



**Facultad 1**  
**Universidad de las Ciencias Informáticas.**

**Título:** Sistema para la administración remota de dispositivos USB.  
Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autora:** Yusmila Mora Vega

**Tutor:** Ing. Nadia Porro Lugo

**Co-Tutor:** Ing. Edilberto Blez Deroncele

Ciudad de la Habana

26 de abril de 2012



*"... Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes."*



## DATOS DE CONTACTO

**Ing. Nadia Porro Lugo:** Graduada de “Ingeniera en Ciencias Informáticas” en el 2007. Categoría docente: Instructor. Posee 1 año de experiencia en el tema y 5 años de graduada.

Email: nporro@uci.cu

**Ing. Edilberto Blez Deroncele:** Graduado de “Ingeniero en Ciencias Informáticas” en el 2011. Categoría docente. Instructor Recién Graduado en Adiestramiento. Posee 3 años de experiencia en el tema y 1 año de graduado.

Email: eblez@uci.cu



*A mis padres, por ser las personas que más amo en la vida, por ser los mejores padres del mundo, por apoyarme en todo momento y darme fuerzas para seguir adelante, por haber sacrificado tantas cosas en su vida para que este momento llegara a la mía, por inculcarme todos los valores necesarios para convertirme en una persona de bien, por su dedicación y sacrificio, por su confianza y seguridad en mí, por sus consejos que gracias a ellos he llegado a realizar uno de los anhelos más grandes de la vida y por lo cual les estaré eternamente agradecida. Los amo con todo mi corazón.*

*A mi querida hermanita Yubi, a pesar de tan solo llevarme un año de edad, te considero mi segunda madre, por esos consejos que siempre sabes darme, por tu amor y confianza, por haberme escuchado atentamente cuando las cosas no han ido bien, a ti que me has brindado tu apoyo incondicional, eres mi ejemplo a seguir, para mí significas mucho, te quiero con toda el alma.*

*A mi sobrinita, más que sobrina mi hermana más chiquita Yaneisita, por ese derroche de amor que sentimos una por la otra, por esos consejitos tan sabios, siendo aún tan pequeña, te agradezco por haber llegado a nuestras vidas y ganarte un lugar muy grande en mi corazón. Te quiero mucho.*

*A mis hermanos Joandy y Yordanys, por ser mi ejemplo a seguir, por sus consejos y confianza en mí, por enseñarme que en la vida hay que sacrificarse para lograr lo que nos proponemos, por apoyarme siempre, por todo su amor y cariño. Los quiero mucho.*

*A mi novio Pedro, por ser tan especial, por todo su apoyo incondicional, por quererme, ayudarme y comprenderme, por confiar en mí, gracias por existir. Te amo.*

*A toda mi familia de forma general, a mi cuñada Daimí, a mis sobrinos Melany y Yoandito. Los quiero.*

*A mis amigas Elizabeth, Lianna, Yanisel y Yaniris porque saben escuchar y brindar ayuda cuando es necesario. Por todo su apoyo, gracias. A mis compañeros por esa linda amistad que cultivamos, Rayko y Jose.*

*A mi tutora Nadia, por todo su apoyo. A mi co-tutor Edito por su paciencia, ayuda y los conocimientos que me transmitió.*

*A la Revolución por darme la oportunidad de formarme como ingeniera.*



*Dedico este trabajo a mis padres, por su apoyo incondicional y gracias a ellos he tratado de ser cada día una mejor persona, a mis hermanos por su confianza en mí, a mi sobrina Yaneisi, por todo su amor y a mi novio por quererme y comprenderme tanto.*

Declaro que soy el único autor de este trabajo y autorizo al centro de desarrollo CESOL de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año 2012.

---

Yusmila Mora Vega

**Firma del Autor**

---

Ing. Nadia Porro Lugo

**Firma del Tutor**

---

Ing. Edilberto Blez Deroncele

**Firma del Co-tutor**



**Resumen**

La necesidad de evitar el robo de información sensible en cualquier dispositivo de almacenamiento, ya sea en una memoria o un disco duro, como parte de la política de seguridad en los Organismos de la Administración Central del Estado (OACE), impone contar con un sistema que permita restringir el uso de ciertos dispositivos USB.

Este trabajo presenta un sistema para la administración remota de dispositivos USB, que garantiza una mayor seguridad y el manejo de los dispositivos de almacenamiento. Para el desarrollo de dicho sistema, se realiza un diseño que cumple con los parámetros establecidos por la arquitectura, definiéndose una arquitectura n capas y basada en plugin, se emplean como herramientas de modelado Visual Paradigm, como lenguaje de programación se utilizó C para el lado del cliente y C++ para el lado del servidor, se escogió el protocolo de comunicación SSH, luego de un estudio realizado se decidió escoger como Sistema Gestor de Base de Datos (SGBD) SQLite, entre otras tecnologías seleccionadas para la implementación. Además se realizó un periodo de pruebas para verificar el correcto funcionamiento de la aplicación. Se presenta como resultado un sistema listo para su instalación y funcionamiento.

**Palabras Claves:** Administración, Seguridad, Dispositivos USB.

Índice de Contenido

Introducción .....1

Capítulo 1. Fundamentación teórica .....5

    Introducción.....5

    1.1 Seguridad.....5

    1.2 Dispositivos USB.....7

        1.2.1 Dispositivos de almacenamiento.....7

    1.3 Administración de dispositivos USB.....9

    1.4 Estudio de sistemas que permiten la administración de dispositivos USB .....11

    1.5 Metodología de Desarrollo.....14

        1.5.1 Proceso Unificado de Desarrollo.....14

        1.5.2 Extreme Programing (XP) .....15

        1.5.3 OpenUP.....17

    1.6 Lenguajes de Desarrollo.....19

    1.7 Herramientas de Modelado .....20

        1.7.1 Rational Rose .....20

        1.7.2 Visual Paradigm.....21

    1.8 Entorno de Desarrollo (IDE) .....22

    1.9 Protocolos de Comunicación .....22

    1.10 Sistemas Gestores de Base de Datos (SGBD) .....23

        1.10.1 SQLite.....23

        1.10.2 MySQL.....23

        1.10.3 PostgreSQL .....24

1.10.4 Resultados de la comparación de los SGBD.....	24
Conclusiones .....	25
Capítulo 2.Características y Diseño del sistema .....	26
Introducción.....	26
2.1. Modelo de dominio .....	26
2.1.1. Descripción de las clases.....	27
2.2. Requisitos Funcionales y No funcionales .....	27
2.2.1. Funcionales .....	27
2.2.2. No Funcionales.....	28
2.3. Diagrama CU del sistema.....	29
2.3.1. Descripción de actores.....	30
2.3.2 Descripción de los casos de uso del sistema .....	30
2.3.3 Especificación de los casos de uso del sistema.....	31
2.4 Patrones de arquitectura .....	36
2.4.1. Arquitectura N-capas .....	36
2.4.2. Arquitectura basada en plugin.....	37
2.5 Arquitectura del Sistema.....	38
2.5.1 Descripción de los componentes de la arquitectura .....	38
2.6 Diagrama de paquetes .....	38
2.6.1 Descripción de los Paquetes.....	39
2.7 Diagrama de clases del diseño.....	40
2.7.1 Diagrama de clases del diseño para el Servidor .....	40
2.7.2 Diagrama de estructuras de datos para el Cliente.....	41

2.7.3 Descripción de las clases.....	42
2.7.4 Patrones de diseño. ....	44
2.8 Diseño de la BD .....	45
2.8.1 Diagrama de clases persistentes .....	45
2.8.2 Modelo Entidad- Relación .....	45
2.9 Conclusiones.....	47
Capítulo 3. Implementación y Prueba .....	48
Introducción.....	48
3.1. Modelo de implementación.....	48
3.1.1. Diagrama de componentes .....	48
3.2. Estándares de codificación.....	49
3.3 Pruebas.....	50
3.3.1 Pruebas de funcionalidades .....	50
3.4 Resultados de las pruebas de funcionalidades.....	61
3.4. Modelo despliegue .....	62
3.5 Conclusiones.....	63
Conclusiones Generales .....	64
Recomendaciones .....	65
Bibliografía Referenciada.....	66
Bibliografía Consultada .....	69

Ilustración 1 Modelo de Dominio .....	27
Ilustración 2 Diagrama CU del sistema .....	30
Ilustración 3 Arquitectura del Sistema .....	38
Ilustración 4 Diagrama de paquetes.....	39
Ilustración 5 Diagrama de clases del diseño capa Acceso a Datos.....	40
Ilustración 6 Diagrama de clases del diseño capa Servicios .....	41
Ilustración 7 Diagrama de clases del diseño capa Presentación .....	41
Ilustración 8 Diagrama de Estructura de Datos (cliente).....	42
Ilustración 9 Diagrama Entidad-Relación .....	46
Ilustración 10 Diagrama de componentes .....	49
Ilustración 11 . Diagrama de No Conformidades .....	61
Ilustración 12 Diagrama de No Conformidades por Casos de Uso.....	62
Ilustración 13.Modelo de Despliegue .....	62



Tabla 1 Descripción de actores.....	30
Tabla 2 CU Dar Acceso a Dispositivo .....	32
Tabla 3 Gestionar usuario.....	35
Tabla 4 Asignar Permiso.....	36
Tabla 5 Descripción clase ServicioConexionCliente.....	42
Tabla 6 Descripción clase ServicioPrincipal.....	43
Tabla 7 Descripción clase ServicioInterfazUsuario.....	43
Tabla 8 Descripción clase ServicioManejadorResultados .....	44
Tabla 9 Descripción clase ServicioAccesoDato .....	44
Tabla 10 Descripción clase ServicioAccesoDato .....	44
Tabla 11 Descripción clase Usuario.....	46



## Introducción

Muchas empresas no son conscientes, o escogen ignorar, la amenaza presentada por los dispositivos portátiles en sus redes empresariales, hasta que ocurre algún acontecimiento desafortunado o incluso catastrófico.

Aunque la mayoría de las empresas tienen software antivirus, firewall, seguridad de contenido de correo y web para protegerse contra amenazas externas, pocos se dan cuenta de lo sencillo que es para un empleado o para un intruso copiar enormes cantidades de información sensible en una memoria USB o en discos duros extraíbles, además del riesgo de la introducción en la red de virus, software malicioso o ilegal.

Por otro lado, la utilización de un Sistema Operativo (SO) como GNU/Linux puede implicar cierta seguridad con respecto a otros, pero no resulta infalible. Con la aparición de los mencionados dispositivos USB, es más fácil transportar información, pero estos pueden llegar a ser perjudiciales e incluso constituir una de las mayores amenazas para la información confidencial.

GNU/Linux Nova es el sistema operativo al que está migrando todo el territorio cubano, hoy en día muchas instituciones lo utilizan, como es el caso de la Universidad de las Ciencias Informáticas (UCI), y otros pertenecientes a los Organismos de la Administración Central del Estado (OACE). Esto se debe, a que además de ser un sistema hecho por cubanos y para los cubanos, adiciona a las 4 libertades del software, 4S o sea soberanía tecnológica, socio-adaptabilidad, sostenibilidad y seguridad. Sin embargo en algunas de las entidades donde se utiliza el sistema, es necesario, como parte de su política de seguridad, restringir el uso de ciertos dispositivos electrónicos.

Para estos organismos, es imprescindible, evitar el robo de información en una memoria o cualquier otro dispositivo de almacenamiento, o que se conecte una impresora que no pertenezca a la entidad e imprima documentación sensible, es por esto que se hace necesaria la administración remota de los dispositivos USB en las computadoras de dichos organismos.

Actualmente no existe una herramienta para la administración remota de dispositivos USB para el sistema operativo Linux, por lo que es necesario desarrollar una aplicación que permita la administración remota

de dispositivos USB en el sistema operativo GNU/Linux Nova que satisfaga las políticas de seguridad de OACE.

Por lo anteriormente descrito queda definido el siguiente **Problema de Investigación**: ¿Cómo administrar de forma remota los dispositivos USB?

Siendo el **Objeto de Estudio** la administración remota de dispositivos USB, y el **Campo de Acción** la administración remota de dispositivos externos de bloques que almacenan información.

El **objetivo general** de la investigación es diseñar e implementar un sistema para la administración remota de dispositivos externos de bloques que almacenan información.

Como **objetivos específicos** se desglosan los siguientes:

- Diseñar la arquitectura del sistema para administrar de forma remota los dispositivos externos de bloques que almacenan información, y que permita que la aplicación sea escalable.
- Definir el protocolo de comunicación entre el ordenador cliente y el ordenador servidor para garantizar los mecanismos de seguridad a través de la red.
- Implementar la base del sistema para la administración remota de dispositivos externos de bloques que almacenan información.
- Validación del sistema.

Planteándose la **Idea a defender**:

Con el diseño e implementación de un sistema se podrán administrar remotamente los dispositivos externos de bloques que almacenan información.

Para organizar el presente trabajo se definieron como **tareas de investigación**:

- Realización de la fundamentación teórica de la investigación.
- Estudio del estado del arte de las aplicaciones que se dedican a la administración de dispositivos USB.

- Selección de las herramientas y metodologías de desarrollo para el desarrollo del sistema.
- Levantamiento de los requisitos funcionales y no funcionales que deberá contener el sistema.
- Implementación del sistema.
- Ejecución de las pruebas para evaluar y validar la calidad del producto.

Para realizar esta investigación se utilizaron los siguientes **métodos científicos**:

### **Métodos teóricos:**

- Analítico-Sintético: Este método permite estudiar las teorías y los documentos referentes a los dispositivos USB y a la administración de los mismos, facilitando de esta forma la extracción de los elementos más importantes relacionados con el objeto de estudio por partes y en su conjunto.
- Histórico-Lógico: Este método permite estudiar lo relacionado con la administración remota de los dispositivos USB, para así obtener conocimiento histórico de su evolución, desarrollo y comportamiento a nivel internacional y nacional.

Con este trabajo se pretende alcanzar como **posible resultado**: Un sistema que permita en el sistema operativo Nova:

- Detectar dispositivos USB
- Otorgar permisos a los dispositivos USB
- Obtener información de estos dispositivos
- Desactivar dispositivos USB

La investigación será abordada en tres capítulos:

**Capítulo 1. Fundamentación teórica:** Se expone todo lo relacionado con la tecnología y la metodología que se utilizará para alcanzar el objetivo. Se analiza el estado actual referente a la administración remota de dispositivos USB y se exponen los conceptos fundamentales relacionados. Se realiza un estudio de las herramientas y la metodología a utilizar para el desarrollo del sistema.

**Capítulo 2. Características y Diseño del sistema:** En este capítulo se presentan los requerimientos del sistema a ser desarrollados, su descripción y todo el proceso de análisis y diseño de manera detallada de acuerdo a lo establecido en la metodología utilizada, además de cómo quedará desplegado el sistema.

**Capítulo 3. Implementación y prueba:** Se expone todo lo relacionado a la implementación del sistema, su estructuración en clases y componentes que garantizan la capacidad operacional del producto, con la explicación detallada de cada componente y su funcionalidad.

## Capítulo 1. Fundamentación teórica

### Introducción

En el presente capítulo se describen los fundamentos teóricos de la presente investigación: la seguridad, la administración remota y los dispositivos USB. Se presenta el estudio de varias herramientas que permiten la administración remota de dispositivos, las cuáles sirvieron de base para la determinación de los requisitos. Se determina la metodología, herramientas, lenguajes, y mecanismos de seguridad para el desarrollo del sistema.

#### 1.1 Seguridad

La seguridad de software es un tema que se trata como algo borroso y su definición se maneja con cierto grado de incertidumbre, teniendo diferentes definiciones por distintas personas. Según el diccionario de la Real Academia Española de la Lengua (RAE), el término seguridad se refiere a la certeza o garantía de que algo va a cumplirse; mecanismo que previene algún riesgo o asegura el buen funcionamiento de alguna cosa, precaviendo que falle. Si se aplica el concepto al campo de la informática se puede definir que la Seguridad Informática generalmente consiste en asegurar que los recursos tecnológicos y activos tangibles e intangibles de un sistema de información (material informático o programas) de una organización sean utilizados de la manera que se decidió. Esto se traduce en que el sistema se encuentra en un estado que nos indica que está libre de peligro, daño o riesgo (1).

Sin embargo todos coinciden en que las tendencias actuales en el mundo de la seguridad de software establecen que “El objetivo de la Seguridad Informática será mantener la Integridad, Disponibilidad, Privacidad, Control y Autenticidad de la información manejada por la computadora” (1).

Como conclusión la seguridad de un sistema de software es la combinación de su capacidad para garantizar la integridad, disponibilidad, privacidad, control y autenticidad de la información y demás activos.

Una vez definido el concepto de seguridad, se pueden establecer objetivos básicos para un software seguro (2):

**Independencia de la seguridad:** la seguridad debe construirse y utilizarse de manera independiente de la aplicación.

**Uniformidad:** la seguridad debe aplicarse de manera correcta y consistente a través de toda la aplicación

y del proceso que desarrolla la misma.

**Modularidad:** mantener la seguridad separada. Entre otras ventajas, esto nos brinda mayor flexibilidad y menor costo de mantenimiento.

**Ambiente seguro:** se debe partir de un entorno confiable. Es decir, las herramientas de desarrollo y lenguajes de programación no deben contener agujeros de seguridad.

**Seguridad desde el comienzo:** la seguridad debe ser considerada como un requerimiento desde el inicio del diseño.

**Seguridad:** Seguridad es toda aquella acción que tiende a garantizar el cumplimiento de cuatro objetivos importantes (2):

- **Confidencialidad:** Los objetos de un sistema serán accedidos únicamente por entidades autorizadas.
- **Integridad:** Los objetos sólo pueden ser modificados por entidades autorizadas y de manera controlada.
- **Disponibilidad:** Los objetos deben permanecer accesibles a entidades autorizadas.
- **Autenticación:** Verificar la identidad del emisor y del receptor

Los tres elementos que se deben proteger son el hardware (servidor, cableado, etc.), el software (sistema operativo, aplicaciones, etc.) y los datos (documentos, bases de datos, etc.). Las amenazas a las que pueden estar expuestos estos elementos son:

- **Interrupción:** Un recurso del sistema es destruido o se vuelve no disponible. También es denominado Negación de Servicio.
- **Intercepción:** Una entidad no autorizada consigue acceso a un recurso.
- **Modificación:** Una entidad no autorizada no solo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad.
- **Fabricación:** Una entidad no autorizada inserta objetos falsificados en el sistema.

## 1.2 Dispositivos USB

Los dispositivos USB tienen varios nombres, incluyendo dispositivos thumb (pulgares en inglés, ya que es común que su tamaño sea el de un pulgar). Los dispositivos USB actualmente están disponibles con USB 2.0, pero los USB 3.0 están entrando al mercado ahora (3).

Estos pequeños dispositivos de **plug and play**<sup>1</sup> son extraíbles, regrabables, y excelentes para el almacenamiento de datos personales y profesionales, ya que muchos son dispositivos de hardware encriptados para mayor seguridad. Los dispositivos USB vienen en colores, estilos y capacidades que se adaptan a todas las edades y grupos demográficos (3).

### 1.2.1 Dispositivos de almacenamiento

Dispositivo de almacenamiento es todo aparato que se utilice para grabar los datos de la computadora de forma permanente o temporal. Una unidad de disco, junto con los discos que graba, es un dispositivo de almacenamiento.

#### Medios extraíbles:

Los medios extraíbles son aquellos medios de almacenamiento diseñados para ser extraídos de la computadora sin tener que apagarla. Este término medio extraíble también hace referencia a algunos dispositivos de almacenamiento extraíbles, cuando éstos son usados para transportar o almacenar datos.

Por ejemplo:

- Memorias USB.
- Discos duros externos.

#### Dispositivos externos de bloques

Un dispositivo de bloques, en el campo de la informática, es un componente de la computadora en el que la comunicación con la unidad central de proceso los datos se transmiten en conjuntos indivisibles, o sea, aquello que no se puede dividir.

---

<sup>1</sup> **Plug-and-play** se refiere a la capacidad de un sistema informático de configurar automáticamente los dispositivos al conectarlos. Permite poder enchufar un dispositivo y utilizarlo inmediatamente, sin preocuparse de la configuración.

Un ejemplo típico de dispositivo de bloque es un disco duro, que está ordenado internamente en sectores que típicamente son de 512 bytes, y al escribir o al leer del disco es necesario transferir esa cantidad de información como conjunto.

### **Memorias USB**

Una memoria USB (de Universal Serial Bus), es un dispositivo de almacenamiento que utiliza una memoria flash para guardar información. Se lo conoce también con el nombre de unidad flash USB, lápiz de memoria, lápiz USB, minidisco duro, unidad de memoria, llave de memoria, entre otros. Los primeros modelos requerían de una batería, pero los actuales ya no. Estas memorias son resistentes a los rasguños (externos), al polvo, y algunos hasta al agua, factores que afectaban a las formas previas de almacenamiento portátil, como los disquetes, discos compactos y los DVD (4).

Estas memorias se han convertido en el sistema de almacenamiento y transporte personal de datos más utilizado, desplazando en este uso a los tradicionales disquetes y a los CD.

### **Discos duros portátiles o extraíbles**

Los discos duros portátiles son una de las líneas de productos periféricos para computadora de más rápido crecimiento. Algunas personas desean un disco duro portátil para crear espacio adicional para almacenamiento de archivos en las computadoras que carecen de almacenamiento interno extenso, especialmente para las computadoras portátiles. Otras usan un disco duro externo portátil para respaldar archivos y archivarlos para mantenerlos ultra seguros en caso de una falla de la computadora. (5)

Hay muchas razones para elegir un dispositivo portátil por encima de un dispositivo de escritorio, incluyendo:

1. Un disco duro portátil no requiere de una fuente de alimentación externa.
2. Un disco duro portátil es excelente para usarlo de manera móvil, es lo suficientemente pequeño como para llevarlo y con frecuencia está diseñado con resistencia para el transporte.
3. Los discos duros portátiles se pueden usar fácilmente en múltiples PC para compartir archivos.
4. Un disco duro portátil permite archivar datos rápido y fácilmente, además permiten programar respaldos automáticos (5).

A continuación se detallan cada una de las razones mencionadas anteriormente:

1. Sin fuente de alimentación externa

A los discos duros externos portátiles, en términos de la industria, se les llama “plug-and-play” (conectar y usar). Los usuarios simplemente conectan el dispositivo al puerto USB y están listos para empezar a arrastrar y soltar los archivos. No necesita de una fuente de alimentación adicional, a diferencia de una unidad de disco duro de escritorio.

2. Excelente para usarlo móvil

De manera ideal, los discos duros portátiles pueden viajar con las personas. Muchos vienen en elegantes diseños delgados y con frecuencia tienen un revestimiento resistente capaz de soportar caídas accidentales.

3. Compartir archivos

Si se trabaja con archivos grandes, un disco duro portátil es un lugar excelente para guardarlos.

4. Respaldos automáticos:

Los discos duros portátiles guardan grandes cantidades de información y le permiten programar respaldos periódicos automáticos. A menudo, los discos duros portátiles USB cuentan con un software para respaldo que se configura una vez.

### 1.3 Administración de dispositivos USB

**Administración** como concepto viene del latín ad (hacia, dirección, tendencia) y minister (subordinación u obediencia), y significa aquel que realiza una función bajo el mando de otro, es decir, aquel que presta un servicio a otro.

Según Idalberto Chiavenato, la **administración** es "el proceso de planear, organizar, dirigir y controlar el uso de los recursos para lograr los objetivos organizacionales"(6).

Reinaldo O. Da Silva, define la **administración** como "un conjunto de actividades dirigido a aprovechar los recursos de manera eficiente y eficaz con el propósito de alcanzar uno o varios objetivos o metas de la organización"(6).

En el área de la informática el concepto es ampliamente utilizado también, refiriéndose al manejo y control eficiente de los recursos tecnológicos para el logro de un buen desempeño de la tecnología.

Apareció también el concepto de la administración remota la cual surge ante la necesidad de tener los servidores en lugares físicamente seguros, aislados y protegidos, y de que un mismo administrador pueda gestionar distintos servidores ubicados en lugares distantes (7).

El objeto de estudio de este trabajo se refiere a la Administración Remota de los dispositivos USB, la cual haciendo una extrapolación y análisis de los conceptos anteriormente planteados podemos definirla como el manejo y control eficiente de los dispositivos USB para lograr, en el caso particular de la investigación, que se cumplan las políticas de seguridad de los organismos donde será desplegado el sistema.

Uno de los mecanismos para garantizar seguridad en las instituciones es mediante la administración de dispositivos USB, como es el caso de la herramienta que se pretende desarrollar.

La clave para administrar los dispositivos portátiles en el entorno empresarial es dar a los administradores control directo sobre qué dispositivos están en uso en su red. Por lo que existen varias aplicaciones que permiten la administración de dispositivos USB, y admiten conocer varias informaciones de cuando se conectan dispositivos USB de almacenamiento en la computadora.

El problema de los dispositivos de almacenamiento extraíble es que es difícil controlar qué información puede querer extraer de la empresa un empleado y con qué fines. Además, desde estos dispositivos pueden ejecutarse aplicaciones malware<sup>2</sup> de todo tipo que pueden poner en peligro la seguridad, en este caso el escenario es particularmente arriesgado.

Para darle cumplimiento al objetivo de la investigación fue necesario estudiar algunas herramientas que se dedican a la administración de dispositivos USB. A continuación se muestra el estudio y los resultados que se obtuvieron.

---

<sup>2</sup>**Malware:** también llamado **badware**, **código maligno**, **software malicioso** o **software malintencionado**, es un tipo de software que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento de su propietario.

## 1.4 Estudio de sistemas que permiten la administración de dispositivos USB

Para la investigación se consultaron un total de 5 aplicaciones de administración remota de dispositivos USB, las cuales se abordan a continuación.

### GFI Endpoint Security

Esta aplicación proporciona soluciones para la seguridad de redes, de contenido y mensajería, con un fuerte enfoque para pequeñas y medianas empresas. Sólo funciona en el Sistema Operativo Windows.

Es la solución de seguridad que ayuda a mantener la integridad de los datos evitando el acceso no autorizado y la transferencia de contenidos desde y hacia los siguientes dispositivos de medios portátiles:

- Puertos USB (por ejemplo, flash y lectores de tarjetas de memoria, Pen Drives) (8).
- Puertos FireWire (por ejemplo, cámaras digitales, lectores de tarjetas FireWire)
- Conexiones de datos inalámbricas (por ejemplo infrarrojos)
- Las unidades de disco (interna y externa)
- Las unidades ópticas, tales como CD, DVD y MO (magneto óptico) unidades de disco (interna y externa)- extraíble USB unidades de disco duro.
- Otras unidades, como unidades Zip y unidades de cinta (interna y externa).

A través de su tecnología, GFI EndPointSecurity puede permitir o denegar el acceso a un dispositivo, así como asignar (si procede) los privilegios de “completo o sólo lectura”, en cada dispositivo compatible (por ejemplo, unidades CD / DVD, PDA) y en cualquier directorio local o de usuario de Active / grupo de usuarios.

También se puede grabar la actividad de todos los dispositivos portátiles que se utilizan en los equipos de destino (incluyendo la fecha / hora de uso por el cual los dispositivos se utilizaron).

Entre sus principales características podemos encontrar que evita la fuga y el robo de información mediante el control integral del acceso a dispositivos portátiles de almacenamiento con mínimo esfuerzo administrativo, impide la introducción de software malicioso o no autorizado en la red, proporciona a los administradores mayor control y puede bloquear dispositivos por clase, extensiones de archivo, puerto físico o identificador de dispositivo, desde un único lugar, además permite a los administradores conceder acceso temporal al dispositivo o puerto durante un período de tiempo estipulado, monitoriza centralizadamente la red, detecta dispositivos conectados y realiza varias tareas (8).

### **MyUSBOnly**

Ayuda a proteger los puntos finales de la computadora, por lo que se puede restringir el uso de los puertos USB y otros puntos en la computadora de escritorio o portátil y prevenir eficazmente la introducción de datos no deseados o la fuga de datos desde la computadora (8).

Esta aplicación permite prevenir el robo de datos mediante el bloqueo de todos ellos, además se comprueba una lista blanca de dispositivos permitidos según su marca comercial (si se desea permitir que un determinado USB de determinado marca se utilice) o por número de serie (para determinados dispositivos USB que se desee permitir). Secretamente registra toda la conexión USB, desconecta la actividad y recibe un mensaje de correo electrónico de notificación cuando un dispositivo de almacenamiento USB no autorizado está conectado a la computadora (8).

Esta aplicación se usa solamente en el sistema operativo Windows.

### **USBDeview**

Es una pequeña aplicación que muestra todos los dispositivos USB que están conectados a un ordenador. Para cada dispositivo USB, la información se muestra extendida: Nombre del dispositivo y su descripción, tipo de dispositivo, número de serie (para dispositivos de almacenamiento masivo), la fecha y hora en que el dispositivo fue añadido y mucho más (8).

También permite desinstalar los dispositivos USB que previamente ha usado, y desconectar los dispositivos USB que están conectados al ordenador.

También se puede utilizar USBDeview en un equipo remoto, siempre y cuando se conecte a la computadora con el usuario administrador. Esta utilidad funciona en Windows 2000/XP/2003/Vista. Windows 98 (8).

### **USB over Network**

El programa es absolutamente fácil de instalar y utilizar. Se compone de dos partes: cliente y servidor. Cuando se instale USB over Network a través del servidor de red se pueden compartir dispositivos USB locales con otros ordenadores en una red. Los ordenadores con USB over Network a través de la red del cliente tendrán acceso a estos dispositivos compartidos (9).

Esta aplicación se usa solamente en el sistema operativo Windows.

## Truecrypt

TrueCrypt es una aplicación gratuita, cuyo código fuente está disponible, permite crear volúmenes cifrados (ya se trate de un disco virtual contenido en un fichero, una partición de disco o una unidad de almacenamiento USB, por ejemplo), de tal forma que todo lo que contengan pueda ser accedido sólo si se conoce la contraseña y/o fichero clave que se utilizó para su creación. Además el cifrado y el descifrado ocurren en la memoria RAM, de forma transparente al usuario, antes de ser escritos al disco (8).

Los algoritmos de cifrado que utiliza son AES, Serpent y Twofish o una combinación de los mismos.

Existen versiones para sistemas operativos Windows XP/2000/2003/Vista/7, Mac OS X, Linux y MorphOS (en este último bajo el nombre Kryptos). TrueCrypt se distribuye gratuitamente y su código fuente está disponible, aunque bajo una licencia restrictiva (8).

## Resultados obtenidos del análisis de las herramientas que permiten la administración remota de los dispositivos USB

El estudio realizado de las aplicaciones que permiten la administración de dispositivos USB valió para detectar las funcionalidades de mayor aparición en las aplicaciones analizadas. A continuación se mencionan estas funcionalidades:

- Permitir o denegar el acceso a un dispositivo.
- Asignar los privilegios de “completo o sólo lectura”.
- Grabar la actividad de todos los dispositivos portátiles que se utilizan en los equipos de destino (incluyendo la fecha y hora de uso por el cual los dispositivos se utilizaron).
- Restringir el uso de los puertos USB
- Mostrar la información extendida con todos los dispositivos USB que están conectados a la PC. Por ejemplo: Nombre del dispositivo y su descripción, tipo de dispositivo, número de serie, la fecha, hora que el dispositivo fue añadido, entre otros.

Las 5 herramientas tratadas sirvieron para guiar y dar una vista de cómo desarrollar la aplicación, ya que no se puede hacer uso de las mismas porque solo funcionan en el Sistema Operativo Windows, y el sistema requiere que sea para Linux, específicamente para el sistema operativo Nova, por la migración que se está llevando a cabo en nuestro país a software libre.

Teniendo en cuenta el estudio realizado, a continuación se definen, tomando como base las funcionalidades nombradas, las que tendrá el sistema a desarrollar:

- Gestionar usuarios
- Gestionar PC.
- Gestionar Rol.
- Asignar permisos en dependencia del rol que ocupe el usuario, o sea los privilegios de “completo o sólo lectura”.
- Permitir o denegar el acceso a un dispositivo.

## 1.5 Metodología de Desarrollo

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo, pero elegir la mejor para un equipo en un determinado proyecto podría marcar la diferencia a la hora de obtener un software eficiente y con la calidad requerida.

A continuación se describen metodologías de desarrollo usadas ampliamente en la universidad y en el mundo para definir cuál de ellas deberá utilizarse en el desarrollo del sistema propuesto.

### 1.5.1 Proceso Unificado de Desarrollo.

RUP por sus siglas en inglés Rational Unified Process, traducido al español Proceso Unificado de Desarrollo, es una metodología robusta que se distingue por ser: dirigida por casos de uso, centrada en la arquitectura, iterativo e incremental, esto es lo que la hace un proceso unificado. RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes (10).

Como RUP es un proceso, en su modelación define como sus principales elementos:

- Trabajadores (“Quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (“Qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

- Flujo de actividades (“Cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable (10).

**Principales características:**

- Proceso dirigido por casos de uso
- Proceso centrado en la arquitectura
- Proceso iterativo e incremental

**Fases de RUP:**

**Inicio:** tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico, obteniéndose como uno de los principales resultados una lista de los casos de uso y una lista de los factores de riesgo del proyecto(10).

**Elaboración:** tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema, además se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido(10).

**Construcción:** está compuesta por un ciclo de varias iteraciones, en las cuales se van incorporando sucesivamente los casos de uso, de acuerdo a los factores de riesgo del proyecto. Este enfoque permite por ejemplo contar en forma temprana con versiones del sistema que satisfacen los principales casos de uso. Los cambios en los requerimientos no se incorporan hasta el inicio de la próxima iteración (10).

**Transición:** se inicia con una versión “beta” del sistema y culmina con el sistema en fase de producción (10).

**1.5.2 Extreme Programming (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad. Esta metodología ágil es utilizada para proyectos de corto plazo con equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (11).

**Principales características:**

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que si se adelanta en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (12).

**Fases de XP:**

**Exploración:** En esta fase, 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 (11).

**Planificación de la Entrega:** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días (11).

**Iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio) (11).

**Producción:** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase(11).

**Mantenimiento:** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura (11).

**Muerte del Proyecto:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo (11).

### 1.5.3 OpenUP

OpenUp está pensada para proyectos pequeños. Es un proceso de desarrollo iterativo del software que es mínimo, completo, y extensible:

- Mínimo: Sólo incluye el contenido del proceso fundamental
- Completo: Puede ser manifestado como proceso entero para construir un sistema.
- Extensible: Puede ser utilizado como base para agregar o para adaptar más procesos.

La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances (12).

OpenUP presenta varios beneficios ya que es apropiado para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito. Permite detectar errores tempranos a través de un ciclo iterativo. Evita la

elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP (12).

Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas. Permite, además, equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto.

#### **Principales Características:**

- Este principio promueve prácticas que permiten a los participantes de los proyectos desarrollar una solución que maximice los beneficios obtenidos y que cumpla con los requisitos y restricciones del proyecto.
- Se centra en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo, este último es evolutivo para obtener retroalimentación y mejoramiento continuo.
- Este principio promueve prácticas que permiten a los equipos de desarrollo obtener retroalimentación temprana y continua de los clientes del proyecto, permitiendo demostrarles incrementos progresivos en la funcionalidad (13).

#### **Fases del OpenUP:**

**Concepción:** Primera de las 4 fases en el proyecto del ciclo de vida, acerca del entendimiento del propósito y objetivos y obteniendo suficiente información para confirmar que lo que el proyecto debe lograr. El objetivo de ésta fase es capturar las necesidades de los **stakeholder**<sup>3</sup> en los objetivos del ciclo de vida para el proyecto.

**Elaboración:** Es el segundo de las 4 fases del ciclo de vida del OpenUP donde se trata los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base de la elaboración de la arquitectura del sistema.

**Construcción:** Esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.

---

<sup>3</sup> **Stakeholder** : las partes interesadas

**Transición:** Es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad y performance del último entregable de la fase de construcción (13).

Una vez realizado el estudio de las diferentes metodologías de desarrollo, se decide utilizar la metodología OpenUp dada las facilidades que brinda para el desarrollo de proyectos pequeños, además esta es la metodología usada en el proyecto Nova.

## 1.6 Lenguajes de Desarrollo

Para el desarrollo del sistema se escogieron los lenguajes de desarrollo que se utilizarían, quedando definido como lenguaje de modelado se usa UML y como lenguajes de programación C para el lado del cliente y C++ para el lado del servidor.

### Lenguaje Unificado de Modelado (UML)

Para crear la documentación se utiliza **UML** como lenguaje de modelado, por las potencialidades descriptivas que posee. Es un estándar de propósito general orientado a objetos y permite no solo modelar y obtener modelos de arquitectura de la aplicación, de comportamiento y de estructura, sino que también posibilita generar esquemas de negocio y de datos (14).

Se escogió C para el lado del cliente porque además de ser una restricción del diseño, es el lenguaje en el que se desarrolla el sistema operativo GNU/Linux Nova y es el que interactúa directamente con el sistema.

El lenguaje C fue inventado e implementado por primera vez por Dennis Ritchie en un DEC PDP-11 en Bell Laboratorios. Se denomina como un lenguaje de nivel medio, puesto que combina elementos de lenguajes de alto nivel (Fortran, Pascal, Basic...) con el funcionalismo del lenguaje ensamblador (16).

Permite la manipulación de bits, bytes y direcciones (los elementos básicos con que funciona la computadora).

Otras características del C es que posee muy pocas palabras clave (32, donde 27 fueron definidas en la versión original y cinco añadidas por el comité del ANSI, (enum, const, signed, void y volatile). Todas las palabras clave de C están en minúsculas (C distingue entre las mayúsculas y minúsculas) (16).

Principales características:

- Es un lenguaje de programación de propósitos generales.

- Permite la programación estructurada, la abundancia de operadores y tipos de datos.
- No está asociado a ningún sistema operativo ni a ninguna máquina.
- Es popular y eficaz
- Permite el desarrollo de Sistemas Operativos y programas de aplicación, tiene sólo 32 palabras reservadas, entre otras (16).

En el caso del lenguaje C++ se escogió para el lado del servidor por la cantidad de abstracción y la cantidad de clases que posee y este no tiene interacción con el sistema operativo, sino con la Base de datos, además por el soporte que brinda el mismo. A continuación se describen las características de este lenguaje.

El C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él, la intención de su creación fue extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido (15).

En la actualidad, el C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original (15).

## 1.7 Herramientas de Modelado

Dentro de las principales aplicaciones CASE existentes para el modelado visual se encuentran el Rational Rose y Visual Paradigm.

### 1.7.1 Rational Rose

Es una herramienta para el modelado visual mediante UML de sistemas de software. Rational Rose tiene una interfaz muy amigable conformada principalmente por un navegador que le permite al usuario navegar rápidamente por las distintas vistas del modelo. Esta herramienta con plataforma independiente ayuda a mejorar la comunicación en el trabajo en equipos y permite visualizar el sistema completo utilizando un lenguaje común UML. Los sistemas operativos que los soportan son Microsoft Windows y permite la integración con Borland JBuilder versiones 7.0 a 10.0, Microsoft Visual Studio, Enterprise Editions 3.0 entre otros (18).

### ¿Qué ofrece?

- Crear y visualizar diagramas UML.
- Especificar, analizar, y diseñar el sistema antes de codificarlo.
- Mantiene la consistencia de los modelos del sistema software.
- Chequeo de la sintaxis UML.
- Generar documentación automáticamente.
- Generar código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic a partir de los Modelos.
- Ingeniería inversa (crear modelo a partir de código).

### 1.7.2 Visual Paradigm

Constituye una herramienta profesional para UML. Permite realizar modelos de todo el ciclo de vida del desarrollo del software desde el modelamiento del negocio hasta el despliegue del producto. Con este software de modelado se logra una mayor rapidez en el desarrollo de aplicaciones que pueden ser de mayor calidad y menos costosas. Además de permitir la creación de diagramas que indican la manera en que se debe construir un sistema, brinda también la posibilidad de generar el código de dichas aplicaciones a partir de los diagramas y además permite generar la documentación (17).

### ¿Qué ofrece?

- Crear y visualizar diagramas UML.
- Generar código C++, CORBA, Java, esquemas XML a partir de los diagramas.
- Generar documentación.
- Realizar ingeniería inversa - código a modelo, código a diagrama.
- Generar bases de datos.

Las dos herramientas estudiadas anteriormente soportan el ciclo completo del desarrollo de software y cuentan con una interfaz amigable para los usuarios, así como abundante documentación, pero se decide escoger para la solución como herramienta de modelado, Visual Paradigm, ya que es una herramienta muy potente que soporta todos los diagramas UML y además es multiplataforma con soporte para el sistema operativo GNU/Linux.

## 1.8 Entorno de Desarrollo (IDE)

Como entorno de desarrollo para la realización del sistema se decide utilizar IDE Netbeans. Los entornos de desarrollo conocidos como IDEs son aplicaciones que se utilizan para facilitar la programación en diferentes lenguajes. Muchas veces se especializan en un lenguaje específico pero en otros casos ofrecen un amplio abanico de los mismos. Estas herramientas pueden ser multiplataforma o funcionar en un solo Sistema Operativo (19).

### IDE Netbeans

El IDE NetBeans es un entorno de desarrollo integrado para los desarrolladores de software. Con su uso se obtienen todas las herramientas necesarias para crear aplicaciones de escritorio, de empresa, web y aplicaciones móviles con el lenguaje Java, C / C + +, e incluso lenguajes dinámicos como PHP, Java Script, Groovy, y Ruby. Es fácil de instalar y utilizar, además, se ejecuta en muchas plataformas, incluyendo Windows, Linux, Mac OS X y Solaris (17).

## 1.9 Protocolos de Comunicación

Luego de un estudio realizado de protocolos de comunicación remota se determinó que escoger el protocolo Security SSH (Secure Shell) ya que permite la comunicación segura.

Detallando este protocolo facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor, como es el caso del sistema a desarrollar ya que permite a los usuarios conectarse a un host remotamente.

A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas (20).

El protocolo SSH encripta todo lo que envía y recibe, por lo que se puede usar para asegurar protocolos inseguros.

Si se utiliza SSH para inicios de sesión de Shell (capa) remota y para copiar archivos, se pueden disminuir las amenazas a la seguridad, ya que el cliente SSH y el servidor usan firmas digitales para verificar su identidad. Adicionalmente, toda la comunicación entre los sistemas cliente y servidor es encriptada, por lo que no servirán de nada los intentos de falsificar la identidad de cualquiera de los dos lados de la comunicación ya que cada paquete está cifrado por medio de una llave conocida sólo por el sistema local y el remoto (20).

## 1.10 Sistemas Gestores de Base de Datos (SGBD)

Una base de datos es una colección de información organizada por filas, campos y columnas, “es un conjunto de datos relacionados entre sí, entendiéndose por dato los hechos que pueden registrarse y que tienen un significado implícito”. Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten a los usuarios crear, mantener, construir y manipular base de datos para diversas aplicaciones (21).

A continuación se analizan y se describen tres Sistema Gestor de Base de Datos (SGBD) ampliamente utilizados, con sus principales características, para finalmente escoger el que más se adapte a las necesidades del sistema a desarrollar.

### 1.10.1 SQLite

SQLite es un sistema gestor de base de datos relacional, que a diferencia de los sistemas de gestión de bases de datos cliente-servidor, no establece un proceso independiente con el que el programa principal se comunica si no que su biblioteca se enlaza con el programa pasando a ser parte integral del mismo.

Entre sus principales características se encuentran:

- Varios accesos de lectura pueden ser servidos en paralelo.
- No posee configuración, esto significa que de la forma en que fue creado y diseñado SQLite no necesita ser instalado.
- Varios procesos o hilos pueden acceder a la misma base de datos sin problemas.
- Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente (22).

### 1.10.2 MySQL

Sistema Gestor de Base de Datos relacional, multihilo y multiusuario, que funciona en diferentes plataformas, sencillo de usar e increíblemente rápido. Es gratis para aplicaciones no comerciales. Posee un sistema de privilegios y contraseñas que son muy flexibles y seguros, las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. También soporta a grandes bases de datos.

Principales características que presenta:

- Aprovecha la potencia de sistemas multiprocesadores consumiendo muy pocos recursos y memoria.
- Gracias a su implementación multihilo, soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas, soporta hasta 32 índices por tabla, gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos y otras (23).

### 1.10.3 PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre, que está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee una serie de características positivas respecto a otros, por ejemplo, a diferencia de MySQL, tiene gran escalabilidad haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

Principales características:

- Es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima.
- Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, aunque no se considera un sistema de gestión de bases de datos puramente orientado a objetos.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas aunque no entre objetos, ya que no existen y soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos (24).

### 1.10.4 Resultados de la comparación de los SGBD

Se escogió como SGBD SQLite, a continuación se muestran algunas razones.

- Tamaño: SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- Rendimiento de base de datos: SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.

- Portabilidad: se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- Estabilidad: SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- SQLite: implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.
- Interfaces: cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, groovy, etc.
- Costo: SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.
- Es portable, lo que significa que puede ser ejecutado en diferentes sistemas operativos, como Windows, Linux, BSD, Mac OS X, Solaris y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración (25).

## Conclusiones

Este capítulo los conceptos fundamentales de la investigación, los cuales son seguridad, administración remota, y dispositivos USB. Se hizo un estudio de las herramientas de administración remota de dispositivos USB que sirvió de base a la determinación de las funcionalidades del sistema a desarrollar. Para el desarrollo del sistema del sistema se seleccionó como metodología de desarrollo OpenUP, como sistema gestor de base de datos SQLite, como lenguaje de modelado UML y como herramienta de herramienta de modelado Visual Paradigm, como lenguaje de programación C para el lado del cliente y C++ para el lado del servidor, todo con el fin de relacionar conceptos y lograr un buen entendimiento de la posterior solución.

## Capítulo 2. Características y Diseño del sistema

### Introducción

En este capítulo se detallan las características del sistema a desarrollar, definiendo el modelo de dominio según el objeto de estudio. Se mencionan y se desarrollan los requisitos funcionales y no funcionales que debe cumplir la aplicación. Además, se explican los patrones de arquitectura utilizados y el diagrama de clases del diseño realizado para la aplicación.

### 2.1. Modelo de dominio

Un modelo conceptual o de dominio, brinda los conceptos significativos para el dominio del problema. Este modelo representa objetos o componentes del mundo real, no específicamente del software. Además de conceptos, pueden mostrar los atributos de los conceptos y las asociaciones entre los mismos (26).

En el caso del sistema para la administración remota de dispositivos USB, se decidió representar este modelo, ya que dicho sistema no consta de un negocio real y no existe ningún proceso organizado, manual, ni automatizado que tributen a los objetivos de este trabajo de diploma.

A continuación se representa el modelo del dominio, el cual recoge y describe los conceptos más importantes dentro del contexto del sistema así como las relaciones entre ellos.

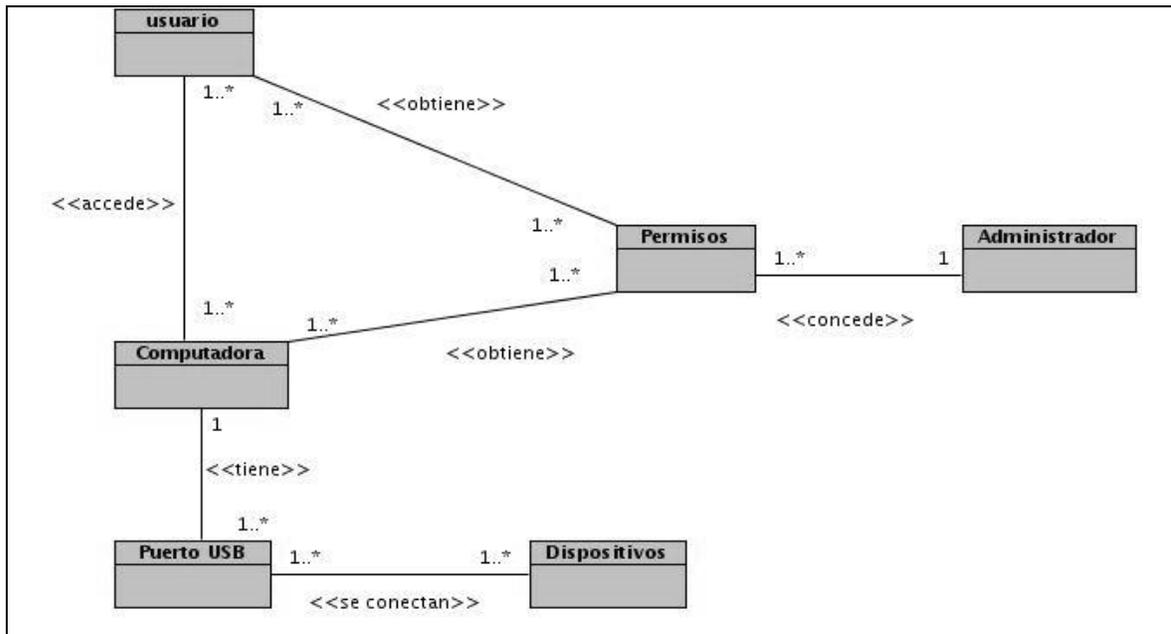


Ilustración 1 Modelo de Dominio

### 2.1.1. Descripción de las clases

**Usuario:** accede a una computadora para manipular información a través de dispositivos USB.

**Computadora:** estación de trabajo que permite la gestión de información.

**Permisos:** permisos que le son otorgados a las personas y a las computadoras con el fin, en el caso de las personas, que puedan acceder según los distintos permisos establecidos de lectura, escritura, entre otros a los dispositivos conectados en la computadora.

**Puerto USB:** permite conectar a la computadora diversos dispositivos USB.

**Dispositivos:** permite almacenar información.

**Administrador:** es el encargado de establecer el control de los dispositivos que se conecten a la computadora y de asignar los permisos a los usuarios, a los roles y a las computadoras, para el manejo de los dispositivos USB.

## 2.2. Requisitos Funcionales y No funcionales

### 2.2.1. Funcionales

#### \*Gestionar Usuario:

RF1. Insertar usuario: Permite insertar un usuario a la base de datos.

RF2. Eliminar usuario: Permite eliminar un usuario de la base de datos.

RF3. Modificar usuario: Permite modificar los datos de un usuario que se encuentra en la base de datos.

RF4. Mostrar usuario: Permite mostrar los datos de un usuario.

#### **\*Gestionar Rol**

RF5. Insertar rol: Permite insertar un rol a la base de datos.

RF6. Eliminar rol: Permite eliminar un rol de la base de datos.

RF7. Modificar rol: Permite modificar los datos de un rol que se encuentra en la base de datos.

RF8. Mostrar rol: Permite mostrar los datos de un rol.

#### **\*Gestionar Computadoras**

RF9. Insertar Computadoras: Permite insertar una pc a la base de datos.

RF10. Eliminar Computadoras: Permite eliminar una pc de la base de datos.

RF11. Modificar Computadoras: Permite modificar los datos de una pc.

RF12. Mostrar Computadoras: Permite mostrar una pc.

RF13. **Asignar Permiso:** Permite asignar diferentes permisos a los roles.

RF 14. **Dar Acceso a Dispositivo**(o sea permitir o denegar el acceso a un dispositivo).

### **2.2.2. No Funcionales**

Los requerimientos no funcionales son fundamentales en el éxito del producto y están vinculados a requerimientos funcionales. A continuación se describen los requisitos no funcionales del sistema.

#### **Requerimientos de diseño:**

- Los lenguajes de programación que se usarán serán el C++ para el lado del servidor y C para el lado del cliente.

#### **Usabilidad:**

- La Herramienta debe funcionar correctamente para cualquier usuario que desee conectar un dispositivo en la computadora.

Ambiente:

- Sistema Gestor de Bases de Datos SQLite.

**Apariencia o interfaz externa:**

- El sistema no contará con una interfaz visual (de usuario), sino que ejecutará las funcionalidades en modo consola, de esta manera se garantiza un menor consumo de recursos.

**Soporte:**

- El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.
- Netbeans 6.9.
- Visual Paradigm 6.4.

**2.3. Diagrama CU del sistema**

**Caso de uso del sistema:** Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores (27).

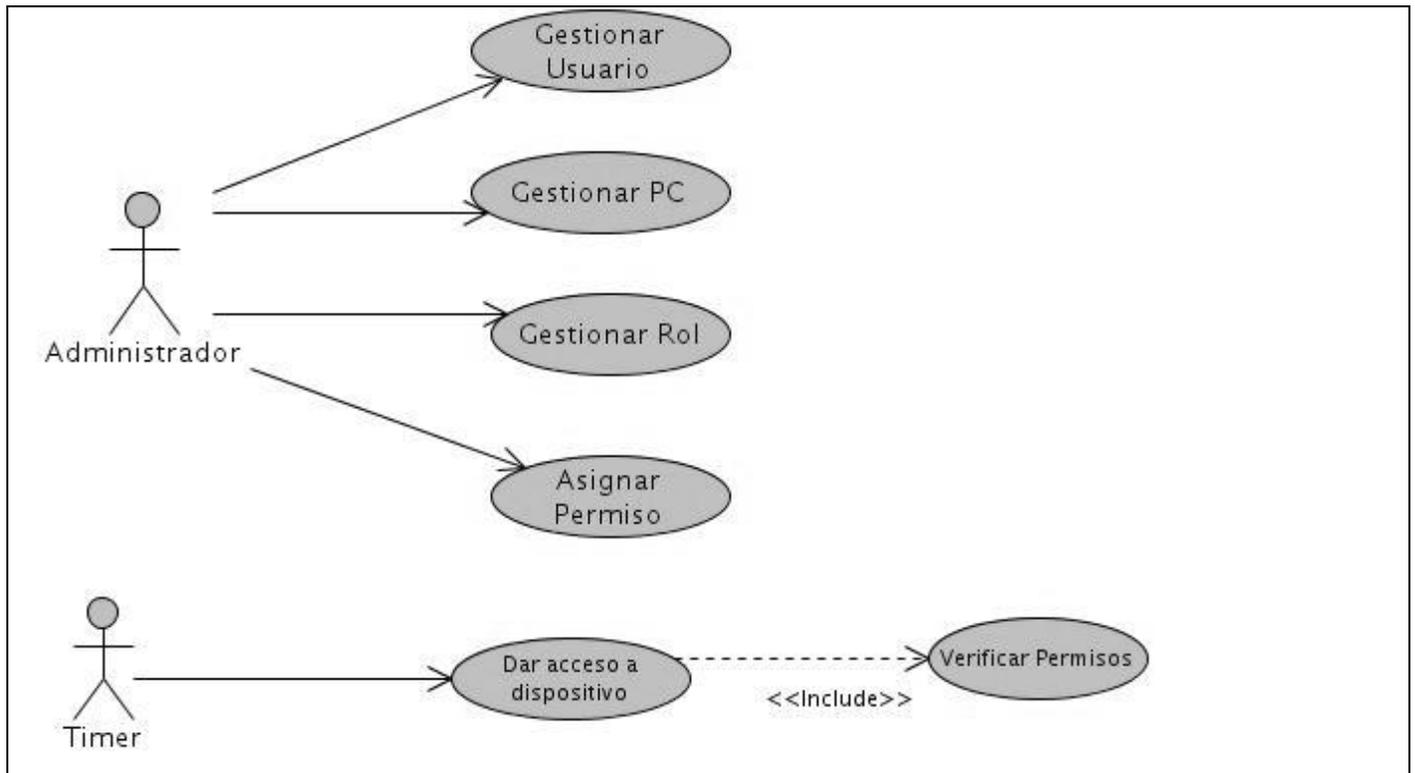


Ilustración 2 Diagrama CU del sistema

### 2.3.1. Descripción de actores

Un actor es un agente externo que interactúa con el sistema en pos de obtener un resultado esperado.

El sistema cuenta con los actores que se especifican a continuación:

Actor	Descripción
<b>Administrador</b>	Persona encargada de la gestión de roles, usuarios y pc. Además se encarga de la asignación de permisos a los diferentes roles.
<b>Timer</b>	Es el sistema, encargado luego de verificar los permisos, de dar acceso o no al dispositivo que se ha conectado a la computadora.

Tabla 1 Descripción de actores.

### 2.3.2 Descripción de los casos de uso del sistema

**CU1. Gestionar usuario:** Este caso de uso tiene como objetivo que el administrador pueda insertar, eliminar, modificar y mostrar un usuario.

**CU2. Gestionar Rol:** Este caso de uso tiene como objetivo que el administrador pueda insertar, eliminar, modificar y mostrar un rol.

**CU3. Gestionar PC:** Este caso de uso tiene como objetivo que el administrador pueda insertar, eliminar, modificar y mostrar una PC.

**CU4. Asignar Permiso:** Este caso de uso tiene como objetivo que el administrador pueda asignar permisos a los usuarios.

**CU5. Dar acceso a Dispositivo:** Este caso de uso tiene como objetivo que el sistema en dependencia de los permisos, pueda dar o no acceso a un dispositivo.

### 2.3.3 Especificación de los casos de uso del sistema

A continuación se reconocen y se detallan los casos de uso a partir de los requisitos funcionales anteriormente especificados.

Se especifican los casos de usos Dar acceso a dispositivo, Gestionar usuario y Asignar Permiso, el resto de los requisitos se detallan en la sección **Anexos**.

#### CU5. Dar Acceso a Dispositivo

<b>Caso de Uso:</b>	Dar acceso a Dispositivo	
<b>Actores:</b>	Timer	
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de que el sistema pueda darle acceso a un dispositivo.	
<b>Resumen:</b>	El caso de uso se inicia cuando el sistema le da acceso a un dispositivo y termina cuando el sistema realiza la acción.	
<b>Precondiciones:</b>	El usuario debe conectar el dispositivo en la computadora.	
<b>Referencias</b>	RF14	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	

1. El caso de uso se inicia cuando el usuario conecta un dispositivo en la PC.	2. Ir a sección Verificar. 4. El sistema le da acceso al dispositivo.
<b>Sección “Verificar Permisos”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. El sistema verifica si el usuario está autenticado en el sistema y pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo. En otro caso verifica si la pc pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2. Si el usuario no está autenticado, o si el usuario o la pc no pertenecen al dominio no se puede montar el dispositivo.

Tabla 2 CU Dar Acceso a Dispositivo

**CU1. Gestionar usuario**

<b>Caso de Uso:</b>	Gestionar usuario
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de que el administrador pueda insertar, eliminar, modificar y mostrar un usuario.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador desea insertar, eliminar, mostrar o modificar los datos de un usuario y termina cuando el sistema realiza una de estas acciones.

<b>Precondiciones:</b>	El administrador tiene que estar autenticado en el sistema.	
<b>Referencias</b>	RF1, RF2, RF3, RF4	
<b>Prioridad</b>	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso se inicia cuando el administrador selecciona una de las opciones siguientes:</p> <ul style="list-style-type: none"> <li>- Insertar usuario.</li> <li>- Eliminar usuario.</li> <li>- Modificar usuario.</li> <li>- Mostrar usuario.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el administrador.</p> <ul style="list-style-type: none"> <li>- Si selecciona "Insertar usuario", ver sección "Insertar usuario".</li> <li>- Si selecciona "Eliminar usuario", ver sección "Eliminar usuario".</li> <li>- Si selecciona "Modificar usuario", ver sección "Modificar usuario".</li> <li>- Si selecciona "Mostrar usuario", ver sección "Mostrar usuario".</li> </ul>	
	<p>3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el administrador.</p>	
Sección "Insertar usuario"		
Acción del Actor	Respuesta del Sistema	
<p>1. El administrador introduce los datos del usuario:</p> <ul style="list-style-type: none"> <li>-nombre</li> <li>-dominio</li> </ul>	<p>2. El sistema verifica que los datos estén correctos y que no exista el usuario en la BD.</p>	

	3. El sistema procesa la información a partir de la acción realizada por el administrador y almacena los datos del usuario.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2. Si los datos del usuario están incorrectos o el usuario ya existe en la BD el sistema muestra un mensaje "Datos incorrectos" o "El usuario ya existe en la BD". Volver a la sección 1.
<b>Sección "Eliminar usuario"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador entra el nombre del usuario que desea eliminar.	2. El sistema verifica que el usuario exista en la BD. 3. El sistema muestra un mensaje de confirmación de eliminar al usuario.
4. El Administrador selecciona la opción Sí.	5. El sistema elimina el usuario.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2. Si el usuario no existe en la BD, el sistema muestra un mensaje de error. Volver a la sección 1.
4. El Administrador selecciona la opción No.	5. El sistema cancela la acción de eliminar usuario.
<b>Sección "Modificar usuario"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

1. El administrador introduce el nombre del usuario que desea modificar.	2. El sistema verifica que exista en la BD. 3. El sistema permite modificar los datos del usuario.
4. El administrador modifica los datos del usuario.	5. El sistema procesa la información a partir de la acción realizada por el administrador y modifica los datos del usuario.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Si el usuario no se encuentra en la BD o los datos del usuario están incorrectos el sistema muestra un mensaje.
<b>Sección “Mostrar usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador inserta el nombre del usuario que desea mostrar.	2. El sistema verifica que el usuario exista en la BD. 3. El sistema muestra los datos del usuario
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Si el usuario no existe en la BD el sistema muestra un mensaje.

Tabla 3 Gestionar usuario

#### CU4. Asignar Permiso

<b>Caso de Uso:</b>	Asignar permiso
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de que el administrador pueda

	asignar permisos a los usuarios.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador asigna permisos a un usuario y termina cuando el sistema realiza la acción.
<b>Precondiciones:</b>	El administrador tiene que estar autenticado en el sistema.
<b>Referencias</b>	RF4
<b>Prioridad</b>	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el administrador selecciona la opción Asignar Permiso.	2. El sistema muestra la lista de permisos a asignar y los usuarios y las pc que están en la BD.
3. El administrador introduce el id del usuario o de la pc y entre paréntesis el id de los permisos que desea a asignar a dicho usuario o pc.	3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el administrador y asigna los permisos al usuario o a la pc.

Tabla 4 Asignar Permiso

## 2.4 Patrones de arquitectura

Los patrones arquitectónicos, o patrones de arquitectura, ofrecen soluciones a problemas de arquitectura de software. Dan una descripción de los elementos y el tipo de relación que tienen con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones (28).

### 2.4.1. Arquitectura N-capas

El desarrollo de aplicaciones en n-capas es un proceso iterativo de división del problema en piezas manejables denominadas componentes. Estos componentes, o "Componentes de Negocio, son "modelos software" basados típicamente en la "vista" de un objeto real, evento o proceso de negocio. Los

componentes software individuales pueden formar parte y adaptarse tanto de estructuras independientes como de sistemas colaborativos (29).

### **Ventajas del modelo**

- Desarrollos paralelos (en cada capa)
- Aplicaciones más robustas debido al encapsulamiento
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica)
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad. (29)

### **2.4.2. Arquitectura basada en plugin**

Un plugin es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

El sistema utiliza una arquitectura basada en plugin, lo que significa, que permite extender la aplicación sin necesidad de acceder al código fuente, solamente agregando plugins que implementen las funcionalidades deseadas. Además este tipo de arquitectura trae como beneficios que el sistema sea escalable.

## 2.5 Arquitectura del Sistema

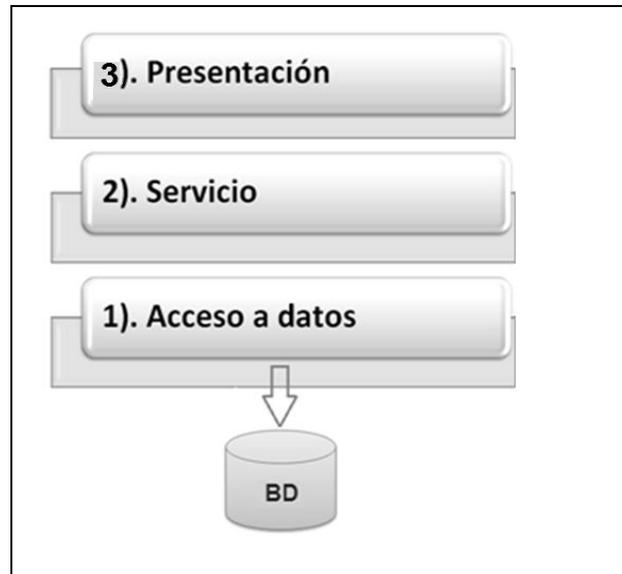


Ilustración 3 Arquitectura del Sistema

### 2.5.1 Descripción de los componentes de la arquitectura

- **Acceso a datos:** permite la comunicación y extracción de los datos almacenados, contiene los datos de los usuarios, las computadoras y el dominio al cual pertenecen los mismos.
- **Servicio:** brinda todas las funcionalidades de la aplicación.
- **Presentación:** muestra al administrador la interfaz de la aplicación.
- **BD:** se encuentran almacenados los datos.

## 2.6 Diagrama de paquetes

Como se hace referencia en la arquitectura del Sistema, el mismo está compuesto por 4 capas, la capa de Presentación, la de Servicios, la de Acceso a Datos y la capa de Dominio, las cuales se van a representar a continuación como subpaquetes dentro del diagrama de paquetes.

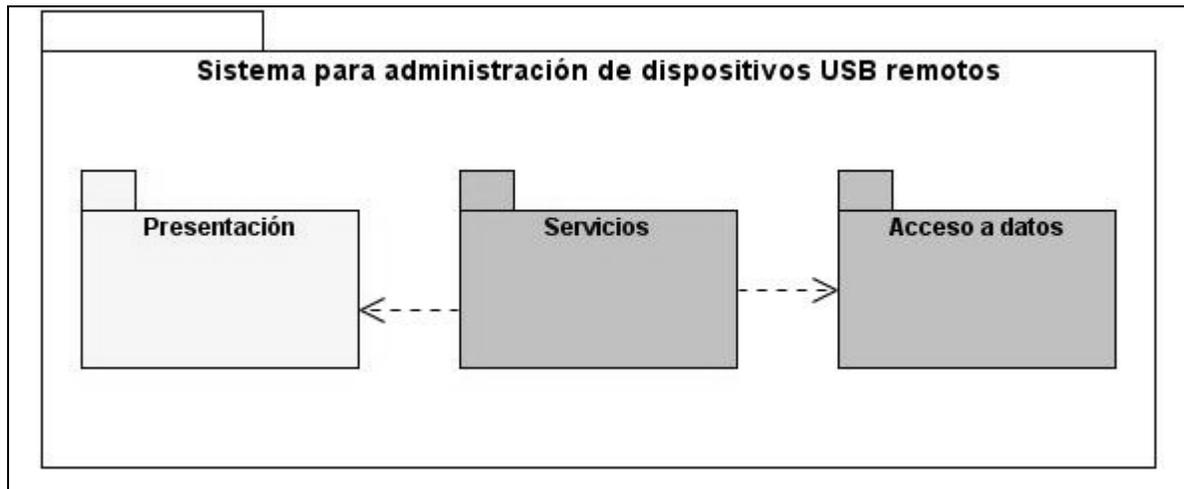


Ilustración 4 Diagrama de paquetes

### 2.6.1 Descripción de los Paquetes

**1. Paquete de Presentación:** esta capa contiene la clase `ProveedorCapaInterfazUsuario` que permite la comunicación de la capa de servicio con la de interfaz de usuario, que implementa esta interfaz, además contiene la clase `InterfazPluginInterfazUsuario` que la misma permite implementar un plugin para mostrar una interfaz de usuario al administrador. Actualmente se encuentra implementado un plugin que brinda una interfaz en consola con comandos para realizar todas las acciones posibles.

**2. Paquete de Servicios:** esta capa contiene la clase servicio principal que es la que inicia la aplicación, la cual inicializa la clase `ServicioConexionCliente`, y las dos clases `ServicioInterfazUsuario` y la clase `ServicioAccesoDato` que son las que se conectan con las capas correspondientes, también contiene la clase `ServicioManejadorResultados` que se conecta con la clase `InterfazPluginManejadorResultados` que permite implementar un plugin para controlar la respuesta del sistema. Actualmente está implementado un plugin que muestra los resultados en la pantalla.

**3. Paquete de Acceso a Datos:** esta capa controla el acceso del sistema a la Base de Datos (BD), contiene la clase `ProveedorCapaAccesoDato` que permite la comunicación entre la capa de servicio y la capa de Acceso a Datos, además contiene la clase `AccesoDatos` que es la que implementa esa interfaz, además contiene la clase `InterfazPluginConexionBaseDato` que es la que permite crear plugin que se conecten a distintas BD. En esta capa se encuentra implementado el plugin de conexión a BD SQLite, además consta de tres clases, la clase `usuario` la cual contiene los datos de un usuario, la clase `permiso`

que contiene los datos de un permiso, y por último la clase computadora que contiene los datos de una computadora.

## 2.7 Diagrama de clases del diseño

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema y describen su estructura. Los diagramas de clases contienen los siguientes elementos:

- Clases
- Interfaces
- Colaboraciones
- Relaciones de dependencia, generalización y asociación.

Estos diagramas son lo más importantes del diseño, contienen toda la información de las clases y sus relaciones (30).

### 2.7.1 Diagrama de clases del diseño para el Servidor

A continuación se representa el diagrama de clases del diseño, dividido por cada una de las capas que lo componen.

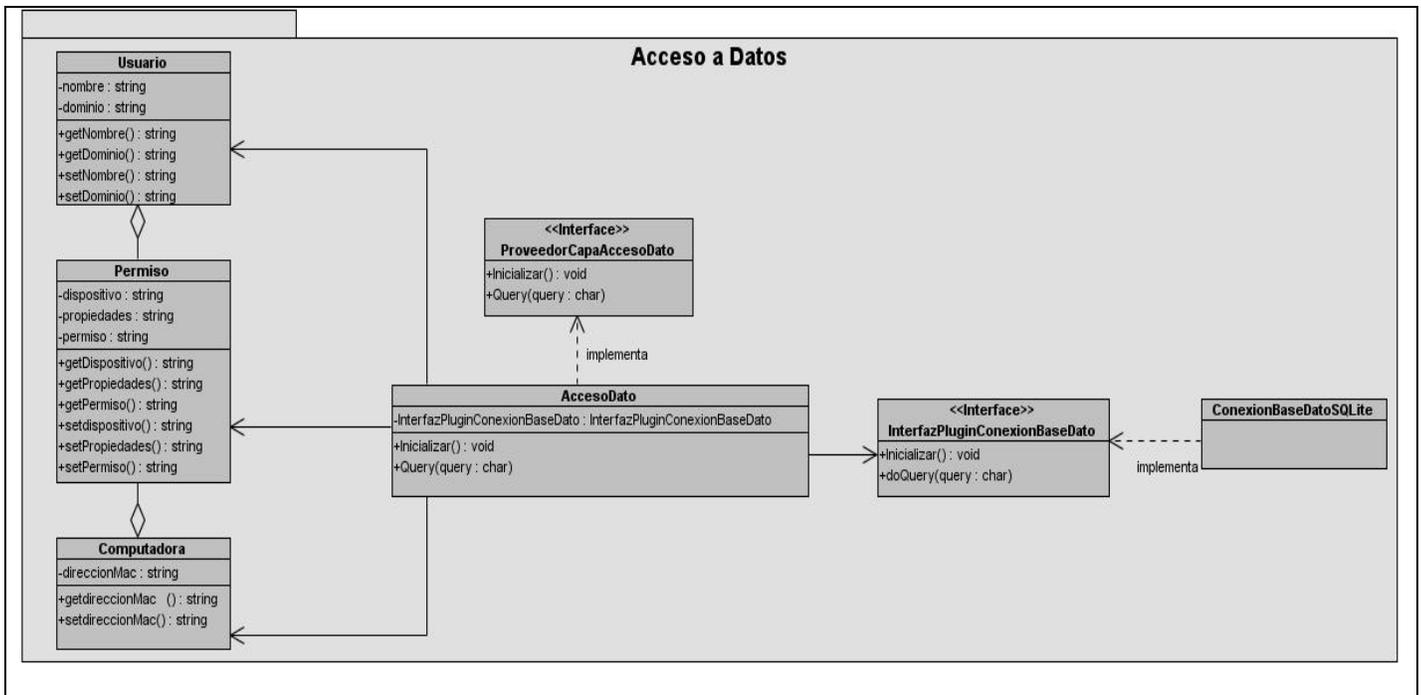


Ilustración 5 Diagrama de clases del diseño capa Acceso a Datos

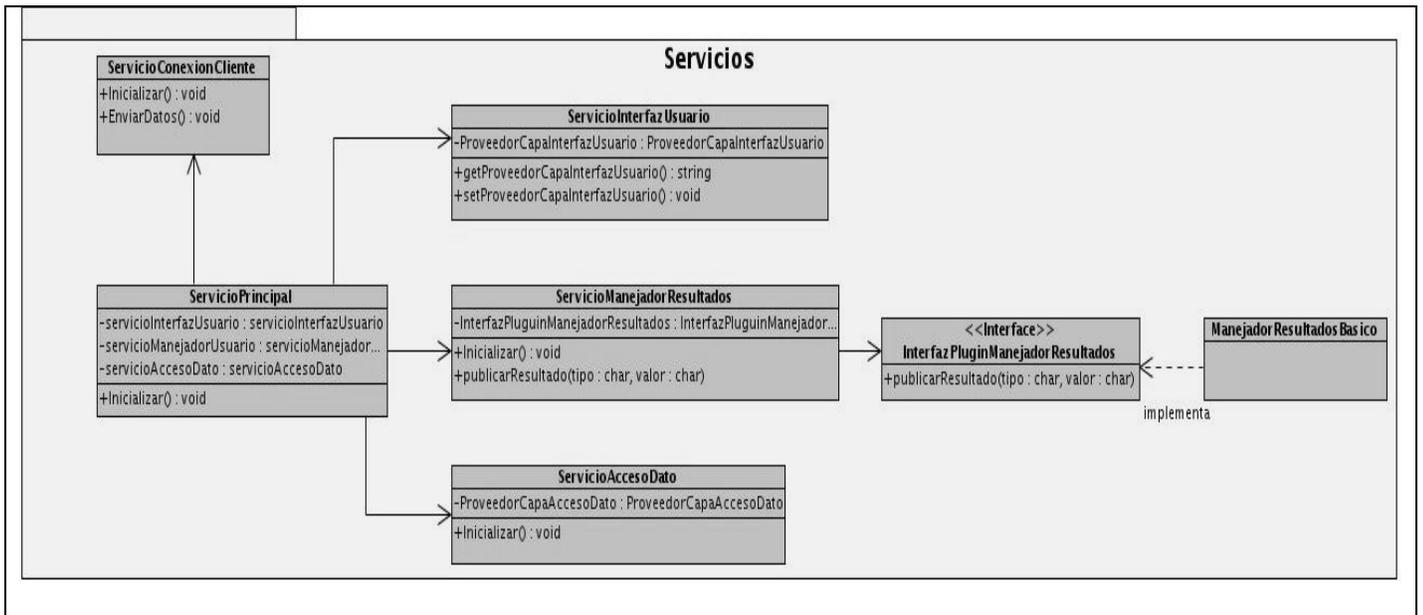


Ilustración 6 Diagrama de clases del diseño capa Servicios

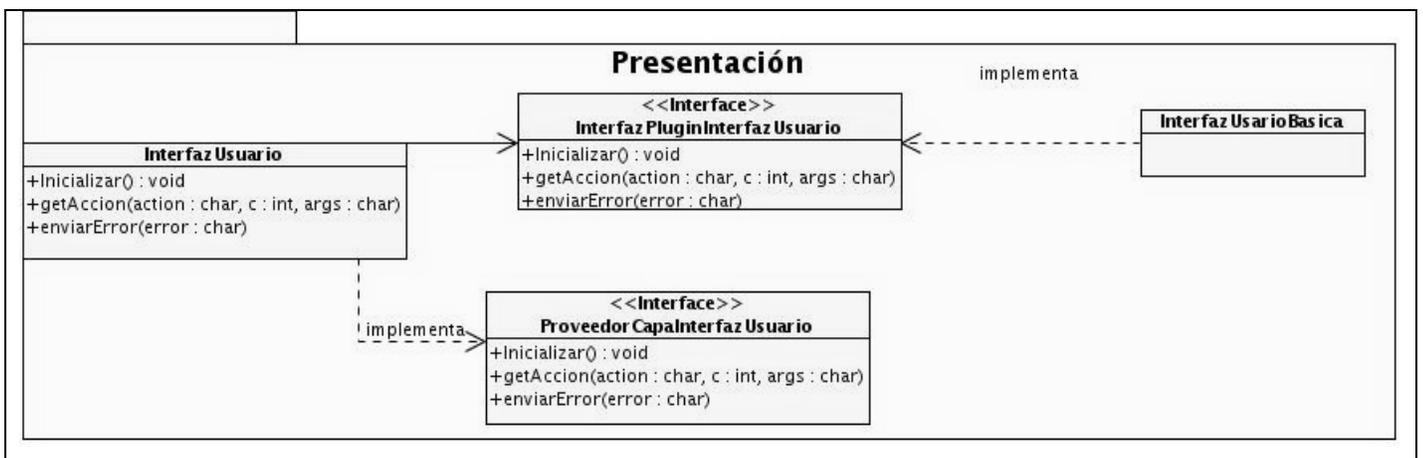


Ilustración 7 Diagrama de clases del diseño capa Presentación

### 2.7.2 Diagrama de estructuras de datos para el Cliente

El sistema a desarrollar está basado en una arquitectura cliente - servidor, la parte del cliente está programada en C, el mismo es un lenguaje estructurado, por lo que no posee el tipo de datos definido como clase, en dichas estructuras se definen comportamientos (métodos) y propiedades (atributos). A

continuación se representa el diagrama de estructuras de datos que es muy similar a un diagrama de clases.

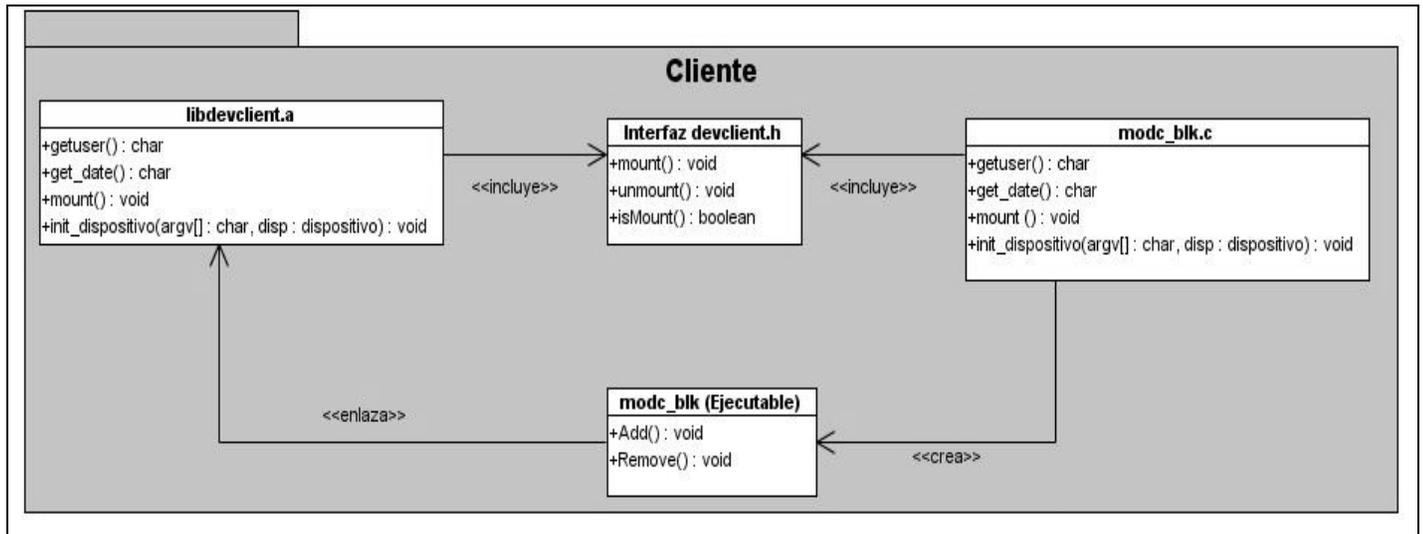


Ilustración 8 Diagrama de Estructura de Datos (cliente)

### 2.7.3 Descripción de las clases

A continuación se describe cada una de las clases pertenecientes a la capa de Servicios, el resto de las tablas se encuentran descritas en los anexos del documento.

<b>Nombre: ServicioConexionCliente</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
Funciones	
Inicializar	
EnviarDatos	
Descripción	Esta clase tiene la responsabilidad de implementar las funcionalidades que permiten enviar y recibir los datos del cliente.

Tabla 5 Descripción clase ServicioConexionCliente

<b>Nombre: ServicioPrincipal</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
servicioInterfazUsuario	servicioInterfazUsuario
servicioManejadorUsuario	servicioManejadorUsuario
servicioAccesoDato	servicioAccesoDato
Funciones	
Inicializar	
<b>Descripción</b>	Esta clase es la que inicia la aplicación, y permite el acceso a la capa de servicio.

Tabla 6 Descripción clase ServicioPrincipal

<b>Nombre: ServicioInterfazUsuario</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
ProveedorCapaInterfazUsuario	ProveedorCapaInterfazUsuario
Funciones	
getProveedorCapaInterfazUsuario	
setProveedorCapaInterfazUsuario	
<b>Descripción</b>	Esta clase conecta la capa de servicio con la caspa interfaz de usuario.

Tabla 7 Descripción clase ServicioInterfazUsuario

<b>Nombre: ServicioManejadorResultados</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
InterfazPluguinManejadorResultados	InterfazPluguinManejadorResultados
Funciones	
Inicializar	
publicarResultado	
<b>Descripción</b>	Esta clase contiene los métodos para manejar los resultados de las acciones de la aplicación, esta carga un plugin para definir como se muestra estos resultados.

--	--

Tabla 8 Descripción clase ServicioManejadorResultados

<b>Nombre: ServicioAccesoDato</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
ProveedorCapaAccesoDato	ProveedorCapaAccesoDato
Funciones	
Inicializar	
<b>Descripción</b>	Esta clase conecta la capa de servicio con la capa acceso a datos.

Tabla 9 Descripción clase ServicioAccesoDato

<b>Nombre: InterfazPluginManejadorResultados</b>	
<b>Tipo de clase: Servicio</b>	
Atributo	Tipo
ProveedorCapaAccesoDato	ProveedorCapaAccesoDato
Funciones	
publicarResultado	
<b>Descripción</b>	Esta clase representa la interfaz en la cual los plugin que implemeten el manejador de resultados deben heredar.

Tabla 10 Descripción clase ServicioAccesoDato

### 2.7.4 Patrones de diseño.

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Representa un esquema o microarquitectura que supone una solución a problemas (dominios de aplicación) semejantes; una estructura común que tienen aplicaciones semejantes (31).

#### Instancia única (Singleton):

El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una, se puede implementar en cualquier lenguaje orientado a objetos.

Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Se implementa creando en una clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado) (32).

**Singleton provee una única instancia global gracias a que:**

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea instanciable directamente (32).

**2.8 Diseño de la BD**

Para diseñar la Base de Datos es necesario primeramente presentar el diagrama de clases persistentes con sus atributos y relaciones.

**2.8.1 Diagrama de clases persistentes**

Las clases persistentes son aquellas que tienen durabilidad en el tiempo. Son el punto de partida para la creación del modelo físico de datos.

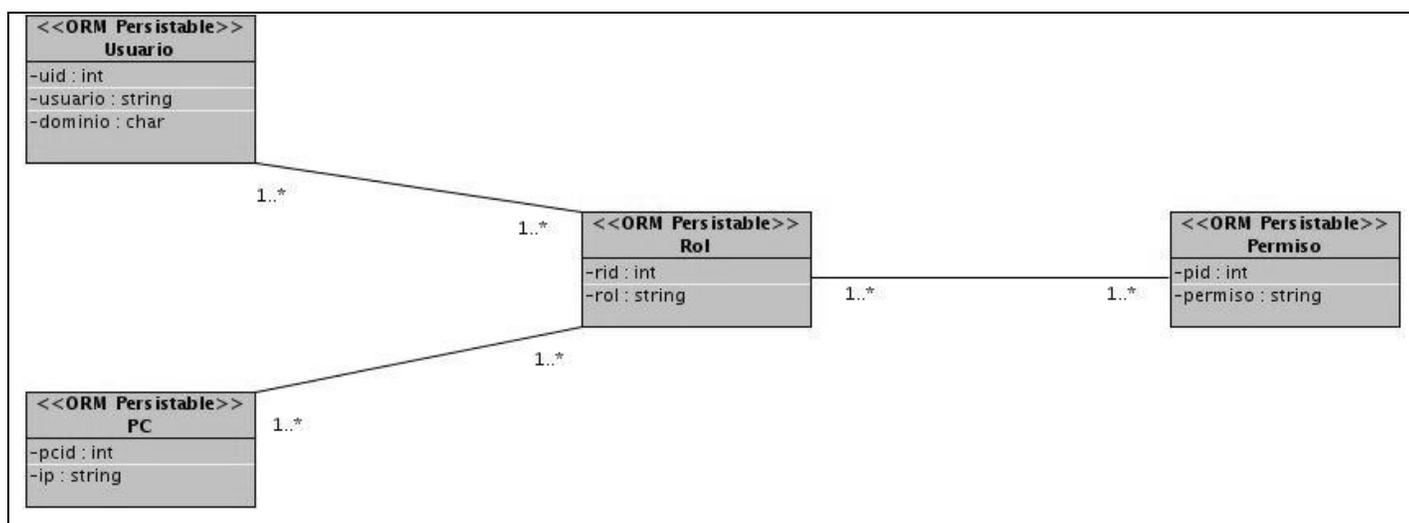


Ilustración 9 Diagrama de Clases Persistentes

**2.8.2 Modelo Entidad- Relación**

Partiendo del diagrama de clases persistente visto anteriormente, se define el siguiente modelo físico de la

base de datos o modelo Entidad-Relación:

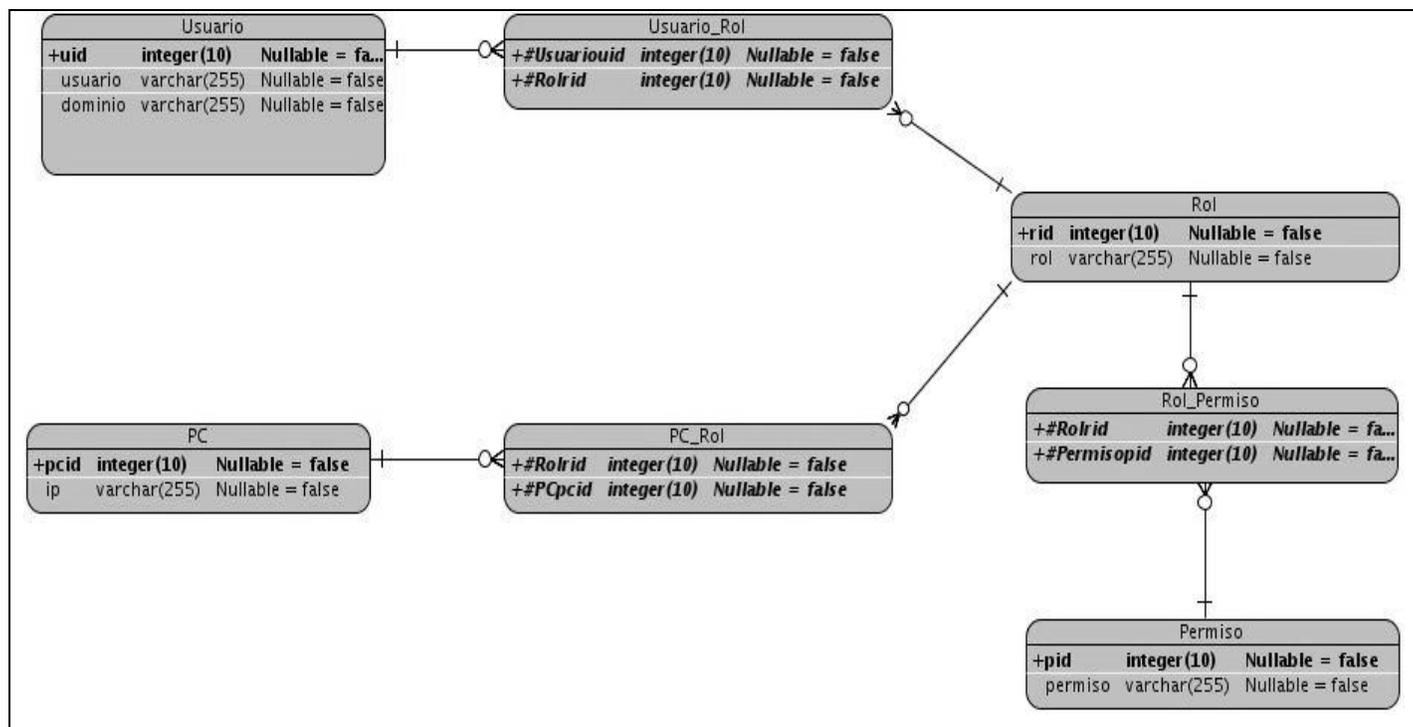


Ilustración 9 Diagrama Entidad-Relación

### 2.8.2.1. Descripción de las tablas

A continuación se describe la tabla usuario, la cual almacena los datos de los usuarios de la aplicación. El resto de las tablas se encuentran descritas en los anexos del documento.

Nombre: <b>usuario</b>		
Descripción: <b>Tabla para almacenar los datos de los usuarios</b>		
Atributo	Tipo	Descripción
<b>Uid</b>	<b>Int</b>	<b>Identificador del usuario</b>
<b>Usuario</b>	<b>Varchar</b>	<b>Nombre del usuario</b>
<b>Dominio</b>	<b>Varchar</b>	<b>Dominio al que pertenece el usuario</b>

Tabla 11 Descripción clase Usuario

## 2.9 Conclusiones

En la realización de este capítulo se presenta el diagrama de CU del sistema, el diagrama de clases persistentes, el modelo entidad-relación, el modelo de dominio y sus entidades, así como el modelo de diseño, entre otros. Se muestra el listado de requisitos funcionales y no funcionales, y se describen cada uno de los casos de uso del sistema que los engloba. Se plantea que el sistema a desarrollar consta de una arquitectura compuesta por 3 capas, la de acceso a datos, la capa de servicios y la de presentación, cada una de ellas con sus funciones pertinentes, además se escogió como patrón de diseño el de instancia única (singleton).

Partiendo de todo lo planteado y analizado en este capítulo se logrará una eficaz implementación, pues el diseño es factor elemental para la comprensión de la estructura del sistema y su composición.

## Capítulo 3. Implementación y Prueba

### Introducción

En este capítulo se detallan los estándares de codificación que sirven de base para tener un buen manejo de los elementos de programación. Se realiza la validación del sistema a través del plan de pruebas y se abordan aspectos del modelo de implementación, como el diagrama de componentes y despliegue.

### 3.1. Modelo de implementación

El modelo de implementación permite organizar los componentes de acuerdo con los mecanismos de estructuración y modularización disponible en el entorno de implementación y en el lenguaje utilizado, y muestra la dependencia entre componentes.

#### 3.1.1. Diagrama de componentes

Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interface (33).

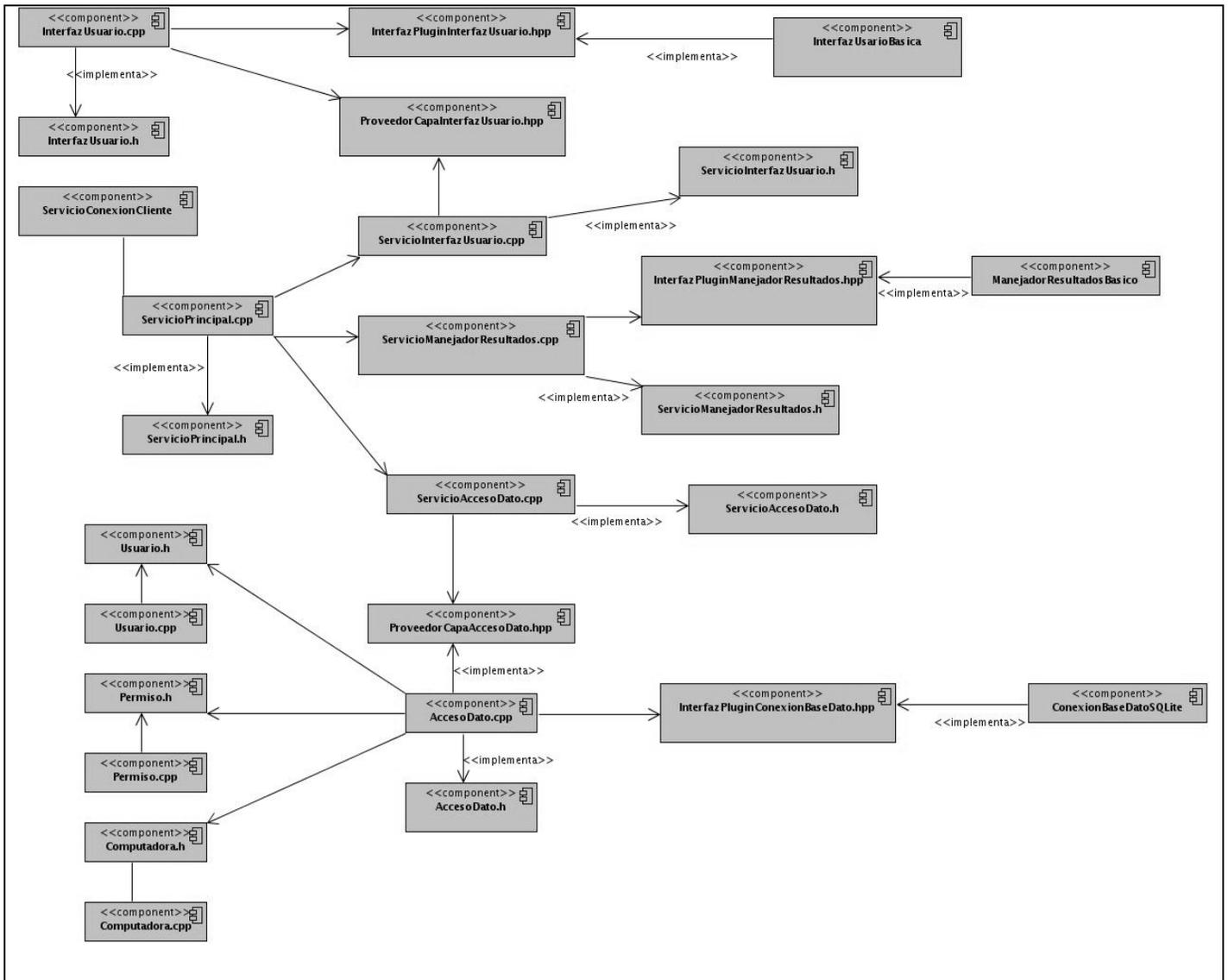


Ilustración 10 Diagrama de componentes

### 3.2. Estándares de codificación

Los estándares de codificación son pautas de programación que están enfocadas a la estructura de la lógica del código para facilitar su lectura, comprensión y mantenimiento (34).

A continuación se presentan algunas de estas convenciones para la programación en lenguaje C++.

### Identificadores de variables

- Cuando las variables están compuestas por más de una palabra la 1era palabra comienza con minúscula y el resto comienzan con mayúscula.
- Para distinguir palabras dentro del nombre deberá emplearse una letra mayúscula o un guión bajo (\_), sin mezclar ambas formas en un mismo programa.

#### Ejemplos:

```
nombreDeVariable  
nombre_de_variable
```

### Identificadores de tipos definidos por el usuario

- La primera letra será mayúscula.

**Ejemplo:** class Clase o struct Estructura

### Identificadores generales

- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.

## 3.3 Pruebas

Las pruebas de software permiten evaluar un producto desde un punto de vista crítico. En este proceso, el probador que es el responsable de llevar a cabo las pruebas, somete al producto a una serie de acciones inquisitivas y este a su vez responde con un comportamiento determinado (35).

### 3.3.1 Pruebas de funcionalidades

Este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y se deben realizar pruebas de comportamiento ante distintos escenarios (36).

A continuación se muestran los Casos de Pruebas correspondientes a los Casos de Uso descritos anteriormente.

**Descripción de las variables**

	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	string	No	Puede contener cualquier cadena alfa-numérica. El primer carácter no puede ser un número.
2	Dominio	string	No	Puede contener siguientes caracteres: solo letras y ptos. Ejemplo: uci.cu

**Caso de Prueba Gestionar Usuarios**

Escenario	Descripción	Nombre	Dominio	Respuesta del sistema	Flujo central
EC 1.1 Insertar usuario correctamente.	El administrador introduce los datos del usuario correctamente.	V	V	El sistema verifica que los datos estén correctos, y que el usuario no exista en la BD, de ser así, almacenará el usuario en el sistema.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Insertar Usuario". El sistema le permite introducir los datos. El administrador introduce los datos y el sistema los verifica.
		"ymorav"	"uci.cu"		
EC 1.2 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario de forma incorrecta.	I	I	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
		(vacío)	"001.00"		
EC 1.3 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario de forma incorrecta.	I	I	El sistema verifica que el usuario no se encuentre en la BD, de no ser así, emitirá un mensaje de error indicando que el usuario ya existe.	
		"ymorav"	"üci.cu"		

Escenario	Descripción	Nombre	Dominio	Respuesta del sistema	Flujo central
EC 2.1 Modificar usuario correctamente.	El administrador modifica los datos del usuario correctamente.	V "ymorav"	V "uci.cu"	El sistema verifica que los datos estén correctos, y que el usuario exista en la BD, de ser así, modificará los datos del usuario.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Modificar Usuario". El sistema le permite introducir el nombre del usuario y luego le muestra los datos para que este los modifique.
EC 2.2 Modificar usuario incorrectamente.	El administrador modifica los datos del usuario incorrectamente.	I "fhf5d"	I (vacío)	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos o que el usuario no se encuentra en la BD.	

Escenario	Descripción	Nombre	Respuesta de sistema	Flujo Central
EC 3.1 Eliminar usuario correcto	El administrador inserta el usuario que desea eliminar	"ymorav"	El sistema verifica que el usuario exista en la BD, en caso de ser así muestra un mensaje de confirmación de la acción: "¿Seguro que desea eliminar el usuario ymorav?". El administrador escoge la opción "Si" y el sistema elimina el usuario de la base de datos.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los usuarios existentes en la BD y el administrador elige la opción eliminar.

EC 3.1 Eliminar usuario incorrecto	El administrador inserta el usuario que desea eliminar	"51512"	El sistema verifica que el usuario exista en la BD, en caso de no ser así muestra un mensaje de error, indicando que el usuario no se encuentra en la BD.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los usuarios existentes en la BD y el administrador elige la opción eliminar
EC 3.2 Cancelar la operación.	El administrador introduce la opción NO		El sistema deshace la operación de eliminar.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los usuarios existentes en la BD y el administrador elige la opción eliminar El sistema le permite eliminar un usuario y cancelar la operación. El administrador escoge la opción "NO".

Escenario	Descripción	Nombre	Respuesta del sistema	Flujo central
EC 1.1 Mostrar usuario correctamente	El administrador inserta el usuario que desea mostrar.	"ymorav"	El sistema verifica que el usuario exista en la BD, de ser así, mostrará los datos del usuario.	El administrador accede al sistema y elige la opción mostrar usuario, el sistema la brinda la posibilidad de entrar el nombre, una vez que el administrador inserta el nombre del usuario el sistema verifica que se encuentre en la BD, en caso de ser así muestra los datos de dicho usuario.
EC 1.1 Mostrar usuario incorrectamente	El administrador inserta el usuario que desea mostrar.	"hgvhg"	El sistema verifica que el usuario exista en la BD, de no ser así, mostrará un mensaje de error que el usuario es incorrecto o que no existe en la BD.	

**Descripción de las variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	ID de la PC	Campo de Texto	No	valores numéricos
2	IP de la PC	Campo de Texto	No	Puede contener cadenas alfa-numéricas y puntos, puede empezar nunca con puntos ni terminar con puntos.

**Caso de Prueba Gestionar PC**

Escenario	Descripción	ID de la PC	IP de la PC	Respuesta del sistema	Flujo central
EC 1.1 Insertar pc correctamente.	El administrador introduce los datos de la pc correctamente.	V "5"	V "10.33.4.210"	El sistema verifica que los datos estén correctos, y que la pc no exista en la BD, de ser así, almacenará la pc en el sistema.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Insertar PC". El sistema le permite introducir los datos. El administrador introduce los datos y el sistema los verifica.
EC 1.2 Insertar pc incorrectamente.	El administrador introduce los datos de la pc de forma incorrecta.	I (vacío)	I "58.09j"	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
EC 1.3 Insertar pc incorrectamente.	El administrador introduce los datos de la pc de forma	I	I	El sistema verifica que la pc no se encuentre en la BD, de no ser así,	

	incorrecta.	"5"	"10.33.4.210"	emitirá un mensaje de error indicando que la pc ya existe.	

Escenario	Descripción	id de PC	IP de la PC	Respuesta del sistema	Flujo central
EC 2.1 Modificar pc correctamente.	El administrador modifica los datos de la pc correctamente.	V	V	El sistema verifica que los datos estén correctos, y que la pc exista en la BD, de ser así, modificará los datos de la pc.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Modificar PC". El sistema le permite introducir el id de la PC y el ip y luego le muestra los datos para que este los modifique.
		"5"	"10.33.4.210"		
EC 2.2 Modificar pc incorrectamente.	El administrador modifica los datos de la pc incorrectamente.	I	I	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos o que la pc no se encuentra en la BD.	
		"p"	(vacío)		

Escenario	Descripción	ID PC	IP de la PC	Respuesta de sistema	Flujo Central
EC 3.1 Eliminar pc correctamente	El administrador inserta el id de la pc que desea eliminar	"5"	"10.33.4.210"	El sistema verifica que la pc exista en la BD, en caso de ser así muestra un mensaje de confirmación de la acción: "¿Seguro que desea eliminar el la pc 5?". El administrador escoge la opción "Si" y el sistema elimina la pc de la base de datos.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra las pc existentes en la BD y el administrador elige la opción eliminar.

EC 3.1 Eliminar pc incorrectamente	El administrador inserta la pc que desea eliminar	“pg”	“10.33.210”	El sistema verifica que la pc exista en la BD, en caso de no ser así muestra un mensaje de error, indicando que la pc no se encuentra en la BD.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra las pc existentes en la BD y el administrador elige la opción eliminar
EC 3.2 Cancelar la operación.	El administrador introduce la opción NO			El sistema deshace la operación de eliminar.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra las pc existentes en la BD y el administrador elige la opción eliminar El sistema le permite eliminar una pc y cancelar la operación. El administrador escoge la opción “NO”.

Escenario	Descripción	ID de la PC	Ip de la PC	Respuesta del sistema	Flujo central
EC 1.1 Mostrar pc correctamente	El administrador inserta la pc que desea mostrar.	“5”	“10.33.4.210”	El sistema verifica que la pc exista en la BD, de ser así, mostrará los datos de la pc.	El administrador accede al sistema y elige la opción Mostrar PC, el sistema la brinda la posibilidad de entrar el id de la pc y el ip, una vez que el administrador inserta el ID y el ip, el sistema verifica que se encuentre en la BD, en caso de ser así muestra los datos de dicha pc.
EC 1.1 Mostrar pc incorrectamente	El administrador inserta la pc que desea mostrar.	“hgg”	“10..210”	El sistema verifica que la pc exista en la BD, de no ser así, mostrará un mensaje de error que la pc es incorrecta o que no existe en la BD.	

**Descripción de las variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	ID del Rol	Campo de Texto	No	valores numéricos
2	Rol	Campo de Texto	No	Puede contener cadena de caracteres.

**Caso de Prueba Gestionar Rol**

Escenario	Descripción	id Rol	Rol	Respuesta del sistema	Flujo central
EC 1.1 Insertar rol correctamente.	El administrador introduce los datos del rol correctamente.	V	V	El sistema verifica que los datos estén correctos, y que el rol no exista en la BD, de ser así, almacenará el rol en el sistema.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Insertar rol". El sistema le permite introducir los datos. El administrador introduce los datos y el sistema los verifica.
		"5"	"secretaria"		
EC 1.2 Insertar rol incorrectamente.	El administrador introduce los datos del rol de forma incorrecta.	I	I	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
		(vacío)	"58.09j"		
EC 1.3 Insertar rol incorrectamente.	El administrador introduce los datos del rol de forma incorrecta.	I	I	El sistema verifica que el rol no se encuentre en la BD, de no ser así, emitirá un mensaje de error indicando que el rol ya existe.	
		"5"	"secretaria"		

Escenario	Descripción	id Rol	Rol	Respuesta del sistema	Flujo central
EC 2.1 Modificar rol correctamente.	El administrador modifica los datos del rol correctamente.	V "5"	V "secretaria"	El sistema verifica que los datos estén correctos, y que el rol exista en la BD, de ser así, modificará los datos del rol.	El administrador con los permisos pertinentes accede a la consola de administración y elige la opción "Modificar Rol". El sistema le permite introducir el nombre del rol y luego le muestra los datos para que este los modifique.
EC 2.2 Modificar rol incorrectamente.	El administrador modifica los datos del rol incorrectamente.	I "pepe"	I (vacío)	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos o que el rol no se encuentra en la BD.	

Escenario	Descripción	ID Rol	Rol	Respuesta de sistema	Flujo Central
EC 3.1 Eliminar rol correctamente	El administrador inserta el rol que desea eliminar	"5"	"secretaria"	El sistema verifica que el rol exista en la BD, en caso de ser así muestra un mensaje de confirmación de la acción: "¿Seguro que desea eliminar el rol 5?". El administrador escoge la opción "Si" y el sistema elimina el rol de la base de datos.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los roles existentes en la BD y el administrador elige la opción eliminar.
EC 3.1 Eliminar rol incorrectamente	El administrador inserta el rol que desea eliminar	"pepe"	"5"	El sistema verifica que el rol exista en la BD, en caso de no ser así muestra un mensaje de error, indicando que el rol no se encuentra en la BD.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los roles existentes en la BD y el administrador elige la opción eliminar

EC 3.2 Cancelar la operación.	El administrador introduce la opción NO			El sistema deshace la operación de eliminar.	El Administrador con los permisos pertinentes accede al sistema. El sistema le muestra los roles existentes en la BD y el administrador elige la opción eliminar El sistema le permite eliminar un rol y cancelar la operación. El administrador escoge la opción "NO".
-------------------------------	---	--	--	--	---

### Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	ID de permiso	Campo de texto	No	valores numéricos.
2	ID PC	Campo de texto	No	valores numéricos.
3	ID Usuario	Campo de texto	No	Contiene una cadena de caracteres..

### Caso de prueba Asignar Permisos:

Escenario	Descripción	ID Usuario	ID PC	ID Permisos	Respuesta del sistema	Flujo central
EC1.1 Asignar Permisos correctamente.	El administrador introduce el ID del usuario y de los permisos correctamente.	V "4"	V	V " 1,2,3,4 "	El sistema verifica que los datos estén correctos, de ser así, asignará los permisos correctamente al usuario.	El administrador con los permisos pertinentes accede al sistema y en la consola de administración elige la opción Asignar Permisos. El sistema muestra una lista de permisos, usuarios y pc y el administrador debe introducir el id del usuario o la pc a la cual
EC1.2 Asignar Permisos correctamente.	El administrador introduce el ID de la pc y de los permisos correctamente.	V	V "5"	V " 1,2,3,4 "	El sistema verifica que los datos estén correctos, de ser así, asignará los permisos	

					correctamente a la pc.	se le quiere otorgar los permisos y el id de esos permisos entre paréntesis. El sistema verifica la entrada y en caso de que esté incorrecta emite un mensaje de error.
EC 1.2 Asignar Permisos incorrectamente.	El administrador introduce el ID del usuario y de los permisos incorrectamente.	 "4e"		 (vacío)	El sistema verifica que los datos estén correctos, de no ser así, mostrará un mensaje de error indicando que los id no corresponden a ningún usuario o permiso de la BD.	
EC 1.2 Asignar Permisos incorrectamente.	El administrador introduce el ID de la pc y de los permisos incorrectamente.		 "34j5"	 (vacío)	El sistema verifica que los datos estén correctos, de no ser así, mostrará un mensaje de error indicando que los id no corresponden a ninguna pc o permiso de la BD.	

**Caso de Prueba Dar Acceso a Dispositivo**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Dar Acceso a un dispositivo.	El usuario conecta un dispositivo en la computadora.	El sistema verifica si el usuario está autenticado en el sistema y pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo. En otro caso verifica si la pc pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo, en caso de ser así le da al usuario acceso al dispositivo.	El usuario conecta una computadora en la pc, el sistema verifica que esté autenticado o que pertenezca al dominio y tenga los permisos necesarios para ver el contenido del dispositivo. En otro caso verifica si la pc pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo, en caso de ser así le da al usuario acceso al dispositivo
EC 1.2 Denegar Acceso a un dispositivo.	El usuario conecta un dispositivo en la computadora.	El sistema verifica si el usuario está autenticado en el sistema y pertenece al dominio y tiene	

		<p>los permisos pertinentes para acceder al dispositivo. En otro caso verifica si la pc pertenece al dominio y tiene los permisos pertinentes para acceder al dispositivo, en caso de no ser así el sistema no le da acceso al dispositivo.</p>	
--	--	---	--

### 3.4 Resultados de las pruebas de funcionalidades

Durante las pruebas internas<sup>4</sup> realizadas al sistema se detectaron un grupo de no conformidades. Se probaron 13 casos de pruebas, los cuales arrojaron en la primera iteración 7 no conformidades, de las cuales todas fueron resueltas, y en una segunda iteración 2 no conformidades que se resolvieron de igual manera. A continuación se muestra un gráfico que representa las no conformidades. De esto se puede inferir que las pruebas de funcionalidades fueron exitosas.

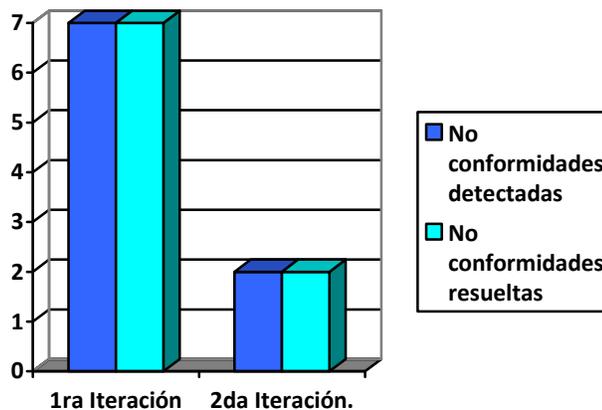


Ilustración 11 . Diagrama de No Conformidades

A continuación se representa un gráfico donde se evidencia que el caso de uso con mayor número de No conformidades fue el Gestionar Usuarios y el que menos errores presentó en las pruebas fue el Asignar Permiso, en el caso de uso Dar acceso a Dispositivo no se detectaron errores.

<sup>4</sup> **Internas:** Llevadas a cabo por el equipo de desarrollo de la aplicación.



Ilustración 12 Diagrama de No Conformidades por Casos de Uso.

### 3.4. Modelo despliegue

Un diagrama de despliegue muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos.

- Permiten modelar la disposición física o topología de un sistema.
- Muestra el hardware usado y los componentes instalados en el hardware.
- Muestra las conexiones físicas entre el hardware y las relaciones entre componentes (37).

**Computadora Cliente:** computadora a la cual un usuario conecta un dispositivo.

**Servidor de Aplicaciones:** funciona como intermediario entre las computadoras clientes y el servidor que guarda la información (servidor BD).

**BD SQLite:** Contiene toda la información del sistema.



Ilustración 13. Modelo de Despliegue

### 3.5 Conclusiones

En el presente capítulo se realizó el diagrama de componentes. Luego se realizaron las pruebas al sistema, o sea, los casos de pruebas correspondientes a los casos de uso. Por último se analizó el diagrama de despliegue del sistema.

## Conclusiones Generales

Una vez desarrollado el sistema para la administración remota de dispositivos USB, se puede arribar a las siguientes conclusiones:

- Las herramientas y tecnologías libres aseguran la independencia tecnológica del sistema desarrollado, lo cual resulta acorde con la política del país de migrar a software libre.
- La arquitectura de software utilizada permitió lograr un sistema escalable, o sea, que brinda la posibilidad de incluir nuevos plugin en un futuro.
- La utilización del protocolo SSH permitió implementar mecanismos que hacen más seguras la comunicación entre el cliente y el servidor.
- La base del sistema implementado permite la administración exitosa de dispositivos USB de almacenamiento.
- Las pruebas realizadas permitieron lograr un producto que cumple con los requisitos planteados inicialmente.

## Recomendaciones

A partir del estudio realizado se recomienda:

- Incluir nuevos plugin o modificar los plugin de interfaz de usuario y el plugin de Manejador de Resultados Básico con el fin de lograr una interfaz más amigable para el administrador y en caso que se quiera modificar las conexiones a las Bases de Datos modificar el plugin Conexión a la Base de Datos SQLite.
- Continuar ampliando el sistema en cuanto a categorías de dispositivos USB que se pueden administrar remotamente mediante él.

## Bibliografía Referenciada

1. **Aldegani, G. M.** Seguridad Informática. Argentina : MP Ediciones, 1997.
2. Security through Aspect Oriented Programming in Proceedings of the 1st working conference on Network Security. **Win, Bart De, Vanhaute, Bart y Decker, Bart De.** 2001.
3. verbatim-latinoamerica.com. [En línea] [Citado el: 5 de mayo de 2012.] <http://www.verbatim-latinoamerica.com/subcat/hard-drives/portable/>.
4. Ecured. [En línea] [Citado el: 12 de mayo de 2012.] [www.ecured.cu/index.php/Memoria\\_flash](http://www.ecured.cu/index.php/Memoria_flash).
5. OCU. [En línea] [Citado el: 20 de mayo de 2012.] [www.ocu.org/disco-duro-externo/](http://www.ocu.org/disco-duro-externo/).
6. Promoción de Negocios. [En línea] 2011. [Citado el: 8 de mayo de 2012.] <http://www.promonegocios.net/administracion/definicion-administracion.html>.
7. Sostic. [En línea] 2011. [Citado el: 9 de mayo de 2012.] <http://sostic.farvista.net/?q=node/23>.
8. Manual de Ayuda de la Aplicación. *Manual*.
9. USB over Network. [En línea] 2010. [Citado el: 11 de mayo de 2012.] <http://www.usb-over-network.com/usb-over-network.html>.
10. Introducción a la Ingeniería de Software. [En línea][Citado el: mayo 18, 2012.] [http://eva.uci.cu/file.php/58/Bibliografia/Ingenieria\\_de\\_SW/Conferencias/Conferencias\\_IS1\\_2007-2008/Conferencias.rar](http://eva.uci.cu/file.php/58/Bibliografia/Ingenieria_de_SW/Conferencias/Conferencias_IS1_2007-2008/Conferencias.rar).
11. Sánchez, María Mendosa. Metodologías de desarrollo de software. [En Línea][Citado el: mayo 18, 2012.] [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
12. Buenas Tareas. [En línea] [Citado el: 20 de mayo de 2012.] <http://www.buenastareas.com/ensayos/Metodologia-Open-Up/446545.html>.
13. WordPress. [En línea] 2 de septiembre de 2008. [Citado el: 13 de mayo de 2012.] <http://cbasqa.wordpress.com/2008/09/02/proceso-de-desarrollo-openup/>.
14. **UML.ORG. Object Management Group - UML.** UML.ORG. [En línea] [Citado el: 9 de mayo de 2012.] [http://www.uml.org/..](http://www.uml.org/)
15. Informática Blog. [En línea] 2009. [Citado el: 16 de mayo de 2012.] <http://informatica-full2.blogspot.com/2009/06/historia-del-lenguaje-c.html>.

16. Monografías. [En línea] [Citado el: 16 de mayo de 2012.] <http://www.monografias.com/trabajos4/lenguajec/lenguajec.shtml>.
17. Ecured. [En línea] [Citado el: 8 de mayo de 2012.] [http://www.ecured.cu/index.php/Visual\\_Paradigm](http://www.ecured.cu/index.php/Visual_Paradigm).
18. Rational. [En línea] [Citado el: 10 de mayo de 2012.] <http://www.rational.com.ar/herramientas/roseenterprise.html>
19. Ecured. [En línea] [Citado el: 11 de mayo de 2012.] [http://www.ecured.cu/index.php/IDE\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n).
20. web.mit.edu. [En línea] [Citado el: 18 de mayo de 2012.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>.
21. cavsi. [En línea] [Citado el: 12 de mayo de 2012.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
22. blackberry. [En línea] [Citado el: 12 de mayo de 2012.] [http://docs.blackberry.com/en/developers/deliverables/16511/SQLite\\_overview\\_701956\\_11.jsp](http://docs.blackberry.com/en/developers/deliverables/16511/SQLite_overview_701956_11.jsp).
23. desphpparatodos blog. [En línea] 2011. [Citado el: 13 de mayo de 2012.] <http://desphpparatodos.blogspot.com/2011/01/introduccion-mysql-conceptos-basicos.html>.
24. Blog de Daniel Pecos. [En línea] [Citado el: 14 de mayo de 2012.] [http://danielpecos.com/docs/mysql\\_postgres/x15.html](http://danielpecos.com/docs/mysql_postgres/x15.html).
25. Aplicaciones Empresariales. [En línea] [Citado el: 17 de mayo de 2012.] <http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>.
26. Ecured. [En línea] [Citado el: 12 de mayo de 2012.] <http://www.ecured.cu/index.php/> .
27. mastermagazine. [En línea] [Citado el: 18 de mayo de 2012.] <http://www.mastermagazine.info/termino/4184.php>.
28. slideshare. [En línea] [Citado el: 20 de mayo de 2012.] <http://www.slideshare.net/josecuartas/patrones-de-arquitectura-softwarecapa-de-datos>.
29. Buenas Tareas. [En línea] [Citado el: 19 de mayo de 2012.] <http://www.buenastareas.com/materias/arquitectura-n-capas/0>.
30. blogspot. [En línea] [Citado el: 13 de mayo de 2012.] <http://egdamar877.blogspot.com/2009/05/expocicion.html>.

31. microsoft. [En línea] [Citado el: 14 de mayo de 2012.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
32. **microsoft**. [En línea] [Citado el: 12 de mayo de 2012.] <http://msdn.microsoft.com/es-es/library/bb972272.aspx>.
33. —. microsoft. [En línea] [Citado el: 15 de mayo de 2012.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
34. Blog de Miguel Matas. [En línea] [Citado el: 18 de mayo de 2012.] <http://www.miguelmatas.es/blog/2008/05/06/como-definir-un-estandar-de-codificacion-yo-trabajo/>.
35. Pruebas de Software. [En línea] [Citado el: 25 de mayo de 2012.] [pruebasdesoftware.com/laspruebasdesoftware.htm](http://pruebasdesoftware.com/laspruebasdesoftware.htm)
36. Guia Web. [En línea] [Citado el: 26 de mayo de 2012.] <http://www.guiaweb.gob.cl/guia/capitulos/cuatro/queprobar.htm>.
37. scribd. [En línea] [Citado el: 17 de mayo de 2012.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.

## Bibliografía Consultada

1. **Graut et al(2004)**. "Metodología de la Investigación". s.l. : Fondo editorial Coruniversitaria, 2da ed, 2004.
2. **Hill, Mc Graw**. Sampieri et al(2006) "Metodología de la Investigación". s.l. : 4ta ed, 2006.
3. Arquitectura de software. [En línea] [Citado el: 18 mayo, 2012.] <http://www.sei.cmu.edu/architecture/definitions.html>.
4. **Menéndez-Barzanallana**, Rafael Asensio. Principales herramientas CASE del mercado y su uso. [En línea] [Citado el: mayo 17, 2012.] [http://www.um.es/docencia/barzana/LAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/LAGP/Enlaces/CASE_principales.html).