la métrica TOC Fuente: (Fornaris, 2009).....	18
Tabla 4: Rango de valores para los atributos de calidad relacionados con la métrica RC. Fuente: (Fornaris, 2009).....	19
Tabla 5: Resultados de bonificación por Niveles. Fuente: (Gaceta, 2014)	26
Tabla 6: Resultados de bonificación por Publicaciones Científicas. Fuente: (Gaceta, 2014)	26
Tabla 7: Actores del sistema y su descripción. Fuente: (Elaboración propia)	28
Tabla 8: historia de usuario 15. Fuente (Elaboración propia).....	29
Tabla 9: Prioridad para el cliente por historia de usuario. Fuente (Elaboración propia)	31
Tabla 10: Lista de reserva del producto (Requisitos no funcionales).	32
Tabla 11: Plan de duración de las iteraciones. Fuente: (Elaboración propia).....	34
Tabla 12: Plan de duración de las iteraciones. Fuente (Elaboración propia).....	35
Tabla 13: Tarjeta CRC número 1. Fuente: (Roberth, y otros, 2007)	37
Tabla 14: HU Tarea de ingeniería Adicionar bonificación a un evento. Fuente (Elaboración propia) ..	38
Tabla 15: Métrica RC para las clases principales del sistema. Fuente (Elaboración propia).....	43
Tabla 16: Resultado para el Acoplamiento. Fuente (Elaboración propia)	44
Tabla 17: Resultado para la cantidad de prueba y Complejidad de mantenimiento. Fuente (Elaboración propia).....	44
Tabla 18: Resultado para la Reutilización. Fuente (Elaboración propia).....	45
Tabla 19: Tamaño de Clases. Fuente (Elaboración propia)	46
Tabla 20: Cantidad de clases por clasificación. Fuente (Fornaris, 2009)	47
Tabla 21: Caso de prueba para el camino 3. Fuente (Elaboración propia)	52
Tabla 22: Caso de prueba para gestionar bonificación Fuente (Elaboración propia)	53
Tabla 23: Caso de prueba de aceptación. Fuente (Elaboración propia)	55

Figura 1 Arquitectura del sistema Fuente: (Velthunis, 2010)	36
Figura 2 Patrón MVC. Fuente (Potencier, 2010).....	37
Figura 3: Estructura de carpetas Fuente: (Elaboración propia).....	40
Figura 4: Fragmento de código, muestra el uso de llaves Fuente: (Elaboración propia)	41
Figura 5: Resultado de la métrica RC. Fuente (Elaboración propia)	45
Figura 6: Resultado de la métrica TOC. Fuente (Elaboración propia).....	48
Figura 7: Fragmento de código perteneciente al método createStaticAction. Fuente: (Elaboración propia).....	50
Figura 8: Grafo de flujo del fragmento de código de la Figura 7 Fuente: (Elaboración propia).....	51
Figura 9: Cantidad de No Conformidades (NC) por iteraciones Fuente: (Elaboración propia)	54
Figura 10: Pruebas de aceptación Fuente: (Elaboración propia)	56

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) ofrecen un amplio marco de posibilidades en la informática. Esto trae como beneficio un aumento de la productividad y una mejor gestión de las empresas u organizaciones. Gracias al creciente y rápido avance de las tecnologías existen medios, sistemas y aplicaciones que tratan temas relacionados con las notificaciones, avisos, y eventos. El aprovechamiento de estas tecnologías ha impactado contextos referentes a las noticias, el deporte, la cultura, la medicina y la educación, como principales áreas de prioridad en Cuba por su carácter social.

Con el uso de las TIC se han podido desarrollar conocimientos en las diferentes esferas de la investigación. Varias son las empresas u organizaciones que lograron informatizar sus procesos mediante aplicaciones. Cuba no está exenta de estos cambios y ha venido realizando estrategias para lograr la informatización de la sociedad, además ha enfocado sus esfuerzos en el desarrollo científico, técnico e innovador. La Universidad de las Ciencias Informáticas, fue creada con el objetivo de lograr la informatización de la sociedad y resolver problemas existentes en disímiles esferas del ámbito social. En esta universidad los estudiantes se forman desde la producción, investigación y desarrollo de software, garantizando una amplia capacidad investigativa que facilita las condiciones necesarias para el desempeño de diferentes roles productivos.

Conforme a la resolución 79/2015 del Ministerio de Educación Superior, resulta necesario incentivar a los profesionales recién graduados, destacados por sus resultados en trabajos científicos y exámenes de premio, mediante un sistema que integre las bonificaciones obtenidas en actividades, tales como: eventos científicos, concursos de certificación de rol, exámenes de premio y publicaciones científicas. Es por ello que en el curso 2014-2015, se realizó la tesis de diploma, Sistema de Gestión para la Certificación de Roles 2.0. La misma se dirigía a: perfeccionar la ponderación establecida por competencia en cada rol a certificar, así como la combinación de dos o más roles en un mismo plan de formación.

En el Sistema de Gestión para la Certificación de Roles 2.0, una vez que se evalúan las tareas realizadas por un estudiante, se brinda la opción de especificar si esta tributa o no, al Mérito Científico. Teniendo en cuenta las tareas evaluadas en el sistema de Certificación de Rol 2.0, se cuenta con tres de los cuatro indicadores que maneja la resolución para el Mérito Científico: la participación y/o premios en eventos científicos, las publicaciones científicas y la certificación de roles. Por esta razón se decidió aprovechar estos indicadores para incorporar al sistema anterior la gestión de evidencias del Mérito Científico.

Este proceso de gestión del Mérito Científico, se realiza de forma manual, archivándose toda la información en formato duro y en el mejor de los casos en documentos Word u hojas de cálculo, ocasionando así una mala gestión de las evidencias, bonificaciones y en ocasiones distorsión en la información. Muchas veces el estudiante certificaba algún rol y no tenía conocimientos previos de la bonificación que este le proporcionaba para la distinción al Mérito Científico. Igualmente existían convocatorias que no tenían alto nivel científico y se bonificaban. Los datos de los indicadores correspondientes al Mérito Científico que almacena el Sistema de Gestión para la Certificación de Roles 2.0, se encuentran dispersos, resultando engorroso el trabajo a desarrollar por el Profesor Principal de Año y para el propio estudiante interesado en optar por el premio al Mérito Científico.

Teniendo en cuenta la **situación problemática** explicada anteriormente, se plantea el siguiente **problema** a resolver: ¿Cómo identificar los estudiantes potenciales a obtener la distinción “Mérito Científico” en el Sistema de Certificación de Roles 2.0 para la Práctica Profesional?

El presente trabajo tiene como **objeto de estudio**: el mantenimiento de software en ámbitos académicos; siendo el **campo de acción** el mantenimiento perfectivo de software.

Para dar solución a este problema se traza el siguiente **objetivo general**: desarrollar el módulo “Mérito Científico” para la identificación de los estudiantes potenciales a obtener la distinción en el Sistema de Certificación de Roles 2.0 para la disciplina de Práctica Profesional.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación mediante un estudio de los referentes teóricos de la gestión académica basada en evidencias, metodología de desarrollo, técnicas y herramientas de software.
2. Realizar el análisis y diseño del módulo “Mérito Científico” para la identificación de los estudiantes potenciales a obtener esta distinción, en el Sistema de Certificación de Roles 2.0, mediante la metodología de desarrollo de software seleccionada.
3. Implementar el módulo “Mérito Científico” para la identificación de los estudiantes potenciales a obtener la distinción, Mérito Científico en el Sistema de Certificación de Roles 2.0, mediante el uso de las herramientas y tecnologías seleccionadas.
4. Valorar la efectividad de la solución propuesta mediante pruebas de software para dar cumplimiento al objetivo de la investigación.

Para el cumplimiento de los objetivos propuestos se tienen en cuenta las siguientes **tareas**:

- Revisión de los referentes teóricos de los sistemas de gestión académica basados en evidencias
- Selección de la metodología y herramientas a utilizar para el desarrollo del sistema que gestionará el Mérito Científico
- Análisis del sistema para determinar los requisitos funcionales y no funcionales
- Realización del modelo de diseño del módulo “Mérito Científico”
- Confección del modelo de implementación del módulo “Mérito Científico”
- Ejecución de pruebas de caja blanca al código fuente obtenido en el módulo “Mérito Científico” del Sistema de Gestión de Certificación de Roles 2.0
- Valoración de la efectividad del módulo “Mérito Científico” mediante las pruebas de caja negra para comprobar su correcto funcionamiento

Posibles resultados

- El módulo “Mérito Científico” para la identificación de los estudiantes potenciales a obtener la distinción al Mérito Científico en el Sistema de Certificación de Roles 2.0 para la disciplina de Práctica Profesional.

En la actividad científica existe una unión inseparable entre el método, la teoría y la práctica, pues los conocimientos se obtienen usando un método científico a partir de los datos de la práctica y la teoría precedente, se integran en una nueva teoría explicativa que regresa a la práctica para su comprobación, contribuyendo al mismo tiempo a su transformación. (Castellanos, 1997) (Hernández, 2002).

Para la búsqueda de información se emplearon los siguientes métodos científicos de la investigación:

Métodos Teóricos:

Histórico-Lógico: Se basan en el estudio histórico del fenómeno, ponen de manifiesto la lógica interna de su desarrollo, de su teoría y el conocimiento más profundo de su esencia. Estos métodos expresan en forma teórica la esencia del objeto, explican la historia de su desarrollo, reproducen el objeto en su forma superior y permiten unir el estudio de la estructura del objeto de investigación con su concepción

histórica (Hernández, 2002). Este método permitió analizar la trayectoria completa del proceso de Mérito Científico (fenómeno) y su funcionamiento en los diferentes años desde su creación, revelando las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales.

Analítico-sintético: Permite el procesamiento de la información y arribar a conclusiones prácticas y teóricas de la investigación (León, 2011). Mediante este método se realizó un análisis de los documentos, las publicaciones, resoluciones, bibliografías y en general toda la información relacionada con la bonificación de los eventos científicos y el empleo de sistemas de gestión académica basados en evidencias. Se realizó un estudio sobre las tendencias actuales, posibilitando hacer una síntesis de estas principales fuentes y llegar a conclusiones parciales sobre el tema en estudio

Métodos Empíricos

Técnica de Entrevista: La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información valiosa. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos por lo que es importante una buena comunicación. Los resultados a lograr en la misión dependen del nivel de comunicación entre el investigador y los participantes en la misma. (León, 2011).

Se aplicó en encuentros sostenidos con la asesora de la Secretaría Docente de la Facultad 3. La técnica de entrevista permitió recopilar información referente al proceso de conformación del expediente para la obtención del Mérito Científico en la Universidad de las Ciencias Informáticas. Además, ayudó a contribuir al entendimiento de este negocio, y así validar las funcionalidades identificadas para el desarrollo de la solución.

El presente documento está estructurado en tres capítulos, y a continuación, se expone una breve descripción de los mismos.

Capítulo 1. Fundamentación teórica del módulo “Mérito Científico”: abarca todo lo relacionado con la fundamentación teórica que sustenta la presente investigación, el estado del arte, así como de las herramientas, lenguajes, gestor de base de datos a utilizar y metodología a emplear.

Capítulo 2. Diseño e implementación de la propuesta de solución: en este capítulo se realiza la descripción del comportamiento del sistema a través de las Historias de Usuario. Como parte del diseño se define el patrón de arquitectura y se identifican los patrones de diseño que se van a utilizar. Posteriormente se confeccionan las Tarjetas CRC (Clase-Responsabilidad-Colaborador) que contienen

el nombre de las clases, las responsabilidades y sus colaboradores. Luego se elaboran las tareas de ingeniería y se determinan los estándares de codificación para llevar a cabo la implementación del sistema.

Capítulo 3. Validación de resultados: en este capítulo se realizan validaciones de los resultados que se obtuvieron del sistema en su ejecución, haciendo uso de algunos de los métodos definidos en el marco teórico de la investigación que permiten establecer comparaciones entre los resultados obtenidos. Además, se valida que el diseño realizado cumpla con la calidad requerida y que el sistema implementado satisfaga las necesidades del cliente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO “MÉRITO CIENTÍFICO”

Introducción

En el siguiente capítulo se hace un estudio del estado del arte relacionado con los conceptos fundamentales que sustentan la investigación sobre el módulo Mérito Científico. Una vez identificados los conceptos fundamentales, se describe el proceso para la gestión del Mérito Científico, se selecciona la metodología de software a emplear en la solución propuesta, el marco de trabajo, lenguajes de programación y herramientas que permitirán el desarrollo de la investigación. Además, se identifican los patrones arquitectónicos y de diseño, las métricas de validación para el diseño y finalmente se describen las pruebas de software a emplear en el desarrollo del módulo Mérito Científico.

1.1. Conceptos fundamentales

La educación y formación de los estudiantes, es un proceso ininterrumpido que se inicia en la edad temprana, continúa en los diferentes niveles de la enseñanza y se mantiene como una necesidad durante toda la vida. Tal necesidad está siempre en constante desarrollo con las apremiantes exigencias sociales y la renovación constante de la información. Seguidamente se exponen los conceptos identificados, en el proceso de gestión del Mérito Científico.

Gestión académica: tiene su razón de ser en la gestión y tramitación de los procesos, funciones y procedimientos administrativos orientados a la vida académica del estudiante. Las principales tareas que lleva a cabo están relacionadas con el acceso, la admisión de los estudiantes a los diferentes niveles de los estudios universitarios, la matriculación de los mismos, la gestión de las becas y ayudas, las convalidaciones y reconocimientos de asignaturas y créditos (León, 2016).

Mantenimiento de software académico: el mantenimiento de software es la modificación de un producto de software después de la entrega, para corregir errores, mejorar el rendimiento, u otros atributos. Siendo el mantenimiento del software una de las actividades más comunes en la ingeniería de software (Calleja, 2010).

Evidencias: Una evidencia es una muestra verificada y certera obtenida en una investigación. El término en sí puede ser más genérico, es decir, adaptado a muchos campos de la vida cotidiana y el estudio científico. (General, 2015). Las evidencias constituyen en la realización del trabajo,

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO “MÉRITO CIENTÍFICO”

un aspecto fundamental en la gestión de la participación en eventos, ya que, a través de estas evidencias obtenidas por los estudiantes, permiten la conformación de bonificaciones.

Galardón: Premio o recompensa de los méritos o servicios. (Real Academia Española, 2015).

Mérito Científico: Consiste en un diploma acreditativo emitido por el Rector(a) y para su otorgamiento se tendrá en cuenta el acumulado de puntos que alcanzó el estudiante recogido en su expediente (Gaceta, 2014).

Investigación científica: La investigación es considerada una actividad humana orientada a la obtención de nuevos conocimientos y su aplicación para la solución a problemas o interrogantes de carácter científico. Hace referencia al procedimiento de reflexión, de control y de crítica que funciona a partir de un sistema, y que se propone aportar nuevos hechos, datos, relaciones o leyes en cualquier ámbito del conocimiento científico (Investigación, 2015).

Gestión: Se entiende por gestión, a toda acción y efecto de administrar o gestionar un negocio. A través de la cual se llevarán a cabo diligencias y trámites que conducirán al logro de objetivos. La misma podrá estar orientada a resolver un problema específico, a concretar un proyecto, pero también puede estar involucrada en la dirección y administración que se realiza en una empresa (Drucker, 1954).

Evento: Proviene del latín *eventus* y de acuerdo al diccionario de la Real Academia Española (RAE), un evento es un suceso importante y programado, de índole social, académica, artística o deportiva (RAE, 2010).

Es una actividad especial que requiere ser planeada, ejecutada, controlada y encaminada a lograr objetivos propuestos, bien sea de índole empresarial, asociativa o particular (INC, SCRIBD, 2013b).

Eventos nacionales: son los eventos que reúnen asistentes de diferentes provincias o estados de un mismo país (CESAR, 2012).

Eventos internacionales: son los eventos que reúnen asistentes de varios países y regiones. Por otra parte, para que sean de carácter internacional debe intervenir la decisión de una asamblea general ordinaria del organismo internacional que lo convoca (CESAR, 2012).

Una vez relacionados los conceptos fundamentales relativos al proceso del Mérito Científico, es preciso conocer las características del mismo para comprender la necesidad de contribuir a su mejora, a continuación, se resume la situación actual de este proceso.

1.2. Escenario actual del proceso

El premio al Mérito Científico se otorga, al culminar los estudios, donde se involucran los estudiantes desde la entrada en su primer año hasta el cierre de su carrera. Este proceso hoy en día se realiza de forma manual y comienza con la convocatoria de diferentes eventos. La primera información emitida la hace llegar la facultad, cuando informa la convocatoria de un evento próximo. Es aquí donde el profesor principal del año, es el encargado de enviar a sus estudiantes por vía correo una notificación informándoles cuáles son las actividades que tributan al Mérito Científico. Luego el estudiante es el encargado de gestionar su participación. Como paso inicial, debe de solicitar participar en el evento del cual fue lanzada la convocatoria realizando la inscripción. Antes de inscribirse, se le leen algunos requisitos a cumplir para poder participar en el evento, en dependencia del evento si es externo o interno.

Luego que el estudiante participa en el evento y obtiene un resultado, esta bonificación debe de llegar a la facultad donde pertenece el estudiante, para así registrar y hacer constancia de su participación. Luego el Profesor Principal de Año debe actualizar en una hoja de cálculo, la bonificación obtenida por el estudiante, en uno de los tres primeros lugares, para así actualizar la evidencia de su resultado en el evento.

Este proceso se repite para cada evento, excepto para el Concurso de certificación de roles en el desarrollo de software. En este se acumulan puntos de acuerdo al nivel certificado, por ejemplo, los niveles pueden ser: **Avanzado, Intermedio o Básico**. El Profesor Principal de Año, debería obtener esta información mediante el tutor del estudiante en su asignatura de Producción Investigación y Desarrollo (PID). Una vez obtenida todas las bonificaciones de los estudiantes durante un semestre, se archiva esta información en cada uno de los expedientes de los estudiantes hasta que se gradúen al finalizar sus estudios, para poder determinar quiénes se merecen el Mérito Científico.

Este método de registrar los resultados obtenidos por los estudiantes, estaba ocasionando varios problemas, y en ocasiones distorsiones en la información. No solo se bonificaba al autor principal del trabajo, como acordaba la resolución, sino que a veces estudiantes que no tenían una

intervención directa en los eventos obtenían este galardón sin habérselo ganado. No se tenía en cuenta las evidencias del trabajo científico investigativo del estudiante por lo que quedaban fuera del proceso Mérito Científico. Muchas veces el estudiante certificaba algún rol y no tenía conocimientos previos de la bonificación que este le proporcionaba para la distinción al Mérito Científico.

1.3. Metodología, herramientas y tecnologías de desarrollo

1.3.1. Metodología de Desarrollo de Software

La selección de la metodología de desarrollo apropiada es esencial para contribuir al éxito del desarrollo de la solución informática. "La metodología de desarrollo de software se encarga de elaborar estrategias; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega. Su principal objetivo es elevar la calidad del software a través de un mayor control sobre el proceso" (Sommerville, 2005).

Las metodologías de desarrollo se pueden dividir en dos grupos, de acuerdo a sus características y los objetivos que persiguen, ágiles y tradicionales (Velthunis, 2010).

Metodologías tradicionales o pesadas: Hacen mayor énfasis en la planificación y control del proyecto, en la especificación precisa de requisitos y modelado, estableciendo estrictamente las actividades involucradas, los roles definidos, los artefactos que se deben producir, las herramientas y la documentación usada (AUP (Proceso Unificado Ágil) RUP (Rational Unified Procces), MSF (Microsoft Solution Framework), Win-Win Spiral Model, Iconix). Estas metodologías han demostrado ser efectivas y necesarias en proyectos de larga duración, pero han presentado problemas en proyectos donde el entorno es muy cambiante y en el cual se exige reducir el tiempo de desarrollo, pero sin que afecte la calidad del producto (Santos, (2009-2010)).

Metodologías ágiles o ligeras: Están orientadas a la generación de código con ciclos muy cortos de desarrollo manteniendo un proceso incremental. Son capaces de permitir cambios en los requisitos de último momento, además el equipo de desarrollo mantiene una comunicación constante con el cliente.

Extreme Programming (XP), SCRUM, Crystal Clear, Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD)), se caracterizan por ser flexibles al cambio y por poder reducir el tiempo de desarrollo del software.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO “MÉRITO CIENTÍFICO”

Además, demuestran que es más importante contar con un software funcional que una documentación excesiva. Estas metodologías han demostrado tener éxito en proyectos pequeños (Roberth, y otros, 2007).

Luego de analizar los dos grupos de metodologías (**ver Tabla 1**) se optó por el uso de un enfoque de desarrollo ágil, escogiéndose la programación extrema (XP) para el desarrollo del módulo propuesto. Esta metodología, se centra en las necesidades del cliente, en la comunicación y retroalimentación del equipo de desarrollo para lograr un producto de buena calidad. La misma admite en la solución cambios de última hora, con el objetivo de favorecer las entregas rápidas del software funcional en un período que no exceda los seis meses. Ofrece la posibilidad de cambiar los requisitos en cualquier momento de la vida de un proyecto. El cliente se convierte en miembro del equipo y decide qué se implementa, además de saber el estado real y el progreso del proyecto. Puede añadir, cambiar o quitar requerimientos en cualquier momento como clave para el éxito en el desarrollo de software.

Tabla 1: Diferencias entre Metodologías ágiles y no ágiles Fuente: (Schenone, 2004)

Metodologías ágiles	Metodologías tradicionales
Se basan en heurísticas provenientes de prácticas de producción de código	Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo
Preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente por el equipo	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso muy controlado, numerosas normas
Contrato flexible e incluso inexistente	Contrato prefijado
El cliente es parte del desarrollo	Cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10)	Grupos grandes

Pocos artefactos	Más artefactos
Menor énfasis en la arquitectura del software	La arquitectura del software es esencial

1.3.2. Marco de trabajo

Uno de los pasos más importantes en el desarrollo de un software es la selección del marco de trabajo. Un marco de trabajo (framework en inglés) simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona una estructura al código fuente, forzando al desarrollador a crear código más legible. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (Potencier, 2010).

Symfony 2.1: Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Potencier, 2010).

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. Symfony es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows. A continuación, se muestran algunas de sus características:

- ✓ Fácil de instalar y configurar en la mayoría de las plataformas
- ✓ Independiente de un sistema gestor de bases datos
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- ✓ Basado en la premisa de convenir en vez de configurar, en la que el desarrollador solo debe configurar aquello que no es convencional

- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la Web
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas de arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- ✓ Fácil de extender, lo que permite su integración con bibliotecas desarrolladas por terceros

Esta herramienta será de gran utilidad ya que el módulo a desarrollar tiene como base el Sistema de Certificación de Roles 2.0. Utilizando Symfony en la versión 2.1.7 permitiendo que se pueda exportar el proyecto sin que el servicio sufra cambios en tiempo de ejecución, solo se necesitan los archivos bundles del negocio creado para actualizar el proyecto. Siendo los bundles la parte más importante de Symfony, permiten utilizar funcionalidades construidas por terceros o empaquetar tus propias funcionalidades para distribuirlas y reutilizarlas en otros proyectos. Además, facilitan mucho la activación o desactivación de determinadas características dentro de una aplicación (Potencier, 2010).

1.3.3. Lenguajes de programación

Los lenguajes de programación permiten crear programas mediante un conjunto de instrucciones, reglas de sintaxis y operadores que un equipo debe ejecutar (Bonanata, 2013). Para darle solución al módulo propuesto se propone la combinación de diferentes lenguajes de programación web, para permitir así el funcionamiento del sistema deseado. A continuación, se enuncian los lenguajes de programación que, de acuerdo con sus características, se emplearán para el desarrollo de la solución.

Del lado del Servidor

PHP 5.3. Es un lenguaje de programación multiplataforma orientado a objetos y con una gran cantidad de bibliotecas de funciones. Permite la conexión con la mayoría de los gestores de base de datos, y posee capacidad para expandir su potencial utilizando módulos. No requiere definición de tipos de variables, ni manejo detallado de bajo nivel (Hernández, 2002).

Este lenguaje se emplea en la construcción del módulo Mérito Científico de contenido dinámico, permitiendo el manejo de excepciones y el empleo de técnicas de programación orientada a objetos. Proporciona estabilidad a la aplicación ya que utiliza su propio sistema de administración de recursos y dispone de un método de manejo de variables, conformando así un sistema robusto. Posibilita configurar diferentes niveles de seguridad para evitar ataques al código. En la UCI es una tendencia desarrollar aplicaciones web empleando este lenguaje, producto a las facilidades que brinda. Existe una comunidad considerable de desarrolladores formalizada, con basta documentación sobre la tecnología.

Twig 2.1. Es un motor de plantillas, que posee un ambiente amigable para el diseñador y el desarrollador, permitiendo añadir funcionalidades a los entornos de plantillas. Posee una sintaxis corta y concisa, similar a la de otros lenguajes de programación. Además, implementa un mecanismo de herencia de plantillas, para acelerar el rendimiento del sistema que se desarrolla (Pacheco, 2013).

Este lenguaje permite compilar las plantillas hasta código PHP regular optimizado, lo que proporciona rapidez durante la implementación. Garantiza la seguridad del código de las plantillas ya que posee un modo de recinto de seguridad para evaluar el código de plantilla que no es confiable. Permite al desarrollador definir sus propias etiquetas y filtros personalizados, garantizando flexibilidad a la aplicación. Posibilita la depuración de las plantillas creadas, mediante mensajes de error con el nombre del archivo y el número de línea donde se produjo el problema (Pacheco, 2013).

Del lado del Cliente

HTML 5.0. Es un lenguaje para escritura de hipertexto, en otras palabras, documentos de texto estructurados, incluye enlaces (links, según su denominación en inglés) que conducen a otros documentos o a otras fuentes de información y permite la inclusión de información por formularios, entre otros. HTML5 se caracteriza por ofrecer una mejor estructura eliminando el uso excesivo de las etiquetas <div>, esta se emplea para definir un bloque de contenido o sección de la página; con el objetivo de que la web sea más coherente y comprensible. (Moral, 2004).

CSS 3.0. Es un lenguaje creado para controlar el aspecto o la presentación de los documentos electrónicos definidos con HTML4, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir. Esta forma de descripción de estilos ofrece

a los desarrolladores el control total sobre estilo y formato de sus documentos. Es empleado para separar los contenidos y su presentación y es imprescindible para crear páginas web complejas ya que separa la definición de los contenidos y la descripción de su aspecto, presentando numerosas ventajas. Mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Jeremy, 2014).

Bootstrap 3.0.1. Es una de las bibliotecas más utilizadas y conocidas en la actualidad, desarrollada por Twitter, pensado y diseñado para crear interfaces web. Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto ofrece agilidad a la hora de cargar y al adaptarse a otros dispositivos. Una de las principales características que posee, es que permite crear interfaces web con CSS5 y JavaScript. Adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa. En otras palabras, automáticamente se adapta al tamaño de un ordenador o de una tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Designy. Es una de las corrientes que marcan esta nueva era digital (Otto, 2014) (Niska, 2014).

JavaScript 1.6. Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (Polo, 2008).

1.3.4. Herramientas de base de datos

Sistema Gestor de Bases de Datos: Los Sistemas de Gestión de Base de Datos (SGBD) se usan como interfaz entre las aplicaciones, los usuarios y las bases de datos. Tienen como propósito principal gestionar convenientemente la información a almacenar o recuperar, de manera sencilla y fiable para el usuario (Mora, 2014).

PostgreSQL 9.4. PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es considerado como un SGBD de código abierto robusto, muy conocido y usado en entornos de software libre, brinda un control de concurrencia multiversión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta

gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación (Postgresql, 1996-2013).

Se selecciona el gestor de base de datos PostgreSQL porque presenta gran capacidad de almacenamiento y buena escalabilidad. Se ajusta al número de CPU (Unidad Central de Procesamiento) mediante la fracción de tiempo en que los recursos de CPU están ocupados en el sistema principal. La cantidad de memoria disponible de forma óptima, se debe a la capacidad de la unidad de disco que se posea, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta.

1.3.5. Servidor web

Un servidor web o servidor HTTP¹, es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Para la transmisión de todos estos datos generalmente se utiliza el protocolo HTTP o el protocolo HTTPS² (la versión cifrada y autenticada), para estas comunicaciones pertenecientes a la capa de aplicación del modelo OSI³.

Apache 2.4 Está diseñado para ser un servidor web potente y flexible, continuamente actualizado y adaptado a los nuevos protocolos (HTTP) y puede funcionar en la más amplia variedad de plataformas y entornos. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular, el cual permite elegir las características del servidor seleccionando qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Proporciona el código fuente completo y viene con una licencia sin restricciones. Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información (Foundation, 2011).

Es un software de libre distribución, que publica su código fuente, lo que permite que pueda ser modificado para colaborar en su desarrollo. Tiene interfaz con todos los sistemas de autenticación. Facilita la integración de complementos de los lenguajes de programación de

¹ HTTP: Protocolo de Transferencia de Hipertexto.

² HTTPS: Protocolo seguro de Transferencia de Hipertexto

³ OSI: Modelo de Interconexión de Sistemas Abiertos

páginas web dinámicas más comunes. Tiene integración estándar del protocolo de seguridad SSL y provee interfaz a todas las bases de datos (Kabir, 2003).

Apache presenta muchas otras características, entre ellas un elaborado índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online (Kabir, 2003).

Apache es seleccionado por la necesidad de un servidor web compatible con la plataforma de desarrollo, que permita rapidez en la navegación y la seguridad e integridad de los datos con los que se trabajen. Además, Apache permite configurar los módulos a utilizar en el servidor, logrando desechar elementos innecesarios para lograr mayor rapidez en la renderización de las páginas.

1.4. Patrones

Un patrón de arquitectura de software es un esquema genérico, probado para solucionar un problema particular recurrente que surge en un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades, relaciones y las formas en que colaboran (González, 2003).

1.4.1. Patrón arquitectónico

Modelo-Vista-Controlador (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, donde la vista es la página HTML, el modelo es la capa de abstracción que se sitúa sobre el Sistema de Gestión de Base de Datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. La vista representa el modelo en un formato adecuado para interactuar con el usuario. Transforma el modelo en una página web, que permite al usuario interactuar con ella. Se encarga de la presentación visual de los datos captados por el modelo, para después mostrarlos al usuario. Los controladores reciben la entrada, o sea los eventos producidos por el usuario mediante el uso del teclado o el mouse a través de las vistas. Los eventos son traducidos para servir las demandas del modelo o las vistas (García, 2013).

Se seleccionó el patrón MVC, ya que permite el desarrollo de aplicaciones web robustas. Con este patrón de diseño reducimos y eliminamos el uso de presentación en un mismo lugar. Si hay alguien encargado de maquetar la aplicación, o alguien más se encarga de crear las reglas del negocio y demás actividades, cada uno puede trabajar independientemente del otro sin sufrir afectaciones. Además de poder aplicar distintos diseños de presentación sin que se altere la lógica del negocio. Proporciona reutilización de los componentes, facilidad para la realización de pruebas unitarias y simplicidad en el mantenimiento del sistema.

1.4.2. Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. La utilización de patrones de diseño, permite ahorrar grandes cantidades de tiempo en la construcción de software. El producto obtenido es más fácil de comprender, mantener y extender. Existen versiones ya implementadas de estas funcionalidades comunes (marcos de trabajo) que permiten centrarse en desarrollar sólo la funcionalidad específica requerida por cada aplicación y, además dan mejor imagen de profesionalidad y calidad.

Los patrones de diseño se agrupan en dos grandes categorías: GRASP y GoF. Los primeros describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de este grupo de patrones se encuentran los siguientes: Experto, Creador, Bajo Acoplamiento, Controlador y Alta Cohesión. Los patrones de diseño GoF son 23 y se clasifican según su propósito en Creacionales, Estructurales, Comportamiento y según su ámbito en Objeto y Clase.

1.5. Métricas de validación

Las métricas constituyen un método de evaluación de los productos y procesos de software, las cuales suelen ser aplicadas a muchas organizaciones, procesos y productos. Ayudan en la planificación, seguimiento y control de un proyecto de software y evalúan la calidad del producto (IEEE, 2015). Las Métricas de validación permiten tener una visión de la eficacia del software. Por ello se estudiaron varios métodos de validación de manera que fuesen validados todos los artefactos generados durante esta investigación.

1.5.1. Métricas para el diseño

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO “MÉRITO CIENTÍFICO”

Se estudiaron las métricas orientadas a clases dada la importancia que tiene para el equipo de desarrollo conocer la complejidad del proceso de implementación y las características del diseño.

Dentro de las métricas orientadas a clases se encuentra la métrica Tamaño de clase (TC) en la que se tienen en cuenta los siguientes aspectos:

- El número total de operaciones (de instancia heredada y privada) que están encapsuladas dentro de la clase.
- El número de atributos (de instancia heredada y privada) que están encapsulados por la clase.
- Promedio general de los dos anteriores para el sistema completo.

Estos dos primeros valores se suman de acuerdo a la clase que se analiza y los resultados son tomados como Cantidad de Procedimientos (CP). Cuanto menos sean los valores para el TC, más probable será que las clases dentro del sistema puedan reutilizarse ampliamente (**ver Tabla 2**), de lo contrario significa que la clase posee un alto grado de responsabilidad y se reduce su reutilización. Luego la CP son comparados para determinar el Tamaño Operacional de Clases (TOC) de cada clase (Fornaris, 2009).

Tabla 2: Clasificación de valores Fuente: (Fornaris, 2009)

Clasificación	Valores
Pequeño	$CP \leq 20$
Medio	$CP > 20 \wedge \leq 30$
Grande	$CP > 30$

Tamaño Operacional de Clase (TOC) (ver Tabla 3)

Consiste en medir el tamaño general de una clase tomando el valor de la cantidad de operaciones que están encapsuladas dentro de dicha clase (Lincke, 2009).

Tabla 3: Rango de valores para la evaluación de los atributos de calidad con la métrica TOC Fuente: (Fornaris, 2009)

Atributo	Categoría	Criterio
Responsabilidad	Baja	$CP \leq \text{Promedio}$
	Media	$\text{Promedio} \leq CP \leq 2 * \text{Promedio}$
	Alta	$CP > 2 * \text{Promedio}$
Complejidad de Implementación	Baja	$CP \leq \text{Promedio}$
	Media	$\text{Promedio} \leq CP \leq 2 * \text{Promedio}$
	Alta	$CP > 2 * \text{Promedio}$
Reutilización	Baja	$CP > 2 * \text{Promedio}$
	Media	$\text{Promedio} \leq CP \leq 2 * \text{Promedio}$
	Alta	$CP \leq \text{Promedio}$

Relaciones entre Clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad (Desarrollo web, 2013) (ver **Tabla 4**).

Acoplamiento: Un aumento del RC implica un aumento del acoplamiento de la clase.

Complejidad de mantenimiento: Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.

Cantidad de pruebas: un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 4: Rango de valores para los atributos de calidad relacionados con la métrica RC. Fuente: (Fornaris, 2009)

Atributo	Categoría	Criterio
----------	-----------	----------

Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	Cantidad de relaciones de uso > 2
Complejidad del Mantenimiento	Baja	Cantidad de relaciones de uso <= Promedio
	Media	Promedio <= Cantidad de relaciones de uso <= 2* Promedio
	Alta	Cantidad de relaciones de uso > 2* Promedio
Cantidad de Pruebas	Baja	Cantidad de relaciones de uso <= Promedio
	Media	Promedio <= Cantidad de relaciones de uso <= 2* Promedio
	Alta	Cantidad de relaciones de uso > 2* Promedio

1.6. Pruebas

Anteriormente se intentaba construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. A continuación, se muestran los tipos de pruebas, donde se crean una serie de casos de prueba que intentan demoler el software construido además demuestran hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se llevan a cabo las pruebas proporcionan una buena indicación de la fiabilidad del software y, de alguna manera, indican la calidad del software como un todo (Pressman, 2005).

1.6.1. Pruebas de Caja Blanca

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Aunque las pruebas de caja blanca son aplicables a varios niveles unidad, integración y sistema, habitualmente se aplican a las unidades de software. Su objetivo es comprobar los flujos de ejecución dentro de cada unidad (función, clase, módulo, etc.) pero también pueden probar los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas de sistema. Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración). Mediante los métodos de prueba de caja blanca, se pueden obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los ciclos en sus límites y con sus límites operacionales
- Ejerciten las estructuras internas de datos para asegurar su validez.

Según (Pressman, 2005), las principales técnicas de diseño de pruebas de caja blanca son:

- Pruebas de la estructura de control
- Pruebas de caminos básicos

Se pretende utilizar la prueba de caminos básicos ya que ésta permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

1.6.2. Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Las pruebas de caja negra no son una alternativa a las técnicas de pruebas de caja blanca, sino que se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los

métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- ❖ **Técnica de la Partición de Equivalencia:** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ❖ **Técnica del Análisis de Valores Límites:** esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ❖ **Técnica de Grafos de Causa-Efecto:** Según (Pressman, 2005) es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones

En este caso se escoge dentro del método de caja negra la técnica de la Partición de Equivalencia que es una de las más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así el número de casos de prueba que hay que desarrollar.

1.6.3. Pruebas de aceptación

La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar cómo el sistema satisface sus criterios de aceptación, validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y procesos de negocio. Está considerada como la fase final del proceso, para crear un producto confiable y apropiado para su uso (Pressman, 2005).

1.7. Conclusiones parciales

Después de una búsqueda de información referente a las necesidades del cliente, para llegar a la obtención de la propuesta de solución. Se estudiaron y analizaron los elementos que sustentan la solución del problema, llegando a las siguientes conclusiones:

- El Sistema para la Certificación de Roles 2.0 almacena tres de los cuatro indicadores que mide la resolución para el Mérito Científico, ellos son: la certificación de roles, la participación y/o premios en eventos y las publicaciones. Se hace necesario desarrollar un módulo que gestione el Mérito Científico e incorpore la obtención de resultados en exámenes de premio.
- Se opta por un enfoque de desarrollo ágil, seleccionando XP como metodología de desarrollo de software para el desarrollo del módulo Mérito Científico para el proceso de certificación de roles.
- Los lenguajes y herramientas de desarrollo seleccionados contribuyen a la implementación del módulo Mérito Científico para el proceso de certificación de roles.
- Los tipos de pruebas seleccionadas permiten identificar deficiencias en el diseño y en la implementación de las funcionalidades del sistema, con el objetivo de solventarlas y obtener un sistema que contribuya a solucionar el problema planteado.

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se exponen los artefactos generados propios de la metodología para el desarrollo del sistema tales como: historias de usuario, actores del sistema, tarjetas CRC, el listado de reserva del producto y se describen en qué plazo van a realizarse las iteraciones. Cada iteración depende de los requisitos definidos por el usuario y la prioridad que tengan para el proyecto. Además, se describe la arquitectura del sistema, los patrones de diseño que fueron utilizados incluyendo la descripción de los requisitos funcionales y no funcionales.

2.1. Descripción de los procesos del negocio

La Dirección de Investigaciones, entre otras funciones, tiene la misión de llevar un control de la participación de sus trabajadores y estudiantes en diferentes eventos científicos, tanto nacionales como internacionales, así como la organización de los diferentes eventos científicos organizados por la propia Universidad como: Jornada Científico Estudiantil (JCE), Fórum de Ciencia y Técnica (FCT) y Conferencia Científica Uciencia, por citar algunos. Dicho control también es de interés de las facultades. La organización de los eventos científicos desarrollados por la Universidad, queda a cargo de los asesores y/o vicedecanos de Producción-Investigación de las diferentes facultades o por la propia Dirección de Investigaciones (DI), en dependencia del nivel en el que se desarrolle. Durante este proceso se envían los trabajos que son arbitrados posteriormente por una Comisión Científica, quienes valoran la aceptación o no para ser presentados en el evento. Todo evento que se desarrolla cuenta con un Comité Organizador, que como su nombre lo indica, es el grupo encargado de la organización de los mismos. Lleva a cabo la acreditación y con ello un control de todos los ponentes, conferencistas y otros invitados. Una vez que se desarrolla el evento, se dan las premiaciones en caso que el evento lo conciba y se hacen las conclusiones del mismo.

Para la participación en eventos científicos internacionales, la propuesta la debe realizar el Profesor Principal de Año al departamento de Investigación y Postgrado y este ser autorizado por el Decano(a), para así cada Profesor Principal de Año de dicha facultad pueda hacerle llegar mediante una notificación a cada estudiante, tras haber sido aceptado el trabajo por el Comité Organizador del evento y después de haber sido aceptada la participación del estudiante, por el responsable para ello (Decano, Líder de Proyecto Productivo o Proyecto Investigación y Desarrollo o por el tutor de PID).

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Esta propuesta debe ser valorada por la Dirección de Investigaciones tomando como fuente de información la Dirección De Cooperación Internacional, la Dirección de Economía y Finanzas y el Consejo de Dirección UCI. Una vez que se materializa la participación en el evento, la Dirección de Investigaciones debe dejar constancia de dicha participación. Luego mediante la vía correo o de alguna manera se debe retroalimentar al área a la cual pertenece el participante.

Actualmente todo este proceso de ponderación del Mérito Científico, tanto para eventos internos como nacionales e internacionales, se realiza de forma manual, archivándose toda la información en papel y en el mejor de los casos en documentos Word u hojas de cálculo. Las informaciones son enviadas hacia los niveles superiores a través de correo electrónico. Todo esto trae como consecuencia un difícil y engorroso procesamiento de los datos, que se cometan errores humanos con mayor frecuencia, retrasos en la información, que exista inconsistencia de la misma en las diferentes estructuras de la Universidad y que no se cuente en todo momento con la información más actualizada.

El premio al Mérito Científico se otorga al culminar los estudios. En él se involucran los estudiantes desde la entrada en su primer año hasta el cierre de su carrera. La obtención del Mérito Científico es opcional, depende de la voluntad del estudiante. Para cada expediente de los estudiantes que opten por el premio, se reflejan un conjunto de actividades llamadas “**Eventos Científicos**”, que el estudiante debe realizar a lo largo de cada semestre. Para ello cada facultad cuenta con una hoja de cálculo de apoyo a la identificación de los estudiantes potenciales a obtener esta distinción. Estas hojas de cálculo traen malas consecuencias cuando no se les da un uso correcto. La persona encargada de actualizar la bonificación obtenida por cada estudiante, no tiene como gestionar la seguridad del mismo, y cualquiera que tenga mínimos conocimientos de hojas de cálculo puede cambiar las fórmulas, e incluso eliminarlas.

Según lo planteado en la resolución 79/2015 por el Ministerio de Educación Superior, para el otorgamiento del diploma al Mérito Científico se tendrá en cuenta el acumulado de puntos recogido en el expediente del estudiante y para ello se tiene:

Resultados de eventos por Niveles

- **Internacional**
- **Nacional**
- **Provincial**
- **CES:** Centros de Educación Superior

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

En los eventos científicos se podrán presentar un máximo de tres trabajos por cada autor y se aceptarán en el caso de los trabajos hechos por un colectivo de estudiantes los que fueron realizados hasta por tres autores. Cuando se trate de más de un autor se otorgará la puntuación correspondiente a cada uno de ellos por igual.

Tabla 5: Resultados de bonificación por Niveles. Fuente: (Gaceta, 2014)

Resultados por niveles	Premio Relevante	Premio Destacado	Mención	Participación
Internacional	---	---	---	3
Nacional	5	4	3	---
Provincial	4	3	2	---
CES	3	2	1	---

Tabla 6: Resultados de bonificación por Publicaciones Científicas. Fuente: (Gaceta, 2014)

Resultados por niveles	Primer autor	Segundo Autor	3er Autor
Primer Nivel	6	5	4
Segundo Nivel	5	4	3
Tercer Nivel	4	3	2
Cuarto Nivel	3	2	1

Para el **Concurso de certificación de roles en el desarrollo de software** se acumulan puntos de acuerdo a la siguiente escala:

1. **Avanzado:** 5 puntos
2. **Intermedio:** 4 puntos
3. **Básico:** 3 puntos

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Para los exámenes de premio: A los estudiantes que obtengan los tres primeros lugares se le otorgarán los puntos de acuerdo a la escala siguiente:

1. **Primer premio....** 5 puntos.
2. **Segundo premio...** 4 puntos.
3. **Tercer premio....** 3 puntos.

Los **exámenes de premio** se rigen por lo establecido en la Resolución Ministerial No. 210 de fecha 31 de julio del 2007, “Reglamento para el Trabajo Docente y Metodológico” del Ministerio de Educación Superior.

2.2. Propuesta de solución

A raíz de la falta de sistemas que gestionen el Mérito Científico y faciliten la ponderación del mismo, se hace imperiosa la necesidad de implementar un módulo que sea capaz de informatizar este trabajo en las oficinas del Vicedecanato de Formación (VDF) de la Facultad 3 de la Universidad de las Ciencias Informáticas. Algunos de los beneficios de esta investigación una vez que sea implantada, evaluada y aprobada son:

- El módulo mostrará al usuario aquellas opciones a las que está autorizado a acceder y realizar cambios en las funcionalidades permitidas, teniendo en cuenta el rol que desempeñe en el sistema.
- El módulo debe permitir la utilización de tres de los cuatro indicadores que posee el Sistema de Certificación de Roles 2.0, además de incorporar la bonificación por resultados en examen de premio, para poder gestionar los estudiantes candidatos al premio Mérito Científico.
- Permitirá la gestión de evidencias para poder validar los resultados obtenidos.
- Brindará la posibilidad de utilizar las tareas que tributan al Mérito Científico, asignadas por el Sistema de Certificación de Roles 2.0, para agregarlas al expediente del estudiante.
- El módulo brindará la posibilidad de conocer el estado en que se encuentran cada bonificación de forma personal y estando autenticado.

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

- Permitirá archivar toda la información que el Profesor Principal de Año considere necesaria, posibilitando que pueda ser reutilizada en otro momento como la conformación del expediente para la integralidad del estudiante.
- El módulo brindará la posibilidad de llevar un control de la información del estudiante.

2.2.1. Actores del sistema

Un actor es una entidad externa al sistema que se modela y que puede interactuar con él. Puede ser una persona o un grupo de personas homogéneas, otro sistema, o una máquina. Los actores son externos al sistema que vamos a desarrollar. Por lo tanto, al identificarlos, estamos comenzando a delimitar el sistema y a definir su alcance (Sommerville, 2005). A continuación, se muestran los principales actores del módulo Mérito Científico

Tabla 7: Actores del sistema y su descripción. Fuente: (Elaboración propia)

Actores relacionados con el sistema	Descripción
Profesor Principal de Año	Se encarga de notificar al estudiante de los eventos próximos, gestionar los estudiantes que participan, así como la bonificación obtenida, y además es el encargado(a) de gestionar el expediente de cada estudiante, al terminar cada semestre.
Estudiante	Es el encargado de incorporar evidencias a cada evento realizado que se asigne en el sistema por el Profesor Principal de Año, además se encarga de solicitar alguna participación en algún evento externo, para su posterior aprobación, y a partir de 3er año continúa con su formación en PID
Subdirector de Formación	Se encarga de velar por el correcto funcionamiento del proceso a través de la aplicación, por lo que tendrá una credencial que le permitirá visualizar todos los niveles del

	<p>sistema, sin que esto implique hacer modificaciones en el mismo.</p> <p>Define el comienzo y el fin de un semestre en curso, es el encargado de gestionar por cada Profesor Principal de Año los grupos asociados al mismo antes de empezar el semestre.</p>
Administrador del sistema	Se encarga de administrar el sistema, con privilegios y credenciales para realizar cualquier cambio en la aplicación

2.3. Exploración

En la fase de exploración se define el alcance general del proyecto, el cliente describe lo que necesita mediante la redacción de sencillas “historias de usuarios” y plantean a grandes rasgos su interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (Roberth, y otros, 2007).

2.3.1. Historias de usuario

Entre los artefactos que define la metodología seleccionada se encuentran las historias de usuario que son utilizadas para especificar las funcionalidades que brindará el sistema. Cada historia de usuario es una representación de un requerimiento de software escrito en una o dos frases utilizando el lenguaje común del usuario (**ver Tabla 8**). Representan una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Además, son la base para realizar las pruebas de aceptación, así como la estimación y planificación del proyecto (Joskowicz, 2008). Como resultado del trabajo realizado durante la fase de exploración se identificaron un total de 31 historias de usuario, y a continuación se muestra una de ella:

Tabla 8: Historia de usuario 15. Fuente (Elaboración propia)

Historia de Usuario	
Número: 15	Nombre: Gestionar evidencias por cada evento
Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad del negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas /etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 Semana
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, adjuntar evidencias a cada evento en que participaron y se les haya asignado por el Profesor Principal de Año. Esta acción depende del evento en cuestión.	
Observaciones: El Profesor Principal de Año debe de agregar el evento en el que participó dicho estudiante, para que cuando se suba la evidencia, se emita la bonificación obtenida.	

2.4. Planificación

Es aquí donde el cliente plantea a grandes rasgos la prioridad de las historias de usuario en correspondencia con las necesidades más inmediatas para luego asignarlas, por el orden de relevancia a las iteraciones planificadas. Además, las estimaciones de esfuerzo asociada a la implementación de las historias de usuario, la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. El tiempo de implementación de una historia de usuario generalmente es de uno a tres puntos.

2.4.1. Estimación de esfuerzo por historia de usuario

Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación de esfuerzo (ver **Tabla 9**) para cada una de las historias de usuario, arrojando los siguientes resultados:

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Tabla 9: Prioridad para el cliente por historia de usuario. Fuente (Elaboración propia)

Número	Historia de usuario	Prioridad para el cliente
1	Insertar eventos externos.	Alta
2	Modificar eventos externos.	Alta
3	Salvar datos modificados del evento	Alta
4	Mostrar eventos externos	Alta
5	Eliminar eventos externos	Alta
6	Cerrar evento	Alta
7	Realizar búsqueda de eventos externos	Media
8	Mostrar listado de los eventos externos realizados	Media
9	Crear solicitud para participación en evento	Media
10	Eliminar solicitud de participar en evento	Media
11	Listar solicitudes realizadas	Media
12	Crear evento UCI	Alta
13	Modificar evento UCI	Media
14	Agregar solicitud de publicación científica	Media
15	Gestionar evidencias por cada evento	Media
16	Aprobar solicitud	Media
17	Buscar resultados de eventos UCI	Media
18	Asignar eventos a estudiante	Alta
19	Eliminar eventos a estudiante	Media
20	Enviar alerta con la convocatoria	Baja
21	Mostrar desempeño por cada estudiante	Media
22	Adicionar punto de bonificación por evento	Baja
23	Eliminar evidencia	Media
24	Gestionar examen de premio	Alta
25	Mostrar estudiantes candidatos al mérito	Baja
26	Actualizar datos de bonificación	Baja
27	Mostrar reporte del sistema	Baja
28	Adicionar evidencias al expediente	Media
29	Exportar expediente	Baja
30	Mostrar bonificación alcanzada por el estudiante	Media

31	Asignar estudiante a Profesor Principal de Año.	Alta
-----------	---	------

2.4.2. Lista de reserva del producto

Las listas de reserva del producto en una aplicación son de gran importancia ya que son las cualidades que todo sistema debe poseer para un correcto funcionamiento. A continuación, se definen los requisitos no funcionales mediante la lista de reserva del producto (**ver Tabla 10**), elaboradas en la fase de planificación:

Tabla 10: Lista de reserva del producto (Requisitos no funcionales).

RNF1	<p>Usabilidad</p> <p>Facilidad de uso por parte de los usuarios:</p> <p>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.</p>
RNF2	<p>Confiabilidad</p> <p>El sistema debe ser preciso con la información que maneja y le proporciona al usuario. Haciendo énfasis en los resultados de los análisis que ejecutará, para evitar errores que puedan incidir negativamente en la toma de decisiones.</p>
RNF3	<p>Especificación de la terminología utilizada</p> <p>El sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mejor comprensión de la herramienta de trabajo</p>
RNF4	<p>Emplear perfiles de usuario</p> <p>Diferenciar las interfaces y opciones para los usuarios que accedan al sistema con diferentes roles (profesores o estudiantes).</p>
RNF5	<p>Seguridad de la información</p>

	El sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
RNF6	Software(Cliente)
	Las PC clientes deben tener instalado un Navegador Web Mozilla Firefox o Google Chrome.
RNF7	Software(Servidor)
	Se debe contar con un Servidor web Apache versión 2.0 o superior.
	El servidor puede poseer un sistema operativo (Devian o cualquier versión Libre) o (Windows 8.1) se recomienda esta versión pero el sistema es multiplataforma
RNF8	Hardware(cliente)
	Las PC clientes deben poseer requisitos mínimos como: Procesador 1ghz, con 512 de RAM. Cada PC cliente debe tener conexión de red.
RNF9	Hardware(servidor)
	Se debe disponer de un servidor con requisitos mínimos como: Procesador AMD A6, con 2GB de RAM o más. Se debe disponer como mínimo de un disco duro de 500 GB o superior y el servidor debe tener conexión de red

2.5. Desarrollo de iteraciones

Una vez definidas las historias de usuarios y estimado el esfuerzo para la realización de cada una de ellas se establece el plan de iteraciones, regulando la cantidad de historias de usuario a implementar dentro del rango establecido por la estimación efectuada. Tomando como referencia los aspectos antes tratados la aplicación se desarrollará en tres iteraciones, las cuales se describen a continuación:

Iteración 1

La iteración tiene como finalidad implementar las historias de usuario que se consideraron más necesarias atendiendo a su relevancia e impacto para el negocio y para el cliente. Se da respuesta a las funcionalidades de asignar un estudiante a un Profesor Principal de Año, de registrar eventos UCI y eventos externos, gestionar exámenes de premio, además de modificar y eliminar los datos insertados.

Iteración 2

En esta iteración se realizan todas las historias de usuarios relacionadas con la gestión del Mérito Científico: como adicionar puntos de bonificación por eventos, mostrar desempeño por cada estudiante, mejor bonificación hasta el momento, entre otras de prioridad media para el cliente.

Iteración 3

En esta iteración se realizan todas las historias de usuario relacionadas con los reportes que se generan en el sistema. Algunas de estas son: crear solicitud de participación en eventos externos, mostrar reporte del sistema, entre otras de prioridad baja para el cliente.

Después de realizada la estimación del esfuerzo y el plan de iteraciones, continuando los pasos que propone la metodología XP. En la tabla 11 se muestran las tres iteraciones analizadas previamente con las historias de usuario que incluyen el plan de duración de las iteraciones. Este tiene como objetivo fundamental mostrar la duración de cada iteración, según la prioridad asignada por el cliente.

Tabla 11: Plan de duración de las iteraciones. Fuente: (Elaboración propia)

Iteración	Historias de usuario/Prioridad	Duración total de las Iteraciones(Semanas)
Iteración I	10 /Alta	6
Iteración II	15 /Media	4
Iteración III	6 /Baja	2
Total	31	12

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

A continuación, se presenta el plan de entregas ideado para la fase de implementación. Atendiendo al mismo se harán entregas del sistema al finalizar cada iteración en la fecha aproximada que se indica en la siguiente tabla:

Tabla 12: Plan de duración de las iteraciones. Fuente (Elaboración propia)

Iteración	Fecha de Entrega
Iteración I	21/03/2016
Iteración II	2/05/2016
Iteración III	16/05/2016

2.6. Diseño de la solución

La metodología XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema robusto y reutilizable tratando de mantener su simplicidad, es decir, crear un diseño evolutivo que se va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente. A la hora de darle cumplimiento a la actividad de diseñar, XP no especifica ninguna técnica de modelado. Pueden utilizarse indistintamente sencillos esquemas en una pizarra, diagramas de clases utilizando UML o tarjetas CRC (Clase, Responsabilidad y Colaboración) siempre que sean útiles, tributen a la comprensión y no requieran mucho tiempo en su creación.

Como base para el desarrollo de la aplicación propuesta se utilizó el marco de trabajo Symfony 2 que implementa, como lo hacen muchos otros, el patrón arquitectónico Modelo Vista Controlador (MVC) que obliga a organizar el código de acuerdo a sus convenciones. Symfony 2 proporciona una estructura en forma de árbol de archivos para organizar de forma lógica todos esos contenidos, además de ser consistente con la arquitectura MVC utilizada y con la agrupación de los directorios de la forma **proyecto / src / bundle**. Cada vez que se crea un nuevo proyecto, aplicación o módulo, se genera de forma automática la parte correspondiente de esa estructura. Además, la estructura se puede personalizar completamente, para reorganizar los archivos y directorios o para cumplir con las exigencias de organización de un cliente.

2.6.1. Arquitectura del sistema

Se definió una arquitectura en tres capas, basada en el patrón arquitectónico MVC. El nivel superior de presentación o vista es la capa que contiene los componentes con los que interactuará el usuario,

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

encapsula las interfaces de usuario representadas por las clases twig, hojas de estilo, funciones JavaScript y formularios necesarios para la interacción con el cliente. En este nivel se responde a un diseño y comportamiento estándar con vista a facilitar el trabajo con ellos y lograr la estandarización de la aplicación.

El nivel de negocio o capa de control contiene las clases controladoras que se encargan de dar respuestas a las peticiones realizadas por el usuario. Es aquí donde se recibe una petición de la capa superior o sea la capa de presentación por medio de una ruta, se gestiona o procesa la misma de ser necesario, solicitándola a la capa de acceso a datos y finalmente enviando la respuesta a quien realizó la petición. En el nivel de acceso a datos o modelo, se encuentra la capa de negocio que tiene como principal función separar la lógica de negocio del resto de la aplicación y obtener una mayor reutilización del código. Esta capa se encarga de recibir las peticiones, procesar la información, hacer pedidos y devolver respuestas a las clases controladoras, las que a su vez envían a las clases enmarcadas en la capa vista.

En aras de esclarecer el acceso entre cada capa de la arquitectura propuesta, se considera necesario exponer que la misma incluye una capa transversal denominada capa de dominio. Esta comunica con las restantes tres capas de forma que cada una, en el momento que necesite, pueda acceder a los datos almacenados en las entidades. En la imagen que se muestra se representa esta capa y la relación antes descrita (*ver Figura 1*).

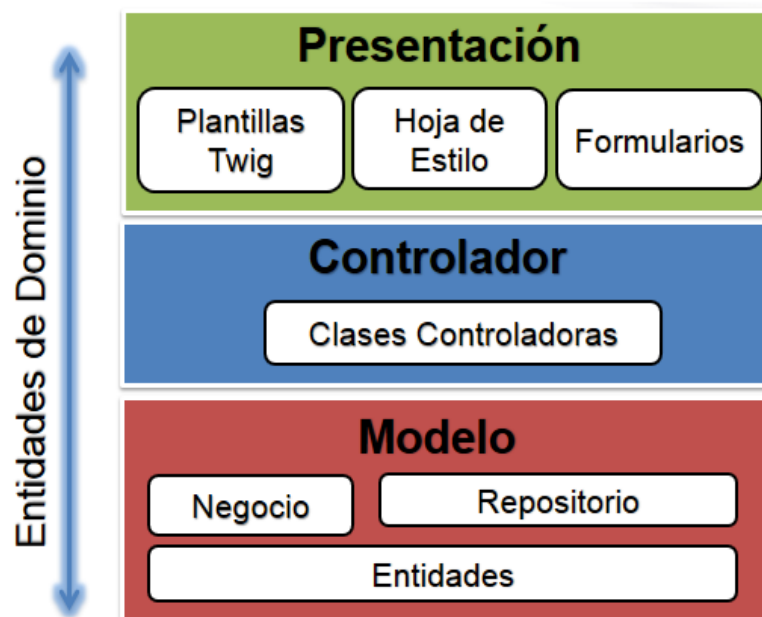


Figura 1 Arquitectura del sistema Fuente: (Velthunis, 2010)

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Inicialmente se arma una (vista) donde el usuario interactúa con la misma. El controlador recibe la petición de la acción solicitada mediante una ruta o dirección y gestiona el evento. El controlador accede al modelo actualizándolo o buscando la información requerida para dar respuesta. El controlador delega a los objetos de la vista, la tarea de desplegar la interfaz de usuario y mostrar los datos del modelo para generar la interfaz apropiada (Potencier, 2010) (*ver Figura 2*).

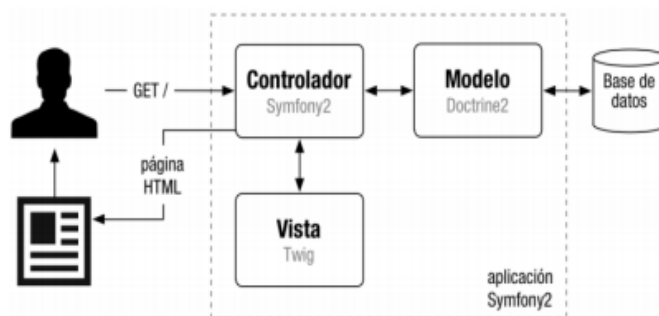


Figura 2 Patrón MVC. Fuente (Potencier, 2010)

2.6.2. Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas) (Casas, 2009). Como resultado del trabajo realizado se muestra un ejemplo de tarjetas CRC confeccionadas durante la fase de diseño.

Tabla 13: Tarjeta CRC número 1. Fuente: (Roberth, y otros, 2007)

PublicacionCientificaController	
Responsabilidad	Colaboradores
1. Adicionar publicación científica.	1. Persona
2. Modificar publicación científica.	2. Usuario
3. Eliminar publicación científica.	3. Publicación científica
4. Aprobar publicación científica.	4. Rol

5. Listado de publicaciones científica por estudiante.	
6. Visualizar publicaciones científicas.	
7. Verificar si existe publicación científicas.	

2.7. Codificación de la solución

En esta fase se genera todo el código fuente necesario para satisfacer las historias de usuario definidas para la solución. Al inicio de cada iteración, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Todas las HU son traducidas en tareas de programación.

2.7.1. Tareas de ingeniería

Para llevar a cabo una correcta implementación de las HU se deben definir por parte del equipo de desarrollo las Tareas de Ingeniería (TI) (**ver Tabla 14**) que se realizan en cada una de las iteraciones. Las TI también conocidas como tareas de implementación permiten a los desarrolladores obtener un nivel de detalle más avanzado que el que propicia las HU. A continuación, se describen una de las tareas de ingeniería pertenecientes a la primera iteración.

Tabla 14: HU Tarea de ingeniería Adicionar bonificación a un evento. Fuente (Elaboración propia)

Tarea de ingeniería	
Número tarea: 9	Historia de usuario: 9, Adicionar bonificación
Nombre de tarea: Implementar RF_Adicionar_bonificacion_evento	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 18-02-2016	Fecha fin: 20-02-2016
Programador responsable: Leonardo Torres Silega.	
Descripción: se debe implementar el requisito funcional Adicionar bonificación por evento, el cual debe permitir al Profesor Principal de Año registrar por cada evento que el estudiante participe, el resultado obtenido.	

Para una correcta comprensión y ejecución de la codificación resulta imprescindible el uso de estándares de codificación.

2.7.2. Estándares de Codificación

Los estándares de codificación se definen por el equipo de desarrollo para lograr estandarización en la programación del software. Estos se basan en la estructura y apariencia física de un programa con el fin de facilitar la lectura, comprensión, mantenimiento del código, reutilización a lo largo del proceso de desarrollo de un software y no en la lógica del programa. La generalización de aspectos tan simples como el trato de las mayúsculas, ayuda a eliminar conflictos de funcionalidades implementadas con nombres iguales, y guían de forma clara el proceso de desarrollo. Los estándares de codificación por lo general son reglas que se siguen para la escritura del código fuente, de tal manera que a otros programadores se les facilite entender el código (Calleja, 2009).

A continuación, se muestran algunas pautas del estándar definido por el equipo de desarrollo, así como ejemplos de su uso.

1. Todas las nomenclaturas a utilizar se definen en idioma español.
2. Los identificadores para las variables y los parámetros se escriben con letras en minúsculas y en caso de ser un nombre compuesto se escriben juntas utilizando la notación Camello, variante (LowerCamelCase).

```
private $id;
```

```
private $fechaFin;
```

3. En caso de que los métodos se nombren con una sola palabra. Esta se escribe en minúsculas, en caso de ser un nombre compuesto. Las palabras que lo conforman se escriben juntas, de la segunda en adelante se escriben con letra inicial mayúscula. Se emplea la notación Camello, variante (LowerCamelCase).

```
public function createStaticAction (Request $request) {
```

4. El bundle del módulo Mérito se escribe con el nombre Merito seguido de la palabra Bundle (MeritoBundle) **(ver Figura 3)**.

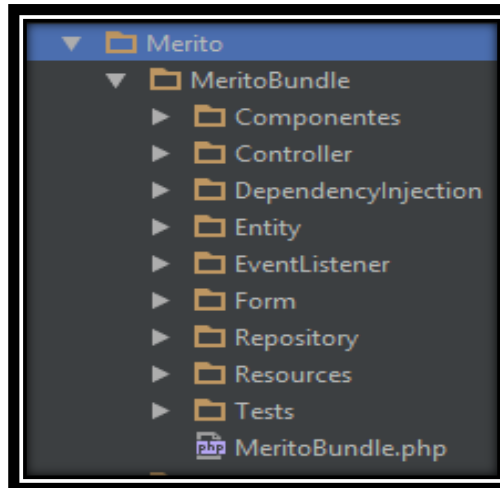


Figura 3: Estructura de carpetas Fuente: (Elaboración propia)

5. Los nombres de las clases se escriben con la primera letra de cada palabra que lo compone en mayúscula, haciendo uso de la notación Camello, con la variante UpperCamelCase.
6. Guía para la declaración de variables
 - ❖ Las variables deben ser declaradas lo más cercano a la línea de su primer uso.
 - ❖ Solamente una declaración de variable por línea
7. Las clases formularios comienzan con el nombre del formulario, seguido de la palabra Type (ConvocatoriaType).

```
class ConvocatoriaType extends AbstractType {
```

8. Se hizo uso de llaves para ganar en comprensión del código

```
public function newAction()  
{  
    $entity = new EventoUci();  
    $form = $this->createForm(new EventoUciType(),  
    $entity);  
  
    return $this->  
>render('MeritoBundle:EventoUci:new.html.twig',  
    array(  
        'entity' => $entity,  
        'form' => $form->createView(),  
    ));  
}
```

Figura 4: Fragmento de código, muestra el uso de llaves Fuente: (Elaboración propia)

2.8. Patrones de diseño utilizados

El desarrollo de un módulo conlleva, en la mayoría de las ocasiones, a darle solución a problemas complejos que ya alguien ha resuelto con anterioridad o de algún sistema que carece de una u otras funcionalidades. Por esta razón uno de los pasos a tener en cuenta cuando se decide desarrollar un módulo para un proyecto de software es identificar qué patrones pueden ser utilizados. Entiéndase por patrón como una solución estándar para un problema común de programación. Los patrones de diseño son herramientas que proveen facilidades para crear un software reutilizable de buena calidad. Cada patrón describe un problema que ocurre repetidamente en nuestro entorno. A continuación, se muestran algunos patrones que utiliza el marco de trabajo Symfony aplicados en el diseño del sistema.

2.8.1. Patrones GRASP

Patrón Controlador: la clase controladora `EventoUciController`, constituye un ejemplo de la aplicación de este patrón. La misma tendrá en cuenta la responsabilidad de manejar los eventos uci en el sistema.

Alta cohesión: es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta. En el caso del sistema propuesto se utiliza el patrón alta cohesión, en la asignación a cada clase de las responsabilidades que le corresponde. Se establecen las condiciones, para que cada una colabore con las demás en la resolución de tareas.

Bajo acoplamiento: las clases que implementan la lógica de negocio y de acceso a datos se encuentran en el modelo. Estas clases no tienen asociaciones con las de la vista, por lo que la dependencia en este caso es baja, demostrándose así el uso de este patrón. Consiste en tener las clases lo menos relacionadas posible, ya que en caso de producirse una modificación en alguna de ellas, se tenga una repercusión mínima en las demás (Viscoti, 2004). Esta característica permitió potenciar la reutilización y disminuyó la dependencia entre las clases.

2.8.2. Patrones GoF

Patrón Decorador: En el sistema este patrón permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidad. Se aplica con la intención de proporcionar una forma flexible de introducir o eliminar funcionalidad de un componente sin modificar su apariencia externa o su función. Por ejemplo, añadir el menú y el pie de página a las plantillas que lo requieran, se trata de decorar las plantillas con elementos adicionales reutilizables.

2.9. Conclusiones parciales

Una vez finalizado el presente capítulo se arriba a las siguientes conclusiones:

- ✓ La obtención de los requisitos funcionales y no funcionales, permitió definir el comportamiento y las restricciones del módulo Mérito Científico para su implementación.
- ✓ La propuesta de arquitectura del sistema se sustenta en un conjunto de componentes reutilizables que tienen como base el patrón arquitectónico MVC, lo que conforma un sistema robusto y flexible a cambios.
- ✓ Además, se describieron los estándares de codificación y patrones de diseño que se utilizan durante la implementación del módulo, los cuales permiten una legibilidad del código fuente y una mayor reutilización por parte del usuario.
- ✓ Las entregas del producto al cliente dividido en tres iteraciones, permitió una mejora constante en cada iteración, teniendo en cuenta las funcionalidades que se le fueron incorporando al sistema, obteniéndose como resultado la solución de la propuesta planteada.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se exponen las pruebas que se le realizaron al sistema en aras de validar la propuesta de solución. Estas fueron realizadas en cada una de las iteraciones, las cuales fueron necesarias para avanzar hacia la próxima iteración ya que para lograr un producto con calidad es necesario un plan de pruebas para así darle seguimiento a los cambios.

3.1. Validación del diseño

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

3.1.1. Relaciones entre clases (RC)

La métrica RC evalúa la cantidad de relaciones de uso que existe entre las distintas clases que forman el diseño propuesto. Los aspectos de calidad que se miden son: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas. Se procede a calcular el promedio de las mismas y teniendo ambos valores, según los criterios expuestos en el Capítulo 1, se determina la incidencia de los atributos de calidad en cada una de las clases.

Tabla 15: Métrica RC para las clases principales del sistema. Fuente (Elaboración propia)

No	Clases	Relaciones de uso
1	BonificacionExtraController	2
2	ConvocatoriaController	0
3	EventoController	2
4	EventoEstudianteController	6
5	EventoExternoController	3
6	EventoUciController	3
7	EvidenciaController	3
8	ExamenPremioController	2
9	IndexController	2
10	ProfesorEstudianteController	3
11	PublicacionCientificaController	1
12	ReportesController	1

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

13	ResultadoController	1
14	SolicitudParticipacionEventoController	1
15	UsuarioController	1
16	BonificacionExtra	3
17	Categoria	1
18	Convocatoria	1
19	Evento	1
20	EventoExterno	7
21	EventoUci	0
22	Evidencia	2
23	ExamenPremio	6
24	ExpedienteIntegralidad	3
25	NomencladorAlcance	3
26	NomencladorPais	1
27	PublicacionCientifica	4
28	Resultado	1
29	SolicitudParticipacionEvento	0
30	LoginListener	4
	Total	68

Para medir el acoplamiento según los resultados de esta métrica, se plantean valores según los criterios expuestos en el Capítulo 1. Para esta métrica se obtuvieron los siguientes resultados.

Tabla 16: Resultado para el Acoplamiento. Fuente (Elaboración propia)

Acoplamiento	Criterio	Cantidad de clases
Ninguno	0	3
Bajo	1	10
Medio	2	5
Alto	> 2	12

Tabla 17: Resultado para la cantidad de prueba y Complejidad de mantenimiento. Fuente (Elaboración propia)

Categoría	Criterio	Cantidad de clases
Baja	\leq Prom.	18
Media	Entre Prom. y $2 \cdot$ Prom.	9

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Alta	> 2*Prom.	3
------	-----------	---

Tabla 18: Resultado para la Reutilización. Fuente (Elaboración propia)

Categoría	Criterio	Cantidad de clases
Baja	>2* Prom.	3
Media	Entre Prom. y 2*Prom.	9
Alta	<= Prom.	18

Al analizar los resultados obtenidos para cada uno de los atributos de calidad para esta métrica, se evidencia que el 40 % de las clases poseen un alto acoplamiento, un 17 % media y el 33 % bajo. Por otra parte el 60 % de las clases presentan una complejidad de mantenimiento baja, el 30 % media y un 10 % alta. En el caso de los valores de cantidad de pruebas se obtiene que el 60 % de las clases poseen una cantidad de pruebas baja, el 30 % media y un 10 % alta. En cuanto a la reutilización se puede apreciar que el 60 % de las clases poseen una alta reutilización, el 30 % media y un 10 % baja.

En el siguiente gráfico se puede observar los resultados de la métrica.

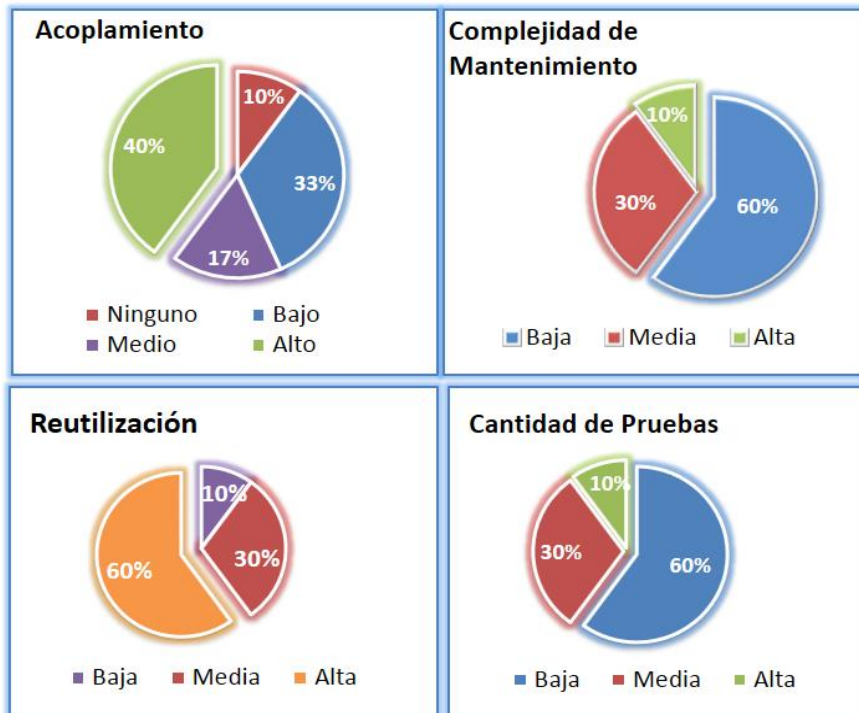


Figura 5: Resultado de la métrica RC. Fuente (Elaboración propia)

Estos valores obtenidos en el gráfico anterior, permiten medir de forma cuantitativa la calidad de los atributos internos del software. Esto permite evaluar la calidad durante el desarrollo del sistema. El acoplamiento consiste en el grado de dependencia o interconexión de una clase o estructura de clase,

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

con otras, está muy ligada a la característica de reutilización. El valor de la reutilización obtenido anteriormente, consiste en el grado reutilizable presente en una clase o estructura de clase, dentro del diseño del módulo implementado. La complejidad de mantenimiento, consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirectamente en el proyecto, pero lo que más afecta es la planificación del proyecto. Por otra parte, la cantidad de pruebas consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto, o sea del módulo implementado

3.1.2. Tamaño operacional de clases TOC

Para evaluar las métricas son necesarios los umbrales. Esta métrica se aplica a las mismas clases en las que fue aplicada la métrica RC. En este caso las clases se clasifican en tres grupos según la Cantidad de Procedimientos (CP), junto con los umbrales seleccionados para su clasificación (**ver Tabla 3**).

Nota: Cuanto menor sea el valor medio para el tamaño, más probable es que las clases existentes en el sistema se puedan reutilizar.

Tabla 19: Tamaño de Clases. Fuente (Elaboración propia)

No	Clases	No. Atributos	No. Operaciones
1.	BonificacionExtraController	0	9
2.	ConvocatoriaController	0	9
3.	EventoController	0	14
4.	EventoEstudianteController	0	5
5.	EventoExternoController	0	17
6.	EventoUciController	0	17
7.	EvidenciaController	0	10
8.	ExamenPremioController	0	11
9.	IndexController	0	15
10.	ProfesorEstudianteController	0	7
11.	PublicacionCientificaController	0	15
12.	ReportesController	0	4
13.	ResultadoController	0	13
14.	SolicitudParticipacionEventoController	0	10
15.	UsuarioController	0	8

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

16.	BonificacionExtra	4	7
17.	Categoria	3	5
18.	Convocatoria	5	8
19.	Evento	11	26
20.	EventoExterno	4	8
21.	EventoUci	3	9
22.	Evidencia	6	12
23.	ExamenPremio	2	4
24.	ExpedienteIntegralidad	11	27
25.	NomencladorAlcance	3	6
26.	NomencladorPais	3	6
27.	PublicacionCientifica	7	16
28.	Resultado	5	9
29.	SolicitudParticipacionEvento	5	9
30.	LoginListener	5	4
	Total	77	298

Tomando como muestra un total de 30 clases implementadas, seleccionadas por su importancia para el negocio en cuanto a funcionalidades, se obtiene un promedio de 2.56 atributos y 9.93 de operaciones. Para evaluar las métricas son necesarios los valores de los umbrales. Existen tres umbrales para esta métrica, a continuación, se muestran la cantidad de clases por clasificación

Tabla 20: Cantidad de clases por clasificación. Fuente (Fornaris, 2009)

Clasificación	Valores de los Umbrales	Cantidad de Clases
Pequeño	Cantidad de Procedimientos ≤ 20	27
Medio	Cantidad de Procedimientos $20 < \leq 30$	1
Grande	Cantidad de Procedimientos > 30	2

Se procede a calcular el promedio de las mismas y teniendo ambos valores, según los criterios expuestos en el Capítulo 1.

De acuerdo a los umbrales mostrados en la **(Tabla 16)** el 90 % de las clases son pequeñas, lo cual significa que estas clases se pueden reutilizar ampliamente. Al analizar los resultados obtenidos para cada uno de los atributos de calidad para esta métrica, se evidencia que el 63 % de las clases poseen

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

una responsabilidad baja, un 30 % media y el 7 % alta. Por otra parte, el 63 % de las clases presentan una complejidad baja, el 30 % media y un 7 % alta. Estos atributos de calidad son directamente proporcionales. En el caso de la reutilización, se obtiene que el 65 % de las clases poseen una alta reutilización, el 28 % media y un 7 % baja. En el siguiente gráfico (ver **Figura 6**) se observan los resultados de la métrica.

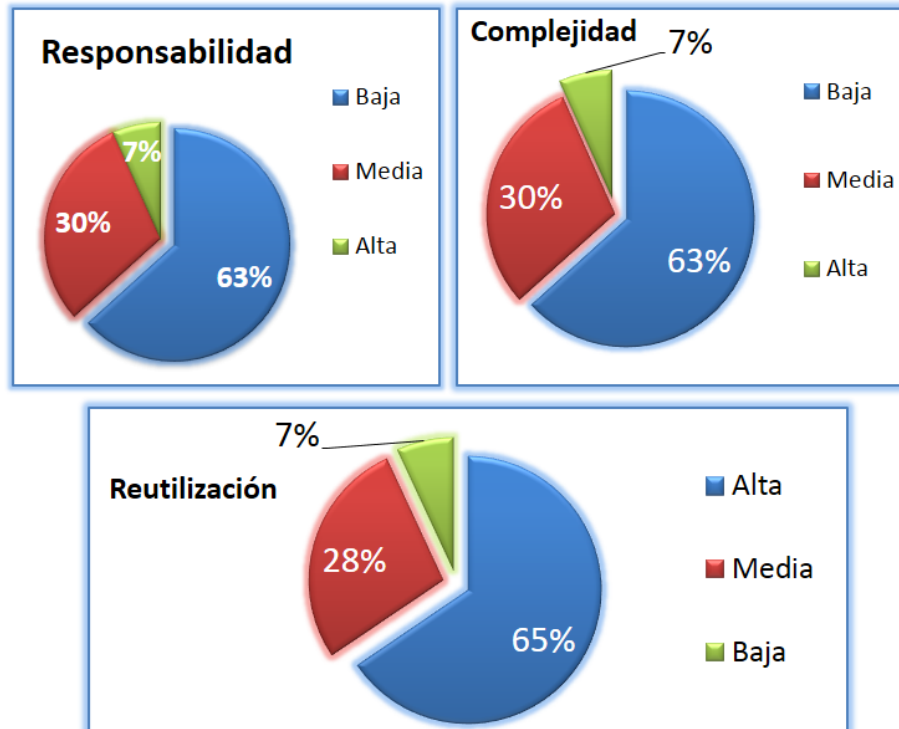


Figura 6: Resultado de la métrica TOC. Fuente (Elaboración propia)

Haciendo un análisis de los resultados obtenidos para los atributos de la métrica TOC se puede observar que el atributo reutilización cuenta con un porcentaje alto. Se demuestra que el componente cuenta con una elevada reutilización, baja responsabilidad y complejidad en el diseño propuesto. Por lo que se concluye que los resultados obtenidos en esta métrica son positivos.

3.2. Verificación del sistema

Las pruebas tienen como objetivo valorar y mejorar la calidad de los productos del trabajo generado durante el desarrollo y modificación del software. Según (Pressman, 2005) verificación es: *“el conjunto de actividades que aseguran que el software implemente correctamente una función específica, y la validación es un conjunto diferente de actividades que aseguran que el software*

construido corresponde y satisface los requisitos del cliente.”. Se tuvieron en cuenta los siguientes niveles de pruebas:

- ✓ Nivel de Unidad
- ✓ Nivel de Aceptación

3.2.1. Nivel de unidad

Las pruebas unitarias son aplicadas para verificar que el software cumple los requisitos funcionales y también son empleadas para asegurar la calidad del código entregado. Además, son la mejor forma de detectar fallas tempranamente en el desarrollo y está demostrado que mientras más pronto se encuentren los errores, menos costará corregirlos. A este nivel se realizan las pruebas de funcionalidad, utilizando los métodos de prueba de caja blanca y Caja Negra para comprobar que tanto el código como la interfaz no contengan errores y se ejecutan adecuadamente.

3.2.2. Pruebas de caja blanca

El método de pruebas de caja blanca precisa del acceso al código del programa de modo que se pueda comprobar su lógica interna. Al sistema desarrollado se le aplicaron tanto pruebas unitarias como de aceptación, tal y como propone XP. El código que se muestra en la **Figura 7**, se utilizó para la realización de la prueba de caja blanca. Este método es fundamental en el sistema, el mismo tiene que ver con la creación de un nuevo evento externo es el sistema.

```

1 public function createStaticAction(Request $request) {
    $entity = new EventoExterno();
    $form = $this->createForm(new EventoExternoType(), $entity);
    $form->bind($request);

2     if ($form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $eventoExterno = $em->getRepository('MeritoBundle:EventoExterno')
3         ->findOneBy(array('nombre' => $entity->getNombre()));
        if ($eventoExterno) {
4             return $this->render('MeritoBundle:EventoExterno:newStatic.html.twig', array(
                'entity' => $entity,
                'form' => $form->createView(),
                'errors' => 'Ya existe un "Evento UCI" con ese nombre'
            ));
        }
        $em->persist($entity);
        $em->flush();
5         return $this->redirect($this->generateUrl('evento_externo_home'));
    } $errors = $form->getErrors(true);
    $errors_toString = '';
6     foreach ($errors as $error) {
7         $errors_toString .= $error->getMessage() . '\n';
    }

8     return $this->render('MeritoBundle:EventoExterno:new.html.twig', array(
        'entity' => $entity,
        'form' => $form->createView(),
        'errors' => $errors_toString
    )); }

```

Figura 7: Fragmento de código perteneciente al método createStaticAction. Fuente: (Elaboración propia)

El método **createStaticAction** consiste en mostrar la página **EventoExterno:new.html.twig** enviándole varios datos gestionados para que sean procesados en dicha página. Para realizar esta operación el método recibe la entrada por el usuario de la creación del nuevo evento, y luego se obtienen los datos enviados desde el formulario creándose un objeto de tipo **EventoExterno**. En caso de que ocurra algún error de entrada, el método genera la excepción del determinado error, y finalmente se le da una respuesta a la página que envió los datos por Ajax.

Partiendo del fragmento de código tomado se obtuvo el siguiente grafo de flujo (ver Figura 8).

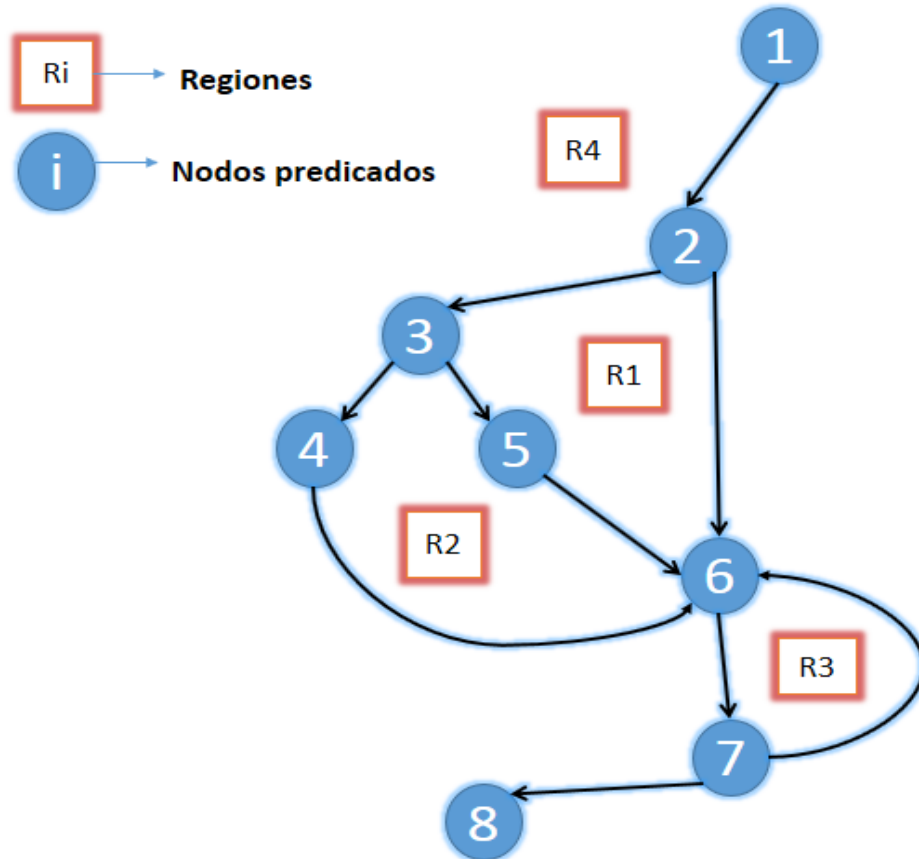


Figura 8: Grafo de flujo del fragmento de código de la Figura 7 Fuente: (Elaboración propia)

Luego de haber realizado la construcción del grafo de flujo se procede a calcular la complejidad ciclomática. En la fórmula $V(G)$ representa el valor del cálculo.

$$V(G) = (A - N) + 2$$

Donde **A** es el número de aristas y **N** es el número de nodos contenidos en el grafo.

$$V(G) = (10 - 8) + 2$$

$$V(G) = 4$$

El número de regiones del grafo es igual a la complejidad ciclomática. El cálculo efectuado anteriormente dio como resultado que la complejidad ciclomática es de cuatro. Este valor indica que existen cuatro posibles caminos por donde el flujo puede circular y determina el número de casos de prueba que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. A continuación, se representan los caminos básicos por los que puede transitar el flujo:

- **Camino básico 1:** 1 - 2 - 3 - 4 - 6 - 7 - 8
- **Camino básico 2:** 1 - 2 - 3 - 5 - 6 - 7 - 8
- **Camino básico 3:** 1 - 2 - 6 - 7 - 8
- **Camino 4:** 1 - 2 - 6 - 7 - 6 - 7 - 8

Luego de establecidos los caminos básicos se procede a realizar los casos de prueba para cada uno de ellos, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino. A continuación, se presenta un ejemplo (**ver Tabla 21**) de los cuatro casos de pruebas realizados a esta funcionalidad.

Tabla 21: Caso de prueba para el camino 3. Fuente (Elaboración propia)

Entrada	Introducir correctamente los datos del evento externo
Resultados Esperados	Que el evento no exista en la base datos, y que los parámetros introducidos sean válidos.
Condiciones	<pre> if (\$form->isValid()) { \$em = \$this->getDoctrine()->getManager(); \$eventoExterno = \$em->getRepository('MeritoBundle:EventoExterno')->findOneBy(array('nombre' => \$entity->getNombre())); if (\$eventoExterno) { return \$this->render('MeritoBundle:EventoExterno:newStatic.html.twig', array('entity' => \$entity, 'form' => \$form->createView(), 'errors' => 'Ya existe un "Evento UCI" con ese nombre')); } \$em->persist(\$entity); \$em->flush() return \$this->redirect(\$this->generateUrl('evento_externo_home')); } </pre>

3.2.3. Prueba de caja negra

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Para la realización de las pruebas de caja negra se empleó la técnica partición de equivalencia que garantizó efectividad al examinar los valores válidos e inválidos de las entradas existentes en el software. A continuación, se brinda un ejemplo (**ver Tabla 22**) de un caso de prueba de uno de los requisitos que componen la funcionalidad gestionar bonificación.

Tabla 22: Caso de prueba para gestionar bonificación Fuente (Elaboración propia)

Escenario	Descripción	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Adicionar bonificación con campos correctos	Permite asignarle a un estudiante el resultado obtenido en un evento	Válido	El sistema debe adicionarle la bonificación al estudiante, y actualizar su bonificación total.	Seleccione la opción Estudiantes, después Eventos Estudiantes y luego en la pestaña gestionar resultados
EC 1.2 Adicionar bonificación con campos incompleto	El sistema no permite adicionar porque existen campos obligatorios que no han sido seleccionados	Inválido	El sistema muestra un mensaje de ¡Error! Informando "Debe de seleccionar un resultado"	
EC 1.3 Adicionar bonificación con campos repetidos	El sistema no permite adicionar porque existen campos obligatorios que no han sido seleccionados	Inválido	El sistema muestra un mensaje de ¡Error! "Ya existe un campo igual para este estudiante en este evento"	

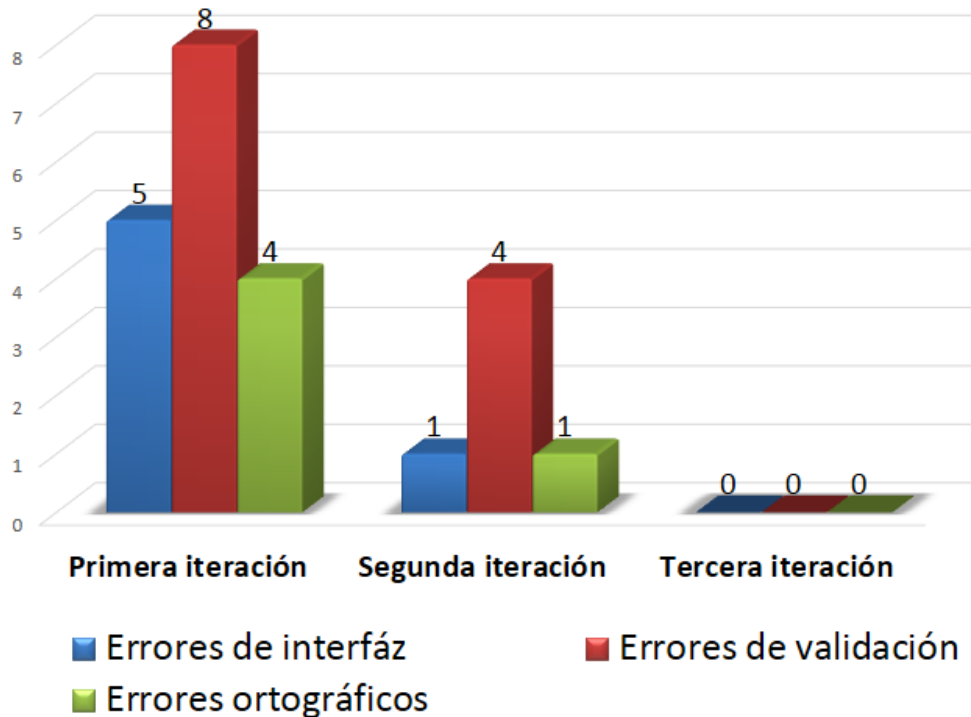


Figura 9: Cantidad de No Conformidades (NC) por iteraciones Fuente: (Elaboración propia)

De las no conformidades encontradas en la primera iteración, 8 errores eran de validaciones, estaban relacionadas con la validación de los datos que se introducen en el sistema, 5 errores de interfaces, y 4 errores ortográficos. En una segunda iteración se detectó un error de interfaz, 4 de validaciones y 1 error ortográfico. Se pudo comprobar en la tercera iteración realizada que estos errores fueron corregidos y que la aplicación desarrollada funciona correctamente.

Después de concluidas las pruebas por el grupo de calidad. Se liberó la aplicación módulo “Mérito Científico” para el Sistema de Certificación de Roles 2.0 en la disciplina de Práctica profesional, entregándose al equipo de desarrollo el Acta de Liberación Interna de Productos del Software en la que consta que la aplicación está apta para ser utilizada.

3.3. Validación del sistema

Con la validación del sistema se pretende comprobar que el software cumple las expectativas que el cliente espera. Para llevar a cabo la validación de la aplicación se aplicaron las pruebas de aceptación por cada historia de usuario definida en la etapa de planificación.

3.3.1. Nivel de aceptación

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

En este proceso se ejecutan pruebas dirigidas al sistema en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos definidos en el análisis y diseño. Las pruebas no pueden asegurar la ausencia de defectos, sólo pueden demostrar que existen errores. En la metodología XP las pruebas de aceptación son las destinadas a comprobar que la funcionalidad implementada sea la esperada por el cliente y usuarios finales de la aplicación. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación está apta para su uso y despliegue.

A cada una de las historias de usuario se le realizaron pruebas de aceptación. Estas pruebas son reflejadas mediante casos de pruebas de aceptación (**ver Tabla 23**), las cuales están conformadas por

siete parámetros, que dan información acerca de la prueba realizada. Los parámetros a medir son:

- **Código:** muestra un identificador para cada prueba realizada (normalmente se pone PA: pruebas de aceptación y seguido de las letras el nombre del componente).
- **Nombre de la historia de usuario:** indica el nombre de la prueba.
- **Descripción:** se describe cuál es la funcionalidad que se va a medir del componente al que se le esté realizando la prueba.
- **Condiciones de Ejecución:** indica cuáles son las condiciones que se tienen que cumplir para que el componente realice correctamente la funcionalidad que se va a medir.
- **Entrada / Pasos de ejecución:** indica los pasos a seguir para realizar la prueba.
- **Resultados esperados:** muestra cuál es el resultado que se obtendría de un correcto funcionamiento de la prueba.
- **Evaluación de la prueba:** indica el estado de la prueba si es satisfactoria o insatisfactoria

Tabla 23: Caso de prueba de aceptación. Fuente (Elaboración propia)

Caso de prueba de aceptación	
Código: PA9-HU9	Historia de usuario: Crear solicitud para participación en evento externo
Nombre de la persona que realiza la prueba: Maybel Díaz Capote	

Descripción: Se crea una nueva solicitud informando la participación en un evento.

Condición de ejecución: El estudiante debe estar autenticado en la aplicación.

Entrada/Pasos de ejecución:

1. El usuario presiona el botón solicitud de participación en evento externo.
2. El usuario introduce el nombre del evento.
3. El usuario introduce la descripción del evento.
4. El usuario presiona el botón **salvar y adicionar**.

Resultado esperado: Se envía una solicitud al Profesor Principal de Año

En la realización de pruebas de aceptación se llevaron a cabo dos iteraciones. En la primera iteración se evaluó un total de 17 casos de pruebas, de ellos 13 fueron satisfactorios pues no se detectaron no conformidades. Tres fueron insatisfactorios debido a que presentaban no conformidades y uno no fue evaluado puesto que no se encontraba en funcionamiento. En la segunda iteración, todos los requisitos fueron evaluados de satisfactorios, quedando culminada las pruebas de aceptación

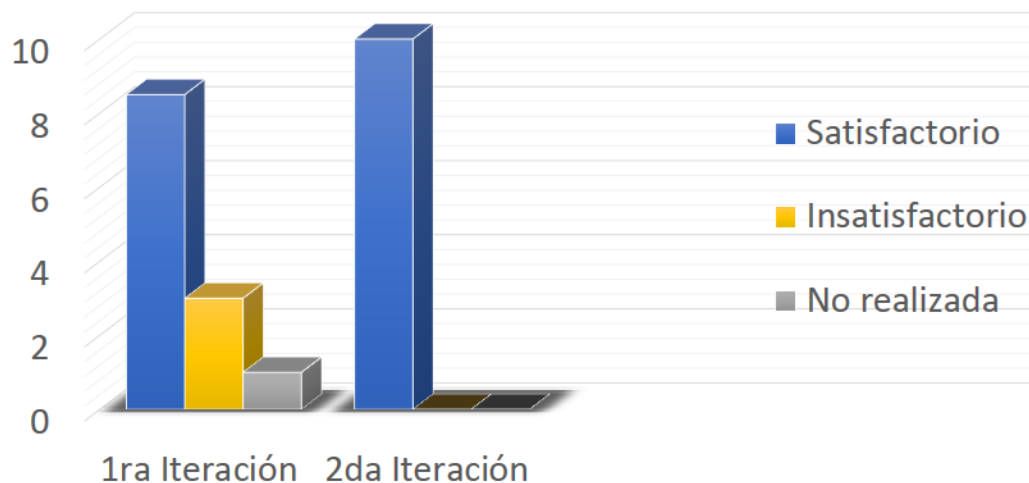


Figura 10: Pruebas de aceptación Fuente: (Elaboración propia)

3.4. Conclusiones parciales

Mediante las métricas para validar el diseño, las pruebas de aceptación, las pruebas de unidad, y realizando casos de pruebas a todas las funcionalidades del sistema en tres iteraciones, se detectaron todos los errores que presentaba el sistema para darle así solución a los mismos, luego se llegaron a las conclusiones que:

- ✓ La aplicación de métricas de diseño, demostró que las clases del diseño poseen bajo acoplamiento, que existe además una baja responsabilidad y complejidad de implementación y una alta reutilización en el diseño propuesto para el módulo mérito científico.
- ✓ Las pruebas realizadas al sistema, tanto de unidad como de aceptación arrojaron resultados satisfactorios, permitiendo afirmar que la implementación del sistema esta apta para su uso.

CONCLUSIONES GENERALES

Con la realización del presente trabajo de diploma se desarrolló un módulo que facilitará la gestión de los estudiantes merecedores del premio al Mérito Científico durante los estudios de su carrera en la Universidad de las Ciencias Informáticas. La bonificación generada en la participación de eventos y las evidencias obtenidas. Es por ello que se puede concluir afirmando que:

- ✓ Mediante el marco teórico de la investigación se identificaron las herramientas necesarias, los lenguajes de programación y los artefactos que con su uso adecuado permitieron dar solución a la propuesta planteada.
- ✓ Las tecnologías y herramientas seleccionadas, así como la utilización de XP como metodología de desarrollo, en correspondencia con los criterios de selección definidos, contribuyeron a la obtención de la solución informática módulo Mérito Científico para la problemática identificada.
- ✓ Con la utilización de patrones de diseño y patrones arquitectónicos, se logró implementar el sistema módulo Mérito Científico de acuerdo a los estándares y modelos utilizados en el desarrollo de software que responden a las necesidades del cliente.
- ✓ Mediante el diseño y aplicación de los casos de prueba se logró valorar los resultados y verificar que las funcionalidades del módulo Mérito Científico cumplieran con las expectativas y necesidades del cliente.

RECOMENDACIONES

A partir de los resultados obtenidos en el desarrollo del trabajo de diploma se recomienda para próxima versión:

- Incluirle al Sistema de Certificación de Roles 2.0, un sistema de alertas tempranas con el propósito de contar con la identificación de posibles acciones correctivas que permitan informar a los estudiantes registrados en el sistema su desempeño académico.

REFERENCIAS

- Bonanata, Maximiliano. 2013.** Programación y algoritmos S.L. Argentina : MP ediciones S.A, 2013. pág. 17. 2013.
- Calleja, M. A and Arias, Manuel. 2009.** Carmen. Etándares de codificación. [En línea] 7 de Mayo de 2009. [Citado el: 27 de Abril de 2015.] <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>.
- Calleja, Manuel Arias. 2010.** Análisis, diseño y mantenimiento del software. [En línea] Noviembre de 2010. <http://www.ia.uned.es/ia/regladas/adms/GuiaDidactica.pdf>.
- Casas, Sandra and Reinaga, Héctor. 2009.** Aspectos Tempranos: Un enfoque basado en tarjetas CRC. Argentina : s.n., 2009.
- Castellanos, S.B. 1997.** Curso Intensivo de Investigación Científica. La Habana : Instituto Pedagógico Latinoamericano., 1997. Folleto,.
- CESAR, P. M. A. A. M. D. 2012.** Tipología y Clasificación de eventos. 2012.
- Chacón, Jesús Mares, [ed.].** Metodología de la Investigacion_5ta_Edición. pág. 613. ISBN: 978-607-15-0291-9.
- Cristiá, Maximiliano. 2011.** Introducción a la ingeniería de requerimientos. [En línea] Agosto de 2011. [Citado el: 1 de Junio de 2016.] <http://www.fceia.unr.edu.ar/~mcristia/publicaciones/ingreq-a.pdf>.
- Desarrollo web, D. 2013.** DSW. [En línea] 2013. [Citado el: 24 de Abril de 2015.] <http://www.desarrolloweb.com/articulos-copyleft/articulo-metricas-de-software.html>.
- Dr Silva, Pedro Horruitiner. 2008.** LA UNIVERSIDAD CUBANA MODELO DE FORMACIÓN. [ed.] Pedro Horruitiner Silva. La Habana : Editorial Universitaria., 2008. SBN 978-959-16-0676-1.
- Drae. 2014.** Real Academia Española. lema.rae.es. [En línea] Octubre de 2014. [Citado el: 18 de Febrero de 2016.] <http://lema.rae.es/drae/srv/search?id=jadYhVT31DXX2vknwZ9Y>.
- Drucker, Peter. 1954.** The Practice of Management. Austria : Collins, 1954. ISBN: 0887306136.
- Eguíluz, Javier. 2016.** symfony.es. [En línea] En Línea, 2016. <http://www.symfony.es/>.
- Fabien Potencier, Francois Zaninotto. 2008.** librosweb.es. [En línea] 30 de Diciembre de 2008. [Citado el: 24 de enero de 2016.] www.librosweb.es.
- Fornaris, Sánchez, Maite and Rabí, Dayanis Alcantara. 2009.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas. 2009.
- Foundation, The Apache Software. 2011.** Documentación del Servidor HTTP Apache 2.0. s.l. : Documentación del Servidor HTTP Apache 2.0, 2011.
- Gaceta, oficial. 2014.** Órgano científico estudiantil de Ciencias Médicas. [En línea] 9 de Diciembre de 2014. http://www.rev16deabril.sld.cu/index.php/16_04/pages/view/resol.
- García, Adolfo Arreola. 2013.** [En línea] 2013. [Citado el: 26 de Marzo de 2015.] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

General, I. 2015. Definición de Evidencia. [En línea] 7 de Enero de 2015. [Citado el: 17 de Marzo de 2016.] <http://conceptodefinicion.de/evidencia/>.

Gestión Institucional Basada en la Evidencia.: **Molina, Ana. Sep,2004.** s.l. : Print version, Sep,2004, Revistas de Ciencias Administrativas y Financieras de la Seguridad Social, Vol. vol.12. ISSN 1409-1259.

González, Asenjo, Andrés, Diego and Alejandro Ríos Peña. 2003. Patrones de Diseño. Habana : s.n., 2003.

Hernández, Rolando Alfredo. 11,2002. Metodología de la Investigación Científica. Ciudad de la Habana : EDUNIV, 11,2002. pág. 114. ISBN: 959-16-0343-6.

IEEE. 2015. [En línea] 2015. http://www.computer.org/cms/Computer.org/Computer.org/s2esc/s2esc_pols/SP-.

INC, S. 2013. Organización de eventos. [En línea] 2013. <http://www.scribd.com/doc/14686867/Organizacionde-eventos>.

Investigación, Científica. 2015. Concepto.de. Concepto.de. [En línea] 2015. <http://concepto.de/investigacion-cientifica/#ixzz47XVpm2WL>.

Jeremy. 2014. Características de Css3. [En línea] 2014. [Citado el: 10 de Abril de 2016.] <http://www.css3.com>.

Joskowicz, José. 2008. Reglas y Prácticas en eXtreme Programming. España : Senior Member, 2008. IEEE : 90334128.

Kabir, Mohammed J. 2003. La Biblia Server Apache. 2003. pág. 815.

León, Hernández. 2011. El Proceso de Investigación Científica. Ciudad de La Habana : s.n., 2011.

León, Universidad de. 2016. servicio de gestión académica. [En línea] 2016. <https://www.unileon.es/universidad/servicios-administrativos/servicio-de-gestion-academica>.

Letelier, Patricio, Canós, José H. & Penadés, M^a Carmen. 2003. Metodologías ágiles en el desarrollo de software: Extreme Programming (XP). 2003.

Lincke, Lundberg and W., J. y Löwe. 2009. Comparing software metrics tools. International Symposium on Software Testing and Analysis. Berlin : s.n., 2009.

Ministerio, de Educación Superior. 2015. Resolución 79/2015. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. pág. 3, Resolución.

Mora, Pérez, Oscar and Marc, Gibert Ginestà. 2014. Bases de datos en PostgreSQL. s.l. : s.l.:UOC, 2014.

Moral, Juan Antonio Breña. 2004. Desarrolloweb.com. [En línea] 2004. [Citado el: 20 de Enero de 2016.] <http://www.desarrolloweb.com>.

Niska, Christoffer. 2014. Extending Bootstrap. s.l. : Published by Packt Publishing Ltd., 2014. ISBN 978-1-78216-841-6.

- Otto, Jacob Mark. 2014.** Bootstrap. [En línea] 2014. [Citado el: 11 de Abril de 2015.] <http://getbootstrap.com/>.
- Pacheco, N. 2013.** Manual de Twig. [En línea] 2013. [Citado el: 3 de Abril de 2015.] <http://gitnacho.github.io/Twig/>.
- Polo, Macario and Villafranca, Daniel. 2008.** [En línea] 2008. [Citado el: 20 de Abril de 2015.] <http://www.infcr.uclm.es/www/mpolo/asig/0708/tutorJavaWeb.pdf>.
- Postgresql. 1996-2013.** [En línea] 1996-2013. <http://www.postgresql.org/about/>.
- Potencier, Fabien. 2010.** Symfony 2.1, el Libro oficial. 2010.
- Pressman, Roger S. 2007.** Ingeniería de Software. Un enfoque práctico 6ta edición. Nueva York : McGraw-Hill, 2007. pág. 384. 0072853182.
- . 2005. Ingeniería del SoftWare Quinta edición. España : s.n., 2005.
- 2014.** Quality and technology. Verificación y Validación de las Entregas . [En línea] 5 de Noviembre de 2014. <http://www.calidadytecnologia.com/2014/11/verificacion-validacion-calidad-entrega-plan-proyecto.html>.
- RAE, La Real Academia Española. 2010.** quees.la. [En línea] 2010. <http://quees.la/evento/>.
- Real Academia Española. 2015.** WordReference.com. [En línea] 2015. <http://www.wordreference.com/es/en/frames.aspx?es=galard%C3%B3n>.
- Roberth, Figueroa y Solís Camilo, Cabrera Armando. 2007.** Metodologías tradicionales y metodologías ágiles. 2007.
- Santos, Félix Óscar Rubio García and Crescencio Bravo. (2009-2010).** Escuela de Ingeniería Civil Informática. [En línea] (2009-2010). http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/Metodologias.pdf.
- Schenone, Marcelo Hernán. 2004.** Diseño de una Metodología Ágil de Desarrollo de. 2004.
- Sommerville, Ian. 2005.** Ingeniería de Software. s.l. : Septima Edición, 2005.
- Tecnología. 2012.** Jojooa - tecnología, marketing y crm. [En línea] 2012. [Citado el: 7 de Junio de 2016.] <https://sites.google.com/site/jojooa/analisis-de-sistemas/definicion-de-actor-que-es-un-actor>.
- Velthunis, Mario G Piattini. 2010.** Análisis y diseño de aplicaciones informáticas de gestión. Madrid : s.n., 2010.
- Vinculando, Reevista. 2010.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 4 de Enero de 2010. [Citado el: 8 de Abril de 2015.] <http://vinculando.org/?s=m%C3%A9tricas+para+requisitos>.
- Visconti, Astudillo Marcello and Hernán. 2004.** Fundamentos de Ingeniería de Software. [En línea] 9 de Septiembre de 2004. [Citado el: 26 de Abril de 2015.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.

ANEXOS

Anexo 1: Historias de usuario

Historia de Usuario	
Número: 1	Nombre del requisito: Insertar eventos externos.
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 2 Semana (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 2 Semanas
Descripción: El sistema debe permitir al usuario con el rol de Profesor Principal de Año, insertar un evento externo siempre y cuando este evento tribute al mérito científico.	
Observaciones: Cada usuario con el rol Profesor Principal de Año, debe de estar autenticado en el sistema.	

Historia de Usuario	
Número: 2	Nombre del requisito: Modificar eventos externos.
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 Semana (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos real: 1 Semanas
Descripción: El sistema debe de permitir al usuario con rol de Profesor Principal de Año, modificar un evento siempre y cuando exista el evento en la lista de eventos.	
Observaciones: Debe de existir el evento en la lista de eventos externos	

Historia de Usuario	
Número: 3	Nombre del requisito: Salvar datos modificados del evento.
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe de permitir al usuario con rol de Profesor Principal de Año, modifique un evento determinado, teniendo en cuenta que no puede quedarse un campo vacío, y la fecha inicial no puede ser mayor que la fecha final.	
Observaciones: Solo pude realizar esta funcionalidad el Profesor Principal de Año	

Historia de Usuario	
Número: 4	Nombre del requisito: Mostrar eventos externos
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe de mostrar un listado de eventos con el nombre, la temática, alcance, país, estado y un conjunto de acciones en las cuales incluye adicionar, eliminar, modificar y pasar el evento a un estado de cerrado.	

Observaciones: Solo el estudiante puede ver el evento y subir evidencia. Una vez finalice el evento externo el Profesor Principal de Año con el rol de Profesor Principal de Año, debe revisar la evidencia y actualizar el resultado obtenido en el evento

Historia de Usuario	
Número: 5	Nombre del requisito: Eliminar eventos externos
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe de permitir al usuario con rol de Profesor Principal de Año, eliminar un evento determinado, siempre y cuando el evento exista dentro del listado de eventos	
Observaciones: Debe de existir al menos un elemento en la lista de eventos	

Historia de Usuario	
Número: 6	Nombre del requisito: Cerrar evento
Modificación de historia de usuario número: 1	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 Semanas

Descripción: Permite cambiar el estado de un evento, informándole al usuario que ya este evento no tributa al Mérito Científico, pero sin embargo puede continuar subiendo evidencias para ser evaluadas, ya que se incluyen en el expediente de integralidad.

Observaciones: Se debe estar autenticado para realizar esta acción

Historia de Usuario	
Número: 7	Nombre del requisito: Realizar búsqueda de eventos externos
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año y Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: De acuerdo al criterio que seleccione es como se debe realiza la búsqueda, puede ser por el nombre del evento, o por el estado del evento. Luego el sistema muestras los posibles resultados. Los estados de un evento pueden ser (Aprobado, no aprobado) .	
Observaciones: Debe de existir al menos un evento en el listado de eventos, para que la búsqueda tenga éxito. Además, este requisito es parejo para los estudiantes y profesores.	

Historia de Usuario	
Número: 8	Nombre del requisito: Mostrar listado de los eventos externos realizados
Modificación de historia de usuario número: ninguna	
Usuario: Estudiantes	Iteración asignada: 1
Prioridad: Alta	Puntos estimados: 2

(Alta / Media / Baja)	(1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio	Puntos reales: 1 semana
(Alto / Medio / Bajo)	
Descripción: El sistema debe permitirle al usuario con el rol de Estudiante, mostrar los eventos en los que haya subido al menos una evidencia.	
Observaciones: El estudiante solo puede ver los eventos asociados a su participación, aunque existan más eventos en el sistema	

Historia de Usuario	
Número: 9	Nombre del requisito: Crear solicitud para participación en evento externo
Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media	Puntos estimados: 1
(Alta / Media / Baja)	(1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo:	Puntos reales: 2
(Alto / Medio / Bajo)	
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, crear una solicitud para participar en un evento externo. Si esta solicitud se aprueba por el Profesor Principal de Año. El profesor debe de crear dicho evento en el sistema para que el estudiante lo evidencie y así poder emitir una nota.	
Observaciones: La solicitud solo es efectiva para los estudiantes asociados a un Profesor Principal de Año.	

Historia de Usuario	
Número: 10	Nombre del requisito: Eliminar solicitud de participar en evento

Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1 Semana
Descripción: El sistema debe permitirle al usuario con el rol de Estudiante, una vez listadas todas las solicitudes de participación, seleccionar una y debe accionar sobre la funcionalidad de eliminar.	
Observaciones: Se elimina siempre que haya al menos una solicitud en la lista de solicitudes.	

Historia de Usuario	
Número: 11	Nombre del requisito: Listar solicitudes realizadas
Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Bajo (Alto / Medio / Bajo)	Puntos reales: 1 Semana
Descripción: El sistema debe permitirle al usuario con el rol de Estudiante que pueda listar todas las solicitudes.	
Observaciones: Solo se mostrarán las que haya realizado.	

Historia de Usuario

Número: 12	Nombre del requisito: Crear evento uci
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 Smenas
Descripción: El sistema debe permitirle al usuario con el rol de Profesor Principal de Año, crear un nuevo evento UCI.	
Observaciones: Se deben de estar autenticado con el rol de Profesor Principal de Año	

Historia de Usuario	
Número: 13	Nombre del requisito: Modificar evento uci
Modificación de historia de usuario número: ninguna	
Usuario: Leonardo Torres Silega	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe de mostrar un listado de eventos con un conjunto de acciones, las cuales incluye adicionar, eliminar, modificar y pasar el evento a un estado de cerrado. Luego de seleccionar la opción de modificar, debe de realizar el cambio deseado y luego dar clic en salvar datos.	
Observaciones: Si los datos no se salvan, el sistema no actualiza la información modificada.	

Historia de Usuario	
Número: 14	Nombre del requisito: Agregar solicitud de publicación científica
Modificación de historia de usuario número: 2	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, registrar solicitud de participación en evento del tipo publicación científica.	
Observaciones: El estudiante debe de esperar a que sea aprobada la publicación, inicialmente se encuentra en un estado no aprobada .	

Historia de Usuario	
Número: 15	Nombre del requisito: Gestionar evidencias por cada evento
Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, registrar evidencias por participación en eventos.	
Observaciones: Debes de estar autenticado	

Historia de Usuario	
Número: 16	Nombre del requisito: Aprobar solicitud
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, aprobar una solicitud tanto de participación en eventos, como de publicaciones científicas o exámenes de premio.	
Observaciones: Debes de estar autenticado.	

Historia de Usuario	
Número: 17	Nombre del requisito: Buscar resultados de eventos uci
Modificación de historia de usuario número: ninguna	
Usuario: Estudiante	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Bajo (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con el rol de Estudiante, buscar los resultados obtenidos en sus eventos, o sea se muestran los eventos, y sus respectivas evidencias.	
Observaciones: Se debe estar autenticado y además hay que tener al menos un resultado en eventos UCI	

Historia de Usuario	
Número: 18	Nombre del requisito: Asignar eventos a estudiante
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración Asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, asignar eventos a los estudiantes. Para ello el estudiante debe de registrar una evidencia al evento participado.	
Observaciones: Debe de estar autenticado en el sistema, y tener evidencias registradas.	

Historia de Usuario	
Número: 19	Nombre del requisito: Eliminar eventos a estudiante
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, eliminar eventos a los estudiantes. Cuando se le elimina un evento a un estudiante, automáticamente el sistema debe de eliminar la bonificación que tenía hasta el momento por el evento.	
Observaciones: Debe de existir el estudiante, y al menos un evento asociado a ese estudiante	

Historia de Usuario	
Número: 20	Nombre del requisito: Enviar alerta con la convocatoria
Modificación de historia de usuario número:	
Usuario: Estudiante	Iteración asignada: 3
Prioridad: Baja (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con el rol Profesor Principal de Año, lanzar convocatorias en las cuales los estudiantes suben sus evidencias por participación. Estas convocatorias no se archivan en el expediente, solo es una visión al estudiante de los eventos en los que puede participar para el Mérito Científico. Luego el Profesor Principal de Año es quien evalúa la evidencia y emite al sistema los resultados	
Observaciones: Cualquier usuario con el rol de Profesor Principal de Año puede enviar alertas de convocatorias, el sistema le muestra estas alertas a todos los estudiantes registrados.	

Historia de Usuario	
Número: 21	Nombre del requisito: Mostrar desempeño por cada estudiante
Modificación de historia de usuario número:	
Usuario: Profesor Principal de Año	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales: 1

Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, mostrar el desempeño de los estudiantes. Para un análisis de cuán importante es

Observaciones: Solo muestra el desempeño de los estudiantes que pertenecen a un mismo año dependiendo del profesor que los atiende.

Historia de Usuario	
Número: 22	Nombre del requisito: Adicionar_bonificacion_evento
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada:
Prioridad: Baja (Alta / Media / Baja)	Puntos estimados: 3 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con el rol de Profesor Principal de Año, adicionar bonificación. Para esta acción se debe primero seleccionar el evento y luego el o los estudiante que participaron.	
Observaciones: Para asignar la bonificación se debe revisar la evidencia subida por el estudiante para ese evento.	

Historia de Usuario	
Número: 23	Nombre del requisito: Gestionar bonificaciones extras
Modificación de historia de usuario número: 3	
Usuario: Profesor Principal de Año	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)

Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con el rol de Profesor Principal de Año, adicionar a un estudiante determinado una bonificación por criterio docente o de cualquier otra índole, siempre y cuando el estudiante sea merecedor de dicha bonificación.	
Observaciones: Se le actualizaría la bonificación obtenida hasta el momento al estudiante, y dicha acción solo la puede realizar el Profesor Principal de Año	

Historia de Usuario	
Número: 24	Nombre del requisito: Gestionar examen de premio
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, crear, modificar o eliminar exámenes de premio.	
Observaciones: El sistema debe de tener al menos un evento, para poder modificarlo o eliminarlo en este caso.	

Historia de Usuario	
Número: 25	Nombre del requisito: Mostrar estudiantes candidatos al mérito
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 3

Prioridad: Baja (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con el rol de Profesor Principal de Año, listar todos los estudiantes que optan por el premio al mérito científico, con sus respectivas bonificaciones obtenida hasta el momento. Informándole al Profesor Principal de Año cuales son los estudiantes que, hasta el momento, tiene más posibilidades de obtener el mérito científico.	
Observaciones: Solo el Profesor Principal de Año puede realizar esta actividad	

Historia de Usuario	
Número: 26	Nombre del requisito: Eliminar evento uci
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 2
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, eliminar algún evento del listado de eventos.	
Observaciones: Debe de existir al menos un evento en la lista.	

Historia de Usuario	
Número: 27	Nombre del requisito: Mostrar reporte del sistema

Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 3
Prioridad: Baja (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El reporte del sistema, dará al Profesor Principal de Año una constancia de como es que se comporta el Mérito Científico hasta el momento. Dará información sobre: cantidad de estudiantes inscrito al mérito científico, cantidad de eventos en el sistema tanto internos como externos, el estudiante de mayor bonificación hasta el momento y la cantidad de Profesores Principales de Año en el sistema.	
Observaciones:	

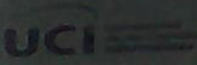
Historia de Usuario	
Número: 28	Nombre del requisito: Adicionar evidencias al expediente
Modificación de historia de usuario número: ninguna	
Usuario: No	Iteración asignada: 2
Prioridad: Media (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Media (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: Esta funcionalidad la hace el mismo sistema, una vez que el Profesor Principal de Año asigne un evento a un estudiante, ese evento el sistema lo registra en el expediente incorporándole su evidencia.	
Observaciones: Una vez que el evento se archive en el expediente, el estudiante puede borrar si lo desea el evento de su perfil.	

Historia de Usuario	
Número: 29	Nombre del requisito: Exportar expediente
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 3
Prioridad: Baja (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Baja (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, exportar el expediente si lo desea para su uso docente.	
Observaciones: Debe de seleccionar el estudiante.	


Historia de Usuario	
Número: 30	Nombre del requisito: Inscribirse al mérito científico
Modificación de historia de usuario número: 3	
Usuario: Estudiante	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 2 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1
Descripción: El sistema debe permitir al usuario con rol Estudiante, una vez que entre al módulo se le muestra un cartel que le informa si desea optar por el Mérito Científico.	
Observaciones: No depende de este cartel para realizar las tareas del Mérito Científico, ya que aunque no opte es importante para el expediente de Integralidad.	

Historia de Usuario	
Número: 31	Nombre del requisito: Asignar estudiante a profesor Profesor Principal de Año
Modificación de historia de usuario número: ninguna	
Usuario: Profesor Principal de Año	Iteración asignada: 1
Prioridad: Alta (Alta / Media / Baja)	Puntos estimados: 1 (1 punto=1 semana / 2 punto=2semanas / ..etc.)
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2
Descripción: El sistema debe permitir al usuario con rol Profesor Principal de Año, asignar los estudiantes que pertenecen al año que atiende.	
Observaciones: Debe de entrar en el sistema al menos una vez, para que el sistema guarde tu información y esta sirva para la selección por el usuario Profesor Principal de Año.	

Anexo 2: Acta de liberación interna del producto de software.



FACULTAD # 3
CENTRO DE GOBIERNO ELECTRONICO



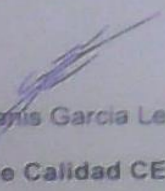
Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: 13/06/2016

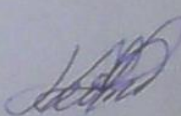
Emitida a favor de: Módulo "Mérito Científico" para el sistema de certificación de roles en la disciplina de Práctica profesional.

Datos del producto

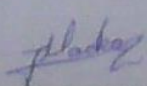
Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App	1.0	0	3	Evaluación dinámica Pruebas de Funcionalidad	13/06/2016




Ing. Yordanis Garcia Leiva
Asesor de Calidad CEGEL



Leonardo Torres Silega
Autor



Jorien Machado Suastegui
Responsable de la liberación



ANEXO 3: Tarjetas CRC (Clase, Responsabilidad y Colaboración)

Tarjeta 1

PublicacionCientificaController	
<p>Responsabilidad</p> <ol style="list-style-type: none"> 1. Adicionar publicación científica. 2. Modificar publicación científica. 3. Eliminar publicación científica. 4. Aprobar publicación científica. 5. Listado de publicaciones científica por estudiante. 6. Visualizar publicaciones científicas. 7. Verificar si existe publicación científicas. 	<p>Colaboradores</p> <ol style="list-style-type: none"> 1. Persona 2. Usuario 3. Publicación científica 4. Rol

Tarjeta 2

BonificacionExtraController	
<p>Responsabilidad</p> <ol style="list-style-type: none"> 1. Asignar bonificación extra a estudiante. 2. Editar bonificación extra. 3. Eliminar bonificación extra. 4. Mostrar bonificación extra asignada. 	<p>Colaboradores</p> <ol style="list-style-type: none"> 1. Resultao 2. Persona 3. expediente

Tarjeta 3

ConvocatoriaController	
Responsabilidad <ol style="list-style-type: none"> 1. Crear convocatoria. 2. Mostrar convocatorias. 3. Listar convocatorias. 4. Eliminar convocatoria. 5. Actualizar convocatorias. 6. Buscar convocatorias. 	Colaboradores <ol style="list-style-type: none"> 1. Persona 2. Convocatoria 3. SolicitudParticipacionEvento

Tarjeta 4

EventoController	
Responsabilidad Obtener la información de los eventos internos y externos	Colaboradores <ol style="list-style-type: none"> 1. Estudiante 2. Evento externo 3. Evento uci

Tarjeta 5

EventoEstudianteController	
Responsabilidad	Colaboradores

<ol style="list-style-type: none"> 1. Asignar eventos a estudiante. 2. Mostrar los eventos asignados. 3. Eliminar eventos asignados. 4. Listar eventos no asignados. 5. Actualizar resultado por evento. 	<ol style="list-style-type: none"> 1. Estudiante 2. Eventos 3. Evidencia
---	---

Tarjeta 6

EventoExternoController	
<p>Responsabilidad</p> <ol style="list-style-type: none"> 1. Adicionar evento externo. 2. Eliminar evento externo. 3. Mostrar evento externo. 4. Listar eventos externos por estudiante. 5. Editar evento. 6. Buscar eventos externos. 	<p>Colaboradores</p> <ol style="list-style-type: none"> 1. Evento 2. Estudiante 3. Examen de premio

Tarjeta 7

EventoUciController	
Responsabilidad <ol style="list-style-type: none"> 1. Adicionar evento uci. 2. Eliminar evento uci. 3. Mostrar evento uci. 4. Listar eventos uci por estudiante. 5. Editar uci. 6. Buscar eventos uci. 	Colaboradores <ol style="list-style-type: none"> 1. Evento 2. Estudiante 3. Examen de premio

Tarjeta 8

EvidenciaController	
Responsabilidad <ol style="list-style-type: none"> 1. Adicionar evidencia por evento. 2. Mostrar evidencias. 3. Listar evidencias. 4. Buscar evidencia 5. Editar evidencia 	Colaboradores <ol style="list-style-type: none"> 1. Evidencia 2. Persona 3. Expediente

Tarjeta 9

ExamenPremioController	
Responsabilidad <ol style="list-style-type: none">1. Crear examen de premio.2. Eliminar examen de premio.3. Mostrar examen de premio.4. Listar exámenes de premio.5. Buscar examen de premio.6. Editar examen de premio	Colaboradores <ol style="list-style-type: none">1. Evento uci2. Persona.

Tarjeta 10

ExpedienteIntegralidadController	
Responsabilidad <ol style="list-style-type: none">1. Gestionar expediente.2. Crear expediente.3. Mostrar expediente.4. Filtrar expediente.	Colaboradores <ol style="list-style-type: none">1. Expediente2. Usuario