



Universidad de las Ciencias Informáticas

Facultad 4

Sistema informático para la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas

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

Autores:

Yuliet Castillo Reyes

Yoency Poll Guerrero

Tutores:

Ing. Yusdel Meriño Almaguer

Ing. Roanny Lamas López

Ciudad de La Habana, Junio 2015

“Año 57 de la Revolución”

Declaración de Autoría

Declaramos ser los autores del presente trabajo de diploma y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Yuliet Castillo Reyes

Yoency Poll Guerrero

Tutores:

Ing. Yusdel Meriño Almaguer

Ing. Roanny Lamas López

Dedicatoria

Dedico esta tesis a la memoria de mi abuela y a mi familia, pero en especial a mis padres que han sacrificado todo por mí.

Yuliet Castillo

Dedicado eternamente a mi pequeña de 5 añitos, a mi bella madre, a mi inigualable padre, en nombre de mi gran hermano. Va también para toda mi familia, y especialmente a los que tanto me quieren y han hecho sacrificios por mí.

Yoency Poll

De Yuliet:

Primero que todo quiero agradecer a mi abuela Nelsa que fue la que me inculcó valores desde chiquita para ser la persona que soy hoy en día. Nunca te olvidaré mami.

A mi familia tan grande y unida, por siempre preocuparse y confiar en mí en todo momento, pero en especial a mis padres y a Baby y Mandy mis segundos padres por cuidar de mí todo este tiempo.

A mi hermano, que a pesar de mortificarme todo el tiempo y no agradecerme en su tesis lo quiero mucho.

A mis tutores, y amigos ante todo Yusdel y Roanny por sus consejos y ayudarnos a que esto fuera posible.

A mi novio Sardina como yo le digo y a todos mis amigos por estar siempre a mi lado en los malos y buenos momentos en estos 5 años. Al igual que aquellos que no se encuentran presentes como Andrés y Jach por quererme como soy.

A mi compañero de tesis Poll, por soportarme en mis momentos de estrés.

Al tribunal y oponente por sus recomendaciones y los tiempos que nos dedicaron.

A los profes que me impartieron clases y los que de una forma u otra se preocuparon por mí.

De Yoency:

A Orlando Gabriel Toledano y Elías Bello por toda la ayuda brindada. A todos los que aportaron su conocimiento y contribuyeron con el desarrollo de esta tesis.

A todos muchas gracias.

Resumen

En la actualidad toda empresa u organización debe tener un control de los mantenimientos y de las reparaciones que se le realicen a los medios como estrategia para su desarrollo. El objetivo de este trabajo consiste en presentar los elementos más importantes de una propuesta de sistema informático para la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas, el cual tiene como propósito fundamental gestionar la información de los mantenimientos planificados y realizados así, como solicitar reportes de afectaciones de los medios; todo esto a través de una aplicación web.

Para dar cumplimiento al objetivo de la investigación se realizó un estudio de las soluciones similares existentes, las metodologías, herramientas y lenguajes a utilizar para la implementación de la propuesta de solución. El proceso de desarrollo estuvo guiado por la metodología XP, basándose en el framework Vaadin y lenguaje de programación Java. Se implementó en el Entorno de Desarrollo Integrado Eclipse, se utilizó el servidor web Apache Tomcat y como gestor de base de datos PostgreSQL. UML fue el lenguaje de modelado en la herramienta Visual Paradigm for UML. La investigación finaliza con las pruebas de aceptación del software, las cuales demostraron la correcta implementación de todas sus funcionalidades.

Palabras clave: mantenimiento, planificación, residencia estudiantil.

Índice

Introducción.....	1
Capítulo 1 Fundamentación Teórica.....	5
1.1 Conceptos generales relacionados con la investigación.....	5
1.1.1 Teoría del mantenimiento.....	6
1.2 Análisis de soluciones existentes.....	8
1.2.1 Soluciones similares existentes a nivel nacional.....	8
1.2.2 Soluciones similares existentes a nivel internacional.....	9
1.3 Estudio de las metodologías y estándares para el desarrollo de <i>software</i>	11
1.3.1 Programación Extrema.....	13
1.3.2 Scrum.....	14
1.3.3 Crystal Clear.....	14
1.4 Lenguaje Unificado de Modelado.....	15
1.5 Selección del <i>framework</i> de desarrollo en el servidor.....	16
1.6 <i>Framework</i> para la capa de acceso a datos.....	18
1.7 Selección del <i>framework</i> CSS.....	18
1.8 Lenguajes de programación.....	18
1.9 Entorno de desarrollo integrado.....	20
1.10 Herramienta CASE para el modelado con UML.....	21
1.11 Servidor de Bases de Datos.....	21
1.12 Servidor Web.....	22
1.13 Conclusiones del capítulo.....	23
Capítulo 2 Exploración y Planificación.....	24
2.1 Descripción de la propuesta de solución.....	24
2.2 Usuarios del sistema.....	25
2.3 Modelo conceptual del negocio.....	25
2.4 Requisitos del sistema.....	26

2.4.1 Requisitos funcionales	26
2.4.2 Aspectos no funcionales del sistema.....	29
2.5 Exploración.....	29
2.5.1 Historia de usuario.....	30
2.5.2 Historias de usuario de la propuesta de solución	30
2.6 Prototipo de interfaz de usuario.....	32
2.7 Planificación.....	34
2.7.1 Plan de Entregas	35
2.7.2 Iteraciones.....	35
2.8 Descripción de la arquitectura.....	36
2.9 Patrones de diseño	39
2.10 Tarjetas Clase-Responsabilidades-Colaboradores	42
2.11 Conclusiones del capítulo.....	44
Capítulo 3 Implementación y Pruebas.....	45
3.1 Diagrama Entidad-Relación.....	45
3.2 Estándares de codificación.....	47
3.3 Tareas de ingeniería.....	47
3.4 Pruebas.....	51
3.4.1 Desarrollo dirigido por pruebas.....	52
3.4.2 Pruebas unitarias	52
3.4.3 Pruebas de aceptación	53
3.5 Análisis de los resultados de las pruebas	58
3.6 Conclusiones del capítulo	59
Conclusiones generales.....	60
Recomendaciones.....	61
Referencias bibliográficas.....	62
Anexos.....	67
Anexo 1: Historias de usuario.....	67

Anexo 2: Tareas de ingeniería para la iteración 1.....	69
Anexo 3: Tareas de ingeniería para la iteración 2.....	72
Anexo 4: Tareas de ingeniería para la iteración 3.....	74
Anexo 5: Tareas de ingeniería para la iteración 4.....	76
Anexo 6: Pruebas unitarias	78
Anexo 7: Casos de pruebas de aceptación para la primera entrega	80
Anexo 8: Casos de pruebas de aceptación para la segunda entrega.....	88
Anexo 9: Casos de pruebas de aceptación para la tercera entrega	90

Introducción

Los avances tecnológicos de la actualidad evidencian un desarrollo acelerado de las Tecnologías de la Información y la Comunicaciones (TIC), las cuales están cada vez más ligadas a las distintas esferas de la vida social. Múltiples sectores se han visto favorecidos por el avance de las denominadas TIC, dentro de los que se destacan, la educación, la salud, el deporte, la producción y los servicios. Tales beneficios han sido posibles debido a las ventajas que dichas tecnologías ofrecen, como la facilidad y rapidez en acceso a la información, la gestión y procesamiento de datos, la comunicación y la posibilidad de poder compartir recursos.

Actualmente la gran mayoría de las empresas cubanas y demás organizaciones económicas, así como la población en general están concientizándose de la necesidad que existe de que se aplique una política de ahorro fundamentada en la planificación y el control del mantenimiento, no solo a los equipos electrodomésticos, sino también a los de computación, existentes en casi todas las entidades como consecuencia del proceso de informatización de la sociedad.

En la Universidad de las Ciencias Informáticas (UCI), se encuentra la residencia estudiantil, la cual es supervisada por una estructura denominada Dirección General de Residencia. En la UCI existen tres instancias de esta, las Direcciones de Residencias 1 y 2 donde radican los estudiantes y la Dirección de Residencia de profesores y especialistas.

En la Dirección de Residencia 2, existe un cúmulo de procesos que contribuyen a la gestión de esta área, uno de ellos es la planificación del mantenimiento a los equipos electrodomésticos, que persigue como uno de sus objetivos, garantizar mayor eficiencia de los servicios, aprovechamiento de los recursos y del tiempo.

Dentro de los distintos tipos de mantenimientos que existen, se encuentran el correctivo y preventivo. El correctivo consiste en el arreglo de las afectaciones de los equipos. El mantenimiento preventivo puede definirse como el conjunto de tareas sistemáticas de limpieza, ajuste y sustitución de piezas que se realiza antes de que se presente una falla en el equipo. (Cuartas, 2008) Aplicar un mantenimiento trae consigo que los equipos funcionen con mayor eficiencia, alarga la vida útil del mismo y reduce gastos en reparaciones fuera del presupuesto.

El proceso de planificación de los mantenimientos preventivos y correctivos en el área de Residencia 2 se ve afectado por diferentes factores y no se aprovechan al máximo las potencialidades de las TIC para lograr una gestión más eficiente de estos. Una vez dañado un equipo se reporta la afectación a la instructora educativa del edificio y esta tramita el reporte a la Dirección de Gestión Tecnológica por vía telefónica, lo que conlleva a que no quede evidencia en la residencia del reporte realizado. Tanto el proceso de reportes de afectaciones como la gestión de los medios electrodomésticos (aires acondicionados, minibar y televisores) del área se realizan de forma manual o mediante herramientas ofimáticas. Los documentos son archivados en copia dura para su posterior análisis, lo que en ocasiones provoca la pérdida de información. Otro factor que influye es que no existe evidencia digital de los mantenimientos realizados ni de los medios o componentes trasladados.

La planificación del mantenimiento se ve afectada porque no se tienen en cuenta determinados factores como son: el tiempo promedio que demora un técnico en realizar labores de mantenimiento a un equipo de un tipo específico, la jornada laboral, la cantidad de técnicos disponibles y la ubicación física de los equipos a revisar. La situación descrita anteriormente provoca incurrir en pérdida de tiempo, imprecisión de la planificación y deterioro de los equipos.

Por todo lo antes expuesto se plantea el siguiente **problema a resolver**: Inexistencia de un sistema automatizado que gestione la planificación del mantenimiento de los equipos electrodomésticos de acuerdo a las necesidades de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas.

De lo planteado anteriormente se define como **objeto de estudio** el proceso de gestión del mantenimiento.

A su vez el **campo de acción** lo comprende la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas.

El **objetivo general** de este trabajo es desarrollar un sistema informático para la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas.

Los **objetivos específicos** a cumplir son:

- Elaborar el marco teórico conceptual que sustenta la investigación.
- Identificar las funcionalidades con que debe cumplir la propuesta de solución.

- Diseñar e implementar el sistema informático para la gestión del mantenimiento.
- Realizar pruebas de *software* a la propuesta de solución.

Las **tareas a realizar** para dar cumplimiento a los objetivos específicos son:

- Revisión bibliográfica para conformar el estado del arte de la investigación.
- Realización de entrevistas sobre el tema de planificación y mantenimiento a técnicos y directivos de la Residencia Estudiantil 2.
- Búsqueda de información sobre soluciones similares existentes en diferentes contextos.
- Selección de la metodología para guiar el proceso de desarrollo del sistema.
- Selección de las herramientas y tecnologías a utilizar para la implementación del sistema.
- Definición de los requerimientos que debe cumplir la propuesta de solución a desarrollar.
- Implementación de la propuesta de solución.
- Realización de las pruebas de *software* a la propuesta de solución.

Los **métodos científicos** empleados en la investigación estuvieron determinados por el objetivo general y las tareas a realizar previstas. A nivel teórico fueron utilizados los métodos: **análisis-síntesis**, con el objetivo de tomar las mejores prácticas de sistemas similares sobre planificación y control que permitió la extracción de los elementos más importantes que se relacionan con el objeto de estudio y a partir del mismo, determinar las características que tendrá la propuesta de solución, así como las tecnologías y herramientas más adecuadas para el desarrollo de la solución. También fueron utilizados **métodos empíricos** como es el caso del **método experimental** para la realización de las pruebas al software una vez concluido el mismo.

Para una mejor comprensión de la investigación se decide realizar una **estructura capitular**. Los capítulos que la conforman son:

Capítulo1: Fundamentación Teórica

En el capítulo se desglosan una serie de conceptos claves asociados al dominio del problema y un profundo análisis sobre el mantenimiento, lo que contribuye a entender mejor todo lo referente al proceso de planificación y control del mantenimiento de los equipos electrodomésticos. Se investiga sobre soluciones similares existentes que planifican y controlan, en aras de obtener una base para la propuesta de solución.

Se realiza un estudio de las metodologías, herramientas y lenguajes de programación que existen en la actualidad, se profundiza en sus características y se seleccionan las más adecuada para dar cumplimiento al objetivo propuesto, desarrollar el sistema.

Capítulo 2: Exploración y Planificación

El capítulo está dedicado a las dos primeras fases de la metodología de desarrollo XP definidas para la investigación: exploración y planificación, donde se describe la estructura interna del sistema mediante los artefactos generados por la metodología de desarrollo utilizada. Se describen las historias de usuario que son las generadas por la metodología de desarrollo y se desglosan los requisitos funcionales y aspectos no funcionales del sistema para una mayor comprensión. Durante la planificación, el cliente establece la prioridad de cada historia de usuario y los programadores realizan una estimación del esfuerzo para implementarlas, se realiza el Plan de Iteraciones y el Plan de Entregas del producto. Se describe la arquitectura de la propuesta de solución, los patrones de diseño empleados y se muestran las tarjetas Clase-Responsabilidades-Colaboradores (tarjetas CRC).

Capítulo 3: Implementación y Prueba

En este capítulo se muestra el modelo de datos mediante un Diagrama Entidad-Relación para un mayor entendimiento de las tablas de la base de datos. Se redactan los estándares de codificación utilizados para implementar la propuesta de solución. Se muestran las tareas de ingeniería de las historias de usuario y las pruebas que propone la metodología empleada.

Capítulo 1 Fundamentación Teórica

En este capítulo se describe una serie de conceptos claves asociados al dominio del problema y se profundiza sobre el tema del mantenimiento. Se aborda un estudio sobre soluciones similares existentes que planifican y controlan los mantenimientos a los equipos, en aras de obtener una base para la propuesta de solución. Se detallan las metodologías de desarrollo, seleccionando la más adecuada, la denominada, Programación Extrema. Se realiza un estudio de las herramientas y lenguajes de programación existentes que serán utilizados durante el desarrollo del sistema

1.1 Conceptos generales relacionados con la investigación

Para contribuir a un mejor entendimiento de los conceptos que se utilizan en la investigación es necesario sentar las bases sobre los mismos, los cuales son mencionados a continuación.

Planificación: Es realizar un plan donde se fijan los recursos que serán necesarios para el logro de las metas trazadas. Es un intento de reducir la incertidumbre a través de una programación de las actividades, tomando en cuenta los escenarios más probables donde se desarrollarán. (Eumed, 2015) Se puede agregar además, que la planificación es establecer prioridades, decidir de forma anticipada cuando hacer alguna tarea y quién la realizará, teniendo en cuenta los factores internos y externos que puedan influir para que se cumpla el objetivo deseado.

Control: Se cerciora que el proceso o sistema que se ha planificado está cumpliendo y alcanzando o no, los resultados que se esperan. Según la Real Academia Española significa comprobación, inspección, fiscalización, intervención. (Real Academia Española, 2015)

Mantenimiento: El mantenimiento puede definirse como toda aquella actividad necesaria para mantener los equipos en buen estado, dígase ajustes, reemplazo, reparación y calibración. Según la Real Academia Española es un conjunto de operaciones y cuidados necesarios para que instalaciones, edificios e industrias puedan seguir funcionando adecuadamente. (Real Academia Española, 2015)

Mantenimiento Preventivo: *“El mantenimiento preventivo puede definirse como el conjunto de tareas sistemáticas de limpieza, lubricación, ajuste y sustitución de piezas antes de que se presente una falla en el equipo”. “Es la conservación planeada de la maquinaria, equipo e instalaciones teniendo como base el*

conocer sistemáticamente el estado de la misma para programar, en los momentos [sic] más oportunos y de menos impacto en la producción, las acciones necesarias para eliminar las posibles averías que originarían la interrupción del funcionamiento". (Ceballos, 2004, p.17)

Mantenimiento Correctivo: *"Se entiende por mantenimiento correctivo la corrección de las averías o fallas, cuando éstas [sic] se presentan. Es la habitual reparación tras una avería que obligó a detener la instalación o máquina afectada por el fallo". (Garrido, 2009, p.5)*

1.1.1 Teoría del mantenimiento

En sus inicios, el mantenimiento era aplicado contra averías o imprevistos que ocurrían en los equipos, pero la planificación del mantenimiento era nula, ya que realizaban las reparaciones luego de producirse la falla. A medida que fue evolucionando surge el mantenimiento correctivo, que se realiza de igual forma, pero en este caso se tienen todos los medios necesarios preparados para así disminuir el tiempo de parada del equipo. Este tipo de mantenimiento desprovisto de planificación no satisfizo las necesidades de la producción y se comenzó a trabajar conjugando el mantenimiento correctivo con el mantenimiento preventivo, conocido fundamentalmente en Cuba como Mantenimiento Preventivo Planificado (MPP), el cual se basa en la "planificación de trabajos y actividades" para lograr mantener el buen estado de los medios e intervenir con anterioridad a la falla. Dicha planificación se realiza teniendo en cuenta las experiencias del operario, los históricos de las máquinas y las especificaciones de los fabricantes. El MPP abarca la lubricación, la limpieza, la pintura y los reemplazos o modificaciones que garanticen la función del equipo (Rodríguez, 2003).

También surge el mantenimiento predictivo que consiste en planificar las inspecciones a los equipos. Estas inspecciones se realizan con la utilización de equipos de medición, teniendo como objetivo detectar los síntomas del fallo antes de que ocurra, para garantizar un reemplazo a tiempo. Debido al elevado costo de los equipos que se necesitan para las inspecciones y la alta calificación del personal para la manipulación de dicho equipamiento, no se mantuvo esa tendencia predictiva a pesar de las ventajas que representaba y se retomaron los mantenimientos correctivos y preventivos (Rodríguez, 2003). A continuación en la figura 1 se muestran las características de estos mantenimientos.

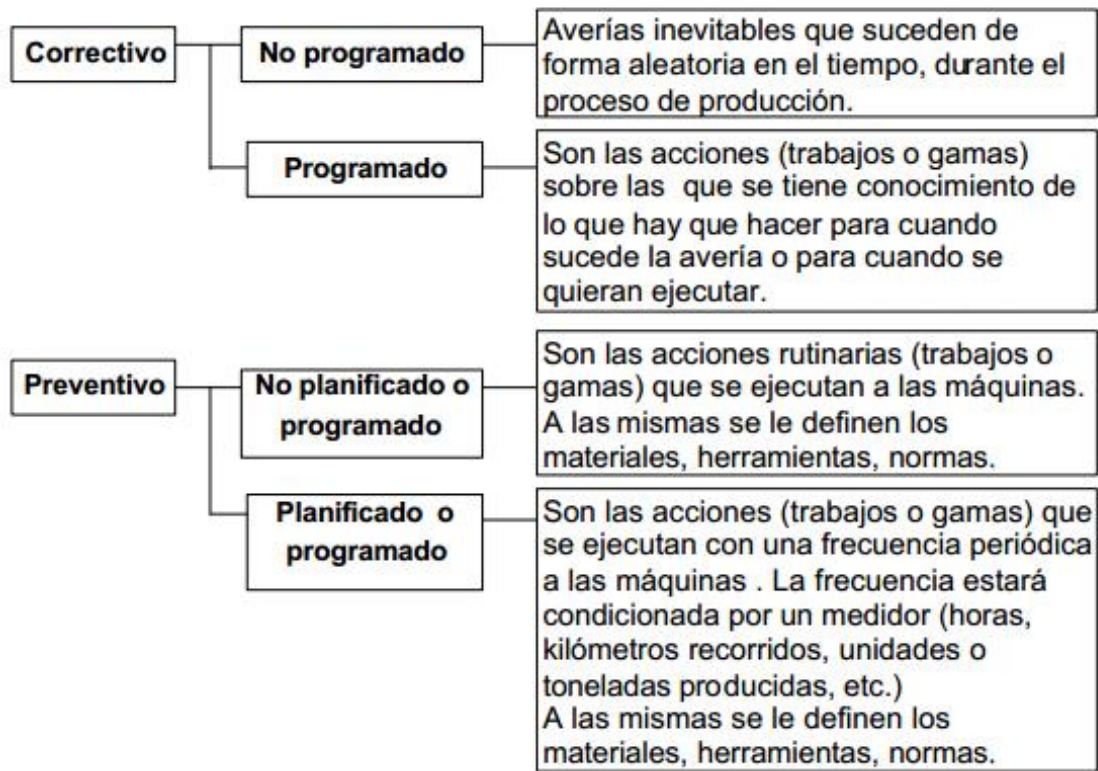


Figura 1: Características de los mantenimientos correctivos y preventivos (Rodríguez, 2003).

La planificación de estos mantenimientos lleva asociada una ardua tarea de organización, planificación y control, así como una elevada manipulación de información, por lo que se ha incurrido al uso de la computación, y se ha revolucionado la gestión del mantenimiento.

Con el desarrollo de la propuesta de solución se pretende que mejore el proceso de planificación del mantenimiento y alargar la vida útil de los equipos. Para planificar los mantenimientos preventivos a los medios electrodomésticos se debe tener un plan de trabajo bien estructurado, y tener presente lo que se quiere hacer, cómo se hará, cuándo y quiénes lo realizarán. Además es importante obtener determinada información de los medios, como el número de inventario, el modelo, las reparaciones que se le hayan efectuado anteriormente, su ubicación, el tipo de mantenimiento (preventivo o correctivo) y la frecuencia de mantenimiento.

En la actualidad cuando ocurre la afectación de un equipo, se le informa a la instructora del edificio y esta tramita el reporte a la Dirección de Gestión Tecnológica por vía telefónica, lo que conlleva a que no quede evidencia en la residencia del reporte realizado. Luego de esto se le orienta a un técnico que realice el mantenimiento correspondiente a la afectación del medio; sin embargo, no tienen en cuenta el orden en que fueron generadas las restantes solicitudes, lo que provoca que otros equipos más afectados no se arreglen y se deterioren.

Un buen control y planificación del mantenimiento trae consigo numerosos beneficios, como la reducción de paros imprevistos, incremento de la vida útil de los equipos, prevención de reparaciones costosas y documentación de la información.

1.2 Análisis de soluciones existentes

El desarrollo de un sistema de gestión que planifique y controle los mantenimientos, necesita de un análisis minucioso de otros sistemas que tengan resultados favorables. Es por ello que se decide realizar un estudio de varios sistemas en el ámbito nacional e internacional para determinar características y funcionalidades que puedan ser utilizadas en la propuesta de solución, a continuación se muestran dichas soluciones similares.

1.2.1 Soluciones similares existentes a nivel nacional

El **GMAC**¹ es un software destinado a la organización, planificación, ejecución y control de la gestión de la actividad de mantenimiento. La versión más reciente de este software se denomina **MacWin**², que es una herramienta informática concebida para ayudar en la resolución de problemas técnicos y de gestión de mantenimiento. MacWin está desarrollado en Access para Windows y basado en la teoría de bases de datos relacionales. La utilización del mismo en las empresas ha logrado reducir los índices de avería, así como un mayor aprovechamiento de la fuerza laboral dedicada al mantenimiento. También han notado crecimientos en la disponibilidad técnica del equipamiento, reducción de inventarios de piezas de repuesto y ahorros de papel debido a que más del 75% de todas las tareas de la gestión del mantenimiento se realizan

¹ Sistema de Gestión del Mantenimiento Asistido por Computadora.

² Mantenimiento Asistido por Computadora sobre Windows.

por computadora. El sistema cuenta con un grupo de módulos que permiten al usuario interactuar con él, a continuación se detallan los más importantes: (Rodríguez, 2003)

Módulo Patrimonio: Tiene la responsabilidad de organizar toda la información inicial que necesita la actividad de mantenimiento, se definen los equipos e instrumentos, su prioridad para el mantenimiento y sus características técnicas, así como el historial de las incidencias que le ocurren a los mismos.

Módulo Mantenimiento: Destinado a cumplir con el objetivo del seguimiento de las órdenes de trabajo correctivas, preventivas y predictivas. En la opción de preventivo permite la confección de la programación del mantenimiento, conformando de esta forma la estrategia a seguir por los departamentos, talleres y brigadas de mantenimiento.

A pesar de los resultados obtenidos en las empresas con el uso del MacWin se concluye que no puede ser utilizado para resolver el problema de la presente investigación, debido fundamentalmente a que solo puede ser utilizado en el sistema operativo Windows, lo que va en contra de las políticas de la universidad que abogan por la soberanía tecnológica y es un software creado con herramientas privativas.

1.2.2 Soluciones similares existentes a nivel internacional

GotelGest.Net: El *Software* de Servicio de Asistencia Técnica GotelGest.Net / SAT se utiliza para gestionar todo el sistema de servicio de asistencia técnica general prestada a clientes en distintos ámbitos como: mantenimientos *hardware* y *software*, mantenimientos a equipos eléctricos, calderas, aires acondicionados y reparaciones de electrodomésticos. En este sitio se facturan los avisos de asistencia emitidos por los clientes, entre los que se destacan: averías, reparaciones y mantenimientos. También se gestionan los distintos contratos de mantenimiento, entre los que se pueden citar: las revisiones periódicas y las garantías. Controlan la agenda de los técnicos en plantilla, así como las intervenciones realizadas y los tiempos empleados. (Servinet, 2015)

Algunas de las características del sistema son:

- Planificación y control de la agenda de los técnicos.
- Control de vacaciones, bajas y horario laboral de los técnicos.
- Distintos tipos de contrato de mantenimiento.
- Envío de alertas programadas.
- Gestión de contratos de mantenimiento.

- Gestión de técnicos de la empresa.

RENOVEFREE: El *software* de mantenimiento RENOVEFREE, es desarrollado por RENOVETEC. Puede implementar un árbol jerárquico de equipos y sistemas, elabora un plan de mantenimiento preventivo y obtiene los indicadores de mantenimiento más habituales. Los programas de Gestión de Mantenimiento llevan un control del mantenimiento programado de cualquier instalación, la carga de trabajo de cada uno de los técnicos, trabajo pendiente, gestión del repuesto, y gestión económica. (Renove Tecnología S.L, 2014)

SatNetwork: El Sistema de gestión de talleres SatNetwork SAT, archiva un historial de las reparaciones realizadas y lleva un control del inventario de repuestos. Brinda información a los clientes del estado de los equipos y vende productos. (Clotet, 2015) Algunas características del sistema SAT son:

- Control de inventario.
- Agendas de clientes y contactos.
- Notación de especificaciones técnicas en productos y servicios.
- Administración de perfiles de usuario.
- Exportación e importación de datos.
- Compatibilidad con Excel, Outlook y LibreOffice.
- Especificaciones técnicas de los equipos.

Los sistemas antes mencionados no pueden ser utilizados para dar solución al problema planteado debido fundamentalmente a que son herramientas privativas desarrolladas para promover servicios de mantenimiento. Su objetivo fundamental consiste en facilitar la comunicación entre clientes y empresas proveedoras de estos servicios.

El estudio y análisis realizado sobre algunos de los sistemas que son utilizados en la gestión del mantenimiento, tanto a nivel nacional como internacional, permitió detectar funcionalidades que pueden ser implementadas en la propuesta de solución. Ejemplo de estas funcionalidades son las siguientes:

- Control de inventario de los medios.
- Confección de la programación del mantenimiento preventivo.
- Exportación e importación de datos.
- Compatibilidad con archivos Excel.

- Administración de perfiles de usuario.
- Horario laboral de los técnicos.

1.3 Estudio de las metodologías y estándares para el desarrollo de *software*

El uso de metodologías para regir el proceso de desarrollo de *software* constituye una buena práctica de la Ingeniería de *Software*. Po tal motivo, se decide realizar un estudio de las principales metodologías existentes, dígase tradicionales o ágiles, con el objetivo de seleccionar la más adecuada para el desarrollo de la propuesta de solución.

Las metodologías tradicionales imponen una disciplina de trabajo en el proceso de desarrollo del *software* con el objetivo de conseguir un *software* más eficiente. Esta metodología no se adapta a los cambios, por lo que no se recomienda usarla cuando los requisitos pueden variar o no están bien definidos. Planifican detalladamente casi todo el desarrollo de *software* a largo plazo, lo que provoca, que si se produce algún cambio en esta planificación se viene abajo.

Sin embargo, las metodologías ágiles proponen procesos que se adaptan y progresan con el cambio, llegando incluso hasta el punto de cambiar ellos mismos. Están orientadas al personal más que orientadas al proceso y existe un ambiente de trabajo agradable. Son una buena elección cuando se trabaja con requisitos desconocidos o variables. Le dan mayor valor a la colaboración directa con el cliente y al desarrollo incremental del producto con iteraciones muy cortas y facilitan la generación rápida de prototipos y de versiones previas a la entrega final. (Cáceres y otros, 2001)

A continuación se muestra la tabla 1 con una comparación entre las metodologías antes mencionadas.

Tabla 1: Comparación entre las metodologías ágiles y tradicionales. (Letelier y Sánchez, 2003).

Metodologías Ágiles	Metodologías Tradicionales
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios
Proceso menos controlado, con pocos principios	Procesos mucho más controlado, con numerosas políticas/normas

El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del <i>software</i>	La arquitectura del <i>software</i> es esencial y se expresa mediante modelos

En resumen, las metodologías tradicionales están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto y obligan al cliente a tomar las decisiones al inicio del mismo, mientras que las metodologías ágiles se deberían aplicar en proyectos donde exista mucha incertidumbre, ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo, y en mantener una buena relación con el cliente. (Figuerola y otros,2015)

La propuesta de solución a desarrollar posee características que hacen que se ajuste a un proceso de desarrollo ágil debido a que:

1. El equipo de desarrollo es de 2 personas.
2. Se cuenta con un período corto de tiempo para el desarrollo de la aplicación.
3. Se añaden funcionalidades en cualquier momento, que tanto el desarrollador como el cliente entiendan necesarias, por lo que el cliente forma parte del equipo de desarrollo.
4. Los requisitos pueden sufrir cambios constantemente a lo largo del proceso de desarrollo.
5. A medida que el proyecto avanza pueden surgir nuevas ideas que serán incorporadas, lo que brinda mayor adaptabilidad al producto.

Debido a lo antes mencionado se requiere de un proceso de desarrollo preferiblemente ágil. Para decidir que metodología implementar se deben tener en cuenta las necesidades del negocio y el equipo. Se realizó un estudio de las principales metodologías ágiles, destacándose: Programación Extrema (XP), Scrum y Crystal Clear de las cuales se abordarán sus principales características.

1.3.1 Programación Extrema

En la metodología XP los equipos de desarrollo suelen trabajar en iteraciones de una o dos semanas de duración. Trabaja en un orden de prioridad estricta; el cliente prioriza las características que se desarrollarán y los desarrolladores trabajarán en ese orden. (Cohn, 2015)

Se consideran los cambios de requisitos en el transcurso del desarrollo del proyecto como un aspecto natural, inevitable e incluso deseable. Este método presenta características fundamentales que se ajustan a la presente investigación como son: (Figueroa y otros, 2015)

- Desarrollo iterativo e incremental que consiste en realizar pequeñas mejoras al *software*, una tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas.
- Programación por parejas.
- Frecuente interacción del equipo de programación con el cliente.
- Entregas frecuentes.
- Simplicidad en el código.

Además de estas características, XP cuenta con las historias de usuario que es la técnica que utilizan para especificar los requisitos del *software*. El trabajo con las mismas es dinámico y flexible, debido a que pueden romperse, añadirse nuevas o ser modificadas por otras más específicas o generales. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (Letelier y Penadés, 2006)

Los roles que propone XP de acuerdo con la propuesta original de Beck son:

- Programador.
- Cliente.
- Encargado de pruebas (*Tester*).
- Encargado de seguimiento (*Tracker*).
- Entrenador (*Coach*).
- Consultor.
- Gestor (*Big boss*).

El ciclo de vida ideal de XP consiste de seis fases:

- Exploración.
- Planificación de la Entrega.

- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

1.3.2 Scrum

Los equipos de Scrum suelen trabajar en iteraciones (llamados sprints) que son de dos semanas a un mes de duración. No permiten cambios en sus requisitos; una vez que se reúnen y planifican la entrega, ese conjunto de requerimientos a ser desarrollados durante cada iteración se mantiene sin cambios hasta el final. Esto permite al equipo trabajar concentrados ya que los mismos no van a cambiar. En esta metodología el cliente prioriza los requisitos, pero el equipo de desarrollo determina la secuencia en la que se desarrollarán. (Cohn, 2015)

En Scrum el desarrollo de *software* se realiza de forma iterativa e incremental, y se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente. Define cuales son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Al cliente ver crecer el producto iteración a iteración se compromete con el proyecto. Por otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta un incremento en la motivación de los integrantes del equipo. (Figuerola y otros, 2015)

1.3.3 Crystal Clear

Crystal Clear presenta algunas características similares con XP como: iteraciones cortas, entregas frecuentes y el contacto directo con usuarios. Pero también posee diferencias como: (Morales, 2007)

- Requiere que en las iteraciones las entregas no se extiendan más allá de 3 meses, y permite que la iteración pueda ser tan larga como el entregable.
- Consta de documentación.
- No existe la programación en pareja, es más flexible, lo que permite al equipo adecuar sus propias convenciones a cada proyecto.

Para comprender mejor se decidió comparar las metodologías Scrum y Crystal Clear con XP (tabla 2 y tabla 3) teniendo en cuenta las diferencias entre ambas. A continuación se muestran dichas tablas:

Tabla 2: Comparación entre las metodologías XP y Scrum.

Características	XP	Scrum
Tiempo de las iteraciones	De 1 o 2 semanas	De 2 semanas a 1 mes
Requisitos	Sensibles al cambio	No permiten cambios
Prioridad del Trabajo	El cliente prioriza y el equipo de desarrollo trabajará en ese orden.	El cliente prioriza y el equipo determina el orden en que se desarrollarán.

Tabla 3: Comparación entre las metodologías XP y Crystal Clear.

Características	XP	Crystal Clear
Tiempo de las iteraciones	De 1 o 2 semanas	No más de 3 meses
Documentación	No requiere casi	Sí
Programación en parejas	Sí	No

Al analizar las metodologías ágiles se decide escoger XP ya que sus características son las que más se ajustan al tema de investigación. Presenta un desarrollo basado en pruebas, la programación se trabaja en parejas y se caracteriza por la simplicidad de su código y diseño. Además, es empleada para proyectos de corto plazo, propone una retroalimentación continua entre el cliente y el equipo de desarrollo, lo que consigue una mayor integración entre los miembros mediante la comunicación.

1.4 Lenguaje Unificado de Modelado

Se decide escoger como lenguaje de modelado para representar gráficamente los distintos artefactos que se generan a través del ciclo de desarrollo de *software*, el Lenguaje Unificado de Modelado (UML³), el cual es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. (Rumbaugh y otros, 1999). Ofrece los elementos necesarios para modelar la solución propuesta y resolver el problema de la

³ De sus siglas en Inglés Unified Modeling Language.

presente investigación. Es aplicable a cualquier metodología de desarrollo de *software*, siempre y cuando esta se enmarque en el paradigma orientado a objetos.

1.5 Selección del *framework* de desarrollo en el servidor

Un *framework* son soluciones completas que contemplan herramientas de apoyo a la construcción de un ambiente de trabajo. Son diseñados con la intención de facilitar el desarrollo de *software*, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de *software* que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Dentro de Java en el ámbito específico de aplicaciones Web se destacan los *framework*: Struts, Spring, JSF⁴ y Vaadin que no son más que un conjunto de librerías API⁵ y herramientas para desarrollar aplicaciones Web. (Tangient, 2014)

Spring: Es una de las herramientas más usadas para crear grandes portales web con Java. Su código está ordenado debido a que presenta la arquitectura modelo vista controlador y anotaciones especiales que permiten separar el código en grupos. La programación de formularios es fácil porque Spring realiza todo el proceso de coger los datos de los formularios, comprobar que estén bien, recargar la página y mostrar los errores. Además es fácil definir etiquetas para ver si un campo está vacío o si un e-mail tiene un formato válido. (Borrego, 2015)

JSF: Es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE⁶. Está basado en el patrón MVC por lo que la separación de la lógica de la presentación le permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo. JSF es posterior a Struts, por lo que se encarga de mejorar deficiencias de este framework. Es sencillo y muy flexible ya que permite crear los propios componentes. JSF trata la vista similar al estilo de Visual Basic o Delphi, donde la programación de la interfaz se hace a través de componentes y basada en eventos. (Pérez, 2006)

Vaadin: Es un marco de desarrollo web Java que está diseñado para hacer fácil la creación y el mantenimiento de interfaces web de usuario de alta calidad. Este *framework* soporta dos modelos de

⁴ De sus siglas en Inglés JavaServer Faces.

⁵ De sus siglas en Inglés Application Program Interface

⁶ De sus siglas en Inglés Java Platform, Enterprise Edition.

programación diferentes: del lado del servidor (su lado más potente) y del lado del cliente. Vaadin es un *framework* RIA⁷, elemento que lo hace popular hoy en día. El presente marco de desarrollo define una separación entre la estructura de la interfaz de usuario y la apariencia, y permite desarrollarlos separadamente. (Grönroos, 2014) Según estudios realizados, se puede apreciar que los *framework* de Java más usados son: Spring, JSF y Vaadin como se muestra en la siguiente figura:

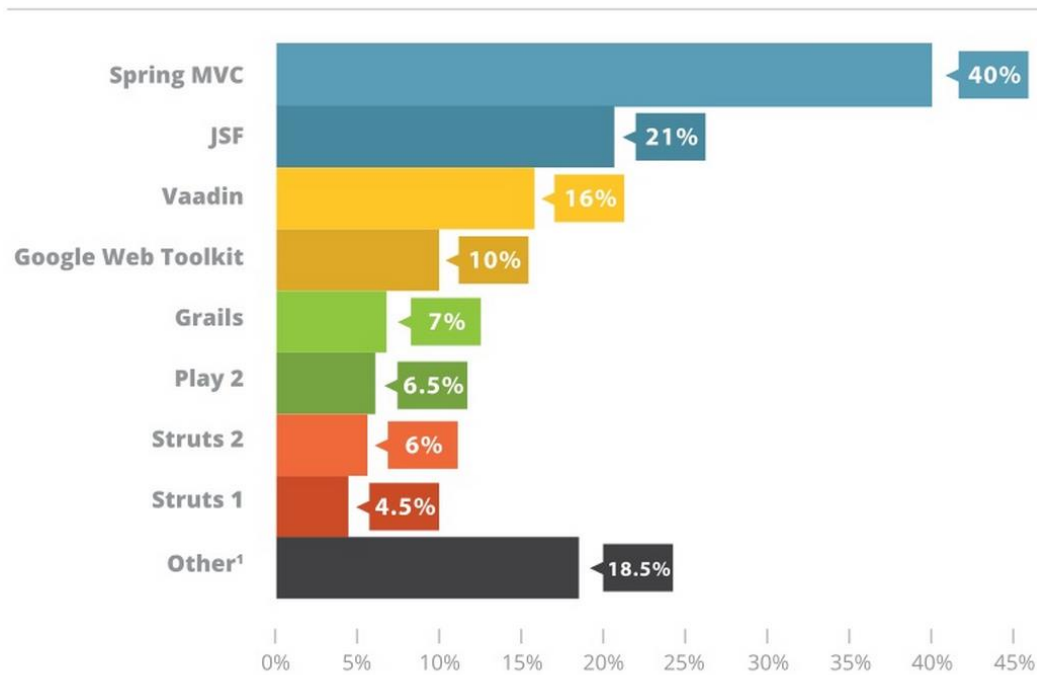


Figura 2: Frameworks Java más usados

A pesar de que Vaadin es utilizado un 16 % se decide escoger como *framework* de desarrollo en su versión 7.1.7 ya que el equipo de desarrollo posee dominio previo de dicho marco de trabajo. Además, cumple con las características necesarias para el desarrollo de la propuesta solución entre las cuales se destacan: que es un *framework* open source, utiliza java como único lenguaje de programación, es de agradable presentación al usuario y es soportado por la mayoría de navegadores web conocidos sin necesidad de plugins.

⁷ De sus siglas en español Aplicaciones de Internet Enriquecidas o en inglés Rich Internet Application.

1.6 Framework para la capa de acceso a datos

Java cuenta con varios marcos de trabajo para el acceso a los datos, los más destacados para este lenguaje son: *Hibernate* y JPA⁸. Estas tecnologías, nos permite realizar la conversión entre los objetos y las tablas de una base de datos. La conversión se llama ORM⁹, y puede configurarse a través de metadatos (mediante XML o anotaciones). JPA también permite seguir el sentido inverso, creando objetos a partir de las tablas de una base de datos, y también de forma transparente. (Marco, 2010)

Para el desarrollo de la propuesta solución será utilizado el lenguaje JPA en su versión 2.0 debido a la compatibilidad con otras herramientas. Esta API se utiliza como ORM y es construida para Java EE.

1.7 Selección del framework CSS

En la actualidad los desarrolladores para facilitar su trabajo en el diseño web utilizan marcos de trabajo. Esto contribuye a un considerable ahorro de tiempo, debido a que es una de las tareas que más tiempo requiere en el proceso de desarrollo de *software*. Dentro de los marcos de trabajo se puede mencionar a Twitter-Bootstrap, el cual es fácil de usar, y se poseen conocimientos previos.

Twitter-Bootstrap es un *framework* de código abierto cuyo objetivo es facilitar el desarrollo de aplicaciones o páginas web teniendo una colección de plantillas CSS, HTML y *plugins* JavaScript. Una vez creado un diseño web la interfaz es simple e intuitiva. Este *framework* con solo agregar algunas clases puede lograr un grupo de botones, formularios, menús desplegables y barras de navegación, y se ahorra tener que escribir código CSS. (Lerner, 2012)

La versión a utilizar será la 3.0.0. En la aplicación se utiliza principalmente para el estilo visual de la interfaz, esto posibilita la compatibilidad con distintos navegadores y dispositivos en los que pueda ser visualizado el contenido.

1.8 Lenguajes de programación

Java: Es un lenguaje de programación que permite crear cualquier tipo de programa y es uno de los más utilizados en la actualidad dadas sus múltiples ventajas. La principal característica de este lenguaje de programación es que es multiplataforma por lo que cualquier programa creado a través de Java podrá

⁸ De sus siglas en Inglés Java Persistence API.

⁹ En Inglés Object Relational Mapping. De sus siglas en español Mapeo Relacional de Objetos.

funcionar correctamente en ordenadores de todo tipo y con sistemas operativos distinto. Presenta una Programación Orientada a Objetos, paradigma muy utilizado hoy en día que facilita y organiza la programación. Además puede correr en el explorador y en dispositivos móviles y su sintaxis es similar a C++ pero más simple. (Adtriboo, 2014)

JavaScript: Es un lenguaje de programación interpretado, por lo que se pueden probar directamente en cualquier navegador sin necesidad de compilar los programas para ejecutarlos. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java, ya que es un lenguaje de programación ligero que añade funcionalidad en una página web. (Eguíluz, 2008) JavaScript es utilizado en el mundo del desarrollo web por ser muy versátil y potente, tanto por la realización de pequeñas tareas como para la gestión de complejas aplicaciones. No es necesario declarar las variables, se pueden utilizar aunque no hayan sido definidas anteriormente, esto se hace mediante la palabra reservada var. Algunas características de este lenguaje son: (Negrino y Smith, 2001)

- Máxima interactividad entre el usuario y la página.
- Verificación de los datos introducidos por el usuario, antes de enviar el formulario al servidor.
- Ejecución de pequeñas cantidades de información al igual que en una base de datos.
- Pre-procesado de información antes de enviarla al servidor.

Se selecciona este lenguaje por su sencillez y su compatibilidad con la mayoría de los navegadores. Además, tiene la ventaja que mediante él se puede realizar gran parte de las funciones del cliente, lo que reduce la carga del servidor.

HTML¹⁰: El Lenguaje de Etiquetado de Hipertexto es un estándar reconocido en todo el mundo. La última versión de este lenguaje es la conocida como HTML5, la cual ha sido diseñada para asegurar una compatibilidad hacia atrás con todo lo que se ha desarrollado hasta el presente, en términos de publicación sobre la Web.

HTML5 brinda numerosas novedades, a continuación se enumeran algunas de ellas: (Lancker, 2013)

- Supresión de etiquetas y atributos de presentación.
- Nuevas etiquetas semánticas o de organización.

¹⁰ Del inglés HyperText Markup Language.

- Nuevas etiquetas de audio y video.

La selección de HTML5 como el lenguaje para realizar el maquetado de la propuesta de solución está respaldada por las herramientas que incorpora para el desarrollo web.

CSS: Es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. Se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos. (Eguíluz, 2007) La última versión del lenguaje CSS es la versión 3 (CSS3), la cual ha incorporado valiosas características, algunas de ellas son: (Lancker, 2013)

- Nuevos selectores (entre los que se pueden citar los contextuales, de clase y por atributo).
- Bordes redondeados y decorados.
- Presentación de un texto en varias columnas.
- Sombra aplicadas al texto y al resto de elementos.
- Degradados de color.
- Los elementos gráficos como los filtros, las máscaras, el efecto espejo o a fusión de imágenes.

Para la selección de CSS3 como lenguaje de programación para escribir el estilo de la propuesta de solución, se tuvo en cuenta que es el usado para aplicar estilo en las aplicaciones y que posee un amplio uso y difusión por parte de diseñadores y desarrolladores web.

1.9 Entorno de desarrollo integrado

Se pueden desarrollar aplicaciones con el *framework* Vaadin en prácticamente cualquier entorno de desarrollo que tenga el SDK de Java y un contenedor de Servlets Java. Existe como soporte para este *framework* los IDEs NetBeans, IntelliJ IDEA y Eclipse, este último tiene un soporte especial para el *framework* Vaadin y es el más usado, por lo que se decide escoger como IDE de desarrollo.

Eclipse Kepler: Es una plataforma de desarrollo, de código abierto, bajo la licencia Eclipse Public License. Este IDE permite crear múltiples proyectos y a su vez proporciona asistentes y ayudas para la creación de los mismos. Posee una plataforma de herramientas web y se integra con Maven. Incluye un potente depurador que ayuda a mejorar el código, tiene una extensa colección de *plug-ins* disponibles y genera métodos y funcionalidades de manera automática, lo que hace el trabajo más fácil. (Rodríguez, 2015)

Para integrar el IDE Eclipse con el *framework* Vaadin se puede realizar mediante la instalación de un *plugin* o mediante Maven. Vaadin se distribuye en el repositorio central de Maven, y puede ser usado con cualquier sistema de gestión de dependencias que tenga acceso al repositorio de Maven, como Ivy o Graddle, además del propio Maven. (Grönroos, 2014)

Apache Maven 3.1.0. Es una herramienta que se puede utilizar para la construcción y gestión de cualquier proyecto basado en Java. Es una comunidad de código abierto y forma parte de la Fundación de Software Apache. Este repositorio local se encarga de gestionar librerías, complementos y drivers para proyectos de Vaadin 7. Maven está diseñado para ser flexible, es capaz de trabajar con varios proyectos al mismo tiempo y es extensible, con la capacidad para escribir plugins en Java o lenguajes de scripting. (Apache, 2015)

1.10 Herramienta CASE para el modelado con UML

Para el modelado del sistema, se decide utilizar **Visual Paradigm para UML** en su versión 8.0, por ser una herramienta que soporta el ciclo de vida completo en el desarrollo de *software*: análisis y desarrollos orientados a objetos, construcción, prueba y despliegue. Permite dibujar todo tipo de diagrama de clases, generación de código a partir de diagramas y generar documentación. Provee el modelado de procesos de negocios, además de un generador de mapeo de objetos-relacionales para los lenguajes de programación Java.NET y PHP. (Free Download Manager.org, 2015)

1.11 Servidor de Bases de Datos

PostgreSQL: Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹¹ y con su código fuente disponible libremente. Debido a sus características ha pasado a ser uno de los sistemas gestores de bases de datos más potentes, algunas de estas son: (Martinez, 2015)

- Posee buen rendimiento frente a grandes volúmenes de datos.
- Ofrece la posibilidad de definir nuevos tipos de datos.
- Varios usuarios pueden acceder al sistema a la vez.
- Es una base de datos 100% ACID¹².
- Regionalización por columna.
- Permite múltiples métodos de autenticación.

¹¹ Del inglés Berkeley Software Distribution.

¹² Atomicity, Consistency, Isolation and Durability.

- Permite el acceso encriptado vía SSL¹³.
- Disponible para Linux, UNIX y Windows, por lo que es multiplataforma.

En la tabla 4 se muestran algunos de los límites de almacenamiento que presenta PostgreSQL.

Tabla 4: Límites de almacenamiento de PostgreSQL.

Límite	Valor
Máximo tamaño base de datos	Ilimitado (Depende del sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (Dependiendo del tipo de dato)
Máximo número de índices por tabla	Ilimitado

Por lo antes expuesto se decide utilizar PostgreSQL en su versión 9.2 como gestor de base de datos. Además utiliza un modelo cliente/servidor y es multiprocesos, lo que permite garantizar la estabilidad del sistema. Un fallo en algunos de sus procesos no altera la funcionalidad de los otros procesos en ejecución y el sistema no se verá afectado.

1.12 Servidor Web

Un servidor web es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y unidireccionales entre el cliente que realiza peticiones y el servidor que genera las respuestas. La rapidez y capacidad de un servidor web serán los factores determinantes para el éxito

¹³ Del inglés Secure Sockets Layer.

de una página web. Para Java existen varios servidores, dentro de los que se pueden citar AOLServer y el Apache Tomcat.

AOLServer: El servidor web AOLServer es el servidor web de código libre desarrollado por AOL (América Online, el proveedor de Internet más importante del mundo). AOL utiliza AOLServer como servidor principal de web para uno de los entornos web de mayor tráfico y uso de Internet. AOLServer es un servidor web con muchas facilidades de uso en entornos de gran escala y sitios web dinámicos. Tiene una amplia base de usuarios, debido a que está integrado con OpenACS, un sistema de gestión de contenidos muy potente, de código libre, desarrollado inicialmente por una empresa llamada ArsDigita y posteriormente liberado bajo licencia GPL. (Mateu, 2004)

Apache Tomcat 7: Es una aplicación de software de código abierto de las tecnologías JavaServlet y JavaServer Pages. Apache Tomcat se desarrolla en un entorno abierto y participativo. Es liberado bajo la licencia Apache versión 2 y es multiplataforma. Tomcat permite el monitoreo y la administración remota de las aplicaciones mediante una conexión segura. (The Apache Software Foundation, 2015)

Se decide optar como servidor web **Apache Tomcat 7** porque está recomendado para Vaadin.

1.13 Conclusiones del capítulo

El estudio realizado apoyado en los métodos de investigación científica permitió elaborar el marco teórico conceptual que soporta la investigación, se definieron las características generales que se tendrán en cuenta para el desarrollo de la propuesta de solución, la cual se desarrollará utilizando la metodología XP para guiar el proceso de desarrollo; el lenguaje UML para el modelado de la solución mediante Visual Paradigm en su versión 8.0. Para la implementación del sistema se utilizará la herramienta Eclipse junto a las tecnologías del lado del cliente y del lado del servidor previamente definidas.

Capítulo 2 Exploración y Planificación

En el presente capítulo se exponen los procesos que serán automatizados en el sistema, basado en el estudio realizado anteriormente. Se hace referencia a lo relacionado con las fases de exploración y planificación, en las que se detallan las Historias de Usuario (HU) para establecer el orden en que serán implementadas, atendiendo a su prioridad. Se detallan los principales patrones de diseño empleados, se describe la arquitectura utilizada y se muestran las tarjetas CRC de la propuesta de solución.

2.1 Descripción de la propuesta de solución

El sistema informático para la gestión del mantenimiento de los equipos electrodomésticos se centra en la Residencia Estudiantil 2, de la Universidad de las Ciencias Informáticas, con el objetivo de que se optimice la planificación de los mantenimientos a los equipos electrodomésticos de esta área. Esto trae consigo el ahorro de recursos en la universidad y el incremento de la vida útil de los medios.

El acceso a las funcionalidades del sistema estará distribuido de acuerdo al rol que juegan los usuarios dentro del sistema como son: Administrador, Supervisor y Técnico. El sistema cuenta con un módulo de seguridad con estos usuarios pre establecidos en su configuración inicial. El módulo Configuración le permite al usuario con rol Administrador seguir una secuencia de pasos para que el sistema realice la planificación automática de los mantenimientos preventivos. Los pasos son: cargar un Excel que contiene todos los medios electrodomésticos de la Residencia 2 e introducir determinados datos como: cantidad de técnicos disponibles, horario laboral de los mismos y la frecuencia de mantenimiento de los equipos. Una vez concluido esta planificación se mostrará en la página inicial de todos los usuarios del sistema. Los usuarios con rol Administrador o Supervisor podrán confirmar la ejecución de los mantenimientos o postergar su planificación en caso necesario.

La propuesta de solución permite además, reportar solicitudes de mantenimientos correctivos; estas solo pueden ser generadas por los usuarios con roles Administrador o Supervisor y son atendidas por los Técnicos. Las solicitudes de reporte generadas se mostrarán en la funcionalidad “Solicitudes” del menú superior, donde el Técnico solo visualizará las solicitudes pendientes, ya que visualizar las solicitudes atendidas, es solo interés de los roles Administrador y Supervisor.

2.2 Usuarios del sistema

Los usuarios del sistema o usuarios clientes, son personas que interactúan con la aplicación haciendo uso de los servicios que proporciona. Estos usuarios según su rol presentan distintas responsabilidades en el sistema, a continuación se describen:

Tabla 5: Descripción de los usuarios del sistema.

Usuarios	Responsabilidades
Administrador	Tiene acceso a todos los módulos y funcionalidades del sistema.
Técnico	Solo tendrá acceso a las siguientes funcionalidades del sistema: Visualizar medio, Visualizar mantenimientos planificados, Autenticar usuario, Editar perfil y al Buscador.
Supervisor	Solo tendrá acceso a las siguientes funcionalidades del sistema: Visualizar medio, Autenticar usuario, Añadir solicitud, Solicitudes atendidas, Solicitudes pendientes, Visualizar mantenimientos planificados, Visualizar mantenimientos realizados, Confirmar mantenimiento realizado, Postergar mantenimiento, Recibir notificaciones de mantenimientos realizados, Recibir notificaciones de la planificación de los mantenimientos, Editar perfil, Buscador y Exportar a pdf.

2.3 Modelo conceptual del negocio

Para poder identificar los conceptos u objetos que estarán relacionados en la solución a desarrollar, se considera necesario generar un modelo conceptual del negocio, que no es más que un artefacto generado por los componentes del lenguaje de modelado UML. Este modelo permite una mayor comprensión entre el equipo de desarrollo y el cliente. A continuación se muestra dicho modelo.

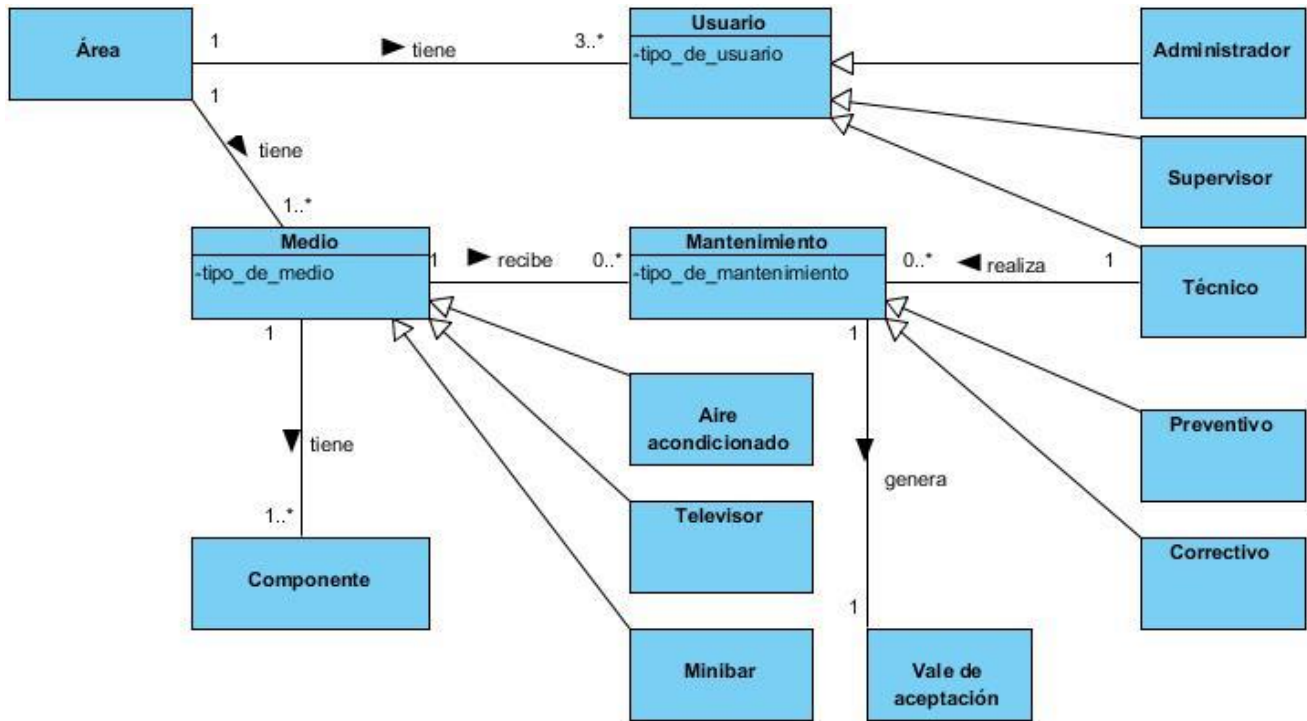


Figura 3: Modelo conceptual del negocio.

2.4 Requisitos del sistema

Los requisitos son una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo. (IEEE, 1990) Los requisitos se dividen en dos grupos: funcionales y no funcionales. En la metodología XP se definen como requisitos funcionales y aspectos no funcionales.

2.4.1 Requisitos funcionales

Los requisitos funcionales describen el comportamiento del sistema a partir de las acciones del usuario sobre el mismo. Las funcionalidades a implementar se muestran en la tabla presentada a continuación, que contiene los siguientes parámetros:

- **Requisito:** Números consecutivos a partir del 1 que identifica las funcionalidades.
- **Descripción:** Se desglosan las funcionalidades y los usuarios que pueden acceder a las mismas.

Tabla 6: Requisitos funcionalidades del sistema.

Requisito	Descripción
RF.1	Gestionar usuario. RF1.1 Crear usuario.(Administrador) RF1.2 Visualizar usuario. (Administrador) RF1.3 Modificar usuario. (Administrador) RF1.4 Eliminar usuario. (Administrador)
RF.2	Usuario perfil. RF2.1 Autenticar usuario. (Administrador, Supervisor y Técnico) RF2.2 Editar perfil. (Administrador, Supervisor y Técnico)
RF.3	Gestionar y trasladar medio. RF3.1 Adicionar medio. (Administrador) RF3.2 Visualizar medio. (Administrador, Supervisor y Técnico) RF3.3 Modificar medio. (Administrador) RF3.4 Eliminar medio. (Administrador) RF3.5 Realizar traslado del medio. (Administrador)
RF.4	Componente RF4.1 Adicionar componente. (Administrador) RF4.2 Realizar traslado del componente. (Administrador)
RF.5	Solicitudes RF5.1 Añadir solicitud. (Administrador y Supervisor)

Capítulo 2: Exploración y Planificación

	RF5.2 Solicitudes atendidas. (Administrador y Supervisor) RF5.3 Solicitudes pendientes. (Administrador y Supervisor)
RF.6	Planificar mantenimientos. RF6.1 Configurar planificación. (Administrador) RF6.2 Períodos de mantenimientos. (Administrador) RF6.3 Cargar Excel. (Administrador)
RF.7	Mantenimientos RF7.1 Visualizar mantenimientos planificados(Administrador, Supervisor y Técnico) RF7.2 Visualizar mantenimientos realizados(Administrador, Supervisor) RF7.3 Confirmar mantenimiento realizado. (Administrador, Supervisor) RF7.4 Postergar planificación del mantenimiento. (Administrador, Supervisor) RF7.5 Recibir notificaciones de mantenimientos realizados. (Administrador y Supervisor) RF7.6 Recibir notificaciones de la planificación de los mantenimientos. (Administrador y Supervisor)
RF.8	Buscar y exportar. RF8.1 Buscador. (Administrador, Supervisor y Técnico) RF8.2 Exportar a PDF. (Administrador y Supervisor)

2.4.2 Aspectos no funcionales del sistema

Los aspectos no funcionales del sistema son propiedades o cualidades que el producto debe tener. Son características para hacer que el software sea más agradable, usable, rápido y seguro. Los aspectos no funcionales asociados al sistema son los siguientes:

Apariencia o interfaz externa

- Debe notificar al usuario ante la presencia de errores de validación.
- Los colores de la aplicación serán claros y la letra estará en tamaño legible.

Portabilidad

- La aplicación será multiplataforma.
- Se podrá acceder a la aplicación desde cualquier estación de trabajo.

Seguridad

- El acceso a la información deberá estar restringido por usuario y contraseña.
- Según el usuario autenticado, se mostrará la información que le corresponde.

Software

- Se debe contar con la máquina virtual de java en su versión 7 o superior, como servidor web Apache Tomcat 7.

2.5 Exploración

En la fase de exploración el cliente plantea a grandes rasgos las historias de usuario que les son de interés para la primera entrega del producto, mientras que los programadores se familiarizan con las herramientas, tecnologías y prácticas que utilizarán. Esta fase tiene un tiempo de duración de pocas semanas a pocos meses, en dependencia de la magnitud y la familiaridad de los programadores con la tecnología. (Beck, 1999)

Al tener redactadas las historias de usuario los programadores hacen una estimación de los tiempos de desarrollo. Las estimaciones realizadas en esta fase son primarias ya que estarán basadas en datos de muy alto nivel, y podrían variar cuando se analicen más en detalle en cada iteración. (Joscowicz, 2008)

2.5.1 Historia de usuario

Las historias de usuario no son más que tarjetas de papel en las cuales el cliente describe las características del sistema. El contenido que ellas abarcan debe ser concreto y sencillo. Se redactan desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. En cualquier momento del desarrollo las historias de usuario pueden ser modificadas o reemplazadas por otras más generales o específicas, así como añadir nuevas. Cada una es lo suficientemente entendible y está delimitada como para que los desarrolladores puedan implementarlas en unas semanas (Jeffries, 2001)

2.5.2 Historias de usuario de la propuesta de solución

La metodología XP no define una plantilla específica para representar las HU. En el área de calidad de la UCI se definió una plantilla que contendrá, el nombre del requisito, el número, el programador, la iteración en la que será implementada, el nivel de prioridad en el negocio (Alta, Media, Baja), un tiempo estimado de desarrollo en días, el riesgo, una breve descripción, un apartado para agregar alguna observación si es necesario y por último un prototipo de interfaz. A continuación se mostrarán las HU 3,5, 6 y 8 que son consideradas las más importantes debido a que gestionan los datos de los medios y su planificación para el mantenimiento. Las restantes 4 se encuentran en el Anexo 1 de la investigación.

Tabla 7: HU Gestionar y trasladar medio

Historia de usuario	
Número: 3	Nombre del requisito: Gestionar y trasladar medio
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alto	
Descripción: El sistema permitirá adicionar, modificar, visualizar, eliminar y trasladar un medio. Las opciones de adicionar, modificar, visualizar, eliminar y trasladar un medio solo las puede realizar el usuario con rol Administrador. Además, a la opción visualizar medio tienen acceso el Supervisor y el Técnico. Para realizar una de las operaciones mencionadas, se debe dirigir al menú superior y seleccionar la opción Medios/Adicionar medio. El Administrador podrá cambiar la ubicación de un medio y una vez trasladado podrá visualizar los	

resultados en Medios/Medios trasladados, donde se generará una traza de todos los traslados de dicho medio.

Observaciones: El usuario deberá estar autenticado. El sistema le indicará los campos que son obligatorios en caso de que la opción seleccionada así lo requiera.

Tabla 8: HU Solicitudes

Historia de usuario	
Número: 5	Nombre del requisito: Solicitudes
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alto	
<p>Descripción: El sistema permitirá al usuario con rol Administrador y Supervisor solicitar un reporte de un medio para su mantenimiento correctivo, así como visualizar las solicitudes de reportes de los medios que se han atendido hasta el momento y las que están pendientes. Para realizar las operaciones descritas, debe dirigirse a la funcionalidad Solicitud del menú superior.</p>	
<p>Observaciones: El usuario deberá estar autenticado con el rol Administrador o Supervisor y se le mostrarán los campos obligatorios para poder generar una solicitud de reporte de un medio. Para visualizar las solicitudes pendientes y atendidas deben existir reportes en el sistema.</p>	

Tabla 9: HU Planificar mantenimientos.

Historia de usuario	
Número: 6	Nombre del requisito: Planificar mantenimientos
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Alto	

Descripción: El sistema permitirá planificar los mantenimientos preventivos que le corresponden a los medios. Para esto el usuario tiene que estar autenticado con el rol Administrador. Cuando este usuario entre al sistema por primera vez se le mostrará en la página principal una guía de pasos que deberá seguir para que el sistema pueda planificar los mantenimientos. Los pasos serán cargar el Excel que contiene los medios de la Residencia 2, introducir los períodos de mantenimientos de cada tipo de medio, la cantidad de técnicos, horas laborales, y los tiempos promedios que se demora un técnico en arreglar un tipo de medio. Finalizado los pasos, el sistema mostrará la vista principal con la información de los mantenimientos planificados.

Observaciones: El usuario con rol Administrador deberá estar autenticado y el sistema le indicará los campos que son obligatorios para poder realizar la planificación de los mantenimientos.

Tabla 10: HU Buscar y exportar.

Historia de usuario	
Número: 8	Nombre del requisito: Buscar y exportar
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 4
Prioridad: Baja	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Bajo	
Descripción: El sistema permitirá a los usuarios con rol Administrador, Técnico y Supervisor buscar un medio por el criterio de búsqueda: número de inventario. El sistema también permitirá al usuario con rol Administrador y Supervisor exportar a documentos en formato PDF la cantidad de acciones de mantenimiento que se han realizado, la cantidad de mantenimientos que quedan pendientes, así como el tipo de mantenimiento realizado: correctivo o preventivo. Además debe mostrar la descripción del medio al que se le realizó dicho mantenimiento.	
Observaciones: El usuario deberá estar autenticado. Solo se podrá buscar los datos de los medios y exportar los datos de los medios y mantenimientos.	

2.6 Prototipo de interfaz de usuario

Realizar los prototipos de interfaz que tendrá la aplicación es una buena práctica para tener una primera aproximación de lo que se pretende implementar, para más adelante, corregir los errores o detalles que no se tuvieron en cuenta inicialmente. A continuación se muestran algunos prototipos de interfaz.

Por favor ingrese sus datos



Usuario

Contraseña

Aceptar

Figura 4: Prototipo de interfaz de la pantalla de autenticación.

Sistema de Mantenimiento

Usuarios Medios Componentes Solicitudes Mantenimientos Buscar

Principal
Editar perfil
Cargar Excel
Configurar planificación
Períodos de mantenimientos
Planificar mantenimientos
Salir

Mantenimientos Planificados

Identificación: 07496
Tipo: Televisor
Ubicación: 136-105
Fecha: 20-5-2015

Identificación: 01165
Tipo: Minibar
Ubicación: 136-105

Figura 5: Prototipo de interfaz de la vista del administrador

El prototipo de interfaz de usuario muestra un sistema de mantenimiento con un menú de navegación superior que incluye 'Medios', 'Solicitudes' y 'Mantenimientos'. A la derecha del menú hay un campo de búsqueda con el botón 'Buscar'. El contenido principal está dividido en dos secciones: a la izquierda, un menú con 'Principal', 'Editar perfil' y 'Salir'; a la derecha, un panel titulado 'Mantenimientos Planificados' que muestra dos tarjetas de mantenimiento. La primera tarjeta contiene los campos: 'Identificación: 07496', 'Tipo: Televisor', 'Ubicación: 136-105' y 'Fecha: 20-5-2015', con botones 'Postergar' y 'Confirmar'. La segunda tarjeta contiene: 'Identificación: 01165', 'Tipo: Minibar' y 'Ubicación: 136-105'.

Figura 6: Prototipo de interfaz de la vista del técnico y supervisor.

2.7 Planificación

La metodología XP plantea la planificación como una fase corta y de un diálogo continuo entre el cliente y el grupo de desarrolladores. En este diálogo se acuerda el orden en que deberán implementarse las HU, y asociadas a estas, las entregas.

Una vez obtenidas las historias de usuario, los programadores estiman el tiempo necesario para implementar cada una, el cual debe ser de entre una y tres semanas. En caso de extenderse más de tres semanas la misma debe ser modificada o dividida en otras más pequeñas. Una duración de menos de una semana indica que es demasiado sencilla y deberá ser agrupada. (Joskowicz, 2008)

Como resultado de la fase de planificación fueron generados el Plan de Entregas y el Plan de Iteraciones los cuales se muestran a continuación.

2.7.1 Plan de Entregas

El plan de entregas es un documento elaborado por el cliente donde quedan plasmadas las entregas de las versiones de la propuesta de solución; así el equipo de desarrollo se planifica para saber que funcionalidades entregará en cada entrega. Para esto se definió el siguiente Plan de Entregas:

Tabla 11: Plan de Entregas.

Entregable	1ra entrega (3ra semana de Marzo)	2da entrega (3ra semana de Abril)	3ra entrega (3ra semana de Mayo)
Sistema informático para la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la UCI	Versión 0.1	Versión 0.2	Versión 1.0

Tabla 12: Funcionalidades disponibles por entrega del producto.

Historia de Usuario	1ra entrega	2da entrega	3ra entrega
HU1. Gestionar usuario.	X	entregado	entregado
HU2. Usuario perfil	X	entregado	entregado
HU3. Gestionar y trasladar medio.	X	entregado	entregado
HU4. Componente.	X	entregado	entregado
HU5. Solicitudes		X	entregado
HU6. Planificar mantenimientos.		X	entregado
HU7. Mantenimientos			X
HU8. Buscar y exportar			X

2.7.2 Iteraciones

La metodología XP es un proceso de desarrollo iterativo, por tanto incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega ideal está compuesto por iteraciones de no más de tres semanas,

en cada iteración se define un conjunto de HU que se van a implementar (González, 2004). Teniendo en cuenta la cantidad de funcionalidades con las que cuentan las iteraciones 3 y 4, se definió un tiempo de desarrollo de 4 semanas para las mismas.

En XP una iteración cumple con los estándares de tiempo de desarrollo, tiempo de prueba y el tiempo de refactorización y no necesariamente termina con una entrega del producto, por lo que una versión puede implicar varias iteraciones del proceso de desarrollo.

Tabla 13: Planificación de las iteraciones.

Iteraciones	Orden de las HU a implementar	Cantidad de tiempo de trabajo
Iteración 1	HU1. Gestionar usuario HU2. Usuario perfil	3 semanas
Iteración 2	HU3. Gestionar y trasladar medio HU4. Componente	3 semanas
Iteración 3	HU5. Solicitudes HU6. Planificar mantenimiento	4 semanas
Iteración 4	HU7. Mantenimientos HU8. Buscar y exportar	4 semanas

2.8 Descripción de la arquitectura

Los patrones arquitectónicos representan el nivel más alto dentro del sistema de patrones y expresan el esquema de la estructura fundamental de la organización para sistemas de software. (Almeira y Perez 2007)

El patrón Modelo Vista Controlador (MVC) es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario. Es el más utilizado en aplicaciones Web y permite no mezclar lenguajes de programación en el mismo código. (Bahit, 2011) MVC divide las aplicaciones en tres niveles de abstracción, las cuales se describen a continuación:

Capítulo 2: Exploración y Planificación

En la capa del **Modelo** es donde se representa la lógica del negocio y es la encargada de acceder de forma directa a los datos actuando como intermediario con la base de datos. (Bahit, 2011) En esta capa actúa el ORM JPA realizando la conversión entre los objetos y las tablas de la base de datos. JPA tiene incluido el lenguaje de consulta orientado a objetos JPQL (Java Persistence Query Language, Lenguaje de Consulta de Persistencia en Java) que nos permite obtener más de un objeto por consulta. (Marco, 2010)

La capa de la **Vista** o de presentación es la encargada de mostrar la información al usuario de forma gráfica y legible. (Bahit, 2011) El framework Vaadin estará presente en esta capa ya que está diseñado para trabajar en la vista.

La capa del **Controlador** es la intermediaria entre la vista y el modelo. Controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que esta lo presente al usuario, de una manera legible. (Bahit, 2011) En esta capa estarán las clases Java de la aplicación.

A continuación se muestra la figura 6 que describe el funcionamiento de la arquitectura de la propuesta de solución y la figura 7 que representa como se evidencia en las clases de la aplicación.

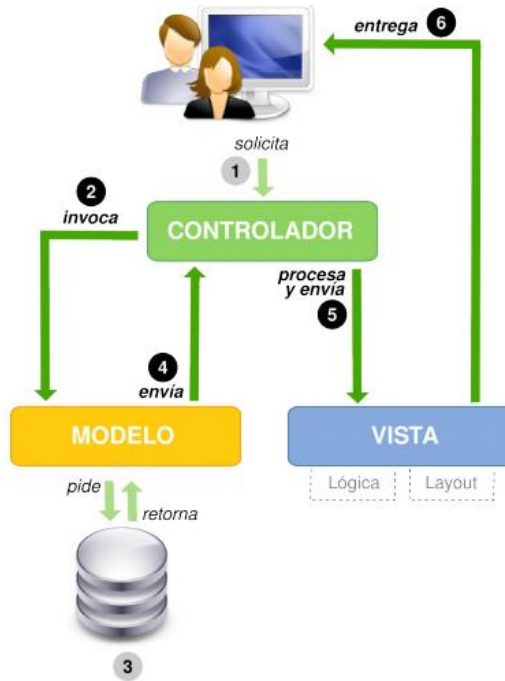


Figura 7 Funcionamiento de MVC. (Bahit, 2011)

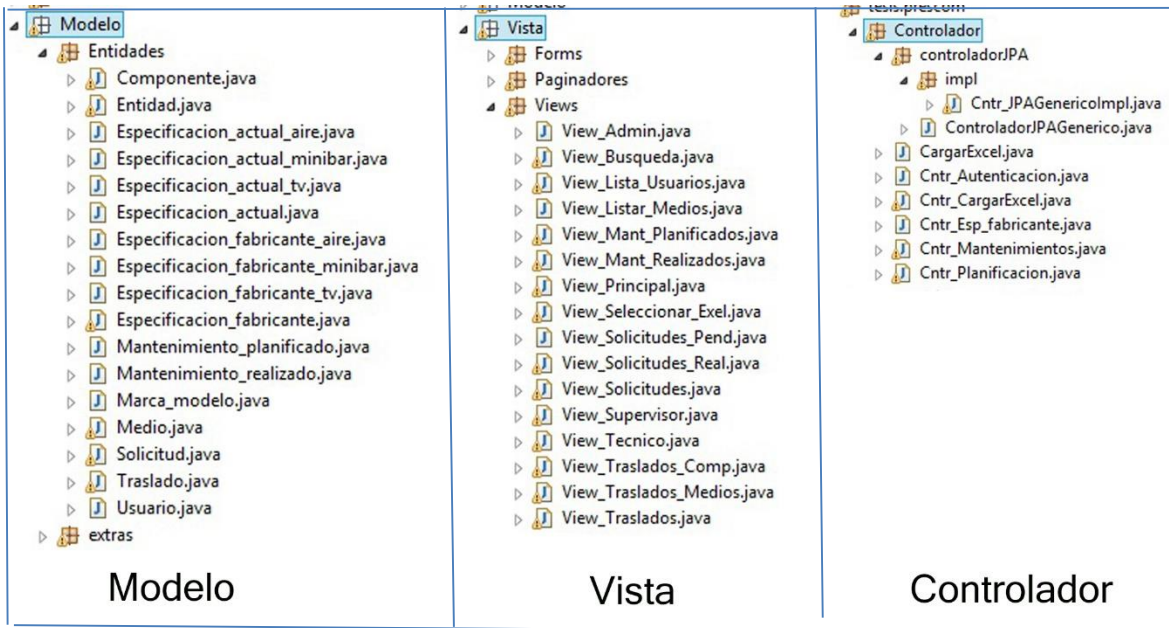


Figura 8: Representación del patrón arquitectónico MVC en la propuesta de solución.

2.9 Patrones de diseño

Para el desarrollo de la propuesta de solución se usaron patrones de diseño para la asignación de responsabilidades, por ser considerados buenas prácticas en el proceso de desarrollo de un software. Los patrones utilizados son los GRASP¹⁴ y los GoF¹⁵. Dentro del patrón GRASP los utilizados son: Experto, Creador y Controlador. Del patrón GoF el empleado es el Decorator.

Experto

Este patrón es utilizado para resolver el problema de *“asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.”* (Larman, 2003, p.193). Es utilizado para identificar la información que tiene cada clase y así otras puedan manipularlas. Este patrón se evidencia en las clases: Componente, Medio, Mantenimiento_planificado, Mantenimiento_realizado, Solicitud, Traslado y Usuario. A continuación se muestra la figura 8 con un ejemplo de la clase Medio.

¹⁴ De sus siglas en español Responsabilidad General Patrones de Software Asignación y de sus siglas en inglés General Responsibility Assignment Software Patterns.

¹⁵ De sus siglas en inglés Gang of Four.

```
Medio.java
76
77
78 public String getEstado_medio() {
79     return estado_medio;
80 }
81
82 public void setEstado_medio(String estado_medio) {
83     this.estado_medio = estado_medio;
84 }
85
86 public String getNo_inventario() {
87     return no_inventario;
88 }
89
90 public void setNo_inventario(String no_inventario) {
91     this.no_inventario = no_inventario;
92 }
93
94 public String getUbicacion_medio() {
95     return ubicacion_medio;
96 }
97
98 public void setUbicacion_medio(String ubicacion_medio) {
99     this.ubicacion_medio = ubicacion_medio;
100 }
101
102 public String getObservaciones() {
103     return observaciones;
104 }
105
106 public void setObservaciones(String observaciones) {
107     this.observaciones = observaciones;
108 }
---
```

Figura 9: Ejemplo del patrón experto en una clase.

Creador

El patrón Creador se encarga de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Tiene como beneficio que soporta el bajo acoplamiento, lo que implica menos dependencias de mantenimiento y mayores oportunidades para reutilizar (Larman 2003). A continuación se muestra la figura 9 con un ejemplo de la clase Medio a la cual se le asigna la responsabilidad de crear instancias de los mantenimientos planificados, mantenimientos realizados, traslado y componente.


```
*Medio.java
26
27 @OneToOne(cascade=CascadeType.ALL,orphanRemoval=true)
28 private Mantenimiento_planificado mant_p;
29
30 @OneToMany(cascade=CascadeType.ALL,orphanRemoval=true)
31 private List<Componente> componentes;
32
33 @OneToMany()
34 private List<Mantenimiento_realizado> mant_r;
35
36 @OneToMany(cascade=CascadeType.ALL,orphanRemoval=true)
37 private List<Traslado> movimientos;
38
39 public Medio() {
40
41     tipo="";
42     estado_medio="";
43     ubicacion_medio="";
44     no_inventario="";
45     observaciones="";
46     tiempo_demora=0;
47     periodo_mant=0;
48     movimientos=new LinkedList<Traslado>();
49     mant_p=new Mantenimiento_planificado();
50     mant_r=new LinkedList<Mantenimiento_realizado>();
51     componentes=new LinkedList<Componente>();
52
53 }
```

Figura 10: Ejemplo del patrón creador en una clase.

Controlador

Se encarga de “asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente”. (Larman, 2003, p.206) Este patrón se evidencia en las clases controladoras de la aplicación como son: Cntr_JPAGenericoImpl, Cntr_Mantenimientos y Cntr_CargarExcel.

Otros patrones de diseño de gran utilidad durante la fase de diseño orientado a objetos son los GoF, conocidos como patrones de la Pandilla de los Cuatro, dado que el libro que los explica fue escrito por cuatro autores. Dentro de este fue utilizado el patrón Decorator, explicado a continuación.

Decorator: Este patrón se encarga de crear una plantilla de diseño base que incorpora todos los componentes (cabecera, pie de página, menú, secciones entre otros) comunes del sistema y luego puede ser llamada para su uso en otras plantillas. Para el caso específico de la investigación la plantilla decorada es la clase plantilla.html (Design, 2003)

2.10 Tarjetas Clase-Responsabilidades-Colaboradores

Las tarjetas Clase-Responsabilidades-Colaboradores (CRC) son una técnica para el diseño de *software* orientado a objetos que permiten identificar y entender los requerimientos del negocio. Son divididas en tres secciones, como se muestra a continuación (Beck y Cunningham, 1989)

Tabla 14: Tarjeta CRC.

Nombre de la clase	
Responsabilidades	Colaboradores

Las tarjetas están compuestas por el nombre de la clase, las responsabilidades y los colaboradores. Una responsabilidad no es más, que los métodos que contendrá esa clase. Cuando una clase tenga que cumplir una responsabilidad y no tenga suficiente información para hacerlo tiene que colaborar con otras clases; aquí es donde los colaboradores ayudarán a ejecutar responsabilidad ya que son clases que el método llama. A continuación se muestran las tarjetas CRC de las clases controladoras de la aplicación:

Tabla 15: Tarjeta CRC #1.

Nombre: Cntr_Mantenimientos.	
Responsabilidad	Colaboraciones
Planificar()	Mantenimiento_planificado ControladorJPAGenericoImpl Medio

Tabla 16: Tarjeta CRC #2.

Nombre: Cntr_JPAGenericoImpl	
Responsabilidad	Colaboraciones
Salvar()	EntityManager

Capítulo 2: Exploración y Planificación

	Entidad
Actualizar()	EntityManager Entidad
ObtenerXId()	EntityManager Entidad
ObtenerUsuario()	EntityManager Entidad
Eliminar()	EntityManager Entidad
FindEntidades()	Query EntityManager
ComprobarAutenticacion()	Query EntityManager
IniciarSesion()	Query EntityManager
PostergarMant()	Query EntityManager
BuscarporRango()	
GetEntityManager()	EntityManager
MediosXmantPlan()	Query

	EntityManager Medio
MediosXmantReal()	Query EntityManager Medio

Tabla 17: Tarjeta CRC #3.

Nombre: Cntr_CargarExcel	
Responsabilidad	Colaboraciones
receiveUpload()	Cntr_JPAGenericoImpl Datos

2.11 Conclusiones del capítulo

Al finalizar el presente capítulo se concluye que la confección del modelo conceptual del negocio contribuyó a la comprensión de los conceptos que se relacionan y a una mayor integración del cliente con el equipo de desarrollo. Se definieron las funcionalidades del sistema y a su vez se elaboró el plan de iteraciones, el mismo cuenta con 4 iteraciones para un total de 8 HU que describen los aspectos principales a tener en cuenta para el desarrollo de la propuesta de solución. Asociado a estas HU se construyó el plan de entregas con un total de 3 entregas. Se elaboraron las tarjetas CRC que representan las funcionalidades a implementar, definiendo las responsabilidades y los colaboradores. Se describió la arquitectura empleada, así como los patrones de diseño utilizados.

Capítulo 3 Implementación y Pruebas

En este capítulo se muestra el diagrama Entidad-Relación y los estándares de codificación para un mejor entendimiento del sistema. Se analizan los resultados de la construcción del sistema para la planificación y control de los mantenimientos a los medios electrodomésticos de la Residencia 2 de la UCI. Se realizan las tareas de ingeniería generadas por cada una de las historias de usuario y las pruebas que propone la metodología empleada.

3.1 Diagrama Entidad-Relación

En la siguiente figura 10 se muestra el Diagrama Entidad-Relación de la propuesta de solución que representa el modelo físico correspondiente a la base de datos.



Figura 11: Diagrama Entidad-Relación

A continuación se muestra una descripción de las tablas que conforman la base de datos que la aplicación utiliza para un mayor entendimiento de la información que se gestiona:

- **mod_sistema.tb_usuario:** Se encarga de almacenar la información de los usuarios del sistema: Administrador, Supervisor y Técnico.
- **mod_sistema.tb_componente:** Se encarga de almacenar los datos de los componentes de un medio.
- **mod_sistema.tb_medio:** Se encarga de almacenar los datos de un medio.

- **mod_sistema.tb_traslado:** Almacena los datos para cuando un equipo sea trasladado de su lugar de origen.
- **mod_sistema.tb_solicitud:** Almacena los datos de cuando se genera un reporte con la rotura de un equipo.
- **mod_sistema.tb_mantenimiento_realizado:** Almacena el tipo de mantenimiento que se le realizó a un medio y otros datos.
- **mod_sistema.tb_mantenimiento_planificado:** Almacena los datos para la planificación de un medio.

3.2 Estándares de codificación

El código es la forma principal de documentación y comunicación entre los desarrolladores, por lo que debe estar lo más claro escrito, de manera que pueda ser identificado fácilmente por cualquier programador de otro equipo de desarrollo. Para la escritura del código fuente, el equipo de desarrollo se rige por reglas que son llamadas estándares de codificación. A continuación se muestran los utilizados en la implementación de la propuesta de solución.

- Los nombres de las tablas comienzan con el prefijo “mod_sistema.tb_” y se escriben en singular y minúscula.
- Los comentarios multilíneas comienzan con los caracteres “/*” y terminan con “*/”, los comentarios de una sola línea comienzan con los caracteres “//”.
- Los nombres de las variables compuestas se escribirán 1er nombre_ 2do nombre y en minúsculas.
- Los nombres de las clases controladoras se escriben Cntr_nombre.

3.3 Tareas de ingeniería

Las HU se descomponen en tareas de programación o ingeniería (TI), que se asignan a un equipo de desarrollo o persona. Estas tareas no tienen que necesariamente ser entendidas por el cliente, pueden ser escritas en lenguaje técnico y son para el uso estricto de los programadores. Es necesario aclarar que el equipo de desarrollo tiene en cuenta para las tareas de ingeniería que una semana equivale a cinco días.

Fueron redactadas 8 HU y se desglosaron en un total de 28 TI, en este epígrafe solo se muestran las TI más relacionadas con el negocio correspondientes a las HU descritas en el capítulo 2, las restantes se encuentran en los anexos 2, 3,4 y 5 de la investigación.

Para la **primera iteración** se definieron las siguientes TI:

HU.1: Gestionar usuario.

- TI.1 Crear usuario.
- TI.2 Visualizar usuario.
- TI.3 Modificar usuario.
- TI.4 Eliminar usuario.

HU.2: Usuario perfil

- TI.5 Autenticar usuario.
- TI.6 Editar perfil.

En el Anexo 2 se muestran las restantes TI definidas para esta iteración.

Para la **segunda iteración** se definieron las siguientes TI:

HU.3: Gestionar y trasladar medio.

- TI.7 Adicionar medio.
- TI.8 Visualizar medio.
- TI.9 Modificar medio.
- TI.10 Eliminar medio.
- TI.11 Realizar traslado del medio.

HU4: Componente

- TI.12 Adicionar componente.
- TI.13 Realizar traslado del componente.

Tabla 18: TI Adicionar medio.

Tarea	
Número de tarea: 7	Número de HU: 3
Nombre de la tarea: Adicionar medio	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 2 de marzo de 2015	Fecha fin: 2 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	

Descripción: Implementar la funcionalidad adicionar medio en el sistema.

Tabla 19: TI Realizar traslado del medio.

Tarea	
Número de tarea: 11	Número de HU: 3
Nombre de la tarea: Realizar traslado del medio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 6 de marzo de 2015	Fecha fin: 6 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: Implementar la funcionalidad que permita cambiar la ubicación de un medio.	

En el [Anexo 3](#) se muestran las restantes TI definidas para esta iteración.

Para la **tercera iteración** se definieron las siguientes TI:

HU5: Solicitudes

- TI.14 Añadir solicitud.
- TI.15 Solicitudes atendidas.
- TI.16 Solicitudes pendientes

HU6: Planificar mantenimientos

- TI.17 Estudiar parámetros para planificar los mantenimientos a los medios y cargar Excel.
- TI.18 Configurar planificación.
- TI.19 Períodos de mantenimientos.
- TI.20 Cargar Excel.

Tabla 20: TI Añadir solicitud.

Tarea	
Número de tarea: 14	Número de HU: 5

Nombre de la tarea: Añadir solicitud.	
Tipo de tarea: Desarrollo	Puntos estimados: 3 días
Fecha inicio: 23 de marzo de 2015	Fecha fin: 25 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Implementar la funcionalidad añadir una solicitud de reporte en el sistema.	

Tabla 21: TI Cargar Excel

Tarea	
Número de tarea: 20	Número de HU: 6
Nombre de la tarea: Cargar Excel	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 10 de abril de 2015	Fecha fin: 13 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Implementar la funcionalidad cargar Excel para poder obtener la información de los medios electrodomésticos de la Residencia 2.	

En el [Anexo 4](#) se muestran las restantes TI definidas para esta iteración.

Para la **cuarta iteración** se definieron las siguientes TI:

HU7: Mantenimientos

- TI.21 Visualizar Mantenimientos planificados.
- TI.22 Visualizar Mantenimientos realizados.
- TI.23 Confirmar mantenimiento realizado.
- TI.24 Postergar planificación del mantenimiento.
- TI.25 Recibir notificaciones de mantenimientos realizados.
- TI.26 Recibir notificaciones de la planificación de los mantenimientos.

HU8: Buscar y exportar

- TI.27 Buscador.
- TI.28 Exportar a PDF.

Tabla 22: TI Buscador

Tarea	
Número de tarea: 27	Número de HU: 8
Nombre de la tarea: Buscador	
Tipo de tarea: Desarrollo	Puntos estimados: 3 días
Fecha inicio: 4 de mayo de 2015	Fecha fin: 6 de mayo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Implementar la funcionalidad Buscador que permitirá buscar un medio por el criterio de búsqueda número de inventario.	

Tabla 23: TI Exportar a pdf.

Tarea	
Número de tarea: 28	Número de HU: 8
Nombre de la tarea: Exportar a pdf.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 7 de mayo de 2015	Fecha fin: 8 de mayo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema permitirá exportar a pdf la cantidad de acciones de mantenimiento que se han realizado y la cantidad de mantenimientos que quedan pendientes, así como el tipo de mantenimiento: correctivo o preventivo. Además debe mostrar la descripción del medio al que se le realizó dicho mantenimiento.	

En el [Anexo 5](#) se muestran las restantes TI definidas para esta iteración.

3.4 Pruebas

La metodología de desarrollo XP propone que se realicen pruebas al *software*, tanto como sea posible. Esto permite aumentar la calidad de los sistemas al reducir el número de errores no detectados y disminuir el tiempo transcurrido entre la aparición de un error y su detección. (Beck, 1999)

Esta metodología divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores y, pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y diseñadas por el cliente final. (Beck, 1999)

3.4.1 Desarrollo dirigido por pruebas

El desarrollo dirigido por pruebas (TDD¹⁶), es una técnica de diseño e implementación de software incluida dentro de la metodología XP. Consiste en guiar el desarrollo de una aplicación, por medio de Test Unitarios; que son algoritmos que emulan lo que la aplicación debería hacer. Esta técnica permitirá saber: qué, cómo, cuáles y cuántos algoritmos se necesitarán desarrollar para que la aplicación funcione como se espera y no tenga fallos imprevistos (Bahit, 2013)

Una vez obtenido los resultados satisfactorios de las pruebas se pasa a producción y se inicia el proceso de refactorización que consiste en limpiar y organizar el código, adaptarlo a los patrones y aumentar su legibilidad, sin modificar su comportamiento externo. (Koskela, 2008)

3.4.2 Pruebas unitarias

Las pruebas unitarias se le realizan al código de la aplicación. Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. (Joskowicz, 2008)

Para el caso específico de Java, que es el lenguaje de programación que se utiliza, es empleada la herramienta JUnit en el IDE Eclipse y las pruebas se realizan por medio de casos de pruebas. A continuación se muestra un ejemplo de prueba unitaria a la clase Cntr_JPAGenericoImpl(), las restantes se encuentran en el [Anexo 6](#) de la investigación.

¹⁶ Del inglés Test Driven Development.



Figura 12: Prueba unitaria Cntr_JPAGenericoImpl().

Una vez realizadas las pruebas se muestran los resultados de la siguiente manera:

Si la prueba arroja resultados satisfactorios, se visualiza un símbolo de aceptación de color verde como se evidencia en la figura anterior y en caso contrario se le indica al usuario a través de un símbolo de error de color rojo. También se puede observar en la interfaz de JUnit la cantidad de ejecuciones realizadas, los errores detectados, los fallos y el tiempo de ejecución de cada prueba.

3.4.3 Pruebas de aceptación

Las pruebas de aceptación la realiza el propio cliente en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Significan la satisfacción del cliente con el producto desarrollado. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de los requisitos. (Ken, 2001)

Las pruebas de aceptación se crean en base a las HU en cada ciclo de iteración. El cliente es responsable de verificar que sean correctos los resultados de las pruebas y de indicar el orden de prioridad de resolución en caso de que el resultado no sea satisfactorio. Una HU se considera terminada al pasar de forma correcta todas las pruebas de aceptación. (Joskowicz, 2008)

Fueron realizados 28 casos de pruebas de aceptación (CPA). En este epígrafe solo se muestra los CPA de las funcionalidades descritas anteriormente en las TI, los restantes se encuentran en los anexos 7,8 y 9 de la investigación.

Para la **primera entrega** se definieron los siguientes CPA:

HU.1: Gestionar usuario.

- CPA.1 Crear usuario.
- CPA.2 Visualizar usuario.
- CPA.3 Modificar usuario.
- CPA.4 Eliminar usuario.

HU.2: Usuario perfil

- CPA.5 Autenticar usuario.
- CPA.6 Editar perfil.

HU.3: Gestionar y trasladar medio.

- CPA.7 Adicionar medio.
- CPA.8 Visualizar medio.
- CPA.9 Modificar medio.
- CPA.10 Eliminar medio.
- CPA.11 Realizar traslado del medio.

HU4: Componente

- CPA.12 Adicionar componente.
- CPA.13 Realizar traslado del componente.

Tabla 24: CPA Adicionar medio

Caso de prueba de aceptación	
Código: HU3_CPA7	Historia de usuario: 3
Nombre: Adicionar medio	
Descripción: Prueba para la funcionalidad que permite adicionar un medio.	
Condiciones de ejecución:	

✓ El usuario debe estar autenticado con el rol Administrador.
<p>Entrada/Pasos de ejecución:</p> <p>Entrada: La entrada consiste en los datos del medio que el usuario debe introducir para ser insertado en la base de datos como son: número de inventario, tipo de medio, estado, ubicación, período de mantenimiento y alguna observación.</p> <p>Pasos de ejecución: Para adicionar un medio debe ir al menú y seleccionar la opción Medios/Adicionar medio. En la ventana que se va a mostrar debe introducir los datos de un medio y presionar en el botón "Registrar".</p>
Resultado esperado: El medio ha sido adicionado correctamente.
Evaluación de la prueba: Prueba satisfactoria.

En el Anexo 7 se muestran los restantes CPA definidas para esta entrega.

Para la **segunda entrega** se definieron los siguientes CPA:

HU5: Solicitudes

- CPA.14 Añadir solicitud.
- CPA.15 Solicitudes atendidas.
- CPA.16 Solicitudes pendientes

HU6: Planificar mantenimientos

- CPA.17 Planificar mantenimientos
- CPA.18 Configurar planificación.
- CPA.19 Períodos de mantenimientos.
- CPA.20 Cargar Excel.

Tabla 25: CPA Añadir solicitud.

Caso de prueba de aceptación	
Código: HU5_CPA14	Historia de usuario: 5
Nombre: Añadir solicitud.	

<p>Descripción: Prueba para la funcionalidad que permite solicitar un reporte de un medio para su mantenimiento correctivo. Para realizar esta operación debe dirigirse a la funcionalidad Solicitud/Añadir solicitud en el menú superior e ingresar los datos que se muestran</p>
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor.
<p>Entrada/Pasos de ejecución:</p> <p>Entrada: Los datos para generar la solicitud como son: tipo de medio, edificio, apartamento, nombre de quien reporta y una descripción de la rotura del medio.</p> <p>Pasos de ejecución: Para realizar un reporte debe ir al menú y seleccionar la opción Solicitudes/Adicionar solicitud y se mostrará un formulario, una vez introducidos los datos se presiona el botón “Registrar” y la solicitud será generada.</p>
<p>Resultado esperado: Solicitud generada correctamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 26: CPA Planificar mantenimientos.

Caso de prueba de aceptación	
Código: HU6_CPA17	Historia de usuario: 6
Nombre: Planificar mantenimientos.	
Descripción: Prueba para la funcionalidad que permite planificar los mantenimientos preventivos.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. ✓ Debe haber cargado el Excel e introducir datos que se le pedirán. 	

<p>Entrada/Pasos de ejecución:</p> <p>Entrada: Los datos que el usuario introduzca para que el sistema pueda planificar los mantenimientos a los medios.</p> <p>Pasos de ejecución: Una vez que el usuario con rol Administrador entre en el sistema, se le mostrará una guía de los pasos que debe seguir para planificar los mantenimientos preventivos. Primeramente debe cargar el Excel que contiene los medios de la Residencia 2, luego se mostrará un formulario con campos a llenar sobre la cantidad de técnicos, horas laborales, y el tiempo promedio que se demora un técnico en arreglar un tipo de medio específico. Además debe introducir los períodos de mantenimientos de cada tipo de medio. Finalizado esto se mostrará la planificación de los mantenimientos preventivos de los medios.</p>
<p>Resultado esperado: Mantenimiento planificado.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

En el Anexo 8 se muestran los restantes CPA definidas para esta entrega.

Para la **tercera entrega** se definieron las siguientes CPA:

HU7: Mantenimientos

- CPA.21 Visualizar Mantenimientos planificados.
- CPA.22 Visualizar Mantenimientos realizados.
- CPA.23 Confirmar mantenimiento realizado.
- CPA.24 Postergar planificación del mantenimiento.
- CPA.25 Recibir notificaciones de mantenimientos realizados.
- CPA.26 Recibir notificaciones de la planificación de los mantenimientos.

HU8: Buscar y exportar

- CPA.27 Buscador.
- CPA.28 Exportar a PDF.

Tabla 27: CPA Exportar a PDF.

Caso de prueba de aceptación	
Código: HU8_CPA28	Historia de usuario: 8
Nombre: Exportar a PDF	

Descripción: Prueba para la funcionalidad que permite exportar a PDF la cantidad de acciones de mantenimiento que se han realizado y la cantidad de mantenimientos que quedan pendientes, así como el tipo de mantenimiento: correctivo o preventivo. Además debe mostrar la descripción del medio al que se le realizó dicho mantenimiento.
Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor.
Entrada/Pasos de ejecución: Pasos de ejecución: Para poder exportar a PDF los mantenimientos realizados y los que quedan pendientes debe ir a menú a la funcionalidad Mantenimientos/Exportar a PDF. Otra manera de exportar datos es Medios/Visualizar medios, se mostrará el listado de medios y la opción de “Exportar a PDF”.
Resultado esperado: Se ha exportado correctamente la información.
Evaluación de la prueba: Prueba satisfactoria.

En el [Anexo 9](#) se muestran los restantes CPA definidas para esta entrega.

3.5 Análisis de los resultados de las pruebas

Se le realizaron al sistema las pruebas de aceptación para comprobar el cumplimiento de los requisitos funcionales y aspectos no funcionales del software. Los errores encontrados en la interfaz u ortografía son clasificados como no conformidades no significativas (NCNS) y los errores en las funcionalidades implementadas como no conformidades significativas (NCS). De manera más detallada se muestra en el siguiente gráfico, cómo se comportó la cantidad de no conformidades encontradas en cada una de las iteraciones realizadas:

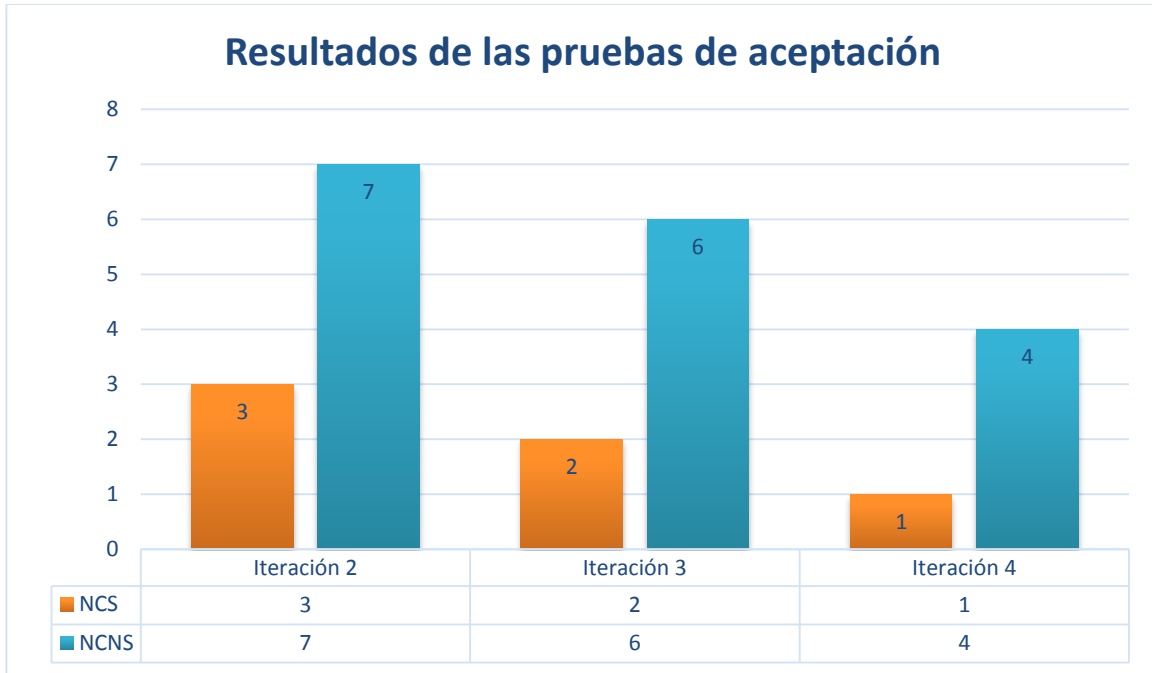


Figura 13: Gráfico de los resultados de las pruebas de aceptación

Todas las no conformidades fueron corregidas en su totalidad, lo que permitió dejar el sistema listo para su posterior uso.

3.6 Conclusiones del capítulo

En este capítulo se generaron artefactos de gran importancia para llevar a cabo la implementación de la propuesta de solución, como el Diagrama Entidad-Relación y las tareas de ingeniería necesarias para desarrollar cada historia de usuario. Se definieron los estándares de codificación, lo que permitió la homogeneidad del código de la propuesta de solución. Fueron escritas y ejecutadas las pruebas unitarias y de aceptación, para comprobar que la aplicación cumplía con los requisitos establecidos y que se encontraba lista para ser desplegada en un entorno real. Las pruebas de aceptación arrojaron como resultado un total de 6 no conformidades significativas y 17 no conformidades no significativas pero todas fueron resueltas en el transcurso de la implementación.

Conclusiones generales

Al finalizar la investigación se arribaron a las siguientes conclusiones:

- El estudio de soluciones similares permitió identificar la necesidad de desarrollar un sistema informático que gestione los mantenimientos de los medios electrodomésticos en la Residencia 2 de la UCI.
- Con el diseño del sistema informático para la gestión del mantenimiento de los equipos electrodomésticos de la Residencia Estudiantil 2 de la UCI se definieron 8 HU, de estas, fueron implementadas 2 dirigidas a la gestión de los usuarios y las restantes 6 a la gestión del mantenimiento.
- Con la realización de las pruebas de aceptación se detectaron 23 no conformidades, 6 significativas y 17 no significativas, las mismas fueron resueltas en el transcurso de la implementación y se comprobó que el software cuenta con la calidad requerida para ser desplegado en un entorno real.
- El cliente firmó una carta de aceptación que representa su satisfacción con el sistema.

Recomendaciones

A partir del trabajo realizado y los resultados obtenidos se recomienda:

- Socializar el uso del software en el resto de las residencias de la universidad.
- Añadir nuevas funcionalidades que permitan obtener reportes estadísticos y gráficos de los mantenimientos realizados en un período determinado de tiempo.

Referencias bibliográficas

1. **Adtriboo**. Java, características y ventajas de uno de los lenguajes de programación más populares. [En línea]. 21 Enero 2014. [Accedido 6 Mayo 2015]. Disponible en: <http://blog.adtriboo.com/2014/01/21/java-ventajas-caracteristicas-lenguaje-programacion-mas-populares/>.
2. **ALMEIRA, Adriana Sandra and PEREZ Cavenago, Vanina**. Arquitectura de Software: Estilos y Patrones. [En línea]. Argentina: Universidad Nacional De La Patagonia San Juan Bosco, 2007. [Accedido 20 Abril 2015]. Disponible en: <http://www.dit.ing.unp.edu.ar/graduate/bitstream/123456789/203/1/Tesina%20Arquitectura%20de%20Soft.pdf>.
3. **AMBLER, S.W.** Process Patterns: Building Large-Scale Systems Using Object Technology. New York: SIGS Books/Cambridge University Press. 1998
4. **Apache Maven Project**. Welcome to Apache Maven. [En línea]. [Accedido 12 Marzo 2015]. Disponible en: <http://maven.apache.org/>.
5. **BAHIT, Eugenia**. POO y MVC en PHP. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC [En línea]. 2011. Disponible en: <http://eugeniabahit.blogspot.com/2011/07/poo-y-mvc-en-php.html>.
6. **BAHIT, Eugenia**. Hackers & Developers. Magazine digital de distribución mensual sobre Software Libre, Hacking y Programación para profesionales del sector de Tecnologías de la Información. 2013.
7. **BECK, Kent**. Extreme Programming Explained. s.l. : Addison-Wesley Professional, 1999.
8. **BECK, K. y CUNNINGHAM, W.** A Laboratory for Teaching Object Oriented Thinking. In: Conference Proceedings on Object-oriented Programming Systems, Languages and Applications [En línea]. New York, NY, USA: ACM, 1989. p. 1–6. [Accedido 7 Abril 2015]. OOPSLA'89. ISBN 0-89791-333-7. Disponible en: <http://doi.acm.org/10.1145/74877.74879>
9. **CÁCERES, Paloma; MARCOS, Esperanza y KYBELE, Grupo**. Procesos ágiles para el desarrollo de aplicaciones Web. Taller de Web Engineering de las Jornadas de Ingeniería del Software y Bases de Datos de, 2001, vol. 2001. Disponible en: <http://dlsi.ua.es/webe01/articulos/s112.pdf>.
10. **CALABRIA, Luis y PÍRIZ, Pablo**. Metodología XP. Universidad ORT Uruguay : Cátedra de Ingeniería de Software., 2003.

11. **CEBALLOS PORRES, Juan Carlos.** “Eficientización del servicio de mantenimiento preventivo mensual de prensas mecánicas de propósito general” [En línea]. Guatemala: UNIVERSIDAD RAFAEL LANDÍVAR, 2004. Disponible en: <http://biblio3.url.edu.gt/Tesis/lote01/Ceballos-Juan.pdf>.
12. **CLOTET, Juan José.** Sistema de gestión de talleres SAT software servicio técnico programa sistema gestión talleres SatNetwork Soft talleres. *SatNetwork Sistemas Informáticos* [En línea]. 2015 2003. [Accedido 19 Enero 2015]. Disponible en: <http://www.satnetwork.com.ar>.
13. **COHN, Mike.** Differences Between Scrum and Extreme Programming. *Mountain Goat Software* [En línea]. [Accedido 16 Marzo 2015]. Disponible en: <http://www.mountangoatsoftware.com/blog/differences-between-scrum-and-extreme-programming>.
14. **CUARTAS PÉREZ, Luis Alberto.** Qué es el mantenimiento [En línea]. 2008. Disponible en: http://www.unalmed.edu.co/tmp/curso_concurso/area3/QUE_ES_EL_MANTENIMIENTO_MECANICO.pdf.
15. **Design Patterns.** Elements of Reusable Object-Oriented Software. s.l. : Pearson Education, 2003.
16. **EGUÍLUZ PÉREZ, Javier.** Introducción a JavaScript [En línea]. 2008. Disponible en: <http://www.librosweb.es/javascript>.
17. **EGUÍLUZ PÉREZ, Javier.** Introducción a CSS [En línea]. 2007. Disponible en: <http://www.librosweb.es/css/>.
18. **EUMED.NET.** Diccionarios, glosarios, vocabularios y catálogos de términos relacionados con la teoría económica, las finanzas y el comercio. [En línea]. [Accedido 19 Enero 2015]. Disponible en: <http://www.eumed.net/cursecon/dic/P3.htm>.
19. **FIGUEROA, Roberth G., CABRERA, Armando A. and SOLÍS, Camilo J.** Metodologías Tradicionales vs. Metodologías Ágiles. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
20. **FREE DOWNLOAD MANAGER.ORG. COPYRIGHT 2004-2015.** Free Download Manager - acelerador y gestor de descargas absolutamente gratuito. [En línea]. [Accedido 14 Enero 2015]. Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

21. **GARRIDO, Santiago García.** *Mantenimiento Correctivo. Organización y gestión de la reparación de averías* [En línea]. Madrid: RENOVETEC, 2009. Disponible en: <http://www.renovetec.com/mantenimientoindustrial-vol4-correctivo.pdf>.
22. **GONZÁLEZ, Carlos Sánchez.** Ciclo de vida de un proyecto XP. Capítulo 5. Metodología. [En línea]. Septiembre 2004. [Accedido 17 Marzo 2015]. Disponible en: <http://oness.sourceforge.net/proyecto/html/index.html>.
23. **GRÖNROOS, Marko.** *Libro de Vaadin. Vaadin 7 Edición-3ª Revisión* [En línea]. Creative Commons CC-BY -ND Licencia Versión 2.0, 2000. Disponible en: <https://vaadin.com/download/book-of-vaadin/vaadin-7/translations/es/book-of-vaadin-es.pdf>.
24. **IEEE-Std. 610** (1990) IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society Press.
25. **JEFFRIES Ron, ANDERSON, Ann and HENDRICKSON, Chet.** Extreme Programming Installed [En línea]. Addison-Wesley Professional, 2001. ISBN 9780201708424. Disponible en: http://www.google.com.cu/books?id=5ZuPjdO8LLoC&lpg=PR13&ots=OHXBtdUX_H&dq=Extreme%20Programming%20Explained.%20Embrace%20Change&lr&pg=PR13#v=onepage&q=Extreme%20Programming%20Explained.%20Embrace%20Change&f=false.
26. **JOSKOWICZ, José.** Reglas y prácticas en Extreme Programming. Universidad de Vigo. España, 2008. <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
27. **Ken Auer, Roy Miller.** Extreme Programming Applied. s.l. : Addison-Wesley Professional, 2001.
28. **KOSKELA, LASSE.** TEST DRIVEN, Practical TDD and Acceptance TDD for Java Developers. Manning Publication Co., 2008. ISBN 1-932394-85-0.
29. **LANCKER, Luc Van.** HTML5 y CSS3: Domine los estándares de las aplicaciones Web [2ª edición] [En línea]. Ediciones ENI, 2013. ISBN 9782746084063. Disponible en: <https://books.google.com.cu/books?id=hpW82IMDmxkC&printsec=frontcover&dq=inauthor:%22Luc+Van+Lancker%22&hl=es&sa=X&ei=i5K8VJe1EOzisATUqYDwDg&ved=0CDcQ6AEwBQ#v=onepage&q&f=false>
30. **LARMAN, Craig.** UML y Patrones. Introducción al análisis y diseño orientado a objetos. [En línea]. Prentice Hall, México, 1999. ISBN 970-17-0261-1. Disponible en: https://docs.google.com/file/d/0B_MWRyBDHOk-UDhZUzNFU2V4OWs/edit?pli=1.
31. **LERNER, Reuven M.** At the forge: Twitter bootstrap. Linux Journal, 2012 . vol. 2012, no 218, p. 6.

32. **LETELIER TORRES, Patricio and SÁNCHEZ LÓPEZ, Emilio A.** Metodologías Ágiles en el Desarrollo de Software. 12 Noviembre 2003. UNIVERSIDAD POLITECNICA DE VALENCIA.
33. **LETELIER, Patricio and M^a CARMEN PENADÉS.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *ww.cyta.com.ar/ta0502/v5n2a1.htm* [En línea]. 15 Abril 2006. [Accedido 18 Noviembre 2014]. Disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.
34. **MARCO, David.** Introducción a JPA 2.0 (I). [En línea]. 24 Enero 2010. [Accedido 5 Marzo 2015]. Disponible en: <http://www.davidmarco.es/articulo/introduccion-a-jpa-2-0-i>.
35. **MARCO, David.** Introducción a JPA 2.0 (IV). *davidmarco.es* [En línea]. 25 Febrero 2010. [Accedido 1 Abril 2015]. Disponible en: <http://www.davidmarco.es/articulo/introduccion-a-jpa-2-0-iv>.
36. **MARTINEZ, Rafael.** Sobre PostgreSQL | www.postgresql.org.es. [En línea]. [Accedido 20 Enero 2015]. Disponible en: http://www.postgresql.org.es/sobre_postgresql.
37. **MATEU, Carles.** Desarrollo de aplicaciones web [En línea]. Marzo 2004. [Accedido 4 Junio 2015] Barcelona, ISBN 84-9788-118-4. Disponible en: http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo_de_Aplicaciones_Web.pdf
38. Fundació per a la Universitat Oberta de Catalunya
39. **MORALES, Andrea.** De XP, Scrum, RUP, CMM (I), Crystal Clear y otras yerbas. *Ideas 3p. El blog técnico de Tercer Planeta* [En línea]. de mayo de 2007. [Accedido 16 Marzo 2015]. Disponible en: <http://blog.tercerplaneta.com/2007/02/de-xp-scrum-rup-cmml-crystal-clear-y.html>.
40. **NEGRINO, Tom y Smith, Dori.** Guía de aprendizaje JavaScript. Madrid: PEARSON EDUCACIÓN, 2001.
41. **PÉREZ GARCÍA, Alejandro.** JSF - Java Server Faces. *DesarrolloWeb.com* [online]. 21 Febrero 2006. [Accedido 5 Mayo 2015]. Disponible en: <http://www.desarrolloweb.com/articulos/2380.php>.
42. **REAL ACADEMIA ESPAÑOLA.** Diccionario de la lengua española. Real Academia Española [En línea]. 2015. [Accedido 14 Enero 2015]. Disponible en: <http://lema.rae.es/drae/?val=control>.
43. **REAL ACADEMIA ESPAÑOLA.** Diccionario de la lengua española. Real Academia Española [En línea]. 2015. [Accedido 14 Enero 2015]. Disponible en: <http://lema.rae.es/drae/?val=mantenimiento>.
44. **Renove Tecnología S.L.** SOFTWARE DE MANTENIMIENTO GRATUITO PMX-PRO y RENOVFREE. Renovetec [En línea]. 2014 2009. [Accedido 19 Enero 2015]. Disponible en: <http://www.renovetec.com/index.php/mantenimiento-industrial/307-software-de-mantenimiento-gratuito-pmx-pro>.

45. **RODRÍGUEZ, Txema, SILES, Fernando, QUIJANO, Juan and GUTIERREZ, Pedro.** Eclipse IDE. *Genveta: dev desarrollo y software* [En línea]. [Accedido 8 Marzo 2015]. Disponible en: <http://www.genbetadev.com/herramientas/eclipse-ide>
46. **RODRÍGUEZ RAMÍREZ, José A.** Gestión de Mantenimiento Asistido por Computadora [En línea]. Cujae, 2003. ISBN 959-261-109-2. Disponible en: <http://pmm-bs.com/doc/images/stories/biblioteca/libros/L1-Libro%20de%20Gestion%20de%20Mantenimiento.pdf>.
47. **RUMBAUGH, James, JACOBSON, Ivar and BOOCH, Grady.** El Lenguaje Unificado de Modelado. Manual de Referencia. Addison-Wesley, 1999.
48. **Servinet.** Software Gestión SAT | Programa Servicio Asistencia Técnica | ERP SAT. [En línea]. 2015. [Accedido 19 Enero 2015]. Disponible en: <http://www.softwaregestionsat.com/>.
49. **SIVIANES CASTILLO, Francisco, SÁNCHEZ ANTÓN, Gemma, ROPERO RODRÍGUEZ, Jorge, RIVERA ROMERO, Octavio, BENJUMER MONDÉJAR, Jaime, BARBANCHO CONCEJERO, Julio y ROMERO TERNERO, María del Carmen.** Servicios en Red [En línea]. Editorial Paraninfo, 2010. [Accedido 18 Enero 2015]. Disponible en: <http://books.google.com.cu/books?id=aOb3rTabO44C&lpg=PA160&dq=servidor%20web%20apache&pg=PR4#v=onepage&q=servidor%20web%20apache&f=false>
50. **Tangient LLC.** Desarrollo Ágil de Software en JAVA - *Frameworks* para Java. [En línea]. 17 Noviembre 2014. [Accedido 12 Marzo 2015]. San Francisco, CA 94103. Disponible en: <http://daitesrc.wikispaces.com/Frameworks+para+Java>.
51. **The Apache Software Foundation.** Apache Tomcat. [En línea]. [Accedido 6 Marzo 2015]. Disponible en: <http://tomcat.apache.org/>

Anexos

Anexo 1: Historias de usuario

Tabla 28: HU Gestionar usuario.

Historia de usuario	
Número: 1	Nombre del requisito: Gestionar usuario
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alto	
Descripción: El sistema permitirá al usuario con rol Administrador crear, visualizar, modificar y eliminar usuarios. Esta operación la podrá realizar en la funcionalidad “Usuarios” que se encuentra en el menú superior.	
Observaciones: El usuario deberá estar autenticado con el rol Administrador. El sistema le indicará los campos que son obligatorios para las operaciones que así lo requieran.	

Tabla 29: HU Usuario perfil

Historia de usuario	
Número: 2	Nombre del requisito: Usuario perfil
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alto	
Descripción: Para el desarrollo de las funcionalidades Autenticar usuario y Editar perfil el usuario debe estar autenticado en el sistema con el rol Administrador, Técnico o Supervisor. Una vez realizada esta operación se le mostrará al usuario la información a la que tiene acceso. También el sistema permitirá a los usuarios visualizar y modificar solamente su información personal.	
Observaciones: El usuario deberá autenticarse.	

Tabla 30: HU Componente.

Historia de usuario	
Número: 4	Nombre del requisito: Componente
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alto	
<p>Descripción: El sistema permitirá al usuario con rol Administrador adicionar o cambiar la ubicación de un componente de un medio a la base de datos. Para realizar la operación de trasladar un componente, se debe dirigir a Componentes/Componentes trasladados donde se generará una traza con las ubicaciones anteriores de dichos componentes.</p>	
<p>Observaciones: El usuario deberá estar autenticado. El sistema le indicará los campos que son obligatorios en caso de que la opción seleccionada así lo requiera y para poder realizarle el traslado a un componente de un medio</p>	

Tabla 31: HU Mantenimientos

Historia de usuario	
Número: 7	Nombre del requisito: Mantenimientos
Programador: Yoency Poll Guerrero y Yuliet Castillo Reyes.	Iteración Asignada: 4
Prioridad: Media	Tiempo Estimado: 10 días
Riesgo en Desarrollo: Medio	
<p>Descripción: Inicialmente el sistema mostrará una ventana de los medios a los que le corresponde realizarle mantenimiento preventivo, pero si se encuentra en otra funcionalidad y desea ver los mantenimientos planificados o los realizados debe dirigirse a la funcionalidad Mantenimientos/Mantenimientos planificados o</p>	

Mantenimientos/Mantenimientos realizados. Una vez mostrado los mantenimientos planificados tendrá la opción de confirmar el mantenimiento o postergarlo, la selección de esta última opción postergará por período de 1 semana la planificación. También el sistema permitirá a los usuarios seleccionar la opción de recibir notificaciones de mantenimientos realizados y de la planificación de los mantenimientos, opción que permitirá a los usuarios recibir correos electrónicos con la información seleccionada. Es válido aclarar que los usuarios con rol Técnico solo podrán visualizar los mantenimientos planificados, mientras que el Administrador y Supervisor si tienen acceso a las demás funcionalidades.

Observaciones: El usuario deberá estar autenticado. Para que el usuario reciba las notificaciones al correo, tendrá que seleccionar dicha opción en su perfil.

Anexo 2: Tareas de ingeniería para la iteración 1

Tabla 32: Crear usuario.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Crear usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 9 de febrero de 2015	Fecha fin: 10 de febrero de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: Se creará un usuario en el sistema.	

Tabla 33: Visualizar usuario

Tarea	
Número de tarea: 2	Número de HU: 1
Nombre de la tarea: Visualizar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 11 de febrero de 2015	Fecha fin: 11 de febrero de 2015

Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.
Descripción: El sistema mostrará los datos del usuario.

Tabla 34: Modificar usuario

Tarea	
Número de tarea: 3	Número de HU: 1
Nombre de la tarea: Modificar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 12 de febrero de 2015	Fecha fin: 12 de febrero de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema modificará los datos del usuario.	

Tabla 35: Eliminar usuario

Tarea	
Número de tarea: 4	Número de HU: 1
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 13 de febrero de 2015	Fecha fin: 13 de febrero de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	

Descripción: El sistema eliminará los datos del usuario.

Tabla 36: Autenticar usuario

Tarea	
Número de tarea: 5	Número de HU: 2
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 16 de febrero de 2015	Fecha fin: 17 de febrero de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema autenticará un usuario.	

Tabla 37: Editar perfil

Tarea	
Número de tarea: 6	Número de HU: 2
Nombre de la tarea: Editar perfil	
Tipo de tarea: Desarrollo	Puntos estimados: 3 días
Fecha inicio: 18 de febrero de 2015	Fecha fin: 20 de febrero de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Cada usuario podrá modificar la información de su perfil.	

Anexo 3: Tareas de ingeniería para la iteración 2

Tabla 38: TI Visualizar medio.

Tarea	
Número de tarea: 8	Número de HU: 3
Nombre de la tarea: Visualizar medio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 3 de marzo de 2015	Fecha fin: 3 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Implementar la funcionalidad visualizar medio en el sistema.	

Tabla 39: TI Modificar medio

Tarea	
Número de tarea: 9	Número de HU: 3
Nombre de la tarea: Modificar medio	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 4 de marzo de 2015	Fecha fin: 4 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: Implementar la funcionalidad modificar medio en el sistema.	

Tabla 40: TI Eliminar medio

Tarea	
Número de tarea: 10	Número de HU: 3
Nombre de la tarea: Eliminar medio	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 5 de marzo de 2015	Fecha fin: 5 de marzo de 2015

Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.
Descripción: Implementar la funcionalidad eliminar medio en el sistema.

Tabla 41: Adicionar componente.

Tarea	
Número de tarea: 12	Número de HU: 4
Nombre de la tarea: Adicionar componente.	
Tipo de tarea: Desarrollo	Puntos estimados: 3 días
Fecha inicio: 9 de marzo de 2015	Fecha fin: 11 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema permitirá adicionar un componente a un medio.	

Tabla 42: Realizar traslado del componente.

Tarea	
Número de tarea: 13	Número de HU: 6
Nombre de la tarea: Realizar traslado del componente.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 12 de marzo de 2015	Fecha fin: 13 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema permitirá cambiar la ubicación de un componente de un medio.	

Anexo 4: Tareas de ingeniería para la iteración 3

Tabla 43: Solicitudes atendidas.

Tarea	
Número de tarea: 15	Número de HU: 5
Nombre de la tarea: Solicitudes atendidas	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 26 de marzo de 2015	Fecha fin: 27 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema mostrará todas las solicitudes que se han atendido hasta el momento	

Tabla 44: Solicitudes pendientes

Tarea	
Número de tarea: 16	Número de HU: 5
Nombre de la tarea: Solicitudes pendientes	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 30 de marzo de 2015	Fecha fin: 31 de marzo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema mostrará todas las solicitudes pendientes que existen.	

Tabla 45: TI Estudiar parámetros para planificar los mantenimientos a los medios.

Tarea	
Número de tarea: 17	Número de HU: 6

Nombre de la tarea: Estudiar parámetros para planificar los mantenimientos a los medios y el Excel.	
Tipo de tarea: Investigación	Puntos estimados: 3 días
Fecha inicio: 1 de abril de 2015	Fecha fin: 3 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: Se investigarán los parámetros a tener en cuenta para una buena planificación.	

Tabla 46: Configurar planificación.

Tarea	
Número de tarea: 18	Número de HU: 6
Nombre de la tarea: Configurar planificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 6 de abril de 2015	Fecha fin: 7 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: El sistema permitirá configurar los datos de la planificación, para luego planificar los mantenimientos preventivos.	

Tabla 47: Períodos de mantenimientos

Tarea	
Número de tarea: 19	Número de HU: 6
Nombre de la tarea: Períodos de mantenimientos	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 8 de abril de 2015	Fecha fin: 9 de abril de 2015

Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes
Descripción: El sistema permitirá insertar los períodos de mantenimientos de cada medio, para poder planificar los mantenimientos preventivos.

Anexo 5: Tareas de ingeniería para la iteración 4

Tarea	
Número de tarea: 21	Número de HU: 7
Nombre de la tarea: Visualizar mantenimientos planificados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 20 de abril de 2015	Fecha fin: 20 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: El sistema mostrará la opción de confirmar mantenimiento.	

Tarea	
Número de tarea: 22	Número de HU: 7
Nombre de la tarea: Visualizar mantenimientos realizados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1 día
Fecha inicio: 21 de abril de 2015	Fecha fin: 21 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: El sistema mostrará la opción de confirmar mantenimiento.	

Tabla 48: Confirmar mantenimientos realizados.

Tarea	
Número de tarea: 23	Número de HU: 7
Nombre de la tarea: Confirmar mantenimientos realizados.	

Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 22de abril de 2015	Fecha fin: 23 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes	
Descripción: El sistema mostrará la opción de confirmar mantenimiento.	

Tabla 49: Postergar planificación del mantenimiento.

Tarea	
Número de tarea: 24	Número de HU: 7
Nombre de la tarea: Postergar planificación del mantenimiento.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 24 de abril de 2015	Fecha fin: 27 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema mostrará la opción de postergar el mantenimiento. Esta selección postergará por período de 1 semana la planificación prevista.	

Tabla 50: Recibir notificaciones de mantenimientos realizados

Tarea	
Número de tarea: 25	Número de HU: 7
Nombre de la tarea: Recibir notificaciones de mantenimientos realizados	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 28 de abril de 2015	Fecha fin: 29 de abril de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	

Descripción: El sistema permitirá a los usuarios seleccionar la opción de recibir notificaciones de mantenimientos realizados.

Tabla 51: Recibir notificaciones de la planificación de los mantenimientos.

Tarea	
Número de tarea: 26	Número de HU: 7
Nombre de la tarea: Recibir notificaciones de la planificación de los mantenimientos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 días
Fecha inicio: 30 de abril de 2015	Fecha fin: 1 de mayo de 2015
Programador responsable: Yoency Poll Guerrero, Yuliet Castillo Reyes.	
Descripción: El sistema permitirá a los usuarios seleccionar la opción de recibir notificaciones por correo electrónico de la planificación de los mantenimientos.	

Anexo 6: Pruebas unitarias

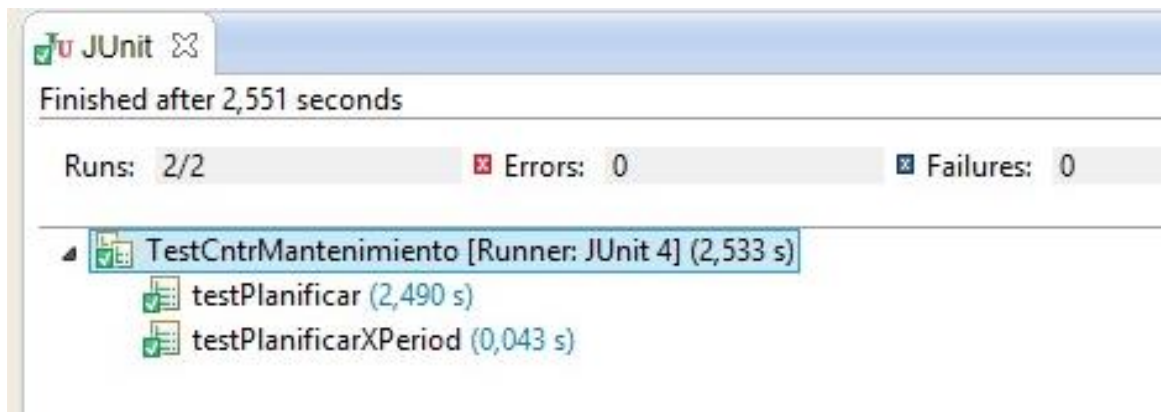


Figura 14: Prueba unitaria a la clase Cntr_Mantenimientos ().

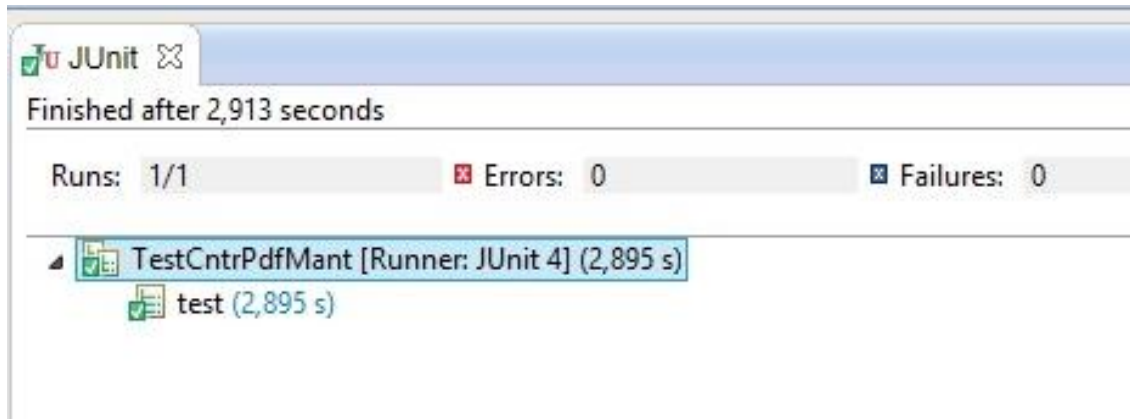


Figura 15: Prueba unitaria a la clase Cntr_Pdf_Mant ().

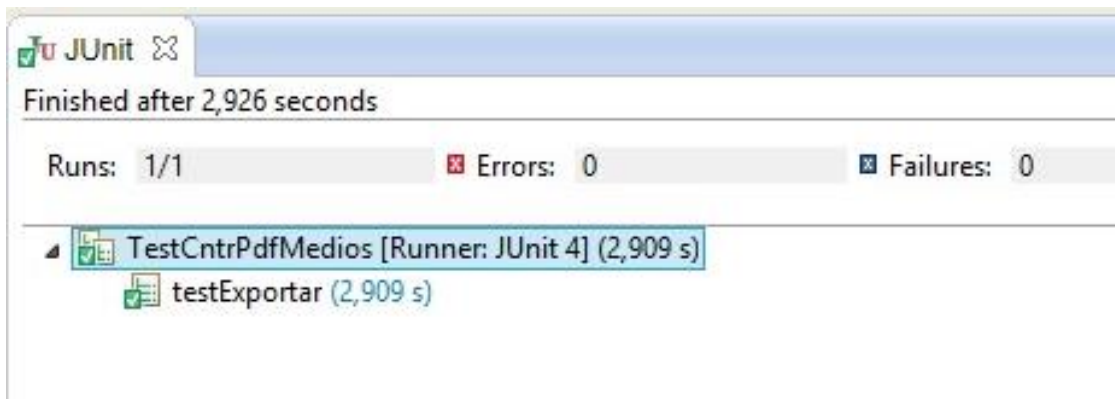


Figura 16: Prueba unitaria a la clase Cntr_Pdf_Medios ().

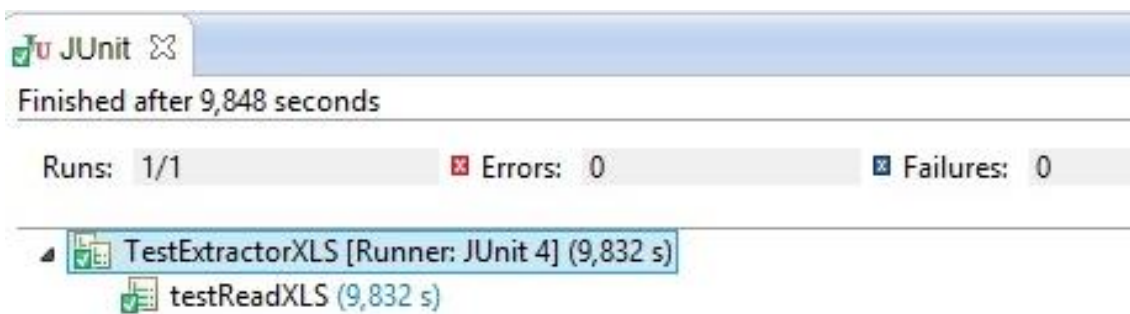


Figura 17: Prueba unitaria a la clase Cntr_ExtractorXLS ().

Anexo 7: Casos de pruebas de aceptación para la primera entrega

Tabla 52: Crear usuario.

Caso de prueba de aceptación	
Código: HU1_CPA1	Historia de usuario: 1
Nombre: Crear usuario.	
Descripción: Prueba para la funcionalidad que permite crear un usuario.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. 	
Pasos de ejecución/Entrada:	
<p>Entrada: La entrada consiste en los datos del usuario que el Administrador debe introducir para ser insertado en la base de datos.</p> <p>Pasos de ejecución: Para crear un usuario debe ir al menú y seleccionar la opción Usuarios/Crear usuario. En la ventana que se muestra se introducen los datos y una vez concluido se selecciona la opción "Registrar".</p>	
Resultado esperado: El usuario ha sido creado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 53: Visualizar usuario

Caso de prueba de aceptación	
Código: HU1_CPA2	Historia de usuario: 1
Nombre: Visualizar usuario	
Descripción: Prueba para la funcionalidad que permite visualizar los usuarios.	
Condiciones de ejecución:	

✓ El usuario debe estar autenticado con el rol Administrador, Supervisor o Técnico y además debe haber algún usuario insertado en la base de datos.
Pasos de ejecución: Pasos de ejecución: Para visualizar un usuario debe ir al menú y seleccionar la opción Usuarios/Visualizar usuarios y se mostrarán todos los usuarios que existan.
Resultado esperado: Se han visualizado todos los usuarios.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 54: Modificar usuario

Caso de prueba de aceptación	
Código: HU1_CPA3	Historia de usuario: 1
Nombre: Modificar usuario	
Descripción: Prueba para la funcionalidad que permite modificar un usuario.	
Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol Administrador, Técnico o Supervisor y además debe haber algún usuario insertado en la base de datos.	
Entrada/Pasos de ejecución: Entrada: Los datos del usuario que el Administrador, Técnico o Supervisor deseen modificar. Pasos de ejecución: Para modificar un usuario debe ir al menú y seleccionar la opción Usuarios/Visualizar usuarios. En la ventana que se va a mostrar debe presionar en el botón que indica editar un usuario. Una vez modificados los datos se presiona en el botón “Aceptar” y seguido se muestra enunciado de “Usuario modificado”.	
Resultado esperado: El usuario ha sido modificado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 55: Eliminar usuario

Caso de prueba de aceptación	
Código: HU1_CPA4	Historia de usuario: 1

Nombre: Eliminar usuario
Descripción: Prueba para la funcionalidad que permite eliminar un usuario.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador y además debe haber algún usuario insertado en la base de datos.
Pasos de ejecución: Pasos de ejecución: Para eliminar un usuario debe ir al menú y seleccionar la opción Usuarios/Visualizar usuarios. En la ventana que se va a mostrar debe presionar en el botón que indica eliminar un usuario y seguido se muestra un enunciado de "Usuario eliminado".
Resultado esperado: El usuario ha sido eliminado correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 56: Autenticar usuario

Caso de prueba de aceptación	
Código: HU2_CPA5	Historia de usuario: 2
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad que permite autenticar un usuario.	
Condiciones de ejecución:	
Entrada/Pasos de ejecución:	
Entrada: El usuario y contraseña del usuario.	
Pasos de ejecución: El usuario entrará su usuario y contraseña y presionará la opción Aceptar.	
Resultado esperado: Usuario autenticado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 57: Editar perfil.

Caso de prueba de aceptación	
Código: HU2_CPA6	Historia de usuario: 2
Nombre: Editar perfil.	
Descripción: Prueba para la funcionalidad que permite que los usuarios puedan modificar solamente su información personal.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador, Supervisor o Técnico. 	
Entrada/Pasos de ejecución:	
Entrada: Los datos que el usuario modifique de su perfil.	
Pasos de ejecución: Para que un usuario pueda modificar su perfil debe dirigirse al menú lateral izquierdo y seleccionar la opción “Editar perfil”, una vez modificados sus datos presiona el botón “Aceptar”.	
Resultado esperado: Se han modificado y guardado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 58: CPA Visualizar medio

Caso de prueba de aceptación	
Código: HU3_CPA8	Historia de usuario: 3
Nombre: Visualizar medio	
Descripción: Prueba para la funcionalidad que permite visualizar los medios.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador, Técnico o Supervisor y además debe haber algún medio insertado en la base de datos. 	
Entrada/Pasos de ejecución:	

Pasos de ejecución: Para visualizar los medios debe ir al menú y seleccionar la opción Medios/Visualizar medio y se mostrará un listado de los medios con sus datos.
Resultado esperado: Se muestra un listado de los medios con sus datos.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 59: CPA Modificar medio

Caso de prueba de aceptación	
Código: HU3_CPA9	Historia de usuario: 3
Nombre: Modificar medio	
Descripción: Prueba para la funcionalidad que permite modificar un medio.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador y además debe haber algún medio insertado en la base de datos. 	
Entrada/Pasos de ejecución:	
Entrada: Los datos del medio que el usuario desea modificar.	
Pasos de ejecución: Para modificar los datos de un medio debe ir al menú y seleccionar la opción Medios/Visualizar medio y se mostrará un listado de los medios con sus datos y diferentes funcionalidades. Se presiona el botón "Modificar" y se mostrará un formulario con los campos a modificar. Una vez modificados se presiona el botón "Aceptar".	
Resultado esperado: Se ha modificado correctamente el medio.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 60: CPA Eliminar medio

Caso de prueba de aceptación	
Código: HU3_CPA10	Historia de usuario: 3
Nombre: Eliminar medio	
Descripción: Prueba para la funcionalidad que permite eliminar un medio.	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador y además debe haber algún medio insertado en la base de datos.
<p>Entrada/Pasos de ejecución:</p> <p>Pasos de ejecución: Para eliminar un medio debe ir al menú y seleccionar la opción Medios/Visualizar medio y se mostrará un listado de los medios con sus datos y diferentes funcionalidades. Se presiona el botón “Eliminar” y se mostrará un mensaje preguntando “si estás seguro de eliminar el medio”.</p>
<p>Resultado esperado: El medio ha sido eliminado correctamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 61: Realizar traslado del medio.

Caso de prueba de aceptación	
Código: HU3_CPA11	Historia de usuario: 3
Nombre: Realizar traslado del medio.	
Descripción: Prueba para la funcionalidad que permite trasladar los medios.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador y además debe haber algún medio insertado en la base de datos. 	
<p>Entrada/Pasos de ejecución:</p> <p>Entrada: La nueva ubicación del medio a trasladar.</p> <p>Pasos de ejecución: Para realizar el traslado de un medio debe ir al menú y seleccionar la opción Medios/Visualizar medios. Se mostrará una ventana de los medios con sus datos y diferentes botones. Se presiona el botón “Editar” del medio que desea trasladar, se mostrará otra ventana para modificar la ubicación del medio, se selecciona la opción “Aceptar” y se mostrará el mensaje “Datos actualizados”.</p>	
Resultado esperado: Se ha trasladado el medio correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 62: Adicionar componente.

Caso de prueba de aceptación	
Código: HU4_CPA12	Historia de usuario: 4
Nombre: Adicionar componente.	
Descripción: Prueba para la funcionalidad que permite insertar un componente de un medio en el sistema.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. 	
Entrada/Pasos de ejecución:	
Pasos de ejecución: Para añadir un componente debe ir al menú y seleccionar la opción Componentes/Adicionar componentes y se mostrará una ventana con campos a llenar, luego de esto se presiona el botón "Aceptar".	
Resultado esperado: Componente insertado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 63: Realizar traslado del componente.

Caso de prueba de aceptación	
Código: HU4_CPA13	Historia de usuario: 4
Nombre: Realizar traslado del componente.	
Descripción: Prueba para la funcionalidad que permite trasladar un componente.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador y además debe haber algún medio insertado en la base de datos. 	
Entrada/Pasos de ejecución:	
Entrada: La nueva ubicación del componente a trasladar.	
Pasos de ejecución: Para realizar el traslado de un componente debe ir al menú y seleccionar la opción Componentes/Visualizar componentes. Se mostrará una ventana de los componentes con sus datos y diferentes botones. Se presiona el botón "Editar" del componente que se desea trasladar, se mostrará otra ventana para	

modificar la ubicación del componente, se selecciona la opción “Aceptar” y se mostrará el mensaje “Datos actualizados”.
Resultado esperado: El componente ha sido trasladado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 64: Solicitudes atendidas.

Caso de prueba de aceptación	
Código: HU5_CPA15	Historia de usuario: 5
Nombre: Solicitudes atendidas	
Descripción: Prueba para la funcionalidad que permite mostrar las solicitudes de reportes de los medios que se han atendido hasta el momento.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor. 	
Entrada/Pasos de ejecución: Pasos de ejecución: Para visualizar las solicitudes realizadas debe ir al menú y seleccionar la opción Solicitudes/Solicitudes realizadas y se mostrará un listado con los reportes de los medios que se han atendido hasta el momento.	
Resultado esperado: Datos mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 65: Solicitudes pendientes.

Caso de prueba de aceptación	
Código: HU5_CPA16	Historia de usuario: 5
Nombre: Solicitudes pendientes.	
Descripción: Prueba para la funcionalidad que permite mostrar las solicitudes de reportes de los medios que están pendientes.	

Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor.
Entrada/Pasos de ejecución: <p>Pasos de ejecución: Para visualizar las solicitudes pendientes debe ir al menú y seleccionar la opción Solicitudes/Solicitudes pendientes y se mostrará un listado con los reportes de los medios que están pendientes porque en algún momento el usuario lo postergó.</p>
Resultado esperado: Datos mostrados correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Anexo 8: Casos de pruebas de aceptación para la segunda entrega

Tabla 66: Configurar planificación.

Caso de prueba de aceptación	
Código: HU6_CPA18	Historia de usuario: 6
Nombre: Configurar planificación.	
Descripción: Prueba para la funcionalidad que permite configurar los parámetros de planificación para que el sistema automáticamente planifique los mantenimientos preventivos luego.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. 	
Entrada/Pasos de ejecución: <p>Entrada: Los datos para que el sistema automáticamente pueda planificar los mantenimientos luego.</p> <p>Pasos de ejecución: Para realizar un reporte debe ir al menú y seleccionar la opción Solicitudes/Adicionar solicitud y se mostrará un formulario, una vez introducidos los datos se presiona el botón “Registrar” y la solicitud será generada.</p>	
Resultado esperado: Parámetros guardados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 67: Períodos de mantenimientos.

Caso de prueba de aceptación	
Código: HU6_CPA19	Historia de usuario: 6
Nombre: Períodos de mantenimientos.	
Descripción: Prueba para la funcionalidad que permite al usuario con rol Administrador insertar los períodos de mantenimientos de cada medio para que el sistema luego planifique los mantenimientos preventivos.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. 	
Entrada/Pasos de ejecución: <p>Entrada: Un número que indicará el tiempo en que le corresponderá mantenimiento nuevamente a un medio.</p> <p>Pasos de ejecución: Para indicar cada cuanto tiempo, a un medio le corresponderá mantenimiento preventivo nuevamente, debe ir al menú lateral izquierdo y seleccionar la funcionalidad “Períodos de mantenimientos” e introducir el tiempo de cada medio.</p>	
Resultado esperado: Tiempo de período de mantenimiento guardado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 68: Cargar Excel.

Caso de prueba de aceptación	
Código: HU6_CPA20	Historia de usuario: 6
Nombre: Cargar Excel.	
Descripción: Prueba para la funcionalidad cargar el Excel.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador. 	
Entrada/Pasos de ejecución:	

Pasos de ejecución: Para cargar el Excel se va al menú lateral izquierdo, se selecciona la opción “Cargar datos desde Excel” y se selecciona la opción “Examinar” para cargar el archivo.
Resultado esperado: Se ha cargado el Excel.
Evaluación de la prueba: Prueba satisfactoria.

Anexo 9: Casos de pruebas de aceptación para la tercera entrega

Caso de prueba de aceptación	
Código: HU7_CPA21	Historia de usuario: 7
Nombre: Visualizar mantenimientos planificados.	
Descripción: Prueba para la funcionalidad que permite visualizar los mantenimientos planificados.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador, Supervisor o Técnico. 	
Pasos de ejecución/Entrada: Pasos de ejecución: Para visualizar los mantenimientos planificados debe ir al menú y seleccionar la opción Mantenimientos/Mantenimientos planificados. Realizado esto se muestran los medios con los datos: ubicación, identificación (#serie), tipo de medio, marca, modelo, fecha de la planificación de dicho medio y las opciones postergar o confirmar.	
Resultado esperado: Se visualizarán los medios planificados.	
Evaluación de la prueba: Prueba satisfactoria.	
Caso de prueba de aceptación	
Código: HU7_CPA22	Historia de usuario: 7
Nombre: Visualizar mantenimientos realizados.	
Descripción: Prueba para la funcionalidad que permite visualizar los mantenimientos realizados.	

Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor.
Pasos de ejecución/Entrada: Pasos de ejecución: Para visualizar los mantenimientos realizados debe ir al menú y seleccionar la opción Mantenimientos/Mantenimientos realizados. Realizado esto se muestran los medios que se han inspeccionado.
Resultado esperado: Se visualizarán los medios realizados.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 69: Confirmar mantenimientos realizados.

Caso de prueba de aceptación	
Código: HU7_CPA23	Historia de usuario: 7
Nombre: Confirmar mantenimientos realizados.	
Descripción: Prueba para la funcionalidad que permite confirmar los mantenimientos realizados.	
Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol Administrador	
Pasos de ejecución/Entrada: Entrada: La entrada consiste en los datos que el usuario con rol Administrador debe introducir como: la ubicación del medio, identificación (#serie), tipo de medio, marca, modelo, fecha de la planificación de dicho medio. Pasos de ejecución: Para confirmar un mantenimiento que se ha realizado debe ir al menú y seleccionar la opción Mantenimientos/Mantenimientos planificados. Realizado esto se muestran los medios con los datos: ubicación, identificación (#serie), tipo de medio, marca, modelo, fecha de la planificación de dicho medio y las opciones postergar o confirmar. En este caso se selecciona la opción Confirmar y el sistema muestra un formulario con los campos costo, tipo de mantenimiento y observaciones.	
Resultado esperado: El mantenimiento ha sido confirmado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 70: Postergar planificación del mantenimiento

Caso de prueba de aceptación	
Código: HU7_CPA24	Historia de usuario: 7
Nombre: Postergar planificación del mantenimiento	
Descripción: Prueba para la funcionalidad que permite postergar la planificación del mantenimiento.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador, Supervisor o Técnico y además debe haber planificado algún mantenimiento. 	
Entrada/Pasos de ejecución:	
<p>Pasos de ejecución: Para postergar un mantenimiento debe ir al menú y seleccionar la opción Mantenimientos/Mantenimientos planificados. Realizado esto se muestran los medios con los datos: ubicación, identificación (#serie), tipo de medio, marca, modelo, fecha de la planificación de dicho medio y las opciones postergar o confirmar. En este caso se selecciona la opción “Postergar” y el sistema muestra un mensaje “Mantenimiento postergado”.</p>	
Resultado esperado: La planificación se postergará.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 71: Recibir notificaciones de mantenimientos realizados.

Caso de prueba de aceptación	
Código: HU7_CPA25	Historia de usuario: 7
Nombre: Recibir notificaciones de mantenimientos realizados.	
Descripción: Prueba para la funcionalidad que permite recibir notificaciones al correo con los mantenimientos realizados.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol Administrador o Supervisor. 	
Entrada/Pasos de ejecución:	

Pasos de ejecución: Se selecciona la opción Editar perfil que se encuentra en el menú lateral izquierdo, se selecciona la opción “Editar” y si el usuario desea recibir correos de los mantenimientos que se han realizado marca el checkbox que tendrá dicha opción.
Resultado esperado: El usuario recibirá notificaciones.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 72: Recibir notificaciones de la planificación de los mantenimientos.

Caso de prueba de aceptación	
Código: HU7_CPA26	Historia de usuario: 7
Nombre: Recibir notificaciones de la planificación de los mantenimientos.	
Descripción: Prueba para la funcionalidad que permite recibir notificaciones al correo de los usuarios Administrador y Técnico sobre la planificación de los mantenimientos.	
Condiciones de ejecución:	
✓ El usuario debe estar autenticado con el rol Administrador o Supervisor.	
Entrada/Pasos de ejecución:	
Pasos de ejecución: Se selecciona la opción Editar perfil que se encuentra en el menú lateral izquierdo, se selecciona la opción “Editar” y si el usuario desea recibir correos de la planificación de los mantenimientos, marca el checkbox que tendrá dicha opción.	
Resultado esperado: El usuario recibirá notificaciones.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 73: Buscador.

Caso de prueba de aceptación	
Código: HU8_CPA27	Historia de usuario: 8
Nombre: Buscador.	
Descripción: Prueba para la funcionalidad que permite buscar un medio por su número de inventario.	

Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol Administrador, Supervisor o Técnico.
Entrada/Pasos de ejecución: Pasos de ejecución: Para buscar un medio por el criterio de búsqueda número de inventario, se debe dirigir al buscador que estará visible en las funcionalidades “Mantenimientos” y “Medios”.
Resultado esperado: Se ha encontrado el medio.
Evaluación de la prueba: Prueba satisfactoria.