



Universidad de las Ciencias Informáticas
Facultad 15

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

*Título: Desarrollo de una Aplicación Web para la
Gestión de los procesos de Extensión y Residencia de la
Facultad 15*

AUTORES:

Imilsy García Brito

Diana Carrillo Feria

TUTOR:

Ing. Yoandris Espinosa Núñez

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

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

Firma del Autor
Imilsy García Brito

Firma del Autor
Diana Carrillo Feria

Firma del Tutor
Ing. Yoandris Espinosa Núñez

Dedicatoria

Imilsy:

A mis padres y a mis hermanas que son el centro de mi vida y que me han apoyado durante estos 5 años de mi carrera, y en especial a mi abuelita paterna que si estuviera entre nosotros sería la persona que más disfrutaría mi éxito.

A mi novio que lo quiero con la vida.

Diana:

A todas las personas que me rodean y me apoyaron con su granito arena, y en especial a mi mamá, a mi papá y a mi hermana.

Y a mi abuela Gina que será la persona que estará doblemente orgullosa por a ver alcanzado este éxito.

Agradecimientos

Imilsy

Agradecer a mi familia, a los que estuvieron siempre al tanto de mis estudios, de mi bienestar, que se preocuparon siempre por mí: mi tía Margarita, Amparo, Marisol, a mi prima Yuyi que me ayudó mucho en mi estancia aquí en la Habana y a su esposo Rolando.

A Yusneisy, a Charel, a Liset, a mi prima Daisely a todas ellas que más que mis amigas son mis hermanas y las quiero con el alma porque forman parte de mi historia.

A Laura, a Yudisleidy, a Mariam, a Dayana, Ailén que me enseñaron el valor de la amistad y con las cuales compartí momentos muy alegres y que llevaré siempre conmigo.

A mi papá que se ha sacrificado para que yo tenga mi futuro y todo lo que necesito, a mi mamá que es lo más grande que yo tengo en la vida y que ha hecho hasta lo imposible porque yo haga realidad mis metas.

A mi novio que a pesar de la distancia se mantuvo siempre a mi lado dándome el apoyo que muchas veces necesité.

A Yoander y a Diana que supieron mostrarme lo mejor de ellos, me ayudaron, me apoyaron, me aconsejaron en muchos, muchos momentos, amigos incondicionales de los pocos que existen

A Yanedi que me ha apoyado todo el tiempo y gracias a ella fue posible la culminación de este trabajo.

A una persona muy especial que ya no tengo conmigo, a mi abuelita Delcida, que gracias a su recuerdo tuve la fuerza para llegar a donde me encuentro hoy.

A todos los que de una forma u otra influyeron en mi vida.

Diana

A mis padres que siempre estuvieron presentes durante toda mi carrera dándome apoyo y fuerzas incluyendo a mi hermana que siempre confió en mí. Y por ser las personas que más quiero

A mis abuelas porque siempre estuvieron pendiente de todo, inculcándome grandes valores, dándome de forma incondicional cualquier cosa que necesitara. Y en especial a mi abuelo Gaspar que ya no está pero que estaría muy orgulloso de ver lograr mis sueños. Y al resto de la familia que me ayudaron.

A mis vecinos: Yamilka, Julio y Dralene que son como de la familia ya que siempre están en mi casa compartiendo y riendo los buenos y los malos momentos.

A mis amigas: Lucia, Ivette y Dileixis por compartir juntas una temporada además de darme alegría y ánimo para seguir adelante a pesar de la distancia, y principalmente a Wilma que más que una amiga es una hermana que ha luchado conmigo desde el principio en las buenas y en las malas pero siempre con amor y paciencia.

A mi compañera de tesis por ser buena amiga y compañera, que a pesar de sus majaderías, ha tenido paciencia conmigo y ha sabido mantener un espíritu optimista buscándole siempre un lado bueno a los problemas. Además por compartir estos 3 años haciendo mejor la estancia en la universidad, y espero no perder nunca la comunicación a pesar de la distancia.

A Yoander por darme lo mejor de él, y porque siempre estuvo y que estará presente en mi vida, al igual que su familia que me brindó todo su cariño.

A todos en general quiero darle las gracias por ayudarme a crecer como persona y por darme la oportunidad de ser quien soy y de alcanzar una de las tantas metas que una persona se traza en la vida.

Resumen

El presente trabajo es para realizar un estudio de las Metodologías de Desarrollo y Lenguajes de Programación existentes, que más se utilizan en la Universidad de las Ciencias Informáticas, así como analizar las características y ventajas que brindan cada uno, con el fin de desarrollar una aplicación Web que controle todos los procesos de extensión y residencia de la facultad 15. Este software será concebido con el objetivo de facilitar una herramienta lo suficientemente general para la gestión de todos los procesos de extensión y residencia, enfocada a la dirección del vicedecanato de dicha facultad, llevando un control diario de todas las evaluaciones de los estudiantes en la residencia, así como mantener supervisados tanto a los estudiantes que son deportistas como a los del movimiento de artistas aficionados. Para ellos se hace un análisis de cómo se llevan a cabo estos procesos y se exponen luego las características del sistema propuesto.

Ya concluido el trabajo se contará con la informatización de algunas de las principales actividades extracurriculares que se llevan a cabo en el área antes mencionada, contribuyendo así a que la gestión de la información sea más organizada, se tenga un mejor control y planificación de todos los procesos.

Índice

Introducción	9
CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA	12
1.1 Introducción.....	12
1.2 Antecedentes	12
1.3 Sistemas de Gestión de Contenido	13
1.3.1 CMS Joomla.....	14
1.3.2 CMS Drupal.....	15
1.4 Tecnologías asociadas a Drupal	17
1.4.1 Tecnologías del lado del cliente.....	17
1.4.1.1 JavaScript.....	17
1.4.1.2 CSS	17
1.4.1.3 HTML.....	18
1.4.2 Tecnologías del lado del servidor	18
1.4.2.1 PHP	18
1.4.2.2 Apache.....	20
1.4.2.3 Gestor de Bases de Datos	21
1.5 Metodología de Desarrollo de Software	24
Lenguaje Unificado de Modelado (UML)	31
1.5.1 Herramientas de Desarrollo	31
1.5.1.1 Herramienta CASE (Rational)	31
1.6 Conclusiones del capítulo	33
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	34
2.1 Introducción	34
2.2 Descripción del Sistema que se propone.	34
2.3 Modelo del dominio	36
2.3.1 Descripción de los conceptos	37
2.3.2 Diagrama del modelo del dominio.....	38
2.4 Requisitos	38
2.4.1 Requisitos Funcionales (RF).....	39
2.4.2 Requisitos no Funcionales (RNF)	45
2.5 Modelo de Caso de Uso del Sistema	46

Ciudad de la Habana, 2010
“Año del 50 Aniversario del Triunfo de la Revolución”

2.5.1 Diagrama de los casos de uso del sistema	46
2.5.2 Descripción de los actores del sistema	46
2.5.3 Descripción de los casos de uso del sistema	47
2.5.3.1 Especificación del Caso de Uso Autenticar	47
2.5.3.2 Especificación del Caso de Uso Gestionar Cuartelería	48
2.5.3.3 Especificación del Caso de Uso Gestionar Guardia	52
2.5.3.4 Especificación del Caso de Uso Gestionar TSU	55
2.6 Conclusiones del capítulo	59
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	60
3.1. Introducción	60
3.2 .Análisis del sistema.....	60
3.2.1. Diagramas de clases del análisis.....	60
3.3 Diagramas de interacción.....	63
3.4 Diseño del sistema.....	66
3.4.1. Diagramas de clases del diseño	66
3.5. Diseño de la Base de Datos	67
3.6. Conclusiones del capítulo	68
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS	69
4.1 Introducción	69
4.2 Modelo de implementación	69
4.3 Modelo de despliegue	70
4.4 Pruebas	71
4.5 Conclusiones del Capítulo.....	72
CONCLUSIONES	73
RECOMENDACIONES	74
REFERENCIAS BIBLIOGRÁFICAS	75
GLOSARIO DE TÉRMINOS.....	77

Introducción

En Cuba, la informática en los últimos tiempos se ha convertido en parte del sustrato tecnológico del proceso de globalización en el cual está inmerso todo el mundo. Esto implica la necesidad de preparar a las nuevas generaciones para la asimilación y utilización de dicha tecnología.

Son innumerables los beneficios que brinda la computación. Rapidez en la obtención de resultados, almacenamiento de grandes volúmenes de información, facilidades para encontrar información adecuada y actualizada por parte de científicos, investigadores, profesionales, estudiantes; con lo cual se trabaja intensamente en el país para ir incorporando de forma progresiva estos elementos al servicio de toda la población.

La Universidad de las Ciencias Informáticas juega un papel importante en todo el proceso de informatización que se lleva a cabo en el país, incluso cuenta con algunos de los principales servicios informatizados, con los cuales se controla la información de todos los estudiantes y profesores, tanto en la docencia como en la producción.

En la UCI el área de Extensión y Residencia, identificada como una de las áreas de resultados claves en el desarrollo creciente de la universidad, se sustenta sobre la base de la planificación, organización y control de todas las actividades no curriculares realizadas por los estudiantes y trabajadores de la UCI. Actualmente persisten problemas en todo el proceso de la extensión universitaria, tales como bajo nivel de planificación de las actividades, producto a la baja gestión documental existente. Problemas organizativos encausados en el cúmulo de gran cantidad de actividades simultáneas para las cuales se dispone de poco tiempo y recursos, así como la gran magnitud de participantes en las mismas. Todo lo que conlleva a que el proceso de control en toda su magnitud se vea afectado considerablemente.

De forma general en la Universidad de las Ciencias Informáticas, la planificación de todas las actividades en los procesos extensionistas se realiza de forma similar, procesos como la organización de la guardia estudiantil, registros de las actividades productivas (TSU), ubicación en la residencia, planificación y control de la cuartería, desarrollo de

actividades deportivas y culturales, son llevadas a cabo de forma tradicional en la cual interactúan la administración y las organizaciones.






Los procesos de Gestión Extensionista y residencial en la facultad 15, actualmente se hacen de forma tradicional, lo que conlleva a una mala gestión de la información, demoras en la gestión de los procesos, bajo control de las actividades; repercutiendo directamente en los procesos de dirección del área de extensión y residencia en la facultad.

A partir de la **situación problemática** descrita se plantea como **problema científico**: ¿Cómo lograr una mejor planificación, organización y control de todos los procesos llevados a cabo en el área de extensión universitaria y residencial de la facultad 15, de tal forma que permita el manejo óptimo de los recursos humanos?

Siendo el **objeto de estudio** el proceso de desarrollo de software para los procesos de extensión universitaria y como **campo de acción** análisis, diseño e implementación de un conjunto de módulos para la gestión de la información de los procesos de extensión y residencia de la facultad 15.

El **objetivo general** de la investigación es desarrollar una aplicación informática basada en una plataforma Web, que viabilice la gestión de los procesos de planificación, organización y control en el área extensión y residencia de la facultad 15 y que permita el manejo óptimo de los recursos humanos.

Tareas de investigación.

-  Realizar el diseño teórico de la investigación.
-  Realizar un estudio de las herramientas y tecnologías posibles a utilizar.
-  Establecer los requisitos que debe cumplir el sistema.
-  Realizar el análisis, diseño e implementación del sistema.
-  Realizar las pruebas de validación de la aplicación.

Se plantea para este trabajo la siguiente **hipótesis**: si se desarrolla una aplicación informática basada en una plataforma Web que viabilice los procesos para el área extensión y residencia de la facultad 15 se logrará una mejor planificación, organización y control de las actividades y el manejo óptimo de los recursos humanos.

Métodos Teóricos

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

Histórico Lógico

Analizan la trayectoria completa del fenómeno, su condicionamiento a los diferentes períodos de la historia, revela las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales. (1)

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

Con el estudio realizado referente a la existencia de herramientas que posibiliten el total control de los procesos de extensión y residencia en la UCI, como punto de partida en el trabajo, se investigó acerca de la importancia que tiene la existencia de las mismas así como las principales técnicas utilizadas.

Hipotético deductivo.

A partir de la hipótesis y siguiendo reglas lógicas de deducción se llega a nuevos conocimientos y predicciones, las que posteriormente son sometidas a verificaciones. (1)

Método Empírico

Observación

Se observará todo lo relacionado con la existencia de herramientas que permiten automatizar los procesos de Extensión y Residencia con el objetivo de mejorar el flujo de información de los mismos en la facultad 15

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realizará un estudio de las herramientas, metodologías, lenguajes de programación a utilizar para el desarrollo de la aplicación Web, y se tomará como punto de partida las características y ventajas que ofrecen cada uno de ellos y así utilizar las que más se ajusten a las necesidades del sistema a desarrollar. Se investigará además acerca de la existencia de sistemas similares al que se desea implementar.

1.2 Antecedentes

La Universidad de las Ciencias Informáticas cuenta con importantes servicios informatizados, uno de los principales es Akademos, con el cual se gestiona la información relacionada con la docencia de los estudiantes de toda la universidad. También se hace imprescindible la existencia de servicios que controlen la información concerniente al área de Extensión y Residencia.

Existen varias aplicaciones que de una forma u otra gestionan los procesos de la Residencia pero no incluyen algunos procesos de esta área, es decir que estos software tratan el control de la guardia y la cuarterería, o la gestión de los edificios para la ubicación de los estudiantes o bien las evaluaciones de los mismos en la beca; sin embargo hay otras actividades que no están presentes.

Entre los software que existen actualmente se encuentra el de la facultad 6 la cual desarrolló un módulo *Residencia*, en el cual incluyen los procesos de Guardia, Cuarterería, Trabajo Socialmente Útil y otras actividades extracurriculares. Este sistema cuenta con muchas de las funcionalidades que se van a desarrollar en este trabajo, lo mismo sucede en la facultad 8 que después de analizar como funcionan estas actividades, llevaron a cabo su informatización. Existen otras facultades que cuentan con aplicaciones similares a la que se quiere desarrollar en este trabajo como la facultad 7, pero solo cuenta con la automatización de los procesos de Guardia y Cuarterería, no brinda la posibilidad de gestionar las demás actividades que se realizan en el área de extensión y residencia, en fin existe variedad en cuanto a lo antes planteado. La facultad 15 en particular dentro del proceso de informatización incluye los eventos deportivos y

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

culturales, además de la Reunión de Estudios Militares (REM) y debido a que no hay ninguna herramienta que incluya estas actividades en conjunto, se hace necesaria la realización de esta aplicación que abarque todos los procesos.

1.3 Sistemas de Gestión de Contenidos

Realizar una Web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En el pasado las herramientas eran básicamente editores que permitían generar una página, que evolucionaron para incorporar el control de la estructura de la Web y otras funcionalidades, pero en general estaban enfocadas más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de sistema de gestión de contenidos. Se trata de herramientas que permiten crear y mantener una Web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las Web. (2)

Los sistemas gestores de contenidos se utilizan fundamentalmente para que la gestión de la Web sea de una forma mucho más sencilla para los usuarios, por lo que también son conocidos como gestores de contenido Web (Web Content Management o WCM). (3)

El gestor de contenidos es una aplicación informática usada para crear, editar, gestionar y publicar contenido digital en diversos formatos. El gestor de contenidos genera páginas dinámicas interactuando con el servidor para generar la página web bajo petición del usuario, con el formato predefinido y el contenido extraído de la base de datos del servidor. (4)

CMS de código abierto

Los CMS de código fuente abierto, son desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente. (2)

El hecho de que el código fuente esté disponible permite que cualquier persona con conocimiento del tema pueda personalizar este producto, que pueda además en caso de errores corregir los mismos, incluso desarrollar nuevas funcionalidades, permitiendo así la evolución del ya mencionado producto aunque exista o no su creador principal.

1.3.1 CMS Joomla

Un CMS robusto y personalizable. Actualmente Joomla representa más que la existencia de una aplicación Web de avanzada. Es la demostración de la superioridad que puede alcanzar el software libre cuando está presente el esfuerzo común y organizado de una numerosa comunidad global. (3)

Joomla ofrece muchas facilidades, con su utilización solo es necesario ocuparse de lo que se quiera publicar, todo lo demás lo gestionará el sistema, ya sea de índole administrativo o técnico

- **Organización del sitio Web:** Joomla está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel de administración de Joomla se podrá crear, editar y borrar las secciones y categorías del sitio de la manera en que más convenga. (5)
- **Publicación de Contenidos:** permite crear páginas ilimitadas y editarlas desde un sencillo editor, que da la posibilidad de formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.
- **Escalabilidad e implementación de nuevas funcionalidades:** Joomla ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de un sitio Web determinado, por ejemplo: galerías de imágenes, foros, clasificados, entre otros.
- **Administración de usuarios:** Joomla permite almacenar datos de usuarios registrados y también la posibilidad de enviar e-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio. (5)
- **Diseño y aspecto estético del sitio:** es posible cambiar todo el aspecto del sitio Web tan solo con un par de clic, gracias al sistema de plantillas que utiliza.
- **Navegación y menú:** totalmente editables desde el panel administrador.
- **Administrador de Imágenes:** Joomla posee una utilidad para subir imágenes al servidor y usarlas en todo el sitio.

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

- **Encuestas:** posee un sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.

1.3.2 CMS Drupal

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. (6)

A continuación se detallan algunas de las características de Drupal.

Ayuda on-line: un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores. (7)

Búsqueda: todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

Módulos: la comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades tales como: categorización de contenidos, encuestas, chat, foros, entre otros.

Personalización: un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.

URL amigables: tiene la capacidad de mejorar las url de los contenidos y colocarla de una manera más agradable al usuario.

Taxonomía: las categorías se utilizan para organizar el contenido, se implementan mediante vocabularios, términos, y las relaciones que se establecen entre ellos, que pueden ser simples o jerárquicas, incluyendo jerarquías múltiples.

Autenticación de usuarios: permite a los usuarios registrarse e iniciar sesión localmente además mediante la integración de diferentes módulos contribuidos puede realizar autenticación mediante diferentes tipos de servidores tales como; LDAP, jabber u otro.

Permisos basados en roles: los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles.

¿Joomla o Drupal?

Cuando se ha tomado la decisión de utilizar un sistema de gestión de contenidos de código abierto, se puede plantear la interrogante de cuál de los existentes seleccionar para el desarrollo de la aplicación, por tal motivo la elección se sustentará en algunos aspectos de gran importancia que contrastan la diferencia entre ellos.

Necesidad de crear perfiles o roles de usuario

En la aplicación que se desea desarrollar va a ser necesario la creación de distintos grupos de usuario, ya sea para acceder a la información o para la parte de administración del sitio, en este caso sería factible utilizar el CMS Drupal, ya que Joomla tiene una serie de grupos con los permisos predefinidos y que no pueden ser modificados o adaptados para necesidades concretas. (8)

Personalización de contenidos

En Joomla se hace difícil mostrar la información personalizada a los diferentes usuarios registrados en el sistema, ya que internamente identifica a los usuarios de la misma forma, es decir que para Joomla todos los usuarios son iguales, en cambio Drupal permite personalizar la información en dependencia del tipo de usuario, es decir que contenido concretamente podrá ver cada uno de los usuarios.

Existen muchas otras diferencias entre estos dos ejemplares sistemas de contenidos y la mayoría de ellas apoyan a Drupal por ser más completo que Joomla a pesar de que existe discrepancia en cuanto al tema. Después de investigar se puede llegar a la conclusión que la selección de un determinado CMS para la construcción de un sitio más bien se

basaría en el tipo de sitio que se desee crear. En el caso de la aplicación a desarrollar en este trabajo el CMS Drupal es el más indicado.

1.4 Tecnologías asociadas a Drupal

1.4.1 Tecnologías del lado del cliente

Existen una gran variedad de tecnologías para la programación del lado del cliente, a continuación se mencionan algunas de las más utilizadas en la actualidad y se hace énfasis en las características y ventajas las cuales se utilizarán para el desarrollo de este trabajo.

1.4.1.1 JavaScript

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. (9)

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Por su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con JavaScript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. (10)

1.4.1.2 CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) es usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores (11)

Una hoja de estilo permite dar características a las etiquetas HTML utilizadas en una página Web, como colores, tamaños, posiciones, entre otras. Esto facilita la modificación del diseño gráfico de un sitio sin que sea necesario realizar cambios en el código HTML. (12)

1.4.1.3 HTML

HTML es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Marcas Hipertextos. Es un lenguaje de marcación, diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, han permitido que el HTML se convierta en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para Web. (13)

HTML también describe la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), además permite incluir hojas de estilos.

El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas Web, generando un producto vistoso, rápido y sencillo.

1.4.2 Tecnologías del lado del servidor

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor Web para generar páginas HTML dinámicamente como respuesta. (14)

1.4.2.1 PHP

Es el acrónimo de Hypertext Pre-Processor. Actualmente existen diferentes lenguajes de programación para desarrollar en la Web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. (15)

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. (16)

¿Cómo Funciona PHP?

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador, normalmente es una página HTML pero también podría ser una página WML

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP. (16)



Imagen 1: ¿Cómo funciona PHP? (16)

Características

Dispone de una gran cantidad de características que lo convierte en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase MySQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) incluso analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP cuenta con la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

Las principales características de PHP son: la rapidez; la facilidad de aprendizaje; el soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

1.4.2.2 Apache

Apache es un servidor Web que es utilizado por varias personas debido a su robustez y estabilidad. Es multiplataforma, es decir corre sobre una variedad de sistemas operativos lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierto, es altamente configurable. Es muy sencillo ampliar las capacidades del servidor Web Apache, actualmente existen muchos módulos que son adaptables a este. Trabaja con gran cantidad de lenguajes, como Perl, PHP y otros lenguajes de script. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en tu servidor (17)

1.4.2.3 Gestor de Bases de Datos

Un **Sistema Gestión de Bases de Datos** (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. El sistema de gestión de base de datos es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma. (18)

Ventajas del SGBD

- **Mejora en la integridad de datos.** La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.
- **Mejora en la seguridad.** La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean vulnerables. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos.
- **Mejora en la accesibilidad a los datos.** Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.
- **Mejora en el mantenimiento.** En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, el cual simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- **Mejora en los servicios de copias de seguridad y de recuperación ante fallos.** Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

¿Qué es MySQL?

- Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB.
- MySQL es un software de código abierto, licenciado bajo la licencia GNU/GPL, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Características de MySQL

- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- El principal objetivo de MySQL es velocidad y robustez.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.

- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo. (19)

¿Qué es PostgreSQL?

PostgreSQL posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977.

Características de PostgreSQL

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x. (20)

Altamente extensible ya que soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

Implementa el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.

¿Por qué PostgreSQL como gestor de Base Dato?

Al comparar los dos gestores de base datos de código abierto más utilizados, se escoge PostgreSQL ya que sus características se ajustan más a las necesidades del sistema que

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

se quiere desarrollar. Es un SGBD muy potente donde la protección de la integridad de los datos es mucho más fuerte que en MySQL, presenta gran escalabilidad y rendimiento en el procesamiento de datos. Es capaz de ajustarse al número de CPU (Unidad Central de Procesamiento en inglés Central Processing Unit) y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de subconsultas y transacciones para ofrecer una mayor estabilidad, haciendo su funcionamiento mucho más fuerte.

1.5 Metodología de Desarrollo de Software

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos sin embargo han presentado problemas en otros. No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. (José H. Canos)

Todo desarrollo de software es riesgoso y difícil de controlar, sino se lleva una metodología para guiar el proceso, lo que se obtiene entonces son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. En inglés: Software Development Methodology (SDM) o System Development Life Cycle (SDLC).

La finalidad de una metodología de desarrollo es garantizar la eficacia (por ejemplo cumplir los requisitos iniciales) y la eficiencia (por ejemplo minimizar las pérdidas de tiempo) en el proceso de generación de software. (21)

RUP

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso. RUP sigue un modelo iterativo que aborda las tareas más riesgosas primero. Con esto se logra reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente. (23)

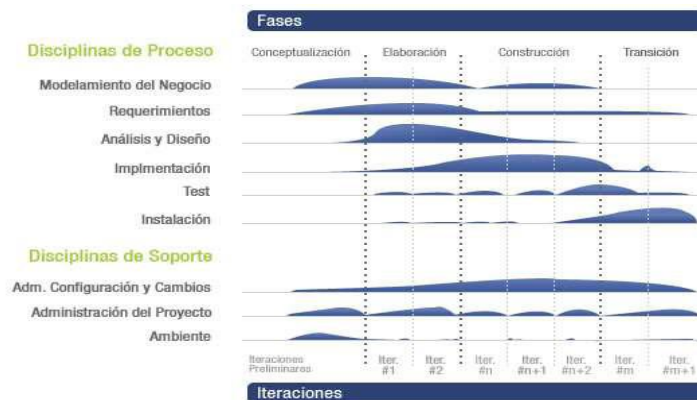


Imagen 2: Fases y flujos de RUP (22)

Características

Tiene una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Pretende implementar las mejores prácticas en Ingeniería de Software, cuenta con un desarrollo iterativo, administración de requisitos, control de cambios, modelado visual del software, además de la verificación de la calidad del software

Administración de requerimientos

RUP describe cómo obtener los requerimientos y organizarlos. Documentar requerimientos de funcionalidad y restricciones, rastrear y documentar decisiones además de captar y comunicar requerimientos del negocio

Arquitecturas basadas en componentes

El proceso se basa en diseñar tempranamente una arquitectura base ejecutable, esta a su vez debe ser flexible, fácil de modificar, intuitivamente comprensible y que promueva la reutilización de componentes

Control de cambios

Los cambios son inevitables, pero es necesario evaluar si éstos son necesarios y rastrear su impacto. RUP indica cómo controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.

Fases (RUP divide el proceso de desarrollo del software en 4 fases)

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas. Se ponen estas versiones a consideración de un subconjunto de usuarios.
- **Transición:** la versión ya está lista para su instalación en las condiciones reales. Puede implicar reparación de errores.

Flujos de trabajo

- **Modelamiento del negocio:** describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** busca los defectos a lo largo del ciclo de vida.
- **Instalación:** se tiene una o varias versiones del producto y se realizan actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

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 y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo,

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

XP

La programación extrema o Xtreme Programming (XP) es un enfoque de la ingeniería de software. Es la más destacada de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Características

- Desarrollo iterativo e incremental
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera, el código es revisado y discutido mientras se escribe, es más importante que la posible pérdida de productividad inmediata.

- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo. La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores

Diferencias entre RUP y XP

Crterios	RUP	XP
Tamaño de los equipos	Está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración.	Se implementan mejor para proyectos cortos y equipos más pequeños

Obtención de requisitos	Se basa en los UseCase (casos de uso) donde se describen los requerimientos de la aplicación desde el punto de vista del usuario.	Se Basa en los UseStories (historias de uso), que al igual que el anterior definen los detalles técnicos sin meterse con los detalles de implementación.
Carga de trabajo	Proceso pesado por estar basado mucho en la documentación, documentación que se ve afectada con los posibles cambios volátiles que a los clientes se les ocurre en cuanto a funcionalidades del software, la justificación de esto es que gracias a su plan de desarrollo con el que se controla el desarrollo se pueden reconocer los problemas y fallos de forma temprana y corregirlos.	Proceso ligero porque no se les asignan roles organizativos al equipo, roles como el modelado o generación de la documentación, esto es reemplazado por la presencia de un representante especializado del cliente, haciendo así más flexibles los posibles cambios que se presenten durante el desarrollo del software.
Relación con el cliente	Al final de cada fase, se le presenta al cliente los artefactos finales de dicha fase, para que sean evaluados por este y se puedan generar las iteraciones necesarias para la siguiente fase.	La comunicación con el cliente es fluida (a través de su representante) después de cada iteración el cliente recibe un pedazo de programa funcional, así el cliente está informado permanentemente y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades.
Desarrollo	Aquí ambos están basados en iteraciones, lo que les permite acercarse poco a poco a la solución sin tener que entrar demasiado rápido a los detalles.	

¿Qué Metodología utilizar?

Después de analizar las características de ambas metodologías se ha decidido utilizar RUP para el desarrollo de la aplicación, una de las razones por la que se selecciona dicha metodología es precisamente porque todos los miembros del equipo comparten una vista

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

de cómo desarrollar el software, o sea permite unificar a todo el equipo y crear un entendimiento común de las tareas a realizar. Al ser RUP una de las metodologías más utilizadas en el mundo, cuenta con una base de conocimiento que ayuda aún más a la unión del equipo, además de que se puede encontrar en línea un sin número de ayuda relacionada con esta metodología.

Otro de los motivos por los cuales se selecciona RUP es que además genera una gran documentación durante todo el proceso de desarrollo de software, documentos que son legibles tanto para el cliente como para los desarrolladores, al final de cada fase se obtienen artefactos los cuales son evaluados por el cliente y junto a toda la documentación generada, pueden ser reutilizados para hacer mejoras en el sistema.

Lenguaje Unificado de Modelado (UML)

Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

Se usa para entender, diseñar, configurar, controlar la información sobre el sistema que se va a construir. El diagrama de clases juega un papel protagónico, este representa una parte importante del sistema, pero solo representa una vista estática, en cambio UML introduce nuevos diagramas que representa una visión dinámica del sistema. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura.

1.5.1 Herramientas de Desarrollo

1.5.1.1 Herramienta CASE (Rational)

Hoy en día, muchas empresas se han extendido a la adquisición de herramientas CASE (Ingeniería Asistida por Computadora), con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio hasta el final

Tecnología Case

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.
- Automatizar:
 - El desarrollo del software
 - La documentación
 - La generación del código
 - El chequeo de errores
 - La gestión del proyecto
- Permitir:
 - La reutilización del software
 - La portabilidad del software
 - La estandarización de la documentación

Rational Rose

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo y da soporte al Proceso Unificado de Rational.

Características

- Capacidad de análisis de calidad de código

- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Integración con otras herramientas de desarrollo de Rational
- Capacidad para integrarse con cualquier sistema de control de versiones incluyendo a Rational ClearCase
- Publicación Web y generación de informes para optimizar la comunicación dentro del equipo

1.6 Conclusiones del capítulo

En el capítulo se hizo referencia a las tecnologías actuales que más se utilizan para el desarrollo de software, por lo que se definen para el diseño y la implementación del sistema algunas de estas tecnologías, metodologías y lenguajes que han sido objeto de estudio y comparaciones: la metodología de desarrollo utilizada es RUP, como lenguaje de modelado UML, la herramienta de modelado visual es el Rational Rose y se hace uso del Sistema Gestor de Contenido (CMS) Drupal. El buen uso de estas tecnologías permitirá obtener un producto con calidad y que cumpla con las funcionalidades establecidas por el cliente, además de que le permita interactuar de manera fácil con la aplicación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se tratan las características que debe tener el sistema, para lo cual se describen los procesos que se van a automatizar y se analiza como se hacen actualmente estos procesos, se definen los requisitos funcionales y los no funcionales con los que debe cumplir la aplicación a desarrollar, además se describen cada uno de los actores del sistema así como la realización del diagrama de caso de uso del sistema.

2.2 Descripción del Sistema que se propone.

Se pretende implementar un sistema que facilite el manejo de toda la información relacionada con los procesos que se realizan en el área de extensión y residencia, el sistema debe cumplir con las exigencias del cliente, el cual es la única persona autorizada de administrar las planificaciones de las actividades así como realizar los cambios que sean necesarios. El sistema debe centrarse en las funcionalidades que se requieren, debe permitir insertar, modificar, así como buscar y guardar la información. A continuación se presenta una breve descripción de los procesos.

Cuartelería

La cuartelería es realizada por todos los estudiantes de lunes a sábado, cada paso de escalera de su respectivo edificio cuenta con un estudiante de cuartelero, el cual es el responsable de controlar todo el personal ajeno que visite el edificio en el horario establecido para la realización de la cuartelería, así como mantener la limpieza del paso de escalera y las áreas alrededor del edificio. Al final de cada día la instructora emite una evaluación (B, M, R) de las cuartelerías realizadas ese día, contribuyendo a la evaluación integral del estudiante.

Guardia estudiantil

El proceso de la guardia estudiantil deben realizarlo todos los estudiantes, el vicedecano de extensión y residencia realiza la planificación de la guardia donde se refleja la brigada que le corresponde realizarla, y el estudiante encargado de hacer la distribución de las postas y de poner el responsable de cada turno, así como de recoger la asistencia a la misma.

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

Al final de cada guardia se elabora un informe donde queda registrado todo lo relacionado a la guardia, el encargado de supervisarla y la asistencia de los estudiantes que la cumplieron.

Trabajo Socialmente Útil

El trabajo socialmente útil surge como una necesidad debido a la falta de personal que existe en la universidad para el cuidado y embellecimiento de las distintas áreas verdes, así como la limpieza de otros lugares como los docentes, entre otros. El TSU en la facultad se realiza en dos sesiones de trabajo (mañana y tarde) en cada una se encuentra una brigada que desarrolla las tareas que le son asignada, cuando se termina el trabajo se genera un reporte del TSU con los datos de la brigada que lo realizó, el lugar, la fecha y las evaluaciones de cada uno de los estudiantes que participó en el mismo.

Juegos Deportivos

Los juegos deportivos se realizan cada año en todas las facultades, siguiendo un orden ascendente, primero los inter-años, después los inter-facultades, las copas, los torneos, tanto en la UCI como fuera de ella. La facultad cuenta con un número determinado de deportistas de los cuales se conoce los deportes en que participa y el rendimiento de cada uno. De los juegos deportivos se registra la fecha de cada juego y los estudiantes que participaron en dicho juego.

Festivales de Artistas Aficionados

Los festivales al igual que los juegos se realizan año tras año en la universidad, se hacen de forma competitiva entre las facultades. En la facultad hay estudiantes con aptitudes para la cultura y la facultad lleva un control de los mismos. De los festivales se registra la fecha y los estudiantes que participaron.

Reunión de Estudios Militares

La Reunión de Estudios Militares (REM), es una actividad que realizan todos los estudiantes en su 4to curso en la Universidad de las Ciencias Informáticas de forma obligatoria, de esta actividad se controla la asistencia de los estudiantes a la misma. En caso de que algún estudiante no realice la REM lo realizaría en su 5to año.

2.3 Modelo del dominio

Cuando se habla en términos de modelo de dominio se debe partir primeramente del concepto de modelo de negocio que es, en el cual se representa el funcionamiento de la organización que se estudia, así como sus relaciones con otras organizaciones asociadas.

¿Cuándo un entorno organizacional es un negocio?

- Flujos de información interconectados.
- Posibilidad de determinar subsistemas.
- No solapamiento de responsabilidades.
- Fácil determinación de responsabilidades: mando, ejecución, recolección, etc.

¿Cuándo un entorno organizacional deja de funcionar como un negocio?

- Flujos de información difusos.
- Imposibilidad de determinar subsistemas.
- Solapamiento de responsabilidades.
- Múltiples responsabilidades.

Según los parámetros antes expuesto y luego de un análisis de los procesos que se van a automatizar se llega a la conclusión de que en el negocio los flujos de información están difusos, es decir el negocio no está bien definido, además de que existen múltiples responsabilidades debido precisamente a que una persona es la encargada de realizar varias tareas, trayendo consigo un exceso de interconexiones y por tal motivo se propone la realización de un modelo de dominio.

El modelo de dominio es un diagrama con los objetos reales que forman parte del sistema que se va a desarrollar y las relaciones que existen entre los mismos, su principal objetivo es comprender los conceptos con los que se trabajan y los que deberá trabajar la futura aplicación.

2.3.1 Descripción de los conceptos

Primeramente se identifican los conceptos a utilizar en el diagrama, los cuales se detallan a continuación.

Vicedecanato: departamento en el cual se controla y dirige todas las actividades extracurriculares, a el pertenece el Vicedecano de extensión y residencia.

Vicedecano: persona encargada de gestionar las actividades que se realizan en la residencia, así como controlar el cumplimiento de las mismas por parte de los estudiantes.

Área de extensión y residencia: área en la que conviven los estudiantes fuera de la docencia y en la cual tienen lugar algunos de los procesos.

Procesos: son aquellas actividades extracurriculares que realizan los estudiantes dentro de la Universidad de las Ciencias Informáticas.

Estudiante: es la persona que realiza las actividades que se llevan a cabo en la residencia.

2.3.2 Diagrama del modelo del dominio

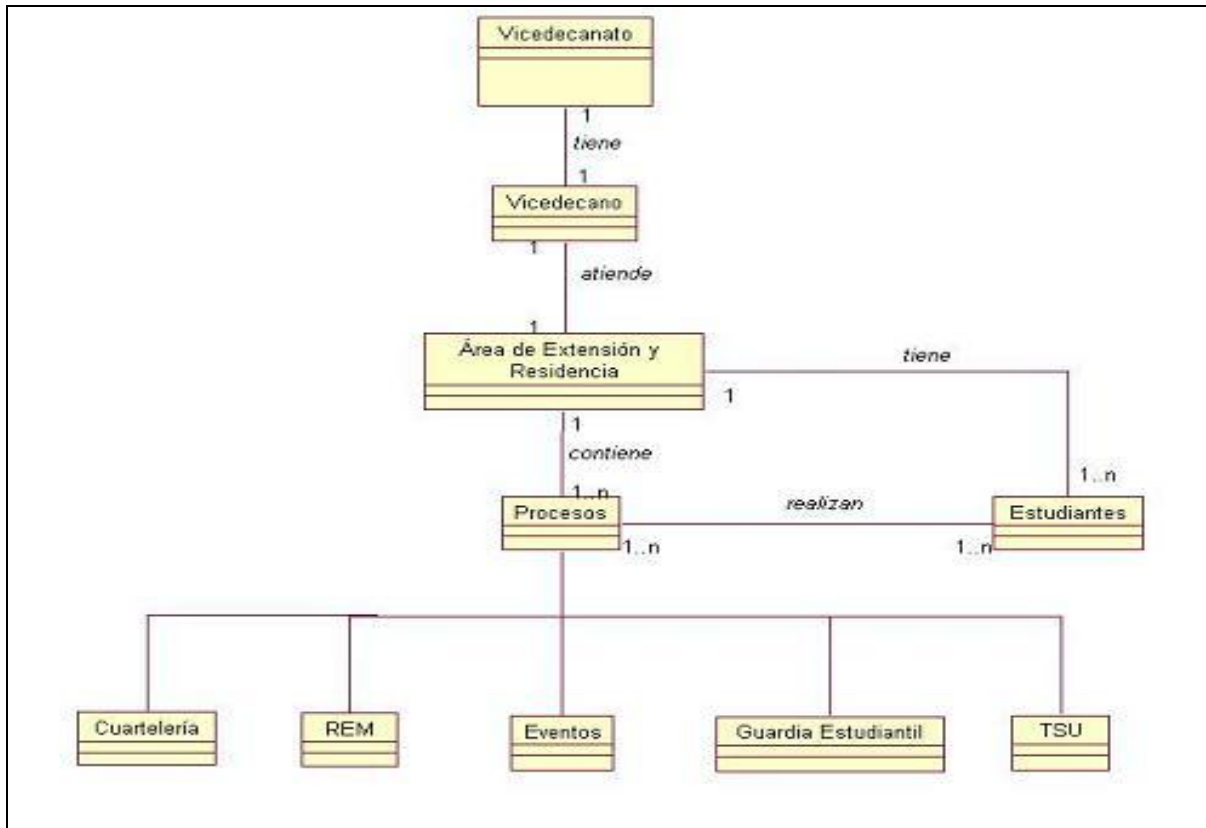


Figura 2.1 Diagrama del Modelo del Dominio

2.4 Requisitos

La especificación de los requerimientos del software es la descripción de lo que debe hacer el sistema que se va a desarrollar, se divide en dos grupos, los requisitos funcionales los cuales son agrupados en casos de uso y los requisitos no funcionales que son propiedades o cualidades que el software debe cumplir, pueden ser restricciones en la implementación, rendimiento, entre otras.

A continuación se muestran los requisitos funcionales con los que cuenta el sistema, los mismos serán identificados de la siguiente forma: RF_N. la n significa el número del requisito y clasificados según su prioridad en Alta (Esencial), Media (Deseado) o Baja (Opcional).

2.4.1 Requisitos Funcionales (RF)

RF_1 Autenticarse

El sistema permitirá que el usuario se autentique especificando los siguientes datos:

- Usuario
- Contraseña

Una vez autenticado permitirá cerrar sesión seleccionando la opción:

- Terminar sesión

Prioridad: Alta

RF_2 Buscar estudiantes.

El sistema permitirá buscar estudiantes especificando cualquiera de los siguientes datos por:

- Grupo.
- Edificio.
- Apartamento.
- Evaluación.

Prioridad: Media

RF_3 Adicionar cuarterería

El sistema permitirá adicionar una cuarterería mostrando los siguientes datos:

- Edificio
- Apartamento

Prioridad: Alta

RF_4 Modificar cuarterería

El sistema permitirá modificar una cuarterería especificando los siguientes datos a modificar:

- Fecha
- Evaluación Áreas y Escaleras
- Evaluación Final

Prioridad: Media

RF_5 Eliminar cuartería.

El sistema permitirá eliminar una cuartería seleccionando la opción eliminar y luego mostrará un mensaje de confirmación.

Prioridad: Baja

RF_6 Mostrar listado de cuartería.

El sistema permitirá listar las cuartería que han sido registradas en la base de datos y permitirá filtrar especificando los siguientes datos:

- Fecha
- Apartamento
- Evaluación

Prioridad: Media

RF_7 Adicionar guardia estudiantil.

El sistema permitirá adicionar una guardia mostrando los siguientes datos:

- Grupo.
- Fecha.

Prioridad: Alta

RF_8 Mostrar Listado de la guardia estudiantil.

El sistema permitirá listar las guardias que han sido registradas en la base de datos y permitirá filtrar especificando los siguientes datos:

- Fecha
- Grupo

Prioridad: Media

RF_9 Modificar guardia estudiantil.

El sistema permitirá modificar una guardia especificando los siguientes datos a modificar:

- Fecha
- Grupo

Prioridad: Media

RF_10 Eliminar guardia estudiantil.

El sistema permitirá eliminar una guardia seleccionando la opción eliminar y luego mostrará un mensaje de confirmación.

Prioridad: Baja

RF_11 Adicionar TSU

El sistema permitirá adicionar un TSU mostrando los siguientes datos:

- Área
- Fecha “Desde”
- Fecha “Hasta”

Prioridad: Alta

RF_12 Asignar TSU

El sistema permitirá asignar un TSU luego de haber seleccionado un grupo determinado, los datos a especificar son:

- TSU
- Fecha
- Área
- Asistencia
- Evaluación

Prioridad: Alta

RF_13 Modificar TSU

El sistema permitirá modificar una guardia especificando los siguientes datos a modificar:

- TSU.
- Fecha.
- Área.
- Asistencia.
- Evaluación.

Prioridad: Media

RF_14 Listar TSU

El sistema permitirá listar los TSU registrados en la base de datos especificando los siguientes datos:

- Fecha
- TSU
- Área

Prioridad: Media

RF_15 Eliminar TSU

El sistema permitirá eliminar un TSU seleccionando la opción eliminar y luego mostrará un mensaje de confirmación.

Prioridad: Baja

RF_16 Adicionar REM

El sistema permitirá adicionar una nueva REM especificando los siguientes datos:

- Nombre de la REM
- Fecha

Prioridad: Alta

RF_17 Listar REM

El sistema permitirá listar las REM registradas en la base de dato, pudiendo especificar algunos de los siguientes filtros:

- Fecha
- Nombre de la REM

Prioridad: Media

RF_18 Registrar asistencia de los estudiantes a la REM

Luego de haber buscado a los estudiantes especificando el grupo, el sistema permitirá registrar la asistencia llenando los siguientes datos:

- Nombre de la REM
- Fecha_Asistencia
- Asistencia

Prioridad: Alta

RF_19 Adicionar Actividad Cultural

El sistema permitirá adicionar una nueva actividad cultural especificando los datos siguientes:

- Categoría
- Año

Prioridad: Alta

RF_20 Registrar Participación en FAA

Luego de haber buscado a los estudiantes especificando el grupo, el sistema permitirá registrar la participación de los estudiantes en el festival llenando los siguientes datos:

- Festival de AA

*Ciudad de la Habana, 2010
"Año del 50 Aniversario del Triunfo de la Revolución"*

- Manifestación Cultural

Prioridad: Alta

RF_21 Listar Participación en FAA

El sistema permitirá mostrar un listado con los estudiantes que participaron en los festivales de artistas aficionados

Prioridad: Media

RF_22 Adicionar Juego Deportivo (JD)

El sistema permitirá adicionar un juego deportivo a la base de datos especificando los siguientes datos:

- Evento
- Nivel

Prioridad: Alta

RF_23 Registrar Participación en JD

Luego de haber buscado a los estudiantes especificando el grupo, el sistema permitirá registrar la participación de los estudiantes en los juegos llenando los siguientes datos:

- Evento Deportivo
- Categoría Deportiva

Prioridad: Alta

RF_24 Listar Deportistas

El sistema permitirá mostrar un listado de los estudiantes deportistas con los que cuenta la facultad 15.

Prioridad: Media

2.4.2 Requisitos no Funcionales (RNF)

Usabilidad

- Portabilidad: permitir que el sistema se ejecute sobre el Sistema Operativo Windows o Linux además de que podrá usar una serie de SGBD como PostgreSQL y MySQL.

Seguridad

- El sistema permite que cada usuario que se autentique sólo tenga acceso a la información que le corresponde de acuerdo a su rol.
- Integridad: la información que sea manejada por el sistema debe ser protegida para evitar su mal uso.

Rendimiento

- El Sistema debe soportar conexiones simultáneas.
- El sistema debe ser eficaz a la hora de atender solicitudes, o sea cuando el cliente desee realizar cualquier acción en el sistema, este debe ser capaz de dar respuesta en el menor tiempo posible.

Confidencialidad

- Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización.
- La autenticación será la primera acción del usuario en el Sistema.
- Si el usuario no está autorizado el sistema muestra un error de acceso.

Disponibilidad

- La información estará disponible para los usuarios autorizados, cuando estos accedan a dicha información.

Interfaz Externa.

- El diseño de la interfaz debe ser fácil y sencillo de entender. Los colores suaves y la información legible.

- La navegabilidad debe ser simple.

Software

- Del lado del Servidor, Servidor Apache 2.0. Gestor de Base Datos Postgres 8.2 .PHP 5. CMS Drupal.
- Del lado del Cliente Sistema Operativo Windows XP o Superior Internet Explorer 6.0 .Mozilla Firefox 2.0 o Superior.

2.5 Modelo de Caso de Uso del Sistema

2.5.1 Diagrama de los casos de uso del sistema

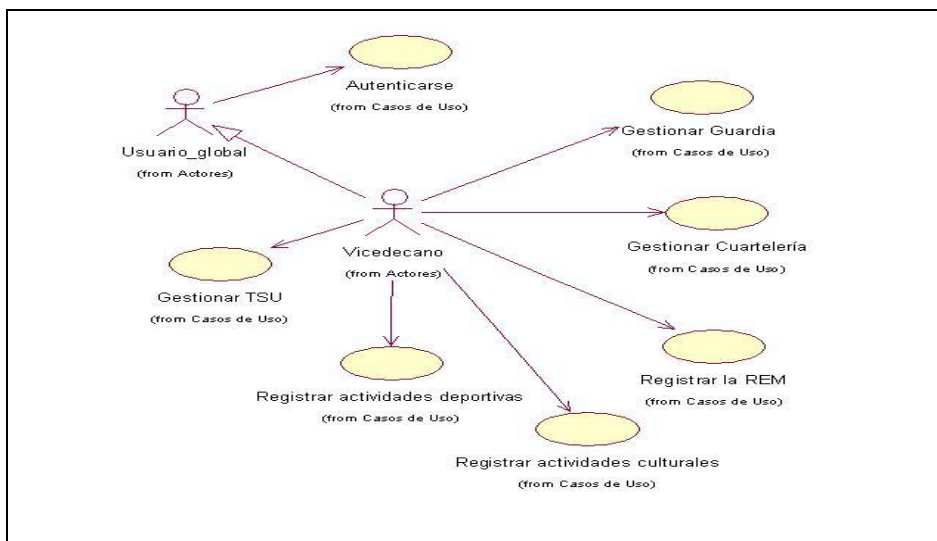


Figura 2.2 Diagrama de Caso de uso del sistema

2.5.2 Descripción de los actores del sistema

Nombre del actor	Justificación
Vicedecano	Es el encargado de controlar la información de los estudiantes en la Residencia, también controlar las evaluaciones de todas las actividades extracurriculares, por lo que es la persona con acceso a todas las funcionales que ofrece el sistema

Usuario	Personas que van a tener algún rol en el sistema, es decir van a poder trabajar con la aplicación. Aquí se incluye al estudiante encargado de supervisar el cumplimiento de la cuartería.
---------	---

2.5.3 Descripción de los casos de uso del sistema

2.5.3.1 Especificación del Caso de Uso Autenticar

CU	<i>Autenticar</i>	
Actores	Usuario	
Resumen	El caso de uso se inicia cuando el usuario le pide al sistema identificarse para ser uso del sistema, según el rol que se le haya asignado a cada uno, además de cerrar su sesión de trabajo, y termina el CUS.	
Precondición		
Poscondición	<ol style="list-style-type: none"> 1. El sistema le da acceso al usuario a los recursos que le son permitidos según su rol. 2. El sistema cierra la sesión de trabajo del usuario autenticado. 	
Referencias	RF_1	
Prioridad	Crítico	
Flujo Normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario desea acceder al sistema.	1.1 El sistema muestra el formulario para ingresar los datos necesarios para ser identificado y otorgar o no el acceso al mismo.	

2. El usuario introduce los datos solicitados por el sistema y presiona el botón “Iniciar sesión”.	2.1 El sistema verifica los datos introducidos por el usuario.
	2.2 Si los datos se introducen correctamente, el sistema le da acceso al mismo, a los recursos que le son permitidos, según los permisos asignados y termina el CUS.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.3 Si los datos introducidos por el usuario no son correctos, el sistema muestra un mensaje de error.
Escenario1: “Terminar Sesión”	
Acción del Actor	Respuesta del Sistema
1. El usuario decide cerrar su sesión de trabajo y presiona el botón: “Terminar Sesión”.	1.1 El sistema cancela todas las acciones que estaba ejecutando el usuario y lo redirecciona para la página de Autenticación y termina el CUS.

2.5.3.2 Especificación del Caso de Uso Gestionar Cuartelería

CU1	<i>Gestionar Cuartelería</i>
Actores	Vicedecano de Extensión y Residencia.
Resumen	El caso de uso se inicia cuando el Vicedecano escoge la opción Gestionar Cuartelería. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.

Precondición	<p>. El sistema debe estar instalado y ejecutándose correctamente.</p> <p>. El usuario debe estar autenticado con los permisos necesarios.</p>
Poscondición	Según la acción realizada el sistema muestra los datos correspondientes.
Referencias	RF_1,RF_2, RF_3,RF_4, RF_5, RF_6
Prioridad	Crítico
Complejidad	Complejo
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. El Vicedecano selecciona la opción Gestionar Cuartelería.	2. El sistema permitirá: <ul style="list-style-type: none"> • Adicionar cuartelería(Ver Sección "Adicionar Cuartelería") • Listar cuartelería (Ver Sección "Listar Cuartelerías").
3. El Vicedecano elige una opción	
Sección: "Adicionar Cuartelería"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
4. Selecciona la opción Adicionar Cuartelería.	5. El sistema da la opción de buscar a un estudiante por: <ul style="list-style-type: none"> • Edificio. • Apartamento.
6. Escoge una de las opciones y da Aplicar (Apply).	7. Valida los datos.

	8. Muestra el listado de estudiantes permitiendo registrar la cuarteraría.
9. Registra la cuarteraría a los estudiantes	10. Adiciona la cuarteraría a la Base de Datos, muestra un mensaje informando que la operación fue realizada con éxito y termina el CUS.
Sección: "Listar Cuarterías"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
11. Selecciona la opción Listar Cuarteraría.	12. El sistema muestra un listado de las Cuarterías registradas y permite además buscar a un estudiante por. <ul style="list-style-type: none"> • Fecha • Apartamento • Evaluación
13. Selecciona el criterio de búsqueda y presiona el botón Aplicar (Apply).	14. Verifica si existe alguna cuarteraría de acuerdo al criterio de búsqueda seleccionado.
	15. Muestra un listado de las cuarterías que coinciden con el criterio de búsqueda seleccionado.
	16. Una vez listadas las cuarterías brinda la posibilidad de : <ul style="list-style-type: none"> • Modificar Cuarteraría (Ver sección "Modificar Cuarteraría"). • Eliminar Cuarteraría (Ver sección "Eliminar Cuarteraría") El CUS termina.
Flujo Alterno	

Acción del Actor	Respuesta del Sistema
13. a. Introduce datos al criterio de búsqueda donde no existen cuarterías en la BD con ese criterio de búsqueda seleccionado.	14.1 Muestra un mensaje dando a conocer que no existe ninguna cuartería que corresponda con el criterio de búsqueda seleccionado.
Sección: "Modificar Cuartería"	
Acción del Actor	Respuesta del Sistema
9. Selecciona la opción Modificar Cuartería.	10. Muestra los datos de la Cuartería a modificar: <ul style="list-style-type: none"> • Fecha • Evaluación Áreas y Escaleras • Evaluación Final
11. Selecciona los datos que desea modificar y la opción Guardar.	13. Actualiza los datos de la Cuartería en la base de datos. Muestra un mensaje de que la cuartería ha sido actualizada y el caso de uso termina.
Sección: "Eliminar Cuartería"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
14. Selecciona la opción Eliminar Cuartería.	15. Muestra un mensaje de confirmación de la opción seleccionada.
16. Selecciona la opción Eliminar.	17. Elimina los datos de la cuartería. Muestra un mensaje de que la cuartería ha sido borrada y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
16. a. No desea eliminar la cuartería selecciona la opción Cancelar.	17.1 No se realiza ninguna operación.

2.5.3.3 Especificación del Caso de Uso Gestionar Guardia

CU1	<i>Gestionar Guardia</i>	
Actores	Vicedecano de Extensión y Residencia	
Resumen	El caso de uso se inicia cuando el Vicedecano escoge la opción Gestionar Guardia. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.	
Precondición	.El sistema debe estar instalado y ejecutándose correctamente. .El usuario debe estar autenticado con los permisos necesarios.	
Poscondición	Según la acción realizada el sistema muestra los datos correspondientes.	
Referencias	RF_1, RF_2, RF_7, RF_8, RF_9,RF_10	
Prioridad	Crítico	
Complejidad	Complejo	
Flujo Normal de eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Guardia.	2. El sistema permitirá: <ul style="list-style-type: none"> • Adicionar Guardia estudiantil.(Ver Sección “Adicionar Guardia estudiantil”) • Mostrar Listado de Guardias (Ver Sección “Mostrar Listado de Guardias”).
	3. El Vicedecano elige la opción.	
Sección: “Adicionar Guardia estudiantil”		

Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
4. Selecciona la opción Adicionar Guardia.	5.El sistema muestra los siguientes datos a seleccionar: <ul style="list-style-type: none"> • Grupo. • Fecha.
6. Selecciona el grupo y le asigna una fecha.	7. Adiciona la Guardia a la base de datos y termina el CU.
Sección: “Mostrar Listado de Guardias”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
8. Selecciona la opción Mostrar Listado de Guardias.	9. El sistema muestra un listado de las guardias registradas y permite además buscar a una Guardia por. <ul style="list-style-type: none"> • Fecha • Grupo
10. Selecciona el criterio de búsqueda y presiona el botón Aplicar (Apply).	11. Verifica si existen guardias de acuerdo al criterio de búsqueda seleccionado.
	12. Muestra un listado de las guardias que coinciden con el criterio de búsqueda seleccionado.
	13. Una vez listadas las guardias que corresponden con el criterio de búsqueda brinda la posibilidad de : Modificar Guardia estudiantil (Ver Sección “Modificar Guardia estudiantil”). Eliminar Guardia estudiantil (Ver Sección

	“Eliminar Guardia estudiantil”). El CUS termina.
Sección : “Modificar Guardia estudiantil”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
14. Selecciona la opción Modificar.	15. Muestra los datos de la guardia. <ul style="list-style-type: none"> • Fecha. • Grupo.
16. Realiza los cambios necesarios a la guardia correctamente y presiona el botón Guardar.	17. Actualiza los datos de la Guardia en la base de datos. Muestra un mensaje de que la Guardia ha sido actualizada.
	18. Guarda los datos de la Guardia en la Base de Datos y termina el CU.
Sección : “Eliminar Guardia estudiantil”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
19. Selecciona la opción eliminar.	20. El sistema muestra un mensaje de confirmación para saber si es la Guardia que se desea eliminar.
21. Selecciona la opción eliminar.	22. Elimina los datos de la Guardia. Muestra un mensaje de que la Guardia ha sido borrada y termina el caso de uso
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
21. a. No desea eliminar la cuartelería selecciona la opción Cancelar.	22.1 No se realiza ninguna operación.

2.5.3.4 Especificación del Caso de Uso Gestionar TSU

CU	<i>Gestionar TSU</i>	
Actores	Vicedecano de Extensión y Residencia	
Resumen	El caso de uso se inicia cuando el Vicedecano escoge la opción Gestionar TSU. Seguidamente el Actor selecciona la acción a realizar e introduce los datos necesarios. El sistema realiza la acción seleccionada por el Actor y termina el caso de uso.	
Precondición	.El sistema debe estar instalado y ejecutándose correctamente. . El usuario debe estar autenticado con los permisos necesarios.	
Poscondición	Según la acción realizada el sistema muestra los datos gestionados.	
Referencias	RF_1, RF_2, RF_11, RF_12, RF_13, RF_14, RF_15	
Prioridad	Crítico	
Complejidad	Complejo	
Flujo Normal de eventos		
	Acción del Actor	Respuesta del Sistema
	1. El Vicedecano selecciona la opción Gestionar TSU.	2. El sistema muestra las siguiente opciones: Adicionar TSU (Ver Sección “ Adicionar TSU ”). Listar TSU (Ver Sección “ Listar TSU ”). Asignar TSU(Ver Sección “ Asignar TSU ”)
	3. Elige una opción.	
Sección : “ Adicionar TSU ”		

Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
4. Selecciona la opción Adicionar TSU.	5. El sistema muestra los siguientes datos para crear el TSU: <ul style="list-style-type: none"> • Área • Fecha "Desde" • Fecha "Hasta"
6. Selecciona el Área y las Fechas de forma correcta para crear el TSU y presiona el botón Guardar	7. Valida los datos. Muestra el mensaje de información del TSU que se ha creado
	8. Adiciona el TSU a la base de datos y el CUS termina
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
6. a. Selecciona los datos de manera incorrecta.	8.1 En caso de faltar algún campo por llenar o por completar, o datos incorrectos, se muestra un mensaje de advertencia.
Sección : " Listar TSU"	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
10. Selecciona la opción Listar TSU.	11. Muestra los siguientes parámetros para realizar el criterio de búsqueda: <ul style="list-style-type: none"> • Fecha • TSU • Área
12. Selecciona el criterio de búsqueda y presiona el botón Aplicar (Apply).	13. Muestra el listado del TSU.

	<p>14. Una vez listados los TSU que corresponden con el criterio de búsqueda brinda la posibilidad de :</p> <p>Modificar TSU (Ver Sección “Modificar TSU”).</p> <p>Eliminar TSU (Ver Sección “Eliminar TSU”).</p> <p>El CUS termina.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
12. a. Selecciona datos al criterio de búsqueda con los que no existen TSU en la base de datos.	13.1 Muestra en un mensaje que no existe ningún TSU para el criterio de búsqueda seleccionado.
Sección : “ Modificar TSU”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
15. Selecciona la opción Modificar.	<p>16. Muestra los datos siguientes del TSU listos para ser modificados.</p> <ul style="list-style-type: none"> • TSU. • Fecha. • Área. • Asistencia. • Evaluación.
17. Modifica los datos de su interés y selecciona la opción Guardar.	18. El sistema valida los datos.
	19. Actualiza los datos del TSU en la Base de Datos, muestra un mensaje dando a conocer que la operación fue

	realizada con éxito y el CUS termina.
Sección : “ Eliminar TSU”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
20. Selecciona la opción Eliminar.	21. Muestra un mensaje de confirmación de la opción seleccionada
22. Selecciona la opción Eliminar	23. Elimina los datos del TSU. Muestra un mensaje dando a conocer que el TSU ha sido borrado.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
22. a. No desea eliminar el TSU seleccionado la opción Cancelar.	23. a.1. No se realiza ninguna operación.
	23. a.2. El caso de uso termina.
Sección :”Asignar TSU”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
24. Selecciona la opción Asignar TSU	25. El sistema muestra el siguiente filtro: <ul style="list-style-type: none"> • Grupo
26. Selecciona el grupo y presiona el botón Aplicar (Apply).	27. El sistema muestra el listado de estudiantes a los que se le quiere registrar el TSU.
28. Selecciona la opción Registrar TSU.	29. El sistema muestra los siguientes datos: <ul style="list-style-type: none"> • TSU • Fecha • Área • Asistencia

	<ul style="list-style-type: none"> • Evaluación
30. Selecciona los datos necesarios y presiona el botón Guardar	31. El sistema guarda los datos y muestra un mensaje dando ha conocer que el TSU ha sido creado correctamente.

2.6 Conclusiones del capítulo

En este capítulo se abordan aspectos de gran importancia, pues partiendo de la comprensión de los procesos a automatizar se definieron las principales funcionalidades del sistema para así satisfacer las necesidades del cliente, para ello:

- Se describieron las principales características del sistema a desarrollar.
- Se describieron los procesos que se van a automatizar.
- Se identificaron los requisitos tanto funcionales como no funcionales del sistema.
- Fueron agrupados en casos de uso los requisitos funcionales especificados y se describieron los casos de usos más importantes.

Luego de la realización de las tareas expuestas anteriormente, se puede obtener una visión general de lo que el sistema debe hacer, por lo que ya se pueden realizar las operaciones descritas en este capítulo, para dar solución a los problemas planteados.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En este capítulo se verán los diagramas de clases del análisis con el objetivo de obtener un mejor entendimiento en cuanto a los procesos relacionados con el sistema. Se realiza un modelo del diseño para explicar los artefactos que intervienen en la elaboración de la aplicación. El propósito principal es lograr una vista más detallada de la aplicación en relación a la propuesta final del sistema así como obtener una implementación que satisfaga las necesidades de los usuarios.

3.2 .Análisis del sistema

En el análisis, se representan las relaciones entre las clases, se centran en los requisitos funcionales dándole un mayor entendimiento de lo que se quiere lograr. Están compuestos por clases de tres tipos fundamentales:

Clase Interfaz: se utiliza para modelar la interacción de entre los actores y el sistema, permitiendo recibir o presentar información hacia los actores. Representa abstracciones de ventanas, formularios, paneles entre otros.

Clase Controladora: las clases de control coordinan el trabajo de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

Clase Entidad: estas son las clases encargadas de modelar la información que posee larga vida y que es a menudo persistente.

3.2.1. Diagramas de clases del análisis

El diagrama de clases del análisis es un artefacto del flujo de Análisis y Diseño que muestra la interacción de sus clases para cada caso de uso, dando una mayor comprensión y representan conceptos y relaciones del dominio.

Autenticar

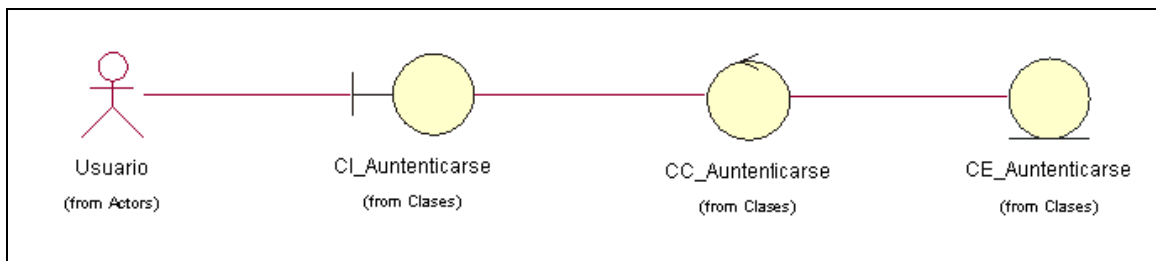


Figura 3.1 Diagrama de clases del Análisis del caso de uso Autenticar

Gestionar Cuartelería

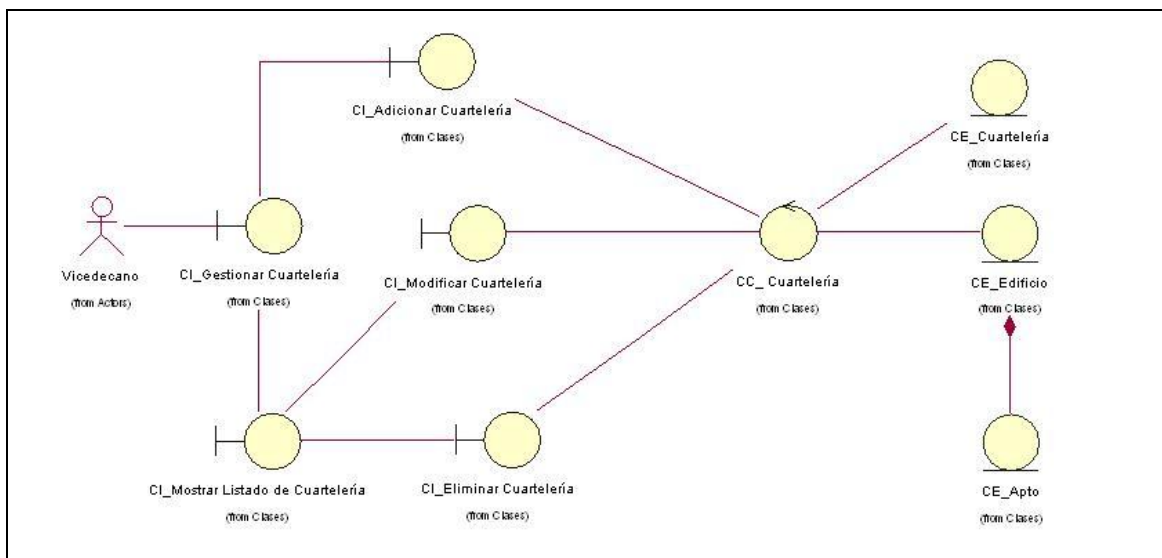


Figura 3.2 Diagrama de clases del Análisis del caso de uso Gestionar Cuartelería

Gestionar Guardia

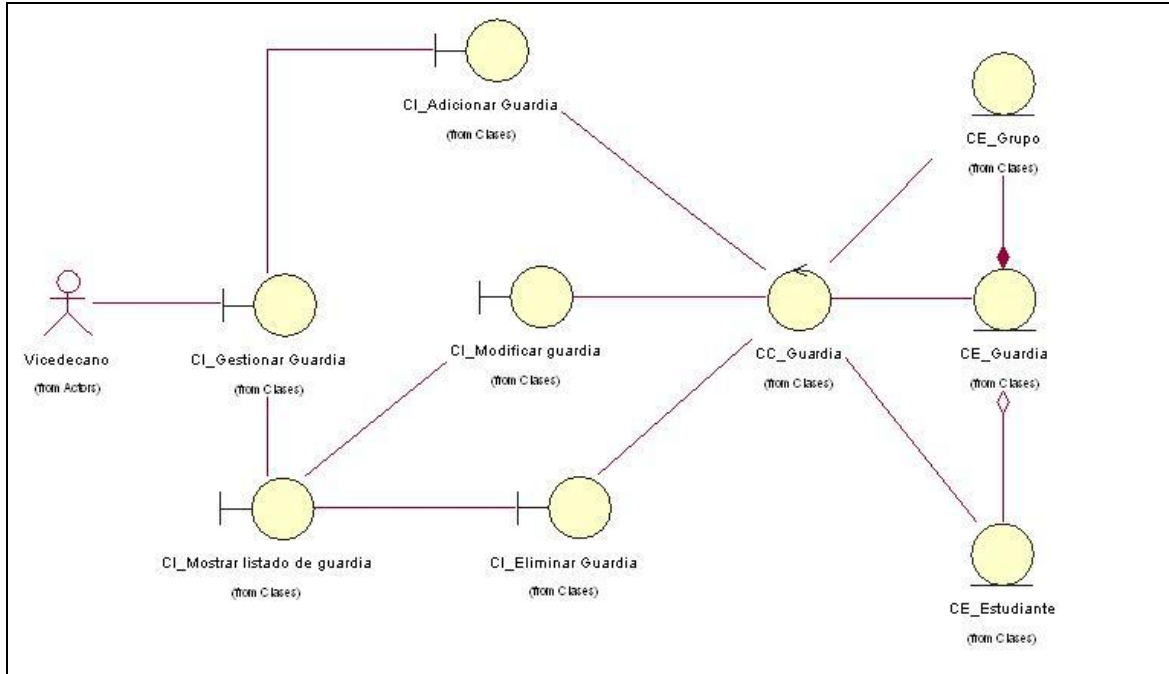


Figura 3.3 Diagrama de clases del Análisis del caso de uso Gestionar Guardia

Gestionar TSU

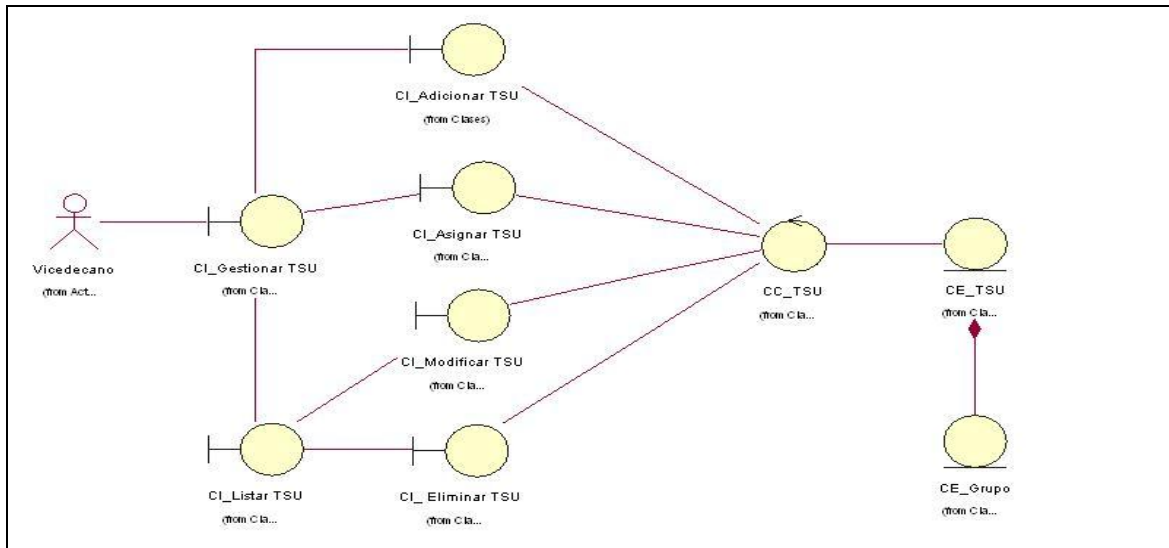


Figura 3.4 Diagrama de clases del Análisis del caso de uso Gestionar TSU

3.3 Diagramas de interacción

El diagrama de interacción, representa la forma de cómo se comunican entre sí el cliente y las clases con la petición de un evento. Esto significa recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades visiblemente. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso. El UML define dos tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje: diagramas de colaboración y diagramas de secuencia.

Las interacciones entre los objetos y las clases, de la solución propuesta como resultado del diseño del software, fueron modeladas a través de diagramas de colaboración.

A continuación algunos ejemplos.

3.3.1 Diagramas de Colaboración del Análisis.

3.3.1.1. Autenticar

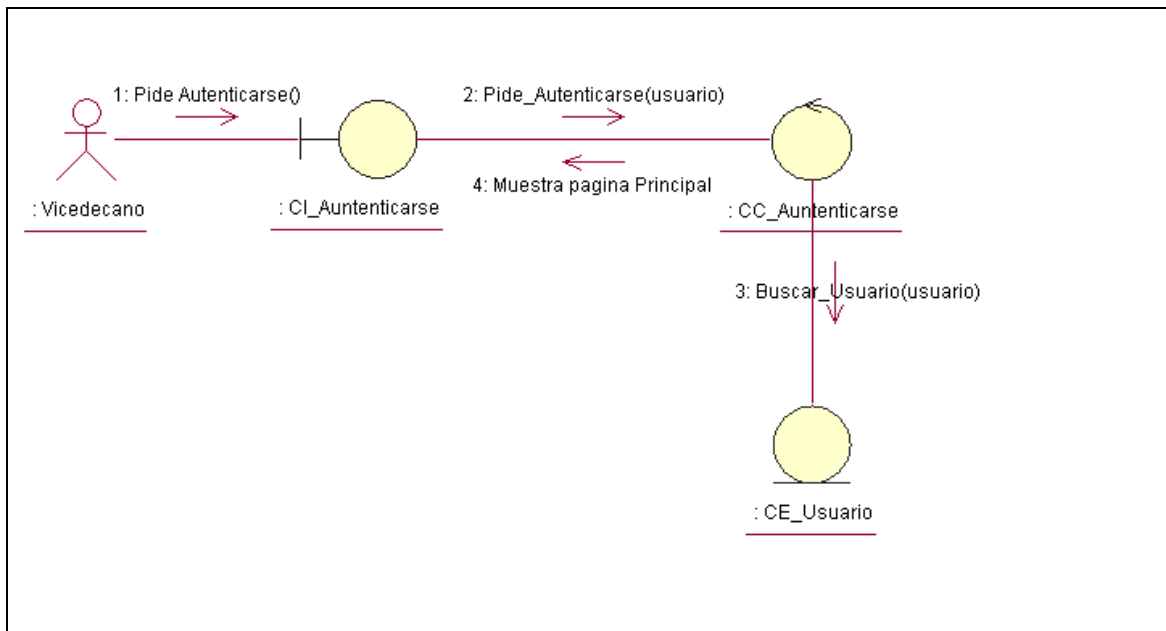


Figura 3.5 Diagramas de Colaboración para el CU Autenticar

3.3.1.2. Gestionar Cuartelería

Adicionar Cuartelería

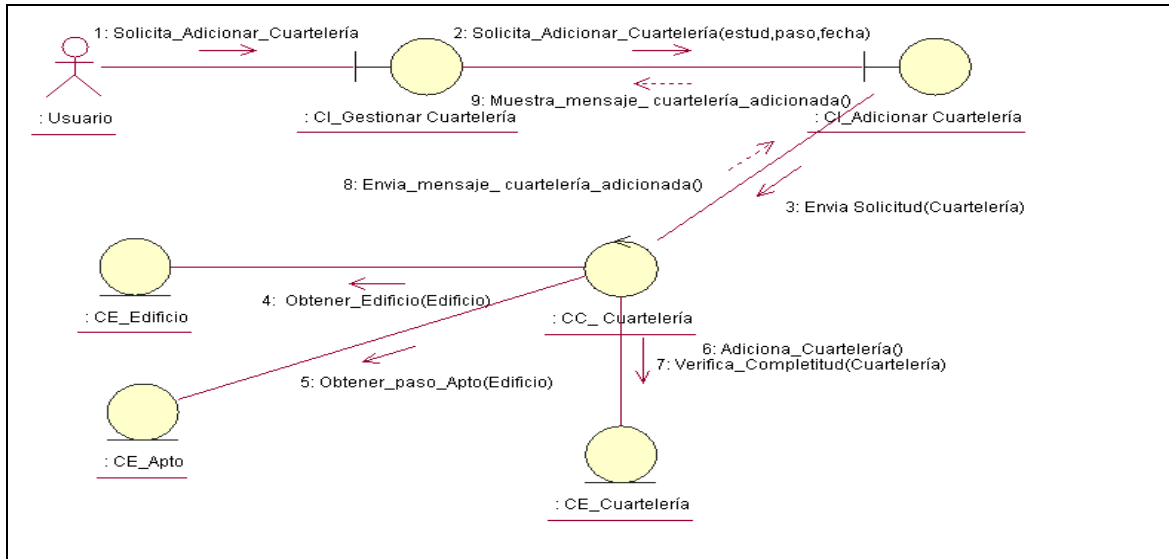


Figura 3.6 Diagramas de Colaboración para el escenario Adicionar Cuartelería

3.3.1.3. Gestionar Guardia

Adicionar Guardia

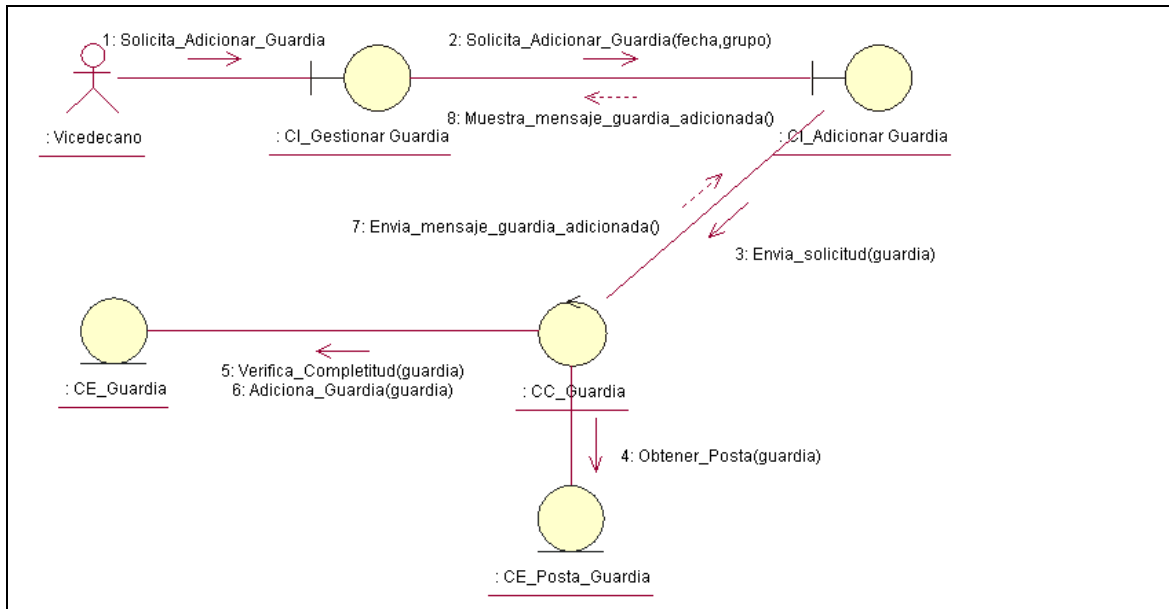


Figura 3.8 Diagramas de Colaboración para el escenario Adicionar Guardia

Mostrar Listado

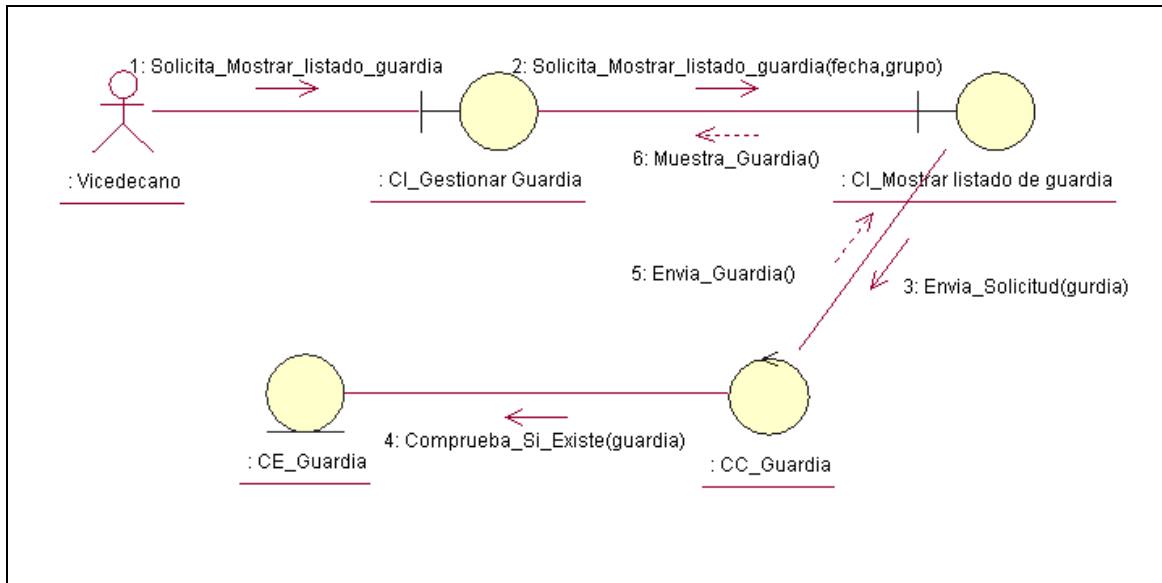


Figura 3.9 Diagramas de Colaboración para el escenario Mostrar Listado

3.3.1.4. Gestionar TSU

Eliminar TSU

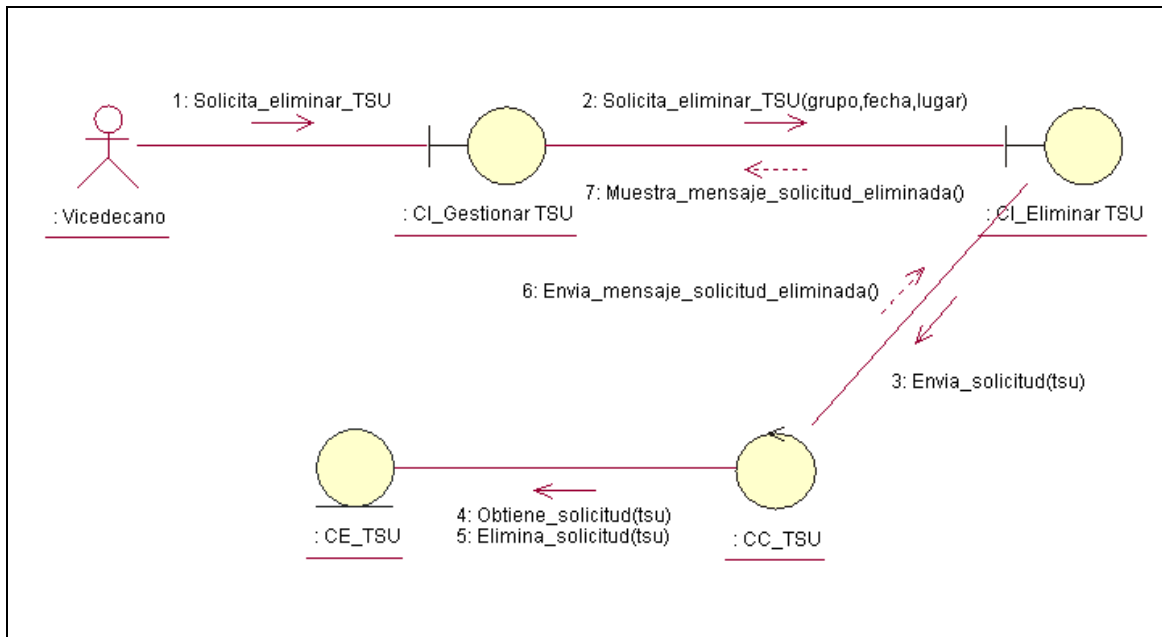


Figura 3.10 Diagramas de Colaboración para el escenario Eliminar TSU

3.4 Diseño del sistema

Con el diseño se representan las relaciones físicas entre los objetos del sistema, especificando los atributos y métodos que se relacionan con los de la implementación, la navegabilidad y la dependencia que existe entre ellas. En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos funcionales y los no funcionales.

3.4.1. Diagramas de clases del diseño

A continuación se muestran algunos ejemplos de estos diagramas

3.4.1.2 Autenticar

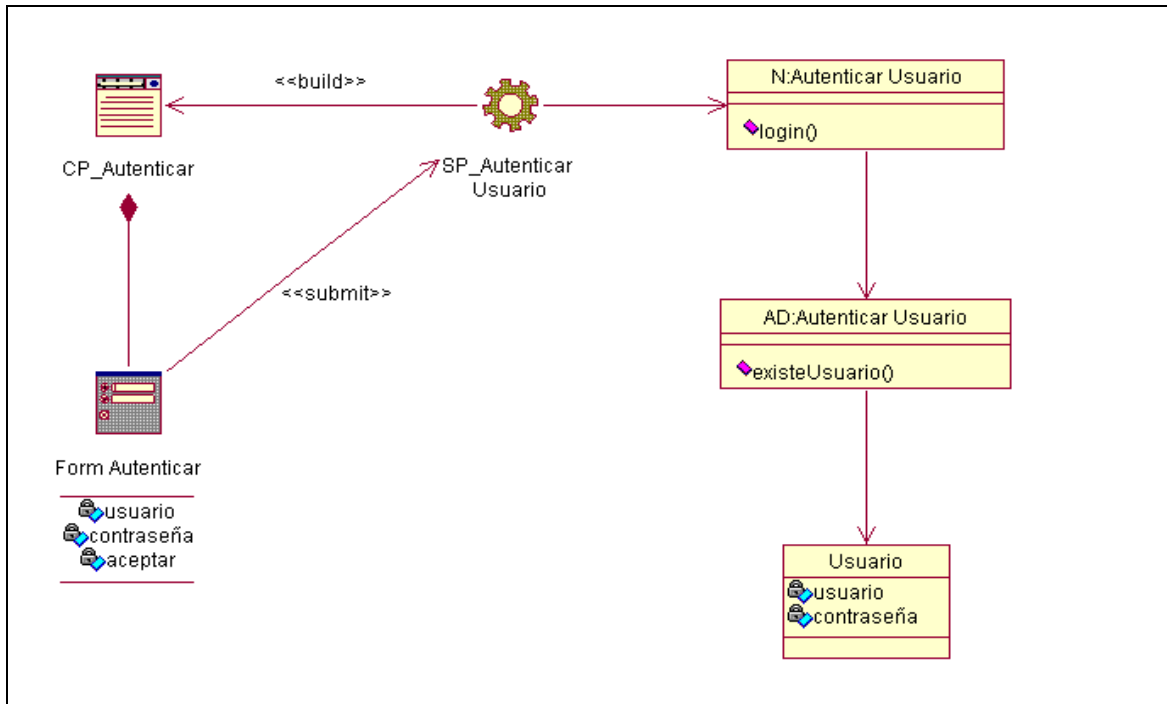


Figura 3.12 Diagrama de clases Web para el CU Autenticar

3.4.1.3 Gestionar Guardia

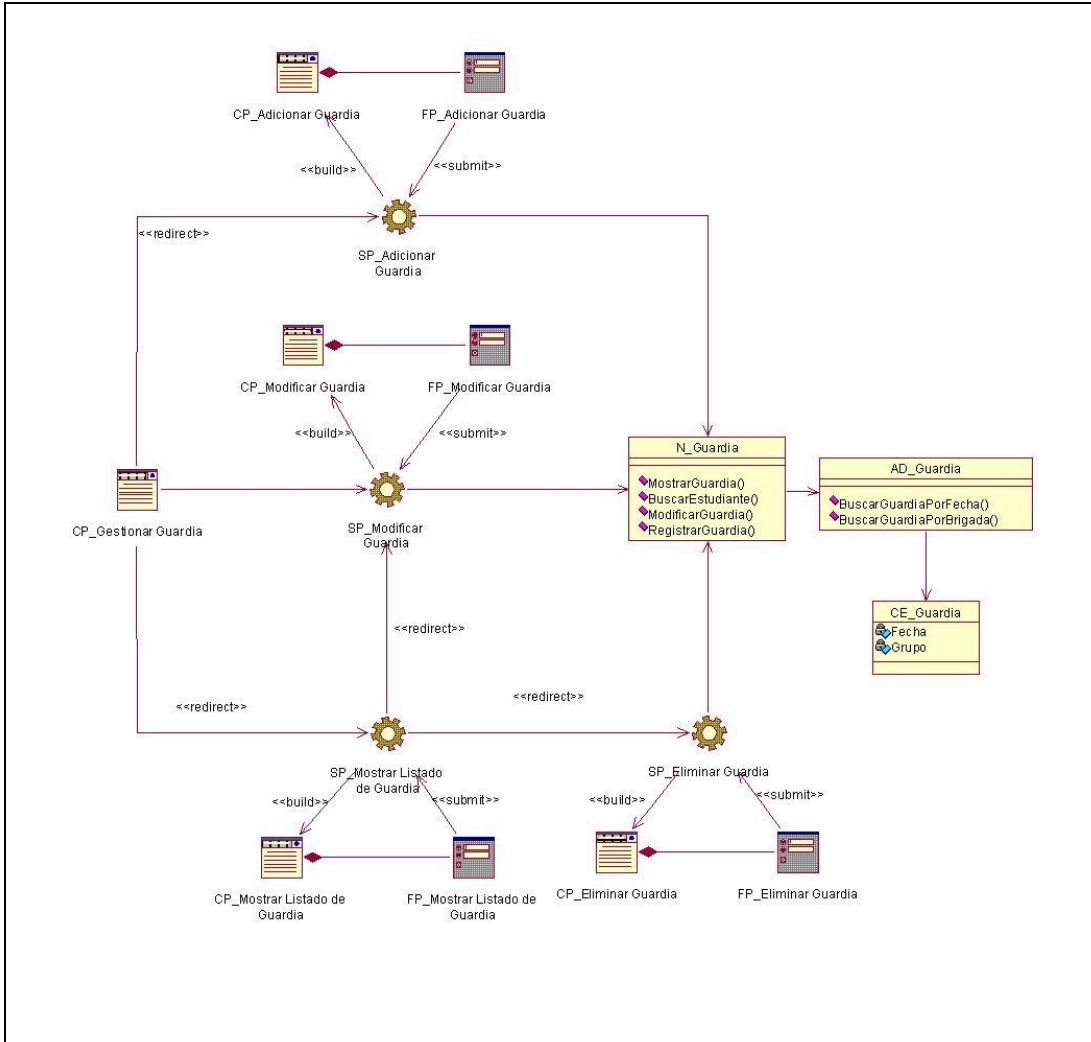


Figura 3.13 Diagrama de clases Web para el CU Gestionar Guardia

3.5. Diseño de la Base de Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, de manera tal que los datos que la conforman puedan ser utilizados en forma dividida cuando sea necesario, utilizando un conjunto de programas que manipulan esos datos

3.5.1 Diagramas de clases persistentes

Las clases persistentes son aquellas que cuentan con características o atributos propios, la información que guardan persisten en el tiempo dentro de la base de datos. Y estarán disponibles para cambiar información o para copia de seguridad, en caso contrario se afectaría el correcto funcionamiento del sistema.

A continuación se presenta el diagrama de clases persistentes

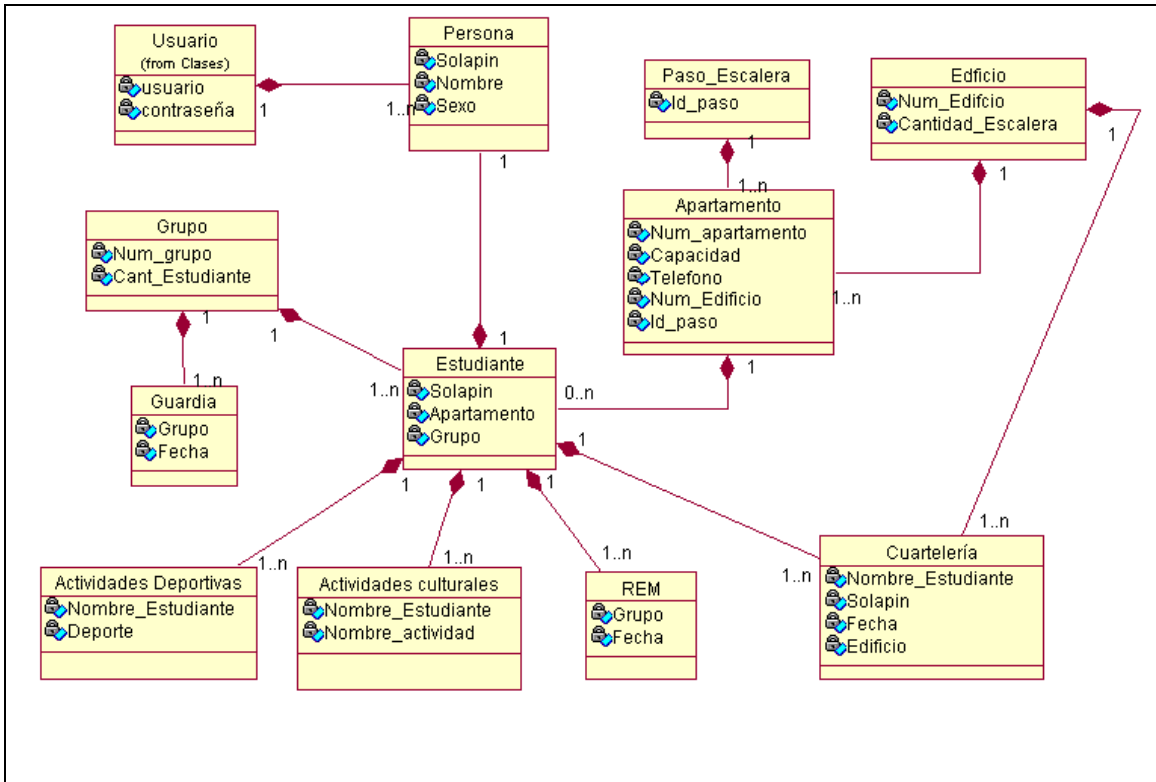


Figura 3.14 Diagrama de clases persistentes.

3.6. Conclusiones del capítulo

En este capítulo se desarrolló la modelación del sistema mediante la realización de los diagramas del sistema y del diseño este último representado permitiendo una mejor comprensión dando a conocer la lógica del sistema. Además de la representación de los diagramas de colaboración, así como el diseño de la base de datos abordando el diagrama de clases persistentes.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

4.1 Introducción

Este capítulo aborda los diferentes diagramas de implementación que representan como el modelo de diseño es implementado en forma de componentes y el despliegue que realiza el sistema. Se describen las pruebas realizadas a cada caso de uso utilizando el método de pruebas de caja negra.

4.2 Modelo de implementación

El modelo de implementación describe cómo los elementos de diseño se implementan en componentes. Además describe como están organizados los componentes de acuerdo con los elementos de organización disponibles en el contexto de implementación y en el lenguaje de programación utilizado, así como la dependencia existente entre estos componentes.

4.2.2. Diagrama de componentes del sistema

Un componente representa algo físico y son creados para poner el código, son creados, eliminados y modificados en el proceso de implementación. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas, pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente.

Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación, cada paquete puede contener componentes y otros paquetes. A continuación se presenta el diagrama de componentes de cómo Drupal estructura los subsistemas de implementación que utiliza, solo se describen los más generales debido a que el cms cuenta con muchos más componentes.

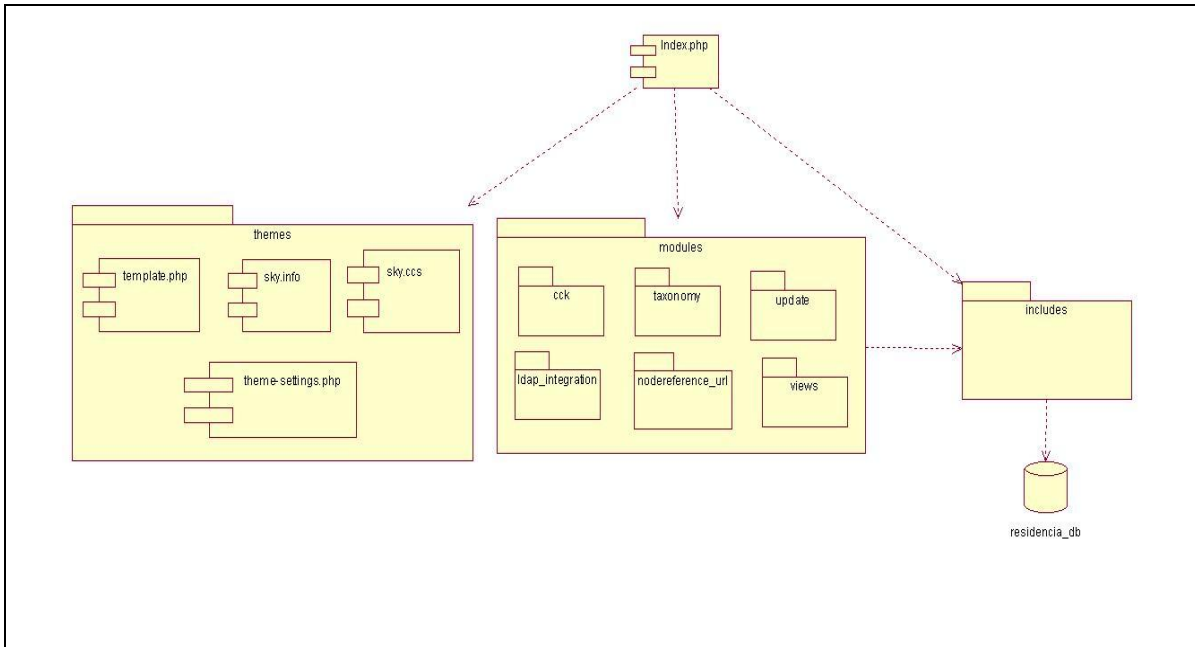


Figura 4.2 Diagrama de Componentes del Sistema

4.3 Modelo de despliegue

En un modelo de despliegue se representan las relaciones físicas que existen entre los componentes de hardware y software, se muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los enlaces de comunicación entre ellos y las instancias de los componentes y objetos que residen en ellos.

4.3.1 Diagrama de Despliegue

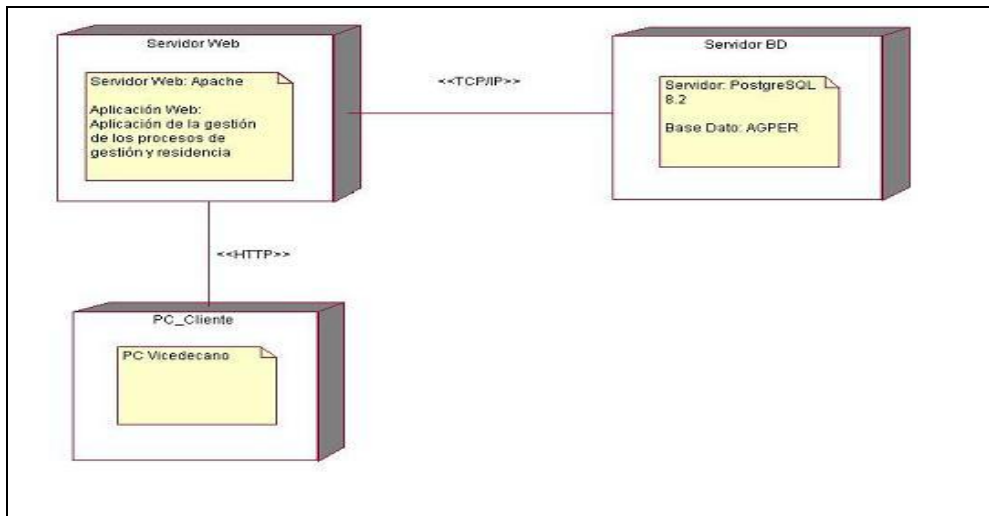


Figura 4.6 Diagrama de despliegue

4.4 Pruebas

La prueba del software es un elemento necesario para la garantía de la calidad del software. La etapa de pruebas tiene como objetivo garantizar la calidad del producto desarrollado. Existen diversos métodos de pruebas como las conocidas **pruebas de caja negra**, las cuales se llevan a cabo sobre la interfaz del software, sin mezclarse con el comportamiento interno ni la estructura del programa. Los casos de prueba de la caja negra pretende demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

A continuación siguiendo el método de caja negra explicado anteriormente se muestran algunos ejemplos de las pruebas efectuadas al sistema para cada caso de uso.

4.4.1 Pruebas de Caja Negra para el Caso de uso “Autenticar”

Condición de Entrada	Casos Válidos	Casos no Válidos
Usuario	Cadena de caracteres	Dejar vacío el campo Usuario
Contraseña	Cadena de caracteres	Dejar vacío el campo Contraseña

Caso de Uso:	Autenticarse
Caso de Prueba:	Autenticarse entrando correctamente los datos
Entrada:	
El Usuario introduce correctamente los datos necesarios para acceder al sistema. Usuario: “yoandry” Contraseña: “vicedecano”	

Resultado:	El sistema le da acceso al usuario autenticado y le muestra la página principal.
Condiciones:	Usuario y Contraseña introducidos correctamente.

Caso de Uso:	Autenticarse
Caso de Prueba:	Autenticarse entrando datos incorrectos
Entrada:	
El Usuario introduce correctamente los datos necesarios para acceder al sistema. Usuario: "Campo Omitido" Contraseña: "vicedecano"	
Resultado:	El sistema muestra un mensaje de alerta notificando el error ("El campo usuario es obligatorio").
Condiciones:	Usuario y Contraseña introducidos incorrectamente.

Para la validación de la aplicación se utilizaron las pruebas de caja negra, se realizaron dos revisiones, en la primera se detectaron tres no conformidades, todas ellas no significativas, dos eran errores ortográficos y la otra una incongruencia, en la segunda iteración no se detectaron no conformidades. Además de las pruebas de caja negra se aplicó una lista de chequeo, para de esta forma asegurar la calidad del software, siendo liberado el mismo con el cumplimiento total de sus funcionalidades.

4.5 Conclusiones del Capítulo

En el presente capítulo se presentó como está estructurado el sistema, tomando como punto de partida el diagrama de componentes, se muestra además el diagrama de despliegue con los nodos necesarios para la utilización del sistema, garantizando que todo funcione correctamente, y por último fueron descritas un conjunto de pruebas que se le realizaron a la aplicación, con los parámetros utilizados durante la ejecución de las mismas así como los resultados obtenidos.

CONCLUSIONES

Como resultado del desarrollo de este trabajo se dio cumplimiento a las tareas de investigación, obteniendo con esto el análisis, diseño e implementación de la aplicación, logrando entender las actividades que se desarrollan en la misma, identificando las funcionalidades que cumpliría el sistema y seleccionando una mejor vía de planificación, control y organización en los procesos de extensión y residencia en la facultad 15. Se hizo la selección de las herramientas, lenguajes y tecnologías necesarias para la confección de este sistema, dándole cumplimiento a los requerimientos definidos, tanto requisitos funcionales como requisitos no funcionales que soportan el sistema.

Se obtuvieron los artefactos del Modelo de Casos de Uso del Sistema, Modelo de Diseño, Modelo de implementación, resultantes de realizar la ingeniería del software según los flujos de trabajo que propone RUP que fue la metodología de desarrollo seleccionada.

Se obtuvo una aplicación que integra un conjunto de funcionalidades, logrando gestionar la información y planificación de algunos de los procesos que se llevan a cabo en el Área de Extensión y Residencia, dicha aplicación informática facilitará que el manejo de la información que se lleva a cabo en toda esta área se realice con la mayor calidad, obteniendo de esta forma un mejor rendimiento en la gestión de todos estos procesos extracurriculares que realizan los estudiantes en la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Luego de haber cumplido con los objetivos propuesto se recomienda:

Continuar con el estudio de los procesos que se llevan a cabo en el Área de Extensión y Residencia que todavía no han sido automatizados, con el fin de integrarlo con esta aplicación si se realizan nuevas versiones.

REFERENCIAS BIBLIOGRÁFICAS

- [1.] Hernández León, R. A., & Coello González, S. (2002). *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. Ciudad de la Habana: Editorial Universitaria.
- [2.] **García Cuerda, Xavier**. Mosaic. [En línea]
[http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/..](http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/)
- [3.] *Herramientas de la web social en la BUS. Documento marco 2008*. [En línea] Abril de 2008. http://herramientas2-0.pbworks.com/f/Documento_marco_de_la_BUS.doc.
- [4.] APCOMPUTACION. [En línea]
http://www.apcomputacion.cl/index.php?option=com_content&view=section&layout=blog&id=4&Itemid=4.
- [5.] Joomlaos.net. *Características de Joomla, Ventajas y características de Joomla CMS*. [En línea] <http://www.joomlaos.net/caracteristicas-de-joomla.php>.
- [6.] *11 RAZONES PARA UTILIZAR DRUPAL PARA SU SITIO WEB en Comunidad Drupal*. [Online] <http://www.pilos.com.co/drupal/11-razones-para-utilizar-drupal-para-su-sitio-web>.
- [7.] **Reyero, Jose**. *Características de Drupal*. [En línea] 17 de Julio de 2005.
<http://drupal.org.es/caracteristicas>.
- [8.] Comunidad Drupal. [En línea] <http://www.pilos.com.co/drupal/joomla-o-drupal-dificil-eleccion>.
- [9.] **Alvarez, Miguel Angel**. Desarrolloweb. *Qué es JavaScript*. [Online]
<http://www.desarrolloweb.com/articulos/25.php>.
- [10.] *JavaScript Patzcuaro MC*. [Online]
http://www.articulosinformativos.com.mx/Javascript_Patzcuaro_MC-r1106253-Patzcuaro_MC.html.
- [11.] *Que son los estandares Web*. [En línea] <http://www.jhonf.com/2010/05/que-son-los-estandares-web/>.

- [12.] **Chamorro, Lilian.** Internet para la rendición de cuentas. *¿Qué es una hoja de estilo?* [En línea] http://www.iprc.org.co/preguntas_frecuentes.shtml?apc=h1a1--&x=1010.
- [13.] Telecomunicaciones. *Lenguajes de marcas(HTML)*. [En línea] 8 de diciembre de 2007. <http://darkaguila.soy.es/lenguajes-de-marcas-html/>.
- [14.] **de la Torre, Aníbal.** *Lenguajes del lado servidor o cliente*. [Online] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- [15.] . **DSA.** *Los diferentes lenguajes de programación para la web*. [En línea] 20 de Febrero de 2010. http://dsapandora.orgfree.com/joomla/index.php?option=com_content&view=article&id=49&Itemid=38
- [16.] **Rodas Hinostroza, Raul.** Linuxcentro. *Características de PHP*. [En línea] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
- [17.] *Una introducción al APACHE*. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro/.
- [18.] **Marqués Andrés, María Mercedes.** *Apuntes de Ficheros y Bases de Datos*. [En línea] <http://www3.uji.es/~mmarques/f47/apun/apun.html>.
- [19.] [En línea] 29 de agosto de 2009. <http://www.buenastareas.com/ensayos/Gestores-De-Bases-De-Datos/12666.html>.
- [20.] **González, Carlos.** *Curso Base de Dato PostgreSQL*. [En línea] <http://www.usabilidadweb.com.ar/postgre.php>.
- [21.] Anónimo. (2007). Recuperado el 2010, de <http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>
- [22.] [En línea] <http://www.geopalm.cl/images/imagenes/metodologia-rup.jpg>
- [23.] Ingenieriasoftwaredos. *Metodología RUP*. [En línea] <http://ingenieriasoftwaredos.wikispaces.com/Grupo+RUP>.

GLOSARIO DE TÉRMINOS

CMS: Content Management Systems en español Sistema Gestores de Contenido

Joomla: es un sistema de administración de contenidos de código abierto construido con PHP.

Drupal: es un sistema de administración de contenido

URL: significa Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

HTML: Hyper Text Markup Language, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

CSS: Hojas de Estilo en Cascada que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe como se va a mostrar un documento en pantalla, por impresora.

ASP: Active Server Page es una tecnología de script que corre del lado de servidor y puede ser usado para crear aplicaciones Web dinámicas e interactivas.

JSP: es el acrónimo de Java Server Pages, es una tecnología Java que permite a los desarrolladores generar contenido dinámico para web, en forma de documentos HTML, XML u otros.

XML: Extensible Markup Language, es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. En la práctica corresponde a un estándar que permite a diferentes aplicaciones interactuar con facilidad a través de la red

PHP: acrónimo de Hypertext Pre-Processor, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor, es un lenguaje muy potente que, junto con HTML, permite crear sitios web dinámicos

*Ciudad de la Habana, 2010
“Año del 50 Aniversario del Triunfo de la Revolución”*