Facultad de Ciencias y Tecnologías Computacionales

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título: "Sistema para la gestión y seguimiento de reservaciones de las áreas compartidas en la Universidad de las Ciencias Informáticas."

Autor:

Eddy Nuñez Oquendo.

Tutoras:

Msc. Juana Elena Acosta García.

Ing. Bárbara Bron Fonseca.

La Habana, junio de 2017.

"Año 59 de la Revolución"

"Educar a un joven no es hacerle aprender algo que no sabía, sino hacer de él alguien que no existía".

John Ruskin

Declaración de autoría

Declaro ser autor de la presente tesis que tiene por título: “Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas.” Reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año 2017.

_____

Firma del autor.

_____ _____

Firma de la tutora. Firma de la tutora.

Datos de contacto

Tutora:

Msc. Juana Elena Acosta García: Graduada de Máster en Ciencias de la Educación y Profesor Auxiliar. Posee 42 años de experiencia en la Educación. Ha realizado tutoría de Trabajos de Diploma y de Tesis de Maestría, así como ha realizado labores de tribunal en variados eventos científicos desarrollados.

Correo electrónico: juana@uci.cu

Tutora:

Ing. Bárbara Bron Fonseca: Graduada en el año 2012 de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Se desempeñó como Especialista B en Informática en la Empresa de Tecnologías para la Defensa XETID. Actualmente es profesora del departamento de ISW en la facultad CITEC de la UCI. Posee publicaciones en revistas científicas y memorias de eventos. Certificado como especialista en calidad para revisiones de Software. Presenta experiencia como tutor y oponente en pregrado.

Correo electrónico: bbron@uci.cu

Agradecimientos

Agradezco el apoyo y la constancia de todas las personas que estuvieron siempre a mi lado, a las que dudaron, por darme fuerzas para ver siempre el lado positivo de las cosas y a las que me acompañaron en este recorrido pasando por muchas cosas incluyendo noches de desvelo y preocupaciones.

A todas y todos gracias por demostrarme que siempre es posible lograr lo que nos proponemos, que aunque sea difícil el camino el éxito siempre se puede alcanzar.

Muchas Gracias.

Resumen

Para la Universidad de las Ciencias Informáticas (UCI) es de vital importancia la realización de talleres de superación, reuniones colectivas, actos conmemorativos y otra serie de actividades que requieren de locales especializados. La reservación de estos locales, también llamados áreas compartidas se realiza de forma manual, lo que ha ocasionado solapamiento de reservaciones y disgustos a los directivos. La presente investigación describe una solución a este problema con la implementación de un Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas (UCI). El cual permite agilizar e informatizar dicho proceso de manera centralizada.

Para guiar el desarrollo del sistema se utilizaron metodologías, herramientas y tecnologías de código abierto acorde a la política de soberanía tecnológica que defiende el país. El sistema obtenido permite la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas, dándole a este proceso una mayor rapidez, organización y control, garantizando además una mayor disponibilidad de la información elevando la precisión en la toma de decisiones por parte de la Dirección de Servicios Generales de la Universidad.

Palabras Claves: *Sistema de gestión, reservación, áreas compartidas.*

Abstract

For the University of Computer Science (UCI) it is vital to carry out workshops, collective meetings, commemorative events and other activities requiring specialized premises. The reservation of these premises, also called shared areas is done manually, which has caused overlap of reservations and dislikes to the managers. The present research describes a solution to this problem with the implementation of a System for the reservation and monitoring of shared areas at the University of Computer Science (UCI). This allows streamlining and computerizing this process centrally.

To guide the development of the system, methodologies, tools and open source technologies were used according to the policy of technological sovereignty that the country defends. The system obtained allows the reservation and monitoring of shared areas at the University of Computer Science, giving this process a greater speed, organization and control, also guaranteeing a greater availability of information by increasing the precision in decision making by The Directorate General Services of the University.

Key Words: *Management system, reservation, shared areas.*

Índice

Índice de figuras

Índice de tablas.

Introducción

Con el desarrollo de la actividad cognoscitiva del hombre, los cambios en las fuerzas productivas, los avances en los modos de producción y del momento histórico vigente, las Tecnologías de la Información y las Comunicaciones (TIC) se han introducido de forma masiva en todas las aristas sociales, redefiniendo los paradigmas de trabajos tradicionales, a partir de la integración tecnológica.

Cuba ha estado inmersa en un profundo y novedoso proceso de transformaciones educacionales y sociales como parte de los programas de la Batalla de Ideas, a partir de la cual se emprendieron nuevos programas destinados a elevar el nivel cultural de la población y su calidad de vida. En estas circunstancias y con la visión de futuro del Comandante en Jefe Fidel Castro Ruz, surge la Universidad de Ciencias Informáticas (UCI).

Fidel, como estratega fundacional, recomendó que la universidad fuese concebida como un centro de nuevo tipo, de alcance nacional, de características atípicas y tareas concretas en el proyecto de informatización de la sociedad cubana, con énfasis en la producción de Software. Un centro que formaría a sus educandos desde una destacada actividad política, cultural y deportiva, objetivos que se han venido cumpliendo mediante la interrelación de la formación-investigación-producción-extensión universitaria como un proceso único y coherente.

La universidad cuenta con un conjunto de áreas vinculadas a las actividades docentes, productivas, investigativas y de servicios, posee además áreas denominadas ¨compartidas¨ entre las que se pueden encontrar los salones de reuniones, teatros, discoteca, etc. Es de vital importancia la gestión de las áreas compartidas con el objetivo de lograr una buena organización, planificación y control de su utilización. También para garantizar su estado de conservación y facilitar el trabajo de las personas que necesitan acceder a ellas.

La Dirección de Servicios Generales (DSG) de la UCI es la encargada de garantizar la reservación de cada una de las áreas compartidas con que cuenta la universidad. Esta actividad se realiza de forma manual lo que ha ocasionado solapamiento de horarios e irregularidades en las actividades a realizar, por lo que el autor en su práctica pudo identificar mediante las entrevistas y encuestas de satisfacción de la calidad del servicio de reserva a técnicos, especialistas y directivos las siguientes insatisfacciones:

- Solapamientos en las reservaciones realizadas.
- Falta de control en la disponibilidad de los locales.
- El cliente que reserva no tiene una información completa de la capacidad y medios con que cuenta el local.

- No existe una estrategia que permita priorizar a que evento se debe asignar una reservación en caso de que existan dos solicitudes iguales.
- Cuando se maneja mucha información no es posible dar información exacta de la disponibilidad de los locales.
- No existen mecanismos para la gestión de los cambios existentes en las reservaciones que permita informar de forma oportuna a los involucrados. Estos cambios son realizados de forma manual lo que genera inconsistencias.

Atendiendo a la situación antes planteada se define como **problema a resolver:** ¿Cómo contribuir en la organización y disminución del margen de errores en el servicio de reservación y seguimiento de las áreas compartidas de la Universidad de las Ciencias Informáticas?

Para dar solución al problema identificado se traza como **objetivo general:** Desarrollar un sistema que contribuya a la organización y disminución del margen de errores en el servicio de reservación y seguimiento de las áreas compartidas de la Universidad de las Ciencias Informáticas.

Una vez planteado el objetivo general se determina como **objeto de estudio:** Los sistemas de reservación de salas, enmarcado en el **campo de acción:** El sistema de reservación y seguimiento de áreas compartidas de la Universidad de las Ciencias Informáticas.

La **idea a defender** de la presente investigación sustenta que: Con el desarrollo de una aplicación informática, se contribuirá a la organización y disminución del margen de errores en el servicio de reservación y seguimiento de las áreas compartidas de la Universidad de las Ciencias Informáticas.

Para guiar el proceso investigativo del presente trabajo se definen las siguientes **tareas investigativas:**

- Investigación y análisis de las fronteras del proceso de reserva de áreas compartidas.
- Estudio científico sobre las soluciones existentes.
- Estudio de las herramientas y tecnologías que serán utilizadas en el desarrollo del sistema.
- Análisis y diseño de la propuesta de solución.
- Implementación de la solución propuesta.
- Validación de la solución propuesta.

Para obtener los conocimientos necesarios con la finalidad de hacer posible el cumplimiento del objetivo trazado en el trabajo, se llevó a cabo un estudio en la que se utilizaron los siguientes métodos de la investigación científica:

Métodos empíricos: usados para poder determinar el grado de complejidad que adquiere el problema y para identificar si quedaron o no satisfechas todas las necesidades previstas a través de(SAMPIERI *et al.* 2006):

- **Entrevista:** Aplicada a miembros de la Dirección de Servicios Generales de la Universidad de las Ciencias Informáticas (UCI) con el objetivo de obtener información sobre el proceso de reservación, seguimiento y control de las áreas compartidas y a partir de las respuestas obtenidas, identificar las principales deficiencias que se cometen en este proceso (ver anexo 1).

Métodos teóricos: usados para estudiar las características del objeto de investigación que no son observables directamente, para apoyar el análisis, la síntesis y su relación con otros fenómenos(SAMPIERI *et al.* 2006):

- **Analítico-Sintético:** Aplicado por el autor para descomponer el problema de la investigación en elementos por separados y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta. Entre los elementos identificados se encuentran el procedimiento creado por la Dirección de Servicios Generales de la Universidad encargados de la tarea de reservación, seguimiento y control de las áreas compartidas, de manera que permita construir correctamente el marco teórico de la investigación.

- **Histórico-Lógico:** Aplicado por el autor para constatar teóricamente cómo han evolucionado los sistemas de gestión en el ámbito laboral, los cuales varían en dependencia de las necesidades que requieran los usuarios y las posibilidades reales de suplirlas de cada institución en dependencia del momento histórico vigente de forma tal que permita un análisis real a la hora de identificar los problemas que existen en la reservación, seguimiento y control de las áreas compartidas de la UCI.

- **Modelación:** Aplicado por el autor para reproducir de forma simplificada la realidad del objetivo de estudio. Creando abstracciones del mismo para explicar la realidad de la necesidad de implementar un sistema que permita realizar reservaciones de las áreas compartidas de la UCI, así como su seguimiento y control.

El presente trabajo de diploma se ha estructurado de la siguiente forma:

Capítulo 1: Fundamentación teórica.

En este capítulo se realiza la elaboración del marco teórico donde se exponen los conceptos asociados a la solución de la problemática y las diferentes soluciones existentes en Cuba y a nivel mundial. De igual manera se describen y caracterizan las herramientas y el lenguaje de programación para el diseño e implementación de la solución, así como la metodología de desarrollo más recomendable.

Capítulo 2: Análisis y diseño del sistema.

Se exponen las principales características y cualidades de la solución a implementar, además se identifican los requisitos, se escoge la arquitectura y los patrones de diseño, se diseñan las clases y se detallan los pasos de la metodología propuesta, incluyendo varios de sus artefactos y diagramas.

Capítulo 3: Implementación y pruebas del sistema.

Se muestra la situación física de los distintos componentes lógicos desarrollados a través del modelo de despliegue y la organización del sistema mediante el modelo de componentes. Y por último se analiza el funcionamiento del sistema desarrollado aplicándole las pruebas necesarias para demostrar que la solución es correcta.

Capítulo 1: Fundamentos teóricos.

Introducción.

El presente capítulo introduce los conceptos básicos relacionados con el funcionamiento de los sistemas de gestión que se encargan de las reservaciones de áreas compartidas. Son analizados sistemas profesionales de corte nacional e internacional y son seleccionadas las herramientas y metodologías a utilizar en el desarrollo del software, justificando el uso de cada una de ellas.

1.1. Conceptos asociados al sistema para la reservación y seguimiento de áreas compartidas en la UCI.

Sistemas de gestión:

Un Sistema de Gestión (SG) es una herramienta o aplicación informática que permite controlar todos y cada uno de los aspectos de una empresa (pedidos, producción, control de presencia, facturación, ventas, administración, etc.), un SG se hace imprescindible ya que, no solamente permite un control exhaustivo del ciclo de producción, sino que además permite la interacción con los dispositivos (CRISTALDO and KLOSTER).

Gestión de la información:

La gestión de la información representa la forma de organizar, evaluar, presentar y comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil; y que esta información esté disponible en el momento que se le necesite. Ella se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización (Ledo, 2012).

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas trazadas en materia de desempeño y de calidad.

Para gestionar la información se han creado algunos sistemas de gestión de la información que están constituidos por los métodos y procedimientos establecidos para registrar, procesar, resumir e informar sobre las operaciones de una entidad. La calidad de la información que brinda el sistema afecta la capacidad de los directivos y ejecutivos para adoptar decisiones adecuadas que permitan controlar las actividades de la entidad (Contraloría General de la República, 2009).

1.2. Sistemas de Reservación de Áreas Compartidas.

A continuación, se exponen algunos sistemas que presentan como principal característica la reservación de salas, salones y espacios compartidos fuera del territorio nacional ya que durante la investigación no se encontraron sistemas de reservación en el territorio nacional que estuviesen acorde con la problemática planteada para la investigación.

1.2.1. Sistema para la solicitud de salas de reunión del Centro de Ciencias Humanas y Sociales (CCHS) de España.

Este sistema se reduce a cuatro pasos para el método de realización de solicitud de reservas de las salas de reuniones del CCHS, a través de la intranet del centro.

Acceder a la aplicación a través de la dirección web del centro y dentro de esta dirección pulsar en la opción de intranet que lleva el nombre de "Reserva de Salas de reunión", que se encuentra en la parte izquierda de la pantalla.

Aparece una pantalla, dónde se muestran en la parte superior, los distintos elementos para que el usuario elija la sala y el día deseados para su reunión.

Al seleccionar el horario que quieras en la sala elegida previamente, aparece una pantalla con diferentes campos llamada "Nueva Reserva" para introducir los datos de la nueva reservación.

Una vez rellenado el formulario de la reserva de sala, se procederá a pulsar el botón SALVAR, y posteriormente, desde la secretaría del centro las o la persona encargada, responderá mediante correo electrónico o llamada de teléfono con la confirmación o denegación de la reserva de la sala.

Este sistema está desarrollado con herramientas privativas y con restricciones de uso, además de no cumplir con las necesidades que se desean automatizar en la Universidad detectadas en la entrevista realizada para esta investigación a la DSG(CCHS).

1.2.2. Configuración salas de reunión Microsoft Office OutLook 2010.

Pasos para la configuración de acceso a los Calendarios de las Salas de Reuniones de Microsoft Office Outlook.

Este acceso es por medio de la lista de calendarios de cliente de correo electrónico Microsoft Exchange 2010 Estándar. La característica de calendario de grupo le permite reservar recursos como equipos audiovisuales y salas de conferencias. Por ejemplo, puede conocer la disponibilidad y reservar una sala de conferencias cuando vaya a programar una reunión. También puede reservar solo un recurso, como el equipamiento de teleconferencia.

Abrir Microsoft Outlook e ir a la opción Mis Calendarios**.** Luego seleccionar la opción Agregar Calendario. Después seleccionar la opción Abrir Calendario Compartido. Seleccionar el botón nombre. A continuación, se visualizará una ventana con la lista de contactos corporativos. Ahí hacer clic en la sección Libreta de direcciones y seleccionar la opción Todos los usuarios. Ubicar los nombres de las salas respectivas y luego hacer clic en el botón Aceptar. Luego aparecerá una ventana con el nombre de la Sala Seleccionada. Para agregarlo a la lista de calendarios del Microsoft Outlook, hacer clic en el botón Aceptar. Finalmente llegarán a la lista de Calendarios de Microsoft Outlook y se podrán visualizar sus calendarios y el que acaban de agregar.

Es un sistema desarrollado con herramientas privativas y con restricciones de uso, complejo e ineficiente según sus características, ya que se vuelve engorroso y complicado para el usuario el agendar una reservación, no brinda posibilidades de reservar servicios tercerizados y para cada reservación en un local distinto hay que realizar demasiados pasos para solicitar una reserva.

1.2.3. Sistema de reservación de salas de reunión Evoko Room Manager.

Desde su aparición Evoko Room Manager previene de incertidumbres al mostrar el estado de las reservas delante de la sala. Tiene pantallas táctiles fuera de las salas de reuniones que muestran si la sala está libre u ocupada, utilizando iluminación LED verde o roja que se puede ver desde lejos. La información básica se muestra automáticamente en la pantalla, incluyendo la hora de inicio y finalización, el tiempo restante de la reunión y quien ha reservado la sala. Se pueden reservas desde Outlook o utilizando los menús de la pantalla táctil, donde también se puede obtener un resumen de las reservas, hacer tu propia reserva o cancelación, y alargar la reunión. Si la sala está ocupada, la pantalla te sugiere salas de reuniones alternativas(EVOKO 2010).

Este sistema está basado en el entorno de Microsoft Exchange Server, además se encuentra vinculado a Microsoft Office Outlook todos creados con herramientas privativas y con restricciones de uso que no están acorde a la política de soberanía tecnológica que implementa Cuba, utiliza pantallas táctiles ubicadas en cada local para hacer modificaciones en las reservaciones con las que no se cuenta en la Universidad.

1.2.4. Sistema de agendamiento de salas de reuniones TouchOne.

El sistema gestiona los problemas más frecuentes que ocurren en las salas de reuniones tales como:

- Doble reserva.
- Reuniones interrumpidas.
- Reuniones espontáneas.

- Uso ineficiente de los recursos.

Se integra y sincroniza con servidores MS Exchange 2007-2016, MS Office 365 y Google Apps for Work para poder mostrar la información actual del calendario de la sala. Muestra una configuración simple tipo Plug&Play con ventajas y beneficios tanto para el integrador como el cliente.

Esta herramienta no responde a las necesidades de la investigación, fue desarrollada con herramientas privativas y con restricciones de uso por lo que no es conveniente su utilización y aplicación en el país.

1.2.5. Sistema de reservación de aulas postgrado.uci.cu.

Sistema web desarrollado en la Universidad de las Ciencias Informáticas con el objetivo de prestar el servicio on-line de reservación de las distintas aulas de los edificios docentes del campus universitario para la realización de clases y otros eventos de postgrado. Se encuentra desarrollado en software libre según la política de soberanía tecnológica que implementa nuestro país, pero es muy simple y no cumple con los requisitos y las necesidades planteadas en esta investigación.

1.2.6. Valoraciones generales.

Después de realizar un análisis de los sistemas expuestos con anterioridad, se puede decir que en su generalidad no cumplen con la necesidad que se plantea en la problemática. Es decir, la información que gestionan estos sistemas, no coincide con la información que se desea controlar y no manejan los mismos tipos de datos en cada uno de ellos.

No se cuenta con tecnología táctil en todas las salas de la Universidad que permita hacer reservaciones con el sistema operativo Androide.

En Cuba las mayores incursiones en el tema de reservación de espacios compartidos a través de aplicaciones web se encuentran en el área turística y en centros de recreación, principalmente para la reservación de habitaciones u hospedajes y para la reservación de entradas a los teatros.

Estas herramientas en línea no satisfacen todos los requerimientos y especificaciones de los usuarios por lo que se decide implementar un sistema de gestión para la reservación de áreas compartidas de la UCI.

1.3. Tecnologías, herramientas y metodologías a considerar.

Las tecnologías, herramientas y metodologías escogidas para el desarrollo del sistema se exponen en el siguiente subtema, para lo cual se analizaron las distintas alternativas y las necesidades del proyecto. El principal elemento que se tuvo en consideración fue que las herramientas debían ser compatibles, basadas en software libre, gratuitas y sin restricciones de uso.

1.4. Metodología de Desarrollo de Software.

Una metodología es una guía para el desarrollo del software; que hace posible que todo el personal de un proyecto se vincule y pueda entenderse, estas son un conjunto de procedimientos y técnicas que ayudan a la documentación para el desarrollo de productos de software, en la que se indican, paso a paso, todas las actividades a realizar para lograr informatizar el proceso deseado; muestran qué personas deben participar en el desarrollo de las actividades y el papel que desempeñan. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Metodología de Desarrollo de Software (OpenUp).

Open UP es una metodología de desarrollo de software, basada en RUP (Rational Unified Process), que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad, de una forma eficiente. Open UP, es un proceso unificado, iterativo e incremental, que se centra en el desarrollo colaborativo de software para generar sistemas de calidad. Los elementos que forman Open Up son tareas, disciplinas, artefactos y procesos. El ciclo de vida de un proyecto, según la metodología Open UP, permite que los integrantes del equipo de desarrollo aporten con micro-incrementos, que pueden ser el resultado del trabajo de unas pocas horas o unos pocos días. El progreso se puede visualizar diariamente, ya que la aplicación va evolucionando en función de este micro-incremento. El objetivo de Open UP es ayudar al equipo de desarrollo, a lo largo de todo el ciclo de vida de las iteraciones, para que sea capaz de añadir valor de negocio a los clientes, de una forma predecible, con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de: una visión del proyecto, transparencia y los medios para que controlen la financiación, el riesgo, el ámbito, el valor de retorno esperado (Infante A., 2013).

Todo proyecto en Open UP consta de cuatro fases: inicio, elaboración, construcción y transición. Cada una de estas fases se divide a su vez en iteraciones (Open Up Basic. Eclipse Foundation, 2011)

Se escoge como metodología de desarrollo de software a Open Up, que permite al sistema integrarse a requerimientos cambiantes que se pueden dar durante el desarrollo del proyecto, es una metodología ágil, que presenta documentación, muy favorable para proyectos que presentan pocos integrantes y es aplicable a sistemas de bajo costo.

1.5. Lenguajes de modelado.

El lenguaje de modelado pretende dar apoyo a la mayoría de los procesos de desarrollo de software, son desarrollados con el objetivo de simplificar y consolidar el gran número de métodos de desarrollo que han surgido. (Jacobson, 2000).

Lenguaje Unificado de Modelado 2.0 (UML).

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y permite la modelación de sistemas con tecnología orientada a objetos. Este lenguaje es fácil de aprender y usar, conforma la colección de las mejores técnicas de ingeniería que han probado ser un éxito en el modelado de sistemas grandes y complejos. Con el uso de UML se facilita la asimilación y entendimiento del contenido, se minimiza el tiempo invertido en el desarrollo del sistema y la documentación del proyecto se confecciona de una forma ordenada.

Se escoge UML 2.0 por ser un lenguaje reconocido y probado a nivel internacional además de poseer disimiles ventajas en cuanto a su uso y facilidad de comprensión.

1.6. Herramientas CASE de modelado.

La Ingeniería de Software Asistida por Computadora, CASE por sus siglas en inglés, constituye la aplicación de métodos y técnicas utilizadas para ayudar a las actividades del proceso del software, como el análisis de requerimientos, el modelo de sistemas, la depuración y las pruebas. Las herramientas CASE representan una forma que permite Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información.

Visual Paradigm 8.0.

Visual Paradigm es una herramienta visual de Ingeniería de Software para el modelado. Brinda una colección de menús, barras de herramientas y ventanas que forman el área de trabajo, lo cual permite crear diferentes tipos de diagramas en un ambiente completamente visual, manejar de manera rápida aplicaciones de mayor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Martín T., 2013).

Es una herramienta profesional multiplataforma, que proporciona tutoriales, demostraciones interactivas y proyectos UML. Posee como peculiaridad sobre el resto de las herramientas que cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones.

Visual Paradigm por brindar licencias gratuitas y comerciales. Es una herramienta multiplataforma, que permite dibujar todos los tipos de diagramas de clases, presenta amplia facilidad de uso, abundantes tutoriales de UML y una potente funcionalidad para la creación de interfaces.

1.7. Lenguajes de Programación.

En la actualidad existen diferentes lenguajes de programación, estos han surgido debido a las tendencias y necesidades de las plataformas existentes.

Desde el comienzo de Internet, surgieron diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías se desarrollaron y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos.

Lenguajes de Programación para el lado del servidor PHP3 5.5.1.

PHP es un lenguaje script que corre del lado del servidor en la Arquitectura Cliente – Servidor. Fue desarrollado originalmente por Rasmus Cerdorf en 1994. Es utilizado para generar páginas Web dinámicas. Su programación es segura y confiable ya que de ninguna manera en el navegador se accede al código fuente en PHP, sino solo a su resultado en HTML4 (González Enrique, 2010)

Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, entre muchas más cosas. Permite aplicar técnicas de programación orientada a objetos.

Como se ha diseñado para su uso en la Web, incorpora gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF en un instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF (González Enrique, 2010).

Una de sus características es su portabilidad ya que está disponible para diferentes sistemas operativos. Dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a bases de datos como PostgreSQL, Microsoft SQL, Oracle, entre otras. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos, no limita su distribución y se puede ampliar con nuevas funcionalidades si se desea. Se puede utilizar como módulo de Apache, lo que lo hace extremadamente veloz. Por estar completamente escrito en C, se ejecuta rápidamente utilizando poca memoria.

Lenguajes de Programación para el lado del servidor Python 2.7.

Python es más fácil de usar, está disponible para sistemas operativos Windows, Mac OS X, Linux y Unix, y le ayuda a realizar su tarea más velozmente.

Es un lenguaje de programación de verdad, ofreciendo mucha mayor estructura y soporte para programas grandes de lo que pueden ofrecer los scripts de Unix o archivos por lotes. Por otro lado, Python ofrece

mucho más chequeo de error que C, y siendo un lenguaje de muy alto nivel, tiene tipos de datos de alto nivel incorporados como arreglos de tamaño flexible y diccionarios. Debido a sus tipos de datos más generales Python puede aplicarse a un dominio de problemas mayor que Awk o incluso Perl, y aun así muchas cosas siguen siendo al menos igual de fácil en Python que en esos lenguajes.

Permite separar tu programa en módulos que pueden reusarse en otros programas. Viene con una gran colección de módulos estándar que puedes usar como base de tus programas, o como ejemplos para empezar a aprender a programar en Python. Algunos de estos módulos proveen cosas como entrada/salida a archivos, llamadas al sistema, sockets, e incluso interfaces a sistemas de interfaz gráfica de usuario como Tk.

Python es un lenguaje interpretado, lo cual puede ahorrar mucho tiempo durante el desarrollo ya que no es necesario compilar ni enlazar. El intérprete puede usarse interactivamente, lo que facilita experimentar con características del lenguaje, escribir programas descartables, o probar funciones cuando se hace desarrollo de programas de abajo hacia arriba. Es también una calculadora de escritorio práctica.

Python permite escribir programas compactos y legibles. Los programas en Python son típicamente más cortos que los programas equivalentes en C, C++ o Java por varios motivos:

- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola instrucción.
- La agrupación de instrucciones se hace por sangría en vez de llaves de apertura y cierre.
- No es necesario declarar variables ni argumentos.

Python es extensible: es fácil agregar una nueva función o módulo al intérprete, ya sea para realizar operaciones críticas a velocidad máxima, o para enlazar programas Python con bibliotecas que tal vez sólo estén disponibles en forma binaria (por ejemplo bibliotecas gráficas específicas de un fabricante). Puedes enlazar el intérprete Python en una aplicación hecha en otro lenguaje y usarlo como lenguaje de extensión o de comando para esa aplicación(ROSSUM 2009).

Leguajes de Programación para el lado del Cliente JavaScript.

Es uno de los lenguajes más utilizados en la creación de aplicaciones web presentando funcionalidades para la creación de contenidos dinámicos. Es un lenguaje de programación para el lado del cliente debido a que la mayor carga de procesamiento la soporta el navegador. Su uso está destinado principalmente para la creación de efectos visuales, así como para la creación de interfaces de usuario. Las aplicaciones que utilizan este lenguaje casi siempre están incrustadas en los códigos HTML de las aplicaciones web y están destinadas para realizar las acciones con el cliente.

CSS.

Es un lenguaje que permite la definición de hojas de estilos diseñadas para el control de las interfaces de usuario definidas en los códigos HTML. Separa los contenidos de la presentación de las aplicaciones lo que permite el diseño de documentos bien definido y muy complejo. Cuando son creados los contenidos de estilos es permitido definir los diferentes tamaños, colores y tipos de fuentes para las letras, así como la gestión de imágenes.

Biblioteca de JavaScript: ExtJS 4.0.7.

Es una biblioteca del Lenguaje JavaScript que facilita la implementación de aplicaciones web que contiene tecnologías como Ajax, DHTML y DOM con grandes funcionalidades para la implementación de Interfaces de Usuarios avanzadas. Fue diseñada originalmente como una extensión de la libraría YUI por Sencha y actualmente contiene además librarías de JQuery y Prototype y su licencia es de Código Abierto(LIBRES 2013).

Los lenguajes de programación del lado del cliente escogidos fueron ExtJS y JavaScript, porque los programas escritos en ellos no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos y son independientes de la plataforma, hardware o sistema operativo y funcionan correctamente siempre y cuando exista un navegador que los soporte.

El lenguaje escogido para el lado del servidor fue Python por poseer una gran capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. Este lenguaje es libre y la amplísima biblioteca que posee simplifica el trabajo de los desarrolladores.

Framework.

Los *frameworks* o marcos de trabajo son aplicaciones compuestas por componentes personalizables que permiten el desarrollo de aplicaciones. Las funcionalidades principales de estas aplicaciones es aligerar de carga a los desarrolladores realizando de manera automática e interactiva tareas comunes dentro de la programación, lo que permite reducir considerablemente el tiempo de desarrollo de aplicaciones complejas.

Framework Symfony 2.8.

Symfony es un poderoso *Framework* orientado para la optimización del desarrollo de aplicaciones web basado en el patrón de diseño MVC. El diseño del mismo separa la lógica del negocio, del servidor y la presentación; además proporciona herramientas para reducir el tiempo de desarrollo de aplicaciones

complejas. Realiza de forma automatizada las tareas básicas durante el desarrollo permitiendo al desarrollador centrarse en las funciones más específicas de la aplicación. El *Framework* está implementado parcialmente en PHP 5.3, es compatible con los Sistemas Gestores de Base de Datos más reconocidos del mundo y es independiente al sistema operativo del servidor, brindando así características muy poderosas para el desarrollo de aplicaciones web (POTENCIER and ZANINOTTO 2010). Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio (Framework, 2014).

Framework Django 1.6.

Django es un framework de desarrollo Web que ahorra tiempo y hace que el desarrollo Web sea divertido. Utilizando Django puedes crear y mantener aplicaciones Web de alta calidad con un mínimo esfuerzo. En el mejor de los casos, el desarrollo web es un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y frustrante. Django te permite enfocarte en la parte divertida -- el quid de tus aplicaciones Web -- al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. Al mismo tiempo, Django intenta no entrometerse, dejándote trabajar fuera del ámbito del framework según sea necesario. Django es sencillamente una colección de bibliotecas escritas en el lenguaje de programación Python, para desarrollar un sitio usando Django escribes código Python que utiliza esas bibliotecas. Django fue diseñado para promover el acoplamiento débil y la estricta separación entre las piezas de una aplicación. Si sigues esta filosofía, es fácil hacer cambios en un lugar particular de la aplicación sin afectar otras piezas(KAPLAN-MOSS 2008).

Django es el *framework* que se escoge para la realización del sistema porque presenta múltiples posibilidades en el trabajo con las validaciones, integración con otras herramientas y es muy factible para realizar pruebas y depuración del código.

1.8. IDE de desarrollo.

Para desarrollar, normalmente solo es necesario un editor de texto, un intérprete o compilador y una terminal de líneas de comando, un Entorno de Desarrollo Integrado (IDE). Esto simplifica el trabajo y ahorra tiempo de desarrollo.

NetBeans 8.1.

NetBeans es una aplicación de código abierto diseñada para el desarrollo de aplicaciones portables entre las distintas plataformas. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de

aplicaciones web, control de versiones, entre otras. Es un programa con Licencia Común de Desarrollo y Distribución (CDDL) que permite su uso libremente, brinda el código del mismo y es totalmente gratuita. Permite escribir, compilar, hacer un debut, ensamblar y desplegar aplicaciones y aunque está escrito en Java, brinda soporte para toda clase de lenguajes de programación. Funciona en sistemas operativos compatibles con la máquina virtual Java (Windows XP, Vista, Windows 7, Ubuntu 9.10, Solaris, Mac OS X 10.5 o superior).

PyCharm 2016.

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma libre propiedad de JetBrains, basado en IntelliJ IDEA. Utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades.

PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional, que es igualmente gratuita pero incluye más características como el soporte a desarrollo web(FERNÁNDEZ 2016).

Las características principales de PyCharm son:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con frameworks web como: Django, Flask, Pyramid, Web2Py, entre otros.
- Frameworks javascripts: jQuery, AngularJS.
- Debugger avanzado de Python y Javascript.
- Integración con lenguajes de plantillas: Mako, Jinja2, Django Template.
- Compatibilidad con SQLAlchemy (ORM), Google App Engine, Cython.
- Soporte para modo VIM (Con plugin).
- Sistemas de control de versiones: Git, CVS, Mercurial.

Para el desarrollo de la aplicación informática se seleccionó PyCharm, por permitir escribir, compilar, ensamblar y desplegar aplicaciones, además de brindar soporte para varios lenguajes de programación mediante el empleo de plugins. Facilita el completamiento para los lenguajes de programación HTML, CSS, JavaScript y PHP.

1.9. Sistemas Gestores de Base de Datos (SGBD).

Un SGBD es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Permite almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas, además de ser ampliamente utilizado en entornos científicos con el objeto de almacenar la información experimental(GARZÓN 2010).

PostgreSQL 9.3.

PostgreSQL es un SGBD Objeto-Relacional basado en el proyecto Postgres, de la Universidad de Berkeley. Es una derivación libre de este proyecto y utiliza el lenguaje SQL. Fue pionero en muchos de los conceptos del sistema objeto-relacional actual. Este proyecto lleva más de una década de desarrollo, siendo hoy día, el sistema libre más avanzado, soportando la gran mayoría de las transacciones SQL y control concurrente. Entre sus principales ventajas se destacan las siguientes:

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios Web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendarias.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Es multiplataforma, disponible en la mayoría de los sistemas operativos.
- Permite implementar reglas, vistas, disparadores, subconsultas y funciones.
- Posee herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.

MySQL.

MySQL es un sistema de gestión de bases de datos relacional multiusuario, cuya idea originaria es de la empresa Opensource MySQL AB. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad. Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal y Delphi. Es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas Linux, Windows, PHP, Perl y Python y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

Después de analizar las características y prestaciones de los SGBD mencionados anteriormente, se aprecian las ventajas que brinda PostgreSQL. Este sistema provee de gran capacidad de almacenamiento, consistencia, escalabilidad y rendimiento bajo grandes cargas de trabajo. Es un SGBD objeto-relacional, distribuido bajo la licencia BSD13 y su código fuente se encuentra disponible libremente. Por lo tanto, se decide desarrollar la solución propuesta en esta investigación haciendo uso del SGBD PostgreSQL en su versión 9.3.

1.10. Servidores Web.

Los servicios web tal como lo define el consorcio World Wide Web (WC3) son un conjunto de aplicaciones o de tecnologías con capacidad para inter-operar en la web (World Wide Web, 2014) estas aplicaciones o tecnologías intercambian datos entre sí, con el objetivo de ofrecer servicios.

Servidor Web Apache.

El servidor Apache es desarrollado por la *Apache Software Foundation*, presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, entre sus principales ventajas se encuentran: posee gran cantidad de extensiones para diversas tecnologías, además de una amplia documentación, es libre, modular y multiplataforma. Se señala como desventaja que no posee interfaz gráfica que facilite su configuración.

Este servidor Web es otro de los más ligeros que hay en el mercado. Está especialmente pensado para hacer cargas pesadas sin perder balance, utilizando poca RAM y poca de CPU. Algunas páginas populares que lo usan son YouTube, Wikipedia y otras que soportan gran tráfico diariamente. Es gratuito y se distribuye bajo Licencia BSD.

Se seleccionó el servidor web Apache debido a que consume pocos recursos, es gratuito, es uno de los más usados y presenta una amplia documentación.

Conclusiones del capítulo.

En este capítulo se realizó una compilación de los principales conceptos al dominio del problema, lo que permitió crear un lenguaje común entre el cliente y el autor, caracterizaron los principales sistemas de

reservación de áreas compartidas en Cuba y el mundo, detallando sus funcionalidades principales. Fue seleccionado un conjunto de herramientas y tecnologías para el desarrollo del sistema. De esta manera al concluir el capítulo quedó confirmada la necesidad de crear un sistema adaptado a las condiciones actuales de la UCI y a las necesidades de la Dirección de Servicios Generales (DSG) de aumentar la disminución del margen de errores en el servicio de reservación de las áreas compartidas de la Universidad de las Ciencias Informáticas.

Capítulo 2: Análisis y diseño del sistema.

Introducción.

En este capítulo se define el modelo de dominio y se da una propuesta de solución al problema planteado. Se identifican, especifican y describen los requisitos no funcionales y funcionales, agrupando en casos de uso estos últimos. Se brinda un acercamiento a la implementación a través de la etapa de diseño, construyéndose para ello los diagramas de interacción y el modelo de clases del diseño. Además, se selecciona el estilo arquitectónico y los patrones de diseño a utilizar y se muestra el modelo de datos con la descripción de las tablas de la base de datos.

2.1. Modelo del Dominio.

Un modelo del dominio o modelo conceptual es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Se realiza cuando no se logra determinar el proceso del negocio con fronteras bien establecidas y donde los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos), cuando existe solapamiento de responsabilidades, así como múltiples responsabilidades. Se representa con un conjunto de diagramas de clases UML en los que no se define ninguna operación (Larman, 2004). En la Figura 1 se muestra el Modelo de Domino del Sistema de reservación, seguimiento y control de las áreas compartidas de la UCI.

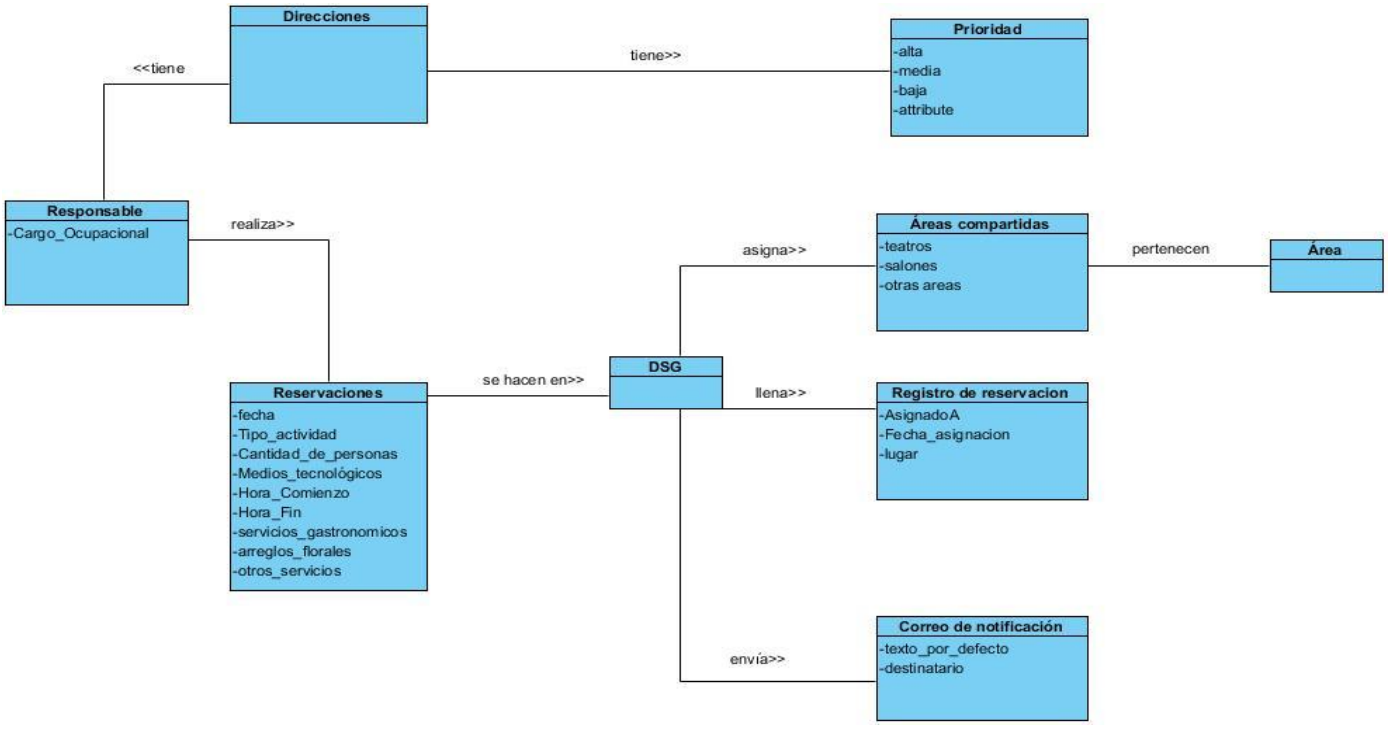

Figura 1: Modelo de Domino del Sistema.

Definición de clases del modelo del dominio.

Direcciones: Son las distintas direcciones de las áreas de la Universidad de las Ciencias Informáticas.

Responsables: Son los encargados de realizar las actividades orientadas por el director(a) del área a la que pertenece.

Prioridad: Está dada por el cargo o condición de un miembro del claustro de la universidad o de alguno de sus restantes trabajadores.

Reservación: Es el acto de solicitar un local o espacio de un área determinada de la Universidad.

Dirección de Servicios Generales (DSG): Es la encargada de tramitar la solicitud de reservación realizada por alguna dirección y de dar respuesta.

Correo de Notificación: Es enviado por la DSG o por el sistema confirmando la solicitud o haciendo aclaración sobre algún cambio realizado en la reservación.

Registro de Reservación: Contiene los datos relacionados a las reservaciones realizadas o solicitadas.

Áreas Compartidas: Son los distintos locales o espacios que se encuentran en las áreas de la Universidad.

Áreas: Hace referencia a las distintas áreas donde se encuentran los locales o espacios compartidos.

Descripción del Modelo de Dominio.

La Universidad de las Ciencias Informáticas (UCI) tiene sus áreas estructuradas por direcciones, estas direcciones tienen responsables encargados de realizar las solicitudes para la utilización de las áreas compartidas de la Universidad, entre las que se encuentran los teatros, los salones, las discotecas u otros espacios abiertos localizados en distintos lugares del centro. Entre las direcciones se encuentra la Dirección de Servicios Generales que es la encargada de tramitar las reservaciones hechas por las restantes direcciones solicitando una local o espacio, también tiene la tarea de realizar seguimientos en las áreas y controlar el estado técnico de las mismas. Las reservaciones contienen los datos de los que las requieren como son nombre, apellidos, cargo del solicitante, también local o espacio requerido, tipo de actividad que se va a realizar, cantidad de participantes, fecha, hora de inicio, hora de fin, entre otros datos según las necesidades de la actividad a realizar.

Cuando una solicitud de reservación llega a la DSG se realiza una revisión de las solicitudes hechas en el Registro de Reservaciones y si ya existe alguna con esa fecha, mismo local o espacio y hora de inicio entonces se revisa el cargo del solicitante y se da prioridad al solicitante con mayor cargo, las prioridades

pueden ser Alta, Media o Baja, la DSG reubica la solicitud en otro local o espacio según la cantidad de personas que asistirán a la actividad y dependiendo de la disponibilidad de lugares libres. Envía un correo a los solicitantes explicando los cambios realizados y la aceptación de la nueva petición y hace los cambios y anotaciones necesarias en el Registro de Reservaciones.

2.2. Levantamiento de Requisitos.

La especificación de los requisitos de software es una descripción completa del comportamiento del sistema que se va a desarrollar. Contiene requisitos funcionales que son las capacidades o condiciones que el sistema debe cumplir, en el que se incluye un conjunto de casos de uso que describen las interacciones que tendrán los usuarios con el software y contiene, además, requisitos no funcionales que son las propiedades o cualidades que el producto debe tener, que imponen restricciones en el diseño o la implementación. En esta etapa se debe generar una información clara y precisa de los aspectos más relevantes del producto, ya que esta actividad es el hilo conductor de todo el desarrollo del software.

2.2.1. Requisitos funcionales.

Los Requisitos Funcionales (RF) definen las condiciones y funciones que el sistema será capaz de realizar. Luego de efectuar la modelación del dominio se obtuvieron los siguientes requisitos funcionales:

Autenticar usuario.

RF 1. Autenticar usuario.

Gestionar usuarios.

RF 2. Adicionar usuario.

RF 3. Eliminar usuario.

RF 4. Actualizar usuario.

RF 5. Buscar usuario.

RF 6. Búsqueda avanzada de usuario.

Gestionar roles.

RF 7. Adicionar rol.

RF 8. Eliminar rol.

RF 9. Actualizar rol.

RF 10. Priorizar rol.

Gestionar locales.

RF 11. Actualizar local.

RF 12. Adicionar local.

RF 13. Buscar local.

RF 14. Eliminar local.

RF 15. Visualizar estado de los locales.

RF 16. Adicionar estado de los locales.

RF 17. Modificar estado de los locales.

Gestionar Áreas.

RF 18. Actualizar área.

RF 19. Buscar área.

RF 20. Adicionar área.

RF 21. Eliminar área.

RF 22. Asignar local a un área.

Otorgar Importancia a las áreas.

RF 23. Ponderar área.

Gestionar reservación.

RF 24. Realizar una reservación.

RF 25. Actualizar una reservación.

RF 26. Asignar servicios al local.

RF 27. Asignar responsable al local.

RF 28. Notificar estado de la reservación.

RF 29. Mover reservación de local.

RF 30. Eliminar reservación.

RF 31. Cancelar reservación.

RF 32. Activar reservación.

Notificar a terceros.

RF 33. Enviar notificación de servicios tercerizados.

Visualizar reportes.

RF 34. Generar reportes de ocupación de las áreas compartidas.

RF 35. Generar reportes de los movimientos de las áreas.

RF 36. Generar reportes de la logística de las áreas.

2.2.2. Requisitos no funcionales.

Los Requisitos No Funcionales (RNF) son propiedades o cualidades, que pueden usarse para juzgar la operación de un sistema en lugar de su funcionalidad. El sistema debe satisfacer las siguientes condiciones o cualidades: que el producto debe tener.

Requerimientos de Software.

RNF 1. El Software se debe ejecutar sobre cualquier plataforma, la PC cliente debe contar con un navegador Web.

Requerimientos de Hardware.

Estaciones de Trabajo:

- **RNF 1.** Se requiere tarjeta de red.
- **RNF 2.** Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- **RNF 3.** Memoria RAM mínimo 512 MB.

Servidores:

- **RNF 4.** Ordenador Dual Core o superior, con 3.0 GHz de velocidad de microprocesador.

RNF 5. Memoria RAM mínimo 2 GB.

RNF 6. Disco Duro con 250Gb de capacidad.

RNF 7. Requiere tarjeta de red.

Requerimientos de Restricciones en el diseño y la implementación.

RNF 8. Servidor de aplicaciones Apache.

RNF 9. Lenguaje de programación PHP y librerías.

Un servidor de BD con los siguientes elementos:

RNF 10. Sistema operativo Ubuntu Server 14.04 o cualquier versión LTS (Long Time Suport, en español, Largo Tiempo de Soporte) superior.

RNF 11. SGBD PostgreSQL, versión 9.3.

RNF 12. En los clientes debe estar instalado un navegador web que cumpla con los estándares de la W3C2, preferiblemente Mozilla Firefox 3.0 o superior.

RNF 13. IDE de desarrollo NetBeans versión 8.1.

Requerimientos de Apariencia o interfaz externa.

RNF 14. Diseño encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.

RNF 15. El diseño debe permitir el uso de los colores azul y blanco. Para la realización del diseño se tendrá en cuenta el control y la transparencia como eje principal en torno al cual girará el sistema.

Requerimientos de Usabilidad.

RNF 16. Los usuarios deben tener un conocimiento básico sobre informática para poder trabajar con el software a desarrollar.

Requerimientos de Portabilidad.

RNF 17. El sistema debe ser multiplataforma.

Requerimientos de Seguridad Confidencialidad.

Confidencialidad:

RNF 18. La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación.

RNF 19. El sistema debe ser accedido solo mediante comunicación segura a través de la red, haciendo uso del protocolo https.

RNF 20. El acceso a las funcionalidades de la solución propuesta debe estar mediado por una jerarquía de roles cumpliendo el principio de mínimo privilegio.

Integridad:

RNF 21. La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes y corrupción., a través de roles y permisos de los usuarios autenticados (R-Back).

Disponibilidad:

RNF 22. A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento.

2.3. Modelo de casos de uso del sistema.

El modelo de casos de uso del sistema sirve para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Permite que desarrolladores de software y clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Está compuesto por un diagrama que muestra las relaciones existentes entre actores y casos de uso del sistema (CUS) y por la especificación de dichos CUS (Larman, 1999).

Casos de Uso del Sistema.

Los CUS representan fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un CUS describe una secuencia de acciones que el sistema debe realizar interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Larman, 2004).

Actores del sistema.

Se considera un actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Por lo general estimula al sistema con eventos de entrada o recibe algo de él. Los actores representan a terceros fuera del sistema que colaboran con el mismo (Larman, 2004). En la tabla 1 se describen los actores que interactúan con el sistema de reservación.

Tabla 1: Actores del Sistema.

Actor	Descripción
Administrador.	Es el administrador del sistema. Encargado de la gestión de los usuarios del sistema, de los roles y de gestionar la información referente a las áreas compartidas.
Responsable de Área.	Realiza la solicitud de la reservación.
Asistente de DSG	Atiende las reservaciones, hace los cambios necesarios, notifica el estado de las reservaciones y notifica a servicios tercerizados.

Diagrama de casos de uso

Un diagrama de casos de uso del sistema (DCU) representa gráficamente a los procesos y su interacción con los actores. Además, facilita el entendimiento de los procesos realizados por el sistema para el desarrollador.

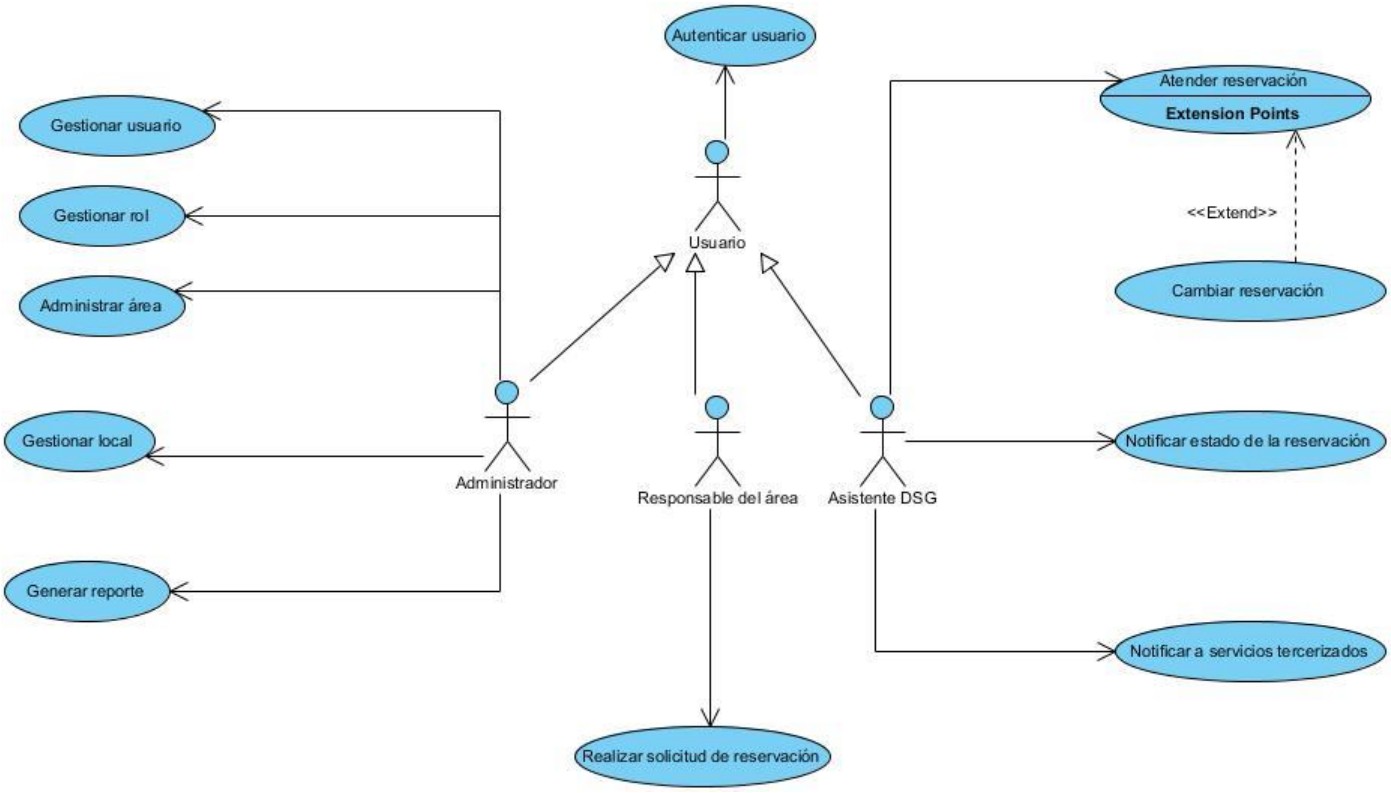

Figura 2: Diagrama de Casos de Uso del Sistema.

Descripción de casos de uso.

Cada CUS se detalla habitualmente mediante una descripción textual que describe la funcionalidad que se construirá en el sistema. En la tabla 2 se muestra la descripción textual del CUS Gestionar locales:

Tabla 2: Descripción de caso de uso: Gestionar locales.

Objetivo	Gestionar locales.
Actores	Administrador (inicia).
Resumen	El administrador solicita la gestión de locales, el sistema muestra una opción en la parte izquierda del menú para la gestión de locales y se muestra la lista de criterios, con la posibilidad de actualizar, buscar, insertar, eliminar, visualizar estado de los locales y modificar estado de los locales.
Complejidad	Alta.
Prioridad	Crítica.
Precondiciones	Debe existir al menos un local para poder modificar o visualizar su estado.
Postcondiciones	Se inserta, actualiza, elimina, busca, visualiza y modifica locales .
Flujo de eventos	
Flujo básico	

	Actor	**Sistema**
1.	Selecciona la opción “Locales” en el menú del lateral izquierdo del sistema.	
2.		Muestra un listado con los locales almacenados y posibilita al administrador realizar algunas de las siguientes acciones: Adicionar local. **Ver Sección 1.** Eliminar local. **Ver Sección 2.** Actualizar local. **Ver Sección 3.**

		Buscar local. **Ver sección 4.** Visualizar estado de los locales. **Ver sección 5.** Adicionar estado de los locales. **Ver sección 6.** Modificar estado de los locales. **Ver sección 7.**
3.	El administrador escoge una de las ocho opciones.	
4.		Finaliza el caso de uso.
Flujo alterno		
Nº 1 "Escoge otra opción."		
	Actor	**Sistema**
1	El administrador indica otra opción del sistema; finalizando el caso de uso.	
Sección 1: "Solicita la adición de un local**"**		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción "Adicionar" de la página principal de las acciones.	
2		Muestra una ventana con un formulario de registro con los siguientes campos: (nombre, tipo de local o espacio, área).
3	Adiciona los datos y da clic en el botón "Aceptar".	
4		Se muestra un mensaje "Local adicionado correctamente".
Flujo alterno		
3.1	El administrador da clic en el botón "Cancelar".	

3.2		Se muestra un mensaje "¿Desea realmente cancelar esta opción?"
3.3	El administrador da clic en el botón "si".	
Sección 2: "Solicita la eliminación de un local**"**		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción "Eliminar" de la página principal de las acciones.	
2		Se muestra una ventana con la lista de locales.
3	Selecciona el local que desee y da clic en el botón "Eliminar".	
4		Se muestra un mensaje "¿Desea eliminar el local seleccionado?"
5	El administrador da clic en el botón "si".	
6		Se muestra un mensaje de notificación "Local eliminado correctamente".
Flujo alterno		
5.1		Se muestra un mensaje "¿Está seguro de que desea eliminar el local seleccionado?"
5.2	El administrador da clic en el botón "no".	
5.3		Cierra el mensaje y no se elimina el local seleccionado.
Sección 3: "Solicita la actualización de un local**"**		
Flujo básico		

	Actor	**Sistema**
1	Selecciona la opción “Actualizar” de la página principal de las acciones.	
2		Se muestra un formulario (nombre, tipo de local o espacio, área).
3	El administrador modifica los campos del formulario del local y da clic en el botón “Aceptar”.	
4		Se muestra un mensaje de notificación “Local actualizado correctamente”.
Flujo alterno		
3.1	El administrador modifica los campos del formulario del local y da clic en el botón “Cancelar”.	
3.2		Se muestra un mensaje “¿Está seguro de que desea cancelar la modificación?”
3.3	El administrador da clic en el botón “si”.	
Sección 4: “Solicita la búsqueda de un local**”**		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción “Buscar” de la página principal de las acciones.	
2		Permite introducir el nombre de un local en el campo de búsqueda.
3	El administrador introduce el criterio de búsqueda del local y da clic en el botón “Buscar”.	
4		El sistema muestra un formulario con el

		local introducido.
Flujo alterno		
4.1		Si no existe el sistema muestra un mensaje de notificación “No existe el local especificado”.
Sección 5: “Solicita visualizar el estado de un local**”**		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción “Visualizar estado” de la página principal de las acciones.	
2		Permite introducir el nombre de un local en el campo de búsqueda.
3	El administrador introduce el criterio de búsqueda del local y da clic en el botón “Buscar”.	
4		Se muestra un formulario con el estado del local seleccionado.
Flujo alterno		
3.1	El administrador da clic en el botón “Cancelar”.	
3.2		Se muestra un mensaje “¿Está seguro de que desea cancelar la búsqueda?”
3.3	El administrador da clic en el botón “si”.	
Sección 6: “Solicita adicionar el estado de un local”		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción “Adicionar estado” de la página principal de las acciones.	

2		Se muestra un listado de locales.
3	El administrador escoge el local deseado.	
4		Se muestra un formulario para introducir el estado del local.
5	El administrador introduce los datos del local y da clic en el botón “Aceptar”.	
6		Se muestra un mensaje de notificación “Estado adicionado correctamente”.
Flujo alterno		
5.1	El administrador da clic en el botón “Cancelar”.	
5.2		Se muestra un mensaje “¿Está seguro de que desea cancelar esta acción?”
5.3	El administrador da clic en el botón “si”.	
Sección 7: “Solicita modificar el estado de un local”		
Flujo básico		
	Actor	**Sistema**
1	Selecciona la opción “Modificar estado” de la página principal de las acciones.	
2		Permite introducir el nombre de un local en el campo de búsqueda.
3	El administrador introduce el criterio de búsqueda del local y da clic en el botón “Buscar”.	
4		Se muestra un formulario con el estado actual del local.
5	El administrador modifica los campos necesarios y da clic en el botón “Aceptar”.	

6		Se muestra un mensaje “El local ha sido modificado satisfactoriamente”.
Flujo alterno		
5.1	El administrador da clic en el botón “Cancelar”.	
5.2		Se cierra la ventana Modificar local y no se guardan las modificaciones realizadas.

2.4. Modelo de Diseño.

El modelo de diseño es un refinamiento y formalización del modelo de análisis, donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado del modelo de diseño son especificaciones detalladas de todos los objetos, incluyendo sus operaciones y atributos (Weitzenfeld, 2005).

El Modelo de Diseño es una disciplina que no se puede obviar en el proceso de desarrollo del software. Es imprescindible para comprender la forma en que va a funcionar el sistema en conjunto con los requisitos, lenguajes de programación, componentes reutilizables y tecnologías de interfaz de usuario que se eligieron para el desarrollo del mismo. Es una representación gráfica, mediante varios diagramas muy explícitos, de la implementación del sistema.

Diagrama de clases del diseño.

Los diagramas de clases muestran cómo se lleva a cabo la colaboración entre las clases para dar cumplimiento a un requisito determinado. Para la elaboración de estos diagramas se hace uso de estereotipos que ayudan a representar de manera más fácil la función y el carácter de las clases dentro de la realización del requisito. A continuación, se muestra el diagrama de clases del diseño del caso de uso del sistema gestionar usuario y gestionar componentes.

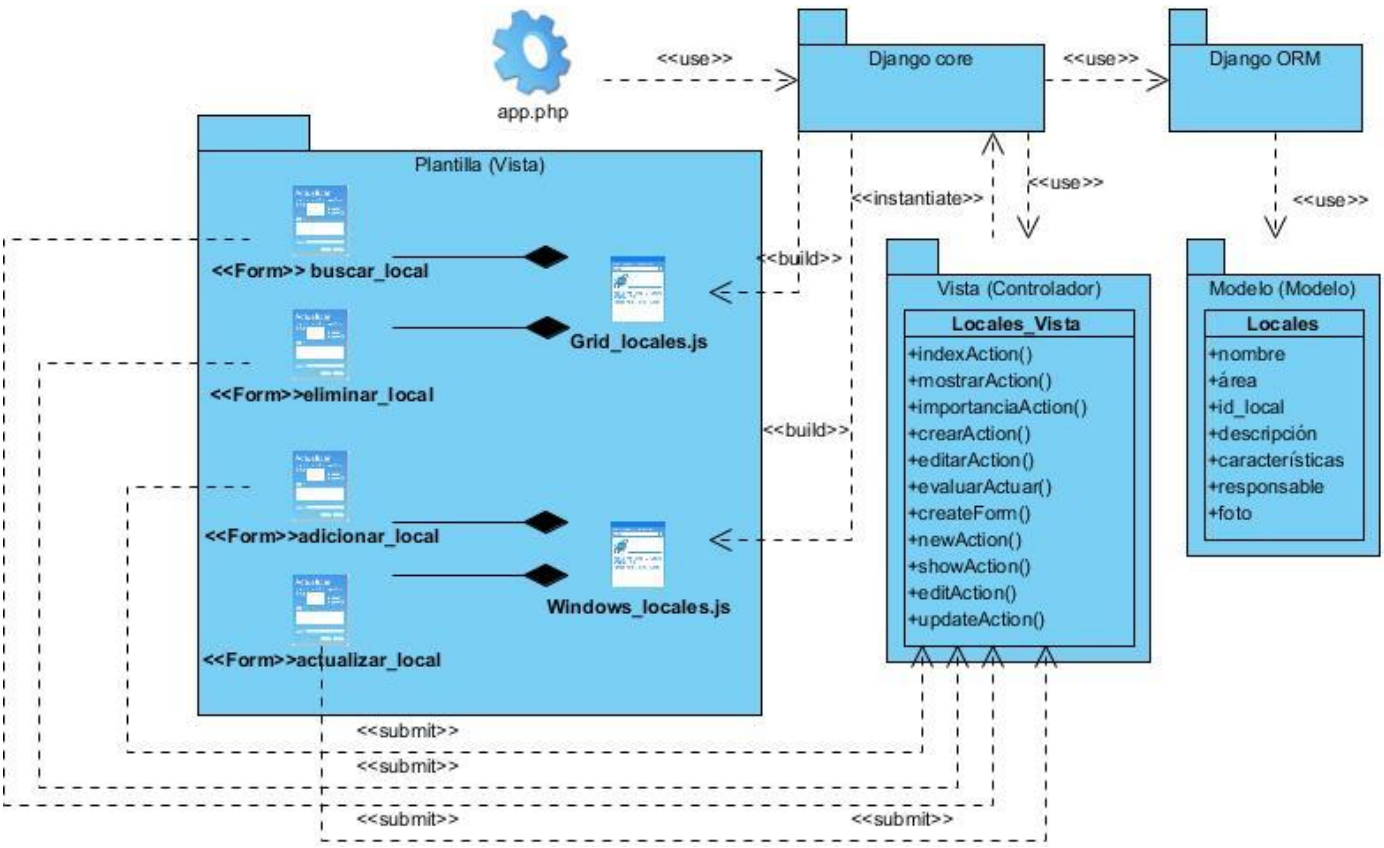

Figura 3: Diagrama de Clases del Diseño: Gestionar locales.

Descripción de patrones arquitectónicos.

Los patrones ayudan al arquitecto a definir la composición y el comportamiento del sistema de software, y una combinación adecuada de ellos permite alcanzar los requerimientos de calidad. A continuación, se describen los utilizados en el desarrollo de la propuesta de solución.

Patrones arquitectónicos.

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone que son plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software. (Buschmann F et al. 1996).

Los patrones arquitectónicos:

- Definen la estructura básica de una aplicación.
- Pueden contener o estar contenidos en otros patrones.

- Proveen un subconjunto de subsistemas predefinidos, incluyendo reglas y pautas para su organización.
- Son una plantilla de construcción.

Patrón arquitectónico Modelo – Vista – Controlador.

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

La necesidad del manejo de la arquitectura de un sistema de software nace con los sistemas de mediana o gran envergadura, que se proponen como solución para un problema determinado. En la medida que los sistemas de software crecen en complejidad, dado por el número de requerimientos o por el impacto de los mismos, se hace necesario establecer medios para el manejo de esta complejidad. En general, la técnica es descomponer el sistema en componentes que agrupan aspectos específicos del mismo y que al organizarse, de cierta manera constituyen la base de la solución de un problema en particular.

MVC son las siglas de Modelo Vista Controlador, que es un patrón de arquitectura de software cuya función es subdividir una aplicación en tres módulos que corresponden a la vista del usuario (la interfaz a la que accede el usuario), una lógica de control para captar los eventos que el usuario ha generado a través de la interfaz, y un modelo que gestiona los datos según le indique la lógica de control.

Modelo: Esta es la representación específica de la información con la cual el sistema opera y se compone por el Sistema de Gestión de Base de Datos y la lógica de negocio. La lógica de negocio asegura la integridad de estos y permite derivar nuevos datos. El Sistema de Gestión de Base de Datos (SGBD) será el encargado de almacenar los cambios en los datos (agregar datos, editarlos o borrarlos) producidos por la lógica de negocio; ejemplos de SGBD son MySQL, Oracle... Es recomendable una capa de abstracción extra denominada Data Access Object (DAO), que es un componente de software que suministra una interfaz común entre la lógica de negocio y el SGBD.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Por lo tanto, la vista es la encargada de presentar los datos al usuario y la interfaz necesaria para modificarlos. Un ejemplo de tecnología podría ser las JSP que, mediante el servidor, genera HTML que interpreta el navegador del usuario mostrándole los datos y los formularios que constituyen la vista para que pueda interactuar con la aplicación.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Por lo general, el controlador sería la unidad central que comunica la vista con

el modelo y viceversa, asociando los eventos del usuario con los cambios que se producirán en el modelo y devolviendo los datos resultantes que genere el modelo a la vista que corresponda.

El patrón arquitectónico Modelo – Vista – Controlador (MVC) divide una aplicación interactiva en tres componentes. El "modelo" contiene la información central y los datos. Las "vistas" despliegan información al usuario. Los "controladores" capturan la entrada del usuario. Es el patrón arquitectónico utilizado por Django, separando la vista de las aplicaciones de la lógica del negocio y del controlador.

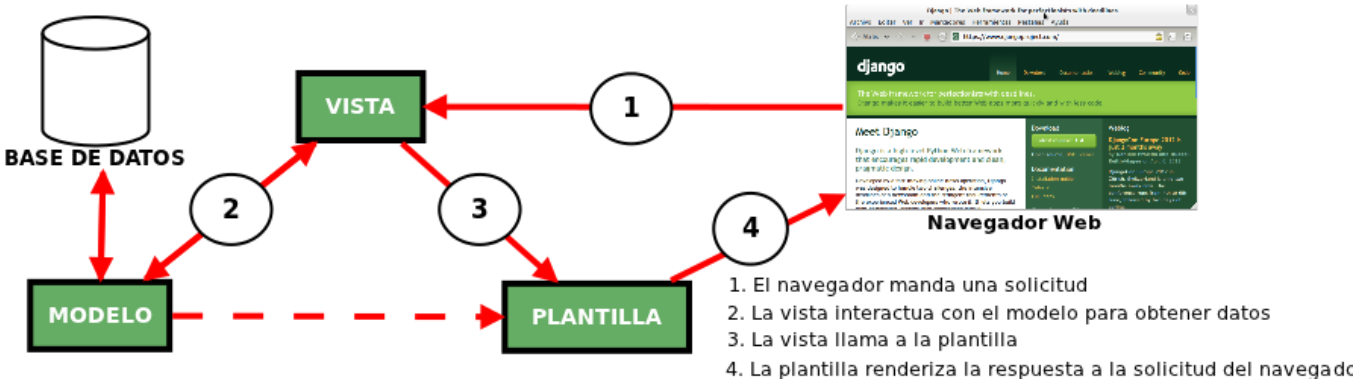

Figura 4: Patrón arquitectónico Modelo Vista Controlador.

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

- M: es la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.
- V: es la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- C: es la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la (C) es manejada por el mismo framework y la parte más emocionante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV. En el patrón de diseño **MTV:**

- M: significa (Model) (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

- T: significa (Template) (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V: significa (View) (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y las plantillas.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario, por ejemplo, al pulsar un enlace). Este punto corresponde en el patrón a la vista.
- El controlador recibe (a través de la interfaz) la notificación de la acción solicitada por el usuario. Es decir, el controlador gestiona el evento que llega desde la vista producida por un usuario.
- El controlador accede al modelo, ya sea con el fin de consultar datos o actualizarlos, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado de las películas que tal usuario tiene). El modelo no debe tener conocimiento directo sobre la vista. Por lo general, el controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice. Sin embargo, en algunas implementaciones, la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Ninguna de las interpretaciones es más correcta que otras. Lo importante es entender los conceptos subyacentes.

Patrones de diseño.

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. (Buschmann F et al. 1996).

Patrones GRASP.

Los Patrones de Software para la Asignación General de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Cardoso E, 2004).

1. **Creador:** Asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro del sistema este patrón se evidencia en las acciones de los controladores, las cuales crean objetos del modelo o los formularios que representan las entidades.

2. **Bajo Acoplamiento:** Determina el nivel de dependencia de una clase con respecto a otras. Una clase con bajo acoplamiento no depende de muchas otras. Este patrón es utilizado por el framework Symfony, y por ende en el sistema, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.

3. **Controlador:** Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones. Se evidencia el uso de este patrón, ya que para cada petición o evento que se genere en el mismo, existe un controlador con la responsabilidad de obtenerla y devolver una respuesta. La respuesta puede ser mostrar una vista, ejecutar un método, devolver un mensaje, etc.

Patrones GoF.

Los patrones GoF (Gang of Four), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Nos permiten crear grupos de objetos para ayudarnos a realizar tareas complejas.

Existen tres tipos de patrones:

De Creación: Abstraen el proceso de creación de instancias.

1. **Singleton:** Asegura que solamente una instancia de una clase es creada. Todos los objetos que usan una instancia de esa clase usan la misma instancia.
2. **De Comportamiento:** Atañen a los algoritmos y a la asignación de responsabilidades entre objetos
3. **Observer:** Se utiliza para mantener “informados” a objetos desacoplados entre sí y que presentan dependencia de un objeto, notificando a los primeros ante cualquier cambio que ocurra en el último. Este patrón es utilizado tradicionalmente en la capa de presentación en el diseño de componentes para las interfaces de usuario de las aplicaciones.

4. **Estructurales:** Se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
5. **Decorator:** Añade responsabilidades adicionales a un objeto dinámicamente. Se utiliza este patrón para las vistas y los *layout's* o plantillas globales que decoran el contenido de la misma.
6. **Facade:** Simplifica el acceso a un conjunto de objetos relacionados al proporcionar un objeto al que todos los objetos fuera del conjunto utilicen para comunicarse con el conjunto.

2.5. Modelo de datos.

El diagrama Entidad-Relación (DER) tiene como objetivo representar mediante una notación gráfica los objetos de datos y sus relaciones. Este define todos los datos que se introducen, se almacenan, se transforman y se producen dentro de una aplicación (Pressman, 2002).

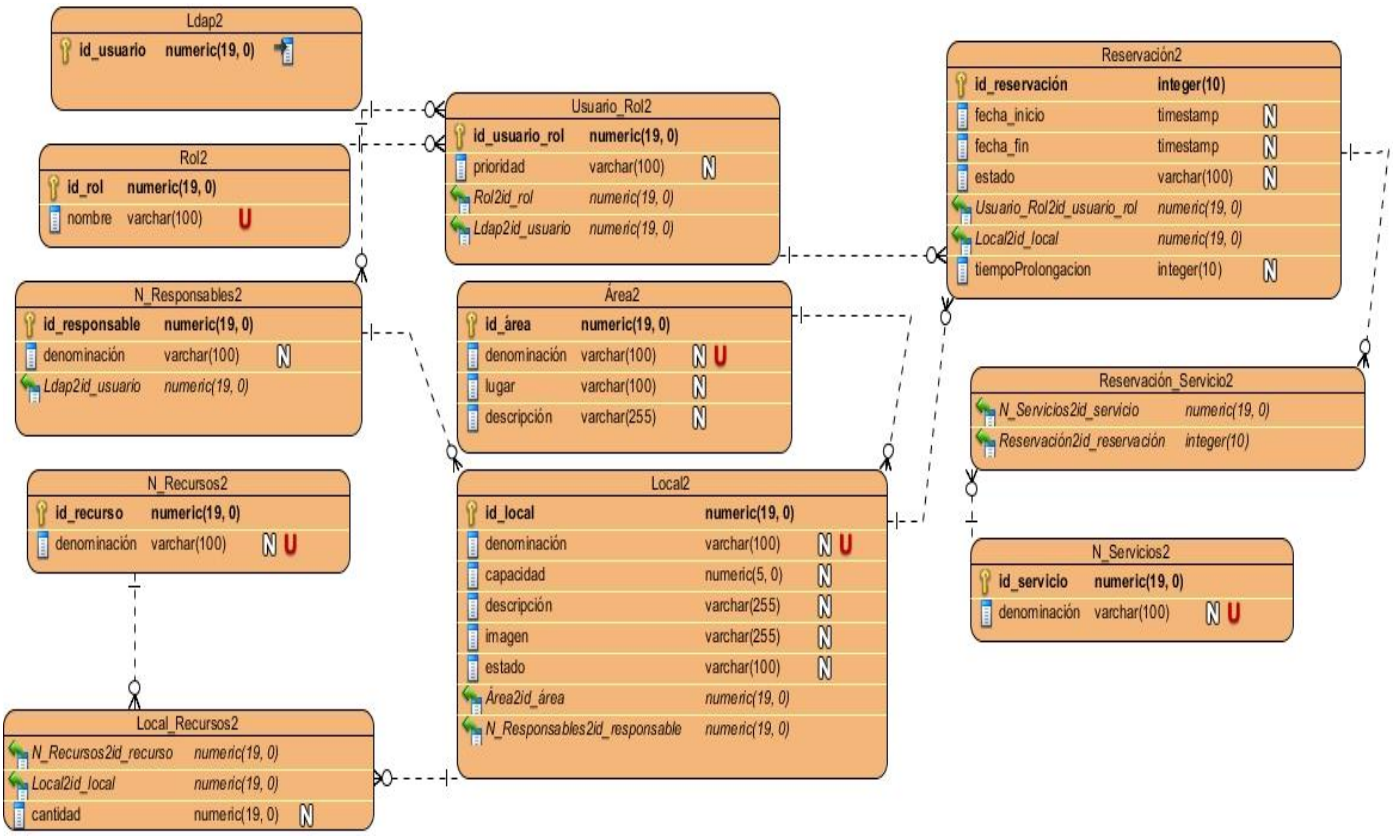

Figura 5: Modelo de datos.

Conclusiones del capítulo.

Con el modelado del dominio, la obtención de los Requisitos Funcionales y No Funcionales y el modelado del sistema, se propicia el entendimiento entre el equipo de desarrollo y el cliente en función de lo que el sistema debe realizar y las características que debe poseer. Como resultado del presente capítulo se

obtuvieron además los artefactos correspondientes al flujo de trabajo de diseño, el cual sirve como material de referencia para futuras ampliaciones y modificaciones del sistema como guía para comenzar su implementación.

Capítulo 3: Implementación y prueba del sistema.

Introducción.

En este capítulo se detallan las fases finales de la construcción del sistema, abordando artefactos fundamentales dentro de la implementación, con el objetivo de obtener un producto que cumpla con los requisitos y que su aplicación sea de manera satisfactoria. Se hace necesaria la comprobación para asegurar que satisface todas las expectativas y cumple con las funcionalidades esperadas, con tal objetivo se introducen, un grupo de ideas fundamentales sobre el proceso de pruebas de la propuesta de solución, abordando las diferentes tipologías de las mismas, centradas en su aplicación sobre sistemas Web.

3.1. Modelo de Implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. También muestra cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. El modelo brinda una mejor visualización del camino a seguir por los desarrolladores del sistema en el momento de enfrentar la implementación del mismo (Jacobson, y otros, 2007).

3.2. Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Se realizan con el objetivo de poseer una vista de forma general del sistema a partir de las dependencias e integraciones de los componentes y módulos.

- En el paquete controlador se encuentran las clases controladoras, encargadas de manejar las peticiones de los usuarios a través de los métodos que tienen implementados.
- El paquete modelo agrupa las entidades del sistema, a través de las cuales se realiza el acceso a la base de datos y el paquete vista, agrupa los archivos que permiten visualizar las respuestas que devuelven los controladores al usuario.

A continuación, se muestra el diagrama de componentes para Reservaciones, donde se encuentran los principales componentes del sistema a desarrollar.

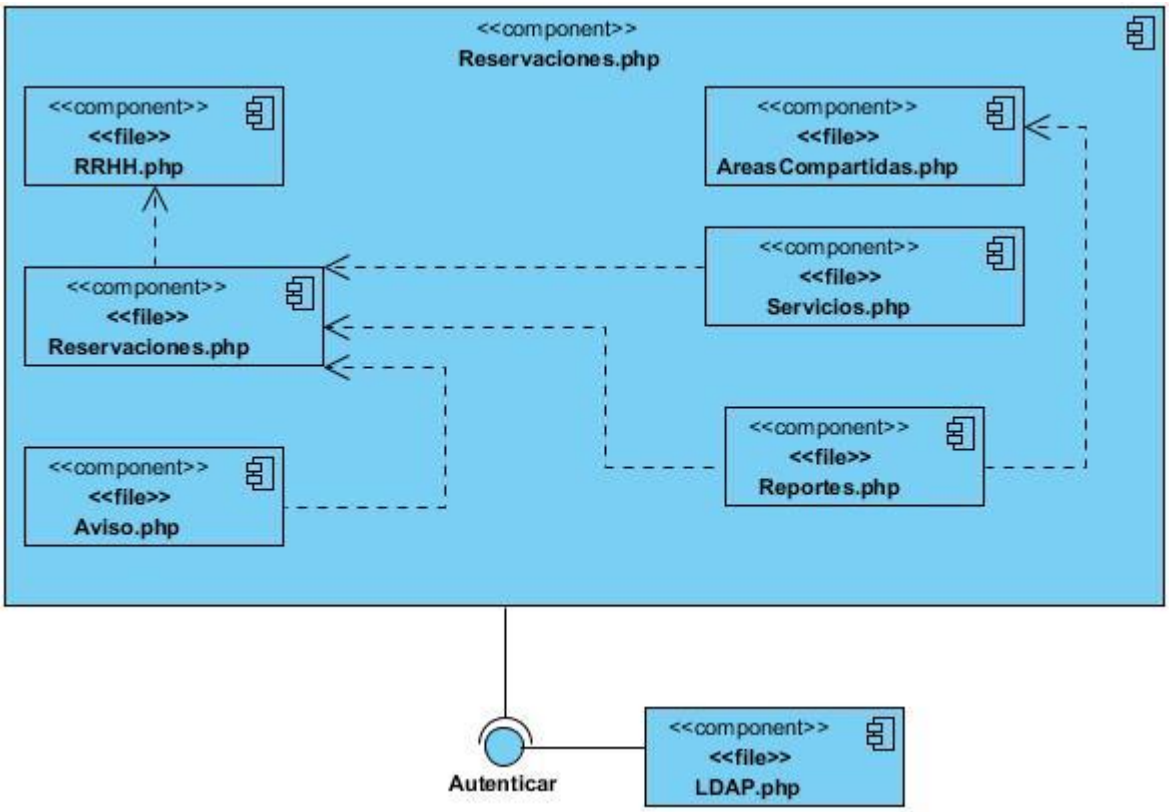

Figura 6: Diagrama de componentes: Reservaciones.

3.3. Código fuente.

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos (Lasso, Iván 2008).

Estándares de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Para facilitar el entendimiento del código y fijar un modelo a seguir, se establecieron estándares de codificación. A continuación, se muestran algunos de estos estándares para el lenguaje PHP utilizados en Symfony2.

Nombres de clases y métodos.

Para la definición de las clases y métodos en el código de la aplicación fue utilizado el estándar. Este es un estilo de escritura que se aplica a frases o palabras compuestas.

```
public function indexAction() {...9 lines }

 public function pdfAction() {...14 lines }

public function mostrarAction() {
    try {...12 lines } catch (\Exception $exc) {...7 lines }
}

public function createAction(Request $request) {...21 lines }

public function editarAction(Request $request) {...24 lines }

public function borrarAction() {...16 lines }

public function evaluarAction() {
```

Figura 7: Estándar de codificación CamelCase.

```
class CriterioController extends Controller {
```

Figura 8: Estándar de codificación UpperCamelCase.

Estructura.

- El código debe usar cuatro espacios en vez de usar el tabulado. Esto minimiza problemas con otras herramientas de desarrollo.
- Las líneas podrían tener 80 caracteres o menos, evitando tener más de 120 caracteres.
- Las llaves de apertura deben ir en la siguiente línea y la llave de cierre debe ir en la siguiente línea después del cuerpo.
- Las llaves de apertura en las estructuras de control deben ir en la misma línea y las llaves de cierre deben de ir después del cuerpo.
- Los paréntesis en las estructuras de control no deben usar espacios antes o después.
- Añadir un solo espacio después de cada limitador de coma.
- Añadir un solo espacio alrededor de los operadores (==, &&…).
- Usa llaves para indicar el control de la estructura sin tener en cuenta el número de declaraciones que el grupo pueda contener.

- Definir una clase por fichero.
- Declarar las propiedades de clase antes que los propios métodos de clase.

3.4. Vista de Despliegue.

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes y objetos que residen en ellos. Está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema.

PC Cliente: Representa las computadoras clientes que se conectan al servidor de aplicaciones, las mismas se comunican con el servidor a través del protocolo seguro HTTP.

Servidor web: Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo IDO.

Servidor de Base de datos: Es donde se almacena toda la información de la aplicación.

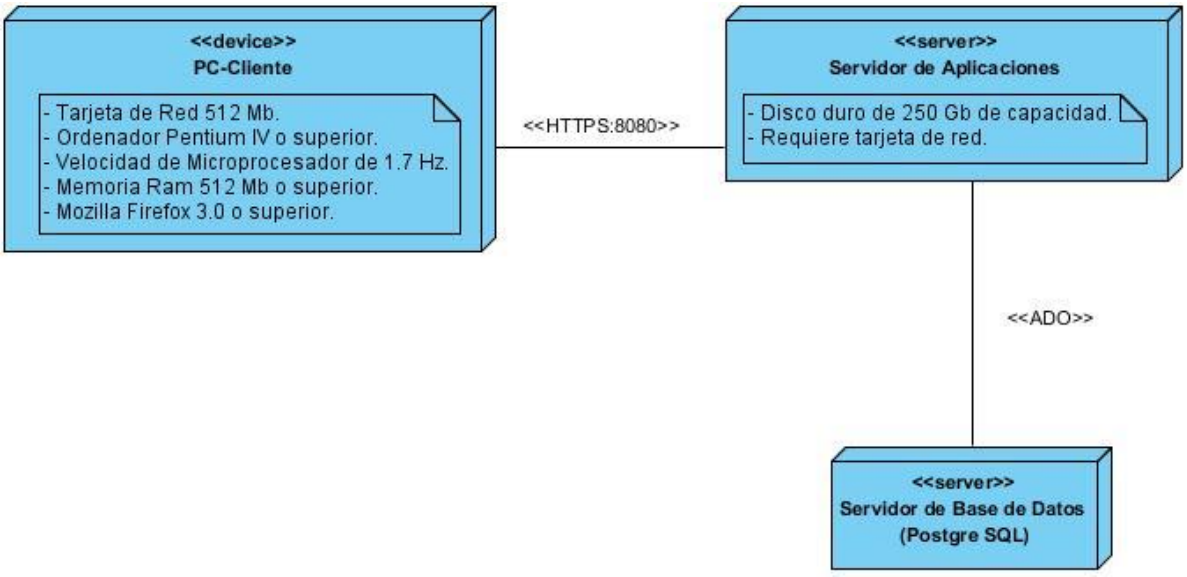

Figura 9: Diagrama de despliegue.

3.5. Pantallas principales de la aplicación.

La interfaz de una aplicación permite el flujo de información entre el usuario y el sistema. A continuación, se muestran algunos ejemplos de las interfaces del sistema desarrollado:

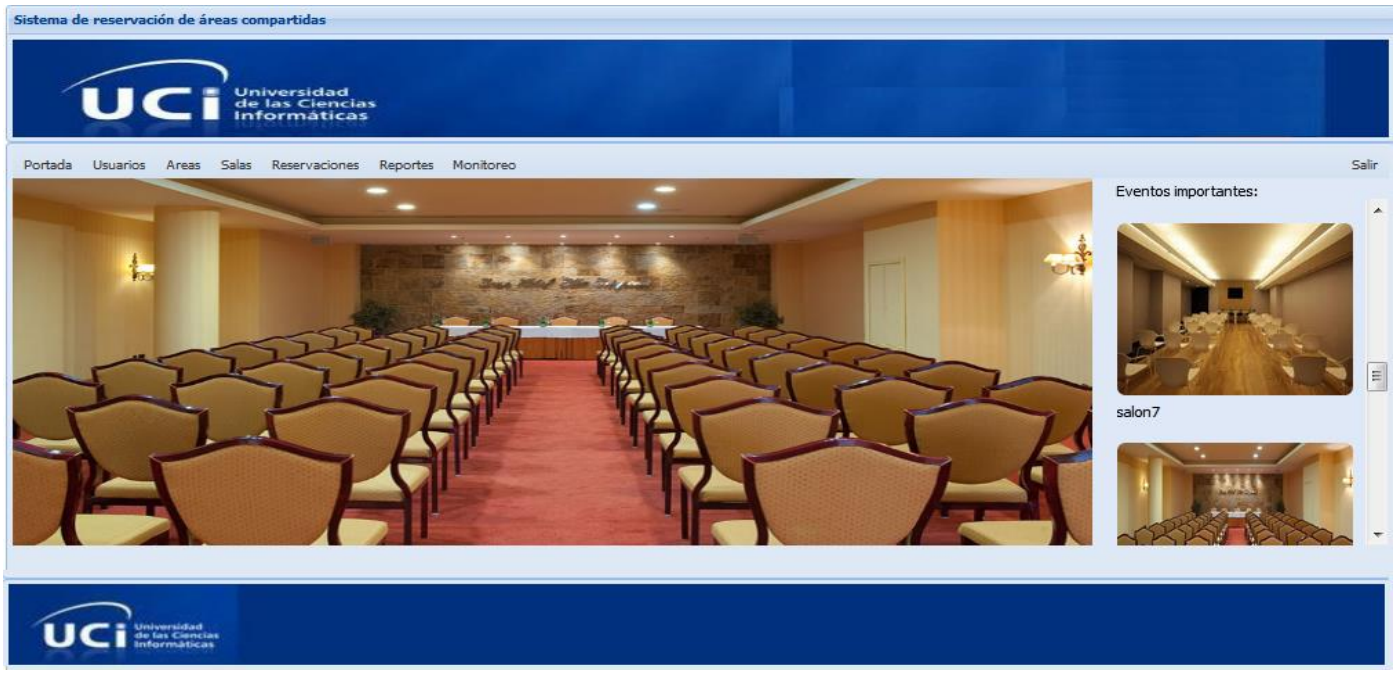

Figura 10: Pantalla de Bienvenida del sistema

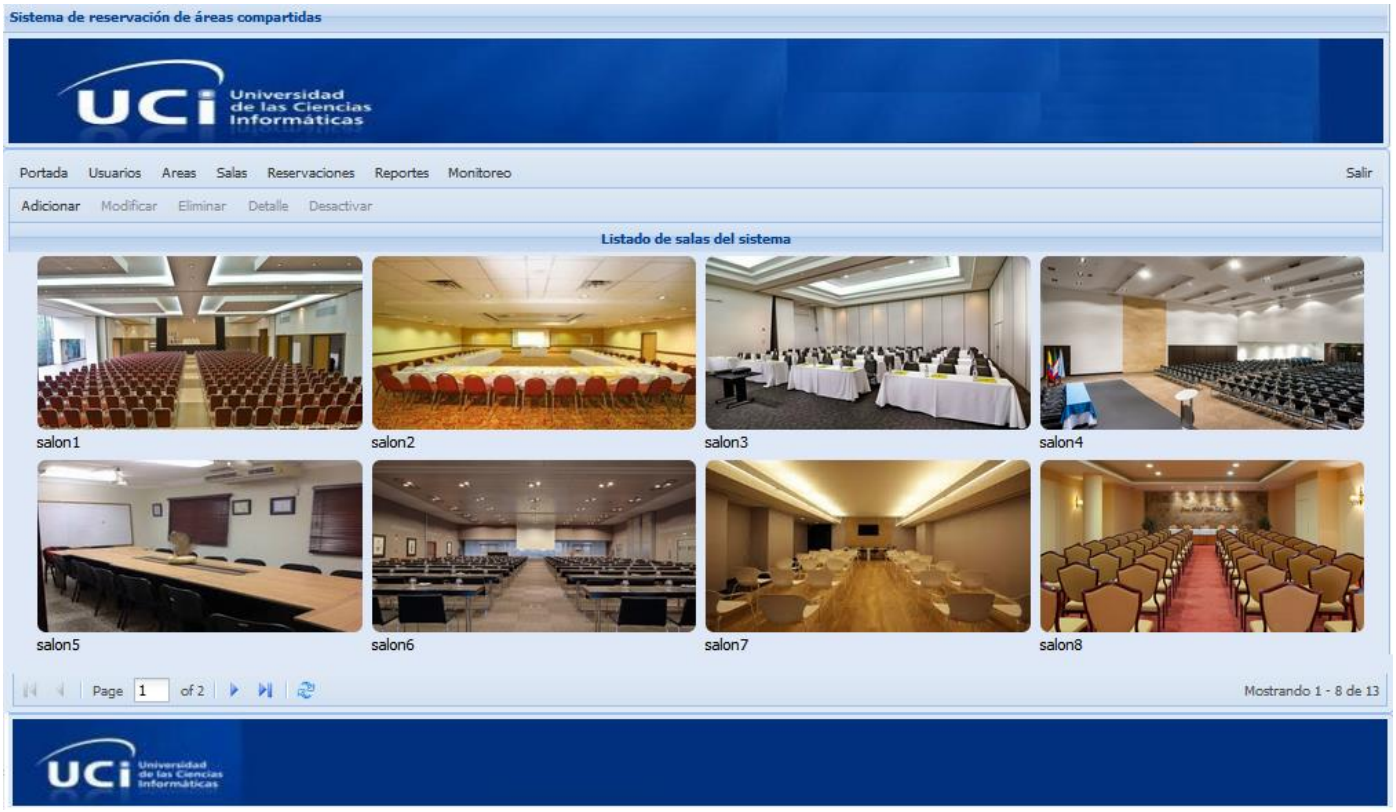

Figura 11: Gestionar áreas compartidas.

3.6. Pruebas al sistema.

Una vez terminada la implementación del producto que se requiere es necesario realizarle pruebas con el objetivo de detectar errores en la aplicación y la documentación; este proceso resulta de gran importancia ya que da una medida de la calidad del mismo siempre que se lleve a cabo de la forma correcta. A continuación, se muestran las pruebas realizadas al sistema y los resultados obtenidos por cada una.

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010).

Niveles de Prueba.

Los niveles de prueba son diferentes ángulos de verificar y validar en determinados momentos el ciclo de vida del software. Existen diferentes niveles de pruebas como: pruebas funcionales, de sistema y de aceptación. En el desarrollo de la fase de pruebas del sistema para la reservación y seguimiento de las áreas compartidas de la Universidad de la Ciencias Informáticas se aplicarán las pruebas que abarcan los siguientes niveles:

- **Desarrollador:** Se realiza con el objetivo de detectar errores en la implementación de requerimientos.

Tipos de Prueba.

Existen diferentes tipos de pruebas que se pueden aplicar para comprobar el correcto funcionamiento de la aplicación, se seleccionaron los que a continuación se muestran:

Pruebas de Funcionalidad: Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Métodos de Prueba.

Pruebas funcionales o de caja negra.

Las pruebas de caja negra, también denominadas pruebas funcionales se centran en los requisitos funcionales del software, es decir, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

- Las pruebas funcionales que se realizarán a la solución, estarán enfocadas o dirigidas a los casos de uso del sistema para verificar su correcto funcionamiento.
- En este tipo de pruebas se ejecutarán los distintos servicios prestados con datos correctos e incorrectos; en caso de que los datos sean incorrectos se verificará que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados (Manas,1994).

A continuación, se muestra el caso de prueba adicionar local.

Tabla 3. DCP Adicionar Local.

Escenario	Identificador	Nombre	Tipo de Local	Área	Descripción	Respuesta del sistema	Flujo Central
EC 1.1 Adicionar local correctamente aplicando el adicionar	1	V(D-Doc.6)	V(Discoteca)	V(Docente 6)	V(Se adiciona un local y se presiona el botón aceptar)	Se muestra el mensaje¨ *El local ha sido adicionado satisfactoriamente*¨. Se cierra el formulario.	1. Seleccione la opción Local del menú de opciones. 2. Selecciona la opción Adicionar. 3. Se llenan los campos: Se presiona la opción Adicionar. Se validan los campos. Se adiciona el local al sistema.
EC 1.2 Adicionar Local correctamente aplicando el aplicar	6	V(T-Doc. 5)	V(Teatro)	V(Docente 5)	V(Se adiciona un local y se presiona el botón Aplicar)	Se limpian los campos para adicionar nuevo local.	1. Seleccione la opción Local del menú de opciones. 2. Selecciona la opción Adicionar. 4.Se llenan los campos: Se presiona la opción Aplicar. Se validan los campos. Se aplican los cambios del local al sistema.

EC1.3 **Adicionar persona incorrectamente.**	1	V(T-Doc. 5)	V(Teatro)	V(Docente 5)	Se muestra un mensaje según el error cometido.	Si ya existe un local con ese identificador se muestra el mensaje *¨El id introducido pertenece a otro local¨.*	1. Seleccione la opción Local del menú de opciones. 2. Selecciona la opción Adicionar. 3. Se llenan los campos: Se presiona la opción Adicionar. Se validan los campos.
	2	V(T-Doc. 4)	I(Fac-6)	V(Docente 4)	Se muestra un mensaje según el error cometido.	Si el nombre contiene números no es válido se muestra el mensaje *¨El nombre no es válido ¨.*	
	3		V(14583)	V(114447)	Se muestra un mensaje según el error cometido.	Si existen datos vacíos se muestra el mensaje *¨Hay campos vacíos ¨.*	
EC1.4 **Adicionar persona correctamente aplicando el Cancelar**	NA	NA	NA	NA	NA (Se presiona el botón Cancelar)	Se muestra un mensaje ¨Desea realmente cancelar ¨.	1. Seleccione la opción Local del menú de opciones. 2. Selecciona la opción Adicionar. 3. Se llenan los campos: Se presiona la opción Cancelar. Se cierra el formulario.

Resultados e interpretación de las pruebas.

El sistema implementado cuenta con un total de 36 requisitos funcionales, para probar cada uno de estos requisitos fue necesario elaborar un diseño de casos de prueba. Cada uno de las No Conformidades (NC) detectadas fueron registradas en el registro de NC del expediente de proyecto de la UCI y fueron debidamente clasificadas.

Se realizaron un total de 3 iteraciones de pruebas. Una primera iteración arrojó un total de 25 no conformidades clasificadas como se muestra en el siguiente gráfico:

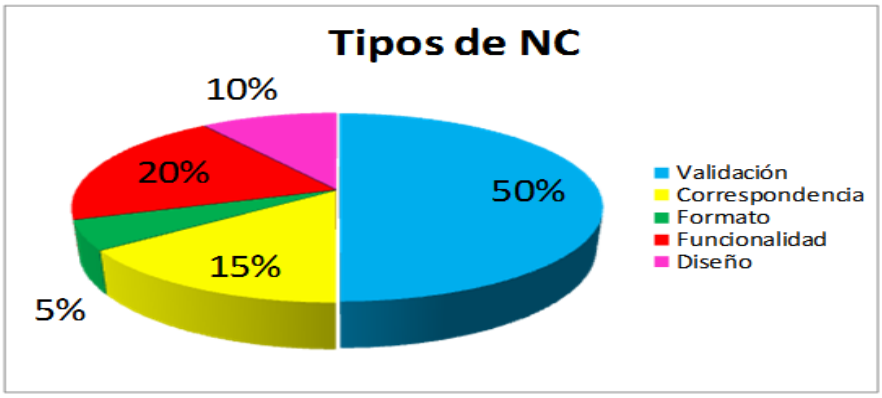

Figura 12: Tipos de No Conformidades 1ra Iteración.

Las 25 NC detectadas fueron debidamente resueltas y se decidió hacer una segunda iteración de pruebas con el objetivo de verificar que no quedaran errores sin ser detectados y revisar que los cambios introducidos al resolver las NC de la primera iteración no habían introducido nuevos errores.

En la segunda iteración se encontraron 10 NC clasificadas de la siguiente manera:

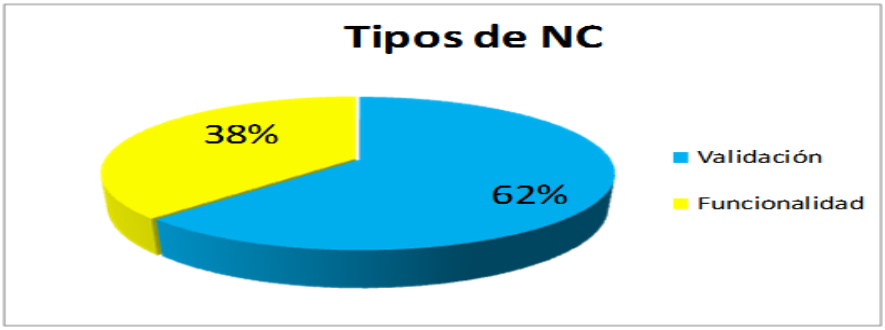

Figura 13: Tipos de No Conformidades 2da Iteración.

La tercera iteración de pruebas se realizó exitosamente ya que no se detectaron NC, garantizando de esta manera que fueran satisfactorios el total de requisitos especificados por el cliente. Los resultados de las 3 iteraciones se muestran en el siguiente gráfico:

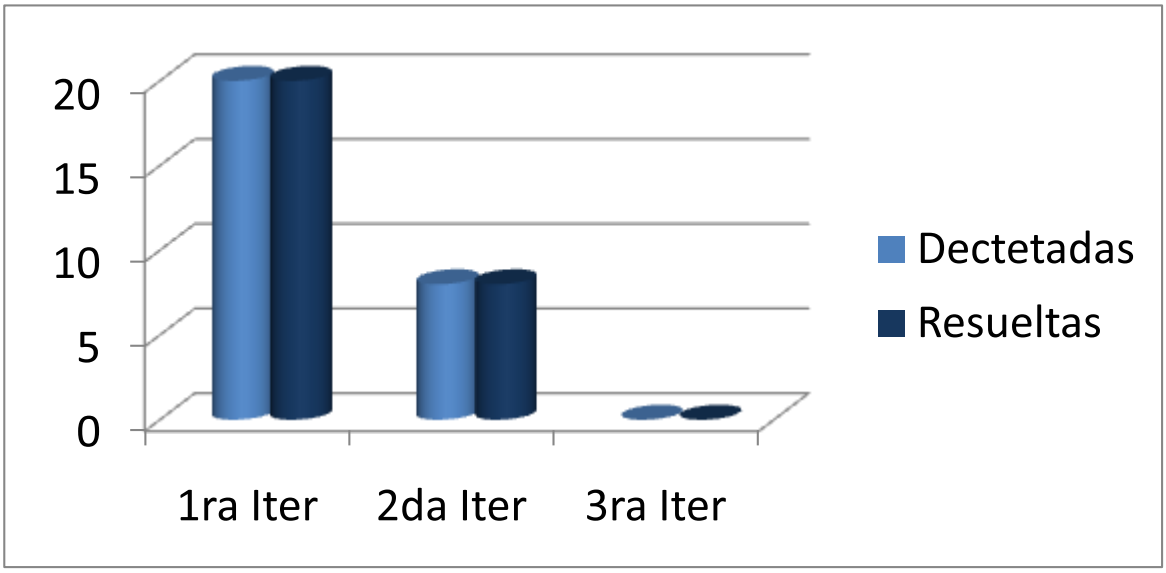

Figura 14: Resultados de las iteraciones.

3.7. Resultados alcanzados en la investigación.

Con la implementación del sistema para la gestión de los recursos humanos se obtuvo una herramienta que:

1. **Locales:** Permite llevar el control de los locales para un seguimiento de su estado físico y material, así como introducir en el sistema todos los cambios que se realicen en los mismos y adicionar nuevos locales si se construyeran en cualquier área de la Universidad.
2. **Reservaciones**: Genera las reservaciones de los locales que se encuentran en las distintas áreas de la Universidad, permite adicionar las reservaciones especificando el tiempo deseado de inicio y fin así como la posibilidad de terminar y liberar un local antes del tiempo previsto o de extender la reservación si existe la opción en ese mismo lugar reservado o sugiere otros con las mismas características que estén libres en ese espacio de tiempo, así como también brinda la posibilidad de solicitar los medios y servicios que se desean utilizar.
3. **Generación de reportes:** Genera reportes estadísticos que apoyan a la toma de decisiones en tiempo real. Estos reportes muestran información actualizada del estado de las reservaciones, las incidencias y el estado de las mismas.

4. **Usuarios y Roles:** Permite asignar roles y permisos a los usuarios del sistema asignándole una prioridad al usuario según su nivel de importancia para tenerlo en cuanta a la hora en que se realice una reservación que coincida en tiempo y espacio con otra. Esto permite facilitar la toma de decisiones en cuanto a nivel de importancia de una reservación.

Conclusiones del capítulo

En el presente capítulo se obtuvo el modelo de componentes que aporta claridad ante el proceso de implementación del sistema, así mismo el modelo de despliegue permite conocer el ambiente en el que debe desempeñarse la propuesta solución, se realizaron las pruebas de software para el futuro despliegue del sistema; tras el flujo de implementación y prueba, el sistema quedó desarrollado. Las pruebas realizadas permitieron identificar y corregir los errores, garantizando un producto de alta calidad el cual cumple los requisitos planteados, tanto funcionales como no funcionales.

Conclusiones

Una vez culminada la investigación se puede afirmar que se les dio cumplimiento a los objetivos planteados, arribando a las siguientes conclusiones:

- El estudio de los principales conceptos relacionados con los sistemas de gestión de información permitió sentar las bases para el desarrollo del Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas.
- A partir de la realización del análisis y diseño del Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas se obtuvieron como resultado los diagramas y artefactos necesarios para guiar el desarrollo del mismo.
- La implementación del Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas dio cumplimiento a los 36 requisitos funcionales identificados en las fases de análisis y diseño.
- El diseño y ejecución de las pruebas a nivel de componentes y de desempeño permitió comprobar el correcto funcionamiento del Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas y el cumplimiento de las funcionalidades incorporadas.
- Como resultado se obtuvo el Sistema para la reservación y seguimiento de áreas compartidas en la Universidad de las Ciencias Informáticas que permite realizar y monitorear las reservaciones realizadas por los miembros de la institución de cualquier área de la Universidad.

Recomendaciones

Se recomienda integrar las notificaciones del sistema con un servidor de correo electrónico para que la información sea enviada por el sistema directamente a cada usuario.

Referencias bibliográficas

BUSCHMANN F, MEUNIER, R., ROHNERT, H., SOMMERLAND, P. y STAL M 1996. Pattern – Oriented Software Architecture. A System of Patterns. Londres.

CARDOSO E, CAMACHO F y NUÑEZ G 2004. Arquitecturas de software. Guía de estudio.S.l.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009a. FolletoSistemaCI, Artículo 3.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009b. Folleto Sistema CI, Artículo 6.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009c. Folleto SistemaCI, Resolución No. 60/11. La Habana.

ECLIPSE FOUNDATION. **2011**. OpenUp Basic... S.l.

EGUILUZ JAVIER 2013. Introducción a AJAX.

EGUILUZ JAVIER 2010. Desarrollo web ágil con Symfony 2.3.

FLORES C y **ALFERÉZ**, **G. 2011**. Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navojoa Usando Open UP. México:

Framework-**Ecured**. [en línea] 2014. [Consulta: 24 febrero 2015]. Disponible en: http://www.ecured.cu/index.php/Framework.

GOBIERNO DE LA REPÚBLICA DE CUBA 2009. Bases para la constitución de la República. [en línea]. S.l.: [Consulta: 23 octubre 2014]. Disponible en: http://www.contraloria.cu/.

GOLDSTEIN A, **WEYL E y LAZIRIS L 2011**. HTML5 & CSS3 for the Real World... S.l.: SitePoint.

GONZALEZ ENRIQUE 2010. Aprender a Programar.

HERNÁNDEZ R A y COELLO S 2002. *El paradigma cuantitativo de la investigación científica.* La Habana: Editorial universitaria.

JACOBSON 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid: Pearson Educación S.A. ISBN 8478290362.

LARMAN, **C**. **2009**. *UML y patrones. Una introducción al análisis y diseño orientado a objetos.* 2. S.l.: s.n.

LOBOS MARIA 2014. Emagister. *www.emagister.com* [en línea]. [Consulta: 23 noviembre 2014]. Disponible en:http://www.emagister.com/curso-aprende-programar/concepto-lenguaje-programacion.

LOPERA C, **GUTIERREZ E y MARTÍN JC 2003**. *Indicadores: ciencia y tecnología en países de América Latina.* Lecturas de Economía. S.l.: s.n.

MARIN ALEJANDRA 1998. Nuevos conceptos del control interno. Valencia.

MAR OMAR 2014. Procedimiento para determinar el índice de control organizacional., vol. 18, no. 2.

MC GRAW, H. **2001**. *Ingeniería de software. Un enfoque práctico.* 5. España: s.n. ISBN 8448132149.

MINISTERIO DE FINANZAS 2003. Resolución del Ministerio de Finanzas y Precios. [en línea]. La Habana: [Consulta: 12 noviembre 2014]. Disponible en: http://www.contraloria.cu/.

NEGRÍN SIMEÓN 1997. La ciencia y la tecnología en Cuba, Revista Cubana de Medicina Tropical., vol. 49, no. 3, pp. 153-160.

R. I. **DE I. DE CIENCIA 2003**. Tecnología (Ricyt), Estado de la Ciencia: Principales indicadores de Ciencia y Tecnología Iberoamericanos/Interamericanos… Buenos Aires.

SALINAS R 2004. *Enfoque Multicriterio Multiexperto para la Toma de Decisiones.* La Habana: s.n.

SANCHEZ R 2002. *Indicadores de los sistemas de ciencia, tecnología e innovación. Economía industrial.* S.l.: s.n. 343.

SILVA DAVID 2009. *Arquitectura de software para un Sistema Automatizado para el Control de Gestión de Indicadores de Refinación.* La Habana: s.n.

ULLMAN LARRY 2007. Php 5 advanced: visual quickpro guide. S.l.

W3C 2014. World Wide Web. [En línea]. [Consulta: 4 diciembre 2014]. Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb.

CCHS. *MANUAL PARA LA SOLICITUD DE SALAS DE REUNIÓN DEL CCHS* SOCIALES, C. D. C. H. Y.

CRISTALDO, P. and M. KLOSTER Sistemas de Gestión.

EVOKO. *Evoko Room Manager Guía y manual del Administrador del sistema.*, 2010. 25.

FERNÁNDEZ, A. C. *Desarrollo de un simulador de eventos discretos en Python*, UNIVERSIDAD DE ALCALÁ ESCUELA POLITÉCNICA SUPERIOR 2016. 98. p.

GARZÓN, T. SISTEMAS GESTORES DE BASES DE DATOS *Innovación y Experiencia Educativa*, 2010, Vol.30.

KAPLAN-MOSS, A. H. Y. J. *El libro de Django* 28 de julio de 2008. 2008. 381 p.

LIBRES, G. D. U. D. T. *Ext JS … más que un simple Framework JavaScript*, Rel@x, 2013. [Disponible en: https://gutl.jovenclub.cu/ext-js-mas-que-un-simple-framework-javascript/

POTENCIER, F. and F. ZANINOTTO. *Symfony 1.2, la guía definitiva*. 2010. pp.10-31 p. ISSN: 978-1590597866

ROSSUM, G. V. *El tutorial de Python*. Septiembre 2009. 2009. p.

SAMPIERI, R.; C. F. COLLADO.*, et al.* Metodología de la investigación *México*, 2006, ISBN: 970-10-5753-8.

Bibliografía

BUSCHMANN F, MEUNIER, R., ROHNERT, H., SOMMERLAND, P. y STAL M 1996. Pattern – Oriented Software Architecture. A System of Patterns. Londres.

CARDOSO E, CAMACHO F y NUÑEZ G 2004. Arquitecturas de software. Guía de estudio.S.l.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009a. FolletoSistemaCI, Artículo 3.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009b. Folleto Sistema CI, Artículo 6.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009c. Folleto SistemaCI, Resolución No. 60/11. La Habana.

ECLIPSE FOUNDATION. **2011**. OpenUp Basic... S.l.

EGUILUZ JAVIER 2013. Introducción a AJAX.

EGUILUZ JAVIER 2010. Desarrollo web ágil con Symfony 2.3.

FLORES C y **ALFERÉZ**, **G. 2011**. Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navojoa Usando Open UP. México:

Framework-**Ecured**. [en línea] 2014. [Consulta: 24 febrero 2015]. Disponible en: http://www.ecured.cu/index.php/Framework.

GOBIERNO DE LA REPÚBLICA DE CUBA 2009. Bases para la constitución de la República. [en línea]. S.l.: [Consulta: 23 octubre 2014]. Disponible en: http://www.contraloria.cu/.

GOLDSTEIN A, **WEYL E y LAZIRIS L 2011**. HTML5 & CSS3 for the Real World... S.l.: SitePoint.

GONZALEZ ENRIQUE 2010. Aprender a Programar.

HERNÁNDEZ R A y COELLO S 2002. *El paradigma cuantitativo de la investigación científica.* La Habana: Editorial universitaria.

JACOBSON 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid: Pearson Educación S.A. ISBN 8478290362.

LARMAN, **C**. **2009**. *UML y patrones. Una introducción al análisis y diseño orientado a objetos.* 2. S.l.: s.n.

LOBOS MARIA 2014. Emagister. *www.emagister.com* [en línea]. [Consulta: 23 noviembre 2014]. Disponible en:http://www.emagister.com/curso-aprende-programar/concepto-lenguaje-programacion.

LOPERA C, **GUTIERREZ E y MARTÍN JC 2003**. *Indicadores: ciencia y tecnología en países de América Latina.* Lecturas de Economía. S.l.: s.n.

MARIN ALEJANDRA 1998. Nuevos conceptos del control interno. Valencia.

MAR OMAR 2014. Procedimiento para determinar el índice de control organizacional., vol. 18, no. 2.

MC GRAW, H. **2001**. *Ingeniería de software. Un enfoque práctico.* 5. España: s.n. ISBN 8448132149.

MINISTERIO DE FINANZAS 2003. Resolución del Ministerio de Finanzas y Precios. [en línea]. La Habana: [Consulta: 12 noviembre 2014]. Disponible en: http://www.contraloria.cu/.

NEGRÍN SIMEÓN 1997. La ciencia y la tecnología en Cuba, Revista Cubana de Medicina Tropical., vol. 49, no. 3, pp. 153-160.

R. I. **DE I. DE CIENCIA 2003**. Tecnología (Ricyt), Estado de la Ciencia: Principales indicadores de Ciencia y Tecnología Iberoamericanos/Interamericanos… Buenos Aires.

SALINAS R 2004. *Enfoque Multicriterio Multiexperto para la Toma de Decisiones.* La Habana: s.n.

SANCHEZ R 2002. *Indicadores de los sistemas de ciencia, tecnología e innovación. Economía industrial.* S.l.: s.n. 343.

SILVA DAVID 2009. *Arquitectura de software para un Sistema Automatizado para el Control de Gestión de Indicadores de Refinación.* La Habana: s.n.

ULLMAN LARRY 2007. Php 5 advanced: visual quickpro guide. S.l.

W3C 2014. World Wide Web. [En línea]. [Consulta: 4 diciembre 2014]. Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb.

CCHS. *MANUAL PARA LA SOLICITUD DE SALAS DE REUNIÓN DEL CCHS* SOCIALES, C. D. C. H. Y.

CRISTALDO, P. and M. KLOSTER Sistemas de Gestión.

EVOKO. *Evoko Room Manager Guía y manual del Administrador del sistema.*, 2010. 25.

FERNÁNDEZ, A. C. *Desarrollo de un simulador de eventos discretos en Python*, UNIVERSIDAD DE ALCALÁ ESCUELA POLITÉCNICA SUPERIOR 2016. 98. p.

GARZÓN, T. SISTEMAS GESTORES DE BASES DE DATOS *Innovación y Experiencia Educativa*, 2010, Vol.30.

KAPLAN-MOSS, A. H. Y. J. *El libro de Django* 28 de julio de 2008. 2008. 381 p.

LIBRES, G. D. U. D. T. *Ext JS … más que un simple Framework JavaScript*, Rel@x, 2013. [Disponible en: https://gutl.jovenclub.cu/ext-js-mas-que-un-simple-framework-javascript/

POTENCIER, F. and F. ZANINOTTO. *Symfony 1.2, la guía definitiva*. 2010. pp.10-31 p. ISSN: 978-1590597866

ROSSUM, G. V. *El tutorial de Python*. Septiembre 2009. 2009. p.

SAMPIERI, R.; C. F. COLLADO.*, et al.* Metodología de la investigación *México*, 2006, ISBN: 970-10-5753-8.

Anexos

Anexo 1: Entrevista realizada para identificar errores en el proceso de aplicación de reservaciones de las áreas compartidas en la Universidad de las Ciencias Informáticas.

Descripción: El autor realiza la entrevista en la Dirección de Servicios Generales (DSG) de la Universidad de las Ciencias Informáticas (UCI) específicamente al Msc. Omar Mar Cornelio, para detallar como se realiza el proceso de reservaciones de las áreas compartidas en la Universidad de las Ciencias Informáticas (UCI), de esta manera se extrae la información que se necesita a través del uso de las técnica de opinión y la propia experimentación de los implicados (entrevistador y entrevistados), la información de la entrevista está presente en el siguiente trabajo respetando las respuestas de los entrevistados.

Tabla 4: Encuesta realizada por el autor en la DSG.

Preguntas	Respuestas
¿Existe alguna aplicación para realizar las reservaciones?	No.
¿Se realizan de forma manual según las solicitudes que llegan por correo y por las actividades que salen por plan de trabajo?	Si.
¿Con qué frecuencia hay que estar revisando el correo para saber si llegan solicitudes de reservaciones?	Todo el día.
¿Quiénes pueden realizar solicitudes de reservaciones?	Todo el personal de la Universidad.
¿Quién es el/la encargado/a de planificar las reservaciones y todo lo que ellas impliquen y de enviar la información a los involucrados en las mismas?	El/La asistente de control de la Dirección de Servicios Generales o en su ausencia alguien dispuesto en el cargo.
¿Si se realiza algún cambio por cualquier	Si

circunstancia, se tendría que arreglar la información en los libros donde se anotan las reservas y volver a realizar todo el trabajo de informar a los involucrados?	
¿Se da seguimiento a las reservaciones?	El seguimiento que se le da a las reservaciones es pobre.
¿Qué tipo de seguimiento se le da a las reservaciones?	A través de correo electrónico, verbal y a veces alguna visita.
¿Cómo se identifican las reservaciones críticas para su cumplimiento?	Muy pocas veces se sabe cuáles son las reservaciones críticas.
¿Se da prioridad a las reservaciones de alguna forma?	No se establece ninguna prioridad específica.
¿Existe algún orden específico para garantizar las reservaciones?	No existe, se tratan de garantizar todas las reservaciones.
¿Cuál es la principal causa de lo planteado anteriormente?	Que no se han determinado nunca, solo se trata de cumplir con todas las reservaciones que se realicen.
¿Existen otras razones además de la antes planteada?	Si, muchas veces las reservaciones no se realizan con el tiempo de antelación necesario y sin solicitar todos los servicios que se requieren, además de que los cambios pueden ser inesperados.
¿Es confiable el resultado obtenido?	No.

¿Cuáles son los problemas más frecuentes en el proceso de reservación?	La presencia de errores humanos durante el procedimiento de reservación.
¿Es necesario digitalizar este proceso para un mejor funcionamiento de la gestión y control de las reservaciones de las áreas compartidas de la universidad?	Totalmente.