

Universidad de las Ciencias Informáticas

Conceptualización e implantación de un Laboratorio Industrial de  
Pruebas de Software.

Tesis en opción al título de  
Máster en Ciencias

**Por**

**Autor: Ing. Tayché Capote García**

**Tutores: Dra. Ailyn Febles Estrada**

**MSc. Ramsés Delgado Martínez**

**La Habana, Cuba**

Junio de 2011

## **Agradecimientos**

A papi, que aunque no está físicamente lo siento bien cerca de mi todos los días.

A mami, por ser mi razón de ser, mi inspiración. Gracias por junto a papi darme la vida y guiar cada paso que doy, los adoro.

A mis primos-hermanos Thais, Jeily y Jorgito, por sus sonrisas y malcriadeces.

A nana, piti y yoyo, por sus enseñanzas y por estar siempre para mí.

A Guille (mi Tatica), por su confianza, ayuda e incondicionalidad, gracias por tanto amor, sin ti no hubiera llegado a la meta.

A mi familia de Güira (Maikel, Isa, Guillermo, Pilar y Anaida), por quererme y ayudarme tanto.

A los chicos del LIPS y el GIPS, de cada uno de ustedes hay un pedacito en esta tesis, gracias por confiar en esta idea loca y por ser un colectivo único, excepcional, como ninguno.

A mi tutora y amiga Ailyn, sin tu impulso final y tus ideas esta tesis no se hubiera concretado, gracias por enseñarme cada día a ser mejor profesional.

A Ramsés (Raki) por ser mi super tutor, por soportar mis despistes y tener tanta paciencia conmigo y con mi Haier...jajaja, por ser amigo.

A los amigos de siempre, que aunque estén un poco lejos a veces, siempre están en mi corazón: la Negra, Yeni, Nely, Lídice, Pepe, el Negro, Pepote, Irina C (y el sobrino), Geiser, Yoan...

A Alicia por ser mi segunda madre, por aconsejarme y quererme tanto.

A la profe Alina, por ser un ejemplo a seguir para mí y su cariño de siempre.

A los Doctores Vivian y Febles, gracias por su ayuda, su preocupación y estresarnos tanto con el PEFCI.

A todos los que me han apoyado y dado ánimos, a los que están y los que no, gracias por todo.

A los que se me olvidaron ahora mismo, pues después de 2 días sin dormir pueden ser muchos....a los que alguna vez me preguntaron ¿cómo va la tesis?

*Especialmente quisiera agradecer a Fidel, Comandante en Jefe eterno, por sus ideas brillantes y acertadas, por hacer esta gran UCI que ha sido mi segunda casa durante estos 9 años.*

## DECLARACIÓN JURADA DE AUTORÍA Y AGRADECIMIENTOS

Declaro por este medio que yo Tayché Capote García, con carné de identidad 84031003054, soy el autor principal del trabajo final de maestría Conceptualización e implantación de un Laboratorio Industrial de Pruebas de Software, desarrollada como parte de la Maestría en Calidad de Software y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firman la presente declaración jurada de autoría en La Habana a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Ing. Tayché Capote García  
Autora

---

Dra. Ailyn Febles Estrada  
Tutora

---

MsC. Ramsés Delgado Martínez  
Tutor

## **Resumen**

Propiciar la calidad en el software es una actividad que ha surgido como consecuencia de la fuerte demanda y de la competencia, debido a la vorágine de ofertas en el mercado y las condiciones que impone el entorno a las organizaciones, convirtiéndose en una necesidad prioritaria para las entidades que lo desarrollan. Un requisito indispensable para ello, lo constituye la realización de pruebas de software, en el momento adecuado y del tipo adecuado, las que figuran como un instrumento para evaluar y controlar la calidad, al detectar los errores existentes.

Ante los constantes cambios y avances que se experimentan en la Industria del Software, se hace imprescindible para Cuba, específicamente para la UCI por el volumen de proyectos productivos que desarrolla, el trabajo dirigido a la capacitación y buena preparación de sus profesionales en temas de calidad, proporcionando el desarrollo de productos eficientes y eficaces.

El presente trabajo conceptualiza y muestra los resultados de la implantación de un Laboratorio Industrial de Pruebas de Software (LIPS) en una universidad productiva, llevándolo a un escalón superior donde se vincula la actividad productiva con la docente, certificando el rol de probador a todos los estudiantes de 2do año de la carrera, a través de la asignatura Práctica Profesional 2 y una sinergia total con el nuevo modelo de integración formación, producción e investigación de la UCI. Además se comienzan a utilizar herramientas automatizadas para la gestión de los procesos de prueba y para la ejecución de estas.

# Índice de contenidos

Introducción.....	1
1 Fundamentación teórica.....	11
1.1 Introducción.....	11
1.2 Calidad de software, conceptos esenciales.....	11
1.3 Las pruebas de software como parte del control de calidad.....	13
1.4 Las pruebas de liberación.....	16
1.5 Principales Modelos de Calidad.....	17
1.5.1 Modelo de McCall.....	18
1.5.2 Modelo de Boehm.....	19
1.5.3 ISO/IEC 9126-1 Parte 1: Modelo de Calidad. ....	20
1.5.4 ISO/IEC 25000, Software Product Quality Requirements and Evaluation (SquaRE )......	24
1.6 Modelo para el desarrollo de las pruebas de software.....	24
1.7 Algunas organizaciones que realizan pruebas de software en el mundo.....	27
1.8 Evolución de las pruebas de software en la UCI.....	29
1.9 Conclusiones.....	30
2 Propuesta de solución.....	31
2.1 Introducción.....	31
2.2 Departamento de Pruebas de Software.....	32
2.2.1 Grupo de Ingeniería de Pruebas.....	33
2.2.2 Laboratorio Industrial de Pruebas de Software.....	35
2.2.2.1 Elementos fundamentales y novedosos en la concepción del LIPS.....	39
2.3 Descripción general del funcionamiento del LIPS.....	55
2.4 Conclusiones parciales.....	55
3 Validación de la hipótesis y presentación de resultados.....	56
3.1 Introducción.....	56
3.2 Validación de la propuesta .....	57
3.2.1 Calidad de las pruebas.....	57
3.2.1.1 Comparación entre la cantidad de NC detectadas en las pruebas de liberación y las pruebas de aceptación.....	57
3.2.1.2 Encuesta de satisfacción a los clientes.....	58
3.2.1.3 NC detectas en nuevos tipos de prueba.....	62
3.3 Optimización de los recursos.....	64
3.4 Tiempo de ejecución de la prueba.....	65
3.5 Conclusiones.....	67
4 Conclusiones.....	68
5 Recomendaciones.....	69
6 Referencias bibliográficas.....	70
7 Anexos.....	75
7.1 Anexo 1. Tipos de prueba definidos según las características de calidad de la ISO/IEC 9126.....	75
7.2 Anexo 2. Organización de los puestos de trabajo para los turnos del LIPS.....	77
7.3 Anexo 3. Ejemplo de una no conformidad redactado por un estudiante en el Redmine.....	77
7.4 Anexo 4. Estructura de la clase en el LIPS.....	78
7.5 Anexo 5. Guía para llenar los campos de la solicitud de pruebas en el Redmine del LIPS y Plantilla para adjuntar.....	79
7.6 Anexo 6. Vista de las Solicitudes de liberaciones en el Redmine del LIPS.....	81
7.7 Anexo 7. Vista del Sistema de Gestión de Evaluaciones de Software.....	82
7.8 Anexo 8. Vista del Sistema de Control de Acceso al LIPS.....	82
7.9 Anexo 9. Encuesta para evaluar el desempeño de las pruebas de liberación.....	83

## Índice de figuras

Figura 1: Mapa conceptual que describe las condiciones del entorno.....	11
Figura 2: Representación del Modelo de McCall.....	19
Figura 3: Representación del Modelo de Boehm.....	20
Figura 4: Representación de la calidad interna y externa. Norma ISO/IEC 9126.....	21
Figura 5: Representación de la calidad del producto. Norma ISO/IEC 25 000.....	24
Figura 6: Organización inicial del Laboratorio Industrial de Pruebas de Software.....	30
Figura 7: Estructura de Calisoft.....	32
Figura 8. Estructura del DPSW.....	33
Figura 9: Estructura del GIPS.....	34
Figura 10: Cantidad de artefactos probados por meses.....	36
Figura 11: Representación proceso industrial del LIPS.....	41
Figura 12: El LIPS en el MIFPI de la UCI a través de la asignatura PP2.....	45
Figura 13: Cambios de estado de las NC.....	49
Figura 14: Estructuras de las carpetas en el SVN para los Expedientes de Prueba.....	51
Figura 15: Proceso del LIPS.....	55
Figura 16: Valor de las NC según su impacto por roles.....	60
Figura 17: Valor de las NC según su impacto por proyecto.....	60
Figura 18: Consideraciones sobre qué mejoras debe tener el proceso de pruebas.....	61

## Índice de tablas

Tabla 1: Comparación entre los modelos de calidad.....	22
Tabla 2: Presencia de las características de calidad en los modelos.....	23
Tabla 3: Algunas organizaciones y laboratorios de prueba en el mundo.....	28
Tabla 4: Herramientas automatizadas identificadas por tipos de prueba.....	55
Tabla 5: Comparación de NC detectadas en PL y PA.....	58
Tabla 6: Cantidad de NC en Pruebas de Instalación y Configuración.....	63
Tabla 7: Cantidad de NC en Pruebas de Seguridad.....	64
Tabla 8: Cantidad de NC en Pruebas de Carga y Estrés.....	64
Tabla 9: Tiempo dedicado a una iteración de pruebas.....	66

## **Introducción**

En un mundo como el actual, globalizado y muy competitivo, es necesario lograr un mejor desempeño de las organizaciones y una profundización y agilidad del proceso de toma de decisiones, realizando un manejo y uso de la información a favor de que las instituciones y los individuos, mantengan vigentes las bases del conocimiento que requieran, para alcanzar sus objetivos.

El nuevo entorno impone condiciones que hace que las organizaciones tengan en cuenta un grupo de elementos para su gestión, esenciales para lograr ser productivas e insertarse en el mercado con éxito. Estos elementos están relacionados con las estrategias, los retos en plazos definidos y que los objetivos de la organización estén bien claros; para lo cual es esencial contar con información disponible, compartir conocimientos, globalizar mercados, optimizar recursos, descentralizar y diversificar.

En este contexto debe insertarse la industria del software, que a pesar de ser de las últimas en llegar al mercado cuenta con índices de desarrollo elevados, que presuponen un reto para cualquier empresa.

El desarrollo de la industria de software ha sido acelerado, lo que puede evidenciarse en el crecimiento de las exportaciones en países como Argentina, que aumentaron su cifra en un 33% del año 2002 al 2003 (Finquelievich 2004) y mostró desde el 2000, que obtuvo 35 millones de dólares, un ritmo sostenido hasta el 2007 en este crecimiento (Santos 2009); Uruguay reportó en el 2000 una cifra de 79 millones de dólares, mientras que en el 2007 alcanzaba la cifra de 188 millones (Martínez 2009); India, que se ha convertido en una gran potencia en esta área había alcanzado entre 1995-1996, 734 millones de dólares y en 1999-2000 alcanzó 3900 millones de dólares, cifra que ha sido superada en los años posteriores (Cairó 2000), en los que se han calculado alrededor de 5 mil millones de dólares como ingresos anuales (SÍNTESIS 2005). Cuba aunque llegó hace poco a este mercado, ha obtenido algunos resultados alentadores (en el 2005 se obtuvieron 627 800 dólares, mientras que en el 2007 fue de 1 194 100 dólares) (Santos 2009). Estos números son indicadores de la existencia de un mercado que, aunque nada fácil de conquistar, puede constituir en pilar

del crecimiento económico de no pocos países. (Delgado 2010)

Aunque los resultados alcanzados no son malos, no cubren las expectativas inicialmente vislumbradas debido básicamente a que la productividad que se alcanza en general es baja, la cantidad de recursos a consumir -en tiempo principalmente- es alta y el trabajo realizado casi nunca tiene la calidad requerida. Los proyectos se concluyen en fecha posterior a lo planificado y los problemas no se detectan a tiempo en este medio indisciplinado y caótico de desarrollo. (Febles, Álvarez, and Fernández 2000) (SIME 1997) (Febles 2000) En las condiciones descritas del entorno actual, estos problemas deben ser mitigados si la organización se propone insertarse en el mercado, para lo cual debe ofrecer productos y/o servicios de calidad, que la hagan creíble y confiable.

Al profundizar en las raíces o causas de tales insatisfacciones afloran de manera reiterada la aplicación inadecuada de las técnicas de Ingeniería de Software, la no utilización de los roles y procesos apropiados para el desarrollo de las tareas de la empresa de software y el no utilizar modelos de calidad en ellas. (Febles, Álvarez, and Fernández 2000) (Febles 2000) (Baeza 1995)

A pesar de que existe un interés gubernamental debido a que la innovación tecnológica, la velocidad en el desarrollo y la satisfacción del cliente, se han convertido en la consigna de las organizaciones que quieren sobrevivir y cada vez ser más competitivas en el mundo; informes de instituciones dedicadas al análisis de software muestran los siguiente indicadores, para nada alentadores (Reo 2002) (Alejandro G Bedini 1995) (Butler, Faith, Verla, and Elaine 2011) (Camou 2002):

- ◆ El 25% de los proyectos de software son abortados.
- ◆ Se liberan productos a sus clientes con remanentes del 15% de defectos.
- ◆ Muchas empresas gastan de 30% a 44% de su tiempo y dinero en retrabajo sobre software ya liberado.
- ◆ Se cumplen las planificaciones de tiempo solamente el 53% de las veces.

En el reporte de CHAOS 2009 la estadística que se muestra no es diferente a los números que se han presentado en años anteriores, pues se refleja que solo el 32% de los proyectos

de software son catalogados de exitosos, el 44% fueron cancelados (entrega tardía, por encima del presupuesto y/o con menos requisitos de los pactados con el cliente) y el 24% fallaron (cancelados antes de concluirse o entregados antes de su uso)(CHAOS 2009). A pesar de estas cifras, en el reporte publicado en el 2011 se precisa que existe un marcado aumento en las tasas de éxito del proyecto desde 2008 hasta 2010, lo que representa un alza en estas tasas con respecto al estudio anterior, así como una disminución en el número de fracasos. El punto más bajo en los últimos cinco períodos de estudio fue de 2004, en el que sólo el 28% de los proyectos tuvieron éxito. Esto refleja que las organizaciones desarrolladoras de software van comprendiendo que deben resolver los problemas que provocan el fracaso en los proyectos, aspecto indispensable para ser más productivas e insertarse en el mercado.(CHAOS 2011)

Dentro de estas dificultades se destacan dos variables que necesariamente deben considerarse para el desarrollo de un software: la calidad y la productividad. La primera se encuentra afectada por factores como el personal a cargo del proceso, el tipo de herramientas utilizadas y las metodologías aplicadas. El segundo aspecto se refiere a la capacidad de la empresa de reducir costos previendo las fallas antes que se produzcan. (Baeza 1995)

En el año 2002 entra a formar parte de esta industria la Universidad de las Ciencias Informáticas (UCI), la cual tiene dentro de sus misiones esenciales producir software con eficiencia y calidad, para lo cual se trabaja en la formación de ingenieros bien preparados desde la producción, valiéndose de las buenas prácticas y buen uso de los estándares y normas de calidad, para entregar al cliente un producto confiable y que se ajuste a sus necesidades.

Esta universidad productiva tiene un gran reto, pues es responsable de la informatización de la sociedad cubana, pero además está comenzando a tener un impacto en la economía por concepto de exportación de software y servicios informáticos, el cual debe ser mayor, logrando diversificar su mercado y aportando cada vez más recursos a la nación. La UCI como universidad actual, al igual que cualquier otra organización, debe ser consciente de

que el futuro ya no es la prolongación del presente, necesita anticiparse al impacto de las fuerzas que interactúan en su entorno, creando e innovando sus propios instrumentos de gestión e implementando estrategias que le permita asegurar los resultados que se ha propuesto alcanzar institucionalmente.

La UCI se propuso como objetivo estratégico para el 2012 el ingreso de mil millones de dólares por concepto de software, sin dudas una cifra ambiciosa pero alcanzable (UCI 2008). Ello requiere diversificar el mercado, lograr un estadio superior en el mercado venezolano, con el que se viene trabajando desde hace un tiempo; pero en cualquiera que se proponga solidificarse el trabajo o insertarse, se pondrá como premisa que el producto tenga la calidad requerida.

En la universidad se realizó, por parte de la Dirección de Calidad de Software, en diciembre del 2007 un diagnóstico a los proyectos de la UCI. Fueron evaluados 137 proyectos de diferentes tipos y características, detectándose entre otras las siguientes dificultades (DCS 2008):

### **Organización**

- ◆ Gran diversidad de roles.
- ◆ Solapamiento de roles.
- ◆ No se tiene definición de las actividades de los roles.
- ◆ Poca capacitación en el rol a desempeñar.
- ◆ No todos los proyectos se encuentran regidos por una metodología.

### **Planificación y seguimiento**

- ◆ No se gestionan los requisitos del sistema.
- ◆ No todos los desarrolladores participan en la comprensión de los requerimientos del sistema.
- ◆ No se gestionan los costos de los proyectos.
- ◆ Gran diversidad de lenguajes de programación.
- ◆ No se realizan actividades básicas dentro de todo proceso de desarrollo de software.

## **Calidad**

- ◆ No se utilizan estándares para el desarrollo.
- ◆ Los artefactos que se entregan a los clientes poseen diferentes formatos.
- ◆ Se entregan al cliente artefactos con alto número de no conformidades.
- ◆ Muy difícil ejecutar acciones de control de la calidad.
- ◆ No se documenta el trabajo realizado.

## **Mediciones**

- ◆ No se cuenta con datos históricos para realizar futuras mediciones.

Estos problemas no están alejados de los que existen a nivel mundial en la industria del software, donde los relacionados con la calidad deben constituir un foco de atención. Asegurar y controlar la calidad en el proceso de desarrollo de software es esencial si se quiere alcanzar un nivel superior, ser más competitivo y por supuesto, es clave para garantizar la satisfacción del cliente.

Dentro de las actividades de control de la calidad están las pruebas de software, elemento esencial en la verificación y validación de un software. Un buen proceso de pruebas generará un conjunto de beneficios para la industria del software, dentro de los que se pueden destacar (A. de Amescua García, Velasco, and Sanz 2007):

- ◆ Se produce un incremento de la satisfacción del cliente al utilizar un software con una cantidad de errores inferior.
- ◆ Se incrementa la eficiencia del proceso de desarrollo.
- ◆ Se facilita la definición y cumplimiento de los objetivos de calidad.
- ◆ Se incrementa la satisfacción de los trabajadores debido a que se proporcionan herramientas y recursos apropiados para la realización eficiente del trabajo.

Dependiendo de la estructura organizativa de la organización, las actividades relacionadas con las pruebas software se desarrollan siguiendo diferentes modelos. Actualmente, se está comenzando a realizar lo que se ha nombrado outsourcing o tercerización de las actividades relacionadas con las pruebas de software, o lo que es lo mismo, subcontratar las pruebas de software con el objetivo de reducir costes, tiempos y mejorar tanto el proceso de

pruebas como la calidad del producto. (Koomen, van der Aalst, Broekman, and Vroon 2006)  
En esta nueva tendencia los equipos de prueba son independientes del equipo de desarrollo, por lo que no están comprometidos con los tiempos de entrega, algo que en muchas ocasiones condiciona los resultados de la prueba. Si bien es cierto que este modelo no sustituye las pruebas que debe hacer el desarrollador o el equipo de calidad del proyecto, es un paso importante en el control de la calidad del producto, visto como un paso previo a la entrega al cliente.

Con esta concepción se creó desde sus inicios el laboratorio de pruebas dentro de la Dirección de Calidad de Software (DCS) de la UCI, hoy Centro de Calidad para Soluciones Informáticas (Calisoft). Se comenzaron a liberar los artefactos de los proyectos de exportación, los cuales fueron en aumento y se sumaron además proyectos con clientes nacionales, por lo que la demanda de artefactos a probar se fue haciendo cada vez mayor.

En un primer momento se pasó de tener un laboratorio central con estudiantes como probadores y un especialista al frente, a un esquema en el que existía un grupo de calidad en cada facultad. Este era guiado por un asesor de calidad, nuevo rol que se crea, y dirigidos todos desde la DCS, que gestionaba centralmente las solicitudes de prueba y utilizaba estos recursos en función de la disponibilidad y capacidad.

El crecimiento de los artefactos a probar que venía experimentando el laboratorio, donde se probaron desde enero de 2008 hasta julio de 2010 un total de 5513 artefactos, teniendo como media por meses 178 entregables, hizo que algunos problemas que ya existían se hicieran más críticos y se fueran convirtiendo en barreras para cumplir con eficiencia y calidad los compromisos del laboratorio. Las dificultades esenciales acentuadas son: (Febles et al. 2011)

- ◆ Pobre asistencia de los estudiantes de los grupos de calidad a los turnos de trabajo planificados.
- ◆ Coincidencia de actividades docentes o extensionistas con el horario planificado, siendo las primeras de prioridad para los estudiantes.
- ◆ Los estudiantes priorizan otras actividades que se le planifican por la facultad por encima de la producción, pues esta no tiene repercusión ninguna en evaluaciones.

- ◆ Pocos estudiantes en los grupos de calidad de las facultades para asumir la demanda de artefactos a probar.
- ◆ No se incorporan a los grupos de calidad estudiantes de 1ero y 2do años como generalidad.
- ◆ El horario de producción de 1ero y 2do año está condicionado por la asignatura Práctica Profesional.
- ◆ La calidad en la ejecución de las pruebas no siempre es la más óptima y no se tiene un mecanismo de peso para evaluar el trabajo de los estudiantes, de manera que cuente como parte de su formación y repercuta en sus resultados productivos o académicos.

La necesidad de hacer más eficiente el laboratorio de pruebas de manera que se puedan asumir todas las solicitudes de pruebas, aumentando la calidad en su ejecución, conlleva a definir el siguiente **Problema de investigación**:

¿Cómo aumentar la calidad de las pruebas, optimizar los recursos y disminuir el tiempo de ejecución de estas en un laboratorio de pruebas de una universidad productiva?

Se formuló la siguiente **Hipótesis**:

Con la implantación de un Laboratorio Industrial de Pruebas de Software en una universidad productiva se aumentará la calidad de las pruebas ejecutadas, se optimizarán los recursos y se disminuirá su tiempo de ejecución.

Para enfrentar este problema, se definió como **Objeto de la investigación**:

Procesos de pruebas de software.

Y como **Campo de acción**, asociado a este objeto:

Una Universidad productiva con un modelo de integración formación, producción e investigación.

El **Objetivo general** de la investigación es:

Implantar un Laboratorio Industrial de Pruebas de Software en una universidad productiva para aumentar la calidad de las pruebas ejecutadas, optimizar los recursos utilizados en estas y disminuir su tiempo.

De este se derivan los **Objetivos específicos**:

1. Analizar las concepciones teóricas de las pruebas de software.
2. Estudiar los modelos más significativos de calidad del producto.
3. Valorar las organizaciones que realizan pruebas de software, precisando la relación que tienen con las universidades.
4. Conceptualizar un Laboratorio Industrial de Pruebas de Software (LIPS) en una universidad productiva.
5. Caracterizar los elementos novedosos que se insertan en la organización del LIPS.
6. Validar la implantación del Laboratorio Industrial de Pruebas de Software a partir de los resultados obtenidos en cuanto al aumento de la calidad de las pruebas, la optimización del uso de recursos y la disminución del tiempo de ejecución de estas.

Los métodos empíricos utilizados para la investigación:

- ◆ La observación participativa para obtener información de las discusiones en sesiones científicas y encuentros que se realicen en entornos que aporten información para la investigación.
- ◆ El grupo focal para realizar discusiones con los especialistas del laboratorio de pruebas de Calisoft con el objetivo de identificar los problemas esenciales que afectan el buen desempeño de las pruebas de software. También fue útil para discutir los elementos que ayudaron a la conceptualización e implantación del Laboratorio Industrial de Pruebas de Software (LIPS).
- ◆ El método de la entrevista a profundidad para precisar cómo conectar el LIPS con el nuevo modelo de integración formación- producción-investigación de la UCI y conocer el impacto de la propuesta al ser presentada en eventos internacionales.
- ◆ El cuasi experimental para comprobar la utilidad de los resultados obtenidos a partir de la propuesta realizada.
- ◆ El estudio de casos permite realizar un examen intensivo de organizaciones que realizan pruebas de software para profundizar en su organización y relación que tienen con las universidades. Además permitió validar la propuesta a partir del estudio

del comportamiento de proyectos en las pruebas de software antes y después de la implantación del LIPS.

- ◆ La encuesta para determinar la satisfacción de los proyectos con el servicio brindado por el LIPS.

Como métodos teóricos:

- ◆ Hipotético-deductivo para la elaboración de la hipótesis central de la investigación y para proponer nuevas líneas de trabajo a partir de los resultados parciales.
- ◆ Método histórico-lógico y el dialéctico para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
- ◆ La sistematización permitió organizar la información dispersa encontrada sobre los temas investigados.
- ◆ Análisis-síntesis, al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.
- ◆ Enfoque sistémico, para lograr que los elementos que forman parte del modelo sean un todo que funcione de manera armónica.

### **Resultados a alcanzar:**

Como resultado del presente trabajo se obtendrá la conceptualización de un Laboratorio Industrial de Pruebas de Software, donde serán liberados todos los entregables de los proyectos de los Centros de Desarrollo de la UCI o de cualquier otra entidad desarrolladora de software que solicite el servicio de pruebas de liberación. Se logrará además una relación armónica con el nuevo modelo de integración formación, producción e investigación de la UCI, pues con el trabajo en el LIPS los estudiantes de 2do año de la universidad deben acreditar el rol de probador, lo cual constituye su vinculación real a la producción desde el ciclo básico.

La implantación del LIPS permitirá aumentar la calidad en la ejecución de las pruebas de liberación, optimizar el uso de los recursos y minimizar el tiempo en que se ejecutan estas,

con lo cual se logra mayor eficiencia en el proceso de pruebas y liberar a los clientes productos con mayor calidad. Además introduce el uso de herramientas automatizadas para la gestión de las pruebas de software y la nueva organización permite que los profesionales del área de pruebas se puedan especializar según las características de calidad definidas en normas internacionales.

**Estructura del documento:**

El documento se encuentra estructurado en tres capítulos. En el Capítulo 1 se realiza el estudio del estado del arte de las pruebas de software, profundizando en los modelos de calidad del producto; se busca información sobre las organizaciones que realizan pruebas de software en el mundo, estableciendo puntos en común en su organización y su vinculación con las universidades; así como la evolución de las pruebas de software en la UCI, que conllevó a la propuesta actual. En el Capítulo 2 se conceptualiza el Laboratorio Industrial de Pruebas de Software, profundizando en los elementos novedosos y esenciales que incorpora. Por último en el Capítulo 3 se realiza un estudio de los resultados de la implantación de LIPS en la UCI, teniendo en cuenta las 3 variables planteadas en la hipótesis de la investigación: calidad de las pruebas, optimización de los recursos y disminución del tiempo de ejecución.

# 1 Fundamentación teórica.

## 1.1 Introducción

En el capítulo se presenta una reseña sobre los temas esenciales que deben ser abordados como fundamentación de la propuesta del trabajo investigativo. Estos están relacionados esencialmente con los conceptos esenciales de la calidad de software, específicamente las pruebas de software y algunas organizaciones que las definen dentro de sus servicios esenciales, a partir de servicio outsourcing o tercerización de pruebas. Se precisan los modelos más importantes que definen las características de calidad que deben evaluarse en los productos de software. También se hace referencia a cómo ha sido la evolución de las pruebas en la UCI.

## 1.2 Calidad de software, conceptos esenciales.

Las condiciones que ha impuesto el entorno ha hecho que las organizaciones busquen estrategias para ser más competitivas y poder insertarse en el mercado. En el siguiente mapa conceptual elaborado por la autora a partir de la revisión bibliográfica (Figura 1), se resume lo que se ha explicado en la Introducción:

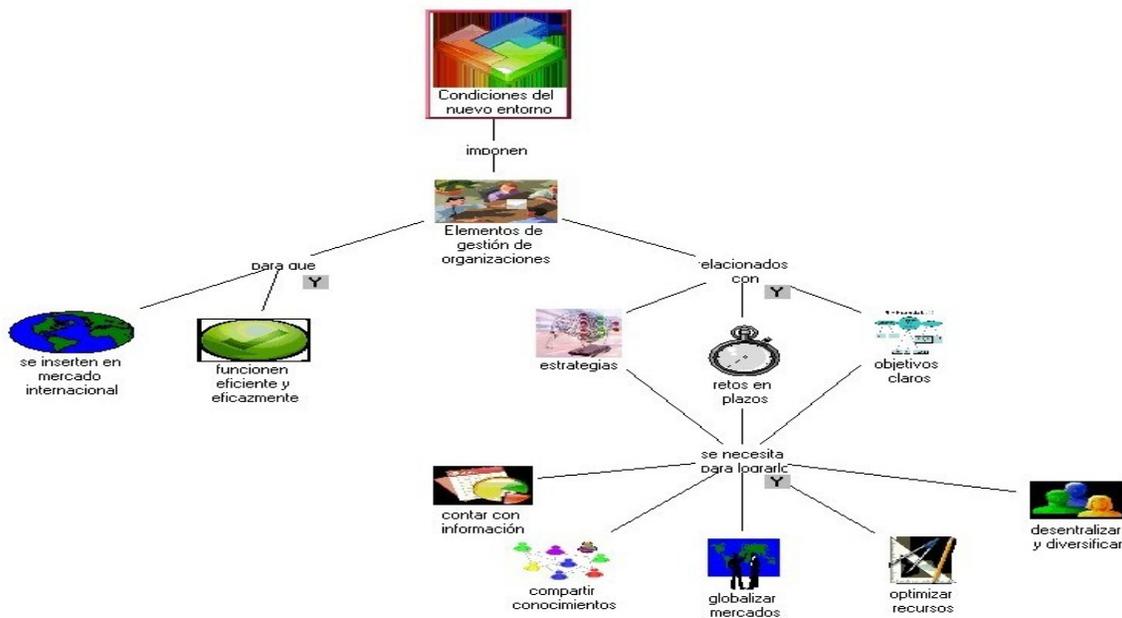


Figura 1: Mapa conceptual que describe las condiciones del entorno.

Dado este escenario no es una sorpresa que se haya producido en la década del 90 un auge a nivel mundial de la calidad en diversas áreas, a lo que no fue ajena la industria del software. (Alejandro Bedini 2007) (Garvin 1984)

El propósito de la ingeniería de software es generar y mantener sistemas de software dentro de las restricciones de tiempo, funcionalidad y costos acordados con el cliente. Las metas de esta disciplina tecnológica son mejorar la calidad de los productos desarrollados y aumentar la productividad de los ingenieros de software (García Mireles and Rodríguez Jacobo 2001). Realmente la ingeniería de software se enfrenta a muchos más obstáculos para poder dominar la calidad si se compara con otras ciencias, entre ellos se pueden mencionar:

- ◆ No existe una definición estándar y universal de qué es calidad.
- ◆ La calidad debe satisfacer a una amplia gama de entes relacionados pero no mutuamente excluyentes como son: clientes, procesos, organismos, productos, etc.
- ◆ Crear una cultura de la calidad.

Uno de los problemas que se presenta cuando se analiza el término calidad es lo ambiguo que es y los disímiles conceptos que se han definido de este. Se han llegado a precisar múltiples aristas dentro de las que se ha conceptualizado como son: la popular, la profesional, de software y la administrativa. (Kan 2000)

Algunos conceptos de calidad más reconocidos, que están relacionados con el software, son: Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor; según la Real Academia de la Lengua Española. (RAE)

IEEE: la calidad es el grado mediante el cual un sistema, componente o procesos satisface las necesidades o expectativas de un cliente o usuario. (IEEE 1990)

Pressman precisó que es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentos, y con las características implícitas que se espera de todo software desarrollado profesionalmente. (Pressman 1998)

Para CMMI calidad es la habilidad de un conjunto de características inherentes a un producto, componente o proceso para cubrir los requerimientos del cliente. (CMMI 2002)

ISO la define como el grado mediante el cual una serie de características inherentes al producto satisfacen los requerimientos. (ISO 2005)

En todos los conceptos anteriores hay puntos comunes como son los requisitos y el cliente. Los requisitos describen lo que quiere el cliente y es la guía para verificar la calidad del software; el cliente por su parte define la calidad del producto, a partir de que se cumpla lo solicitado y sus expectativas.

La autora define la calidad a partir de los análisis anteriores como la concordancia con los requisitos funcionales y no funcionales establecidos, con normas y estándares que deba cumplir el producto, logrando satisfacer las necesidades y expectativas de un cliente o usuario.

Cuando no hay calidad en un software se pueden presentar las siguientes situaciones (Rodríguez 2010):

- ◆ Programas que no hacen exactamente lo que se espera.
- ◆ Proyectos que no terminan nunca.
- ◆ Sistemas informáticos que no se utilizan por la dificultad de su manejo.
- ◆ Productos software que son imposibles de mantener cuando desaparece la persona o personas que lo desarrollaron.
- ◆ Software poco seguro.

La calidad de software debe ser construida desde el comienzo, no es algo que puede ser añadido después (Humphrey 1997). Es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software (Carrasco, D. G León, and Benavides 2004).

### **1.3 Las pruebas de software como parte del control de calidad.**

La IEEE define como control de la calidad a un conjunto de actividades para evaluar la calidad de los productos desarrollados. (IEEE 1990)

El control de la calidad del software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos

fundamentales: (1) mantener bajo control un proceso y (2) eliminar las causas de los defectos en las diferentes fases del ciclo de vida, (Pressman 2002). Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el control de la calidad del software es la “Prueba de Software”.

En el caso específico del desarrollo de software, las pruebas se han definido de diversas maneras y existen enfoques dinámicos y estáticos. Los enfoques dinámicos apuntan a ejecutar una parte o todo el software para determinar si funciona según lo esperado. El enfoque estático se refiere a la evaluación del software sin ejecutarlo, usando mecanismos automatizados (herramientas asistidas) y manuales tales como controles de escritorio, inspecciones y revisiones (Daich, Price, Raglund, and Dawood 1994).

La evaluación estática busca faltas sobre el sistema en reposo, o sea, estudia los distintos modelos que componen el software buscando posibles faltas en los mismos. Las técnicas de evaluaciones estáticas se pueden aplicar, tanto a requisitos, modelos de análisis, diseño, código o cualquier otro artefacto que no requiera ejecutar una aplicación.

Por su parte las evaluaciones dinámicas generan entradas al sistema con el objetivo de detectar fallos, cuando este ejecuta dichas entradas. Los fallos se observan cuando se detectan incongruencias entre la salida esperada y la salida real. La IEEE (IEEE 1990) las define como una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente. SWEBOK (SWEBOK 2004) precisa que la prueba de software es la verificación dinámica del comportamiento de un programa contra el comportamiento esperado, usando un conjunto finito de casos de prueba, seleccionados de manera adecuada desde el dominio infinito de ejecución.

De forma general la autora define la prueba de software como actividades que se realizan, en las cuales se utilizan diseños de prueba, el artefacto es ejecutado o evaluado bajo requisitos especificados y se registran los resultados obtenidos, los que deben ser comparados con los resultados esperados, para emitir una evaluación basada en aspectos determinados del artefacto. Este concepto se adoptará de manera general en la investigación e integra la

esencia de las evaluaciones estáticas y dinámicas dentro de uno solo.

Los términos Verificación y Validación suelen usarse indistintamente en algunos contextos. Para las pruebas de software ambos términos indican conceptos diferentes. La definición dada por IEEE (IEEE 1990) de estos es la siguiente:

- ◆ Verificación: proceso de evaluación de un sistema o componente para determinar si un producto de una determinada fase de desarrollo satisface las condiciones impuestas al inicio de la fase.
- ◆ Validación: proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar cuándo se satisfacen los requerimientos especificados.

Boehm (Boehm 1984) usa dos preguntas para clarificar la diferencia entre Verificación y Validación:

- ◆ Verificación: ¿Estamos elaborando correctamente el producto?
- ◆ Validación: ¿Estamos elaborando el producto correcto?

CMMI (CMMI 2002) define el propósito de la Verificación y la Validación de la siguiente forma:

- ◆ El propósito de la Verificación es asegurar que los Productos Internos seleccionados cumplen con su especificación de requerimientos. Los métodos de verificación pueden ser, entre otros: inspecciones, revisiones por pares, auditorías, recorridas, análisis, simulaciones, pruebas y demostraciones.
- ◆ El propósito de la Validación es demostrar que un producto o componente de producto cumple su uso previsto cuando es puesto en su ambiente previsto. Deben ser seleccionados los productos internos (por ejemplo: requerimientos, diseño, prototipos) que mejor indican cuán bien el producto y los productos internos deben satisfacer las necesidades del usuario.

De manera general en la bibliografía consultada, la prueba de software es tomada como una parte del proceso de Verificación y Validación y bajo el entendido de que esta requiere ejecutar el código. Se puede precisar además que en la Verificación se ejecutan pruebas

para revisar si el software se está construyendo de forma correcta, teniendo en cuenta la documentación definida durante el desarrollo. En el caso de la Validación es esencial la intervención del cliente y se realizan las pruebas en un ambiente real, determinando si el software que se ha construido es el correcto, según las necesidades de su cliente.

Las pruebas de software constituyen una etapa importante y son esenciales en el control de la calidad, si se quiere lograr el objetivo de entregar al cliente un producto de calidad, según los parámetros establecidos por él mismo. En algunos sectores cuando se desarrolla un software, estas se vuelven más críticas, pues como se precisa en (Riascos 2008) los fallos en un sistema pueden llegar a ocasionar desastres en las organizaciones (Piattini, F García, and Caballero 2007). Es tanto el impacto que pueden llegar a tener estos fallos que pueden producir hasta pérdidas humanas, como lo menciona NIST (NIST 2002), estableciendo como ejemplo el sector aeroespacial –el desastre del Ariane 5, y en la aeronáutica, los de algunos aviones comerciales–.

En la presente investigación el concepto de pruebas de software definido por la autora se inserta dentro de la verificación, pues se llevan a cabo dentro del proceso de pruebas de liberación, donde se realizan pruebas con el objetivo esencial de verificar que el producto esté construido correctamente, tomando como base los requisitos definidos por el cliente y otras normas y/o estándares que apliquen al producto en cuestión y se hayan precisado por este.

#### **1.4 Las pruebas de liberación.**

Desde el punto de vista del cliente y los usuarios, la calidad de un producto de software es percibida principalmente por las fallas que encuentran en el producto y por la gravedad que éstas tienen para el negocio del cliente. Para ser competitivas, las empresas desarrolladoras de software necesitan asegurarse de la calidad de sus productos previo a su instalación en el ambiente del cliente. (Pérez Lamancha 2006)

Un paso previo a la instalación del sistema en el ambiente del cliente o la entrega de cualquier artefacto que se obtenga como parte del desarrollo de software, lo constituye la

prueba de liberación, teniendo en cuenta que para el Departamento de Pruebas de Software (DPSW) estas constituyen una etapa dentro del ciclo de vida del software, se ejecutan una vez que se haya concluido el desarrollo y realizadas las pruebas internas en el proyecto, a todos los artefactos que constituyen entregables al cliente, o que deben ser utilizados como parte del desarrollo de otro proyecto.

Dentro de las pruebas de liberación se realizan evaluaciones estáticas y dinámicas, dependiendo del artefacto que deba ser evaluado, las cuales se enmarcan en el concepto de pruebas de software al que llegó la autora a partir de las definiciones estudiadas.

Después de definidos estos conceptos puede surgir una pregunta: ¿Es realmente posible encontrar un conjunto de propiedades en un software que den una indicación de su calidad? Este es uno de los principales problemas que se presentan cuando se habla de la calidad de un software y que encuentra su respuesta en los Modelos de Calidad.

### **1.5 Principales Modelos de Calidad.**

Los Modelos de Calidad no son más que documentos que integran en sí las mejores prácticas para un desarrollo de software eficiente. Proponen temas de administración en los que cada organización que los aplique debe hacer énfasis, e integran diferentes prácticas dirigidas especialmente a procesos clave. De esta forma permite medir la evolución de la calidad durante todo el ciclo de vida del sistema informático.

La ventaja de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar. Estos ayudan también a comprender las relaciones que existen entre diferentes características de un producto de software. (IEEE 1990)

La elaboración de un modelo de calidad pasa por la identificación de aquellas características, tanto internas como externas, que influyen en la calidad de un producto software. Un modelo de calidad es una descomposición jerárquica, y su elaboración puede abordarse de diferentes formas, como puede ser la identificación de objetivos y métricas que permitirían la caracterización de la calidad.

En Ingeniería del Software se han elaborado diferentes modelos de calidad y estándares relacionados con la calidad que ofrece un producto software. Entre estos modelos se pueden identificar dos tendencias: por un lado están los que persiguen identificar características del producto y, por otro, los que tratan de identificar características de calidad en el proceso de desarrollo del producto software (Carretero 2007). En la presente investigación son de interés los modelos que identifican las características de calidad del producto.

Son varios los modelos de calidad que se han definido y se aplican en función del objetivo que se persigue y la adecuación de estos al contexto en el que se deben aplicar. Se pueden mencionar los modelos estudiados por la autora durante la investigación: Modelo de McCall, Modelo de Boehm, Modelo de Dromey, Modelo SATC (Software Assurance Technology Center ), Modelo C-QM, WebQEM (Web Quality Evaluation Method ), ISO/IEC 9126 y Modelo FURPS (Funcionality – Usability – Reliability – Performance – Supportability ). De estos modelos los que mayor relevancia han tenido y constituyen de forma general la base del resto son el Modelo de McCall, el de Boehm y la Norma ISO/IEC 9126, de los cuales se especifican algunas características a continuación.

### **1.5.1 Modelo de McCall.**

El modelo de Jim McCall, desarrollado inicialmente para la Fuerza Aérea de los EE.UU en 1977, es uno de los más conocidos y referenciados. Este modelo busca reducir la brecha entre usuarios y desarrolladores enfocándose en un número de factores de calidad que reflejen las prioridades de ambos. El modelo establece una jerarquía de 3 Perspectivas, 11 Factores, 23 Criterios de Calidad y 41 Métricas. En la Figura 2 se representa la relación entre las 3 Perspectivas: operación (características operacionales), reutilización (capacidad de adaptarse a nuevos entornos) y revisión (capacidad de soportar cambios), con los 11 Factores.

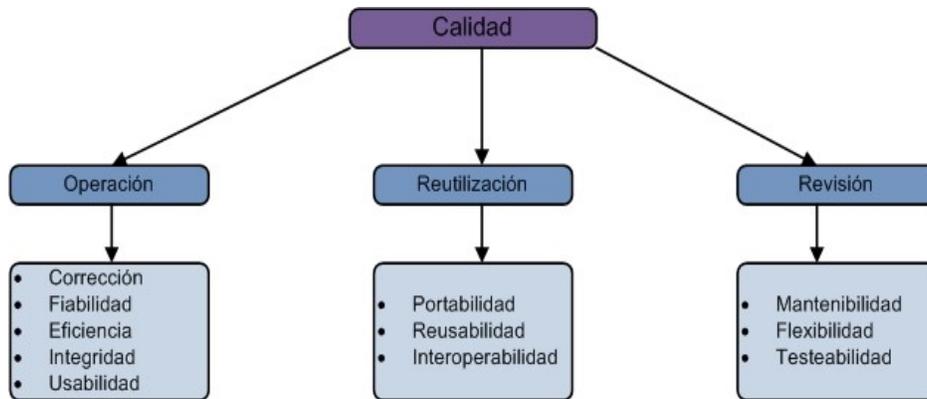


Figura 2: Representación del Modelo de McCall.

El gran aporte de este modelo fue el establecimiento de relaciones entre características de calidad y métricas, aunque haya sido criticado por la objetividad de algunas de las métricas propuestas. (McCall and Richards 1977)

### 1.5.2 Modelo de Boehm.

Barry Boehm, un conocido y además precursor ingeniero de software de los años 70 también realizó sus aportes propios al tema de la calidad del software. El modelo que propone es similar al de McCall al representar la estructura de características de calidad de forma jerárquica, contribuyendo cada una de ellas a la calidad total. En este modelo también están consideradas las necesidades del usuario, pero se encuentran características de calidad en el terreno del hardware que no están presentes en el otro modelo representado anteriormente (Boehm, Brown, and Lipow 1978). En la Figura 3 se representa este modelo.



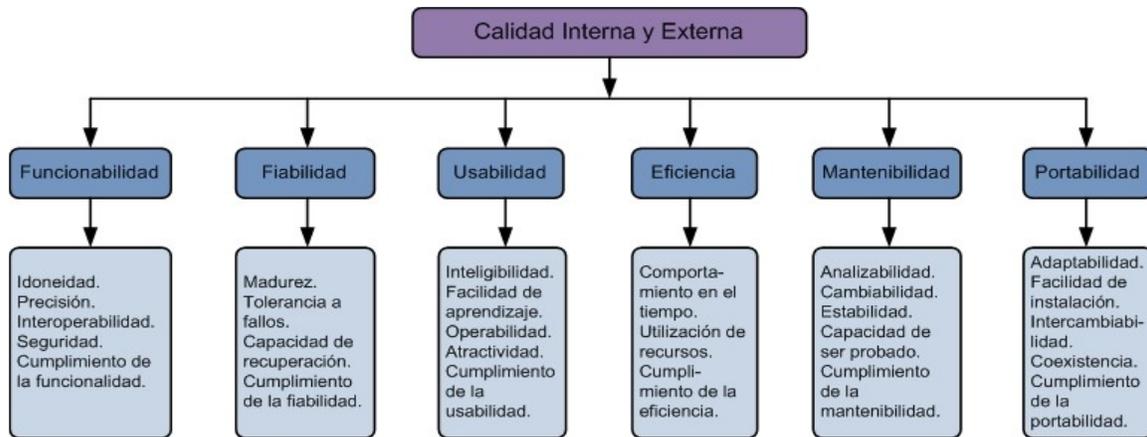


Figura 4: Representación de la calidad interna y externa. Norma ISO/IEC 9126.

### 1.5.4 ISO/IEC 25000, Software Product Quality Requirements and Evaluation (SquaRE ).

Esta norma nace por las inconsistencias entre ISO/IEC 9126 e ISO/IEC 14598, la cual establece los procesos para realizar la evaluación de la calidad del producto software y está directamente relacionada con la ISO/IEC 9126. Su objetivo es aglutinar bajo una misma familia el modelo de calidad y el proceso de evaluación. (Rodríguez 2010)

La versión de la primera publicación es de 2005 y está formada por 5 partes, aunque en este momento no se había terminado de definir todavía. Se precisan para la calidad del producto 8 características, 6 de ellas contenidas en la ISO/IEC 9126 y se adicionan 2 características. En la calidad de uso se precisan 5 características, una más que en la ISO/IEC 9126. En la Figura 5 se representa la relación entre las características y subcaracterísticas de calidad del producto definidas en la ISO/IEC 25 000.



Figura 5: Representación de la calidad del producto. Norma ISO/IEC 25 000.

Se considera que debe darse seguimiento a la norma ISO/IEC 25 000 e ir incluyendo en el trabajo del laboratorio, en la medida que vaya adquiriendo madurez en su definición, los aportes que realice. Teniendo en cuenta que estaba en definición cuando se realizó la investigación para este trabajo, no se tuvo en cuenta en la comparación que se realiza a continuación de los principales modelos de calidad.

Después de estudiados en profundidad los Modelos de McCall, Boehm y la ISO/IEC 9126 y consultados otros modelos que no fueron detallados en la presente documento al considerarse que los presentados son los más completos y que más aportan a la investigación; la autora se propuso hacer una comparación entre los modelos, de manera que se pudiera conocer cuáles características de calidad tienen más ocurrencia en estos y con esta base decidir cuál de ellos utilizar en el Laboratorio Industrial de Pruebas de Software (LIPS). En la búsqueda bibliográfica se encontró una tesis de maestría de la Universidad Tecnológica Nacional de Argentina, donde se realizó una comparación entre los modelos que fue estudiada a partir de los análisis ya realizados por la autora, la cual se presenta a continuación en la Tabla 1 y posteriormente el análisis de los resultados en los que se sustenta la definición del modelo a utilizar. (Scalone 2006)

Características/Modelos	Boehm	Dromey	McCall	SATC	C-QM	WebQM	ISO 9126	FURPS
Facilidad de uso		X	X			X	X	X
Integridad			X	X				
Corrección			X	X				
Confiabilidad	X	X	X			X	X	X
Eficiencia	X	X	X			X	X	
Facilidad de mantenimiento		X	X	X	X		X	
Facilidad de prueba	X		X					
Flexibilidad			X					
Facilidad de reutilización			X	X	X			
Interoperabilidad			X					

Portabilidad	X	X	X				X	
Ingeniería humana	X							
Fácil de entender	X			X				
Fácil de modificar	X							
Funcionalidad		X			X	X	X	X
Performance								X
Facilidad de soporte								X
Ambigüedad				X				
Trazabilidad				X				
Estructura/Arquitectura				X				
Documentación				X				
Conformidad					X			

Tabla 1: Comparación entre los modelos de calidad.

A partir de los datos precisados en la tabla anterior se determinó en cuántos modelos está definida cada característica de calidad, señalando en la Tabla 2 los que mayor presencia tienen:

Características de calidad	Cantidad de modelos en los que está presente
<b>Facilidad de uso</b>	<b>5</b>
Integridad	2
Corrección	2
<b>Confiabilidad</b>	<b>6</b>
<b>Eficiencia</b>	<b>5</b>
<b>Facilidad de mantenimiento</b>	<b>5</b>
Facilidad de prueba	2
Flexibilidad	1
Facilidad de reutilización	3

Interoperabilidad	1
<b>Portabilidad</b>	<b>4</b>
Ingeniería humana	1
Fácil de entender	2
Fácil de modificar	1
<b>Funcionalidad</b>	<b>5</b>
Performance	1
Facilidad de soporte	1
Ambigüedad	1
Trazabilidad	1
Estructura/Arquitectura	1
Documentación	1
Conformidad	1

Tabla 2: Presencia de las características de calidad en los modelos.

En la tabla anterior se señalan las características de calidad que más presencia tienen en los modelos comparados y resultan ser las 6 características que comprende la norma ISO/IEC 9126, por lo que se considera la más completa.

### 1.6 Modelo para el desarrollo de las pruebas de software.

Dependiendo de la estructura organizativa de la institución, las actividades relacionadas con las pruebas software se desarrollan siguiendo diferentes modelos. Entre ellos destacan los siguientes (Runeson, Andersson, and Thelin 2006):

- ◆ Desarrolladores que hacen las funciones de probadores.
- ◆ Equipo de pruebas integrado en el equipo de desarrollo.
- ◆ Equipos de pruebas independientes que realizan las labores de verificación y validación.
- ◆ Grupos de SQA que dedican parte de su esfuerzo a realizar pruebas.

- ◆ Outsourcing de las actividades relacionadas con las pruebas de software.

Este último modelo, el outsourcing de las pruebas de software, o lo que es lo mismo, subcontratar las pruebas de software con el objetivo de reducir costes, tiempos y mejorar tanto el proceso de pruebas como la calidad del producto, se ha convertido en una tendencia actual. (Koomen, van der Aalst, Broekman, and Vroon 2006)

En esta nueva tendencia los equipos de prueba son independientes del equipo de desarrollo, por lo que no están comprometidos con este, algo que en muchas ocasiones condiciona los resultados de la prueba. Si bien es cierto que el modelo no sustituye las pruebas que debe hacer el desarrollador o el equipo de calidad del proyecto, es un paso importante en el control de la calidad del producto, visto como un paso previo a la entrega al cliente.

Las ventajas que se obtienen al tener una organización de prueba independiente de las de desarrollo son (Myers 2004):

- ◆ Motivación en el proceso de pruebas.
- ◆ Oposición de intereses con la organización de desarrollo.
- ◆ Separación del proceso de pruebas del control gerencial de la organización de desarrollo.
- ◆ El conocimiento especializado que la organización independiente tiene respecto a las pruebas.

Como regla general sucede que un equipo de pruebas que pertenece a la propio entidad que desarrolla software detecta menos errores que evaluadores independientes y algunas de las razones de este fenómeno son (BugHuntress QA Lab 2011):

- ◆ La prueba es una disciplina de la ingeniería en particular, se requieren métodos especiales, herramientas, conocimientos, habilidades, experiencia y talento peculiar.
- ◆ El objetivo de los programadores es demostrar que el programa funciona. De manera simplista, la tarea de los probadores es detectar los fallos (es decir, probar que el programa no funciona) y asegurarse de que el programa cumple las necesidades de los usuarios. Estos dos puntos de vista en el programa son muy diferentes.
- ◆ Los programadores se acostumbran al código, e inconscientemente puede evitar

algunos cuellos de botella en el programa cuando lo prueban. Es un proceso objetivo.

- ◆ El equipo interno siente la presión directa cuando hay atrasos en el desarrollo y una fecha de entrega límite que cumplir, por lo que puede no profundizar lo necesario al ejecutar la prueba.

Durante la etapa de pruebas es conveniente que el software se mantenga en un ambiente separado del entorno de desarrollo y lo ideal es preparar un ambiente de pruebas lo más parecido al que tendrá posteriormente en casa del cliente, para asegurar su correcto funcionamiento en esta etapa futura y esencial. Esto se puede lograr con la tercerización del servicio de pruebas, además de garantizar que el equipo que gestione las pruebas tenga la experiencia requerida y la formación necesaria para detectar una gran cantidad de errores, así como una metodología específica que le permite hacer su trabajo de manera correcta. (Sánchez and Garzás 2010)

Algunas instituciones han encontrado como una solución apropiada para aplicar los principios del outsourcing o tercerización de las pruebas de software, las Factorías de Pruebas. Han definido como Factoría de Pruebas a una organización cuya actividad de negocio principal consiste en proporcionar servicios de pruebas a empresas. Estas dispondrán de los recursos necesarios, tanto personas como procesos e infraestructura, que permitan proporcionar y gestionar de forma eficiente los servicios ofrecidos. (González 2010)

Algunos conceptos que están presentes en una factoría de prueba son: grupo dedicado que provee pruebas de software como un servicio compartido; emplea un conjunto de personas que se especializan en uno o múltiples tipos de prueba o uno o más dominios de negocio o tecnológicos. (Bizcaia 2010)

Estas definiciones que se han presentado son las que siguen de manera general las organizaciones que se dedican a brindar servicios de pruebas de software, aunque no siempre se utilice el término “factoría”. La esencia es que siguiendo el modelo de tercerización de pruebas se obtienen un conjunto de beneficios dentro de los que se pueden destacar: permite simplificar la gestión de los recursos de prueba; mayor control de la calidad, tiempo y costes de los servicios de aseguramiento; mayor productividad y reducción

de los costes de las actividades de verificación y validación del software debido a la industrialización del proceso; mayor especialización, ya que se dispone de recursos expertos en actividades y herramientas específicas de aseguramiento de la calidad y mayor capacidad de absorber de trabajo, al disponer de una gran flexibilidad en los equipos de pruebas. Este es el modelo bajo el cual se organiza el DPSW.

### 1.7 Algunas organizaciones que realizan pruebas de software en el mundo.

Como se ha explicado la calidad del software se ha convertido en un factor crítico, teniendo en cuenta un entorno cada vez más competitivo y clientes más exigentes. El desarrollo de software es una actividad que posee un impacto directo en todos los ámbitos de la sociedad moderna. La buena calidad de los desarrollos y la conciencia colectiva en el sector empresarial de la importancia de las áreas de pruebas han contribuido a que el mercado internacional proponga nuevos estándares de aceptación. El mercado de las pruebas de software crecerá hasta 100.000 millones de euros en 2014 desde la actualidad, según la consultora francesa PAC (Cusmai 2011).

Dentro de la calidad, las pruebas de software como actividad fundamental del control de la calidad, juegan un papel protagónico y como tendencia se ha generalizado la creación de entidades que se dediquen a probar software, bajo el concepto de la tercerización del servicio.

Como parte de la presente investigación se realizó un estudio de casos, basado en varias organizaciones que brindan servicio de pruebas de software, con el objetivo de conocer su concepción, características esenciales y si tenían relación con universidades. Se escogieron laboratorios de varias regiones, para hacer el análisis más abarcador. En la Tabla 3 se muestran las organizaciones estudiadas con mayor profundidad:

Institución	Laboratorios de Pruebas de Software	País
Hexaware Technologies (HEXAWARE)	Test Center of Excellence	India
Red de Parques Tecnológicos de Software, ParqueSoft Nation. (ParqueSoft)	ParqueSoft	Colombia
Choucair Testing. (Choucair)	Choucair Testing S.A.	Colombia
Concyteg. (CONCYTEG)	Laboratorio de desarrollo y pruebas de software en el estado	México

E-Quallity (e-Quallity)	e-Quallity	México
Cámara Uruguaya de Tecnologías de la Información y Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República de Uruguay (UdelaR), a través de la Fundación Julio Ricaldoni. (CES)	Laboratorio de Pruebas Funcionales. Centro de Ensayos de Software de Uruguay, (CES)	Uruguay
Instituto Nacional de Tecnologías de la Información. (INTECO)	Laboratorio Nacional de Calidad de Software, INTECO	España
Grupo CapGemini. (Sogeti)	Software Control & Testing, Sogeti	España
Universidad de Limerick. (SQRL)	Software Quality Research Laboratory (SQRL)	Irlanda

Tabla 3: Algunas organizaciones y laboratorios de prueba en el mundo.

Con la información adquirida a partir del estudio de las organizaciones, se pueden precisar algunos elementos importantes y que se tienen en cuenta de una forma u otra como generalidad en estas organizaciones:

- ◆ Establecen una estrategia de pruebas.
- ◆ Definen sus procesos, infraestructura, herramientas (automatizadas fundamentalmente) y técnicas.
- ◆ Establecen la gobernabilidad, las métricas y el marco de la comunicación.
- ◆ Realizan el plan de gestión del conocimiento para atender las demandas de recursos y al modelo de prestación de configuración centralizada.
- ◆ Desarrollan los recursos humanos. Contar con un equipo multidisciplinario para el diseño y ejecución de las pruebas.

En algunos casos se pudo constatar que tienen relación con universidades desde el punto de vista académico, para impartir conferencias, cursos y realizar otras actividades formativas. Solo en el caso del CES de Uruguay se utilizan los estudiantes más aventajados para formar parte de los equipos de prueba, pero a partir de una selección, no es una vinculación masiva de estudiantes. En el caso del SQRL se precisó que utilizan a estudiantes en el desarrollo de herramientas y en la parte investigativa de forma general.

En Cuba, hasta la creación de Calisoft, no existía ninguna organización que se dedicara por completo a los temas relacionados con la calidad y pruebas de software. Esto era algo que las propias empresas productoras de software debían garantizar. Con la creación de este

centro se garantiza este servicio, el cual comenzó brindándose a los proyectos de la UCI, pero hoy se proyecta para el resto de la Industria Cubana de Software.

El hecho de garantizar que los productos de software que se desarrollen en el país tengan la calidad requerida, tiene mayor impacto al definirse dentro de los Lineamientos de la Política Económica y Social del Partido y la Revolución, aprobados en el VI Congreso del Partido Comunista de Cuba, el lineamiento número 83 que precisa: Trabajar para garantizar, por las empresas y entidades vinculadas a la exportación, que todos los bienes y servicios destinados a los mercados internacionales respondan a los más altos estándares de calidad. (Lineamientos PCC 2011)

### **1.8 Evolución de las pruebas de software en la UCI.**

Desde el primer año de creación de la Universidad de las Ciencias Informáticas (UCI) se estudia la creación de un órgano evaluador, con el objetivo de garantizar la revisión de los productos desarrollados por los proyectos antes de ser entregados al cliente. En el segundo curso de la Universidad es que se evidencia con mayor fuerza la necesidad de realizar las pruebas de software, por lo que se crean grupos de investigación y se asignan tareas a diferentes especialistas para la concepción de este proyecto y llevarlo a la práctica, supervisados por la ya existente Dirección de Calidad de Software (DCS), debido a la premura de probar los productos desarrollados por los proyectos de exportación existentes, los cuales ya debían hacer entregas al cliente. Se crea en el 2005 un grupo de calidad central con estudiantes de 3ero y 4to año, los cuales pertenecían al Laboratorio de Pruebas de la DCS de la UCI que era guiado por un especialista de esta Dirección. Es en estos momentos se comienza a definir cómo hacer las pruebas.

Este modelo evoluciona y se decide en el año 2006 crear un grupo de calidad por cada facultad, formado a partir de los estudiantes del grupo central y nuevas incorporaciones, los cuales eran guiados por un Asesor de Calidad, nuevo rol que aparece en el proceso de pruebas. Se comienzan a utilizar los recursos de las facultades para las pruebas, guiados por los estudiantes de mayor experiencia y profesores. En un primer momento se hacían

pruebas cruzadas, o sea, los estudiantes no probaban artefactos elaborados en su propia facultad, pero los problemas que se fueron presentando y han sido mencionados en el presente trabajo, hicieron que se cambiara esta concepción, vinculando los grupos de calidad de la facultad a la liberación de los productos de su propia facultad, de manera que estuvieran más comprometidos con el desarrollo del trabajo, aunque implicaba el riesgo de que no se probara con la profundidad requerida, el que era mitigado por el trabajo intencionado de los especialistas del laboratorio al frente de las pruebas.

Se continua consolidando el trabajo del laboratorio como elemento clave en la liberación de productos software y entregables de la producción y se definen un conjunto de temas de investigación apoyados por las tesis de grados, una tesis de maestría y una tesis de doctorado, tutoradas por la Dirección de Calidad de Software, que se ve fortalecida y protagonista de un importante paso en la producción de la UCI.

Se comienzan a presentar un grupo de problemas que se especificaron en la situación problémica y por esto se decide cambiar la organización del laboratorio de pruebas. La concepción inicial de esta nueva forma de organización determinaba 1 Planificador, 4 grupos de trabajo: Grupo de Pruebas Exploratorias y Muestreo, Grupo de Diseño de Pruebas, Grupo de Herramientas Automatizadas y Grupo de Gestión de Pruebas y un Laboratorio de Pruebas Funcionales donde se ejecutaban las pruebas (Figura 6). De esta forma cada grupo tenía la responsabilidad de una parte del proceso de pruebas de liberación.

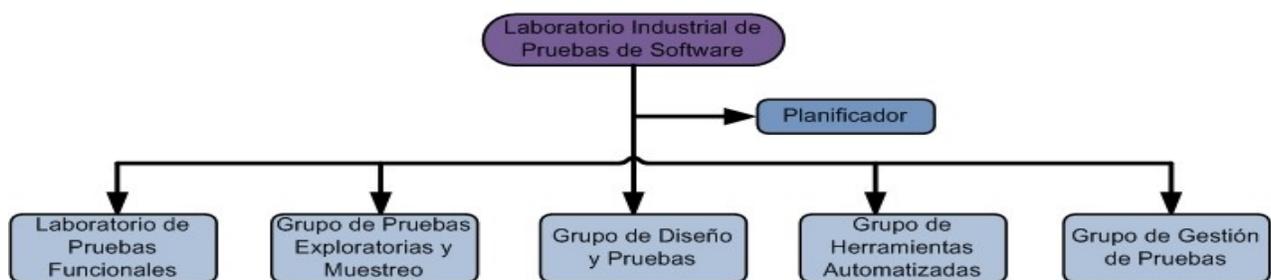


Figura 6: Organización inicial del Laboratorio Industrial de Pruebas de Software.

Teniendo en cuenta la experiencia adquirida con esta organización y la transformación de la Dirección de Calidad de Software en Centro de Calidad para Soluciones Informáticas (Calisoft), a finales del curso 2009-2010 se trabajó en una nueva organización del

Laboratorio, convirtiéndolo en el Departamento de Pruebas de Software de Calisoft (DPSW).

## **1.9 Conclusiones.**

Después de estudiar los conceptos relacionados con la calidad del software y las pruebas de software, se llegaron a enunciar definiciones propias de la autora que fundamentaron las pruebas de liberación. Se analizaron los principales modelos de calidad de software relacionados con los productos, determinando que la ISO/IEC 9126 es la norma más completa que define las 6 características de calidad que mayor presencia tienen en los modelos estudiados.

Se mencionan los modelos que adoptan las organizaciones en función de su estructura organizativa, los cuales guían la forma de desarrollar las actividades relacionadas con las pruebas de software. Se profundiza en la nueva tendencia del outsourcing o tercerización de pruebas, muy utilizada en el mundo y en la que se sustenta la formación del LIPS, al ser la que se adapta a sus características y condiciones.

Algunas organizaciones que se dedican a brindar servicios de pruebas de software fueron estudiadas y se pudo precisar que tienen características comunes como son: que establecen sus estrategias, definen procesos, infraestructura, herramientas automatizadas y técnicas, desarrollan los recursos humanos con el fin de contar con un equipo multidisciplinario y que esté especializado en temas relacionados con las pruebas de software. La relación con la universidad en el caso de algunas organizaciones se establece a partir de impartición de conferencias, clases o porque utilizan a estudiantes en investigaciones y/o desarrollo de herramientas, pero solo en uno de los casos los estudiantes se integran a los equipos de profesionales para el desarrollo de las pruebas y es solo los más aventajados. A pesar de que las organizaciones estudiadas tienen estos aspectos en común, sucede que el entorno en el que se desarrolla el LIPS y el hecho de tener una universidad productiva en este, lo hacen diferente del resto de las instituciones y son elementos novedosos que aportan valor a la concepción del laboratorio.

## 2 Propuesta de solución

### 2.1 Introducción

La concepción del Laboratorio de Pruebas existente en la Dirección de Calidad de Software fue evolucionando en función de los cambios que se realizaban en la vida productiva y docente de la UCI, así como el aumento de la demanda en las solicitudes de pruebas de liberación, determinada por el crecimiento de los proyectos de exportación y el hecho de que se incorporaran a este proceso los entregables de los proyectos nacionales y de informatización. Ante el escenario descrito, el proceso de pruebas de liberación definido no respondía a la realidad que se presentaba, ni la propia organización del laboratorio.

En el presente Capítulo se conceptualiza el Laboratorio Industrial de Pruebas de Software (LIPS), lo que conllevó a redefinir el proceso de pruebas de liberación, donde se incluye el uso de herramientas automatizadas tanto en la gestión de las pruebas como en la ejecución de estas, así como otros elementos novedosos que se insertaron en el proceso para resolver los problemas presentados en el laboratorio.

### 2.2 Departamento de Pruebas de Software.

La evolución de la Dirección de Calidad de Software (DCS) en el Centro de Calidad para Soluciones Informáticas (Calisoft), hizo que se reanalizara la estructura interna, por lo que se decide adoptar la mostrada en la Figura 7 siguiente:



Figura 7: Estructura de Calisoft.

Se decide redimensionar la actividad de las pruebas de software creando un Departamento de Pruebas de Software (DPSW), teniendo en cuenta elementos como el crecimiento de la

cantidad de especialistas, la cantidad de artefactos a probar que iba en aumento como tendencia, solo contando con las liberaciones a los proyectos de la UCI, pero con solicitudes ya realizadas de empresas productoras de software en el país, que se han mostrado interesadas en hacer pruebas de liberación a sus productos. La creación del DPSW es un paso estratégico, una proyección a futuro, por la importancia que tiene garantizar la calidad de los productos de software, para satisfacer a los clientes y ser más competitivos en el mercado.

Una vez creado el Departamento, se organiza su estructura interna, manteniendo el Laboratorio Industrial de Pruebas de Software (LIPS), pero con una concepción diferente y creando el Grupo de Ingeniería de Pruebas (GIPS). Este grupo define a partir de las características de calidad de la norma ISO/IEC 9126, los tipos de prueba a realizar y su diseño, haciendo herramientas más específicas para su utilización en el LIPS, donde se tienen en cuenta las características de los probadores.

La estructura del DPSW se muestra en la siguiente Figura 8:

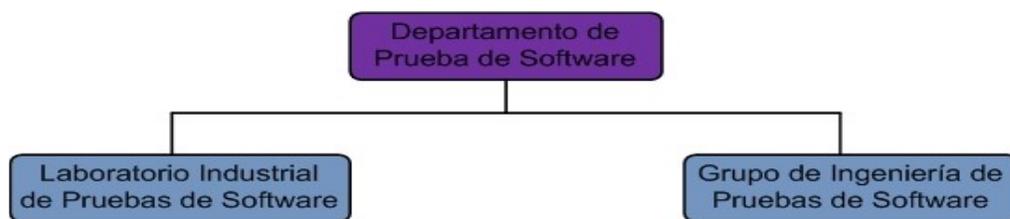


Figura 8. Estructura del DPSW.

### 2.2.1 Grupo de Ingeniería de Pruebas.

El Grupo de Ingeniería de Pruebas (GIPS) tiene la misión de definir y diseñar los tipos de prueba que deben realizarse en el laboratorio de pruebas. En el Capítulo 1 se precisaron los modelos y normas más reconocidos que establecen las características, subcaracterísticas y atributos de calidad para medir la calidad de un producto, estos son estudiados y analizados de forma continua por este grupo para lograr mejorar los procesos de pruebas.

Después de este análisis se decide que los especialistas se organizaran en el GIPS a partir de las características de calidad que define la norma ISO/IEC 9126, por considerarse la más

completa, y han trabajado en la adaptación o elaboración de diseños de prueba más específicos. En la Figura 9 Se muestra la organización de los especialistas en el GIPS:



Figura 9: Estructura del GIPS.

Al estudiar la norma ISO/IEC 25000, la cual constituye una actualización de la ISO/IEC 9126, se pudo precisar que es un poco más amplia que la anterior, pasando de 6 características de calidad a 8. Las 2 nuevas características están contempladas en la primera norma como subcaracterísticas de calidad, por lo que se han tenido en consideración de alguna manera en la organización de este grupo. La característica Seguridad se ha trabajado dentro de Funcionalidad y siempre se le ha dado un peso importante; mientras que Compatibilidad se tenía en cuenta en Portabilidad.

Esta organización permite lograr una especialización de los profesionales del DPSW, garantizado que se puedan definir y llevar a la práctica tipos de pruebas asociados a las características que deben evaluarse en un producto para determinar su calidad. En la organización anterior del laboratorio, solo se realizaban pruebas de funcionalidad y las de seguridad a nivel de acceso por roles únicamente, de manera que las demás iniciativas que se iban tomando, estaban asociadas a los estudios individuales de los especialistas y la experiencia que se iba adquiriendo en el desarrollo de las pruebas. La decisión de conformar este grupo, de conjunto con la nueva conceptualización del LIPS, aporta como elemento importante el poder diversificar los tipos de prueba a realizar, garantizado que estén asociados a características y subcaracterísticas de calidad definidas por una norma internacional, lo cual permite que se pueda evaluar en qué medida un producto cumple o no con estos aspectos.

Al analizar los servicios prestados por otras organizaciones que brindan servicios de pruebas

de software, como se precisa en el Capítulo 1, se destaca que se realizan varios tipos de prueba, dándole una mayor confianza a los clientes y mostrando mayor especialización. Además se considera muy importante la automatización de estas pruebas, teniendo en cuenta que permiten agilizar el proceso y simular algunos casos que de manera manual sería muy difícil o imposible poder realizar.

Se han identificado tipos de prueba asociados a las características de calidad, algunos que ya se están desarrollando en el LIPS y otros que están en definición, como se muestra en el Anexo 1. Como resultado importante de este grupo se tiene la elaboración de un documento, que es parte de la estrategia de pruebas del DPSW, donde se detalla por cada característica de calidad los tipos de prueba ya definidos, teniendo en cuenta los siguientes elementos:

- ◆ Concepto.
- ◆ Descripción de la prueba (se precisa cómo hacer la prueba, si puede ser ejecutada por los estudiantes en el LIPS o deben desarrollarla especialistas y cómo se deben hacer las pruebas de regresión para este tipo de prueba).
- ◆ Tipos de artefactos a los que se puede realizar.
- ◆ Herramientas definidas para desarrollarla (automatizadas o manuales).
- ◆ Buenas prácticas para el desarrollo de la prueba.

Estos aspectos se determinaron aplicando el método de investigación grupo focal, que puede definirse como una discusión cuidadosamente diseñada para obtener las percepciones sobre una particular área de interés (Krueger 1991), en reunión con todos los especialistas del DPSW, donde cada uno aportó las ideas que tenía a partir de las investigaciones realizadas de manera individual o en los equipos que trabajan las características de calidad.

### **2.2.2 Laboratorio Industrial de Pruebas de Software.**

El Laboratorio Industrial de Pruebas de Software (LIPS) se organiza a partir del análisis realizado de los principales problemas que se iban presentando con la organización anterior del laboratorio, los que se fueron identificando a partir de encuentros desarrollados entre los especialistas, utilizando el método grupo focal.

Los problemas que se detectaron estuvieron relacionados con la pobre asistencia de los estudiantes de los grupos de calidad a los turnos de trabajo, lo que estaba relacionado con que se planificaban otras actividades de corte docente o extensionista, las cuales eran de mayor interés de los estudiantes teniendo en cuenta que estas les repercutían en su evaluación integral en la facultad, o en las notas de las asignaturas y los turnos de prueba no tenían impacto alguno en estos aspectos. También ocurría que los grupos de calidad tenían pocos estudiantes, la facultad que más tenía era 30, pero esta cantidad iba disminuyendo en el tiempo y por otra parte la cantidad de artefactos a probar iba aumentando, dándose la necesidad de que se comenzaran a probar los entregables de los proyectos nacionales y de informatización que iniciaban compromisos más consolidados con los clientes. En la Figura 10 se muestra la cantidad de artefactos probados desde enero de 2008 hasta julio de 2010:

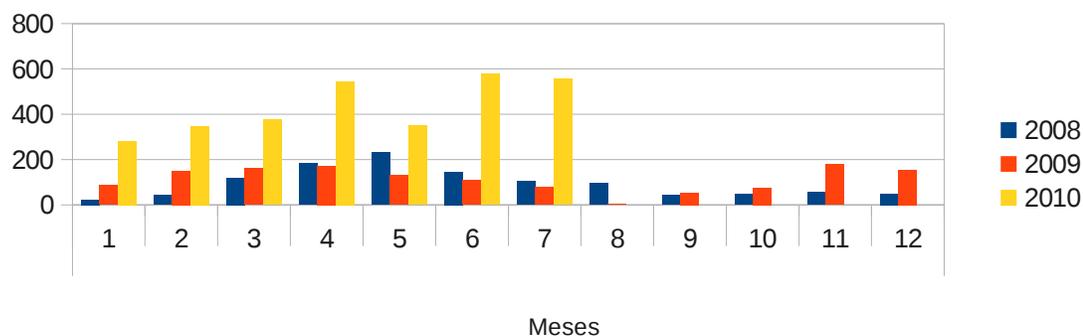


Figura 10: Cantidad de artefactos probados por meses.

Los grupos de calidad de las facultades no tenían estudiantes de 1ero y 2do año como generalidad, pues el tiempo en el que se le podía planificar la producción estaba condicionado por la asignatura Práctica Profesional (PP). Esta situación no permitía que desde los primeros años de la carrera se fuera incorporando en los estudiantes una cultura de calidad, basada en la experiencia que adquirieran probando artefactos de los proyectos reales desarrollados en la UCI.

De forma general los estudiantes no tenían motivaciones para asistir a los turnos de prueba

ni se contaban con mecanismos para poder ejercer un mayor control sobre las tareas que realizaban. Esta situación comenzó a repercutir en la calidad en la ejecución de las pruebas, que no siempre era la más óptima y tampoco se contaba con un mecanismo de peso para evaluar el trabajo, de manera que contara como parte de su formación y le repercutiera en los resultados productivos o académicos.

El cúmulo de estos problemas comenzó a provocar que no siempre pudieran asumirse todos los procesos de prueba, por no tener fuerza de trabajo disponible todo el tiempo. En los casos que se iniciaba un proceso, una iteración podía extenderse más de lo planificado según los horarios que podían realizarse a partir de las demás afectaciones de los estudiantes y la asistencia a los turnos de trabajo. Los estudiantes eran asignados a pruebas específicas y eran los únicos que podían continuar esta, lo que influía directamente en esta extensión del tiempo.

Después de tener identificados todos los problemas que afectaban el desarrollo de las pruebas de liberación, se continuaron haciendo discusiones entre los especialistas del laboratorio con el fin de encontrar soluciones que pudieran contribuir a elevar la calidad de las pruebas, contar con fuerza de trabajo permanente y tener un mecanismo de evaluación del desempeño de los estudiantes que formara parte de su docencia.

Se comenzaron a plantear algunas hipótesis a partir de todos los análisis realizados de la situación presentada. En un primer momento se consideró que se podían vincular al LIPS todos los estudiantes de 1ero y 2do año, a través de la asignatura Práctica Profesional y que los turnos de trabajo les repercutiera en la nota de esta asignatura. Se hicieron varios encuentros con los máximos responsables de la formación en la UCI y la dirección del Departamento Docente Central de PP, para llegar a acuerdos que permitieran utilizar a estos estudiantes. Se comenzó a trabajar con todos los estudiantes de la sede central de los dos años, pero la experiencia demostró que los estudiantes de 1er año no tenían la madurez necesaria para el trabajo en el LIPS, pues acababan de entrar a la Universidad y estaban en un proceso de adaptación a la forma de enseñanza y la vida universitaria de forma general. Teniendo en cuenta estos resultados se consideró que podía incluirse como parte de la

asignatura PP1, en el segundo semestre del curso de 1er año, un tema relacionado con las pruebas de software, donde se le presentan los elementos esenciales relacionados con estas y realiza una actividad práctica que les demuestra cuál será su trabajo en el siguiente año, o sea, es una base para que enfrenten el trabajo en el LIPS. Estas clases de PP1 fueron preparadas por especialistas del DPSW y son impartidas por los profesores de las facultades con el asesoramiento del Departamento.

A pesar de que se establecieron acuerdos con la Vicerrectoría de Formación y se comenzaron a considerar los turnos en el LIPS como parte de una asignatura (PP2), no tenía el peso necesario y se seguían presentando muchos de los problemas que se habían identificado con anterioridad, aunque se tenía una idea más clara de cuál debía ser el camino para su solución.

Se consideró que el proceso de pruebas de liberación debía ser como un proceso industrial donde se tuvieran elementos de entrada a un grupo de actividades consecutivas que fueran ejecutadas por los diferentes roles que intervienen en el proceso y como salida se tendría un artefacto liberado. De esta manera es un proceso que puede ejecutarse constantemente, pudiendo utilizar los recursos de manera más óptima y no condicionando un grupo de probadores a una prueba específica como sucedía antes, sino que se contaba con una fuerza permanente para las pruebas y se asignaba el trabajo en función de los cronogramas de los procesos de liberación que se estaban ejecutando.

Otra de las hipótesis planteadas era que si se garantizaba que los diseños de casos de pruebas (DCP) y las listas de chequeo (LCH) que debían utilizar los estudiantes en el LIPS estaban correctas y bien detalladas, estos serían capaces de detectar una mayor cantidad de no conformidades (NC) en los artefactos, a pesar de que no tuvieran todos los conocimientos técnicos necesarios. Es por esto que las plantillas que se tenían definidas para estas herramientas fueron revisadas por especialistas del laboratorio y se hicieron las adecuaciones necesarias para que las herramientas de prueba estuvieran mejor diseñadas. Se consideró que para poder garantizar esta actividad, era necesario además incluir como parte del proceso de pruebas un subproceso de Diseño de Prueba, donde se pudiera revisar

por especialistas los DCP y las LCH antes de ser utilizadas en el LIPS.

Mediante el método científico observación participativa, se comenzó a obtener información de los talleres que se realizaron para el análisis y definición del nuevo modelo de integración formación, producción e investigación (MIFPI) que se comenzaría a aplicar en la UCI, la cual fue analizada por los especialistas del DPSW y la dirección de Calisoft, para poder proponer cómo se debía insertar el LIPS en este modelo.

### **2.2.2.1 Elementos fundamentales y novedosos en la concepción del LIPS.**

Todo el análisis realizado con anterioridad dio como resultado que se introdujeran en el proceso de pruebas de liberación elementos fundamentales y novedosos para el proceso, que han conceptualizado el LIPS y han marcado la diferencia con el proceso que se desarrollaba con anterioridad. Estos elementos son:

1. Funcionamiento del LIPS como fábrica, proceso industrial.

Tras la Revolución Industrial se generaliza la fábrica, lugar en el que se persigue aumentar la producción para ganar economías de escala aplicando grandes cantidades de capital intelectual, trabajo y tecnología. El sistema fabril permitió la generalización de la división del trabajo. Las sucesivas mejoras organizativas permitieron la especialización de funciones, la estandarización de procedimientos y la producción en cadena. (Piza Escalante 1998)

Dentro de una fábrica se llevan a cabo procesos industriales que pueden definirse como una secuencia de actividades interrelacionadas entre sí, que partiendo de unas entradas o recursos, las transforma sistemáticamente, generando un resultado o producto con valor agregado. (INTESEG)

En el Diccionario de la Lengua Española de la Real Academia Española (RAE) se encontraron los conceptos siguientes:

Proceso: Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial.

Industrial: Perteneiente o relativo a industria.

Industria: Conjunto de operaciones materiales ejecutadas para la obtención, transformación o transporte de uno o varios productos naturales.

Fábrica: Establecimiento dotado de la maquinaria, herramienta e instalaciones necesarias para la fabricación de ciertos objetos, obtención de determinados productos o transformación industrial de una fuente de energía.

Partiendo de los conceptos anteriores se puede precisar que el propósito de un proceso industrial es producir transformaciones a partir de las entradas, permitiendo o facilitando interrelaciones entre recursos a partir de las actividades que lo conforman, generando de esta forma una salida con valor agregado. Los componentes esenciales de estos procesos son: Entradas, Ciclo de transformación y Salidas.

Aplicando las definiciones anteriores al laboratorio de pruebas, se consideran entradas la solicitud de pruebas realizada por el proyecto y el artefacto a probar; dentro del ciclo de transformación estaría la secuencia de actividades que se realizan para probar el artefacto, el cual va sufriendo modificaciones a partir de las NC que se detecten y como salidas se obtendrían el artefacto con mayor calidad, el Acta de Liberación y una Minuta de Cierre con los elementos principales del proceso.

Con estos conceptos incorporados en el laboratorio se aumenta la capacidad para brindar el servicio de pruebas de liberación, contando con gran cantidad de capital intelectual (alrededor de 1700 estudiantes de 2do año), trabajo (demanda cada vez más alta de pruebas de liberación) y tecnología (al menos básica en los 2 laboratorios de prueba y con dos servidores para el montaje de los entornos de prueba). Esta mejora organizativa ha permitido la especialización de los profesionales (GIPS con especialistas dedicados a la investigación y definición de los tipos de prueba a desarrollar), una mayor estandarización del procedimiento (al tener los recursos disponibles de manera constante para probar) y la producción en cadena (entran las solicitudes y van pasando por las diferentes actividades definidas, hasta que se cierra el proceso).

En la siguiente gráfica (Figura 11) se representa el proceso industrial del LIPS:

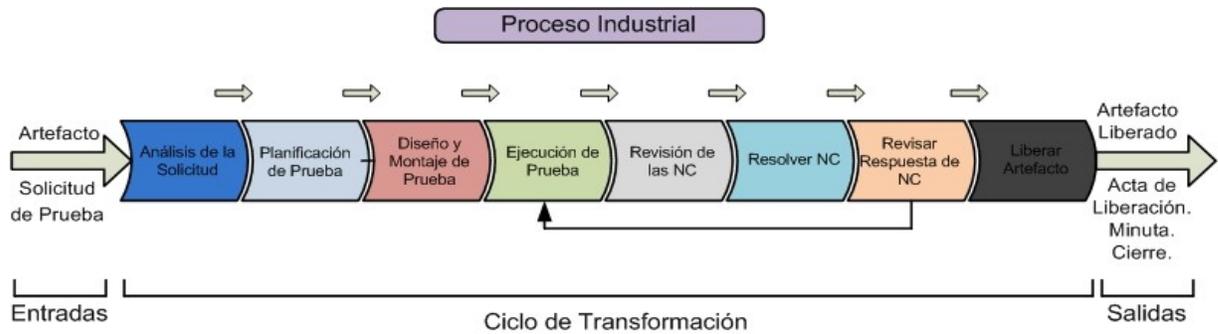


Figura 11: Representación proceso industrial del LIPS.

## 2. Utilización de estudiantes.

Se estudiaron laboratorios de prueba que siguieran el modelo de outsourcing o tercerización de pruebas, que es bajo la filosofía que está organizado el LIPS, precisando cómo es su relación con la universidad. En algunos casos se pudo constatar, como se ha precisado en el Capítulo 1, que establecen puntos de contacto con la universidad para postgrados, maestría, conferencias, actividades de formación de manera general. En un caso se utilizan los estudiantes más aventajados a partir de una selección realizada, directamente en las actividades de prueba, pero no se generaliza para todos los estudiantes, y en otro se vinculan en temas de investigación y desarrollo de herramientas para las pruebas.

No se encontró ningún caso en el que se utilicen de forma masiva estudiantes que desempeñen el rol de probador. En intercambios realizados con profesionales de otros laboratorios de prueba en eventos internacionales en los que se ha presentado la concepción e implementación del LIPS, se ha podido precisar a través de entrevistas a profundidad realizadas, que se considera una propuesta novedosa y que ha logrado llevar a la práctica lo que otras instituciones han intentado, teniendo en cuenta sobretodo que es una fuerza de trabajo más barata y que trae beneficios una vez que el estudiante se gradúa y comienza a producir, por la cultura de calidad que adquiere, pero que por diversos problemas no les ha sido posible llevar a la práctica. Dentro de estos impedimentos se mencionan que los modelos de formación de las universidades no

favorecen que se puedan insertar a los estudiantes en el proceso de pruebas como probadores, a pesar de que se ha propuesto, por ejemplo en España, que se posibilite que hagan el trabajo en los laboratorios de calidad como parte de su proyecto de grado. Un elemento importante relacionado con el planteamiento anterior, que incide en que no se haya podido concretar la idea, es que la formación en las universidades de forma general no se concibe desde la producción.

En una entrevista a profundidad realizada a un graduado de una universidad argentina precisó que si hubieran tenido una vinculación como la que tienen los estudiantes en la UCI con la práctica de la producción y las pruebas de software en proyectos reales, de seguro les resultaría más fácil encontrar una ubicación laboral después de graduados, pues ya contarían con un aval para optar por algunos puestos de trabajo, lo que no sucede hoy porque se gradúan solo con conocimientos teóricos y no con prácticas en proyectos reales.

De manera general se ha podido percibir aceptación del LIPS y el reconocimiento de que este modelo es más barato para las organizaciones, si se compara con lo que cuesta la contratación de profesionales para realizar las pruebas de software. A pesar de que se conocen estos elementos, ha resultado muy difícil en otras partes del mundo implementar alguna propuesta similar, sobretodo porque no se cuenta con un modelo de universidad productiva como el de la UCI.

Los estudiantes de 2do año al realizar el trabajo en el LIPS se vinculan directamente a la producción real, pues los artefactos que son probados en las clases pertenecen a proyectos reales desarrollados por estudiantes de años superiores y profesores de los Centros de Desarrollo de la UCI, o de alguna empresa productora de software del país, algunas de las cuales han comenzado a solicitar el servicio de pruebas de liberación. De esta manera desde el ciclo básico los estudiantes se insertan en el proceso productivo de la UCI.

Al detectar no conformidades en los diferentes tipos de artefactos, los estudiantes van incorporando a su formación elementos de Ingeniería de Software que le serán impartidos

con mayor profundidad en esta asignatura, la cual comienzan a recibir en el 2do semestre de 2do año. Se considera muy importante que se inicien en el proceso de desarrollo realizando pruebas de software, pues de esta manera incorporan en su formación una cultura de calidad, que es esencial para su posterior desarrollo en otros roles dentro de un proyecto productivo. Además como parte de las clases en el LIPS, se les enseñan buenas prácticas para ejecutar los diferentes tipos de prueba que aprenden, dadas incluso en función de artefactos específicos; se precisan las NC más comunes que se pueden detectar en función del artefacto a probar, para que enfoquen su atención en estos detalles y son capaces de clasificar las NC que encuentran; aspectos que contribuyen a que incorporen estos conocimientos y puedan aplicarlos más adelante, previendo estos tipos de errores y construyendo desde el inicio aplicaciones con mayor calidad.

Durante su desempeño en el LIPS, los estudiantes interactúan con herramientas automatizadas como el Subversion (SVN) y el GESPRO, que se utilizan para la gestión del trabajo que deben desarrollar en el turno de clases y de las NC que se detectan al probar los diferentes artefactos respectivamente. Estas herramientas son utilizadas en los proyectos productivos de la UCI para la gestión de los productos de trabajo y de las tareas que deben realizar los miembros de estos proyectos en función del cronograma planificado, por lo que interactuar con ellas en el LIPS permite que se familiaricen con su funcionamiento y que tengan una base para el trabajo que deben realizar con estas en años posteriores. También se utiliza el Entorno Virtual de Aprendizaje (EVA) para el intercambio de los Trabajos Independientes (TI) y de la bibliografía necesaria para desarrollar estos, los cuales son orientados en los turnos de clase y elaborados por los estudiantes en el turno no presencial (NP) que tienen en la semana posterior al laboratorio.

En los Anexos 2 y 3 se muestran imágenes de la organización del SVN, herramienta de la que los estudiantes descargan las orientaciones y recursos que necesitan para el desarrollo de la prueba y de un ejemplo de una NC redactada por un estudiante respectivamente.

### 3. Vínculo con la docencia.

La Universidad de las Ciencias Informáticas para cumplir con su misión estratégica de formar profesionales comprometidos con su Patria, altamente calificados en la rama de la informática y de producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, requirió de cambios profundos en su modelo, transitando hacia la integración de los procesos de formación, producción e investigación. (Lavandero, Verdecia, Martínez, and Díaz 2008)

Este nuevo modelo establece como estrategias esenciales la formación centrada en el aprendizaje y la evaluación de competencias profesionales en el ciclo profesional, sobre la base de la certificación de roles a partir del trabajo realizado en los proyectos, la cual se realiza a través de la asignatura Práctica Profesional.

Teniendo como base la información obtenida a través de la observación participativa en varios encuentros donde se comenzaron a definir los elementos esenciales del nuevo modelo y talleres realizados para comenzar a diseñar los roles que debían ser certificados, se consideró que el LIPS podía insertarse en el modelo si se lograba que los estudiantes de 2do año se vincularan al laboratorio, a través de la asignatura PP2 y con el fin de que certificaran el rol de probador. Esta propuesta fue valorada y aprobada, por lo que la concepción del LIPS tuvo dentro de sus consideraciones la relación con el modelo de formación. El único rol que se certifica en el ciclo básico del estudiante es el de probador. En la Figura 12 se muestra un esquema general del MIFPI, donde se inserta el LIPS:

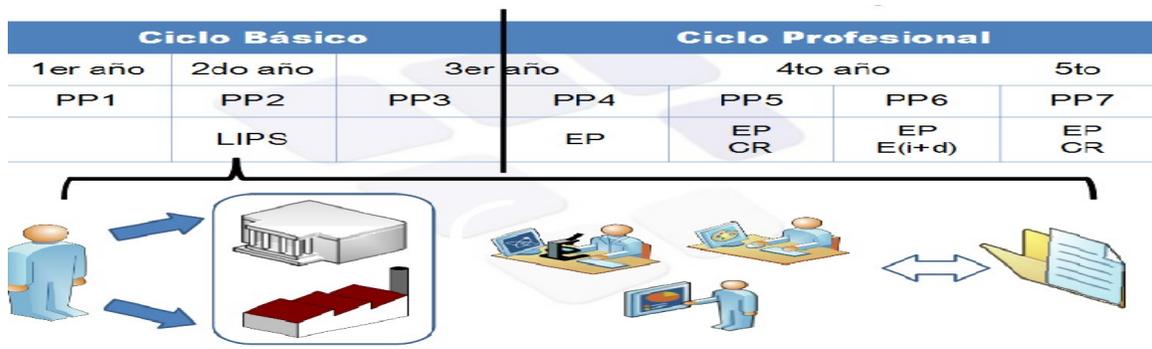


Figura 12: El LIPS en el MIFPI de la UCI a través de la asignatura PP2.

Las clases que se imparten en el LIPS se han clasificado como Prácticas Laborales, teniendo en cuenta que los estudiantes estarán vinculados a prácticas reales, en este caso a las pruebas de software, bajo el principio de estudio- trabajo, donde este tipo de clase se ha considerado por algunos especialistas como la más completa por su integralidad, siempre que esta sea planeada, estructurada y vincule a las demás de forma sistémica. (Martínez Pichardo, Verrier, Benítez, Aguilera, et al.)

Partiendo de la experiencia adquirida en los inicios del LIPS, se pudo constatar que si los turnos duraban menos de 4 horas no eran factibles, pues no se podía concluir la ejecución de un diseño de caso de prueba complejo, por lo que el trabajo de los estudiantes en ocasiones quedaba a medias. Además debe considerarse tiempo en la clase para otros elementos que han sido definidos en la estructura de las prácticas laborales desarrolladas en el LIPS. Las prácticas laborales están consideradas como tal en los P1 elaborados para la asignatura de PP2 por el Departamento Docente Central de Práctica Profesional, de conjunto con especialistas del DPSW; también se ha considerado un turno no presencial que se planifica en la semana siguiente al turno en el LIPS, en el que los estudiantes deben elaborar el Trabajo Independiente orientado por el profesor en el turno de clase.

Se monitorea y mide de manera permanente el desenvolvimiento de los estudiantes y el esfuerzo que dedican a una cantidad de trabajo asignada en el laboratorio, de manera que se retroalimente y perfeccione el método de estimación elaborado por especialistas del DPSW, mediante el cual se planifica el tiempo que dura una iteración de pruebas en el LIPS, a partir del tamaño del artefacto a probar, su complejidad y la cantidad de probadores y PC necesarias.

Los estudiantes durante el desarrollo de la clase son ubicados en dúos en cada puesto de trabajo, de manera que se fomente el trabajo en equipo, se logre que intercambien los puntos de vista sobre una NC detectada y que complementen entre ellos los conocimientos y habilidades que han adquirido. Ha sido una buena práctica utilizada desde que comenzó a funcionar el LIPS, por lo que se ha mantenido dentro de la

concepción del laboratorio. La parte investigativa del LIPS, materializada a través de los Trabajos Independientes, se realiza de forma individual y están relacionados con las 8 prácticas laborales por las que debe transitar un estudiante durante todo el curso en el LIPS. Estas prácticas están relacionadas con los tipos de prueba que deben realizar los estudiantes: pruebas funcionales, prueba de instalación, prueba de configuración, prueba de usabilidad, prueba de seguridad (nivel básico, solo verificando los niveles de acceso según los roles definidos en la aplicación), prueba de recuperación y tolerancia a fallas, prueba de regresión y evaluación estática.

La frecuencia de las clases en el LIPS para cada estudiante es mensual, determinada por la cantidad de horas definidas para la asignatura en su totalidad, de las cuales 32 horas como máximo están destinadas a las prácticas laborales y 12 horas para los turnos no presenciales en el curso.

Por las peculiaridades que tienen las clases en el LIPS y como medio para garantizar que pueda ser bien planeada, se ha definido una estructura para esta, donde se puntualizan los elementos que no deben faltar en el desarrollo de una práctica, incluyendo los aspectos esenciales que deben considerarse cuando se evalúe el trabajo del estudiante en cada clase. Esta estructura puede consultarse en el Anexo 4.

La asignatura de PP2 es impartida por dos profesores en cada grupo de clases, pues los especialistas de Calisoft son los responsables de las clases en el LIPS y hay otro profesor que se considera el principal de la asignatura, que imparte el resto de los contenidos. La comunicación entre los dos profesores es permanente y los estudiantes son evaluados de forma integral en la asignatura.

La evaluación de los estudiantes en el LIPS es guiada por el proceso de certificación del rol de probador, el cual ha sido definido como parte del modelo de integración formación, producción e investigación como se puede evidenciar en (Verdecia, Portuondo, Lavandero, and Arencibia 2011). El estudiante puede adquirir a partir de su desempeño en el LIPS uno de los 3 niveles definidos: básico, intermedio o avanzado, a partir de desarrollar la competencia específica: Ejecutar la prueba generando, clasificando y

describiendo las No Conformidades (NC) detectadas de acuerdo a las normas y directivas de Calisoft.

Todas las semanas se realiza la Preparación Metodológica (PM) con los profesores, donde se presenta la planificación de la semana siguiente, especificando los tipos de artefactos a probar en cada turno de trabajo. En este espacio se discuten los elementos esenciales que deben presentarse a los estudiantes del artefacto que probarán; las NC más comunes que se presentan en este; se repasa la estructura de la clase y se precisan los resultados de los controles a clases realizados, destacando cuáles son los problemas que se han presentado. Los controles a clase se realizan a partir de lo definido en el Reglamento Docente Metodológico: “El control de la actividad docente (Clase, práctica laboral y trabajo investigativo de los estudiantes) es aquel que se realiza a una forma organizativa del proceso docente, prevista en el horario oficial de los estudiantes, y dirigida a comprobar el logro de los objetivos propuestos” . (MES)

Una vez al mes se debe realizar un encuentro donde todos los profesores del LIPS se reúnen, para debatir temas metodológicos relacionados con las clases en el laboratorio, socializar experiencias positivas, realizar clases abiertas y otros tipos fundamentales del trabajo docente-metodológico, que contribuyan a la preparación pedagógica de los profesores, según establece el Reglamento Docente Metodológico (MES). Se ha definido además de conjunto con la Vicerrectoría de Formación y el Departamento Docente Central de PP, que los profesores del LIPS participen mensualmente en una de las PM de las facultades, de manera que intercambien directamente con los otros profesores de PP2 sobre la evolución de los estudiantes y la sinergia entre los temas de la asignatura.

#### 4. Automatización del laboratorio.

En el proceso anterior del laboratorio no se tenían en cuenta herramientas automatizadas ni para la gestión de las pruebas, ni para ejecutarlas. Se realizaron varios encuentros entre los especialistas del laboratorio, para presentar ideas y propuestas encaminadas a insertar herramientas automatizadas en el proceso, pues es una práctica internacional que disminuye el tiempo de las pruebas y contribuye a la optimización en la utilización de

los recursos.

- ◆ Gestión de las NC y solicitudes de prueba.

Se comenzaron a considerar herramientas que permitieran gestionar las NC de manera más ágil, pues estas se registraban en documentos generados por cada probador individualmente, los cuales debían ser revisados y agrupados en un solo reporte de NC por el especialista al frente de la prueba, de forma que se entregara al proyecto un solo documento con todos los errores detectados. El equipo de proyecto una vez resueltos las NC debía devolver el informe entregado, precisando en cada NC si se había resuelto o no procedía, y en este último caso debía precisar los motivos de esa clasificación. Con este documento el especialista al frente de las pruebas debía realizar la prueba de regresión y en los casos en los que no procedía, se organizaba un encuentro con el equipo de proyecto para discutir cada una. Este era el ciclo que se seguía para la gestión de las NC, el cual iba a estar ocurriendo hasta que no se detectaran NC en una iteración, era bastante engorroso y consumía un tiempo considerable de la prueba. (González Jorrín 2007)

Buscando una solución a los problemas presentados en la gestión de las NC se estudió la posibilidad de utilizar la herramienta GESPRO (basada en el Redmine), a partir de que se definió por parte de la máxima dirección de la producción en la Universidad que era la herramienta que debía utilizarse para la gestión de los proyectos. Después de analizar sus funcionalidades y la lógica de funcionamiento, se pudo llegar a la conclusión de que haciendo algunas adecuaciones a los campos que se mostraban y los flujos de las actividades, se podía no solo gestionar las NC de las pruebas de liberación, sino también configurarla para que se realizaran las solicitudes de prueba de liberación, las cuales se enviaban vía correo electrónico.

Para el caso de la solicitud, se hizo una revisión de los datos que eran necesarios que se plasmaran en esta, tomando como punto de partida la plantilla ya definida, e incorporando algunos elementos que se consideraban importantes, a partir de los cambios que se realizaron en el proceso. Se elaboró una Guía para la Solicitud de pruebas en el Redmine

y una Plantilla para adjuntar a la Solicitud de pruebas en el Redmine (Anexo 5). Además se adaptó la Plantilla de Solicitud de pruebas de liberación a partir de la información que se tiene en cuenta en las plantillas anteriores, la cual debe enviarse por correo electrónico y es solo para los casos en los que el proyecto que requiera hacer la solicitud no tenga acceso al Redmine por encontrarse fuera de la UCI. Los Estados por los que transita una solicitud de pruebas durante todo el proceso en función de la etapa en la que está se encuentre son: Nueva, Rechazada, Pospuesta, Asignada, En Curso, Detenida, Abortada, Liberada y Cerrada. Se muestra una vista de las Solicitudes de liberaciones en el Redmine en el Anexo 6.

Para la gestión de las NC se configuraron los campos necesarios, quedando de la siguiente forma:

- ◆ Tema: se precisa el nombre del DCP que se está probando (aplicación) o el nombre del documento (documentación).
- ◆ Descripción: se describe la NC detectada, precisando el lugar donde se encontró.
- ◆ Estado: la NC transita por varios estados, los cuales son graficados en el siguiente esquema (Figura 13) y son modificados por los roles Probador, Especialista al frente de la prueba y Responsable del proyecto, a partir de los permisos que tenga cada cual.

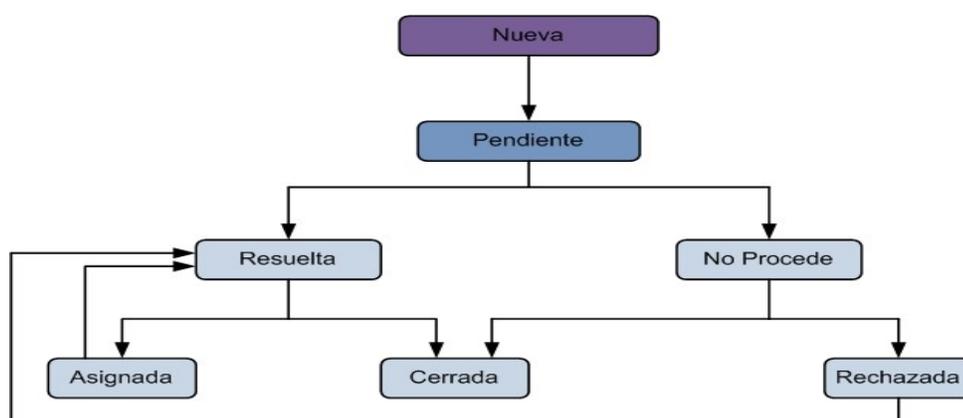


Figura 13: Cambios de estado de las NC.

- ◆ Asignado a: las NC pueden estar asignadas al Especialista al frente de la prueba o al Responsable del proyecto, en dependencia del estado en el que se encuentre.
- ◆ Tipo de error: se clasifican las NC según si el artefacto es aplicación (A) o documentación (D) en Ortografía (A y D), Redacción (A y D), Error de idioma (A y D), Error de interfaz (A), Formato (A y D), Error técnico (D), Excepción (A), Funcionalidad (A), Validación (A), Opciones que no funcionan (A), Diseño de Caso de Prueba (D), Correspondencia con otro artefacto (A y D), Seguridad (A) y Recomendación (A y D).
- ◆ Etapa: se precisa la iteración en la que se detectó la NC.
- ◆ Respuesta a NC: es un campo que se llena por parte del responsable del proyecto de manera obligatoria en los casos que se responda la NC como No Procede, explicando los motivos de esta respuesta. Puede ser utilizado cuando la NC haya sido Resuelta, pero se considera que debe aclararse algo con relación a la solución dada.
- ◆ Fichero: se debe subir una imagen como evidencia de la NC detectada, señalando el error.

Utilizando el Redmine para la gestión de las NC se ha disminuido el tiempo desde que se termina de ejecutar la iteración, a que se entregan todos los errores detectados al equipo de proyecto, si se tiene en cuenta que en la medida que el Especialista al frente de la prueba va revisando las NC registradas por los estudiantes en el LIPS, las va asignando al Responsable del proyecto y este puede ir las consultando y resolviendo en paralelo, sin necesidad de esperar a que estén todas asignadas.

- ◆ Gestión del Expediente de Prueba y del trabajo a realizar en los turnos del LIPS.

En la concepción del laboratorio de pruebas anterior, los documentos que se iban generando durante las pruebas, las herramientas que se utilizaban para probar, así como las versiones de los artefactos en pruebas (elementos que conforman el Expediente de Prueba), se gestionaban a través de un Repositorio de Datos que estaba dividido en dos áreas fundamentales, el área de comunicación e intercambio y el área de

almacenamiento histórico. La gestión en este repositorio se hacía a través de carpetas, a las que se daban permisos específicos en función del rol que jugaban en el proceso. (González Jorrín 2007)

Esta manera de gestionar los Expedientes de Prueba no era óptima si se tiene en cuenta que no podía hacer una correcta gestión de versiones y el tener carpetas compartidas en la PC definida como servidor, tenía muchos riesgos desde el punto de vista de la seguridad informática, a pesar de que se controlaban permisos y se trataba de restringir el acceso.

Se propuso a partir de discusiones entre el equipo de especialistas del LIPS, utilizar el Subversion, una herramienta muy utilizada para la gestión de configuración y que a partir de los elementos estudiados y presentados en el colectivo, se consideró adecuada. La estructura de las carpetas en el SVN para los Expedientes de Prueba se muestra en la siguiente imagen (Figura 14):



Figura 14: Estructuras de las carpetas en el SVN para los Expedientes de Prueba.

En el SVN también se publican las planificaciones de los turnos en el LIPS, para que los profesores sepan el horario de clases que le corresponde, el laboratorio, la prueba a realizar y el especialista al frente de esta. La información es publicada todos los viernes pues la planificación del LIPS se realiza semanal, en función de las facultades que les corresponda asistir a las clases y las pruebas que deban ser ejecutadas a partir de los cronogramas de las liberaciones.

A través del SVN también se preparan los puestos de trabajo de las pruebas que deben realizarse en cada turno del LIPS y es desde este espacio que los estudiantes descargan las orientaciones del trabajo a realizar, el artefacto (en los casos que sea documentación) y la herramienta a utilizar para la prueba. La información se ha organizado a partir del laboratorio, la sesión del turno de trabajo, el puesto de trabajo, artefacto y herramienta.

- ◆ Herramientas para la virtualización del entorno de prueba.

Para el montaje de los entornos de prueba se ha decidido virtualizar los servidores, de manera que se logre un uso más eficiente de los recursos con los que se cuenta, que en este momento son reducidos, pues solo se tienen 2 servidores con 8 y 4 GB de RAM, para montar todas las aplicaciones que son solicitadas para las pruebas de liberación en el LIPS.

Las herramientas que se utilizan para la virtualización son:

- VMWare: Potente herramienta de virtualización probada en entornos de la Universidad. Es multiplataforma y permite la ejecución concurrente de máquinas virtuales.
- vSphere Client: Capacidad para configurar máquinas virtuales hasta 4 CPUs virtuales cada una. Soporte para hardware de servidores de 64 bits hasta 256 GB de memoria RAM física. Incluye la seguridad integrada de red y servidor de seguridad de protección para las máquinas virtuales.

- ◆ Sistema de Gestión de Evaluaciones.

Se desarrolló una aplicación donde se pudiera ir guardando un histórico de los elementos esenciales de cada prueba de liberación, de manera que pueda ser consultado en cualquier momento que se necesite para determinar algún dato sobre una prueba de liberación realizada. Además permite mantener a la dirección de Calisoft y del DPSW actualizada del estado de las pruebas, el cual puede ser consultado a través de esta herramienta web. Este Sistema de Gestión de Evaluaciones tiene como objetivo también confeccionar el parte que se emite semanalmente del estado de las pruebas en función de precisar las que están solicitadas, en ejecución, detenidas o abortadas y otros

elementos generales que se consideren importantes. El sistema se está probando en estos momentos y se comenzaron a introducir los datos históricos. Debe comenzarse a utilizar por todos los especialistas del DPSW a partir del próximo curso. Una vista de la aplicación se muestra en el Anexo 7.

- ◆ Herramienta para el control de accesos al LIPS.

Este sistema se concibe para poder registrar la entrada y salida de los estudiantes al turno del LIPS, a través de un lector de código de barra que marca el código del solapín de los estudiantes para de esta manera gestionar la asistencia. Además gestiona los horarios del LIPS, la planificación de las pruebas a realizar, los laboratorios y sus recursos y los propios estudiantes. Está en fase de prueba para comenzar a utilizarla completamente el curso que vienen en el LIPS. Se muestra una imagen de la aplicación en el Anexo 8.

- ◆ Herramientas automatizadas por tipos de prueba.

Se han identificado herramientas a partir de los tipos de prueba que se desarrollan en el LIPS, las cuales se precisan en la Tabla 4, con una pequeña descripción en cada caso. Algunas de estas herramientas ya se han comenzado a utilizar en el desarrollo de las pruebas por especialistas y otros casos están en estudio para definir cuáles pueden insertarse en las clases del LIPS y de qué manera; y cuáles deben ser utilizadas solo por especialistas por su complejidad.

Tipo de prueba	Herramienta	Descripción
Carga y estrés	JMeter	Herramienta de carga y rendimiento. Organización: The Apache Jakarta Project. Apache Jmeter es una aplicación de escritorio 100% java, diseñada para realizar pruebas de carga funcional y medición de rendimiento. Esta fue diseñada originalmente para probar aplicaciones Web, se ha ampliado desde entonces a otras funciones de prueba.
Funcionales	Selenium	Es una herramienta de Software Libre para pruebas de aplicaciones Web. Las pruebas se ejecutan directamente en un navegador y facilitan las pruebas de compatibilidad en navegadores, también como pruebas funcionales de aceptación de aplicaciones Web. Posee un ambiente de desarrollo llamado Selenium IDE, este facilita el registro de pruebas de aceptación y su depuración.
Benchmark (Comparativa)	Meld	Es una herramienta para comparar ficheros y buscar diferencias entre dos versiones distintas del mismo archivo.

Seguridad	Brutus	Herramienta de hacking, es un programa que permite averiguar contraseñas por el método de la fuerza bruta, es decir, probando secuencialmente cada una de las diferentes posibilidades existentes, hasta dar con la correcta. Brutus soporta múltiples protocolos de autenticación de usuarios, como por ejemplo POP3, HTTP, FTP, SMB, etc.
Seguridad	Dirbuster	Herramienta desarrollada en JAVA que permite encontrar aquellos directorios y archivos que se encuentren en los servidores WEB y que no tengan los permisos asignados correctamente, haciendo uso de la técnica fuerza bruta.
Seguridad	SQLmap	Desarrollada en python para realizar inyección de código sql automáticamente. Su objetivo es detectar y aprovechar las vulnerabilidades de inyección SQL en aplicaciones web. Una vez que se detecta una o más inyecciones SQL en el host de destino, el usuario puede elegir entre varias opciones como son: enumerar los usuarios, los hashes de contraseñas, los privilegios, las bases de datos, entre otras.
Seguridad	SQL Brute	Herramienta para extraer datos de las bases de dato mediante ataques de fuerza bruta usando vulnerabilidades de inyección SQL ciega. Soporta exploits basados en tiempo y en error en un servidor Microsoft SQL., y exploits basados en errores para Oracle.
Seguridad	Webcarab	Es una poderosa herramienta que ayuda al usuario en el Penetration Testing (prueba de penetración). Es un framework para el análisis de aplicaciones que se comunican mediante los protocolos HTTP y HTTPS. Está escrito en Java, por lo que es portable a muchas plataformas. Tiene varios modos de funcionamiento, ejecutado por un número de plugins y en su uso más común, funciona como un proxy de intercepción, que permite al usuario revisar y modificar peticiones creadas por el navegador antes de ser enviadas al servidor, revisar y modificar las respuestas del servidor antes de que sean recibidas por el navegador.
Seguridad	Nikto	Es una herramienta de escaneo de servidores web, la cual se encarga de efectuar diferentes tipos de actividades tales como, detección de malas configuraciones y vulnerabilidades en el servidor objetivo, detección de ficheros en instalaciones por defecto, listado de la estructura del servidor, versiones y fechas de actualizaciones de servidores, test de vulnerabilidades XSS, ataques de fuerza bruta por diccionario, reportes en formatos txt, csv, html, etc.
Seguridad	Wapiti	Puede detectar un razonable número de vulnerabilidades, entre ellas, la inyección XSS, que es la vulnerabilidad detectada en un porcentaje muy elevado de casos. Los ataques que realiza son los primeros que un hacker podría realizar contra una página web. Además permite realizar auditorías de la seguridad de aplicaciones web y escanea las páginas web de las aplicaciones en busca de secuencias de comandos y las formas en que puede inyectar datos. Actúa como un fuzzer, inyectando diferentes cadenas para ver si la aplicación es vulnerable.
Seguridad	OpenVas	Una herramienta que permite identificar las vulnerabilidades de un equipo o servidor desde otra PC remota. Precisa cuáles son las posibles vulnerabilidades a las que puede estar expuesto el equipo analizado durante un ataque remoto, la gravedad de esas vulnerabilidades y unas recomendaciones para su solución.

Seguridad	Nmap	Herramienta de escaneo de puertos de un servidor de hosting o de reconocimiento de una red, que es utilizada para localizar los servicios abiertos, los servicios filtrados y los puertos cerrados de un cierto servidor, el sistema operativo de un host y las versiones de los servidores que operan en el host en los diferentes puertos TCP o UDP. Además rastrea los diferentes hosts en una red, y los routers que existen en esta red.
Seguridad	Metaexploit	Es una herramienta para desarrollar y ejecutar exploits contra una máquina remota. Tiene una consola para poder ejecutar herramientas relacionadas con la seguridad informática. Además su fortaleza y popularidad se basa en que resulta "relativamente fácil" codificar exploits, y una vez codificados bajarlos y usarlos con el propio framework.
Volumen	DBMonster	Es una aplicación para generar datos aleatorios para probar aplicaciones de SQL a bases de datos con mucha carga.

Tabla 4: Herramientas automatizadas identificadas por tipos de prueba.

### 2.3 Descripción general del funcionamiento del LIPS.

Después de haber conceptualizado los elementos esenciales que componen el LIPS, se puede describir de forma general cómo funciona este, a partir de la relación de las actividades y subprocessos que se ejecutan de manera continua y armónica (Figura 15).

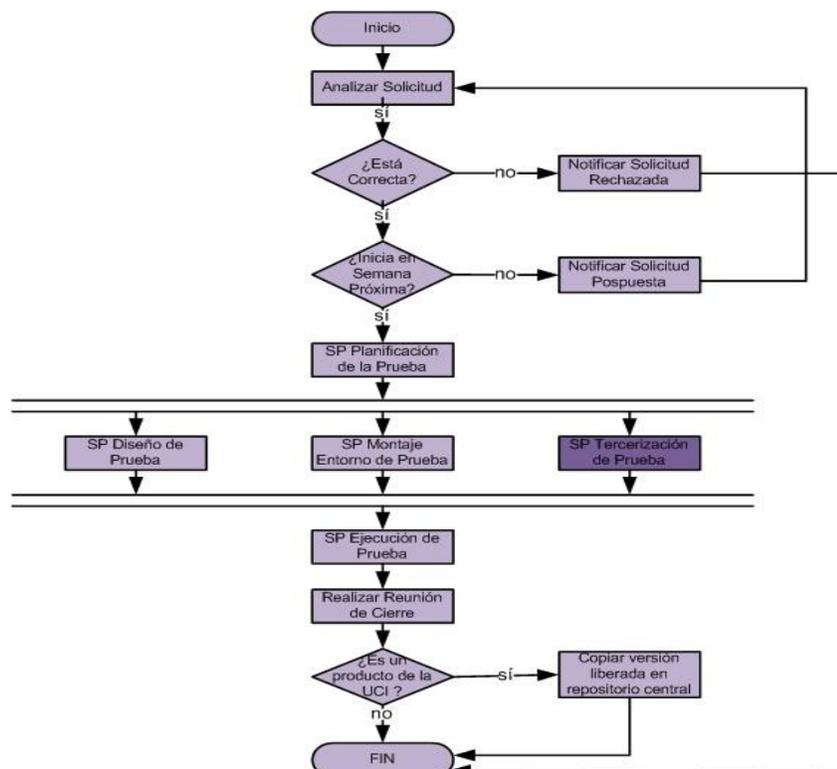


Figura 15: Proceso del LIPS.

El proceso presentado en la figura anterior está descrito con más detalles de forma textual y gráfica en el Libro de Procesos del LIPS, el cual fue elaborado utilizando la plantilla definida por el Grupo de Normalización y Métricas de Calisoft, como parte del Programa de Mejora que se desarrolla en la UCI para lograr la certificación nivel 2 de CMMI, de la cual se han asumidos los epígrafes que aplican para este proceso.

#### **2.4 Conclusiones parciales.**

De lo presentado en el Capítulo se concluye lo siguiente:

1. Se conceptualizó una nueva propuesta de laboratorio de pruebas, materializada en el Laboratorio Industrial de Pruebas de Software.
2. El LIPS funciona similar a un proceso industrial donde se tienen entradas que son transformadas en un ciclo de actividades continuas y se obtienen salidas que agregan valor al cliente.
3. El LIPS tiene una relación directa con el nuevo modelo de integración formación, producción e investigación de la UCI, al formar a los estudiantes de 2do año, con el objetivo de que certifiquen el rol de probador.
4. Las clases del LIPS son Prácticas Laborales, que forman parte de la asignatura Práctica Profesional 2 y se ha definido una estructura para estas prácticas, de forma que todos los profesores tengan una guía y la desarrollen de manera homogénea.
5. Los estudiantes durante su formación en el LIPS interactúan con herramientas para la gestión de la prueba como son el SVN y el Redmine, las cuales utilizarán más adelante en su carrera cuando desempeñen otros roles en los proyectos productivos.
6. Se han introducido como un elemento novedoso en el proceso el uso de herramientas automatizadas, tanto para la gestión de la prueba como para la ejecución de estas.
7. Se ha definido el proceso general del LIPS, el cual se encuentra descrito con más detalles en el Libro de Procesos del Laboratorio Industrial de Pruebas de Software, elaborado a partir de la plantilla definida en el Programa de Mejora de la UCI, donde se precisan todos los elementos que lo conforman.

### **3 Validación de la hipótesis y presentación de resultados**

#### **3.1 Introducción**

En el presente capítulo se realizará un estudio de los resultados alcanzados con la implantación de la concepción descrita del Laboratorio Industrial de Pruebas de Software, como parte esencial del Departamento de Pruebas de Software de Calisoft.

El estudio se basa en resultados obtenidos en las pruebas de liberación a partir de las variables definidas en la hipótesis, haciendo comparaciones de los datos antes de implantada la nueva concepción del LIPS y posterior a su puesta en marcha. También se apoya en datos obtenidos del desarrollo de pruebas de aceptación en Venezuela, así como de la aplicación de una encuesta de satisfacción del cliente diseñada para ser aplicada en las reuniones de cierre de las pruebas de liberación.

#### **3.2 Validación de la propuesta**

Para validar la propuesta se tendrán en cuenta las 3 variables definidas en la hipótesis que son: calidad de las pruebas, optimización de los recursos y tiempo de ejecución de las pruebas. Se precisan en cada caso los elementos que se consideraron para evaluar cada una.

##### **3.2.1 Calidad de las pruebas.**

La calidad es un factor subjetivo, por lo que se han determinado un grupo de aspectos que demuestran un aumento en la calidad de las pruebas de liberación con la implantación del LIPS, los cuales serán presentados a continuación.

##### **3.2.1.1 Comparación entre la cantidad de NC detectadas en las pruebas de liberación y las pruebas de aceptación.**

Uno de los objetivos de las pruebas de liberación (PL) es detectar la mayor cantidad de NC posibles, de manera que el artefacto se entregue al cliente con la menor cantidad de errores. Se ha elaborado una tabla en la que se especifican para una muestra de los proyectos probados en el LIPS, la cantidad de NC que se detectaron por los estudiantes durante las PL

y la cantidad de NC que fueron encontradas en las pruebas de aceptación (PA) por el cliente, tomando en consideración en esta etapa solo los errores que no tienen relación directa con el negocio del proyecto y que por tanto no se hubieran podido detectar en el LIPS, por no tener los conocimientos específicos del negocio. Estos datos se obtuvieron utilizando el método de estudio de casos y se utilizaron proyectos que han sido liberados durante este curso en el LIPS y aceptados posteriormente en Venezuela, haciendo búsquedas en los repositorios y entrevistas a profundidad a los especialistas del DPSW.

En la última columna de la Tabla 5 se haya la razón entre la cantidad de NC detectadas en la PA y la identificada en la PL, siendo satisfactorio el resultado mientras más tienda a cero, pues esto significa que fueron detectadas un número no significativo de NC con respecto a las que pudieron haber llegado al cliente, si no se hubiera realizado una prueba de liberación con calidad. No se consideran en este análisis los Pedidos de Cambio, que como generalidad es lo que más se detecta en las PA, pues estos no son significativos para el LIPS. Se precisan nombres genéricos de los proyectos, por la confidencialidad de la información.

Proyecto	Artefacto	Cantidad NC en PL	Cantidad NC en PA	Cant. NC PA/Cant. NC PL
A	a1	36	0	0
A	a2	107	0	0
A	a3	103	0	0
A	a4	13	0	0
B	b1	49	0	0
C	c1	64	0	0
D	d1	47	2	0.043
F	f1	110	4	0.036
G	g1	20	8	0.4
H	h1	94	5	0.053
H	h2	10	4	0.4
H	h3	675	15	0.022
I	i1	46	0	0

Tabla 5: Comparación de NC detectadas en PL y PA.

### **3.2.1.2 Resultados de la encuesta para evaluar la satisfacción del cliente con el proceso de pruebas de liberación**

Como parte del proceso de pruebas se definió que se realizara una reunión de cierre una vez que el proceso esté listo para concluir, donde se hace un resumen de los principales resultados de la prueba y se realiza una encuesta de satisfacción del cliente a los miembros del proyecto que participaron en la liberación. De esta manera se retroalimenta el proceso de prueba, a partir de los criterios de los clientes. El diseño de la encuesta se presenta en el Anexo 9 y se considera una encuesta preliminar, que se irá modificando para obtener información que permita mejorar el proceso de pruebas de liberación.

Se tomaron 5 proyectos pilotos y la encuesta fue aplicada a 13 miembros de los equipos de proyecto que han participado en el proceso de pruebas, distribuidos en los siguientes roles: 4 Líderes de proyecto, 2 Arquitectos (Arq), 3 Programadores (Prog), 2 Analistas y 2 de Calidad, los cuales pertenecen a 5 proyectos productivos de la UCI. A continuación se hace un análisis de las respuestas dadas según los roles y proyectos:

- ◆ Todos consideran el proceso de pruebas de liberación Importante, lo que demuestra que ha ganado reconocimiento dentro de los proyectos, los cuales han incorporado una cultura de calidad, al menos para determinar que es necesario realizar pruebas de software a sus artefactos antes de ser entregados al cliente.
- ◆ Todos los encuestados se sintieron parte del proceso de prueba como un ente activo. Esto es muy importante, pues al ser los clientes, son esenciales en el proceso y deben sentirse identificados con él.
- ◆ Se elaboró un gráfico (Figura 16) donde se refleja el valor otorgado (Alto, Medio o Bajo) en cuanto al impacto que tuvieron las NC detectadas en el proyecto, desde el punto de vista de los roles, siendo más representativas las de valor Medio (8 personas), seguidas de las clasificadas como Alto (4 personas) y solo 1 Arquitecto consideró que tenían Bajo impacto. En la Figura 17 se precisan los valores otorgados desde el punto de vista de los proyectos, teniendo en cuenta las respuestas dadas por

los miembros de cada uno. De forma general puede considerarse que las NC detectadas en el LIPS a los artefactos de estos proyectos tuvieron un impacto entre Medio y Alto, lo cual es significativo, pues fueron detectadas por los estudiantes en el LIPS.

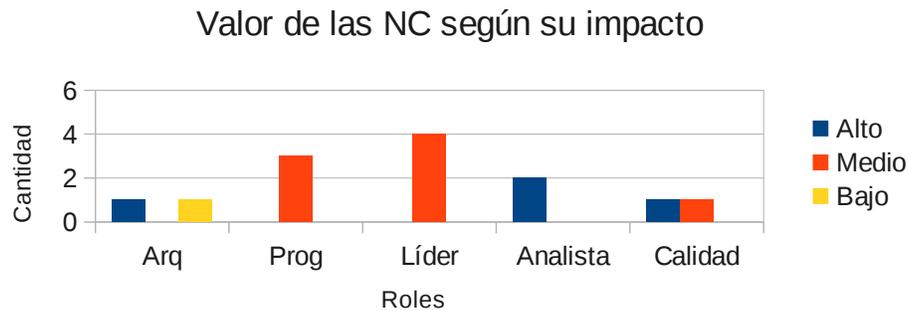


Figura 16: Valor de las NC según su impacto por roles.

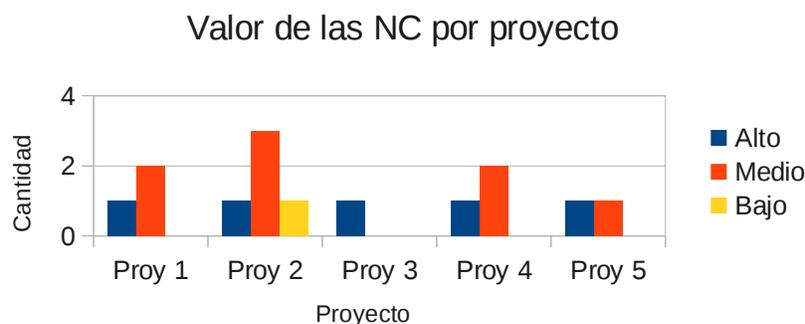


Figura 17: Valor de las NC según su impacto por proyecto.

- ◆ Todos los implicados reconocieron que los Especialistas al frente de las pruebas mostraron conocimientos sobre el proceso en cuestión, factor significativo que demuestra la preparación de los especialistas del DPSW.
- ◆ En cuanto a si se consideran que debe cambiarse algo del proceso de pruebas de liberación definido, se dieron como opciones Todo, Nada o Una parte y en este último caso debían precisar cuál. Las respuestas estuvieron enfocadas a que debía modificarse una parte del proceso, como se muestra en la Figura 18. Todos los

encuestados hicieron referencia al tiempo que transcurre entre que se hace la solicitud de prueba y se realiza la reunión de inicio, el cual nunca pasa de 5 días hábiles. Como parte de las políticas definidas en el proceso de pruebas de liberación, el cual fue aprobado en una reunión de la Red de Centros de Desarrollo de la UCI donde participan todos los Directores de los Centros, se precisa que las solicitudes deben realizarse como mínimo con 1 semana de antelación al inicio de la prueba y que las solicitudes de prueba se revisan por la dirección del DPSW los lunes y jueves de cada semana. Estas políticas respaldan el hecho de que los procesos demoren esa cantidad de días y todo depende del tiempo con que se realice la solicitud, pues como generalidad los proyectos realizan la solicitud el mismo día fijado para el inicio de la prueba. En las reuniones de cierre al intercambiar con los proyectos se precisan estos aspectos, de manera que obtengan mayor dominio del proceso de pruebas de liberación.



Figura 18: Consideraciones sobre qué mejoras debe tener el proceso de pruebas.

- ◆ Todas las personas encuestadas consideraron Satisfactorio el proceso de pruebas, lo que demuestra que a pesar de no concordar en algunos aspectos específicos del proceso como tal, el desarrollo de este fue fructífero y se sintieron satisfechos.
- ◆ La última pregunta se elaboró abierta, de manera que se pudieran obtener criterios que no estuvieran reflejados en las preguntas y que se consideraran importantes por parte de los proyectos, sobre el desarrollo de la prueba. De los encuestados 2 personas no escribieron nada, 2 dieron sugerencias y 9 pusieron criterios positivos

sobre el proceso.

Las sugerencias realizadas fueron que se realizara la encuesta en una aplicación y que se pudiera garantizar la misma calidad en el proceso de prueba de los Centros.

Los criterios emitidos fueron:

- ◆ El proceso se realizó de forma exitosa.
- ◆ Se realizó correctamente el proceso de pruebas de liberación.
- ◆ El proceso permite obtener artefactos liberados con elevado grado de calidad.
- ◆ Se encontraron no conformidades que nunca se habían detectado internamente en el proyecto.
- ◆ Proceso serio.
- ◆ Se cumplieron los tiempos planificados y se monitoreó el cronograma acordado, de manera que se fuera actualizando en función del desarrollo de la prueba.
- ◆ Fue mi primera experiencia en pruebas de liberación y fue buena, pues me ayuda a organizarme para procesos futuros.

### **3.2.1.3 NC detectas en nuevos tipos de prueba**

Se han aplicado, como se ha explicado durante la presente investigación, tipos de prueba que no se realizaban antes en las pruebas de liberación como son: Pruebas de Seguridad, con mayor nivel de profundidad (Tabla 6), Pruebas de Instalación y Configuración (Tabla 7) y Pruebas de Carga y Estrés (Tabla 8). Esto ha sido posible por la concepción del laboratorio y la especialización de los profesionales que trabajan en el DPSW. La tendencia debe ser a continuar profundizando en estos tipos de prueba e incorporar otros, que evalúen con mayor profundidad las características de calidad de la norma ISO/IEC 9126. A continuación se muestran datos relacionados con NC detectadas en estos tipos de prueba, lo que ha aumentado la calidad de estas, al detectar errores que tienen que ver con requisitos no funcionales del software, que antes solo se podían identificar cuando el software se

comenzaba a utilizar por el cliente.

Cuando se detectan NC en las pruebas de seguridad se precisa cuál es la posible solución para resolver el problema, lo que ayuda a los equipos de proyecto cuando van a responder las NC. Estas pruebas se realizan por el Laboratorio de Seguridad del Centro de Desarrollo TLM de la UCI, el cual trabaja de conjunto con el LIPS en el desarrollo de este tipo de prueba, pues al tener personal especializado con experiencia en estas, se decidió tercerizar este servicio, aunque siempre es supervisada la prueba por un especialistas del GIPS que es el responsable de las pruebas de seguridad. Se realizan con herramientas automatizadas a aplicaciones web.

Las pruebas de carga y estrés se realizan a aplicaciones web, utilizando la herramienta Jmeter. Los tipos de errores que más se detectan en estas pruebas están relacionados con el error 302: recursos movidos temporalmente, error 404: no encuentra la url en la petición, timeout: ciclo de carga permanente sin obtener respuesta y que la herramienta capta la contraseña al no estar encriptada desde la PC cliente.

◆ Pruebas de Instalación y Configuración:

Proyecto	Artefacto	Cantidad de NC
A	a1	24
B	b1	20
C	c1	1
D	d1	4
E	e1	25
E	e2	11
F	f1	24
G	g1	7
H	h1	6
I	i1	3

Tabla 6: Cantidad de NC en Pruebas de Instalación y Configuración.

◆ Pruebas de Seguridad:

Proyecto	Artefacto	Cantidad de NC
A	a1	5
B	b1	3
C	c1	4
D	d1	15
E	e1	4
F	f1	3
G	g1	8
H	h1	1

Tabla 7: Cantidad de NC en Pruebas de Seguridad.

◆ Pruebas de Carga y Estrés:

Proyecto	Artefacto	Cantidad de NC
A	a1	27
B	b1	30
C	c1	1
D	d1	8
E	e1	5
E	e2	15
F	f1	9

Tabla 8: Cantidad de NC en Pruebas de Carga y Estrés.

### 3.3 Optimización de los recursos

Según la Real Academia de la Lengua Española optimizar significa mejorar la manera de realizar una actividad (RAE). En este sentido la nueva conceptualización del LIPS permitió optimizar la utilización de los recursos con los que se cuenta, pues se mejoró la forma en la que se gestionan buscando soluciones a los problemas que se identificaron y que afectaban esta variable.

En el caso de la utilización de los estudiantes como probadores se definió en la concepción actual que no hay estudiantes asignados a pruebas específicas, sino que se establecen

horarios de prueba a los que asisten los grupos de clase en función de la planificación de cada semana. Esta planificación se hace semanal, a partir de la definición de 4 semanas tipo que garantizan que todos los grupos pasen por el LIPS una vez al mes. Las facultades envían al DPSW los horarios en los que las brigadas deben asistir al LIPS la semana antes a la que le corresponde y a partir de este, se realiza la planificación de los artefactos a probar, teniendo en cuenta los cronogramas de las pruebas y las prioridades que se establezcan de estas por la dirección de la producción en la UCI. De esta manera se tiene fuerza permanente para las pruebas en los laboratorios, pues de los 22 turnos posibles a cubrir en la semana (4 turnos diarios de lunes a viernes y 2 turnos los sábados), se tienen cubiertos como generalidad alrededor de 19 turnos y los 3 restantes son utilizados para recuperaciones de clases que sean necesarios y por 60 estudiantes de repitencia que realizan su componente laboral en el LIPS ejecutando pruebas de liberación. Estos estudiantes también tienen horarios de trabajo en la noche.

Se hace un mejor uso de las PC al tener los 30 puestos de trabajo en los 2 laboratorios destinados a tiempo completo a la realización de las pruebas, donde se montan diferentes entornos en función de las pruebas que deban realizarse. Al asistir los grupos de clase a los turnos del LIPS todos los puestos de trabajo son cubiertos por los estudiantes.

La utilización de los especialistas también es más óptima en esta nueva concepción, pues al contar con fuerza de trabajo permanente para las pruebas, pueden especializarse en los tipos de prueba y desarrollarlos, hacer pruebas exploratorias y muestreos de las pruebas que se están ejecutando en el LIPS de manera que se verifique la calidad con la que se están ejecutando y poder mejorar los diseños de las herramientas de prueba en función de estos resultados, para lograr que los estudiantes detecten una mayor cantidad de NC, sobretodo de tipo técnicas. Estas actividades que no se podían desarrollar en la organización anterior del laboratorio, pues los especialistas debían enfocarse más en ejecutar pruebas y coordinar los recursos aislados con los que se contaba para su desarrollo.

### 3.4 Tiempo de ejecución de la prueba

El tiempo total de la prueba tiene en consideración el tiempo que es destinado a cada una de las actividades definidas en el proceso de pruebas de liberación, algunas de los cuales son responsabilidad del LIPS y otras del proyecto. Para demostrar cómo ha disminuido el tiempo en la ejecución de las pruebas a partir de la nueva concepción del LIPS, se precisan datos que están relacionados con el tiempo que se demora una iteración de prueba, el cual depende completamente del trabajo en el laboratorio. Con el objetivo de obtener datos que permitieran demostrar esta variable se utilizaron métodos como el estudio de casos y la entrevista a profundidad a especialistas del LIPS.

Se seleccionaron artefactos que tuvieran una magnitud (cantidad de páginas o caso de prueba) y complejidad similar, de manera que tengan validez los resultados. Se presenta la Tabla 9 donde se precisa la cantidad de páginas o casos de prueba (CP) de los artefactos y el tiempo dedicado a una iteración de prueba antes y después de la concepción del LIPS.

Artefacto	Antes del LIPS		Artefacto	Después del LIPS		Cant. días menos
	Cant. Págs - CP	Tiempo (días)		Cant. Págs o CP	Tiempo (días)	
A	487 págs	10	B	485 págs	2	8
C	1000 págs	12	D	1620 págs	3	9
E	45 CP	5	F	45 CP	2	3
G	150 CP	9	H	154 CP	4	5
I	12 CP	3	J	11 CP	1	2
K	193 CP	24	L	195 CP	5	19

Tabla 9: Tiempo dedicado a una iteración de pruebas.

Como se puede apreciar la cantidad de días que disminuye el tiempo de pruebas es significativo, dado porque se cuenta con estudiantes de forma permanente para las pruebas, no asignados a una prueba en específico, e influenciado también por la automatización de la gestión de las NC. Antes de introducir este elemento en el laboratorio, se dedicaba mucho tiempo de la iteración a conformar los informes de NC en un documento word y el fin de la iteración lo marca la entrega de todas las NC detectadas en esta. En el proceso actual las

NC se van asignando en la medida que el especialista las revisa, lo cual puede hacerse al mismo tiempo que se están detectando, por contar con una herramienta web como el Redmine.

### **3.5 Conclusiones**

Del presente capítulo se concluye lo siguiente:

1. Se implantó la nueva concepción del LIPS.
2. Con la implantación del LIPS aumentó la calidad de las pruebas, demostrada a través varios factores que intervienen en determinar esta.
3. En las pruebas de aceptación con el cliente se detectan un número no significativo de NC con relación a las que son encontradas en el proceso de pruebas de liberación.
4. Se han detectado un gran número de NC relacionadas con los nuevos tipos de prueba que se comenzaron a aplicar en el LIPS, lo que garantiza mayor calidad en los artefactos liberados a partir de medir características de calidad determinadas por una norma internacional.
5. Se comenzó a aplicar una encuesta de satisfacción del cliente que demostró un buen desenvolvimiento de los procesos de prueba.
6. En el LIPS se realiza un uso más óptimo de los recursos humanos y tecnológicos, dados por las mejoras en actividades que se realizan con la nueva concepción.
7. Se ha disminuido de manera considerable el tiempo de ejecución de una iteración de pruebas, lo que hace el proceso más ágil, sin disminuir en la calidad.

#### **4 Conclusiones**

1. Se conceptualizó el Laboratorio Industrial de Pruebas de Software a partir de introducir elementos fundamentales y novedosos para el proceso de pruebas de liberación como son el funcionamiento como fábrica (proceso industrial), la vinculación de todos los estudiantes de 2do año, la relación directa con el nuevo modelo de integración formación-producción-investigación de la UCI y automatización de las pruebas.
2. Los estudiantes de 2do año se vinculan al LIPS a través de la asignatura Práctica Profesional 2 con el objetivo de certificar el rol de probador.
3. La creación del Grupo de Ingeniería de Pruebas ha permitido mayor especialización de los profesionales del Departamento, lo que se materializa en la aplicación de nuevos tipos de prueba, asociados a las características de calidad definidas por la norma internacional ISO/IEC 9126.
4. Se cumplió con el objetivo planteado al implantar el LIPS en la Universidad de las Ciencias Informáticas.
5. Se demostró la hipótesis de que con la implantación de un Laboratorio Industrial de Pruebas de Software en una universidad productiva se aumentó la calidad de las pruebas ejecutadas, se optimizó el uso de los recursos y se disminuyó el tiempo de ejecución de la prueba.

## 5 Recomendaciones

El proceso de pruebas de liberación que se ha presentado es una renovación del anterior, teniendo en cuenta los elementos de este que no estaban acordes a las condiciones que impuso el entorno en el que desarrolla Calisoft y a la concepción del LIPS. De igual manera la propuesta actual ya implantada y con resultados, debe evolucionar para que perdure, adaptándose a los nuevos tiempos.

Teniendo en cuenta estos elementos y la necesidad de hacer más productivo el LIPS se propone:

- ◆ Elaborar un Cuadro de Mando Integral (CMI), de manera que se combine el control operativo a corto plazo, con la visión y la estrategia a largo plazo de la organización.
- ◆ Definición de una quinta perspectiva dentro del Cuadro de Mando Integral, aplicado a un laboratorio de pruebas de software.
- ◆ Integrar de las perspectivas del Cuadro de Mando Integral, las disciplinas de las organizaciones inteligentes y las bases de organizaciones celulares, para obtener un laboratorio inteligente de pruebas de software.
- ◆ Lograr mayor incorporación el conocimiento en la toma de decisiones del laboratorio de pruebas.
- ◆ Analizar herramientas automatizadas que soporten el funcionamiento del laboratorio inteligente de pruebas de software.

## 6 Referencias bibliográficas

- Baeza, Yates. Computing in Chile: ¿The Jaguar of the Pacific Rim? *Communications of the ACM* 38, 1995.
- Bedini, Alejandro. Extracto del libro en formato digital “calidad tradicional y de software. 2007.
- Bedini, Alejandro G. Calidad de software. *NOVATICA* 1995.
- Bizcaia. La calidad del software y los procesos de testing. 2010. Available from world wide web: <[www.euskadinnova.net/documentos/363.aspx](http://www.euskadinnova.net/documentos/363.aspx)>.
- Boehm, B. Verifying and Validating Software Requirements and Design Specifications. *IEEE Software* 1, 1984, 75-88.
- Boehm, B, J. R Brown, and M Lipow. Quantitative Evaluation of Software Quality. *Proceed 2nd Int'l Conference on Software Engineering*. 1978, 592-605.
- BugHuntress QA Lab. Outsourcing Software Testing Services. Company profile. 2011. Available from world wide web: <<http://www.bughuntress.com/software-testing-services/independent-verification-validation.html>>.
- Butler, Tresa, Turner Faith, Standley Verla, and Sullivan Elaine. Software Configuration Management: A discipline with added value. *Crosstalk Magazine* 7, July 2011.
- CES. Cámara Uruguaya de Tecnologías de la Información y Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República de Uruguay (UdelaR), a través de la Fundación Julio Ricaldoni. Available from world wide web: <<http://www.ces.com.uy>>.
- CHAOS. THE STANDING GROUP: “New Chaos number show. StartLing results.” 2009. Available from world wide web: <[http://www1.standishgroup.com/newsroom/chaos\\_2009.php](http://www1.standishgroup.com/newsroom/chaos_2009.php)>.
- CHAOS. THE STANDING GROUP: “New Chaos number show. StartLing results.” 2011. Available from world wide web: <<http://www.marketwire.com/press-release/new-standish-group-report-shows-more-projects-are-successful-less-projects-failing-1405513.htm>>.
- CMMI. Capability Maturity Model Integration (CMMI). 2002.
- CONCYTEG. Concyteg. Laboratorio de desarrollo y pruebas de software en el estado. Available from world wide web: <<http://www.concyteg.gob.mx>>.
- Cairó, Gema. La expansión de la industria del software en India. *BOLETIN ECONOMICO DE ICE* 2000. Available from world wide web: <<http://biblioteca.hegoa.ehu.es>>.
- Camou, Elena. Software Development: The Venezuelan competitive project. *cavecom-e* 2002, 7-14.
- Carrasco, O. M. F, D. G León, and A. B Benavides. Un enfoque actual sobre la calidad del software. *Revista ACI* 3, 2004.
- Carretero, Rosa María. Proyecto de Fin de Carrera. “eAula: desarrollando un prototipo desde una perspectiva de usabilidad infantil”. September 2007.
- Choucair. Choucair Testing S.A. Available from world wide web: <<http://www.choucairtesting.com>>.

- Cusmai, Marcelo. Calidad de Software desde Latinoamérica. *Infotesting. Un espacio para testers profesionales. Revista digital*. Abril 2011. Available from world wide web: <<http://www.infotesting.com>>.
- DCS. Diagnóstico 2008. 2008.
- Daich, G, G Price, B Raglund, and M Dawood. Software Test Technologies Report. 1994.
- Delgado, Ramsés. Estrategia para la estandarización de la documentación y las actividades para el desarrollo de software en la Universidad de las Ciencias Informáticas. November 2010.
- Febles, Ailyn. Case Corporativo para el proceso de control de cambios. 2000.
- Febles, Ailyn, S Álvarez, and H Fernández. Aplicación del modelo CMM a la empresa Segurmática. 2000.
- Febles, Ailyn et al. Una experiencia novedosa para el testing desarrollada por un Departamento de Pruebas de Software. *V Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática 2011* February 2011.
- Finkelievich, Susana. Las TIC en la cooperación Sur-Sur: el acuerdo de librecomercio entre la India y el Mercosur. *Revista CTS 1*, September 2004, 223-233.
- García Mireles, G, and J Rodríguez Jacobo. Aplicación del modelado de procesos en un curso de ingeniería de software. *Revista Electrónica de Investigación Educativa 3*, 2001. Available from world wide web: <<http://redie.uabc.mx/vol3no2/contenido-mireles.html>>.
- García, A. de Amescua, M Velasco, and A Sanz. Ten Factors that Impede Improvement of Verification and Validation Processes in Software Intensive Organizations. *EuroSPI 2007, Postdam, Alemania* September 2007.
- Garvin, D. A. "What Does "Product Quality Really Mean?" *Sloan Management Review 26*, 1984, 25-43.
- González, Santiago. La calidad del Software en modo Factoría de Pruebas. November 2010. Available from world wide web: <<http://www.mkm-pi.com/byte-ti/la-calidad-del-software-en-modo-factoria-de-pruebas/>>.
- González Jorrín, Michael. Proceso de pruebas para la liberación de productos de software. Diciembre 2007.
- HEXAWARE. Hexaware Technologies. Tes Center of Excellent. Available from world wide web: <<http://www.hexaware.com/>>.
- Humphrey, W. Introduction to the Personal Software Process. 1997.
- IEEE, 32. Std 610. IEEE Standard Glossary of Software Engineering Terminology. 1990.
- INTECO. Instituto Nacional de Tecnologías de la Información. Laboratorio Nacional de Calidad de Software, INTECO. Available from world wide web: <<http://www.inteco.es>>.
- ISO. ISO / FDIS 9000: 2005. 2005.
- ISO/IEC. ISO/IEC 9126-1.2. Parte 1: Modelo de Calidad. 2001.
- Kan, Stephen H. Metrics and Models in Software Quality Engineering. 2000.

- Koomen, T, L van der Aalst, B Broekman, and M Vroon. TMap Next for result-driven testing. *UTN Publishers. Netherlands*. 2006.
- Krueger, Richard A. *El grupo de discusión: guía práctica para la investigación aplicada*. [Madrid]: Pirámide, 1991.
- Lavandero, J, E Verdecia, O. L Martínez, and A Díaz. Bases para un nuevo modelo de deformación basado en la integración de procesos de formación producción en la UCI. 2008. Available from world wide web: <[http://evapostgrado.uci.cu/file.php/128/Materiales\\_Tema\\_1/Libro\\_Blanco\\_Modelo\\_IFPI](http://evapostgrado.uci.cu/file.php/128/Materiales_Tema_1/Libro_Blanco_Modelo_IFPI)>.
- Lineamientos PCC. Lineamientos de la Política Económica y Social del Partido y la Revolución. de abril 2011.
- MES. Reglamento Docente y Metodológico. Resolución.
- Martínez Pichardo, Rita, Roberto Verrier, María E. Benítez, María Elena Aguilera, and Nieves Garriga. Algunas actividades presenciales y metodológicas del profesor universitario en Cuba.
- Martínez, Enrique. El Uruguay y la exportación de Software. 2009.
- McCall, J. A, and P. K Richards. Factors in Software Quality. *RADC TR-77-369, US Rome Air Development Center Reports NTIS AD/A-049* 1977.
- Myers, G. The art of software testing, 2da edición. *John Wiley & Sons Inc*. 2004.
- NIST. Planning Report 02-3 The Economic Impacts of Inadequate Infrastructure for Software Testing. 2002.
- Pérez Lamancha, B. Proceso de Testing Funcional Independiente. 2006.
- ParqueSoft. Red de Parques Tecnológicos de Software, ParqueSoft Nation. Available from world wide web: <<http://www.parquesoft.com/>>.
- Piattini, M, F García, and I Caballero. Calidad de Sistemas Informáticos. 2007.
- Piza Escalante, Rodolfo. Comentario a la primera ponencia: La Industrialización. *XIV Asamblea Académica, Revista Acta Académica, Universidad Autónoma de Centro América*. November 1998, 254-255.
- Pressman, Roger. *Ingeniería de Software. Un enfoque práctico*. 4ta edición 1998.
- Pressman, Roger. *Ingeniería de Software. Un enfoque práctico*. 5ta edición 2002.
- RAE. Diccionario de la Lengua Española- Vigésima segunda edición. Available from world wide web: <[http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=calidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=calidad)>.
- Reo, Davis A. La gallina, el cerdo y el modelo CMM. September 2002. Available from world wide web: <<http://www.americaxxi.cl/modules.htm>>.
- Riascos, Sandra Critina. Modelo para la evaluación de la efectividad de la tecnología informática en el entorno empresarial. *Ingeniería e Investigación* 28, 2008. Available from world wide web: <[http://www.scielo.unal.edu.co/scielo.php?pid=S0120-56092008000200019&script=sci\\_arttext](http://www.scielo.unal.edu.co/scielo.php?pid=S0120-56092008000200019&script=sci_arttext)>.
- Rodríguez, Moisés. Calidad de Procesos y Productos de Software. Calidad de productos de software-

- ISO/IEC 25000. July 2010. Available from world wide web: <alarcos.inf-cr.uclm.es/per/fruiz/.../santander/mrodriguez-iso25000-update.pdf>.
- Runeson, P, C Andersson, and T Thelin. What Do We Know about Defect Detection Methods? *IEEE Software* 23, 2006, 82-90.
- SÍNTESIS. Revista Síntesis. 2005. Available from world wide web: <58. <http://www.revistasintesis.cl/>>.
- Sánchez, Sandra, and Javier Garzás. Una revisión y comparativa de los modelos y procesos de prueba. 2010.
- SIME. Lineamientos estratégicos para la informatización de la sociedad cubana. 1997.
- SQRL. Software Quality Research Laboratory. Computer Science and Information Systems Dept. University of Limerick. Available from world wide web: <<http://www.sqrl.ul.ie>>.
- SWEBOK. Guide to the Software Engineering Body of Knowledge SWEBOK. 2004. Available from world wide web: <<http://www.swebok.org>>.
- Santos, vismar. LA INDUSTRIA DEL SOFTWARE. ESTUDIO A NIVEL GLOBAL Y AMÉRICA LATINA. *Observatorio de la Economía Latinoamericana* 2009.
- Scalone, Fernanda. Estudio comparativo de los modelos y estándares de calidad del software. June 2006.
- Sogeti. Grupo CapGemini. Software Control & Testing, Sogeti. Available from world wide web: <<http://www.es.sogeti.com/>>.
- UCI. Planeación estratégica. 2008.
- Verdecia, E, R Portuondo, J Lavandero, and J Arencibia. Proposal of a methodology for carrying out the processes of roles certification during the training of an engineer on informatics sciences. *International Technology, Education and Development Conference. Valencia, Spain.* 2011.
- e-Quality. e- Quality. Available from world wide web: <<http://www.e-quallity.net>>.