

*Universidad de las Ciencias Informáticas*

*Facultad 3*



*Sistema para la gestión de los indicadores de Ciencia,*

*Técnica e Innovación en la Facultad 3*

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

***Autor:***

*Yasiel Torres Hernández*

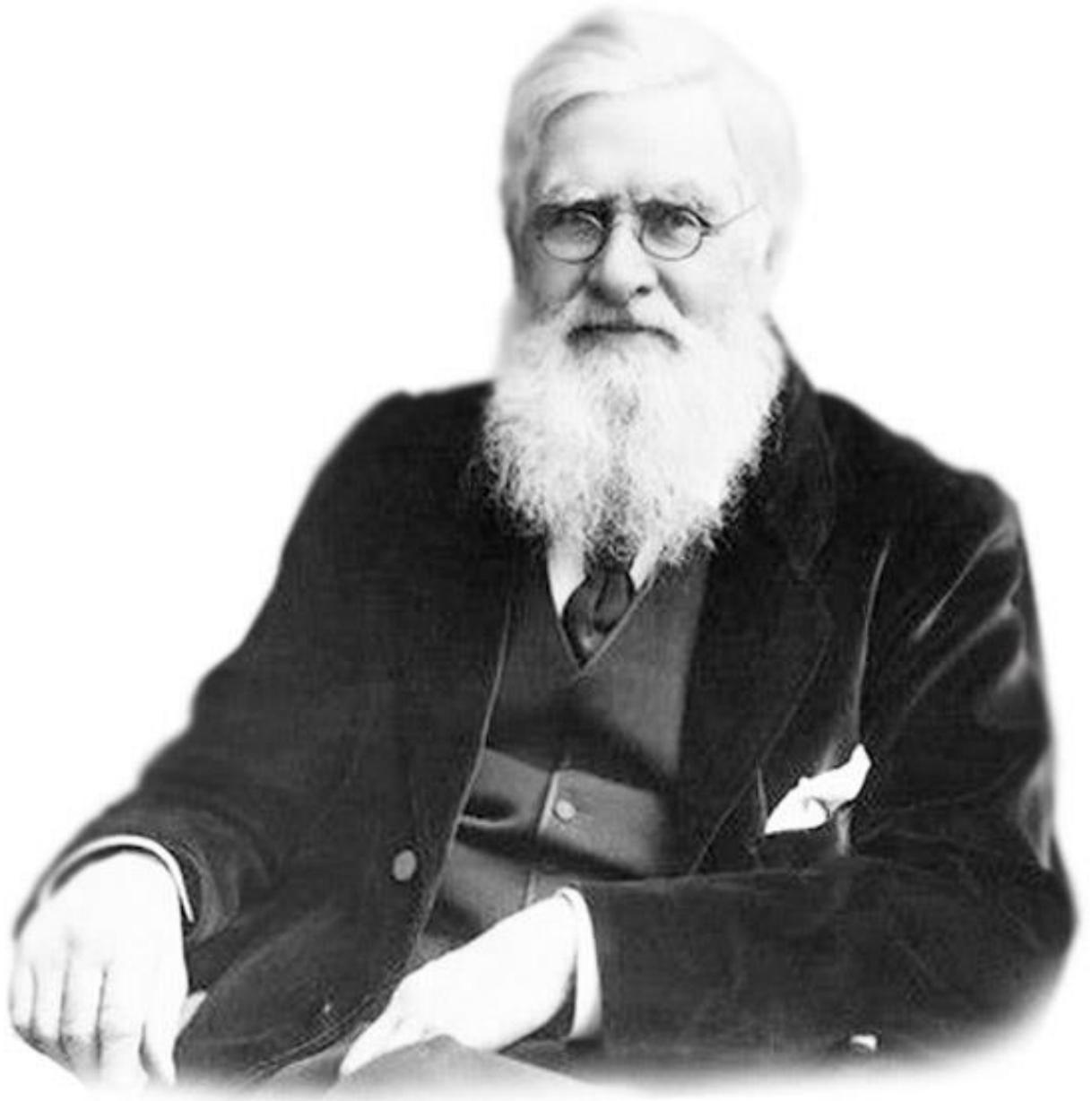
***Tutores:***

*Ing. Daniel Varona Cordero*

*Ing. Ana Cecilia Labrador Valdés*

*La Habana, junio de 2015*

*“Año 57 de la Revolución”*



*“No es el más fuerte ni el más inteligente el que sobrevive,  
sino aquel que más se adapta a los cambios”*

*Charles Darwin*

**DECLARACIÓN DE AUTORÍA:**

Declaro ser el 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 firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2015.

\_\_\_\_\_  
Firma del tesista  
Yasiel Torres Hernández

\_\_\_\_\_  
Firma del tutor  
Ing. Ana Cecilia Labrador Valdés

\_\_\_\_\_  
Firma del tutor  
Ing. Daniel Varona Cordero

***Dedicatoria***

*Le dedico esta tesis, primero que todo, a todas las personas que me quieren y me admiran, que desde el comienzo de mi vida me han dado su apoyo y cariño, especialmente a mis padres Gisela y Antonio por darme todo el amor del mundo y a mi hermano Yeinier por ser el regalo más grande que he recibido en la vida.*

## *Agradecimientos*

*A mis padres que son la fuerza y el ejemplo que me inspira a seguir adelante cada día gracias por siempre estar presente apoyarme en todo, empujarme hacia las mejores decisiones cuando no he elegido el camino correcto, gracias por convertirme en quien soy y ayudarme a realizar mi sueño, esto es para ustedes que son la luz de mi vida. Los quiero.*

*A todos mis tíos por enseñarme lo lindo que es tener una familia unida y que se quiere, gracias por ayudarme a llegar hasta aquí y por cuidar lo que más quiero en el mundo: a mi madre.*

*A mi hermano por ser el pañuelo cuando lloro, por ser el bastón cuando me caigo, por ser mi vida, aunque él no lo sabe.*

*A todos mis compañeros de los cuales he aprendido tanto en estos cinco años de carrera, en especial los que constituyen para mí una familia: Ely, Choy, Dito y Abel, gracias de corazón a todos.*

*A mis tutores por guiarme y ayudarme en todo momento.*

*Yasiel*

## RESUMEN

La gestión del conocimiento marca una diferencia entre países desarrollados y en vías de desarrollo. De ello se infiere la importancia que recae en la ciencia métrica para determinar programas de investigación y educación sostenibles. En las universidades constituye una herramienta clave en la gestión de su política científico-tecnológica. Estos sistemas exhiben un conjunto de carencias que han sido arrastradas en la construcción de herramientas informáticas diseñadas para su soporte. En el contexto de la Universidad de las Ciencias Informáticas este proceso presenta limitaciones asociadas a planificación y control. El primero por razones de disponibilidad de tiempo y/o concurrencia de los responsables en cada una de las áreas; mientras que en el segundo por agrupación de evidencias en un árbol de carpetas con siete niveles de profundidad en la mayoría de los casos; afectando la autenticidad y disponibilidad de la información. El objetivo del estudio persigue desarrollar una solución web de apoyo al sistema de gestión de indicadores de Ciencia, Técnica e Innovación en la Facultad 3, que aumente la seguridad y disminuya el tiempo de planificación y consulta de los indicadores. Se utilizó la metodología de desarrollo Proceso Unificado Ágil; arribando a una liberación por el grupo de calidad de la Subdirección del Centro de Informatización de Entidades avalando la herramienta ingenierilmente; cuyos resultados positivos se complementan mediante evaluación de su aplicación en dos áreas. Cabe destacar que constituye una herramienta de apoyo a la planificación y control de los indicadores que no incorpora elementos para la evaluación de estos.

**Palabras claves:** Ciencia Técnica e Innovación, Indicadores CTI, Sistema CTI, Sistemas de gestión de CTI

## **ÍNDICE DE CONTENIDOS**

INTRODUCCIÓN .....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....	6
1.1 Proceso de gestión de indicadores de CTI en la universidad .....	6
1.2 Problemas comunes asociados a la gestión de indicadores de CTI en la Facultad 3 de la UCI ..	6
1.3 Caracterización de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI ....	8
1.4 Valoración de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI estudiadas.....	12
1.5 Metodología de desarrollo de software .....	13
1.6 Arquitectura de software .....	14
1.7 Lenguajes de modelado y desarrollo .....	15
1.7.1 Lenguaje de modelado .....	15
1.7.2 Lenguajes de programación del lado del cliente .....	16
1.7.3 Lenguajes de programación del lado del servidor.....	16
1.8 Marcos de trabajo .....	17
1.8.1 Bootstrap 3.1.1 .....	17
1.8.2 Symfony 2.3.7 LTS.....	17
1.9 Tecnologías y herramientas de desarrollo .....	18
1.9.1 Tecnología Ajax .....	18
1.9.2 Herramienta CASE Visual Paradigm 8.0 .....	18
1.9.3 Entorno de Desarrollo Integrado NetBeans 8.0 .....	19
1.9.4 Servidor de aplicaciones web Apache 2.2.21 .....	19
1.9.5 Sistema gestor de base de datos PostgreSQL 9.1 .....	19
1.9.6 Navegador web Mozilla Firefox .....	20
1.10 Conclusiones del capítulo .....	20
CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN .....	22
2.1 Propuesta de solución .....	22
2.2 Modelado de negocio .....	22
2.2.1 Modelo conceptual .....	23
2.2.2 Procesos de negocio.....	24
2.3 Requisitos.....	26

2.3.1	Técnicas de captura de requisitos .....	26
2.3.2	Requisitos funcionales .....	27
2.3.3	Descripción de requisitos .....	29
2.3.4	Validación de los requisitos funcionales .....	30
2.3.5	Requisitos no funcionales.....	30
2.4	Análisis y diseño .....	32
2.4.1	Diagrama de clases del diseño.....	33
2.4.2	Diseño de la base datos.....	34
2.4.3	Patrones de diseño utilizados.....	36
2.5	Conclusiones del capítulo .....	37
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN .....		39
3.1	Implementación de la solución.....	39
3.1.1	Estándares de codificación.....	39
3.1.2	Diagrama de componentes .....	40
3.1.3	Diagrama de despliegue.....	42
3.2	Gestión de indicadores de CTI en el SGIF3.....	43
3.3	Verificación de la propuesta de solución.....	46
3.3.1	Resultado de la evaluación de las métricas TOC y RC .....	46
3.3.2	Pruebas de software .....	48
3.4	Resultados del cuasi-experimento .....	55
3.5	Conclusiones del capítulo .....	57
CONCLUSIONES GENERALES .....		58
RECOMENDACIONES .....		59
REFERENCIAS BIBLIOGRÁFICAS .....		60
ANEXOS .....		63

**ÍNDICE DE TABLAS**

**Tabla 1.** Operacionalización de las variables dependientes ..... 7

**Tabla 2.** Evaluación de las variables dependientes ..... 8

**Tabla 3.** Descripción del proceso de negocio Levantar el potencial por departamentos ..... 24

**Tabla 4.** Descripción del requisito funcional Adicionar profesor/especialista ..... 29

**Tabla 5.** Descripción del diseño de clases del requisito funcional Gestionar profesor/especialista ..... 33

**Tabla 6.** Evaluación de la variable Tiempo de consulta..... 55

**Tabla 7.** Evaluación de la variable Seguridad ..... 56

**Tabla 8.** Comparación de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI ..... 63

**Tabla 9.** Diseño de caso de prueba Adicionar capacitación recibida ..... 68

**ÍNDICE DE FIGURAS**

**Figura 1.** Fragmento del Modelo de conceptual ..... 23

**Figura 2.** Diagrama del proceso de negocio Levantar el potencial por departamentos ..... 26

**Figura 3.** Diagrama de clase del requisito funcional Gestionar profesor/especialista ..... 33

**Figura 4.** Fragmento del Modelo de datos del SGIF3..... 35

**Figura 5.** Diagrama de componentes ..... 41

**Figura 6.** Diagrama de despliegue..... 42

**Figura 7.** Interfaz Adicionar propuesta para el área..... 44

**Figura 8.** Interfaz Plan actual de la facultad ..... 44

**Figura 9.** Interfaz Adicionar desglose de indicador ..... 45

**Figura 10.** Interfaz Adicionar premio ..... 45

**Figura 11.** Representación en porciento de los resultados obtenidos en el instrumento (TOC) agrupados en los intervalos definidos ..... 47

**Figura 12.** Representación en porciento de los resultados obtenidos en el instrumento (RC) agrupados en los intervalos definidos ..... 48

**Figura 13.** Funcionalidad indexAction ..... 51

**Figura 14.** Grafo de flujo del camino básico..... 52

**Figura 15.** Acta de aceptación de requisitos ..... 64

**Figura 16.** Diagrama de Modelo de datos del SGIF3 ..... 65

**Figura 17.** Acta de liberación ..... 66

**Figura 18.** Acta de aceptación del cliente..... 67

## **INTRODUCCIÓN**

El análisis, la evaluación de la información y el conocimiento resultante de la actividad científica son elementos imprescindibles para los programas de investigación pública, tecnológica y de desarrollo de una sociedad y es ahí donde la Ciencia de la Información brinda una ayuda inestimable, al desarrollar técnicas e instrumentos para medir la producción de conocimiento y su transformación en bienes. Las disciplinas métricas de la información (bibliometría, cienciometría e informetría) han permitido el desarrollo de indicadores que constituyen herramientas claves en la gestión de la política científica y tecnológica y en los procesos de toma de decisiones estratégicas. En términos generales, los indicadores representan una medición agregada y compleja que permite describir o evaluar un fenómeno, su naturaleza, estado y evolución (1).

Los indicadores de Ciencia, Técnica e Innovación (CTI) emergen como elementos centrales para el crecimiento económico y social de un país. Miden acciones sistemáticas relacionadas con la generación, difusión, transmisión y aplicación de conocimientos científicos, tecnológicos e innovadores, así como el desempeño científico de cada profesional que labore en una entidad de carácter académico-científico (1).

La Universidad de las Ciencias Informáticas (UCI) como centro de nuevo tipo presenta características diferentes a las universidades clásicas del país. Entre las principales se destacan la estrecha interrelación docencia-investigación-producción, con el objetivo de elevar el nivel y la calidad de la propia docencia universitaria y de contribuir directamente a mejorar las condiciones económicas del país. La formación curricular y el núcleo fundamental de las investigaciones científicas que se desarrollan se centran en las ramas de las Ciencias de la Informática y de la Computación. Además del trabajo académico, y de investigación y desarrollo para la exportación de productos informáticos con alto valor agregado, la UCI se propone participar de manera decisiva en la producción de software dirigido a la informatización de la sociedad. Por lo que es de vital importancia lograr medir con eficiencia cómo evoluciona la investigación y la introducción de sus resultados en el propio proceso de fabricación de productos informáticos.

Tanto en la universidad como en la Facultad 3, anualmente se realiza la planificación de los indicadores de CTI que debe cumplir cada profesor o especialista según su categoría docente y científica. A lo largo del año, la información relativa a los indicadores debe ir actualizándose con el objetivo de tener un control sobre estos e ir revisando el cumplimiento de la planificación realizada.

Cada profesor o especialista debe ir acumulando las evidencias del cumplimiento de los indicadores, digitalizarlas y hacerlas llegar a su Jefe de departamento para que monitoree el cumplimiento de los

objetivos del año. A su vez, cada Jefe de departamento debe archivarlas en una estructura de carpetas definidas por la Dirección de investigación y postgrado de la UCI para contabilizarlas y trimestralmente realizar un corte con el Vicedecano de investigación y postgrado (VDIP) para ir monitoreando su cumplimiento. Esta información es utilizada para realizar los balances semestrales y anuales del cumplimiento de los indicadores y tomar acciones correctivas en caso de que exista atraso en el cumplimiento de la planificación.

El proceso de planificación de los indicadores se manifiesta al reunir a los jefes de área para analizar su potencial y negociar los compromisos para el año. En ello es necesario emplear como mínimo seis sesiones de trabajo individual con cada área; para que el hacerlo en una sesión grupal no comprometa la calidad del resultado. Por otro lado el control relativo a las evidencias se realiza actualmente de forma manual, con el apoyo de la herramienta Excel para llevar el control de los indicadores y la contabilización de los mismos, resultando complejo para todos los implicados (Jefe de departamento, VDIP) debido a que es necesario agrupar las evidencias digitales en el árbol de carpetas definidas y realizar el conteo.

Por otra parte en la facultad no existe uniformidad en los modos de recopilación de las evidencias, pues varía según la organización estructural de cada una de las áreas. Entre las distintas variantes que se observan destacan un área por entrega personal de las evidencias en el árbol de carpetas definido por la universidad que luego es consolidado en el árbol del área; y otra donde se utiliza un repositorio con una carpeta personal donde cada individuo coloca sus evidencias para luego ser revisadas y ubicadas en la carpeta correspondiente del área. Ambas variantes, aun cuando suponen un paso hacia la organización de las áreas, afectan el correcto flujo de la información. Asimismo en el resto de las áreas donde intervienen activistas que revisan y organizan la información de profesores y/o especialistas para luego despachar con el vicedecanato.

La forma en que se recogen las evidencias afecta el tiempo de consulta considerablemente, pues hay que esperar a que concluya el paso de la información a través de los distintos niveles que representan las áreas. De manera general, tal cual se manifiesta la recolección de evidencias actualmente en la facultad no garantiza la autenticidad y disponibilidad de la información, comprometiendo la seguridad en dos de sus principales aristas.

De lo anterior surge como **pregunta de investigación** si el contar con una herramienta informática que apoye el proceso de levantamiento del potencial, la planificación, y el control de la información relacionada con CTI incidiría positivamente en los tiempos de consulta de la información, tanto como en su disponibilidad y autenticidad.

Teniendo en cuenta lo antes expuesto, es posible identificar el siguiente **problema de investigación**:  
¿Cómo disminuir los tiempos de consulta de la información relacionada con CTI en la Facultad 3 y aumentar sus niveles de disponibilidad y autenticidad?

La investigación se enmarca en el siguiente **objeto de estudio**: gestión de los indicadores de CTI.

Persiguiendo como **objetivo general**: desarrollar una solución web de apoyo al sistema de gestión de indicadores de CTI definidos por la Facultad 3, para disminuir los tiempos de consulta de la información y aumentar su disponibilidad y autenticidad.

Se identifica como **campo de acción**: las herramientas y técnicas de apoyo al proceso de gestión de indicadores de CTI.

Para dar cumplimiento al objetivo general se proponen los **objetivos específicos** que se relacionan a continuación:

- Realizar el marco teórico de la investigación que permita conocer las principales tendencias en la gestión de indicadores de CTI en centros universitarios y de investigación, así como las herramientas y técnicas más difundidas.
- Desarrollar una solución web para la gestión de los indicadores de CTI en la Facultad 3.
- Validar que la propuesta de solución satisface particularidades del problema planteado a partir de su verificación y el análisis de los resultados de su aplicación en dos áreas de la Facultad 3.

Como **idea a defender** se plantea: el desarrollo de una solución web que apoye el sistema de gestión de indicadores de CTI definido en la Facultad 3 permite disminuir los tiempos de consulta de la información referente a los indicadores de CTI y aumentar sus niveles de disponibilidad y autenticidad.

Los **métodos de investigación** que se utilizaron se mencionan a continuación:

### **Métodos teóricos**

- **Histórico-Lógico**

En la determinación de las necesidades históricas y las tendencias actuales de la planificación y el control de evidencias como subprocesos en el sistema de gestión de indicadores de CTI, así como del uso de las herramientas informáticas que dan soporte a este sistema.

## ➤ **Analítico-Sintético**

Para desentramar y comprender el conocimiento, el estudio de las diferentes fuentes y su interdependencia, constituyendo una herramienta que permitió la determinación de aspectos esenciales, así como el arribo a conclusiones empíricas y teóricas.

## ➤ **Modelación**

Se utiliza para la representación de los procesos que enmarcan el objeto de estudio, del modelo lógico y físico de todos los artefactos correspondientes al ciclo de vida del software, ayudando a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

## **Métodos empíricos**

### ➤ **Observación**

Es el registro del comportamiento de los procesos que fueron investigados. Es uno de los métodos más utilizados en la presente investigación, pues permite obtener información directa e inmediata sobre el fenómeno u objeto que se investiga ya que en los distintos acompañamientos se pudo presenciar la ejecución de los procesos de planificación y control de forma completa.

### ➤ **Entrevista**

Se realizan entrevistas al personal encargado de la gestión de los indicadores de CTI en la Facultad 3 con el fin de recopilar toda la información necesaria sobre la realización de los procesos de recopilación y evaluación de los indicadores de CTI.

### ➤ **Experimental**

Para validar a partir de la aplicación con juegos de datos de dos áreas de la Facultad 3 que en comparación con el proceso ejecutado a las mismas áreas sin el apoyo de la solución que se propone, se disminuyen los tiempos de consulta y aumentan los índices de disponibilidad y autenticidad de la información relacionada con las evidencias.

## **Estructura del trabajo**

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

## **CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA**

Se describe el proceso de gestión de indicadores de CTI en la Facultad 3, haciendo énfasis en los procesos de planificación y control; y se realiza la fundamentación teórica sobre las herramientas y técnicas que lo sustentan, su evolución y situación actual a nivel internacional, nacional y en la UCI. Se especifica la metodología de desarrollo de software y arquitectura a utilizar. Se describen las tecnologías, lenguajes y herramientas que sirven de apoyo para la propuesta de solución.

## **CAPÍTULO 2 ANÁLISIS Y DISEÑO DEL SISTEMA**

Comprende el modelado de negocio a tener en cuenta para el desarrollo de la solución, las técnicas de captura de requisitos, los requisitos funcionales y no funcionales. Además se presenta el análisis y diseño de la propuesta de solución a través de los artefactos que establece la metodología de desarrollo de software utilizada.

## **CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN**

Este capítulo está orientado al desarrollo de la solución; se definen los estándares de implementación, diagramas de componentes y de despliegue. Se describe la notación de las clases y algoritmos implementados que responden a los requisitos funcionales del sistema. Se valida la solución propuesta a través de pruebas de caja blanca y caja negra y se aplican los instrumentos de evaluación de las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC). Los resultados de la revisión por parte del grupo de calidad de la Subdirección del Centro de Informatización de Entidades (CEIGE) como un tercero confiable; y por último se exponen los resultados de un cuasi-experimento que sirvió para evaluar la utilización de la herramienta en dos áreas de la facultad y el comportamiento de las variables dependientes de la investigación.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se presentan los fundamentos teóricos de la investigación en la definición de la propuesta de solución. Se describe el proceso de gestión de indicadores de CTI en la Facultad 3 y se realiza el estudio de herramientas y técnicas de apoyo a la gestión de indicadores de CTI en el contexto internacional, nacional y en la universidad. Se especifica la selección de la metodología de desarrollo de software y la arquitectura a utilizar. Se analizan y describen las tecnologías, lenguajes y herramientas propuestas a emplear durante el desarrollo de la solución.

### 1.1 Proceso de gestión de indicadores de CTI en la universidad

En la actualidad el proceso de gestión de los indicadores de CTI de la universidad se realiza en su mayoría de forma manual, con el apoyo de la herramienta Excel. Anualmente se planifican los indicadores que deben cumplir cada profesor o especialista según su categoría docente y científica. A medida que avanza el año, los indicadores deben ir actualizándose con el objetivo de tener un adecuado control e ir revisando el cumplimiento de la planificación realizada.

En la Facultad 3 cada profesor o especialista debe ir acumulando las evidencias del cumplimiento de los indicadores, digitalizarlas y hacérselas llegar a su Jefe de departamento para que monitoree el cumplimiento de la planificación definida. A su vez cada Jefe de departamento debe archivar las evidencias en una estructura de definidas por la Dirección de investigación y postgrado de la UCI para contabilizarlas; trimestralmente se realiza un corte con el VDIP para ir monitoreando la planificación. La información obtenida es utilizada para realizar los balances semestrales y anuales del cumplimiento de los indicadores y tomar acciones correctivas en caso de que exista atraso en el cumplimiento. Seguidamente se abordan algunos de los problemas más comunes asociados a la gestión de indicadores de CTI, particularmente en la Facultad 3 de la UCI.

### 1.2 Problemas comunes asociados a la gestión de indicadores de CTI en la Facultad 3 de la UCI

El proceso de gestión de los indicadores presenta problemas comunes que están asociados al tiempo de consulta y seguridad de la información, a continuación se describe el estado de las variables involucradas, tomando en cuenta las escalas de evaluación que se exhiben en las Tablas 1 y 2.

- **Tiempo de consulta de la información alto:** como resultado de las observaciones realizadas y de entrevistas a jefes de área, puede declararse que se compromete como consecuencia de no poder consultar las evidencias en tiempo real, demorándose la recopilación de la evidencia más de siete días después de realizada la solicitud. También se ve afectado teniendo en cuenta que la forma de recolectar las evidencias referentes a los indicadores de CTI por cada profesor o especialista es mediante el empleo de correo electrónico o dispositivos de

almacenamiento, no siendo la más rápida; considerando además los niveles de comunicación vertical implícitos por determinación del área.

- **Seguridad de la información:** se conoce como un estado de cualquier sistema (informático o no), que indica que está libre de peligro, daño o riesgo. Se entiende como peligro o daño, todo lo que pueda afectar su funcionamiento directo o los resultados que se obtienen. El sistema de gestión de indicadores de CTI en la Facultad 3 se ve afectado directamente en cuanto a la seguridad de la información, comprometiendo los siguientes pilares (2):
  - **Disponibilidad baja:** la información referente a los indicadores de CTI, no se encuentra disponible en todo momento, pues depende de un flujo de transmisión de la información lento. En ocasiones existe pérdida de la información como consecuencia de la manera en que se transmite.
  - **Autenticidad media:** se ve afectada debido que es complejo comprobar la autoría de cada una de las evidencias de los indicadores de CTI, provocando que la información carezca de veracidad en algunos casos.

A continuación se presentan la operacionalización y evaluación de las variables (3):

**Tabla 1.** Operacionalización de las variables dependientes

Variables dependientes	Operacionalización		
	Dimensiones	Parámetros	Indicadores
Tiempo de consulta de la información	Tiempo	Alto	El tiempo que transcurre entre la solicitud de: potencial/negociación de los indicadores/evidencias de cumplimiento; y la respuesta correspondiente es mayor que siete días.
		Medio	El tiempo que transcurre entre la solicitud de: potencial/negociación de los indicadores/evidencias de cumplimiento; y la respuesta correspondiente ocurre después del tercer día y hasta el séptimo.
		Bajo	El tiempo que transcurre entre la solicitud de: potencial/negociación de los indicadores/evidencias de cumplimiento; y la respuesta correspondiente ocurre en un plazo

			de hasta tres días.
<b>Seguridad de la información</b>	<b>Disponibilidad</b>	Baja	La información no está disponible en todo momento y existe pérdida de ella.
		Media	La información no está disponible en todo momento y no existe pérdida de ella. La información está disponible en todo momento y existe pérdida de ella.
		Alta	La información está disponible en todo momento y no existe pérdida de ella.
	<b>Autenticidad</b>	Baja	Menos del 50% de las evidencias, de algún modo, son incoherentes con su declaración.
		Media	Entre el 50% y el 99% de las evidencias, de algún modo, son incoherentes con su declaración.
		Alta	El 100% de las evidencias se muestran en completa coherencia con su declaración.

**Tabla 2.** Evaluación de las variables dependientes

<b>Variables</b>	<b>Evaluación</b>		
	<b>Bajo</b>	<b>Medio</b>	<b>Alto</b>
<b>Tiempo de consulta de la información</b>	Tiempo $\leq$ 3 días	Tiempo > 3 día y Tiempo $\leq$ 7 días	Tiempo > 7 días
<b>Seguridad de la información</b>	Disponibilidad baja y autenticidad baja.	Disponibilidad media y autenticidad media.	Disponibilidad alta y autenticidad alta.
	Disponibilidad media y autenticidad baja.	Disponibilidad alta y autenticidad baja.	Disponibilidad media y autenticidad alta.
	Disponibilidad baja y autenticidad media.	Disponibilidad baja y autenticidad alta.	Disponibilidad alta y autenticidad media.

### 1.3 Caracterización de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI

En el mundo existen diferentes herramientas y técnicas de apoyo a la gestión de indicadores de CTI que permiten medir los resultados científicos alcanzados por las instituciones a través de un conjunto

de indicadores, que varían en dependencia de las necesidades específicas de la institución. El estado del arte realizado se enfocó a sistemas informáticos referentes en sus áreas. Dentro de los sistemas analizados se encuentran: la Red de Indicadores de Ciencia y Tecnología Iberoamericana e Interamericana (RICYT), el Sistema de Información de Ciencia y Tecnología Argentino (SICYTAR), el Sistema Institucional para la Gestión de la Ciencia y la Técnica de la Universidad de Pinar del Río (CuScienTI), el Sistema de Información para la Gestión de Programas y Proyectos de Ciencia Tecnología e Innovación (SiPROCiT), y el Sistema de Indicadores de Ciencia, Tecnología e Innovación de la UCI (SIndiCIT).

Para el análisis de estos sistemas se establecieron los siguientes parámetros de evaluación: tiempo de consulta de la información, seguridad de la información enfocado a los pilares disponibilidad y autenticidad, escalabilidad, generación de reportes, generación de reportes dinámicos, navegación, diseño y multiplataforma.

## ➤ En el ámbito internacional

### **Red de Indicadores de Ciencia y Tecnología Iberoamericana e Interamericana**

El objetivo general de la RICYT es promover el desarrollo y el uso de instrumentos para la medición y el análisis de la ciencia y la tecnología en un marco de cooperación internacional. Manteniendo el propósito de profundizar en el conocimiento de los indicadores de CTI y su utilización como instrumento político para la toma de decisiones, no está concebido para la planificación de los indicadores de CTI y manipula parcialmente la medición y evaluación de los indicadores de CTI (4).

RICYT es un portal web informativo, que proporciona información confiable de indicadores de CTI, apoyándose en talleres que son útiles para el intercambio de experiencias y la discusión teórica sobre su construcción. Los manuales y publicaciones de la RICYT recogen el trabajo realizado por las subredes temáticas y brindan lineamientos para la elaboración de estadísticas referentes a los indicadores. El portal se mantiene actualizado con las novedades y noticias del campo de CTI y los estandariza a partir del uso de políticas internacionales (4).

En el portal web se aprecia orden y estructuración de la información facilitando la navegación dentro del sitio y permitiendo acceder a contenidos de interés con un mínimo de clics. Brinda un diseño atractivo del sitio y adecuado al perfil del usuario. Se puede ejecutar directamente en cualquier plataforma. Las informaciones son actualizadas y están disponibles en todo momento, brindando un bajo tiempo de consulta de la información. Constituye una herramienta eficaz en la generación de reportes estadísticos del área. Se encuentra asesorado por un conjunto de profesores y especialistas de gran conocimiento en el tema, aportándole un alto nivel de autenticidad a la información que brinda el portal.

## **Sistema de Información de Ciencia y Tecnología Argentino**

El SICYTAR está dirigido específicamente a la conformación de tres importantes bases de datos, donde se podrán consultar: los antecedentes curriculares del personal científico y tecnológico argentino, los grupos y proyectos de I+D existentes en Argentina y las instituciones que realizan actividades científicas y tecnológicas en el país.

Tiene entre sus funciones mantener un registro unificado con actualización permanente y disponible en todo momento, lo que mejora considerablemente en los tiempos de consulta de la información. Produce información estadística detallada para elaborar indicadores y evaluar políticas de ciencia y tecnología. Ofrece una herramienta útil para el establecimiento de contactos entre investigadores, así como con los sectores gubernamentales y empresariales. Ofrece a la sociedad argentina en su conjunto el acceso a un registro de datos de carácter público, información agregada de calidad y de alta autenticidad sobre el sector de Ciencia y Tecnología nacional producida a partir de ella. Estandariza formatos para la tramitación de las principales gestiones administrativas en las instituciones del sector, como convocatorias y procesos de evaluación de profesionales, proyectos e instituciones. Posee un adecuado diseño y puede operar en cualquier plataforma informática (5).

### ➤ **En el ámbito nacional**

## **Sistema Institucional para la Gestión de la Ciencia y la Técnica**

El CuScienTI se encuentra actualmente vigente en la Universidad Hermanos Saíz Montes de Oca, ubicada en la provincia Pinar del Río. La aplicación está orientada a la gestión y evaluación de la Ciencia, la Técnica, la Innovación y el Postgrado de dicho centro, apoyada en la utilización de indicadores cuantitativos para la medición de la actividad científica de profesores, investigadores y estudiantes de postgrado (6).

CuScienTI permite la actualización sistemática del Curriculum Vitae<sup>1</sup> de cada profesor. Es una plataforma informática escalable, diseñada para utilizarse como fuente de información, independientemente de tipos y versiones de hardware o software. Permite la gestión integrada de los diferentes procesos desde la base (profesor). Representa, visualiza y genera reportes de la información en diversos formatos, incluyendo mapas de conocimiento. A pesar de las prestaciones que brinda el sistema a la universidad, no realiza una gestión documental de la información, como respaldo a los datos que se introducen en el sistema incidiendo en gran medida en la disminución de la veracidad de la información que se gestiona (6).

---

<sup>1</sup> Se refiere al conjunto de experiencias (educacionales, laborales y vivenciales) de una persona.

### **Sistema de Información para la Gestión de Programas y Proyectos de Ciencia, Tecnología e Innovación**

El SiPROCiT constituye un sistema de apoyo a la planificación, control, evaluación y proyección de los programas y proyectos de ciencia e innovación, acorde a las prioridades establecidas para un período determinado en Cuba y que contribuye a incrementar el nivel de integración del Sistema de Proyectos y Programas (SPP). De manera general con la aplicación se puede acceder a los listados de programas y proyectos nacionales, territoriales y ramales, así como a los datos consolidados, series cronológicas y gráficos estadísticos de programas y proyectos en ejecución y terminados.

La información se introduce de manera descentralizada por cada uno de los secretarios de los programas y es accedido por otros usuarios para recuperar información en forma de reportes dinámicos. El sistema diseñado con interfaces atractivas, brinda organización y orientan al usuario en la navegación, fue desarrollado sobre una arquitectura Cliente Servidor y soportado por una plataforma mixta con prevalencia de tecnologías de código abierto. A pesar que la información no se encuentra totalmente actualizada, el sitio se mantiene todo el tiempo en línea, mejorando en los tiempos de consulta de la información por los usuarios (7).

#### ➤ **En la Universidad de las Ciencias Informáticas**

### **Sistema de Indicadores de Ciencia, Tecnología e Innovación**

El SIIndiCIT que se encuentra actualmente vigente en la UCI, brinda la posibilidad de llevar un control de los indicadores relacionados con las actividades de CTI que se realizan en el centro universitario. Para su procesamiento el sistema depende de una serie de entradas referente a los indicadores cuantitativos que son introducidos por las diferentes estructuras del sistema de CTI en la universidad. A partir de estas entradas y a través de herramientas estadísticas, el sistema permite un conjunto de salidas que posibilitan la evaluación de la producción científica y el análisis del estado actual del centro (8).

La aplicación web está proyectada a permanecer accesible en todo momento e independiente de tipos y versiones de hardware o software, aportándole un alto nivel de disponibilidad a la información y un bajo tiempo de consulta. Es de gran utilidad para el cálculo de los indicadores cuantitativos y se adapta a las características específicas de la universidad. Permite la generación de reportes anuales de todas las áreas de la universidad (8). Sin embargo tiene la limitación de que existe solo un usuario por área, al caso facultad, que debe introducir la información correspondiente a esta. No compendia desde un dato primario (nivel profesor) para arrojar los reportes por área (facultad); luego esto constituye un elemento fundamental a modelar en la propuesta de solución.

## 1.4 Valoración de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI estudiadas

Para la comparación de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI, se analizaron el comportamiento de los sistemas informáticos antes mencionados en cuanto a parámetros de evaluación como: tiempo de consulta de la información, seguridad de la información enfocado a los pilares disponibilidad y autenticidad, escalabilidad, generación de reportes, generación de reportes dinámicos, navegación, diseño y multiplataforma (ver Anexo 1).

Como resultado del análisis comparativo se concluye que los sistemas de apoyo a la gestión de indicadores de CTI coinciden al 100% en que se pueden ejecutar directamente en cualquier plataforma y generan reportes con opciones de descarga para obtener información, sin embargo solo el diseño del 20% se enfoca a la generación de reportes dinámicos. El 60% cuenta con una alta disponibilidad de la información, lo que se traduce en un 80% de los objetos estudiados con tiempo de consulta de la información bajo.

Las herramientas y técnicas estudiadas son escalables en un 60%, propiedad deseable en toda aplicación, que permite manejar el crecimiento continuo de trabajo, reaccionar y adaptarse sin perder calidad. En cuanto a la autenticidad de la información, el 80 % mantiene un nivel alto de autenticidad de la información debido a la administración y asesoramiento de las aplicaciones por personas de marcada referencia en el tema.

La navegación en los sistemas se comporta al 60% de buena aceptación, permitiéndole al usuario la navegación con número limitado de clics y la orientación en todas las vistas de los sistemas a partir del empleo de índices temáticos y jerarquía de contenidos. En cuanto al diseño, se estableció que el 20% de las aplicaciones tienen un aspecto atractivo y acorde al perfil de la institución.

Después de haber realizado un análisis de los sistemas anteriores, se puede constatar que las aplicaciones no cumplen en su totalidad con las propiedades deseables para el desarrollo de la propuesta de solución. De manera general no se adaptan a las necesidades específicas de la situación, sin embargo aportan a la investigación elementos ideales a implementarse en la propuesta de solución.

La mayoría de los sistemas de información relacionados con la actividad de CTI a nivel internacional, constituyen portales web informativos que no están concebidos para la gestión de información. Estas aplicaciones se han desarrollado para difundir y comparar los avances en el área de CTI por países. En el contexto internacional las herramientas informáticas para la gestión de indicadores de CTI no se encuentran disponibles de forma libre para ser adaptadas y utilizadas por otros centros. Existen otros sistemas de gestión que solo manipulan parte de la actividad de CTI, centrándose fundamentalmente

en la gestión de currículum vitae y proyectos de investigación. En el marco de la UCI las soluciones desarrolladas, no tienen alcance local e informatizan de forma aislada las actividades que se realizan a nivel de facultad.

A partir de los resultados obtenidos de la investigación, se propone el desarrollo de un sistema informático, que gestione los indicadores de CTI en la Facultad 3 (SGIF3), mejore el tiempo de consulta de la información y garantice la seguridad a partir de una alta disponibilidad y autenticidad de la información.

Para garantizar que se cumplan los estándares de un buen diseño del sistema y una adecuada navegación, que el sistema permita la generación de reportes y sea escalable ante futuros cambios, es necesario la adecuada utilización de una metodología de desarrollo de software.

## **1.5 Metodología de desarrollo de software**

El desarrollo de toda solución informática es difícil de guiar sin el uso de una metodología que contribuya a agilizar el proceso de desarrollo de software y la entrega de un producto con calidad (9). El autor determina la utilización de la metodología de desarrollo Proceso Unificado Ágil (AUP) en su variante ajustada al entorno UCI por su departamento de calidad por las características de la investigación, inmersión con el proceso y tiempo estimado de terminación. A continuación se presentan las características generales.

### **Proceso Unificado Ágil**

Actualmente la UCI se encuentra inmersa en un proceso de mejora de los procesos que se desarrollan como parte de la construcción del software, basado en el Modelo de Madurez y Capacidad Integrado (CMMI). CMMI mejora y evalúa los procesos de desarrollo y mantenimiento de sistemas y productos de software, es uno de los modelos de certificación internacional más prestigiosos en la ingeniería software. Se decide utilizar la variación de la metodología Proceso Unificado Ágil en unión con el modelo CMMI-DEV v 1.3 para el nivel 2 como metodología rectora en el desarrollo de la solución propuesta (10) (11).

AUP se adapta al ciclo de vida definido para la actividad productiva de la UCI. Brinda a la solución un enfoque simple y fácil para el desarrollo utilizando técnicas y conceptos ágiles. Continuamente vela por la excelencia técnica y el buen diseño y acepta cambios de requisitos en las diferentes etapas del desarrollo, incluso en las tardías. Además cumple con varios principios como son: simplicidad, debido a que todo se describe de forma concisa y en pocas páginas y el principio de agilidad pues se ajusta a los valores y principios de la alianza ágil (11).

Se compone por tres fases: Inicio, Ejecución y Cierre, que brindan los artefactos y la documentación necesaria para el proceso de desarrollo de software. Dado que esta metodología es iterativa-incremental, se genera al final de cada iteración una versión del producto, ayudando a monitorear el progreso del proyecto e ir rectificándolo en caso de ser necesario (11).

Definida la metodología a utilizar, se procede a determinar la arquitectura de software y patrón arquitectónico que representará el flujo e interacción de la información del sistema.

## 1.6 Arquitectura de software

Una arquitectura de software podría definirse de forma general como el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de definir los módulos principales y las responsabilidades que tendrá cada uno de los mismos. Define la interacción que existirá entre dichos módulos, el control y flujo de datos, la secuencia de la información, los protocolos de comunicación y la ubicación en el hardware (12).

La arquitectura definida para el desarrollo de la solución está determinada por la arquitectura Cliente Servidor y el patrón arquitectónico Modelo Vista Controlador (MVC), que a continuación se describen.

### Arquitectura Cliente Servidor

La arquitectura Cliente Servidor centraliza los accesos, recursos y la integridad de los datos al ser controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos. Aporta escalabilidad al sistema, pudiéndose aumentar la capacidad de clientes y servidores por separado. Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes es posible reemplazar, reparar, actualizar o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio. En consecuencia de que las transacciones se dividen en procesos independientes, el servidor no necesita una elevada potencia de procesamiento y se reduce el tráfico de red considerablemente (12).

### Patrón arquitectónico Modelo Vista Controlador

El patrón arquitectónico MVC brinda la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos: modelo, vista y controlador. La manera en que se divide es fundamental para el desarrollo de arquitecturas consistentes y un considerable ahorro de tiempo en la implementación.

El componente modelo representa la información persistente del mundo real. Encapsula los datos, las funcionalidades y preserva la integridad de los datos. El componente vista muestra la información al usuario. Generalmente consiste de toda la información obtenida del modelo que se le muestra al

usuario de la aplicación con elementos de diseño que la hacen amigable e interactiva. Finalmente, el componente controlador responde a eventos e interpreta las operaciones del usuario (13).

Una vez realizado el estudio de la arquitectura de software se procede a la selección de los lenguajes de modelado y desarrollo para facilitar el entendimiento del negocio y su posterior implementación.

## **1.7 Lenguajes de modelado y desarrollo**

### **1.7.1 Lenguaje de modelado**

Se denomina lenguaje de modelado de objetos al conjunto estandarizado de símbolos y las distintas combinaciones de la disposición para modelar un diseño de software (14).

#### **UML 2.0**

Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Mediante él es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. Ofrece un estándar para describir aspectos conceptuales, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Un modelo UML puede ser usado en todas las fases de desarrollo de software, lo que facilita su entendimiento y la comunicación (15).

#### **BPMN 2.0**

Modelo y Notación de Procesos de Negocio (BPMN) es un estándar internacional de modelado de procesos aceptado por la comunidad. Independiente a cualquier metodología de modelado de procesos, crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos. Permite modelar los procesos de una manera unificada y estandarizada facilitando un entendimiento a todas las personas de una organización. Está diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades; contribuye al diseño, control y gestión de los procesos del negocio (16).

Se utilizó el lenguaje de modelado UML en la modelación de los artefactos que se sucedieron a través de las distintas fases de desarrollo definidas por la metodología utilizada. Sin embargo se empleó la notación para el modelado de procesos BPMN para la modelación de los procesos que enmarcan la investigación en aras de lograr un mayor entendimiento de estos a partir de las bondades de la mencionada notación. A continuación se exponen los lenguajes de programación utilizados.

## 1.7.2 Lenguajes de programación del lado del cliente

Las tecnologías del lado del cliente pueden ser directamente "digeridas" por el navegador y no necesitan un tratamiento previo. Son totalmente independientes del servidor (17). Como lenguajes de programación se utilizó JavaScript y HTML5.

### JavaScript 1.6

JavaScript es un lenguaje del lado del cliente basado en objetos similares a una clase Java donde se puede instanciar un objeto, crear propiedades y métodos. Se utiliza principalmente integrado en un navegador web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. La gran mayoría de los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web (18). Al no necesitar compilación y ser interpretado directamente por el navegador ayuda a disminuir los tiempos de ejecución y el consumo de recursos del procesador; también es válido destacar que no compromete la carga de los servicios a través de la red.

### HTML 5

El Lenguaje de Marcado de Hipertexto (HTML) permite escribir texto de forma estructurada, está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Una de las características esenciales del lenguaje es la universalidad, cualquier ordenador, independientemente del sistema operativo, puede leer o interpretar una página web. Se decide utilizar porque es un código que posibilita la creación y edición de documentos web, basado en etiquetas, que tiene como virtud entre otras, la de poder ser implementado por código de otros lenguajes como JavaScript que amplían y mejoran su capacidad original (19).

## 1.7.3 Lenguajes de programación del lado del servidor

Las tecnologías de programación del lado del servidor participan en un proceso o transacción desde el servidor, de forma que son transparentes al usuario final de las aplicaciones que manejan arquitectura Cliente Servidor (17). En la investigación, como se explica a continuación se utiliza el lenguaje de programación PHP 5.5.4 en el lado del servidor.

### PHP 5.5.4

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de las características más potentes y destacables de PHP es su soporte para acceso a un gran número de bases de datos. Además soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, HTTP y derivados (20).

Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible al navegador web y al cliente, permitiendo la programación más segura y confiable. Capacidad de conexión con la mayoría de los motores de

base de datos que se utilizan en la actualidad destacándose su conectividad con MySQL y PostgreSQL. Posee una amplia documentación en su sitio web oficial, destacándose la descripción de todas las funcionalidades del lenguaje y ejemplificadas en un único archivo de ayuda (21).

PHP constituye uno de los lenguajes de programación web más difundidos y utilizados actualmente, debido a las potencialidades descritas, lo que determina que es una tecnología adecuada para el desarrollo de la solución propuesta.

A continuación se presentan los marcos de trabajo seleccionados para el desarrollo de la solución.

## 1.8 Marcos de trabajo

### 1.8.1 Bootstrap 3.1.1

Bootstrap es un marco de trabajo que simplifica el proceso de creación de diseños web combinando Hojas de Estilo en Cascada (CSS) y JavaScript. La mayor ventaja es que permite crear interfaces que se adapten a los distintos navegadores utilizando un marco de trabajo potente con numerosos componentes web que minimizan esfuerzo y tiempo.

Entre las características principales que hacen de Bootstrap una adecuada herramienta para el desarrollo de la propuesta de solución, se encuentran las siguientes (22):

- Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el marco de trabajo de forma sencilla y potente en proyectos con tecnología web.
- Ofrece un diseño sólido usando Lenguaje de Hojas de Estilo (LESS) y estándares como CSS3/HTML5.
- Es un marco de trabajo ligero y funciona con todos los navegadores web.

### 1.8.2 Symfony 2.3.7 LTS

Symfony es un marco de trabajo diseñado para mejorar el desarrollo de las aplicaciones web basado en el patrón MVC. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Simplifica el desarrollo de aplicaciones mediante la automatización de varios patrones empleados para un propósito dado. Además se integra al código, lo que provoca que el desarrollador escriba mejor, más legible y un código más estable. Condiciona que se programe más fácil, empaqueta operaciones complejas en declaraciones simples. Contiene numerosas herramientas y clases para acortar el tiempo de desarrollo de una aplicación web compleja. Automatiza tareas comunes de manera que el desarrollador puede concentrarse por completo en los detalles específicos de una aplicación (23).

## ➤ Modelo

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

## ➤ Vista

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

## ➤ Controlador

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

Symfony viene integrado con el Mapeador de Objeto Relacional (ORM) Doctrine en la versión 2.0 para PHP que se sitúa arriba de la poderosa PHP DBAL (Capa de Abstracción de la Base de Datos) y facilita la persistencia y lectura de información hacia y desde la base de datos. Una de sus características claves es la facultad de escribir consultas a la base de datos en un lenguaje orientado a objetos. Doctrine brinda a los desarrolladores una poderosa alternativa al SQL manteniendo una máxima flexibilidad sin necesidad de duplicar código (24).

En el epígrafe siguiente se describen las tecnologías y herramientas seleccionadas para el desarrollo de la solución en correspondencia con el marco de trabajo a utilizar.

## 1.9 Tecnologías y herramientas de desarrollo

### 1.9.1 Tecnología Ajax

Acrónimo de Asynchronous JavaScript And XML (Ajax), es una tecnología de desarrollo web para crear aplicaciones interactivas que se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Posibilita realizar cambios sobre las páginas sin necesidad de recargarlas, aumentando la interactividad y usabilidad en las aplicaciones (25).

### 1.9.2 Herramienta CASE Visual Paradigm 8.0

Visual Paradigm es una herramienta de modelado basada en la notación UML profesional que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la

documentación. Se caracteriza por el diseño centrado en casos de uso, enfocado al negocio y a la programación orientada a objetos, permitiendo generar un software de mayor calidad. Usa un lenguaje estándar común a todo el equipo de trabajo facilitando la comunicación con modelo y código sincronizados en el ciclo de desarrollo (26).

### 1.9.3 Entorno de Desarrollo Integrado NetBeans 8.0

NetBeans es un Entorno de Desarrollo Integrado (IDE), una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para otro lenguaje de programación. NetBeans es un software de código abierto para profesionales que quieran crear aplicaciones para empresa, web, de escritorio. La versión 8.0 proporciona un conjunto de mejoras en la compatibilidad con PHP e incorpora nuevas herramientas para trabajar con HTML 5.

Se decide utilizar NetBeans 8.0 pues ofrece opciones de desarrollo con *plugins* y herramientas incorporadas, así como completamiento de código, facilitándoles el trabajo de programación para la creación de aplicaciones con rapidez. El editor de código fuente que presenta es ágil y robusto, características que hacen que sea una excelente herramienta para el desarrollo de soluciones informáticas (27).

### 1.9.4 Servidor de aplicaciones web Apache 2.2.21

Apache es un servidor web gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Es multiplataforma, aunque idealmente está preparado para funcionar en entornos Unix. Posee una configuración básica para usuarios estándares aunque permite configuraciones avanzadas en función del nivel de servicios que debe ofrecer el servidor de aplicaciones. Tiene integración con múltiples tecnologías como PHP, aspecto muy provechoso para el trabajo en cuestión debido a la necesaria integración con el lenguaje.

Apache es altamente configurable y permite bases de datos de autenticación y negociado de contenido. Entre sus más interesantes ventajas se pueden mencionar su carácter modular, permitiendo incrementar o quitar los módulos que posee en función de las necesidades específicas del usuario (28).

### 1.9.5 Sistema gestor de base de datos PostgreSQL 9.1

PostgreSQL es el Sistema gestor de base de datos Objeto Relacional, de código abierto más potente del mercado, multiplataforma y que permite el acceso desde los lenguajes de programación más populares, incluyendo PHP. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Funciona con grandes cantidades de datos y una alta

conurrencia de usuarios accediendo a la vez al sistema. Garantiza las propiedades de atomicidad, consistencia, aislamiento y durabilidad de las transacciones. Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit (29) (30).

## 1.9.6 Navegador web Mozilla Firefox 30.0

Mozilla Firefox es el nuevo e innovador navegador de código abierto. Se trata de un práctico y ágil navegador, que está en constante renovación. Presenta la capacidad de modificarse totalmente a gusto del usuario y según las necesidades. Soporta las tecnologías y lenguajes definidos anteriormente. Además posee diversidad de complementos, estilos y opciones que hacen útil y cómoda la navegación (31).

## 1.10 Conclusiones del capítulo

Una vez finalizado el presente capítulo quedaron evidenciadas las bases teóricas que sustentarán el proceso de desarrollo de la solución del problema planteado, para ello se arribó a las siguientes conclusiones:

- Entre los problemas más comunes asociados a la gestión de indicadores de CTI, particularmente en su planificación y control, específicamente en la Facultad 3 de la UCI, se encuentran los relacionados con los tiempos de consulta de la información. Manifiesto en el tiempo que transcurre entre la solicitud y su respuesta, evaluada en 3 rangos: bajo menos de 3 días, medio entre 3 y 7 días, y alto con períodos superiores. También los relacionados con la seguridad. Manifiestos en la disponibilidad y la autenticidad de la información; evaluada por la combinación de los valores bajo, medio, alto entre ambas variables.
- Las herramientas y técnicas que constituyen el estudio del estado del arte exhiben características que conviene implementar en la propuesta de solución, como lo son: constituir solución web, permitir constante retroalimentación de información relacionada con actividades relacionadas al objeto principal sin ser este, constituir sistemas ligeros, disponibilidad de la información y elementos de diseño etc. Asimismo ostentan limitaciones que se deben evitar: necesidad de un único usuario para la entrada de los datos primarios de todo un área, no tienen responsabilidades por compilación de datos sino colección y reportes.
- Para guiar el desarrollo de la solución se utilizó la metodología ágil AUP en su versión ajustada a la UCI. Definiendo una arquitectura Cliente Servidor aplicando el patrón arquitectónico MVC. Con lenguajes de modelado BPMN en la modelación de los procesos y UML en la confección de los artefactos de las distintas fases del ciclo de vida, ambos en su versión 2.0.

- Los lenguajes de programación por la parte cliente fueron JavaScript 1.6 y HTML 5, mientras que por el lado del servidor se utilizó PHP 5.5.4. Utilizando los marcos de trabajo Bootstrap 3.1.1 y Symfony 2.3.7 LTS con ORM Doctrine 2.0.
- Entre las tecnologías de desarrollo utilizadas destacan: Ajax para el desarrollo web, Visual Paradigm 8.0 como herramienta CASE, NetBeans 8.0 como IDE, Apache 2.2.21 como servidor de aplicaciones web, PostgreSQL 9.1 como Sistema gestor de base de datos, y Mozilla Firefox como navegador web.

### **CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN**

En el presente capítulo se define la propuesta de solución del sistema a partir del análisis de las herramientas y técnicas estudiadas para la gestión de indicadores de CTI particularmente la planificación y control. En correspondencia a la utilización de la metodología de desarrollo AUP se describen las disciplinas de Modelado de negocio, Requisitos y Análisis y diseño de la fase Ejecución.

#### **2.1 Propuesta de solución**

Para darle solución a la problemática, en consecuencia con los elementos abordados en el capítulo anterior, se propone desarrollar una aplicación web que gestione la información referente a los indicadores de CTI en la Facultad 3, con la que se disminuyan tiempos actuales de consulta de la información y que aumente la disponibilidad y autenticidad de la información.

El control de acceso en la aplicación estará determinado por la asignación de roles, teniendo en cuenta las acciones que pueden realizar los usuarios en el sistema según sus funciones. Será accedida desde las diferentes áreas de investigación de la facultad donde se introducirá la información de cada usuario para la determinación del potencial con que se planificará el año y las evidencias de la actividad científica una vez terminada el proceso de negociación del compromiso con las áreas. Además permite al Vicedecanato de Investigación y Postgrado de la facultad obtener la información actualizada sobre del cumplimiento de los indicadores de CTI, validar la información, así como visualizar el conjunto de reportes de indicadores que se obtienen de los procesos que se manejan en el sistema de gestión. No contempla la evaluación de los objetivos que conforman el área de resultado clave dado que este es objeto de otra investigación relacionada con la que la presente comparte un eje común y se complementa.

Concluida la descripción de la propuesta de solución se establece la necesidad de llevar a cabo un estudio de la estructura y los procesos del área de investigación y postgrado de la Facultad 3 que serán informatizados.

#### **2.2 Modelado de negocio**

En el ciclo de vida del proceso de ejecución del software el Modelado del negocio es la primera disciplina que propone la metodología AUP. Tiene su mayor peso durante la fase de Ejecución, permitiendo conocer los procesos actuales existentes de la entidad a la cual se le va a desarrollar el sistema y define una visión estática y dinámica del funcionamiento de la organización. El Modelado del negocio se realizó a partir de la confección de documentos como: Modelo conceptual, Descripción de procesos de negocio, Reglas de negocio, Mapa de procesos y Glosario de términos, que pueden ser consultados en el Expediente de proyecto del SGIF3.

### 2.2.1 Modelo conceptual

Los modelos conceptuales se construyen a partir de las definiciones fundamentales que se van a tratar a lo largo de la investigación. Se representa cómo se relacionan los conceptos entre sí para hacer más fácil su entendimiento y contribuir a la solución del problema. Describe el entendimiento común alcanzado por los involucrados respecto a los objetos y/o conceptos del dominio, su relación y el diccionario de datos, que puede ser consultado en el documento Modelo conceptual contenido en el Expediente de proyecto del SGIF3 (32). A continuación se muestra un fragmento del modelo conceptual del negocio en cuestión donde están representados los conceptos asociados a la información personal de cada profesor/especialista.

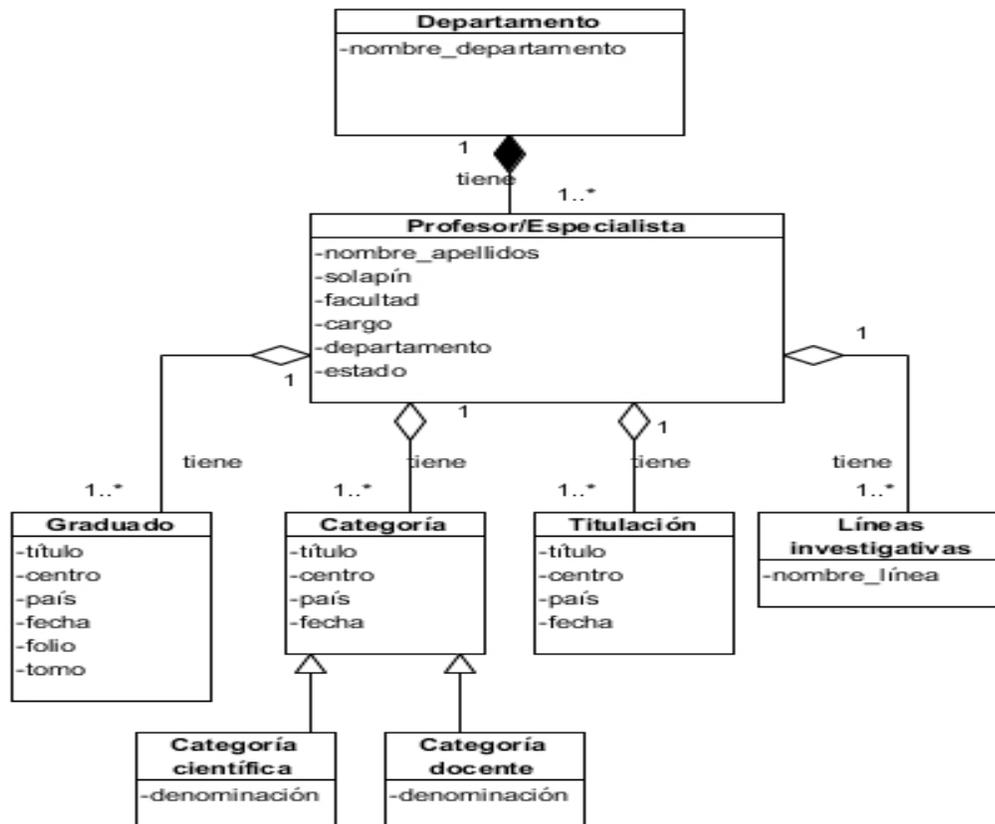


Figura 1. Fragmento del Modelo de conceptual

**Profesor/especialista:** representa el autor de las evidencias de indicadores.

**Departamento:** representa el área a la que pertenecen varios profesores/especialistas.

**Graduado:** representa formación profesional del profesor/especialista.

**Categoría:** representa la categoría docente o científica del profesor/especialista.

**Titulación:** titulación académica y/o grado científico del profesor/especialista.

**Líneas investigativas:** áreas o temas de investigación que potencia el profesor/especialista.

### 2.2.2 Procesos de negocio

El proceso de negocio es un grupo de tareas relacionadas de manera lógica que se llevan a cabo en determinada secuencia y producen o manipulan una colección de datos empleando recursos de la organización para dar resultados que apoyan sus objetivos (33). En la realización de la presente investigación el autor se reunió con el VDIP, con un grupo de profesores y Jefes de departamento de la Facultad 3 para identificar cómo llevan a cabo los procesos de planificación y recopilación de las evidencias de los indicadores de CTI en sus respectivas áreas.

A continuación se presenta la descripción y el diagrama del proceso de negocio: Levantar el potencial por departamentos en la Tabla 3. Los restantes diagramas de procesos de negocio se encuentran en el artefacto Descripción de procesos de negocio contenido en el Expediente de proyecto del SGIF3. A su vez la Figura 2 muestra una representación visual del mismo proceso.

**Tabla 3.** Descripción del proceso de negocio Levantar el potencial por departamentos

<b>Objetivo</b>	Obtener la información de los profesores/especialistas por departamentos.
<b>Evento(s) que lo genera(n)</b>	Solicitud por parte del VDIP el levantamiento del potencial por departamentos.
<b>Pre condiciones</b>	Existe un documento de Información del departamento (Excel).
<b>Marco legal</b>	
<b>Reglas de negocio</b>	Solicitud de la información. Información. Complejidad de la información. Nombre del departamento. Fecha. Estado de un profesor/especialista. Planificación. Indicadores.
<b>Responsable</b>	Jefe de departamento (JD)
<b>Clientes internos</b>	VDIP
<b>Clientes externos</b>	Dirección de investigación y postgrado de la universidad.
<b>Entradas</b>	
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1.	Solicitar información profesor/especialista: el JD solicita a los profesores/especialistas la información personal y las afectaciones para el próximo año para entregar en menos de tres días.
2.	Enviar información profesor/especialista: el profesor/especialista envía al JD la información personal y las afectaciones para el próximo año.
3.	Revisar información profesor/especialista: el JD revisa la completitud de la información.

## Capítulo 2. Análisis y diseño de la solución

4. Registrar información profesor/especialista: registra la información del profesor/especialista.
5. Actualizar información del departamento: actualiza la información general del departamento con los datos del profesor/especialista.

### Pos-condiciones

1. Información del departamento (actualizado).

### Salidas

1. Información del departamento (Excel).

### Flujos paralelos

<Nº Actividad anterior al flujo paralelo +1>. a <Nombre del flujo paralelo>

### Pos-condiciones

### Salidas

### Flujos alternos

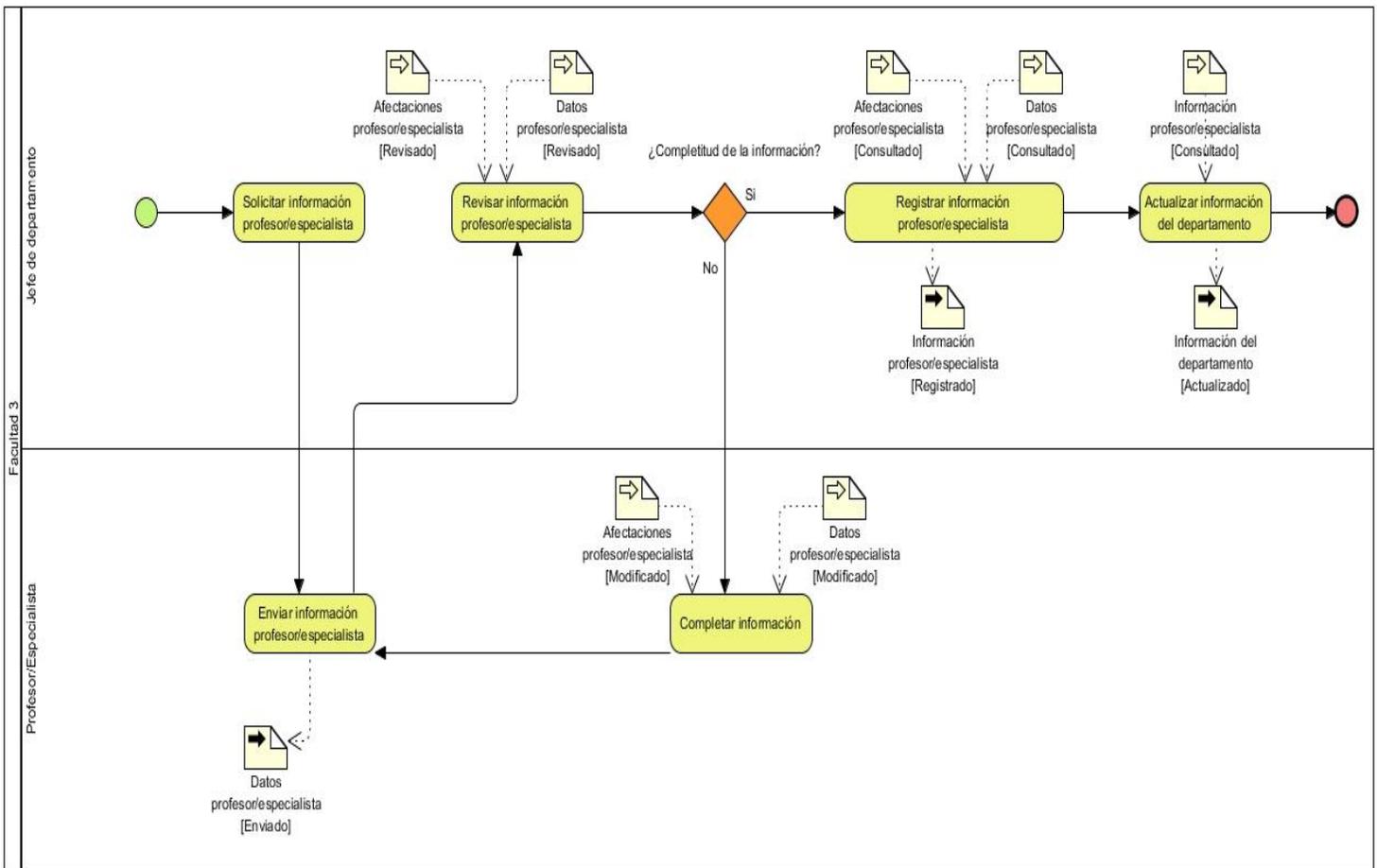
### 3.a Información incompleta

1. Completar información: el profesor/especialista completa la información personal o las afectaciones.  
**Continúa el flujo básico en la actividad 2.**

### Pos-condiciones

### Salidas

### Asuntos pendientes



**Figura 2.** Diagrama del proceso de negocio Levantar el potencial por departamentos

Los artefactos generados en el modelado del negocio constituyen entradas de la disciplina Requisitos, facilitando las actividades correspondientes que continuación se ofrecen.

## 2.3 Requisitos

El objetivo principal de la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Para su realización se generaron los artefactos que dicta la metodología de desarrollo de software utilizada (9).

### 2.3.1 Técnicas de captura de requisitos

En función de la identificación de los requisitos funcionales y no funcionales del sistema se hizo necesaria la utilización de técnicas existentes para la captura de requisitos. Se aplicó específicamente la entrevista a profesores, Jefes de departamento y al VDIP de la Facultad 3 (ver Anexo 2), tormentas de ideas y reuniones, donde el cliente explicó de forma explícita sus necesidades.

### 2.3.2 Requisitos funcionales

Los requisitos funcionales no son más que capacidades o condiciones que el sistema debe cumplir, estos especifican comportamientos particulares de un sistema. A continuación se listan los requisitos funcionales identificados (32).

#### **RF1. Gestionar profesor/especialista:**

- RF1.1 Adicionar profesor/especialista.
- RF1.2 Modificar profesor/especialista.
- RF1.3 Eliminar profesor/especialista.
- RF1.4 Consultar profesor/especialista.

#### **RF2. Gestionar categorías profesor/especialista:**

- RF2.1 Adicionar categorías profesor/especialista.
- RF2.2 Modificar categorías profesor/especialista.
- RF2.3 Eliminar categorías profesor/especialista.
- RF2.4 Consultar categorías profesor/especialista.

#### **RF3. Gestionar evidencias profesor/especialista:**

- RF3.1 Adicionar evidencias profesor/especialista.
- RF3.2 Modificar evidencias profesor/especialista.
- RF3.3 Eliminar evidencias profesor/especialista.
- RF3.4 Consultar evidencias profesor/especialista.

#### **RF4. Revisar evidencias del profesor/especialista.**

#### **RF5. Enviar planificación.**

#### **RF6. Gestionar notificación.**

- RF6.1 Enviar notificación.
- RF6.2 Consultar notificación.
- RF6.3 Eliminar notificación.

#### **RF7. Descargar estructura de carpetas de evidencias.**

#### **RF8. Gestionar desglose de indicadores del profesor/especialista.**

- RF8.1 Elaborar desglose de indicadores del profesor/especialista.

- RF8.2 Modificar desglose de indicadores del profesor/especialista.
- RF8.3 Eliminar propuesta de desglose de indicadores del departamento.
- RF8.4 Consultar desglose de indicadores del profesor/especialista.

### **RF9. Gestionar propuesta de desglose de indicadores del departamento**

- RF9.1 Elaborar propuesta de desglose de indicadores del departamento.
- RF9.2 Modificar propuesta de desglose de indicadores del departamento.
- RF9.3 Consultar propuesta de desglose de indicadores del departamento.
- RF9.4 Eliminar propuesta de desglose de indicadores del departamento.
- RF9.5 Aceptar propuesta de desglose de indicadores del departamento.

### **RF10. Gestionar planificación de indicadores de la facultad**

- RF10.1 Elaborar planificación de indicadores de la facultad.
- RF10.2 Modificar planificación de indicadores de la facultad.
- RF10.3 Consultar planificación de indicadores de la facultad.

### **RF11. Generar reporte**

- RF11.1 Reportar resumen de evidencias profesor/especialista.
- RF10.2 Reportar resumen de evidencias departamento.
- RF11.3 Reportar resumen de evidencias facultad.
- RF11.4 Reportar potencial de los profesores.
- RF11.5 Reportar planificación actual

### **RF12. Exportar información.**

- RF12.1 Exportar resumen de evidencias departamento.
- RF12.2 Exportar resumen de evidencias facultad.
- RF12.3 Exportar potencial de los profesores.
- RF12.4 Exportar planificación actual.

### **RF13. Gestionar nomenclador**

- RF13.1 Adicionar nomenclador.
- RF13.2 Modificar nomenclador.
- RF13.3 Eliminar nomenclador.

- RF13.4 Consultar nomenclador.

### 2.3.3 Descripción de requisitos

A continuación se presenta la descripción del requisito funcional: Adicionar profesor/especialista presente en el artefacto Descripción de requisitos del Expediente de proyecto del SGIF3.

**Tabla 4.** Descripción del requisito funcional Adicionar profesor/especialista

<b>Precondiciones</b>	El usuario no se ha registrado anteriormente.
<b>Flujo de eventos</b>	
<b>Flujo básico Adicionar profesor/especialista</b>	
1.	El sistema valida (ver validación 2) que se pueda adicionar información del profesor/especialista.
2.	Se introducen los datos para adicionar la información del profesor/especialista. Nombre, Apellidos, Área, Usuario, Contraseña, Contraseña UCI
3.	El sistema valida (ver validación 1) los datos introducidos.
4.	Si los datos son correctos el sistema los registra.
5.	El sistema confirma el registro de los datos.
6.	Concluye el requisito.
<b>Post-condiciones</b>	
1.	Se registró en el sistema la información del profesor/especialista.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 5.a Información errónea</b>	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
<b>Post-condiciones</b>	
1	N/A
<b>Flujo alternativo 5.a Información incompleta</b>	
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 4 del flujo básico.
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo *.a El usuario cancela la acción</b>	
1	Concluye el requisito.
<b>Pos-condiciones</b>	
1	No se registran los datos.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo conceptual.
2	El usuario no tiene los permisos para adicionar.
<b>Conceptos</b>	<b>Profesor/especialista</b> Visibles en la interfaz: Nombre, Apellidos, Departamento, Estado Utilizados internamente: N/A
<b>Requisitos especiales</b>	N/A
<b>Asuntos pendientes</b>	N/A

### 2.3.4 Validación de los requisitos funcionales

La validación de requisitos tiene como misión demostrar que la definición de los requisitos representa el sistema que el usuario necesita o el cliente desea. De esta forma puede asegurarse que los requisitos validados representan una descripción aceptable del sistema que se debe implementar. Esto implica verificar que los requisitos sean consistentes y que estén completos (32).

Para la validación de los requisitos funcionales identificados se utilizó la técnica: Construcción de prototipos. Su empleo ofreció como resultado que el cliente tuviera una visión inicial de la estructura de la interfaz de usuario que tendría el sistema. Se aplicaron además criterios para evaluar los requisitos del cliente, entre los que se destacan: transacciones, reutilización, criticidad, dependencia, estabilidad y frecuencia; características que reflejan la complejidad y prioridad para el cliente de cada uno de los requisitos (recogidos en el Expediente de proyecto del SGIF3 en el documento de Evaluación de los requisitos). Se consideraron además para validar los requisitos del cliente, criterios como: trazabilidad, ambigüedad, identificador único y completitud; elementos que definen la aprobación del requisito (ver documento Criterios para validar los requisitos del cliente que se encuentra en el Expediente de proyecto del SGIF3).

Por último se aplicó la Revisión técnica formal a cada descripción de los requisitos identificados. Con la utilización de esta técnica se validó que no existieran errores en el contenido o malas interpretaciones, información incompleta, inconsistencias y que los requisitos no fueran contradictorios, imposibles o inalcanzables, obteniendo como resultado que fueran aprobados por el cliente en el Acta de aceptación de requisitos (ver Anexo 3).

### 2.3.5 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que debe tener el producto. Son las características que hacen al producto atractivo, usable, rápido o confiable. Entre los requisitos no funcionales del sistema propuesto se encuentran (32):

#### Usabilidad

**RNF1.** El sistema debe presentar una interfaz amigable que permita la fácil interacción y llegar de manera rápida y efectiva a la información buscada. Debe ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

**RNF2.** Las etiquetas de cada funcionalidad y los campos de cada interfaz tendrán títulos asociados a su función de negocio y los mensajes de error deberán incluir una descripción textual del error.

**RNF3.** La consistencia de la interacción entre usuario y sistema estará determinada por el diseño de la interfaz de usuario que mantendrá los elementos como menús, banners y zona de trabajo, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones.

### **Seguridad**

**RNF4.** Se debe garantizar la integridad, disponibilidad, confidencialidad y autenticidad de la información manipulada por el sistema.

**RNF5.** El sistema debe verificar que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.

**RNF6.** El sistema debe mostrar las funcionalidades de acuerdo a los permisos del usuario que se encuentre activo.

### **Disponibilidad**

**RNF7.** La información se encontrará disponible en todo momento para usuarios autorizados a acceder al sistema.

### **Restricciones de diseño**

**RNF8.** La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará basándose en definiciones de procesos del negocio.

**RNF9.** La capa del negocio mantendrá el estado de los procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.

**RNF10.** La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas.

### **Soporte**

**RNF11.** El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.

### **Rendimiento**

**RNF12.** Teniendo en cuenta que el producto se debe diseñar sobre una arquitectura Cliente Servidor y el nivel de concurrencia que pueda existir, el sistema debe ser capaz de prestar servicio sin que se deterioren los tiempos de respuestas.

**RNF13.** El sistema debe garantizar el acceso concurrente para todos los usuarios del sistema durante la jornada laboral establecida.

### **Interfaz**

**RNF14.** El sistema debe contar con una interfaz fácil de usar y sencilla permitiendo que los usuarios sean capaces de interactuar con la aplicación.

**RNF15.** Será diseñada para adaptarse a la resolución del usuario, utilizando colores refrescantes, agradables y un diseño visual de acuerdo al perfil del usuario.

### **Portabilidad**

**RNF16.** El sistema debe ser compatible con los sistemas operativos Windows y Linux.

### **Escalabilidad**

**RNF17.** El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial del sistema.

### **Validación de información**

**RNF18.** El sistema debe validar automáticamente la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos y manejo de tipos de datos.

### **Hardware**

#### Cliente:

**RNF19.** Requisitos mínimos: procesador Pentium IV a 800 MHz, 1gb de memoria RAM y un navegador web.

**RNF20.** Tarjeta de red.

#### Servidor:

**RNF21.** Requisitos mínimos: procesador Dual Core a 3.00 GHz, 4gb de memoria RAM y una capacidad de 1Tb de disco duro.

**RNF22.** Tarjeta de red.

### **Software**

#### Cliente:

**RNF23.** Sistema operativo con interfaz gráfica y soporte para red.

**RNF24.** Las interfaces deben ser compatibles con Mozilla Firefox 3.5 o superior.

#### Servidor:

**RNF25.** Un servidor de aplicaciones web Apache 2.2.21 o superior.

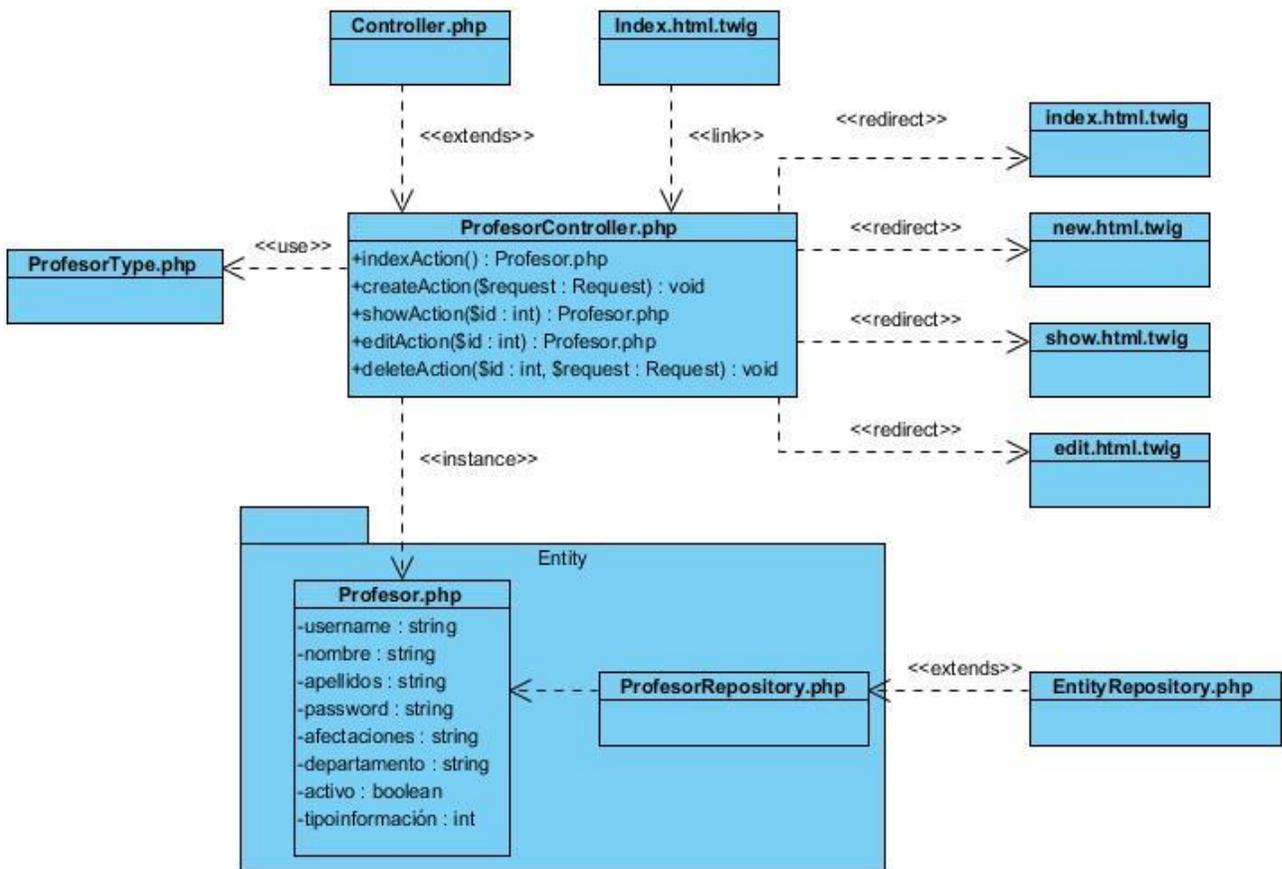
**RNF26.** Gestor de base de datos PostgreSQL 9.1 o superior.

## **2.4 Análisis y diseño**

En el presente epígrafe se especifican los artefactos generados como parte de la disciplina Análisis y diseño que facilitan el modelado del sistema.

## 2.4.1 Diagrama de clases del diseño

El diseño de los diagramas de clases permite describir la estructura del sistema mostrando sus clases, atributos y sus relaciones. A continuación se muestra el diagrama de clase del diseño del requisito funcional Gestionar profesor/especialista, donde se representan las principales clases, operaciones y relaciones necesarias para la realización del requisito. Los diagramas de clases del diseño de los requisitos restantes se encuentran en el Expediente de proyecto del SGIF3.



**Figura 3.** Diagrama de clase del requisito funcional Gestionar profesor/especialista

A continuación se muestra la descripción de las clases del diseño del requisito funcional Gestionar profesor/especialista.

**Tabla 5.** Descripción del diseño de clases del requisito funcional Gestionar profesor/especialista

Clases	Descripción
Controller	Controlador base que proporciona métodos a las características comunes que se necesitan en los controladores.
Index	Página principal de la aplicación.

Profesor_Controller	Clase controladora que maneja las acciones realizadas por el usuario; tiene la información de la solicitud, crea y devuelve una respuesta.
Index_Profesor	Página encargada de visualizar el listado de los profesores
Create_Profesor	Página encargada de visualizar el formulario para adicionar la información del profesor.
Edit_Profesor	Página encargada de visualizar el formulario para modificar la información del profesor.
Show_Profesor	Página encargada de visualizar el formulario para consultar la información del profesor.
EntityRepository	Clase que contiene métodos predefinidos para la recuperación de datos.
Profesor	Clase que contiene los atributos asociados al profesor.
Profesor_Repository	Clase que contiene las consultas dql para acceder a los datos almacenados en la clase profesor y extender las funcionalidades del repositorio.
ProfesorType	Clase que provee las características y validación de los campos de los formularios relacionados con la información del profesor.

### 2.4.2 Diseño de la base datos

El diseño de la base de datos es la herramienta conceptual para la descripción de datos, relaciones entre ellos, semántica y restricciones de consistencia. Tal representación permite contar con una vista del modelo a implementar a través de la perspectiva de entidades.

#### Modelo de datos

El modelo Entidad Relación (E-R) es un modelo de datos de alto nivel. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades y de relaciones entre estos objetos (35).

El modelo de datos propuesto en la solución cuenta con un total de 35 tablas, de ellas 2 son encargadas de la seguridad, 5 son nomencladores encargados de gestionar conceptos específicos del negocio ya predefinidos, ejemplo: la tabla “nom\_departamento” gestiona los departamentos definidos para los profesores. Las 28 tablas restantes se encargan de gestionar los datos que son necesarios tener registrados, por ejemplo, la tabla “dat\_profesor\_especialista” almacena los principales datos relacionados con los profesores.

En el diseño del Modelo de datos del SGIF3 se evidencia el empleo de varios patrones como solución a problemas comunes en el diseño de la base de datos. El patrón “llave subrogadas” genera una llave primaria única para cada entidad en vez de usar un atributo identificador. En la aplicación del mismo se definen secuencias de valores enteros como llaves primarias, que permite que las tablas sean más fáciles de consultar por el identificador dado que se conoce el mismo tipo en cada tabla.

En la gestión de los datos relacionados con la planificación se utilizó el patrón “árbol simple” para la tabla “dat\_propuesta”, en la modelación de la relación propuesta-contrapropuesta (padre-hijo), donde los nodos padre e hijos representan elementos del mismo tipo. En la relación no existen ciclos, un hijo no puede ser su propio padre.

El modelo de datos que a continuación se presenta es un fragmento de la estructura lógica de los datos del SGIF3, donde se especifica de las entidades que guardan la información de cada profesor/especialista: tipo de dato, forma en que se relacionan y las restricciones de integridad. Para consultar el Modelo de datos del SGIF3 (ver Anexo 4).

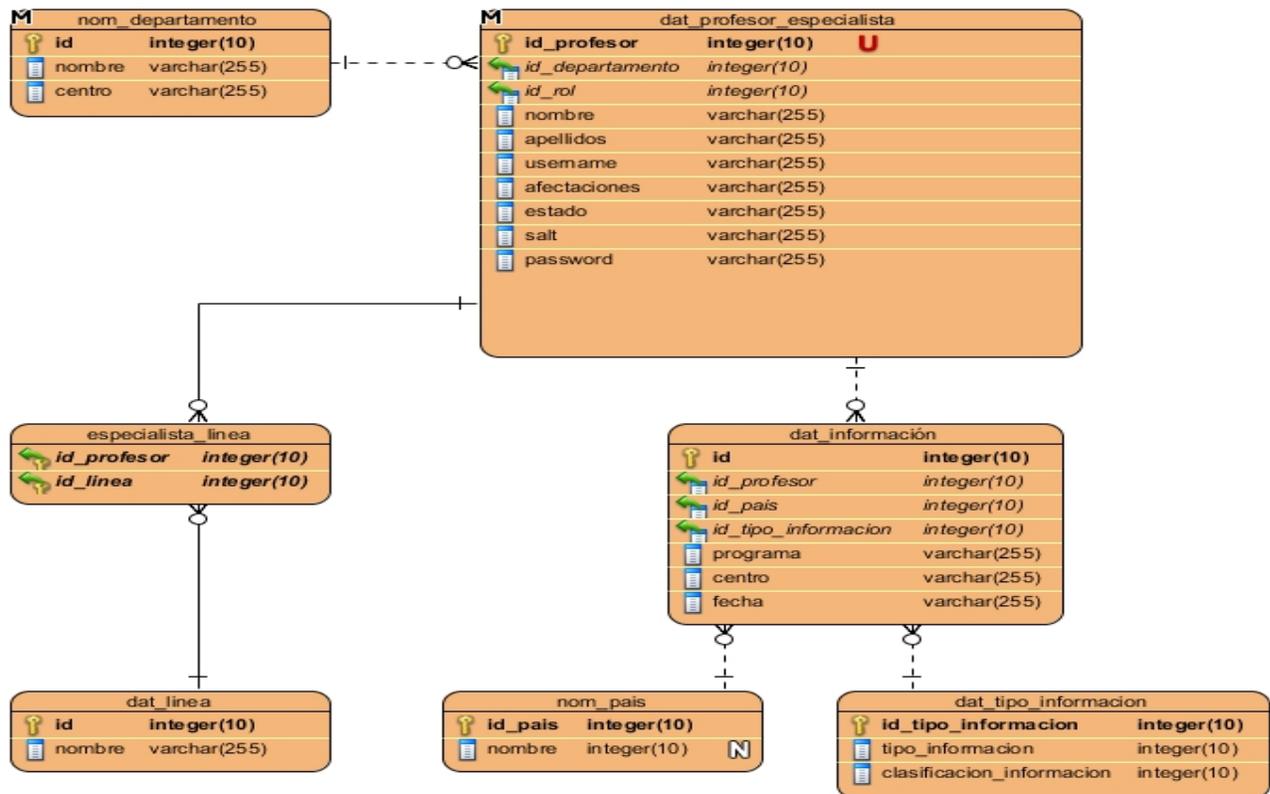


Figura 4. Fragmento del Modelo de datos del SGIF3

### 2.4.3 Patrones de diseño utilizados

El uso de patrones de diseño en la generación de los artefactos necesarios para el desarrollo facilitó crear una entrada apropiada como punto de partida a las actividades de implementación. Los patrones empleados en el proceso de desarrollo de la solución son mencionados a continuación:

**Patrones GRASP** describen los principios fundamentales de la asignación de responsabilidades a objetos.

- **Experto:** indica que la responsabilidad de la creación de un objeto o la implementación de un método debe recaer sobre la clase que conoce toda la información necesaria para ejecutar la tarea. Refuerza el encapsulamiento y redundancia en bajo acoplamiento. Se evidencia en el diseño del sistema en las clases entidades: Profesor, Indicador que contienen la información necesaria para la creación de objetos.
- **Creador:** ayuda a identificar quién debe ser el responsable de la instanciación o creación de nuevas clases u objetos. La visibilidad entre la clase creada y la clase creadora es una de las facilidades que se deriva del uso del patrón y además conduce a un bajo acoplamiento, supone facilidad de mantenimiento y reutilización. Se evidencia en la implementación de las clases controladoras: Profesor\_Controller, Indicador\_Controller, donde se encuentran las acciones definidas para el sistema y se crean los objetos de las clases que representan las entidades.
- **Controlador:** encargada de recibir los datos del usuario y enviarlos a las distintas clases según el método llamado a modo de intermediario entre una determinada interfaz y el algoritmo que la implementa. El patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control. La clase controladora Profesor\_Controller, es un ejemplo de la aplicación del patrón, la misma tendrá a cargo la responsabilidad de manejar los eventos relacionados con: la gestión de la información asociada al profesor.
- **Bajo acoplamiento:** las clases se encuentran lo menos relacionadas entre sí, para que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en las demás, potenciando la reutilización y disminuyendo la dependencia entre las clases. En el modelo de datos se definieron un conjunto de clases persistentes entre las que se establecieron las relaciones necesarias de manera que fueran más independientes y reutilizables para reducir el impacto de los cambios y acrecentar la oportunidad de una mayor productividad.

- **Alta cohesión:** indica que la información que almacena una clase debe ser coherente, de manera que todos sus métodos tengan un comportamiento bien definido. Si el sistema fallara por alguna razón es mucho más fácil encontrar responsabilidades si las clases del sistema son cohesivas. El patrón fue utilizado en el diseño del componente de manera general; agrupando las clases en dependencia de los requisitos, según la premisa de que la información almacenada en una clase debe ser coherente y estar relacionada con ésta en mayor medida, enfocada en sus responsabilidades.

**Patrones GOF** son soluciones concretas, técnicas, que se utilizan en situaciones frecuentes y favorecen la reutilización de código. A continuación se presentan los patrones GOF que se identifican en el funcionamiento del marco de trabajo Symfony:

- **Observador:** define una dependencia “uno-a-muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente. Este patrón consiste en un objeto llamado “container”, que tiene una lista de observadores, llamados “listeners” y cuando se llama a un método del objeto se notifica a los observadores de que se ha producido ese evento.
- **Decorador:** permite agregar funcionalidades y responsabilidades a objetos de forma dinámica y transparente para el usuario, esto se realiza por medio de relaciones con otras clases extendiendo su funcionalidad al incorporar las de las clases asociadas. En Symfony este patrón es visible en la plantilla base que es global en toda la aplicación y contiene el código HTML que es común a la mayoría de las páginas. Su uso aporta una mayor flexibilidad que la herencia estática, permitiendo, entre otras cosas, añadir una funcionalidad varias veces.
- **Método de fábrica:** El marco de trabajo Symfony define una interfaz para crear un objeto, pero deja que sean las subclasses quienes decidan que clase instanciar. Permite que una clase delegue en sus subclasses la creación de objetos. Las clases delegan la responsabilidad en una de entre varias clases auxiliares. Está presente en el marco de trabajo en las relaciones que existen entre las clases `FormTypeInterface`, `AbstractType` y cada una de las clases que representan los formularios.

### 2.5 Conclusiones del capítulo

En el capítulo que anterior se definió la propuesta de solución para el Sistema para la gestión de los indicadores de CTI en la Facultad 3, destacándose las siguientes conclusiones:

- La solución que se propone incluye en la planificación de los indicadores, acciones relacionadas con el levantamiento del potencial de las áreas y la negociación con estas de los

objetivos del año; mientras que en lo relacionado al control, acciones relacionadas con la recolección, validación y reporte de las evidencias registradas por los profesores/especialistas de cada área. No incluye evaluación de los objetivos del área de resultados claves dado que este objetivo se recoge en investigaciones relacionadas con las que se comparte un eje común.

- El modelo conceptual y la descripción de los procesos permitieron identificar 43 requisitos funcionales con los que la solución debe de cumplir para su correspondencia con las expectativas del cliente. De ellos el 26% de complejidad alta, 44% con complejidad media. Mientras que el 35% presentan una prioridad alta y el resto prioridad media. Así como 26 requisitos no funcionales, entre los que destacan de tipo: Usabilidad, Seguridad, Restricciones de diseño, Rendimiento; por solo mencionar las categorías más pobladas.
- El diseño exhibe la utilización de patrones de asignación de responsabilidades GRASP y los patrones GOF embebidos en el marco de trabajo Symfony.

### CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN

El capítulo que a continuación se presenta comprende la implementación del SGIF3, a partir de los resultados del Análisis y diseño. Serán especificados los estándares de codificación a utilizar en el código fuente, así como los diagramas de componente y despliegue para mostrar la comunicación del sistema. Incluye además la validación del diseño mediante métricas de calidad y el resultado de las pruebas a nivel de unidad aplicadas al sistema para evaluar su comportamiento. Finalmente se presentan los resultados de la comparación de los resultados de su aplicación a dos áreas de la facultad para evaluar el comportamiento de las variables.

#### 3.1 Implementación de la solución

La disciplina Implementación de la fase de Ejecución de la metodología AUP es fundamental en el desarrollo de software; se define y organiza el código de la propuesta de solución. Entre los principales objetivos cabe destacar la implementación de los diferentes elementos del diseño y los artefactos obtenidos del capítulo anterior.

##### 3.1.1 Estándares de codificación

Los estándares de codificación se definen para establecer reglas sobre la escritura del código fuente, logrando un estilo de programación homogéneo y entendible por todos los participantes, siendo en consecuencia apto al mantenimiento (36). A continuación se definen las tres secciones fundamentales de los estándares de codificación:

- Convención de nomenclatura: define cómo nombrar variables, funciones y clases.
- Convenciones de legibilidad de código: es la forma de organizar el código y lograr que independientemente de quién desarrolle se entienda como un todo.
- Convenciones de documentación: define cómo establecer comentarios, archivos de ayuda, entre otros.

#### Nomenclaturas utilizadas

- **PascalCase:** establece que los nombres de los identificadores, las variables, métodos y clases están compuestos por una o más palabras juntas, iniciando cada palabra con letra mayúscula y el resto en minúscula. Todas las clases están nombradas siguiendo el estándar PascalCase, de acuerdo al propósito y la función que responden. Ejemplo: las clases ProfesorController y IndicadorController.
- **CamelCase:** es similar a PascalCase, la diferencia consiste en que la letra inicial del identificador no comienza con mayúscula. Esta notación fue utilizada para el nombre de funciones, atributos y variables.

### **Nomenclatura de las clases**

Los nombres de las clases deben comenzar con mayúscula y en el caso de que la clase contenga un nombre compuesto entonces se escribe la segunda palabra con minúscula seguida sin el uso de artículos, contribuyendo a un buen entendimiento a la hora de leer el nombre de la clase.

### **Nomenclatura según el tipo de clases**

Clases controladoras: las clases controladoras después del nombre llevan la palabra: “Controller”.

Clases entidad: las clases entidad después del nombre llevan la palabra: “Entity”.

### **Nomenclatura de las funcionalidades**

Los nombres de las funciones se escriben con la primera letra minúscula y si es compuesto con mayúscula la inicial de la segunda palabra.

Clases controladoras: en las clases controladoras las funcionalidades luego de escribir el nombre llevan la palabra: “Action”.

### **Llaves de apertura y cierre**

Son utilizadas llaves de apertura y cierre, incluso en situaciones en las que son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

### **Declaraciones**

Toda variable local tendrá que ser inicializada en el momento de ser declarada, exceptuando el caso en que su valor inicial dependa de una llamada a otro método en determinado momento de ejecución. Las declaraciones deben situarse al principio de cada bloque principal en el que se utilicen y nunca en el momento de su uso.

### **Nomenclatura de los comentarios**

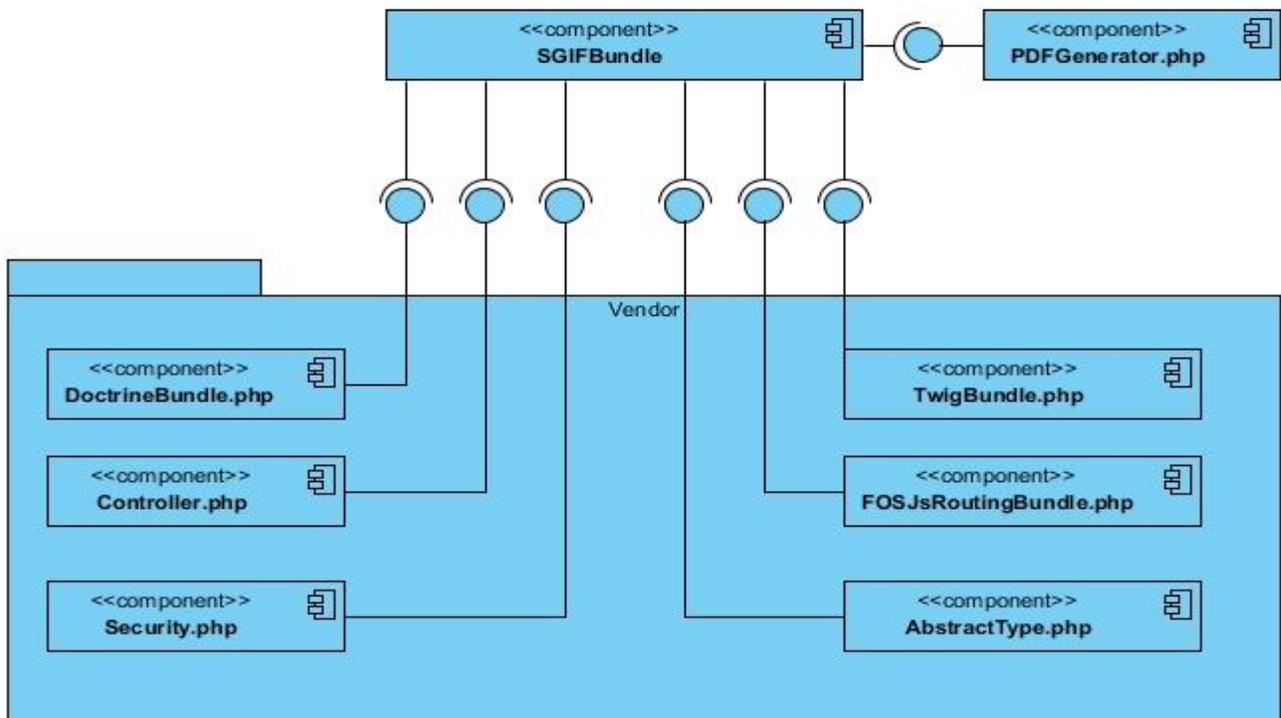
Los comentarios tienen que estar escritos claros y entendibles con el propósito de evidenciar lo que se está desarrollando y así poder tener una mejor comprensión del código.

#### **3.1.2 Diagrama de componentes**

El diagrama de componentes representa de forma general las relaciones entre los componentes físicos utilizados para el desarrollo de una aplicación. Generalmente un componente físico puede contener clases, ejecutables, documentos, binarios, bibliotecas, ficheros o tablas. Un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones pero un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clases (37).

## Capítulo 3. Implementación y validación de la solución

Seguidamente se muestra, en la Figura 5, el diagrama de componentes obtenido luego del desarrollo de la solución, donde se visualizan con más facilidad la estructura general del sistema y el comportamiento de los servicios que estos componentes proporcionan y utilizan.



**Figura 5.** Diagrama de componentes

El diagrama de componentes permitió traducir el modelo de diseño en un software operacional. En función de informatizar los procesos de planificación, recolección de evidencias de CTI, el componente SGIFBundle representa las interfaces y clases principales para la gestión del negocio. Para la generación de reportes en formato pdf consume los servicios del componente PDFGenerator.php.

SGIFBundle se comunica con componentes o bibliotecas de terceros que están contenidas en el paquete Vendor y que son utilizados de forma individual por los servicios que brindan para resolver tareas comunes en el desarrollo web como:

- El control y manejo de la seguridad de la aplicación.
- Mapeo de peticiones a los métodos del controlador adecuado.
- Creación y envío de formularios.
- El manejo del sistema de plantillas.

- El acceso a datos.
- El control y manejo del negocio.

### 3.1.3 Diagrama de despliegue

El diagrama de despliegue se muestra en la Figura 6, en él se encuentran representadas las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Se utiliza como entrada principal en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia fundamental en su diseño.

La utilización del diagrama permitió apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor de aplicaciones web (Apache 2.2.21 o superior) y maneja los datos persistentes sobre el servidor de base de datos (PostgreSQL 9.1 o superior) y utiliza los servicios del directorio activo de la universidad, para autenticar el registro de los usuarios.

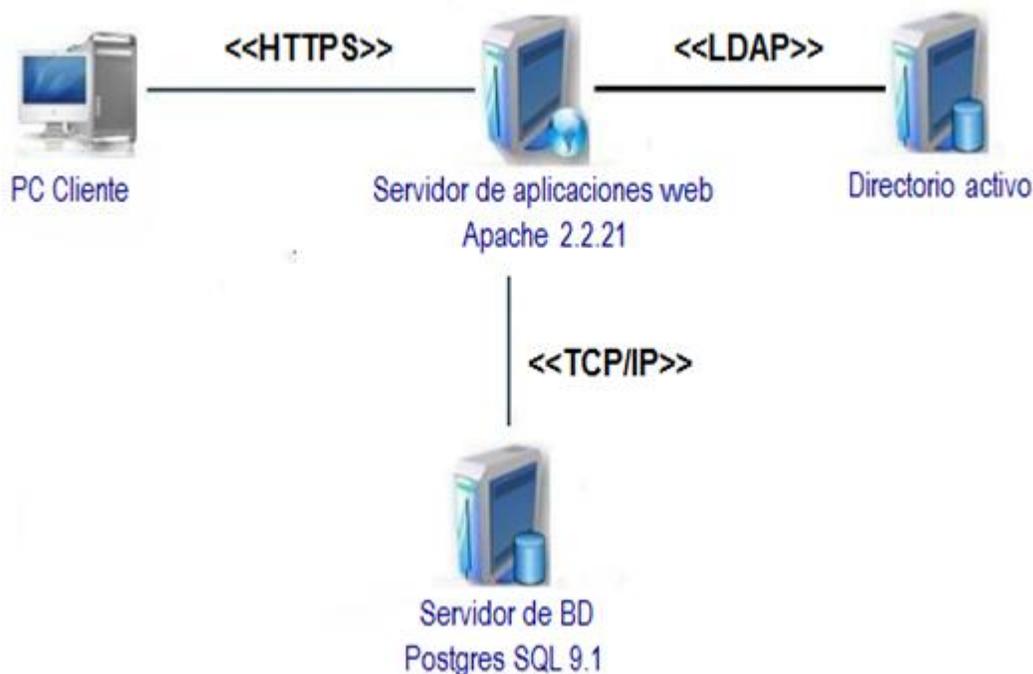


Figura 6. Diagrama de despliegue

### **Descripción de elementos e interfaces de comunicación**

#### **Nombre del tipo de conexión: características físicas de la conexión**

**HTTPS:** el Protocolo de Transferencia de Hipertexto Seguro es un protocolo que define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxis) para comunicarse. Es la combinación del protocolo HTTP y protocolos criptográficos. Se utiliza para conseguir conexiones seguras en la red, habitualmente para transacciones de pagos o cada vez que se intercambie información importante (38).

**TCP:** el Protocolo de Control de Transmisión permite a dos anfitriones establecer una conexión e intercambiar datos. Garantiza la entrega de datos, que no se pierdan durante la transmisión y también que los paquetes sean entregados en el mismo orden en el cual fueron enviados (39).

**LDAP:** el Protocolo Ligero/Simplificado de Acceso a Directorios (LDAP) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP es ampliamente utilizado por cualquier compañía u organización que requiera acceso a una red utilizando información de autenticación. Además de información de contactos o personas, es capaz de acceder a certificados encriptados, punteros a impresoras y otros servicios de una red y proporcionar un acceso único cuando una contraseña para un usuario está compartida entre determinados servicios. LDAP es apropiado para cualquier información de tipo directorio, donde la norma es acceder de forma rápida a dicha información y actualizarla de manera poco frecuente (40) (41).

### **3.2 Gestión de indicadores de CTI en el SGIF3**

El SGIF3 como resultado final del desarrollo de la propuesta de solución facilita la planificación y control de los indicadores anuales de la facultad, desglosado en áreas y estas a su vez por profesores/especialistas que la integran según su categoría docente y científica; mediante una interfaz amigable que a la misma vez da una mayor visibilidad y organización de la información a la hora de consultarla. A continuación se muestran algunas de las interfaces principales del sistema teniendo en cuenta los diferentes niveles de acceso:

Mediante el rol Vicedecano es posible acceder a las secciones de: Reportes, Evidencias, Planificación y Profesores. Es el rol con mayores privilegios en el sistema, tiene permisos para gestionar la creación, envío y aceptación de las propuestas de planificación para las áreas (ver Figura 7), gestionar usuarios, controlar las evidencias de las áreas, generar reportes del estado actual de la planificación y el cumplimiento de los indicadores (ver Figura 8), controlar el potencial académico, docente y científico de los profesores/especialistas.

**Adicionar propuesta para el área**

Área: CEGEL  
 Proyectos I+D: 2  
 Publicaciones: 3  
 Premios: 2  
 Patentes y registros: 4  
 Capacitaciones recibidas: 3  
 Capacitaciones ofertadas: 6  
 Resultados introducidos: 5  
 Trabajos presentados: 6  
 Empleos de estudiantes: 7

Tipo de información	Cantidad
Proyecto	41
Publicación	41
Patente y Registro	0
Capacitación recibida	50
Capacitación ofertada	50
Resultado introducido	50
Trabajo presentado	1
Empleo de estudiantes	41
Premio	1

Figura 7. Interfaz Adicionar propuesta para el área

**Plan actual de la facultad**

INDICADOR \ ÁREA	Facultad	CEIGE
Proyectos I+D	2	1
Publicaciones	2	1
Premios	22	11
Patentes y registros	2	1
Capacitaciones recibidas	2	1
Capacitaciones ofertadas	22	11
Resultados introducidos	2	1
Trabajos presentados	2	11
Empleos de estudiantes	2	11

Figura 8. Interfaz Plan actual de la facultad

El rol Jefe de departamento cuenta con funcionalidades de control y gestión de los usuarios, para ello dicho rol tiene permisos para activar los usuarios en correspondencia con el área, así como desglosar los indicadores de los profesores/especialistas por área (ver Figura 9), y validar las evidencias de cada uno de ellos.



Figura 9. Interfaz Adicionar desglose de indicador

El rol profesor es el de menor privilegio, tiene los permisos de agregar la información personal del usuario y la creación de las evidencias de los indicadores (ver Figura 10). Solo tiene acceso a las secciones Planificación y Evidencias.

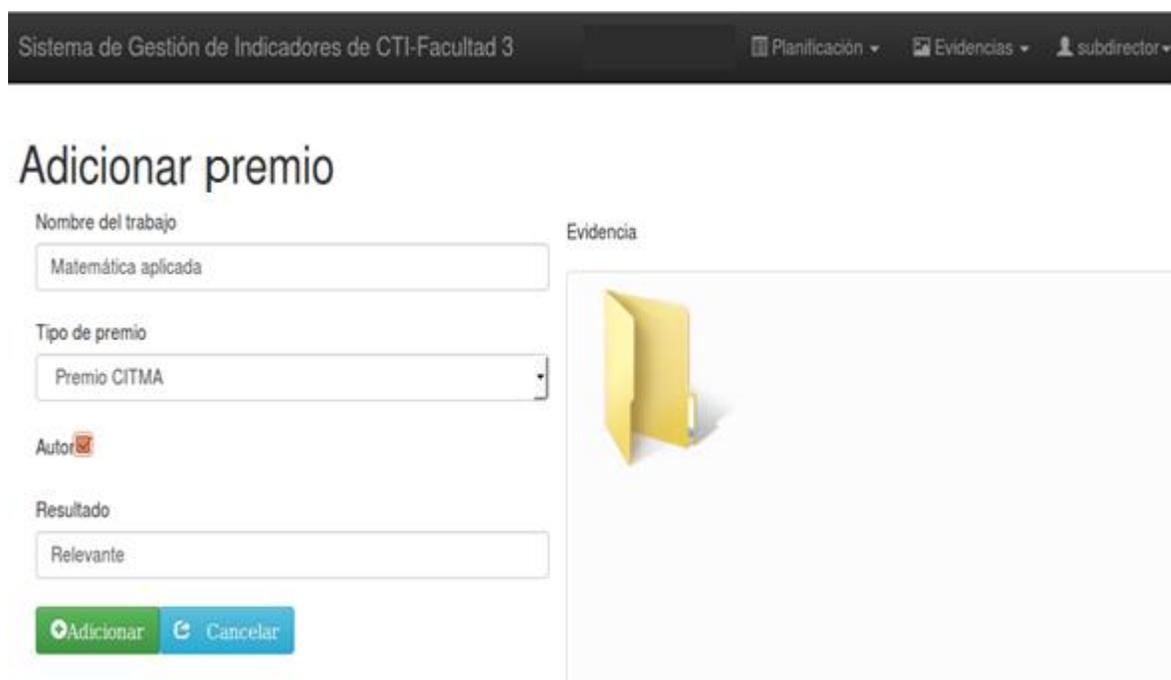


Figura 10. Interfaz Adicionar premio

### **Impacto de la solución:**

- Puntualiza los indicadores con los que debe cumplir cada usuario, en este caso profesor/especialista, garantizando que se encuentre identificado con las actividades de CTI que debe desempeñar en el año.
- Compartimentación de la información mediante la gestión de los roles y permisos de acceso a todos los niveles.
- Acumulación de evidencias digitales del cumplimiento de los indicadores planificados.
- Visibilidad de las evidencias por los Jefes de departamento correspondientes y revisión en tiempo real del cumplimiento de la planificación realizada para cada profesor/especialista subordinado.
- Actualización de los indicadores a medida que avanza el año.
- Disminución del tiempo de consulta de la información teniendo en cuenta que las evidencias en cada nivel están organizadas en la estructura de carpetas definida por la Dirección de investigación y postgrado de la facultad.
- Conteo automático de las evidencias en cada nivel reduciendo el margen de error.
- Apoyo a la toma de decisiones mediante el análisis y visualización de la información a través de los reportes predefinidos.
- Alta veracidad de la información del sistema, a partir de la asignación de responsabilidades en el manejo y revisión de las evidencias de los indicadores.

### **3.3 Verificación de la propuesta de solución**

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es medible y varía de un sistema a otro, convirtiéndose en un elemento de gran importancia en la actualidad ya que influye directamente en la competitividad de las empresas (42). Para poder detectar y corregir los posibles errores en el diseño e implementación de la solución se lleva a cabo la evaluación de las métricas TOC y RC para la validación del diseño y se describen las pruebas de software aplicadas.

#### **3.3.1 Resultado de la evaluación de las métricas TOC y RC**

La aplicación de métricas al diseño y la implementación de un producto de software constituyen un elemento fundamental a la hora de evaluar la calidad del mismo. Seguidamente se hará una evaluación del diseño propuesto para el sistema donde se tienen en cuenta tanto atributos como métricas de calidad.

## Capítulo 3. Implementación y validación de la solución

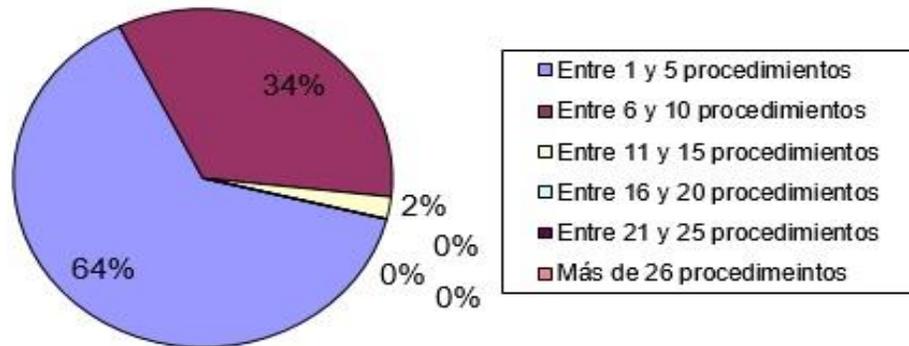
A continuación se presentarán las métricas aplicadas para evaluar la calidad del diseño propuesto:

**TOC:** se refiere al número de métodos pertenecientes a una clase. Está determinada por los atributos: Responsabilidad, Complejidad de implementación y Reutilización, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado.

**RC:** dado por el número de relaciones de uso de una clase. Está determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Cantidad de pruebas y Reutilización, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado.

### Resultados del instrumento de evaluación de la métrica TOC

Representación en por ciento de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

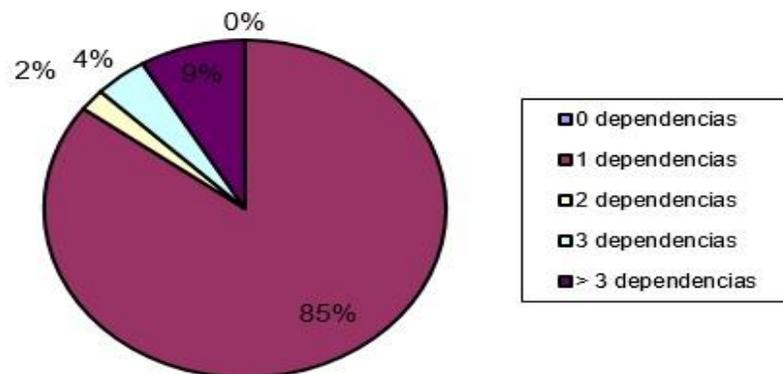


**Figura 11.** Representación en por ciento de los resultados obtenidos en el instrumento (TOC) agrupados en los intervalos definidos

El instrumento de medición de la métrica TOC se aplicó a un total de 47 clases. Al analizar los resultados obtenidos luego de la evaluación, se puede concluir que el diseño propuesto para el sistema está entre los límites aceptables de calidad, teniendo en cuenta que la mayoría de las clases (64%) posee menos cantidad de operaciones que la media registrada en las mediciones.

Los atributos de calidad evaluados se encuentran en un nivel satisfactorio; de manera que la responsabilidad que posee una clase respecto a otras, la repercusión que tiene un cambio o error en una clase dentro de la solución y el grado de dificultad de implementación del diseño en función del número de operaciones definidas en cada clase fue bajo para un 83% de las clases, viéndose favorecida la reutilización del diseño de software.

### Resultados del instrumento de evaluación de la métrica RC



**Figura 12.** Representación en porcentaje de los resultados obtenidos en el instrumento (RC) agrupados en los intervalos definidos

El instrumento de medición de la métrica RC se aplicó a un total de 47 clases. Al analizar los resultados obtenidos luego de la evaluación, se puede concluir que el diseño propuesto para el sistema está entre los límites aceptables de calidad, teniendo en cuenta que la mayoría de las clases (85%) posee dependencia directa con una sola clase y solamente un 9% más de tres dependencias.

Los atributos de calidad evaluados se encuentran en un nivel satisfactorio; de manera que el nivel de esfuerzo necesario para sustentar, mejorar o corregir la solución presentada, las dependencias o interconexión de una clase o estructura de clase respecto a otras y el número o grado de esfuerzo necesario para realizar las pruebas de calidad al sistema fue el bajo para 87% de las clases, viéndose favorecida la reutilización del diseño de software.

#### 3.3.2 Pruebas de software

Al conjunto de técnicas experimentales para la búsqueda de fallos en los programas, que determinan en cierto grado la calidad de un producto, se les denomina pruebas de software. La competencia mundial en el ámbito informático exige productos de calidad, de esta forma se muestra la necesidad de contar con pruebas de este tipo desde las primeras etapas de concepción de un proyecto (32).

#### Niveles de Prueba

A la hora de evaluar dinámicamente un sistema se debe comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Las pruebas se aplican en distintos niveles de trabajo, dentro de estos se distinguen:

Pruebas de unidad: prueba individual a las unidades separadas de un sistema de software.

## Capítulo 3. Implementación y validación de la solución

---

Pruebas de integración: los componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.

Pruebas del sistema: son usualmente conducidas para asegurar que todos los módulos trabajan como sistema sin error. Es similar a la prueba de integración pero con un alcance mucho más amplio.

Pruebas de aceptación: son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales (32).

**Métodos de Prueba**: enfoque sistemático, independiente del nivel en que se enmarque la prueba, que ayuda a encontrar buenos conjuntos de casos de prueba<sup>2</sup> para detectar diferentes tipos de errores.

Dos enfoques alternativos:

Caja negra: se comprueban las funcionalidades sin tener en cuenta la estructura interna.

Caja blanca: se comprueban los componentes internos.

**Objetivo**: conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. El aspecto fundamental que rige esta etapa de pruebas es determinar cómo y en qué sentido el componente cumple con las expectativas del cliente, a partir de los requisitos establecidos y las restricciones impuestas. En ese ámbito se trazan un conjunto de objetivos dentro de los que se sitúan:

1. Verificar la implementación del componente.
2. Verificar la integración adecuada del componente.
3. Verificar que todos los requisitos se han implementado correctamente.
4. Identificar los errores y asegurar que estos sean corregidos de la mejor manera.

Con la idea de diseñar pruebas que sistemáticamente muestren diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo (32).

---

<sup>2</sup> Casos de prueba: especifican una forma de probar el componente, incluye: la entrada, las condiciones bajo las cuales ha de probarse y los resultados esperados.

### Prueba de Software que será aplicada al sistema

**Descripción general de las pruebas para el Nivel de unidad:** se le denomina Prueba de unidad al proceso de separar y probar el correcto funcionamiento de las partes individuales de un programa para asegurar que cada uno de estos se ejecuta correctamente por separado.

Ventajas de usar este tipo de prueba:

1. Los errores son más fáciles de localizar.
2. Los errores están más acotados.
3. Se fomenta el cambio.

El autor partir de los conocimientos adquiridos durante la implementación del sistema se encargará de aplicar las pruebas para este nivel mediante los métodos antes presentados y las técnicas correspondientes que serán especificadas más adelante (32).

**Descripción y aplicación de la prueba de caja blanca o estructural:** la prueba de caja blanca también se conoce como prueba de caja transparente o de cristal. La prueba consiste específicamente en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa, examinándose la lógica interna sin considerar los aspectos de rendimiento.

Dentro de la prueba de caja blanca se incluyen las técnicas de pruebas que serán descritas a continuación:

Prueba del camino básico: permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto de caminos básicos.

Prueba de condición: ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.

Prueba de flujo de datos: se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.

Prueba de bucles: se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución todos los bucles en sus límites operacionales (32).

**Aplicación:** según la descripción de la prueba de caja blanca presentada con anterioridad y a partir de la necesidad de crear un producto de alta calidad es preciso valorar qué tan certera ha sido la

## Capítulo 3. Implementación y validación de la solución

implementación del sistema y para ello es necesario aplicar una de las técnicas que esta comprende, en este caso la del camino básico.

Para ello es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Se debe comenzar por un análisis del código, posteriormente son enumeradas cada una de las instrucciones, se construye el grafo de flujo asociado y según de las fórmulas pertinentes se calcula dicha complejidad:

1. Se seleccionó el algoritmo: indexAction correspondiente a la clase PublicacionController, mediante el cual se carga en la interfaz correspondiente el listado de evidencias de las publicaciones del año en dependencia del rol del usuario en sesión; publicaciones por facultad, publicaciones por departamento o publicaciones por profesor (ver Figura 13).
2. Se enumeraron las sentencias de código para luego construir el grafo de flujo asociado representado a través de nodos, aristas y regiones (ver Figura 14).
3. Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres vías o fórmulas de manera tal que quede justificado el resultado, siendo el mismo en cada caso:

```
public function indexAction() {  
    $em = $this->getDoctrine()->getManager(); //1  
    $sc = $this->get('security.context'); //1  
    $profesor = $this->getUser()->getId(); //1  
  
    if ($sc->isGranted('ROLE_VICEDECANO') || $sc->isGranted('ROLE_ADMIN')) { //2  
        $entities = $em->getRepository('SIGFBundle:Publicacion')->publicacionVicedecano(); //3  
        return $this->render('SIGFBundle:Publicacion:index.html.twig', array('entities' => $entities)); //4  
    }  
    else if ($sc->isGranted('ROLE_JEFEDPTO')) { //5  
        $idpto = $this->getUser()->getDepartamento()->getId(); //6  
        $entities = $em->getRepository('SIGFBundle:Publicacion')->publicacionByDpto($idpto); //6  
        return $this->render('SIGFBundle:Publicacion:index.html.twig', array('entities' => $entities)); //4  
    }  
    else { //7  
        $entities = $em->getRepository('SIGFBundle:Publicacion')->publicacionByProfesor($profesor); //8  
        return $this->render('SIGFBundle:Publicacion:index.html.twig', array('entities' => $entities)); //4  
    }  
}
```

Figura 13. Funcionalidad indexAction

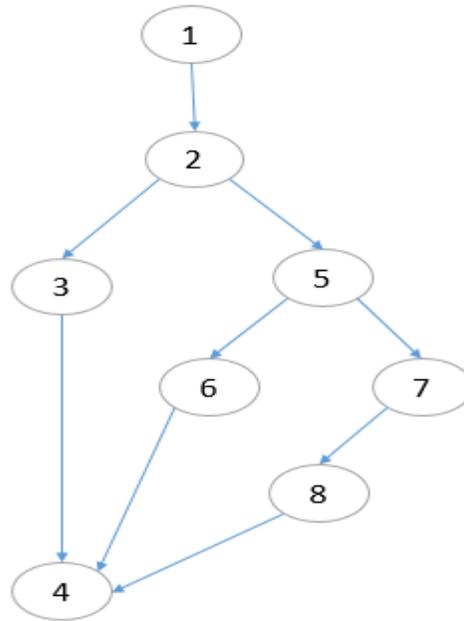


Figura 14. Grafo de flujo del camino básico

### Cálculo de la complejidad ciclomática:

$$V(G) = (9 - 8) + 2 \text{ (A: aristas, N: nodos)}$$

$$V(G) = (9 - 8) + 2 = 3$$

$$V(G) = P + 1 \text{ (P: nodos predicados)}$$

$$V(G) = 2 + 1 = 3$$

$$V(G) = R \text{ (R: regiones)}$$

$$V(G) = 3$$

El cálculo efectuado mediante las fórmulas antes presentadas muestran una complejidad ciclomática de valor 3, de manera que existen 3 posibles caminos por donde el flujo puede circular, este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario especificar los caminos básicos que puede tomar el algoritmo durante su ejecución. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás.

Camino básico #1: 1-2-3-4

Camino básico #2: 1-2-5-6-4

Camino básico #3: 1-2-5-7-8-4

Se procede a ejecutar los casos de pruebas para cada uno de los caminos básicos determinados en el grafo de flujo. Para definir los casos de prueba es necesario tener en cuenta:

## Capítulo 3. Implementación y validación de la solución

---

**Descripción:** se describe el caso de prueba y de forma general se tratan los aspectos fundamentales de los datos de entrada.

**Condición de ejecución:** se especifica cada parámetro para que cumpla una condición deseada y así ver el funcionamiento del procedimiento.

**Entrada:** se muestran los parámetros que serán la entrada al procedimiento.

**Resultados Esperados:** se expone el resultado esperado que debe devolver el procedimiento después de efectuado el caso de prueba.

A continuación, serán descritos los casos de prueba para los 3 caminos básicos determinados:

### **Caso de prueba para el camino básico # 1**

**Descripción:** mostrar en una lista las evidencias de las publicaciones del año, de todos los profesores de la facultad mediante el acceso a la entidad Publicación.

**Condición de ejecución:** para ejecutar el algoritmo es necesario que el usuario en sesión tenga asociado el rol de Vicedecano o Administrador del sistema.

**Entrada:**

\$rol = ROLE\_VICEDECANO

\$profesor = 5

**Resultados esperados:** listado de evidencias de las publicaciones correspondientes a un año de todos los profesores de la facultad al cual pertenece el usuario en sesión.

### **Caso de prueba para el camino básico # 2**

**Descripción:** mostrar en una lista las evidencias de las publicaciones del año asociadas a un departamento, mediante el acceso a la entidad Publicación.

**Condición de ejecución:** para ejecutar el algoritmo es necesario que el usuario en sesión tenga asociado el rol de Jefe de departamento.

**Entrada:**

\$rol = ROLE\_JEFEDPTO

\$idpto= 9

**Resultados esperados:** listado de evidencias de las publicaciones correspondientes a un año de todos los profesores del departamento al cual pertenece el usuario en sesión.

## Capítulo 3. Implementación y validación de la solución

---

### **Caso de prueba para el camino básico # 3**

**Descripción:** mostrar en una lista las evidencias de las publicaciones del año asociadas a un profesor, mediante el acceso a la entidad Publicación.

**Condición de ejecución:** para ejecutar el algoritmo es necesario que el usuario en sesión tenga asociado el rol de profesor.

**Entrada:**

\$rol = ROLE\_PROFESOR

\$profesor = 123

**Resultados esperados:** listado de evidencias de las publicaciones correspondientes a un año del usuario en sesión.

**Descripción y aplicación de la prueba de caja negra o funcional:** a este tipo de prueba también se le conoce como prueba de caja opaca o inducida por los datos. Se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación, esta prueba se limita a brindar solo datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo internamente, es decir, solo trabaja sobre su interfaz externa (32).

En esencia permite encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.

Dentro de la prueba de caja negra se incluyen las técnicas de pruebas que serán descritas a continuación:

**Partición de equivalencia:** divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

**Análisis de valores límites:** prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

**Grafos de causa-efecto:** permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

**Aplicación:** la prueba de caja negra se aplica al requisito funcional Gestionar evidencias–capacitación, en este caso incluye: Adicionar, Eliminar, Editar y Mostrar capacitación. La partición de

## Capítulo 3. Implementación y validación de la solución

equivalencia divide el dominio de entrada de un programa en un número finito de variables de equivalencia. Las variables de equivalencia representan un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa. Se definen dos tipos de variables de equivalencia, las válidas, que representan entradas válidas al programa y las no válidas, que representan valores de entrada erróneos, aunque pueden existir valores no relevantes a los que no sea necesario proporcionar un valor real (ver Anexo 7).

Luego de aplicados los métodos de prueba por el equipo de desarrollo al algoritmo indexAction y al requisito funcional Gestionar publicación antes presentados, es válido señalar que los resultados obtenidos fueron satisfactorios desde el punto de vista interno y funcional del componente, atendiendo al correcto comportamiento del mismo ante diferentes situaciones (entradas válidas y no válidas).

Una vez realizadas las pruebas unitarias por parte del equipo de desarrollo, se ejecutaron las pruebas de liberación definidas por parte del grupo de calidad que pertenece a la Subdirección del CEIGE, en un total de tres iteraciones de pruebas funcionales quedó evidenciado el correcto funcionamiento del sistema a partir de la resolución de las no conformidades detectadas. La evidencia de los resultados de las pruebas se registra en el documento Acta de liberación (ver Anexo 5). Como parte de las pruebas de aceptación se obtuvo un resultado positivo, dado que la solución desarrollada cumple con las preferencias del cliente, evidencia mostrada en el Acta de aceptación del cliente (ver Anexo 6).

### 3.4 Resultados del cuasi-experimento

Una vez verificado la validez desde el punto de vista ingenieril de los artefactos por fase de Ejecución y de obtener la liberación de la propuesta de solución por parte del equipo de calidad perteneciente a la Subdirección del CEIGE y de la aceptación por parte del cliente, se procede a evaluar si efectivamente la herramienta disminuye los tiempos de consulta y aumenta la disponibilidad y autenticidad de la información relativa a las evidencias.

Para ello primeramente se midieron los tiempos del levantamiento del potencial y negociación de los indicadores, así como de consulta de la información por el método tradicional y a través del sistema. La Tabla 6 muestra los elementos de comparación.

**Tabla 6.** Evaluación de la variable Tiempo de consulta

Aspecto	Medición de tiempo (días)		Evaluación	
	Método Tradicional	Sistema	Método Tradicional	Sistema
Levantamiento del potencial	6 días	2 días	Medio	Bajo

## *Capítulo 3. Implementación y validación de la solución*

Negociación de los indicadores	7 días	3 días	Medio	Bajo
Consulta de las evidencias	15 días	0 días	Alto	Bajo

En los casos de medición del tiempo tradicional los tiempos responden desde la solicitud a las áreas hasta que tuvo lugar el último despacho con estas para lograr cada uno de los aspectos objeto de evaluación. Mientras que se simuló en el sistema con los datos que estos reportaron por el método tradicional. Para el aspecto uno se asumió que la solicitud llegaba al área por vez primera y por tanto se necesitaba la carga inicial; incluso así los tiempos de evaluación fueron inferiores al método tradicional. Debe señalarse que no se dedicó jornadas específicas de trabajo a la introducción de datos al sistema sino que se hizo en los momentos en que el autor veía que el VDIP no tenía otra actividad en el plan de trabajo para simular una especie de carga de trabajo de los empleados en las áreas. Puede verse como los tiempos de consulta en cada uno de los aspectos disminuyen mediante la utilización de la herramienta.

Los resultados de la evaluación de la variable Seguridad pueden verse en la Tabla 7. En este caso igualmente la herramienta muestra superioridad por sobre el método tradicional.

**Tabla 7.** Evaluación de la variable Seguridad

Aspecto	Disponibilidad						Evaluación	
	Método Tradicional			Sistema			Método Tradicional	Sistema
	Alta	Media	Baja	Alta	Media	Baja		
Al capturar evidencias			X	X			Baja	Alta
Al reportar evidencias			X	X			Baja	Alta
Evaluación de la variable Disponibilidad							Baja	Alta
	Autenticidad							
Al validar evidencias		X		X			Media	Alta
Evaluación de la variable Autenticidad							Media	Alta
Evaluación de la variable Seguridad							Baja	Alta

En el primer aspecto la evaluación del método tradicional corresponde a que únicamente se tiene disponibilidad sobre una instancia de actualización de captura de evidencias cada 3 meses, por lo que no se tiene acceso en todo momento a información actualizada; mientras que mediante la herramienta no se necesita esperar los momentos de actualización sino que se presenta la posibilidad de hacerlo de forma continua. En lo relativo al aspecto de reporte de las evidencias al vicedecanato, la evaluación del método tradicional responde a que hubo pérdida de información por la longitud del

## *Capítulo 3. Implementación y validación de la solución*

---

nombre de varios archivos condicionado por la profundidad de este en el árbol de carpetas. Mientras que la herramienta presenta la posibilidad de descargarlos directamente del FTP que exporta. Por último se detectó mediante el método tradicional algunas evidencias que no se correspondían con la declaración de autoría, o repetidas, o de años anteriores; que pudo haberseles pasado a los activistas de las áreas por lo tedioso que resulta revisar una por una las evidencias en la estructura de carpetas. Mientras que la herramienta sólo reportó aquellas que habían sido previamente verificadas por el jefe de área.

En este aspecto también la herramienta muestra superioridad por sobre el método tradicional a partir de que exhibe un aumento en las variables Disponibilidad y Autenticidad, y en consecuencia en la Seguridad de la información relacionada con las evidencias del cumplimiento de los indicadores de CTI que se planificaron con anterioridad.

### **3.5 Conclusiones del capítulo**

En el capítulo que recién finaliza se realizó la implementación y validación del SGIF3 arribando a las siguientes conclusiones:

- Se determinaron estándares de codificación para asegurar una mejor comprensión del código que engloba nomenclatura de clases, atendiendo a nombre y tipo, así como funcionalidades con la utilización del patrón lowerCamell, y de comentarios.
- Se establecieron 9 aspectos para caracterizar el impacto de la solución entre los que destacan los relacionados con acumulación, compartición, y visibilidad de las evidencias; las variables dependientes de la investigación y el apoyo a la toma de decisiones asociadas a la gestión de la actividad de postgrado e investigación.
- De la verificación del sistema pudo constatarse que el diseño de clases presenta niveles adecuados de calidad, manifiesto en la simplicidad e independencia de estas, lo que se traduce en menor esfuerzo por reutilización y mayor disposición a la mantenibilidad.
- Se aplicaron pruebas de unidad, utilizando la técnica del camino básico que permitió corregir algunos errores en el código que no fueran defectos del software durante las pruebas de liberación por parte del grupo de calidad de la Subdirección del CEIGE.
- Se desarrolló un cuasi experimento orientado a evaluar el comportamiento de las variables dependientes en la investigación, por el método tradicional con datos reales y simulando su entrada en la herramienta. En todos los casos la propuesta de solución se mostró superior al método tradicional.

### **CONCLUSIONES GENERALES**

Con la culminación del presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir el objetivo propuesto, resaltando que:

- Se realizó el marco teórico referente a la investigación donde se establecieron las principales tendencias en cuanto a la gestión de indicadores de CTI en centros universitarios y de investigación.
- Se desarrolló el SGIF3 abarcando las etapas establecidas por la metodología de desarrollo mediante la utilización de las herramientas, lenguajes y tecnologías definidas.
- Fue comprobada la calidad de la solución propuesta a partir de los resultados satisfactorios obtenidos en las evaluaciones realizadas al sistema.

## RECOMENDACIONES

Luego de concluida la investigación se proponen las siguientes recomendaciones:

- Adicionar la funcionalidad de búsqueda de evidencias por años, que permita realizar un estudio histórico del cumplimiento de los indicadores al SGIF3.
- Integrar al SGIF3 un módulo de evaluación de Áreas de Resultados Claves, que posibilite la evaluación de los objetivos de trabajo para el año en cuestión.

## REFERENCIAS BIBLIOGRÁFICAS

1. La evaluación de la investigación científica: una aproximación teórica desde la ciencia métrica. Ricardo Arencibia Jorge, Félix de Moya Anegón. s.l. : Acimed, 2008.
2. Dunier Bichot González, Yosbel Echevarría Millares, Raúl Velázquez Álvarez. Guía para el aseguramiento de la seguridad lógica en el desarrollo de sistemas informáticos de gobierno electrónico. La Habana : s.n., 2012.
3. Jesús Ferrer. Conceptos básicos de la Metodología de la Investigación. [En línea] [Citado el: 20 de febrero de 2015.] [http://metodologia02.blogspot.com/p/operacionalizacion-de-variable\\_03.html](http://metodologia02.blogspot.com/p/operacionalizacion-de-variable_03.html).
4. Organización de los Estados Americanos (OEA), Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED). Red de Indicadores de Ciencia y Tecnología Iberoamericana e Interamericana. [En línea] 1994. [Citado el: 22 de enero de 2015.] <http://www.ricyt.org/>.
5. Ministerio de Ciencia Tecnología e Innovación Productiva. Sistema de Información de Ciencia y Tecnología Argentino. Sistema de Información de Ciencia y Tecnología Argentino. [En línea] [Citado el: 10 de febrero de 2015.] <http://www.sicytar.secyt.gov.ar/>.
6. Dayron Armas Peña, Maidelyn Díaz Pérez. Sistema Institucional para la Gestión de la Ciencia y la Técnica en Universidades: una perspectiva cuantitativa para su análisis y evaluación. Pinar del Río : s.n.
7. CITMATEL. Sistema de Información para la Gestión de Programas de Ciencia e Innovación. [En línea] CITMATEL, 2015. [Citado el: 20 de febrero de 2015.] <http://siprocit.ceniai.inf.cu/>.
8. Propuesta de solución para la gestión de la información de la actividad de ciencia, tecnología e innovación en la Universidad de las Ciencias Informáticas. Yunaysy Ortiz Batista, Yeslidier López Reinoso, Yordanis Medina León, Susana Gonce Fernández, David Batard Lorenzo, Jorge Gulín González. La Habana : s.n., 2010.
9. Rodríguez, Tamara. Metodología de desarrollo. La Habana : UCI, 2014.
10. CMMI, Equipo del Producto. CMMI-DEV, V1.3. [En línea] 2010. [Citado el: 20 de febrero de 2015.] <http://www.sei.cmu.edu>.
11. Sánchez, Tamara Rodríguez. Metodología de desarrollo. La Habana : UCI, 2014.
12. Arebas, Jesus Barranco. Metodologías del análisis estructurado de sistemas. s.l. : Madrid. ISBN 84-8468-043-6.
13. Seth Ladd y Darren, Davidson . Expert Spring MVC and Web Flow. s.l. : Appress, 2006.
14. Mairelys Fernández González, Osley Zorrilla Rivera. Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. La Habana : UCI, 2010.
15. Stevens, Pooley. El Lenguaje Unificado de Modelado. s.l. : Wesley, 2002.

16. Stephen A. White, Derek Miers. BPMN Modeling and Reference Guide. Florida, Estados Unidos : Future Strategies, 2009.
17. Lenguaje por parte del servidor o el cliente. [En línea] [Citado el: 25 de enero de 2015.] [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).
18. Introducción a JavaScript. Introducción a JavaScript. [En línea] [Citado el: 8 de diciembre de 2014.] <http://librosweb.es/javascript/>.
19. Pilgrim, Mark. HTML5 Up and Running. s.l. : O'Reilly, 2010.
20. Stig Saether Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben, Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, Jouni Ahto. Manual de PHP. s.l. : editado por Rafael Martínez.
21. Relacionados., Historia de PHP y Proyectos. Historia de PHP y Proyectos Relacionados. [En línea] [Citado el: 22 de enero de 2015.] <http://php.net/manual/es/history.php>.
22. Bootstrap. Bootstrap. [En línea] [Citado el: 18 de diciembre de 2014.] <http://twitter.github.io/bootstrap/>.
23. Symfony2. [En línea] [Citado el: 20 de enero de 2015.] <http://symfony.com/doc/2.3/book/index.html>.
24. Francois, Zaninotto Fabien Potencier. The Definitive Guide to Symfony (1era edición). s.l. : Apress, 2007. ISBN 9781430203797.
25. Introducción a Ajax. Introducción a Ajax. [En línea] 5 de diciembre de 2014. [Citado el: 5 de diciembre de 2014.] <http://www.librosweb.es/ajax/index.html>.
26. Visual Paradigm. [En línea] [Citado el: 20 de enero de 2015.] <http://www.visual-paradigm.com/support/vpuml/releasenotes/800.jsp>.
27. NetBeans. NetBeans. [En línea] [Citado el: 10 de diciembre de 2014.] <https://netbeans.org/community/releases/80/index.html>.
28. Apache. Apache. [En línea] Apache Foundation. [Citado el: 21 de diciembre de 2014.] <http://www.apache.org>.
29. PostgreSQL. Postgresql-es. [En línea] [Citado el: 10 de diciembre de 2014.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
30. 2010, CAVSI. Sistemas Gestores de Bases de Datos. [En línea] [Citado el: 21 de enero de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
31. Mozilla Foundation. Firefox. Firefox. [En línea] [Citado el: 21 de abril de 2015.] <http://www.getfirefox.net/>.
32. Pressman, Roger S. Ingeniería de software: Un enfoque práctico. La Habana : Félix Varela, 2005.

33. Yanirys Montes de Oca Hernández, Yuliesky Brito Díaz. Módulo para la Gestión de Información de Trámites Protocolizables Complejos en la notaría Buen Viaje de Santa Clara. [En línea] [Citado el: 20 de febrero de 2015.] <http://www.eumed.net/libros-gratis/2012b/1232/index.htm>.
34. Guillermo Storti, Gladys Ríos, Gabriel Campodónico. Bases de datos. Modelo Entidad Relación. [En línea] 2007. [Citado el: 25 de mayo de 2015.] [www.belgrano.esc.edu.ar/matestudio/carpeta\\_de\\_access\\_introduccion.pdf](http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf).
35. Lisandra Santamaria Pérez, Alejandro José Rodríguez Cruz. Gestor de reglas y criterios de medidas para evaluar el cumplimiento de objetivos estratégicos en el Sistema de Planificación de Actividades. La Habana : Universidad de las Ciencias Informáticas, 2014.
36. Microsoft. Diagramas de componentes de UML: Referencia. Developer Network. [En línea] [Citado el: 20 de mayo de 2015.] <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
37. Explicando el protocolo HTTPS. GenBeta Web+Software. [En línea] WSL, 5 de septiembre de 2011. [Citado el: 10 de mayo de 2015.] <http://www.genbeta.com/guia-de-inicio/explicando-el-protocolo-https>.
38. Kevin Fall, W. Richard Stevens. TCP/IP Illustrated, The Protocols. Michigan : Inc Pearson Education, 2011.
39. Ed Sermersheim. Lightweight Directory Access Protocol (LDAP): The Protocol. Lightweight Directory Access Protocol (LDAP): The Protocol. [En línea] [Citado el: 15 de mayo de 2015.] <http://tools.ietf.org/html/rfc4511?ref=driverlayer.com>.
40. Mozos, Guillermo Muñoz. Disector de protocolos LDAP. Madrid : UAM. Departamento de Tecnología Electrónica y de las Comunicaciones, 2014.
41. ACIMED. Un enfoque actual sobre la calidad del software. [En línea] Ciencias Médicas, 2 de febrero de 2015. [http://scielo.sld.cu/scielo.php?pid=S1024-94351995000300005&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94351995000300005&script=sci_arttext).

## ANEXOS

### Anexo 1. Comparación de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI

**Tabla 8.** Comparación de las herramientas y técnicas de apoyo a la gestión de indicadores de CTI

Parámetros	Herramientas y técnicas				
	<i>RICYT</i>	<i>SICYTAR</i>	<i>CuScienTI</i>	<i>SiPROCIT</i>	<i>SIndiCIT</i>
Tiempo de consulta de la información bajo	Si	Si	No	Si	Si
Alta disponibilidad de la información	Si	Si	No	No	Si
Autenticidad de la información	Si	Si	No	Si	Si
Escalabilidad	No	No	Si	No	Si
Generación de reportes	Si	Si	Si	Si	Si
Generación de reportes dinámicos	No	No	No	Si	No
Buena navegación	Si	No	No	Si	Si
Buen diseño	Si	Si	No	Si	No
Multiplataforma	Si	Si	Si	Si	Si

### Anexo 2. Entrevista realizada al personal involucrado en la gestión de indicadores de CTI en la Facultad 3

La siguiente entrevista desarrollada por estudiantes de Tesis de grado de la Facultad 3 está vinculada a los procesos de gestión de los indicadores de CTI.

Nombre y Apellidos: \_\_\_\_\_

Cargo: \_\_\_\_\_

Área: \_\_\_\_\_

1. ¿Cómo se realiza la comunicación entre los diferentes involucrados, con respecto a la realización de las tareas y períodos de entrega de la información y evidencias de los indicadores de CTI?
2. ¿Cómo se recoge la información asociada a cada profesor/especialista?
3. ¿Existe negociación de la planificación entre los distintos involucrados?

4. ¿Cómo se recoge en el área las evidencias que respaldan el sistema de indicadores de CTI?
5. ¿De qué manera se organiza la recolección de la información y las evidencias?
6. ¿Cuáles son los mecanismos utilizados para revisar la información y evidencias de los profesores/especialistas?
7. ¿Cómo se otorga la evaluación?
8. ¿Qué deficiencia se observa en el actual sistema de gestión de indicadores de CTI?
9. ¿Qué mejoras se propone para el actual sistema de gestión de indicadores de CTI?

**Anexo 3.** Acta de aceptación de requisitos

**UCI** Universidad de las Ciencias Informáticas | **CENTRO DE INFORMATIZACIÓN DE ENTIDADES**

RF11.4 Exportar potencial de la facultad.  
RF11.4 Exportar plan actual de la facultad.  
RF11.4 Exportar plan actual del departamento.

**RF13. Gestionar nomenclador**  
RF12.1 Adicionar nomenclador.  
RF12.2 Modificar nomenclador.  
RF12.3 Eliminar nomenclador.  
RF12.4 Consultar nomenclador.

Y para que así conste se suscribe la presente Acta en la de La Habana el 1ero de junio de 2015.

Nombre y Apellidos: MSc Yan Martínez Anguila  
Cargo: Vicedecano de Seguimiento e Investigación (en precursores)  
Firma: [Firma manuscrita]

**Figura 15.** Acta de aceptación de requisitos

Anexo 4. Diagrama de Modelo de datos del SGIF3

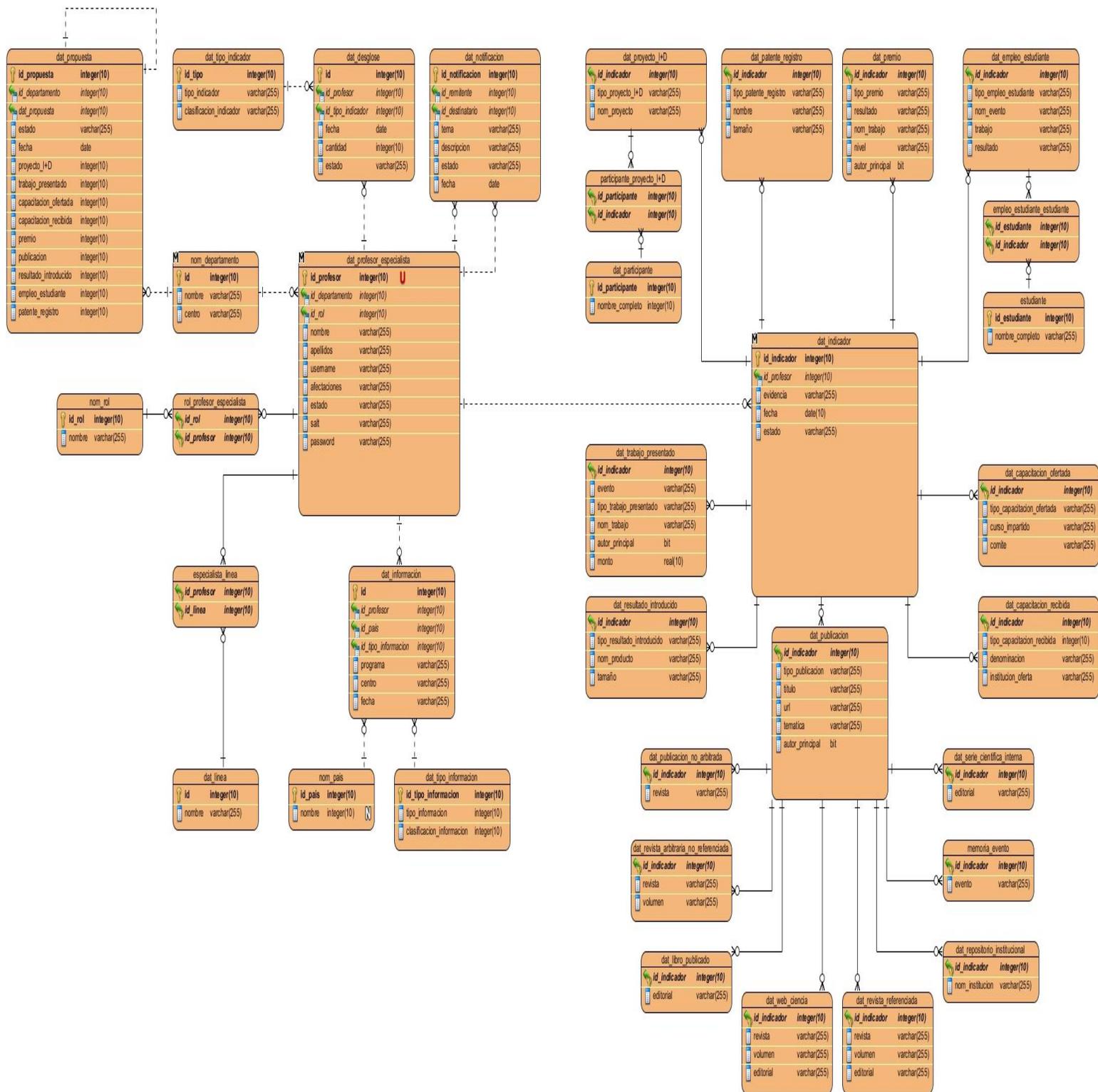
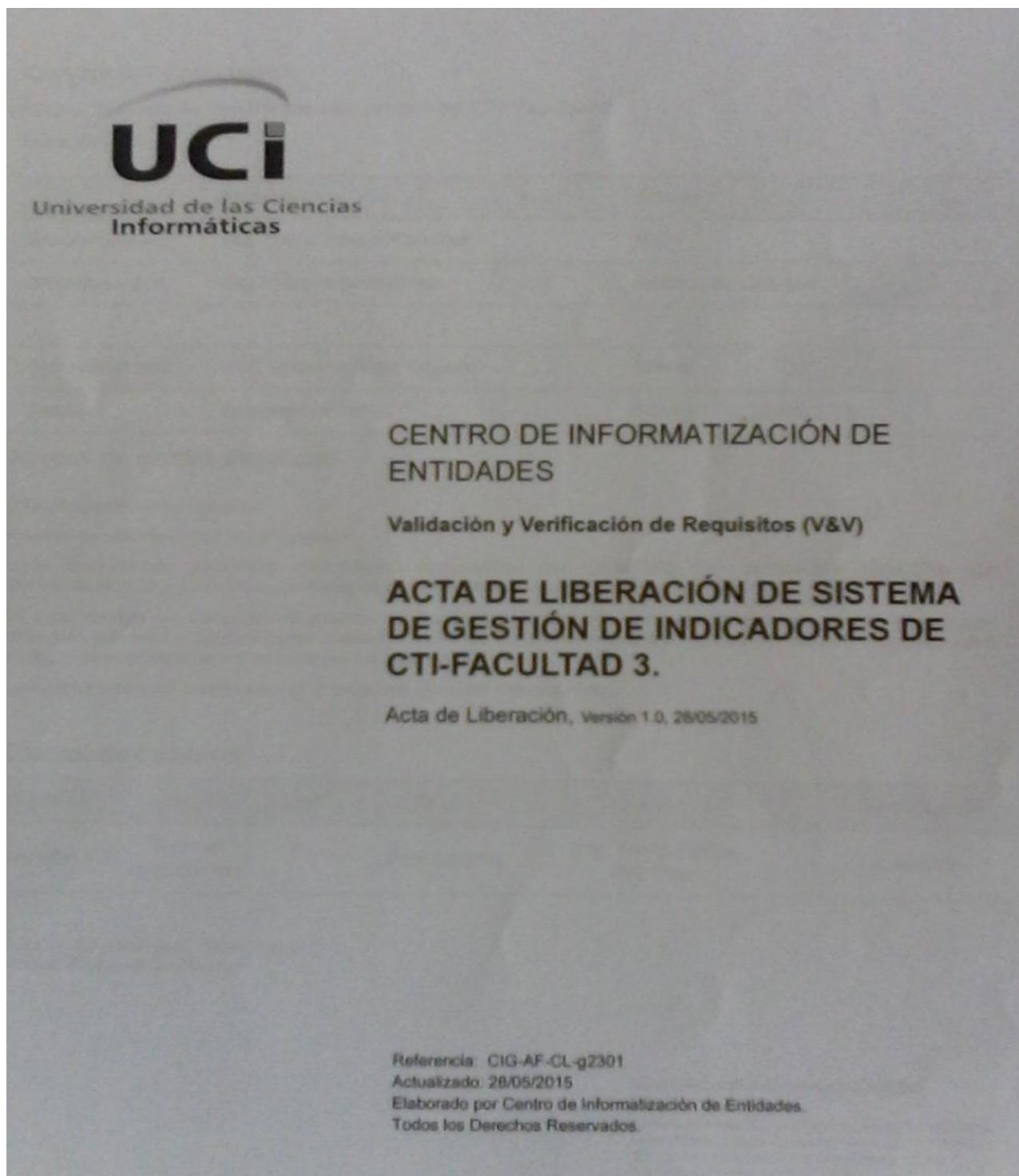


Figura 16. Diagrama de Modelo de datos del SGIF3

**Anexo 5. Acta de liberación**



**Figura 17. Acta de liberación**

Anexo 6. Acta de aceptación del cliente

**UCI** Universidad de las Ciencias Informáticas | **CENTRO DE INFORMATIZACIÓN DE ENTIDADES**  
Acta de aceptación del producto

En la Habana, 1ero de junio de 2015

De una parte, el Vice-Deanato de Investigación y Postgrado de la Facultad 3, representado por Yoan Martínez Márquez, quien a los fines y efectos derivados del presente documento se denominará como **PARTE CLIENTE**, y de otra Parte el equipo de desarrollo de la "Sistema para la gestión de los indicadores de Ciencia, Técnica e Innovación en la Facultad 3.", representado en este acto por Yasiel Torres Hernández y Rolando Ajete Valdés, que a los fines y efectos derivados del presente documento se denominarán como **PARTE COMERCIALIZADORA**.

Primero: Que en cumplimiento de los acuerdos, han sido efectuadas las actividades que se describen, **Las partes DECLARAN:**

CONSIDERANDO: Que se han efectuado las actividades siguientes:

1. Modelado del negocio del sistema.
2. Levantamiento de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Pruebas internas y de liberación.

CONSIDERANDO: Que las actividades realizadas han sido desarrolladas con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las partes**.

CONSIDERANDO: Que las actividades que se han ejecutado cumplen con los requisitos de la **PARTE CLIENTE**.

CONSIDERANDO: que la **PARTE COMERCIALIZADORA** ha entregado la documentación que avala la ejecución de este acto a la **PARTE CLIENTE**.

**POR TANTO:** **Las partes** acuerdan formalizar mediante el Acta, Aceptadas las actividades que han sido ejecutadas en esta fecha.

Y para que así conste, se extiende la presente Acta en dos (2) ejemplares, rubricados por **Las partes**.

Por la PARTE CLIENTE  
Yoan Martínez Márquez

Por la PARTE COMERCIALIZADORA  
Yasiel Torres Hernández      Rolando Ajete Valdés

Figura 18. Acta de aceptación del cliente

Anexo 7. Diseño de caso de prueba Adicionar capacitación recibida

Tabla 9. Diseño de caso de prueba Adicionar capacitación recibida

Escenario	Descripción	Tipo	Curso recibido	Institución que oferta	Evidencia	Respuesta del sistema	Flujo central
EP 1.1: Adicionar una nueva capacitación recibida introduciendo datos válidos.	Se introducen los datos de la capacitación recibida: Tipo Curso recibido Institución que oferta Evidencia	V	V	V	V	El sistema actualiza el Listado de Capacitaciones recibidas.	1- Se presiona el botón <b>Adicionar</b> . 2- Se insertan todos los datos. 3- Se presiona el botón <b>Adicionar</b> . 4- El sistema muestra el mensaje de información que ha sido creada correctamente la evidencia. 5- El sistema muestra un formulario con los datos de la capacitación recibida creada. 6- Se presiona el botón: <b>Ir al listado</b> . 7- El sistema muestra el Listado de las Capacitaciones recibidas actualizado.
		Participante en diplomado	Minería de procesos en la planificación	UCI	caprec.jpg		
EP 1.2: Adicionar una nueva capacitación recibida dejando campos requeridos en blanco.	Se introducen los datos de la capacitación recibida: Tipo Curso recibido Institución que oferta Evidencia	I	V	V	V	Se muestra el mensaje de error: "Por favor seleccione un elemento de la lista".	1- Se presiona el botón <b>Adicionar</b> . 2- Se introducen los datos dejando campos requeridos en blanco. 3- Se presiona el botón <b>Adicionar</b> . 4- El sistema muestra el mensaje de error correspondiente a cada campo, indicando que debe ser completado.
		Vacío	Minería de procesos en la planificación	UCI	Evidencia1.jpg		
EP 1.3: Cancelar	Se cancela la operación de adicionar una nueva capacitación recibida.	NA	NA	NA	NA	Se cancela la operación y el sistema retorna al Listado de Capacitaciones recibidas	1- Se presiona el botón <b>Adicionar</b> . 2- Se introducen, o no, datos. 3- Se presiona el botón <b>Cancelar</b> . 4- El sistema retorna al Listado de las Capacitaciones recibidas.