



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 5**

**CENTRO DE CONSULTORÍA Y DESARROLLO DE ARQUITECTURAS  
EMPRESARIALES**

**Tablero de Control para la Dirección de Posgrado de la  
Universidad de las Ciencias Informáticas**

Trabajo de Diploma para optar por el título de  
Ingeniera en Ciencias Informáticas

**Autora:** Claudia Seijo Pérez

**Tutores:** Dr.C. Arturo César Arias Orizondo

Ing. Luis Angel Carbonel Hidalgo

**La Habana, junio de 2016**

## **DECLARACIÓN JURADA DE AUTORÍA**

Declaro ser autora del presente trabajo de diploma y se reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los 5 días del mes de julio del año 2016.

Nombre de la autora

Claudia Seijo Pérez

---

Nombre del tutor principal

Dr.C. Arturo César Arias Orizondo

---

## AGRADECIMIENTOS

*A toda mi familia por su dedicación y apoyo.*

*A mis amigos de la UCI.*

*A la Universidad y a mis profesores por su dedicación.*

*A la dirección de posgrado de la universidad por haberme acogido como parte de su equipo durante todo el desarrollo de la tesis.*

*A mis tutores por su dedicación y confianza.*

*A mis vecinos de al lado que son parte de mi gran familia.*

## DEDICATORIA

*A mi madre y a mi abuelo Ramón que han sido mi fuerza en toda la trayectoria de mi carrera y de mi vida; quienes han estado a mi lado en todos los momentos difíciles, y de quienes recuerdo sus consejos y el amor que me brindaron.*

*A mi mami Magnolia que me ha cuidado desde que nací.*

## RESUMEN

Actualmente la evaluación del desempeño del proceso de formación posgraduada en la Universidad de las Ciencias Informáticas (UCI), se realiza mediante el análisis de un conjunto de indicadores consolidados en una hoja de datos en Microsoft Excel. Con ellos se generan gráficos que muestran el estado actual y el comportamiento histórico de los indicadores. Esta forma de registrar y consolidar los datos trae algunos inconvenientes, pues la información solo está disponible en una estación de trabajo, lo que dificulta el acceso para su actualización y consulta. Solo una persona puede actualizar la información, por lo que el registro de la misma no siempre se logra en el momento oportuno y los indicadores pueden estar desactualizados. Además, no es posible manejar la información que está relacionada, por lo que hay que introducir información redundante, lo que no propicia evitar inconsistencias en la misma. Estas insuficiencias dificultan la adecuada gestión de los indicadores, lo que afecta el apropiado control del proceso de formación posgraduada en la UCI. Para contribuir a solucionar la problemática anteriormente descrita, se desarrolló un tablero de control desplegable en la web. En su construcción se emplearon tecnologías tales como Spring, JSF (Java Server Faces), Primefaces, Hibernate y PostgreSQL y se utilizó el Proceso Unificado Ágil como metodología de desarrollo de software. Las pruebas al producto obtenido garantizaron la confiabilidad del mismo y por tanto la entrega de una solución útil para la Dirección de Posgrado de la UCI, la que con su empleo fortalece sus mecanismos de control.

**Palabras clave:** aplicación web, cuadro de mando integral, tablero de control

# CONTENIDOS

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 Marco teórico y referencial de la investigación .....</b>	<b>5</b>
Herramientas para el control de la gestión organizacional .....	5
Indicadores de gestión .....	6
Cuadro de Mando Integral .....	7
Tablero de control.....	9
Diferencias entre Cuadro de Mando Integral y tablero de control.....	12
Tecnologías para el desarrollo del tablero de control.....	13
Apache Tomcat.....	13
JSF (Java Server Faces) .....	14
Primefaces.....	15
Spring .....	16
Hibernate .....	17
PostgreSQL .....	18
Metodología para el desarrollo del tablero de control .....	19
Herramientas para el desarrollo del tablero de control.....	20
Conclusiones del capítulo.....	20
<b>CAPÍTULO 2: Planificación, diseño y desarrollo del sistema .....</b>	<b>21</b>
Descripción general del proceso a informatizar .....	21
Indicadores que maneja el tablero de control .....	22
Sistema informático propuesto .....	23
Funcionalidades del sistema.....	23
Historias de usuarios .....	25
Estimación de esfuerzo por Historias de Usuario.....	37
Plan de iteraciones .....	38
Plan de entrega.....	38

Modelo de datos.....	39
Modelo físico de datos .....	39
Diccionario de datos .....	39
Arquitectura de la aplicación.....	41
Propiedades del sistema.....	41
Estilo arquitectónico empleado .....	42
Estructura de paquetes de implementación .....	43
Patrones de diseño empleados.....	46
Diagrama de clases .....	49
Conclusiones del capítulo.....	50
<b>CAPÍTULO 3 Evaluación de los resultados .....</b>	<b>51</b>
Pruebas de software.....	51
Pruebas de caja blanca.....	51
Pruebas de caja negra .....	57
Pruebas de aceptación .....	62
Aplicación obtenida .....	68
Evaluación de los resultados alcanzados .....	69
Conclusiones del capítulo.....	69
<b>CONCLUSIONES .....</b>	<b>70</b>
<b>RECOMENDACIONES.....</b>	<b>71</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>
<b>ANEXOS.....</b>	<b>75</b>
Anexo 1. Aval de la Dirección de Posgrado de la UCI .....	75

## Índice de figuras

Figura 1 Perspectivas del Cuadro de Mando Integral .....	8
Figura 2 Ejemplo de implementación mediante herramienta informática de la perspectiva financiera de un Cuadro de Mando Integral.....	8
Figura 3 Ejemplo de tablero de control.....	11
Figura 4 Tecnologías fundamentales empleadas en la implementación del tablero de control....	13
Figura 5 Diagrama de objetos de negocio.....	21
Figure 6 Área para los gráficos que conforman el tablero de control.....	35
Figura 7 Modelo físico de datos del sistema.....	39
Figura 8 Arquitectura de tres capas del sistema.....	42
Figura 9 Estructura de paquetes de implementación.....	44
Figura 10 Estructura de paquetes vista desde la herramienta de desarrollo.....	45
Figura 11 Ejemplo de empleo del patrón controlador .....	46
Figura 12 Ejemplo del uso del patrón experto .....	47
Figura 13 Ejemplo del uso del patrón controlador .....	47
Figura 14 Ejemplo del uso del patrón singleton .....	48
Figura 15 Diagrama de clases del gestionar Área .....	49
Figura 16 Flujo del camino básico para el código numerado.....	53
Figura 17 Resultados de las pruebas de aceptación para los requisitos seleccionados a modo de ejemplo.....	67
Figura 18. Ejemplo de interfaz de actualización de datos .....	68
Figura 19. Ejemplos de gráficos que conforman el tablero de control.....	68

## Índice de tablas

Tabla 1. Perspectivas que conforman un Cuadro de Mando Integral.....	7
Tabla 2. Indicadores que maneja el tablero de control.....	22
Tabla 3. Historia de usuario: gestionar doctores de la UCI.....	27
Tabla 4. Historia de usuario: gestionar institución.....	28
Tabla 5. Historia de usuario: gestionar programa de formación.....	29
Tabla 6. Historia de usuario: gestionar aspirantes.....	30
Tabla 7. Historia de usuario: mostrar compromisos de defensas.....	31
Tabla 8. Historia de usuario: gestionar defensas.....	32
Tabla 9. Historia de usuario: mostrar información consolidada.....	33
Tabla 10. Historia de usuario: mostrar información histórica.....	34
Tabla 11. Historia de usuario: autenticar usuario.....	36
Tabla 12. Estimación del esfuerzo por historia de usuario.....	37
Tabla 13. Planificación de la implementación del sistema.....	38
Tabla 14. Plan de entrega.....	38
Tabla 15. Caminos básicos para ejemplo de prueba de caja blanca.....	54
Tabla 16. Caso de prueba de ejemplo para mostrar estrategia de pruebas de caja negra.....	58
Tabla 17. Resultados de las pruebas de caja negra realizadas a toda la aplicación.....	61
Tabla 18. Iteración 1, caso de prueba de aceptación para el RF6 (maestrías).....	63
Tabla 19. Iteración 2, caso de prueba de aceptación para el RF6 (maestrías).....	63
Tabla 20. Iteración 3, caso de prueba de aceptación para el RF6 (maestrías).....	64
Tabla 21. Iteración 1, caso de prueba de aceptación para el RF6 (doctorados).....	64
Tabla 22. Iteración 2, caso de prueba de aceptación para el RF6 (doctorados).....	65
Tabla 23. Iteración 3, caso de prueba de aceptación para el RF6 (doctorados).....	65
Tabla 24. Iteración 1, caso de prueba de aceptación para el RF9.....	66
Tabla 25. Iteración 2, caso de prueba de aceptación para el RF9.....	66

## **INTRODUCCIÓN**

Las organizaciones modernas requieren manejar adecuadamente información relevante de sus procesos, para monitorearlos y tomar decisiones objetivas en función de mejorar el desempeño de los mismos. El uso de las Tecnologías de la Información y las Comunicaciones (TIC) ha contribuido significativamente a este propósito, en organizaciones donde hardware, software, procesos, datos y personas conforman un sistema complejo en el que se genera, procesa y distribuye información para tomar decisiones en todos los niveles de la organización y generar conocimiento (Topi, Valacich et al. 2010).

La adecuada presentación de la información es esencial para una efectiva toma de decisiones basada en datos y contribuir así a una evaluación más objetiva de los procesos de una organización. La mejora continua de estos requiere monitorear de manera permanente los resultados que generan, a modo de identificar aquellos procesos que requieren mayor atención por no alcanzar los niveles de desempeño esperados.

Existen varias herramientas para evaluar el desempeño de los procesos de una organización. El Tablero de Control (TC) es una de ellas y es un marco conceptual que ayuda a convertir la visión y la estrategia de una empresa, en mediciones y objetivos tangibles (Cruz, González et al. 2010). El TC es un sistema de comunicación, información y aprendizaje. La elaboración de un tablero de control debe originarse en la visión y estrategia de la empresa, para luego entrar a definir los factores críticos necesarios para poder alcanzar el éxito empresarial. Con el empleo de un TC se ofrece mayor coherencia a los indicadores de gestión de la organización. Ello permite evaluar la actuación de la organización en su cumplimiento con el rumbo estratégico trazado (Ramírez 2009).

Actualmente en la Universidad de las Ciencias Informáticas (UCI) son pocos los procesos claves (actividad de desarrollo-producción, formación posgraduada y otros) evaluados bajo este enfoque. Ello dificulta ofrecer una evaluación más integral, objetiva y coherente de estos procesos sustantivos de la universidad.

El proceso de formación posgraduada no escapa de esta realidad. Si bien el mismo tiene bien definidos los indicadores más importantes para evaluar dicho proceso, la

información a partir de la cual se calculan los indicadores se maneja de manera manual y se consolida en una hoja de datos en Microsoft Excel. A partir de los datos consolidados se generan un conjunto de gráficos que muestran el estado actual y el comportamiento histórico de los indicadores, para evaluar y controlar el desempeño del proceso de formación posgraduada en la UCI.

Esta forma de registrar y consolidar los datos incide negativamente en los siguientes atributos de la información:

- Disponibilidad: la información solo está disponible en una estación de trabajo, lo que dificulta el acceso para su actualización y consulta.
- Oportunidad: solo una persona puede actualizar la información, por lo que el registro de la misma no siempre se logra en el momento oportuno y los indicadores pueden estar desactualizados.
- Consistencia: no es posible manejar la información que está relacionada, por lo que hay que introducir información redundante, lo que no propicia evitar inconsistencias en la misma.

Las insuficiencias anteriores dificultan la adecuada gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada en la UCI, lo que afecta el apropiado control de dicho proceso.

Por lo antes expuesto se identificó el siguiente **problema**: ¿cómo mejorar la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada en la UCI, para contribuir a perfeccionar el control de dicho proceso?

En función de solucionar el problema identificado se formuló la siguiente **idea a defender**: si se desarrolla para la Dirección de Posgrado de la UCI un tablero de control desplegable en la web, se mejora la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada, lo que contribuye a perfeccionar el control de dicho proceso.

El **objeto de estudio** se centró en el proceso de evaluación del desempeño organizacional mediante tableros de control. Dentro de este se fijó como **campo de acción** la informatización del tablero de control.

Para ofrecer una solución al problema planteado se determinó como **objetivo general** de este trabajo: desarrollar para la Dirección de Posgrado de la UCI, un tablero de control desplegable en la web, para mejorar la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada y contribuir a perfeccionar el control de dicho proceso.

El objetivo general fue desglosado en los siguientes **objetivos específicos**:

1. Construir el marco teórico y referencial de la investigación, relacionado con tableros de control y las herramientas informáticas apropiadas para su desarrollo.
2. Diseñar la solución informática consistente en un tablero de control desplegable en la web.
3. Implementar la solución informática propuesta.
4. Evaluar la calidad y aceptación del tablero de control desarrollado.

En correspondencia con el objetivo planteado, se empleó el Proceso Unificado Ágil (AUP por sus siglas en inglés) para conducir el diseño y desarrollo de la solución informática propuesta. Bajo este enfoque fueron articulados y aplicados con rigor científico los siguientes métodos y técnicas.

**Métodos teóricos:**

- **Analítico - sintético:** para descomponer el problema de investigación en elementos, profundizar en su estudio y luego sintetizarlos en la solución propuesta.
- **Histórico – lógico y dialéctico:** con el fin de realizar un estudio crítico sobre la evolución de los diferentes enfoques relativos a tableros de control y las tecnologías empleadas para su implementación.
- **Modelación:** en el diseño de la arquitectura del sistema y de cada una de las vistas que la conforman.
- **Sistémico:** para integrar armónicamente los componentes diseñados en la solución propuesta.

**Métodos empíricos:**

- **Análisis documental:** en la revisión de la literatura especializada, tanto académica como empresarial, para extraer la información necesaria que permitió realizar el proceso de investigación.
- **Tormenta de ideas:** para refinar el diseño de la solución propuesta.
- **Pruebas de calidad:** para evaluar la calidad de la solución desarrollada.
- **Controles de aplicabilidad:** para validar con los profesionales la aplicabilidad de la solución desarrollada.

El **aporte** más importante de este trabajo consiste en el desarrollo de un tablero de control desplegable en la web, que mejora la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada de la UCI y contribuye a perfeccionar el control de dicho proceso.

El presente trabajo de diploma está compuesto por introducción, tres capítulos, conclusiones, recomendaciones y bibliografía. Se adjuntan anexos que contribuyen a su comprensión.

- **Capítulo 1:** dedicado al marco teórico y referencial de la investigación, relacionado con el tablero de control y las herramientas informáticas apropiadas para su desarrollo.
- **Capítulo 2:** aborda la concepción de la solución informática propuesta en términos de su diseño funcional y arquitectónico.
- **Capítulo 3:** contiene el diseño y el resultado de las pruebas de calidad aplicadas al sistema informático desarrollado.

## CAPÍTULO 1 Marco teórico y referencial de la investigación

En este capítulo se abordan técnicas gerenciales existentes que facilitan el control de la gestión de una organización, así como la forma en que son implementadas mediante herramientas informáticas. Se determinan además las tecnologías que pueden emplearse específicamente para el desarrollo de un tablero de control, una de las técnicas gerenciales de control analizadas.

### Herramientas para el control de la gestión organizacional

El control de gestión es un proceso que sirve para guiar la gestión de la organización hacia el cumplimiento de los objetivos y un instrumento para evaluarla. Los objetivos son la categoría rectora. El proceso de toma de decisiones está orientado a alcanzar los objetivos marcados y luego estos son el patrón para evaluar a la gestión, o sea, el grado en que los resultados de la gestión se acercan a los objetivos previstos (Amaya 2005).

El control de gestión ha ido evolucionando con el tiempo a medida que la problemática organizacional planteaba nuevas necesidades y exigencias. Se inicia y desarrolla en Estados Unidos a partir de 1930 bajo diversas formas (control de costos, control presupuestario, etcétera). Luego, en la segunda guerra mundial, surgen nuevas técnicas (investigación de operaciones, planificación a mediano y largo plazo, entre otras). Asimismo, el empleo del ordenador electrónico aporta nuevas posibilidades al desarrollo de la función hasta el surgimiento del *controlling* a finales de la década de los años 80.

Más tarde, a inicios de los años 90, se da a conocer el **Cuadro de Mando Integral** o *Balanced Scorecard*, como una potente herramienta de gestión que traduce la estrategia de la empresa en un conjunto coherente de indicadores; así como el **tablero de control**, cuyo objetivo y utilidad básica es diagnosticar adecuadamente una situación mediante un conjunto de indicadores cuyo seguimiento y evaluación periódica ofrece un mayor conocimiento de la situación de la organización apoyándose en tecnologías informáticas (Cerde and Chacur 2006).

Teniendo en cuenta que ambas herramientas están basadas en el empleo de indicadores, a continuación se profundiza en este concepto.

### ***Indicadores de gestión***

Todo se puede medir y por tanto todo se puede controlar, allí radica el éxito de cualquier operación porque lo que no se mide, no se puede administrar.

Uno de los factores determinantes para que todo proceso se lleve a cabo con éxito, es implementar un sistema adecuado de indicadores para medir la gestión de los mismos. Para que se puedan implementar indicadores en posiciones estratégicas que reflejen un resultado óptimo en el mediano y largo plazo, se requiere un buen sistema de información que posibilite medir las diferentes etapas del proceso bajo control (Matiz 2011).

Un indicador se define como la relación entre variables cuantitativas o cualitativas, que permite observar la situación y las tendencias de cambio generadas en el objeto o fenómeno observado, respecto a objetivos y metas previstos e influencias esperadas. Se define como un número que sirve para informar continuamente sobre el funcionamiento o comportamiento de una actividad en una organización. Los indicadores pueden ser valores, unidades, índices, series estadísticas, etc. (Taylor, Benavides et al. 2012).

Los objetivos principales de los indicadores de gestión son:

- Establecer un sistema de instrumentos que permita en forma rápida y proactiva, administrar la organización y hacer posible la comparación de los resultados con las metas propuestas y otras entidades.
- Definir parámetros que permitan que el diseño de los objetivos, los planes y las metas sean en condición y tiempo real para controlar las operaciones diarias que se realizan dentro de la organización.
- Crear mecanismos de detección de fallas que garanticen la posibilidad de llevar a cabo acciones concretas que permitan obtener soluciones reales y de aplicación inmediata.

Además de las definiciones estratégicas, la medición de indicadores de gestión está condicionada por la capacidad de la institución de generar la información necesaria y de elaborar los indicadores, con niveles adecuados de calidad, certeza y confiabilidad (Matiz 2011).

A continuación se expone el estudio realizado sobre dos de las herramientas de control de la gestión tradicionalmente empleadas en las organizaciones, a modo de identificar la más apropiada en la solución de la problemática planteada en esta investigación.

### ***Cuadro de Mando Integral***

El Cuadro de Mando Integral (CMI) es una herramienta de gestión que facilita la toma de decisiones y que recoge un conjunto coherente de indicadores que proporciona a la alta dirección y a las funciones responsables, una visión comprensible del negocio o de su área de responsabilidad. La información aportada por el CMI permite enfocar y alinear los equipos directivos, las unidades de negocio, los recursos y los procesos con las estrategias de la organización (Nogueira, Medina et al. 2004).

El CMI ayuda a transformar la estrategia en acción, pues permite alinear los esfuerzos del cambio en torno a la estrategia de la organización, controlar y predecir los resultados que transforman la información de la organización en conocimientos y evaluar los objetivos en el sistema administrativo a través de la medición del desempeño (Orizondo 2015).

El CMI no sustituye a los métodos de gestión existentes, ni elimina las medidas e indicadores actuales, sino que les da a estos una mayor coherencia y los ordena jerárquicamente, según el modelo de relaciones causa-efecto y el árbol de indicadores, que permiten evaluar la actuación empresarial en su cumplimiento con el rumbo estratégico trazado (Kaplan 2010). Las interrelaciones entre los factores clave que permiten la creación de valor futuro para la organización se realiza a través de las cuatro perspectivas del CMI reflejadas en la siguiente tabla.

*Tabla 1. Perspectivas que conforman un Cuadro de Mando Integral*

<b>Perspectiva</b>	<b>Pregunta</b>
Financiera	¿Cómo nos ve el organismo superior?
Clientes	¿Cómo nos ven nuestros clientes?
Procesos	¿En cuáles procesos debemos ser excelentes?
Desarrollo	¿Cuáles son los recursos clave para mejorar?

En la siguiente figura se aprecian las relaciones entre las perspectivas del Cuadro de Mando Integral.



Figura 6 Perspectivas del Cuadro de Mando Integral (Peñalver 2016)

En la siguiente figura se muestra un ejemplo de implementación mediante una herramienta informática de la perspectiva financiera de un Cuadro de Mando Integral.



Figura 7 Ejemplo de implementación mediante herramienta informática de la perspectiva financiera de un Cuadro de Mando Integral (Páez 2016)

### ***Tablero de control***

El tablero de control es una herramienta del campo de la administración de empresas, aplicable a cualquier organización y nivel de la misma, cuyo objetivo y utilidad básica es diagnosticar adecuadamente una situación. Se lo define como el conjunto de indicadores cuyo seguimiento y evaluación periódica permitirá contar con un mayor conocimiento de la situación de su empresa o sector apoyándose en nuevas tecnologías informáticas (Cortés 2010).

El diagnóstico y monitoreo permanente de determinados indicadores e información ha sido y es la base para mantener un buen control de situación en muchas de las disciplinas de la vida. Como ejemplo de estos podemos señalar a la medicina, basada en mediciones para el diagnóstico de la salud de los pacientes, a la aviación, cuyos indicadores del tablero de control sintetiza la información del avión y del entorno para evitar sorpresas y permite a los pilotos dirigir el avión a buen puerto; el tablero de un sistema eléctrico o de una represa son otros ejemplos. En todos estos casos el tablero permite a través del color de las luces y alarmas, ser el disparador para la toma de decisiones. En todos estos ejemplos es fundamental definir los indicadores a monitorear (Cruz, González et al. 2010).

La organización es sujeta de parametrización para facilitar el diagnóstico y la toma de decisiones. Si bien hay indicadores genéricos para todas las organizaciones, especialmente en áreas como las económicas financieras, cada organización o sector requiere definiciones a la medida de sus propios parámetros y definir quién y cómo va a monitorear esa información (Matiz 2011).

El método de trabajo comienza identificando como áreas clave a aquellos temas relevantes a monitorear y cuyo fracaso permanente impediría la continuidad y el progreso de la empresa o sector dentro de un entorno competitivo, aun cuando el resultado de todas las demás áreas fuera bueno. El método fue creado inicialmente para ser aplicado a la empresa en su conjunto, con una visión global, pero demostró ser aplicable también a un sector o función dentro de la empresa.

Los indicadores claves son los datos, índices, mediciones o ratios que generan información de la situación de cada área clave. A partir de definir áreas e indicadores y con el apoyo de las tecnologías informáticas, se puede conformar una potente

herramienta de diagnóstico. El tablero propiamente dicho contendrá entonces las áreas e indicadores que sintetizan un diagnóstico completo de situación, por lo cual podría ser llevado en papel, pero su uso se potencia mucho más utilizando modernas herramientas informáticas del mundo del *Business Intelligence* (Caralt and Díaz 2011). De esta forma se puede acceder a la información relevante para completar el diagnóstico e implementar acciones correctivas.

En todos los tableros después de la definición de los temas e indicadores clave se deberá definir:

- **Reporte o pantalla:** muestra la información clave para el diagnóstico, con su formato y configuración. La arquitectura del reporte debe reflejar la necesidad de información y cultura de uso de la empresa y es recomendable que sea estándar a distintos niveles de la organización.
- **Período del indicador:** día, mes, acumulado del ejercicio, proyectado a fin del período fiscal o para los próximos meses, etc.
- **Apertura:** forma en la cual se podrá abrir y clasificar la información para acceder a sucesivos niveles de desagregación, en tablas o matrices por producto, sector geográfico, dimensión de análisis, etc.
- **Frecuencia de actualización:** tiempo que transcurre entre distintas actualizaciones de los datos. Ejemplo: on-line, diaria, semanal, mensual.
- **Referencia:** base sobre la cual se desean calcular las desviaciones. Puede ser un estándar, la historia, el mes anterior, el promedio de los últimos doce meses, el presupuesto inicial o revisado, un objetivo o una meta, etc.
- **Parámetro de alarma:** niveles por encima o por debajo de los cuales el indicador es preocupante, por ejemplo, más o menos 5% sobre una base de referencia.
- **Gráfico:** la mejor forma de representar gráficamente la realidad que nos muestra la información. Por ejemplo, gráficos de pastel, barras, líneas, etc.
- **Responsable de monitoreo:** es quien debe informar al nivel superior cuando haya en el indicador alguna sorpresa desagradable.
- **Avisos automáticos:** se emiten para informar problemas de acuerdo a parámetros incluidos en el sistema.

El tablero tiene determinado alcance que limita, pero a su vez refuerza su utilidad:

- **Refleja solo información cuantificable:** el tablero tiene un alcance limitado como para poder recoger toda la información informal y cualitativa.
- **Evalúa situaciones, no responsables:** permite saber cómo está la empresa o un sector, pero no identifica directamente quién es el responsable de que ocurra.
- **No focaliza totalmente la acción directiva:** en principio establece qué mirar para diagnosticar y generar un buen ambiente de análisis.
- **No reemplaza el juicio directivo:** siempre habrá que aplicar el sentido común para emitir juicio a partir de la información.
- **No pretende reflejar totalmente la estrategia:** no es una herramienta ideal para implementar una estrategia global. Para ello habrá que seleccionar y priorizar aquellos indicadores del tablero en los que habrá que ser excelentes.

En la siguiente figura se ofrece un ejemplo de implementación de un tablero de control.



Figura 8 Ejemplo de tablero de control (Quántico 2016)

### ***Diferencias entre Cuadro de Mando Integral y tablero de control***

De una manera errada algunas personas llaman Cuadro de Mando Integral, *Balanced Scorecard* o tablero de comando a cualquier modelo que contenga: un mapa estratégico, una tabla con indicadores, objetivos y planes de acción. Y los llaman así sin tener en cuenta su contenido. Es decir, se confunde un Cuadro de Mando Integral con cualquier sistema de medición, sea o no estratégico. Y como resultado de esta confusión, en la mente de algunos profesionales no hay distinciones entre lo que es o debería ser un Cuadro de Mando Integral o tablero de comando y lo que es o debería ser un cuadro de mando o tablero de control (Caralt and Díaz 2011).

El Cuadro de Mando Integral es un sistema y una herramienta por medio de la cual una organización logra ejecutar su plan estratégico trasladando el mismo a la acción. Su importancia radica en que –por medio de sus indicadores- muestrea de manera continua cuándo la organización y sus colaboradores alcanzan las metas del plan estratégico.

El tablero de control, en cambio, es una herramienta operativa útil para controlar lo que pasó en un área o departamento. Sus indicadores están focalizados a procesos y no al avance del plan estratégico de la organización. Es un instrumento útil para medir el avance de los resultados operativos en una organización, pero es insuficiente para lograr la ejecución de un plan estratégico. Por eso, denominar a ambos de la misma manera es incorrecto y solo genera una mala reputación al Cuadro Mando Integral.

Los elementos que componen un Cuadro de Mando Integral son: un mapa estratégico, formado por objetivos que provienen de la visión de la organización, que es desdoblada en objetivos para asegurar su cumplimiento y el FODA, que analiza las fortalezas, oportunidades, debilidades y amenazas de la organización.

Un tablero de control, en cambio, es un cuadro operacional, motivo por el cual sus objetivos no provienen de la estrategia sino de los diferentes planes y programas que tiene que cumplir cada área o departamento. Los elementos que componen un tablero de control son: objetivos operativos del área o departamento e indicadores de gestión que muestran lo que pasó durante un período.

Por todo lo anteriormente analizado, para evaluar el proceso de formación posgraduada en la UCI, el tablero de control constituye una herramienta efectiva.

## Tecnologías para el desarrollo del tablero de control

Para desarrollar el tablero de control es necesario la selección de las herramientas y tecnologías apropiadas, teniendo en cuenta que este se implementa como un conjunto de componentes desplegados en un portal web. En la siguiente figura se identifican las principales tecnologías a emplear y se describen a continuación.

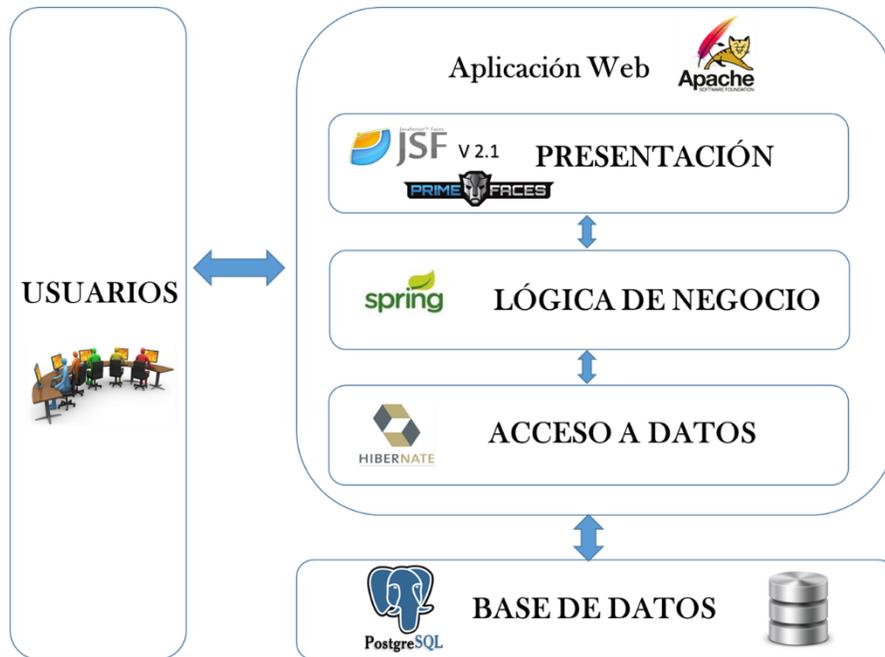


Figura 9 Tecnologías fundamentales empleadas en la implementación del tablero de control (elaboración propia)

### Apache Tomcat

Tomcat es un contenedor web basado en el lenguaje Java que actúa como motor de servlets y JSPs. Se ha convertido en la implementación de referencia para las especificaciones de servlets y JSPs. Fue seleccionado como la implementación de referencia de contenedores de componentes web Sun (JSPs/Servlets) (Apache Software Foundation 2016).

Tomcat es desarrollado y actualizado por miembros de la *Apache Software Foundation* y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software License*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones estables

más recientes son las 8.x que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Tomcat es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

### ***JSF (Java Server Faces)***

Es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador). Al igual que Struts, JSF pretende normalizar y estandarizar el desarrollo de aplicaciones web. Hay que tener en cuenta JSF es posterior a Struts, y por lo tanto se ha nutrido de la experiencia de este, mejorando algunas sus deficiencias (Keen, Coutinho et al. 2012).

Los principales componentes de la tecnología Java Server Faces son:

- Una API y una implementación de referencia para:
- Representar componentes de interfaz de usuario y manejar su estado.
- Manejar eventos, validar en el lado del servidor y convertir datos.
- Definir la navegación entre páginas.
- Soportar internacionalización y accesibilidad
- Proporcionar extensibilidad para todas estas características.
- Una librería de etiquetas Java Server Pages (JSP) personalizadas para dibujar.

Una de las ventajas de que JSF sea una especificación estándar es que pueden encontrarse implementaciones de distintos fabricantes. Esto permite no vincularse

exclusivamente con un proveedor concreto, y poder seleccionar el que más interese en cada caso según el número de componentes que suministra, el rendimiento de éstos, soporte proporcionado, precio, política de evolución, etc.

Una de las grandes ventajas de la tecnología Java Server Faces es que ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones web construidas con tecnología JSP conseguían parcialmente esta separación. Sin embargo, una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos de los componentes o manejar elementos UI (*User Interface*) como objetos con estado en el servidor. Separar la lógica de negocio de la presentación también permite que cada miembro del equipo de desarrollo de la aplicación web se centre en su parte asignada del proceso diseño, y proporciona un modelo sencillo de programación para enlazar todas las piezas (Java Code Geeks 2015).

Otro objetivo importante de la tecnología Java Server Faces es mejorar los conceptos familiares de componente-UI y capa-web, sin limitarnos a una tecnología de script particular o un lenguaje de marcas.

### ***Primefaces***

Primefaces es una librería de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web (Çivici 2014). Primefaces está bajo la licencia de Apache License v2. Una de las ventajas de utilizar Primefaces es que permite la integración con otros componentes como por ejemplo Richfaces.

En cuanto a la experiencia de los usuarios finales los componentes de Primefaces son amigables al usuario, además cuentan con un diseño innovador. Pero en lo que respecta al programador, es que sus desarrolladores no respetan un principio básico del desarrollo de componentes: la compatibilidad hacia atrás, es decir, un componente de una nueva versión de Primefaces por lo general no es compatible al 100% con una aplicación desarrollada con la versión previa a la misma.

En el desarrollo de la aplicación se utilizó la versión 5.3.

## **Spring**

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control (IoC), de código abierto para la plataforma Java.2 (Winterfeldt 2013).

La primera versión fue escrita por Rod Johnson, quien lo lanzó junto a la publicación de su libro *Expert One-on-One J2EE Design and Development* (Johnson 2002). El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003. El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005. La versión 1.2.6 de Spring Framework obtuvo reconocimientos *Jolt Awards* y *Jax Innovation Awards* en 2006. Spring Framework 2.0 fue lanzado en 2006, la versión 2.5 en noviembre de 2007, Spring 3.0 en diciembre de 2009 y Spring 3.1 dos años más tarde. El inicio del desarrollo de la versión 4.0 fue anunciado en enero de 2013. La versión actual es 4.1.1 (Johnson, Hoeller et al. 2015). En el sistema se utiliza Spring 4.1.0.

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de programación en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (*Enterprise Java Bean*).

Spring Framework comprende diversos módulos que proveen un rango de servicios (Johnson, Hoeller et al. 2015):

- **Contenedor de inversión de control:** permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java. Se lleva a cabo principalmente a través de la inyección de dependencias.
- **Programación orientada a aspectos:** habilita la implementación de rutinas transversales.
- **Acceso a datos:** se trabaja con RDBMS en la plataforma java, usando Java Database Connectivity y herramientas de mapeo objeto relacional con bases de datos NoSQL.

- **Gestión de transacciones:** unifica distintas APIs de gestión y coordina las transacciones para los objetos Java.
- **Modelo vista-controlador:** implementa el patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **Framework de acceso remoto:** permite la importación y exportación estilo RPC, de objetos Java a través de redes que soporten RMI, CORBA y protocolos basados en HTTP incluyendo servicios web (SOAP).
- **Convención sobre configuración:** el módulo Spring Roo ofrece una solución rápida para el desarrollo de aplicaciones basadas en Spring Framework, privilegiando la simplicidad sin perder flexibilidad.
- **Procesamiento por lotes:** un framework para procesamiento de mucho volumen que como características incluye funciones de registro/trazado, manejo de transacciones, estadísticas de procesamiento de tareas, reinicio de tareas, y manejo de recursos.
- **Autenticación y autorización:** procesos de seguridad configurables que soportan un rango de estándares, protocolos, herramientas y prácticas a través del subproyecto Spring Security (antiguamente Acegi).
- **Administración remota:** configuración de visibilidad y gestión de objetos Java para la configuración local o remota vía JMX.
- **Mensajes:** registro configurable de objetos receptores de mensajes JMS, una mejora del envío de mensajes sobre las API JMS estándar.
- **Testing:** soporte de clases para desarrollo de unidades de prueba e integración.

### ***Hibernate***

Cuando se trabaja bajo el paradigma “Orientado a objetos” y a la vez se utiliza el de sistemas de bases de datos relacionales, los desarrolladores afrontan las desavenencias de estos dos enfoques de trabajo. Un ORM es un framework que propone una nueva forma de modelar los datos, permitiendo solucionar la diferencia que existe entre estos dos paradigmas (Bauer and King 2007).

Se decide utilizar Hibernate en su versión 4.3.7 pues es una herramienta ORM de código abierto muy madura y completa. Agiliza la relación entre la aplicación y la base de datos. Tiene como objetivo facilitar el mapeo de atributos entre una base de datos relacional y un modelo de objetos, a través de archivos (XML) o con anotaciones en las entidades (Hennebrueder 2013).

### **PostgreSQL**

PostgreSQL es un sistema gestor de bases de datos relacionales de código abierto. Brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación, entre el que se encuentra Java (The PostgreSQL Global Development Group 2016).

Posee características significativas el motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (*constraints*) y los disparadores (*triggers*). El código fuente se encuentra disponible para todos sin costo alguno. Controla la integridad referencial y tiene interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG, PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Por todas estas características se decide la utilización de su versión 9.3 en la solución planteada.

## **Metodología para el desarrollo del tablero de control**

Las metodologías para el desarrollo del software imponen un proceso disciplinado con el fin de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema.

El desarrollo de la solución fue guiado por la metodología Proceso Unificado Ágil (AUP por sus siglas en inglés) (Edeki 2013). La misma ha sido designada por la universidad para la actividad productiva, específicamente para los proyectos de desarrollo de pequeña escala.

Es una versión simplificada del Proceso Unificado de Rational (RUP por sus siglas en inglés). AUP describe de una manera simple y fácil de entender, la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Ha sido adaptada a las necesidades de desarrollo de software en la universidad. En la variación efectuada se proponen 3 fases para guiar el proceso de desarrollo de los proyectos en la universidad:

- **Inicio:** durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
- **Cierre:** en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

## Herramientas para el desarrollo del tablero de control

Para el desarrollo del tablero de control fueron seleccionadas las siguientes herramientas:

- **Visual Paradigm para UML 8.0:** herramienta multiplataforma diseñada para soportar el ciclo de vida completo del proceso de desarrollo del software, a través de la representación de todo tipo de diagramas. Con capacidad de ingeniería directa e inversa enfocada al negocio, genera un software de mayor calidad. Emplea **UML 2.0** como lenguaje de modelado estándar para el desarrollo de sistemas de software a partir del paradigma de Programación Orientada a Objetos (POO), permitiendo especificar, visualizar, construir y documentar todos los elementos que conforman dichos sistemas.
- **Eclipse STS 3.6.1:** *Spring Tool Suite* (STS) es una herramienta libre desarrollada por *Spring Source*, basada actualmente en Eclipse 3.x, para construir aplicaciones empresariales enriquecidas por los proyectos de *Spring Source*. Una de las principales razones para emplear STS es que incluye herramientas para el desarrollo en lenguaje Java para *Spring*.

## Conclusiones del capítulo

Luego de la revisión y análisis de la literatura especializada se arribaron a las siguientes conclusiones:

- A través del control de la gestión mediante tableros de control y/o cuadros de mando integral, la organización configura sus decisiones estratégicas, evalúa la implementación de las decisiones de forma operativa, verifica el cumplimiento de los procedimientos y procesos y realiza los análisis sobre la base de un sistema informativo eficiente, oportuno y eficaz, que permita la corrección de las desviaciones y su seguimiento.
- Se identificó el tablero de control como una solución factible para perfeccionar el control del proceso de formación posgraduada en la UCI, pues este facilita la gestión de los indicadores que evalúan su desempeño.
- Mediante el empleo de las tecnologías, herramientas y metodología seleccionada, es viable el desarrollo de un tablero de control desplegable en un portal web.

## CAPÍTULO 2: Planificación, diseño y desarrollo del sistema

En el capítulo se realiza el análisis y diseño detallado del sistema. Además, se planifica el desarrollo del software bajo los principios de la metodología seleccionada (AUP).

### Descripción general del proceso a informatizar

La Dirección de Posgrado de la UCI es la encargada de dirigir metodológicamente la formación académica y la superación profesional de los profesores y especialistas de la UCI. Para ello necesita mantener un constante monitoreo de los estudiantes matriculados, tanto en programas de maestrías y doctorados de la propia universidad como de otras instituciones de educación superior nacionales y extranjeras.

De cada aspirante se conoce el programa de posgrado que cursa, la fecha planificada de defensa de tesis (con lo que se conforma el plan anual de defensas), así como la fecha real de defensa (con lo que se verifica el cumplimiento de los compromisos).

Con esta información es posible evaluar el estado actual y el comportamiento histórico de la formación de másteres y doctores de la UCI, identificar tendencias y en función de ellas tomar decisiones que ayuden a mejorar el proceso.

Los principales conceptos de negocio identificados y descritos anteriormente, son resumidos en el siguiente diagrama de objetos de negocio.

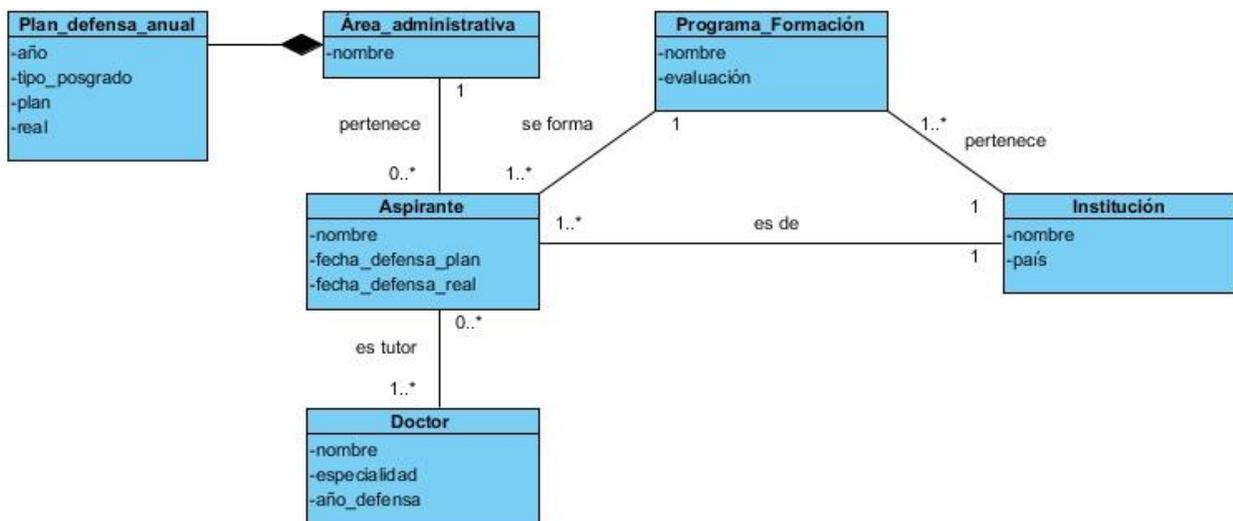


Figura 10 Diagrama de objetos de negocio (elaboración propia)

## Indicadores que maneja el tablero de control

Una vez descrito el proceso a informatizar es necesario especificar los indicadores que maneja el tablero de control. Los mismos son resumidos en la siguiente tabla.

Tabla 2. Indicadores que maneja el tablero de control

No	Nombre del indicador	Descripción
1	Doctorados defendidos en la UCI en un período dado	Muestra desde la fundación de la UCI (año 2002) la cantidad de profesionales que han defendido su doctorado (sin distinguir el tipo de programa).
2	Cantidad de doctores por año	Muestra la cantidad de doctores con que ha contado la UCI por año.
3	Doctorados defendidos por área de la UCI	Muestra la cantidad de profesionales que han defendido su doctorado por áreas administrativas.
4	Composición de la matrícula del programa doctoral de la UCI	Muestra los matriculados en el programa doctoral de la UCI (propios de la UCI, nacionales y extranjeros)
5	Aspirantes de la UCI por tipo de programa	Muestra los aspirantes que están matriculados en el programa doctoral propio de la UCI, los que están matriculados en otros programas en Cuba y los que están matriculados en programas en el extranjero.
6	Composición de doctores por ramas de la ciencia	Muestra la distribución en porciento de los doctores de la UCI por las diferentes ramas de la ciencia.
7	Aspirantes por ramas de la ciencias	Muestra la distribución en porciento de los aspirantes de la UCI por las diferentes ramas de la ciencia.
8	Compromisos de defensa por año	Muestra por año los compromisos futuros de defensa de los aspirantes de la UCI.
9	Maestrías defendidas en la UCI	Muestra por año las defensas de maestría realizadas.
10	Maestrías defendidas por área	Muestra por área las defensas de maestría realizadas.
11	Matrícula de las maestrías propias	Muestra la matrícula actual de las maestrías propias de la UCI.

## **Sistema informático propuesto**

El sistema propuesto consiste en la informatización del tablero de control. El mismo contribuye a perfeccionar el control del proceso de formación posgraduada en la UCI, al facilitar la gestión de los indicadores que evalúan su desempeño. Para ello es necesario actualizar toda la información relativa a los programas de maestría y doctorado que cursan aspirantes de la UCI y de otras instituciones en la universidad. Con esta información consolidada y mediante un conjunto de reportes gráficos organizados en un tablero de control, es posible monitorear el estado de la ejecución de los programas de posgrado, así como el comportamiento de los principales indicadores definidos en el epígrafe precedente.

Para cumplir con lo anterior el sistema propuesto contiene un conjunto de formularios para actualizar la información de posgrado. En otra sección aparecen organizados los gráficos y tablas resúmenes que conforman el tablero de control para monitorear la actividad.

### ***Funcionalidades del sistema***

El sistema implementado debe ser capaz de gestionar las principales actividades del área de formación de posgrado. Para ello se implementaron las funcionalidades que se listan a continuación:

1. Gestionar doctores:
  - a. Adicionar doctores.
  - b. Modificar doctores (actualizando la fecha de baja se da baja al doctor).
  - c. Eliminar doctores.
  - d. Listar doctores.
  - e. Buscar doctores.
  
2. Gestionar institución (autorizada a impartir programas de posgrado y/o de los aspirantes):
  - a. Adicionar institución.
  - b. Modificar institución.
  - c. Eliminar institución.
  - d. Listar institución.
  - e. Buscar institución

3. Gestionar programa formación:
  - a. Adicionar programa formación.
  - b. Modificar programa formación.
  - c. Eliminar programa formación.
  - d. Listar programa formación.
  - e. Buscar programa formación.
  
4. Gestionar aspirantes:
  - a. Adicionar aspirantes.
  - b. Modificar aspirantes.
  - c. Eliminar aspirantes.
  - d. Listar aspirantes.
  - e. Buscar aspirantes.
  
5. Gestionar defensas de maestrías y doctorados.
  - a. Actualizar fecha de defensa real del aspirante.
  
6. Mostrar compromisos de defensas de maestrías y doctorados por área.
  - a. Mostrar cantidad de defensas previstas por áreas por año.
  - b. Mostrar cantidad real de defensas contra compromisos de los aspirantes por área por año.
  
7. Mostrar información consolidada.
  - a. Mostrar matrícula por programas de formación de maestrías y doctorados.
  - b. Composición de la matrícula por programas de formación
  - c. Defensas de maestrías y doctorados por áreas.
  
8. Mostrar información histórica.
  - a. Defensas de maestría y doctorados ocurridas en un período.
  - b. Cantidad de doctores de la UCI por años.
  - c. Defensas realizadas vs compromisos futuros.
  
9. Autenticar usuario (función propia de la administración del sistema):
  - a. Autenticar usuario
  - b. Gestionar permisos de acceso al sistema

### **Historias de usuarios**

Entre las técnicas ágiles que utiliza AUP se encuentran el modelado ágil que permite encapsular los requisitos funcionales en Historias de Usuarios (HU) y la descripción de requisitos por procesos o por Casos de Uso.

Las HU son descripciones cortas y simples de las funcionalidades del sistema, narradas desde la perspectiva de la persona que desea dicha funcionalidad. Poseen una descripción escrita que será utilizada para planificar y posteriormente disgregar los detalles con el dueño del producto (Sánchez, Torres et al. 2003).

Las HU se clasifican de la siguiente manera:

1. Teniendo en cuenta la Escala Nominal de Prioridad en el Negocio:

- Alta:** se le otorga a las HU que constituyen funcionalidades de vital importancia en el desarrollo del proyecto.
- Media:** se le otorga a las HU que para el cliente constituyen funcionalidades a tener en cuenta sin que tengan una afectación directa sobre el proyecto que se está desarrollando.
- Baja:** se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de la estructura y que no tienen nada que ver con el proyecto en desarrollo.

2. Teniendo en cuenta la escala nominal de riesgo en desarrollo:

- Alta:** se otorga cuando para la implementación de la HU se considera la posible existencia de errores que lleven a la inoperatividad del código.
- Media:** se otorga cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- Baja:** se otorga cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU estarán constituidas mediante tablas divididas por las siguientes secciones:

- Número:** número de la historia de usuario incremental en el tiempo.
- Nombre de Historia de Usuario:** el nombre de la historia de usuario para identificarlas mejor entre los desarrolladores y el cliente.
- Usuario:** involucrados en la ejecución de la HU.
- Iteración Asignada:** número de la iteración.
- Prioridad en negocio:** muy alta, alta, media o baja.
- Riesgo en Desarrollo:** alto, medio o bajo.
- Programador responsable:** involucrados en el desarrollo de la HU.
- Puntos estimados:** tiempo estimado que se demorará el desarrollo de la HU, debe ser menor que tres semanas.
- Descripción:** breve descripción de la HU.
- Observaciones:** señalamiento o advertencia del sistema.
- Prototipo de interfaz:** prototipo de interfaz si aplica.

A continuación se detallan las historias de usuario que definen las funcionalidades del sistema.

Tabla 3. Historia de usuario: gestionar doctores de la UCI

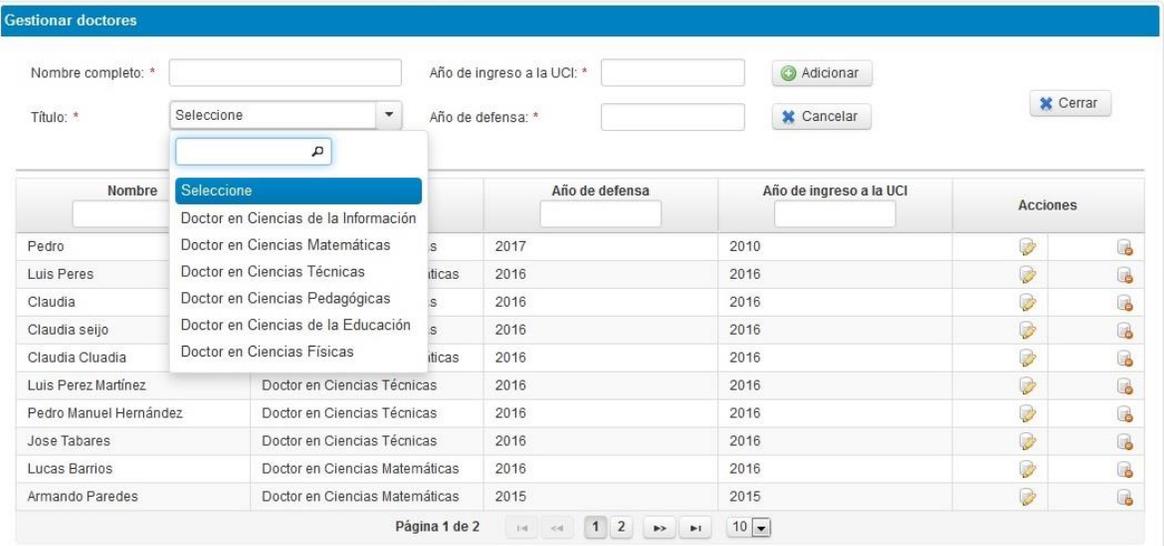
Historias de usuario	
<b>Número:</b> HU1	<b>Nombre del requisito:</b> Gestionar doctores de la UCI.
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 0.8 semana
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas
<b>Descripción:</b> Permite gestionar los doctores de la universidad en el sistema, facilitando adicionar, eliminar y modificar los siguientes datos: nombre del doctor, año de defensa, año de ingreso a la universidad, si es baja o no (indicado por el año en que es baja).	
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.	
<b>Prototipo de interfaz:</b> 	

Tabla 4. Historia de usuario: gestionar institución

Historias de usuario	
<b>Número:</b> HU2	<b>Nombre del requisito:</b> Gestionar Institución
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 0.6 semana
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas
<b>Descripción:</b> Permite gestionar las instituciones autorizadas a impartir programas de posgrado y de los aspirantes a través de las siguientes operaciones: adicionar, modificar o eliminar, insertando los siguientes datos: nombre de la institución y el país.	
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.	
Prototipo de interfaz:	
<p>The screenshot displays a web interface titled "Gestionar institución". At the top, there is a blue header bar with the title. Below it, a form contains two input fields: "Nombre" (with a red asterisk) and "País" (with a dropdown menu and a red asterisk). To the right of the "Nombre" field is a green "Adicionar" button. To the right of the "País" field is a blue "Cancelar" button. Further right is a blue "Cerrar" button. Below the form is a table with three columns: "Nombre", "Título", and "Acciones". The table lists several institutions, each with a yellow edit icon and a red delete icon in the "Acciones" column. At the bottom of the table, there is a pagination bar showing "Página 1 de 1" and navigation controls for page 1 of 10.</p>	

Tabla 5. Historia de usuario: gestionar programa de formación

Historias de usuario																																																			
<b>Número:</b> HU3	<b>Nombre del requisito:</b> Gestionar programa formación.																																																		
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 1																																																		
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 1.5 semana																																																		
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas																																																		
<b>Descripción:</b> Permite gestionar un programa de formación de posgrado de la universidad, facilitando adicionar, modificar o eliminar los siguientes datos: nombre del programa, nombre del tipo de posgrado (maestría o doctorado), institución a la que pertenece y la evaluación del programa (autorizado, ratificado, certificado, excelencia).																																																			
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.																																																			
<b>Prototipo de interfaz:</b>																																																			
<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo de Postgrado</th> <th>Evaluación</th> <th>Institución</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Informática aplicada</td> <td>Maestría</td> <td>Aprobado</td> <td>Universidad de las Ciencias Informáticas</td> <td> </td> </tr> <tr> <td>Informática</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad de las Ciencias Informáticas</td> <td> </td> </tr> <tr> <td>Ciencias de la Educación</td> <td>Doctorado</td> <td>Aprobado</td> <td>Instituto Superior Politécnico José Antonio Echeverría</td> <td> </td> </tr> <tr> <td>Ciencias Pedagógicas</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad de Ciencias Pedagógicas Enrique José Varona</td> <td> </td> </tr> <tr> <td>Ciencias Pedagógicas</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad de Camaguey</td> <td> </td> </tr> <tr> <td>Automática</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad Central de las Villas</td> <td> </td> </tr> <tr> <td>Ciencias de la Educación</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad de La Habana</td> <td> </td> </tr> <tr> <td>Computación</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad Central de las Villas</td> <td> </td> </tr> <tr> <td>Ciencias Matemáticas</td> <td>Doctorado</td> <td>Aprobado</td> <td>Universidad de La Habana</td> <td> </td> </tr> </tbody> </table>		Nombre	Tipo de Postgrado	Evaluación	Institución	Acciones	Informática aplicada	Maestría	Aprobado	Universidad de las Ciencias Informáticas		Informática	Doctorado	Aprobado	Universidad de las Ciencias Informáticas		Ciencias de la Educación	Doctorado	Aprobado	Instituto Superior Politécnico José Antonio Echeverría		Ciencias Pedagógicas	Doctorado	Aprobado	Universidad de Ciencias Pedagógicas Enrique José Varona		Ciencias Pedagógicas	Doctorado	Aprobado	Universidad de Camaguey		Automática	Doctorado	Aprobado	Universidad Central de las Villas		Ciencias de la Educación	Doctorado	Aprobado	Universidad de La Habana		Computación	Doctorado	Aprobado	Universidad Central de las Villas		Ciencias Matemáticas	Doctorado	Aprobado	Universidad de La Habana	
Nombre	Tipo de Postgrado	Evaluación	Institución	Acciones																																															
Informática aplicada	Maestría	Aprobado	Universidad de las Ciencias Informáticas																																																
Informática	Doctorado	Aprobado	Universidad de las Ciencias Informáticas																																																
Ciencias de la Educación	Doctorado	Aprobado	Instituto Superior Politécnico José Antonio Echeverría																																																
Ciencias Pedagógicas	Doctorado	Aprobado	Universidad de Ciencias Pedagógicas Enrique José Varona																																																
Ciencias Pedagógicas	Doctorado	Aprobado	Universidad de Camaguey																																																
Automática	Doctorado	Aprobado	Universidad Central de las Villas																																																
Ciencias de la Educación	Doctorado	Aprobado	Universidad de La Habana																																																
Computación	Doctorado	Aprobado	Universidad Central de las Villas																																																
Ciencias Matemáticas	Doctorado	Aprobado	Universidad de La Habana																																																

Tabla 6. Historia de usuario: gestionar aspirantes

Historias de usuario	
<b>Número:</b> HU4	<b>Nombre del requisito:</b> Gestionar aspirantes
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 1.5 semana
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas
<b>Descripción:</b> Permite gestionar los aspirantes a través de las siguientes operaciones: adicionar, modificar o eliminar los siguientes datos: nombre del aspirante, cargo, si es externo o no a la UCI (para identificar estudiantes extranjeros o de otras instituciones del país), institución a la que pertenece (de ser externo), área a la que pertenece (si es de la UCI), año de defensa por plan, programa de formación en el que está matriculado y el tutor asignado.	
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.	
Prototipo de interfaz:	

Tabla 7. Historia de usuario: mostrar compromisos de defensas

Historias de usuario																							
Número:HU5		Nombre del requisito: Mostrar compromisos de defensas de maestrías y doctorados por área.																					
Programador: Claudia Seijo Pérez										Iteración Asignada: 2													
Prioridad: media										Tiempo Estimado: 1 semana													
Riesgo en Desarrollo: <ul style="list-style-type: none"> <li>Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>Problemas eléctricos.</li> </ul>										Tiempo Real: 40 horas													
<b>Descripción:</b> Permite consolidar los compromisos de defensas de maestrías y doctorados por área, así como las defensas que realmente ocurrieron en el año.																							
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.																							
Prototipo de interfaz:																							
Año	Fac 1		Fac 2		Fac 3		Fac 4		Fac 5		Fac 6		FICI		otras áreas		Fuera de la UCI		TOTAL				
	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Plan	Real	Compromiso	Potencial	Defendidos	% cumplimiento	
2002																1						1	
2003																						0	
2004																1						1	
2005								1		1												2	
2006																						0	
2007																1						1	#iDIV/0!
2008	1																					1	#iDIV/0!
2009																2						2	#iDIV/0!
2010																1						1	#iDIV/0!
2011	1									1						2						4	#iDIV/0!
2012	2				3		1		1							1						8	#iDIV/0!
2013	1		3		1		2		1			2									10	10	100%
2014	2		2		2		1		3		2										12	10	83%
2015	2				2	2			3	3	1	1	2	2		0			1		10	9	90%
2016	2		1		3		2		1	1	1		2								8	8	13%
2017																						39	
2018																						42	
2019																						37	
	7		3		8		5		11		5		2		9		1				126	51	

Tabla 8. Historia de usuario: gestionar defensas

Historias de usuario																																													
<b>Número:</b> HU6	<b>Nombre del requisito:</b> Gestionar defensas de maestrías y doctorados																																												
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 2																																												
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 0.5 semana																																												
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas																																												
<b>Descripción:</b> Una vez defendida la tesis por el aspirante, se registra en el sistema la fecha en que ocurrió la defensa. Esto indica que el aspirante concluyó su formación como máster o doctor. Se emplea la misma interfaz usada para gestionar un aspirante.																																													
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.																																													
Prototipo de interfaz:																																													
<div style="border: 1px solid #0070C0; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px;"><b>Gestionar aspirantes</b></div> <div style="padding: 10px;"> <p>Nombre completo: * <input type="text"/> Institución: * <input type="text" value="Seleccione"/></p> <p>Año de defensa real: <input type="text"/> Área: * <input type="text" value="Seleccione"/> <input type="button" value="Adicionar"/></p> <p>Año de defensa en plan: * <input type="text"/> Cargo: * <input type="text" value="Seleccione"/> <input type="button" value="Cancelar"/> <input type="button" value="Cerrar"/></p> <p>Tipo de postgrado: * <input type="text" value="Seleccione"/> Programa de formación: * <input type="text" value="Seleccione"/></p> <p>Tipo de programa: * <input type="text" value="Seleccione"/></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Nombre</th> <th>Institución</th> <th>Plan</th> <th>Real</th> <th>Área</th> <th>Cargo</th> <th>Tipo de Postgrado</th> <th>Tipo de Programa</th> <th>Programa de formación</th> <th colspan="2">Acciones</th> </tr> </thead> <tbody> <tr> <td>Lucas Barrios</td> <td>Universidad de las Ciencias Informáticas</td> <td>2016</td> <td>2016</td> <td>Facultad 6</td> <td>Director</td> <td>Maestría</td> <td>Ciencias Básicas</td> <td>Informática aplicada</td> <td></td> <td></td> </tr> <tr> <td>Eduardo González Cruz</td> <td>Universidad de las Ciencias Informáticas</td> <td>2014</td> <td>2014</td> <td>Facultad 5</td> <td>Director</td> <td>Doctorado</td> <td>Otros programas</td> <td>Informática</td> <td></td> <td></td> </tr> <tr> <td>Claudia Seijo</td> <td>Universidad de las</td> <td>2015</td> <td>2015</td> <td>Facultad 3</td> <td>Director</td> <td>Doctorado</td> <td>Educación-</td> <td>Informática</td> <td></td> <td></td> </tr> </tbody> </table> </div> </div>		Nombre	Institución	Plan	Real	Área	Cargo	Tipo de Postgrado	Tipo de Programa	Programa de formación	Acciones		Lucas Barrios	Universidad de las Ciencias Informáticas	2016	2016	Facultad 6	Director	Maestría	Ciencias Básicas	Informática aplicada			Eduardo González Cruz	Universidad de las Ciencias Informáticas	2014	2014	Facultad 5	Director	Doctorado	Otros programas	Informática			Claudia Seijo	Universidad de las	2015	2015	Facultad 3	Director	Doctorado	Educación-	Informática		
Nombre	Institución	Plan	Real	Área	Cargo	Tipo de Postgrado	Tipo de Programa	Programa de formación	Acciones																																				
Lucas Barrios	Universidad de las Ciencias Informáticas	2016	2016	Facultad 6	Director	Maestría	Ciencias Básicas	Informática aplicada																																					
Eduardo González Cruz	Universidad de las Ciencias Informáticas	2014	2014	Facultad 5	Director	Doctorado	Otros programas	Informática																																					
Claudia Seijo	Universidad de las	2015	2015	Facultad 3	Director	Doctorado	Educación-	Informática																																					

Tabla 9. Historia de usuario: mostrar información consolidada

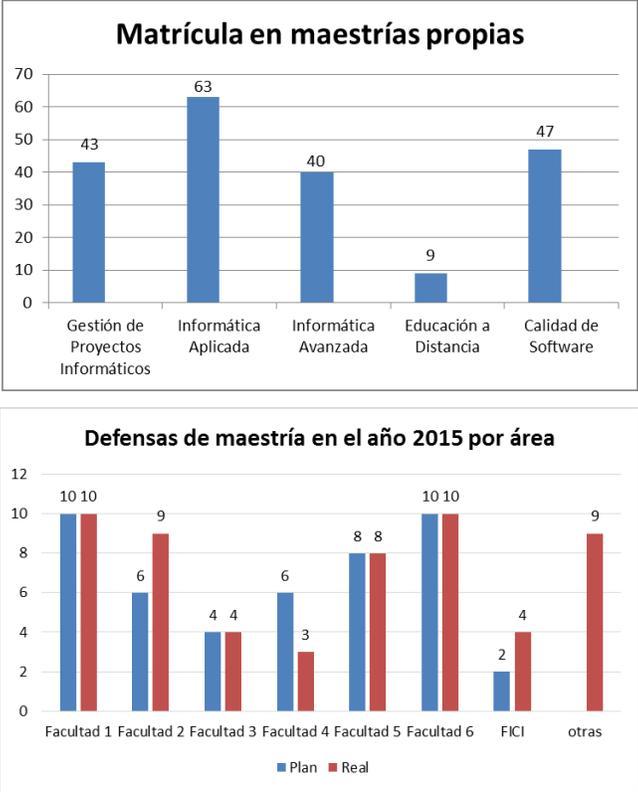
Historias de usuario																																								
<b>Número:</b> HU7	<b>Nombre del requisito:</b> Mostrar información consolidada.																																							
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 3																																							
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 1.8 semana																																							
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas																																							
<b>Descripción:</b> Permite mostrar la cantidad de matriculados por programas y las defensas de maestrías y doctorados por área.																																								
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.																																								
<b>Prototipo de interfaz:</b>																																								
 <p><b>Matrícula en maestrías propias</b></p> <table border="1"> <thead> <tr> <th>Programa</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr> <td>Gestión de Proyectos Informáticos</td> <td>43</td> </tr> <tr> <td>Informática Aplicada</td> <td>63</td> </tr> <tr> <td>Informática Avanzada</td> <td>40</td> </tr> <tr> <td>Educación a Distancia</td> <td>9</td> </tr> <tr> <td>Calidad de Software</td> <td>47</td> </tr> </tbody> </table> <p><b>Defensas de maestría en el año 2015 por área</b></p> <table border="1"> <thead> <tr> <th>Área</th> <th>Plan</th> <th>Real</th> </tr> </thead> <tbody> <tr> <td>Facultad 1</td> <td>10</td> <td>10</td> </tr> <tr> <td>Facultad 2</td> <td>6</td> <td>9</td> </tr> <tr> <td>Facultad 3</td> <td>4</td> <td>4</td> </tr> <tr> <td>Facultad 4</td> <td>6</td> <td>3</td> </tr> <tr> <td>Facultad 5</td> <td>8</td> <td>8</td> </tr> <tr> <td>Facultad 6</td> <td>10</td> <td>10</td> </tr> <tr> <td>FICI</td> <td>2</td> <td>4</td> </tr> <tr> <td>otras</td> <td>0</td> <td>9</td> </tr> </tbody> </table>		Programa	Cantidad	Gestión de Proyectos Informáticos	43	Informática Aplicada	63	Informática Avanzada	40	Educación a Distancia	9	Calidad de Software	47	Área	Plan	Real	Facultad 1	10	10	Facultad 2	6	9	Facultad 3	4	4	Facultad 4	6	3	Facultad 5	8	8	Facultad 6	10	10	FICI	2	4	otras	0	9
Programa	Cantidad																																							
Gestión de Proyectos Informáticos	43																																							
Informática Aplicada	63																																							
Informática Avanzada	40																																							
Educación a Distancia	9																																							
Calidad de Software	47																																							
Área	Plan	Real																																						
Facultad 1	10	10																																						
Facultad 2	6	9																																						
Facultad 3	4	4																																						
Facultad 4	6	3																																						
Facultad 5	8	8																																						
Facultad 6	10	10																																						
FICI	2	4																																						
otras	0	9																																						

Tabla 10. Historia de usuario: mostrar información histórica

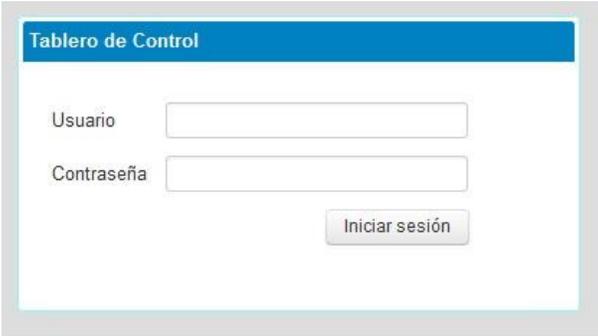
Historias de usuario																															
<b>Número:</b> HU8	<b>Nombre del requisito:</b> Mostrar información histórica.																														
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 3																														
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 1.8 semana																														
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas																														
<b>Descripción:</b> Permite mostrar las defensas de maestrías y doctorados ocurridas en un periodo de tiempo así como la cantidad de doctores de la UCI por año ,las defensas realizadas y los compromisos futuros.																															
<b>Observaciones:</b> El usuario debe estar autenticado en el sistema.																															
<b>Prototipo de interfaz:</b>																															
<table border="1"> <caption>Cantidad de Doctores de la UCI (2012-2016)</caption> <thead> <tr> <th>Año</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr><td>2010</td><td>28</td></tr> <tr><td>2011</td><td>44</td></tr> <tr><td>2012</td><td>46</td></tr> <tr><td>2013</td><td>58</td></tr> <tr><td>2014</td><td>63</td></tr> <tr><td>2015</td><td>63</td></tr> <tr><td>2016</td><td>64</td></tr> </tbody> </table>		Año	Cantidad	2010	28	2011	44	2012	46	2013	58	2014	63	2015	63	2016	64														
Año	Cantidad																														
2010	28																														
2011	44																														
2012	46																														
2013	58																														
2014	63																														
2015	63																														
2016	64																														
<table border="1"> <caption>Doctorados defendidos en la UCI (2002-2015)</caption> <thead> <tr> <th>Año</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr><td>2002</td><td>1</td></tr> <tr><td>2003</td><td>0</td></tr> <tr><td>2004</td><td>1</td></tr> <tr><td>2005</td><td>2</td></tr> <tr><td>2006</td><td>0</td></tr> <tr><td>2007</td><td>1</td></tr> <tr><td>2008</td><td>1</td></tr> <tr><td>2009</td><td>2</td></tr> <tr><td>2010</td><td>1</td></tr> <tr><td>2011</td><td>4</td></tr> <tr><td>2012</td><td>8</td></tr> <tr><td>2013</td><td>10</td></tr> <tr><td>2014</td><td>10</td></tr> <tr><td>2015</td><td>9</td></tr> </tbody> </table>		Año	Cantidad	2002	1	2003	0	2004	1	2005	2	2006	0	2007	1	2008	1	2009	2	2010	1	2011	4	2012	8	2013	10	2014	10	2015	9
Año	Cantidad																														
2002	1																														
2003	0																														
2004	1																														
2005	2																														
2006	0																														
2007	1																														
2008	1																														
2009	2																														
2010	1																														
2011	4																														
2012	8																														
2013	10																														
2014	10																														
2015	9																														

A nivel de prototipo de interfaz de usuario, las dos historias de usuario anteriores se representan unificadas en la aplicación. Esto puede ser corroborado en la siguiente figura.



Figura 6 Área para los gráficos que conforman el tablero de control (elaboración propia)

Tabla 11. Historia de usuario: autenticar usuario

Historias de usuario	
<b>Número:</b> HU9	<b>Nombre del requisito:</b> Autenticar usuario.
<b>Programador:</b> Claudia Seijo Pérez	<b>Iteración Asignada:</b> 3
<b>Prioridad:</b> alta	<b>Tiempo Estimado:</b> 1.3 semana
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>• Rotura de alguna estación de trabajo perteneciente al proyecto.</li> <li>• Problemas eléctricos.</li> </ul>	<b>Tiempo Real:</b> 40 horas
<b>Descripción:</b> Permite autenticar usuarios en el sistema empleando el LDAP ( <i>Lightweight Directory Access Protocol</i> ) de la UCI.	
<b>Observaciones:</b> El LDAP de la UCI debe estar disponible.	
<b>Prototipo de interfaz:</b>	
	

**Estimación de esfuerzo por Historias de Usuario**

La metodología Proceso Unificado Ágil basa sus procesos de planificación en estimaciones temporales de las historias de usuario, las cuáles deben ser realizadas por los desarrolladores durante las diversas reuniones de planificación. El objetivo principal es estimar cuánto tiempo llevará implementar las historias. Todas las estimaciones que maneja la programación extrema se cuantifican en semanas de desarrollo ideal, lo cual no es más que la cantidad de trabajo que se puede realizar durante una semana sin distracciones (llamadas telefónicas, reuniones, enfermedades, comidas, etc.), sin trabajos extras, sin la asignación de otras tareas, sin dependencias, y suponiendo que el desarrollador sabe exactamente lo que hay que hacer.

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo del sistema:

Tabla 10. Estimación del esfuerzo por historia de usuario

No	Historias de Usuario	Puntos de estimación
1	Gestionar doctores	0.8
2	Gestionar Institución	0.6
3	Gestionar programa formación	1.5
4	Gestionar aspirantes	1.5
5	Mostrar compromisos de defensas de maestrías y doctorados por área.	1.0
6	Gestionar defensas de maestrías y doctorados	0.5
7	Mostrar información consolidada	1.8
8	Mostrar información histórica	1.8
9	Autenticar usuarios	1.3

**Plan de iteraciones**

Luego de estimar el esfuerzo dedicado a cada historia de usuario se procedió a planificar la implementación del sistema. La misma quedó organizada en tres iteraciones.

Tabla 11. Planificación de la implementación del sistema

Iteración	Historias de Usuario	Duración total	Porcentaje
1	Gestionar doctores de la UCI Gestionar institución Gestionar programa de formación	3.5 semanas	27%
2	Gestionar aspirantes Mostrar compromisos de defensa de maestría y doctorado por área Gestionar defensas de maestría y doctorado	3 semanas	28%
3	Mostrar información consolidada Mostrar información histórica Autenticar usuarios	4.9 semanas	45%
		Total 12 semanas	100%

**Plan de entrega**

El plan de entrega marca las fechas de terminación de cada una de las iteraciones que tendrá el sistema, así como los componentes que serán obtenidos en cada una de ellas.

Tabla 12. Plan de entrega

Sistema	Final de la iteración 1	Final de la iteración 2	Final de la iteración 3
Tablero de control para la Dirección de Posgrado de la Universidad de las Ciencias Informáticas	26/2/2016	22/3/2016	29/4/2016

## Modelo de datos

### Modelo físico de datos

La información manejada por el sistema persiste en una base de datos relacional. El modelo físico para estructurar y almacenar los datos es el que se muestra en la siguiente figura.

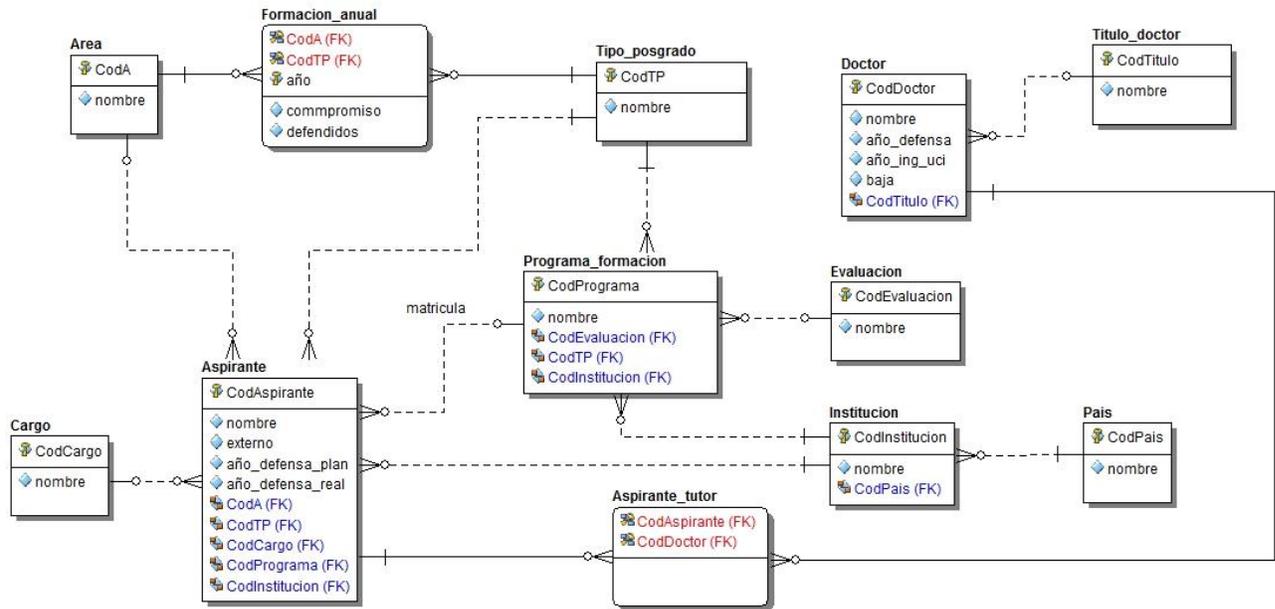


Figura 7 Modelo físico de datos del sistema (elaboración propia)

### Diccionario de datos

**Tipo\_posgrado:** En esta tabla se almacenan los diferentes tipos de posgrados (maestrías y doctorados). La misma contiene nombre y código. Además, tiene relación con las tablas área y aspirante. Con área la relación es de muchos a muchos por lo que se crea otra tabla.

**Área:** Contiene las diferentes áreas (facultades y otras estructuras administrativas). Almacena el código del área y su nombre. Tiene relación con las tablas Tipo\_posgrado y Aspirante.

**Formación\_ anual:** Es el resultado de la relación entre las tablas tipo\_posgrado y área, donde para cada año se registran los compromisos y los defendidos de las áreas.

**Aspirante:** En esta tabla se almacenan todos los datos del aspirante como su nombre, si es externo a la UCI o no, el código que lo identifica, además del año que debe defender por el plan y el año que defendió su tesis. Tiene relación con las tablas área, tipo\_posgrado, doctor, programa\_formacion e institución y cargo.

**Cargo:** Almacena el nombre y código que identifica al cargo de los aspirantes. Presenta relación con la tabla aspirante.

**Programa\_ formación:** Almacena todos los programas de formación de posgrado, tanto de maestrías como doctorados, propios de la UCI o que se imparten por otras instituciones. De cada programa se almacena el nombre, código que lo identifica y la evaluación vigente del programa (Autorizado, Ratificado, Certificado, Excelencia). Presenta relación con las tablas aspirante, institución y tipo de posgrado.

**Evaluación:** Almacena las categorías evaluativas de un programa de posgrado. Para registrarlas se almacena de cada una su nombre y el código. Se relaciona con la tabla programa formación.

**Institución:** En esta tabla se almacena el nombre de todas las instituciones registradas, tanto las autorizadas a impartir posgrado como las de procedencia de los aspirantes. De cada institución se registra el código, el nombre y el país al cual pertenece. Presenta una relación con las tablas aspirante y programa formación.

**País:** Se guarda el nombre del país y el código. Guarda relación con la tabla institución, para reflejar el país al que pertenece.

**Doctor:** En esta tabla se almacena los datos de cada doctor de la UCI. De cada uno se almacena el nombre, código, el año en que ingresó a la universidad, título de doctor que alcanzó, rama a la cual pertenece, año en el cual defendió y si es baja o no de la institución.

**Aspirante\_tutor:** Es el resultado de la relación entre las tablas aspirante y doctor. En la misma se almacena de cada aspirante sus tutores.

**Título\_ doctor:** Guarda el código del título y el nombre del grado científico alcanzado. Guarda relación con la tabla doctor.

## **Arquitectura de la aplicación**

### ***Propiedades del sistema***

El sistema debe cumplir con determinadas cualidades y propiedades; por estas razones se describen a continuación, algunas de las características necesarias para el funcionamiento del producto final.

### **Usabilidad**

El manejo del sistema debe ser sencillo e intuitivo empleando iconos que sean de rápida identificación con su funcionalidad. La navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

### **Ambiente de ejecución del software**

El ambiente de ejecución de la aplicación es multiplataforma, se recomienda al cliente utilizar un sistema operativo donde se pueda instalar un navegador web para acceder al sistema.

Para el servidor: se usará un sistema operativo:

- Ubuntu desde la versión 12.04 hasta la 16.04.

- Debian 6 o 7

- Nova

Gestor de base de datos: PostgreSQL.

Se recomienda el uso del navegador Mozilla Firefox en su versión 25 o superior para acceder a las interfaces del sistema y Chrome en su versión 51.0.2704.

## Interfaz de usuario

El sistema deberá respetar las pautas de diseño de la UCI, así como con las de la dirección de posgrado, para su integración con dicho sistema.

## Seguridad

Políticas de seguridad por usuarios y roles, el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado. El sistema garantizará la autenticación como primera acción.

## Estilo arquitectónico empleado

Para el desarrollo de la solución se utilizó una arquitectura en tres capas, la que permite que las distintas partes de la aplicación se puedan modificar sin que se afecten las demás capas; es decir, separar los diferentes aspectos del desarrollo tales como presentación, lógica de negocio y mecanismos de almacenamiento.



Figura 8 Arquitectura de tres capas del sistema (elaboración propia)

**Capa de presentación:** Es la interfaz del programa frente al usuario, esta debe cumplir su propósito con el usuario final, fácil de usar y amigable. Las interfaces deben ser consistentes con la información dentro del software.

**Capa de lógica de negocios:** Se encuentran los programas que son ejecutados, recibe las peticiones del usuario y posteriormente envía las respuestas tras el proceso. La lógica de negocio es la intermediaria entre la capa de presentación y la capa de acceso a datos.

**Capa de datos:** Se encarga de hacer las transacciones con la base de datos y con otros sistemas para descargar o insertar información al sistema. La consistencia en los datos es sumamente importante. Mantendrá la comunicación con la de lógica de negocio al enviar la información que será procesada e ingresada en objetos según sea necesario. La capa de acceso a datos es la encargada de satisfacer las peticiones del negocio a través de los Objetos de Acceso a Datos (DAO), para ello, se utilizan interfaces DAO y sus implementaciones.

### ***Estructura de paquetes de implementación***

Estructuralmente la aplicación está organizada por paquetes de implementación. En ellos existen un conjunto de clases con responsabilidades específicas que en colaboración permiten responder a las solicitudes de los usuarios. Por cada petición se procesa la vista que está contenida en el paquete UI. Para esto se utilizan los paquetes Convertidor (contiene clases que son implementaciones de la interfaz Converter y se encargan de hacer las transformaciones: cadena>Tipo JAVA y viceversa) y Validador (está definido por la interfaz javax.faces.validator.Validator y sus múltiples implementaciones aseguran que el tipo de un dato y su valor introducido por el usuario sean correctos).

Una vez procesada la vista, la clase controladora que atiende la petición actualiza el objeto de negocio que se manipula. Dicha clase controladora está ubicada en el paquete Controlador, el cual contiene las clases que actúan como conductos entre la interfaz de usuario y los servicios de negocio. Por otro lado, las clases que generan los objetos de negocio están organizadas en el paquete Entidades. La clase controladora además invoca la función de negocio que se requiera del paquete Servicio (contiene las clases encargadas de las funciones de negocio). Este utiliza su implementación correspondiente para satisfacer la petición y hace uso del paquete OAD (contiene las clases que manejan

la conexión con la base de datos para obtener y guardar los datos) para acceder a los datos necesarios que se encuentran en el servidor de base de datos.

La siguiente figura ilustra los paquetes que conforman la aplicación, así como las relaciones de dependencia entre estos explicadas anteriormente.

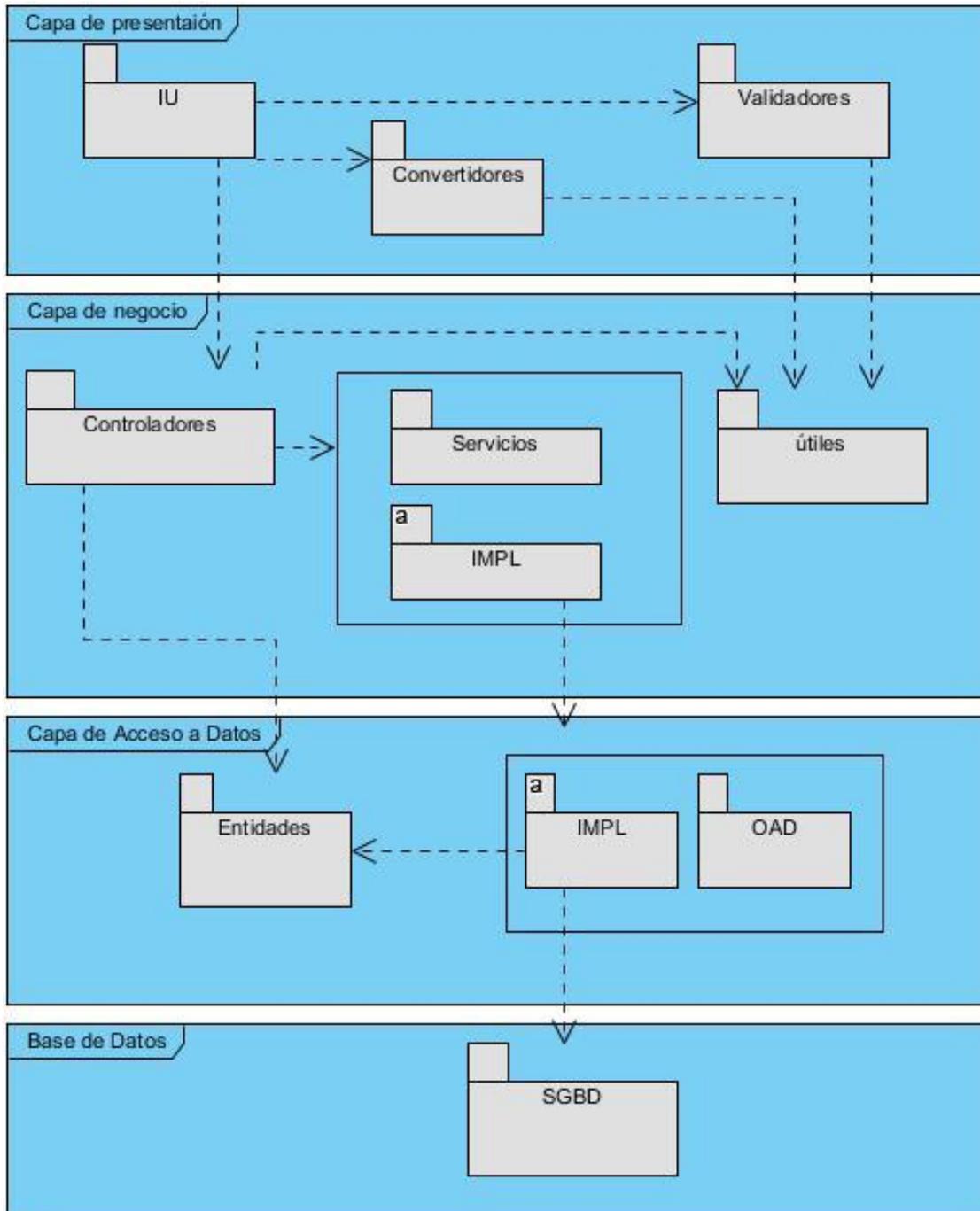


Figura 9 Estructura de paquetes de implementación (elaboración propia)

La estructura de paquetes anteriormente descrita quedó reflejada en la forma en que fue organizado el código de la aplicación. La siguiente figura ilustra los paquetes de implementación en la herramienta de programación empleada.

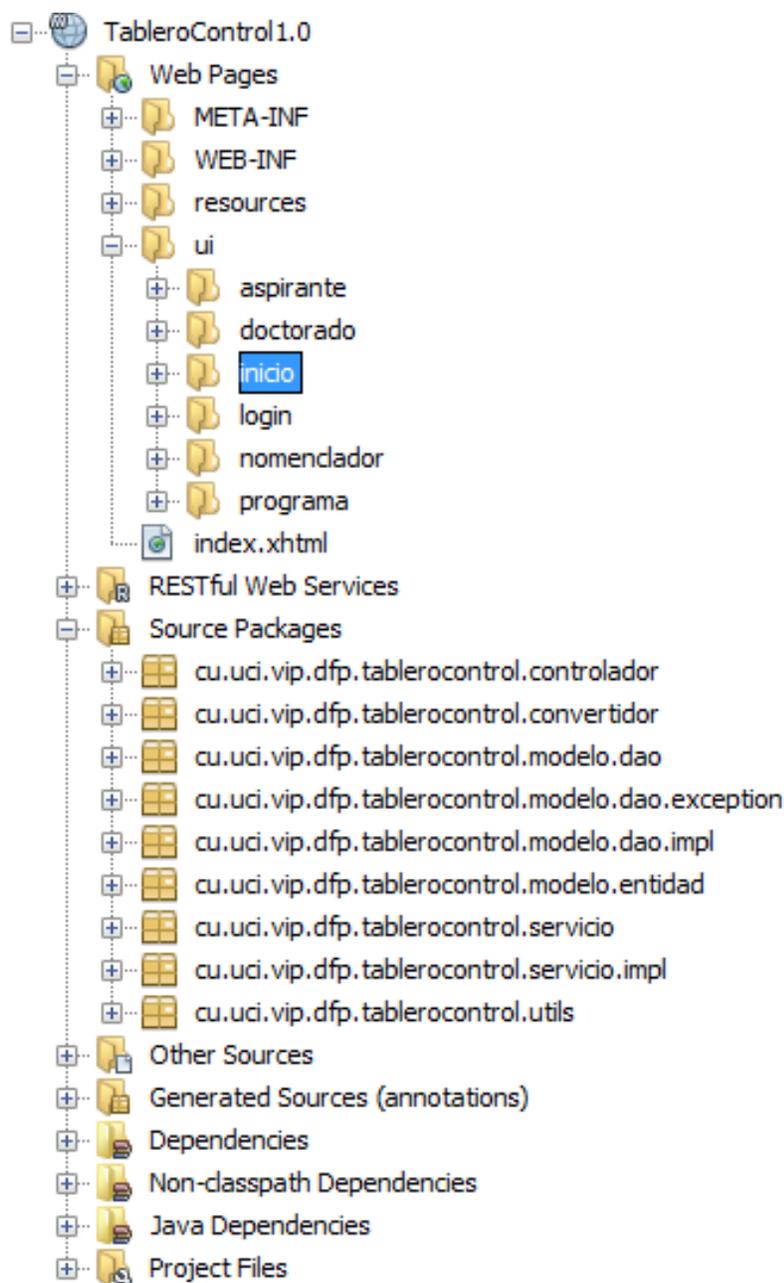


Figura 10 Estructura de paquetes vista desde la herramienta de desarrollo (elaboración propia)

### Patrones de diseño empleados

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Representan una descripción de las clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. El uso de patrones posibilita estandarizar el modo en que se realiza el diseño y proporciona reusabilidad, extensibilidad y mantenimiento del código (Larman 2004).

### Patrones GRASP

Los patrones GRASP describen la asignación de responsabilidades a objetos. Para el desarrollo del sistema fueron utilizados los siguientes patrones de esta clasificación, que en su mayoría son implementados de forma nativa por los marcos de trabajo empleados:

- **Creador:** este patrón es el encargado de que una clase B cree una instancia de una clase A, siempre que: La clase B contenga a la clase A; B sea una agregación (o composición) de A; B almacene a A; B tenga los datos de inicialización de A o B use a A.

El patrón se refleja en la implementación de la clase **AreaControlador**, que contiene la información necesaria para crear y usar la instancia de la clase **Area**.

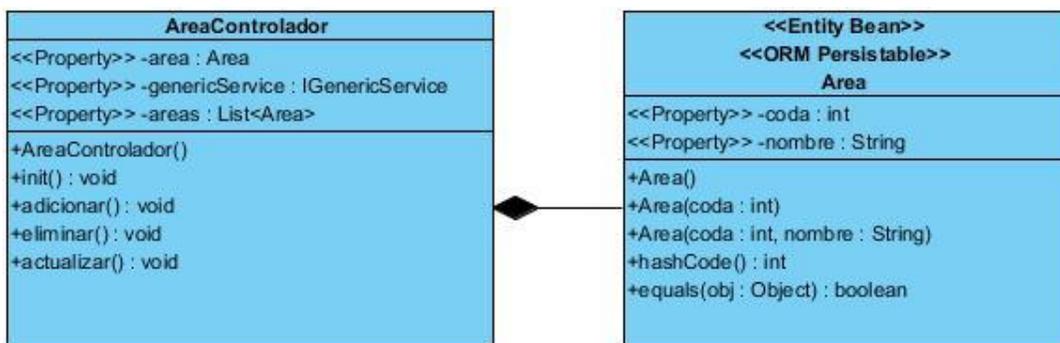


Figura 11 Ejemplo de empleo del patrón controlador (elaboración propia)

- **Experto:** se basa en que se le asigne la responsabilidad de la creación de un objeto o la implementación de un método a la clase que conoce toda la información necesaria para crearlo.

Ejemplo de lo expuesto anteriormente es la clase **AspiranteDao** la que se especializa en aplicar cambios en la base de datos.



Figura 12 Ejemplo del uso del patrón experto (elaboración propia)

- **Controlador:** el patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal manera que es el que recibe todas las peticiones y ejecuta en las diferentes clases el método solicitado. Ejemplo de dicho patrón son las clases **AreaControlador**, **AspiranteControlador**, **CargoControlador**, encargadas de desencadenar eventos según las peticiones que se realicen al sistema, separando la capa de presentación de la lógica de negocio, lo cual permite una mayor reutilización del código.

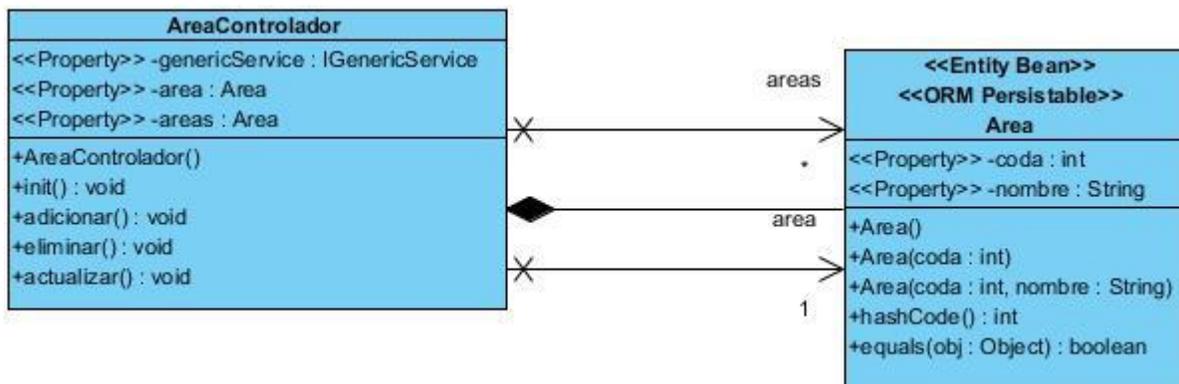


Figura 13 Ejemplo del uso del patrón controlador (elaboración propia)

- **Alta cohesión:** el patrón define que la información que almacena una clase debe de ser coherente y estar en mayor medida relacionada con la clase.
- **Bajo acoplamiento:** es utilizado con el objetivo de tener las clases del sistema con la menor dependencia entre ellas. De tal manera que, en caso de producirse alguna modificación, el impacto sea el mínimo posible en el resto de clases, potenciando la reutilización.

## Patrones GOF

Dentro de los patrones de diseño se encuentran los patrones *Gang of Four* o Grupo de los Cuatro. Estos patrones definen el comportamiento entre las clases y los objetos. Los patrones de diseño GOF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Gamma, Helm et al. 1994). Dentro de ellos se identificaron los siguientes como los más importantes empleados durante el desarrollo de la aplicación.

- **Singleton:** garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. El patrón de diseño *singleton* (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Ejemplo de esto es la inyección de dependencia que proporciona Spring al sistema. Todas las clases controladoras tienen una única instancia de la clase `IGenericService`.

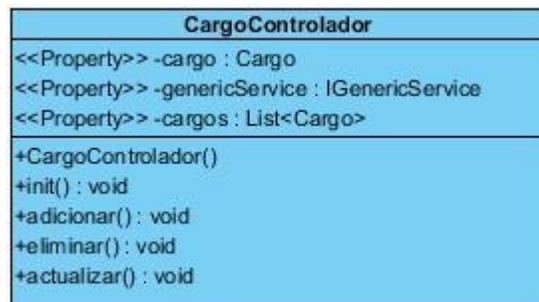


Figura 14 Ejemplo del uso del patrón singleton (elaboración propia)

- **DAO:** encapsula el acceso a la base de datos, por lo que cuando la capa de lógica de negocio necesite interactuar con la base de datos, va a hacerlo a través de la API que le ofrece DAO. Generalmente esta API consiste en métodos CRUD (*Create, Read, Update* y *Delete*). Ejemplo de lo planteado se refleja en el diagrama de la figura 11.

### Diagrama de clases

Un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama que describe la estructura estática de un sistema y muestra las clases que lo conforman, sus atributos, operaciones (o métodos) y las relaciones de colaboración entre estas. El diagrama que a continuación se muestra a modo de ejemplo, ilustra las clases de diseño que son necesarias para gestionar una de las entidades de negocio de la aplicación bajo la arquitectura definida para la misma.

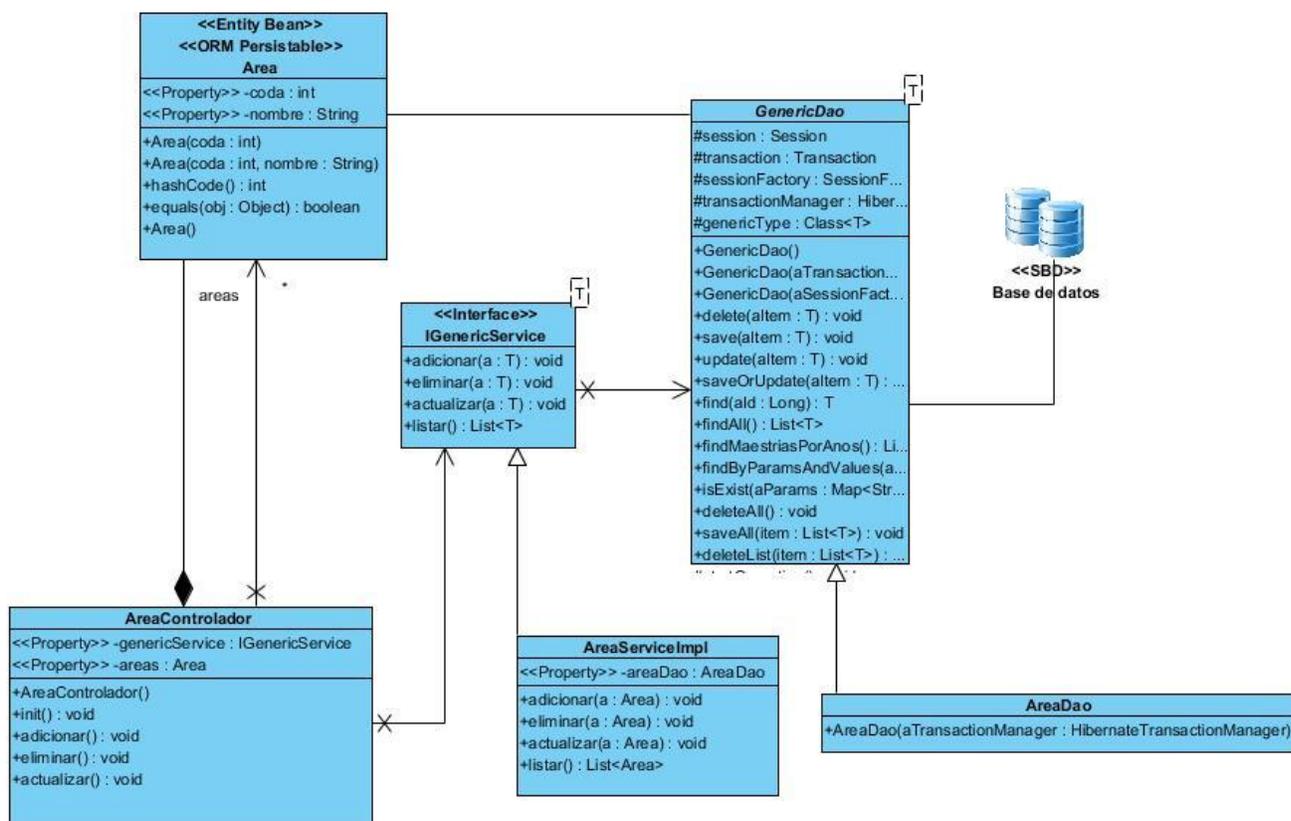


Figura 15 Diagrama de clases del gestionar Área (elaboración propia)

El diagrama anterior muestra las relaciones entre las clases que intervienen en el proceso de gestionar áreas de la institución. La clase controladora **AreaControlador** actualiza la entidad **Area** con los datos proporcionados por la vista y delega a la clase **AreaServiceImpl**, que es una implementación de la interfaz **IGenericService**, las diferentes acciones que ejecuta el usuario. Esta a su vez se encarga de aplicar la lógica del negocio haciendo uso de la clase **GenericDao** y de su especificación **AreaDao**, necesarias para acceder a los datos almacenados en el servidor de base de datos.

## **Conclusiones del capítulo**

Luego de realizada la planificación del desarrollo, así como el análisis, diseño e implementación del software, se arribaron a las siguientes conclusiones:

- Como resultado del proceso de obtención de requisitos, se identificaron los indicadores que satisfacen las necesidades informativas para el control del proceso de formación posgraduada de la UCI, con énfasis en la formación doctoral.
- Mediante el empleo adecuado de los métodos de desarrollo de software, patrones de diseño, herramientas y las tecnologías seleccionadas, se diseñó e implementó el tablero de control, que facilita la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada de la UCI.

## **CAPÍTULO 3 Evaluación de los resultados**

### **Pruebas de software**

Las pruebas de software consisten en la verificación del comportamiento de un programa en un conjunto finito de casos de prueba debidamente seleccionados. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; para probar el comportamiento del mismo (Muller, Beer et al. 2010).

Para realizar el proceso de pruebas es necesario el análisis y diseño de las mismas. En esta actividad los objetivos de las pruebas generales se transforman en condiciones de pruebas y casos de pruebas tangibles. Para ello se ejecutaron las siguientes tareas:

- Revisar la base de pruebas, los informes de análisis de riesgos, la arquitectura, el diseño y las especificaciones de interfaz.
- Evaluar la “testabilidad” de la base de pruebas y de los objetos de pruebas.
- Identificar y priorizar las condiciones de prueba en base al análisis de los elementos, la especificación, el comportamiento y la estructura del software.
- Diseñar y priorizar los casos de prueba.
- Identificar los datos de pruebas necesarios para soportar las condiciones y los casos de pruebas.

Teniendo en cuenta lo anterior, para evaluar la calidad del producto obtenido, se realizaron pruebas de caja negra, de caja blanca, pruebas funcionales y de aceptación. Las mismas son descritas a continuación.

### ***Pruebas de caja blanca***

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Las pruebas de caja blanca se desarrollaron aplicando la técnica del camino básico, la que 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 de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa (Pressman 2010).

Para realizar esta técnica es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se ejemplifica la estrategia de pruebas de caja blanca. Para la misma se emplea un segmento de código de la aplicación desarrollada mediante el cual se obtienen los doctorados defendidos por área. En este se enumeran las sentencias de código como indica el procedimiento.

```
private BarChartModel initBarModelDoctoradosDefPorArea() {  
    BarChartModel model = new BarChartModel();           1  
    int contadorArea = 0;                                 1  
    List<Aspirante> aspiranteTemp = this.getAspirantes(); 1  
    List<String> listAreas = new ArrayList();             1  
    for (Aspirante aspirant : aspiranteTemp) {           2  
        if (aspirant.getTipoPosgrado().getCodtp() == 2) { 3  
            listAreas.add(aspirant.getArea().getNombre()); 4  
        }  
    }  
    Collections.sort(listAreas);5  
    ChartSeries areas = new ChartSeries();5  
    areas.setLabel("Áreas");5  
    for (int i = 0; i < listAreas.size(); i++) {6  
        String var = listAreas.get(i);7  
        for (String area : listAreas) {8  
            if (var.equals(area)) {9  
                contadorArea++; 10  
            }  
        }  
        areas.set(var, contadorArea); 11  
        if (this.contadorAreaMayor < contadorArea) { 12  
            this.contadorAreaMayor = contadorArea; 13  
        }  
    }  
}
```

```

contadorArea = 0; 14
for (int h = 0; h < listAreas.size(); h++) { 15
    if (listAreas.get(h).equals(var)) { 16
        listAreas.remove(listAreas.get(h)); 17
        h--; 18
    }
}
i--; 19
}
model.addSeries(areas); 20
return model; 21
}
    
```

La siguiente figura describe el flujo del camino básico del código seleccionado para ejemplificar la estrategia de prueba de caja blanca seguida.

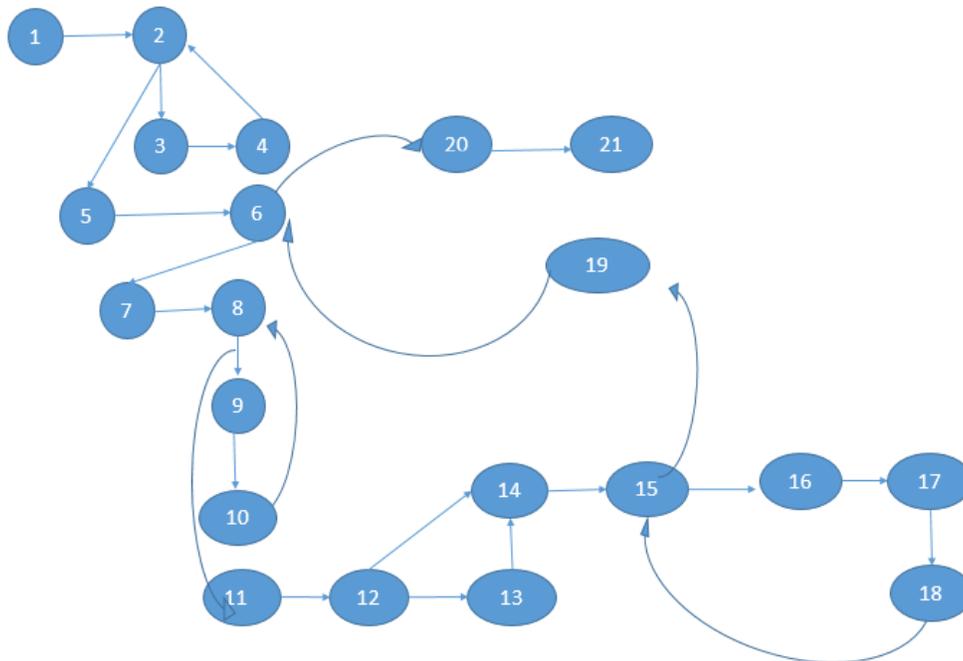


Figura 16 Flujo del camino básico para el código numerado (elaboración propia)

Fórmulas para calcular la complejidad ciclomática:

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

Donde "A" es la cantidad de aristas y "N" la cantidad de nodos.

$$V(G) = (25 - 21) + 2$$

$$V(G) = 6$$

$$V(G) = P + 1$$

Siendo "P" la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = 5 + 1$$

$$V(G) = 6$$

$$V(G) = R$$

Donde "R" representa la cantidad de regiones en el grafo.

$$V(G) = 6$$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de 6, lo que significa que existen seis posibles caminos por donde el flujo puede circular. Este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Tabla 13. Caminos básicos para ejemplo de prueba de caja blanca

Número	Caminos Básicos
1	1-2-5-6-7-8-11-12-13-14-15-19-6-20-21
2	1-2-5-6-20-21
3	1-2-5-6-7-8-11-12-14-15-19-6-20-21
4	1-2-5-6-7-8-9-10-8-11-12-13-14-15-16-17-18-15-19-6-20-21
5	1-2-3-4-2-5-6-7-8-9-10-8-11-12-13-14-15-16-17-18-15-19-6-20-21
6	1-2-5-6-7-8-9-10-8-11-12-14-15-16-17-18-15-19-6-20-21

Posteriormente de haber determinado los caminos básicos se procede a ejecutar los casos de pruebas para cada uno de estos. Para definir los casos de prueba fue necesario tener en cuenta los siguientes elementos:

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

Caso de prueba para el camino básico # 1.

**Descripción:** En este camino no se logra obtener una lista de áreas referentes a los aspirantes que cursan un doctorado.

**Condición de ejecución:**

**Entrada:**

-complejidad=Media, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que se construya el grafico de barras referente a los doctorados defendidos por áreas.

**Resultado obtenido:** No satisfactorio.

Caso de prueba para el camino básico # 2.

**Descripción:** En este camino se declaran las variables y se construye un gráfico nulo.

**Condición de ejecución:**

**Entrada:**

-complejidad=Baja, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que se construya el gráfico de barras referente a los doctorados defendidos por áreas.

**Resultado obtenido:** No satisfactorio.

Caso de prueba para el camino básico # 3.

**Descripción:** En este camino se construye un gráfico nulo.

**Condición de ejecución:**

**Entrada:**

-complejidad=Alta, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que se construya el grafico de barras referente a los doctorados defendidos por áreas.

**Resultado obtenido:** No satisfactorio.

Caso de prueba para el camino básico # 4.

**Descripción:** En este camino se construye un gráfico nulo.

**Condición de ejecución:**

**Entrada:**

-complejidad=Alta, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que la lista de áreas obtuviera valor.

**Resultado obtenido:** No satisfactorio.

Caso de prueba para el camino básico # 5.

**Descripción:** En este camino se construye un gráfico con al menos un valor.

**Condición de ejecución:**

**Entrada:**

-complejidad=Alta, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que se construya el grafico correctamente.

**Resultado obtenido:** Satisfactorio.

Caso de prueba para el camino básico # 6.

**Descripción:** En este camino se construye un gráfico nulo.

**Condición de ejecución:**

**Entrada:**

-complejidad=Alta, representa la complejidad del camino seleccionado.

**Resultado esperado:** Se espera que se construya el gráfico correctamente.

**Resultado obtenido:** No Satisfactorio.

Después de haber ejecutado las pruebas de caja blanca aplicando el método del camino básico para el segmento de código seleccionado, se obtuvieron los siguientes resultados:

- Se determinó que la complejidad ciclomática del código seleccionado es de 6, lo que representa el límite mínimo de casos de prueba para la historia de usuario.
- En una primera iteración de los 6 casos solo uno (el caso 5) fue satisfactorio porque por este camino se logra construir el gráfico (doctorados defendidos por áreas) con al menos un valor.
- En una segunda iteración, luego de corregir el código e inconsistencias en los juegos de datos todos los caminos tuvieron un resultado satisfactorio.
- La misma estrategia fue empleada para todas las pruebas de caja blanca ejecutadas en esta fase del desarrollo del software. En general se requirieron dos iteraciones para probar las historias de usuario del sistema.

### ***Pruebas de caja negra***

Las pruebas de caja negra permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos (Pressman 2010).

La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la caja blanca.

A continuación se muestra el caso de prueba del requisito “gestionar institución”, para ilustrar la estrategia seguida con las pruebas de caja negra.

Tabla 16. Caso de prueba de ejemplo para mostrar estrategia de pruebas de caja negra

Escenarios	Descripción	Respuesta del sistema	Flujo central
<b>EC</b> 1.1 Gestionar institución. Insertar Institución correctamente.	El usuario realiza el proceso de inserción correctamente.	5. El sistema adiciona la institución y muestra un mensaje de alerta “La institución ha sido adicionada satisfactoriamente”. 6. El sistema adiciona a la tabla de instituciones la institución.	1. El usuario introduce el nombre de la institución en el campo ^Nombre ^ 2. El usuario hace clic en la lista desplegable de países. 3. El usuario selecciona un país. 4. El usuario hace clic en el botón "Adicionar".

<p><b>EC 1.2</b> Insertar institución incorrectamente.</p>	<p>El usuario realiza el proceso de inserción incorrectamente.</p>	<p>2. El sistema coloca los campos nombre y país en rojo y muestra un mensaje de alerta: “Debe introducir el nombre de la institución “y “Debe seleccionar el país de la institución “respectivamente.</p>	<p>1. El usuario hace clic en el botón adicionar.</p>
<p><b>EC 1.3</b> Cancelar inserción de la institución.</p>	<p>El usuario realiza el proceso de cancelar la inserción de la institución.</p>	<p>2. El sistema limpia los campos.</p>	<p>1. El usuario hace clic en el botón cancelar.</p>
<p><b>EC 2.0</b> Eliminar institución correctamente.</p>	<p>El usuario realiza el proceso de eliminar correctamente.</p>	<p>2. El sistema muestra la ventana eliminar institución con las opciones Si/No.  4. El sistema elimina la institución y cierra la ventana.  4.1El sistema cierra la ventana y no elimina la institución.</p>	<p>1. El usuario hace clic en el icono “eliminar” que se encuentra en la columna acciones de la tabla de institución.  3. El usuario selecciona la opción de que sí.  3.1 El usuario selecciona la opción de que no.</p>

<p><b>EC 2.1</b> Eliminar institución incorrectamente.</p>	<p>El usuario realiza el proceso de eliminar incorrectamente.</p>	<p>2.El sistema muestra la ventana eliminar institución con las opciones Si/No.</p> <p>4.El sistema no elimina la institución y cierra la ventana.</p> <p>4.1 El sistema cierra la ventana.</p>	<p>1. El usuario hace clic en el icono “eliminar” que se encuentra en la columna acciones de la tabla de institución.</p> <p>3. El usuario selecciona la opción de que sí.</p> <p>3.1. El usuario selecciona la opción de que no.</p>
<p><b>EC 3.0</b> Actualizar Institución.</p>	<p>El usuario realiza el proceso de actualizar correctamente.</p>	<p>2.El sistema muestra la ventana actualizar Institución que contiene los campos nombre y país con los datos de la institución seleccionada.</p> <p>4.El sistema actualiza los datos de la entidad institución .</p>	<p>1.El usuario hace clic en el icono “Actualizar” que se encuentra en la columna acciones de la tabla de institución.</p> <p>3.El usuario actualiza al menos un dato de la institución y da clic en el botón actualizar.</p>
<p><b>EC 4.0</b> Cerrar</p>	<p>El usuario realiza el proceso de</p>		<p>1.El usuario hace clic en el botón cerrar.</p>

	cerrar la sesión de gestionar institución.	2.El sistema cierra la sesión del gestionar institución.	
--	--	--	--

### Resultados obtenidos en la prueba de caja negra

Se realizaron tres iteraciones de las pruebas de caja negra con la utilización de casos de pruebas, para comprobar que el sistema funciona correctamente y si está adecuado a los requerimientos del cliente. Durante la primera iteración se encontraron cinco no conformidades, una de prioridad alta y cuatro de prioridad baja. La no conformidad de prioridad alta fue con respecto a la construcción y persistencia de las entidades en la base de datos y las bajas estuvieron relacionadas con mal funcionamiento de mensajes de confirmación. Todas fueron resueltas por la desarrolladora de la aplicación. Se realizó una segunda iteración, en la cual se encontraron no conformidades con respecto a la validación de formularios. Se realizó una tercera iteración donde no se encontraron no conformidades, evaluándose de esta manera todos los casos de prueba como satisfactorios.

En general, los resultados de las pruebas de caja negra realizadas a toda la aplicación se resumen en la siguiente tabla.

Tabla 17. Resultados de las pruebas de caja negra realizadas a toda la aplicación

No Conformidades	Iteración		
	1	2	3
Funcionalidad	12	4	0
Interfaz	15	5	0
Validación	8	3	0

### **Pruebas de aceptación**

Las pruebas de aceptación son a menudo responsabilidad de los clientes o usuarios. Tienen como objetivo crear confianza con el sistema; aunque su objetivo principal es el de no localizar defectos. Evalúan la buena disposición de un sistema para su despliegue y uso (Sommerville 2005).

Las pruebas de aceptación poseen gran importancia ya que permiten confirmar que las historias de usuario han sido implementadas correctamente al final de cada iteración. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. A continuación se muestran los datos que se recogen en las pruebas de aceptación:

- **Iteración:** número de la iteración realizada.
- **Nombre del rasgo a probar:** nombre del rasgo al que se le realiza la prueba.
- **No. del rasgo a probar:** número del rasgo al que se le realiza la prueba.
- **Descripción del rasgo a probar:** descripción breve del rasgo que se prueba.
- **Precondiciones:** condiciones necesarias para poder realizar el caso de prueba y obtener los resultados esperados.
- **Pasos:** pasos lógicos a seguir durante el desarrollo de la prueba para la obtención del resultado esperado.
- **Resultados esperados:** descripción breve de los resultados esperados luego de realizar la prueba.
- **Evaluación:** clasificación de la prueba dependiendo de la comparación del resultado obtenido con el resultado esperado.

Para mostrar la estrategia seguida durante las pruebas de aceptación y los resultados alcanzados, se exponen tres historias de usuario a modo de ejemplo y se describen las pruebas realizadas.

Diseño del caso de prueba: mostrar información consolidada.

Tabla 18. Iteración 1, caso de prueba de aceptación para el RF6 (maestrías)

<b>Iteración No.: 1</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a las maestrías defendidas.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> Muestra el gráfico referente a las maestrías defendidas por año.	
<b>Precondiciones:</b> El usuario debe estar previamente registrado y autenticado en el sistema.	
<b>Pasos:</b> El usuario accede a la sección de maestrías.	
<b>Resultados esperados:</b> Se muestra un gráfico que contempla la cantidad de maestrías que son defendidas por año.	
<b>Evaluación:</b> <i>Prueba no satisfactoria.</i>	
<b>Observaciones:</b> <i>La cantidad de maestrías por año no refleja el valor correcto.</i>	

Tabla 19. Iteración 2, caso de prueba de aceptación para el RF6 (maestrías)

<b>Iteración No.: 2</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a las maestrías defendidas.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> Muestra el gráfico referente a las maestrías defendidas por año.	
<b>Precondiciones:</b> El usuario debe estar previamente registrado y autenticado en el sistema.	
<b>Pasos:</b> El usuario accede a la sección de maestrías.	
<b>Resultados esperados:</b> Se muestra un gráfico que contempla la cantidad de maestrías que son defendidas por año.	
<b>Evaluación:</b> <i>Prueba parcialmente satisfactoria.</i>	
<b>Observaciones:</b> <i>La cantidad de maestrías por año refleja el valor correcto pero los años no estaban ordenados ascendentemente.</i>	

Tabla 20. Iteración 3, caso de prueba de aceptación para el RF6 (maestrías)

<b>Iteración No.: 3</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a las maestrías defendidas.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> <i>Muestra el gráfico referente a las maestrías defendidas por año.</i>	
<b>Precondiciones:</b> <i>El usuario debe estar previamente registrado y autenticado en el sistema.</i>	
<b>Pasos:</b> <i>El usuario accede a la sección de maestrías.</i>	
<b>Resultados esperados:</b> <i>Se muestra un gráfico que contempla la cantidad de maestrías que son defendidas por año.</i>	
<b>Evaluación:</b> <i>Prueba satisfactoria.</i>	
<b>Observaciones:</b> <i>La cantidad de maestrías por año refleja el valor correcto y los años están ordenados ascendentemente.</i>	

Tabla 21. Iteración 1, caso de prueba de aceptación para el RF6 (doctorados)

<b>Iteración No.: 1</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a los doctorados defendidos por área.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> <i>Muestra el gráfico referente a los doctorados defendidos por área.</i>	
<b>Precondiciones:</b> <i>El usuario debe estar previamente registrado y autenticado en el sistema.</i>	
<b>Pasos:</b> <i>El usuario accede a la sección de doctorados.</i>	
<b>Resultados esperados:</b> <i>Se muestra un gráfico que contempla la cantidad de doctorados defendidos por área.</i>	
<b>Evaluación:</b> <i>Prueba no satisfactoria.</i>	
<b>Observaciones:</b> <i>La cantidad de doctorados defendidos por área no corresponden con los valores esperados.</i>	

Tabla 22. Iteración 2, caso de prueba de aceptación para el RF6 (doctorados)

<b>Iteración No.: 2</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a los doctorados defendidos por área.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> <i>Muestra el gráfico referente a los doctorados defendidos por área.</i>	
<b>Precondiciones:</b> <i>El usuario debe estar previamente registrado y autenticado en el sistema.</i>	
<b>Pasos:</b> <i>El usuario accede a la sección de doctorados.</i>	
<b>Resultados esperados:</b> <i>Se muestra un gráfico que contempla la cantidad de doctorados defendidos por área.</i>	
<b>Evaluación:</b> <i>Prueba parcialmente satisfactoria.</i>	
<b>Observaciones:</b> <i>No se muestran la cantidad de áreas esperadas.</i>	

Tabla 23. Iteración 3, caso de prueba de aceptación para el RF6 (doctorados)

<b>Iteración No.: 3</b>	
<b>Nombre del requisito a probar:</b> <i>Mostrar información referente a los doctorados defendidos por área.</i>	<b>Número del requisito a probar:</b> <i>RF6</i>
<b>Descripción del requisito a probar:</b> <i>Muestra el gráfico referente a los doctorados defendidos por área.</i>	
<b>Precondiciones:</b> <i>El usuario debe estar previamente registrado y autenticado en el sistema.</i>	
<b>Pasos:</b> <i>El usuario accede a la sección de doctorados.</i>	
<b>Resultados esperados:</b> <i>Se muestra un gráfico que contempla la cantidad de doctorados defendidos por área.</i>	
<b>Evaluación:</b> <i>Prueba satisfactoria.</i>	
<b>Observaciones:</b> <i>Se muestra la información deseada.</i>	

Tabla 24. Iteración 1, caso de prueba de aceptación para el RF9

<b>Iteración No.:1</b>	
<b>Nombre del requisito a probar:</b> Gestionar Institución	<b>Número del requisito a probar:</b> RF9
<b>Descripción del requisito a probar:</b> Muestra un formulario y una tabla que permiten añadir y visualizar las instituciones respectivamente.	
<b>Precondiciones:</b> El usuario debe estar previamente registrado y autenticado en el sistema.	
<b>Pasos:</b> 1. El usuario accede a la sección de Instituciones.	
<b>Resultados esperados:</b> Se espera que los datos de la institución añadida sean reflejados en la tabla de instituciones.	
<b>Evaluación:</b> Prueba parcialmente satisfactoria.	
<b>Observaciones:</b> La institución es añadida correctamente pero no se refleja en la tabla de instituciones.	

Tabla 25. Iteración 2, caso de prueba de aceptación para el RF9

<b>Iteración No.:2</b>	
<b>Nombre del requisito a probar:</b> Gestionar Institución	<b>Número del requisito a probar:</b> RF9
<b>Descripción del requisito a probar:</b> Muestra un formulario y una tabla que permiten añadir y visualizar las instituciones respectivamente.	
<b>Precondiciones:</b> El usuario debe estar previamente registrado y autenticado en el sistema.	
<b>Pasos:</b> 1. El usuario accede a la sección de Instituciones.	
<b>Resultados esperados:</b> Se espera que los datos de la institución añadida sean reflejados en la tabla de instituciones.	
<b>Evaluación:</b> Prueba satisfactoria.	
<b>Observaciones:</b> La institución es añadida correctamente y es reflejada en la tabla de instituciones.	

La siguiente gráfica resume los resultados de las pruebas de aceptación seleccionadas para ejemplificar la estrategia seguida.

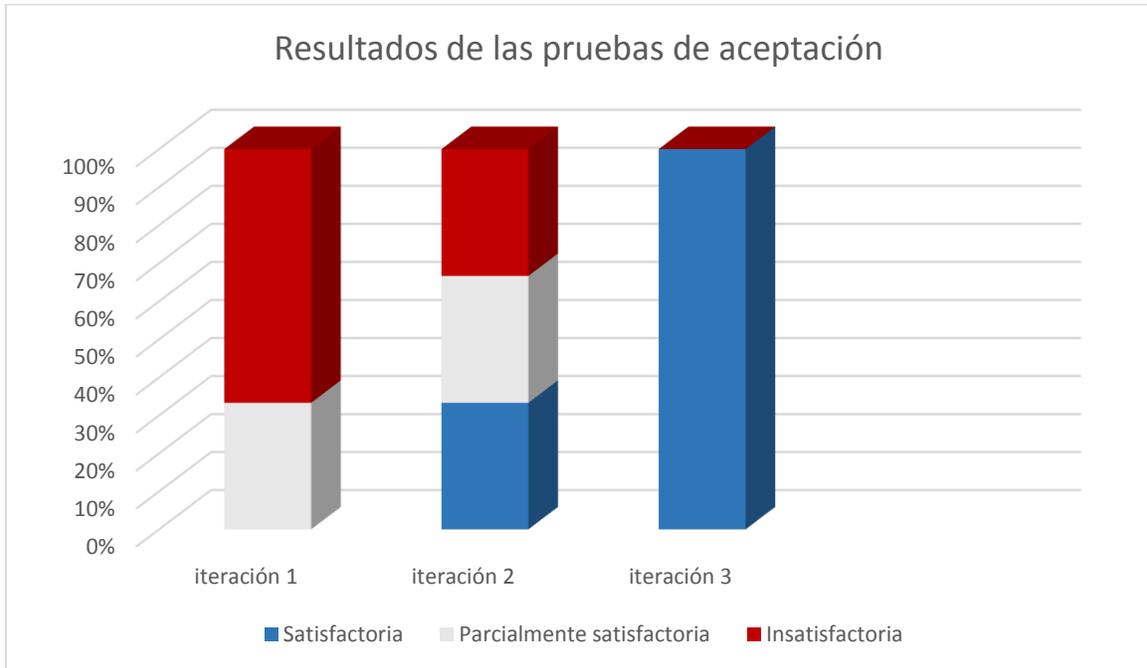


Figura 17 Resultados de las pruebas de aceptación para los requisitos seleccionados a modo de ejemplo (elaboración propia)

En general, para toda la aplicación se requirieron tres iteraciones para que la misma fuera aceptada por el cliente final.

## Aplicación obtenida

A continuación se ejemplifica mediante algunas interfaces de usuario la aplicación desarrollada. La misma quedó totalmente funcional, por lo que resulta una solución útil para la Dirección de Posgrado de la UCI, la que con su empleo fortalece sus mecanismos de control.



Figura 18. Ejemplo de interfaz de actualización de datos (elaboración propia)

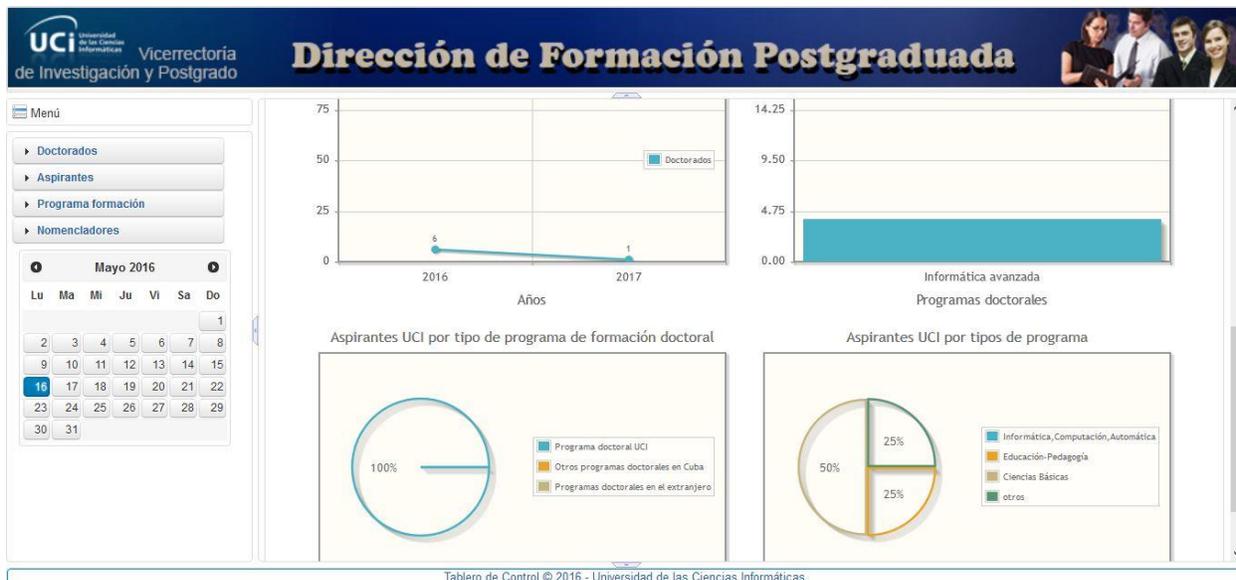


Figura 19. Ejemplos de gráficos que conforman el tablero de control (elaboración propia)

## **Evaluación de los resultados alcanzados**

La forma en que la aplicación informática desarrollada registra y consolida los datos del proceso de formación posgraduada de la UCI, contribuye a solucionar las insuficiencias inicialmente detectadas teniendo en cuenta que:

- La información ahora está disponible mediante una aplicación web que interactúa con un servidor de bases de datos, lo que facilita el acceso para su actualización y consulta.
- Varias personas con el adecuado control de acceso pueden actualizar la información, por lo que el registro de la misma es posible lograrlo en el momento oportuno y mantener así los indicadores actualizados.
- Se maneja consistentemente la información que está relacionada, por lo que no es necesario introducir información redundante.
- El amigable diseño de interfaz facilita el uso de la aplicación a sus potenciales usuarios.

Las mejoras anteriores inciden favorablemente a la adecuada gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada en la UCI, lo que incrementa el control sobre dicho proceso y la efectividad en la toma de decisiones.

## **Conclusiones del capítulo**

Luego de realizadas el conjunto de pruebas de calidad al producto de software obtenido, se arribaron a las siguientes conclusiones:

- Los resultados alcanzados durante la fase de pruebas demostraron que se obtuvo un software confiable que satisface los requisitos planteados para el tablero de control.
- Con el producto obtenido es factible gestionar los indicadores que evalúan el desempeño del proceso de formación posgraduada en la UCI.
- Los potenciales usuarios del tablero de control mostraron satisfacción con el software desarrollado una vez concluidas las pruebas de aceptación.

## **CONCLUSIONES**

1. El tablero de control constituye una herramienta eficaz para el monitoreo y control de los indicadores fundamentales que marcan la salud del proceso de formación posgraduada de la UCI y su empleo contribuye a tomar decisiones objetivas basadas en datos.
2. Mediante la adecuada selección y empleo de métodos, tecnologías y herramientas, se desarrolló para la Dirección de Posgrado de la UCI, un tablero de control desplegable en la web, que mejora la gestión de los indicadores que evalúan el desempeño del proceso de formación posgraduada y contribuye a perfeccionar su control.
3. El conjunto de pruebas de calidad a que fue sometido el tablero de control desarrollado, avaló que se obtuvo un software confiable, que satisface los requisitos planteados, es aplicable en su versión actual y satisface a sus potenciales usuarios.

## **RECOMENDACIONES**

Para darle continuidad al presente trabajo se recomienda lo siguiente:

1. Emplear el tablero de control desarrollado para monitorear el proceso de formación posgraduada de la UCI.
2. Contrastar el valor de los indicadores del tablero de control, con los objetivos estratégicos de la UCI en el área de posgrado propuestos para cada año, como umbral que permite la evaluación cualitativa del proceso.
3. Integrar el tablero de control desarrollado al Sistema de Gestión Universitaria de la UCI, para acceder a datos que este último almacena y que son de interés para indicadores manejados por el tablero de control.

## REFERENCIAS BIBLIOGRÁFICAS

Amaya, J. A. (2005). Gerencia: Planeación y Estrategia, Universidad Santo Tomás de Aquino. ISBN: 958-97305-8-2.

Apache Software Foundation (2016). "Apache Tomcat." Fecha consulta 31 de marzo 2016, from <http://tomcat.apache.org/>.

Bauer, C. and G. King (2007). Java Persistence with Hibernate. New York, Manning Publications Co. ISBN: 1-932394-88-5.

Caralt, J. C. and J. C. Díaz (2011). Introducción al Business Intelligence. Barcelona, Editorial UOC. ISBN: 978-84-9788-886-8.

Cerda, J. R. S. and A. A. Chacur (2006). "Sistemas de control de gestión, análisis para organizaciones sin fines de lucro." Revista Ingeniería Industrial vol 5, núm 1, p. 61-76. ISSN: 0717-9103

Çivici, Ç. (2014). "PrimeFaces User's Guide 5.0." from [http://www.primefaces.org/docs/guide/primefaces\\_user\\_guide\\_5\\_0.pdf](http://www.primefaces.org/docs/guide/primefaces_user_guide_5_0.pdf).

Cortés, A. P., Implementación de Tableros de Control en el área de mejora continua en una empresa de manufacura. Tesis de maestría. Tutor J. B. P. Victorino. 2010, Instituto Tecnológico de Puebla, Puebla, México.

Cruz, B. C. d. I., D. J. H. González, et al., Tablero de Control. Tesis de Grado. Tutor A. P. d. Lira. Desarrollada en Escuela Superior de Comercio y Administración Santo Tomás, 2010, Instituto Politécnico Nacional, Ciudad de México.

Edeki, C. (2013). "Agile Unified Process." International Journal of Computer Science and Mobile Applications vol 1, núm 3, p. 13-17. ISSN: 2321-8363

Gamma, E., R. Helm, et al. (1994). Design patterns, software engineering, object-oriented programming, Addison-Wesley. ISBN: 0-201-63361-2.

Hennebrueder, S. (2013). "Guide to Java Persistence and Hibernate." from <http://www.laliluna.com/download/java-persistence-developer-guide.pdf>.

Java Code Geeks (2015). "JSF 2.0 Programming Cookbook." from <https://www.javacodegeeks.com/wp-content/uploads/2015/05/JSF-2.0-Programming-Cookbook.pdf>.

Johnson, R. (2002). Expert One-on-One J2EE Design and Development, Wrox. ISBN: 978-0764543852.

Johnson, R., J. Hoeller, et al. (2015). "Spring Framework Reference Documentation. 4.2.6.Release." from <http://docs.spring.io/spring/docs/current/spring-framework-reference/pdf/spring-framework-reference.pdf>.

Johnson, R., J. Hoeller, et al. (2015). "Spring Framework Reference Documentation. 4.3.0.RELEASE."

Kaplan, R. S. (2010). Conceptual Foundations of the Balanced Scorecard, Harvard Business School, Harvard University.

Keen, M., R. Coutinho, et al. (2012). "Developing Web Applications using Java Server Faces." from <http://www.redbooks.ibm.com/redpapers/pdfs/redp4883.pdf>.

Larman, C. (2004). UML Y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado, Prentice Hall. ISBN: 8420534382.

Matiz, D. A. S. (2011). "Teoría de indicadores de gestión y su aplicación práctica." Revista Gestión Integral en Ingeniería Neogranadina vol 3, núm 2. ISSN: 2145-5759

Muller, T., A. Beer, et al. (2010). Programa de estudio de nivel básico para probadores certificados, International Software Testing Qualifications Board.

Nogueira, D., A. Medina, et al. (2004). Fundamentos para el Control de la Gestión Empresarial. La Habana, Pueblo y Educación. ISBN: 959-13-1192-3.

Orizondo, A. C. A., Modelo para evaluar los resultados de la actividad de desarrollo producción de la Universidad de las Ciencias Informáticas. Tesis de Diplomado. Tutor J. Lavandero. Desarrollada en Centro de Estudio de Técnicas de Dirección, 2015, Facultad de Ingeniería Industrial. Instituto Superior Jose Antonio Echeverría, La Habana.

Páez, F. (2016). "Cuadro de Mando Integral." from <http://www.cmigestion.es/servicios/consultoria-estrategica/cuadro-de-mando-integral/>.

Peñalver, P. (2016). "Cómo implementar un cuadro de mando integral en tu negocio." from <http://blog.wearemarketing.com/como-implementar-un-cuadro-de-mando-integral-en-tu-negocio>.

Pressman, R. S. (2010). Ingeniería del software. Un enfoque práctico, McGrawHill, 7ma Edición. ISBN: 978-607-15-0314-5.

Quántico (2016). "Tablero de Control." from <http://www.quanticotrends.com/funcionalidades-monitor/dashboard-monitoreo-social/>.

Ramírez, C. A. C., Tablero de Control para un Proceso del Negocio en una Compañía de Seguros. Tesis de Grado. Tutor P. G. L. Giraldo. 2009, Universidad Nacional de Colombia, Medellín.

Sánchez, E., P. O. L. Torres, et al. Mejorando la gestión de historias de usuario en eXtreme programming. Jornadas de Ingeniería del Software y Bases de Datos (JISBD), celebrado en Alicante (12 al 14 de noviembre, 2003). ISBN: 84-688-3836-5

Sommerville, I. (2005). Ingeniería del Software, Pearson, 7ma Edición. ISBN: 978-847-82-9074-1.

Taylor, E. R., M. C. Benavides, et al. (2012). Guía para la construcción de indicadores de gestión. Bogotá, Departamento Administrativo de la Función Pública (DAFP). ISBN: 978-958-8125-57-2.

The PostgreSQL Global Development Group (2016). "PostgreSQL 9.3.11 Documentation." from <http://www.postgresql.org/files/documentation/pdf/9.3/postgresql-9.3-A4.pdf>.

Topi, H., J. S. Valacich, et al. (2010). Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, ACM & AIS.

Winterfeldt, D. (2013). "Spring by Example. Versión 1.3." from <http://www.springbyexample.org/>.

## ANEXOS

### Anexo 1. Aval de la Dirección de Posgrado de la UCI



Dirección de Formación Posgraduada  
Vicerrectoría de Investigación y Posgrado

La Habana, 22 de junio de 2016

Por medio del presente documento avalamos la utilidad de la aplicación informática desarrollada "Tablero de Control", para la gestión de los principales indicadores que maneja la Dirección de Formación Posgraduada de la UCI. De igual forma expresamos conformidad con las funcionalidades implementadas en su versión actual, la que será puesta en práctica de inmediato para emitir los informes regularmente solicitados a nuestra dirección.

Para que así conste firmamos,

Dr.C. Carlos Eulalio Novo Soto  
Metodólogo



MSc. Dainys Gainza Reyes  
Metodóloga

Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, municipio de La Lisa. La Habana, Cuba.