

FACULTAD  
INFORMÁTICA MATEMÁTICA

DPTO. INGENIERÍA INFORMÁTICA

# SISTEMA DE GESTIÓN DE REPORTE DE INCIDENCIAS PARA EL DEPARTAMENTO DE SOPORTE TÉCNICO DE LA UNIVERSIDAD DE HOLGUÍN

TESIS PRESENTADA EN OPCIÓN  
AL TÍTULO DE INGENIERO INFORMÁTICO

Autor: Antonio Baptista Miguel

HOLGUÍN 2019



FACULTAD  
INFORMÁTICA MATEMÁTICA

DPTO. INGENIERÍA INFORMÁTICA

# SISTEMA DE GESTIÓN DE REPORTE DE INCIDENCIAS PARA EL DEPARTAMENTO DE SOPORTE TÉCNICO DE LA UNIVERSIDAD DE HOLGUÍN

TESIS PRESENTADA EN OPCIÓN  
AL TÍTULO DE INGENIERO INFORMÁTICO

Autor: Antonio Baptista Miguel

Tutores: M.Sc. Hamlet Nuñez Gumila

Ing. Allen M. Jiménez González

HOLGUÍN 2019



PENSAMIENTO

*“Se quiere más aquello que se ha conseguido con muchas  
fatigas.”*

*Aristóteles*

## DEDICATORIA

Mi dedicatoria primeramente va dirigida a Dios todo poderoso principalmente, por permitirme este logro y sueño que a la vez parecía inalcanzable, y gracias a su amor, bendición y poder de espíritu pude conseguir, gracias señor.

*A la memoria de mi mamá “María Daconceção Massanga” quien le debo todo lo que soy en reconocimiento de su amor, esfuerzo y sacrificio, a mi padre João Tónico por el apoyo incondicional que me han dado hasta ahora, por la atención, amor, cariño, aun estando lejos de mí.*

A mis hermanos y hermanas “Francisco, João, Rosa, Guigúí, Mádo” y toda la familia gracias por estar presente en mi vida.

A mi querida esposa “Elizabeth Berrillo Espronceda y mi hijo Anthony Baptista Miguel Berrillo “quienes estuvieron pendientes de mí, que comparten amor alegrías, son mi iluminación, mi ejemplo de lucha y constancia para continuar intentando ser cada día mejor.

## AGRADECIMIENTOS

Primeramente, quisiera agradecer a Dios todo poderoso principalmente, por permitirme este logro y sueño que a la vez parecía inalcanzable, y gracias a su amor, bendición y poder de espíritu pude conseguir, gracias señor.

*A mis tutores Hamlet Nuñez Gumila y Allen M. Jiménez González, por tener en todo momento preocupación y dedicación por mi trabajo, quienes no escatimaron esfuerzos para que esta tesis concluyera lo mejor posible.*

*A mis padres, por toda la ayuda y apoyo brindado.*

*A mis enemigos y a los que me critican, por obligarme a perfeccionarme cada día.*

*A los de verdad, los que no fingen la amistad, que saben que los quiero hasta más infinito.*

*A todos los que me han brindado su ayuda incondicional y constantemente han estado preocupados por la prosperidad de este trabajo.*

*A mis profesores, de los que tanto he aprendido y de los que he visto clases magistrales como las de Reynol Solórzano y las de Leydis Lamothe.*

*A mis compañeros de aula, por brindarme su amistad desinteresadamente.*

*A los Informáticos todos...*

## RESUMEN

La actual investigación se realizó en el departamento de Soporte técnico de la Universidad de Holguín donde la gestión de reportes de incidencias y mantenimientos se efectúa para que los técnicos puedan ofertar un mejor servicio a todo el equipamiento de la entidad. Hasta el momento dicho proceso se lleva a cabo de forma manual pues no existe un producto informático que apoye esta tarea donde el jefe del departamento de soporte técnico es el encargado de distribuir incidencias y el mantenimiento. El objetivo del presente trabajo es desarrollar un sistema de reportes de incidencias mediante la cual se logre gestionar los reportes de incidencias y el mantenimiento de tal forma que se garantice una mejor planificación de reportes de incidencias y los mantenimientos en la entidad. Teniendo como problemática la situación anteriormente planteada se pretende obtener como resultado una aplicación que facilite la gestión de reportes de incidencias y el procesamiento de datos que esta conlleva para que se convierta en un producto sostenible. En la presente investigación se describe el proceso mediante el cual se obtuvo como resultado dicho sistema de gestión de reportes de incidencias. Para el desarrollo de un sistema de gestión de reportes de incidencias se emplearon diversas tecnologías en concordancia con las limitaciones que impuso el sistema; entre las que se encuentran la metodología de desarrollo de *software* XP, el lenguaje de programación PHP, framework Yii, el Sistema Gestor de Base de Datos MySQL y el servidor de aplicaciones Web Apache.

## ABSTRACT

The current investigation was carried out in the technical support department of the University of Holguin where the management of incidents and maintenance reports is carried out so that the technicians can offer a better service to all the equipment of the entity. So far this process is carried out manually because there is no computer product to support this task where the head of the technical support department is responsible for distributing incidents and maintenance. The objective of this paper is to develop a system of incident reports through which incident reports and maintenance can be managed in such a way as to guarantee a better planning of incident reports and maintenance in the entity. Having as problematic the situation previously raised is intended to obtain as a result an application that facilitates the management of incident reports and the processing of data that this entails so that it becomes a sustainable product. The present investigation describes the process by which said incident management system was obtained as a result. For the development of a management system for incident reports, various technologies were used in accordance with the limitations imposed by the system; among which are the XP software development methodology, the PHP programming language, framework Yii, the MySQL Database Management System and the Apache Web application server.

## TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1. MARCO TEÓRICO.....	7
1.1 FUNDAMENTOS TEÓRICOS DEL PROCESO DE REGISTRO Y CONTROL DE INCIDENCIAS DE SOPORTE TÉCNICO .....	7
1.1.1 Clasificación de Incidente .....	8
1.1.2 Proceso de gestión de registro de incidentes .....	10
1.2 PROCESO DE REGISTRO Y CONTROL DE INCIDENCIAS DE SOPORTE TÉCNICO EN LA UNIVERSIDAD DE HOLGUÍN.....	11
1.3 SISTEMAS INFORMÁTICOS PARA EL REGISTRO Y CONTROL DE INCIDENCIAS .....	14
1.4 TECNOLOGÍAS Y HERRAMIENTAS .....	18
ARQUITECTURA CLIENTE/SERVIDOR .....	19
1.4.1 Lenguajes de programación .....	21
1.4.2 Framework Yii .....	22
1.4.3 Servidor Apache.....	23
1.5 METODOLOGÍAS DE DESARROLLO DE SOFTWARE .....	25
1.5.1 Metodología RUP .....	26
1.5.2 Metodología ICONIX.....	26
1.5.3 Metodología de desarrollo de software XP .....	26
CONCLUSIONES DEL CAPÍTULO.....	30
CAPÍTULO 2: DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA.....	31
2.1 FASE DE EXPLORACIÓN.....	31
2.1.1 Descripción de requisitos .....	31
2.1.2 Historias de usuario .....	34
2.2 PLANIFICACIÓN.....	39
2.2.1 Iteración .....	39
2.2.2 Plan de entregas .....	40
2.3 IMPLEMENTACIÓN .....	41
2.3.1 Primera Iteración.....	41
2.3.2 Segunda Iteración .....	43
2.3.3 Tercera Iteración .....	44
2.3.4 Cuarta Iteración.....	45
2.3.5 Quinta Iteración.....	46
2.4 DIAGRAMA DE DESPLIEGUE.....	47
2.5 VALORACIÓN DE SOSTENIBILIDAD DE LA PROPUESTA .....	48
2.6 PRUEBAS .....	51

2.6.1 Pruebas de aceptación .....	51
CONCLUSIONES DEL CAPÍTULO.....	55
CONCLUSIONES GENERALES.....	56
RECOMENDACIONES .....	57
REFERENCIAS BIBLIOGRÁFICAS.....	58
GLOSARIO DE TÉRMINOS.....	1
ANEXO 1: HISTORIAS DE USUARIO .....	I
ANEXO 2: TAREAS.....	II
ANEXO 3: CASOS DE PRUEBA DE ACEPTACIÓN .....	V

## ÍNDICE DE TABLAS

Tabla 1: Historia de Usuario # 1 "Gestionar elemento de reporte" .....	35
Tabla 2: Historia de Usuario # 2 "Gestionar equipo" .....	35
Tabla 3: Historia de Usuario # 3 "Gestionar incidencias" .....	36
Tabla 4: Historia de Usuario # 4 "Gestionar plan de mantenimiento por equipo".....	36
Tabla 5: Historia de Usuario # 5 "Gestionar mantenimientos realizados a los equipos" .....	37
Tabla 6: Historia de Usuario # 6 "Gestionar solicitud de mantenimiento" .....	37
Tabla 7: Historia de Usuario # 7 "Reportar como atendido la notificación de incidencia" .....	38
Tabla 8: Historia de Usuario # 8 "Mostrar incidencia por atender" .....	38
Tabla 9: Historia de Usuario # 9 "Gestionar planes por realizar" .....	39
Tabla 10: Distribución de las historias de usuario por iteración.....	40
Tabla 11: Cronograma de entregas.....	41
Tabla 12: Tarea # 1 "Insertar datos del elemento de reporte" .....	41
Tabla 13: Tarea # 2 "Editar elemento de reporte" .....	42
Tabla 14: Tarea # 3 "Visualizar datos de la Gestión de elemento de reporte" ..	42
Tabla 15: Tarea # 4 "Eliminar datos de la Gestión de elemento de reporte" ..	43
Tabla 16: Tarea # 5 "Insertar equipo" .....	43
Tabla 17: Tarea # 6 "Modificar equipos" .....	44
Tabla 18: Tarea # 7 "Eliminar equipo" .....	44
Tabla 19: Tarea # 8 "Mostrar incidencia por atender" .....	45
Tabla 20: Tarea # 9 "Gestionar planes por realizar" .....	45
Tabla 21: Tarea # 10 "Gestionar reporte por departamento" .....	46
Tabla 22: Tarea # 11 "Gestionar locales" .....	46

Tabla 23: "Caso de prueba de aceptación HU2_P1" .....	52
Tabla 24: "Caso de prueba de aceptación HU2_P2" .....	52
Tabla 25: "Caso de prueba de aceptación HU3_P3" .....	53
Tabla 26: "Caso de prueba de aceptación HU4_P4" .....	53
Tabla 27: "Caso de prueba de aceptación HU4_P5" .....	54
Tabla 28: "Caso de prueba de aceptación HU4_P6" .....	54

## ÍNDICE DE FIGURAS

Figura 1: Clasificación de incidente.....	9
Figura 2: Diagrama de prioridades.....	10
Figura 3:Funcionamiento del Lenguaje PHP.....	22
Figura 4: Diagrama de despliegue .....	47

## INTRODUCCIÓN

La Universidad de Holguín tiene la responsabilidad de satisfacer las necesidades sociales mediante la formación integral y continúa de profesionales en las ciencias exactas, económicas, técnicas, agropecuarias, jurídicas, sociales y humanísticas, aportando resultados científico-técnicos relevantes y la extensión de su acción a la comunidad y al país. Cuenta para ello con cuatro sedes, la sede Oscar Lucero Moya (OLM) donde radican las facultades de Ciencias Económicas y Administración, Ingeniería, Ingeniería Industrial y Turismo, Informática y Matemática. La sede Celia Sánchez Manduley (CSM) con las facultades de Ciencias Humanísticas, Ciencias Sociales y Jurídicas. La sede Manuel Fajardo Rivero (MFR) con la facultad de Cultura Física y Deporte y la sede José de la Luz y Caballero (JLC) donde radican las facultades de Educación Infantil, Psicopedagogía y Arte, Educación Media Superior, Ciencias Agropecuarias. Cuenta la universidad además con 10 Centro Universitarios Municipales (CUM) y 3 Filiales Universitarias Municipales (FUM).

Durante el día a día de los departamentos de la universidad de Holguín pueden ocurrir numerosas incidencias que impidan a los usuarios desarrollar su actividad laboral correctamente provocando que los departamentos no funcionen de la manera adecuada. Estas incidencias pueden ser muy variadas, desde un fallo de un programa o con las conexiones, un problema hardware como la avería de un PC o una impresora hasta una petición para dar de alta a un nuevo usuario en el sistema o para solicitar derechos de accesos para unas aplicaciones o carpetas. Sin una herramienta de gestión de incidencias adecuada la resolución de estas incidencias podría llevar mucho tiempo lo que podría provocar graves pérdidas para los departamentos lo cual es del todo inviable. Desde hace unos años cada vez es más frecuente ver en las empresas numerosas aplicaciones online. Estas aplicaciones normalmente están instaladas en un servidor de la empresa integrado dentro de la Intranet de la misma, por lo que consecuentemente para poder utilizarla no es necesario instalarla en el PC del usuario, sino bastaría con una conexión al ejecutable del servidor. Este funcionamiento hace mucho más sencillo y práctico el funcionamiento de estas herramientas y facilita enormemente su administración ya

que al estar centralizada permiten su correcto funcionamiento tanto desde la Intranet de las empresas como desde cualquier PC exterior con conexión a la misma. Otro punto favorable es el bajo coste necesario para el buen funcionamiento de estas aplicaciones y además para el usuario es mucho más fácil manejarse con ellas ya que cada día los usuarios están más acostumbrados a manejarse en Internet. Actualmente en las Universidades para dar de alta una incidencia los usuarios lo hacen vía telefónica llamando normalmente un departamento de soporte técnico, el cual da de alta las incidencias y, a partir de ese punto, el usuario pierde total conocimiento sobre qué está pasando con su incidencia lo que provoca malestar e inseguridad en él (TRILLO, 2007).

La **asistencia técnica** o **soporte técnico** en el sector de las tecnologías de información de comunicación (TIC) es un rango de servicios por medio del cual se proporciona asistencia a los usuarios al tener algún problema al utilizar un producto o servicio, ya sea este el hardware o software de una computadora, periféricos, artículos electrónicos, maquinaria, o cualquier otro sistema informático. Soporte técnico o simplemente servicio de asistencia es llevado a cabo por un técnico de mantenimiento informático que debe asistir a los usuarios a distancia para solucionar los problemas con los que se encuentran al usar herramientas informáticas, especialmente cuando el dispositivo de cómputo no funciona. Este servicio se brinda de manera remota (generalmente por teléfono). Cuando un técnico de soporte trabaja en un centro de atención al cliente (también conocido como departamento de soporte técnico) puede asistir a clientes de compañías diferentes.

Usualmente, el técnico de soporte solo se ocupa de problemas de "nivel 1". Esto significa que anota el nombre de la persona que llama y el problema que tiene. Luego, ingresa estos datos en la base de datos para consultas. Con la ayuda de una base de conocimientos en la que aparecen las preguntas y respuestas más frecuentes, realiza un diagnóstico y trata de encontrar una solución dentro de un plazo estipulado por el empleador. También se puede decir que el soporte técnico es un rango de servicios que proporcionan asistencia con el hardware o software de una computadora, o algún otro dispositivo electrónico. En general los servicios de

soporte técnico tratan de ayudar al usuario a resolver determinados problemas con algún producto en vez de entrenar o personalizar.

La mayoría de las compañías que venden hardware o software ofrecen soporte técnico de manera telefónica o en línea. Las instituciones y compañías por lo general tienen sus propios empleados de soporte técnico (Silva 2018, Vialfa 2018).

La Universidad de Holguín tiene su propio departamento de soporte técnico, el cual cumple con el objetivo de proyectar tareas de soporte técnico y mantenimiento de software y hardware. El mantenimiento a los equipos de cómputo y software se realiza en conjunto con la Dirección de Informatización. Esta última con posesión exclusiva de las Unidades de Procesamiento Central (CPU). Aunque compartiendo las reparaciones de componentes adicionales tales como monitores, impresoras, mouse y teclados con el Departamento de soporte técnico que pueden ser realizadas por ambos según disponibilidad de fuerza de trabajo y recursos.

En la actualidad recibe las solicitudes de las diferentes estructuras universitarias, dichas solicitudes son evaluadas y si se cuentan con todos los recursos necesarios para su ejecución, se emite una orden de trabajo donde se recoge una serie de datos como el listado de los recursos que se van a utilizar, el operario que va a ejecutar, la descripción del mantenimiento que se va a realizar, entre otros (Jorge, 2017).

Este departamento surge para controlar el trabajo realizado por los operarios en el día y el control de recursos materiales extraídos del almacén (Jorge, 2017). Se detectaron ciertas deficiencias en el proceso de reporte de incidencia, entre las que se encuentran:

- Desactualización de la documentación
- Lentitud en la búsqueda de datos
- Lentitud en la elaboración de los planes de mantenimiento
- Lentitud en la elaboración de informes
- Pérdida y deterioro de documentos

Teniendo en cuenta estos aspectos se puede definir como **problema de la investigación**: ¿Cómo mejorar el proceso de reporte de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín?

Partiendo del problema se delimita como **objeto de estudio**: El proceso de reporte de incidencias.

Para dar solución a esta problemática se determina el siguiente **objetivo** para la investigación: Desarrollar un sistema de gestión de información que mejore el proceso de reporte de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín.

El **campo de acción**: Informatización del proceso de reporte de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín.

Para guiar la investigación se trazaron las **preguntas científicas** siguientes:

- ¿Cuáles son los fundamentos teóricos del proceso de reporte de incidencias?
- ¿Cuál es el estado de los sistemas informáticos de registro y control de incidencias?
- ¿Cómo desarrollar un sistema de gestión de información que mejore el proceso de reporte de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín?
- ¿Será sostenible el sistema desarrollado?
- ¿Cómo valorar el grado de aceptación del sistema desarrollado?

Para cumplir el objetivo propuesto, se plantean estas **tareas de la investigación**:

1. Elaborar los fundamentos teóricos del proceso de registro y control de incidencias.
2. Investigar las especificidades del proceso de registro y control de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín.
3. Estudiar los sistemas informáticos de registro y control de incidencias a nivel internacional.
4. Desarrollar un sistema de gestión de información que mejore el proceso de reporte de incidencias en el Departamento de Soporte Técnico de la Universidad de Holguín.
5. Valorar la sostenibilidad del sistema desarrollado.
6. Valorar el grado de aceptación del sistema mediante pruebas de aceptación.

Para lograr el objetivo planteado en la investigación, se utilizaron diferentes métodos, tanto teóricos como empíricos, los cuales se mencionan a continuación.

### **MÉTODOS TEÓRICOS**

**Análisis y síntesis:** para el análisis del objeto de estudio y así comprender las relaciones esenciales y características generales del proceso.

**Histórico – Lógico:** para el estudio del desarrollo histórico de los procesos de mantenimiento a software.

**Modelación:** para el diseño de la biblioteca de clases propuesta a través de los diagramas de la metodología XP y de esta forma desarrollar un sistema de gestión de reportes de incidencias para el departamento de soporte técnico de la Universidad de Holguín.

**Inductivo – deductivo:** para el diseño y aplicación del procedimiento para la ejecución del mantenimiento al producto software.

### **MÉTODOS EMPÍRICOS**

**Observación:** empleado esencialmente para llevar a cabo la implantación y diagnóstico del resultado obtenido de esta investigación.

**Entrevista:** análisis de sugerencias, criterios y necesidades que pudieron influir en la correcta concepción de las soluciones, de ahí que se aplicara este método para obtener información, búsqueda de opiniones y conocimientos de expertos; además de recopilar elementos para el análisis del sistema.

**Revisión de documentos:** empleado para lograr un mejor entendimiento del proceso mediante el estudio de bibliografía especializada.

El informe de la investigación consta de introducción, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas, glosario de términos y anexos.

Capítulo 1. Marco teórico: aborda los fundamentos teóricos que sustentan la investigación, además de una descripción de las herramientas necesarias para dar cumplimiento a los objetivos trazados en esta investigación y se abordan las metodologías seleccionadas para el desarrollo del software.

Capítulo 2. Descripción y elaboración de la solución propuesta: describe el proceso de desarrollo del sistema propuesto a través de la metodología seleccionada en el Capítulo 1, y se analizan aspectos como el estudio de sostenibilidad del producto,

se describe el sistema. se describe el proceso ingeniería que permite dar solución al problema planteado y dar cumplimiento así al objetivo de la investigación, se documenta el proceso de ingeniería de software que incluye desde la fase de análisis hasta la fase de pruebas.

## **CAPÍTULO 1. MARCO TEÓRICO**

En el presente capítulo se realiza un análisis de los fundamentos teóricos que sirven de base para la presente investigación, donde se reflejan los conceptos fundamentales en los cuales se basa el desarrollo del sistema de gestión de reportes de incidencias para el departamento de soporte técnico en la Universidad de Holguín. Se sistematizan los fundamentos que guían la investigación, con el objetivo de establecer y comprender mejor los elementos teóricos que se relacionan con la gestión de desarrollo del sistema. Se relacionan las tendencias y tecnologías actuales escogidas para el desarrollo del sistema, así como una valoración de las metodologías de desarrollo de *software* con énfasis en lo seleccionado.

### **1.1 Fundamentos teóricos del proceso de registro y control de incidencias de soporte técnico**

El departamento de Soporte Técnico Empresarial proporciona asistencia técnica de tecnología de información (TI) y consultoría para organizaciones gubernamentales. El departamento Empresarial es responsable de dar soporte a la puerta de enlace para que el usuario acceda a la red del servidor y los sistemas de escritorio. Administran los sistemas de misión crítica utilizados por agencias, empresas, y las agencias más pequeñas que no tienen su propio personal de TI.

El Departamento de Soporte Técnico brinda servicios como los que se listan a continuación:

- **Servidor de red y gestión de software**

La administración del servidor y del software garantiza que la infraestructura de red del usuario esté funcionando y se ejecute mediante auditorías del sistema para identificar posibles problemas con el sistema informático; aplica actualizaciones del sistema operativo: instala, configura software nuevo y también implementa actualizaciones al software existente.

- **Gestión de usuarios de red**

La gestión de usuarios entrega la administración y gestión de los registros de cuentas de usuarios. Los administradores de usuarios de red tienen la capacidad de identificar y controlar el estado de los usuarios conectados a la red.

- **Gestión de seguridad de usuarios de red**

La Gestión de seguridad del usuario supervisa los sistemas para evitar amenazas y vulnerabilidades de seguridad de la información antes de que se conviertan en incidentes; realiza la gestión e implementación de parches; gestión de cortafuegos; Consultoría sobre medidas generales de seguridad.

- **Administración de dispositivos de red (escritorio, móvil, impresora, etc.)**

Administración de dispositivos de red proporciona la gestión del ciclo de vida de los dispositivos de los usuarios de la red, incluida la consultoría sobre la provisión de dispositivos, la instalación y configuración, la protección antivirus, la supervisión y el mantenimiento, la desactivación y la preparación de dispositivos para el excedente.

- **Acceso remoto**

Proporciona a los usuarios autorizados acceso encriptado y seguro a las aplicaciones y servidores estatales desde ubicaciones remotas.

- **Email**

Proporciona configuración y administración de cuentas, filtrado de SPAM, protección antivirus y asistencia al usuario final.

- **Mesa de ayuda**

El único punto de contacto para las ofertas o problemas de los Servicios profesionales.

### 1.1.1 Clasificación de Incidente

El registro inicial debe ser adecuado para asignar a la incidencia un código de categorización. Esto será importante más adelante, cuando se revisen tipos de incidencias para fijar prioridades y posteriormente aplicarlos en los procesos de gestión de reporte de incidencias y otras actividades de la gestión de servicios de Tecnología de Información(TI). Tener en cuenta que la comprobación de solicitudes de servicio en este proceso no implica que las solicitudes de servicio sean incidentes. Esto es simplemente el reconocimiento del hecho de que las solicitudes de servicio a veces se registran incorrectamente como incidentes (por ejemplo, un usuario entra de forma incorrecta la solicitud como un incidente del sistema)(GOMERO, 2017).

Para poder determinar un nivel de prioridad cuando existan múltiples incidencias concurrentes es necesario asignar una prioridad para la resolución de las mismas como se muestra en la figura 1. Los parámetros en los que se basa el nivel de prioridad son:

- **Impacto:** El impacto determina la importancia del incidente dependiendo de cómo éste afecta a los procesos del mantenimiento y/o de la cantidad de usuarios afectados.
- **Urgencia:** La urgencia dependerá del tiempo máximo de demora que esté de acuerdo el cliente para resolver el incidente y/o el nivel de servicio convenido en el departamento.

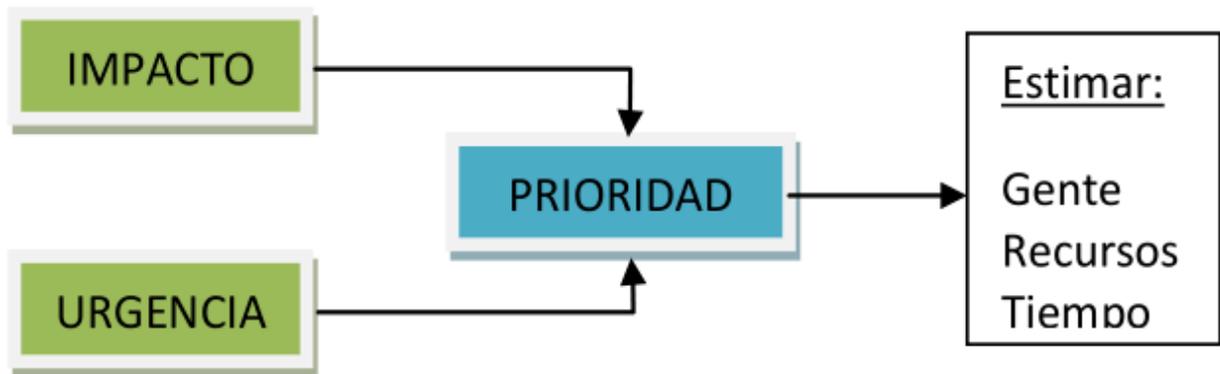


Figura 1: Clasificación de incidente.

Se debe tomar en cuenta factores auxiliares tales como el tiempo esperado para solucionar la incidencia y los recursos necesarios, dependiendo de la prioridad se asignarán los recursos necesarios para solucionar el incidente. Es conveniente establecer un procedimiento para determinar la prioridad del incidente como por ejemplo se puede diferenciar entre las prioridades (Huércano, 2017): Crítica, Alta, Media, Baja. En la figura 2 se indica un posible diagrama de prioridades en función de la urgencia e impacto del incidente.

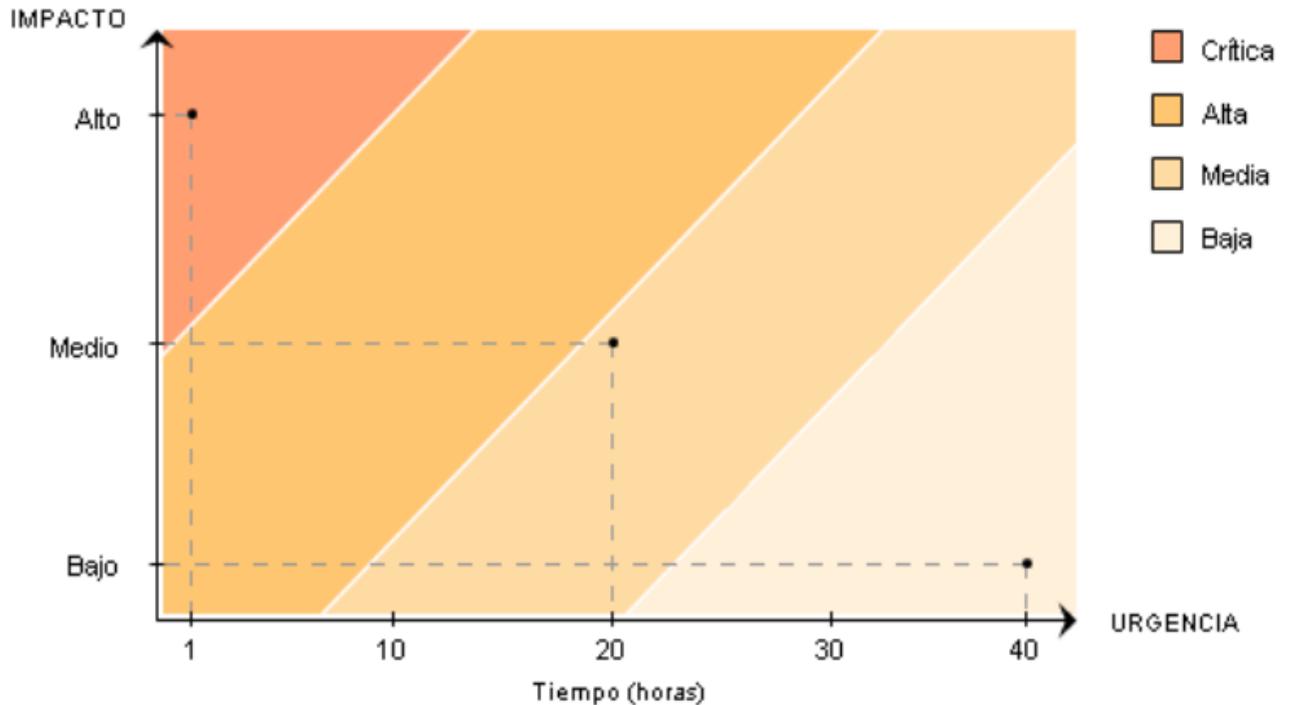


Figura 2: Diagrama de prioridades.

### 1.1.2 Proceso de gestión de registro de incidentes

Cuando un usuario reporta una incidencia a la mesa de ayuda, el responsable de la mesa de ayuda tiene el deber de hacer el registro del mayor número posible de indicios del incidente que se está reportando como parte del primer diagnóstico. También se tiene que hacer el intento de determinar la causa de lo que está fallando y de qué manera se podría corregir. En este contexto resultan de gran utilidad la información de los errores que ya son conocidos. De ser posible, el responsable de la mesa de ayuda debe resolver la incidencia inmediatamente y luego cerrarla. Si no puede resolverla, debe escalar la incidencia.

El proceso habitual de gestión de registro de incidentes comienza con la detección y registro del incidente, para lo cual se debe dar cuenta que el incidente proviene desde el usuario. El proceso de registro de la incidencia es el primer paso para una correcta gestión del mismo. Las incidencias pueden provenir de diversos orígenes tales como usuarios, gestión de aplicaciones, Centro de Servicios o en los departamentos de las Empresas, entre otros. El proceso del registro debe realizarse inmediatamente ya que resulta más costoso hacerlo posteriormente y se corre el

riesgo de que la aparición de nuevas incidencias demore indefinidamente el proceso (Huércano, 2017).

Se deberá comprobar que el incidente aún no ha sido registrado ya que es probable que más de un usuario notifique la misma incidencia y por lo tanto se tiene que evitar duplicaciones innecesarias. Se tiene que asignar una referencia o número de caso único al incidente que lo identificará tanto en los procesos internos como en las comunicaciones con el cliente. El registro inicial debe ser ingresado en la base de datos asociada la información básica necesaria para el procesamiento del incidente como, por ejemplo: hora, descripción del incidente, sistemas afectados, etc. (Paul, 2011). Se solicitará información de apoyo la misma que se incluirá para la resolución del incidente que puede ser solicitada al cliente a través de un formulario específico, etc.

En los casos en que el incidente pueda afectar a otros usuarios estos deben ser notificados para que conozcan como esta incidencia puede afectar su flujo habitual de trabajo. Como siguiente paso se debe hacer la clasificación del incidente el mismo que tiene como objetivo principal el recopilar toda la información que pueda ser de utilizada para la resolución del mismo y se tiene que asignar una categoría dependiendo del tipo de incidente o del grupo de trabajo responsable de su resolución, se tiene que establecer el nivel de prioridad dependiendo del impacto y la urgencia para poder asignar recursos con el personal adecuado y responsable de su resolución y en este proceso también monitorear el estado y el tiempo de respuesta esperado (Figuerola, 2012).

## **1.2 Proceso de registro y control de incidencias de soporte técnico en la Universidad de Holguín**

Los procesos necesarios para crear un registro y controlar las incidencias son administrados por el departamento de soporte técnico, la recopilación de todo mantenimiento y de toda la información llega desde los departamentos de la entidad y las agencias vía correo electrónico, donde es procesada y se asigna un recurso humano para su atención. Esta área influye en gran medida en el cumplimiento de las tareas programadas para el logro de objetivos; además el volumen del mantenimiento y la información documental está en constante incremento.

En la Universidad de Holguín, existe un Departamento de Soporte Técnico que se encarga de prestar servicio de mantenimiento al equipamiento informático, el cual cumple con el objetivo de proyectar tareas de soporte técnico y mantenimiento de software y hardware. Tiene como tareas fundamentales:

1. Asesorar en la adquisición, actualización y mantenimiento del equipamiento informático.
2. Participar en la elaboración de políticas del uso y manejo de los recursos informáticos.
3. Brindar soporte técnico y asistencia a usuarios de equipos informáticos en el ámbito de la Universidad.
4. Documentar procesos y capacitar a los usuarios en el uso del recurso informático.
5. Asesorar a los usuarios y sus necesidades en la compra de software, hardware y dispositivos informáticos.
6. Reparar y poner en funcionamiento el equipamiento informático.
7. Atender las demandas de los clientes en el menor tiempo posible.
8. Proporcionar servicio de mantenimiento a equipos de cómputo rápido y eficiente en la comodidad del puesto de trabajo.
9. Mantener una comunicación constante con los clientes, dándoles la seguridad de que sus equipos funcionan de manera adecuada en cualquier momento.

Este departamento posee una estructura jerárquica que en lo vinculado al equipamiento consta de:

- Jefe del departamento: encargado de administrar y dirigir todos los procesos del departamento
- Especialista de equipos: persona que controla dichos medios y planifica los mantenimientos
- Técnicos: encargados de ejecutar los mantenimientos y las reparaciones.

Actualmente, el proceso de registro y control de incidencias se lleva de manera manual ocasionando graves problemas de mantenimiento, retrasos con el desarrollo de plan de mantenimiento, pérdida de información, descontento del

personal, mala imagen del departamento de Soporte Técnico dentro de la Universidad, imposibilidad de obtener estadísticas del trabajo y del soporte del área de Sistemas de manera rápida y precisa, entre otros. En la actualidad el departamento de Soporte Técnico recibe las solicitudes de las diferentes estructuras universitarias, dichas solicitudes son evaluadas y si se cuenta con los recursos necesarios para su ejecución, se emite una orden de plan de mantenimiento donde se recoge una serie de datos como el listado de piezas que se van a utilizar, el operario que la va a ejecutar, y realizar la descripción del mantenimiento, entre otros (Jorge, 2017).

La evaluación de los reportes se lleva a cabo por la parte técnica si el reporte abierto es de carácter interno o externo, siendo los de carácter interno todos aquellos reportes que son atendidos directamente por Departamento de Soporte técnico o grupo de trabajo y los externos todos aquellos reportes que deben ser atendidos por otra zona o grupo de trabajo o por una dependencia externa del Departamento. Los reportes de carácter interno constituidos en el servicio a restablecer o equipo a reparar pasan a ser automáticamente continuados por la parte técnica relacionada con el mismo, sea el propio técnico del departamento, el personal técnico del taller, brigadas móviles. Si el reporte es de carácter externo, se contactará con la dependencia o entidad encargada de reparar este servicio. Una vez reparado el servicio afectado o restablecido el servicio interrumpido, las dependencias antes mencionadas notificarán al supervisor de comunicaciones que el servicio interrumpido se encuentra restablecido y estarán disponibles para el chequeo del servicio en cuestión. Una vez resuelta la interrupción, el reporte debe cerrarse en consulta con quien lo generó y una vez recibida la aprobación del cliente, se procede a cerrar el reporte. Cerrado el reporte se procede a completar todos los datos inherentes a este en el Registro de Reportes. Un reporte está integrado por un identificador, el supervisor que lo abrió, el supervisor que lo cierra, la hora de apertura, la hora en que es cerrado, el tipo (interno, externo), el estado del reporte (en proceso, restablecido, pendiente a compra, servicio restablecido), si afecta servicio o no y una nota técnica. Una queja está formada por un identificador, el subsistema interrumpido o averiado, el departamento en el que se ha generado la

queja, quien la emitió, la fecha y la descripción de la misma. En la actualidad el departamento de soporte técnico de la Universidad de Holguín, intenta simular una mesa de ayuda a través de los sub departamentos de soporte técnico que se encuentra en otras sedes.

Los objetivos principales de la Gestión de incidencias son:

- detección y registro del incidente
- clasificación y soporte inicial
- investigación y diagnóstico
- solución y restablecimiento del servicio
- cierre del incidente
- monitorización, seguimiento y comunicación del incidente.

### **1.3 Sistemas informáticos para el registro y control de incidencias**

Los sistemas informáticos en las empresas son los que ayudan a lograr su correcto funcionamiento, se tiene muy en cuenta el papel esencial que juegan los sistemas informáticos en las empresas, la tramitación y la gestión de la información que esta maneja. Con el avance tecnológico, se ha incrementado indiscutiblemente el uso de las nuevas tecnologías en el desarrollado mundo de la computación, la electrónica y las comunicaciones; brindando infinitas facilidades en los entornos empresariales (Michelena, 2007).

A consecuencia de la gran significación que tienen los sistemas de informática y las comunicaciones en las empresas, la tramitación de las incidencias que ocurren en estos es de vital importancia para el restablecimiento de los servicios empresariales. La incidencia en un sistema de comunicación se define como “cualquier acontecimiento inesperado que pueda dar lugar a un período operacionalmente importante, durante el cual una instalación, equipo o sistema no facilite servicio dentro de las tolerancias especificadas” (Michelena, 2007).

Glenis González Cedeño afirma que la tramitación a las incidencias de los sistemas de comunicaciones desempeña un papel fundamental, sobre todo si se habla de gestionarlas y atenderlas de forma correcta en el momento adecuado para solucionar las averías y restablecer los servicios. Es precisamente este detalle el que puede marcar la diferencia con la competencia ya que existe una estrecha

relación entre la tramitación, la eficiencia y el éxito” (Cedeño, 2007).

Uno de los aspectos esenciales en la operación del servicio es la búsqueda de un equilibrio entre estabilidad y la capacidad de respuesta. La organización de tecnología de información no debe comprometerse en la prestación de servicios para los que carezca de capacidad tecnológica o los necesarios recursos humanos ni tampoco caer en el exceso de la infraestructura tecnología de información encareciendo innecesariamente el costo de los servicios prestados.

Los principales procesos asociados directamente a la fase de operación del servicio son:

- **Gestión de eventos:** responsable de monitorizar todos los eventos que acontezcan en el departamento de soporte técnico de la Universidad de Holguín, con el objetivo de asegurar su correcto funcionamiento y ayudar a prever incidencias futuras.
- **Gestión de incidencias:** responsable de registrar todas las incidencias que afecten a la calidad del servicio y restaurarlo a los niveles acordados de calidad en el más breve plazo posible.
- **Gestión de cambios de servicios de equipamientos informáticos:** responsable de gestionar las peticiones de usuarios que habitualmente requieren cambios en la prestación del servicio de mantenimiento.
- **Gestión de problemas:** responsable de analizar y ofrecer soluciones a aquellos incidentes que por su frecuencia o impacto degradan la calidad del servicio del departamento.
- **Gestión de acceso a los servicios de equipamientos informáticos:** responsable de garantizar que sólo las personas con los permisos adecuados puedan acceder a la información de carácter restringido(OverTI, 2013).
- **Gestión de incidencias:** el proceso de gestión de incidencias cubre todo tipo de incidencias, ya sean fallos, preguntas o consultas planteadas por usuarios al personal técnico ubicados en el departamento de soporte técnico al usuario o bien detectado automáticamente por herramientas de monitoreo(Bon, 2008).

El proceso de gestión de incidentes tiene una estrecha relación con la función mesa de servicio (Service Desk), siendo una parte fundamental para una correcta entrega de servicios de equipamiento informático.

Una mesa de servicio es un conjunto de recursos tecnológicos y humanos, para prestar servicios con la posibilidad de gestionar y solucionar todas las posibles incidencias de manera integral (Foundations ITIL, 2005).

Para garantizar el correcto funcionamiento de las empresas del país, se necesita conocer el comportamiento de las diferentes instalaciones, equipos o sistemas que en ellos existen; entre los aspectos característicos del funcionamiento de los mismos se encuentran los indicadores técnicos:

- disponibilidad
- confiabilidad

La disponibilidad de una estación se define como: “la relación entre el tiempo real de funcionamiento y el tiempo de funcionamiento especificado de una determinada estación, equipo o sistema” (Michelena, 2007).

La confiabilidad de una estación se define como: “la probabilidad de que una instalación terrestre, equipo o sistema funcione dentro de las tolerancias especificadas en un período de tiempo determinado” (Michelena, 2007).

Los estados de control de una incidencia son:

- Incidencia asignada: se encontrará en dicho estado cuando está en análisis, hasta llegar a uno de los estados descritos a continuación. Si se encuentra la incidencia en estado Resuelta o Anulada y un grupo verifica que debe continuar su estudio, la cambiará a estado “Asignada”.
- Incidencia anulada: se encontrará en este estado cuando el grupo técnico que la ha estudiado ha verificado que no se reproduce, o es inexistente, o se encuentra duplicada con otra incidencia creada anteriormente. No es un estado final, ya que el grupo usuario que reportó la incidencia deberá verificar que efectivamente puede ser cancelada.
- Incidencia cancelada: es un estado final de la incidencia. Una incidencia se encontrará en dicho estado tras pasar por el estado “Anulada” si el equipo creador comprueba que así procede.

- **Incidencia resuelta:** una incidencia se encontrará en este estado cuando el equipo técnico ha detectado y corregido la causa raíz que la provocaba. No es un estado final, ya que el grupo usuario que reportó la incidencia deberá verificar que efectivamente se encuentra solucionada y puede ser cerrada.
- **Incidencia cerrada:** es un estado final de una incidencia. La incidencia pasará al estado cerrada una vez el grupo usuario ha comprobado que la incidencia ha quedado totalmente solucionada y se ha aportado la información necesaria para tomar las acciones necesarias con el cliente (por ejemplo, tiempo de afección, clientes totales afectados por esta incidencia, etc.).

Además, cuando una incidencia se encuentra en estado asignada, puede encontrarse pendiente de una acción por parte de alguno de los grupos involucrados. Los sub-estados que se podrán dar ante una incidencia en estado asignada, serán:

- **Incidencia asignada, pendiente de información:** se encontrará en este estado cuando el equipo que está analizándola, necesite más datos para su estudio.
- **Incidencia asignada, pendiente de re-priorización:** se encontrará en este estado cuando se ha solicitado por cualquier grupo involucrado en la incidencia el cambio de prioridad (S.A, 2007).

Los problemas de gestión de incidencias en la actualidad son muy relevantes en las empresas que tiene muchos equipamientos de tecnología de información y comunicación. Estas empresas necesitan de un departamento de soporte técnico para que puedan reportar sus incidencias que ocurren en el día adía.

Los clientes realizan sus quejas directamente a los vendedores sin que ellos den una solución valedera que beneficie a los intereses tanto del cliente como de la empresa debido a la falta de conocimiento.

Los Administradores de Ventas confunden sus funciones y cuando reciben una queja dejan de hacer su trabajo y lo intentan solucionar en la mayoría de casos, no escalan el problema al personal con mejor conocimiento.

Cuando un cliente no está satisfecho con la solución que le dieron vuelve a llamar reportando el mismo problema con otro usuario generando una duplicación del caso

y malentendidos internos. No hay un registro de los problemas ni de las soluciones que se le entregó al cliente haciendo imposible generar estadísticas reales del manejo de incidentes o satisfacción de los clientes (Huércano, 2017).

### **Otros tipos de sistemas de gestión de incidencias**

Otros sistemas de gestión de incidencias que actualmente se está usando en las empresas para comunicar y resolver incidencias son:

- **MantisBT:** 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 muchos escenarios.
- **Lotus Notes:** es un gestor de correo muy parecido a Outlook Exprés, a través del cual uno se puede comunicar con los distintos usuarios mediante correos electrónicos y también mediante un chat que debe estar perfectamente configurado y apuntando a un servidor propio para su correcto funcionamiento (Montesinos, 2014).
- **Kayako Fusion:** es un completo sistema de gestión de incidencias, orientado principalmente a grandes empresas que necesitan gestionar el gran número de peticiones y notificaciones de incidencias que reciben a diario, que permite reducir la carga de trabajo y aumentar el porcentaje de resoluciones por tiempo y priorizar lo más importante de una forma más rápida y sencilla (Montesinos, 2014).
- **NetSupport ServiceDesk:** cumple todos los requisitos como instrumento de gestión de problemas e incidentes completamente compatibles con la funcionalidad de ITIL, y a la vez se puede integrar con facilidad con NetSupport DNA un conjunto de programas de gestión de activos de TI (Montesinos, 2014).

Ideal para la pequeña y mediana empresa, y completamente escalable para grandes empresas, NetSupport ServiceDesk ha sido desarrollado para minimizar el período de inactividad de sistemas a la vez que ofrece un soporte eficaz para todos los usuarios (Montesinos, 2014).

### **1.4 Tecnologías y herramientas**

El análisis de las tecnologías a utilizar constituye un factor importante en el

desarrollo del sistema propuesto. Después de analizado y realizado el estudio de la problemática de la presente investigación y los diferentes conceptos de los sistemas de información, se necesita conocer las tendencias, tecnologías y herramientas actuales disponibles para el desarrollo de la posible aplicación informática. Desde el comienzo de toda aplicación se debe tener en cuenta, las tecnologías y herramientas a utilizar para un correcto desarrollo, utilizando aquellas tecnologías que de acuerdo a sus características le aporten a la aplicación seguridad, confiabilidad, persistencia y portabilidad entre otros aspectos que hacen de un sistema un producto de calidad.

Para el desarrollo del sistema de gestión de reportes de incidencias se utilizarán las tecnologías y las herramientas siguientes:

- Lenguaje de programación PHP
- Framework Yii
- Servidor de aplicaciones Web (servidor http Apache)
- Sistema Gestor de Base de Datos (MySQL)

### Arquitectura Cliente/Servidor

La arquitectura cliente - servidor consiste básicamente en que un programa, el Cliente informático, realiza peticiones a otro programa, el servidor, que le da respuesta. Según *International Business Machines* (IBM) la arquitectura Cliente-Servidor: "Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas" (FJRP, 2008).

En esta arquitectura se pueden distinguir 3 capas o niveles:

- Manejador de Bases de Datos (Nivel de almacenamiento)
- Procesador de aplicaciones o reglas del negocio (Nivel lógico)
- Interfaces del usuario (Nivel de presentación)

Esta arquitectura facilita que los recursos sean centralizados, pues el servidor puede administrar los recursos que son comunes a todos los usuarios. Posee seguridad mejorada gracias a que la cantidad de puntos de entrada que permite el acceso a los datos no es muy alta (Renaud, 2008).

Además, es escalable, fácil de mantener, presenta centralización del control pues los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado, no pueda dañar el sistema. La existencia de plataformas de *hardware* cada vez más baratas, permite que los costes se hayan reducido considerablemente. Además, se pueden utilizar componentes (tanto de *hardware* como de *software*) de diferentes fabricantes, lo cual favorece la implantación de soluciones. El servidor presenta a todos sus clientes una interfaz única y bien definida. El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa. Muestran información de forma sencilla a través de un hipervínculo(FJRP, 2008).

El Cliente/Servidor también contribuye a una disminución de los costos de entrenamiento del personal, pues favorece la construcción de interfaces gráficas interactivas, las cuales pueden hacer que los "datos" se conviertan en "información" y además son más intuitivas y fáciles de usar.

Actualmente la gran mayoría de las aplicaciones de gestión empresarial hacen uso del modelo Cliente/Servidor, precisamente en el desarrollo de la aplicación se tendrá en cuenta por las siguientes razones:

- Petición de sistemas fáciles de usar
- Productividad y calidad
- Bajo costo y alto rendimiento
- Acceso fácil a la información
- Nuevas tecnologías y herramientas de alta productividad.

### 1.4.1 Lenguajes de programación

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Estas instrucciones permiten la construcción de programas con los cuales se puede realizar operación de entrada y salida, almacenamiento, cálculos y lógica de comparación. Un lenguaje de programación permite a un programador especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.

Existen numerosos lenguajes de programación empleados para el desarrollo de aplicaciones *web*, entre los cuales se encuentran: PHP, Perl, Ruby, Python, entre otros (Ellis, 2011).

#### **Lenguaje de programación PHP**

PHP es un lenguaje de scripting y del lado de servidor, que es utilizado para incluir funcionalidades dentro de código HTML dentro de las etiquetas “<?php ?>” o incluso en páginas con su propia extensión (php, php4, php5 o phtml). A diferencia de lenguajes de script, del lado de cliente como Javascript, el servidor procesa en su totalidad el código programado de su lado, entregándole al cliente un archivo en HTML procesado con los resultados que fueron programados en el script, de forma que cualquier petición entre el servidor, el cliente y aplicativos de terceros (como bases de datos o sistemas de autenticación) se realiza de forma transparente como se muestra en la (figura 3).

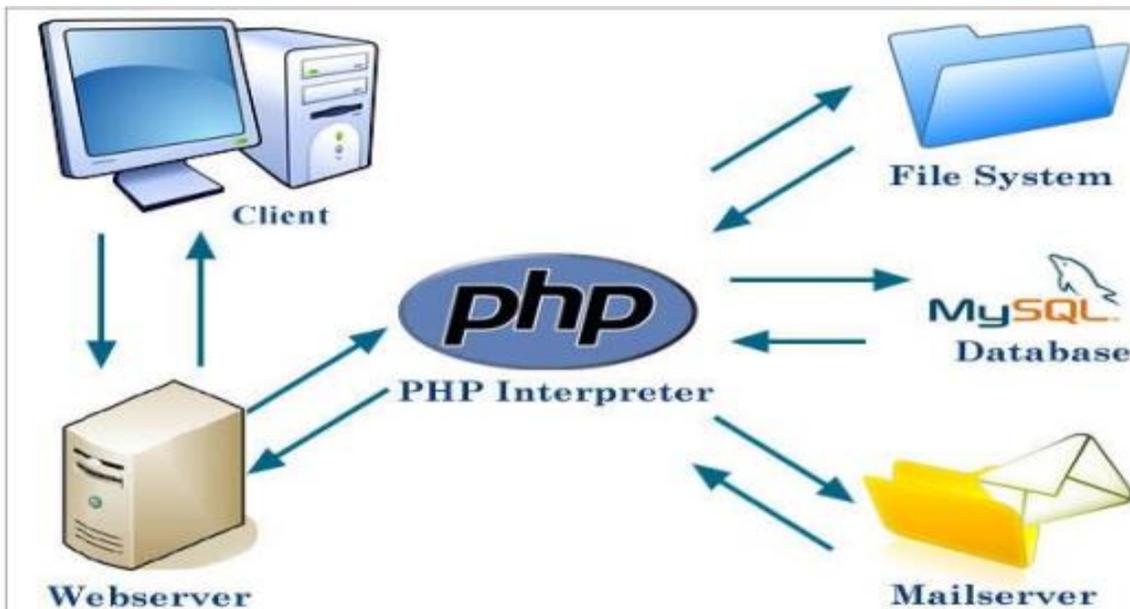


Figura 3:Funcionamiento del Lenguaje PHP

La ventaja de usar este lenguaje es la facilidad con la que el usuario puede ver resultados y obtener funcionalidades sin necesidad de ejecutar comandos externos o desde el sitio en PHP, además de explotar las funcionalidades de la ejecución de un aplicativo en un típico esquema cliente servidor. El funcionamiento de las páginas en PHP alojadas en un servidor es el siguiente:

- El navegador del cliente solicita el documento PHP.
- Llega la solicitud del servidor y el servidor localiza el documento, lanza el intérprete de PHP y ejecuta todo su código.
- Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al cliente.
- El servidor transfiere el resultado en HTML y es mostrado en el navegador del cliente (Ellis, 2011).

#### 1.4.2 Framework Yii

Yii es un framework orientado a objetos, *software libre*, de alto rendimiento basado en componentes, PHP y *framework* de aplicaciones web.

El proyecto Yii comenzó el 1 de enero de 2008, con el fin de solucionar algunos problemas con el framework Prado. Por ejemplo, Prado es lento manejando páginas complejas, tiene una curva de aprendizaje muy pronunciada y tiene varios controles

que dificultan la personalización, mientras que Yii es mucho más fácil y eficiente. En Octubre de 2008 después de 10 meses de trabajo en privado, la primera versión alfa de Yii fue lanzada. El 3 de diciembre del mismo año, Yii 1.0 fue formalmente presentado (Winesett, 2009).

### 1.4.3 Servidor Apache

Gracias a los avances en conectividad y la gran disponibilidad de banda ancha, hoy en día es muy común establecer los servidores *web* dentro de la propia empresa, sin tener que recurrir a caros alojamientos en proveedores externos. Esto es posible gracias a Apache, uno de los mejores y el más utilizado entre los servidores *web* que existen. Apache ha construido una gran reputación entre los servidores *web* gracias a su gran estabilidad, confiabilidad, documentación y hasta el mismo código en sí.

Entre sus principales ventajas se encuentran:

- Es personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida.
- Tiene un formato simple, los archivos de configuración de Apache están en ASCII, por lo que pueden ser editados tan solo con un editor de texto.
- Es multiplataforma y código abierto.
- Popular (fácil conseguir ayuda/suporte).
- Provee una muy buena base para la seguridad del sistema, gracias a los módulos de Autenticación, Autorización y Control de Acceso al Servidor *Web*.
- Trabaja en conjunto con gran cantidad de lenguajes de programación interpretados como PHP (Lenguaje Personal Home Page), Perl, Java, JSP (Java Server Pages) y otros lenguajes de script, que son el complemento ideal para los sitios *web* dinámicos.

Se seleccionó como servidor *web* apache por las ventajas que brinda, además por ser este una solución sencilla y rápida para tener sitios *web* funcionando al 100% de forma gratuita (Ellis, 2011).

### **Sistema Gestor de Bases de Datos MySQL**

Es un sistema de base de datos relacional orientada a objetos. Es un proyecto de

código libre, soporta casi toda la sintaxis SQL y posibilita muchas características tales como: consultas complejas, integridad referencial, vistas, integridad transaccional, es una de las bases de datos más potentes y robustas del mercado. Maneja muy bien grandes cantidades de datos y su rendimiento es muy bueno ante el acceso múltiple de usuarios al sistema (Autores, 2010).

MySQL es un potente SGBD objeto-relacional, multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y está liberado bajo la licencia Berkeley *software* Distribución (BSD). Es ampliamente considerado como el sistema gestor de bases de datos de código abierto más avanzado del mundo. Fue pionero en muchos conceptos que estuvieron disponibles en algunos sistemas de bases de datos comerciales de alto calibre, como, por ejemplo: control de concurrencia multiversión, gestión de transacciones y puntos de salvos. MySQL tiene las siguientes ventajas (KCHWABER, 2002):

- Organiza los datos mediante un modelo objeto-relacional.
- Capaz de manejar procedimientos, rutinas complejas y reglas.
- Soporta *tablespaces*, transacciones anidadas, copias de seguridad en línea y soporte para parte de los estándares SQL 92, 99, 2003 y 2008.
- Cuenta con una API sumamente flexible propia para el trabajo con varios lenguajes de programación y procedurales como C, C++, NET, Bash, Delphi, PL/Java, PL/Perl, PL/Tcl, PL/pgSQL, PL/Ruby, PL/PHP, PL/Python, PL/Scheme y PL/R.
- Ofrece transacciones que permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos declarados por el usuario; soporta además sobrecarga de operadores, sobrecarga de procedimientos, vistas materializables, particionamiento de tablas y datos.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información dentro de la base de datos.
- Ofrece un control de acceso simultáneo a través de la gestión de múltiples versiones de un mismo registro (MVCC).

- Las restricciones y disparadores tienen la función de mantener la integridad y consistencia en las bases de datos.
- Usa una arquitectura cliente/servidor basada en un proceso por usuario. Existe un proceso maestro que se ramifica para proporcionar conexiones adicionales por cada cliente que se intenta conectar a MySQL.
- Tiene incorporado el mecanismo *Write Ahead Logging* (WAL), que incrementa la confiabilidad de las bases de datos al registrar los cambios antes de ser escritos al disco; lo que asegura que, en caso de ocurrir un fallo crítico en las bases de datos, exista un registro de transacciones del cual se pueda restaurar (LAGO, 2007).

### 1.5 Metodologías de desarrollo de software

Las metodologías de desarrollo de *software* surgen debido a la necesidad de emplear una serie de procedimientos y técnicas a la hora de desarrollar un producto de *software*, con el propósito de brindarle una guía al desarrollador. Debido a que no todos los sistemas que se desarrollan tienen la misma complejidad existen una gran variedad de metodologías para la creación de los mismos, están las metodologías tradicionales orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán y las metodologías ágiles orientadas a la interacción con el cliente y el desarrollo incremental del *software*, mostrando versiones parcialmente funcionales del *software* al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando (Ellis, 2011).

Desarrollar un sistema de gestión de reporte de incidencias que cumpla los requisitos establecidos por el cliente depende en gran medida de las etapas que complementan la metodología que se escoja para guiar el mismo. Estas etapas contienen en su interior una serie técnicas y procedimiento que necesitan los desarrolladores para implementar y sostener el *software*.

El éxito del producto depende en gran parte de la metodología escogida por el equipo, ya sea tradicional o ágil, donde los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempos establecidos.

Las metodologías de desarrollo de *software* surgieron a raíz de la necesidad de

controlar y documentar proyectos cada vez más complejos. La implantación de una metodología es necesaria si se quieren gestionar adecuadamente los proyectos(Burback, 1998).

### **1.5.1 Metodología RUP**

La metodología tradicional RUP (Proceso Unificado de Modelado), es un proceso formal que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de *software* de alta calidad que satisfaga los requerimientos de los usuarios finales.

RUP hace uso de cuatro etapas en su metodología descritas de la siguiente manera: una fase de concepción, una fase de elaboración, una fase de construcción y una de transición. Dentro de cada una de ellas se realiza varias iteraciones dependiendo de la dimensión del proyecto (Jacobson, 2000).

### **1.5.2 Metodología ICONIX**

Iconix es una metodología de desarrollo del *software* que se halla a medio camino entre la complejidad de RUP y la simplicidad de XP, es una metodología simplificada en comparación a otras más tradicionales, la cual unifica un conjunto de métodos de orientación a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar, cuenta con una secuencia de pasos que se deben seguir y determina claramente las actividades a desarrollar en cada etapa del ciclo de vida del proyecto que la utilice.

Iconix se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final. Presenta claramente las actividades de cada fase y exhibe una secuencia de pasos que deben ser seguidos, ofrece trazabilidad, que no es más que la capacidad de seguir una relación entre los diferentes artefactos del *software* producidos. La metodología brinda un uso dinámico del UML porque utiliza solo algunos diagramas, al contrario de RUP (Cruz, 2009).

### **1.5.3 Metodología de desarrollo de software XP**

Las metodologías ágiles (como por ejemplo XP, SCRUM, DSDM, Crystal, etc.)

forman parte del movimiento de desarrollo ágil de *software*, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. De forma que una metodología ágil es la que tiene como principios que:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El *software* que funciona es más importante que la documentación exhaustiva.
- La colaboración con el cliente en lugar de la negociación de contratos.
- La respuesta delante del cambio en lugar de seguir un plan cerrado.

Se puede decir que, este movimiento empezó a existir a partir de febrero de 2001, cuando se reunieron los representantes de cada una de estas metodologías y terminaron poniendo en común sus ideas en una declaración conjunta.

La programación extrema (XP) es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. Este modelo de programación se basa en una serie de metodologías de desarrollo de *software* en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación.

Una de las características principales de este método de programación, es que sus ingredientes son conocidos desde el principio de la informática. Los autores de XP han seleccionado aquellos que han considerado mejores y han profundizado en sus relaciones y en cómo se refuerzan los unos con los otros. El resultado de esta selección ha sido esta metodología única y compacta.

Por esto, aunque no está basada en principios nuevos, el resultado es una nueva manera de ver el desarrollo de *software*. El objetivo que se perseguía en el momento de crear esta metodología era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos aplicando el sentido común.

La Programación Extrema se basa en 12 principios básicos agrupados en cuatro categorías:

- Retroalimentación a escala fina

**El principio de pruebas:** se tiene que establecer un periodo de pruebas de aceptación del programa (llamado también periodo de caja negra) donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento.

**Proceso de planificación:** en esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias de Usuario (User Stories). Entre 20 y 80 historias (todo dependiendo de la complejidad del problema) se consideran suficientes para formar el llamado Plan de Liberación, el cual define de forma específica los tiempos de entrega de la aplicación para recibir retroalimentación por parte del usuario. Por regla general, cada una de las Historias de Usuario suelen necesitar de una a tres semanas de desarrollo.

**El cliente en el sitio:** se le dará poder para determinar los requerimientos, definir la funcionalidad, señalar las prioridades y responder las preguntas de los programadores. Esta fuerte interacción cara a cara con el programador disminuye el tiempo de comunicación y la cantidad de documentación, junto con los altos costos de su creación y mantenimiento. Este representante del cliente estará con el equipo de trabajo durante toda la realización del proyecto.

**Programación en parejas:** uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo pone sus dudas. Requiere que todos los programadores XP escriban su código en parejas, compartiendo una sola máquina. De acuerdo con los experimentos, este principio puede producir aplicaciones más buenas, de manera consistente, a iguales o menores costos (LOUCOPOULOS, 1989).

- **Proceso continuo en lugar de por lotes**

**Integración continua:** permite al equipo hacer un rápido progreso implementando las nuevas características del *software*. En lugar de crear builds (o versiones) estables de acuerdo a un cronograma establecido, los equipos de programadores XP pueden reunir su código y reconstruir el sistema varias veces al día. Esto reduce los problemas de integración comunes en proyectos largos y estilo cascada.

**Refactorización:** permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo. Los programadores evalúan continuamente el diseño y recodifican lo necesario. La finalidad es mantener un sistema enfocado a proveer el valor de negocio mediante la minimización del código duplicado y/o ineficiente.

**Entregas pequeñas:** colocan un sistema sencillo en producción rápidamente que se actualiza de forma rápida y constante permitiendo que el verdadero valor de negocio del producto sea evaluado en un ambiente real. Estas entregas no pueden pasar las 2 o 3 semanas como máximo.

- Entendimiento compartido

**Diseño simple:** se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos. Desarrollo simple se enfoca en proporcionar un sistema que cubra las necesidades inmediatas del cliente, ni más ni menos. Este proceso permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.

**Metáfora:** desarrollada por los programadores al inicio del proyecto, define una historia de cómo funciona el sistema completo.

**Propiedad colectiva del código:** un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo. Este método difiere en mucho a los métodos tradicionales en los que un simple programador posee un conjunto de código. Los defensores de XP argumentan que mientras haya más gente trabajando en una pieza, menos errores aparecerán.

**Estándar de codificación:** define la propiedad del código compartido, así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.

**Bienestar del programador:** La programación extrema sostiene que los programadores cansados escriben código de menor calidad. Minimizar las horas extras y mantener los programadores frescos, generará código de mayor calidad.

## Conclusiones del capítulo

Se realizó un estudio sobre las bases teóricas que fundamentan la presente investigación y las tecnologías para el desarrollo de aplicaciones fundamentalmente *web*, destacándose aquellas se adaptan mejor a las necesidades del contexto donde se desarrolla la investigación. El estudio realizado arrojó como resultado:

La necesidad de realizar una aplicación *web* que controle y agilice el proceso de gestión de reporte de incidencias en la Universidad de Holguín.

Se seleccionaron como tecnologías para el desarrollo de la aplicación el lenguaje de programación PHP, empleando como gestor de bases de datos MySQL.y el servidor *web* Apache, guiados por la metodología de desarrollo de *software* XP. Las herramientas y tecnologías propuestas se adaptan correctamente al marco donde se desarrolló la aplicación, asegurando una correcta compatibilidad.

## **CAPÍTULO 2: DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA**

En el presente capítulo se expone en detalle la aplicación de la metodología de *software* XP en el mantenimiento de la aplicación, además, se exponen las principales características del sistema y se detallan los requerimientos funcionales y no funcionales que dan paso a las historias de usuario.

Los requerimientos del sistema representan las acciones que el mismo debe realizar para satisfacer las necesidades para las cuales se creó. Es por ello que la elaboración detallada de estos, para ser utilizados como base de las futuras pruebas sea de vital importancia en el proceso de desarrollo de la aplicación.

### **2.1 Fase de exploración**

La metodología de desarrollo XP comienza con su fase de exploración. Durante esta fase se realiza el proceso de identificación de las Historias de Usuario (HU, del inglés *User Histories*). Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

#### **2.1.1 Descripción de requisitos**

Los requisitos del sistema representan las Tareas que el sistema debe ser capaz de realizar para trabajar correctamente. Es por ello que la redacción detallada de estos, para ser utilizados como base de las futuras pruebas sea de vital importancia.

Requerimientos funcionales

RF1- Gestionar elemento de reporte

RF2- Gestionar equipo

RF2.1- Insertar equipo

RF2.2- Modificar equipo

RF2.3- Eliminar equipo

RF2.4- Listar equipos

RF3- Gestionar incidencias

RF3.1- Insertar incidencias

RF3.2- Modificar incidencias

RF3.3- Eliminar incidencias

RF3.4- Listar incidencias

**RF4- Gestionar plan de mantenimiento por equipo**

RF4.1- Insertar plan de mantenimiento por equipo

RF4.2- Modificar plan de mantenimiento por equipo

RF4.3- Eliminar plan de mantenimiento por equipo

**RF5- Gestionar mantenimientos realizados a los equipos**

RF5.1- Insertar mantenimientos realizados a los equipos

RF5.2- Modificar mantenimientos realizados a los equipos

RF5.3- Eliminar mantenimientos realizados a los equipos

**RF6- Gestionar solicitud de mantenimiento**

RF6.1- Insertar solicitud de mantenimiento

RF6.2- Modificar solicitud de mantenimiento

RF6.3- Eliminar solicitud de mantenimiento

RF7- Reportar como atendido la notificación de incidencia

**RF8- Muestrar incidencia por atender**

**RF9- Gestionar planes por realizar**

**RF10- Gestionar reporte por departamento**

RF11-Gestionar locales

**Requerimientos no funcionales**

Los requerimientos no funcionales están relacionados con la apariencia, usabilidad, integración y otras cualidades que no dependen de lo que debe hacer la aplicación, sino de cómo lo debe hacer.

**Apariencia o Interfaz Externa**

- Los colores deben ser usados en tonalidades claras de forma que el usuario visualice una interfaz agradable, el tamaño de las letras debe ser adecuado, evitando fuentes pequeñas que interfieran en una buena visualización.
- La interfaz no debe de recargarse con imágenes para proporcionar una

navegación cómoda.

- El flujo de trabajo en el sistema debe asemejarse al proceso rutinario de gestión del proceso.
- Debe ser simple de usar y explícito para que los usuarios se sientan cómodos a la hora de trabajar con el mismo.
- Usabilidad
- Los usuarios que trabajarán el sistema deben tener conocimientos básicos de computación.
- El sistema debe ser accesible desde todas las áreas de la UHOLM que cuentan con conectividad.
- El sistema debe ser sencillo para agilizar el tiempo de conexión al mismo.
- Ayuda y documentación
- Debe contar con un Manual de usuario que propicie las principales funcionalidades que tiene el sistema como apoyo para la interacción con el producto informático.
- Rendimiento
- Los tiempos de respuestas deben ser rápidos ya que el sistema es una aplicación cliente servidor, con una apariencia web.
- La disponibilidad de trabajo de la red contra el servidor es constante.
- Portabilidad
- El sistema puede ser usado bajo los Sistemas Operativos Windows y Linux sistemas operativos MacOS.
- El sistema corre sobre una plataforma web, codificada en PHP con el framework y sistema de base de datos en MySQL.
- Seguridad
- El encargado debe validar el ingreso al sistema de los usuarios, ya que este es el que tendrá acceso a la base de datos, a los ficheros fuentes y es el responsable de la autorización en general del mismo.
- Sólo los usuarios autorizados podrán acceder al sistema.
- No existe información que se pueda tener sin ser usuario del sistema.
- Software

- Máquina Servidor:
- Debe tener instalado el sistema Operativo Windows o Linux, Servidor Web Apache Tomcat versión 7.0, MySQL 5.5.35.
- Máquina Cliente:
- Se debe disponer del navegador Web Mozilla Firefox 9.0.1 o superior.
- El sistema operativo debe ser Windows, Linux o sistemas operativos MacOS.
- Hardware
- Máquina Servidor:
- PC Intel Pentium IV o superior con 2.6 GHz o superior.
- Memoria RAM de 1.0 GB o superior.
- Espacio en disco de 1.0 GB o más disponibles para la instalación del software requeridos.
- Máquina cliente:
- PC Intel Pentium IV o superior con 1.6 GHz o superior.
- Memoria RAM de 256 MB o superior.

### 2.1.2 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requerimientos del *software*. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (LETELIER, 1989). Las Tablas 1 hasta 9 muestran algunas historias de usuarios, las demás se muestran en el Anexo 1.

Tabla 1: Historia de Usuario # 1 "Gestionar elemento de reporte"

Historia de usuario	
<b>Número:</b> 1	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar elemento de reporte	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 1
<b>Descripción:</b> El administrador debe ser capaz de gestionar elementos de reporte en el sistema.	
<b>Información adicional:</b> Gestionar elemento de reporte incluye insertar, editar, visualizar, eliminar.	

Tabla 2: Historia de Usuario # 2 "Gestionar equipo"

Historia de usuario	
<b>Número:</b> 2	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar equipo	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 1
<b>Descripción:</b> El administrador debe ser capaz de gestionar el equipo en el sistema.	
<b>Información adicional:</b> Gestionar equipo, incluye insertar, modificar, listar y eliminar equipo del sistema.	

Tabla 3: Historia de Usuario # 3 "Gestionar incidencias"

Historia de usuario	
<b>Número:</b> 3	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar incidencias.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 1
<b>Descripción:</b> El administrador debe ser capaz de Gestionar equipo en el sistema.	
<b>Información adicional:</b> Gestionar equipo, incluye Insertar modificar, eliminar, Listar, equipos.	

Tabla 4: Historia de Usuario # 4 "Gestionar plan de mantenimiento por equipo"

Historia de usuario	
<b>Número:</b> 4	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar plan de mantenimiento por equipo.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 2
<b>Descripción:</b> El administrador debe ser capaz de Gestionar plan de mantenimiento por equipo	
<b>Información adicional:</b> Gestionar plan de mantenimiento por equipo, incluye insertar, editar, modificar, eliminar.	

Tabla 5: Historia de Usuario # 5 "Gestionar mantenimientos realizados a los equipos"

Historia de usuario	
<b>Número:</b> 5	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar mantenimientos realizados a los equipos.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 2
<b>Descripción</b> El sistema debe de ser capaz de Gestionar plan de mantenimiento por equipo.	
<b>Información adicional:</b> Gestionar mantenimientos realizados a los equipos incluye insertar, modificar, eliminar.	

Tabla 6: Historia de Usuario # 6 "Gestionar solicitud de mantenimiento"

Historia de usuario	
<b>Número:</b> 6	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar solicitud de mantenimiento.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 2
<b>Descripción</b> El sistema debe de ser capaz de Gestionar solicitud de mantenimiento.	
<b>Información adicional:</b> Gestionar solicitud de mantenimiento incluye insertar, modificar, eliminar mantenimientos realizados a los equipos.	

Tabla 7: Historia de Usuario # 7 "Reportar como atendido la notificación de incidencia"

Historia de usuario	
<b>Número:</b> 7	<b>Usuario:</b> administrador
<b>Nombre:</b> Reportar como atendido la notificación de incidencia.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 2
<b>Descripción:</b> El sistema debe de ser capaz de Reportar como atendido la notificación de incidencia.	
<b>Información adicional:</b> Reportar como atendido la notificación de incidencia.	

Tabla 8: Historia de Usuario # 8 "Mostrar incidencia por atender"

Historia de usuario	
<b>Número:</b> 8	<b>Usuario:</b> administrador
<b>Nombre:</b> Mostrar incidencia por atender.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 3
<b>Descripción:</b> El sistema debe de ser capaz mostrar incidencia por atender.	
<b>Información adicional:</b> Mostrar incidencia por atender.	

Tabla 9: Historia de Usuario # 9 "Gestionar planes por realizar"

Historia de usuario	
<b>Número:</b> 9	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar planes por realizar.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 5 días	<b>Iteración asignada:</b> 3
<b>Descripción</b> El sistema debe de ser capaz de Gestionar planes por realizar.	
<b>Información adicional:</b> Gestionar planes por realizar	

## 2.2 Planificación

En esta etapa se priorizan las historias de usuario. Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. Se deben incluir varias iteraciones para lograr un mejor desempeño. El cronograma fijado en la etapa de planeamiento se realiza a un número de iteraciones, cada una toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo.

Esto es alcanzado seleccionando las historias que harán cumplir la construcción de la estructura para el sistema completo. El cliente decide las historias que se seleccionarán para cada iteración. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para producción (Hurtado, 2005).

### 2.2.1 Iteración

Para la selección del trabajo de cada iteración se tuvo en cuenta que este no tuviera más de 3 semanas de desarrollo, siguiendo las prácticas básicas de XP, como se muestra en la siguiente tabla.

Tabla 10: Distribución de las historias de usuario por iteración

Iteraciones	Distribución de las historias de usuario por iteración	Tiempo
1	RF1- Gestionar elemento de reporte RF2- Gestionar equipo RF3- Gestionar incidencias	3 semanas
2	RF4- Gestionar plan de mantenimiento por equipo RF5- Gestionar mantenimientos realizados a los equipos RF6- Gestionar solicitud de mantenimiento RF7- Reportar como atendido la notificación de incidencias	3 semanas
3	RF8-Mostrar incidencia por atender RF9- Gestionar planes por realizar	3 semanas
4	RF10- Gestionar reporte por departamento. RF11-Gestionar locales	3semanas

### 2.2.2 Plan de entregas

El plan de entregas es el compromiso final del equipo de trabajo con los clientes. Es una cuestión de vital importancia para el negocio entre ambas partes, ya que la entrega tardía o temprana de la solución, repercute notablemente en la economía y moral de todos los involucrados. La estimación es uno de los temas más complicados del desarrollo de un proyecto que utiliza la metodología XP, y es por ello que resulta de vital importancia tener bien claros los requerimientos del cliente, el estilo de trabajo del equipo de desarrollo y el tiempo con que dispone el cliente para tener en sus manos la solución. La siguiente tabla expone el cronograma de entregas del módulo.

Tabla 11: Cronograma de entregas

Plan de entregas				
1ra iteración	2da iteración	3ra iteración	4ta iteración	5ta iteración
1 de febrero 2019	21 de febrero 2019	12 de Marzo 2019	2 de abril 2019	23 de abril 2019

## 2.3 Implementación

La metodología XP propone comenzar la implementación partiendo de una arquitectura lo más flexible posible para evitar grandes cambios en las próximas iteraciones y en los cambios que generalmente el cliente propone.

### 2.3.1 Primera Iteración

El principal objetivo de esta iteración es crear una aplicación con la estructura necesaria para implementar un sistema de gestión de reportes de incidencias en el estilo arquitectónico seleccionado. La tabla siguiente muestra algunas tareas de la iteración (Anexo 2). Para ello se trazaron las siguientes Tareas:

**Tarea número 1:** Insertar datos del elemento de reporte

**Tarea número 2:** Editar datos del elemento de reporte

**Tarea número 3:** Visualizar datos de la Gestión de elemento de reporte

**Tarea número 4:** Eliminar datos de la Gestión de elemento de reporte

Tabla 12: Tarea # 1 "Insertar datos del elemento de reporte".

Tarea	
<b>Número:</b> 1	<b>Número de la historia:</b> HU_01
<b>Nombre de la Tarea:</b> Insertar datos del elemento de reporte	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 5 días
<b>Fecha de inicio:</b> 1 de febrero del 2019	<b>Fecha de fin:</b> 5 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario insertar y guardar en la base de datos la información referente a gestionar elemento de reporte.	

Tabla 13: Tarea # 2 "Editar elemento de reporte".

Tarea	
<b>Número: 2</b>	<b>Número de la historia: HU_01</b>
<b>Nombre de la Tarea:</b> Editar datos del elemento de reporte.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 4 días
<b>Fecha de inicio:</b> 7 de febrero del 2019	<b>Fecha de fin:</b> 11 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario editar y actualizar en la base de datos elemento de reporte.	

Tabla 14: Tarea # 3 "Visualizar datos de la Gestión de elemento de reporte".

Tarea	
<b>Número: 3</b>	<b>Número de la historia: HU_01</b>
<b>Nombre de la Tarea:</b> Visualizar datos de la Gestión de elemento de reporte.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 12 de febrero del 2019	<b>Fecha de fin:</b> 15 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario visualizar los datos de de la gestión de elemento de reporte.	

Tabla 15: Tarea # 4 "Eliminar datos de la Gestión de elemento de reporte".

Tarea	
<b>Número:</b> 4	<b>Número de la historia:</b> HU_01
<b>Nombre de la Tarea:</b> Eliminar datos de la Gestión de elemento de reporte.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 16 de febrero del 2019	<b>Fecha de fin:</b> 19 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario eliminar de la base de datos los datos la Gestión de elemento de reporte.	

### 2.3.2 Segunda Iteración

El principal objetivo de esta iteración es lograr una versión de la herramienta informática que permita gestionar equipo. Para ello se trazan las siguientes tareas:

**Tarea número 5:** Insertar equipo

**Tarea número 6:** Modificar equipo

**Tarea número 7:** Eliminar equipo

Tabla 16: Tarea # 5 "Insertar equipo".

Tarea	
<b>Número:</b> 5	<b>Número de la historia:</b> HU_02
<b>Nombre de la Tarea:</b> Insertar equipo.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 21 de febrero del 2019	<b>Fecha de fin:</b> 24 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario insertar en la base de datos del sistema.	

Tabla 17: Tarea # 6 "Modificar equipos".

Tarea	
<b>Número: 6</b>	<b>Número de la historia: HU_02</b>
<b>Nombre de la Tarea:</b> Modificar equipo.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 24 de febrero del 2019	<b>Fecha de fin:</b> 27 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Modificar equipo y actualizar en la base de datos del sistema.	

Tabla 18: Tarea # 7 "Eliminar equipo".

Tarea	
<b>Número: 7</b>	<b>Número de la historia: HU_02</b>
<b>Nombre de la Tarea:</b> Eliminar equipo	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 28 de febrero del 2019	<b>Fecha de fin:</b> 31 de febrero del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario eliminar de la base de datos el equipo del sistema.	

### 2.3.3 Tercera Iteración

El principal objetivo de esta iteración es lograr una versión de la herramienta informática que permita mostrar incidencia por atender. Para ello se trazan la siguiente Tarea:

**Tarea número 8:** Mostrar incidencia por atender

**Tarea número 9:** Gestionar planes por realizar

Tabla 19: Tarea # 8 "Mostrar incidencia por atender".

Tarea	
<b>Número:</b> 8	<b>Número de la historia:</b> HU_08
<b>Nombre de la Tarea:</b> Mostrar incidencia por atender.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 17 de Marzo del 2019	<b>Fecha de fin:</b> 20 de Marzo del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario mostrar incidencia por atender .	

Tabla 20: Tarea # 9 "Gestionar planes por realizar".

Tarea	
<b>Número:</b> 09	<b>Número de la historia:</b> HU_09
<b>Nombre de la Tarea:</b> Gestionar planes por realizar .	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 17 de Marzo del 2019	<b>Fecha de fin:</b> 20 de Marzo del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario gestionar planes por realizar.	

### 2.3.4 Cuarta Iteración

El principal objetivo de esta iteración es lograr una versión de la herramienta informática que permita gestionar plan de mantenimiento por equipo Para ello se trazan las siguientes Tareas:

**Tarea número 10:** Gestionar reporte por departamento

Tabla 21:Tarea # 10 "Gestionar reporte por departamento".

Tarea	
<b>Número:</b> 10	<b>Número de la historia:</b> HU_10
<b>Nombre de la Tarea:</b> Gestionar reporte por departamento.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 21 de Marzo del 2019	<b>Fecha de fin:</b> 24 de Marzo del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario gestionar reporte por departamento.	

### 2.3.5 Quinta Iteración

El principal objetivo de esta iteración es lograr una versión de la herramienta informática que permita gestionar locales. Para ello se trazan las siguientes Tareas:

**Tarea número 11:** Gestionar locales

Tabla 22:Tarea # 11 "Gestionar locales".

Tarea	
<b>Número:</b> 11	<b>Número de la historia:</b> HU_11
<b>Nombre de la Tarea:</b> Gestionar locales.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 9 de Abril del 2019	<b>Fecha de fin:</b> 12 de Abril del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al usuario visualizar y gestionar locales.	

## 2.4 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema (figura 4) en términos de cómo se distribuye la funcionalidad entre los nodos computacionales. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una principal influencia en el diseño. Su objetivo fundamental es la distribución de las condiciones físicas de *hardware* y *software* que se espera tenga la implantación del sistema. En diagrama de despliegue se muestra como se comunican el servidor web (Apache) con el servidor de la base de datos (MySQL) y el cliente navegador (Renaud, 2008).

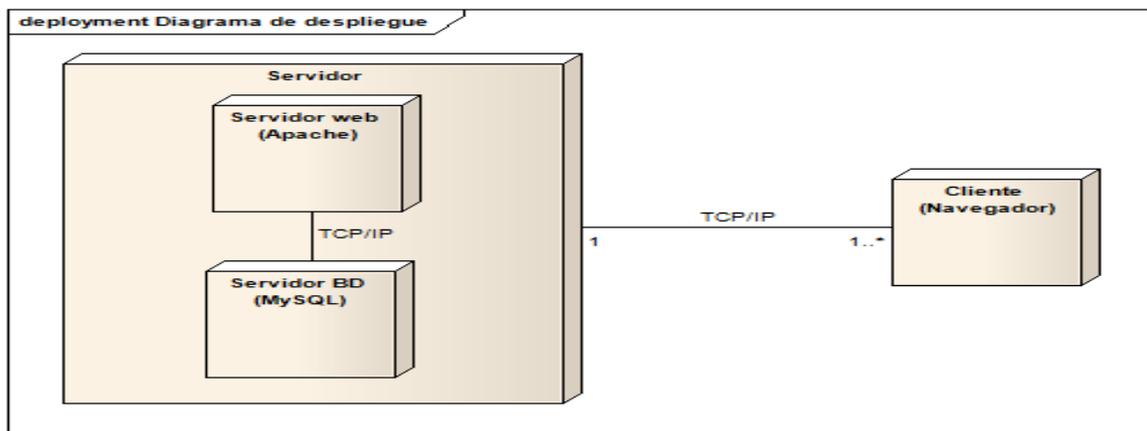


Figura 4: Diagrama de despliegue

## 2.5 Valoración de Sostenibilidad de la propuesta

La valoración de sostenibilidad es el proceso donde se realiza la apreciación de los diferentes impactos que tiene un producto informático en diferentes dimensiones (Administrativa, Socio-Humanista, Tecnológica y Ambiental). A la hora de desarrollar un *software* se deben tener en cuenta desde las primeras fases el análisis de los procesos a informatizar, el personal vinculado a la actividad, las características de los usuarios que manipularan el sistema, el estudio acerca de aplicaciones anteriores a la propuesta que se quiere desarrollar y las diferencias que existen entre ellas, tratando en todo momento que la solución final se acerque lo más posible en su apariencia externa al trabajo clásico que llevan las personas para las que se trabaja (modelos, formularios, vistas etc.), para así asegurar que el cambio que se va a producir sea asimilado de una forma positiva (Concepción, 2009).

### **Dimensión Administrativa**

Esta dimensión se sustenta en el ahorro, gasto, calidad de la producción y los servicios, de administración de recursos y en la toma de decisiones administrativas que brinda como resultado. El gasto en recursos informáticos es mínimo pues la infraestructura que sirve de soporte al proyecto ya estaba creada. El sistema informático optimiza la mano de obra. Aporta mejoras significativas para la gestión de información optimizando el tiempo y favoreciendo el flujo de trabajo de los usuarios que interactúan con el *software*. Proporciona además una disminución considerable en los tiempos de creación de los sistemas y garantiza la perdurabilidad de información valiosa. No se incurren en gastos de electricidad asociado al uso de las computadoras y el servidor porque esta situación ya se presentaba.

Las herramientas utilizadas para el desarrollo del *software* en su totalidad son libres, con lo que se absuelve a la universidad de incurrir en gastos para llevar a cabo el desarrollo del proyecto. A partir de lo analizado y los beneficios que proporciona el *software*, se arribó a la conclusión que éste es sostenible desde la dimensión administrativa.

## **Dimensión Socio-Humanista**

La satisfacción de los trabajadores es un factor esencial para lograr la calidad de sus servicios, por lo que actualmente las entidades hacen énfasis en lograr su bienestar. La automatización de los procesos proporciona el bienestar de estos en el desempeño laboral a partir de las comodidades brindadas.

El sistema informático propuesto resolverá la necesidad planteada en el Departamento de Soporte Técnico de la Universidad de Holguín, pues facilitará que el proceso de comunicación del departamento se realice de una forma cómoda propiciando así que disminuya el tiempo que se emplea en las tareas que se ejecutan. Con el desarrollo del sistema, la carga de trabajo se reducirá, al realizar las labores con el apoyo de la herramienta informática. Estas mejoras en las condiciones de trabajo darán lugar a una gran satisfacción del personal implicado, lo que favorecerá una mejor calidad en la atención de los Departamentos. Se garantizará que los reportes tanto diarios como mensuales se realicen de forma correcta, segura y en un tiempo breve.

Se tuvo en cuenta el rechazo al cambio que podía surgir al establecer el sistema, lo cual se considera normal, debido a la tendencia del ser humano a hacer costumbre de lo habitual y rutinario y a la resistencia inconsciente ante los cambios de su entorno. Para favorecer la aceptación del sistema se llevaron a cabo entrevistas con los usuarios potenciales, para explicarles en profundidad las ventajas que el sistema proporcionaría y cómo el sistema podía serles fácil y cómodo de usar. El sistema se desarrolló en una interfaz con ambiente *Web*. A partir de lo analizado anteriormente, se arribó a la conclusión de que el sistema es sostenible desde la dimensión socio-humanista.

## **Dimensión Ambiental**

La creación de una cultura y una conciencia ambiental se hace cada día una tarea más ardua, aumentan considerablemente los productos que no tienen en cuenta su impacto negativo en la naturaleza, por lo que la presente investigación plantea una solución razonable ambientalmente.

En la Universidad de Holguín existen condiciones de trabajo para la correcta explotación del sistema, como son: con medidas adecuadas para evitar daños a la

salud que serían mayoritariamente daños en la vista y en la postura de los implicados, efectos provocados por el tiempo en que deben estar intercambiando información.

En este aspecto también se tiene en cuenta el uso de colores e imágenes que resulten agradables y eviten el estrés; las correctas alineaciones de textos y otros elementos en las interfaces como: iluminación, tamaño de letra, espaciamiento entre caracteres, tipografía. Todo esto ayuda a evitar el cansancio visual, los daños en la columna y el estrés de los usuarios al minimizar el tiempo frente al monitor, por lo que se favorece la calidad en las labores productivas. Debido a las características que presenta, no genera contaminación por ruido, ni tiene un impacto directo desfavorable en el medio ambiente.

Se podrán reutilizar los componentes y recursos del sistema, debido a las características de las tecnologías empleadas y a que se tuvo en cuenta la generalidad en el diseño del mismo. A partir de lo analizado anteriormente se arribó a la conclusión de que el sistema es sostenible desde la dimensión ambiental.

### **Dimensión Tecnológica**

Para el empleo del sistema los usuarios se encuentran capacitados y no necesitan de preparación informática, debido a que hacen uso de computadoras para su labor diaria; además el sistema será intuitivo y fácil de usar y poseerá un manual de usuario que facilitará su manipulación. En cuanto a la infraestructura electrónica, el Departamento de Soporte Técnico de la Universidad de Holguín cuenta con los recursos precisos para la implantación y aplicación del sistema. Las características que poseen las computadoras utilizadas por los trabajadores cumplen con los requerimientos necesarios para hacer uso del sistema y además se encuentran conectadas a la red.

Con el fin de facilitar el continuo mantenimiento del sistema, se describieron con comentarios las funciones fundamentales y se le pusieron nombres intuitivos para proporcionar una mejor comprensión y entendimiento del código. Por otra parte, el sistema permitirá su evolución en el tiempo. Además, permitirá cambios, ya sea de mejoras de *hardware*, red e incluso de plataforma. A partir de lo analizado anteriormente se arribó a la conclusión de que el sistema es sostenible desde la

dimensión tecnológica.

### **Conclusiones de la valoración de sostenibilidad**

Una vez analizadas la dimensión administrativa, socio-humanística, tecnológica y ambiental se concluye que la herramienta informática es perdurable en el tiempo. El producto informático permite que el trabajo se realice de una forma más cómoda, se minimice el tiempo en la comunicación y favorece el ahorro de materiales de oficina. Además, prevé continuidad por lo que se puede concluir que es un producto sostenible.

## **2.6 Pruebas**

Las pruebas son en XP muy importantes ya que son las que finalizan el proceso de implementación. Esta etapa también está diseñada para el usuario en conjunto con el programador, esta fórmula genera buenos resultados.

### **2.6.1 Pruebas de aceptación**

Las pruebas de aceptación de los usuarios se describen a partir de casos de pruebas de caja negra. Son aquellas pruebas finales que se realizan antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Además, verifican si el producto satisface los requerimientos del usuario, si se comporta como se desea, según las especificaciones funcionales del diseño. A continuación, aparecen algunas de las pruebas de aceptación realizadas (Pressman, 2013).

Tabla 23: "Caso de prueba de aceptación HU2\_P1".

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Insertar equipo.	
<b>Descripción:</b> Prueba la funcionalidad de insertar equipo en el sistema.	
Condiciones de Ejecución: Se debe contar con el equipo a insertar en el sistema.	
<b>Entrada/Pasos de Ejecución:</b> Después de acceder al sistema se selecciona la opción insertar equipo en el sistema ,esto muestra una segunda página que es un formulario para insertar equipo en el sistema, el administrador da clic en el botón registrar.	
<b>Resultado Esperado:</b> El sistema muestra un mensaje el mismo refleja que se han insertado satisfactoriamente el equipo.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Tabla 24: "Caso de prueba de aceptación HU2\_P2".

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Modificar equipo.	
<b>Descripción:</b> Prueba la opción de modificar equipo en sistema	
<b>Condiciones de Ejecución:</b> El equipo a modificar debe de estar registrado en el sistema.	
<b>Entrada/Pasos de Ejecución:</b> El administrador accede al listado de los equipo, selecciona el equipo a modificar y da clic en el botón modificar, el sistema muestra un alerta y el administrador selecciona sí.	
<b>Resultado Esperado:</b> El sistema redirecciona el administrador al listado de equipo.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Tabla 25: "Caso de prueba de aceptación HU3\_P3".

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P3	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Eliminar equipo.	
<b>Descripción:</b> Prueba la opción de eliminar equipo en el sistema.	
<b>Condiciones de solicitud:</b> El administrador eliminar equipo en el sistema.	
<b>Entrada/Pasos de solicitud:</b> El administrador accede al sistema y Selecciona el equipo y da clic en el botón eliminar, el sistema muestra un mensaje diciendo que el equipo fue eliminado.	
<b>Resultado Esperado:</b> El sistema muestra un mensaje diciendo que el equipo fue eliminado correctamente.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Tabla 26: "Caso de prueba de aceptación HU4\_P4".

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P4	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Listar equipos.	
<b>Descripción:</b> Prueba la opción de listar equipos.	
<b>Condiciones de solicitud:</b> El administrador lista equipos en el sistema y renvía la lista de equipo.	
<b>Entrada/Pasos de solicitud:</b> El administrador accede al sistema y lista equipo y da clic en el botón listar, y el sistema muestra un mensaje diciendo que fue listado el equipo.	
<b>Resultado Esperado:</b> El sistema muestra un mensaje diciendo que fue listado correctamente.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Tabla 27: "Caso de prueba de aceptación HU4\_P5"

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P5	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Insertar incidencias.	
<b>Descripción:</b> Prueba la funcionalidad de insertar los datos de una incidencia en sistema.	
<b>Condiciones de solicitud:</b> Para insertar una incidencia debes de ser el administrador del sistema.	
<b>Entrada/Pasos de solicitud:</b> El administrador accede al sistema y inserta una incidencia y da clic en el botón insertar, y el sistema muestra un mensaje diciendo que la incidencia fue insertado.	
<b>Resultado Esperado:</b> incidencia insertado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 28: "Caso de prueba de aceptación HU4\_P6".

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P6	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Modificar incidencias.	
<b>Descripción:</b> Prueba la funcionalidad de modificar incidencias en el sistema.	
<b>Condiciones de solicitud:</b> Para modificar una incidencia debes de ser el administrador del sistema.	
<b>Entrada/Pasos de solicitud:</b> El administrador accede al sistema y da modificar incidencias, y el sistema muestra un mensaje diciendo que la incidencia fue modificado correctamente.	
<b>Resultado Esperado:</b> incidencias fue modificado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

La metodología XP ofrece como ya se había visto anteriormente las pruebas de aceptación que diseña y ejecuta el cliente y son aceptadas por el equipo de trabajo como acuerdo de trabajo realizado. Lo que quiere decir que el trabajo se considera

completado cuando todas las pruebas de aceptación son satisfactorias para el cliente.

En el caso de la solución propuesta, todas las pruebas resultaron satisfactorias para el cliente, lo que permite confirmar la hipótesis planteada en la introducción de este trabajo.

### **Conclusiones del capítulo**

En el presente capítulo se alcanzaron a las siguientes conclusiones:

- La aplicación desarrollada asiste el trabajo en las diferentes áreas y permite mantener la información actualizada en relación a la realización de los mantenimientos a equipos.
- Las tecnologías y herramientas escogidas para el desarrollo del sistema, por los resultados obtenidos, demuestran adaptarse al desarrollo de sistemas con este propósito.
- El sistema es sostenible según las dimensiones Administrativa, socio-humanista, tecnológica y ambiental, por lo que su utilización resulta sostenible en la Universidad de Holguín.
- La puesta en marcha de forma parcial del sistema facilitó la labor del personal de mantenimiento, pues se agiliza la gestión de reporte de incidencias la que en este se maneja.

## CONCLUSIONES GENERALES

El proceso de desarrollo de la investigación requirió de una amplia revisión bibliográfica con el fin de desarrollar un sistema de gestión de reportes de incidencias, que permitió realizar una correcta modelación de la solución propuesta. La utilización de una adecuada combinación de tecnologías y la metodología de desarrollo XP, facilitó el desarrollo del “Sistema de gestión de reportes de incidencias en la Universidad de Holguín” para gestionar este proceso en el Departamento de Soporte Técnico, potenciando las funcionalidades del mismo. Una vez realizada la valoración de sostenibilidad del sistema de gestión de reportes de incidencias elaborado, se puede afirmar que el mismo es sostenible, según su impacto social, económico, tecnológico y ambiental. El empleo de pruebas de aceptación permitió obtener la aceptación por parte de los usuarios y concluir que la solución propuesta satisface las necesidades que dieron origen a esta investigación.

## RECOMENDACIONES

Para ampliar los resultados obtenidos en la investigación se recomienda continuar perfeccionando el sistema mediante la inclusión de nuevas funcionalidades tales como:

- Generar reportes estadísticos para ayudar en la toma de decisiones.
- Visualizar los equipos a los que se les realizó mantenimiento fuera del plan.
- Vincular el sistema de gestión de reporte de incidencias desarrollado con la familia de sistemas existentes en la Universidad de Holguín a partir de la validación de usuarios a través de LDAP y la retroalimentación de datos del equipamiento.

## REFERENCIAS BIBLIOGRÁFICAS

- Bon, J. (2008). Fundamentos de la gestión de servicios de TI: Basado en ITIL. Gran Bretaña: Van Haren Publishing, Zaltbommel.
- Burback, R., ed. (1998). Software engineering methodology: The watersluice, Phd thesis.
- Cruz, M. (2009). El método Delphi en las investigaciones educativas. La Habana: Editorial Academia.
- Ellis, R. (2011). CodeIgniter Guía del Usuario en Español.
- Figuerola, N. (2012). ITIL v3 ¿Por dónde empezar? .
- FJRP. (2008). Arquitectura Cliente / Servidor
- Foundations ITIL, X. c. (2005). Texto expositivo para Curso ITIL. Buenos Aires.
- Gomero, E. L. O. (2017). Sistema informático para el proceso de gestión de incidencias en la Empresa Insecorp sac.
- Huércano, M. L. y. S. R. (2017). ITIL v3 Manual íntegro.(Diseño de los procesos de gestión de incidencias y servicedek, alineado a las buenas prácticas de itil, aplicado a la empresas delltex industrial s.a.).
- Jacobson, I., G. Booch, and J. Rumbaugh (2000). El Proceso Unificado de Desarrollo de Software.
- Jorge, A. (2017). Sistema de Gestión para el control de las órdenes de trabajo en la Universidad de Holguín.
- Letelier, P. P., M. C. (1989). Metodologías ágiles para el desarrollo de software. eXtreme Programming (XP). Valencia, España: Universidad Politécnica de Valencia.
- Michelena, S. (2007). Tramitación de interrupciones. Cálculo de la disponibilidad y confiabilidad. .
- Montesinos, J. F. (2014). Implementación de un sistema de un sistema de incidencias
- OverTI. (2013). Consultoría y Soluciones para la Gestión de servicios TI (2013). Obtenido.
- Paul, A. D. (2011). ITIL Hero's HandBook

Pressman, R. S. (2013). Ingeniería del Software, un enfoque práctico (Sexta Edic.).  
Renaud, P. (2008). Manual de procedimientos para Soporte Técnico / diagrama despliegue. .  
Trillo, A. M. (2007). Herramienta de gestión de incidencias.  
Winesett, J. M. (2009).frameworkyii.

## GLOSARIO DE TÉRMINOS

**Sistema:** Conjunto de elementos interrelacionados que trabajan juntos para obtener un resultado deseado.

**Incidencia:** Problema informático que tiene un usuario con el sistema que precisa de la ayuda o intervención de un especialista para su resolución.

**Cliente:** Persona que utiliza el sistema informático y que puede tener incidencias en su uso.

**Técnico de Soporte:** Especialista informático encargado de solucionar los problemas que los usuarios tienen con el sistema.

**Intervención:** Acción realizada para solucionar parte o toda la problemática de una incidencia de un usuario.

**HTML:** Es el Lenguaje de Marcado de Hipertexto (HyperText Markup Language), que se utiliza para crear los documentos a los que se accede a través de Navegadores.

**Software:** Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

**Hardware:** Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

**SQL:** *Structured Query Language*. Lenguaje de preguntas estructurado, lenguaje que utiliza bases de datos para pedir información de las misma.

## ANEXO 1: Historias de usuario

Historia de usuario	
<b>Número:</b> 10	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar reporte por departamento	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 6 días	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Gestionar reporte por departamento	
<b>Información adicional:</b> Gestionar reporte por departamento	

Historia de usuario	
<b>Número:</b> 11	<b>Usuario:</b> administrador
<b>Nombre:</b> Gestionar locales.	
<b>Prioridad en el negocio:</b> Alta	<b>Nivel de complejidad:</b> Alta
<b>Estimación:</b> 2 semanas	<b>Iteración asignada:</b> 3
<b>Descripción:</b> El administrador debe ser capaz de gestionar locales	
<b>Información adicional:</b> Gestionar locales	

## ANEXO 2: Tareas

Tarea	
<b>Número:</b> 20	<b>Número de la historia:</b> HU_10
<b>Nombre de la tarea:</b> Gestionar reporte por departamento	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 20 de abril del 2019	<b>Fecha de fin:</b> 23 de abril del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al especialista Gestionar reporte por departamento.	

Tarea	
<b>Número:</b> 21	<b>Número de la historia:</b> HU_11
<b>Nombre de la tarea:</b> Gestionar locales	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 2 días
<b>Fecha de inicio:</b> 20 de abril del 2019	<b>Fecha de fin:</b> 23 de abril del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al especialista Gestionar locales	

Tarea	
<b>Número:</b> 22	<b>Número de la historia:</b> HU_5
<b>Nombre de la tarea:</b> Gestionar mantenimientos realizados a los equipos	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 23 de abril del 2019	<b>Fecha de fin:</b> 26 de abril del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al especialista gestionar mantenimientos realizados a los equipos.	

Tarea	
<b>Número:</b> 23	<b>Número de la historia:</b> HU_6
<b>Nombre de la tarea:</b> Gestionar solicitud de mantenimiento en el año.	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 26 de abril del 2019	<b>Fecha de fin:</b> 29 de abril del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al especialista Gestionar solicitud de mantenimiento.	

Tarea	
<b>Número:</b> 24	<b>Número de la historia:</b> HU_7
<b>Nombre de la tarea:</b> Reportar como atendido la notificación de incidencias	
<b>Tipo de tarea:</b> Desarrollo	<b>Estimación:</b> 3 días
<b>Fecha de inicio:</b> 2 de mayo del 2019	<b>Fecha de fin:</b> 5 de mayo del 2019
<b>Programador responsable:</b> Antonio Baptista Miguel	
<b>Descripción:</b> Implementar una vista que permita al reportar como atendido la notificación de incidencias	

### ANEXO 3: Casos de Prueba de Aceptación

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P6	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Insertar mantenimientos realizados a los equipos	
<b>Descripción:</b> Prueba la funcionalidad de insertar mantenimientos realizados a los equipos	
<b>Condiciones de solicitud:</b> Para poder insertar un nuevo mantenimiento realizados a los equipos en el sistema, debe de ser el Administrador.	
<b>Entrada/Pasos de solicitud:</b> Acceder al sistema, hacer clic en el botón Insertar mantenimientos realizados a los equipos y llenar los campos correspondientes.	
<b>Resultado Esperado:</b> mantenimientos realizados a los equipos insertado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P7	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Modificar mantenimientos realizados a los equipos	
<b>Descripción:</b> Prueba la funcionalidad de modificar mantenimientos realizados a los equipos en el sistema.	
<b>Condiciones de solicitud:</b> Para modificar plan de mantenimiento por equipo debes de ser el Administrador del sistema.	
<b>Entrada/Pasos de solicitud:</b> Ir al botón modificar plan de mantenimiento por equipo elegir el plan de mantenimiento por equipo a modificar. Clic en el botón modificar.	
<b>Resultado Esperado:</b> plan de mantenimiento por equipo fue modificado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P8	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Eliminar mantenimientos realizados a los equipos.	
<b>Descripción:</b> Prueba la funcionalidad eliminar mantenimientos realizados a los equipos en el sistema.	
<b>Condiciones de solicitud:</b> Para eliminar un mantenimiento realizado a los Equipos debes de ser el Administrador en el sistema.	
<b>Entrada/Pasos de solicitud:</b> Ir al botón insertar mantenimientos realizados a los equipos elegir eliminar. clic en el botón eliminar.	
<b>Resultado Esperado:</b> Mantenimientos realizados a los equipos fue eliminado en el sistema	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	