"SRMC: Sistema gestor para reportes de incidencias de mantenimiento constructivo en la Universidad de las Ciencias Informáticas"

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

Autores: Yoan Pérez Alfonso.

Gabriel Hurtado Turiño.

Tutores: Ing. Osmany Montes de Oca Rodríguez

Ing. Yusdel Meriño Almaguer

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 4

16 de junio del 2015

Declaración de autoría

Declaramos que somos los únicos autores del trabajo **"SRMC: Sistema gestor para reportes de incidencias de mantenimiento constructivo en la Universidad de las Ciencias Informáticas"** y autorizamos al Departamento de Mantenimiento en la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____ _____

Autor: Yoan Pérez Alfonso **Autor:** Gabriel Hurtado Turiño

_____ _____

Tutor: Ing. Osmany Montes de Oca Rodríguez **Tutor:** Ing. Yusdel Meriño Almaguer

"La Revolución no se lleva en los labios para hablar de ella, se lleva en el corazón para morir por ella."

Dedicatoria

De Gabriel:

A mis padres por su apoyo incondicional, por ser mis guías, por su amor y comprensión.

A mi hermanita por ser lo más grande que tengo en la vida. Espero seguir siendo su ejemplo.

A mi viejita linda que ha sido mi única abuela y segunda madre.

A Rudy por llegar a mi vida desde bien pequeño e inculcarme valores excepcionales.

A la china por aparecer mágicamente en nuestras vidas y organizar todo lo que estaba patas arriba.

A mi dos hermanastros Elvis y Manolín, por permitirme conocer lo que es sentirse un hermano menor y por cada uno de los momentos compartidos.

A dos personas que a pesar de no estar desde hace mucho tiempo entre nosotros siempre quise y he querido tanto. Mi tía Luisa y mi abuelo Erasmo.

A Maide por poner en orden mi vida, por su amor, comprensión, paciencia, apoyo e incondicionalidad.

A mis amigos Edier, Davisito, Luis Ernesto y Ocmel. Por ser de esos amigos que nunca se olvidan aunque estén a kilómetros de distancia.

De Yoan:

A mi familia, fieles y leales amigos que ante todo cuento siempre con su apoyo.

A mi hermano, a pesar de ser tan insoportable, lo quiero con la vida.

A mi novia, por soportarme y mimarme durante estos años de universidad.

A los amigos que son parte de mi familia, sin importar la distancia y las dificultades siempre podré contar con ellos.

A mis padres, los motores en mi vida, esto es un éxito que sin ustedes no sería posible lograrlo.

Agradecimientos

De Gabriel:

A mi mamá por formarme desde pequeño, educarme, enseñarme a luchar y tener siempre metas trazadas; por ser madre y amiga; por siempre apoyar incondicionalmente mis decisiones; por hacerme el hombre que hoy soy.

A mi hermanita por ser esa personita especial que me impulsa en cada decisión que tomo, por ser lo más grande que tengo en la vida, por respetarme y por compartir todo lo que tiene de forma incondicional. Espero seguir siendo su ejemplo.

A mi papá por su apoyo, por ser esa especie de profesor que siempre le exige más y más a su alumno, por su apoyo en cada una de las decisiones tomadas durante mi vida por descabelladas que pudieron parecer, por poco a poco llegar a entender que ya su hijo se estaba convirtiendo en un hombre y respetarlo como tal. Por ser más que padre, amigo.

A mi viejita linda que ha sido mi única abuela y segunda madre. Por ser capaz de vivir con necesidad para ofrecerle a su nieto todo lo que tiene. Por pedirle a Dios a cada minuto que guíe mi camino. Por su fuerza y sus ganas de vivir.

A Rudy por llegar a mi vida desde bien pequeño e inculcarme valores excepcionales. Por siempre estar al tanto de lo que necesito. Por su paciencia, su respeto, su apoyo y su cariño.

A la china por aparecer mágicamente en nuestras vidas y organizar todo lo que estaba patas arriba. Por lograr que mi pequeña familia esté más unida. Por regalarme esos dos hermanos excepcionales a los cuales quiero tanto.

A mi dos hermanastros Elvis y Manolín, por permitirme conocer lo que es sentirse un hermano menor y por cada uno de los momentos compartidos. Especialmente a Manu que tiene ese don de virar el mundo al revés y hacer de la vida una fiesta.

A dos personas que a pesar de no estar desde hace mucho tiempo entre nosotros siempre quise y he querido tanto. Mi tía Luisa y mi abuelo Erasmo.

A Maide por llegar a mi vida y ponerla en orden, por ser mi pareja y mi amiga, por apoyarme siempre, por aguantarme, por tener tanta paciencia y nunca darse por vencida, por siempre luchar por

un nosotros, por permitirme ser parte de su vida, por los buenos momentos y los no tan buenos. Por ser tan especial.

A Petra, Marciano, Marquito, Gleny, Alex, Yoelvis, Yaubel, Mario, Rosa y los demás que conforman la familia de los muchos por permitirme ser un mucho más.

A mis amigos Edier, Davisito, Luis Ernesto y Ocmel. Por ser de esos amigos que nunca se olvidan aunque estén a kilómetros de distancia.

A Lázaro por ser como mi hermano desde el primer día de conocernos. Por su ayuda incondicional y por cada uno de los momentos compartidos.

A Rache que llegó para quedarse, por ser una persona tan especial.

A Henrry que tanto me apoyó y ayudó incondicionalmente.

A Angel Luis, a Yeni y a Ari, por todo lo que vivimos, por la ayuda que siempre me brindaron y por su amistad.

A Yanet y a Jochi por su apoyo, su amistad y por ese amor que surgió y el fruto que dará del cual fui cómplice.

A Leo por haberme brindado su amistad, por sus criterios dados sobre el SRMC y por ser mi apoyo y compañero durante el despliegue de fiscalía.

A René por su ayuda en varias ocasiones.

A mis amigos de la música que aunque no ganamos fama, nos demostramos que éramos capaces de lograr más de lo que creíamos podíamos lograr: Abelito, Roberto Daniel, Luis Felipe y Peña.

A Peña por ser mi amigo y haber sido mi guía en la mudanza a Cienfuegos. Por los momentos compartidos en la discomuro y por su ayuda.

A mis compañeros de aula, a los que están y los que no. Por cada momento compartido. En especial a Nolberto, el Viki y a Olga.

A los traquetos Jose, Doll y Andrés por convertirse en personas importantes en mi vida en tan poco tiempo.

Agradecimientos

A la tropa de la FEU por tantas noches de insomnio, por los logros y los fallos. Especialmente a Manuel, Livan, Roberto y a Enier por todo lo que me enseñaron y por los tantos momentos que compartimos juntos.

A la tropa de televisión y radio universitaria: Papito (mi maestro), Odalys, Linett, Yaily, Karel, Chavdar y Zenia. Por haberme permitido estar delante de las cámaras y detrás de los micrófonos desenvolviéndome en un mundo que no conocía. Por sus consejos, por RJ10, por Mediaplayer Deportivo y tantas cosas más.

A Arianna y Luz María por su amistad, ayuda y todas las rositas de maíz que hicimos juntos.

A la profe Rosalba que tanta paciencia me tuvo en los años siendo su alumno; por su ayuda y por ese carisma característico y ese amor con el que imparte sus clases.

A Osmany por haber aceptado tutorearnos cuando acudí a él. Por toda la ayuda brindada a pesar de estar desarrollando su tesis de maestría.

A Yusdel por ser de esas personas que conocí desde que puse por primera vez los pies en la UCI, por sus consejos y por su ayuda.

A Linet y Mayara que siguen ahí para lo que nos haga falta, gracias por su amistad.

A mi gemelo del alma Sandy que ha sido como mi hermano, por su ayuda, por permitirme incurrir también en el teatro y por parecerse tanto a mí. Por seguir siendo idénticos.

A mi compañero de tesis al que tanto le debo, no solo por compartir la tesis sino como amigo y hermano. Por su paciencia, conocimiento y capacidad como persona, dirigente y amigo.

A todos aquellos que de una forma u otra formaron parte en mi formación como ingeniero.

De Yoan:

Se termina mi paso por la universidad y resulta difícil recordar el primer año y empezar estas líneas. Por ello, sin ánimos de dejar de mencionar a alguien, doy gracias a todas y cada una de las personas que en algún momento nos dimos un abrazo, un beso, tuvimos una charla o simplemente estrechamos las manos. Ahora sí:

Agradecimientos

A la UCI, por haberme dado la oportunidad de formarme como profesional, por su rica y extensa vida universitaria. Por robarme horas de sueño, de estudio y de compartir con amistades, por permitirme hacer tantas cosas que ni imaginaba.

Al Docente 5, si las paredes hablaran contarían todos los momentos que viví en ese laberinto, por suerte no lo hacen.

A la FEU, esa organización que en su momento fui dirigente. Todos la llamamos el bichito que te dejan caer en el cuerpo y luego es imposible desprenderlo,

A los compañeros que fueron parte del Secretariado de la FEU, a los viejos Linet, Olia, Aballe, Leo, Mayara, de ellos aprendí muchísimo. A los nuevos Yasiel, Niurka, Pepe, Frank, Michel, Lázaro Gerardo, por los extensos y agotadores momentos que compartimos.

A los profesores y compañeros de clase, por soportarme y ayudarme siempre que podían; por los momentos alegres y otros no tanto pero igual se recuerdan todos.

Al MAA, gracias a Arcadio por permitirme ser parte de su colectivo de danza, por esas horas de ensayo para poder dar lo mejor en el escenario, a los compañeros de la cumbia y bachata. A Sandy, el flaco, por dejarme entrar al mundo teatral, recuerdo la primera obra en el primer año, pero nunca olvidaré "Las monjas", a eso le llamo hacer historia.

A Mario, Tocallo, Bell, Malidia, Yasiel, Ivan, Alberto, Arturo, amigos que hice en este paso por la universidad, por estar en el momento que necesité conversar o simplemente un consejo.

A Dianelis, Sandy, Dairelis, Yosleidys, Yelaine, y mi "prima" Evelyn, otro grupo de amigos muy especiales, por las horas de estudio, por darme aliento cuando suspendía un examen, por siempre acompañarme en mis inventos, travesuras y fiestas,. Gracias por todos y cada uno de los momentos que compartimos juntos.

Agradecimientos

A mis tutores Osmany y Yusdel, por cada recomendación y alerta, por su tiempo dedicado a este fin y por ser los Más Mejores.

A Rachy, Jorge Carlos, Nelson, Claudia, Lizandra, Yanelis, amigos que hacemos en algún momento de la vida y los vamos arrastrando, por ser parte de mi familia. Ellos conocen de los buenos y malos momentos por los que he pasado y siempre han estado ahí para escucharme.

A Gabriel, compañero de tesis y ante todo amigo, gracias por ser parte de este éxito en mi vida.

A mi familia, Tía Marlen, Alanys y Aliannys, mis abuelos Mima y Pipo, ustedes conocen lo que ha costado este trayecto estudiantil. No me alcanzaría la vida para agradecerles por cada consejo, por darme aliento, los quiero.

A Yanet, por llegar a mi vida y ser parte de ella, por ser amiga, compañera, consejera, por darme fuerzas cuando pensaba que todo estaba perdido. Thank you, I love you.

Al inso de mi hermano, espero que vea en mí un reflejo, te quiero con la vida.

A mis padres, Maribel y Juanito, ustedes han sido la luz al final del camino, me han guiado y apoyado en toda esta etapa estudiantil, han luchado para siempre salir adelante a pesar de las adversidades. Por cada entrega y dedicación de su parte, este es el fruto de su trabajo.

Resumen:

En la actualidad, el proceso de reportes de incidencias de mantenimiento constructivo en la Universidad de las Ciencias Informáticas se lleva a cabo de forma manual. Esto lo convierte en un proceso muy complejo, debido a que pueden provocarse reportes duplicados, falta de retroalimentación entre los trabajadores de mantenimiento y la comunidad universitaria. El presente trabajo se fundamenta en el desarrollo de una aplicación web para llevar a cabo este proceso, desarrollado con tecnologías libres. El progreso del sistema está guiado por la metodología de desarrollo Programación Extrema. Como lenguajes de programación del lado del cliente se utilizaron HTML, JavaScript, *JQuery* y CSS apoyado en el *framework* de diseño *Twitter-Bootstrap*. Como lenguaje del lado del servidor se manipuló PHP, a través del *framework* de desarrollo *Symfony*, apoyado en los componentes y librearías YAML, Twig y Doctrine. Se utilizó Apache como servidor web y *PostgreSQL* como gestor de bases de datos. Fueron implementadas veintiséis funcionalidades para satisfacer las necesidades del cliente y se realizaron las pruebas de software a la aplicación, las cuales garantizan un alto grado de satisfacción del cliente y en gran medida la correcta implementación de las funcionalidades.

Palabras clave: incidencia, mantenimiento, reporte.

Índice

Índice de ilustraciones

Índice de tablas

Introducción

El 23 de septiembre del 2002 abre las puertas a su primer curso académico la Universidad de las Ciencias Informática (UCI), con la visión futurista del Comandante en Jefe Fidel Castro Ruz. La remodelación de antiguos edificios existentes en la zona donde está actualmente situada la UCI y la construcción de otros, permitió que en solo 106 días surgiera lo que hoy es llamada “La Universidad del Futuro”.

La concepción y puesta en marcha de esta institución se desarrolló de forma distinta a como habían sido concebidas las demás universidades. Debido a la cantidad de estudiantes, profesores y trabajadores no docentes que actuarían en la universidad, la UCI se convertiría en una de las universidades más grandes y con mayor cantidad de instalaciones constructivas del país.

La universidad tiene más de 12 años de creada, de ahí que presente varios problemas de mantenimiento constructivo debido al desgaste que provoca años de explotación por una numerosa comunidad universitaria. En la universidad se pueden encontrar áreas conocidas como: docentes, residencia, rectorado, entre otras. Actualmente, algunas de estas presentan a diario incidencias que deben ser solucionadas para contribuir con el bienestar del personal que radica en la institución. La dirección de mantenimiento es la encargada de darle solución, pero presenta insuficiencia en la informatización de procesos, esto provoca retrasos en las prestaciones de servicios referentes a incidencias o afectaciones en el área de mantenimiento.

La UCI cuenta con 8 brigadas de trabajo y está distribuida en 6 áreas donde se presentan diariamente incidencias y afectaciones, sucesos que recaen en los hombros de los trabajadores de mantenimiento pertenecientes a la brigada correspondiente a dicha área, encargados de darle solución. En ocasiones varias de las incidencias son resueltas, pero otras no pueden ser mitigadas debido a la falta de recursos o por no contar con los materiales necesarios para dar solución.

El proceder de los trabajadores de mantenimiento es dar solución a estas afectaciones o una respuesta, en caso de no ser solucionadas. En ocasiones estas respuestas no son recepcionadas por la persona que realiza el reporte, debido a que debe realizar llamadas telefónicas constantes al técnico para conocer el estado del reporte, convirtiéndose en un proceso engorroso.

Actualmente este proceso se realiza vía telefónica por la persona que detectó la avería hacia el técnico que radica en las oficinas de mantenimiento. Este último se encarga de

procesar tal suceso de forma manual en un documento excel donde quedan plasmados datos de interés para el personal de la dirección como son el nombre de la persona que realiza el reporte, el número de solapín, el teléfono, la dirección donde se encuentra la afectación y una breve descripción de lo sucedido. El técnico después de haber recopilado los datos necesarios, realiza otra llamada telefónica o envía un correo al jefe de la brigada que atiende el área donde se presentó el incidente. El jefe se encarga de distribuir las tareas a partir de los reportes, por lo que asigna primeramente aquellas afectaciones prioritarias. Por su parte, los operarios se encargan de verificar la veracidad de la afectación y si estas son reales y pueden ser solucionados, el Jefe de la brigada solicita al almacén los recursos necesarios para dar solución a dichas afectaciones. Luego de contar con los recursos necesarios los operarios se encargan de mitigar la afectación. Este proceso se puede observar en el Anexo #1.

Este proceso no automatizado, genera un número del reporte (identificador) con el cual la persona que realizó dicho reporte podrá llamar para que el personal encargado de documentar los reportes haga una búsqueda y le informen el estado de la incidencia. Algunas situaciones tardan en ser solucionadas debido al cúmulo de reportes que se tienen diariamente. La lista de estos puede ser inmensa y la espera larga.

A partir de los resultados obtenidos, mediante la encuesta realizada vía web por el Centro de Innovación y Calidad de la Educación (CICE), a una muestra de 1144 usuarios de la UCI y las entrevistas no formales con el personal que trabaja en las oficinas de mantenimiento de la UCI, se detectaron las deficiencias siguientes:

- El 92.56% de la muestra no conoce el teléfono de la oficina donde radica el técnico de mantenimiento y el 94.23% no conoce donde radica la oficina donde atienden los reportes. Por lo que queda en manos de los instructores reportar la incidencia existiendo la posibilidad de que estos no la realicen en tiempo o no la realicen.
- Desvío de recursos al no poder realizar un control efectivo y sistemático sobre la cantidad de reportes existentes que se manejan en un documento excel.
- Al no contar con una herramienta automatizada, los reportes se hacen actualmente vía telefónica donde cualquier persona puede reportar una incidencia y el personal que lo recibe puede no detectar la duplicación del reporte y existe la posibilidad que la información sea falsa, comprometiendo la veracidad de las incidencias.
- No existe una herramienta a la cual los directivos de las distintas áreas puedan acceder para conocer el estado de los reportes; por lo que deben solicitar vía

correo esta información a la dirección de mantenimiento, para tener control y poder darle seguimiento a los problemas de mantenimiento de sus respectivas áreas.

- Se hace engorroso la asignación de prioridad a los diferentes reportes.
- Actualmente hay falta de retroalimentación ya que en varias ocasiones la incidencia reportada no es solucionada y el 44.34% de la muestra plantean que nunca reciben respuesta y el 51.78% en ocasiones.
- El 65.62% de la muestra plantea que es mejor recibir los reportes con una aplicación web y una notificación al correo.
- El 70.23% de la muestra plantea que es más efectivo realizar reportes mediante una aplicación web y vía telefónica.

Por lo planteado anteriormente, este trabajo tiene como **problema a resolver**: ¿Cómo contribuir a la gestión de incidencias sobre mantenimiento constructivo en las áreas de la UCI para facilitar el seguimiento de los reportes hechos por la comunidad universitaria?

A partir del problema de investigación identificado se definió como **objeto de estudio**: los procesos de reportes de incidencias.

El **campo de acción** se centra en los procesos de gestión de reportes de incidencias de mantenimiento constructivo en la UCI.

En el marco de este trabajo se trazó como **objetivo general:** Desarrollar una aplicación informática que contribuya con los procesos de gestión de incidencias sobre mantenimiento constructivo en las áreas de la UCI para facilitar el seguimiento de los reportes hechos por la comunidad universitaria.

En correspondencia con el objetivo general se propone realizar los siguientes **objetivos específicos**:

- Realizar un estudio del estado del arte de los sistemas de gestión de reportes de incidencias.
- Analizar y diseñar la aplicación para la gestión de reportes.
- Implementar la aplicación para la gestión de reportes.
- Validar mediante pruebas el funcionamiento de la aplicación informática.

Se tiene como **resultado esperado** que el desarrollo de una aplicación informática permitirá a la comunidad dar seguimiento a los procesos de reporte de incidencias de mantenimiento constructivo en la UCI.

Para la realización de esta investigación se utilizó la combinación de los **métodos teóricos y empíricos**, los que permitieron develar la parte de la ciencia que está siendo objeto de estudio. Entre los primeros se emplean: ***Histórico-lógico***: Se empleó al analizar la evolución histórica de los sistemas de gestión para concebir el sistema actual. ***Analítico-sintético***: Este método es empleado al analizar la documentación y elementos teóricos relacionados con las herramientas para la gestión de reportes.

Como métodos empíricos: se realizaron ***Entrevistas*** al cliente, usuarios y especialistas en el tema, para determinar las funcionalidades a implementar y desarrollar el sistema deseado. También, se realizó una ***Encuesta***, para recopilar datos por medio de un cuestionario previamente diseñado. Los datos se obtuvieron al realizar un conjunto de preguntas dirigidas a una muestra representativa de la población, con el fin de conocer si la comunidad universitaria conoce acerca del trabajo en la oficina de mantenimiento constructivo de la UCI.

La presente investigación está estructurada por un documento que contiene introducción, tres capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos. Los capítulos abordan las siguientes temáticas:

Capítulo 1. Fundamentación teórica: Enfatiza en los principales conceptos relacionados con los sistemas de gestión de reportes. Se realiza un estudio del estado del arte a nivel mundial y nacional, además de las herramientas, lenguajes de desarrollo y metodologías, para así determinar cuáles son las apropiadas en el desarrollo del sistema informático a desarrollar.

Capítulo 2. Características del sistema: Hace referencia a la propuesta del sistema a desarrollar. Queda plasmado el diseño de la aplicación informática, los requisitos que el sistema debe cumplir y los artefactos que propone la metodología seleccionada.

Capítulo 3. Implementación y prueba: Se plantean las funcionalidades que requiere el sistema, se logra cumplir el objetivo general de los estudios que se presentan y se describen las pruebas propuestas por la metodología seleccionada con el objetivo de asegurar la eficiencia de la solución.

Capítulo 1: Fundamentación teórica

Con el estudio de las metodologías, herramientas y tecnologías existentes, se pretende dar cumplimiento al objetivo propuesto. El presente capítulo tiene como principal objetivo exponer los fundamentos teóricos generales, los cuales son necesarios para ser tomados como punto de partida y dar solución al presente trabajo de diploma.

1.1 Conceptos asociados al dominio del problema

Para el estudio y desarrollo del presente trabajo es necesario el dominio de varios conceptos que permitirán una mejor comprensión del tema, mostrados a continuación:

Mantenimiento: Conjunto de operaciones y cuidados necesarios para que instalaciones, edificios e industrias puedan seguir funcionando adecuadamente (1).

Construcción: Toda nueva edificación que se ejecute o reparación de una ya deteriorada (2).

Mantenimiento constructivo: Se conceptúan aquellos trabajos cuyo objetivo es prevenir o corregir las afectaciones constructivas de elementos o partes aisladas de una obra, mediante la participación limitada de algunas especialidades constructivas, sin que esto modifique sus funciones, forma o dimensión original. Son los trabajos que tienen como objetivo, el cuidado sistemático y preventivo de una obra, no alteran el valor inicial del medio sobre el que se realizan y por lo general se ejecutarán en un período menor o igual a un año (3).

Incidencia: anomalía, consulta, mejora, inconsistencia. Lo que sobreviene en el decurso de un asunto o negocio y tiene con él alguna conexión (4).

Reporte es un informe, noticia o documento (puede ser impreso, digital, audiovisual) que pretende transmitir una información, aunque puede tener diversos objetivos. Existen reportes divulgativos, persuasivos y de otros tipos (5). La primera acción del reporte, luego de detectar/identificar una incidencia, es registrarla y hacerla pública, para que los encargados de darle solución procedan.

Importancia del reporte: El reporte de los incidentes permite responder a los mismos en forma sistemática, facilita una reparación rápida y eficiente, minimiza la pérdida de materiales y la interrupción de los servicios, mejora continuamente el proceso de tratamiento de incidentes.

Software gestor de reportes: es una herramienta que tiene como principal objetivo la creación de reportes, permite efectuar informes y generar consultas desde un sistema por medio de la utilización de cualquier tipo de base de datos independiente de su estructura y diseño (6).

1.2 Análisis de soluciones similares

El estudio del estado del arte y de aplicaciones existentes a nivel mundial, nacional y específicamente en la UCI, permite arribar a conclusiones y poder dar solución a la gestión de reportes de incidencias.

1.2.1 Herramientas internacionales

En el mundo existen un gran número de universidades y empresas que cuentan con sistemas informáticos de gestión, con el objetivo principal de resolver problemas específicos presentados en áreas que necesiten obtener reportes informáticos sobre sus datos. Entre los que se pueden citar:

Mantis: es una aplicación de software libre multiplataforma que permite gestionar las incidencias en una empresa, en un sistema o en un proyecto. Es un sistema fácil de usar y adaptable a varios escenarios. Además cuenta con diferentes *plugins* que aumentarán la capacidad de trabajo con la herramienta. Cuenta con una gran variedad de funcionalidades que permitirán que todos los objetivos queden cubiertos completamente (7).

Entre todas ellas cabe destacar (7):

- El reporte de incidencias, que permite a los distintos usuarios realizar reportes de cualquier tipo, ya sean incidencias técnicas, peticiones de soporte o *bugs* de un sistema. El usuario puede añadir un breve título a la incidencia, especificar una descripción detallada del mismo y algunos detalles técnicos si fueran necesarios. Estos comentarios pueden ser revisados por los responsables de solucionar las distintas incidencias, que a la vez pueden asignarlas a otras personas, añadir comentarios como respuesta o pedir datos para seguir correctamente el hilo de la actividad hasta que quede resuelta.
- Un sistema de permisos de usuario que permite identificar a los distintos usuarios que acceden al sistema. Los distintos niveles son: espectador, informador, actualizador, desarrollador, manager y administrador. Cada uno de ellos cuenta con unos roles específicos según las acciones que puedan realizar, por ejemplo:

el informador puede reportar incidencias y añadir nuevas notas, pero no puede ni asignar ni modificar los distintos defectos. Solo el administrador del sistema puede configurar los permisos de cada rol.

- Notificaciones de usuario: Mantis permite al usuario recibir notificaciones mediante correos electrónicos. De esta manera, tanto el responsable de resolver la incidencia como cualquier usuario, puede conocer cualquier dato o novedad en las incidencias detectadas.
- Flexibilidad para personalizar el sistema: pueden definirse nuevos estados para los defectos detectados, generar nuevos campos o personalizar los permisos de los distintos roles de usuario. Además, permite gestionar las etiquetas y organizar la información por proyectos, públicos o privados, y con acceso a diversos usuarios. Dentro de ellos, pueden generarse otros subproyectos o categorías.

Sistema de reportes de incidencias del Departamento de Seguridad Informática en la Universidad Nacional de Lujan en Argentina, herramienta que recoge reportes de incidencias registrándolas y haciéndolas pública entre los miembros del equipo de proyecto asignados para su análisis, resolución y seguimiento. Una vez reportada la incidencia identificada, el Analista Funcional lo analiza para determinar si la incidencia tiene que ver con el requerimiento de la universidad o si se encuentra fuera del alcance para la entrega comprometida y planificada. En caso de no corresponder para esa entrega y ser verdaderamente una incidencia, se le tratará de acuerdo con lo que se haya definido dentro del proyecto y pasa al Desarrollador quien efectuará una nueva evaluación de la misma (8).

Sistemas de gestión para la toma de decisiones: La Vicerrectoría de Investigación y Desarrollo, personal de Recursos Humanos y de Biblioteca de la universidad de Chile, cuentan con el sistema de información institucional *Business Objects*, que simplifica el proceso de medición, apoya la toma de decisiones y favorece la transparencia. La plataforma permite acceder fácilmente a reportes transversales, cifras, información estadística y gráficos de cada área. Los antecedentes pueden ser exportados a un documento excel o PDF, organizados de acuerdo a los criterios del usuario y analizados a través de diferentes herramientas de gestión dispuestas en el sistema (9).

En lo específico, el sistema de gestión de la Vicerrectoría de Investigación y Desarrollo reúne reportes de publicaciones generales, publicaciones por facultad, autores, tipos de documentos y artículos. El sistema de Recursos Humanos, administrado por la Dirección

de RR.HH, recoge reportes de contratos, remuneraciones, licencias y accidentabilidad, situaciones del personal y evaluación del académico. *Business Objects* de Bibliotecas contiene reportes de colecciones, infraestructura, presupuesto, recursos humanos, servicios presenciales, estudiantes y académicos de cada una de las 48 unidades bibliotecarias. La plataforma es administrada por el Sistema de Servicios de Información y Bibliotecas (SISIB) (9).

1.2.2 Herramientas nacionales

El estudio realizado está enfocado en los diferentes sistemas gestores de reportes en búsqueda de obtener las mejores características que ellos brindan. Luego de consultar la bibliografía se concluye que la mayoría de los sistemas gestores de reportes a nivel nacional fueron desarrollados en la UCI.

GEReport es una herramienta destinada al diseño, generación y configuración de los reportes relacionados con los datos históricos almacenados en una fuente de datos. Además, luego de contar con toda la información del reporte es posible exportarlo como imagen y en los formatos HTML, PDF y Excel. Para la interacción con las aplicaciones externas, el sistema implementa un servicio que expone los metadatos de los reportes para poder utilizarlos sin restricciones de lenguajes y plataformas. El sistema se creó sobre un entorno web y se desarrolló siguiendo lo establecido por el Proceso Unificado de Desarrollo (RUP), utiliza UML como lenguaje de modelado. Para la gestión de la base de datos se seleccionó *PostgreSQL*, en la implementación se utilizaron los lenguajes de programación PHP, con *CodeIgniter* 2.0 como marco de trabajo del lado del servidor y JavaScript con *Dojo Toolkit* 1.8 para el trabajo del lado del cliente. Se utiliza en la Universidad de Cienfuegos por analistas y programadores del Grupo de Estudios y Desarrollo de Ingeniería y Sistemas (GEDIS), perteneciente a la Facultad de Ingeniería (10).

GDR: Generador Dinámico de Reportes por sus siglas. Desarrollado en centro DATEC de la facultad 6 de la Universidad de las Ciencias Informáticas. Entre sus principales características se encuentran que es una aplicación para la web, las cual necesita de la instalación de un servidor de aplicaciones web: Apache, un sistema gestor de bases de datos: *PostgreSQL* y un servidor de gráficos, preferentemente el "*Rich Chap Server*". Su código fuente es el de la herramienta PHP *Reports* y el trabajo con grandes cantidades de volúmenes de datos es deficiente al no tener implementado una solución de paginado para el trabajo con los mismos. Es un sistema multiplataforma y muy empleado por los servicios internos de la institución. El lenguaje de programación que

se emplea para trabajar es PHP, y se cuenta con un grupo de desarrollo para generar plantillas en formatos *jrxml* para lograr una integración con la librería *jasperreports* y poder trabajar además con Java (11).

Olympia: Aplicación desarrollada sobre el *framework* Symfony y escrita en PHP. Con el objetivo de desarrollar una herramienta que permita la generación dinámica de reportes para el proyecto productivo alasGRATO de la Facultad 6 de la Universidad de las Ciencias Informáticas. Este generador cuenta con una serie de características importantes para construir y guardar reportes; además puede generar informes con información extraída de diversos gestores de bases de datos. Es multiplataforma y soporta imágenes, gráficas y sub-reportes, así como varios orígenes de datos. Proporciona a los usuarios, entre otras opciones, agilizar la toma de decisiones, generar reportes en varios formatos y con gran variedad de opciones en su diseño, al marcar una diferencia entre los reportes tradicionales y los reportes dinámicos, objetos de este producto.

Está compuesto por varias aplicaciones entre las que se encuentran el Visor de reportes, Diseñador de modelos y el Diseñador de reporte. Aunque permite abstraerse en parte de los conocimientos relacionados con los gestores de bases de datos, el usuario aún debe poseer conocimientos básicos de los mismos. Además, su entorno de trabajo está estructurado de forma que es difícil guiarse en la creación y generación de reportes. Las consultas que soporta la misma son las de SQL estándares, mediante el lenguaje de programación PHP, no presenta buen rendimiento dentro de las bases de datos Oracle y necesita de un servidor web (Apache) para poder entrar en funcionamiento (12).

Manage Engenier ServiceDesk: plataforma utilizada por el centro de soporte de la Universidad de las Ciencias Informáticas, para informar cualquier hecho o evento que sucede de manera inesperada y afecta el correcto funcionamiento de una aplicación informática. Es una plataforma desarrollada con código privativo y fue creada con fines de reportar incidencias mediante el envío de notificaciones al correo (13).

1.2.3 Conclusiones de soluciones similares

Los estudios realizados, permitieron adquirir información sobre los sistemas de gestión de reportes en el ámbito internacional y nacional. Mediante el análisis de las principales características de estos sistemas se llegó a la conclusión de que las herramientas desarrolladas no son de carácter general, sino que resuelvan solamente los problemas detectados internamente por los especialistas del equipo de desarrollo, específicamente incidencias de soporte técnico, por lo que no pueden ser utilizados. Además, no cumplen

con las peticiones de los clientes y adaptar cualquiera de estos sistemas a sus requerimientos sería más costoso en cuanto a tiempo y trabajo. A pesar de esto, es válido aclarar que algunos de estos sistemas poseen funcionalidades que se pudieran tener en cuenta en el desarrollo del sistema. Entre estas funcionalidades se citan: asignación de estados, notificación por correo electrónico y asignación de prioridades. Se necesita una aplicación específica que permita crear reportes y que no se dupliquen, para lograr que la comunidad pueda reportar por sí sola y que el trabajo del personal de mantenimiento sea revisar y dar solución a las afectaciones.

1.3 Metodologías de desarrollo

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental para guiar el desarrollo de un producto de software; pero los requisitos de un software a otro son tan variados y cambiantes que ha dado lugar a que exista una gran variedad de metodologías (14).

Entre las metodologías de desarrollo existen metodologías tradicionales centradas específicamente en el control del proceso. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos). Aplicar propuestas tradicionales obliga a forzar al cliente a que tome la mayoría de las decisiones al principio. Luego el coste de cambio de una decisión tomada puede llegar a ser muy elevado. Por otro lado, las metodologías ágiles son menos orientados al documento, exigiendo una cantidad pequeña de documentación para una tarea dada, son más bien orientadas al código.

Tabla 1. Diferencia entre los tipos de metodologías

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado

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 software	La arquitectura del software es esencial y se expresa mediante modelos

El equipo de desarrollo se centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, al promover el trabajo en equipo, quienes se preocupan por el aprendizaje de los desarrolladores y se propicia un buen clima de trabajo. El equipo de desarrollo está compuesto por dos integrantes, los cuales tienen una retroalimentación continua junto al cliente, existe comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. El presente proyectos se define con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Principales valores del desarrollo ágil (15):

- **El individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas**: el usuario es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo.
- **Desarrollar software que funciona más que conseguir una buena documentación**: la documentación debe ser corta y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que le van a servir para ponerse al día son: el propio código y la interacción con el equipo.
- **La colaboración con el cliente más que la negociación de un contrato:** las características particulares del desarrollo de software hacen que varios proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente

manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo

- **Responder a los cambios más que seguir estrictamente un plan**: la habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir.

Luego de un análisis de las metodologías de desarrollo y estudiado sus características, etapas de desarrollo y ventajas que posibilitan, se decide escoger las metodologías ágiles y dentro de ellas**:**

Extreme Programming (XP)

Una de las prácticas significativas que posee XP es que simplifica el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que no se necesiten, ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código (16).

Existen disimiles autores que definen el ciclo de vida de XP, definiendo sus fases, los autores de la presente investigación emplean la bibliografía de José Carlos Carvajal, donde define que está compuesto de seis fases: Exploración, donde los clientes escriben las historias de usuario (HU) describiendo las funcionalidades que serán añadidas al sistema, con actualizaciones regulares en caso de modificación; Planificación, donde se establece la prioridad de las HU y se realiza una estimación temporal para una primera entrega del proyecto; Iteraciones, donde se realizan iteraciones continuas decidiendo las historias de usuario a implementar, así como las pruebas funcionales ejecutadas al final con una retroalimentación continua; Producción, donde se realizan pruebas de rendimiento y funcionamiento que son necesarias antes de entregar el producto; al realizar una pequeña entrega; Mantenimiento, donde se actualiza el sistema una vez entregada la primera versión, el sistema debe estar en mantenimiento mientras existan iteraciones en la fase de producción; Cierre del proyecto, es donde se realiza la entrega final al cumplir las necesidades del cliente y se documentan todos los pasos (17).

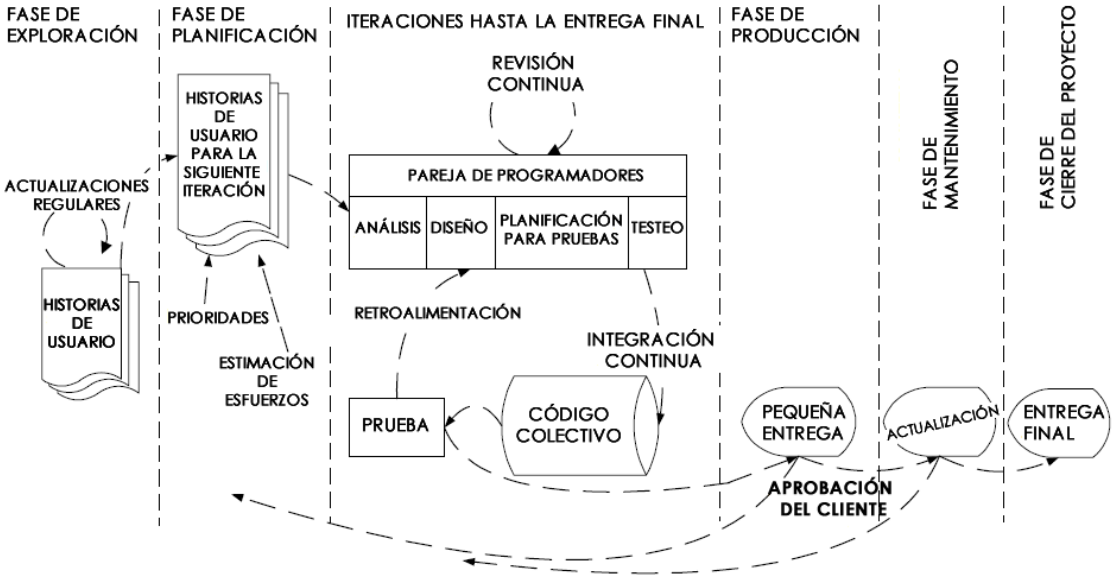


Ilustración 1. Fases de XP

1.4 Lenguajes de desarrollo

Un usuario cualquiera, usa un navegador y escribe básicamente la URL (*Uniform Resource Locator*) de una página, puede hacer una petición a través del protocolo HTTP (*Hipertext Transfer Protocol*) a un servidor web, este atiende la petición, procesa la misma y envía una respuesta al usuario a través del navegador que muestra la respuesta en formato HTML. La arquitectura cliente/servidor fue una de las razones por la cual se eligió hacer un sistema basado en la web.

A continuación se detallan las herramientas y tecnologías utilizadas para el desarrollo del presente trabajo. Estas fueron seleccionadas de acuerdo con las ventajas que proporcionan, teniendo en cuenta la tendencia actual y las novedades en este campo.

1.4.1 Tecnología del lado del servidor

Consiste en procesar una petición de un usuario mediante la interpretación de un script en el servidor web para generar una respuesta que cualquier cliente puede interpretar. A continuación se hace una descripción de varios lenguajes:

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía *Sun Microsystems* con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Una de las principales características es que es un lenguaje independiente de la plataforma, es decir que al

hacer un programa en Java podrá funcionar en cualquier ordenador del mercado, siempre y cuando se cuente con la máquina virtual de java (18).

ASP (*Active Server Pages*) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web y utiliza el lenguaje *Visual Basic Script* o *Jscript* (Javascript de Microsoft).

Con las ASP se pueden realizar varias aplicaciones distintas. Permite el acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. Actualmente se ha presentado la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro (19).

PHP

Es el acrónimo de *Hypertext Preprocessor* o preprocesador de hipertexto. Es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y extensa documentación. Fue creado originalmente en 1994 por Rasmus Lerdorf. PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo (21).

A partir de información disponible en el sitio web *Digital Learning,* basándose en el análisis de consultas que se realizan en los motores de búsqueda más utilizados, además de las actividad en comunidades de programadores, la demanda de conocimientos y experiencia de programación en el mercado de trabajo, el volumen de libros vendidos y encuestas realizadas en Internet entre programadores (20); se concluye optar por PHP como tecnología de servidor. Se considera un lenguaje flexible, potente y de alto rendimiento, además ha atraído el interés de múltiples sitios con gran demanda de tráfico (21).

1.4.2 Tecnologías del lado del cliente

Tecnologías ejecutadas en el navegador del usuario. Son las páginas dinámicas que se procesan en el cliente, en las cuales el navegador soporta toda la carga del procesamiento de los efectos y sus funcionalidades.

JavaScript

JavaScript es un lenguaje de programación interpretado usado para crear páginas dinámicas, las variables no necesitan ser introducidas antes de su uso y los tipos de variables se resuelven dinámicamente durante su ejecución. El código JavaScript que se encuentra dentro de las páginas web puede ser interpretado por todos los navegadores, porque soporta la carga de procesamiento (22).

Características:

- Es un lenguaje de alto nivel, multiplataforma y no necesita compilación.
- Está basado en objetos.
- Admite la programación estructurada.
- Maneja la mayoría de los eventos que se pueden producir sobre la página web.

CSS

Las hojas de estilo en cascada, conocidos como CSS se definen según Rivera como "…un lenguaje de hojas de estilo creado para controlar la presentación de los documentos" (23). Se decide seleccionar CSS para mejorar el diseño visual de la solución si fuera necesario, separar contenidos de presentación y realizar cambios a múltiples elementos dentro del código, lo que agiliza el proceso de cambios.

HTML5

HTML es un lenguaje de etiquetas (también llamado lenguaje de marcado) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. De hecho, las letras "ML" de la sigla HTML significan *"markup language"*, que es como se denominan en inglés a los lenguajes de marcado. La principal ventaja de los lenguajes de etiquetas es que son muy sencillos de leer y escribir por parte de las personas y de los sistemas electrónicos. La principal desventaja es que pueden aumentar el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos (24).

Durante los últimos meses se ha generado gran expectativa alrededor de un sin número de tecnologías que trae el nuevo lenguaje de la web, no solo por parte de los

desarrolladores, sino también de usuarios, debido a que las nuevas especificaciones del HTML mejoran sensiblemente la experiencia de la navegación.

Con HTML5 se pretende facilitar el desarrollo aplicaciones web, las cuales siguen unos parámetros de compatibilidad para usuarios de ordenadores de escritorio. A continuación se mencionan algunas características:

- HTML5 Proporciona soporte para vídeo y audio: Ayuda a acabar con el Flash Player o cualquier otro *plugin* externo para poder reproducir audio y video. HTML5 lo reproduce nativamente, de una forma muy simple y efectiva.
- Obtener un código limpio: Una de las principales características de HTML5 es su simpleza en el código. Este lenguaje de marcado permite escribir un código conciso que ayuda a distinguir el significado del código, del estilo y contenido. Además, HTML5 permite organizar efectivamente tus CSS.
- Permite una mejor interacción: HTML 5 ofrece una solución rápida para crear un sitio web dinámico que proporcione a los usuarios una interfaz intuitiva con la máxima interacción en este sentido, con una etiqueta especial y potente que permite hacer su sitio más interactivo.

Compatibilidad multi-plataforma: No sólo en los distintos navegadores actuales y populares como Chrome, Firefox, Safari y Opera 9, sino también algunos navegadores antiguos como Internet Explorer 6 (25).

Jquery

JQuery es un nuevo tipo de biblioteca o marco de trabajo de JavaScript que permite simplificar la manera de interactuar con las vistas en una aplicación web, permite manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX al sistema, la cual permite el intercambio asíncrono de datos entre cliente y servidor de manera sencilla. *JQuery* al igual que otras librerías, ofrece funcionalidades basadas en JavaScript con las que se logran grandes resultados en menos tiempo y espacio. La gran ventaja de *JQuery* es que permite cambiar el contenido de la página web sin necesidad de recargarla y utiliza DOM y AJAX de manera sencilla, así como manipular el estilo CSS (26).

1.5 Entorno de desarrollo integrado

IDE JetBrains PhpStorm 8.0

PhpStorm es un IDE de programación desarrollado por *JetBrains* escrito en Java. Es uno de los entornos de programación completos de la actualidad para las tareas de un desarrollador enfocado en PHP. Actualmente es compatible con Sistemas Operativos Windows, Linux y Mac OS X. Algo que destaca en *PhpStorm* es la ejecución del código en la misma interfaz del IDE. Así como también la interpretación y visualización inmediata de código PHP hasta en cinco navegadores web (27).

Algunas características principales de este editor de código:

- Permite la gestión de proyectos fácilmente.
- Proporciona un fácil autocompletado de código.
- Soporta el trabajo con PHP 5.5.
- Sintaxis abreviada.

1.6 Framework

Mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, un *framework* simplifica el desarrollo de una aplicación. Además, proporciona estructura al código fuente, potencia al desarrollador a crear código legible y fácil de mantener. Permite la programación de aplicaciones y encapsula operaciones complejas en instrucciones sencillas.

Rails es un *framework* de desarrollo de aplicaciones web escrito en el lenguaje de programación Ruby. Está diseñado para hacer que la programación de aplicaciones web sea cómoda, haciendo supuestos sobre lo que cada desarrollador necesita para comenzar (28).

Zend es un *framework* de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Implementa código totalmente orientado a objetos. Presenta una arquitectura débilmente acoplada permitiendo a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). Ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de utilizar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos (29).

Para el desarrollo del presente trabajo se selecciona **Symfony** por ser un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones WEB, gracias a sus características.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios WEB de comercio electrónico de primer nivel. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación WEB. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación WEB compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos complejos.
- Basado en la premisa de convenir en vez de configurar, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la WEB.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo. (30)

Symfony se hace necesario para que las peticiones entre el cliente y el servidor se realicen de manera asíncrona para mejorar el tiempo de envío y respuesta.

Bootmetro 1.0

Para aplicar estilo a la web se seleccionó **BootMetro,** un *framework* basado en *Twitter Bootstrap* que permite construir páginas web con la interfaz metro, de manera rápida y flexible. Hay que tener en cuenta conocimientos de HTML, CSS y *Javascript* para ponerla en marcha.

Entre sus principales características ofrece:

- **Creación de *Hubs***: Estos permiten darle enfoque al contenido del sitio y reduce las distracciones.
- **Acciones**: La *ApplicationBar* permite agregar iconos en el menú contextual para mostrar las diferentes acciones.
- **Navegación**: Implementa una navegación pura.
- **Paneles**: Puedes crear paneles para mostrar opciones de configuración.
- **IconMoon**: Es un conjunto de iconos de fuente que puedes utilizar en tu sitio con una variedad de colores y dimensiones. (31)

1.7 Herramientas

Yaml

YAML es un formato para guardar objetos de datos con estructura de árbol. Este lenguaje es muy legible para las personas, más legible que JSON y sobretodo que XML. Normalmente se utiliza para definir archivos de configuración, aunque también es posible serializar objetos, es decir, escribir la estructura de un objeto en modo cadena de texto para posteriormente poderlo recuperar (32).

YAML es un formato ideal para los archivos de configuración, los archivos *YAML* son tan expresivos como los archivos *XML* (33).

Twig

Las plantillas en Twig son poco trabajosas de hacer y resultan muy intuitivas; Twig, seguro motor de plantillas para PHP, añade útil funcionalidad a los entornos de plantillas, es flexible, rápido e implementa un novedoso mecanismo de herencia de plantillas y cachea en auténticas clases PHP todo el contenido de las mismas, para acelerar el rendimiento de la aplicación.

Las características clave son:

- Rápido: Twig compila las plantillas hasta código PHP regular optimizado. El costo general en comparación con código PHP regular se ha reducido al mínimo.
- Seguro: Twig tiene un modo de recinto de seguridad para evaluar el código de plantilla que no es confiable. Esto te permite utilizar Twig como un lenguaje de plantillas para aplicaciones donde los usuarios pueden modificar el diseño de la plantilla.

- Flexible: Twig es alimentado por flexibles analizadores léxico y sintáctico. Esto permite al desarrollador definir sus propias etiquetas y filtros personalizados, y crear su propio DSL. (34)

Visual Paradigm

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Ordenador, del inglés *Computer Aided Software Engineering*) que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo de desarrollo de un software, desde la fase de análisis hasta el despliegue del mismo. Visual Paradigm está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Además, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo a través de un lenguaje gráfico (35).

Highcharts

Highcharts es una biblioteca de gráficos escritos en JavaScript que ofrece una manera factible de agregar gráficos interactivos al sitio web o aplicación web. Actualmente soporta líneas, ranuras, área, columna, barras, pastel, circulares, de dispersión, medidores angulares, caja de trama, las barras de error, embudo, cascada y tipos de gráficos polares. Tiene licencias libres para un sitio web personal, una escuela o una organización sin fines de lucro. Una de las características clave de Highcharts es que bajo cualquier licencia, libre o no, se permite descargar el código fuente y hacer ediciones. Esto permite modificaciones personales y una gran flexibilidad. Con el módulo de exportación activado, los usuarios pueden exportar el gráfico a formato SVG, PNG, JPG, PDF o imprimir el gráfico directamente desde la página web (36).

Ajax

Es la técnica de desarrollo Web para aplicaciones interactivas más utilizada actualmente. Engloba a todo un grupo de tecnologías (XHTML, JavaScript, CSS, API y DOM) y mantiene una comunicación asíncrona con el servidor en segundo plano, lo que permite realizar continuos cambios sin necesidad de recargar las páginas. Además, añadiéndole la potencia de PHP, podrá crear fácilmente las aplicaciones profesionales. Con el uso de ajax en las aplicaciones, reduce al mínimo el código JavaScript. Permite la comunicación entre cliente y servidor, para crear aplicaciones web verdaderamente

flexibles, así como para mejorar notablemente la experiencia de usuario a través de tareas como la validación de formularios a tiempo real (37).

Valentina Studio

Permite el manejo de base de datos y una característica de gran utilidad es la de poder clonar base de datos. Permite crear diagramas, tablas, ejecutar los últimos SQL's y además soporta Postgress.

Entre sus principales características cabe destacar las siguientes:

- Capacidad para crear, editar, importar y administrar bases de datos MySQL, PostgreSQL y SQLite además de la base de datos propietaria Valentina DB
- Capacidad para gestionar múltiples bases de datos y múltiples operaciones en vistas de pestaña.
- Crear, borrar, modificar y ver tablas, campos y métodos.
- Editor de esquema con vistas de árbol y de columna.
- Editor de datos, navegación, ordenación y filtrado de registros, exportar registros hallados y previsualización de imágenes.
- Editor SQL con autocompletado, coloreado de sintaxis, modo de consola, y capacidad de guardar las consultas recientes o marcadas como favoritas.
- Añadir y borrar usuarios en los servidores de bases de datos, registrar o desregistrar bases de datos, ejecutar diagnósticos, crear, modificar y ver eventos planificados.
- Utilidades para diagnosticar, comprimir, desfragmentar y volcar datos. (38)

1.8 Sistemas gestores de base de datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, al garantizar la seguridad e integridad de los mismos (39). Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos. En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Un SGBD también debe

encargase de cumplir las reglas de integridad y redundancias. Debe realizar copias de seguridad y de recuperación de datos (40).

1.8.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto potente en el mercado. Utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (41).

A continuación se muestra un gráfico que ilustra de manera general los componentes más importantes en un sistema PostgreSQL:

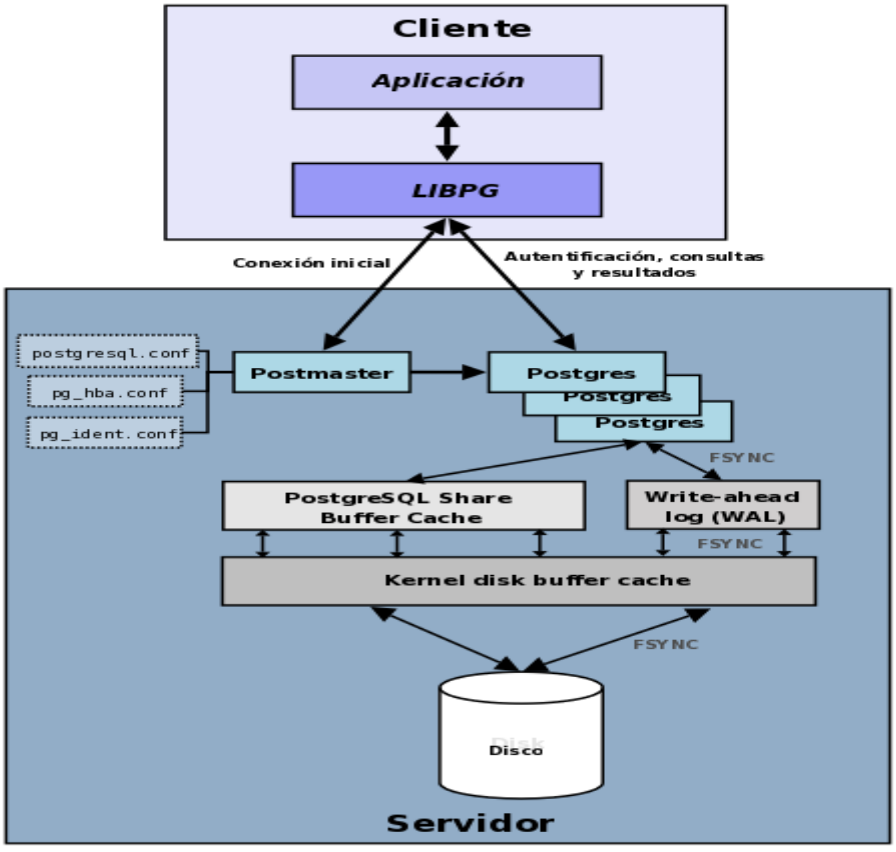


Ilustración 2. Componentes principales de un sistema PostgreSQL

Características:

- Soporta distintos tipos de datos.
- Incorpora una estructura de datos *array*.

- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Ventajas:

- Instalación ilimitada.
- Mejor soporte que los proveedores comerciales.
- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad legendarias.
- Extensible.
- Multiplataforma.
- Herramientas gráficas de diseño y administración de bases de datos.

Para gestionar el gestor de bases de datos PostgreSQL, se trabaja con **PgAdmin III** con licencia *Open Source*. Está escrita en C++ y usa la librería gráfica multiplataforma *wxWidgets*, lo que permite que se pueda usar en Linux, *FreeBSD*, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como *Pervasive Postgres*, *EnterpriseDB*, *Mammoth Replicator* y SRA *PowerGres*.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y soporte para el motor de replicación *Slony-I*. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix *Domain* Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.

Doctrine

Una de las tareas comunes y a la vez complejas de la programación web, consiste en la persistencia de la información en una base de datos. Afortunadamente, Symfony incluye la librería Doctrine, que proporciona herramientas para simplificar el acceso y manejo de la información de la base de datos.

Trabajar con Doctrine, en vez de trabajar con filas y tablas, permite guardar y obtener objetos enteros a partir de la información de la base de datos (42). El truco para que esto funcione consiste en mapear una clase PHP a una tabla de la base de datos y después, mapear las propiedades de la clase PHP a las columnas de esa tabla:

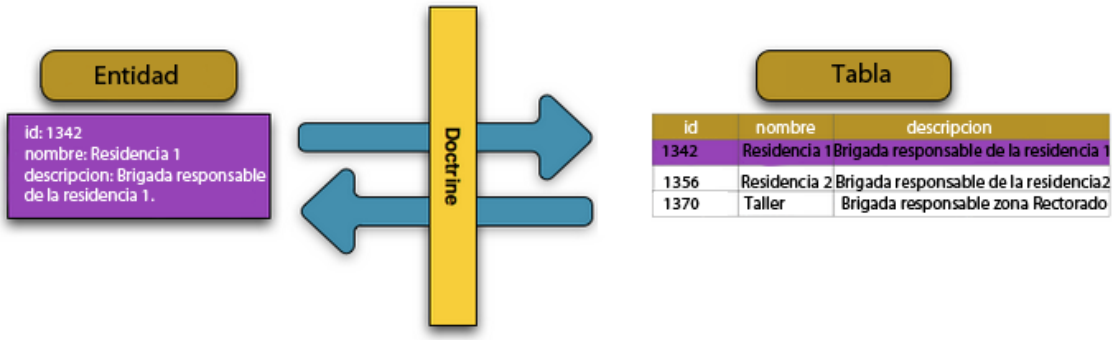


Ilustración 3. Mapeando objeto PHP a una tabla

Doctrine simplifica al máximo este proceso, de manera que sólo tienes que añadir algunos metadatos a la clase PHP para configurar. Estos metadatos se pueden configurar en archivos YAML, XML o directamente mediante anotaciones en la propia clase PHP.

1.9 Servidores web

Apache 2.2.4

Para el desarrollo de la propuesta de solución se seleccionó el servidor web Apache, por ser una herramienta libre y brindar más seguridad que IIS. Apache funciona en una multitud de SO, mientras que IIS solo puede ser ejecutado en Windows. Además IIS es un software propiedad de Microsoft, lo que dificulta la obtención de la licencia para su uso, mientras que Apache es un software libre y facilita su uso a toda la comunidad. Apache es el servidor web hecho por excelencia, su configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es un servidor web de código libre por lo que permite realizar cualquier cambio en el código fuente (43).

Principales características de Apache:

- Corre en una multitud de Sistemas Operativos (SO), lo que lo hace prácticamente universal.
- Posee una tecnología gratuita de código fuente abierto.
- Servidor altamente configurable de diseño modular.
- Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Conclusiones parciales

La definición de los principales conceptos a tratar durante el desarrollo del presente trabajo permitió un mayor entendimiento durante la investigación. Entre los sistemas estudiados, Mantis posee características y utilización de algoritmos que pudieran ser tomados en cuenta para el desarrollo del presente trabajo. El estudio de las herramientas y tecnologías existentes en el mundo actual, permitió definir cuáles utilizar para implementar una aplicación informática que automatice los procesos de gestión de incidencias sobre mantenimiento constructivo en las áreas de la UCI y así contribuir al seguimiento de los reportes hechos por la comunidad universitaria.

Capítulo 2: Características del sistema

En este capítulo se describen las principales características del sistema a desarrollar. Se identifican los procesos que intervienen en el sistema con el objetivo de desarrollar un correcto diseño e implementación de los mismos. La metodología XP propone técnicas importantes de diseño, en la fase de planeación se tienen las historias de usuario, que facilitan la comprensión del sistema a desarrollar. Se definen además los requerimientos no funcionales y se precisan los actores que intervienen en el sistema.

2.1 Propuesta del Sistema

Para la solución del problema se propone una aplicación web, la cual será diseñada para la dirección de mantenimiento de la UCI y permitirá gestionar las incidencias de reportes. El sistema proporcionará una interfaz como forma de presentación al usuario desde la cual se podrá autenticar. Una vez autenticado, la aplicación ofrecerá información general sobre el sistema y le brindará de acuerdo a su rol, una serie de funciones. El usuario autenticado conseguirá realizar reportes mediante una interfaz que contará con un formulario a llenar, una vez reportado recibirá una notificación al correo con los datos de dicho reporte. El sistema también brindará al técnico general la opción de realizar reportes y llevar un control de los reportes realizados según una serie de parámetros como son: especialidad, local, área, instalación, afectación, entre otros. Además, el sistema contará con la posibilidad de conciliar y generar un documento excel con el cual, los directivos de la oficina de mantenimiento constructivo UCI, obtendrán estadísticas sobre los reportes realizados, según el estado, ya sea solucionado, pendiente o sin solución. Además, lograrán obtener estadísticas mediante gráficos, generados por un rango de fecha.

2.2 Usuarios relacionados con el sistema

Se definen como usuarios relacionados al sistema todos aquellos que realizan una función o interactúan con él de una forma u otra. A continuación se describe la definición de los siguientes usuarios.

Tabla 2. Usuarios involucrados en el sistema

Involucrados en el sistema	Descripción
Usuario	Todos los usuarios deben autenticarse para acceder a las funcionalidades del sistema.

	Encargado de realizar el reporte y tener acceso a la información de sus reportes.
Técnico General	Persona que tendrá acceso al sistema como usuario estándar y podrá chequear los reportes y posibilitar dar solución o una respuesta.
Administrador	Los administradores disponen de posibilidades ilimitadas para ejecutar todas las funciones administrativas del sistema.
Jefe Brigada	Persona que tendrá acceso al sistema para chequear los reportes y las tareas que están relacionados con su brigada.
Directivo	Persona que tendrá acceso al sistema para chequear el estado de los reportes y obtener estadísticas.

2.3 Requisitos del Sistema

Para el desarrollo del presente trabajo se definen características que debe poseer el sistema para cumplir con los objetivos trazados y satisfacer las necesidades del cliente.

2.3.1 Requisitos funcionales

Estos requisitos definen el comportamiento interno del sistema, describiendo la manera en que éste debe reaccionar ante entradas particulares y cómo se debe comportar en situaciones específicas (44).

- Autenticar usuario: el usuario debe autenticarse para poder acceder a las funcionalidades del sistema.
- Gestionar usuarios: permitir adicionar usuarios y mostrar datos del mismo según LDAP-UCI.
- Gestionar roles: permitir mostrar el rol de cada usuario.
- Gestionar reportes: permitir adicionar, mostrar, editar y eliminar reportes.
- Conciliar: permitir exportar a documento excel los datos referentes todos los reportes realizados hasta la fecha actual.
- Exportar: permitir exportar los reportes a documento excel.
- Gestionar área: permitir adicionar, mostrar, editar y eliminar área.
- Gestionar brigada: permitir adicionar, mostrar, editar y eliminar brigada.
- Gestionar recurso: permitir adicionar, mostrar, editar y eliminar recurso.

- Gestionar afectación: permitir adicionar, mostrar, editar y eliminar afectación.
- Gestionar especialidad: permitir adicionar, mostrar, editar y eliminar especialidad.
- Gestionar local: permitir adicionar, mostrar, editar y eliminar local.
- Gestionar instalación: permitir adicionar, mostrar, editar y eliminar instalación.
- Importar excel: permitir importar reportes de un documento excel.
- Graficar datos estadísticos: permitir mostrar mediante un gráfico los reportes realizados en un rango determinado.
- Asignar rol: permite asignar rol a los usuarios.
- Notificar por correo: envía un correo al usuario que realiza un reporte con los datos ingresados.
- Filtrar reporte: permitir filtrar reportes mediante varios criterios.
- Filtrar especialidad: permitir filtrar especialidades mediante varios criterios.
- Filtrar afectación: permitir filtrar afectaciones mediante varios criterios.
- Filtrar recurso: permitir filtrar recursos mediante varios criterios.
- Filtrar brigada: permitir filtrar brigadas mediante varios criterios.
- Filtrar área: permitir filtrar áreas mediante varios criterios.
- Filtrar instalación: permitir filtrar instalaciones mediante varios criterios.
- Filtrar local: permitir filtrar locales mediante varios criterios.
- Exportar gráfico: permitir exportar en varios formatos.

2.3.2 Aspectos no funcionales

Para lograr un correcto funcionamiento del sistema se deben tener en cuenta un conjunto de aspectos no funcionales; son cualidades o propiedades que el producto debe tener (45). A continuación se listan algunos de estos:

Requisitos de seguridad: El acceso a la aplicación se realizará según el rol de cada usuario.

Requisitos de usabilidad: Claridad y buena organización de la aplicación, permitiendo la interpretación correcta e inequívoca de la información. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- Conocimientos básicos relativos al uso de una computadora.
- Conocimientos básicos del sistema operativo Windows y Linux.
- Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

Requisitos de soporte: El sistema permitirá la agregación o modificación de nuevas funcionalidades siempre que sea necesario, asegurar su extensibilidad y mejores prestaciones.

Requisito de interfaz: Su funcionamiento debe ser intuitivo y requerir de información mínima. Debe ser amigable para los usuarios.

Requisitos de configuración y seguridad: La aplicación requiere validación de uso por el servidor central. Al no obtener la validación, no podrá iniciar. La validación permite al servidor determinar, los datos que enviará a la aplicación, dependiendo del nivel de acceso inscrito al usuario.

Requisitos legales: Una vez terminada la aplicación web debe ser sometida a una evaluación y certificación por parte del cliente del producto.

2.4 Historias de usuario

Las historias de usuario (HU) son la forma en que se especifican en XP los requisitos del sistema, las mismas no deben ser descritas en más de tres líneas e idealmente es el cliente quien las redacta y prioriza por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica (46). Luego de esto se le suma un tiempo estimado de desarrollo que lo define el desarrollador. Deben describirse con claridad, de manera sencilla y concreta para su mejor entendimiento (47).

A continuación se muestra una HU con sus principales elementos y la explicación de cada uno de ellos:

Tabla 3. Campos de una HU

Historia de Usuario	
Número Usuario	Nombre
Prioridad en el negocio	Puntos Estimados
Riesgo de desarrollo	Iteración asignada
Descripción	
Observaciones	

Campos de las historias de usuario:

Número: Es el número que se asigna a la HU.

Usuario: El usuario que interactúa o protagoniza la HU.

Nombre: Atributo que contiene el nombre de la HU.

Prioridad en el negocio: Prioridad que se le asigna a la historia de usuario en el negocio. Si es importante debe ser implementada lo antes posible, esta se clasifica en:

- Alta: Se le otorga a las historias de usuarios que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control integral del sistema.
- Media: Se le otorga a las historias de usuarios que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- Baja: Se le otorga a las historias de usuarios que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Riesgo de desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna un valor que puede ser:

- Alto: Cuando en la implementación de las historias de usuarios se considera la posible existencia de errores que lleven a la inoperatividad del código.
- Medio: Cuando pueden aparecer errores en la implementación de las historias de usuarios que puedan retrasar la entrega de la versión.
- Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad, sin que traigan perjuicios para el desarrollo del proyecto. (48)

Puntos Estimados: Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como cinco días hábiles. Cuando el valor de dicho atributo es de 0.5 equivale a dos días y medio de trabajo, lo que se traduce en veinte horas.

Iteración asignada: Iteración en la que se implementará la HU.

Descripción: Describe lo que realizará la HU.

Observaciones: Propone información adicional para hacer entendible la HU.

El equipo de desarrollo y el cliente trabajaron en conjunto para definir las HU. A continuación se muestran las HU de los requisitos más significativos de la aplicación:

Tabla 4. HU_1

Historia de Usuario	
Número: 1 Usuario: Usuario, Técnico General, Administrador, Jefe Brigada, Directivo	Nombre: Autenticar usuario.
Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 1
Descripción: El sistema muestra la vista y luego el usuario entra sus datos (usuario y contraseña) en el formulario, el sistema verifica que los datos entrados estén correctos según el dominio UCI y se le permite el acceso según el rol.	
Observaciones: Según el rol que desempeñe el usuario será la interfaz que se mostrará.	

Tabla 5. HU_2

Historia de Usuario	
Número: 2 Usuario: Usuario, Técnico General, Administrador, Jefe Brigada, Directivo	Nombre: Gestionar usuarios.
Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 1
Descripción: Permite adicionar a la base de datos el usuario que se autentica la primera vez que accede al sistema y mostrar los datos según el LDAP UCI.	
Observaciones: Una vez que el usuario se autentica, podrá ver sus datos según está registrado en el sistema LDAP UCI.	

Tabla 6. HU_3

Historia de Usuario	
Número: 3 Usuario: Administrador, Técnico General, Usuario, Jefe Brigada, Directivo	Nombre: Gestionar roles.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Alto	Iteración asignada: 1

Descripción: Permite mostrar los roles asignados a los usuarios.
Observaciones: Una vez autenticado cualquier usuario podrá ver el rol que le fue asignado.

Tabla 7. HU_4

Historia de Usuario	
Número: 4 Usuario: Técnico General, Administrador, Jefe Brigada	Nombre: Gestionar reportes.
Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 1
Descripción: Permite crear, mostrar, editar y eliminar reportes, mientras que el usuario autenticado solo tendrá acceso a los reportes realizados por él.	
Observaciones: El Técnico General o el Administrador, debe estar autenticado para acceder a esta funcionalidad. El Jefe de Brigada una vez autenticado, tendrá acceso a ver y editar, los reportes asignados a su brigada por el Técnico General.	

Tabla 8. HU_5

Historia de Usuario	
Número: 5 Usuario: Técnico General, Administrador	Nombre: Conciliar
Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 1
Descripción: Muestra un documento excel con los siguientes datos pertenecientes a las brigadas Residencia 1, Residencia 2, Residencia profesores, Docente, Guardia, Carpintería metálica, Taller y a los reportes de la afectación Filtración y de la especialidad Clima (Total de reportes pendientes del pasado año, total de reportes del presente año, mes y semana, total de reportes resueltos del presente año, mes y semana, total de reportes pendientes del presente año, mes y semana, total de reportes del pasado año resueltos esta semana y total de reportes del presente año resueltos esta semana), así como porcientos calculados automáticamente y otros datos de interés.	

Observaciones: El Técnico General o el Administrador, debe estar autenticado para acceder a esta funcionalidad.

Las restantes historias de usuario se encuentran definidas en el Anexo # 2 del documento.

2.5 Estimación de esfuerzos por HU

Para el desarrollo de la aplicación propuesta según la fase de planificación, se realiza una estimación del esfuerzo por cada una de las HU identificadas:

Tabla 9. Estimación de esfuerzo por HU

No	Historia de Usuario	Estimación (semanas)
1	Autenticar usuario	1.5
2	Gestionar usuarios	1.5
3	Gestionar roles	1
4	Gestionar reportes	1.5
5	Conciliar	1.5
6	Exportar	1.5
7	Gestionar área	1
8	Gestionar brigada	1
9	Gestionar recurso	1
10	Gestionar afectación	1
11	Gestionar especialidad	1
12	Gestionar local	1
13	Gestionar instalación	1
14	Importar excel	0.5
15	Graficar datos estadísticos	0.5
16	Asignar rol	0.5
17	Notificar por correo	1.5
18	Filtrar reporte	0.5

19	Filtrar especialidad	0.5
20	Filtrar afectación	0.5
21	Filtrar recurso	0.5
22	Filtrar brigada	0.5
23	Filtrar área	0.5
24	Filtrar instalación	0.5
25	Filtrar local	0.5
26	Exportar gráfico	0.5

2.6 Plan de iteraciones

Como lo plantea el ciclo de vida de XP se crea un plan de duración de las iteraciones para una mayor organización del trabajo; donde se especifican cuáles son las HU definidas por el cliente que serán implementadas en cada iteración del sistema. Las funcionalidades son planificadas al generar, en cada una, entregables funcionales que implementa las historias de usuario asignadas a la iteración. De esta manera se garantiza que el cliente tenga un prototipo del sistema con un porciento de funcionalidades listas para ser probadas. Además, permite tomar decisiones en cuanto a cambios que puedan requerirse para mantener el camino correcto rumbo a los objetivos propuestos (49). A continuación se reflejan cómo quedaron definidas las iteraciones:

Iteración 1: Se implementan las historias de usuario con mayor prioridad para el cliente, para así obtener una primera versión del producto y lograr que el sistema obtenga las primeras funcionalidades, relacionadas con las historias de usuarios 1, 2, 3, 4, 5.

Iteración 2: En esta iteración se implementan las demás funcionalidades con prioridad alta y media, relacionadas con las historias de usuarios 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18. Una vez concluida esta iteración en conjunto con la primera, serán mostradas al cliente para si este desea realizar cambios según lo previsto anteriormente.

Iteración 3: Durante esta iteración serán implementadas las restantes funcionalidades de prioridad baja, relacionadas con las historias de usuario 19, 20, 21, 22, 23, 24, 25, 26. En ese instante el sistema será puesto a prueba por un período de tiempo, para evaluar el desempeño de la aplicación y la satisfacción del cliente.

Tabla 10. Plan de duración de las iteraciones

Iteración	Orden de las HU	Duración total (semanas)
1	Autenticar usuario Gestionar usuarios Gestionar roles Gestionar reportes Conciliar	7
2	Exportar Gestionar local Gestionar instalación Gestionar área Gestionar brigada Gestionar recurso Gestionar afectación Gestionar especialidad Importar excel Graficar datos estadísticos Asignar rol Notificar por correo Filtrar reporte	12
3	Filtrar especialidad Filtrar afectación Filtrar recurso Filtrar brigada Filtrar área Filtrar instalación Filtrar local Exportar gráfico	4

2.7 Plan de entregas

En el plan de entregas se realiza un cronograma de entregas donde el cliente establece la prioridad de cada HU, cuáles serán agrupadas para conformar una entrega y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (clientes y desarrolladores). Este plan se ajusta a las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la

fase de implementación (50). La finalidad de este plan es reflejar la duración de cada iteración, lo que permite tener una idea aproximada del tiempo que durará la confección del sistema.

Tabla 11. Plan de entrega

Historia de usuario	1era iteración	2da iteración	3ra iteración
Cantidad de UH	5	13	8
Fecha de entrega	19/12/2015	12/03/2015	15/04/2015

2.8 Prototipo no funcional de interfaz de usuario

Una vista preliminar del sistema a desarrollar se obtiene a partir de un prototipo de interfaz, el cual puede ser modificable según las características definidas de la propuesta de solución. A continuación se muestra la interfaz donde cualquier usuario, una vez autenticado, puede realizar el reporte de la incidencia. Esta propuesta puede cambiar en cualquier momento, según los criterios del cliente.

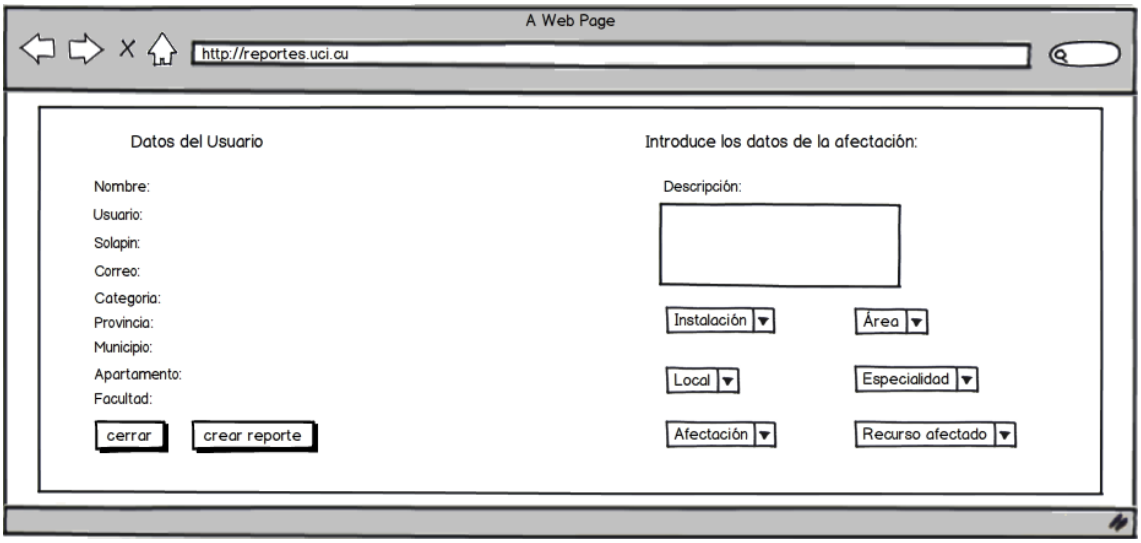


Ilustración 4. Prototipo no funcional

2.9 Tarjetas contenido responsabilidad colaboración

Las tarjetas contenido responsabilidad colaboración (CRC) constituyen una forma simple de organizar las clases notables para las funcionalidades del sistema, con el objetivo de desarrollar una representación organizada de las clases. Para el diseño de aplicaciones, la metodología XP no requiere la presentación del sistema mediante

diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración) (50).

Un modelo CRC es una colección de tarjetas índices estándar que representan clases, donde las responsabilidades definidas son todos los servicios que un objeto puede realizar y representan la parte pública de los objetos, mientras que los colaboradores representan peticiones de un cliente a un servidor para cumplir la responsabilidad del cliente. Se diseña una tarjeta CRC por cada una de las funcionalidades directas al negocio, es decir aquellas que fueron desarrolladas desde la raíz. Obteniéndose un diseño simple y sin implementar características que no son necesarias. A continuación se muestran las tarjetas CRC referentes a las HU de la primera iteración:

Tabla 12. Tarjeta CRC_1

Autenticar usuario	
Responsabilidades	**Colaboraciones**
fetchData()	Rol, rol

Tabla 13. Tarjeta CRC_2

Gestionar usuario	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request) showAction($id)	

Tabla 14. Tarjeta CRC_3

Gestionar rol	
Responsabilidades	**Colaboraciones**
indexAction($id)	

Tabla 15. Tarjeta CRC_4

Gestionar reportes

Responsabilidades	Colaboraciones
indexAction() indexTecnicoAction() createAction(Request $request) tecnicoCreateAction(Request $request) createCreateForm(Reporte $entity) newAction() showAction($id) editAction($id) createEditForm(Reporte $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction()	Especialidad, especialidad, Afectacion, afectacion, Local, local, Usuario, usuario, Brigada, brigada, Recurso, recurso

Tabla 16. Tarjeta CRC_5

Conciliar	
Responsabilidades	**Colaboraciones**
conciliarAction()	Reporte

Las tarjetas CRC restantes se encuentran definidas en el Anexo # 3 del documento.

2.10 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular permitiendo una solución a un problema en un contexto determinado. Cada patrón describe un problema que ocurre una y otra vez en el entorno, describe también el núcleo de su solución, de forma que puede utilizarse varias veces sin repetirse (51). En

el diseño de la aplicación propuesta se utilizan los Patrones Generales de Software para Asignar Responsabilidades (GRASP) y los GoF (en inglés *Gang of Four*).

A continuación se pone de manifiesto el uso de patrones GRASP utilizados:

- Experto: consiste fundamentalmente en asignar las responsabilidades a aquellos objetos o clases que cuenta con la información necesaria para cumplir la responsabilidad. Esto se evidencia en el getConciliado(): método que devuelve si el reporte ya fue conciliado o no, de la entidad Reporte.
- Creador: Consiste en determinar quién es el encargado o debería tener la responsabilidad de crear un determinado objeto. Esto se evidencia en la clase RolController, encargada de crear roles.
- Alta Cohesión: caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme; es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Su uso se evidencia en las clases ReporteController, encargada de tener todos los métodos relacionados con reportes, Reporte, encargada de los métodos relacionados con el manejo de la información de un reporte y ReporteRepository, encargada de los métodos de obtención de los reportes de la BD.
- Bajo Acoplamiento: cada clase en el sistema depende solo de las clases necesarias para su implementación. No existe una sobrecarga de dependencia entre las clases. Esto se evidencia en clases que no dependen de otras: Rol, Area, Especialidad y clases con dependencias: Usuario que depende de Rol, Local de Instalacion, Instalacion de Area, Afectacion de Especialidad y Brigada de Area y Usuario. (52)

A continuación se pone de manifiesto el uso de patrones GoF utilizados:

- Singleton: Este patrón garantiza la creación de un mecanismo de acceso global a una instancia y la existencia de una única instancia para una clase. El uso de este patrón se evidencia en el Anexo # 4.
- Decorator: Se evidencia en el uso del mecanismo de herencia de plantilla de Twig. Se muestra su funcionamiento en la Ilustración # 6. (52)

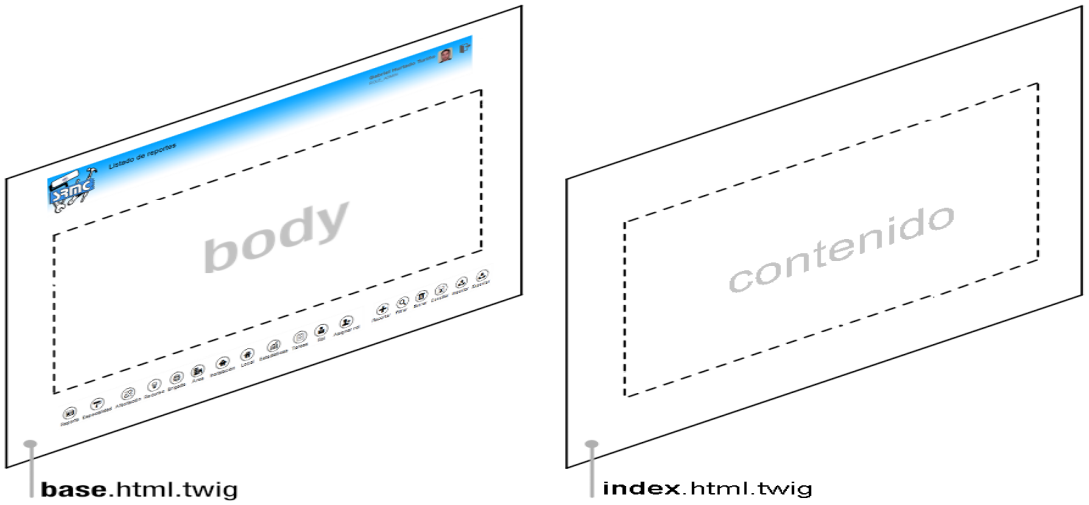


Ilustración 5. Herencia de plantillas

Conclusiones parciales

Mediante las fases de exploración, planificación e iteraciones quedaron definidas las características para el desarrollo de la solución. Fueron concebidas por el equipo de desarrollo y el cliente veintiséis historias de usuario, para lograr una planificación en el desarrollo de la aplicación. La estimación del esfuerzo y plan de duración permitieron obtener una apreciación de cinco meses y medio aproximadamente. Las cinco funcionalidades concebidas a desarrollar en la primera iteración permitieron obtener una visión de la primera versión del sistema. Con la conformación de las tarjetas CRC se obtuvo una representación de las principales clases y funcionalidades del sistema.

Capítulo 3: Implementación y prueba

Dentro de las fases propuestas por la metodología utilizada se lleva a cabo la implementación y las pruebas del sistema. Dentro de la implementación se examinará el análisis y diseño realizado al sistema, así como las tareas de ingeniería utilizadas como base para la implementación del software. Luego se evaluará la calidad de la aplicación a través de las pruebas, una vez concluido el sistema, para permitir el desarrollo del mismo de forma rápida y segura, documentando además los resultados obtenidos.

3.1 Fase de Implementación

Se especifica en esta fase la implementación de las historias de usuario en su correspondiente iteración, obteniéndose de cada una de ellas una versión funcional del producto. Lo primero es hacer un chequeo de cada HU en conjunto con el plan de iteraciones y se modifica en caso de ser necesario, para esto se crean tareas de desarrollo y poder organizar la implementación. Estas tareas, al contrario de las historias de usuario escritas en el lenguaje del cliente, son escritas en un lenguaje técnico. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones de desarrollo sobre el sistema.

3.1.1 Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad para el cliente:

Tabla 17. Fase de implementación Iteración_1

Historias de usuario	**Tiempo de implementación**	
	Estimación	**Real**
Autenticar usuario	1.5	2
Gestionar usuarios	1.5	1
Gestionar roles	1	0.5
Gestionar reportes	1.5	1
Conciliar	1.5	1

A continuación se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración:

Tabla 18. Tarea_de_ingeniería_1

Tarea	
Número de tarea: 1	**Número de HU: 1**
Nombre: Autenticar usuario.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Fecha inicio: 28/10/2014	Fecha fin: 7/11/2014
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se especifican los datos que mostrará el sistema para llenar por los usuarios (nombre y contraseña) mediante un formulario y el botón iniciar sesión.	

Tabla 19. Tarea_de_ingeniería_2

Tarea	
Número de tarea: 2	**Número de HU: 2**
Nombre: Gestionar usuarios	
Tipo de tarea: desarrollo	Puntos de estimación: 1.5
Fecha inicio: 10/11/2014	Fecha fin: 19/11/2014
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Permite adicionar y mostrar datos del usuario.	

Tabla 20. Tarea_de_ingeniería_3

Tarea	
Número de tarea: 3	**Número de HU: 3**
Nombre: Gestionar roles	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1
Fecha inicio: 19/11/2014	Fecha fin: 28/11/2014

Programador responsable: Gabriel Hurtado y Yoan Pérez
Descripción: Permite mostrar el rol asignado al usuario

Tabla 21. Tarea_de_ingeniería_4

Tarea	
Número de tarea: 4	**Número de HU: 4**
Nombre: Gestionar reportes	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 1.5
Fecha inicio: 1/12/2014	Fecha fin: 10/12/2014
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Permite adicionar, mostrar, editar y eliminar reportes	

Tabla 22. Tarea_de_ingeniería_5

Tarea	
Número de tarea: 5	**Número de HU: 5**
Nombre: Conciliar	
Tipo de tarea: desarrollo	Puntos de estimación: 1.5
Fecha inicio: 10/12/2014	Fecha fin: 19/12/2014
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Permite mostrar un documento excel con datos de los reportes seleccionados.	

3.1.2 Iteración 2

Tabla 23. Fase de implementación Iteración_2

Historias de usuario	**Tiempo de implementación**	
	Estimación	**Real**
Exportar	1.5	1

Gestionar local	1	1
Gestionar instalación	1	1
Gestionar área	1	1
Gestionar brigada	1	0.5
Gestionar recurso	1	1
Gestionar afectación	1	0.5
Gestionar especialidad	1	0.5
Importar Excel	0.5	1
Graficar datos estadísticos	0.5	2
Asignar rol	0.5	0.5
Notificar por correo	1.5	0.5
Filtrar reporte	0.5	1

Las tareas de ingeniería generadas en la iteración 2 se encuentran en el Anexo # 5 del documento.

3.1.3 Iteración 3

Tabla 24. Fase de implementación Iteración_3

Historias de usuario	**Tiempo de implementación**	
	Estimación	**Real**
Filtrar especialidad	0.5	0.5
Filtrar afectación	0.5	0.5
Filtrar recurso	0.5	0.5
Filtrar brigada	0.5	0.25
Filtrar área	0.5	0.25
Filtrar instalación	0.5	0.25
Filtrar local	0.5	0.25

Exportar	0.5	0.5

Las tareas de ingeniería generadas en la iteración 3 se encuentran en el Anexo # 5 del documento.

3.2 Trabajo con Symfony

Symfony divide los proyectos web en aplicaciones y módulos. Dentro de un proyecto, se pueden definir diferentes aplicaciones, todas ellas compartiendo el mismo modelo de datos y un mismo nombre de dominio. El Sistema de Reportes de Mantenimiento Constructivo (SRMC) está formado por una interfaz pública a la que acceden los usuarios y una interfaz de administración con la que los responsables del sitio manejan su funcionamiento y gestionan sus contenidos. Además, cada interfaz se divide en varios módulos. Cada módulo se corresponde con una tabla de la base de datos.

3.2.1 Arquitectura del sistema

La arquitectura de software, es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes desde la visión del resto del sistema y las formas en que los componentes interactúa. Además, guía al equipo de desarrollo a través del ciclo de vida del sistema (53).

Una arquitectura de diseño es el Modelo Vista Controlador (MVC) para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio. Es una especialización de un modelo de capas, con la diferencia que se usa para entornos web como patrón por excelencia:

MVC:

- El modelo, es la capa encargada de encapsular toda la lógica de negocio de la aplicación.
- La vista, es la respuesta de cada controlador mostrando al usuario la información que este necesita.
- El controlador, es el eje central de la arquitectura, encargada de gestionar todas las peticiones, validar los inputs recibidos y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas. Recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización (54).

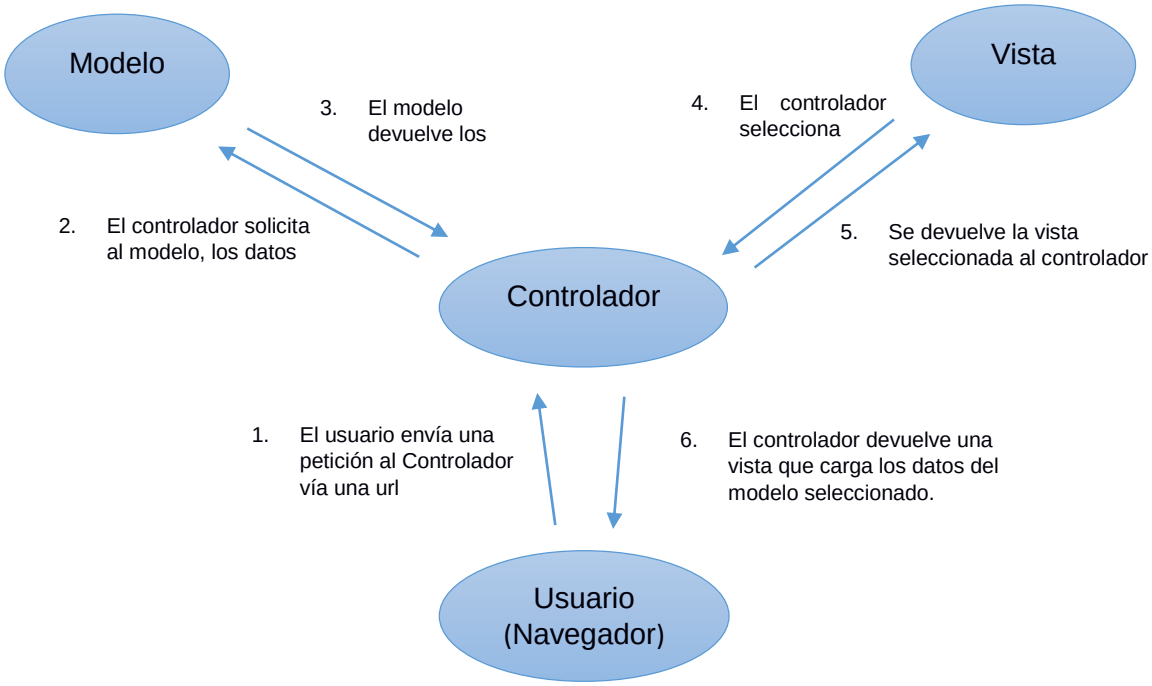


Ilustración 6. Arquitectura MVC

Para la implementación de Symfony se utilizan varios patrones, situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

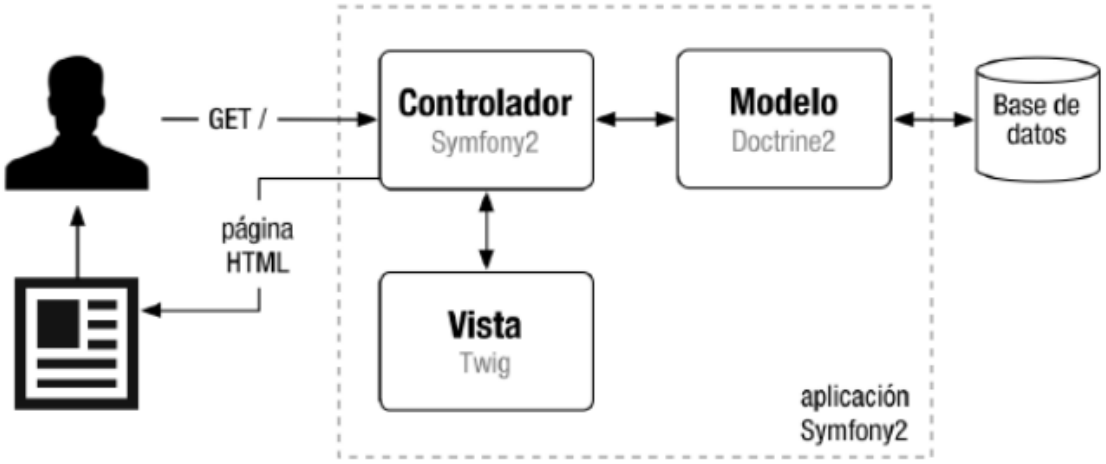


Ilustración 7. Arquitectura MVC en Symfony

3.3 SRMC

Con el desarrollo del sistema de reportes de incidencias de mantenimiento constructivo para la universidad, los trabajadores de mantenimiento podrán gestionar los procesos de mantenimiento en las áreas de la UCI y la comunidad a su vez podrá dar seguimiento a estos procesos. Este sistema tiene implementada todas las funcionalidades

requeridas por el cliente, permite al usuario una vez autenticado realizar reportes mediante una interfaz con los formularios para este fin. Una vez terminado el reporte recibirá una notificación vía correo y ante la modificación realizada por parte del técnico de mantenimiento donde se da respuesta o solución a dicho reporte, volverá a ser notificado mediante un correo.

El sistema se encuentra en el directorio Reportes\src\Reportes mostrándose en la siguiente ilustración:

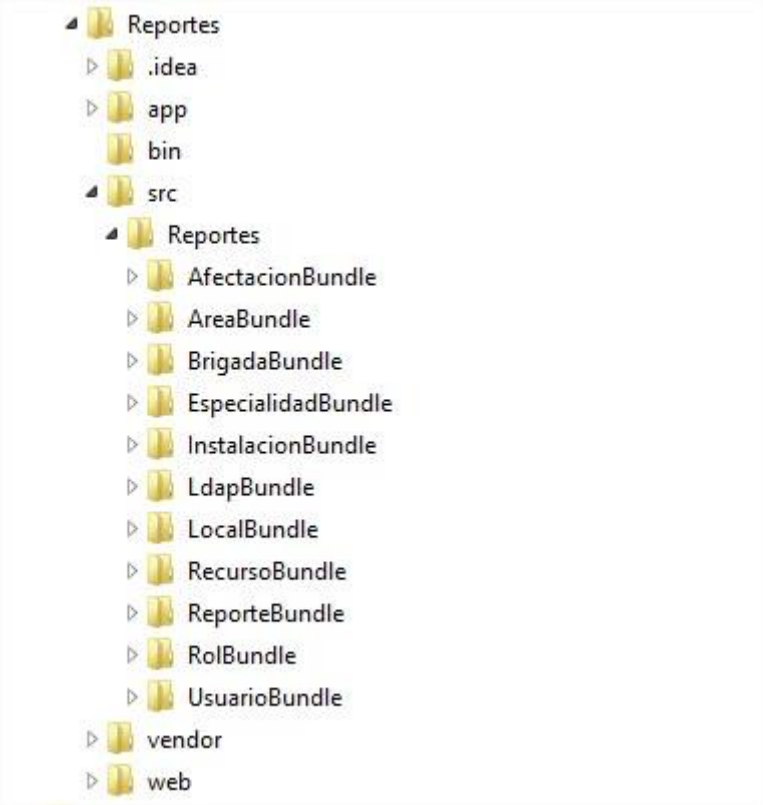


Ilustración 8. Ubicación del sistema

3.4 Modelo de despliegue

El Diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores – dispositivos – componentes de software (55).

La configuración propuesta estará conformada por un Servidor Web. Este a su vez se conectará con el servidor LDAP para verificar la autenticidad del usuario, posteriormente se conectará con la Base de Datos para almacenar toda la información del sistema. Las máquinas clientes a su vez estarán conectadas con el Servidor Web.

3.5 Trabajo con la base de datos

Una de las tareas principales en el diseño de una aplicación es la construcción de la base de datos, en esta se ponen de manifiesto los datos necesarios para el correcto funcionamiento de la misma. Una de las tareas complejas para cualquier aplicación involucra la persistencia y lectura de información hacia y desde una base de datos. *Symfony* viene integrado con Doctrine, una biblioteca, cuyo objetivo es dotar de poderosas herramientas. (56)

Doctrine puede crear automáticamente todas las tablas de la base de datos necesarias para cada entidad mediante el uso del comando: doctrine:schema:create y crear la base de datos mediante el comando: doctrine:database:create (57).

En la base de datos se cuenta con las siguientes tablas: afectacion, area, brigada, afectacion_especialidad, especialidad, instalacion, local, recurso, recurso_afectacion, reporte, rol, usuario, usuario_rol, recurso_especialidad, donde se acumula toda la información que necesitan los usuarios que interactúan con la aplicación. El diagrama del modelo base de datos se muestra a continuación en la Ilustración 9:

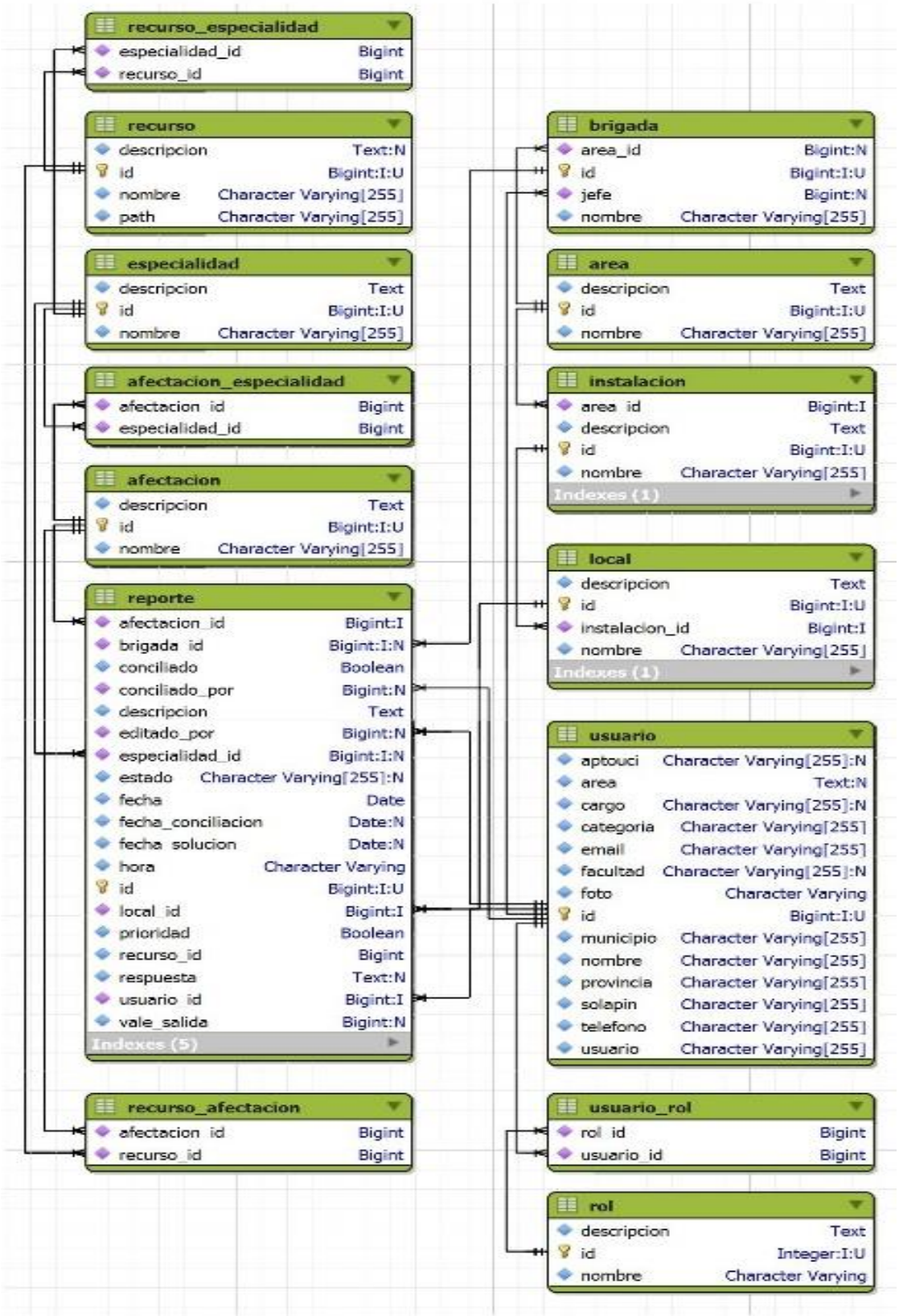


Ilustración 9. Modelo base de datos

3.6 Pruebas

Uno de los pilares de XP es el proceso de pruebas. Anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas y reduce el número de errores no detectados, disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. Las pruebas permiten comprobar la eficacia del sistema; estas son las responsables de verificar si los objetivos trazados fueron cumplidos en la etapa de implementación. Con ellas se reduce el número de errores no detectados durante la implementación, el tiempo entre la introducción de estos en el sistema y su detección; son las encargadas de aumentar la seguridad y de evitar efectos colaterales no deseados a la hora de realizar modificaciones en la aplicación (58).

3.6.1 Pruebas unitarias

Las pruebas unitarias deben ser construidas antes que el código, permitiéndole a los programadores tener máxima claridad de lo que van a programar antes de hacerlo, así como conocer cada una de las historias de usuarios que deberán pasar, lo que optimizará el trabajo y el código será de mayor calidad (59).

Este tipo de pruebas es realizada por una persona especializada en *Software testing*, la cual está familiarizada en el uso de herramientas de depuración y pruebas. Algunas de las herramientas que se utilizan para pruebas unitarias son: JUnit, La Suite de Mercury, CPPUnit, entre otras. El objetivo fundamental de las pruebas unitarias es asegurar el correcto funcionamiento de las interfaces mediante el código, o flujo de datos entre componentes (60).

Lo importante en este tipo de pruebas es que se deben tener claros los siguientes aspectos:

- Los datos de entrada deben ser preparados con minuciosidad, ya que el resultado de las pruebas dependen de estos.
- Se debe conocer qué componentes interactúan en cada historia de usuario.
- Se debe conocer de antemano qué resultados debe devolver el componente según los datos de entrada utilizados en la prueba.
- Finalmente se deben comparar los datos obtenidos en la prueba con los datos esperados, si son idénticos se puede decir que la aplicación superó la prueba.

Symfony integra una biblioteca independiente llamada *PHPUnit* que proporciona una plataforma de pruebas. Todas las pruebas en symfony están ubicadas bajo el directorio test/ del proyecto. Este tiene a su vez dos sub-directorios, uno para pruebas unitarias (*test/unit/*) y otro para las pruebas funcionales (*test/functional/*). Esta biblioteca permite mostrar un mensaje de error aún en caso de que sean pocas las pruebas que se ejecutan. Las pruebas implican llamar a un método o a una función con un conjunto predefinido de argumentos y comparar la salida con los resultados esperados. Esta comparación determina si una prueba pasa o no. Se muestra a continuación un ejemplo de prueba unitaria ejecutada al código del método getSlug() de la clase Recurso:

```
<?php
namespace Reportes\ReporteBundle\Tests\Entity;
use Reportes\ReporteB
undle\Util\Util;
use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
class UtilTest extends \PHPUnit_Framework_TestCase
{
   public function testGetSlug()
   {
      $util = new Util();
      $this->assertEquals('herraje-de-bano',$util->getSlug("Herraje de baño"));
   }
}
Time: 516 ms, Memory: 15.75Mb
OK (1 test, 1 assertion)
```

3.6.2 Pruebas de aceptación

Las pruebas de aceptación son realizadas como técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento, estas son creadas a partir de las historias de usuario y desde la perspectiva del cliente. En las pruebas de aceptación se deben especificar uno o varios escenarios para comprobar que una HU ha sido correctamente implementada. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. En estos escenarios de prueba se indican las posibles respuestas que tiene el software en la utilización de cada funcionalidad.

Como resultado del trabajo realizado durante la fase de pruebas, se ejemplifican a continuación algunos escenarios de las pruebas de aceptación:

Tabla 25. Caso de prueba P1_1

Caso de prueba de aceptación.	
Código: P1	**No. de HU: 1**
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad autenticar usuario.	
Condiciones de ejecución: El caso de prueba inicia cuando el usuario del sistema introduce una contraseña y un usuario válido.	
Pasos de ejecución: El usuario introduce una contraseña y un usuario válido en el sistema y al hacer clic en iniciar sesión accede al sistema. Si es la primera vez que le usuario es autenticado, se le asigna el rol usuario por defecto y se muestra la interfaz correspondiente a dicho rol, si no se muestra la interfaz correspondiente al rol que posea.	
Resultado esperado: El usuario se autentica y accede al sistema según su rol.	
Evaluación de la prueba: Satisfactoria.	

Tabla 26. Caso de prueba P2_2

Caso de prueba de aceptación.	
Código: P2	**No. de HU: 2**
Nombre: Mostrar datos del usuario	
Descripción: Prueba para la funcionalidad de mostrar datos de usuario	
Condiciones de ejecución: Cualquier usuario autenticado.	
Pasos de ejecución: El usuario presiona en el banner, en su foto o nombre.	
Resultado esperado: Se muestra la interfaz con los datos del usuario autenticado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 27. Caso de prueba P3_3

Caso de prueba de aceptación.	
Código: P3	**No. de HU: 3**
Nombre: Mostrar rol	

Descripción: Prueba para la funcionalidad de mostrar rol
Condiciones de ejecución: Administrador autenticado.
Pasos de ejecución: El usuario presiona el botón Asignar rol y luego presiona en cualquiera de los usuarios existentes en el listado.
Resultado esperado: Se muestran los datos del usuario seleccionado, donde aparece el rol que posee.
Evaluación de la prueba: Satisfactoria.

Tabla 28. Caso de prueba P4_4

Caso de prueba de aceptación.	
Código: P4	**No. de HU: 4**
Nombre: Crear reporte	
Descripción: Prueba para la funcionalidad de crear reporte	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución El administrador o el técnico general del sistema acceden a la opción Reportar, luego introducen los datos correctamente para crear un nuevo reporte de afectación, luego acceden a la opción Crear reporte. El sistema muestra el mensaje "La afectación fue creada correctamente con número de id #" y envía una notificación de reporte de afectación vía correo al usuario que realiza el reporte.	
Resultado esperado: El sistema crea el reporte y muestra el mensaje "La afectación fue creada correctamente con número de id #" y envía una notificación de reporte de afectación vía correo al usuario que realiza el reporte.	
Evaluación de la prueba: Satisfactoria.	

Tabla 29. Caso de prueba P5_4

Caso de prueba de aceptación.	
Código: P5	**No. de HU: 4**
Nombre: Editar reporte	
Descripción: Prueba para la funcionalidad de crear reporte	
Condiciones de ejecución: Técnico general o Administrador autenticado.	

Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos del reporte seleccionado. Todos los campos exceptuando Vale de salida son obligatorios. Dan clic en el botón actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente" y envía una notificación de reporte actualizado vía correo al usuario que reportó la afectación.
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra el mensaje "Los datos fueron actualizados correctamente " y envía una notificación de reporte actualizado vía correo al usuario que reportó la afectación.
Evaluación de la prueba: Satisfactoria.

Tabla 30. Caso de prueba P6_4

Caso de prueba de aceptación.	
Código: P6	**No. de HU: 4**
Nombre: Mostrar reporte	
Descripción: Prueba para la funcionalidad de crear reporte	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Mostrar datos del reporte correspondiente al reporte deseado y el sistema muestra una interfaz con los datos del reporte seleccionado.	
Resultado esperado: El sistema muestra una interfaz con los datos del reporte seleccionado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 31. Caso de prueba P7_4

Caso de prueba de aceptación.	
Código: P7	**No. de HU: 4**
Nombre: Eliminar reporte	
Descripción: Prueba para la funcionalidad de crear reporte	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente al reporte que desea eliminar. Cuando dan clic en borrar el sistema	

muestra el mensaje "¿Seguro que desea eliminar el reporte?" seleccionan la opción Aceptar. El sistema muestra el mensaje "El reporte ha sido eliminado correctamente".
Resultado esperado: El sistema elimina el reporte seleccionado y muestra el mensaje "El reporte ha sido eliminado correctamente".
Evaluación de la prueba: Satisfactoria.

Los restantes casos de prueba están definidos en el Anexo # 6 del documento.

Resultados de las pruebas de aceptación

El grado de satisfacción del cliente atendiendo a sus necesidades se pudo obtener a partir de cada entrega realizada. Para ello se realizaron 3 iteraciones de las pruebas de aceptación y se detectaron no conformidades significativas referentes a errores de validación, no significativas que se centraron fundamentalmente en errores ortográficos y varias recomendaciones. Las no conformidades detectadas quedan registradas en la siguiente tabla:

Tabla 32. No conformidades detectadas

Número de iteración	**No conformidades**		**Recomendaciones**	**No conformidades resueltas**
	Significativas	**No significativas**		
1	2	1	0	3
2	3	2	2	7
3	1	0	1	2

Las pruebas se realizaron de forma iterativa e incremental, y se comprobó en cada iteración la corrección de los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del software. Razón por la cual la aplicación quedó aprobada por parte del cliente, quien evaluó el 100 % de los casos de prueba de forma satisfactoria.

Conclusiones parciales

Al concluir este capítulo se demuestra que una historia de usuario es un resumen de una funcionalidad del sistema, mientras que los desarrolladores van ampliando sus conocimientos a medida que se va implementando. Se demuestra la posibilidad de realizar un correcto diseño de la aplicación con buenas prácticas, como son el uso de

patrones arquitectónicos y de diseño. Además, se evidenció la necesidad de realizar pruebas al software para corregir errores no deseados; donde el cliente comprueba el cumplimiento y la satisfacción de lo que esperaba con el sistema.

Conclusiones generales

Al finalizar la investigación se arribaron a las siguientes conclusiones:

- El estudio del estado del arte permitió definir las características de la solución y los métodos científicos a utilizar.
- El análisis y diseño de la solución propuesta estuvo guiado por la metodología de desarrollo de software XP, la cual permitió generar los artefactos que documentaron la solución, para continuar con el desarrollo futuro de nuevas versiones y centró al equipo de desarrollo en los requerimientos de esta.
- Las pruebas de aceptación permitieron validar el sistema, demostrar su correcto funcionamiento y satisfacer las necesidades definidas por el cliente.
- Con el desarrollo del SRMC, la comunidad universitaria cuenta con un sistema que contribuye con el proceso de gestión de reportes de incidencias de mantenimiento constructivo en la Universidad de las Ciencias Informáticas.

Recomendaciones

Luego de haber finalizado la presente investigación se recomienda:

- Incorporar nuevas restricciones y roles al SRMC que permitan mejorar los procesos de gestión de reportes.
- Integrar el sistema SRMC con ASET, sistema que utilizan los directivos de mantenimiento de la UCI para controlar la existencia de los recursos en el almacén.

Referencias

1. **Real Academia Española.** Real Academia Española. [En línea] [Citado el: 03 de 12 de 2014.] http://lema.rae.es/drae/?val=mantenimiento.

2. **Habana, Oficina del historiador de la.** Plan maestro. [En línea] [Citado el: 22 de 11 de 2014.] http://www.planmaestro.ohc.cu/index.php/instrumentos/glosario#m.

3. **ONEI.** *Construcción en Cuba.Indicadores seleccionados.*

4. **Farlex.** The Free Dictionary. [En línea] [Citado el: 17 de 04 de 2015.] http://es.thefreedictionary.com/incidencia.

5. **Definicion.de.** Definicion.de. [En línea] 2015. [Citado el: 03 de 11 de 2014.] http://definicion.de/reporte/.

6. **Caschile.** Gestor de Reportes. [En línea] [Citado el: 10 de 11 de 2014.] http://www.caschile.cl/index.php?option=com_content&view=article&id=458:nuevo-sistema-gestor-de-reportes&catid=35:noticias-destacadas..

7. **Sandra Gomez.** Reporte de incidencias con Mantis. [En línea] 2013. [Citado el: 22 de 12 de 2014.] http://www.globetesting.com/2013/01/reporte-de-incidencias-mantis/.

8. **Universidad deLujan .** Universidad de Lujan Argentina. [En línea] [Citado el: 25 de 11 de 2014.] http://www.seguridadinformatica.unlu.edu.ar/?q=node/4.

9. **Universidad de Chile.** [En línea] [Citado el: 11 de 12 de 2014.] http://www.uchile.cl/portal/presentacion/vicerrectoria-de-asuntos-economicos-y-gestion-institucional/convenio-de-desempeno/areas-de-accion-y-proyectos/72479/sistemas-de-gestion-para-la-toma-de-decisiones.

10. *GeReport: Sistema de Gestión de Reportes Dinámicos.* **Cinthya Rodriguez Hernández, Rafael Felipe Bomate Gavio, Yenia Román Bu, Carlos Manuel Delgado Rivero, Manuel Cortés Cortés.** s.l. : RCCI,Revista Cubana de Ciencias Informáticas, 2014.

11. **Hechavarría, José Luis Chamizo.** *Módulo de Reportes del Sistema Integral de Análisis de Información.* Facultad 2, UCI, Cuidad de la Habana : UCI, Julio, 2013.

12. **Yanioski Hernandez, Agneri Martínez, José Orlando Lafaurie.** *Sistema para la generación de reportes en la plataforma alasGRATO.* La Habana : UCI, 2008.

13. **UCI.** Soporte de la UCI. . [En línea] [Citado el: 26 de 12 de 2014.] http://soporte.uci.cu..

14. **Avison, D.E. y Fitzgerald, G.** *Information Systems Development: Methodologies, Techniques, and Tools.* McGraw-Hill : s.n., 1995.

15. **Patricio Letelier.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* Universidad Politécnica de Valencia (UPV) : s.n.

16. **Beck, Kent.** *Embracing Change with Extreme Programming. .* ISBN-10: 0201616416 | ISBN-13: 978-0201616415. : Addison Wesley Longman. Inc, 1999.

17. **Riola, Jose Carlos Carvajal.** *Metodologías Agiles:Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial. .* Barcelona : UPC-Barcelona, 2008.

18. **Álvarez, Miguel Angel.** Desarrollo web. [En línea] [Citado el: 05 de 02 de 2015.] http://www.desarrolloweb.com/articulos/497.php..

19. **Álvarez, Miguel Angel.** Desarrollo web. [En línea] [Citado el: 20 de 01 de 2015.] http://www.desarrolloweb.com/articulos/393.php..

20. **Achour, Mehdi.** Manual de PHP. . [En línea] [Citado el: 10 de 01 de 2015.] http://docs.php.net/manual/es/.

21. **Learning, Digital.** Digital Learning. [En línea] http://www.digitallearning.es/blog/rankings-de-lenguajes-de-programacion/.

22. **Almira, Yanet de los Angeles Cardenas.** *MÓDULO RECOMENDADOR DE LIBROS PARA LA TIENDA VIRTUAL DEL PORTAL CUBALITERARIA. .* La habana : s.n., 2014.

23. **Rivera, E.A., Zamora, R.G. y Soria, M.G.** *Sistema de Educación a Distancia.* s.l. : IV Congreso de Tecnología en Educación, 2012.

24. **Libros Web.** Libros Web. . [En línea] [Citado el: 04 de 02 de 2015.] http://librosweb.es/xhtml/capitulo_2.html..

25. **PHP-UCI, Comunidad.** <PHP>La comunidad uci de PHP. . [En línea] [Citado el: 04 de 02 de 2015.] https://php.uci.cu/articles.php?cat_id=20.

26. **jQuery.** The jQuery Foundation. [En línea] 2013. [Citado el: 15 de 02 de 2015.] http://jquery.com/.

27. **Editores de código.** Editores de código. [En línea] [Citado el: 26 de 02 de 2015.] http://www.editoresdecodigo.com/2014/06/descargar-phpstorm-full-ide-para-php-y-mas.html.

28. **libros web.** LibrosWeb. . [En línea] [Citado el: 23 de 01 de 2015.] http://librosweb.es/libro/introduccion_rails/capitulo_2.html.

29. **Guerrero, Ariagna.** *Sistema para la Gestión de Mantenimiento para Proyectos Productivos en la UCI.* s.l. : UCI, 2009.

30. **libros web.** libros web. [En línea] [Citado el: 26 de 11 de 2014.] http://librosweb.es/symfony/capitulo_1/symfony_en_pocas_palabras.html..

31. **José Luis Murga.** Blog de SEO, Posicionamiento en Buscadores, Diseño Web y Desarrollo Web. [En línea] 2013. [Citado el: 17 de 02 de 2015.] http://www.jlmurgas.net/recursos/crea-tu-proyecto-web-al-estilo-modern-de-windows-8-con-bootmetro/.

32. **web tutoriales.** Web Tutoriales . [En línea] [Citado el: 05 de 02 de 2015.] http://www.webtutoriales.com/articulos/yaml..

33. **Symfony.** Symfony en español. [En línea] [Citado el: 23 de 11 de 2014.] http://gitnacho.github.io/symfony-docs-es/components/yaml.html.

34. **sitio web.** web, sitio . [En línea] [Citado el: 07 de 12 de 2014.] http://gitnacho.github.io/Twig/intro.html..

35. **Ing. Grettel Susel Incencio Piñeiro.** *Sistema informático para la evaluación de atributos de calidad en componentes biométricos.* Granma : s.n., 2014.

36. **Torstein Hønsi.** Highcharts product. [En línea] 2011. [Citado el: 02 de 04 de 2015.] http://www.highcharts.com/products/highcharts.

37. **Lee Babin.** *Dialnet.* España : ANAYA MULTIMEDIA, 2007.

38. **Valentina.** Valentina. [En línea] 2015. [Citado el: 11 de 04 de 2015.] https://www.valentina-db.com/en/valentina-studio-overview.

39. **Avila, Katty.** CAVSI. [En línea] [Citado el: 05 de 12 de 2014.] http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/..

40. **Alegsa, Leandro. .** Alegsa. [En línea] [Citado el: 05 de 12 de 2014.] http://www.alegsa.com.ar/Dic/sgbd.php..

41. **PostgreSql.** PostgreSql-es. [En línea] [Citado el: 12 de 02 de 2015.] http://www.postgresql.org.es/sobre_postgresql..

42. **Fabien Potencier, Ryan Weaver.** *Symfony 2.4, el libro oficial.* 2015.

43. **Kabir, Mohammed J.** *La Biblia Servidor Apache 2.* s.l. : Anaya Multimedia, 1999. ISBN: 8441514682.

44. **Google Sites.** Metodología Gestión de Requerimientos. [En línea] [Citado el: 28 de 03 de 2015.] https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales.

45. —. Metodología Gestión de Requerimientos. [En línea] [Citado el: 28 de 03 de 2015.] https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales.

46. **Priolo, Ing. Sebastian.** *Programación Extrema.* 2011.

47. **Beck, K.** *Extreme Programming Explained.* s.l. : Addison-Wesley, 2000.

48. **Luis Olfrides Pérez Rodríguez, Daliana González Martínez.** *Sistema de Control de Laboratorios (CLAB).* 2013.

49. **Solis, Ricardo Tercero.** *XP Programación Extrema para Desarrollo de Sistema Basados en Web. Sistematización de Control del Parque Vehicular de la Secretaría del Medio Ambiente y Recursos Naturales .* s.l. : Delegación Morelos, 21 de abril de 2010. 1.1.

50. **Bajo de Luque, M.J. y et.al.** Metodologías ágiles . [En línea] Agile Alliance. [Citado el: 25 de 02 de 2015.] http://metologiasagiles.herobo.com/index.php/2011-12-05-16-09-55/metologia-xp.

51. **Félix Prieto.** *Patrones de diseño.* Universidad de Valladolid : s.n., 2008.

52. **Larman, Craig.** *UML y Patrones Introducción al análisis y diseño orientado a objetos. .* México : s.n., 1999.

53. **Carlos Billy Reynoso.** *Introducción a la Arquitectura del software.* Buenos Aires : Universidad de Buenos Aires, 2004.

54. **José Jorge Márquez Gómez.** *ARQUITECTURA MVC.*

55. **LinkedIn Corporation © 2015.** Diagramas de Despliegue. [En línea] [Citado el: 27 de 03 de 2015.] http://es.slideshare.net/arcangelsombra/diagramas-de-despligue-uml-1475353.

56. **Nacho Pacheco.** Symfony en español. [En línea] 2013. [Citado el: 26 de 02 de 2015.] http://gitnacho.github.io/symfony-docs-es/book/doctrine.html.

57. —. Symfony en Español. [En línea] [Citado el: 26 de 03 de 2015.] http://gitnacho.github.io/symfony-docs-es/book/doctrine.html.

58. **J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres .** *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN .* s.l. : Department de Lenguajes y Sistemas Informáticos .

59. **Saboya Vargas, Aniceto.** *Uso de Recomendadores, Asistentes y Ayudantes en sistemas.* Catalunya, España : s.n., 2005.

60. **Ing. Alexander Oré B.** Calidadysoftware.com. [En línea] [Citado el: 26 de 03 de 2015.] http://www.calidadysoftware.com/testing/pruebas_unitarias1.php.

61. **Hospital Universitario.** Virgen de las Nieves. [En línea] [Citado el: 20 de 11 de 2014.] http://www.hvn.es/servicios_generales/gestionambiental/gestion_ambiental/reporte_incidencias.php..

62. **Lemus, Juan Manuel.** Maestros del Web. [En línea] [Citado el: 02 de 05 de 2015.] http://www.maestrosdelweb.com/hojas-de-estilo-con-yaml/..

63. **Stephen Burns.** *Dialnet.* España : ANAYA MULTIMEDIA, 2008.

64. **Revista Cubana de Ciencias Informáticas.** Revista Cubana de Ciencias Informáticas. [En línea] [Citado el: 10 de 12 de 2014.] http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path[]=726 ..

Glosario de términos

Ajax: Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

CSS: Las hoja de estilo en cascada (en inglés Cascading Style Sheets) contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llama “cascada” porque una hoja puede heredar los formatos definidos en otra hoja de forma de que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación de un Sitio Web.

Doctrine: Es un Mapeador de Objeto Relacional (ORM) creado para PHP.

Entorno de desarrollo integrado (IDE): Programa compuesto por un conjunto de herramientas para un programador que puede dedicarse en exclusiva a un sólo lenguaje de programación o bien puede utilizarse para varios. Consisten en un editor de código, un compilador, un depurador y un GUI.

Frameworks: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

HTML: Hipertext Transfer Protocol (Protocolo de transferencia de hipertexto), es un método común de intercambio de información en la World Wide Web, mediante el cual se transfieren las páginas web a un ordenador.

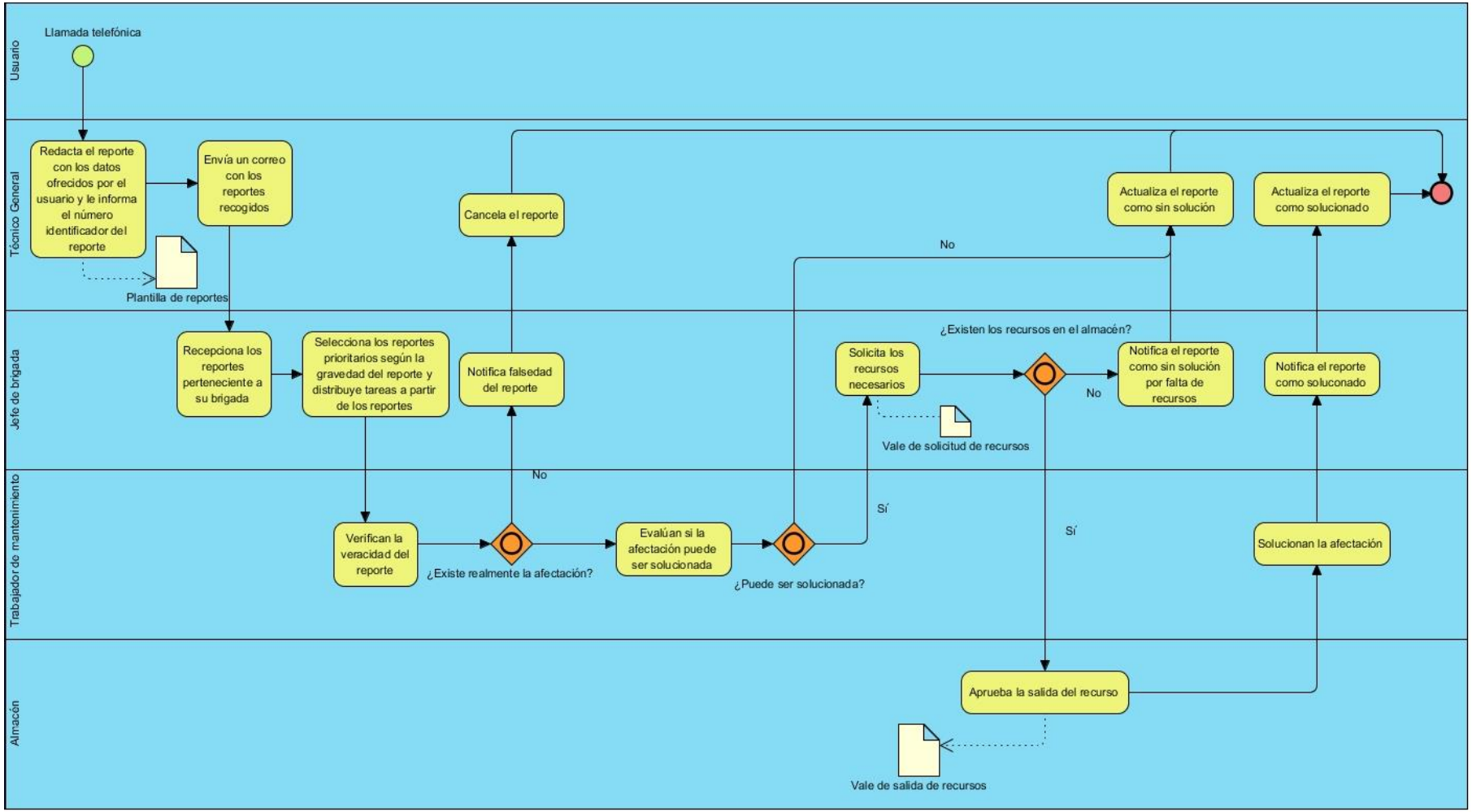
HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

JavaScript: Lenguaje de programación interpretado, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Actualmente todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web.

PHP: (Hypertext Preprocessor) es un lenguaje multiplataforma, multiparadigma, script (no se compila para conseguir códigos máquina sino que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código) para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código.

Web: Es un sistema para presentar información en Internet basado en hipertexto. Cuándo se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

Anexo 1


Usuario
Técnico General
Jefe de brigada
Trabajador de mantenimiento
Almacén
Llamada telefónica
Redacta el reporte con los datos ofrecidos por el usuario y le informa el número identificador del reporte
Envía un correo con los reportes recogidos
Plantilla de reportes
Recepciona los reportes perteneciente a su brigada
Selecciona los reportes prioritarios según la gravedad del reporte y distribuye tareas a partir de los reportes
Verifican la veracidad del reporte
¿Existe realmente la afectación?
No
Notifica falsedad del reporte
Cancela el reporte
Evalúan si la afectación puede ser solucionada
¿Puede ser solucionada?
Sí
No
Solicita los recursos necesarios
Vale de solicitud de recursos
¿Existen los recursos en el almacén?
No
Notifica el reporte como sin solución por falta de recursos
Actualiza el reporte como sin solución
Sí
Aprueba la salida del recurso
Vale de salida de recursos
Solucionan la afectación
Notifica el reporte como soluconado
Actualiza el reporte como solucionado

Anexo 2

Historia de Usuario	
Número: 6 Usuario: Técnico General, Administrador	Nombre: Exportar
Prioridad en el negocio: Media	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 1
Descripción: Permite exportar a documento excel todos los reportes realizados.	
Observaciones: El Técnico General o el Administrador, debe estar autenticado para acceder a esta funcionalidad. Los reportes se almacenan en un documento excel para ser analizados en la reunión mensual que realiza la dirección de mantenimiento.	

Historia de Usuario	
Número: 7 Usuario: Técnico General, Administrador	Nombre: Gestionar áreas.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar las áreas, mientras que al usuario autenticado solo le brindará la información necesaria.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 8 Usuario: Técnico General, Administrador	Nombre: Gestionar brigadas.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar las brigadas, mientras que al usuario autenticado solo le brindará la información necesaria.	

Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.

Historia de Usuario	
Número: 9 Usuario: Técnico General, Administrador	Nombre: Gestionar recursos.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar recursos, mientras que al usuario autenticado solo le brindará la información necesaria.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 10 Usuario: Técnico General, Administrador	Nombre: Gestionar afectaciones.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar las afectaciones, mientras que al usuario autenticado solo le brindará la información necesaria.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 11 Usuario: Técnico General, Administrador	Nombre: Gestionar especialidades.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2

Descripción: Permite crear, mostrar, editar y eliminar las especialidades, mientras que al usuario autenticado solo le brindará la información necesaria.
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.

Historia de Usuario	
Número: 12 Usuario: Técnico General, Administrador	Nombre: Gestionar local.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar los locales, mientras que al usuario autenticado solo le brindará la información necesaria.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 13 Usuario: Técnico General, Administrador	Nombre: Gestionar instalaciones.
Prioridad en el negocio: Alta	Puntos Estimados: 1
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite crear, mostrar, editar y eliminar las instalaciones, mientras que al usuario autenticado solo le brindará la información necesaria.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 14 Usuario: Técnico General, Administrador	Nombre: Importar excel
Prioridad en el negocio: Media	Puntos Estimados: 0.5

Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite desde un documento excel importar los reportes a la aplicación informática.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 15 Usuario: Técnico General, Administrador	Nombre: Graficar datos estadísticos
Prioridad en el negocio: Media	Puntos Estimados: 0.5
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Muestra un gráfico de los reportes realizados en un rango de fecha determinado	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 16 Usuario: Administrador	Nombre: Asignar rol
Prioridad en el negocio: Alta	Puntos Estimados: 0.5
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Se le asigna el rol de acuerdo a la autenticación de un usuario	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 17 Usuario: Técnico general, Administrador	Nombre: Notificar por correo

Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alto	Iteración asignada: 2
Descripción: Una vez realizado un reporte o modificado, se le envía una notificación de correo al usuario.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 18 Usuario: Técnico general, Administrador	Nombre: Filtrar reporte
Prioridad en el negocio: Alta	Puntos Estimados: 0.5
Riesgo de desarrollo: Medio	Iteración asignada: 2
Descripción: Permite filtrar los reportes	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 19 Usuario: Técnico general, Administrador	Nombre: Filtrar especialidad
Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según la especialidad	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 20 Usuario: Técnico general, Administrador	Nombre: Filtrar afectación

Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según la afectación	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 21 Usuario: Técnico general, Administrador	Nombre: Filtrar recurso
Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según el recurso	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 22 Usuario: Técnico general, Administrador	Nombre: Filtrar brigada
Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según la brigada	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 23 Usuario: Técnico general, Administrador	Nombre: Filtrar área

Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según el área	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 24 Usuario: Técnico general, Administrador	Nombre: Filtrar instalación
Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según la instalación	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 25 Usuario: Técnico general, Administrador	Nombre: Filtrar local
Prioridad en el negocio: Baja	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite filtrar según el local	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Historia de Usuario	
Número: 26 Usuario: Técnico general, Administrador	Nombre: Exportar gráfico

Prioridad en el negocio: Media	Puntos Estimados: 0.5
Riesgo de desarrollo: Bajo	Iteración asignada: 3
Descripción: Permite exportar datos estadísticos a formato pdf, png, jpg, svg.	
Observaciones: El Técnico General o el Administrador deben estar autenticado para acceder a esta funcionalidad.	

Anexo 3

Exportar	
Responsabilidades	**Colaboraciones**
streamAction() cellColor($phpExcelObject, $cells, $color)	Reporte

Gestionar área	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request) createCreateForm(Area $entity) newAction() createEditForm(Area $entity) updateAction(Request $request, $id) deleteAction(Request $request, $id) borrarAction() showAction($id) editAction($id)	

Gestionar brigada	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request)	Usuario, usuario, Area, area

createCreateForm(Brigada $entity) newAction() showAction($id) createEditForm(Brigada $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction() editAction($id)	

Gestionar recurso	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request) createCreateForm(Recurso $entity) newAction() showAction($id) editAction($id) createEditForm(Recurso $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction()	Afectacion, afectacion, Especialidad, especialidad

Gestionar afectación	
Responsabilidades	**Colaboraciones**

indexAction() createAction(Request $request) createCreateForm(Afectacion $entity) showAction($id) editAction($id) createEditForm(Afectacion $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction()	Especialidad, especialidad

Gestionar especialidad	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request) createCreateForm(Especialidad $entity) newAction() showAction($id) editAction($id) createEditForm(Especialidad $entity) updateAction(Request $request,$id) deleteAction(Request $request, $id) borrarAction()	

Gestionar local

Responsabilidades	Colaboraciones
indexAction() createAction(Request $request) createCreateForm(Local $entity) newAction() showAction($id) editAction($id) createEditForm(Local $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction()	Instalacion, instalacion

Gestionar instalación	
Responsabilidades	**Colaboraciones**
indexAction() createAction(Request $request) createCreateForm(Instalacion $entity) newAction() showAction($id) editAction($id) createEditForm(Instalacion $entity) updateAction(Request $request,$id) deleteAction(Request $request,$id) borrarAction()	Area, area

Importar excel	
Responsabilidades	**Colaboraciones**
leerAction()	Afectacion, Local, Brigada, Especialidad, Recurso, Reporte, Usuario, Rol, Area, Instalacion, afectacion, local, brigada, especialidad, recurso, reporte, usuario, rol, area, instalacion, afectacion_especialidad, recurso_afectacion, recurso_especialidad

Graficar datos estadísticos	
Responsabilidades	**Colaboraciones**
graficarAction() consultasAction()	Reporte, reporte

Asignar rol	
Responsabilidades	**Colaboraciones**
indexAction() showAction($id) editAction($id) createEditForm(Usuario $entity) updateAction(Request $request, $id)	Usuario, usuario, usuario_rol

Notificar por correo	
Responsabilidades	**Colaboraciones**
emailAction($name, $usuario, $entity)	Reporte, Usuario, reporte, usuario

emailActualizarAction($name, $usuario, $entity)	

Filtrar reporte	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar especialidad	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar afectación	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar recurso	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar brigada	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar área	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar instalación	
Responsabilidades	**Colaboraciones**
indexAction()	

Filtrar local	
Responsabilidades	**Colaboraciones**
indexAction()	

Anexo 4

```
$soap = new
\SoapClient('https://autenticacion2.uci.cu/v6/PasarelaAutenti
cacionWS.wsdl');
        $user = $soap-
>ObtenerPersonaDadoCredencial($_POST['solapin']);
        $em = $this->getDoctrine()->getManager();
        $usuario_login = $em-
>getRepository('UsuarioBundle:Usuario')-
>findOneBy(array('usuario' => $this->getUser()-
>getUsername()));
        $entity = new Reporte();
        $form = $this->createCreateForm($entity);
        $form->handleRequest($request);
        //Verifica que el campo usuario no esté vacío
        if ($user->Nombres != "") {
            $local = $em->getRepository('LocalBundle:Local')-
>find($_POST['reportes_reportebundle_reporte']['local']);
            $especialidad = $em-
>getRepository('EspecialidadBundle:Especialidad')-
>find($_POST['reportes_reportebundle_reporte']['especialidad'
]);
```

```
            $afectacion = $em-
>getRepository('AfectacionBundle:Afectacion')-
>find($_POST['reportes_reportebundle_reporte']['afectacion'])
;
            $descripcion =
$_POST['reportes_reportebundle_reporte']['descripcion'];
            $area = $em->getRepository('AreaBundle:Area')-
>find($_POST['areas']);
            $brigada = $em-
>getRepository('BrigadaBundle:Brigada')-
>findOneBy(array('area' => $area));
            $recurso_id = explode("_", $_POST['recurso']);
            $recurso_id = array_pop($recurso_id);
            $recurso = $em-
>getRepository('RecursoBundle:Recurso')->find($recurso_id);
            $fecha = new \DateTime();
            $hora = $fecha->format("g:i:s a");
            $usuario_resultado = $em-
>getRepository('UsuarioBundle:Usuario')-
>findOneBy(array('solapin' => $_POST['solapin']));
//Si no está lo adiciono en la BD
            if (count($usuario_resultado) == 0) {
                $usuario = new Usuario();
                $usuario->setNombre($user->Nombres . " " .
$user->Apellidos);
                $usuario->setUsuario($user->Usuario);
                $usuario->setSolapin($user->Credencial);
                $usuario->setEmail($user->Correo);
                $usuario->setCategoria($user->Categoria);
                $usuario->setProvincia($user->Municipio-
>Provincia->NombreProvincia);
                $usuario->setMunicipio($user->Municipio-
>NombreMunicipio);
                $usuario->setAptoUCI($user->Residencia-
>Apartamento);
                $usuario->setTelefono($user->Residencia-
>Telefono);
                $usuario->setFacultad($user->Area-
>NombreArea);
                $usuario->setCargo($user->Cargo-
>NombreCargo);
                $usuario->setArea($user->Area->NombreArea);
                $usuario->setFoto($user->Foto->UrlFoto);
                $roles = array();
                $rol = array();
                $roles[0] = "ROLE_USUARIO";
                foreach ($roles as $item) {
```

```
                    $este_rol = $em-
>getRepository('RolBundle:Rol')->findOneBy(array('nombre' =>
$item));
                    array_push($rol, $este_rol);
                }
                $usuario->setRol($rol);
                $em->persist($usuario);
            } else {
                $usuario = $usuario_resultado;
            }
            $entity->setUsuario($usuario);
            $entity->setEditadoPor($usuario_login);
            $entity->setFecha($fecha);
            $entity->setHora($hora);
            $entity->setEstado("Pendiente");
            $entity->setConciliado(FALSE);
            $entity->setRecurso($recurso);
            $entity->setBrigada($brigada);
            if ($especialidad->getNombre() == "Clima" or
$especialidad->getNombre() == "Electrohogar") {
                $copextel = $em-
>getRepository("BrigadaBundle:Brigada")-
>findOneBy(array('nombre' => "Copextel"));
                $entity->setBrigada($copextel);
            }
            $reporte = $em-
>getRepository('ReporteBundle:Reporte')-
>existeReporteNoSolucionado($local, $especialidad,
$afectacion, $recurso_id);
//Si no existe algún reporte sin solucionar con esos datos.
Asegura que no se duplique el reporte
            if (count($reporte) == 0 or
isset($_POST["forzar_reporte"])) {
                $em = $this->getDoctrine()->getManager();
                $em->persist($entity);
                $em->flush();
```

Anexo 5

Tarea	
Número de tarea: 6	**Número de HU: 6**
Nombre: Conciliar	
Tipo de tarea: desarrollo	Puntos de estimación: 1.5

Fecha inicio: 10/12/2014	Fecha fin: 19/12/2014
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite exportar a documento excel los reportes seleccionados.	

Tarea	
Número de tarea: 7	**Número de HU: 7**
Nombre: Gestionar local	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 13/01/2015	Fecha fin: 16/01/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar local	

Tarea	
Número de tarea: 8	**Número de HU: 8**
Nombre: Gestionar instalación	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 19/01/2015	Fecha fin: 22/01/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar instalación	

Tarea	
Número de tarea: 9	**Número de HU: 9**
Nombre: Gestionar área	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 23/01/2015	Fecha fin: 27/01/2015

Programador responsable: Gabriel Hurtado y Yoan Pérez
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar área

Tarea	
Número de tarea: 10	**Número de HU: 10**
Nombre: Gestionar brigada	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.5
Fecha inicio: 27/01/2015	Fecha fin: 30/01/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar brigada	

Tarea	
Número de tarea: 11	**Número de HU: 11**
Nombre: Gestionar recurso	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 30/01/2015	Fecha fin: 10/02/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar recurso	

Tarea	
Número de tarea: 12	**Número de HU: 12**
Nombre: Gestionar afectación	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 10/02/2015	Fecha fin: 13/02/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	

Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar afectación

Tarea	
Número de tarea: 13	**Número de HU: 13**
Nombre: Gestionar especialidad	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 16/02/2015	Fecha fin: 19/02/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite adicionar, mostrar, editar y eliminar especialidad	

Tarea	
Número de tarea: 14	**Número de HU: 14**
Nombre: Importar excel	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 19/02/2015	Fecha fin: 23/02/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite importar un documento excel	

Tarea	
Número de tarea: 15	**Número de HU: 15**
Nombre: Graficar datos estadísticos	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.5
Fecha inicio: 24/02/2015	Fecha fin: 27/02/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite mostrar un gráfico	

Tarea	
Número de tarea: 16	**Número de HU: 16**
Nombre: Asignar rol	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.5
Fecha inicio: 27/02/2015	Fecha fin: 2/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite asignar rol dependiendo de la función del usuario	

Tarea	
Número de tarea: 17	**Número de HU: 17**
Nombre: Notificar por correo	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 3/03/2015	Fecha fin: 6/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite enviar una notificación al correo una vez hecho un reporte o modificado el mismo	

Tarea	
Número de tarea: 18	**Número de HU: 18**
Nombre: Filtrar reporte	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 9/03/2015	Fecha fin: 11/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por reporte	

Tarea

Número de tarea: 19	**Número de HU: 19**
Nombre: Filtrar especialidad	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 16/03/2015	Fecha fin: 19/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por especialidad	

Tarea	
Número de tarea: 20	**Número de HU: 20**
Nombre: Filtrar afectación	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 19/03/2015	Fecha fin: 23/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por afectación	

Tarea	
Número de tarea: 21	**Número de HU: 21**
Nombre: Filtrar recurso	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 23/03/2015	Fecha fin: 26/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por recurso	

Tarea	
Número de tarea: 22	**Número de HU: 22**

Nombre: Filtrar brigada	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 26/03/2015	Fecha fin: 31/03/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por brigada	

Tarea	
Número de tarea: 23	**Número de HU: 23**
Nombre: Filtrar área	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 1/04/2015	Fecha fin: 6/04/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por área	

Tarea	
Número de tarea: 24	**Número de HU: 24**
Nombre: Filtrar instalación	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 6/04/2015	Fecha fin: 9/04/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por instalación	

Tarea	
Número de tarea: 25	**Número de HU: 25**

Nombre: Filtrar local	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 9/04/2015	Fecha fin: 13/04/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite filtrar por local	

Tarea	
Número de tarea: 26	**Número de HU: 26**
Nombre: Exportar gráfico	
Tipo de tarea: desarrollo	Puntos de estimación: 0.5
Fecha inicio: 13/04/2015	Fecha fin: 16/04/2015
Programador responsable: Gabriel Hurtado y Yoan Pérez	
Descripción: Se implementa la funcionalidad que permite exportar a formato pdf, png, jpg, svg	

Anexo 6

Caso de prueba de aceptación.	
Código: P8	**No. de HU: 5**
Nombre: Conciliar	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El usuario autenticado selecciona la opción Conciliar del menú de opciones en la interfaz principal.	
Resultado esperado: Se muestra un documento excel con los datos de la conciliación.	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P9	**No. de HU: 6**
Nombre: Exportar	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución:	
Resultado esperado:	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P10	**No. de HU: 7**
Nombre: Crear área	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Área luego introducen los datos correctamente para crear una nueva área El sistema muestra el mensaje "El área fue creada correctamente".	
Resultado esperado: El sistema crea el área y muestra el mensaje "El área fue creada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P11	**No. de HU: 7**
Nombre: Editar área	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	

Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos del área. Los campos Área y Descripción son obligatorios a la hora de editar un área. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"
Evaluación de la prueba: Satisfactoria.

Caso de prueba de aceptación.	
Código: P12	**No. de HU: 7**
Nombre: Eliminar área	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente al área que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar el área?" seleccionan la opción Aceptar. El sistema muestra el mensaje "El área ha sido eliminada correctamente".	
Resultado esperado: El sistema elimina el área seleccionada y muestra el mensaje "El área ha sido eliminada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P13	**No. de HU: 9**
Nombre: Crear recurso	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	

Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Recurso, luego introducen los datos correctamente para crear una nuevo recurso. El sistema muestra el mensaje "El recurso fue creado correctamente".
Resultado esperado: El sistema crea el recurso y muestra el mensaje "El recurso fue creado correctamente".
Evaluación de la prueba: Satisfactoria.

Caso de prueba de aceptación.	
Código: P14	**No. de HU: 9**
Nombre: Editar recurso	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos del recurso. Los campos Recurso y Descripción son obligatorios a la hora de editar un recurso. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".	
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P15	**No. de HU: 9**
Nombre: Eliminar recurso	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente a al recurso que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar el recurso?"	

seleccionan la opción Aceptar. El sistema muestra el mensaje "El recurso ha sido eliminada correctamente".
Resultado esperado: El sistema elimina el recurso seleccionado y muestra el mensaje "El recurso ha sido eliminado correctamente".
Evaluación de la prueba: Satisfactoria.

Caso de prueba de aceptación.	
Código: P16	**No. de HU: 10**
Nombre: Crear afectación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Afectación, luego introducen los datos correctamente para crear una nueva afectación. El sistema muestra el mensaje "La afectación fue creada correctamente".	
Resultado esperado: El sistema crea la afectación y muestra el mensaje "La afectación fue creada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P17	**No. de HU: 10**
Nombre: Editar afectación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos de la afectación. Los campos Afectación y Descripción son obligatorios a la hora de editar una afectación. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".	

Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"
Evaluación de la prueba: Satisfactoria.

Caso de prueba de aceptación.	
Código: P18	**No. de HU: 10**
Nombre: Eliminar afectación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente a la afectación que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar la afectación?" seleccionan la opción Aceptar. El sistema muestra el mensaje "La afectación ha sido eliminada correctamente".	
Resultado esperado: El sistema elimina la afectación seleccionada y muestra el mensaje "La afectación ha sido eliminada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P19	**No. de HU: 11**
Nombre: Crear especialidad	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Especialidad, luego introducen los datos correctamente para crear una nueva especialidad. El sistema muestra el mensaje "La especialidad fue creada correctamente".	
Resultado esperado: El sistema crea la especialidad y muestra el mensaje "La especialidad fue creada correctamente".	

Evaluación de la prueba: Satisfactoria.

Caso de prueba de aceptación.	
Código: P20	**No. de HU: 11**
Nombre: Editar especialidad	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos de la especialidad. Los campos Especialidad y Descripción son obligatorios a la hora de editar una especialidad. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".	
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P21	**No. de HU: 11**
Nombre: Eliminar especialidad	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente a la especialidad que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar la especialidad?" seleccionan la opción Aceptar. El sistema muestra el mensaje "La especialidad ha sido eliminada correctamente".	
Resultado esperado: El sistema elimina la especialidad seleccionada y muestra el mensaje "La especialidad ha sido eliminada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P22	**No. de HU: 12**
Nombre: Crear local	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Local, luego introducen los datos correctamente para crear uno nuevo local. El sistema muestra el mensaje "El local fue creado correctamente".	
Resultado esperado: El sistema crea el local y muestra el mensaje "El local fue creado correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P23	**No. de HU: 12**
Nombre: Editar local	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos del local. Los campos Local y Descripción son obligatorios a la hora de editar un local. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".	
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P24	**No. de HU: 12**
Nombre: Eliminar local	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente a al local que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar el local?" seleccionan la opción Aceptar. El sistema muestra el mensaje "El local ha sido eliminado correctamente".	
Resultado esperado: El sistema elimina el local seleccionado y muestra el mensaje "El local ha sido eliminado correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P25	**No. de HU: 13**
Nombre: Crear instalación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Instalación, luego introducen los datos correctamente para crear una nueva instalación. El sistema muestra el mensaje "La instalación fue creada correctamente".	
Resultado esperado: El sistema crea la instalación y muestra el mensaje "La instalación fue creada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.

Código: P26	**No. de HU: 13**
Nombre: Editar instalación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Editar para modificar los datos de la instalación. Los campos Instalación y Descripción son obligatorios a la hora de editar una instalación. Dan clic en el botón Actualizar, el sistema actualiza los datos y muestra el mensaje "Los datos fueron actualizados correctamente".	
Resultado esperado: El sistema actualiza los datos que hayan sido modificados y muestra un mensaje "Los datos fueron actualizados correctamente"	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P27	**No. de HU: 13**
Nombre: Eliminar instalación	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Borrar correspondiente a la instalación que desea eliminar. Cuando dan clic en borrar el sistema muestra el mensaje "¿Seguro que desea eliminar la instalación?" seleccionan la opción Aceptar. El sistema muestra el mensaje "La instalación ha sido eliminada correctamente".	
Resultado esperado: El sistema elimina la instalación seleccionada y muestra el mensaje "La instalación ha sido eliminada correctamente".	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.

Código: P28	**No. de HU: 14**
Nombre: Importar excel	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El usuario presiona el botón Importar excel, presiona la opción examinar, luego selecciona el archivo xls que desea importar (se recomienda que en fichero sea un fichero previamente exportado y modificado), presiona el botón Abrir y luego el botón Importar.	
Resultado esperado: Se crean todos los reportes existentes en el documento excel si sus datos son correctos y si la celda correspondiente al id está vacía o se actualizan los reportes si la celda correspondiente al id contiene el id del reporte y dicho id existe, también crea las áreas, instalaciones, locales, brigadas y recursos si no existen en la base de datos, así como sus dependencias. Se muestra una alerta de resultado satisfactorio con la cantidad de adiciones y de actualizaciones realizadas o una alerta de error si ocurrió algún percance al importar.	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P29	**No. de HU: 15**
Nombre: Graficar	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Técnico general o Administrador autenticado.	
Pasos de ejecución: El administrador o el técnico general del sistema acceden a la opción Graficar, luego dan clic en Usuarios, dan clic en el botón Generar estadísticas, posteriormente seleccionan el período y llenan los campos Fecha de inicio y Fecha final y dan clic en el botón Generar gráfico. El sistema muestra un gráfico con la cantidad de reportes de los usuarios en las fechas en los que se reportaron afectaciones.	
Resultado esperado: El sistema muestra un gráfico con la cantidad de reportes realizados por usuarios en las fechas que se reportaron afectaciones.	
Evaluación de la prueba: Satisfactoria.	

Caso de prueba de aceptación.	
Código: P30	**No. de HU: 16**
Nombre: Asignar rol	
Descripción: Prueba para la funcionalidad de conciliar	
Condiciones de ejecución: Administrador autenticado.	
Pasos de ejecución: El usuario presiona el botón Asignar rol, selecciona la acción asignar rol del usuario al que desee asignarle el rol, selecciona el rol que desea asignar y presiona el botón Asignar rol.	
Resultado esperado: Se asigna el rol al usuario, se muestran los datos del usuario al que le fue asignado el rol y se muestra una alerta de resultado satisfactorio.	
Evaluación de la prueba: Satisfactoria.	

Anexo 7

Encuesta sobre reportes de mantenimiento constructivo en la Universidad de las Ciencias Informáticas.

- ¿Conoces el número telefónico al que puedes llamar para reportar las afectaciones de mantenimiento constructivo?

  __Sí
  __No
  En caso afirmativo poner el número: _____

- ¿Conoces dónde radica la oficina que atiende los reportes de mantenimiento constructivo?
  __Sí
  __No
  En caso afirmativo poner: Edificio____ Apartamento____

- Sería más efectivo realizar reportes mediante:

  __una aplicación web
  __vía telefónica
  __ambos

- ¿Recibes respuesta a tus reportes?
  - __Sí
  - __No
  - __En algunas ocasiones

- Selecciona la manera que crea más efectiva para recibir respuesta a sus reportes:

  - __Llamada telefónica
  - __Correo electrónico
  - __Llamar al técnico que recibió tu reporte
  - __Mediante una aplicación web
  - __Mediante una aplicación web y una notificación al correo.