

**UCI**

**Universidad de las Ciencias Informáticas  
Facultad 2**

*Sistema Integral de Gestión de los Laboratorios*

**SIGLA**

*Módulo: "Control de Solicitudes de Software"*

*Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias  
Informáticas*

***Autores:*** Iosmara Suárez Roque.

Yadil Mujica Hernández.

***Tutor:*** Ing. Salvador González Gómez.

Ciudad de La Habana, Julio del 2008

"Año 50 de la Revolución"

005.1  
Sua  
5

Fe: 02-03-09

NR: TD-1867-08




*"No se trata de sueños ni fantasías; comienzan a ser visibles realidades.  
¡Bien vale la pena luchar por ellas!"*

*Giddens*

## **DECLARACIÓN DE AUTORÍA**

*Declaramos ser autores del presente Trabajo de Diploma, de esta forma reconocemos y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo como considere necesario.*

*Para que así conste, firmamos la presente a los \_\_\_\_ días del mes \_\_\_\_\_ de 2008.*



---

*Firma de los autores*  
*losmara Suárez Roque*



---

*Firma de los autores*  
*Angel Yadil Mujica Hemández*



---

*Firma del tutor*  
*Ing. Salvador González Gómez*

## **OPINIÓN DE USUARIO DEL TRABAJO DE DIPLOMA**

El Trabajo de Diploma, titulado Sistema Integral de Gestión de los Laboratorios, Módulo; "Control de Solicitudes de Software" fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente \_\_\_\_
- Parcialmente en un \_\_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

---

---

---

---

---

---

---

---

---

---

---

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de julio del 2008.

---

*Representante de la entidad*

---

*Cargo*

---

*Firma*

---

*Cuño*

## **OPINIÓN DE TUTOR DEL TRABAJO DE DIPLOMA**

Título: Módulo: Control de Solicitudes de Software.

Autores: losmara Suárez Roque

Angel Yadil Mujica Hernández

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan:

---

---

---

---

---

---

---

---

---

---

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_\_.

\_\_\_\_\_

\_\_\_\_\_ de julio del 2008.

*Ing. Salvador González Gómez*

# *Agradecimientos*

*De Mara:*

*El desarrollo de este trabajo me ha formado y perfeccionado como persona, estudiante y profesional; ya que se han ampliado mis horizontes al modo de ver y enfrentar los problemas y estos resultados se los debo:*

*A los profesores que me han asistido y aconsejado cuando los he necesitado en especial a Yadilka, que si no fuese por su apoyo, insistencia y hasta incluso sus lágrimas hoy no estaría aquí culminado mis estudios universitarios. A Michel, que fue uno de los profesores que nos dedicó el tiempo necesario para aclarar las dudas con respecto a la tesis.*

*A mi tutor, que se ha comportado incondicionalmente con nosotros, representándonos y apoyándonos e todo momento.*

*A Baby, Daily, Rafe, abuela Blanca; que aunque se encuentren lejos, en estos 5 años los he sentido muy cerca, dándome su cariño y fuerzas para seguir adelante.*

*A mis hermanos, Kire, Sopa y Kenia; que me han brindado sus consejos en cada momento difícil por los que he pasado.*

*A mis lindos sobrinos, que aunque son muy pequeños aún para entender la vida los quiero como si fuesen mis propios hijos.*

*A mis amigos: Aymee, Ismary, Castilo, Yaily, Yunier, Orley, Yuliet, Yunay, que me han ofrecido siempre voluntaria y desinteresadamente una mano cada vez que los necesito y a mis compañeros de estudio, que han sido mi familia universitaria en estos años que llevamos juntos.*

*A la Revolución y al Comandante en Jefe por ofrecerme tanta riqueza de conocimientos e información.*

*A mi amor Geldri Berroa, quien es la persona que me hace sentir cerca de casa, quien ha soportado mis malcriadeces y en ocasiones asume el papel de compañero, hermano y amigo.*

*Y en especial a Saturnino y Ángela Rosa mis queridos, especiales, adorables, serviciales e incondicionales padres. Los amo mucho mucho mucho y les doy gracias por guiarme y formarme en lo que soy hoy.*

*De Yadil:*

*Agradecimientos a toda mi familia por el apoyo que siempre me han brindado sin importar nada ni esperar nada a cambio.*

*Agradecimientos especiales a mi mamá por ser la persona mas importante en este mundo y por darme tanto amor durante toda mi vida, hoy quisiera decirte que te quiero mucho y te agradezco por la forma en que me enseñaste a enfrentar la vida.*

*Agradecimientos a mi hermana, por ser mi otra mamá, por estar siempre ahí para tu hermano menor, tengo que confesarte que sin ti me sentiría un poco solitario en este mundo, gracias por demostrarme tanto amor.*

*Agradecimientos a todos mis amigos (Mario, Osmany, Annioldys, Orlando, Erik, Alexey, Luis Alberto, Ernesto, Serguey, Lachy, Sergio) y todos los que puedan haber pasado por alto pero que saben que tienen en mi a un amigo.*

*Agradecimientos a mis compañeros de proyecto (Castillo, Orley, Yaily, Ismary) por la ayuda brindada y en especial a Yunier (Piri) de todo corazón te agradezco tu ayuda y te pido disculpas si llegué a molestarte.*

*Agradecimientos a mi tutor.*

*Agradecimientos a Yoslane por ser mi mejor amiga, por estar ahí cuando he necesitado su ayuda, por ser una persona especial.*

*Agradecimientos a mi prima Anaibis por ser mi psicóloga particular y guarda mis secretos.*

*Agradecimientos a Deysi por toda la ayudada que me ha dado.*

*Agradecimientos a la Revolución por darme la posibilidad de formarme como profesional*

*Agradecimientos a toda la gente de mi Guild.*

*Agradecimientos a todas aquellas personas que durante toda mi vida me han ayudado de una u otra forma.*

*Hoy quisiera agradecer incluso a Dios.*



## *Dedicatoria*

*Dedico este trabajo a las personas más especiales e importantes en mi vida: a mis padres, que me han apoyado, comprendido, educado y formado. Quienes han jugado el papel del árbol protector, que cuando niños nos alimentan con sus frutos y regocijan con su sombra, una vez crecidos nos brindan sus troncos para que formemos nuestro hogar y cuando estamos agotados y agobiados de los problemas que nos impone la vida, siguen ahí para nosotros, y aunque ya sean ancianitos les queda aún su tronco para que reposemos y descarguemos todos nuestros problemas.*

*También se lo dedico a mis amigos y hermanos, quienes más que hermanos han sido las personas que me han enseñado a enfrentar la vida y no verla tan complicada como lo parece, a mis lindos sobrinos, a toda mi familia y a mi amor, que me ha ayudado a levantarme tras cada caída, me ha comprendido, amado, consentido y guiado hasta el final de este difícil camino.*

*Iosmara Suárez Roque*

*Dedico este trabajo a mis padres por haberme traído al mundo, por la educación que supieron darme, en especial a mi mamá por vivir para mí desde que nací, por inculcarme todos mis principios, por apoyarme en todo, a mi hermana por llevarme la contraria cuando cree que estoy actuando mal, a todas esas personas que saben que son muy especiales en mi vida y que nunca podrán ser remplazadas, espero nunca llegara defraudarlos.*

*Angel Yadil Mujica Hernández*

## RESUMEN

El proceso de solicitud de instalación de programas en los laboratorios docentes es imprescindible en el cumplimiento del plan de estudio de la Universidad de las Ciencias Informáticas. En la actualidad para realizar el pedido de instalación se llevan a cabo una serie de trámites que en muchas ocasiones se tornan engorrosos, además, la recopilación de información se realiza de forma manual, tal es el caso que en ocasiones no se disponen de programas que son indispensables para impartir clases prácticas de ciertas asignaturas.

El presente trabajo tiene como objetivo desarrollar el módulo: "Control de Solicitudes de Software" (perteneciente al proyecto SIGLA) para la gestión de los pedidos de instalación de programas en los laboratorios docentes, permitiendo obtener con su elaboración un control adecuado de los solicitudes, anulación de los atrasos en el proceso docente educativo y aprovechamiento de la jornada laboral por parte del personal encargado de las actividades de petición e instalación de programas.

Para modelar, especificar, construir y documentar la aplicación se tuvo en cuenta que los lenguajes, herramientas y tecnologías utilizadas no incurrieran en gastos de licencias, utilizando por ello en su construcción el Visual Paradigm; el lenguaje de programación PHP, la librería ExtJS, con la que se construyó la interfaz de usuario y el sistema gestor de base de datos PostgreSQL.

Palabras Claves: peticiones, programas, instalación, modelar, construir, laboratorios docentes.

# INDICE

INTRODUCCIÓN .....	1
<b>CAPÍTULO 1: MARCO TEÓRICO DE LA INVESTIGACIÓN.....</b>	<b>4</b>
1.1. INTRODUCCIÓN .....	4
1.2. ESTUDIO DE SISTEMAS EXISTENTES VINCULADOS A LA SOLICITUD DE SERVICIOS.....	4
1.3. TECNOLOGÍAS, LENGUAJES DE PROGRAMACIÓN Y LIBRERÍAS.....	5
1.3.1. <i>Lenguajes de Programación</i> .....	5
1.3.1.1. <i>PHP</i> .....	5
1.3.2. <i>Librería ExJS 2.0</i> .....	6
1.3.3. <i>.AJAX</i> .....	6
1.3.4. <i>Servidor Web</i> .....	10
1.3.5. <i>Sistema Gestor de Base de Datos</i> .....	10
1.4. ARQUITECTURA.....	11
1.4.1. <i>Patrón Modelo – Vista – Controlador</i> .....	11
1.4.2. <i>Arquitectura Cliente/Servidor</i> .....	11
1.5. MODELACIÓN DE FUNCIONES MEDIANTE IDEF0.....	12
1.6. LENGUAJE DE MODELADO Y METODOLOGÍA DE DESARROLLO.....	13
1.6.1. <i>Lenguaje de Modelado UML</i> .....	13
1.6.2. <i>Metodología de Desarrollo RUP</i> .....	14
1.7. HERRAMIENTAS UTILIZADAS.....	15
1.7.1. <i>Visual Paradigm</i> .....	15
1.7.2. <i>Aptana</i> .....	16
1.7.3. <i>GIMP</i> .....	16
1.7.4. <i>PGAdmin III</i> .....	17
1.9 PALABRAS CLAVES PARA LA COMPRESIÓN DEL PROBLEMA EXISTENTE.....	17
1.10 CONCLUSIONES.....	19
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>20</b>
2.1 INTRODUCCIÓN .....	20

2.2	OBJETO DE ESTUDIO.....	20
2.2.1	<i>Problema y Situación Problémica.....</i>	20
2.2.2	<i>Objeto de Automatización.....</i>	21
2.2.3	<i>Información que se Maneja.....</i>	22
2.3	PROPUESTA DEL SISTEMA.....	23
2.4	MODELO DE NEGOCIO.....	23
2.4.1	<i>Descripción y Representación de los Procesos del Negocio.....</i>	24
2.5	ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.....	30
2.5.1	<i>Requerimientos Funcionales.....</i>	30
2.5.2	<i>Requerimientos No Funcionales.....</i>	31
2.6	MODELO DE CASOS DE USO DEL SISTEMA.....	34
2.6.1	<i>Definición de los actores del sistema.....</i>	34
2.1.1	<i>Diagrama de casos de uso del sistema.....</i>	35
2.1.2	<i>Descrpción de los Casos de Uso del Sistema.....</i>	36
2.2	CONCLUSIONES.....	36
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO.....</b>		<b>37</b>
3.1	INTRODUCCIÓN.....	37
3.2	ANÁLISIS.....	37
3.2.1	<i>Modelo de Clases del Análisis.....</i>	38
3.3	DISEÑO.....	42
3.3.1	<i>Diagrama de Clases del Diseño.....</i>	42
3.4	DISEÑO DE LA BASE DE DATOS.....	53
3.4.1	<i>Modelo Lógico de Datos.....</i>	53
3.4.2	<i>Modelo Físico de Datos.....</i>	54
3.5	CONCLUSIONES.....	55
<b>CAPÍTULO 4: IMPLEMENTACIÓN.....</b>		<b>56</b>
4.1	INTRODUCCIÓN.....	56
4.2	DIAGRAMA DE DESPLIEGUE.....	56

4.3	DIAGRAMA DE COMPONENTES .....	56
4.4	CONCLUSIONES .....	57
<b>CAPÍTULO 5: ESTUDIO DE LA FACTIBILIDAD. ....</b>		<b>58</b>
5.1-	INTRODUCCIÓN. ....	58
5.2-	PLANIFICACIÓN. ....	58
	5.2.1 <i>Puntos de Casos de Uso</i> .....	58
	5.2.1.1 <i>Cálculo de Puntos de Casos de Uso sin ajustar</i> .....	58
	5.2.1.2 <i>Cálculo de Puntos de Casos de Uso Ajustados</i> .....	60
	5.2.1.3 <i>Estimación de esfuerzo a través de los puntos de casos de uso</i> .....	63
	5.2.1.4 <i>Calcular esfuerzo del proyecto</i> .....	64
5.5	BENEFICIOS TANGIBLES E INTANGIBLES. ....	65
5.6	ANÁLISIS DE COSTO. ....	66
5.7	CONCLUSIONES. ....	66
<b>CONCLUSIONES. ....</b>		<b>67</b>
<b>RECOMENDACIONES. ....</b>		<b>68</b>
<b>BIBLIOGRAFÍA.....</b>		<b>69</b>
<b>ANEXOS .....</b>		<b>72</b>
	ANEXO. NO 1: DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA. ....	72
<b>GLOSARIO DE TÉRMINOS .....</b>		<b>102</b>

## INTRODUCCIÓN

La Universidad de las Ciencias Informáticas(UCI) está compuesta actualmente por 10 205 estudiantes, los cuales se agrupan en 10 facultades, todos se gradúan como Ingenieros en Ciencias Informáticas; pero en dependencia de la facultad a la que pertenecen y a los proyectos productivos a los que estén vinculados reciben asignaturas adicionales.

La mayoría de las disciplinas que se imparten en la universidad requieren de prácticas en los laboratorios, y para que estas tengan los resultados esperados se necesitan instalar determinados programas. Estos se instalan mediante una imagen a principio de cada semestre, según lo necesitado por cada facultad; aunque puede presentarse la situación de que en el transcurso del semestre algún profesor necesite un programa que no esté instalado para impartir un curso optativo.

¿Cómo se realiza el proceso de petición de programas en la universidad?

Los Departamentos Centrales hacen llegar un listado con los posibles programas a instalar a los profesores de los departamentos de cada facultad, y estos solicitan al intermediario los programas que necesitan.

El papel del intermediario es representado por un profesor encargado de realizar todos los trámites para que los programas sean instalados, estas diligencias son: pedir la autorización de las instalaciones al vicedecano(a) de formación y solicitar la instalación de los programas con el permiso obtenido al técnico de software que atiende su docente.

Estos trámites se realizan para solicitar la instalación de una imagen; pero si se requiere un programa que no haya sido integrado en la imagen para impartir un curso optativo por una o varias semanas del semestre, el profesor interesado le da a conocer el programa que necesita al intermediario y este realiza los trámites (mencionados en el párrafo anterior) concernientes a las solicitudes de instalación.

En ocasiones estos pasos se violan, dándose el caso que es el especialista general de software quien acude a los departamentos centrales para pedir la propuesta de los programas, o es el propio educador cuando necesita un determinado programa quien hace la petición directamente al especialista general de software sin una previa autorización.

Estos incumplimientos se deben a que el proceso se realiza de forma manual, y en su desarrollo están implicadas muchas personas, las cuales en ocasiones no se ponen de acuerdo y trabajan sin una previa organización; provocando que el nuevo programa solicitado por el profesor que va a impartir el

curso optativo no se encuentre disponible para la fecha prevista y errores en los programas instalados mediante la imagen desarrollada; ya que al personal que la confecciona no le llega toda la información requerida, o puede que lleguen datos equívocos. Todas estas fallas en las peticiones de programas ocasionan a su vez, pérdida innecesaria del tiempo de la jornada laboral por parte de las personas involucradas en las actividades de petición e instalación y atrasos en el proceso docente-educativo.

La Universidad de las Ciencias Informáticas tiene la particularidad de ser una universidad docente-productiva, pues, además de formar a especialistas de primer nivel en ciencias informáticas, posee la filosofía que prioriza la creatividad, busca soluciones útiles y deshace problemas. Debido a esto, existen en la misma más de 160 proyectos productivos que proporcionan la solución de diversas dificultades de ámbito internacional, nacional y las existentes en la universidad; ejemplo de ello es el proyecto SIGLA(Sistema Integral de Gestión en los Laboratorios), el cual está compuesto por 9 módulos(Administración/Configuración, Control de Personal, Control de Hardware, Control de Medios, Control de Solicitudes de Software, Gestión de Informes y Reportes, Sistema de Reservación, Gestión de Base de Datos y Documentación), encargados de resolver todos los inconvenientes relacionados con el funcionamiento de los laboratorios.

El curso anterior se elaboró el sistema: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC, el cual no arrojó los resultados esperados, por lo que, el módulo Control de Solicitudes de Software tiene la tarea de informatizar las peticiones de programas que se realicen en cada facultad del centro, y con su cumplimiento se espera ofrecer las siguientes mejoras:

- 1- . Un control adecuado de las peticiones de programas.
- 2- Disminución de errores en las imágenes desarrolladas
- 3- Eficiencia y calidad en la instalación de programas.
- 4- Aprovechamiento fructífero de la jornada laboral por parte del personal involucrado en las actividades de solicitudes e instalación de programas en los laboratorios docentes.
- 5- Anulación de los atrasos en el proceso docente-educativo.

Para darle solución a la problemática antes planteada se enfocarán los esfuerzos en resolver el problema: ¿Cómo desarrollar un sistema para el control de solicitudes de instalación de programas en los laboratorios de la UCI?

Además se realizará un estudio de los procesos de control y organización de los laboratorios en la UCI, quedando enmarcado específicamente en los procesos de solicitud de instalación de programas en los laboratorios docentes de la UCI.

Como objetivo general de la investigación se plantea desarrollar un sistema para la informatización del proceso de control de solicitudes de instalación de programas en los laboratorios de la UCI y para darle cumplimiento a este objetivo y alcanzar los resultados requeridos se realizarán las siguientes tareas de investigación:

- 1- Estudiar los principales sistemas existentes que brinden servicios de gestión de solicitud de programas.
- 2- Estudiar la tesis: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC desarrollada el curso anterior (2006-2007).
- 3- Analizar las herramientas y lenguajes que se utilizarán en la elaboración del sistema.
- 4- Estudiar las metodologías y librería con las cuales se desarrollará el sistema.

La estructuración del trabajo de diploma está compuesta por 5 capítulos, los cuales contienen el resultado de la investigación.

Capítulo 1: "*Marco Teórico de la Investigación*". Se refleja una valoración del estudio realizado al trabajo de diploma desarrollado el curso anterior (2006-2007) además, se hace una breve descripción de las tecnologías que se utilizarán en el desarrollo del producto y el significado de las palabras esenciales para la comprensión de las actividades del negocio.

Capítulo 2: "*Características del Sistema*". Se reflejan los requerimientos tanto funcionales como no funcionales con los que cumplirá la aplicación. Además se presentan y describen los procesos del negocio y los actores y casos de usos del sistema.

Capítulo 3: "*Análisis y Diseño del Sistema*". Se representa el modelado de las clases del análisis y del diseño, así como la secuencia en la que se realizan las acciones entre las clases, paquetes e interfaces representadas, y por último se muestra el modelo lógico y físico de datos.

Capítulo 4: "*Implementación*". Se representan los diagramas propios de la disciplina, como son: el diagrama de componentes y de despliegue.

Capítulo 5: "*Estudio de la Factibilidad*". Se detalla la estimación del esfuerzo, el tiempo, el costo y los beneficios que aportará la aplicación a desarrollar.



### **CAPÍTULO 1: MARCO TEÓRICO DE LA INVESTIGACIÓN.**

#### **1.1. Introducción**

En el presente capítulo se refleja una valoración del estudio realizado al Módulo de Servicio: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC. Se exponen además, mediante una breve descripción, los principales conceptos que sustentan la investigación, y se muestra el estado del arte de las tecnologías, lenguajes y herramientas a emplear en el desarrollo de la aplicación.

#### **1.2. Estudio de sistemas existentes vinculados a la solicitud de servicios.**

Según los resultados obtenidos en la investigación no existe actualmente un sistema que gestione los pedidos de instalación de programas, sí existen sistemas que instalan programas en los ordenadores, como es el caso de Appupdater, y servicios que se prestan a domicilio de instalación de programas, pero una aplicación que controle el pedido de instalación de programas sólo se encontró en la Universidad de las Ciencias Informáticas, y es el caso del Módulo de Servicio: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC. Este sistema se realizó el curso anterior (2006-2007) como un trabajo de diploma; el mismo ofrece las funcionalidades de gestionar las solicitudes de instalación de programas en los laboratorios docentes; es decir, se encarga de las peticiones de instalación de imagen y de nuevos programas que se necesiten. Además, corrige los posibles errores que puedan presentar las instalaciones realizadas. Para su desarrollo se utilizó el lenguaje de programación PHP, el gestor de base de datos MySQL 5.0 y se diseñó la interfaz en el Dreamweaver 8.0.

Este sistema no se publicó ni utilizó en la universidad, debido a que poseía errores y no proyectó los resultados esperados por el cliente, además, incumple con requerimientos tan importantes como el rendimiento y la prevención de errores innecesarios, que son objetivos esenciales del módulo Control de Solicitudes de Software, el cual se encargará de la configuración de solicitudes de instalación de programas en los laboratorios docentes de la universidad, y se utilizarán en su construcción tecnologías y herramientas de código abierto.

## 1.3. Tecnologías, Lenguajes de Programación y Librerías.

En la actualidad, gracias al avance de Internet, han surgido diversas tecnologías y lenguajes para la creación de aplicaciones Web, las cuales ofrecen grandes facilidades para establecer comunicación con el usuario mediante las páginas Web, debido a su capacidad para ser visualizadas desde cualquier parte del mundo haciendo uso de un navegador.

### 1.3.1. Lenguajes de Programación.

#### 1.3.1.1. PHP.

PHP es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones; convirtiéndolo en el favorito de millones de programadores en todo el mundo.

#### Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.

#### PHP 5.2 es una versión de PHP que además incluye:

- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.

- Mejor soporte a XML.

### 1.3.2. Librería ExtJS 2.0.

ExtJS es una librería de componentes que facilita las herramientas necesarias para la creación de aplicaciones Web con excelentes gráficos; ya que posee una considerable colección de elementos para el diseño de interfaces, ventanas, pestañas, menús, tablas, etc.

Brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

Actualmente ExtJS es considerado un FrameWork independiente; ya que a principios del 2007 se creó una compañía para comercializar y dar soporte al mismo, dicha compañía proporciona los servicios de consultoría necesarios para ayudar a los clientes en el aprovechamiento máximo de las ventajas de ExtJS. Es importante señalar que la ExtJS 2.0 tiene dos tipos de licencias, LGPL (Open Source) y la comercial, esta última es obligatoria si se desea obtener soporte.

### 1.3.3. .AJAX.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML) facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor. De esta forma, es posible realizar cambios sobre una página sin necesidad de recargarla, aumentando la interactividad, velocidad y usabilidad de la misma.

“AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.”<sup>1</sup>

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.

---

<sup>1</sup> Eguíluz Pérez, Javier. 2007. Libros Web. *Introducción a AJAX*. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosWeb.es/ajax>.

- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además:

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, actualizando solamente porciones de la página.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial.

A continuación se explican las características más importantes de algunas de las tecnologías que componen AJAX:

### **JSON**

Notación de Objetos de JavaScript (JavaScript Object Notation) constituye un formato ligero para el intercambio de datos abierto y basado en texto. JSON está pensado principalmente para usarse en aplicaciones Web ya que permite transmitir información estructurada de forma asíncrona entre el servidor y los clientes.

JSON es un subconjunto de la notación literal de objetos de JavaScript y es usado generalmente con ese lenguaje. Sin embargo, los tipos básicos y las estructuras de datos de la mayoría de los lenguajes de programación también pueden ser representados en JSON y el formato puede por tanto ser usado para intercambiar datos estructurados entre programas escritos en diferentes lenguajes. Existe código para analizar y generar JSON en una gran cantidad de lenguajes como: ActionScript, C, C#, Java, JavaScript, Perl, PHP, Python, Ruby y Smalltalk.

#### Características de JSON:

- La sintaxis es muy concisa por lo que requiere menos codificación y procesamiento.
- No se necesita código de aplicación adicional para analizar texto.

- Es legible e independiente de la plataforma.
- Soporta diversos lenguajes de programación.

Este formato de intercambio de datos es relativamente nuevo, por lo que no cuenta con el soporte de proveedores de XML; sin embargo es ante todo un formato de datos que no está limitado a las aplicaciones Web y prácticamente se puede usar en cualquier escenario en que los sistemas necesiten intercambiar o almacenar información estructurada como texto.

### **XHTML**

"XHTML es el acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos."<sup>2</sup>

XHTML reúne la capacidad de formato de HTML y se consolida con la formalidad del XML (y sus reglas) a la hora de estructurar documentos para la portación de datos. Está encaminado al uso de un etiquetado correcto, por lo que exige una serie de requisitos básicos a cumplir en cuanto al código.

Algunos de estos requisitos son:

- Elementos correctamente anidados.
- Etiquetas en minúsculas.
- Elementos cerrados correctamente.
- Atributos de valores entrecomillados.

### **CSS**

Es un lenguaje de hojas de estilos (Cascading Style Sheets) creado para controlar la presentación de documentos estructurados y escritos en XHTML, aspectos como: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en

---

<sup>2</sup> Eguíluz Pérez, Javier.2007. Libros Web. *Introducción a XHTML*. [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosWeb.es/xhtml>.

un documento, esto permite un control mayor del documento y sus atributos, convirtiendo al XHTML en un documento muy versátil y liviano.

“Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.”<sup>3</sup>

Entre los beneficios concretos de CSS se encuentran<sup>4</sup>:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, entre otros).

### **JavaScript**

Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

---

<sup>3</sup> **Eguíluz Pérez, Javier. 2007.** Libros Web. *Introducción a CSS*. [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosWeb.es/css>.

<sup>4</sup> **HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio web. [Online] [Cited: Marzo 15, 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.

### **1.3.4. Servidor Web.**

Apache es un servidor Web potente, flexible y disponible para distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

#### Características de Apache:

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web tiene una fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación ha dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP, los cuales instalan el Apache y el PHP configurados para su uso.

### **1.3.5. Sistema Gestor de Base de Datos.**

PostgreSQL es un gestor de bases de datos relacional orientado a objetos, libre y gratuito. Posee una gran escalabilidad, ya que es capaz de soportar una gran cantidad de peticiones simultáneas, ajustándose al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima.

#### Presenta las siguientes propiedades:

- **Atomicidad:** Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.
- **Consistencia:** Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- **Durabilidad:** Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.
- **Aislamiento:** Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.

#### PostgreSQL presenta las siguientes características:

- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.
- Posee confiabilidad, seguridad y flexibilidad.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Soporta el uso de índices, reglas y vistas.

### 1.4. Arquitectura.

#### 1.4.1. Patrón Modelo – Vista – Controlador.

El Modelo-Vista-Controlador (MVC) se creó para Smalltalk a finales de los setenta. A partir de entonces su uso se ha ido extendiendo cada día más para la construcción de sistemas software con interfaz gráfica. MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

La **Vista** es la información presentada al usuario. Una vista puede ser una página Web o una parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

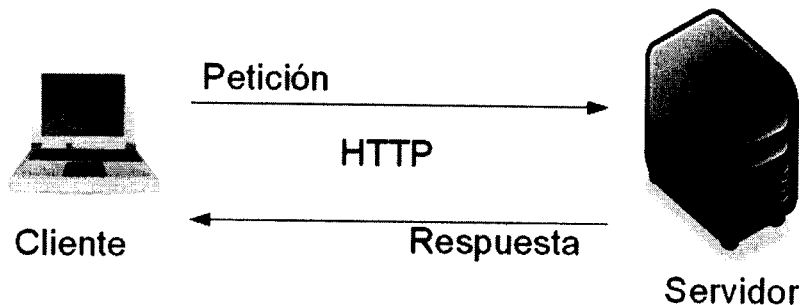
#### 1.4.2. Arquitectura Cliente/Servidor.

La arquitectura cliente-servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente, que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos.

Puede presentarse como uno a varios clientes y uno o más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos. Un único servidor típicamente sirve a una multitud de



clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.



**Figura 1 Arquitectura Cliente/Servidor**

Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

## 1.5. Modelación de Funciones mediante IDEF0.

IDEF0 (Integration Definition for Function Modeling) consiste en una serie de normas que definen la metodología para la representación de funciones. Representa de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 posee las siguientes características:

- Es capaz de representar gráficamente los principales procesos del negocio de una empresa a cualquier nivel de detalle.
- Permite analizar, documentar y mejorar los procesos del negocio.

- Facilita la comunicación y captura de información.
- Es muy coherente y simple, pero a la vez rigurosa y exacta.
- Puede ser generado fácilmente por una gran variedad de herramientas gráficas en computadores.
- Posee flexibilidad ante cambios.

La notación IDEF puede ser representada de dos formas:

La forma **AS-IS** (tal como es) es la empleada por el negocio para representar los procesos a la manera actual, antes de la automatización.

La forma **TO-BE** (como debe ser) es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización.

### **1.6. Lenguaje de Modelado y Metodología de Desarrollo.**

#### **1.6.1. Lenguaje de Modelado UML.**

UML es un lenguaje utilizado para el modelado de un sistema, permitiendo en mayor o menor medida representar todas las disciplinas de un proyecto informático: desde el análisis con los casos de uso, hasta la implementación y configuración mediante los diagramas de despliegue.

UML permite:

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

Aporta las siguientes ventajas:

- Permite realizar una verificación y validación del modelo realizado.
- El modelado con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.
- Permite generar código a partir de los modelos y a la inversa, lo que posibilita la constante actualización.

- Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es suficientemente expresivo como para modelar sistemas que no son informáticos, como: flujos de trabajo en una empresa y diseño de la estructura de una organización.

### 1.6.2. Metodología de Desarrollo RUP.

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”<sup>5</sup>

RUP (Rational Unified Process) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML), constituye una metodología estándar muy utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

- Se utiliza en proyectos que se desarrollan a largo plazo.
- Permite una mejor comunicación entre ingenieros de software y de negocio, manejando un lenguaje común para facilitar la comprensión de procesos del negocio y su traducción a las funcionalidades que brindará el sistema.
  - Genera un volumen considerable de documentación, posibilitando que los cambios realizados en los miembros del equipo no resulte un factor negativo para el avance del proyecto.
  - Propone el desarrollo en ciclos e iteraciones con los artefactos que se generan, siendo esto un elemento importante para alcanzar una categoría de certificación en el desarrollo del software.
  - Asegura la producción de software de calidad dentro de plazos y presupuestos predecibles.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen en los diferentes flujos de trabajo, representan la realización de los casos de uso.

---

<sup>5</sup> Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000. El Lenguaje Unificado de Modelado. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e incremental:** Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

### **1.7. Herramientas Utilizadas.**

#### **1.7.1. Visual Paradigm.**

Visual Paradigm para UML (VP-UML) es una poderosa herramienta CASE multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software. Está diseñado para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistema y quienes estén interesados en la construcción de sistemas de software confiables mediante el uso de la Orientación a Objetos.

Este software facilita una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML soporta un conjunto de lenguajes (Java, C + +, PHP, Ada y Python), tanto en generación de código como ingeniería inversa.

Entre sus características principales se pueden citar:

- Soporta UML en su versión 2.1.
- Facilita la comunicación de todo el equipo de desarrollo mediante el uso de un lenguaje estándar común.
- Posibilita el desarrollo de la ingeniería directa e inversa.
- Durante todo el ciclo de desarrollo el modelo y el código permanecen sincronizados, permitiendo la generación de código a partir de diagramas y viceversa.
- Se encuentra disponible en múltiples versiones y plataformas.
- Permite la generación de bases de datos a partir de la transformación de diagramas de Entidad-Relación en tablas de base de datos y viceversa.

### 1.7.2. Aptana.

Aptana Studio es un IDE de desarrollo para aplicaciones de la Web 2.0, gratuito y de código fuente abierto. Está basado en el conocido entorno de desarrollo Eclipse, también de código abierto; pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a HTML, CSS y JavaScript.

Soporta varias librerías como: Prototype, Yahoo UI y JQuery, pudiendo combinarlas fácilmente en una aplicación. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse y se puede encontrar para tres plataformas fundamentales: Windows, Mac OS y GNU/Linux.

La gestión de proyectos, vista previa, autocompletado de código y gestión de documentación, son algunas de las características similares que presenta Aptana con otros entornos de desarrollo integrado (Eclipse, C++ Builder, Visual Studio. NET), pero además:

- Visualiza los errores de sintaxis a medida que se escribe.
- Presenta una previa visualización de estilos CSS con el editor CSS.

### 1.7.3. GIMP.

GIMP es el acrónimo para GNU Image Manipulation Program. Es un programa libre distribuido bajo la licencia GPL, apropiado para tareas como: arreglo fotográfico, composición y edición de imagen. Es especialmente útil para la creación de logotipos y otros gráficos para páginas Web. Inicialmente se desarrolló para sistemas Unix; sin embargo, actualmente existen versiones totalmente funcionales para Windows y para Mac OS.

#### Características fundamentales de GIMP:

- Lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, gif, png, pcx y tiff.
- Es capaz de importar ficheros en pdf y también imágenes vectoriales en formato svg.
- Permite la automatización de muchos procesos mediante macros o secuencias de comandos.
- Dispone de varias herramientas o filtros para la manipulación de los colores y el aspecto de las imágenes.

### 1.7.4. PGAdmin III.

Es una aplicación gráfica usada para la gestión de PostgreSQL, siendo la más completa y popular con licencia Open Source. PGAdmin está escrito en C++ y utiliza la librería gráfica multiplataforma wxWidgets, permitiendo que se pueda usar en sistemas operativos como: GNU/Linux, FreeBSD, Solaris, Mac OS y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3, ejecutándose en cualquier plataforma.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples, hasta desarrollar bases de datos complejas.

El producto también incluye:

- Editor SQL con resaltado de sintaxis.
- Editor de código de la parte del servidor.
- Amplia documentación para ayudar a los usuarios menos experimentados.

## 1.9 Palabras claves para la comprensión del problema existente.

- 1- Intermediario: Es un profesor que se encarga de realizar todos los trámites requeridos para la instalación de una imagen o de un nuevo programa en los laboratorios. Es quien se relaciona con los profesores que hacen el pedido y el personal autorizado a gestionar la solicitud.
- 2- Técnico de software: Es la persona responsable de los programas que están o deben instalarse en los laboratorios docentes de la universidad.
- 3- Especialista general de software: Es el encargado de supervisar a los técnicos de software, y en caso de que el técnico de software tenga alguna dificultad a la hora de tomar una decisión debe consultarlo con este individuo.
- 4- Departamento central: Es donde se agrupa el personal responsable de una determinada especialidad a nivel central.
- 5- Departamento de la especialidad: Es donde se agrupan los profesores pertenecientes a una especialidad determinada de una facultad específica.
- 6- Autorización: Es la firma que refleja el vicedecano(a) de formación en el listado de los programas solicitados aprobando la instalación de los mismos.
- 7- Programas: Son los componentes no tangibles que necesita un ordenador para que los usuarios puedan realizar operaciones y tareas.

- 8- Imagen: Es el conjunto de programas que se instalan en los laboratorios docentes a principios de semestre.
- 9- Características de un programa: Es el espacio en disco y la memoria RAM que necesita un programa. También, la categoría a la que pertenece y la compatibilidad con los sistemas operativos Linux y Windows.
- 10- Área: es el espacio que ocupa en la UCI un edificio docente.
- 11- Edificio docente: Sitio en el que los estudiantes de la UCI reciben las asignaturas de la carrera.
- 12- Laboratorio: Es un local equipado con diversos medios computacionales (monitores, CPU, teclados, ratones, cables de red, switch, etc.), donde se realizan clases prácticas de las disímiles asignaturas que se imparten en la UCI.
- 13- Facultad: Es una subdivisión (que representa un perfil) de la universidad, la UCI posee 10 subdivisiones, sin tener en cuenta las correspondientes a las MINI-UCI.
- 14- Prácticas en los laboratorios: Son las clases que reciben los estudiantes en los laboratorios, en las cuales se practican lo estudiado en las conferencias.
- 15- Curso Optativo: Asignatura adicional que se imparte en el transcurso del semestre, en la cual no es obligatorio matricular y contribuye a la formación profesional de los estudiantes.
- 16- Vicedecano de formación: Es la persona que supervisa y controla todas las actividades y ajustes en la docencia.
- 17- Asignatura: conocimientos propios de la carrera de ingeniería que se imparten en la UCI.
- 18- Elementos de un programa: Subprogramas que integran el programa principal(es el programa que se solicita).
- 19- Estado de un programa: Categoría que se le asignan a los programas solicitados para darle a conocer a la persona que realizó el pedido si su programa puede o no ser instalado en los laboratorios docentes de la UCI.
- 20- Perfil: Término que se utiliza para representar las ramas en las que se especializan cada una de las facultades de la UCI.

### **1.10 Conclusiones.**

En este capítulo se ha mostrado el resultado del estudio realizado al sistema desarrollado en el curso (2006-2007) Módulo de Servicio: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC, ha quedado reflejada la investigación efectuada sobre las metodologías, lenguajes y herramientas que se van a utilizar en el desarrollo de la aplicación y la definición de las palabras claves para el entendimiento del negocio.



## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.**

### **2.1 Introducción**

En este capítulo se describen brevemente los problemas existentes en la Universidad de las Ciencias Informáticas referentes a las solicitudes de instalación de programas en los laboratorios docentes. Se mencionan y detallan los procesos que serán automatizados, las actividades que se realizan en el negocio y capacidades, condiciones y características que debe poseer el producto a elaborar. Además, se representa el modelo de casos de uso del sistema con sus respectivas descripciones.

### **2.2 Objeto de Estudio.**

#### **1.7.5. Problema y Situación Problemática.**

En la UCI se imparten hoy en día una serie de materias, las cuales, según el plan de estudio del centro, precisan realizar prácticas en los laboratorios y para que estos encuentros se realicen con calidad y generen en los estudiantes los conocimientos requeridos es preciso tener instalados una serie de programas. Estos se instalan a principios de cada semestre mediante una imagen; pero puede darse el caso que un profesor en el transcurso del semestre requiera un programa que no esté instalado en los laboratorios para impartir un curso optativo por una o varias semanas.

Para efectuar las solicitudes de instalaciones se realizan una serie de actividades que son indispensables para el control adecuado de las mismas. Estas acciones son:

- Entregar el listado de los posibles programas a instalar a los departamentos de las facultades por parte de los departamentos centrales.
- Hacer entrega del listado de los programas necesitados por cada departamento de la facultad al intermediario.
- Pedir autorización de las instalaciones al vicedecano(a) de formación de la facultad.
- Hacer el pedido de los programas que se deben instalar en los laboratorios al técnico de software que atiende el docente al cual pertenece la facultad.

Estas son las tareas que se realizan en una solicitud de instalación de imagen, pero si se desea un programa que no haya sido incorporado en la imagen se hacen las siguientes gestiones:

- El profesor que impartirá el curso optativo se dirige al intermediario y le entrega el programa que necesita
- El Intermediario realiza las tareas (mencionadas anteriormente) correspondientes a una solicitud de instalación.

Estas tareas en muchas ocasiones no se realizan, debido a que: los departamento centrales no les hacen llegar a los departamentos de las facultades la propuesta de los programas a instalar; además, no se le asigna el papel del intermediario a ningún profesor, provocando que cualquier educador se dirija al Especialista General de Software a pedir un determinado programa sin una previa autorización.

Estas deficiencias en los procesos de solicitudes se deben a que los pedidos se realizan de forma manual, y en su realización están involucradas muchas personas, las cuales trabajan sin una apropiada organización, provocando que el nuevo programa solicitado no esté instalado para la fecha establecida y que la imagen elaborada no posea todos los programas requeridos; ya que a sus desarrolladores les hacen llegar datos erróneos; todas estas fallas ocasionan a su vez, pérdida innecesaria del tiempo de la jornada laboral por parte de las personas involucradas en el proceso de petición e instalación de programas, insuficiencia en el control de las peticiones de instalación de programas y atrasos en el proceso docente-educativo.

La universidad no cuenta con un sistema que gestione las acciones referentes a las solicitudes de instalación de programas en los laboratorios docentes, siendo esta la causa que ocasiona el problema a resolver por el módulo: Control de Solicitudes de Software, el cual se encargará de configurar toda la información de las peticiones que se realicen.

### **1.7.6. Objeto de Automatización.**

Para llevar un control adecuado de las solicitudes de programas se desean automatizar los siguientes procesos:

Configurar solicitud de instalación, tanto de imagen como de nuevos programas: Permite realizar una nueva solicitud y modificar una ya existente. Además, el sistema debe registrar los datos del intermediario (nombre, apellidos, número de solapín, área y facultad a la que pertenece).

Mostrar Información de las Solicitudes de Instalación, tanto de imagen como de nuevos programas: Brinda al usuario que realizó la solicitud la información referente a la misma, incluyendo los estados de los programas pedidos. El sistema, en caso de que un programa no pueda ser instalado en los

laboratorios, es decir, que su estado sea cancelado, le proporciona al intermediario una justificación de dicha decisión y le da la posibilidad de introducir nuevos programas que sustituyan a los cancelados.

Asignar Estados a los Programas: Permite registrar los estados (Aceptado, Cancelado o Pendiente) de los programas solicitados. El pendiente indica que todavía no se ha estudiado o está en análisis la instalación y el cancelado que no puede ser instalado por algún motivo, motivo que debe registrar el sistema para dárselo a conocer al solicitante en el momento que lo desee.

Mostrar Historial de Solicitudes: Brinda la posibilidad al personal autorizado de observar, según el criterio introducido (mostrar por área, por curso y por rango de fecha), el listado de todas las solicitudes realizadas, así como la información contenida en cada una de ellas y permite generar un documento que es posible imprimir, el cual contiene los programas cuyas instalaciones fueron aceptadas.

Mostrar Programas Instalados en los Laboratorios Docentes: Ofrece a los usuarios del sistema los programas instalados en los laboratorios de cada uno de los docentes de la universidad.

En estos momentos la universidad no cuenta con un sistema que gestione los pedidos de instalación de programas en los laboratorios docentes; aunque el curso pasado se desarrolló una aplicación Web que se encargaba de controlar las peticiones de instalación de programas en los laboratorios y reportar los errores que poseían dichas instalaciones. Para su desarrollo se utilizó el lenguaje de programación PHP, el gestor de base de datos MySQL 5.0 y se diseñó la interfaz de usuario en el Dreamweaver 8.0.

Este sistema no se publicó ni utilizó en la universidad, debido a que poseía errores y no cumplió con las necesidades planteadas por el cliente.

### **1.7.7. Información que se Maneja.**

La información que se opera son los datos referentes a las solicitudes que se realizan, ya sea para solicitar la instalación de una imagen, como la de un nuevo programa que no haya sido instalado a principios de semestre. De estas solicitudes es necesario conocer: la fecha en las que fueron realizadas, las características de los programas solicitados y los datos del intermediario (nombre y apellidos, área y facultad a la que pertenece, número solapín y carnet de identidad).

Además se maneja el estado de los programas demandados, que son: aceptado, cancelado y pendiente. Se procesa también un documento que es donde van a estar implícitos todos los programas con sus características y la firma del vicedecano(a) de formación autorizando las instalaciones, el cual servirá como constancia de que esos fueron los programas que en verdad se solicitaron, y no se reclamen luego de instalada la imagen otros que no fueron incluidos en la solicitud.

### **2.3 Propuesta del Sistema.**

Se propone, para resolver los inconvenientes existentes en el control de las solicitudes de instalación de programas, desarrollar una aplicación Web que presente una interfaz amena, sencilla, entendible y fácil de usar para todos los usuarios. El sistema debe permitir realizar solicitudes de instalación de imagen y de nuevos programas, en el caso de una solicitud de nuevos programas, el usuario debe especificar los laboratorios y la fecha en la que deben estar instalados, aunque podrá ser insertada sólo si se realiza con 15 días como mínimo de antelación. Una vez que la aplicación haya registrado las solicitudes podrá ofrecer a sus usuarios las posibilidades de modificarlas, observar los programas instalados en todos los laboratorios docentes, consultar el historial de solicitudes en dependencia del criterio introducido, que puede ser por cursos, por docentes y dado un rango de fechas y decidir si un programa puede o no ser instalado, es decir, asignarle el estado de aceptado, cancelado o pendiente; en caso de cancelar un programa el usuario debe introducir una justificación de la decisión que ha tomado y una vez guardada, la aplicación le proporcionará al solicitante dicha justificación.

A diferencia de la aplicación: "Módulo de Servicio. Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC" el nuevo sistema a desarrollar cumplirá cabalmente con todas las necesidades planteadas por el cliente además, en su desarrollo se utilizarán tecnologías y herramientas de código abierto, contribuyendo y reforzando así la batalla de migrar a software libre que se está librando en Cuba debido a las limitaciones impuestas de comercializar e importar tecnologías y software productivos en todo el mundo.

### **2.4 Modelo de Negocio.**

El módulo Control de Solicitudes de Software está compuesto por 3 procesos:

- 1- Solicitud de Instalación de Imagen.
- 2- Solicitud de Instalación de Nuevos Programas.
- 3- Asignar Estados a los Programas Pedidos.

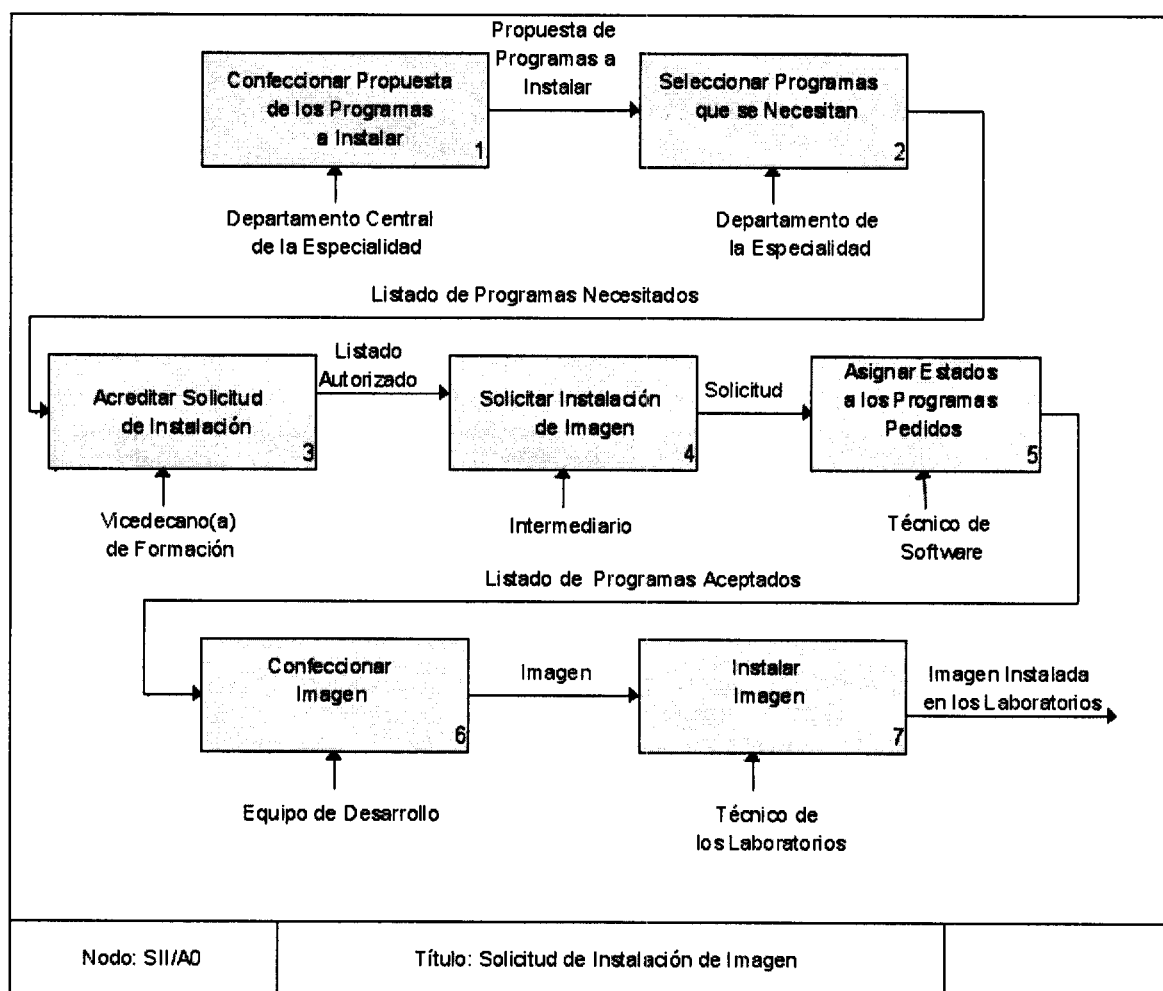
El primero se realiza a principios de cada semestre, cuando los departamentos de las especialidades realizan los pedidos de los programas que necesitan para impartir sus clases en los laboratorios docentes. El segundo se efectúa cuando un profesor tiene la necesidad de impartir un curso optativo, y requiere un determinado programa que no fue incluido en la imagen, este se lleva a cabo en el

transcurso del semestre y el último se realiza como una actividad más en los dos procesos antes mencionados, aunque se origina en el proceso Solicitud de Instalación de Imagen.

### **1.7.8. Descripción y Representación de los Procesos del Negocio.**

Proceso: Solicitud de Instalación de Imagen.

- 1- Confeccionar Propuesta de los Programas a Instalar: La actividad se inicia cuando los profesores del Departamento Central confeccionan, según los programas que existan en la universidad, una propuesta de los programas que deben ser instalados en los laboratorios docentes. Esto lo hacen basándose en las necesidades de cada facultad, según su perfil. Esta actividad arroja como resultado un listado (propuesta) con los programas que deben ser instalados.
- 2- Seleccionar Programas que se Necesitan: Comienza cuando los profesores de los departamentos de cada especialidad seleccionan los programas que necesitan para impartir sus clases basándose en la propuesta del Departamento Central. Esta actividad tiene como resultado un listado de los programas requeridos por cada departamento.
- 3- Acreditar Solicitud de Instalación: Se inicia cuando el intermediario se dirige con las necesidades de cada departamento al vicedecano(a) de formación de su facultad para que autorice las instalaciones requeridas. La autorización es representada mediante una firma en el listado de los programas pedidos por los profesores. Esta actividad es realizada por el vicedecano(a) de formación y arroja como resultado el listado acreditado.
- 4- Solicitar Instalación de Imagen: Esta actividad se inicia una vez que el intermediario se dirige, con la autorización obtenida, al técnico de software de su docente y le hace un pedido de instalación de imagen. Esta actividad tiene como resultado la solicitud de instalación de imagen, la cual va a contener el listado autorizado y los datos del intermediario.
- 5- Asignar Estados a los Programas (Consultar descripción del proceso: Asignar Estados a los Programas Pedidos).
- 6- Confeccionar Imagen: Se inicia una vez que se le hace llegar al equipo de desarrollo el listado de los programas que fueron aceptados y confeccionan la imagen. Esta actividad aporta como resultado la imagen lista para ser instalada en los laboratorios.
- 7- Instalar Imagen: Comienza cuando se les entrega a los técnicos de los laboratorios la imagen para que sea instalada en todas las computadoras; resultando así, la instalación de los programas requeridos.

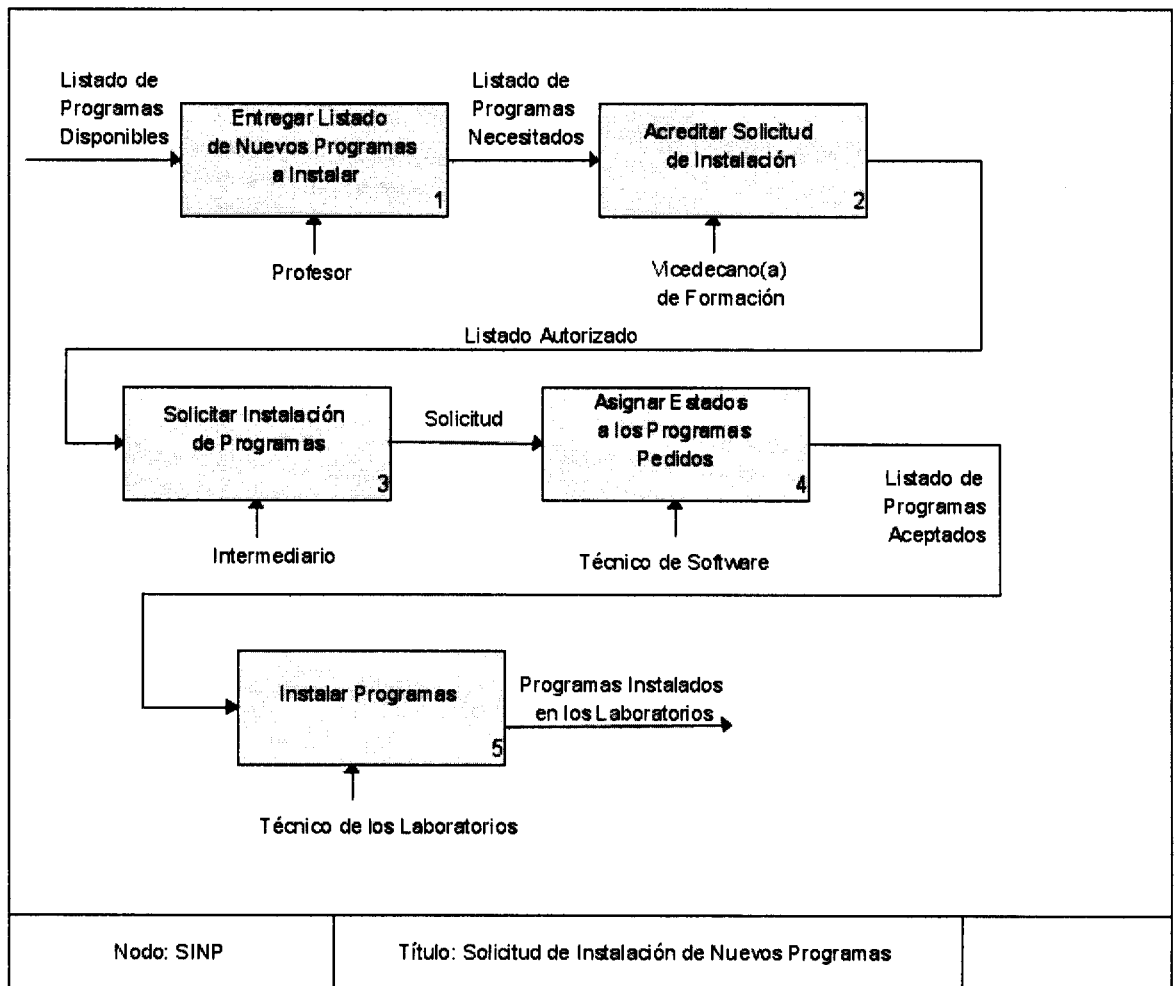


**Figura 2.1: Proceso del Negocio: Solicitud de Instalación de Imagen.**

Proceso: Solicitud de Instalación de Nuevos Programas.

- 1- Entregar Listado de Nuevos Programas a Instalar: La actividad se inicia cuando el profesor que va a impartir el curso optativo le hace entrega de los programas que necesita al intermediario. Esta acción arroja como resultado una solicitud que contiene los programas que precisa el educador.
- 2- Acreditar Solicitud de Instalación: Se inicia cuando el intermediario se dirige con las necesidades del profesor al vicedecano(a) de formación de su facultad para que autorice las instalaciones. La autorización es representada mediante una firma en el listado de los programas pedidos. Esta actividad es realizada por el vicedecano(a) de formación y arroja como resultado el listado acreditado.

- 3- Realizar Petición de la Nueva Instalación: Una vez obtenida la autorización el intermediario se dirige al técnico de software de su docente con el documento firmado y hace un pedido de instalación de los programas que requiere el profesor en uno o varios laboratorios hasta la fecha de culminación del curso optativo, es válido aclarar que esta petición debe realizarse con 15 días como mínimo de antelación, porque si no, no se garantiza la instalación del mismo para la fecha que lo necesite el profesor. Esta actividad va a proyectar como resultado un pedido de instalación, el cual consiste en los datos de la persona que realiza la solicitud con el documento autorizado por el vicedecano(a) de formación.
- 4- Asignar Estados a los Programas (Consultar descripción del proceso: Asignación de Estados a los Programas Solicitados).
- 5- Instalar Programas: Esta actividad se realiza cuando se le entrega a los técnicos de los laboratorios los nuevos programas a instalar, estos son quienes ejecutan la tarea, y como resultado ofrecen los programas requeridos instalados en los laboratorios.



**Figura 2.2: Proceso del Negocio: Solicitud de Nuevos Programas.**

Proceso: Asignación de Estados a los Programas Solicitados.

1- Verificar Características de los Programas: Esta actividad comienza cuando el técnico de software comprueba si las características de los programas solicitados son compatibles con el hardware que poseen las computadoras donde serán instalados. Mediante la misma se arrojan tres posibles respuestas: Una positiva, una negativa y una inexacta. La positiva se obtiene cuando las características de los programas son compatibles con las del hardware de los ordenadores. El caso de la respuesta negativa es cuando sus características no son las adecuadas al hardware de las computadoras y la respuesta inexacta es cuando ocurre lo mismo que la positiva, pero en este caso el



programa que se ha pedido requiere de mucho espacio en disco o memoria RAM, pero puede instalarse, y tendría entonces que analizarse su instalación.

2- **Aceptar la Instalación de Programas:** Esta actividad se realiza cuando se decide que uno o varios programas serán instalados (respuesta positiva). Es realizada por el técnico de software y arroja como resultado el listado de los programas aceptados.

3- **Ubicar en Pendiente la Instalación de Programas:** Esta actividad se inicia cuando el técnico de software comprueba que uno o varios de los programas solicitados requieren mucha memoria y espacio en disco, que se pueden instalar, pero en su lugar se podrían ubicar otros programas que serían más óptimos. Esta actividad es realizada por el técnico de software y se obtiene como resultado el o los programas que están pendientes a ser instalados.

4- **Analizar la Instalación de Programas:** Esta tarea se inicia cuando el técnico de software le propone al intermediario otros programas que pueden ser instalados en lugar de los que están en estado pendiente. Esta actividad es efectuada por el técnico de software y el solicitante, y arroja como resultado dos decisiones: que se instalen o no los programas pendientes.

5- **Cancelar la Instalación de Programas:** Esta actividad se efectúa cuando se decide no realizar la instalación de uno o varios programas (respuesta negativa). Es realizada por el técnico de software y arroja como resultado el listado de los programas con sus estados cancelados.

5- **Solicitar otros Programas que Sustituyan los Cancelados:** Esta actividad se inicia una vez que se le informa al intermediario los programas que no pueden ser instalados debido a sus características; y este adiciona en su solicitud otros programas que sustituyan los anulados. Esta actividad es realizada por el intermediario y arroja como resultado el listado de los nuevos programas solicitados.

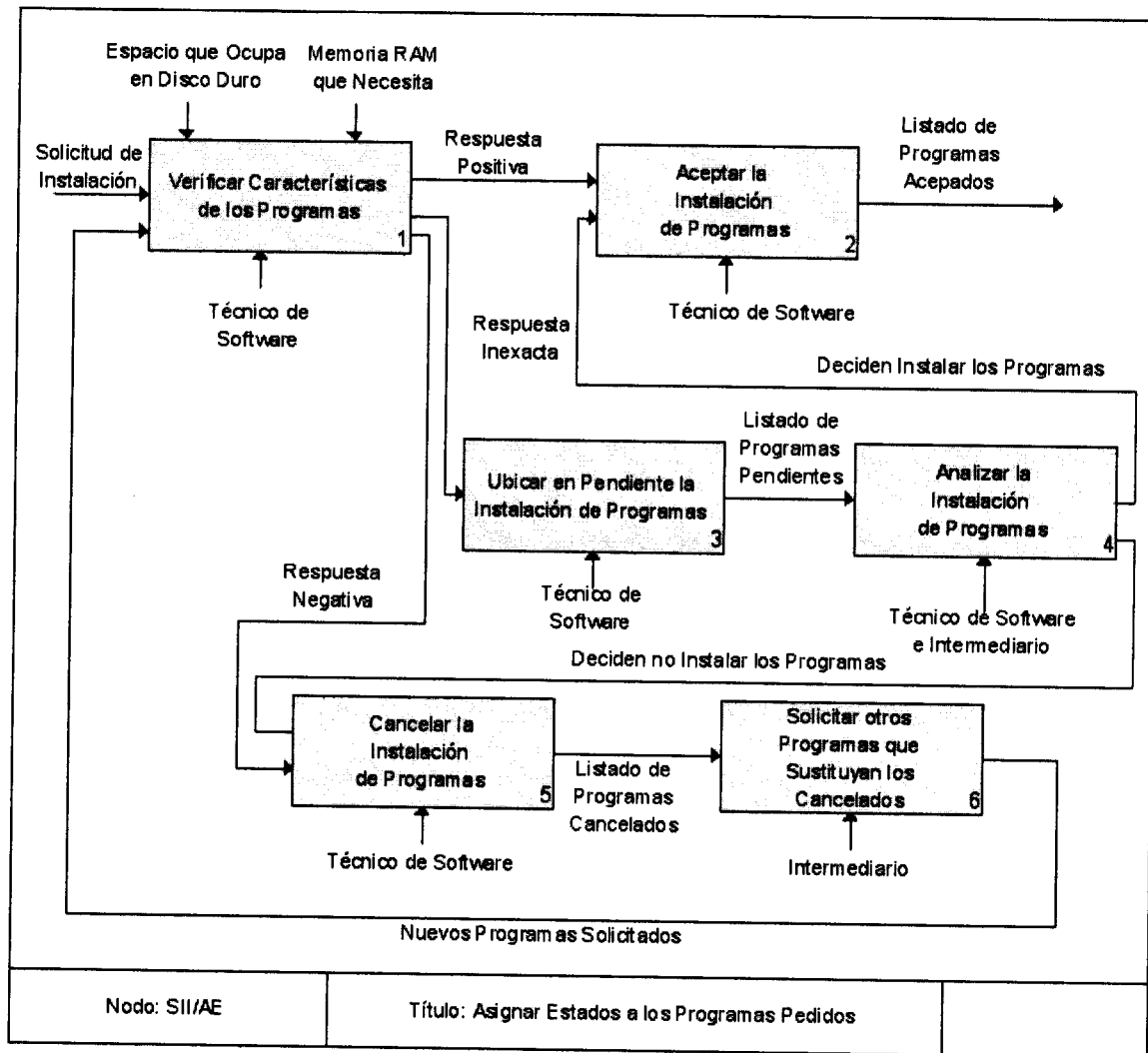


Figura 2.3: Proceso del Negocio: Asignar Estados a los Programas Pedidos.

## 2.5 Especificación de los requisitos del software.

### 1.7.9. Requerimientos Funcionales.

Las condiciones o capacidades que el sistema a desarrollar deben cumplir son:

R1: Administrar solicitud de instalación de imagen.

R1.1: Insertar una nueva solicitud de instalación de imagen.

R1.2: Modificar una solicitud de instalación de imagen.

R1.3: Mostrar solicitud de instalación de imagen.

R2: Administrar solicitud de instalación de nuevos programas.

R2.1: Insertar una nueva solicitud de instalación de nuevos programas.

R2.2: Modificar una solicitud de instalación de nuevos programas.

R2.3: Mostrar solicitud de instalación de un nuevo programa.

R2.4: Insertar fecha de inicio y fin de la instalación.

R2.5: Seleccionar los laboratorios en los que serán instalado el nuevo programa.

R3. Filtrar el nombre de un programa cuando se desea seleccionar y no está a simple vista.

R4. Sustituir los programas que su instalación fue cancelada.

R5. Controlar el estado de los programas.

R6. Generar un documento con los programas solicitados que fueron aceptados, sus características y los datos del intermediario.

R7. Visualizar los programas instalados en los laboratorios de los docentes.

R8. Mostrar historial de solicitudes.

R8.1- Mostrar solicitudes por curso.

R8.2- Mostrar solicitudes por área.

R8.3- Mostrar solicitudes dado un rango de fechas.

### **2.5.1 Requerimientos No Funcionales.**

Las propiedades, características, cualidades o condiciones que el producto (incluyendo todos los módulos del proyecto) debe cumplir son:

#### Requerimientos de Seguridad:

- La información será almacenada en una base de datos, dejando registradas todas las operaciones mediante copias de seguridad.
- El uso y manejo del sistema estará controlado; ya que la información podrá ser consultada y modificada solamente por el personal autorizado; estableciendo para ello una serie de roles con funcionalidades específicas.
- La aplicación estará funcionando a tiempo completo; de esta forma es posible que los usuarios tengan acceso (según sus permisos) en todo momento a la información solicitada.
- Es muy importante que los servidores estén sincronizados correctamente con el día y hora, ya que si ocurre un fallo el sistema no funcionaría correctamente.

#### Requerimientos de Confiabilidad:

- Todas las salidas del sistema tendrán 100% de veracidad y precisión.
- El sistema tendrá la capacidad de recuperarse rápidamente ante cualquier fallo mediante las copias de seguridad que serán realizadas.
- Se verificará constantemente la conexión con el servidor y en caso de fallo se le notificará al usuario.
- En caso de alguna dificultad con el funcionamiento del sistema, el tiempo medio de reparación deberá ser menor de 1 día.

#### Requerimientos de Rendimiento:

- El sistema debe tener un tiempo de respuesta rápido y eficiente, inferior a 10 segundos.
- La aplicación debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

### Requerimientos de Apariencia o Interfaz Externa:

- La interfaz será agradable, sencilla y sugerente; ya que su diseño simulará una aplicación de escritorio, presentará por defecto un color azul claro y textos que muestran la acción a realizar con un tipo de letra Verdana de tamaño 10pt.
- Se le brindará al usuario la posibilidad de personalizar la apariencia de interfaz externa.
- Debe contener pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema; por lo que se emplearán íconos sencillos que representarán la operación a realizar.
- El sistema será diseñado para una resolución de pantalla de 1024 X 768 píxeles.

### Requerimientos de Hardware:

El servidor Web y de base de datos deben tener como mínimo las siguientes características:

- Computador PENTIUM IV.
- 1 GB de memoria RAM.
- 10 GB de espacio libre en disco duro.

Las computadoras situadas en los puestos de trabajo de los usuarios requieren como mínimo:

- Computador PENTIUM IV.
- 512 MB de memoria RAM.

Además es necesario contar con una impresora para poder imprimir los reportes y tarjetas de red de 10/100 MB/s para la conexión.

### Requerimientos de Software:

- Las estaciones de trabajo clientes utilizarán como sistema operativo GNU/Linux en sus diversas distribuciones o la familia de Windows superior a Windows 98.
- Las máquinas clientes contarán con un navegador, ya sea el Internet Explorer en versiones superiores a la 5.0 o Mozilla Firefox en versiones superiores a la 1.5, ambos con las opciones de JavaScript y cookies habilitadas.
- El servidor Web funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el Apache en su versión 2.0 y PHP 5.2 con las librerías PDO, pgsq y json habilitadas.

- El servidor de base de datos funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el gestor de bases de datos PostgreSQL 8.2.

### Requerimientos de Usabilidad

- La interfaz será fácil de usar para los diversos usuarios que interactúen con ella.
- El sistema estará bien documentado, con el fin de lograr un mejor uso de los servicios que este ofrecerá, para ello se realizará una ayuda que explique paso a paso cada una de las funcionalidades del software.

### Requerimientos de Soporte:

- El sistema contará con la documentación apropiada para agilizar su mantenimiento y configuración.
- Se ofrecerán servicios de adiestramiento al personal que trabajará con el software.

### Requerimientos de Ayuda y Documentación en Línea:

- El sistema proporcionará en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de que necesiten realizar alguna consulta.

### Requerimientos Legales:

- Los módulos de desarrollo del proyecto SIGLA así como la documentación del mismo, son propios de la UCI y específicamente del área de los laboratorios.

## 2.6 Modelo de Casos de Uso del Sistema.

### 2.6.1 Definición de los actores del sistema.

Actor	Descripción
Intermediario	Es el encargado de gestionar solicitudes de instalación de programas, es decir; puede adicionar una nueva solicitud y modificar una ya existente y sustituir los programas que fueron cancelados. Además, puede observar las solicitudes que ha realizado.
Especialista General de Software.	Es el facultado para observar las solicitudes que se han realizado (historial de solicitudes). Su rol puede ser representado por el Técnico de Software.
Técnico de Software	Es el encargado de asignarle estados a los programas de la solicitud, o sea aceptar, cancelar o dejar en modo pendiente la instalación de cada programa del pedido.
Usuario	Es el autorizado para observar los programas que están instalados en los laboratorios de cada docente de la universidad.  Su rol es representado por: especialistas generales de software, profesores, intermediarios y técnicos de software.

**Tabla 2.1: Definición de los actores del sistema.**

2.6.2 Diagrama de casos de uso del sistema.

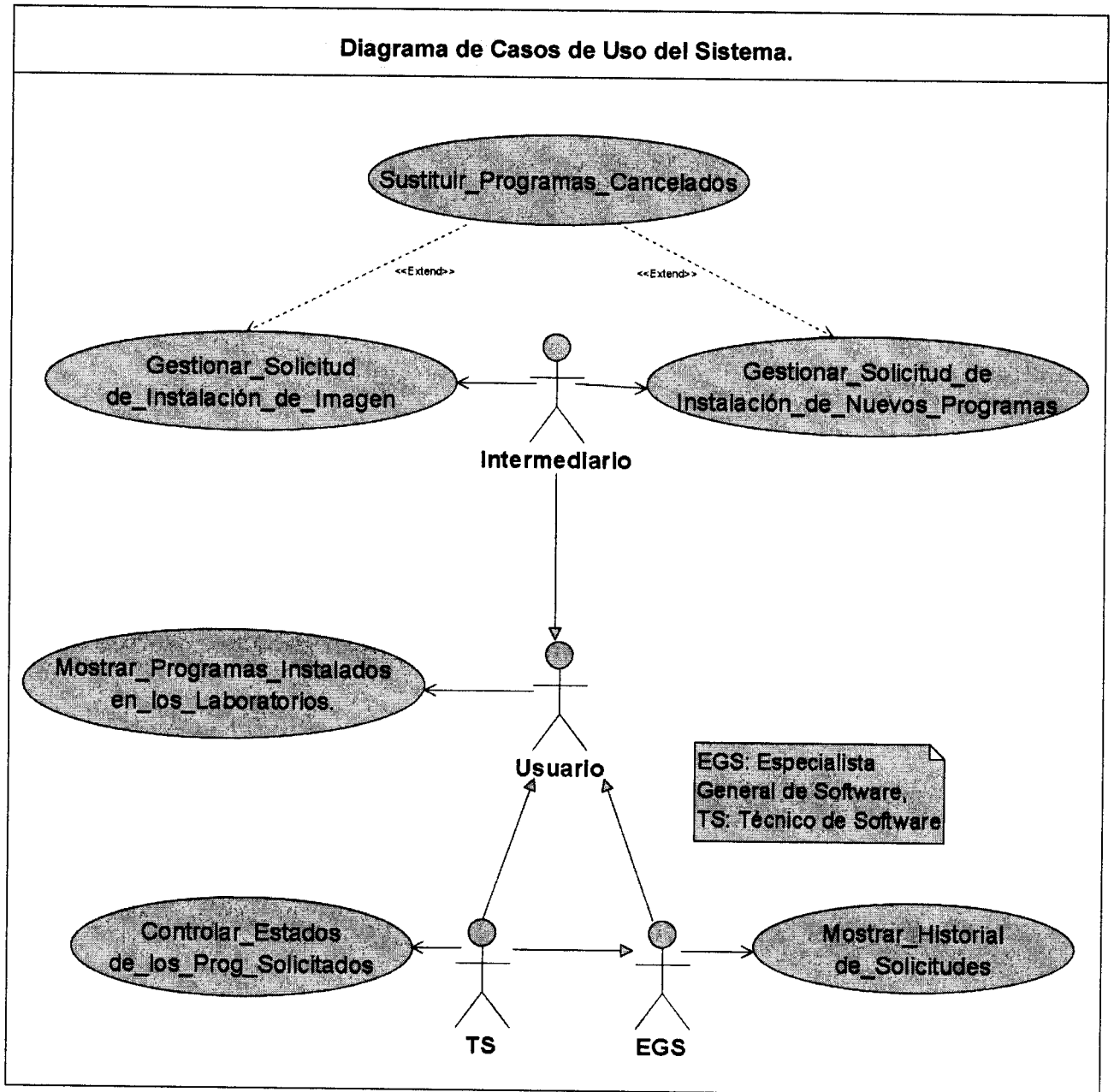


Figura 2.4: Diagrama de Casos de Uso del Sistema



### **2.1.1 Descripción de los Casos de Uso del Sistema.**

Esta descripción se encuentra en los anexos (Anexo No 1)

## **2.2 Conclusiones.**

En el desarrollo del capítulo se describió brevemente los problemas existentes en el control de las solicitudes de instalación de programas en los laboratorios docentes, las funcionalidades que brindará el sistema para resolver estos inconvenientes y los requerimientos tanto funcionales como no funcionales que debe cumplir. Además, se detallaron las actividades que se realizan en el negocio, las acciones que podrán realizar en la aplicación los usuarios en dependencia del rol que se les otorgue y se representó el diagrama de casos de uso del sistema con sus respectivas descripciones.

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO.**

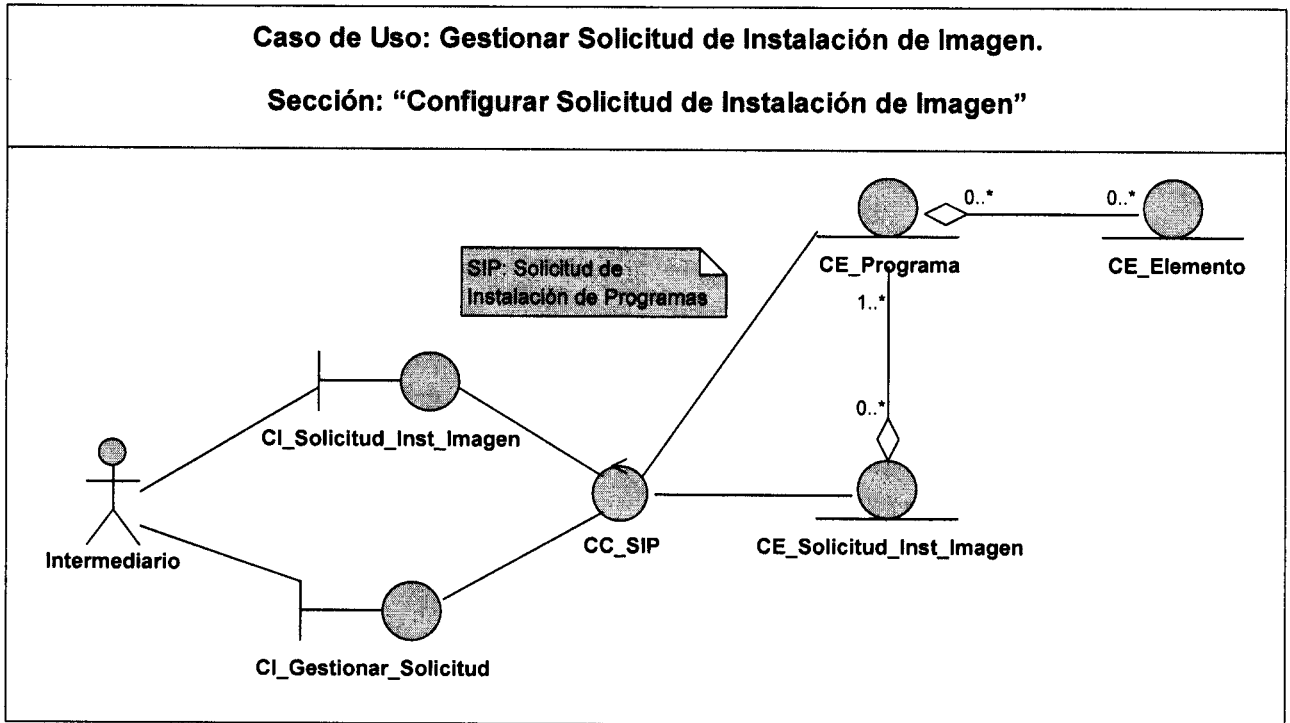
### **3.1 Introducción.**

En el desarrollo de este capítulo se representa la estructura de la solución planteada por el cliente mediante las clases del análisis, además se reflejan las clases del diseño de los casos de uso, donde se especifican las funcionalidades del sistema y las necesidades de cada una para obtener el resultado esperado, así como la secuencia en la que se realizan las acciones entre las clases, paquetes e interfaces representadas, y por último se muestra el modelo lógico y físico de datos.

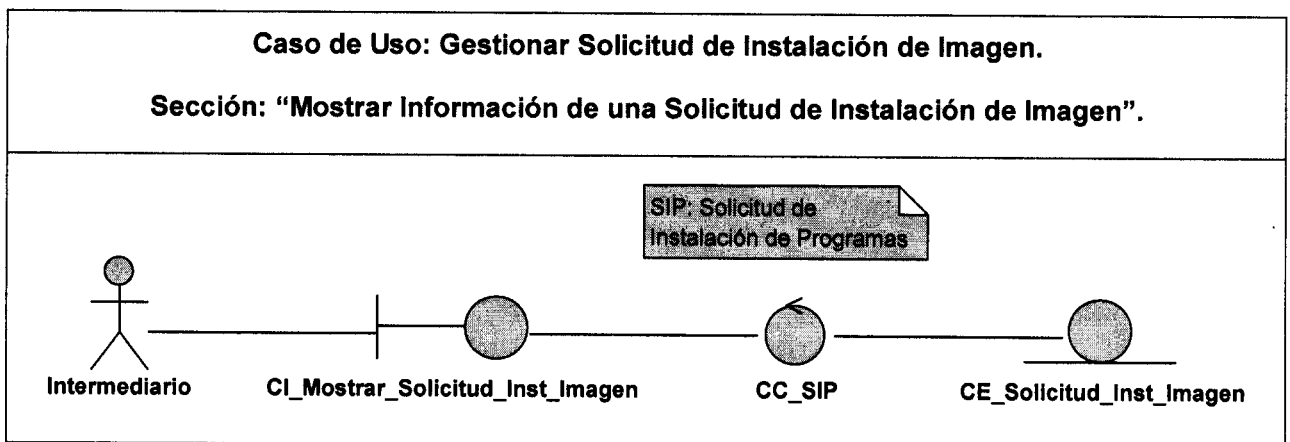
### **3.2 Análisis.**

El desarrollo de sistemas y aplicaciones figuran una secuencia de acciones que comienzan con la identificación de los procesos del negocio, sigue hacia una descripción de los objetivos, define las funcionalidades y finalmente se obtienen los requisitos que conducen hacia un modelo de análisis, el cual representa una abstracción de varias clases del diseño. Este modelo de clases se representa a continuación detalladamente por casos de uso.

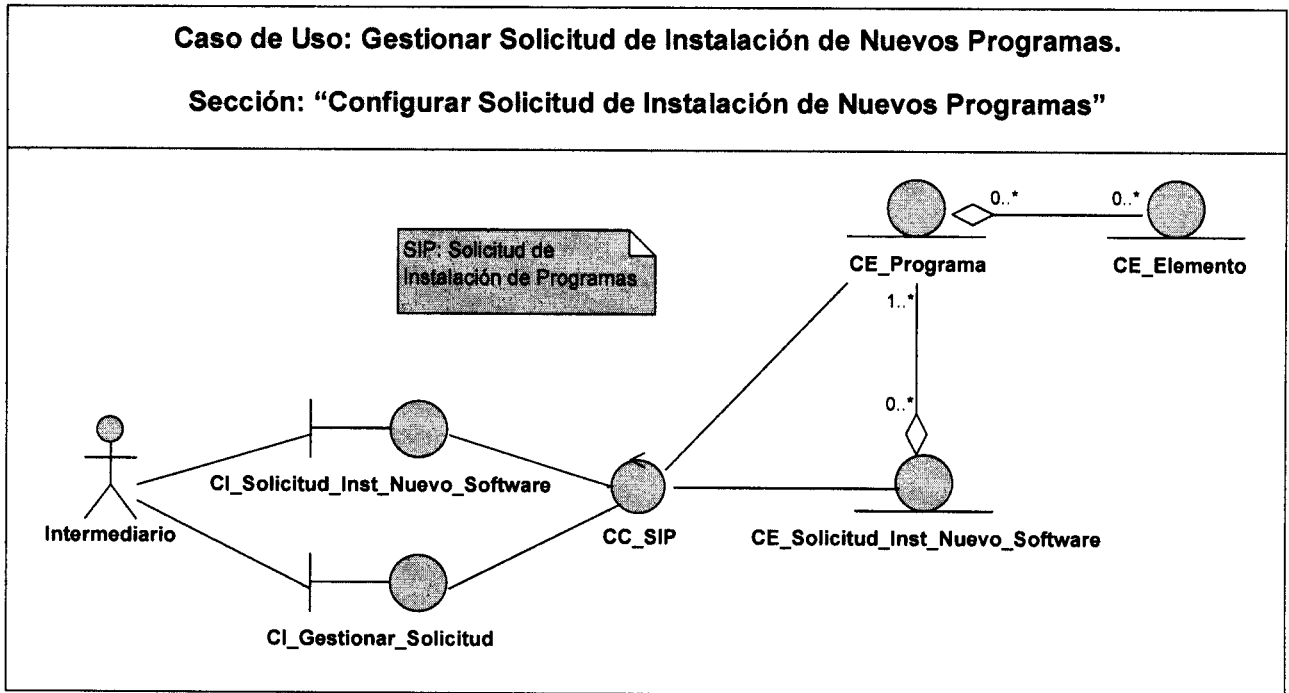
3.2.1 Modelo de Clases del Análisis.



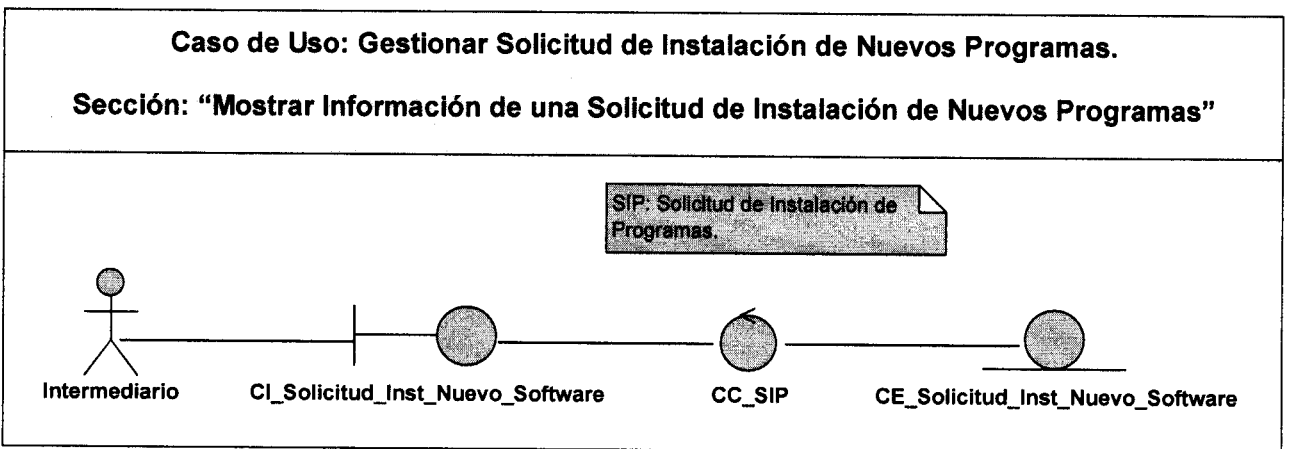
**Figura 3.1: Diagrama de Clases del Análisis: Gestionar Solicitud de Instalación de Imagen. Sección: "Configurar Solicitud de Instalación de Imagen"**



**Figura 3.2: Diagrama de Clases del Análisis: Gestionar Solicitud de Instalación de Imagen. Sección: "Mostrar Información de una Solicitud de Instalación de Imagen"**



**Figura 3.3: Diagrama de Clases del Análisis: Gestionar Solicitud de Instalación de Nuevos Programas. Sección: "Configurar Solicitud de Instalación de Nuevos Programas"**



**Figura 3.4: Diagrama de Clases del Análisis: Gestionar Solicitud de Instalación de Nuevos Programas. Sección: "Mostrar Información de una Solicitud de Instalación de Nuevos Programas"**

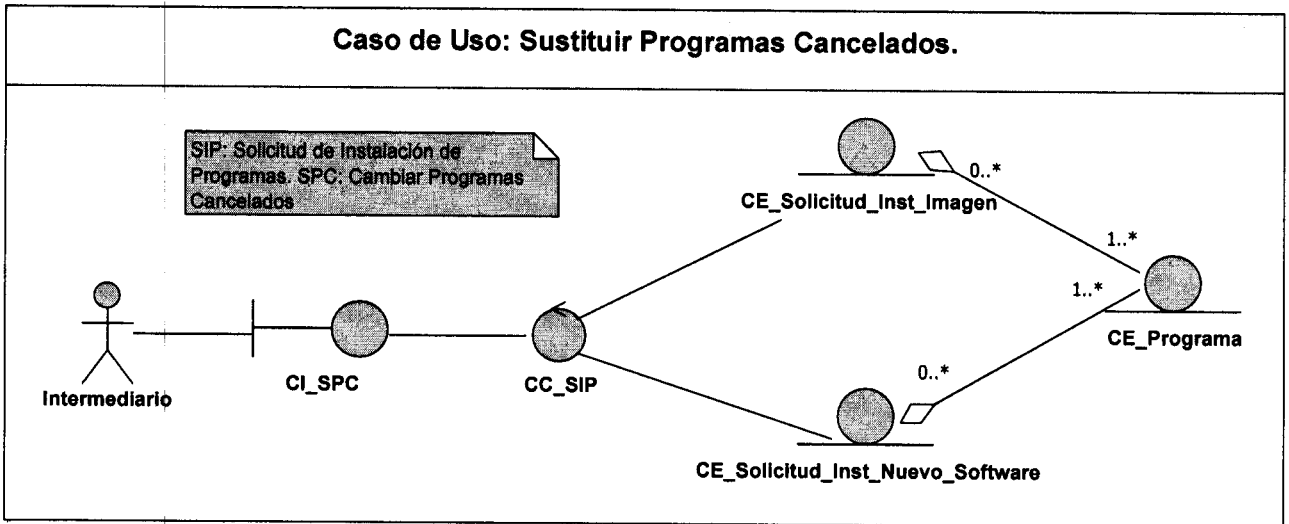


Figura 3.5: Diagrama de Clases del Análisis: "Sustituir Programas Cancelados"

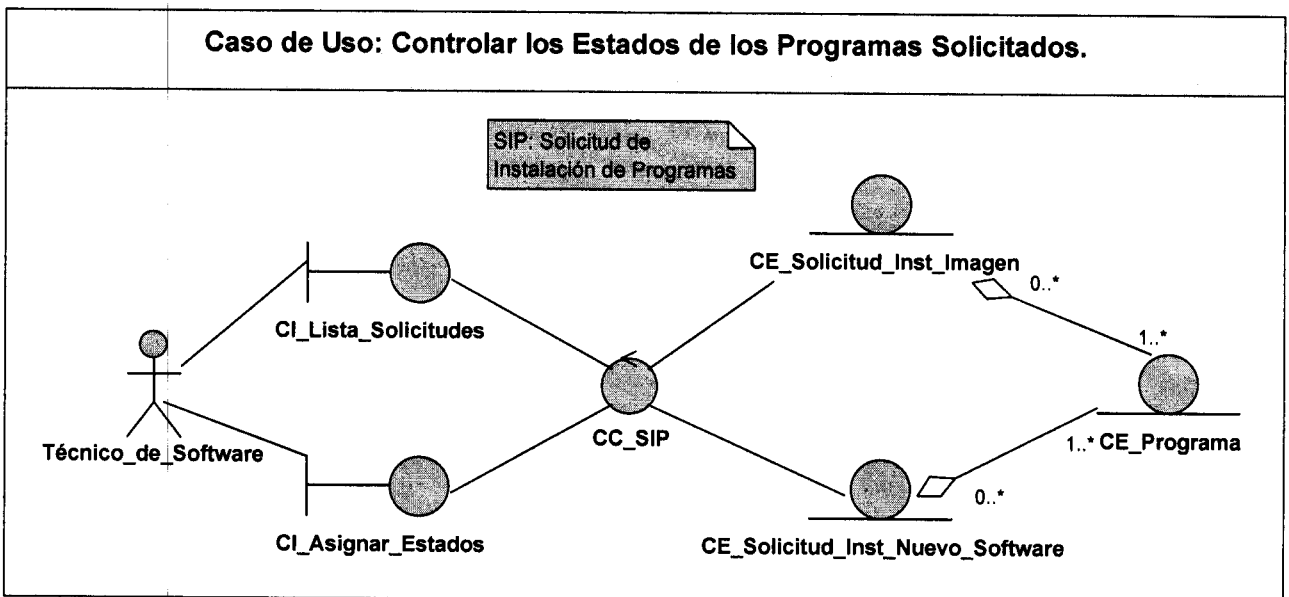
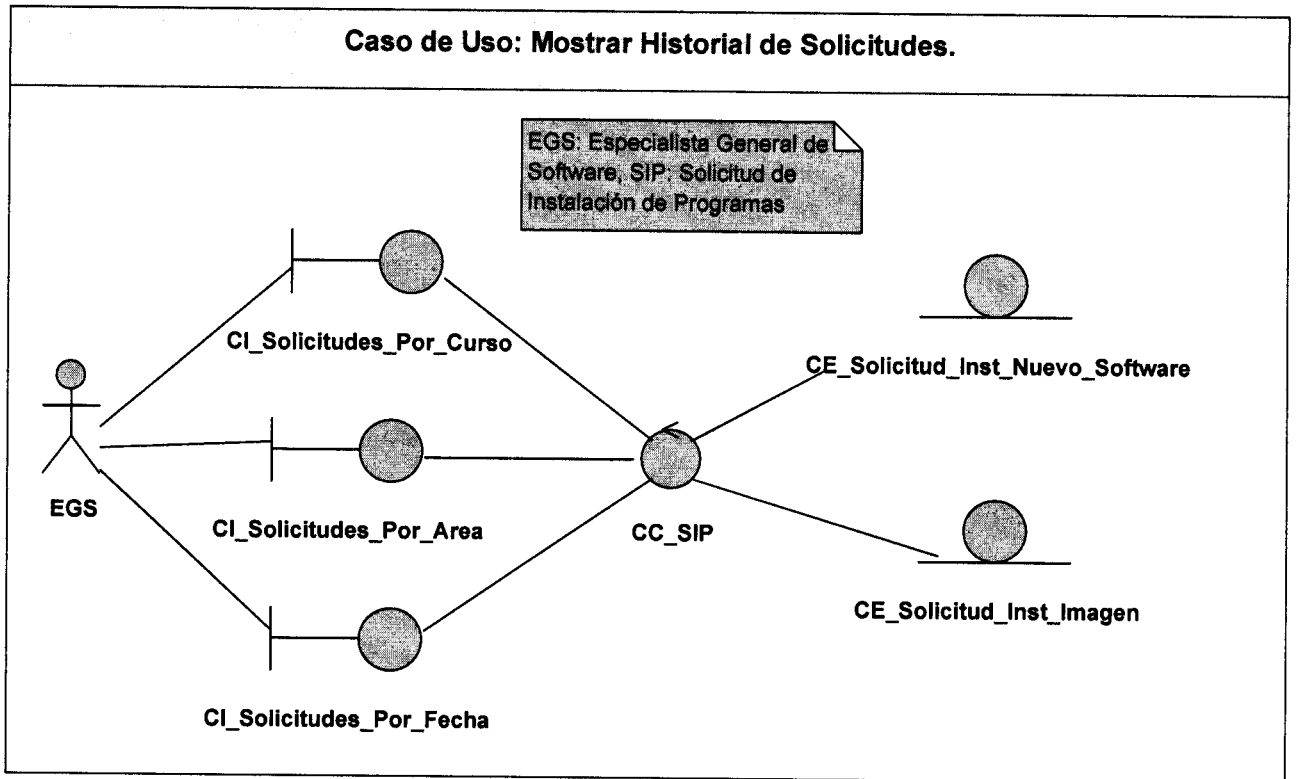
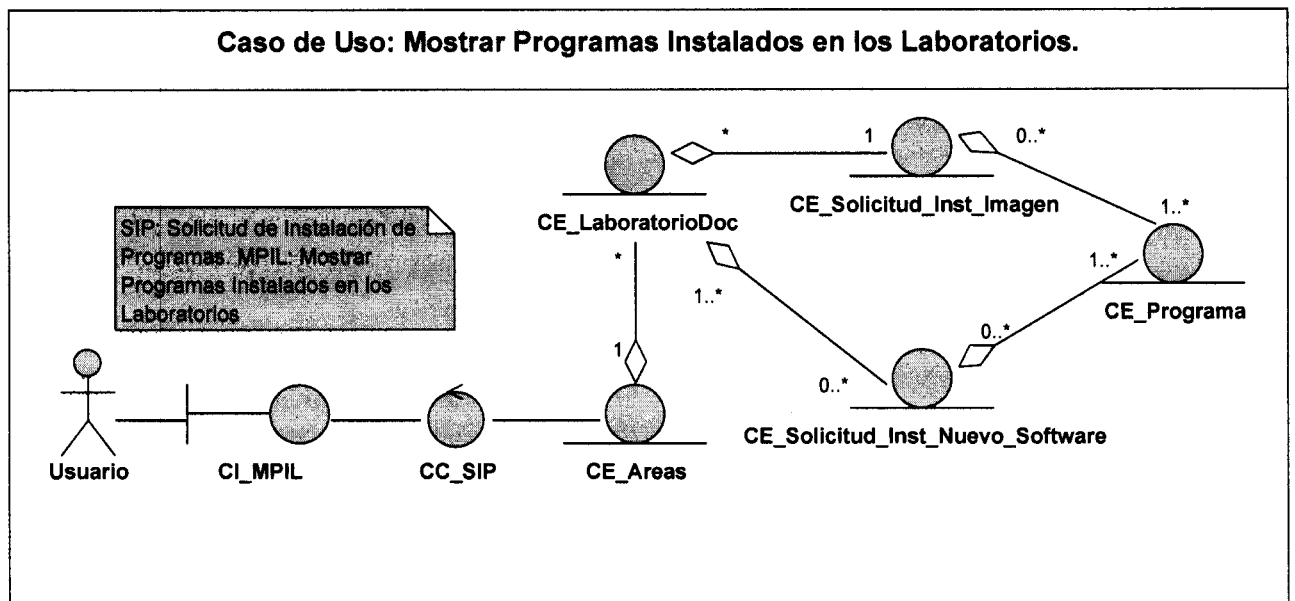


Figura 3.6: Diagrama de Clases del Análisis: "Controlar Estados de los Programas Solicitados"



**Figura 3.7: Diagrama de Clases del Análisis: "Mostrar Historial de Solicitudes"**



**Figura 3.8: Diagrama de Clases del Análisis: "Mostrar Programas Instalados en los Laboratorios"**

### 3.3 Diseño.

El modelo de diseño transforma el modelo de la información, creado en el análisis, en las estructuras de datos necesarias para implementar el sistema. Es el proceso de definición de la arquitectura (interfaces, clases, funcionalidades y datos) que satisface tanto, los requerimientos funcionales como los no funcionales. Este modelo se representa a continuación detalladamente por casos de uso.

#### 3.3.1 Diagrama de Clases del Diseño.

Debido a que las tecnologías de desarrollo Web han evolucionado considerablemente, al equipo del módulo: Control de Solicitudes de Software se le hizo difícil la tarea de representar los diagramas de clases del diseño utilizando los estereotipos Web estándares, por lo que se decidió representar el Framework (ExtJS) mediante un paquete que está conformado por los componentes JavaScript que construyen los elementos de los formularios.

En la tabla que se muestra a continuación se representan los accesos que realizan las clases de acceso a datos a las entidades, con el fin de esclarecer la relación query dirigida al paquete Dominio,

el cual contiene las clases persistentes representadas en la Figura 3.15: Diseño de Bases de Datos "Modelo Lógico de Datos"

Nombre del caso de uso	No. Figura	Clase Modelo	Entidades
Gestionar Solicitud de Instalación de Imagen. Sección: Configurar Solicitud de Instalación de Imagen.	3.8	SolicitudDAO	SolicitudSoftware
		ProgramaDAO	Programa
Gestionar Solicitud de Instalación de Imagen. Sección: Mostrar Información de una Solicitud de Instalación de Imagen.	3.9	SolicitudDAO	SolicitudSoftware
Gestionar Solicitud de Instalación de Nuevos Programas. Sección: Configurar Solicitud de Instalación de Nuevos Programas.	3.10	SolicitudDAO	SolicitudSoftware
		ProgramaDAO	Programa
		AuxiliarDAO	Facultad
Gestionar Solicitud de Instalación de Nuevos Programas. Sección: Mostrar Información de una Solicitud de Instalación de Nuevos Programas.	3.11	SolicitudDAO	SolicitudSoftware
Controlar Estados de los Programas Solicitados.	3.12	SolicitudDAO	SolicitudSoftware
		AuxiliarDAO	Area
Mostrar Historial de Solicitudes.	3.13	SolicitudDAO	SolicitudSoftware
		AuxiliarDAO	Area
			Curso



Mostrar Programas Instalados en los Laboratorios.	3.14	AuxiliarDAO	LaboratorioDocente
			Area
		SolicitudDAO	SolicitudSoftware
Sustituir Programas Cancelados	3.15	SolicitudDAO	SolicitudSoftware
		ProgramaDAO	Programa

**Tabla 3.1: Relaciones entre clases modelos y entidades.**

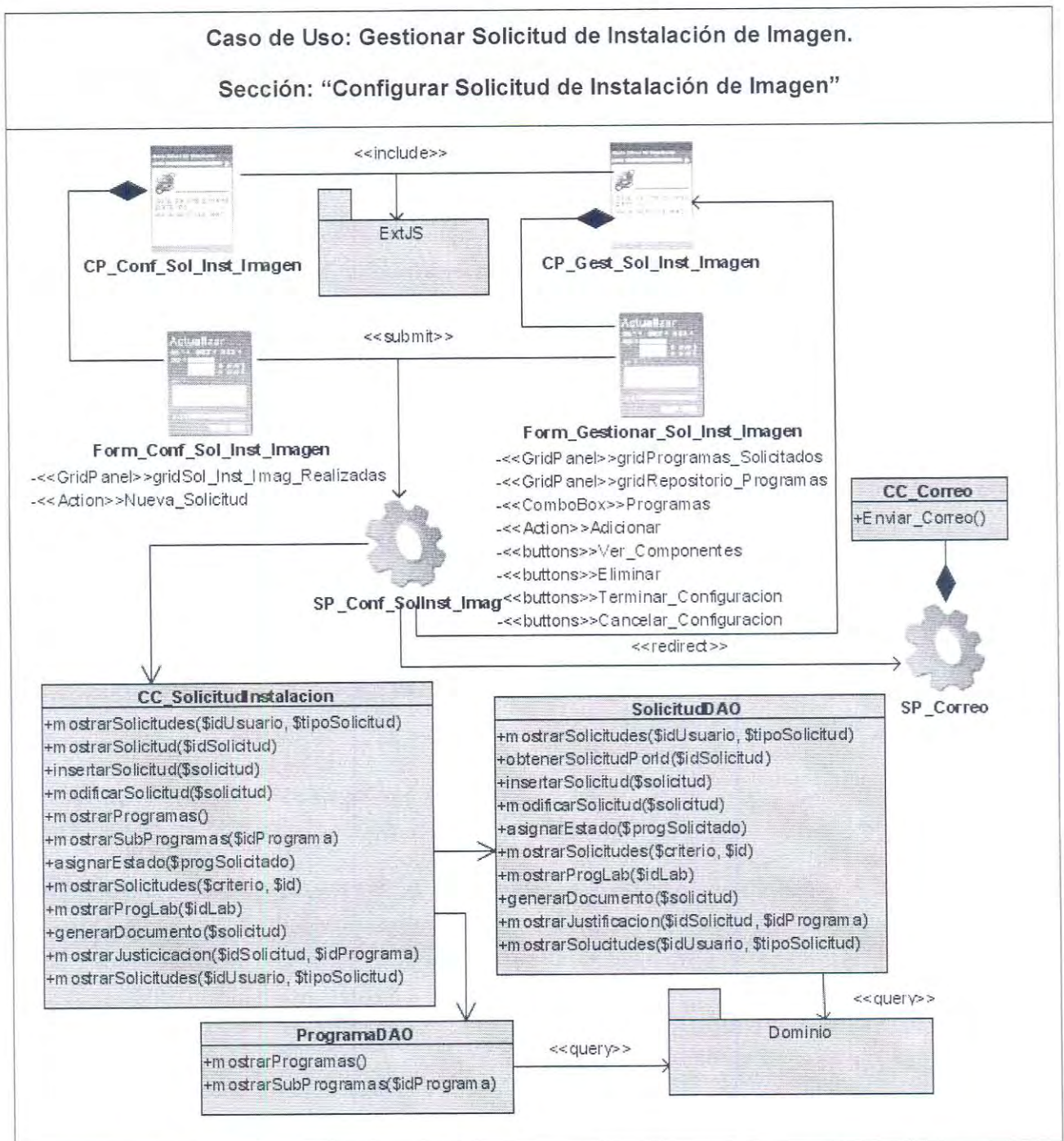


Figura 3.8: Diagrama de Clases del Diseño: Gestionar Solicitud de Instalación de Imagen.

Sección: "Configurar Solicitud de Instalación de Imagen"

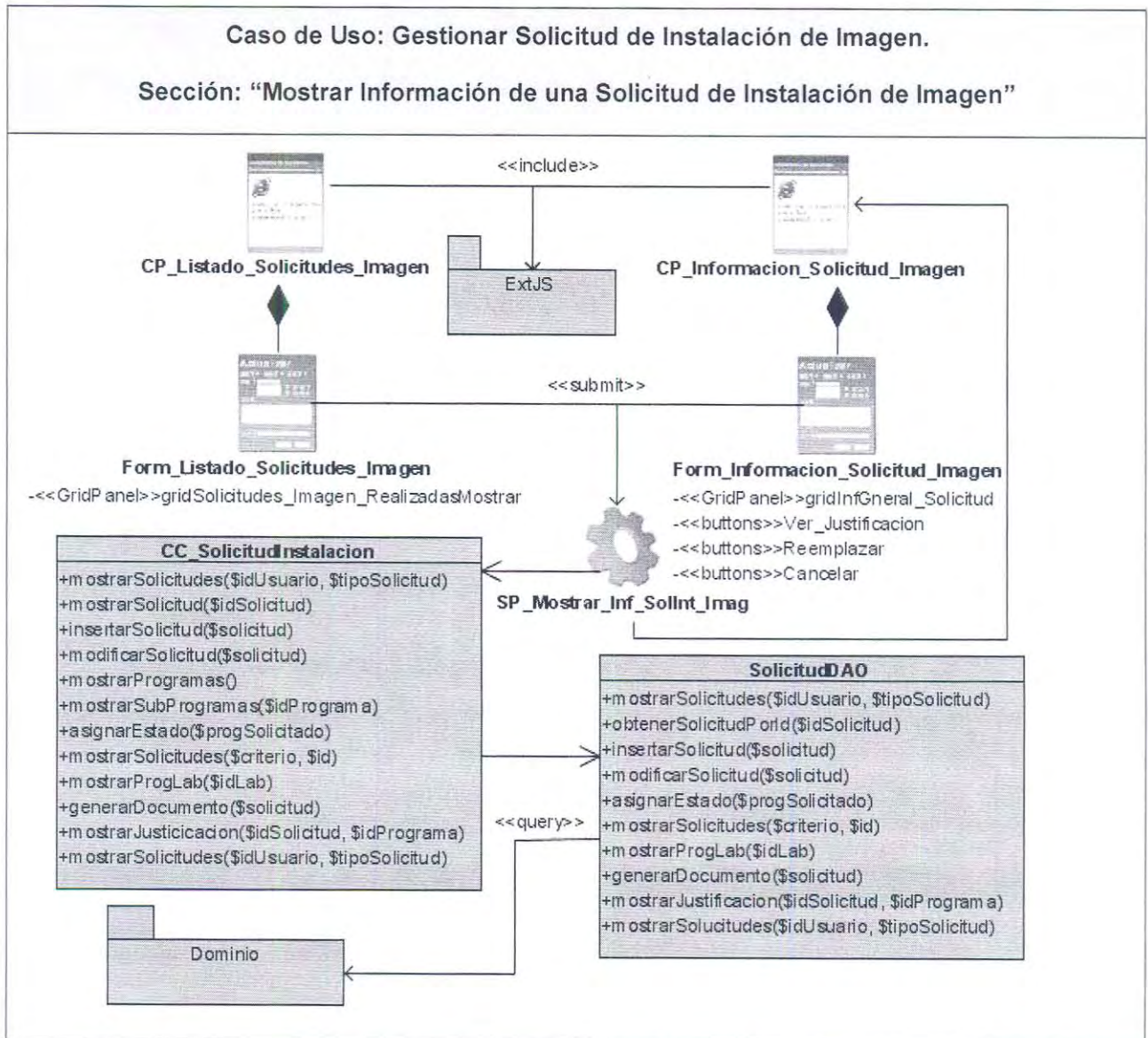
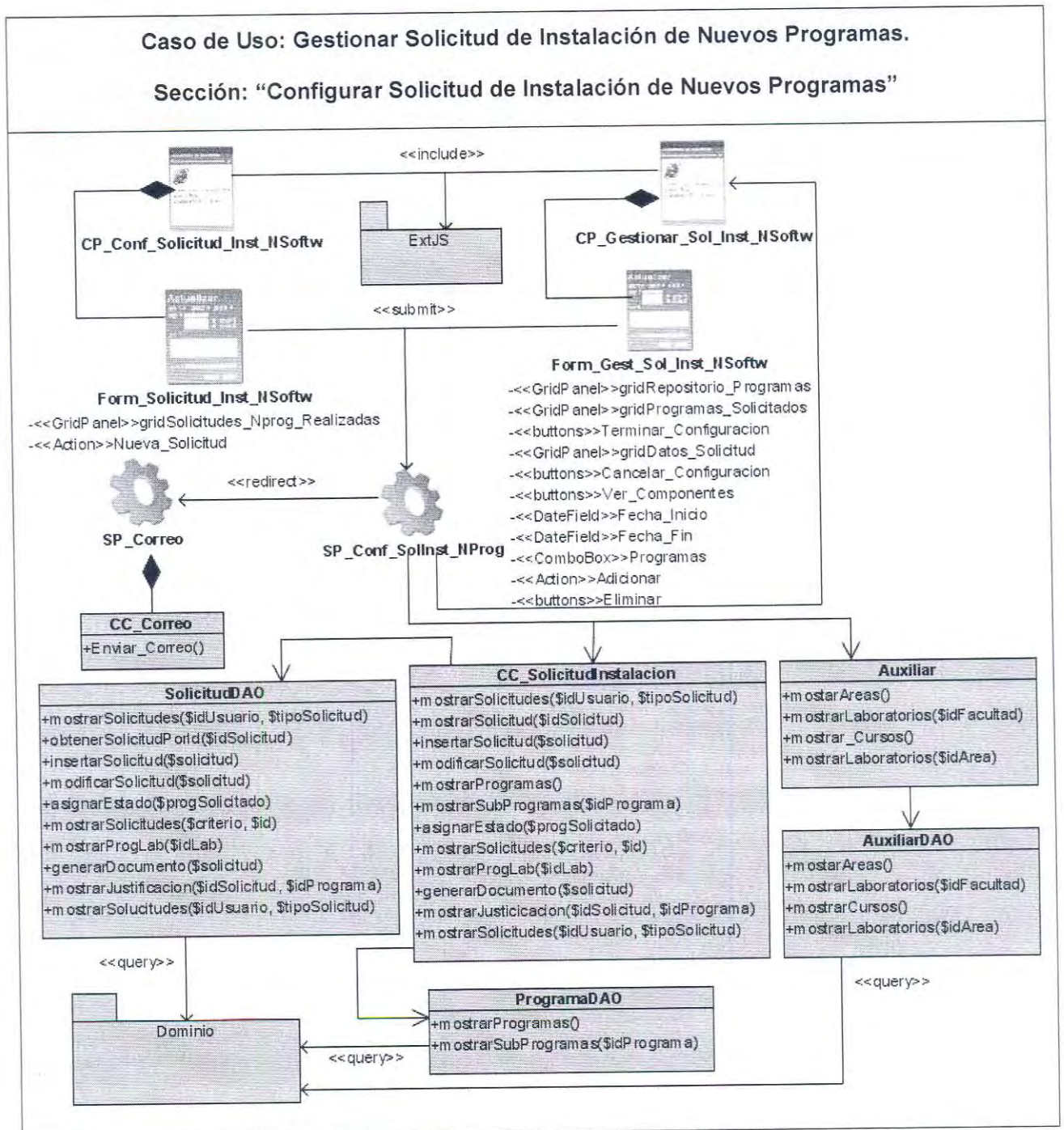


Figura 3.9: Diagrama de Clases del Diseño: Gestionar Solicitud de Instalación de Imagen.

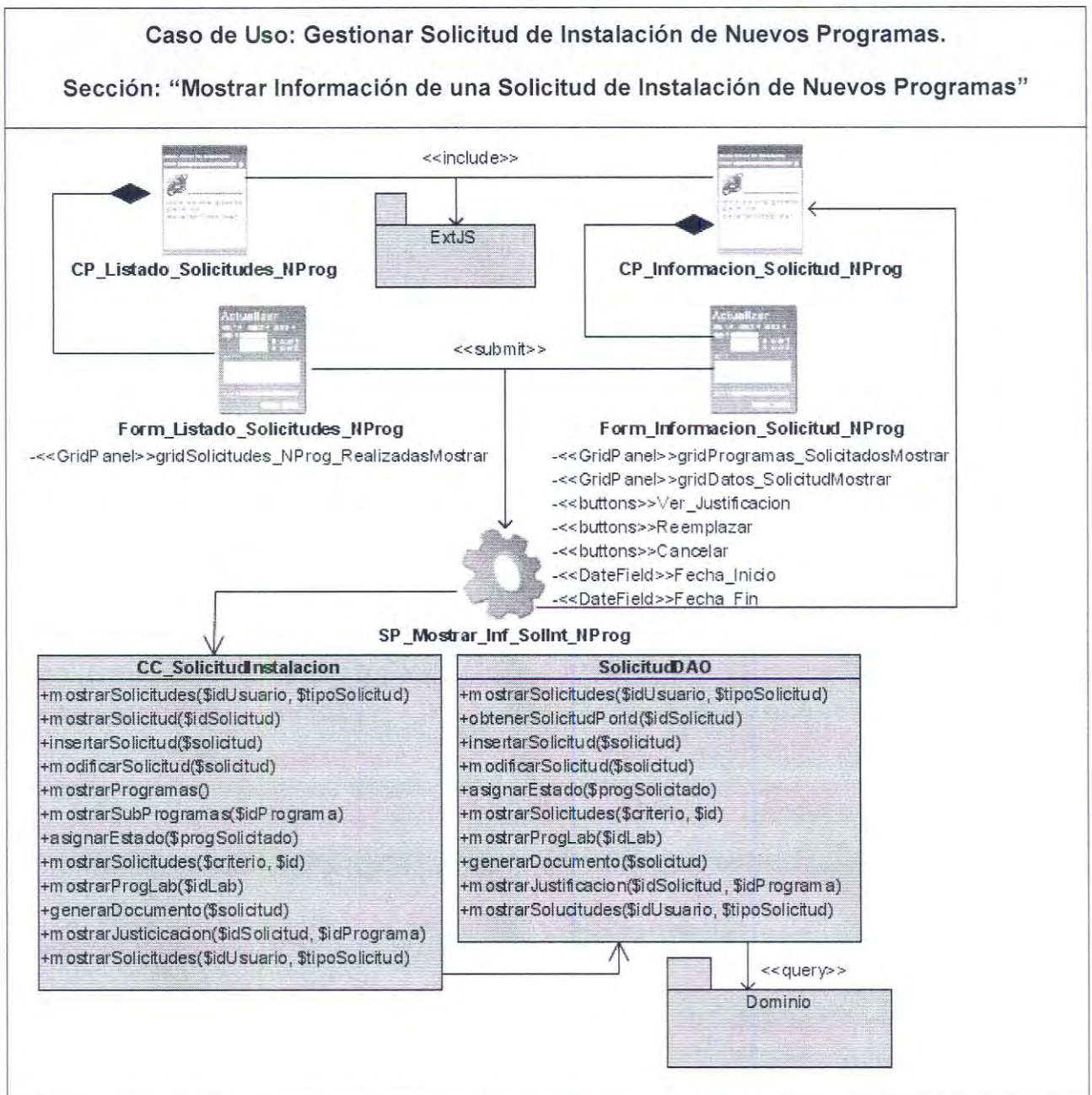
Sección: “Mostrar Información de una Solicitud de Instalación de Imagen”





**Figura 3.10: Diagrama de Clases del Diseño: Gestionar Solicitud de Instalación de Nuevos Programas. Sección: "Configurar Solicitud de Instalación de Nuevos Programas"**





**Figura 3.11: Diagrama de Clases del Diseño: Gestionar Solicitud de Instalación de Nuevos Programas. Sección: "Mostrar Información de una Solicitud de Instalación de Nuevos Programas"**

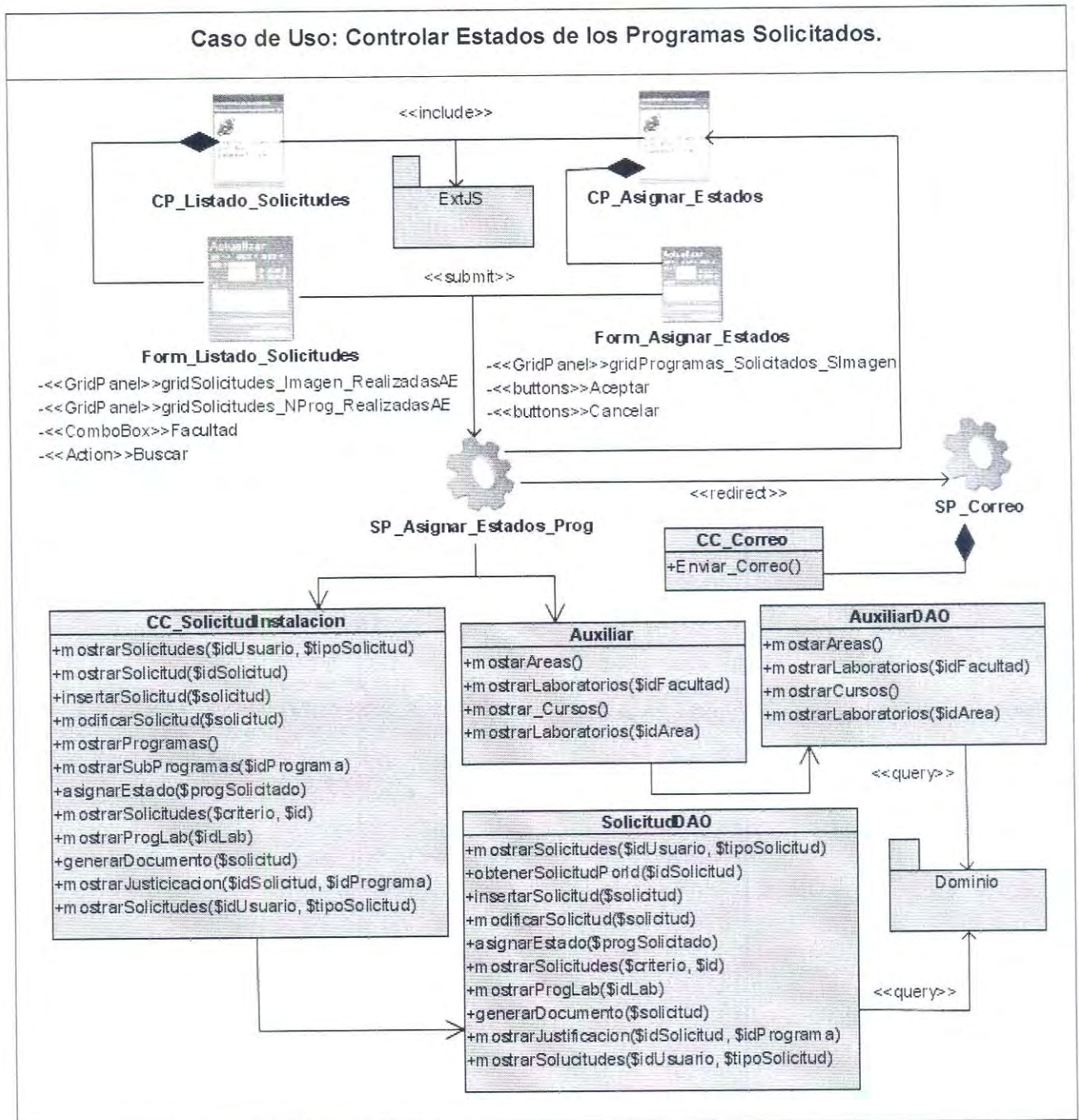


Figura 3.12: Diagrama de Clases del Diseño: "Controlar Estados de los Programas Solicitados"



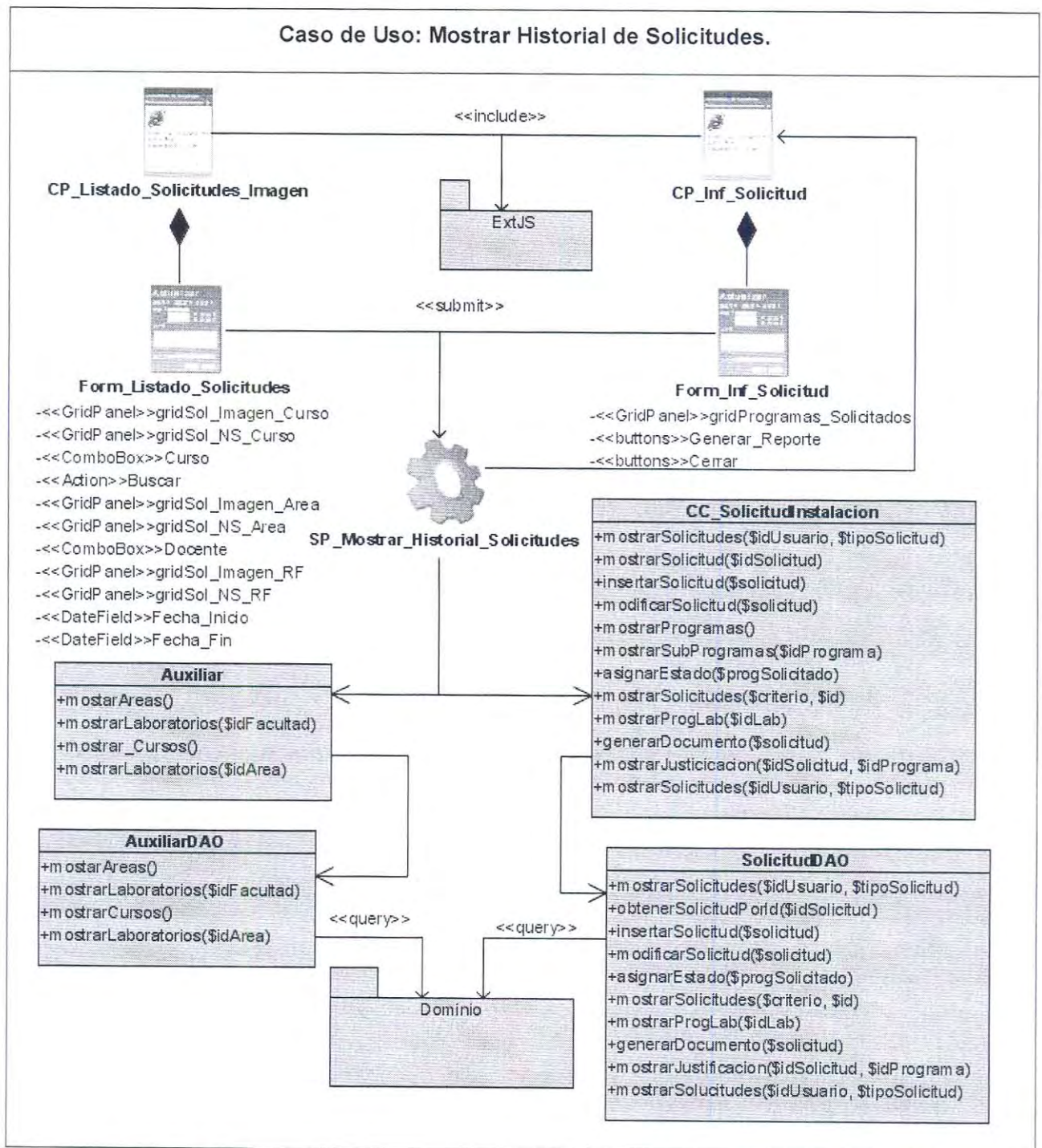


Figura 3.13: Diagrama de Clases del Diseño: "Mostrar Historial de Solicitudes"

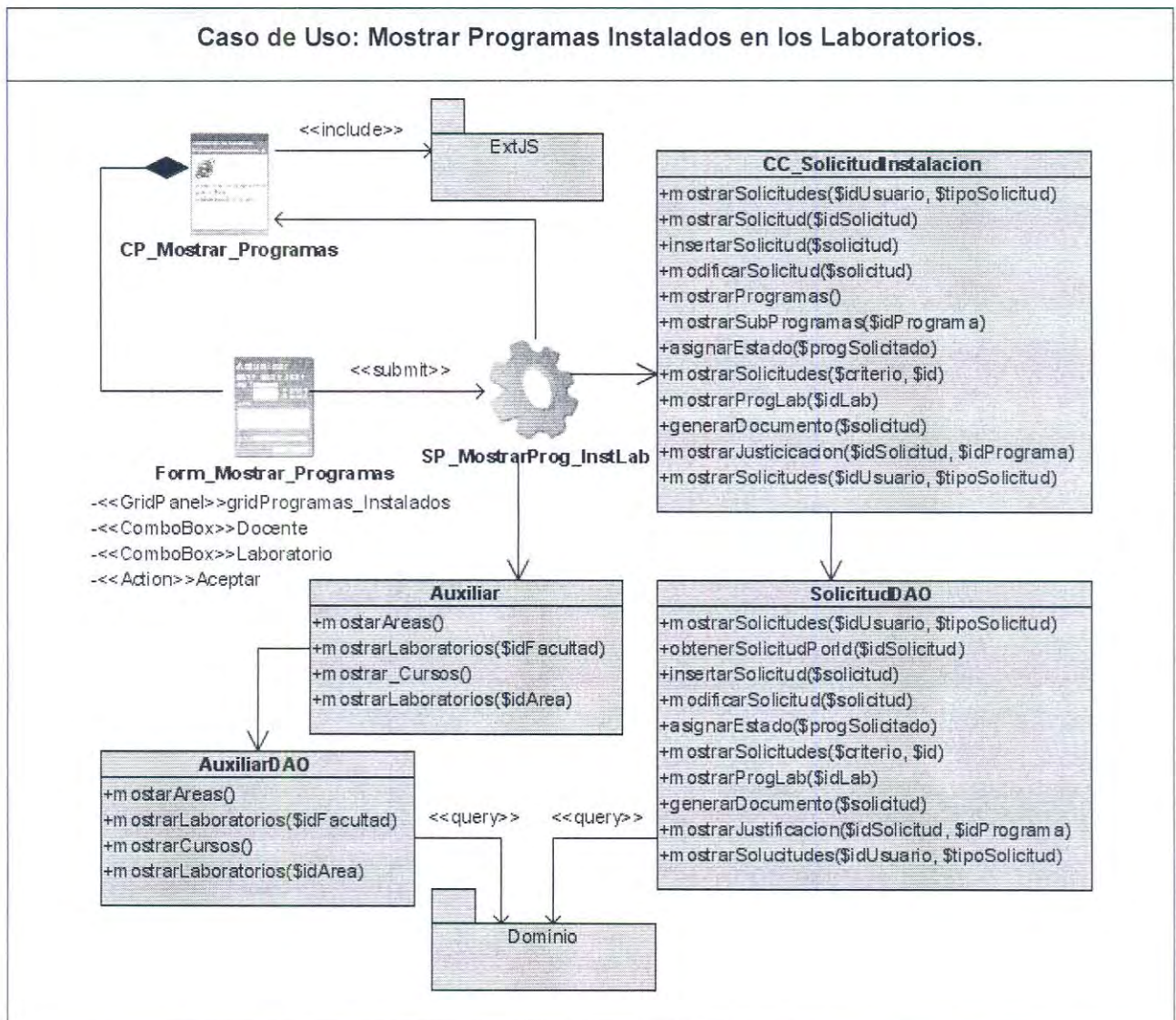


Figura 3.14: Diagrama de Clases del Diseño: "Mostrar Programas Instalados en los Laboratorios"



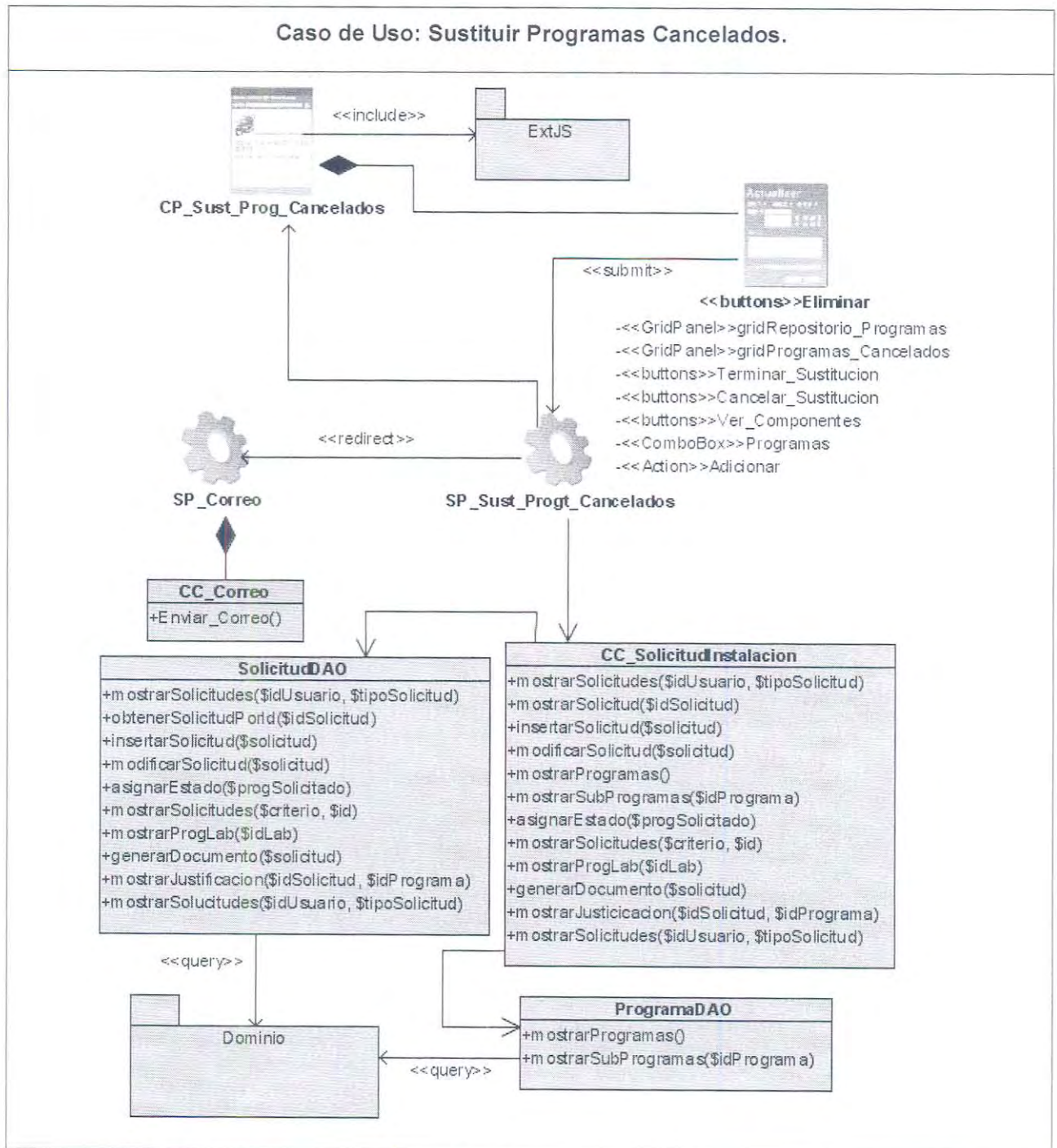


Figura 3.15: Diagrama de Clases del Diseño: "Sustituir Programas Cancelados"

### 3.3.2 Diagramas de Interacción.

Esta descripción se encuentra en los anexos (Anexo No 2), sólo se encuentran los diagramas de los casos de uso críticos.

## 3.1 Diseño de la Base de Datos.

### 3.1.1 Modelo Lógico de Datos.

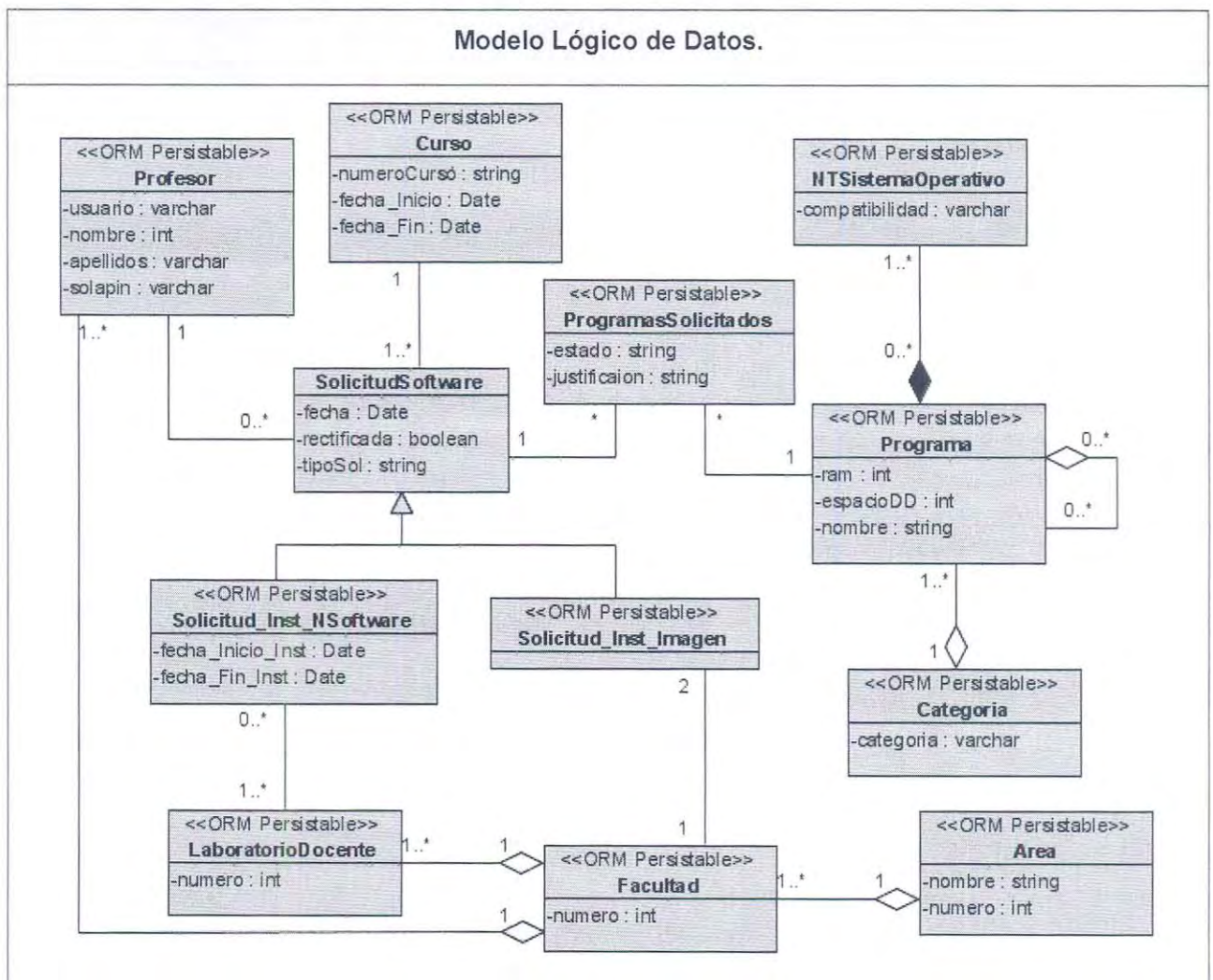


Figura 3.16: Diseño de Bases de Datos "Modelo Lógico de Datos"



3.1.2 Modelo Físico de Datos.

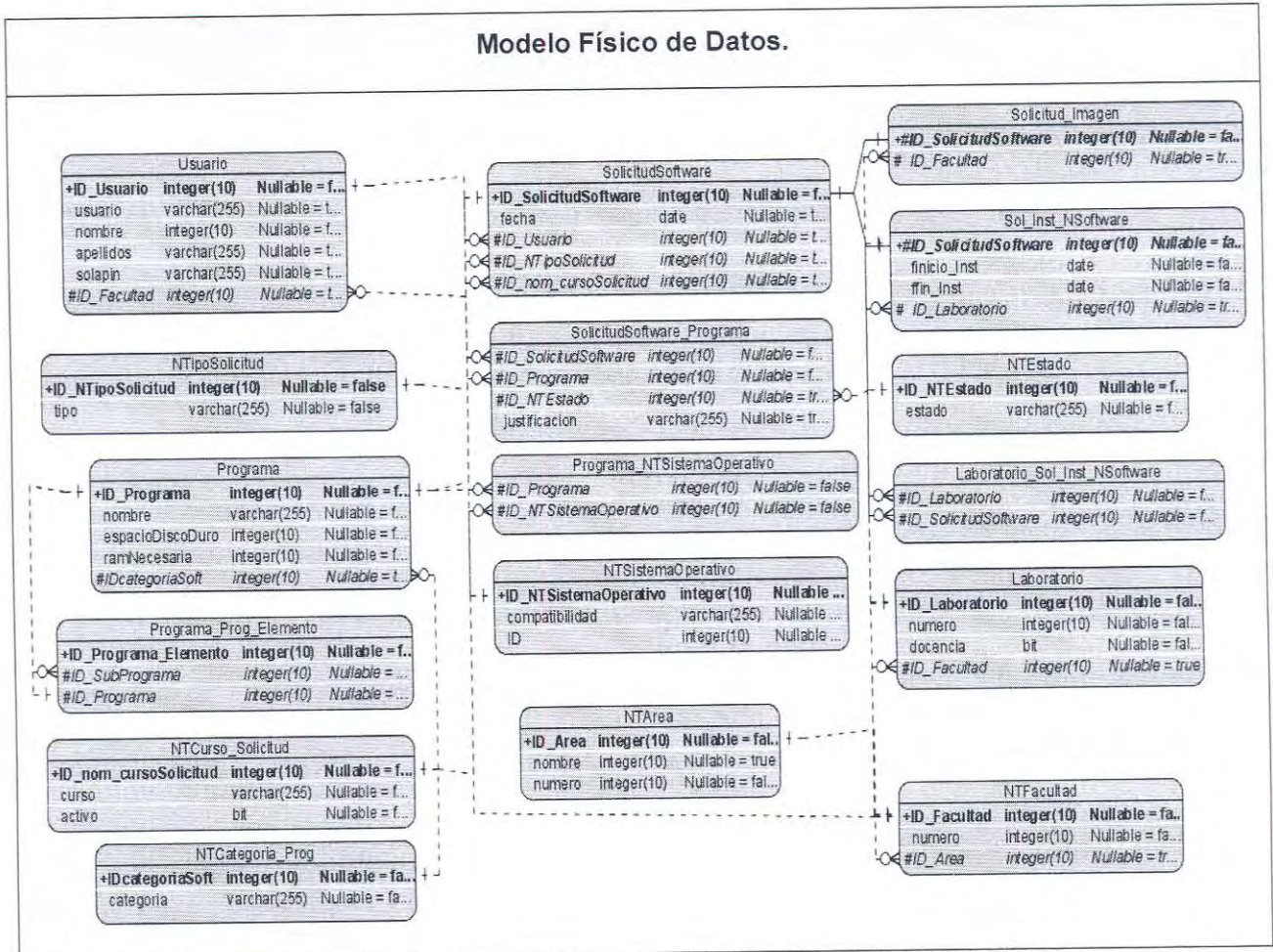


Figura 3.17: Diseño de Bases de Datos "Modelo Físico de Datos"

### **3.2 Conclusiones.**

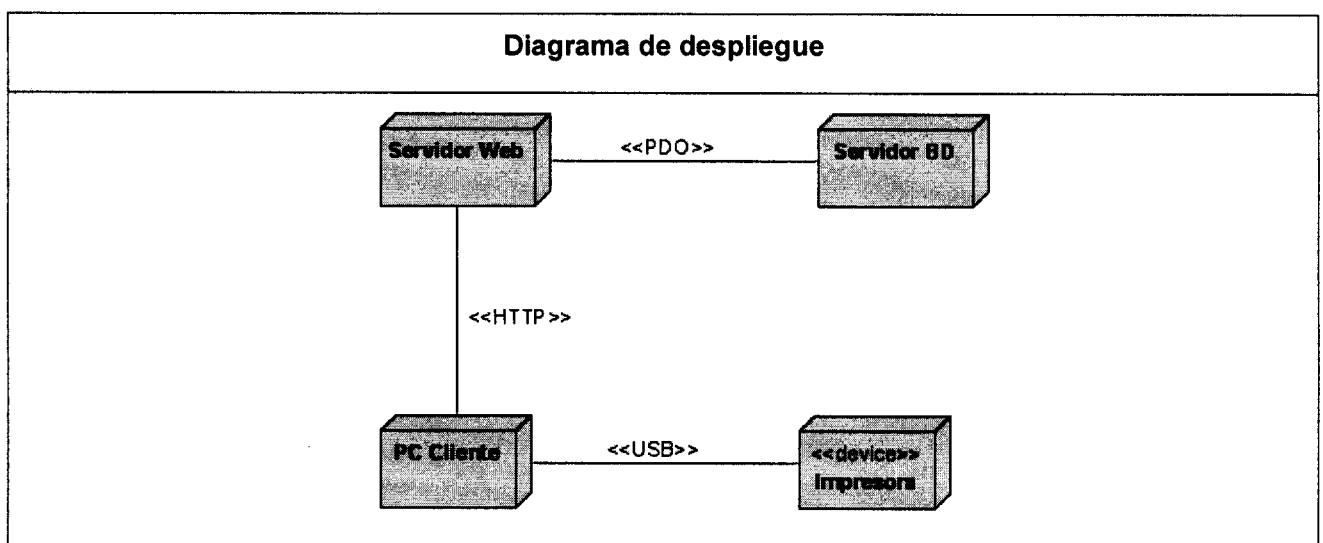
En el desarrollo del capítulo se representó el modelo de análisis, el cual constituye una visión del sistema sobre las necesidades planteadas por el cliente. Además, se confeccionó la estructura de la aplicación mediante las clases del diseño, de las cuales se realizaron descripciones secuenciales de sus funcionalidades y por último se mostraron los modelos lógico y físico de datos.

## CAPÍTULO 4: IMPLEMENTACIÓN.

### 4.1 Introducción.

En este capítulo se modelan los diagramas de la etapa de implementación (componentes y despliegue). El propósito del mismo es dejar listo un producto software en su versión operativa inicial que cumpla con los requerimientos tanto funcionales como no funcionales y posea la calidad, robustez, portabilidad, y flexibilidad adecuada para su aplicación.

### 4.2 Diagrama de despliegue.



*Figura 4.1: "Diagrama de Despliegue"*

### 4.3 Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas y reflejar las relaciones que existen entre los elementos de implementación. Entre los objetivos que se persiguen con estos diagramas están: mostrar la estructura del modelo de implementación, los componentes software que constituyen una parte reusable y las dependencias entre los ficheros de código en tiempo de compilación.

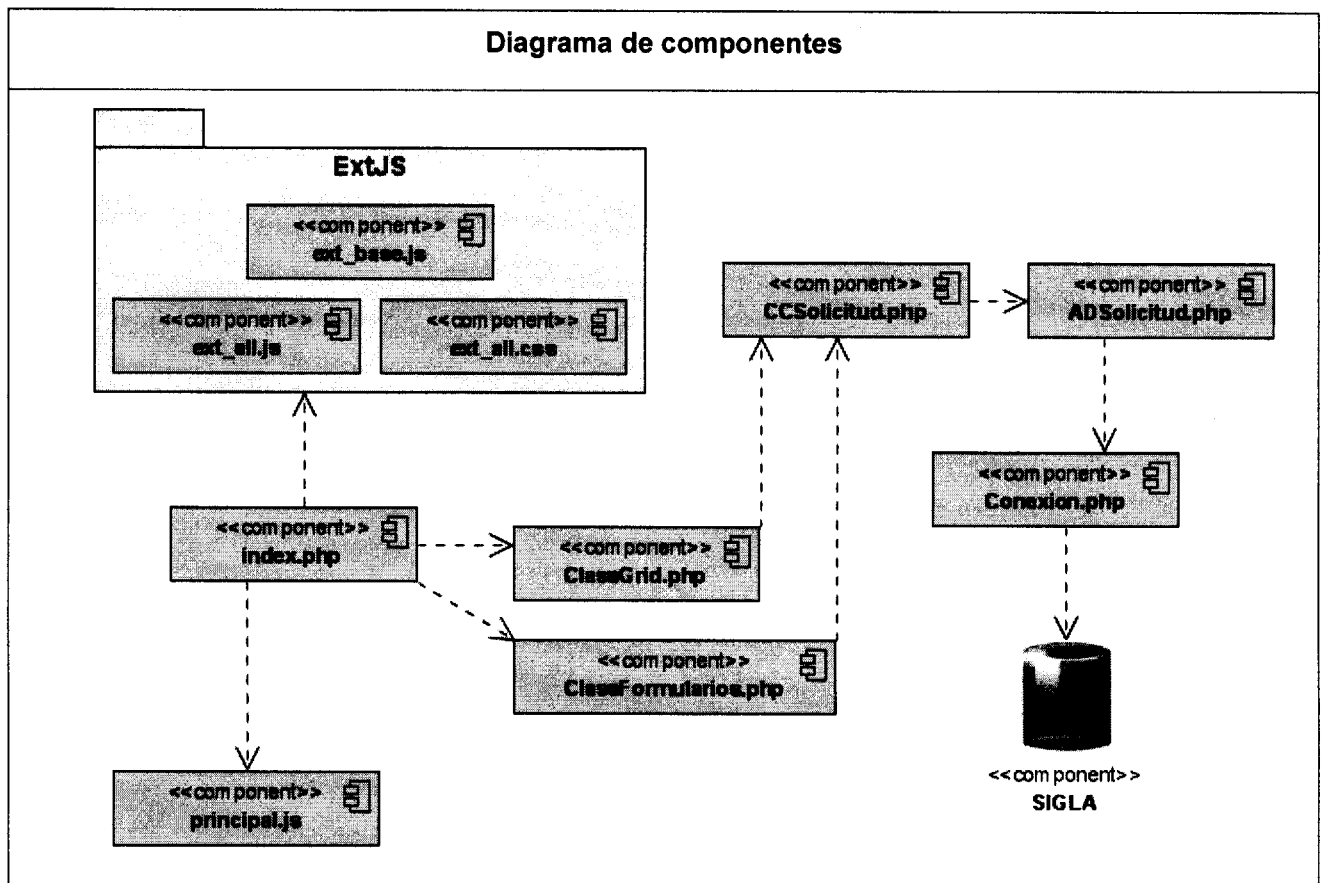


Figura 4.2: "Diagrama de componentes"

#### 4.4 Conclusiones

En el desarrollo del capítulo se representó el sistema a través de los diagramas de componentes y de despliegue para lograr una mejor comprensión del módulo y establecer el fundamento de la aplicación; cumpliendo así, con lo requerido y necesitado por el cliente.

## **CAPÍTULO 5: ESTUDIO DE LA FACTIBILIDAD.**

### **5.1- Introducción.**

En este capítulo se detallan los cálculos para obtener la estimación del costo, esfuerzo y tiempo que demora el desarrollo de la aplicación. Además, se reflejan los beneficios dirigidos a controlar las solicitudes de instalación de programas y el análisis, según los resultados obtenidos, de la factibilidad del sistema.

### **5.2- Planificación.**

El objetivo de la planificación de un proyecto software es proporcionar un marco de trabajo que permita hacer estimaciones razonables de recursos, costos y planificación temporal. Estas apreciaciones se realizan mediante un proceso de estudio y análisis minucioso de la información que se posee para poder determinar correctamente la decisión adecuada y más factible.

#### **5.2.1 Puntos de Casos de Uso.**

La estimación mediante el análisis de Puntos de Casos de Uso es un método que se utiliza para evaluar el esfuerzo de un proyecto de desarrollo de software a partir de los casos de uso y actores; se manejan además, factores de entorno y de complejidad técnica, a los cuales se les asignan determinados pesos para estimar el tiempo y costo del proyecto.

##### **5.2.1.1 Cálculo de Puntos de Casos de Uso sin ajustar.**

Los puntos de casos de uso sin ajustar se calculan mediante la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Para calcular el Factor de Peso de los Actores sin Ajustar (UAW) se utilizan los criterios y las expresiones representadas en la siguiente tabla:

Tipo	Descripción	Peso	Cant*pe- so
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	4*3
<b>Total</b>			<b>12</b>

**Tabla 5.1: Peso de los Actores sin ajustar.**

**Resultado:** El Factor de Peso de los Actores sin Ajustar (UAW) es 12.

El Factor de Peso de los Casos de Uso sin Ajustar (UUCW) se calcula mediante la clasificación de los casos de uso, la cual se obtiene en dependencia de las transacciones (pares de pasos acción-usuario > respuesta-sistema de los escenarios de los casos de uso) que realicen, a cada tipo se le asigna un peso y luego se multiplican la cantidad de casos de uso de ese tipo por el peso establecido y se suman los resultados, tal como se refleja en la siguiente tabla.

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	2*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	4*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	0*15
<b>Total</b>			<b>50</b>

**Tabla 5.2: Peso de los Casos de Uso sin ajustar.**

**Resultado:** El Factor de Peso de los Casos de Uso sin Ajustar tiene un valor de 50.



Quedando el valor de los Puntos de Casos de Uso sin Ajustar como se muestra a continuación:

$$UUCP = UAW + UUCW$$

$$UUCP = 12 + 50$$

$$UUCP = 62$$

**5.2.1.2 Cálculo de Puntos de Casos de Uso Ajustados.**

Cuando se obtiene el valor de los Puntos de Casos de Uso sin Ajustar, se debe ajustar mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

El Factor de Complejidad Técnica (TCF) se calcula utilizando la siguiente ecuación:

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

De la cual, el término de la sumatoria se calcula asignándole a los criterios descritos en la siguiente tabla valores entre 0 y 5 (donde el cero representa un aporte irrelevante y el 5 uno relevante); luego se toman estos valores y se multiplican por los pesos correspondientes, y la suma de estos cálculos dan como resultado el término requerido.

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	1	1
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8

T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	2
<b>Total</b>				<b>46.5</b>

**Tabla 5.3: Factor de Complejidad Técnica.**

Utilizando los resultados de los cálculos realizados se obtiene que:

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$TCF = 0.6 + 0.01 * 46.5$$

$$TCF = 0.6 + 0.465$$

$$TCF = 1.065$$

$$TCF = 1.07$$

**Resultado:** El Factor de Complejidad Técnica posee un valor de 1.07.

El cálculo del Factor de Ambiente (EF) se realiza mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

Donde el término de la sumatoria se calcula de manera similar al del Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	1	4.5

E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	6
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3
<b>Total</b>				<b>15.5</b>

**Tabla 5.4: Factor de Ambiente.**

Utilizando los resultados de los cálculos realizados se obtiene que:

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor})$$

$$EF = 1.4 - 0.03 * 15.5$$

$$EF = 1.4 - 0.465$$

$$EF = 0.935$$

$$EF = 0.935$$

$$EF = 0.94$$

**Resultado:** El Factor de Ambiente posee un valor de 0.94.

Quedando el valor de los Puntos de Casos de Uso Ajustados como se muestra a continuación:

$$UCP = UUCP * TCF * EF$$

$$UCP = 62 * 1.07 * 0.94$$

$$UCP = 62.36$$

### 5.2.1.3 Estimación de esfuerzo a través de los puntos de casos de uso.

#### Para la fase de implementación

El valor del esfuerzo se calcula utilizando la siguiente ecuación:

$$E = UCP \times CF$$

Mediante este método se obtiene la estimación del esfuerzo en horas-hombre. Donde el término Factor de Conversión (CF) se obtiene utilizando los siguientes criterios:

$$CF = 20 \text{ horas-hombre (si Total}_{EF} \leq 2)$$

$$CF = 28 \text{ horas-hombre (si Total}_{EF} = 3 \text{ ó Total}_{EF} = 4)$$

$$CF = \text{abandonar o cambiar proyecto (si Total}_{EF} \geq 5)$$

$$\text{Total}_{EF} = \text{Cant EF} < 3 \text{ (entre E1 –E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)}$$

Siguiendo estos criterios se obtiene que:

$$\text{Total}_{EF} = \text{Cant EF} < 3 \text{ (entre E1 –E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)}$$

$$\text{Total}_{EF} = 2 + 0$$

$$\text{Total}_{EF} = 2$$

**Resultado:** El Factor de Conversión tiene un valor de 20 horas-hombre.

Quedando el cálculo del valor del esfuerzo como se muestra a continuación:

$$E = UCP \times CF$$

$$E = 62.36 * 20 \text{ horas-hombre}$$

$$E = 1247 \text{ horas-hombre}$$

**5.2.1.4 Calcular esfuerzo del proyecto.**

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	312 horas-hombre
Diseño	20%	624 horas-hombre
Implementación	40%	1247 horas-hombre
Prueba	15%	468 horas-hombre
Sobrecarga	15%	468 horas-hombre
<b>Total</b>	<b>100%</b>	<b>3119 horas-hombre</b>

**Tabla 5.5: Esfuerzo del proyecto.**

**ET = 3119 horas-hombre**, y se estima que cada mes tiene como promedio 240 horas laborables, dando como resultado un  $E_T = 13$  mes-hombre.

**Resultado:** Se estima que una persona podrá desarrollar las funcionalidades del módulo en un tiempo aproximado de 13 meses.

**Costo del proyecto.**

Se toma como salario promedio mensual: \$125.00; ya que el módulo está integrado por dos individuos, de los cuales uno recibe mensualmente un salario de \$150, y el segundo de \$100.

Para calcular el costo se manejan los siguientes términos:

**CH:** Cantidad de hombres.

**Tiempo:** Tiempo total de desarrollo del módulo.

**CHM:** Costo por Hombre - Mes.

**Resultado:** La cantidad de hombres que integran el módulo son 2.

El Costo por Hombre - Mes se calcula mediante la ecuación:

$$\text{CHM} = 2 * \text{Salario Promedio}$$

$$\text{CHM} = 250.00 \text{ \$/mes}$$

Quedando el cálculo del costo y el tiempo como se muestra a continuación:

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

$$\text{Costo} = 250.00 * 13 / 2$$

$$\text{Costo} = \$1625.$$

$$\text{Tiempo} = \text{ET} / \text{CH}$$

$$\text{Tiempo} = 13 / 2$$

$$\text{Tiempo} = 6.5$$

De los resultados obtenidos se estima que con dos hombres trabajando en el módulo, el mismo se desarrolle en un tiempo aproximado de 7 meses, con un costo total estimado en \$1625.

### **5.5 Beneficios tangibles e intangibles.**

El módulo Control de Solicitudes de Software tiene la tarea de resolver las trabas existentes en el control de solicitudes de instalación de programas en los laboratorios docentes de la universidad, y como automatizará una solución a estos problemas internos, no brindará ningún aporte económico; pero sí proporcionará significativos beneficios intangibles, además de ofrecer una aplicación Web flexible, sugerente, amena y fácil de usar.

Estos beneficios intangibles serán:

- Ahorro del material de oficina.
- Control adecuado de las peticiones de programas.
- Eficiencia y calidad en la instalación de programas.

- Disminución de errores en las imágenes desarrolladas.
- Aprovechamiento fructífero de la jornada laboral por parte del personal involucrado en las actividades concernientes a las solicitudes e instalación de programas en los laboratorios.
- Anulación de los atrasos en el proceso docente-educativo.

### **5.6 Análisis de costo.**

El desarrollo de una aplicación que controle las solicitudes de instalación de programas tiene un costo asociado, y como la misma no proporcionará bienes monetarios el análisis para llegar a la conclusión de su factibilidad debe estar centrado en las facilidades que obtendrá el personal encargado de realizar las tareas de pedidos e instalación de programas, a las mejoras que aportará al proceso docente educativo y a que no incurrirá en gastos de compras de licencias; ya que las herramientas y tecnologías que se van a utilizar en su construcción son libres. Por tanto, luego de realizar los cálculos y estudios pertinentes, se puede arribar sin ningún riesgo a la conclusión de que su desarrollo es plenamente factible.

### **5.7 Conclusiones.**

En este capítulo se realizó el estudio de la factibilidad del módulo: "Control de Solicitudes de Software", basándose en los cálculos relativos a la estimación del costo, esfuerzo y el tiempo de desarrollo del sistema y también, de los beneficios que aportará su implementación a los procesos de solicitud de instalación de programas en los laboratorios docentes de la Universidad de las Ciencias Informáticas.

### **CONCLUSIONES.**

Después de realizada la investigación se puede concluir que las tareas de investigación propuestas y el objetivo general del trabajo se cumplieron cabalmente, obteniéndose por tanto los siguientes resultados:

- Se realizó un estudio de las herramientas y tecnologías de desarrollo que se utilizaron en la confección de la solución.
- Se analizó el Módulo de Servicio: Gestión de Solicitudes de Modificación de Imágenes de SO en los laboratorios docentes y clonado de PC.
- Se realizó una descripción de los procesos actuales relacionados con los pedidos de instalación de programas en los laboratorios docentes de la UCI.
- Se documentaron y modelaron los flujos de trabajo: Negocio, Requerimientos, Análisis, Diseño e Implementación.
- Se construyó una aplicación capaz de garantizar rapidez y efectividad en los procesos de solicitud de instalación de programas.



### **RECOMENDACIONES.**

Luego de concluir la realización del presente trabajo de diploma se recomienda al personal encargado del control de solicitudes e instalación de programas y a los miembros del proyecto SIGLA que se realicen las siguientes actividades y mejoras:

- Realizarse las pruebas pertinentes al sistema, para asegurar que no existan deficiencias en las funcionalidades implementadas.
- Desarrollar un catálogo de programas, ya que con el mismo se podrán facilitar y mejorar las acciones de solicitudes de instalación de programas.
- Realizar un manual de usuario para esclarecer cualquier duda que posean los usuarios cuando interactúen con el sistema.
- Incorporar nuevas funcionalidades que perfeccionen los servicios que brinda el sistema.

---

## BIBLIOGRAFÍA

1. **Worsley, John and Drake, Joshua.** *Características de PostgreSQL.* [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
2. —. *¿Qué es PostgreSQL?* [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node12.html>.
3. **Welling, Luke and Thomson, Laura.** *Desarrollo Web con PHP y MySQL.* [Online] [Cited: Enero 8, 2008.] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.
4. **Vizcaino, Aurora, García, Felix Óscar and Caballero, Ismael.** *Una Herramienta CASE para ADOO: Visual Paradigm.* [Online] [Cited: Noviembre 27, 2007.] [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1\\_VP.pdf](http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf).
5. **San Félix, Alvaro del Castillo.** *El servidor de Web Apache: Introducción práctica.* [Online] 2000. [Cited: Enero 8, 2008.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.
6. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** *El Lenguaje Unificado de Modelado.* [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
7. **Pitschke, Jurguen.** *Visual Paradigm Suite.* [Online] 2002-2008. [Cited: Enero 20, 2008.] <http://www.visual-paradigm.eu/content/view/12/1/lang,en/>.
8. **López Guzmán, Clara.** *Modelo para el Desarrollo de Bibliotecas Digitales Especializadas.* [Online] Marzo 2000. [Cited: Enero 20, 2008.] [http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec\\_26.htm](http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_26.htm).
9. **Lago, Ramiro.** *Patrón "Modelo-Vista-Controlador".* [Online] 2007. [Cited: Diciembre 10, 2007.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
10. **Eguíluz Pérez, Javier.** *Libros Web. Introducción a JavaScript.* [Online] 2007. [Cited: Diciembre 5, 2007.] <http://www.librosWeb.es/javascript>.
11. —. *Libros Web. Introducción a CSS.* [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosWeb.es/css>.
12. —. *Libros Web. Introducción a XHTML.* [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosWeb.es/xhtml>.
13. —. *Libros Web. Introducción a AJAX.* [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosWeb.es/ajax>.

- 
14. **Artiles Visual, Leticia, et al.** *PERFECCIONAMIENTO DE LA GESTION UNIVERSITARIA EN EL ISCMH: Proyecto Tocororo.* [Online] [Cited: Enero 8, 2008.] [http://www.ucmh.sld.cu/rhab/vol5\\_num1/rhcm10106.htm](http://www.ucmh.sld.cu/rhab/vol5_num1/rhcm10106.htm).
15. **Álvarez Romero, Eduardo and Pueyo, Daniel.** *Integration Definition For Funcion Modeling.* [Online] 2004-2005. [Cited: Diciembre 12, 2007.] <http://dmi.uib.es/~burguera/download/IDEF0trabajo.doc>.
16. **Alvarez, Miguel Angel.** *Aptana Studio.* [Online] [Cited: Diciembre 7, 2007.] <http://www.desarrolloWeb.com/articulos/aptana-studio.html>.
17. **Visual Paradigm.** *Visual Paradigm for UML.* [Online] [Cited: Diciembre 10, 2007.] <http://www.visual-paradigm.com/product/vpumil/>.
18. **Apache.** *Versión 2.2 de la documentación del Servidor de HTTP Apache.* [Online] 1995-2006. [Cited: Enero 20, 2008.] <http://www.apache2.org/>.
19. **Departamento de Control de Calidad y Auditoría Informática.** *Sistemas en Arquitectura Cliente/Servidor.* [Subdirección de Sistemas] s.l. : Dirección General de Servicios de Cómputo Académico, 2001.
20. **Edinet.** *Qué es XML?* [Online] 2002. [Cited: Diciembre 12, 2007.] <http://www.edinet.com/sabia2.asp>.
21. **DesarrolloWeb.** *Qué es PHP.* [Online] [Cited: Noviembre 17, 2007.] <http://www.desarrolloWeb.com/articulos/392.php>.
22. **pgAdmin PostgreSQL Tools .** *pgAdmin.* [Online] [Cited: Diciembre 10, 2008.] <http://www.pgadmin.org/>.
23. **ProgramacionWeb.net .** *MVC - Modelo Vista Controlador.* [Online] 2003-2008. [Cited: Enero 9, 2008.] <http://www.programacionWeb.net/articulos/articulo/?num=505>.
24. **XperimentoS.** *Los mejores 12 FrameWorks JavaScript.* [Online] [Cited: Noviembre 25, 2007.] <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript/>.
25. **Microsoft.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft.* [Online] Diciembre 2006. [Cited: Marzo 28, 2008.] [http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real\\_World\\_SOA.pdf](http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf).

- 
26. **World Wide Web Consortium.** *Guía Breve de Tecnologías XML* . [Online] 1994-2005. [Cited: Noviembre 25, 2007.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.
27. **GNU Image Manipulation Program.** *GIMP*. [Online] 2001. [Cited: Diciembre 10, 2007.]
28. **Gábilos Software.** *Gábilos Software*. [Online] 2001-2007. [Cited: Diciembre 2, 2007.] <http://www.gabilos.com/programasdegestion/prestacionesaraper.htm>.
29. **Maestros del Web.** *Conociendo los principales editores de páginas Web del mercado*. [Online] 1997-2007. [Cited: Enero 10, 2008.] <http://www.maestrosdelWeb.com/editorial/conociendo-los-principales-editores-de-paginas-Web-del-mercado/>.
30. **HTMLPOINT.com.** *¿Qué es Apache?* [Online] 1997-2006. [Cited: Diciembre 10, 2007.] <http://www.htmlpoint.com/faq/apache/01.htm>.
31. **Ext JS.** *Ext JS - JavaScript Library*. [Online] 2006. [Cited: Diciembre 10, 2007.] <http://extjs.com/>
32. **Martínez, Leticia and Peláez, Orfilio.** Radio Nuevitas. *Radio Nuevitas*. [Online] Radio Nuevitas Digital, Marzo 14, 2002. [Cited: Diciembre 10, 2007.] [http://www.radionuevitas.co.cu/2007/10/07/universidad\\_ciencias\\_informaticas.asp](http://www.radionuevitas.co.cu/2007/10/07/universidad_ciencias_informaticas.asp).

## ANEXOS

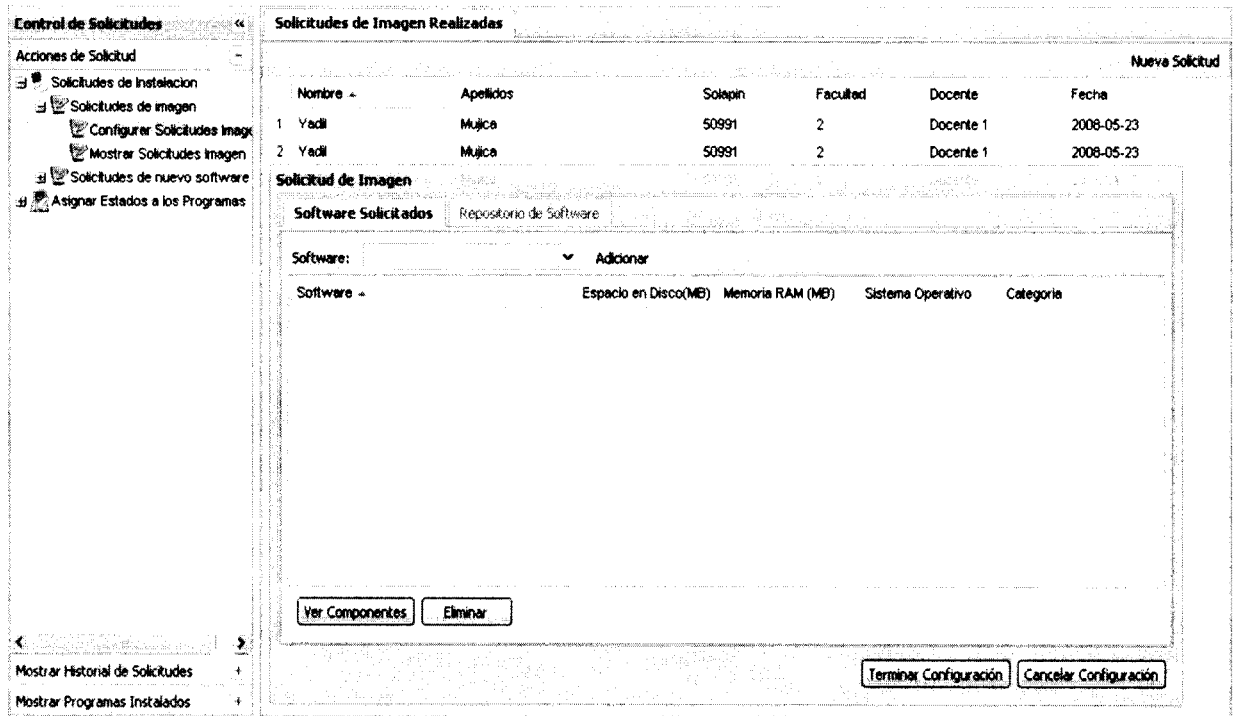
### Anexo. No 1: Descripción de los casos de uso del sistema.

<b>Caso de Uso:</b>	<b>Gestionar Solicitud de Instalación de Imagen.</b>	
<b>Actores:</b>	<b>Intermediario</b>	
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea realizar una nueva solicitud de instalación de imagen, modificar una ya existente o ver los datos de una que haya realizado, las acciones que se realizan para modificar e insertar son: insertar un nuevo programa al listado de la solicitud, eliminar uno ya existente en este listado y modificar la selección de los elementos de algún programa solicitado (si posee elementos). Para ello, se le proporciona los programas disponibles, los solicitados y se le admite realizar las actividades antes mencionadas. El caso de uso termina.	
<b>Precondiciones:</b>	Para poder realizar la acción de gestionar el usuario debe haberse autenticado y tener los permisos requeridos para realizar esta operación.	
<b>Referencias</b>	R1.1 - R1.2 - R1.3 - R1.4 - R3.	
<b>Prioridad</b>	<b>Crítica</b>	
<b>Flujo Normal de Eventos</b>		
<b>Sección "Configurar Solicitudes de Instalación de Imagen"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El Intermediario selecciona la opción: "Solicitudes de Instalación".	2- Muestra las opciones: Solicitudes de Instalación de Imagen y de Nuevo Programa.	
3- Selecciona la opción: "Solicitudes de Imagen"	4- Muestra las acciones referentes a la gestión de solicitudes de instalación de imagen.	

	<ul style="list-style-type: none"> <li>• Configurar Solicitudes de Instalación de Imagen.</li> <li>• Mostrar Información de una Solicitud de Instalación de Imagen.</li> </ul>
5- Selecciona la opción: "Configurar Solicitudes de Instalación de Imagen".	6- Muestra el listado de las solicitudes con la fecha en las que fueron realizadas y la información del solicitante (nombre, apellido, número de solapín, facultad y docente al que pertenece). Y permite: <ul style="list-style-type: none"> <li>• Configurar una de las solicitudes.</li> <li>• Configurar una nueva solicitud.</li> </ul>
7- Selecciona la opción: "Configurar Nueva Solicitud"	8- Muestra el listado de los programas disponibles con sus características (espacio en disco duro que ocupa, memoria RAM que necesita, su compatibilidad con Linux y Windows, y la categoría).
9- Selecciona un programa.	10-Permite: <ul style="list-style-type: none"> <li>• Insertar el programa seleccionado.</li> <li>• Eliminar el programa seleccionado</li> <li>• Modificar los elementos del programa seleccionado.</li> </ul>
11-Selecciona la opción: Incluir Programa.	12-Se incluye el programa con el estado por defecto en pendiente.
13-Escoge la opción: Terminar Configuración.	14-Comprueba la longitud del listado de los programas solicitados.
	15-Guarda la configuración de la solicitud.

	<p>16-Envía automáticamente un correo al Técnico de Software del área a la que pertenece el intermediario informándole que se ha realizado una configuración de solicitud.</p>
	<p>17-El caso de uso termina.</p>

**Prototipo de Interfaz**



**Flujos Alternos**

\*.a El actor selecciona la opción de Cancelar configuración.

Acción del Actor	Respuesta del Sistema
	<p>*.a.1- Elimina los datos creados.</p>
	<p>*.a.2- Muestra un mensaje de información: "La</p>

	acción ha sido cancelada”.
	*.a.3- Termina el caso de uso.
<b>Prototipo de Interfaz</b>	
6. a El actor selecciona la opción de configurar la solicitud seleccionada.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6. a.1-Muestra dos listados de programas disponibles y los solicitados.
	6. a.2- Continúa en el paso 9 del flujo básico.
9. a El usuario teclea el nombre del programa que desea seleccionar.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	9. a.1- Muestra, a medidas que el Intermediario introduce los caracteres, los programas que sus nombres están compuestos por esas letras.
9. a.2- Selecciona un programa	
	9. a.3- Continúa en el paso 10 del flujo básico.
10. a El usuario selecciona la opción de eliminar el programa seleccionado.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	10. a.1- Muestra un mensaje confirmando la acción.
10. a.2- Confirma que desea eliminar el programa.	10. a.3- Elimina el programa del listado de los solicitados.



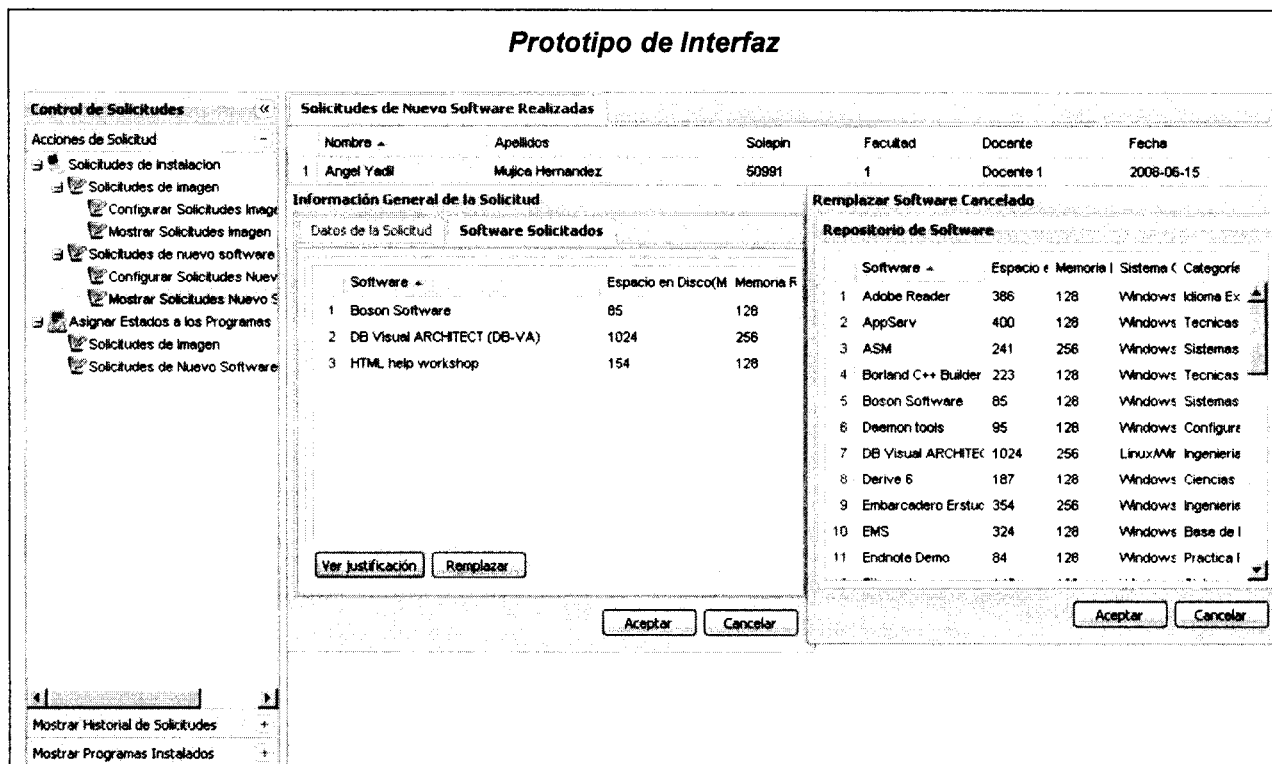
	10. a.4- Continúa en el paso 13 o regresa al 9 del flujo básico.
10. a.2.a- El usuario cancela la acción de eliminar el programa.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	10. a.2.a.1- Cancela la acción.
	10. a.2.a.2- Continúa en el paso 13 o regresa al 9 del flujo básico.
10. b El usuario modifica los elementos del programa seleccionado (Por defecto todos los elementos están seleccionados).	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	10. b.1- Posibilita continuar en el paso 13 del flujo básico o regresa al paso 9 del flujo básico.
14. a- El listado de los programas solicitados está vacío.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	14.a.1- Muestra un mensaje de información: "Debe incluir al menos un programa"
	12. a.2- Regresa al paso 9 del flujo básico.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Mostrar Información de una Solicitud de Instalación de Imagen"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- Selecciona la opción: "Mostrar Solicitudes de Imagen".	2- Muestra el listado de las solicitudes con las fechas en las que fueron realizadas

	<p>y la información del solicitante (nombre, apellido, facultad y docente al que pertenece).</p>
<p>3- Selecciona la solicitud de la cual desea ver su información.</p>	<p>4- Muestra el listado de los programas solicitados con sus características (espacio en disco duro que ocupa, memoria RAM que necesita, su compatibilidad con Linux y Windows, la categoría y el estado).</p> <p>En caso de que el estado sea cancelado permite ver la justificación del porqué fue derogada la instalación del programa.</p>
<p>5- Selecciona la opción: Ver Justificación.</p>	<p>6- Muestra la justificación introducida por el Técnico de Software y brinda la posibilidad de insertar programas que sustituyan a los que fueron cancelados (Ver descripción del caso de uso: <b>Sustituir Programas Cancelados.</b>).</p>
	<p>7- El caso de uso termina.</p>

<b>Prototipo de Interfaz</b>																									
<p><b>Control de Solicitudes</b></p> <ul style="list-style-type: none"> <li>Acciones de Solicitud</li> <li>  Solicitudes de Instalacion</li> <li>  Solicitudes de imagen               <ul style="list-style-type: none"> <li>Configurar Solicitudes Imagen</li> <li>Mostrar Solicitudes Imagen</li> </ul> </li> <li>  Solicitudes de nuevo software</li> <li>  Asignar Estados a los Programas</li> </ul> <p>Mostrar Historial de Solicitudes +</p> <p>Mostrar Programas Instalados +</p>	<p><b>Solicitudes de Imagen Realizadas</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Nombre</th> <th>Apellidos</th> <th>Solepin</th> <th>Facultad</th> <th>Docente</th> <th>Fecha</th> </tr> </thead> <tbody> <tr> <td>1 Angel Yadi</td> <td>Mujica Hernandez</td> <td>50991</td> <td>1</td> <td>Docente 1</td> <td>2008-06-15</td> </tr> </tbody> </table> <p><b>Información General de la Solicitud</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Software</th> <th>Espacio en Disco</th> <th>Memoria RAM (MB)</th> <th>Sistema Operativo</th> <th>Categoría</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>1 Borland C++ Builder 6</td> <td>223</td> <td>120</td> <td>Windows</td> <td>Tecnicas de Progr</td> <td>Pendiente</td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="Ver justificación"/> <input type="button" value="Reemplazar"/> </p> <p style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p>	Nombre	Apellidos	Solepin	Facultad	Docente	Fecha	1 Angel Yadi	Mujica Hernandez	50991	1	Docente 1	2008-06-15	Software	Espacio en Disco	Memoria RAM (MB)	Sistema Operativo	Categoría	Estado	1 Borland C++ Builder 6	223	120	Windows	Tecnicas de Progr	Pendiente
Nombre	Apellidos	Solepin	Facultad	Docente	Fecha																				
1 Angel Yadi	Mujica Hernandez	50991	1	Docente 1	2008-06-15																				
Software	Espacio en Disco	Memoria RAM (MB)	Sistema Operativo	Categoría	Estado																				
1 Borland C++ Builder 6	223	120	Windows	Tecnicas de Progr	Pendiente																				
<b>Poscondiciones</b>	Se gestionó una solicitud de instalación de Imagen.																								

<b>Caso de Uso:</b>	<b>Sustituir Programas Cancelados.</b>
<b>Actores:</b>	<b>Intermediario</b>
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea sustituir los programas que fueron cancelados por el técnico de software por otros que solicitará. Para ello, se le proporciona los programas que fueron cancelados y los que aún no se hayan solicitado, y se le permite eliminar los anulados e insertar otros en su lugar. El caso de uso termina.
<b>Precondiciones:</b>	El usuario debe haberse autenticado, tener los permisos requeridos para realizar esta operación y haber algún programa en estado cancelado.
<b>Referencias</b>	<b>R4</b>

Prioridad	Secundario.
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	1- Muestra, además de los datos establecidos de una solicitud, los programas que fueron cancelados y los que aún no han sido solicitados.
2- Selecciona un programa.	3- Permite: <ul style="list-style-type: none"> <li>• Insertar el programa seleccionado.</li> <li>• Eliminar el programa seleccionado.</li> <li>• Modificar los elementos del programa seleccionado.</li> </ul>
4- Selecciona la opción: Incluir Programa.	5- Se incluye el software con el estado por defecto en pendiente.
6- Escoge la opción: Concluir Sustitución.	7- Se comprueba que haya en el listado de los programas de la solicitud la misma cantidad que los mostrados.
	8- Guarda los cambios.
	9- Envía automáticamente un correo al Técnico de Software del área a la que pertenece el intermediario informándole que se ha realizado una sustitución de los programas cancelados.
	10- El caso de uso termina.



**Flujos Alternos**

\*.a El actor selecciona la opción de Cancelar sustitución.

Acción del Actor	Respuesta del Sistema
	*.a.1- Elimina los datos creados.
	*.a.2- Muestra un mensaje de información: "La acción ha sido cancelada".
	*.a.3- El caso de uso termina.

2. a El usuario teclea el nombre del programa que desea seleccionar.

Acción del Actor	Respuesta del Sistema

	2. a.1- Muestra, a medidas que el Intermediario introduce los caracteres, los software que sus nombres están compuestos por estas letras en el orden en que lo valla tecleando el usuario.
2. a.2- Selecciona un programa	
	2. a.2- Continúa en el paso 3 del flujo básico.
3. a El usuario selecciona la opción de eliminar el software seleccionado.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. a.1- Muestra un mensaje confirmando la acción.
3. a.2- Confirma que desea eliminar el programa.	3. a.3- Elimina el programa del listado de los solicitados.
	3. a.4- Continúa en el paso 6 o regresa al 2 del flujo básico.
3. a.2.a- El usuario cancela la acción de eliminar el programa.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. a.2.a.1- Cancela la acción.
	3. a.2.a.2- Continúa en el paso 6 o regresa al 2 del flujo básico.
3. b El usuario modifica los elementos del software seleccionado.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. b.1- Continúa en el paso 6 o regresa al 2 del

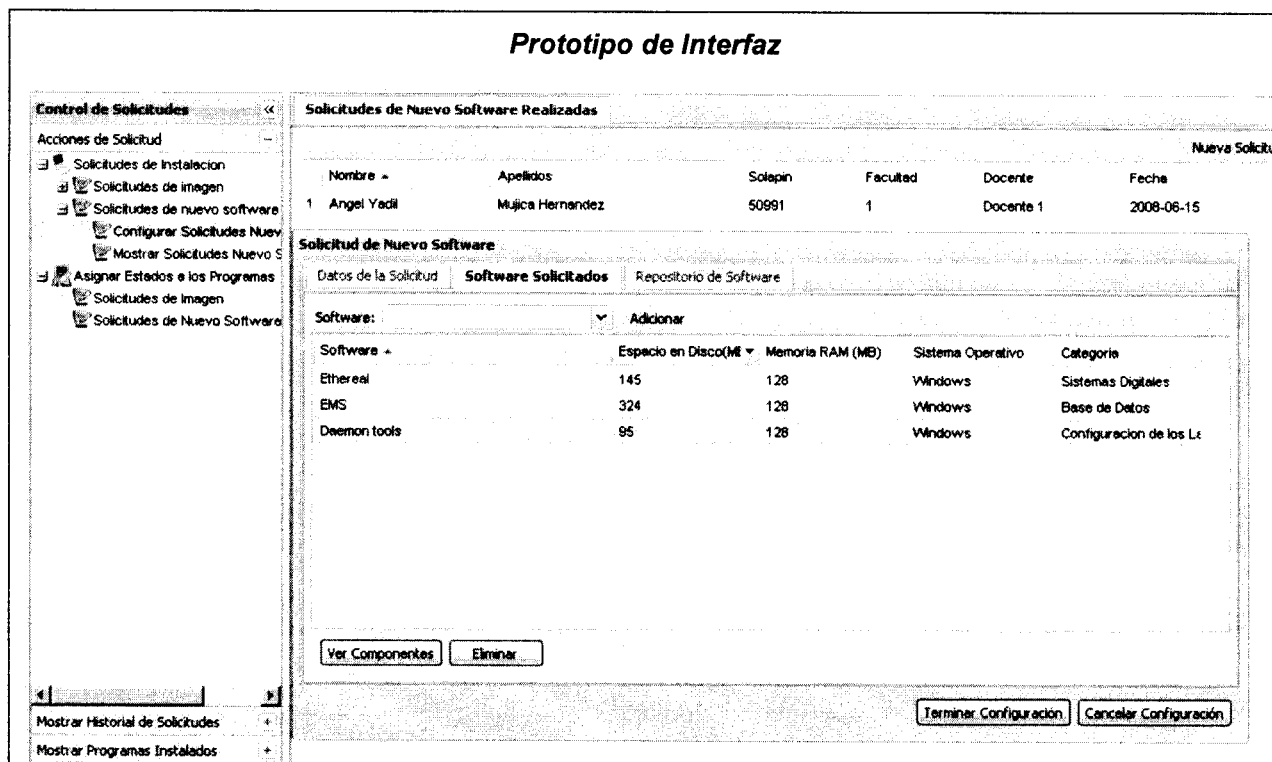
	flujo básico.
8. a- Se comprueba que haya en el listado de los programas de la solicitud la misma cantidad que los mostrados.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	8.a.1- Muestra un mensaje de información: "Debe haber en el listado de los programas solicitados la misma cantidad que la mostrada"
	8. a.2- Regresa al paso 2 del flujo básico.

<b>Caso de Uso:</b>	<b>Gestionar Solicitud de Instalación de Nuevos Programa.</b>
<b>Actores:</b>	<b>Intermediario</b>
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea realizar una solicitud de instalación de nuevos programas, modificar una ya existente o mostrar la información de una solicitud, las acciones que se realizan son: insertar un nuevo programa al listado de la solicitud, eliminar uno ya existente en este listado y modificar la selección de los elementos de algún programa solicitado (si posee elementos). Para ello, se le proporciona los programas disponibles, los solicitados y se le admite realizar las actividades antes mencionadas. El caso de uso termina.
<b>Precondiciones:</b>	Para poder realizar la acción de gestión el usuario debe haberse autenticado y tener los permisos requeridos para realizar esta acción.
<b>Referencias</b>	<b>R2.1 - R2.2 - R2.3 - R.3 - R.4.</b>
<b>Prioridad</b>	<b>Crítica</b>
<b>Flujo Normal de Eventos</b>	

<b>Sección "Configurar Solicitudes de Instalación de Nuevos Programas"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Intermediario selecciona la opción: "Solicitudes de Instalación".	2- Muestra las opciones: Solicitudes de Instalación de Imagen y de Nuevo Programa.
3- Selecciona la opción: "Solicitudes de Instalación de un Nuevo Programa".	4- Muestra las acciones referentes a la gestión de solicitudes de instalación de imagen.
5- Selecciona la opción: " Configurar Solicitudes de Instalación de un Nuevo Programa"	6- Muestra el listado de las solicitudes con la fecha en las que fueron realizadas y la información del solicitante (nombre, apellido, número de solapín, facultad y docente al que pertenece). Y permite: <ul style="list-style-type: none"> <li>• Configurar una de las solicitudes.</li> <li>• Configurar una nueva solicitud.</li> </ul>
7- Selecciona la opción: "Configurar Nueva Solicitud"	8- Muestra el listado de los programas disponibles con sus características (espacio en disco duro que ocupa, memoria RAM que necesita, su compatibilidad con Linux y Windows, y la categoría).  Brinda la posibilidad de introducir: <ul style="list-style-type: none"> <li>8- Fecha para la que necesita que esté instalado el programa (Fecha de inicio).</li> <li>9- Fecha hasta la que necesita que esté instalado el programa (fecha tope).</li> </ul>



	10-Laboratorios donde serán instalados los programas.
9- Introduce la fecha en la que necesita que esté instalado el programa.	10- Verifica la fecha para la que debe estar instalado el programa (fecha de inicio).
11- Selecciona los laboratorios en los que desea que esté instalado el programa.	
12- Selecciona un programa.	13- Permite: <ul style="list-style-type: none"> <li>• Insertar el programa seleccionado.</li> <li>• Eliminar el programa seleccionado</li> <li>• Modificar los elementos del programa seleccionado.</li> </ul>
14- Selecciona la opción: Incluir Programa.	15- Se incluye el programa con su estado como pendiente.
16- Escoge la opción: Terminar Configuración.	17- Comprueba la longitud del listado de los programas solicitados y que se haya seleccionado al menos un laboratorio.
	18- Guarda la configuración de la solicitud.
	19- Envía automáticamente un correo al Técnico de Software del área a la que pertenece el intermediario informándole que se ha realizado una configuración de solicitud.
	20- El caso de uso termina.



**Flujos Alternos**

\*.a El actor selecciona la opción de Cancelar configuración.

Acción del Actor	Respuesta del Sistema
	*.a.1- Elimina los datos creados.
	*.a.2- Muestra un mensaje de información "La acción ha sido cancelada".
	*.a.3- Termina el caso de uso.

6. a. El actor selecciona la opción de configurar la solicitud seleccionada.

Acción del Actor	Respuesta del Sistema

	<p>6. a.1- Muestra de la solicitud a modificar:</p> <ul style="list-style-type: none"> <li>• Los listados de los programas disponibles y los solicitados</li> <li>• Fecha para la que necesita que esté instalado el programa (Fecha de inicio).</li> <li>• Fecha hasta la que necesita que esté instalado el programa (fecha tope).</li> <li>• Laboratorios donde serán instalados los programas.</li> </ul>
	<p>6.a.2- Permite modificar:</p> <ul style="list-style-type: none"> <li>• Fecha para la que necesita que esté instalado el programa (Fecha de inicio).</li> <li>• Fecha hasta la que necesita que esté instalado el programa (fecha tope).</li> <li>• Laboratorios donde serán instalados los programas.</li> </ul>
<p>6. a.3- Modifica la fecha en la que quiere que esté instalado el programa.</p>	
<p>6. a.4- Modifica la selección de los laboratorios en los que será instalado el programa.</p>	
	<p>6. a.5- Continúa en el paso 12 del flujo básico.</p>
<p>10. a- Se verifica que la fecha de inicio de la instalación tenga una diferencia como mínimo de 15 días con la fecha en la que se realiza la solicitud.</p>	

Acción del Actor	Respuesta del Sistema
	10. a.1- Muestra un cartel de información: "Para la fecha que ha introducido no se garantiza que el nuevo programa esté instalado, por favor, realice su pedido con 15 días como mínimo de antelación".
	10. a.2- Regresa al paso 9 del flujo básico.
12. a El usuario teclea el nombre del programa que desea seleccionar.	
Acción del Actor	Respuesta del Sistema
	12. a.1- Muestra, a medidas que el Intermediario introduce los caracteres, los programas que sus nombres están compuestos por esas letras.
12. a.2- Selecciona un programa	
	12. a.3- Continúa en el paso 13 del flujo básico.
13. a El usuario selecciona la opción de eliminar el programa seleccionado.	
Acción del Actor	Respuesta del Sistema
	13. a.1- Muestra un mensaje confirmando la acción.
13. a.2- Confirma que desea eliminar el programa.	13. a.3- Elimina el programa del listado de los solicitados.
	13. a.4- Continúa en el paso 16 o regresa al 12 del flujo básico.

13. a.2.a- El usuario cancela la acción de eliminar el programa.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	13. a.2.a.1- Cancela la acción.
	13. a.2.a.2- Continúa en el paso 16 o regresa al 12 del flujo básico.
13. b El usuario modifica los elementos del programa seleccionado (Por defecto todos los elementos están seleccionados).	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	13. b.1- Continúa en el paso 16 o regresa al 12 del flujo básico.
17. a- El listado de los programas solicitados está vacío.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	17.a.1- Muestra un mensaje de información: "Debe incluir al menos un programa"
	17. a.2- Regresa al paso 12 del flujo básico.
17. c- No se ha seleccionado ningún laboratorio.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	17. c.1- Muestra un mensaje de información: "Debe seleccionar al menos un laboratorio."
	17. c.2- Regresa al paso 11 del flujo básico.

<b>Flujo Normal de Eventos</b>	
<b>Sección "Mostrar Información de una Solicitud de Instalación de Nuevos Programas"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El Intermediario selecciona la opción: Solicitudes de Instalación.	2- Muestra las opciones: Solicitudes de Instalación de Imagen y de Nuevo Programa.
3- Selecciona la opción: "Solicitudes de Instalación de un Nuevo Programa".	4- Muestra las acciones referentes a la gestión de solicitudes de Instalación de un Nuevo Programa.
5- Selecciona la opción: "Mostrar Solicitudes de Nuevos Programas".	6- Muestra el listado de las solicitudes con las fechas en las que fueron realizadas y la información del solicitante (nombre, apellido, facultad y docente al que pertenece).

<p>7- Selecciona la solicitud de la cual desea observar su información.</p>	<p>8- Muestra los datos referentes a la solicitud realizada.</p> <ul style="list-style-type: none"><li>• Fecha para la que necesita que esté instalado el programa (Fecha de inicio).</li><li>• Fecha hasta la que necesita que esté instalado el programa (fecha tope).</li><li>• Laboratorios en los que serán instalados los programas.</li><li>• Listado de los programas solicitados con sus características (espacio en disco duro, memoria RAM, compatibilidad, categoría y estado).</li></ul> <p>En caso de que el estado sea cancelado permite ver la justificación del porqué fue derogada la instalación del programa.</p>
<p>9- Selecciona la opción: " Ver Justificación"</p>	<p>10- Muestra la justificación introducida por el Técnico de Software y brinda la posibilidad de insertar programas que sustituyan a los que fueron cancelados (Ver descripción del caso de uso: <b>Sustituir Programas Cancelados</b>).</p>
	<p>11- El caso de uso termina.</p>

**Prototipo de Interfaz**

**Control de Solicitudes**

**Solicitudes de Nuevo Software Realizadas**

Nombre	Apellidos	Solepin	Facultad	Docente	Fecha
1. Angel Yañil	Mujica Hernandez	50991	1	Docente 1	2008-06-15

**Información General de la Solicitud**

Software	Especio en Disco(M)	Memoria RAM (MB)	Sistema Operativo	Categoría
1. Boson Software	85	128	Windows	Sistemas Digita
2. DB Visual ARCHITECT (DB-VA)	1024	256	Linux/Windows	Ingeniería y Ge
3. HTML help workshop	154	128	Windows	Practica Profes

Mostrar Historial de Solicitudes +  
Mostrar Programas Instalados +

**Poscondiciones**      Se gestionó una solicitud de instalación de Nuevos Programas.

<b>Caso de Uso:</b>	<b>Mostrar Historial de Solicitudes.</b>
<b>Actores:</b>	<b>Especialista General de Software.</b>
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción de ver el historial de solicitudes, y se le proporciona todas las solicitudes que se han realizado tanto por curso, como por área o por un rango de fecha que introduzca. El caso de uso termina.
<b>Precondiciones</b>	Para poder realizar la acción deben haberse registrado solicitudes de instalación, el usuario debe estar autenticado y tener los permisos requeridos



	para realizar esta operación.
<b>Referencias</b>	R8
<b>Prioridad</b>	<b>Secundaria</b>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona la opción: "Mostrar Historial de Solicitudes".	2- Brinda la posibilidad de seleccionar los siguientes criterios de búsqueda: <ul style="list-style-type: none"> <li>• Solicitudes por Curso.</li> <li>• Solicitudes por Área.</li> <li>• Solicitudes por fecha.</li> </ul>
3- Selecciona un criterio de búsqueda.	
4- Selecciona la opción: "Mostrar Solicitudes por Curso".	5- Muestra los tipos de solicitudes que se realizan (solicitudes de instalación de imagen y de nuevo programa).
6- Selecciona el tipo de solicitud del cual desea ver el historial.	7- Muestra el listado de los cursos ordenados descendientemente.
8- Selecciona el curso del cual desea ver las solicitudes.	
9- Selecciona la opción: "Buscar"	10- Consulta los datos y muestra un listado de posibles coincidencias.  De las solicitudes encontradas se reflejan los siguientes datos: <ul style="list-style-type: none"> <li>• Fecha en las que fueron realizadas</li> <li>• La información del solicitante</li> </ul>

	(nombre, apellido, facultad y docente al que pertenece).
11-Selecciona una de las solicitudes.	<p>12-Muestra de la solicitud seleccionada el listado de los programas solicitados con sus características (espacio en disco duro, memoria RAM, compatibilidad, categoría y estado).</p> <p>En caso de que sea una solicitud del tipo: Instalación de un nuevo programa se muestran además.</p> <ul style="list-style-type: none"><li>• Fecha para la que necesita que esté instalado el programa (Fecha de inicio).</li><li>• Fecha hasta la que necesita que esté instalado el programa (fecha tope).</li><li>• Laboratorios en los que serán instalados los programas.</li></ul> <p>Y brinda la posibilidad de generar un documento que contenga toda la información referente a la solicitud y los datos del solicitante.</p>
	13-El caso de uso termina.

**Prototipo de Interfaz**

**Control de Solicitudes**

Acciones de Solicitud

Mostrar Historial de Solicitudes

- [-] Solicitudes por Curso
- [-] Imagen Por Curso
- [-] Nuevo Software Por Curso
- [+] Solicitudes por Area
- [+] Solicitudes por Rango de Fecha

Mostrar Programas Instalados

**Solicitudes de Imagen por Curso**

Curso: 2007-2008 Buscar

Nombre	Apellidos	Solapin	Facultad	Docente	Fecha	
1	Angel Yadi	Mujica Hernandez	50991	1	Docente 1	2008-06-15
2	Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-12
3	Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-12
4	Osmany	Lopez Cardoso	50984	2	Docente 1	2008-06-12
5	Osmany	Lopez Cardoso	50984	2	Docente 1	2008-06-12
6	Yoslaine de la Caridad	Ramirez Morales	50993	1	Docente 1	2008-06-13
7	Yoslaine de la Caridad	Ramirez Morales	50993	1	Docente 1	2008-06-13
8	Yoslaine de la Caridad	Ramirez Morales	50993	1	Docente 1	2008-06-12

**Información General de la Solicitud**

**Software Solicitados**

Software	Espacio en Disco(M)	Memoria RAM (MB)	Sistema Operativo	Categoria	Estado
1 Borland C++ Builder 6	223	128	Windows	Tecnicas de Programacion	Pendiente

### Flujos Alternos

2.a- El usuario selecciona la opción: “Mostrar Solicitudes por Área”

Acción del Actor	Respuesta del Sistema
	2. a.1- Muestra los tipos de solicitudes (de instalación de imagen y de un nuevo programa).
2. a.2- Selecciona el tipo de solicitud del cual desea ver el historial.	2. a.3- Muestra el listado de los docentes.
2. a.4- Selecciona el docente del cual desea ver las solicitudes.	
	2. a.5- Continúa en el paso 9 del flujo básico.

2. b- El usuario selecciona la opción: "Mostrar Solicitudes por fecha."	
Acción del Actor	Respuesta del Sistema
	2. b.1- Muestra los tipos de solicitudes (de instalación de imagen y de un nuevo programa).
2. b.2- Selecciona el tipo de solicitud del cual desea ver el historial.	2. b.3- Muestra los campos de fecha (inicio y fin de la instalación) a introducir.
2. b.4- Introduce las fechas.	2. b.5- Continúa en el paso 9 del flujo básico.
10. a- El sistema no encuentra ninguna coincidencia.	
Acción del Actor	Respuesta del Sistema
	10. a.1- Muestra un mensaje de Información: "No se han encontrado coincidencias".
	10. a.2- Regresa al paso 3 del flujo básico.
12. a: El usuario selecciona la opción Generar documento.	
Acción del Actor	Respuesta del Sistema
	12. a.1- Genera un documento con la información de la solicitud, pero de los programas de la solicitud sólo se van a mostrar en el documento los que su instalación fue aceptada.
	12. a.1- El caso de uso termina.
<b>Poscondiciones</b>	Se Mostraron los datos referentes a una solicitud.

<b>Caso de Uso:</b>	<b>Asignar Estados a los Programas.</b>	
<b>Actores:</b>	<b>Técnico de Software.</b>	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario determina, según las características de los programas solicitados, el estado que pueden tener; es decir, acepta, cancela o deja en modo pendiente la instalación de los mismos, y en caso de cancelar la instalación el usuario debe introducir una justificación de su decisión. El caso de uso termina.	
<b>Precondiciones</b>	Para poder realizar la acción, un intermediario perteneciente al docente que atiende el técnico de software debió con anterioridad realizar una solicitud de instalación; además, el usuario debe estar autenticado y tener los permisos requeridos para realizar esta operación.	
<b>Referencias</b>	<b>R5</b>	
<b>Prioridad</b>	<b>Crítica</b>	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1- El Técnico de Software selecciona la opción: "Asignar estados a los programas".	2- Muestra los tipos de solicitudes que se realizan (solicitudes de instalación de imagen y de nuevo programa).
	3- Selecciona un tipo de solicitud.	4- Muestra el listado de áreas.
	5- Selecciona un área.	
	6- Selecciona la opción: "Buscar"	7- Busca las solicitudes realizadas por el área y tipo indicado.
		8- Muestra el listado de las solicitudes del tipo indicado con la fecha en las que

	<p>fueron realizadas y la información del solicitante (nombre, apellido, número de solapín, facultad y docente al que pertenece) y además, indica cuales han sido modificadas.</p>
<p>9- Selecciona la solicitud a la que desea asignarle estados a sus programas.</p>	
<p>10- Selecciona la opción: Asignar Estados.</p>	<p>11-Muestra de los programas solicitados:</p> <ul style="list-style-type: none"> <li>• Espacio en disco duro y memoria RAM que necesitan.</li> <li>• Compatibilidad con los sistemas operativos Linux y Windows.</li> <li>• Categoría a la que pertenece.</li> <li>• Estado de cada uno de los programas (En caso de que sea la primera vez que el técnico consulta la solicitud se muestran los estados en modo pendientes).</li> </ul>
<p>12-Asigna el estado a los programas, los estados son:</p> <ul style="list-style-type: none"> <li>• Aceptado.</li> <li>• Cancelado.</li> <li>• Pendiente.</li> </ul>	
<p>13-Acepta los estados asignados a los programas de la solicitud.</p>	<p>14-Guarda los estados de los programas de la solicitud.</p>

15-Envía automáticamente un correo al intermediario que realizó esa solicitud, informándole que a los programas se le han asignado estados.

16-El caso de uso termina.

### Prototipo de Interfaz

**Control de Solicitudes**

Acciones de Solicitud

- Solicitudes de Instalacion
- Asignar Estados a los Programas
- Solicitudes de Imagen
- Solicitudes de Nuevo Software

**Asignar Estado a las Solicitudes de Imagen**

Docente: Docente 2    Buscar

Nombre	Apellidos	Solepin	Facultad	Docente	Fecha
1 Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-13
2 Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-13
3 Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-13
4 Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-13
5 Orlando	Sanchez Castro	50994	3	Docente 2	2008-06-13

**Información General de la Solicitud**

**Software Solicitados**

Solicitud Instalada

Software	Espacio en Disco(M)	Memoria RAM (MB)	Sistema Operativo	Categoria	Estado
1 PostgreSQL	441	128	Linux/Windows	Base de Datos	Cancelado
2 Win Prolog 4040	83	64	Windows	Tecnicas de Progra	Aceptado
3 WinRar	54	64	Windows	Configuracion de lo	Cancelado

Mostrar Historial de Solicitudes +  
Mostrar Programas Instalados +

Aceptar    Cancelar

### Flujos Alternos

\*.a Cancela la asignación de los estados a los programas solicitados.

#### Acción del Actor

#### Respuesta del Sistema

\*.a.1- Anula los estados asignados y vuelven a su estado pendiente.

\*.a.2- El caso de uso termina.

12. a- Coloca en cancelado el estado de un programa.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	12. a.1- Permite introducir una justificación de la cancelación de la instalación del programa solicitado.
12. a.2- Introduce justificación.	12. a.3- Verifica que haya justificado el estado de instalación cancelado.
	12. a.4- Continúa en el paso 12 o cambia al 13 del flujo básico.
12. a.3.a.- No se ha introducido la justificación de la cancelación.	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	12.a.3.a.1- Muestra un cartel de información: "Debe introducir la explicación de su decisión"
12. a.3.a.2- Introduce justificación.	12. a.3.a.3- Continúa en el paso 12 o cambia al 13 del flujo básico.
<b>Poscondiciones</b>	Se le asignaron estados a los programas de la solicitud.



<b>Caso de Uso:</b>	<b>Mostrar Programas Instalados en los Laboratorios.</b>	
<b>Actores:</b>	<b>Usuario.</b>	
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea ver los programas que han sido instalados en los laboratorios docentes de la universidad. Una vez mostrada la información requerida termina el caso de uso.	
<b>Precondiciones:</b>	Para poder realizar la acción deben haberse realizado solicitudes de instalación de programas, el usuario debe estar autenticado y tener los permisos requeridos para realizar esta operación.	
<b>Referencias</b>	<b>R7</b>	
<b>Prioridad</b>	<b>Segundaria</b>	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1- El usuario selecciona la opción: "Ver programas instalados en los laboratorios docentes"	2- Muestra el listado de los docentes y sus laboratorios.
	3- Selecciona el docente y el laboratorio del cual desea ver los programas que están instalados.	
	4- Selecciona la opción: "Buscar".	5- Busca los programas instalados.
		6- Muestra el listado de los programas solicitados y de estos: <ul style="list-style-type: none"> <li>• El Nombre.</li> <li>• Espacio en disco duro y memoria</li> </ul>

---

	<p>que requieren.</p> <ul style="list-style-type: none"><li>• Compatibilidad con los sistemas Linux y Windows.</li><li>• La categoría</li></ul>
<b>Poscondiciones</b>	Se mostraron los programas instalados en los laboratorios del docente seleccionado.

## GLOSARIO DE TÉRMINOS

### A

**Aplicación:** Programa que resuelve un problema determinado para un cliente final.

**Automatizar:** Hacer automático el funcionamiento de un proceso que se realiza manualmente.

**Asíncrona:** Tipo de comunicación donde cada byte se transmite del emisor al receptor de modo independiente.

**APPSERV:** Es una herramienta OpenSource para Windows que facilita la instalación de Apache, MySQL y PHP en una sola herramienta.

### B

**Biblioteca:** Es un conjunto de procedimientos y funciones (subprogramas) agrupadas en un archivo con el fin de ser aprovechadas por otros programas.

### C

**Correo electrónico:** Servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente mediante sistemas de comunicación electrónicos.

**Cookies:** Archivo de texto que se graba en el ordenador del visitante del cual se sirven los servidores Web para guardar información acerca del cliente de un sitio. Sirve para identificar a visitantes recurrentes.

**Código abierto:** Es el término con el que se conoce al software distribuido y desarrollado libremente. Un programa de código abierto puede ser libre, pero también puede ser semilibre o incluso completamente no libre y muchos de estos programas restringen los derechos de los usuarios.

**Código:** Texto escrito en un lenguaje de programación que ha de ser compilado o interpretado para ejecutarse en una computadora.

### D

**DOM:** [Document Object Model o Modelo de objetos en documentos]. Es una interfaz independiente de la plataforma y del lenguaje, permite que los programas y scripts tengan acceso dinámicamente y actualicen el contenido, la estructura y estilo de los documentos.

### E

**Entidad:** Término utilizado para hacer referencia a una organización o institución.

**Estaciones de trabajo:** Son los ordenadores que van a utilizar los clientes para acceder a la aplicación.

### F

**FrameWork:** Es una estructura de soporte definida en la cual, otro proyecto de software puede ser organizado y desarrollado e incluye soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**FTP:** Sigla de File Transfer Protocol (Protocolo de transferencia de ficheros). Es un protocolo que permite copiar ficheros desde un sistema a otro a través de la red.

### H

**Herramienta CASE:** CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas al aumento de la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y dinero.

**HTTPS:** Hypertext Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

**Historial de solicitudes:** Conjunto de solicitudes que se han almacenado al pasar de los cursos.

**Hoja de estilo:** Conjunto de instrucciones que se asocian a los archivos de texto y se ocupan de los aspectos de formato y presentación de los contenidos como son: tipo, fuente y tamaño de letras, justificación del texto, colores y fondos, etc.

### I

**INFOMED:** Es el conjunto de individuos, organizaciones, productos y servicios de información que tributan a la salud en Cuba y en el resto del mundo.

**Informatizar:** Aplicar sistemas y equipos informáticos al tratamiento de la información.

**Ícono:** Pequeña imagen gráfica mostrada en la pantalla que representa un objeto manipulable por el usuario y pretende asociar un comportamiento del sistema de un modo intuitivo.

**Interfaz:** Es la parte de una aplicación que se encarga de interactuar con el usuario. Comprende las pantallas y los elementos que informan al usuario sobre lo que puede hacer, o sobre lo que está sucediendo.

**IDE:** Entorno Integrado de Desarrollo. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica; que puede utilizar uno o varios lenguajes de programación.

### **J**

**JQuery:** Librería de JavaScript que enfatiza la interacción entre JavaScript y HTML.

### **L**

**Lenguaje:** Tipo de código compuesto por un conjunto de palabras reservadas, códigos especiales, funciones y comandos que representan acciones específicas.

**Librería:** Conjunto de subprogramas utilizados para desarrollar programas. Contiene código "ayuda" y datos que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos.

**Licencias LGPL:** La Licencia Pública General Menor (inglés: Lesser General Public License) es una modificación de la licencia GPL, que posibilita a los desarrolladores utilizar programas bajo la GPL o LGPL sin estar obligados a someter el programa final bajo dichas licencias. Además, LGPL permite la utilización simultánea de software con este tipo de licencia tanto en desarrollos libres como en desarrollos privativos.

**Licencia GPL:** Es una licencia que está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**Log:** Ficheros de registro en los que van anotando los pasos que hace un cierto usuario en el transcurso de una conexión al servidor.

### **M**

**MB (Megabyte):** Unidad de medida de cantidad de datos informáticos.

**Metodología:** Conjunto de procedimientos y técnicas para el desarrollo y documentación de productos software.

**Materia:** Se refiere a una asignatura.

**Memoria RAM:** Siglas de Random Access Memory, traducido al español como Memoria de Acceso Aleatorio. Es un conjunto de Chips que junto con el microprocesador es parte fundamental del ordenador. El tamaño de la RAM determina la rapidez y comodidad de trabajo ante el ordenador, así como el número de programas que se pueden utilizar de forma simultánea.

### **P**

**Proyecto productivo:** Proceso o acción estratégica encaminado a conseguir un objetivo previamente fijado, con unos recursos económicos y temporales limitados, en los que la finalidad es desarrollar una actividad mediante la cual se obtiene un producto (bienes y/o servicios).

**PC:** Ordenador personal o Computadora Personal.

**Plataformas:** Es el basamento, ya sea de hardware o software, sobre el cual un programa puede ejecutarse.

**PDO:** PHP Data Objects o PDO es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

**Prototype:** Es un framework basado en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones Web

**Plug-in:** Programa accesorio que añade capacidad al programa principal.

### **R**

**Rol:** Papel que ejerce una persona en una actividad o proyecto.

**Requerimiento:** Necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.

### **S**

**SIGLA:** Sistema Integral de Gestión de los Laboratorios.

**Sistema:** conjunto de hardware, software y de un soporte humano; que utiliza dispositivos para capturar, almacenar y procesar datos.

**Software libre:** Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

### U

UCI: Universidad de las Ciencias Informáticas.

### W

wxWidgets: Bibliotecas multiplataforma y libres para el desarrollo de interfaces gráficas programadas en lenguaje C++.

W3C: World Wide Web Consortium. Es la organización internacional que define normas y reglas para Internet.