

# Universidad de las Ciencias Informáticas



## Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Título: Herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB en las estaciones de trabajo del CESIM.**

**Autores:** Greicel Martínez Rosa

Luis Miguel Rojas Aguilera

**Tutor:**

Ing. Isledy Gainza Martínez

Ing. Yoandry González Castro

La Habana, junio 2014

## ***Pensamiento***



***“El día que el hombre se dé cuenta de sus profundas equivocaciones, se habrá acabado el progreso de la ciencia.”***

***Charles Chaplin***

## ***Declaración de autoría***

### **Declaración de autoría:**

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Informática Médica de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

---

#### **Firma de la autora**

Greicel Martínez Rosa

---

#### **Firma del autor**

Luis Miguel Rojas Aguilera

---

#### **Firma del tutor**

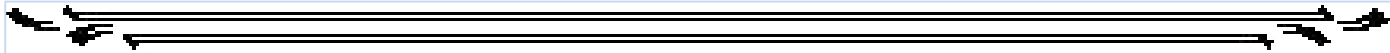
Ing. Isledy Gainza Martínez

---

#### **Firma del cotutor**

Ing. Yoandry González Castro

## ***Datos de contacto***



### **Ing. Yoandry González Castro**

Graduado de Ingeniero en Ciencias Informáticas en el año 2013 en la Universidad de Ciencias Informáticas (UCI). Recién Graduado en Adiestramiento en el Centro de Informática Médica (CESIM). Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desempeña como desarrollador.

### **Ing. Isledy Gainza Martínez**

Graduada de Ingeniería en Ciencias Informáticas en el año 2008 en la Universidad de Ciencias Informáticas (UCI). Posee categoría docente de Instructor. Ha impartido las asignaturas de Sistemas Operativos, Practica Profesional 3, 4, 5 y 6, y la asignatura electiva AISS. Vinculada a la Facultad 7 y se desempeña como analista del departamento Atención Primaria a la Salud en el Centro de Informática Médica (CESIM). Correo electrónico: [igainza@uci.cu](mailto:igainza@uci.cu)

# Agradecimientos

## *Agradecimientos de Greicel*

*En este momento tan especial en el que logro alcanzar uno de mis sueños me siento agradecida a muchas personas, que de una forma u otra estuvieron presentes a lo largo del camino. Personitas que con perseverancia, astucia, dedicación y algunas que sin quererlo lograron que creciera como persona y contribuyeron a convertirme en toda una profesional.*

*Quiero agradecer a:*

*Mis padres, por ser mi razón de ser, mi mayor tesoro. Gracias por estar siempre que los necesite, por apoyarme en todas mis decisiones, por ese ánimo que en los momentos más difíciles me dieron. Este título es por ustedes y para ustedes. Gracias por todo el sacrificio que tuvieron que hacer para apoyarme en mi sueño, por todo lo que me han querido y enseñado. La vida no me alcanzaría para agradecerles, ni para agradecerle a dios por haberme obsequiado tan valioso regalo, unos padres como ustedes.*

*A mi hermanito del alma, la luz de mis ojos. Por todo el amor que me has brindado, por cuidarme, apoyarme y por sobre todo por querer siempre lo mejor para mí. Y por ser mi fuente de inspiración.*

*A mis tías y abuelos por estar siempre conmigo, por guiarme siempre por el mejor camino y por sobre todo apoyarme. A mi abuelita Walkiria que a pesar de no estar conmigo físicamente, está presente en cada paso que doy, un día me dijiste que lo iba a lograr y no te equivocaste.*

*A mi novio Franly, por todo el amor, cariño, comprensión y apoyo brindado. Gracias por estar ahí cada vez que lo necesite, por darme fuerzas y ánimo para seguir adelante. Por ser mi amigo, compañero y confidente. Por llegar a mi vida en el momento preciso.*

*A todos los profesores que durante cinco años contribuyeron a mi formación profesional e integrar. Gracias por todos los conocimientos brindados y por todos los valores que me inculcaron.*

## Agradecimientos

*A Antonia, Chávez, Nanda y a mis suegros Walter y Maite por la ayuda ofrecida y por estar ahí para mí. Por todos los momentos que compartimos juntos y por abrir las puertas de su casa y hacerla sentir como si fuera mía.*

*A mis amigos, los viejos y los que conocí en la universidad, por estar siempre conmigo, por los consejos brindados y los momentos compartidos juntos. En especial a mi hermanito del alma Pedro, por apoyarme y brindarme tu amistad todos estos añitos. Gracias por no defraudarme nunca y por sobre todo aconsejarme y no juzgarme en ningún momento.*

*A mi compañero de tesis Rojas, gracias por ayudarme a convertir en realidad mi sueño, sin ti no lo hubiera conseguido.*

*A mis compañeros de cuarto Ania, Alejandro y Arianne por compartir conmigo este último añito en la universidad. Ustedes fueron como mi tablita en medio del mar.*

*A mis tutores Isledy y Yoandry por el apoyo brindado, a la Revolución y la UCI por permite realizar este sueño.*

*Gracias a todos.*

## Agradecimientos

---

### *Agradecimientos de Luis Miguel*

*A mi familia , en especial a mis padres Idania Aguilera Fernández y Luis Rojas Puro y a mi hermana Yanela Rojas Aguilera por convertirme en la persona que soy con su educación, cariño, entrega, dedicación y aportes tanto material como espiritualmente.*

*A Yadini Pérez López, mi compañera en todo momento y en todos los planos de la vida, por estar conmigo en las buenas y en las malas, siempre que lo necesité, por su amor y comprensión.*

*A Maritza López y Nicasio Pérez quienes me han dado todo su apoyo y se han comportado como unos padres para mí.*

*A mi compañera de tesis por su preocupación y responsabilidad con esta tarea que hemos llevado juntos, por su sinceridad y dedicación.*

*A mis tutores por su preocupación y apoyo en cada paso en la realización de esta tesis.*

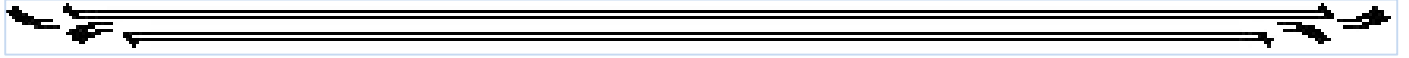
*A mi perro Kinke por todo el cariño que me transmite desinteresada y fielmente y por hacerme reír siempre que lo necesitaba.*

*A mis compañeros de estudio y de cuarto por enseñarme lo interesante y complicada que puede ser la convivencia y por compartir conmigo buenos y malos momentos.*

*A mis amigos por su apoyo y preocupación, muchas gracias.*

*Al tribunal y oponente pues con sus señalamientos hicieron posible alcanzar la calidad de la tesis.*

## **Dedicatoria**



*Dedicatoria de Greicel*

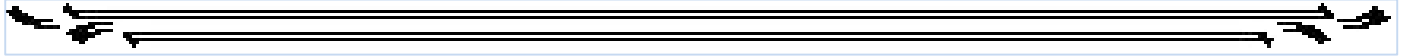
*A mi familia, a ella me debo.*

*Dedicatoria de Luis Miguel*

*A mi familia.*



# *Resumen*



## **Resumen**

En el mundo empresarial, la seguridad de la información es factor fundamental para garantizar el prestigio de las empresas y éxito de los negocios. Al mismo tiempo enfrenta una gran amenaza provocada por la utilización de dispositivos externos de almacenamiento, con los que se puede obtener gran cantidad de información de forma no autorizada de las instituciones. El objetivo de la presente investigación, es desarrollar una herramienta para controlar el acceso y monitorizar el flujo de información en dispositivos externos de almacenamiento. Para el desarrollo de la misma se definieron las herramientas y tecnologías necesarias y a través de la metodología XP (Extreme Programming) se guió el proceso de desarrollo. Se describieron las funcionalidades que debe brindar la herramienta, se realizó el diseño y siguiendo el estilo arquitectónico cliente-servidor se llevó a cabo la implementación para llegar a una solución. Con el objetivo de comprobar el correcto funcionamiento de la herramienta se realizaron pruebas unitarias y de aceptación. La solución propuesta está compuesta por un servicio que estará presente en las estaciones de trabajo cliente, iniciando con el sistema operativo, permitiendo detectar la conexión de un dispositivo y los eventos que se generan sobre el mismo y por una aplicación web, que se encuentra en el servidor, brindando la posibilidad al administrador de gestionar el control de acceso de los dispositivos, visualizar todas las acciones realizadas sobre el dispositivo y filtrar por diferentes criterios de búsqueda.

**Palabras claves:** dispositivos USB; control de acceso; monitorización de flujo de información; seguridad de información.

# Introducción

## Índice

Introducción .....	1
Capítulo 1. Fundamentación teórica .....	6
1.1 Conceptos básicos .....	6
1.1.1 Control de acceso .....	6
1.1.2. Control de dispositivos .....	6
1.1.3. Monitoreo de flujo de información .....	6
1.1.4. USB .....	7
1.2 Herramientas existentes .....	7
1.2.1 Herramientas existentes a nivel internacional .....	7
1.3 Medidas para mejorar el control y seguridad del entorno de desarrollo .....	12
1.4 Herramientas empleadas para el desarrollo de la solución.....	12
1.4.1 Entorno de desarrollo integrado .....	12
1.4.2 Lenguaje de programación.....	13
1.4.3 Plataforma de desarrollo web Symfony 2.0 .....	13
1.4.4 Framework de diseño de interfaces web ExtJs 4.0 .....	14
1.4.5 Gestor de base de datosPgAdmin3.....	14
1.4.6 Servidor de base de datos PostgreSQL .....	15
1.4.7 Visual Paradigm como herramienta de modelado .....	15
1.5 Metodología de desarrollo de software Programación Extrema (XP- por sus siglas en ingles) .....	15
1.5.1 Características de XP .....	16
Conclusiones parciales:.....	18
Capítulo 2. Análisis y diseño .....	19
2.1 Análisis de la solución propuesta.....	19
2.1.1 Requisitos no funcionales .....	20
2.2.1 Historias de usuarios.....	22
2.3 Fase de planificación.....	23
2.3.1 Estimación de esfuerzo por cada historia de usuario .....	23
2.3.2 Plan de iteraciones.....	24

# Introducción

2.3.3 Plan de duración de las iteraciones.....	25
2.4Diseño de la herramienta.....	26
2.4.1 Tarjetas Contenido, Responsabilidad, Colaboración (CRC) .....	26
2.5 Descripción de la arquitectura .....	27
2.5.1 Patrón arquitectónico .....	27
2.5.2 Patrones de diseño .....	29
2.6 Modelo de datos.....	32
Capítulo 3. Implementación y prueba.....	33
3.1 Fase de Implementación .....	33
3.1.1 Estándares de codificación .....	33
3.1.2 Implementación de historias de usuarios por iteraciones .....	35
3.2 Fase de pruebas .....	41
3.2.1 Pruebas unitarias .....	41
3.2.1 Pruebas de aceptación .....	41
Conclusiones parciales .....	43
Conclusiones generales.....	44
Anexos.....	46
Anexo 1.Historias de usuarios .....	46
Anexo 2. Tarjetas CRC.....	51
Anexo 3. Pruebas de aceptación.....	52
Referencias.....	60

## Índice de tablas

Tabla 1. HU Detectar dispositivo .....	22
Tabla 2. Estimación de esfuerzo .....	24
Tabla 3. Plan de duración de iteraciones .....	26
Tabla 4 . Tarjeta CRC AttachDetector .....	27
Tabla 5. Descripción de tarea de ingeniería 1 Detectar dispositivo .....	36
Tabla 6. Descripción de tarea de ingeniería 2 Verificar dispositivo .....	36
Tabla 7. Descripción de tarea de ingeniería 3 Detectar cambios .....	37
Tabla 8. Descripción de tarea de ingeniería 4 Autenticar usuario .....	37
Tabla 9. Descripción de tarea de ingeniería 5 Registrar sucesos .....	38
Tabla 10. Descripción de tarea de ingeniería 6 Enviar al servidor .....	38
Tabla 11. Descripción de tarea de ingeniería 7 Detectar que se desconecta .....	39
Tabla 12. Descripción de tarea de ingeniería 8 Filtrar registros .....	39
Tabla 13. Descripción de tarea de ingeniería 9 Añadir usuario .....	40
Tabla 14 . Descripción de tarea de ingeniería 10 Modificar permisos .....	40
Tabla 15. Descripción de tarea de ingeniería 11 Visualizar trazas .....	41
Tabla 16. Pruebas de aceptación para la HU1 Detectar la conexión de un nuevo dispositivo. ....	42
Tabla 17. Pruebas de aceptación 1 para la HU10 Verificar dispositivo .....	42
Tabla 18. Pruebas de aceptación 2 para la HU10 Verificar dispositivo .....	43
Tabla 19. HU Detectar cambios en el sistema de archivos del dispositivo conectado .....	46
Tabla 20. HU Registrar los sucesos ocurridos .....	47
Tabla 21. HU Enviar al servidor .....	47
Tabla 22. HU Detectar cuando se desconecta un dispositivo .....	48
Tabla 23 . HU Modificar permisos .....	48
Tabla 24. HU Visualizar trazas .....	49
Tabla 25. HU Filtrar registros .....	49
Tabla 26. HU Autenticar usuario .....	50
Tabla 27. HU Verificar que el dispositivo está autorizado .....	50
Tabla 28. HU Añadir usuario .....	51

## Índice de tablas

Tabla 30. Tarjeta CRC MonitorObserver.....	<b>¡Error! Marcador no definido.</b>
Tabla 31. Tarjeta CRC USBWatcher.....	51
Tabla 32. Tarjeta CRC Watcher.....	52
Tabla 33. Tarjeta CRC ActivityHandler.....	52
Tabla 36. Tarjeta CRC USBWatcherClient.....	<b>¡Error! Marcador no definido.</b>
Tabla 38. Pruebas de aceptación para la HU2Detectar cambios en el sistema de archivos.....	53
Tabla 39. Pruebas de aceptación 1 para la HU9Autenticar usuario. ....	53
Tabla 40. Pruebas de aceptación 2 para la HU9Autenticar usuario. ....	54
Tabla 41. Pruebas de aceptación para la HU3Registrar los sucesos ocurridos.....	55
Tabla 42. Pruebas de aceptación para la HU5Detectar cuando se desconecta un dispositivo.....	55
Tabla 43. Pruebas de aceptación para la HU4 Enviar al servidor la base de dato.....	56
Tabla44. Pruebas de aceptación1 para la HU8Filtrar registros. ....	56
Tabla 45. Pruebas de aceptación2 para la HU8Filtrar registros. ....	57
Tabla 46. Pruebas de aceptación3 para la HU8Filtrar registros ....	58
Tabla 47. Pruebas de aceptación para la HU11Añadir usuario .....	58
Tabla 48. Pruebas de aceptación para la HU6Modificar permisos .....	58
Tabla 49. Pruebas de aceptación para la HU7 Visualizar trazas.....	59

## *Índice de figuras*

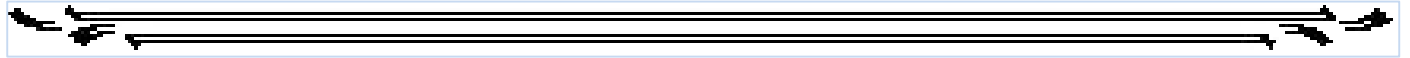


Figura 1. Patrón arquitectónico Cliente-Servidor.....	29
Figura 2. Ejemplo de patrón Experto.....	30
Figura 3. Ejemplo de patrón Controlador .....	30
Figura 4. Ejemplo de patrón Alta cohesión.....	31
Figura 5. Ejemplo de patrón Bajo acoplamiento.....	31
Figura 6. Modelo de datos .....	32

# ***Introducción***

## **Introducción**

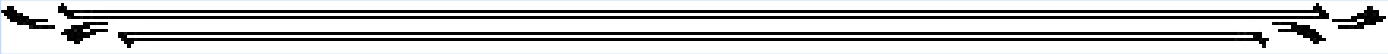
En el mundo actual, las tecnologías de la información y las comunicaciones (TIC) han avanzado sorprendentemente, convirtiéndose en parte importante de la vida diaria de las personas, las organizaciones empresariales e instituciones educativas. Permitiendo la comunicación social, mejorando la calidad del aprendizaje y la productividad económica, así como el desarrollo de la enseñanza y educación, a través de la elaboración de nuevos modelos pedagógicos basados en el uso de las capacidades y potencialidades que ofrecen estos adelantos. La evolución de la tecnología que se logra día tras día, junto a la industria del software, incluyendo aquellas universidades que se encargan del desarrollo de servicios, productos y soluciones informáticas, han llevado a todas las esferas de la sociedad las ventajas del desarrollo de las TIC. Sectores como la educación, salud, cultura, empresarial, prensa y el militar se benefician haciendo uso de productos de software y servicios que ayudan al desarrollo de sus actividades diarias, así como a la toma de decisiones, el incremento productivo y la gestión de información.

A medida que evoluciona la tecnología informática y se hace más grande su influencia en el desarrollo económico y social, surgen acciones ilícitas conocidas como delitos informáticos, que con el desarrollo de los dispositivos, la programación y el internet se han vuelto más frecuentes y sofisticados (1). Delitos como el acceso no autorizado a la información o la modificación, omisión y destrucción de la misma son acciones que enfrentan con frecuencia las organizaciones, pudiendo provocar considerables daños en el ámbito social y económico. Es por ello que se debe garantizar la seguridad de la información que es manejada en su quehacer diario. Esta no es más que el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, disponibilidad e integridad de la misma (2). Esto implica que solamente tengan acceso a la información y permisos para modificarla, la (s) persona autorizada para hacerlo.

La información manejada en cada institución posee una alta importancia para el desarrollo de sus actividades diarias, por lo que mantener una adecuada seguridad contribuiría a alcanzar y mantener el éxito de los negocios. Por tal motivo se debe ver la seguridad de la información como un proceso continuo, identificando y minimizando las vulnerabilidades y mitigando las amenazas. Con frecuencia la información que es utilizada en las instituciones, resulta de interés para terceras personas y para obtenerla, ponen en práctica cualquier técnica o herramienta que tengan a su disposición. En ocasiones

## ***Introducción***

---



algo tan simple como la utilización de un dispositivo externo de almacenamiento puede poner en peligro dicha información. Es precisamente la amenaza que representan estos dispositivos uno de los principales problemas que pueden enfrentar hoy día las mismas. Es por ello que a nivel mundial han ido surgiendo alternativas que permiten el control de los dispositivos externos de almacenamiento de información mediante la conexión USB, desarrollándose así una serie de herramientas que facilitan dicha labor.

Con el objetivo de convertir la informática en una de las ramas más productivas y aportadoras de recursos para el país, como afirmó el comandante Fidel Castro Ruz, se crea la Universidad de las Ciencias Informáticas (UCI), que mediante la vinculación estudio-trabajo como modelo de formación tiene la misión de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la informática. Además de producir aplicaciones y servicios informáticos sirviendo de soporte a la industria cubana de la informática, por lo que no se encuentra ajena a la gran amenaza que representan los dispositivos externos de almacenamiento. Dentro de la Universidad existen diferentes centros encargados del desarrollo de software y servicios informáticos, el Centro de Informática Médica (CESIM) es uno de ellos. El cual tiene la misión de desarrollar productos, servicios y soluciones informáticas para la optimización del trabajo y mejoramiento de la calidad de la atención médica, contribuyendo a la formación integral de profesionales y permitiendo un posicionamiento en el mercado nacional e internacional.

Cada sistema desarrollado en el CESIM, genera un cúmulo considerable de códigos fuente y documentos, encaminados a alcanzar el éxito de cada proyecto. Es por ello que se implementan un conjunto importante de medidas para mejorar el control y seguridad del entorno en el que se genera y gestiona la información. Sin embargo, no se controla la utilización de dispositivos externos de almacenamiento de información (DEA) mediante la conexión USB, con los que se puede sustraer fácilmente información de las estaciones de trabajo. Gran parte del personal perteneciente al CESIM cuentan con DEA, que pueden utilizar sin necesidad de una previa autorización, lo que implica que cualquier persona pueda utilizar el dispositivo que desee en el puesto de trabajo al que tenga acceso, almacenando información sensible perteneciente a los proyectos desarrollados en el centro. La información que es almacenada en el dispositivo puede ser consultada, modificada o eliminada, sin que quede un registro de las acciones que fueron realizadas sobre el dispositivo. Poniendo en peligro la seguridad de la información que es generada, la confidencialidad e integridad de la misma, pudiendo provocar retrasos en la fecha de entrega de los sistemas, que los mismos no se ajusten a las necesidades de los clientes, pues se podría modificar documentos que registren las funcionalidades especificadas por los mismos sin que quede registrado el usuario que realizó el cambio. Todo ello restaría prestigio al centro y significaría además pérdidas económicas para el mismo.



# Introducción

Lo anteriormente expuesto conlleva al planteamiento del **problema científico**: ¿Cómo contribuir a la seguridad de la información gestionada en las estaciones de trabajo del Centro de Informática Médica? Teniendo como **objeto de estudio**: la seguridad de la información gestionada en las estaciones de trabajo del Centro de Informática Médica, delimitado por el **Campo de Acción**: control de acceso y monitorización del flujo de información en los dispositivos USB en las estaciones de trabajo del CESIM y como **objetivo general**: implementar una herramienta que controle el acceso y monitorice el flujo de información en dispositivos de almacenamiento externos conectados por USB, que se utilizan en las estaciones de trabajo del Centro de Informática Médica.

Derivándose los siguientes objetivos específicos:

- ✓ Elaborar el marco teórico de la investigación sobre la herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB en las estaciones de trabajo del CESIM.
- ✓ Realizar el análisis y diseño de la herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB en las estaciones de trabajo del CESIM.
- ✓ Implementar la herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB en las estaciones de trabajo del CESIM.
- ✓ Validar la solución propuesta mediante las pruebas de aceptación y unitarias.

Para dar cumplimiento a los objetivos definidos se plantean las siguientes tareas investigativas:

- ✓ Realización del estudio del estado del arte del tema tratado.
- ✓ Estudio de las políticas de seguridad informática de la universidad.
- ✓ Definición de las herramientas y tecnologías para el desarrollo de la solución.
- ✓ Implementación de las funcionalidades que componen la solución propuesta.
- ✓ Realización de pruebas al sistema desarrollado.

Partiendo de lo antes planteado se define la siguiente **idea a defender**: Con la aplicación de la herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB externos de almacenamiento se permitirá potenciar la seguridad de la información en las estaciones de trabajo del CESIM.

Para la realización de la investigación se emplearon los siguientes métodos:

**Métodos teóricos:**

# Introducción

- ✓ Histórico-Lógico: se utiliza durante todo el proceso de desarrollo de la investigación. Incluye el estudio de los temas relacionados con el control de acceso y monitorización de flujo de información en dispositivos USB y la exploración de las posibles herramientas a utilizar para la solución.
- ✓ Analítico-Sintético: se utilizó para realizar un análisis y síntesis de las políticas y medidas de seguridad implementadas en los centros productivos de la UCI.

## Métodos empíricos:

- ✓ Entrevista no estructurada o libre: se realizó con el objetivo de obtener información referente al control de acceso en dispositivos USB de almacenamiento de información en el CESIM. Es una entrevista en la que se trabaja con preguntas abiertas sin un orden preestablecido, adquiriendo un carácter de conversación.
- ✓ Análisis documental: se utilizó con el objetivo de obtener información mediante la recolección y selección de documentos relacionados con el tema a tratar.

Como resultado de la investigación se realizará la implementación de una herramienta para el control de acceso y monitorización de flujo de información en dispositivos USB externos de almacenamiento, que contribuirá a mantener la seguridad de la información generada en el CESIM.

La presente investigación se estructura de la siguiente forma:

**Capítulo 1 Fundamentación teórica:** en este capítulo se realiza la fundamentación teórica de la investigación. Se lleva a cabo el estudio del arte, a través del cual se realiza una investigación de varias herramientas que permiten el control de acceso y monitorización de flujo de información en dispositivos USB. Además se definen las herramientas y tecnologías a utilizar para darle solución al problema científico planteado en la introducción.

**Capítulo 2 Análisis y diseño:** en este capítulo se realiza el análisis y diseño de la herramienta, proporcionando una descripción de la arquitectura, definiendo los patrones de diseño y arquitectónico utilizados, se definieron todas las funcionalidades y los requerimientos no funcionales con los que debe cumplir la herramienta. Se elaboran las tarjetas CRC correspondientes.

**Capítulo 3 Implementación y prueba:** en este capítulo se implementa la herramienta para el control de acceso y monitorización de flujo de información de dispositivos USB externos de almacenamiento y se especifican los estándares de codificación a tener en cuenta para ello. Se definen las tareas de ingeniería

## ***Introducción***

a realizar para dar solución a las historias de usuarios definidas en el capítulo 2. Además de aplicar pruebas para validar el correcto funcionamiento de la herramienta.

# Capítulo 1. Fundamentación teórica

## Capítulo 1. Fundamentación teórica

En el presente capítulo se tratan conceptos básicos que sirven como base para entender la solución que se propone. Se muestra el estudio del arte donde se analizan los sistemas para el control de acceso y monitorización de flujo de información de dispositivos USB externos existentes y se especifican las herramientas y metodología a utilizar para el desarrollo de la solución.

### 1.1 Conceptos básicos

#### 1.1.1 Control de acceso

El control de acceso constituye un poderoso mecanismo para proteger la información que es almacenada en cada estación de trabajo. Está basado en tres conceptos fundamentales: identificación, autenticación y autorización.

Estos controles constan por lo general de tres pasos:

- Primeramente la identificación del usuario y el dispositivo.
- En segundo lugar, la autenticación, que identifica al usuario y al dispositivo de almacenamiento externo (USB) que intenta acceder a la estación de trabajo.
- En tercer lugar, procede la cesión de derechos, es decir, la autorización, que dota al usuario de privilegios para poder conectar o no el dispositivo externo de almacenamiento en la estación de trabajo. (3)

#### 1.1.2. Control de dispositivos

El control de dispositivos regula el acceso de los dispositivos de almacenamiento externo. El cual ayuda a evitar la pérdida y la fuga de datos, así como mantener y elevar la seguridad de la información. (4)

#### 1.1.3. Monitoreo de flujo de información

El monitoreo de flujo de información hace referencia a registrar todas las acciones que realizan los usuarios sobre la información en los dispositivos. Como copiar, modificar, consultar y eliminar.

## **Capítulo 1. Fundamentación teórica**

### **1.1.4. USB**

Universal Serial Bus (USB- por sus siglas en inglés). Es un concepto de la informática para nombrar al puerto que permite conectar periféricos a una computadora.

La creación USB se remonta a 1996, cuando un grupo de empresas (entre las que se encontraban IBM, Intel y Microsoft) desarrolló el formato para mejorar la capacidad de interconexión de los dispositivos tecnológicos (5). Desde sus inicios los puertos USB constituyen una gran ventaja pues permiten conexión instantánea, alimentación a través del cable, compartición del ancho de banda, configuración automática y son transparentes al software\_(6). Una de sus principales desventajas es la gran amenaza que representan para las empresas, pues estos puertos son la principal vía de fuga de información.

### **1.2 Herramientas existentes**

Con el objetivo de mitigar la gran amenaza que representan los dispositivos USB para las instituciones, a nivel mundial se han ido desarrollando un grupo de herramientas, implementadas para alcanzar un control sobre estos dispositivos y garantizar la seguridad de la información almacenada en las estaciones de trabajo. A continuación se muestran las principales características de dichas herramientas.

#### **1.2.1 Herramientas existentes a nivel internacional**

##### **1.2.1.1 NetWrix USB Blocker**

NetWrix USB Blocker es una herramienta útil que tiene como objetivo bloquear la entrada de dispositivos en toda una red de ordenadores. El programa se encarga básicamente de proteger los puertos USB ante accesos no autorizados. Logrando que nadie pueda extraer información de las estaciones de trabajo. (7)

USB Blocker se caracteriza por ser robusto. Ofrece mecanismos de protección que permiten que no se lleve a cabo ninguna actividad no autorizada en las estaciones de trabajo. Con esta herramienta se puede permitir o prohibir los dispositivos de acuerdo a su marca, modelo o un ID exacto. Ofrece una gran facilidad de despliegue, configuración, mantenimientos y es compatible con Windows 7, Windows Vista, XP y Server 2003 y 2008. (8)

#### **Ventajas de la herramienta USB Blocker:**

- Previene al usuario el uso de dispositivos externos no autorizados.
- La herramienta posee licencia gratuita y pagada.

## Capítulo 1. Fundamentación teórica

- No requiere de la instalación de programas en las PCs clientes.
- No requiere largos períodos de aprendizaje, por lo que se convierte en una herramienta fácil de utilizar.

### Desventajas de la herramienta USB Blocker:

- Al estar en presencia de la versión comercial el usuario deberá pagar por obtener la herramienta y su licencia.
- Si el usuario hace uso de la versión gratuita, no podrá obtener un registro de las actividades desarrolladas por los usuarios sobre el dispositivo.
- Es compatible únicamente con Windows, lo que implica que un usuario que haga uso de sistema operativo libre no podrá utilizar USB Blocker.

### 1.2.1.2 DeviceLock

DeviceLock es una solución de seguridad de prevención de filtrado de datos de terminales, basada en directivas que permiten a los administradores de red controlar de forma centralizada las actividades de carga y descarga de datos a través de dispositivos de equipos locales, protocolos de red y aplicaciones. Con la herramienta se puede negar el acceso de los usuarios no autorizados a los dispositivos conectados a puertos USB, FireWire, adaptadores WiFi y BlueTooth, CD-Roms, disquetes, puertos infrarrojos seriales y paralelos, impresoras de red y locales.

Permite el control total sobre los puertos USB de las estaciones de trabajo. Trata de identificar qué usuarios pueden acceder a puerto y dispositivos sobre un equipo. Es administrable a través de la red y está disponible para utilizarse en Windows NT/2000/XP/Vista/7 y Windows Server 2003/2008. DeviceLock permite gestionar: control de accesos de los usuarios o grupos, crea una lista blanca USB, facilita la realización de auditoría de puertos. (9)

### Ventajas de la herramienta DeviceLock

- Posee una interfaz amigable y fácil de usar.
- Brinda la posibilidad de controlar un gran número de computadoras a través de la red.
- Garantiza la seguridad de la información en cada estación de trabajo.

### Desventajas de la herramienta DeviceLock

## Capítulo 1. Fundamentación teórica

- Es compatible únicamente con Windows, lo que implica que un usuario que haga uso de sistema operativo libre no podrá utilizar DeviceLock.
- Es una herramienta privativa.

### 1.2.1.3 Programa de Vigilancia para Dispositivos USB (USB Devices Monitoring Software)

USB Devices Monitoring Software es capaz de realizar un seguimiento de las actividades en línea y fuera de ella y en unidades USB. Proporciona una poderosa manera de prevenir el acceso en cualquier red. El programa de protección de puertos USB es además capaz de notificar a los administradores de redes a través de sonidos bip si se está produciendo algún acceso no autorizado utilizando dispositivos extraíbles. El software para el bloqueo del puerto USB puede llevar un archivo de bitácora que incluye toda la compleja información que ayudará a los administradores a rastrear las pistas de cualquier acceso no autorizado. La aplicación proporciona el estado de inserción o eliminación, nombre de la computadora cliente o dirección IP, nombre de la unidad USB, fecha u hora de la información recibida en el servidor. La herramienta es compatible con los sistemas operativos Windows 2000, Windows 2003, Windows Vista Starter, Windows Vista Home Basic, Windows Vista Home. Soporta todos los tipos de dispositivos extraíbles tales como unidades flash, pendrives, tarjetas multimedia, iPod, reproductores MP3, cámaras digitales. (10)

#### Ventajas de la herramienta USB Devices Monitoring Software

- Analiza cualquier dispositivo USB.
- Los datos recogidos por la herramienta son muy completos.
- Brinda una interfaz amigable, facilitando su uso.
- La herramienta mantiene archivos de registro de toda la información, los cuales pueden ser fácilmente entendidos por cualquier técnico y usuarios no técnicos.

#### Desventajas de la herramienta USB Devices Monitoring Software

- El monitoreo y registro en tiempo real están disponible únicamente en la versión Pro.
- Es compatible solo con el sistema operativo Windows.
- Es una herramienta privativa.

### 1.2.1.4 USB Flash Block/Unblock

USB Flash Block/Unblock controla el acceso de memorias USB a los ordenadores. Es una herramienta sencilla con la que se puede restringir el uso de memorias USB. La herramienta tiene tres modos de

## **Capítulo 1. Fundamentación teórica**

funcionamiento: el modo Normal, donde se puede leer y escribir sobre cualquier memoria USB; el modo solo Lectura, donde se pueden ver los contenidos de las memorias USB pero no se puede escribir sobre ellas y el modo Disabled, que impide al ordenador reconocer memorias USB externas. Es compatible con los sistemas operativos Windows 2000, Windows XP, Windows 2003, Windows Vista, Windows 7, Windows 8. USB Flash Block/Unblock es una herramienta portable, por tal motivo no necesita ser instalado en la computadora (11).

### **Ventajas de la herramienta USB Flash Block/Unblock**

- Es apto para cualquier tipo de usuario pues brinda una interfaz de manejo fácil.
- En caso de producirse algún tipo de error esta herramienta le enviará al usuario un mensaje de alerta.

### **Desventajas de la herramienta USB Flash Block/Unblock**

- Los usuarios que hacen uso de software libre no pueden utilizar USB Flash Block/Unblock, pues es compatible solamente con Windows.
- Es una herramienta privativa.

#### **1.2.1.5 Endpoint Protector**

Endpoint Protector es la solución de seguridad que permite proteger los datos confidenciales almacenados en las estaciones de trabajo que cuenten con sistemas operativos Mac, Linux y Windows. Permite una gestión completa de dispositivos y puertos, así como el análisis del contenido de los ficheros transferidos desde la estación de trabajo del usuario. Con este software, se pueden habilitar, controlar y auditar puertos USBs, DVDs, Discos Externos, y la información transferida hacia el dispositivo.

### **Ventajas de la herramienta Endpoint Protector**

- Permite registrar la información transferida desde diferentes aplicaciones como E-Mail Clients, Outlook, Lotus Notes, Thunderbird, Web Browsers, Internet Explorer, Firefox, Chrome, Yahoo Messenger, Facebook, etc.
- Con Endpoint Protector se puede obtener una licencia completamente gratuita.
- Es una herramienta multiplataforma.

### **Desventajas de la herramienta Endpoint Protector**



## Capítulo 1. Fundamentación teórica

- La licencia que se obtiene gratis cubre 5 estaciones de trabajo únicamente, imposibilitando su uso a grandes empresas.
- La licencia es legal solamente para negocios pequeños.
- Para poder visualizar los registros realizados por la herramienta el usuario deberá conectarse obligatoriamente al servidor de la empresa McAfee, empresa desarrolladora de Endpoint Protector.

### 1.2.1.5 Módulo Control de dispositivos del Kaspersky Antivirus (KAV)

Control de dispositivos permite configurar restricciones de acceso flexibles a dispositivos de almacenamiento de datos (discos duros, dispositivos extraíbles, transmisores por secuencias, unidades de CD/DVD), dispositivos de transmisión de datos (módems), dispositivos de salida de datos (impresoras) e interfaces de conexión (USB, Bluetooth, IR) (12).

#### Ventajas del módulo Control de dispositivos del KAV

- Garantiza la seguridad, pues permite controlar los buses de conexión, restringiendo el uso de dispositivos.
- Se pueden crear reglas haciendo uso de los id de los dispositivos, que permitan que se utilicen solamente las memorias de una empresa determinada. Así como restringir el uso de determinados dispositivos una vez finalizada la jornada laboral.

#### Desventajas del módulo Control de dispositivos del KAV

- Si el dispositivo está autorizado para utilizarse en las estaciones de trabajo, el módulo no registra las acciones realizadas sobre el mismo.

Una vez realizado el estudio del arte se concluye que no se utilizan ninguna de las herramientas mencionadas anteriormente pues no responde a las necesidades del problema científico planteado. Para poder obtener estas herramientas el usuario deberá pagar, en el caso de la herramienta DeviceLock el cliente podrá obtener una versión completamente gratuita pero deberá limitarse a hacer uso de algunas de las funcionalidades que posee la misma. Ninguno de los sistemas analizados obtiene registro de las actividades realizadas por los usuarios sobre sus dispositivos. Cada sistema corre sobre plataforma Windows, imposibilitando su uso a personas que utilicen sistemas operativos Linux, exceptuando las herramientas Endpoint Protector y KAV pues ambas son herramientas multiplataforma. Endpoint Protector tiene el inconveniente de que la licencia gratuita que se puede obtener solo cubre 5 estaciones de trabajo

## **Capítulo 1. Fundamentación teórica**

y cada laboratorio de producción cuenta con 30 estaciones. Para poder ver los registros realizados por la herramienta, el usuario debe hacer uso obligatoriamente del sitio web y del servidor de la McAfee, empresa desarrolladora de Endpoint Protector. Mientras que el módulo Control de dispositivos del KAV se configura para que bloquee el acceso a los dispositivos que desee el usuario, pero en caso de poder utilizarse el dispositivo, este módulo no registra las acciones que realiza el usuario sobre el mismo.

### **1.3 Medidas para mejorar el control y seguridad del entorno de desarrollo**

En la universidad se implementan un conjunto de políticas que tienen como objetivo fundamental proteger la información. Algunas hacen referencia a la seguridad de la información que se maneja en la institución y en los proyectos productivos de la misma. “Políticas de seguridad informática para el desarrollo de software en la UCI”, es un complemento de las Políticas de seguridad en la universidad, con el objetivo de establecer directivas adicionales a las áreas que intervienen en el proceso productivo, por lo que son aplicables a todos los centros de desarrollo de la UCI. Estas establecen que la información fundamental a proteger en el proceso de producción de software es la que está relacionada con el expediente de los proyectos y los códigos fuentes de las aplicaciones desarrolladas. Deberá garantizarse que toda la información se encuentre en los repositorios centrales de los servidores de producción, así como que las áreas utilizadas para el desarrollo de software estarán protegidas con medidas adecuadas que garanticen el acceso físico exclusivamente del personal autorizado. Se deberá llevar un registro de la entrada y salida del personal en los laboratorios de producción (13). Entre otras medidas que establecen el uso de antivirus, el control de quién está utilizando la computadora en cada instante de tiempo y el empleo de herramientas que permiten el cifrado y el control de versiones.

### **1.4 Herramientas empleadas para el desarrollo de la solución**

Con el objetivo de facilitar el desarrollo del software y de obtener una herramienta capaz de controlar y monitorear de manera eficiente y segura el acceso de los dispositivos USB de almacenamiento y el flujo de información de los mismos en las estaciones de trabajo del CESIM, se utilizaron una serie de herramientas, las que se describen a continuación.

#### **1.4.1 Entorno de desarrollo integrado**

Para el desarrollo de la solución planteada se hizo uso de la herramienta Geany. Este es un entorno de desarrollo ligero, compatible con los sistemas operativos GNU/Linux, Mac OS X, BSD, Solaris y Windows. Dentro de sus utilidades destacan el hecho de que es un entorno compatible con la mayoría de los lenguajes, posee varios paneles para poder acceder mejor a los datos, permite que el desarrollador pueda

## **Capítulo 1. Fundamentación teórica**

ver el estado del programa y los mensajes del compilador. Geany soporta un conjunto de lenguajes dentro de los que se puede encontrar Java, Java Script, PHP, CSS, Python. (14)

Posee varias herramientas para compilar, brindando la posibilidad adicional de ejecutar directamente del entorno, mediante una consola que se integra en el programa. Otra de las ventajas que ofrece Geany es la descomposición, representación de las clases y estructuras del código. Es una herramienta completamente gratuita y de código abierto (15).

### **1.4.2 Lenguaje de programación**

Para llevar a cabo la implementación de la herramienta se utilizó el lenguaje de programación Python. Es un lenguaje dinámico y orientado a objetos. El principal objetivo que persigue es la facilidad, tanto de lectura, como de diseño. Es un lenguaje de programación multiparadigma, pues permite varios estilos como: programación orientada a objetos, programación estructural y funcional. Se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Se integra con otros lenguajes y herramientas. Ofrece varias utilidades tales como: desarrollo de prototipos del sistema, lenguaje integrador para combinar varios componentes de un programa, elaboración de aplicaciones clientes, desarrollos web y de sistemas distribuidos. Una de sus potencialidades es la gran cantidad de librerías que dispone para la realización de diversas tareas como: acceso a ficheros, manejo de cadenas, servicios web, retoque de imágenes, multimedia, interfaces gráficas, XML y acceso a base de datos. Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa, las expresiones pueden ser introducidas una a una, pudiendo verse el resultado de su evaluación inmediatamente. El uso de este lenguaje trae aparejado una serie de ventajas dentro de las que se pueden encontrar: su sencillez y velocidad, sus bibliotecas las que hacen gran parte del trabajo, soporta varias bases de datos, la rapidez de su desarrollo y es un lenguaje de libre distribución. (16)

### **1.4.3 Plataforma de desarrollo web Symfony 2.0**

Symfony es un framework completo diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo-Vista-Controlador. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Todas estas ventajas

## Capítulo 1. Fundamentación teórica

tienen como resultado que no se deba reinventar la rueda cada vez que se crea una nueva aplicación web Symfony. Es compatible con varios gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. Posee además una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo. (17)

### 1.4.4 Framework de diseño de interfaces web ExtJs 4.0

ExtJs es una biblioteca de Java Script para el desarrollo de aplicaciones web interactivas la cual hace flexible el manejo de componentes de la página como el DOM, realiza peticiones AJAX y DHTML y permite crear interfaces de usuario bastante funcionales. ExtJs 4 ofrece la cartografía más avanzada y capacidades gráficas de cualquier marco de Java Script, sin depender de plugins, ofreciendo una visualización perfecta de píxeles en cualquier navegador de cualquier sistema operativo. Utilizar este framework de diseño le ofrece al desarrollador una serie de ventajas tales como:

- Crear aplicaciones complejas utilizando componentes predefinidos.
- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores.
- El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta. (18)

### 1.4.5 Gestor de base de datos PgAdmin3

Es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL y funciona sobre casi todas las plataformas. (19) Permite desde ejecución de consultas SQL simples hasta la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración. La aplicación incluye un editor de la sintaxis SQL y un editor de código del lado del servidor. Una de las características interesantes de PgAdmin3 es que, cada vez que realiza alguna modificación en un objeto, escribe la(s) sentencia(s) SQL correspondiente(s), lo que hace que, además de una herramienta muy útil, sea a la vez didáctica. También incorpora funcionalidades para realizar consultas, examinar su ejecución (como el comando explain) y trabajar con los datos.

## **Capítulo 1. Fundamentación teórica**

### **1.4.6 Servidor de base de datos PostgreSQL**

PostgreSQL es un sistema de base de datos relacional que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Pertenece al ámbito del software libre, está bajo la licencia Berkeley Software Distribution (BSD). Cuenta con versiones para las plataformas \*nix y Windows. Permite la realización de transacciones seguras, vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, implementa internamente lenguajes de consulta de muy alto nivel como son el plpgsql, plperl, el plpython y el c. En PostgreSQL el tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla. El número de índices por tabla es también ilimitado.

### **1.4.7 Visual Paradigm como herramienta de modelado**

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos.

Provee soporte para la generación de código. Tiene integración con diversos IDEs como NetBeans, JDeveloper, Eclipse, JBuilder, así como la posibilidad de realizar ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate.

Es portable y posee gran facilidad de uso. Su diseño se centra en casos de uso y se enfoca al negocio que genera un software de mayor calidad. También tiene disponibilidad en múltiples plataformas. Soporta Business Process Modeling Notation (BPMN) y Subversión. (19)

### **1.5 Metodología de desarrollo de software Programación Extrema (XP- por sus siglas en ingles)**

El desarrollo de un software implica un conjunto de actividades que, de no estar bien organizadas pueden ocasionar conflicto entre las partes involucradas, retraso en el desarrollo del software, hasta poner en peligro el producto final. Por tal motivo se hace necesario determinar que metodología de desarrollo se debe utilizar. Para ello hay que partir de las características que presenta el proyecto que se quiere desarrollar.

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (20). Existen dos grupos, comprendidos por las metodologías ágiles y las tradicionales. El primer grupo pone vital importancia en la

## Capítulo 1. Fundamentación teórica

capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y mantener una buena relación con el cliente. Mientras que el segundo está pensado para el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto (21). El enfoque ágil está concebido para equipos de trabajo de 10 o menos integrantes, el cliente es parte del equipo de desarrollo y todos trabajan en un mismo sitio. Este enfoque responde más a los cambios que ha seguir estrictamente una planificación. Mientras que el enfoque tradicional está concebido para proyectos más grandes, el cliente interactúa con el equipo de desarrollo mediante reuniones y pueden estar distribuidos en varios sitios. Este enfoque centra su atención en cumplir con un plan de proyecto, definido en la etapa inicial del mismo y en generar una extensa documentación.

Para desarrollar la solución propuesta se seleccionó el enfoque ágil, pues se presenta una solución pequeña, sujeta a cambios en los requisitos y se desea obtener el mejor código posible. Como metodología de desarrollo se utilizó Programación extrema (XP- por sus siglas en inglés). Por ser la más representativa de su categoría.

### 1.5.1 Características de XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas, coraje para enfrentar los cambios y se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes. Es una metodología ágil de desarrollo de software, que en vez de planificar, analizar y diseñar para el futuro distante, realiza estas actividades poco a poco, a través de todo el proceso de desarrollo. Tiene como objetivos fundamentales mejorar la productividad en desarrollo de software y garantizar la calidad del mismo, haciendo que este supere las expectativas del cliente. El proceso de desarrollo de XP consta de varias fases que se mencionan a continuación:

**Planeación:** la fase de planeación comienza creando una serie de Historias de usuario que describen las características y funcionalidades requeridas en el software que se desarrollará. Una historia es una descripción de una característica o funcionalidad que el cliente quiere que posea el software que se va a construir (22). Cada historia la escribe el cliente y le asigna una prioridad basándose en el valor de impacto que tiene en el software que se desarrolla.

## Capítulo 1. Fundamentación teórica

---

**Diseño:** el diseño en XP sigue de manera rigurosa el principio “mantenerlo simple”. Siempre se prefiere un diseño simple respecto a otro más complejo. Por otro lado, el diseño ofrece una guía de implementación para una historia como está escrita, ni más ni menos.

**Codificación:** se recomienda que después de diseñar las historias y realizar el trabajo de diseño preliminar no se proceda directamente a la codificación, sino que se desarrollen una serie de pruebas de unidad que ejerciten cada una de las historias que vayan a incluirse en el incremento de software actual.

**Pruebas de aceptación:** también llamadas pruebas del cliente, las especifica el cliente y se centran en las características generales y las funcionalidades del sistema, elementos visibles y revisables por el cliente.

La metodología XP define 12 prácticas, de esas solamente se escogieron 7 para desarrollar la solución propuesta:

**El juego de la planificación:** hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario. Los clientes deciden sobre el tiempo de duración de cada iteración.

**Entregas Pequeñas:** producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con todas las funcionalidades del sistema, pues ya constituye un resultado de valor para el negocio.

**Diseño simple:** se debe diseñar la solución lo más simple que pueda y ser implementada en un momento determinado del proyecto. Lo que permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.

**Refactorización:** es una actividad constante de reestructuración del código con el objetivo de remover duplicaciones de código, mejorar su legibilidad, simplificarlo, hacerlo más flexible a los cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.

**Integración continua:** cada pieza de código es integrada en el sistema una vez que esté lista. Así el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

## **Capítulo 1. Fundamentación teórica**

**Estándares de programación:** XP enfatiza que la comunicación de los programadores es a través del código, por lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

**Programación en parejas:** uno de los principios más importantes, XP define que los programadores escriban sus códigos en parejas compartiendo una misma estación de trabajo.

### **Conclusiones parciales:**

Una vez finalizado el presente capítulo se pudo arribar a las siguientes conclusiones:

- ✓ Las herramientas estudiadas no se consideran una alternativa aplicable pues no se ajustan al proceso de migración a software libre en la universidad. Estas son herramientas privativas y compatibles solo con el sistema operativo Windows. Exceptuando la herramienta Endpoint Protector y KAV que son multiplataforma. Endpoint Protector cuenta con una licencia gratuita pero cubre solamente 5 estaciones de trabajo y cada laboratorio del centro cuenta con 30 estaciones de trabajo como mínimo. Mientras que el módulo Control de dispositivos del KAV se configura para que bloquee el acceso a los dispositivos que desee el usuario, pero en caso de poder utilizarse el dispositivo, este módulo no registra las acciones que realiza el usuario sobre el mismo.
- ✓ La metodología de desarrollo, herramientas y lenguajes seleccionados posibilitará desarrollar una solución que permitirá controlar el acceso y monitorización de flujo de información en dispositivos USB.



## Capítulo 2. Análisis y diseño

### Capítulo 2. Análisis y diseño

El análisis y diseño de un software se lleva a cabo con el objetivo de determinar las necesidades del usuario y poder realizar la especificación de los requisitos que van a servir como base para el desarrollo de la herramienta, los requerimientos de hardware, software, disponibilidad y usabilidad. El análisis y diseño permite al ingeniero realizar el modelado de la herramienta, siguiendo una serie de pasos que permiten describir todos los elementos de la herramienta a construir.

El presente capítulo comprende las fases de exploración, planificación y diseño de la herramienta. Muestra las funcionalidades y los requisitos no funcionales con los que debe contar la misma. Se identifica además el patrón arquitectónico a emplear así como los patrones de diseños utilizados, las tarjetas CRC correspondientes y el modelo de datos.

#### 2.1 Análisis de la solución propuesta

Se propone desarrollar una herramienta que permita controlar el acceso y monitoreo de flujo de información de dispositivos USB externos de almacenamiento. La misma permitirá o bloqueará el acceso de los dispositivos de acuerdo al número de serie, id vendor o id del producto. La herramienta se estará ejecutando en tiempo real en segundo plano, en modo promiscuo, registrando todos los sucesos que ocurren desde que un usuario inserta un dispositivo USB hasta que lo retira. Una vez que el usuario inserta el dispositivo, la herramienta comprueba con los datos que ofrece el control de acceso almacenado en la base de datos, que el dispositivo este autorizado para utilizarse en esa estación de trabajo. En el caso de que no esté autorizado se bloquea el dispositivo, imposibilitando su uso. Si el mismo está autorizado la herramienta comienza a registrar todos los eventos realizados sobre el dispositivo, así como los datos de la computadora, dispositivo y usuario que esté haciendo uso del mismo, el nombre y la ruta completa del archivo y la fecha en que ocurrió el evento, a la base de datos central. Si por problemas de conexión o cualquier otro motivo no se puede enviar al servidor, los datos son almacenados en una base de datos local, la cual haciendo uso de un mecanismo se intenta almacenar en el servidor en determinado intervalo de tiempo. La herramienta contará con una interfaz web en la que el administrador, que será la persona encargada de conceder permisos, podrá ver el registro de todas las acciones que realiza el usuario sobre el dispositivos, podrá filtrar los registros por usuario, dispositivo o estación de trabajo, añadir un nuevo dispositivo a la lista de dispositivos autorizados para conectarse a la estación de trabajo y eliminar dispositivos de dicha lista. El usuario que no es administrador, pero que este registrado en la base de datos podrá acceder a la aplicación web pero solamente verá la lista de trazas.

## Capítulo 2. Análisis y diseño

A continuación se muestran las funcionalidades con las que debe contar la herramienta y que serán descritas posteriormente en las historias de usuarios.

**HU1.** Detectar dispositivo.

**HU2.** Detectar cambios en el sistema de archivos del dispositivo conectado.

**HU3.** Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.

**HU4.** Enviar al servidor los eventos detectados.

**HU5.** Detectar cuando se retira el dispositivo.

**HU6.** Modificar permiso del dispositivos.

**HU7.** Visualizar trazas.

**HU8.** Filtrar registros.

**HU9.** Autenticar usuario.

**HU10.** Verificar dispositivo.

**HU11.** Adicionar usuario.

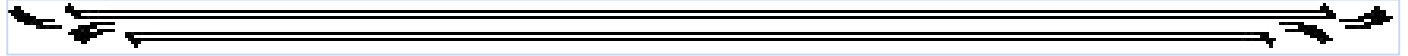
### 2.1.1 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. A través de estos se especifican propiedades del sistema como restricciones de ambiente, desarrollo, rendimiento, dependencias de plataformas y mantenibilidad. Los requisitos no funcionales son las características que hacen al producto atractivo, usable, rápido y confiable (23). En el presente documento se hace referencia a los requisitos de software, hardware, seguridad, disponibilidad y usabilidad.

#### 2.1.1.1 Requisitos de hardware

Para la instalación de la herramienta se debe contar con un microprocesador con una capacidad similar a un Intel Pentium. Las estaciones de trabajo cliente deben contar como mínimo con 1 GB de RAM. El servidor de base de datos debe tener como mínimo con 2 GB de memoria RAM y 400 GB de espacio en disco. Mientras que el servidor de aplicaciones debe contar con 2 GB de memoria RAM.

## Capítulo 2. Análisis y diseño



### 2.1.2.2 Requisitos de software

Las estaciones de trabajo deben contar con sistemas operativos Linux o Windows. El servidor de aplicaciones debe tener un servidor web Apache versión 2.2, así como PHP en su versión 5.3. Mientras que el servidor de base de datos contará con PostgreSQL. Las estaciones clientes solo requieren un navegador web.

### 2.1.2.3 Seguridad

La herramienta está diseñada para monitorear todos los cambios realizados por el usuario al sistema de archivos del dispositivo de almacenamiento de información que se conecte en la estación de trabajo, el usuario que haga uso de la estación de trabajo no podrá cerrar la herramienta ni manipular los datos recogidos. Solamente el administrador podrá acceder a todos los registros, que serán almacenados en la base de datos y tomar alguna acción en el caso lo requiera. Para poder acceder a la aplicación web el usuario deberá autenticarse, por tal motivo solamente podrá acceder a la misma la persona autorizada para hacerlo.

### 2.1.2.4 Usabilidad

Se estará en presencia de una herramienta fácil de usar, por lo que la persona encargada de trabajar con la misma se familiarizará en un tiempo reducido con la herramienta.

### 2.1.2.5 Disponibilidad

La herramienta debe mantenerse funcionando las 24 horas del día y los 7 días de la semana. La misma estará monitoreando constantemente los dispositivos de almacenamiento que sean conectados a las estaciones de trabajo.

## 2.2 Fase de exploración

En la fase de exploración, los clientes plantean a grandes rasgos las Historias de usuario que son de interés para la primera entrega del producto. 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 (24).

## Capítulo 2. Análisis y diseño

### 2.2.1 Historias de usuarios

En cuanto a la información contenida en cada historia de usuarios, existen varias planillas que pueden ser utilizadas, pero aún no se ha llegado a un consenso sobre cual emplear. En algunas se recoge nombre y descripción del requisito, en otras se registra solamente la descripción, otras recogen fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado, cosas por terminar y comentarios. Las Historias de usuarios de la herramienta que se quiere desarrollar contendrán los siguientes datos: nombre y descripción del requisito, prioridad, número de la historia, puntos estimados, iteración asignada y observaciones en los casos que lo requieran. En la tabla 1 se muestra un ejemplo de las historias especificadas. La descripción de las restantes Historias de usuarios se encuentra en el anexo 1.

Historia de usuario	
<b>Número:</b> 1	<b>Nombre de historia:</b> Detectar dispositivo
<b>Usuario:</b>	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Alta	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Cuando el dispositivo es conectado a la estación de trabajo, se muestra un mensaje indicándolo.	
<b>Observaciones:</b> Cuando el dispositivo es detectado la herramienta puede permitir o bloquear el acceso del mismo.	

**Tabla 1. Detectar dispositivo (elaboración propia)**

**Número:** posee el número asignado a la Historia de usuario.

**Nombre de historia:** atributo que contiene el nombre de la historia de usuario.

**Usuario:** usuario del sistema que protagoniza o utiliza la historia. Este puede ser el administrador y en caso de que el atributo se encuentre vacío, el sistema es el que ejecuta la acción.

**Prioridad:** prioridad que tiene la Historia de usuario para el negocio.

## Capítulo 2. Análisis y diseño

**Puntos estimados:** este atributo no es más que una estimación realizada por el equipo de desarrollo del tiempo de duración de cada Historia de usuario. Un punto equivale a una semana de trabajo.

**Iteración asignada:** no es más que la iteración a la que fue asignada la historia, para ser implementada.

**Descripción:** este atributo contiene una breve descripción de la Historia de usuario que se desea implementar.

**Observaciones:** este atributo contiene alguna aclaración que se deba hacer de la Historia de usuario.

### 2.3 Fase de planificación

La fase de planificación se encarga de estimar el esfuerzo necesario para desarrollar cada una de las historias de usuarios que fueron definidas anteriormente, así como la prioridad que tiene cada una de ellas, la cual es establecida por el cliente. Dicha prioridad puede ser alta, media o baja. Una vez definidos estos aspectos se procede a elaborar el plan de entrega, que no es más que definir por cada iteración que Historias de usuarios se debe implementar.

#### 2.3.1 Estimación de esfuerzo por cada Historia de usuario

Las estimaciones de esfuerzo asociado a la implementación de las historias las establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana de programación, las historias generalmente valen de 1 a 3 puntos (25).

La tabla 2 muestra la estimación de esfuerzo por cada una de las Historias de usuarios que fueron definidas en la fase de exploración.

Historias de usuarios	Puntos de estimación
Detectar dispositivo.	1 semana
Detectar cambios en el sistema de archivos del dispositivo conectado.	2 semana
Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.	3 semanas
Enviar al servidor los eventos detectados.	2 semanas

## Capítulo 2. Análisis y diseño

Detectar cuando se desconecta un dispositivo.	1 semana
Modificar permisos del dispositivo.	1 semana
Visualizar trazas.	1 semana
Filtrar registros.	2 semana
Autenticar usuario	1 semana
Verificar dispositivo	2 semanas
Adicionar usuario	1 semana

**Tabla 2. Estimación de esfuerzo (elaboración propia)**

### 2.3.2 Plan de iteraciones

El Plan de iteraciones está compuesto por iteraciones de no más de tres semanas. Al comienzo de cada iteración el cliente deberá seleccionar las Historias de usuario que serán implementadas, para ello se tiene en cuenta la prioridad de cada historia y la relación que existe entre ellas en cuanto a funcionalidad. Se realizó la herramienta en 6 iteraciones las cuales se describen a continuación:

**Iteración 1:** se desarrollan las Historias de usuarios de alta prioridad HU1, HU10. Las mismas se encargarán de detectar cuando se conecta un nuevo dispositivo y de verificar que el mismo este autorizado para conectarse en la estación de trabajo.

**Iteración 2:** se desarrollarán las HU2 y HU9 que tienen una alta prioridad para el negocio. Las mismas se encargan de detectar cambios en el sistema de archivo del dispositivo conectado y de pedir autenticación.

**Iteración 3:** tiene como objetivo implementar la Historia de usuario HU3, la cual tiene alta prioridad. La misma se encarga de registrar todas las acciones realizadas sobre el dispositivo en una base de datos local.

**Iteración 4:** se implementarán las HU4 y HU5, que tienen media y baja prioridad respectivamente. Las mismas se encargan de enviar al servidor los eventos detectados y detecta cuando un dispositivo ha sido desconectado.

## Capítulo 2. Análisis y diseño

**Iteración 5:** se desarrollarán las HU11 y HU8 que tienen baja prioridad. Las mismas se encargan de filtrar las trazas por usuarios, dispositivos o estaciones de trabajo y Añadir un usuario.

**Iteración 6:** se desarrollan las Historias de usuario de baja prioridad, HU6 y HU7. Las mismas se encargan de modificar los permisos de un dispositivo y visualizar las trazas registradas hasta el momento.

### 2.3.3 Plan de duración de las iteraciones

Una vez culminado el plan de iteraciones se prosigue a elaborar el plan de duración de cada una de las iteraciones definidas. Este plan tiene como objetivo mostrar la duración que tendrá cada iteración y el orden en que serán implementadas teniendo en cuenta la prioridad definida por el cliente.

Iteraciones	Historias de usuarios	Duración de las iteraciones (semanas)
Iteración 1	Detectar dispositivo.  Verificar que el dispositivo está autorizado.	3
Iteración 2	Detectar cambios en el sistema de archivo del dispositivo conectado.  Autenticar usuario.	3
Iteración 3	Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.	3

## Capítulo 2. Análisis y diseño

Iteración 4	Enviar al servidor los eventos detectados. Detectar cuando se desconecta un dispositivo.	3
Iteración 5	Filtrar registros. Añadir usuario	3
Iteración 6	Modificar permisos del dispositivo. Visualizar trazas.	2

Tabla 3. Plan de duración de iteraciones (elaboración propia)

### 2.4 Diseño de la herramienta

La metodología XP hace énfasis en los diseños simples y claros, para ello se recomienda no adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se está trabajando.

XP hace uso de tarjetas Contenido, Responsabilidad, Colaboración (CRC) para la descripción del sistema. En esta fase se definen el patrón arquitectónico utilizado, los patrones de diseños empleados durante la implementación de la herramienta y se muestra el modelo de datos perteneciente a la solución.

#### 2.4.1 Tarjetas Contenido, Responsabilidad, Colaboración (CRC)

EL objetivo de las tarjetas CRC es hacer un inventario de las clases que se van a necesitar para implementar el sistema y la forma en que van a interactuar, facilitando el análisis y discusión de las



## Capítulo 2. Análisis y diseño

mismas por parte de varios actores del equipo de proyecto, con el objeto de que el diseño sea simple (26). Cada tarjeta está estructurada de la siguiente forma:

**Nombre de la clase:** hace referencia a las clases de la herramienta que se quiere desarrollar.

**Responsabilidad:** hace refiere a la función que realiza la clase dentro de la herramienta.

**Colaboración:** hace referencia a la relación que tiene una clase determinada con otras clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

Durante todo el proceso de diseño de la herramienta, se confeccionaron un total de 10 tarjetas CRC. En la tabla 4 se muestra una de ellas. En el Anexo 2 se podrá encontrar ejemplos de otras tarjetas. El resto queda reflejado en el artefacto Tarjetas CRC.

Tarjeta CRC	
<b>Nombre de la clase:</b> AttachDetector	
<b>Responsabilidad:</b> Se encarga de monitorear la gestión de dispositivos para detectar cuando un nuevo dispositivo se conecte.	<b>Colaboración:</b> threading.Thread

Tabla 4. Tarjeta CRC AttachDetector (elaboración propia)

### 2.5 Descripción de la arquitectura

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación (27). La arquitectura establece la estructura, funcionamiento e interacción entre las partes del software.

#### 2.5.1 Patrón arquitectónico

Los patrones de arquitectura están orientados a representar los diferentes elementos que componen una solución de software y las relaciones entre ellos, junto a un conjunto de restricciones de cómo estos elementos pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural

## Capítulo 2. Análisis y diseño

---

esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones (28).

Para el desarrollo de la herramienta se empleó el patrón arquitectónico cliente-servidor. El cual es una red de comunicaciones en la que los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta y que pone a disposición de los clientes cada vez que estos son solicitados. El cliente por su parte es quien inicia el diálogo mediante el envío de peticiones al servidor, es por ello que tiene un papel activo en la comunicación. Está diseñado para soportar la interacción con el usuario final y no requiere equipos de altas prestaciones. Mientras que el servidor es la parte pasiva pues espera por las peticiones realizadas por los clientes, procesa dichas peticiones, envía una respuesta, gestiona y comparte los recursos con los clientes que sirve. La cantidad de clientes a los que presta servicios puede ser variable (29). El uso del patrón cliente-servidor permite la centralización del control pues el acceso, los recursos y la integridad de los datos son controlados por el servidor de manera tal que, estos no puedan ser accedidos, ni dañados por clientes no autorizados. Otro de sus beneficios es que favorece escalabilidad pues en cualquier momento se pueden añadir nuevos clientes y servidores o bien estos pueden ser mejorados.

En la herramienta a desarrollar, en el cliente se contará con un servicio que inicia con el sistema operativo. Para desarrollar sus funciones hace uso de varios servicios del sistema operativo tales como inotify, udev y wmi. Una vez que se conecta un dispositivo a la estación de trabajo, la aplicación hace uso del servicio udev en el caso de Linux o wmi en el caso de Windows para detectar dicha conexión. Una vez que se detecta el dispositivo la aplicación utiliza un servicio web que se encuentra en el servidor para obtener los datos del control de acceso de la base de datos central, donde se relaciona qué usuario puede utilizar qué dispositivo en qué estación de trabajo. Si el dispositivo está autorizado para utilizarse, la aplicación comienza a detectar todos los eventos que realiza el usuario sobre el mismo, haciendo uso del servicio del sistema inotify en Linux o wmi en Windows. Los eventos detectados se almacenan directamente en la base de datos central que se encontrará en el servidor. Si por problemas de conexión o algún otro motivo no se pueden almacenar los eventos en el servidor, estos se almacenan en una base de datos local que se encontrará en las estaciones de trabajo cliente, la cual mediante un mecanismo intenta almacenarse en el servidor cuando pasa cierto intervalo de tiempo. En las estaciones clientes haciendo uso de un navegador se accede a la aplicación web con la que contará la herramienta, la cual se encontrará en el servidor. La aplicación permitirá gestionar los permisos, así como mostrar los registros de todas las acciones realizadas por el usuario sobre los dispositivos.

## Capítulo 2. Análisis y diseño

La comunicación del cliente con el servidor se hará haciendo uso de un servicio presentando credenciales mediante un certificado x.509 firmado por una autoridad certificadora creada para la solución, los datos serán cifrados antes de ser enviados. En la figura 1 se muestra como se relacionan cada uno de los elementos mencionados anteriormente.

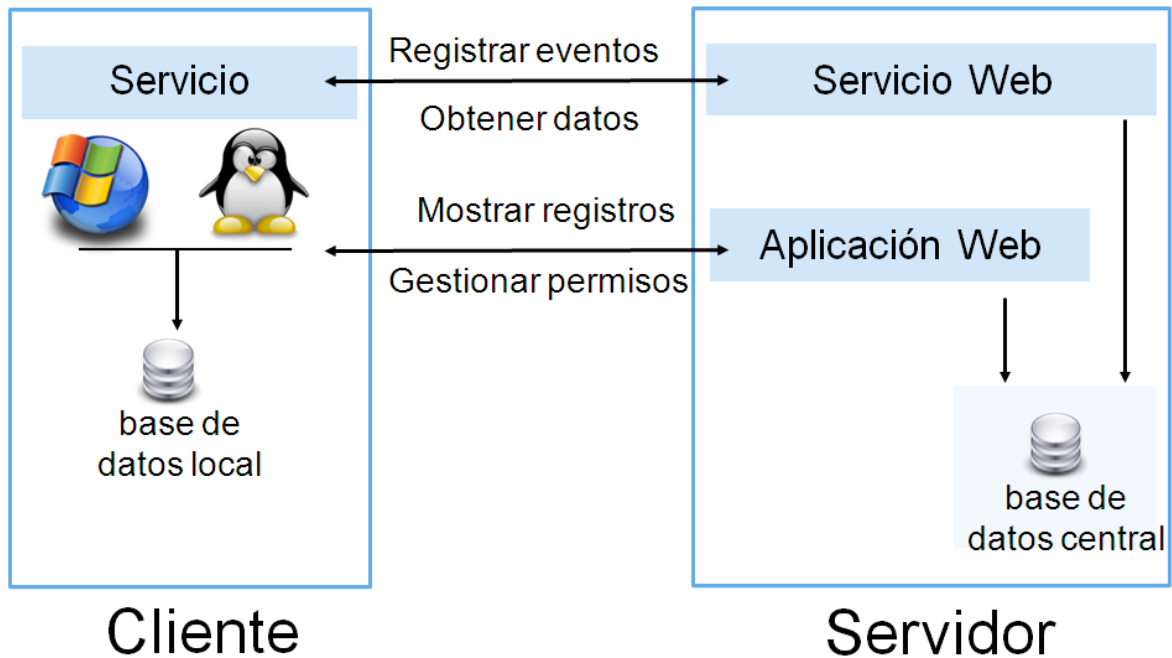


Figura 1. Patrón arquitectónico Cliente-Servidor (elaboración propia).

### 2.5.2 Patrones de diseño

Los patrones de diseños brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Son un conjunto de estrategias o buenas prácticas, que pueden facilitar el trabajo en muchas situaciones a la hora de realizar una aplicación. Además de ser relativamente fáciles de comprender. Hacen más fácil el trabajo, y sobre todo, hacen el código más legible. Los patrones de diseño son independientes del lenguaje en el que se utilicen y generalmente se representan como diagramas UML (30).

Para el desarrollo de la solución se emplearon una serie de patrones de diseños pertenecientes al conjunto de los patrones GRASP. Estos son patrones que permiten la asignación de responsabilidades y ayudan a refinar el diseño y a asignar las responsabilidades a las distintas clases, haciéndolas más sencillas y reutilizables. A continuación se explican los patrones empleados en el desarrollo de la solución.

## Capítulo 2. Análisis y diseño

**Experto:** es el principio básico de asignación de responsabilidades. Este patrón indica que las responsabilidades deben ser asignadas a aquellas clases que cuentan con la información necesaria para llevarlas a cabo.

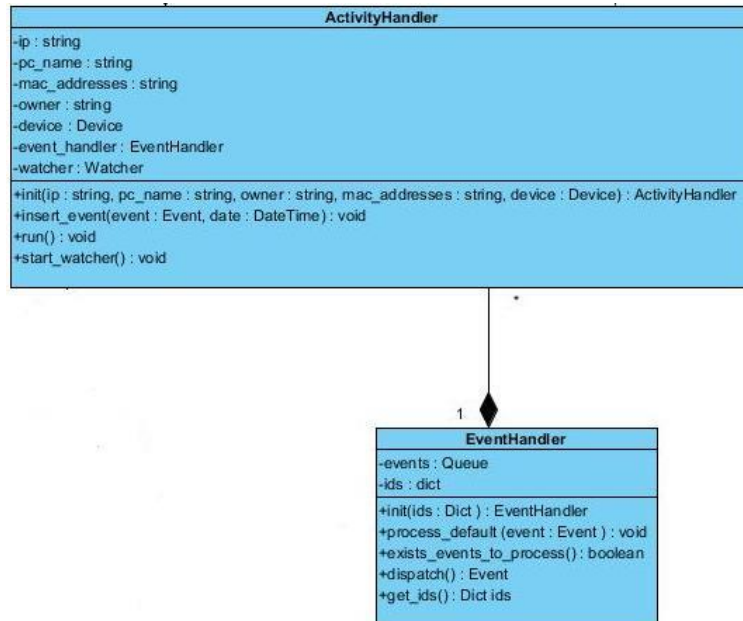


Figura 2. Ejemplo de patrón Experto (elaboración propia)

**Controlador:** este patrón se encarga de que una clase actúe como intermediaria para el manejo de eventos. Con el objetivo de aumentar la reutilización de código y a la vez tener un mayor control (31).

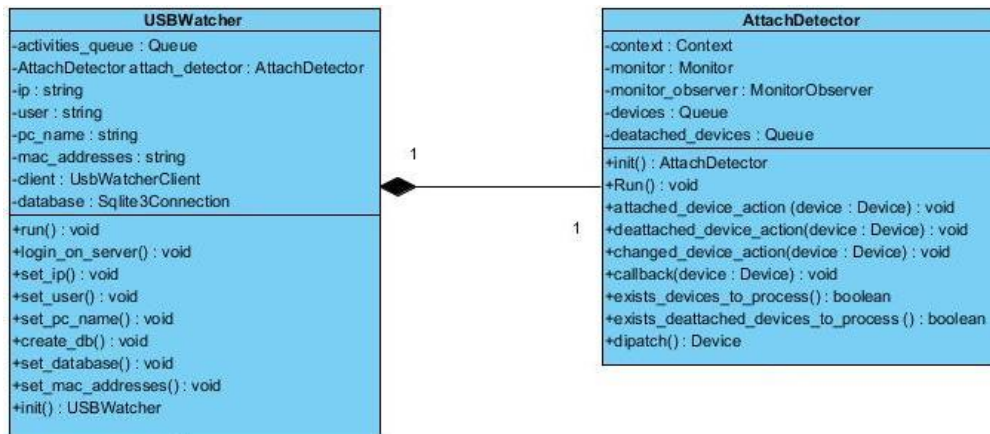


Figura 3. Ejemplo de patrón Controlador (elaboración propia)

## Capítulo 2. Análisis y diseño

**Alta cohesión:** cada elemento del diseño realiza una labor única dentro del sistema. Los algoritmos que desarrollan las principales funcionalidades de la herramienta deben estar distribuidos entre las diferentes clases, evitando la sobrecarga de responsabilidad en alguna de ellas.

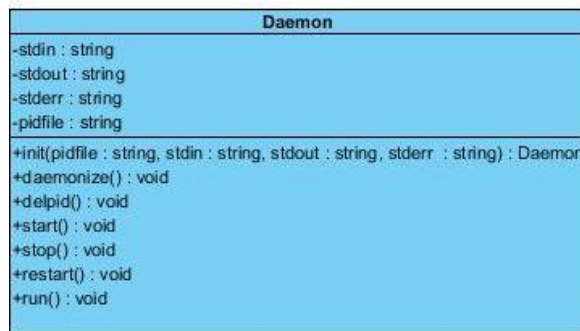


Figura 4. Ejemplo de patrón Alta cohesión (elaboración propia)

**Bajo acoplamiento:** hace referencia a que se logre la menor dependencia entre las clases. Permitiendo que en caso de que se produzca alguna modificación en alguna de las clases afecte lo menos posible el resto, favoreciendo la reutilización.

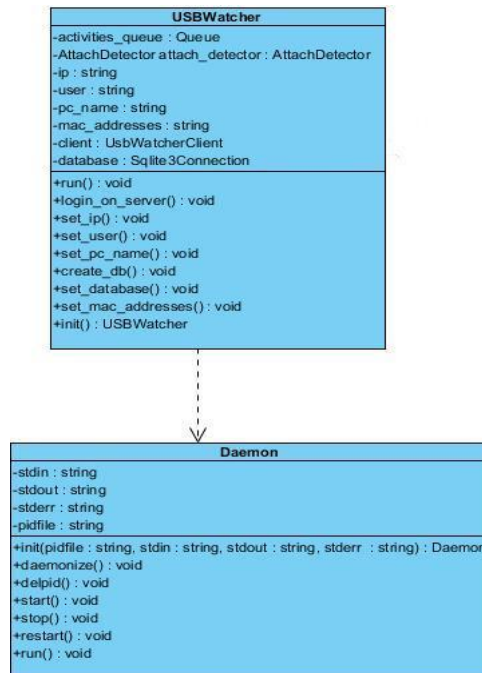


Figura 5. Ejemplo de patrón Bajo acoplamiento (elaboración propia)

## Capítulo 2. Análisis y diseño

### 2.6 Modelo de datos

Un modelo de datos es un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. (32). En la figura 7 se muestra el modelo correspondiente a la solución propuesta.

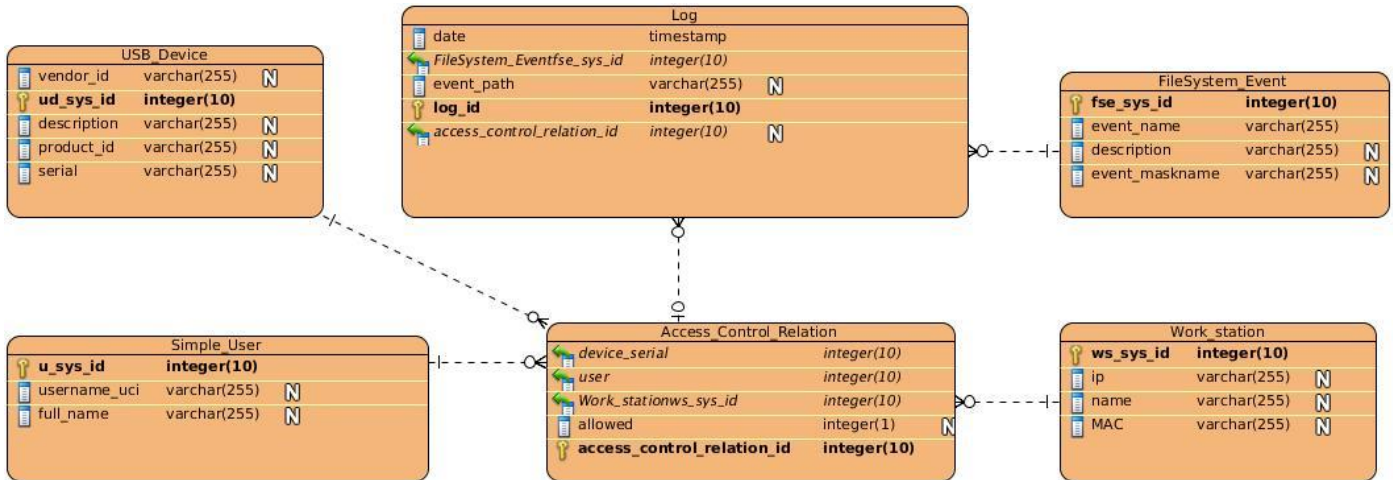


Figura 6. Modelo de datos (elaboración propia)

Cada evento en el sistema de archivos (FileSystem\_Event) es realizada por un usuario (Simple\_User) sobre un dispositivo (USB\_Device) en una estación de trabajo (Work\_station) en un espacio de tiempo determinado (Date\_time). De ahí que en la tabla Log se registren los identificadores correspondientes a cada elemento que ejecuta el evento. Mientras que la tabla Acces\_Control\_Relation se encarga de especificar que usuario puede utilizar que dispositivo en que estación de trabajo.

### Conclusiones parciales

Una vez concluido el presente capítulo se puede arribar a las siguientes conclusiones:

- ✓ Con la creación de las Historias de usuarios se logró identificar las funcionalidades que debe cumplir la herramienta a desarrollar.
- ✓ El diseño de la solución realizado satisface los requerimientos del cliente y representa la base sobre la cual se llevará a cabo la implementación.

## Capítulo 3. Implementación y prueba

---

### Capítulo 3. Implementación y prueba

Dentro del proceso de desarrollo de software la fase de implementación es la encargada de implementar todas las Historias de usuarios definidas por el cliente. La metodología XP define que la implementación debe realizarse haciendo uso de estándares de codificación manteniendo el código fácilmente entendible para todo el equipo de desarrollo. Por otro lado la fase de prueba permite comprobar que el trabajo realizado por los desarrolladores se corresponde con lo definido por el usuario, para ello se hace uso de las pruebas de aceptación y pruebas unitarias, creadas en base a las historias implementadas.

En el presente capítulo se definen los estándares de codificación utilizados para la implementación, se especifican las tareas de ingenierías realizadas, así como las pruebas realizadas a la herramienta, con el objetivo de verificar que se cumplió con lo establecido por el cliente.

### 3.1 Fase de Implementación

#### 3.1.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (33). Al comenzar la implementación de un sistema de software se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

Para el desarrollo de la herramienta para el control de acceso y monitorización de flujo de información de dispositivos USB se emplearon los siguientes estándares de codificación:

**Identación:** en el lenguaje informático la indentación equivale a la sangría en el lenguaje natural. Así como para el lenguaje formal, cuando se realiza una redacción, se debe respetar ciertas sangrías, los lenguajes informáticos, requieren una indentación.

En el desarrollo de la solución se indenta utilizando 1 tabulación dentro de cada bloque de código, o sea, luego de la declaración de un método, al declarar un bucle, cumpliendo con las normas de indentación que exige el lenguaje de programación Python. A continuación se muestra un ejemplo de la aplicación de este estándar dentro del código de la herramienta:

## Capítulo 3. Implementación y prueba

```
def start_watcher (self):  
    print "\n","Starting Watcher...","\n"  
    try:  
        self.watcher.start ()  
    except Exception as e:  
        print "\n","Error Ocurrred Starting Watcher ", " Message: {}".format (e.message), "\n"
```

**Nombrado de los símbolos:** se empleó para establecer un esquema para nombrar las funciones, clases, atributos de las clases, parámetros y así evita utilizar nombres establecidos en otras partes. En el código de la solución para los nombres de las clases se utiliza siempre letra inicial mayúscula en cada palabra que compone el nombre y no existen espacios entre palabras. A continuación se muestra un ejemplo de la aplicación de este estándar dentro del código de la herramienta:

```
class USBWatcher (Daemon)  
    class AttachDetector (threading.Thread)  
    class ActivityHandler(threading.Thread)
```

Para los nombres de los atributos y métodos se utiliza siempre letra minúscula y separación entre letras con un underscore, como se muestra a continuación:

```
def attached_device_action(self, device)  
    self.activities_queue=Queue()
```

**Inicialización:** en la implementación se inicializan las variables donde se declaran, la única razón para no realizarlo así es que su valor dependa de cálculos que deban realizarse. A continuación se muestra un ejemplo de la aplicación de este estándar dentro del código de la herramienta:

```
self.activities_queue=Queue ()  
self.attach_detector=AttachDetector ()  
self.ip=ip  
self.pc_name=pc_name  
self.mac_addresses=mac_addresses  
self.owner=owner  
self.device=device
```



## Capítulo 3. Implementación y prueba

**Comentarios:** estos se realizan con el objetivo de que el código sea más fácil de entender, en el desarrollo de la solución se comenta utilizando tres comillas dobles, como se muestra a continuación:

```
""" Clase para detectar cuando un nuevo dispositivo ha sido conectado en la estación de trabajo """
```

```
class AttachDetector(threading.Thread)
```

**Espacios dentro del código:** con el objetivo de facilitar la lectura del código, es recomendable señalar un espacio de separación entre los diferentes elementos. En el desarrollo de la solución entre la declaración de la clase y la declaración del primer método y entre cada bloque de código perteneciente a cada método existirá un espacio de una línea. Dentro del bloque de código se escribirá una línea debajo de la otra.

### 3.1.2 Implementación de Historias de usuarios por iteraciones

En la fase de exploración se definieron las Historias de usuarios que integran la solución, la misma está compuesta por 11 historias. Cada historia será dividida en tareas más pequeñas conocidas como tareas de ingeniería. Las cuales pueden estar escritas en un lenguaje técnico, no necesariamente entendible por el cliente, serán las que guíen la implementación y serán asignadas a un programador responsable de su implementación.

Teniendo en cuenta la planificación realizada en el capítulo anterior, se definieron 6 iteraciones. A cada una de ellas se le asignaron varias Historias de usuarios. A continuación se muestran las tareas de ingenierías realizadas por iteración.

#### 3.1.2.1 Iteración 1

La iteración 1 implementa las HU1 y 10. Encargadas de detectar la conexión de un nuevo dispositivo y de verificar que el mismo este autorizado para conectarse a la estación de trabajo. Ambas tienen una prioridad alta.

Tarea de ingeniería	
Número de tarea:1	Historia de usuario (No. 1): Detectar la conexión de un nuevo dispositivo.
Nombre de la tarea: Detectar dispositivo	
Tipo de tarea: Desarrollo	Puntos estimados:1
Fecha de inicio:1 de enero del 2014	Fecha de fin:8 de enero del 2014

## Capítulo 3. Implementación y prueba

<b>Programador responsable:</b> Luis Miguel Rojas Aguilera
<b>Descripción:</b> Una vez que el usuario conecta un dispositivo, la herramienta lo detecta mostrando una interfaz indicándole al usuario que el dispositivo se ha conectado.

**Tabla 5. Descripción de tarea de ingeniería 1 Detectar dispositivo (elaboración propia)**

Tarea de ingeniería	
<b>Número de tarea:</b> 2	<b>Historia de usuario (No. 10):</b> Verificar dispositivo.
<b>Nombre de la tarea:</b> Verificar dispositivo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 10 de enero del 2014	<b>Fecha de fin:</b> 23 de enero del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> Una vez que el usuario conecta un dispositivo, la herramienta verifica que el dispositivo se encuentra en la lista de dispositivos autorizados para conectarse en la estación de trabajo. Si el dispositivo no está autorizado, el usuario no podrá trabajar con él.	

**Tabla 6. Descripción de tarea de ingeniería 2 Verificar dispositivo (elaboración propia)**

### 3.1.2.2 Iteración 2

La iteración 2 implementa las HU2 y 9 que tienen una alta prioridad para el negocio. Las mismas se encargan de detectar cambios en el dispositivo conectado y de pedir autenticación.

Tarea de ingeniería	
<b>Número de tarea:</b> 3	<b>Historia de usuario (No. 2):</b> Detectar cambios en el sistema de archivos del dispositivo conectado.
<b>Nombre de la tarea:</b> Detectar cambios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2

## Capítulo 3. Implementación y prueba

<b>Fecha de inicio:</b> 25 de enero del 2014	<b>Fecha de fin:</b> 10 de febrero del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> Una vez que se comprueba que el dispositivo está autorizado para conectarse a la estación de trabajo, la herramienta debe detectar todos los cambios que realiza el usuario sobre el dispositivo.	

**Tabla 7. Descripción de tarea de ingeniería 3Detectar cambios (elaboración propia)**

Tarea de ingeniería	
<b>Número de tarea:</b> 4	<b>Historia de usuario (No. 9):</b> Autenticar usuario.
<b>Nombre de la tarea:</b> Autenticar usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 12 de febrero del 2014	<b>Fecha de fin:</b> 19 de febrero del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> En la aplicación web con que cuenta la herramienta el usuario debe autenticarse, introduciendo el usuario y contraseña. La herramienta verifica que los datos estén correctos y permite el acceso a la misma.	

**Tabla 8. Descripción de tarea de ingeniería 4Autenticar usuario (elaboración propia)**

### 3.1.2.3 Iteración 3

La iteración 3 implementa la Historia de usuario HU3, la cual tiene alta prioridad. La misma se encarga de registrar todas las acciones realizadas sobre el dispositivo en una base de datos local.

Tarea de ingeniería	
<b>Número de tarea:</b> 5	<b>Historia de usuario (No. 3):</b> Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.

## Capítulo 3. Implementación y prueba

<b>Nombre de la tarea:</b> Registrar sucesos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 21 de febrero del 2014	<b>Fecha de fin:</b> 17 de marzo del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> Cuando la herramienta detecta algún cambio en el dispositivo conectado y por alguna razón no puede enviarse directamente al servidor, se almacenan en una base de datos local.	

**Tabla 9. Descripción de tarea de ingeniería 5 Registrar sucesos (elaboración propia)**

### 3.1.2.4 Iteración 4

La iteración 4 implementa las HU4 y 5, que tienen media y baja prioridad respectivamente. Las mismas se encargan de enviar al servidor los eventos detectados y detectar cuando un dispositivo ha sido desconectado.

Tarea de ingeniería	
<b>Número de tarea:</b> 6	<b>Historia de usuario (No. 4):</b> Enviar al servidor los eventos detectados.
<b>Nombre de la tarea:</b> Enviar al servidor	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 19 de marzo del 2014	<b>Fecha de fin:</b> 4 de abril del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> Una vez que el usuario realiza alguna acción sobre el dispositivo, el evento se registra directamente en el servidor.	

**Tabla 10. Descripción de tarea de ingeniería 6 Enviar al servidor (elaboración propia)**

Tarea de ingeniería	
<b>Número de tarea:</b> 7	<b>Historia de usuario (No. 5):</b> Detectar cuando se desconecta un dispositivo.

## Capítulo 3. Implementación y prueba

<b>Nombre de la tarea:</b> Detectar que se desconecta	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 6 de abril del 2014	<b>Fecha de fin:</b> 12 de abril del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> Una vez que el usuario retira el dispositivo de la estación de trabajo, la herramienta muestra un mensaje indicándolo.	

**Tabla 11. Descripción de tarea de ingeniería 7 Detectar que se desconecta (elaboración propia)**

### 3.1.2.5 Iteración 5

La iteración 5 desarrolla las HU11 y 8 que tienen baja prioridad. Las mismas se encargan de filtrar los registros por usuarios, dispositivos o estaciones de trabajo y eliminar dispositivos de la lista de dispositivos autorizados para conectarse a la estación de trabajo.

<b>Tarea de ingeniería</b>	
<b>Número de tarea:</b> 8	<b>Historia de usuario (No. 8):</b> Filtrar registros.
<b>Nombre de la tarea:</b> Filtrar registros.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 14 de abril del 2014	<b>Fecha de fin:</b> 30 de abril del 2014
<b>Programador responsable:</b> Luis Miguel Rojas Aguilera	
<b>Descripción:</b> La herramienta muestra una interfaz en la que el usuario autorizado para trabajar con la misma, podrá filtrar los registros de las acciones realizadas por el usuario sobre el dispositivos de acuerdo a un usuario, una estación de trabajo o un dispositivo.	

**Tabla 12. Descripción de tarea de ingeniería 8 Filtrar registros (elaboración propia)**

<b>Tarea de ingeniería</b>	
<b>Número de tarea:</b> 9	<b>Historia de usuario (No. 11):</b> Añadir usuario.
<b>Nombre de la tarea:</b> Añadir usuario	

## Capítulo 3. Implementación y prueba

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 1 de mayo del 2014	<b>Fecha de fin:</b> 7 de mayo del 2014
<b>Programador responsable:</b> Greicel Martínez Rosa	
<b>Descripción:</b> La herramienta muestra una interfaz en la que la persona autorizada podrá añadir un nuevo usuario a la lista de los autorizados a utilizar un dispositivo en una estación de trabajo.	

**Tabla 13. Descripción de tarea de ingeniería 9Añadir usuario (elaboración propia)**

### 3.1.2.6 Iteración 6

La iteración 6 implementa las Historias de usuario de baja prioridad, HU6 y 7. Las mismas se encargan de añadir un dispositivo a la lista de los dispositivos autorizados para acceder a las estaciones de trabajo y ver los registros de las acciones realizadas por el usuario sobre el dispositivos.

Tarea de ingeniería	
<b>Número de tarea:</b> 10	<b>Historia de usuario (No.6):</b> Modificar permisos del dispositivo.
<b>Nombre de la tarea:</b> Modificar permisos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 8 de mayo del 2014	<b>Fecha de fin:</b> 14 de mayo del 2014
<b>Programador responsable:</b> Greicel Martínez Rosa	
<b>Descripción:</b> La herramienta muestra una interfaz en la que la persona autorizada podrá modificar los permisos de un dispositivo.	

**Tabla 14. Descripción de tarea de ingeniería 10Modificar permisos (elaboración propia)**

Tarea de ingeniería	
<b>Número de tarea:</b> 11	<b>Historia de usuario (No.7):</b> Visualizar trazas.
<b>Nombre de la tarea:</b> Visualizar trazas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1

## Capítulo 3. Implementación y prueba

<b>Fecha de inicio:</b> 15 de mayo del 2014	<b>Fecha de fin:</b> 21 de mayo del 2014
<b>Programador responsable:</b> Greicel Martínez Rosa	
<b>Descripción:</b> La herramienta muestra una interfaz en la que la persona autorizada podrá ver todos los registros de las acciones realizadas por el usuario sobre el sistema de archivo de los dispositivos.	

**Tabla 15. Descripción de tarea de ingeniería 11 Visualizar trazas (elaboración propia)**

### 3.2 Fase de pruebas

Dentro del proceso de desarrollo de software juega un papel importante la fase de pruebas, pues son las que permiten determinar que tan bien implementadas están las funcionalidades definidas por el cliente. Las pruebas son una pieza clave para probar el sistema, dejarlo libre de errores y alcanzar la satisfacción del cliente.

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que se implementan. Todos los módulos del sistema que se desea desarrollar deben pasar las pruebas antes de ser liberados. Para ello XP divide las pruebas en unitarias y de aceptación.

#### 3.2.1 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del mismo. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o re-codificar parte del mismo (34). Las pruebas unitarias son realizadas generalmente por el programador, pues él conoce con mayor detalle el código generado. Se probaron cada una de las funcionalidades del sistema verificando que cada una de ellas funciona correctamente, arrojando resultados satisfactorios.

#### 3.2.1 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las Historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una Historia de usuario ha sido implementada correctamente. Las pruebas son consideradas como pruebas de caja negra

## Capítulo 3. Implementación y prueba

y son las encargadas de comprobar que el sistema cumple con todas las funcionalidades definidas por el cliente.

Los clientes son los encargados de verificar que los resultados obtenidos con las pruebas definidas son correctos. Una Historia de usuario no se puede considerar terminada hasta que no pase todas las pruebas de aceptación. De no pasar las pruebas el cliente debe definir la prioridad en que las historias deben ser resueltas. Para cada Historia de usuario perteneciente a una iteración se definieron pruebas, a continuación se muestran las pruebas de aceptación pertenecientes a la iteración 1, el resto se describe en los anexos.

Caso de prueba de aceptación	
<b>Código:</b> HU1_p1	<b>Historia de usuario (No.1):</b> Detectar dispositivo
<b>Nombre:</b> Conectar dispositivo.	
<b>Descripción:</b> Se desea probar que la herramienta reconozca cuando se conecta un nuevo dispositivo.	
<b>Condiciones de ejecución:</b> Conectar un dispositivo.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta reconoce el dispositivo	
<b>Resultado:</b> Muestra un mensaje indicando que el dispositivo ha sido conectado.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 16. Pruebas de aceptación para la HU1 Detectar la conexión de un nuevo dispositivo (elaboración propia).

Caso de prueba de aceptación	
<b>Código:</b> HU10_p1	<b>Historia de usuario (No.10):</b> Verificar dispositivo.
<b>Nombre:</b> Verificar dispositivo.	
<b>Descripción:</b> Se desea probar que la herramienta verifique si el dispositivo conectado puede utilizarse en la estación de trabajo.	
<b>Condiciones de ejecución:</b> Dispositivo no autorizado.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta busca en la listas de dispositivos autorizados para utilizarse en la estación de trabajo.	
<b>Resultado:</b> La herramienta bloquea el acceso al dispositivo.	
<b>Evaluación de la prueba:</b> Satisfactorio	

Tabla 17. Pruebas de aceptación 1 para la HU10 Verificar dispositivo (elaboración propia).



## Capítulo 3. Implementación y prueba

Caso de prueba de aceptación	
<b>Código:</b> HU10_p2	<b>Historia de usuario (No.10):</b> Verificar dispositivo.
<b>Nombre:</b> Verificar dispositivo.	
<b>Descripción:</b> Se desea probar que la herramienta verifique si el dispositivo conectado puede utilizarse en la estación de trabajo.	
<b>Condiciones de ejecución:</b> Dispositivo autorizado.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta busca en la listas de dispositivos autorizados para utilizarse en la estación de trabajo.	
<b>Resultado:</b> La herramienta permite el acceso al dispositivo.	
<b>Evaluación de la prueba:</b> Satisfactorio	

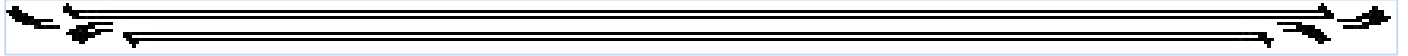
Tabla 18. Pruebas de aceptación 2 para la HU10 Verificar dispositivo (elaboración propia).

### Conclusiones parciales

Al finalizar el presente capítulo se arribo a las siguientes conclusiones:

- ✓ Con la implementación de la herramienta se garantiza el control de acceso y la monitorización de flujo de información en los dispositivos USB. Cumpliendo así con los requisitos establecidos por el cliente.
- ✓ Los resultados obtenidos durante las pruebas de aceptación y las pruebas unitarias evidenciaron el correcto funcionamiento de la aplicación.

## **Conclusiones generales**

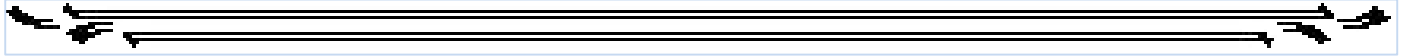


### **Conclusiones generales**

Una vez finalizado el presente Trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resultando que:

- ✓ Se seleccionaron la metodología de desarrollo, lenguajes y herramientas adecuadas para culminar en el tiempo establecido y con la calidad requerida la solución propuesta, teniendo en cuenta la cantidad de personas involucradas en el desarrollo, así como la magnitud de la solución.
- ✓ El análisis y diseño realizado permitió el desarrollo de una solución enfocada a las necesidades del cliente.
- ✓ La implementación de la herramienta posibilitó obtener un producto funcional capaz de contribuir a la seguridad de la información gestionada en las estaciones de trabajo.
- ✓ Las pruebas realizadas permitieron verificar el correcto funcionamiento de la herramienta así como el ajuste a las necesidades y requisitos del cliente.

## ***Recomendaciones***



Como parte del proceso de desarrollo de la investigación se recomiendan los siguientes aspectos:

- ✓ Con el objetivo de elevar la seguridad de la herramienta se recomienda implementar una política de seguridad en las estaciones de trabajo en la que el usuario tenga los permisos necesarios para realizar las tareas que se le asignan y no más, logrando que no intervengan en el funcionamiento de la herramienta.

# Anexos

## Anexos

### Anexo 1.Historias de usuarios

Historia de usuario	
<b>Número:</b> 2	<b>Nombre de historia:</b> Detectar cambios en el sistema de archivos del dispositivo conectado
<b>Usuario:</b>	<b>Puntos Estimados:</b> 2
<b>Prioridad:</b> Alta	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Una vez que el dispositivo fue aceptado, la herramienta detecta cada uno de los cambios realizados en el sistema de archivos del dispositivo, tales como copiar, eliminar, detectar los archivos que son accedidos.	
<b>Observaciones:</b> Esto ocurre cuando no se bloquea el acceso de los dispositivos que se quieren conectar.	

**Tabla 19.Detectar cambios en el sistema de archivos del dispositivo conectado (elaboración propia)**

Historia de usuario	
<b>Número:</b> 3	<b>Nombre de historia:</b> Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.
<b>Usuario:</b>	<b>Puntos Estimados:</b> 3
<b>Prioridad:</b> Alta	<b>Iteración asignada:</b> 2

## Anexos

**Descripción:**

Cuando la herramienta detecta algún cambio en el sistema de archivos del dispositivo y por algún motivo no se puede registrar directamente en el servidor, este es almacenado en una base de datos local.

**Observaciones:** Esta historia de usuario depende de los eventos detectados en la historia de usuario 2.

**Tabla 20. Registrar los sucesos ocurridos (elaboración propia)**

Historia de usuario	
<b>Número:</b> 4	<b>Nombre de historia:</b> Enviar al servidor los eventos detectados.
<b>Usuario:</b>	<b>Puntos Estimados:</b> 2
<b>Prioridad:</b> Media	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Una vez que el usuario realiza algún evento en el sistema de archivos del dispositivo este es almacenado en el servidor.	
<b>Observaciones:</b> Esto ocurre cuando el usuario realiza alguna acción.	

**Tabla 21. Enviar al servidor (elaboración propia)**

## Anexos

Historia de usuario	
<b>Número:</b> 5	<b>Nombre de historia:</b> Detectar cuando se desconecta un dispositivo.
<b>Usuario:</b>	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Baja	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Una vez desconectado el dispositivo la herramienta muestra un mensaje indicándolo.	
<b>Observaciones:</b>	

**Tabla 22. Detectar cuando se desconecta un dispositivo (elaboración propia)**

Historia de usuario	
<b>Número:</b> 6	<b>Nombre de historia:</b> Modificar permisos de dispositivo.
<b>Usuario:</b> Administrador	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Baja	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Permite modificar los permisos de los dispositivos registrados en la base de datos.	
<b>Observaciones:</b>	

**Tabla 23. Modificar permisos (elaboración propia)**

## Anexos

Historia de usuario	
<b>Número:</b> 7	<b>Nombre de historia:</b> Visualizar trazas.
<b>Usuario:</b> Administrador	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Baja	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Permite ver los registros de las acciones realizadas por el usuario sobre el sistema de archivo de los dispositivos hasta el momento.	
<b>Observaciones:</b>	

**Tabla 24. Visualizar trazas (elaboración propia)**

Historia de usuario	
<b>Número:</b> 8	<b>Nombre de historia:</b> Filtrar registros.
<b>Usuario:</b> Administrador	<b>Puntos Estimados:</b> 2
<b>Prioridad:</b> Baja	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Permite filtrar los registros de las acciones realizadas por el usuario sobre el sistema de archivo de los dispositivos por usuarios, estaciones de trabajo o dispositivos.	
<b>Observaciones:</b>	

**Tabla 25. Filtrar registros (elaboración propia)**

## Anexos

Historia de usuario	
<b>Número:</b> 9	<b>Nombre de historia:</b> Autenticar usuario.
<b>Usuario:</b> Administrador	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Alta	<b>Iteración asignada:</b> 4
<b>Descripción:</b> El administrador debe autenticarse para trabajar con la herramienta. Esto permite que solo la persona autorizada pueda acceder a la misma.	
<b>Observaciones:</b>	

Tabla 26. Autenticar usuario (elaboración propia)

Historia de usuario	
<b>Número:</b> 10	<b>Nombre de historia:</b> Verificar dispositivo.
<b>Usuario:</b>	<b>Puntos Estimados:</b> 2
<b>Prioridad:</b> Alta	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Una vez conectado el dispositivo la herramienta verifica si está autorizado para ser utilizado en esa estación de trabajo.	
<b>Observaciones:</b>	

Tabla 27. Verificar que el dispositivo está autorizado (elaboración propia)



## Anexos

Historia de usuario	
<b>Número:</b> 11	<b>Nombre de historia:</b> Añadir usuario
<b>Usuario:</b> Administrador	<b>Puntos Estimados:</b> 1
<b>Prioridad:</b> Baja	<b>Iteración asignada:</b> 5
<b>Descripción:</b> Permite al administrador añadir un nuevo usuario a la lista de los usuarios autorizados a trabajar en una estación de trabajo.	
<b>Observaciones:</b>	

Tabla 28. Añadir usuario (elaboración propia)

### Anexo 2. Tarjetas CRC

Tarjeta CRC	
<b>Nombre de la clase:</b> USBWatcher	
<b>Responsabilidad:</b> Con ayuda otras clases, USBWatcher se encarga de detectar cuando se conecta un dispositivo, si está autorizado o no y permite además que la aplicación se ejecute en segundo plano. Así como identificar los eventos que se generan en el sistema de archivo del dispositivo. Esta es la clase controladora. Permite además verificar si existen eventos registrados localmente para enviarlos al servidor.	<b>Colaboración:</b> AttachDetector Deamon ActivityHandler DeviceValidator ServerData Persistence UploadDataThread

Tabla 31. Tarjeta CRC USBWatcher (elaboración propia)

## Anexos

Tarjeta CRC	
<b>Nombre de la clase:</b> Watcher	
<b>Responsabilidad:</b> Se comunica con la aplicación del sistema de Linux llamada inotify que notifica los cambios producidos en el sistema de archivos.	<b>Colaboración:</b> EventHandler

Tabla 32. Tarjeta CRC Watcher (elaboración propia)

Tarjeta CRC	
<b>Nombre de la clase:</b> ActivityHandler	
<b>Responsabilidad:</b> La clase ActivityHandler se encarga de gestionar los eventos que son detectados y los envía al servidor. En caso de que no pueda enviarse al servidor la clase hace uso de la clase Persistence para almacenar los eventos localmente.	<b>Colaboración:</b> Watcher EventHandler ServerData Persistence

Tabla 33. Tarjeta CRC ActivityHandler (elaboración propia)

### Anexo 3. Pruebas de aceptación

#### Iteración 2

Caso de prueba de aceptación

## Anexos

<b>Código:</b> HU2_p1	<b>Historia de usuario (No.2):</b> Detectar cambios en el sistema de archivos del dispositivo conectado.
<b>Nombre:</b> Cambios en el sistema de archivos.	
<b>Descripción:</b> Se desea probar que la herramienta detecte todos los cambios realizados en el sistema de archivo del dispositivo conectado.	
<b>Condiciones de ejecución:</b> Realizar cambios.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta detecta todos los cambios realizados.	
<b>Resultado:</b> La herramienta detecta todos los cambios realizados al sistema de archivo del dispositivo.	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla 38. Pruebas de aceptación para la HU2Detectar cambios en el sistema de archivos (elaboración propia).**

Caso de prueba de aceptación	
<b>Código:</b> HU9_p1	<b>Historia de usuario (No.9):</b> Autenticar usuario.
<b>Nombre:</b> Introducir datos	
<b>Descripción:</b> Se desea probar que la herramienta permita el acceso solo a la persona autorizada (la persona con rol de administrador).	
<b>Condiciones de ejecución:</b> Datos incorrectos.	
<b>Entrada/ Pasos de ejecución:</b> Usuario no autorizado. La herramienta comprueba que el usuario y la contraseña pertenezcan a la persona autorizada para trabajar con la herramienta.	
<b>Resultado:</b> La herramienta deniega el acceso a la persona, mostrando el siguiente mensaje “El usuario o la contraseña es incorrecta”.	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla 39. Pruebas de aceptación 1 para la HU9Autenticar usuario (elaboración propia).**

## Anexos

Caso de prueba de aceptación	
<b>Código:</b> HU9_p2	<b>Historia de usuario (No.9):</b> Autenticar usuario.
<b>Nombre:</b> Introducir datos	
<b>Descripción:</b> Se desea probar que la herramienta permita el acceso a la herramienta solo a la persona autorizada (la persona con rol de administrador).	
<b>Condiciones de ejecución:</b> Datos correctos.	
<b>Entrada/ Pasos de ejecución:</b> Usuario autorizado. La herramienta comprueba que el usuario y la contraseña pertenezcan a la persona autorizada para trabajar con la herramienta.	
<b>Resultado:</b> La herramienta permite el acceso a la persona.	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla 40. Pruebas de aceptación 2 para la HU9Autenticar usuario (elaboración propia).**

### Iteración 3

Caso de prueba de aceptación	
<b>Código:</b> HU3_p1	<b>Historia de usuario (No.3):</b> Registrar los sucesos ocurridos en el sistema de archivos, en una base de datos local.
<b>Nombre:</b> Registrar sucesos.	
<b>Descripción:</b> Se desea probar que la herramienta registre todos los sucesos ocurridos en el sistema de archivo del dispositivo en una base de datos local.	
<b>Condiciones de ejecución:</b> Realizar cambios.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta detecta todos los cambios realizados.	
<b>Resultado:</b> Se registren todos los cambios realizados al sistema de archivo del dispositivo en una base de datos local.	
<b>Evaluación de la prueba:</b> Satisfactorio	

## Anexos

Tabla 41. Pruebas de aceptación para la HU3 Registrar los sucesos ocurridos (elaboración propia).

### Iteración 4

Caso de prueba de aceptación	
<b>Código:</b> HU5_p1	<b>Historia de usuario (No.5):</b> Detectar cuando se desconecta un dispositivo.
<b>Nombre:</b> Desconectar dispositivo.	
<b>Descripción:</b> Se desea probar que la herramienta detecte cuando se desconecta un dispositivo.	
<b>Condiciones de ejecución:</b> Desconectar dispositivo.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta detecta que el dispositivo ha sido desconectado.	
<b>Resultado:</b> La herramienta muestra un mensaje indicando que el dispositivo ha sido desconectado.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 42. Pruebas de aceptación para la HU5 Detectar cuando se desconecta un dispositivo (elaboración propia).

Caso de prueba de aceptación	
<b>Código:</b> HU4_p1	<b>Historia de usuario (No.6):</b> Enviar al servidor los eventos detectados.
<b>Nombre:</b> Almacenar en el servidor los eventos.	
<b>Descripción:</b> Se desea probar que la herramienta envíe al servidor los cambios realizados por el usuario.	
<b>Condiciones de ejecución:</b> Realizar cambios en el sistema de archivo del dispositivo.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta registra los cambios en la base de datos central.	

## Anexos

**Resultado:** La herramienta envía los eventos hacia el servidor.

**Evaluación de la prueba:** Satisfactorio

**Tabla 43. Pruebas de aceptación para la HU4 Enviar al servidor la base de dato (elaboración propia).**

### Iteración 5

Caso de prueba de aceptación	
<b>Código:</b> HU8_p1	<b>Historia de usuario (No.8):</b> Filtrar registros.
<b>Nombre:</b> Filtrar registros por usuario.	
<b>Descripción:</b> Se desea probar que la herramienta filtre los registros por usuarios.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción filtrar registros.	
<b>Entrada/ Pasos de ejecución:</b> Se selecciona a un usuario determinado. La herramienta comprueba que exista algún registro correspondiente al usuario especificado.	
<b>Resultado:</b> La herramienta muestra un listado con los registros correspondientes al usuario especificado.	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla44. Pruebas de aceptación1 para la HU8Filtrar registros (elaboración propia).**

Caso de prueba de aceptación	
<b>Código:</b> HU8_p2	<b>Historia de usuario (No.8):</b> Filtrar registros.
<b>Nombre:</b> Filtrar registros por dispositivo.	
<b>Descripción:</b> Se desea probar que la herramienta filtre los registros por dispositivo.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción filtrar registros.	
<b>Entrada/ Pasos de ejecución:</b> Se selecciona a un dispositivo determinado. La herramienta comprueba que exista algún registro correspondiente al dispositivo	

## Anexos

especificado.

**Resultado:** La herramienta muestra un listado con los registros correspondientes al dispositivo especificado.

**Evaluación de la prueba:** Satisfactorio

Tabla 45. Pruebas de aceptación<sup>2</sup> para la HU8Filtrar registros (elaboración propia).

Caso de prueba de aceptación	
<b>Código:</b> HU8_p3	<b>Historia de usuario (No.8):</b> Filtrar registros.
<b>Nombre:</b> Filtrar registros por estación de trabajo.	
<b>Descripción:</b> Se desea probar que la herramienta filtre los registros por estación de trabajo.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción filtrar registros.	
<b>Entrada/ Pasos de ejecución:</b> Se selecciona a una estación de trabajo determinada. La herramienta comprueba que exista algún registro correspondiente a la estación de trabajo especificada.	
<b>Resultado:</b> La herramienta muestra un listado con los registros correspondientes a la estación de trabajo especificada.	
<b>Evaluación de la prueba:</b> Satisfactorio	

## Anexos

Tabla 46. Pruebas de aceptación<sup>3</sup> para la HU8Filtrar registros (elaboración propia).

Caso de prueba de aceptación	
<b>Código:</b> HU11_p1	<b>Historia de usuario (No.11):</b> Añadir usuario.
<b>Nombre:</b> Añadir usuario.	
<b>Descripción:</b> Se desea probar que la herramienta agregue un nuevo usuario a la base de datos.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción Añadir usuario.	
<b>Entrada/ Pasos de ejecución:</b> Se introducen los datos correspondientes.	
<b>Resultado:</b> La herramienta añade un nuevo usuario.	
<b>Evaluación de la prueba:</b> Satisfactorio	

Tabla 47. Pruebas de aceptación para la HU11Añadir usuario (elaboración propia).

### Iteración 6

Caso de prueba de aceptación	
<b>Código:</b> HU6_p1	<b>Historia de usuario (No.6):</b> modificar permisos de los dispositivo
<b>Nombre:</b> Modificar.	
<b>Descripción:</b> Se desea probar que la herramienta modifique los permisos de los dispositivos registrados en la base de datos.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción Modificar permisos.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta modifica los permisos, con los datos introducidos por el usuario.	
<b>Resultado:</b> La herramienta actualiza los permisos.	
<b>Evaluación de la prueba:</b> Satisfactorio	

Tabla 48. Pruebas de aceptación para la HU6Modificar permisos (elaboración propia).

Caso de prueba de aceptación
------------------------------



## Anexos

<b>Código:</b> HU7_p1	<b>Historia de usuario (No.7):</b> Visualizar trazas
<b>Nombre:</b> Visualizar.	
<b>Descripción:</b> Se desea probar que la herramienta permita ver los registros de las acciones realizadas por el usuario sobre el sistema de archivo de los dispositivos.	
<b>Condiciones de ejecución:</b> Se debe acceder a la opción Visualizar trazas.	
<b>Entrada/ Pasos de ejecución:</b> La herramienta busca los registros realizados hasta el momento.	
<b>Resultado:</b> La herramienta muestra un listado con los registros realizados hasta el momento.	
<b>Evaluación de la prueba:</b> Satisfactorio	

**Tabla 49. Pruebas de aceptación para la HU7 Visualizar trazas (elaboración propia).**

## Referencias

### Referencias

1. **Rello, Carem Arlen Mendoza del.** monografias.com. *monografias.com*. [En línea] [Citado el: 12 de 05 de 2014.] <http://www.monografias.com/trabajos89/crimen-y-fraude-informatico/crimen-y-fraude-informatico.shtml>.
2. **Méndez, Zoilibeth Moreno.** Control tecnológico. *Control tecnológico*. [En línea] 19 de 06 de 2011. [Citado el: 11 de 03 de 2014.] <http://controltecnologico.blogspot.com>.
3. Criptonomicón. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.iec.csic.es/criptonomicon/autenticacion/control.html>.
4. trendmicro. [En línea] [Citado el: 09 de 12 de 2013.] [http://docs.trendmicro.com/all/ent/officescan/v10.6/es-es/osce\\_10.6\\_olhsrv/OHelp/Device/dctrl.htm](http://docs.trendmicro.com/all/ent/officescan/v10.6/es-es/osce_10.6_olhsrv/OHelp/Device/dctrl.htm).
5. Definición. [En línea] [Citado el: 09 de 12 de 2013.] <http://definicion.de/usb/>.
6. Usb en castellanos. [En línea] [Citado el: 09 de 12 de 2013.] [http://www.info-ab.uclm.es/labelec/solar/elementos\\_del\\_pc/Puerto\\_usb/index.htm](http://www.info-ab.uclm.es/labelec/solar/elementos_del_pc/Puerto_usb/index.htm).
7. ZMA Productos especiales. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.zma.com.ar/2011/proyectos-especiales/descripcion-de-producto-pe-netwrix-usb-blocker-40-0.html>.
8. SPN Red de software. [En línea] [Citado el: 09 de 12 de 2013.] [http://softpicks.com.es/software/Utilitarios/Control-de-Acceso/USB-Drive-Blocker-Software\\_es-146099.htm](http://softpicks.com.es/software/Utilitarios/Control-de-Acceso/USB-Drive-Blocker-Software_es-146099.htm).
9. Devicelock. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.devicelock.com>.
10. [En línea] [Citado el: 09 de 12 de 2013.] [www.freedownloadmanager.org](http://www.freedownloadmanager.org).
11. Softonic. [En línea] [Citado el: 09 de 12 de 2013.] <http://usb-flash-block-unblock.softonic.com/imagenes-videos>.
12. kasperskyLab. [En línea] 11 de 04 de 2013. [Citado el: 10 de 06 de 2014.] <http://support.kaspersky.com/sp/9366>.
- 13.
14. EcuRed. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.ecured.cu/index.php/Geany>.
15. EcuRed. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.ecured.cu/index.php/Geany>.
16. Python. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.python.org/about/>.
17. Wikipedia. [En línea] [Citado el: 09 de 12 de 2013.] <http://es.wikipedia.org/wiki/Symfony>.

## Referencias

18. InstanCode. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.instancode.com/framework-extjs-4>.
19. **Chavéz, Amalia Isabel Rivero y Pérez, Fidel celorio.** *Implementación del proceso realizar autopsia del módulo anatomía patológica del sistema de información hospitalaria alasHID*. Sistema de gestion hospitalaria, Universidad de las Ciencias Informáticas. 2010.
20. alarcos. [En línea] [Citado el: 13 de 12 de 2013.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
21. **Uñoja, Roger Humberto.** Ingeniería del software. [En línea] 23 de 04 de 2012. [Citado el: 13 de 12 de 2013.] <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>.
22. **Loaiza, Douglas Alfredo.** Ingeniería del software: Metodología XP. [En línea] 07 de 2012. [Citado el: 13 de 12 de 2013.] <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>.
23. **Pressman, Roger S.** *Ingeniería de software, un enfoque práctico*. 2005.
24. **Letelier, Patricio y Penadés, María Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia. 2008.
25. **Wels, Donovas.** Ciclo de vida de un proyecto XP. *Ciclo de vida de un proyecto XP*. [En línea] [Citado el: 11 de 03 de 2014.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
26. Gestión de proyectos y desarrollo de software. *Gestión de proyectos y desarrollo de software*. [En línea] 01 de 2012. [Citado el: 11 de 03 de 2014.] <http://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
27. **Ovalles, Prof. María A. Perez De, Grimán, Ana y E., Luis Mendoza.** <http://prof.usb.ve>. [En línea] 04 de 2004. [Citado el: 1 de 03 de 2014.] <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>.
28. **Reynoso, Carlos y Kicillof, Nicolás.** [En línea] 03 de 2004. [Citado el: 1 de 03 de 2014.] [carlosreynoso.com.ar](http://carlosreynoso.com.ar).
29. [En línea] 3 de 10 de 2008. [Citado el: 19 de 05 de 2014.] <http://ccia.ei.uvigo.es/docencia/SCS>.
30. **Fernandez, Luis H.** Software Guisho. *Software Guisho*. [En línea] 21 de 04 de 2009. [Citado el: 24 de 03 de 2014.] <http://software.guisho.com/patrones-de-diseno>.
31. **Visconti, Marcello y Astudillo, Hernán.** [En línea] [Citado el: 24 de 03 de 2014.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
32. **Departamento de Ciencias de la Computación e I.A.** elvex. *elvex*. [En línea] [Citado el: 2014 de abril de 2014.] <http://elvex.urg.es/idbis/db/docs/intro/C%20Modelado%20de%20datos.pdf>.

## Referencias

33. Microsoft Developer Network. *Microsoft Developer Network*. [En línea] [Citado el: 27 de 03 de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
34. **Joskowicz, Ing. José.** [En línea] 10 de 02 de 2008. [Citado el: 12 de 04 de 2014.]
35. *slideshare*. [En línea] 09 de 12 de 2013. <http://www.slideshare.net/JazmineSheccid23/puerto-usb-5984149>.
36. *Portal de la Universidad de las Ciencias Informáticas*. [En línea] 09 de 12 de 2013. <http://www.uci.cu/mision>.
37. *Objetivos y misión de Centro y el Dpto. Sistema de gestion hospitalaria, Universidad de las iencias Informáticas*. La Habana : s.n., 2013.
38. **Villegas, Jaime.** *tecnoseguro*. [En línea] [Citado el: 09 de 12 de 2013.] <http://www.tecnoseguro.com/faqs/control-de-acceso/%C2%BF-que-es-un-control-de-acceso.html>.
39. Portal del CIM. *Portal del CIM*. [En línea] 2009. [Citado el: 21 de 02 de 2014.] <http://www.cim.sld.cu/>.
40. **Peláez, Juan.** *Arquitectura basada en capas. Arquitectura basa en capas*. [En línea] 26 de 05 de 2009. [Citado el: 24 de 03 de 2014.] [http://Arquitectura basada en capas. \\_ Juan Peláez.htm](http://Arquitectura basada en capas. _ Juan Peláez.htm).
41. **I.A., Departamento de Ciencias de la Computación e. elvex. elvex.** [En línea] [Citado el: 1 de 04 de 2014.] <http://elvex.ugr.es/idbis/db/docs/intro/C%20Modelado%20de%20datos.pdf>.
42. **Joskowicz, Ing. José.** [En línea] 10 de 02 de 2008. [Citado el: 12 de 04 de 2014.]
43. **Perales, Adrian.** *Gadius empire. Gadius empire*. [En línea] 02 de 2013. [Citado el: 2014 de 04 de 29.] <http://gadiempire.blogspot.com/>.
44. [En línea] [Citado el: 2014 de 04 de 29.] <http://www.python.org/about/>.
45. Wikipedia. [En línea] [Citado el: 2014 de 04 de 29.] <http://es.wikipedia.org/wiki/Symfony..>
46. InstanCode. [En línea] [Citado el: 29 de 04 de 2014.] <http://www.instancode.com/framework-extjs-4>.
47. **Douglas, Alfredo.** *Ingenieria del software: Metodologia XP*. [En línea] [Citado el: 29 de 04 de 2014.] <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>.
48. **Fernandez, Luis H.** *Software Guisho*. [En línea] 21 de 04 de 2009. [Citado el: 29 de 04 de 2014.] <http://software.guisho.com/patrones-de-diseno..>