

Universidad de las Ciencias Informáticas
Facultad 1



**Título: Sistema para la administración central de
dispositivos USB remotos.**

Trabajo de Diploma para optar por el título de

Ingeniero en ciencias Informáticas

Autor: Alfonso Fernández Céspedes

Tutor: MsC. Dariem Pérez Herrera.

Co-tutor: Ing. Edilberto Blez Deroncelé

Junio del 2013

“Año 55 de la Revolución”



La conciencia es el resultado del conocimiento.

Hugo Rafael Chávez Frías.

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Alfonso Fernández Céspedes

Firma del Autor

MsC. Dariem Pérez Herrera

Firma del Tutor

Ing. Edilberto Blez Deroncelé

Firma del Co-tutor



DATOS DE CONTACTO

MsC. Dariem Pérez Herrera: Graduado de Ciencias de la Computación en la Universidad Central “Marta Abreu” de Las Villas (UCLV). Fecha de graduado: 14 de julio del 2006. Posee 7 años de graduado y actualmente se desempeña como Administrador de Repositorios del Centro de Software Libre (CESOL) y pertenece al departamento de Sistemas Operativos y Desarrollo de Tecnologías Libres.

Correo: dariemp@uci.cu.

Ing. Edilberto Blez Deroncelé: Graduado de “Ingeniero en Ciencias Informáticas” en el 2011. Categoría docente. Instructor Recién Graduado en Adiestramiento. Posee 4 años de experiencia en el tema y 2 años de graduado. Actualmente se desempeña como Jefe de los proyectos Protección de Puertos USB (PPUSB) y Auditoría de Código Fuente (ACF) del Centro de Software Libre (CESOL) y pertenece al departamento de Sistemas Operativos y Desarrollo de Tecnologías Libres.

Correo: eblez@uci.cu.



AGRADECIMIENTOS

A mis 2 madres que me quieren con la vida, a mi abuela por estar siempre detrás de mí para que estudiase y fuera alguien en esta vida, a mi mamá la madre más cariñosa y preocupada del mundo a veces en exceso.

A mi padre por inculcar en mí grandes valores como el respeto y la solidaridad.

A mis tutores en especial a Eedito que más que tutor fue un amigo, hermano y padre para mí, que bajo su tutela aprendí muchas cosas.

A mis primos Pedro, Jorge y Alejandro que más que primos son hermanos para mí.

A mis hermanitos Ramón, Chang, Tomás, Yannier y Barban.

A mis compañeros del antiguo grupo 2, en especial a Mayi, Made, Leidys, Islen, Greysi y Daymaris.

A mis compañeros de apartamento por el ambiente de solidaridad creado estos 5 años y ser mi otra familia.

A mi hermana Noris a pesar de que la vida puso barreras entre nosotros y nos conocimos tarde, ha demostrado que siempre estuvo presente, a mis sobrinitos y mi cuñado Denys.

A mi familia de Santiago a mí tía Blanca, mi primo Ricardito a María Elena y a Milena que tanto han hecho por mí.

A mi tío Camilo y a su familia por su apoyo incondicional.

A mis vecinos Robert, Mariluz, Melisa, Gisela, Lías, Rosi, Tanita y Dania por más que vecinos ser como una familia.

A mis hermanos de la infancia Marlon, Yasmanis.

A mi abuela Monine y a mi tía Irene.

Al equipo de desarrollo de NOVA por acogerme entre ellos estos últimos años tanto a los estudiantes como a los profesores, en especial a Puente, Haniel, Michel, Abel, Migue, Juan, Allan, Daniel y a Raidel.

A todas las personas que de una forma u otra influyeron en mi personalidad.



DEDICATORIA

A mis madres, mi abuela y mi mamá por todo el cariño y confianza depositada en mí por ser más que madres padres también, por guiarme por el buen camino.

A mi padre el cual es también un ejemplo a seguir, por todo el apoyo y cariño.

A mi hermana Noris por estar siempre presente ahí, a mis sobrinitos y a mi cuñado Denys.

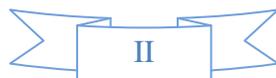
A mis primos Jorge, Pedro y Alejandro por su apoyo incondicional durante estos 5 años de estudio.

A mi familia en general.

A mis vecinos, amigos, compañeros de aulas y a todas las personas que creyeron en mí.

A la memoria de mi abuelo Aniano uno de los hombres más grande que he conocido.

A la memoria del Comandante Chávez ejemplo para todos los latinoamericanos.



RESUMEN

La necesidad de evitar el robo de información sensible en cualquier dispositivo USB de almacenamiento, ya sea en una memoria o un disco duro, o la impresión de un documento confidencial por una impresora no autorizada, como parte de la política de seguridad en las instituciones cubanas, impone contar con un sistema que permita restringir el uso de dichos dispositivos.

Este trabajo presenta un sistema para la administración remota de dispositivos USB, que garantiza una mayor seguridad en el manejo y control de estos. Para el desarrollo del sistema, se realizó un diseño que cumple con los parámetros establecidos por la arquitectura, definiéndose una arquitectura cliente-servidor, para el sistema en general y 3 capas para el servidor, se emplean como herramienta de modelado Visual Paradigm, como lenguaje de programación se utilizó C para el lado del cliente y C++ para el lado del servidor, así como se escogió el protocolo SSH para la comunicación entre el cliente y el servidor, también se decidió escoger como Sistema Gestor de Base de Datos SQLite3, entre otras tecnologías seleccionadas para la implementación. Además se realizó un periodo de pruebas para verificar el correcto funcionamiento de la aplicación, presentándose como resultado un sistema listo para su instalación y funcionamiento.

Palabras claves: Administración, Dispositivo USB, Seguridad.

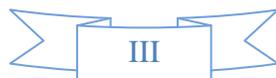


TABLA DE CONTENIDOS

DECLARACIÓN DE AUTORÍA	I
DATOS DE CONTACTO	I
AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
TABLA DE CONTENIDOS	IV
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
Introducción	6
1.1 Seguridad.....	6
1.2 Dispositivos USB.....	8
1.2.1 Dispositivos de almacenamiento.....	9
1.2.2 Dispositivos de impresión.....	11
1.3 Administración de dispositivos USB.	12
1.4 Estudio de sistemas que permiten la administración de dispositivos USB.....	13
1.5 Metodología de desarrollo.....	19
1.5.1. Proceso unificado abierto (OpenUP).....	19
1.6 Lenguajes de modelado y programación.	21
1.6.1 Otras tecnologías a utilizar.....	24
1.7 Herramientas de modelado.....	25

1.7.1 Visual Paradigm.....	25
1.8 Entorno de desarrollo (IDE).....	26
1.9 Protocolo de comunicación.....	26
1.10 Sistema Gestor de Base de Datos (SGBD).....	27
1.10.1 SQLite3.....	27
1.11 Conclusiones Parciales.....	28
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....	29
Introducción.....	29
2.1 Modelo de dominio.....	29
2.1.1. Descripción de las clases del dominio.....	30
2.2. Requisitos del sistema.....	31
2.2.1 Requisitos Funcionales.....	31
2.2.2 Requisitos No Funcionales.....	32
2.3 Diagrama de Casos de Usos del sistema.....	33
2.3.1 Descripción de actores.....	34
2.3.2 Descripción de los CU del sistema.....	35
2.3.3 Especificación de los CU del sistema.....	35
2.4 Patrones de arquitectura.....	41
2.4.1 Arquitectura en capas.....	42
2.4.2 Arquitectura Cliente-Servidor.....	43
2.5 Arquitectura del sistema.....	43
2.5.1 Arquitectura del Servidor.....	44
2.5.2 Descripción de los componentes de la arquitectura del servidor.....	45
2.6 Diagrama de paquetes.....	45
2.6.1 Descripción de los paquetes.....	46
2.7 Diagrama de clases del diseño.....	47

2.7.1 Diagrama de clases del diseño del Servidor.	47
2.7.2 Descripción de las clases del sistema.	51
2.7.3 Diagrama de estructuras de datos para el cliente.	53
2.7.4 Patrones de diseño.	54
2.8 Diseño de la base de datos.	56
2.8.1 Diagrama de clases persistentes.	56
2.8.2 Modelo entidad-relación.....	57
2.8.2.1 Descripción de las tablas.....	57
2.9 Conclusiones parciales.	58
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.	60
Introducción.	60
3.1 Modelo de implementación.	60
3.1.1 Diagrama de componentes.	60
3.2 Estándares de codificación.	62
3.3 Pruebas.	64
3.3.1 Pruebas de caja negra.	64
3.4 Resultados de las pruebas.	69
3.5 Modelo de despliegue	70
3.6 Conclusiones Parciales.	70
CONCLUSIONES	71
RECOMENDACIONES	72
REFERENCIAS BIBLIOGRÁFICAS	73
BIBLIOGRAFÍA	77

ANEXOS78

GLOSARIO DE TÉRMINOS.....92

INTRODUCCIÓN

La curiosidad del hombre y su capacidad para crear, le ha permitido avanzar a nivel tecnológico, a medida que la vida misma se encarga de imponerle nuevos retos, Esta curiosidad aborda diferentes puntos de vista. Por un lado, están aquellos que se dedican a buscar las posibles fallas de los descubrimientos realizados, bien sea para mejorarlos, o simplemente para obtener conocimiento y por el otro, están aquellos que las aprovechan para fines destructivos. En el medio se encuentra el resto de la humanidad, disfrutando o sufriendo con los nuevos descubrimientos.

Curiosamente, el avance tecnológico necesita todos los puntos de vista para seguir creciendo. Este avance trae consigo nuevos problemas que requieren nuevas soluciones. La información ha sido siempre un bien incalculable y protegerla es una tarea continua y de vital importancia. Con el avance de la tecnología y el surgimiento de la información digital se hace cada vez más sencillo guardarla, transferirla y consultarla, pero se hace también más difícil protegerla. A medida que se crean nuevas técnicas para la transmisión de la información, los curiosos idean otras que les permitan acceder a ella sin ser autorizados. La seguridad de la información no es solo la aplicación o uso de un nuevo programa capaz de proteger y preservar nuestra información. Es todo el conjunto de acciones encaminadas a preservar la integridad, disponibilidad, confidencialidad y autenticidad de los datos.

En la actualidad, muchas empresas creen que por tener un potente antivirus, cortafuegos (firewall en inglés) o seguridad de contenido de correo contra amenazas externas, están exentas de sufrir algún tipo de pérdida de información. No siempre se tiene en cuenta el factor humano y se menosprecian las amenazas presentadas por los dispositivos USB¹, ya sea por un dispositivo de almacenamiento o por uno

1 **USB:** De Universal Serial Bus. Es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.

de tipo impresor (impresoras), causando como resultado la exposición de información delicada o introducir algún código malicioso. El uso de un sistema operativo como GNU/Linux puede implicar cierta seguridad en comparación con otros sistemas operativos, sin embargo este no se encuentra libre de las amenazas que traen los dispositivos USB, que pueden llegar a ser fatales para la preservación de información confidencial.

Como parte del proceso de migración a plataformas de *software* libre y de código abierto que lleva a cabo el territorio cubano, muchas instituciones utilizan la distribución cubana de GNU/Linux Nova, tal es el caso de la Universidad de las Ciencias Informáticas (UCI) y algunas instituciones como los Organismos de la Administración Central del Estado (OACE), debido a que es un producto desarrollado por los cubanos y para los cubanos, donde se adiciona a las 4 libertades del *software* libre las 4S o sea: seguridad, soberanía tecnológica, sostenibilidad y socio-adaptabilidad. Sin embargo, es necesario para estos organismos como parte de su política de seguridad, restringir el uso de algunos de estos dispositivos USB.

Todas estas entidades trabajan con información delicada, por lo que es necesario evitar a toda costa el robo de información en un dispositivo de almacenamiento o la conexión de una impresora que no esté autorizada. En un intento para controlar estas amenazas, se han establecido políticas internas que tratan de controlar el uso de dispositivos de almacenamiento o de tipo impresor, pero no siempre se debe confiar en el factor humano. La mejor manera de asegurar un control completo sobre dichos dispositivos es colocando barreras tecnológicas.

Para un administrador de red sería más sencillo y seguro, tener una aplicación que controle el uso de estos dispositivos USB para así tener un control eficiente en cuanto al manejo de información, evitando el robo y la introducción de algún código maligno que pudiese afectar la infraestructura de la entidad.

En sistemas operativos privativos, como Microsoft Windows, existen aplicaciones para controlar dispositivos USB, tal es el caso de GFI Endpoint Security, MyUSBOnly, USBDeview entre otras. El estudio realizado en bibliografías especializadas en el tema, no arroja ninguna referencia a la existencia de una herramienta con este fin en sistemas GNU/Linux. Por lo que se hace necesario desarrollar una aplicación que permita administrar de forma remota los dispositivos USB en el sistema operativo GNU/Linux Nova y que a su vez satisfaga las políticas de seguridad de las entidades cubanas.

Por lo anteriormente expuesto queda definido el siguiente **Problema de Investigación**: ¿Cómo administrar de forma centralizada los dispositivos USB dentro de una red?

Como **Objeto de Estudio**: Los dispositivos USB y sus propiedades.

El **Campo de Acción** de la investigación: Los dispositivos USB de almacenamiento y los de tipo impresor.

El **objetivo general** de la investigación es implementar un sistema para la administración central de dispositivos USB de forma remota en sistemas GNU/Linux haciendo uso de la información que proveen los fabricantes de dichos dispositivos.

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

- 1-Realizar un estudio del arte de sistemas que permiten la administración de dispositivos USB.
- 2-Implementar el uso de protocolos seguros en la comunicación entre el ordenador servidor y los clientes.
- 3-Desarrollar el módulo para el control de los dispositivos USB de almacenamiento masivo.
- 4-Desarrollar el módulo para el control de impresoras.
- 5-Implementar una interfaz gráfica para la aplicación del lado del servidor.

Planteándose la siguiente **Idea a defender**:

Con el desarrollo de un sistema, se podrán administrar remotamente los dispositivos externos de bloques que almacenan información, así como los dispositivos de impresión.

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

- Levantamiento de los requisitos del sistema.
- Realización de una investigación del funcionamiento UDEV.
- Realización de una investigación del funcionamiento y reconocimiento de dispositivos USB de almacenamiento.
- Realización de una investigación del funcionamiento y reconocimiento de impresoras.
- Realización de una investigación del funcionamiento de la biblioteca GTK+-2.0.

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

Métodos teóricos:

- Analítico-Sintético: Este método se utilizó para estudiar las teorías y los documentos referentes a los dispositivos USB y a la administración de los mismos, sintetizando los elementos más importantes relacionados con el objeto de estudio por partes y en su conjunto.
- Histórico-Lógico: Este método se utilizó para estudiar toda la documentación relacionada con la administración remota de los dispositivos USB, para así obtener conocimiento histórico de su evolución, desarrollo y comportamiento a nivel internacional y nacional.

Métodos empíricos:

- Observación: Este método se utilizó para estudiar el comportamiento de aplicaciones utilizadas en el control de los dispositivos USB para hacer un análisis detallado de las principales funcionalidades de estas aplicaciones.

Con esta investigación se pretende alcanzar como **posible resultado** una aplicación que permita en el sistema operativo GNU/Linux Nova:

- Controlar de forma sencilla los dispositivos USB en una red.
- Agregar soporte para los dispositivos de tipo impresor.
- Controlar mediante una interfaz gráfica a los dispositivos USB.

La investigación será abordada en tres capítulos:

Capítulo 1: Fundamentación teórica: En este capítulo se hace un análisis 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: Análisis y Diseño del sistema: En este capítulo se presentan los requisitos 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: En este capítulo quedará plasmado cómo será implementado el sistema, su estructuración en clases y componentes que garantizan la capacidad operacional del producto, con la explicación detallada de cada componente y su funcionalidad; así como la elaboración de las pruebas para detectar las posibles no conformidades y corregirlas.

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 USB, las cuales sentaron las bases para la conformación de los requisitos del sistema. También se determina la metodología, herramientas, lenguajes de programación, entorno de desarrollo (IDE por sus siglas en inglés) y mecanismos de seguridad a utilizar en el proceso de desarrollo del sistema.

1.1 Seguridad.

Según la Real Academia Española de la Lengua (RAE) el término seguridad se refiere a certeza, conocimiento seguro y claro de algo; que previene algún riesgo o asegura el buen funcionamiento de un mecanismo, precaviendo que este falle o se violente.

En el contexto informático se puede definir que la seguridad informática es el conjunto de métodos y herramientas destinados a proteger los bienes o activos (tangibles e intangibles) informáticos de una institución (1). Sin embargo el principal objetivo de la seguridad informática es garantizar cuatro aspectos fundamentales:

- **Confidencialidad:** Los objetos de un sistema serán accedidos únicamente por entidades autorizadas (1).
- **Integridad:** Los objetos sólo pueden ser modificados por entidades autorizadas y de manera controlada (1).

- **Disponibilidad:** Los objetos deben permanecer accesibles a entidades autorizadas (1).
- **Autenticación:** Verificar la identidad del emisor y del receptor (1).

Los tres elementos que se deben proteger son el hardware (servidor, cableado, computadoras, entre otros.), el *software* (sistema operativo, aplicaciones) y los datos (documentos, bases de datos, entre otros.). 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 (1).
- **Interceptación:** Una entidad no autorizada consigue acceso a un recurso (1).
- **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 (1).
- **Fabricación:** Una entidad no autorizada inserta objetos falsificados en el sistema (1).

Para evitar que estas amenazas se conviertan en vulnerabilidades la seguridad informática propone cinco principios básicos:

- **Mínimo privilegio:** se deben otorgar los permisos estrictamente necesarios para efectuar las acciones que se requieran (1).
- **Eslabón más débil:** la seguridad de un sistema es tan fuerte como su parte más débil. Un atacante primero analiza cual es el punto más débil del sistema y concentra sus esfuerzos en ese lugar (1).
- **Proporcionalidad:** las medidas de seguridad deben estar en correspondencia con lo que se protege y con el nivel de riesgo existente. No sería lógico proteger con múltiples recursos un

activo informático que no posee valor o que la probabilidad de ocurrencia de un ataque sobre el mismo es muy baja (1).

- **Dinamismo:** la seguridad informática no es un producto, es un proceso. No se termina con la implementación de los medios tecnológicos, se requiere permanentemente control y mantenimiento (1).
- **Participación universal:** la gestión de la seguridad informática necesita de la participación de todo el personal de una institución. La seguridad que puede ser alcanzada mediante medios técnicos es limitada y debiera ser apoyada por una gestión y procedimientos adecuados, que involucren a todos los individuos (1).

Como conclusión la seguridad de un *software* es la combinación de su capacidad para garantizar la disponibilidad, integridad, privacidad, control y autenticidad de la información y demás activos informáticos cumpliendo con los principios básicos que establece la seguridad informática.

1.2 Dispositivos USB.

Los dispositivos USB tienen varios nombres, incluyendo dispositivos thumb (pulgar en inglés, pues 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 (2).

Estos pequeños dispositivos de **plug and play**² son extraíbles, regrabables, y excelentes para el almacenamiento de datos personales y profesionales, ya que muchos son dispositivos de hardware

² **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.

encriptados para mayor seguridad. Los dispositivos USB vienen en colores, estilos y capacidades que se adaptan a todas las edades y grupos demográficos (2).

1.2.1 Dispositivos de almacenamiento.

El concepto de dispositivos de almacenamiento engloba dos nociones. Los dispositivos son máquinas o sistemas capaces de desarrollar ciertas acciones y cumplir con un objetivo. El almacenamiento, por su parte, es la acción y efecto de almacenar (reunir o guardar información, registrar información) (3).

De esta manera, se puede afirmar que los dispositivos de almacenamiento son dispositivos que escriben o leen datos en formato dado. Estos dispositivos, por lo tanto, trabajan en conjunto con todos los medios donde se almacenan los archivos de una computadora u otro sistema informático, tanto lógicamente como físicamente (3).

Medios extraíbles:

Los medios extraíbles son aquellos medios 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 y discos duros externos (2).

Dispositivos externos de bloque:

Un dispositivo de bloque, 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, que no se puede dividir. Almacenan la información en bloques de tamaño fijo, cada uno con su propia

dirección. La propiedad esencial de un dispositivo de bloque es la posibilidad de leer o escribir en un bloque de forma independiente de los demás, es decir, el programa puede leer o escribir en cualquiera de los bloques (2).

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 (2).

Memorias USB:

Una memoria flash USB es un pequeño dispositivo de almacenamiento masivo electrónico que consta de una pequeña cubierta que protege los circuitos de almacenamiento y un conector de tipo USB. Permite la escritura y borrado de la información (archivos de oficina, videos, música, e incluso sistemas operativos), de manera rápida, sencilla y segura; siendo conectado por medio del puerto USB de la computadora. La definición de la RAE es simplemente "dispositivo portátil pequeño de almacenamiento de datos" (4).

Discos duros extraíbles:

Un disco duro externo, es un dispositivo de almacenamiento magnético, capaz de guardar grandes volúmenes de información, pero que no se encuentra montado dentro del gabinete de la computadora, sino que es posible conectarlo y utilizarlo externamente por medio de un cable hacia el puerto USB, el disco permanece girando todo el tiempo que se encuentra encendido, con un *software* incluido que lo instala en el sistema una vez que este se monta (2). Actualmente es una de las líneas de producto periféricos para computadoras que más auge ha alcanzado en el mercado mundial por las grandes ventajas que estos poseen como:

1. Un disco duro portátil no requiere de una fuente de alimentación externa (2).
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 (2).
3. Los discos duros portátiles se pueden usar fácilmente en múltiples PC para compartir archivos (2).
4. Un disco duro portátil permite archivar datos rápido y fácilmente, además permiten programar respaldos automáticos (2).

1.2.2 Dispositivos de impresión.

Un dispositivo de impresión es un dispositivo periférico que permite producir una gama permanente de textos o gráficos de documentos almacenados en formato electrónico, imprimiéndolos en medios físicos, normalmente en papel o transparencias.

Dispositivos de caracter.

Un dispositivo de caracteres, en informática es un componente del ordenador que se comunica con la unidad central de proceso por medio de bytes individuales, es decir envía carácter a carácter. Un ejemplo de estos dispositivos son los de impresión, el ratón y el teclado.

Impresora.

Una impresora es un periférico de computadora que permite producir una copia permanente de textos o gráficos de documentos almacenados en formato electrónico, imprimiéndolos en medios físicos, normalmente en papel o transparencias, utilizando cartuchos de tinta o tecnología láser y generalmente están permanentemente unidas a la computadora por un cable que puede ser USB o estar conectado al

puerto serial de la computadora. Otras impresoras, llamadas impresoras de red, tienen un interfaz de red interno (5).

Impresora Multifuncional.

Una impresora multifunción, también llamada MFP (*Multi Function Printer*) o simplemente multifunción, es un dispositivo electrónico que reúne en un único aparato varias funciones, como impresora, escáner, fotocopidora y fax. La impresora multifunción puede ser de inyección de tinta o láser, y operar como periférico de ordenador conectada a la computadora sirviendo como impresora o escáner, o bien en modo autónomo, sin necesidad de conectarla a la computadora funcionando como fotocopidora, fax o impresora directa (6).

1.3 Administración de dispositivos USB.

El concepto de administración hace referencia al funcionamiento, la estructura y el rendimiento de las organizaciones. El término proviene del latín *ad-ministrare* (“servir”) o *ad manus trahere* (“manejar” o “gestionar”) (7).

La administración es un proceso que consiste en las actividades de planeación, organización, dirección y control para alcanzar los objetivos establecidos utilizando para ellos recursos económicos, humanos, materiales y técnicos a través de herramientas y técnicas sistematizadas (8).

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

También se puede considerar como administración remota a la funcionalidad de algunos programas que permiten realizar ciertos tipos de acciones desde un equipo local y que las mismas se ejecuten en otro equipo remoto; es decir la administración remota es una función que permite a los administradores del sistema configurar de manera remota cualquier estación de trabajo o grupo de estaciones de trabajo de una red (2).

Luego de un exhaustivo análisis de todos los conceptos anteriormente planteados se puede llegar a la conclusión de que la administración remota de dispositivos USB es: el manejo y control eficiente de los dispositivos USB para lograr en el caso particular de la investigación que cumpla con las políticas de seguridad de las entidades cubanas.

Para administrar los dispositivos en un entorno empresarial, es necesario dar a los administradores toda la información de los dispositivos conectados a la red, para que puedan tener un control absoluto sobre estos, actualmente existen varias aplicaciones que existen con este fin; por lo que para darle cumplimiento al objetivo de la investigación fue necesario estudiar algunas aplicaciones que se dedican a la administración de dispositivos USB. A continuación se muestra el estudio y los resultados que se obtuvieron.

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

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

GFI Endpoint Security.

GFI EndPointSecurity es una solución de seguridad que le ayuda a mantener la integridad de los datos, al prevenir transferencia de contenido sin autorización desde y hacia los siguientes dispositivos portátiles:

- 1-Puertos USB (ejemplo: memorias flash y lectores de tarjeta de memoria, Pen Drives) (9).
- 2-Puertos *FireWire* (ejemplo: cámaras digitales, lectores de tarjetas *FireWire*) (9).
- 3-Conexiones de datos inalámbricas (ejemplo: *Bluetooth dongles*, Infrarrojo) (9).
- 4-Unidades de *Floppy* (internas y externas) (9).
- 5-Unidades ópticas tales como CD, DVD y unidades MO (magnetos ópticas) internas y externas (9).
- 6-Unidades de disco duro removible USB (9).
- 7-Otras unidades tales como unidades *Zip* y unidades de cinta (internas y externas) (9).

A través de su tecnología, GFI EndPointSecurity le da la opción de permitir o negar acceso a un dispositivo también como asignar privilegios de 'control total' o 'solo lectura'. Se puede también registrar la actividad de todos los dispositivos portátiles que son usados en los computadores objetivo incluyendo la fecha y hora de uso y quien utilizó esos dispositivos (9).

Entre sus principales características podemos encontrar, detecta dispositivos conectados y realiza varias tareas, evita la fuga y el robo de información mediante el control integral del acceso a dispositivos portátiles de almacenamiento y de impresión, 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 (9).

Esta aplicación solamente existe para el sistema operativo Microsoft Windows.

MyUSBOnly.

MyUSBOnly es una excelente herramienta que permite proteger las computadoras de intrusos, esta aplicación mantiene un gestor de dispositivos USB con contraseña, de este modo únicamente los USB permitidos podrán ser utilizados en la computadora, entre las características que esta herramienta posee están:

- 1-Bloquea la copia de archivos hacia dispositivos USB o desde los mismos, únicamente permite esto para USB permitidos.
- 2-Genera una bitácora de todos los dispositivos que se conecten a la computadora y todos los movimientos que hagan (copiar, cortar, eliminar).
- 3-Recibe notificaciones de correo cuando un USB no permitido es conectado a la computadora (10).
- 4-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 (9).

El fin de esta aplicación es prevenir el robo de datos mediante el bloqueo de todos ellos, además comprueba una lista blanca de dispositivos permitidos según su marca comercial (si se desea permitir que un determinado USB de determinada marca se utilice) o por número de serie (para determinados

dispositivos USB que se desee permitir) (9). Esta aplicación se usa solamente en el sistema operativo Windows.

USBDeview.

USBDeview es una pequeña aplicación que lista todos los dispositivos que están conectados a un ordenador así como los que fueron anteriormente conectados, para cada dispositivo se muestra la información ampliada como: nombre del dispositivo, descripción, tipo, número de serie, fecha y hora en que el dispositivo fue conectado, identificador del fabricante, identificador del producto entre otras propiedades (11).

Esta aplicación también permite desinstalar los dispositivos USB previamente usados así como activar o desactivar a los dispositivos USB constados en el sistema. También se puede utilizar de forma remota siempre y cuando se conecte al ordenador con el usuario de administrador. Esta aplicación solamente funciona en Microsoft Windows (9).

USB over Network.

USB over Network es una aplicación práctica que permite compartir fácilmente una unidad USB externa o una impresora en una red, es fácil de usar e instalar. Está compuesto de dos partes, cliente y servidor (12).

USB over Network surge ante la dificultad de utilizar un dispositivo que no está físicamente en el ordenador, a través de la parte cliente de USB over Network los ordenadores pueden acceder a un dispositivo USB conectado a un ordenador remoto (donde está instalada la parte del servidor) en una red de área local (LAN) o internet, y así utilizar el dispositivo como si estuviese conectado al ordenador (12).

Principales características de UBS over Network:

- 1-Permite funcionar con dispositivos USB remotos como con los locales.
- 2-Dispositivos USB ilimitados pueden ser compartidos en servidor.
- 3-Dispositivos USB ilimitados se pueden acceder desde el cliente.
- 4-Se admite una amplia variedad de dispositivos USB.
- 5-Intercambio automático de nuevos dispositivos USB.
- 6-Lista de exclusión de uso compartido automático.
- 7-Conexión automática de dispositivos USB que son compartidos en el servidor.
- 8-Admite la eliminación segura de los dispositivos USB. (12)

TrueCrypt.

TrueCrypt es un pequeño *software* de código abierto (open source) multiplataforma que permite encriptar en tiempo real unidades de disco utilizando varios métodos de encriptación. Este programa codifica y decodifica automáticamente los datos que se leen o graban en una determinada unidad. Estos datos no podrán ser leídos si no se ingresa la contraseña correspondiente (13).

TrueCrypt utiliza los algoritmos de cifrado AES (256 bits), Blowfish (448 bits), CAST, IDEA y Triple DES. Además el cifrado y el descifrado ocurren en la memoria RAM, de forma transparente al usuario, antes de ser escritos al disco (13).

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 (2).

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

El estudio realizado a las aplicaciones para administrar dispositivos USB sirvió para detectar las funcionalidades más significativas presentes en cada una de ellas. A continuación se mencionan estas funcionalidades:

- 1-Reconocer el tipo de dispositivo que se conectó.
- 2-Permitir o denegar el acceso a un dispositivo.
- 3-Asignar privilegios “completo, de solo lectura o ejecución”.
- 4-Restringir el uso de los puertos USB.
- 5-Mostrar la información de todos los dispositivos USB conectados a los ordenadores por ejemplo: nombre del dispositivo, modelo, identificador del fabricante, tipo de dispositivo, número de serie, descripción entre otros.

Las herramientas analizadas anteriormente sirvieron para guiar y dar una panorámica de cómo se va a desarrollar el sistema, puesto que muchas de las herramientas anteriormente expuestas funcionan en el sistema operativo Microsoft Windows y el sistema requiere que sea GNU/Linux específicamente para la distribución cubana GNU/Linux Nova por la migración que se está llevando a cabo en el territorio cubano a tecnologías de *software* libre y código abierto; y otras no cumplen algunas condiciones que se necesitan para desarrollar el sistema.

A continuación teniendo en cuenta el estudio realizado y tomando como base las funcionalidades nombradas se mencionan las posibles funcionalidades que va a tener el sistema a desarrollar:

- 1-Reconocer el tipo de dispositivo USB que se conecta (Impresoras, memorias flash y discos duros externos).
- 2-Gestionar usuarios.
- 3-Asignar permisos en dependencia del usuario, o sea las políticas de seguridad en cuanto al montaje del dispositivo “lectura y escritura o solo lectura”.
- 4-Permitir o denegar el acceso al dispositivo.

1.5 Metodología de desarrollo.

Como metodología de desarrollo se decidió escoger OpenUp porque presenta varios beneficios, ya que es apropiada 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. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas (2). Además es la metodología utilizada en el proyecto NOVA.

A continuación se detallan las principales características de OpenUp.

1.5.1. Proceso unificado abierto (OpenUP).

OpenUP conserva las características principales del modelo de desarrollo RUP (por sus siglas en inglés), incluye el desarrollo iterativo, permite identificar los requisitos operacionales del sistema, prever las interacciones con los usuarios y prevenir los posibles riesgos en el desarrollo del sistema (14).

OpenUP es una forma de desarrollo más ágil y ligera, consiste en equipos a los cuales se les asigna una fase del desarrollo que tienen que complementarse entre sí para obtener un buen producto final, no puede ser una sola persona la que realice todo el trabajo pues esto podría ocasionar que se pierda de vista ciertas características importantes, por ejemplo para un proyecto pequeño constituyen equipos de 3 a 6 personas e implican 3 a 6 meses de esfuerzo del desarrollo (14).

Principales características:

1-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 (2).

2-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 (2).

3-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 (2).

Fases de OpenUP

Concepción: Primera de las 4 fases en el proyecto del ciclo de vida, obteniendo suficiente información para confirmar que el proyecto debe hacer. El objetivo de esta fase es capturar las necesidades de los stakeholder³ en los objetivos del ciclo de vida para el proyecto (14).

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 la elaboración de la arquitectura del sistema (14).

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 (14).

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 (14).

1.6 Lenguajes de modelado y programación.

Para el desarrollo del sistema a implementar se escogieron los siguientes lenguajes: como lenguaje de modelado se escogió el lenguaje de modelado unificado (UML) y como lenguajes de programación C para el lado del cliente y C++ para el lado del servidor.

Lenguaje de modelado unificado (UML).

³ **Stakeholder:** las partes interesadas

Se escogió UML como lenguaje de modelado porque es la sucesión de una serie de métodos de análisis y diseño orientados a objetos que aparecen a fines de 1980 y principios de 1990. El UML, fusiona los conceptos de la orientación a objetos e incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos (15).

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. Una de las metas principales de UML es avanzar en el estado de la integración institucional proporcionando herramientas de interoperabilidad para el modelado visual de objetos (15).

Lenguaje de programación C.

Se escogió el lenguaje C porque además de ser una restricción del diseño, es uno de los lenguajes con el cual se desarrolla el sistema operativo GNU/Linux Nova y es el que interactúa directamente con el sistema.

El lenguaje C es un lenguaje de nivel medio, ya que combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador. Es un lenguaje estructurado, pues permite crear procedimientos en bloques dentro de otros procedimientos. Destacar también que C es un lenguaje portable, porque permite utilizar el mismo código en diferentes equipos y sistemas informáticos independientemente de la arquitectura de cualquier computadora (16).

Entre otras características se encuentra, que es un lenguaje con muy pocas palabras reservadas (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) (2).

A modo de conclusión se escogió el lenguaje de programación C para el lado del cliente por las siguientes razones:

1-Es un lenguaje de programación de propósitos generales.

2-No está asociado a ningún sistema operativo ni a ninguna arquitectura de computadora.

3-Es popular y eficaz.

4-Permite el desarrollo de sistemas operativos y programas de aplicación, tiene sólo 32 palabras reservadas.

5-Programación de bajo nivel (nivel bit).

Lenguaje de programación C++.

En el caso del lenguaje C++ se escogió para el lado del servidor por 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 de programación.

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. Además, ha eliminado algunas de las dificultades y limitaciones del C original (2).

1.6.1 Otras tecnologías a utilizar.

UDEV.

UDEV es un gestor de dispositivos para núcleos Linux 2.6. Se encarga de mantener actualizado el directorio */dev* con los dispositivos presentes en el sistema. Además, también puede realizar otras actividades sobre los dispositivos de las que no se encarga el núcleo (scripts de inicialización, permisos, carga de firmwares, etc.) (17).

La forma en la que UDEV gestiona los dispositivos del sistema es a través de reglas. Estas reglas se escriben en archivos de texto con el sufijo *.rules*, y se leen del directorio */etc/udev/rules.d/*. Como en otros sistemas en GNU/Linux, las reglas se ejecutan en orden lexicográfico, por eso suelen contener un número, que ayuda a ejecutar la regla en el orden deseado (17).

Además mediante de UDEV se puede acceder a todos los atributos de un dispositivo USB conectado en el sistema mediante la biblioteca *libudev*.

GTK+

GTK+ es la librería gráfica sobre la que se sustenta todo el interfaz gráfico de GNOME. Es una librería que contiene todo lo necesario para el desarrollo de interfaces gráficas, permitiendo la posibilidad de crear todo tipo de interfaces, desde las más básicas, como botones, etiquetas, cajas de texto, hasta cosas más complejas como selectores de ficheros, colores, fuentes, cajas de texto multilínea con soporte para todo tipo de efectos sobre el texto (18).

La librería GTK+ sigue el modelo de programación orientado a objetos. La jerarquía de objetos comienza en **GObject** de la librería *Glib* del que hereda **GtkObject**. Todos los *widgets* heredan de la clase de

objetos **GtkWidget**, que a su vez hereda directamente de **GtkObject**. La clase **GtkWidget** contiene las propiedades comunes a todos los *widgets*; cada *widget* particular le añade sus propias propiedades (18).

GTK+ utiliza GDK para visualizar los *widgets*. GDK es una interfaz de programación (API) de aplicaciones que se sitúa por encima de la API gráfica nativa. De esta forma portando GDK pueden utilizarse las aplicaciones construidas con GTK+ en otras plataformas (18).

1.7 Herramientas de modelado

Como herramienta de modelado se utilizó el Visual Paradigm. A continuación se menciona las características de esta herramienta.

1.7.1 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. Es multiplataforma (2).

Permite:

- 1-Crear y visualizar diagramas UML.
- 2-Generar código C++, CORBA, Java, esquemas XML a partir de los diagramas.
- 3-Generar documentación.

4-Realizar ingeniería inversa - código a modelo, código a diagrama.

5-Generar bases de datos.

Además se decide escoger para la solución como herramienta de modelado al Visual Paradigm, porque es multiplataforma con soporte para el sistema operativo GNU/Linux.

1.8 Entorno de desarrollo (IDE).

Como IDE se escogió el NetBeans porque posee todas las herramientas necesarias para crear aplicaciones de escritorio, de empresa, web y aplicaciones móviles, posee soporte para lenguajes como Java, C/C++, e incluso lenguajes dinámicos como PHP, Java Script, Groovy, y Ruby. Es fácil de instalar y utilizar, además es multiplataforma.

1.9 Protocolo de comunicación.

Se determinó escoger el protocolo SSH (Secure Shell) porque 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 (19).

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 (19).

1.10 Sistema Gestor de Base de Datos (SGBD).

Una base de datos (BD) 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 (20).

Como SGBD se decidió utilizar Sqlite3 por las siguientes razones:

1.10.1 SQLite3.

SQLite3 es un sistema gestor de base de datos relacional, que a diferencia de los SGBD cliente-servidor, no establece un proceso independiente con el que se comunica con el programa principal, si no que su biblioteca se enlaza con el programa pasando a ser parte integral del mismo (21).

Entre sus principales características se encuentran:

- 1-Varios accesos de lectura pueden ser servidos en paralelo (21).
- 2-No posee configuración, esto significa que de la forma en que fue creado y diseñado SQLite3 no necesita ser instalado (21).
- 3-Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente (21).
- 4-SQLite3 posee una pequeña memoria y una única biblioteca es necesaria para acceder a la bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas (21).

5-SQLite3 realiza operaciones de manera eficiente y es más rápido que *MySQL* y *PostgreSQL* para bases de datos pequeñas como es en el caso del sistema que se desea desarrollar.

6-Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración (21).

7-SQLite3 es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente (21).

1.11 Conclusiones Parciales.

Este capítulo abordó y fundamentó los conceptos más importantes de la presente investigación, los cuales son los siguientes: seguridad, administración remota y dispositivos USB. Se realizó un estudio de varias herramientas de administración de dispositivos USB, que sirvió de base para determinar las funcionalidades del sistema a desarrollar. Además se escogió como metodología OpenUP. Se determinó a utilizar, como lenguaje de modelado UML, Netbeans como IDE de programación, C y C++ como lenguaje de programación para el lado del cliente y del servidor respectivamente y SQLite3 como SGBD, todo con el fin de relacionar todos los conceptos para lograr un buen entendimiento de la posterior solución.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.

Introducción

En este capítulo se detallan las principales características del sistema a desarrollar, definiendo el modelo de dominio según el objeto de estudio de la presente investigación. También se definen los requisitos funcionales y no funcionales con los que debe cumplir el sistema, así como la arquitectura. Se explican los patrones arquitectónicos y del diseño utilizados y el diagrama de clases del diseño.

2.1 Modelo de dominio.

Un modelo de dominio es una representación visual de clases conceptuales o de objetos del mundo real, no de componentes *software*, consiste en un conjunto de diagramas sin definición de operaciones que brinda los conceptos más significativos para el dominio del problema. Además de conceptos, pueden mostrar los atributos de los conceptos y las asociaciones entre los mismos (2).

En el caso del sistema que se desea desarrollar no cuenta con un negocio real y no existe ningún proceso organizado, manual, ni automatizado que tributen a los objetivos de este trabajo de diploma por lo que se decidió utilizar este modelo.

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

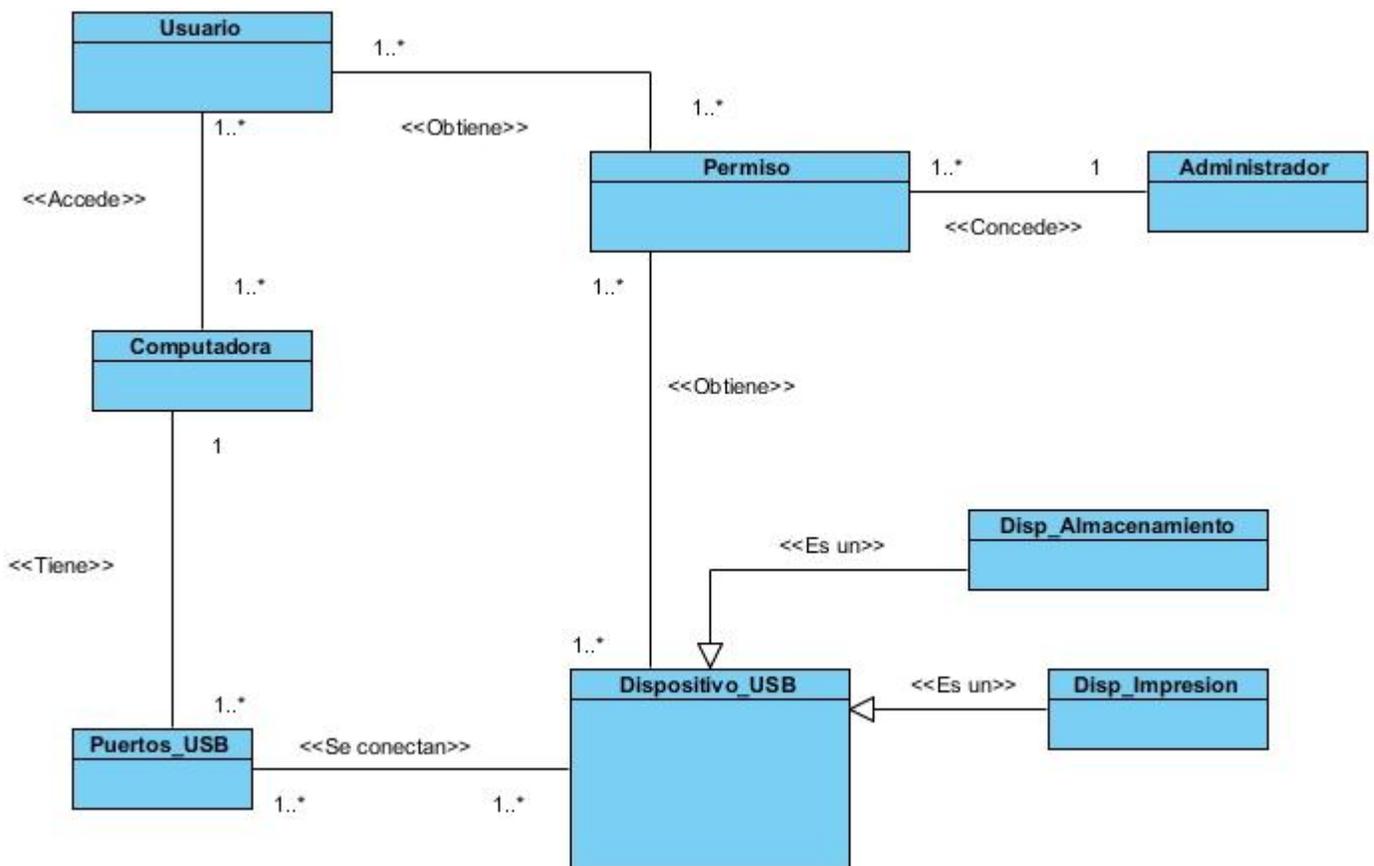


Figura 1: Modelo de dominio.

2.1.1. Descripción de las clases del dominio.

Usuario: Usuario del dominio que accede a una computadora para utilizar un dispositivo USB bien sea de almacenamiento o de tipo impresor.

Computadora: Estación de trabajo donde el usuario conecta el dispositivo.

Permisos: Conjunto de reglas establecidas que le son otorgadas a los usuarios de un dominio y a los dispositivos USB con el fin de autorizar el montaje de estos y que se puedan utilizar.

Puerto_USB: permite conectar a la computadora diversos dispositivos USB.

Dispositivo_USB: Conjunto de dispositivos que se conectan al puerto USB.

Disp_Almacenamiento: Permite el almacenamiento de información.

Disp_Impresión: Permite imprimir un documento digital.

Administrador: es el encargado de establecer el control de los dispositivos USB que se conecten a la computadora mediante la asignación de permisos a los usuarios del dominio así como a los dispositivos USB para utilizarlos.

2.2. Requisitos del sistema.

2.2.1 Requisitos Funcionales.

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

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

RF5. Reconocer conexión de un dispositivo: Permite reconocer cuando se conecta un dispositivo.

RF6. Obtener información (atributos) del dispositivo conectado: Permite una vez conectado el dispositivo obtener la información de este.

RF7. Reconocer que el usuario pertenece a un dominio: Permite reconocer que el usuario pertenece a un dominio.

RF8. Reconocer tipo de dispositivo: Permite reconocer el tipo que tipo de dispositivo es.

RF9. Asignar Permisos: Permite asignar diferentes permisos a los usuarios y a los dispositivos.

RF10. Dar acceso al dispositivo: Permite dar o denegar acceso al dispositivo.

2.2.2 Requisitos No Funcionales.

Los requisitos 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.

Restricciones del diseño y la implementación

1-El sistema debe ser implementado en los lenguajes de programación C y C++.

2-Como sistema gestor de bases de datos se debe usar SQLite3.

3-La interfaz gráfica debe ser implementada utilizando la biblioteca GTK+.

Facilidad de uso

4-El sistema debe contar con una interfaz amigable para que le sea fácil al administrador controlar de forma sencilla a los dispositivos USB.

5-La herramienta debe funcionar correctamente para cualquier usuario del dominio que desee conectar un dispositivo en una computadora.

Hardware

6-Las computadoras donde se vayan a desplegar el sistema deben poseer red.

Soporte

7-El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

Seguridad

8-La comunicación entre el cliente y el servidor debe realizarse mediante el protocolo SSH.

2.3 Diagrama de Casos de Usos del sistema.

Un caso de uso (CU) del sistema no es más que fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Todo sistema tiene como mínimo un diagrama de CU, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (CU). Un diagrama de CU muestra, los distintos requisitos funcionales que se esperan de una aplicación o sistema y como se relaciona con su entorno (usuarios u otras aplicaciones) (22).

A continuación se representa en la figura 2 los casos de usos del sistema a ser desarrollado.

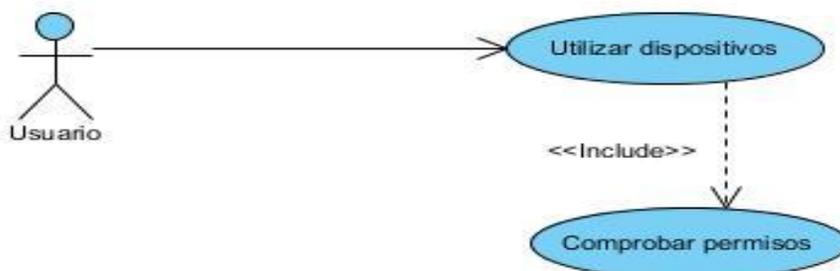
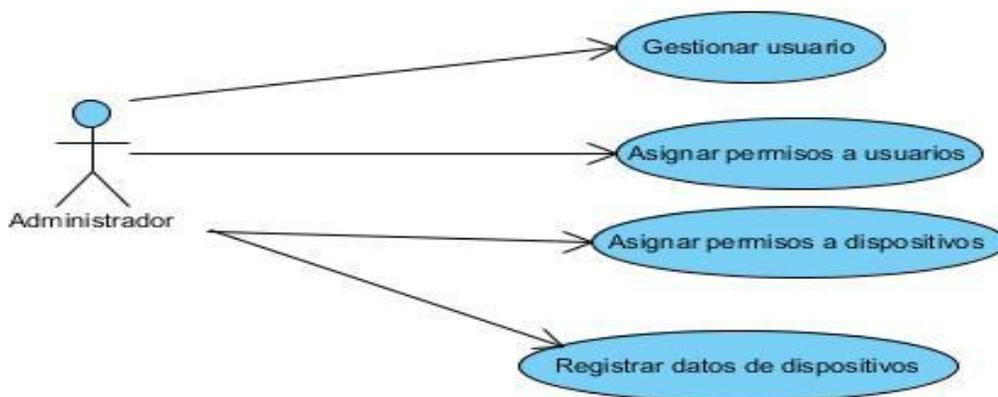


Figura 2: Diagrama de CU del sistema.

2.3.1 Descripción de actores.

Un actor es un agente externo que interactúa con el sistema en pos de obtener un resultado esperado (22).

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

Actor	Descripción
Administrador	Persona encargada de la gestión de usuarios. Además se encarga de la asignación de permisos a los diferentes usuarios del dominio.
Usuario	Es el encargado de conectar el dispositivo para ser utilizado.

Tabla 1: Descripción de actores

2.3.2 Descripción de los CU del sistema.

CU1. Gestionar usuario: Este CU tiene como objetivo que el administrador pueda insertar, eliminar, modificar y mostrar en la base de datos los datos de un usuario del dominio.

CU2. Asignar permisos a usuarios: Este CU tiene como objetivo que el administrador pueda asignar permisos a un determinado usuario del dominio en correspondencia a un dispositivo.

CU3. Asignar permisos a dispositivos: Este CU tiene como objetivo que el administrador pueda asignar permisos por defecto a un determinado dispositivo.

CU4. Registrar datos de dispositivo: Este CU tiene como objetivo que el administrador pueda registrar los datos de un dispositivo que le haya sido notificado.

CU5. Utilizar dispositivos: Este CU tiene como objetivo que el sistema en dependencia de los permisos pueda dar acceso o no al dispositivo.

2.3.3 Especificación de los CU del sistema.

A continuación se reconocen y se detallan los CU más significativos a partir de los requisitos funcionales anteriormente especificados.

Se especifican los CU Gestionar usuario, Asignar permisos a usuarios, Utilizar dispositivos, el resto se detalla en la sección de los **Anexos**.

CU1. Gestionar usuarios.

Caso de uso:	Gestionar usuarios
Actores:	Administrador
Propósito:	Este CU se lleva a cabo con el objetivo de que el administrador pueda insertar, eliminar, modificar y mostrar un los datos de un usuario.
Resumen	El CU 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.
Precondiciones:	El administrador tiene que estar autenticado en el sistema.
Referencias:	RF1, RF2, RF3, RF4
Prioridad:	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. El CU se inicia cuando el administrador selecciona una de las opciones siguientes:</p> <ul style="list-style-type: none"> - Insertar usuario. - Eliminar usuario. - Modificar usuario. - Mostrar usuario. 	<p>2. El sistema realiza la operación según la opción seleccionada por el administrador.</p> <ul style="list-style-type: none"> -Si selecciona "Insertar usuario", ver sección "Insertar usuario". -Si selecciona "Eliminar usuario", ver sección "Eliminar usuario". -Si selecciona "Modificar usuario", ver sección "Modificar usuario". -Si selecciona "Mostrar usuario", ver sección "Mostrar usuario".

3. El CU termina cuando el sistema procesa la información según la acción realizada por el administrador.

Sección "Insertar usuario"

Acción del actor	Respuesta del sistema
1. El administrador introduce los datos del usuario: -nombre -dominio	2. El sistema verifica que los datos estén correctos y que no exista el usuario en la BD.
	3. El sistema procesa la información a partir de la acción realizada por el administrador y almacena los datos del usuario.

Flujos alternos

Acción del actor	Respuesta del sistema
	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".

Sección "Eliminar usuario"

Acción del actor	Respuesta del sistema
1. El sistema muestra un listado con los usuarios del dominio registrados en la base datos 2. El administrador escoge el nombre del usuario del dominio que desea eliminar.	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.

Flujos alternos	
Acción del actor	Respuesta del sistema
	2. Si no hay usuarios en la BD, el sistema muestra un mensaje de error.
	5. El sistema cancela la acción de eliminar usuario.
Sección “Modificar usuario”	
Acción del actor	Respuesta del sistema
1. El sistema muestra un listado con los usuarios registrados en la base datos 2. El administrador escoge el nombre del usuario que desea modificar.	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.
Flujos alternos	
Acción del actor	Respuesta del sistema
	5. Si los datos del usuario están incorrectos el sistema muestra un mensaje.
Sección “Mostrar usuario”	
Acción del actor	Respuesta del sistema
1. El administrador escoge 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.
Flujos alternos	
Acción del actor	Respuesta del sistema
	1. Si el usuario no existe en la BD el sistema muestra

un mensaje de error.

Tabla 2: Especificación CU Gestionar Usuario.

CU2. Asignar permisos a usuarios.

Caso de uso:	Asignar permisos a usuarios.
Actores:	Administrador
Propósito:	Este CU se lleva a cabo con el objetivo de que el administrador pueda asignar un permiso a un usuario del dominio sobre un determinado dispositivo que le haya sido notificado.
Resumen	El CU se inicia cuando el administrador asigna el permiso al usuario del dominio sobre el dispositivo y termina cuando el sistema realiza la acción.
Precondiciones:	El administrador tiene que estar autenticado en el sistema.
Referencias:	RF9
Prioridad:	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El CU se inicia cuando el administrador selecciona la opción "Asignar permiso a usuario".	2. El sistema muestra la lista de permisos a asignar, los usuarios y los dispositivos que están en la BD.
3. El administrador escoge el nombre del usuario del dominio, el identificador del dispositivo y el identificador del permiso que desea asignar a dicho usuario sobre el dispositivo escogido.	3. El CU termina cuando el sistema procesa la información según la acción realizada por el administrador y asigna los permisos al usuario del dominio sobre el dispositivo.

Tabla 3: Especificación CU Asignar permisos.

CU4. Utilizar dispositivos.

Caso de uso:	Utilizar dispositivos	
Actores:	Usuario	
Propósito:	Este CU se lleva a cabo con el objetivo de que el sistema pueda darle acceso a un dispositivo para que pueda ser utilizado por un usuario del dominio.	
Resumen	El caso de uso se inicia cuando un usuario del dominio conecta un dispositivo y termina cuando el sistema le da o no acceso en dependencia de los permisos que tenga el usuario sobre el dispositivo o los permisos que tenga el dispositivo por defecto.	
Precondiciones:	El usuario debe haber conectado un dispositivo en la computadora.	
Referencias:	RF5, RF6, RF7, RF8	
Prioridad:	Crítico	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El CU se inicia cuando el usuario conecta un dispositivo a la computadora.	2. El sistema obtiene todos los atributos del dispositivo. 3. El sistema identifica el tipo de dispositivo que es. 4. El sistema identifica si el usuario pertenece al dominio. 5. El sistema descarga la base de datos del servidor. 6. Ir al sección Comprobar permisos 8. El sistema le da acceso al dispositivo.	
Sección "Comprobar Permisos"		
Acción del actor	Respuesta del sistema	
	7. El sistema verifica: si el usuario pertenece al	

	dominio y tiene los permisos pertinentes para acceder al dispositivo.
Flujos alternos	
Acción del actor	Respuesta del sistema
	7. Si el usuario no pertenece al dominio, o el usuario o el dispositivo no poseen permisos no se puede utilizar el dispositivo.

Tabla 4: Especificación CU Utilizar dispositivos.

La descripción del caso de uso anterior se detalla mejor mediante un diagrama de procesos contenido en la sección de los anexos (Ver anexo 3).

2.4 Patrones de arquitectura.

Los patrones arquitectónicos, o patrones de arquitectura son los que definen la estructura de un sistema de *software*, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño (23).

Es decir ofrecen soluciones a problemas de arquitectura de *software*, brindando una descripción de los elementos más importantes 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 (23).

2.4.1 Arquitectura en capas.

Lo que se conoce como arquitectura en capas es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio, mecanismos de almacenamiento. La necesidad de contar con porciones de la aplicación que se puedan intercambiar sin tener que modificar el resto de la aplicación es lo que impulsa el desarrollo en capas (24).

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 (2).

Ventajas del modelo:

- 1-Facilita la migración. El acoplamiento con el entorno está localizado en las capas inferiores. Estas son las únicas a reimplementar en caso de transporte a un entorno diferente (25).
- 2-Cada nivel implementa unas interfaces claras y lógicas, lo que facilita la sustitución de una implementación por otra (25).
- 3-Permite trabajar en varios niveles de abstracción. Para implementar los niveles superiores sin necesidad de conocer el entorno subyacente, solo las interfaces que proporcionan los niveles inferiores (25).
- 4-Desarrollos paralelos en cada capa (25).
- 5-Aplicaciones más robustas debido al encapsulamiento (25).
- 6-Mantenimiento y soporte más sencillo, es más sencillo cambiar un componente que modificar una aplicación monolítica (25).

7-Mayor flexibilidad, se puede añadir nuevos módulos para dotar al sistema de nuevas funcionalidades (25).

Por las razones anteriormente planteadas se decidió utilizar esta arquitectura para la aplicación del lado del servidor.

2.4.2 Arquitectura Cliente-Servidor.

El modelo cliente-servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes, que es el único elemento de acoplamiento entre ambos (26).

Los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas cliente-servidor son: el proceso cliente que es quien inicia el diálogo, el proceso servidor que pasivamente espera a que lleguen peticiones de servicio y el middleware⁴ que corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes (26).

2.5 Arquitectura del sistema.

Para el sistema en general se escogió la arquitectura cliente- servidor donde la aplicación cliente va a ser la encargada de descargar la base de datos del servidor, para una vez conectado un dispositivo comprobar los permisos que posee el usuario del dominio con respecto a este dispositivo y pueda utilizarlo. Mientras la aplicación del lado del servidor va ser la encargada de actualizar la base de datos.

4 Es un *software* de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

A continuación se muestra la arquitectura que tendrá el sistema.

Arquitectura cliente-servidor



Figura 3: Arquitectura del sistema.

2.5.1 Arquitectura del Servidor.

A continuación la figura 4 muestra la arquitectura 3 capas escogida para el servidor.



Figura 4: Arquitectura del servidor.

2.5.2 Descripción de los componentes de la arquitectura del servidor.

Acceso a datos: Permite la comunicación y la extracción de los datos almacenados en la base de datos, contiene los datos de los usuarios, de los permisos y de los dispositivos.

Servicios: Esta capa sirve de nexo entre las capas Acceso a datos y la capa Presentación.

Presentación: Muestra al administrador la interfaz gráfica de la aplicación.

BD: Se encuentran almacenados todos los datos.

2.6 Diagrama de paquetes.

El servidor está compuesto por 3 capas, la capa de Acceso a datos, la de Servicios y la de Presentación como se hace referencia en la arquitectura del servidor.

A continuación se muestra un diagrama de paquetes que van a representar las capas mencionadas anteriormente como subpaquetes.

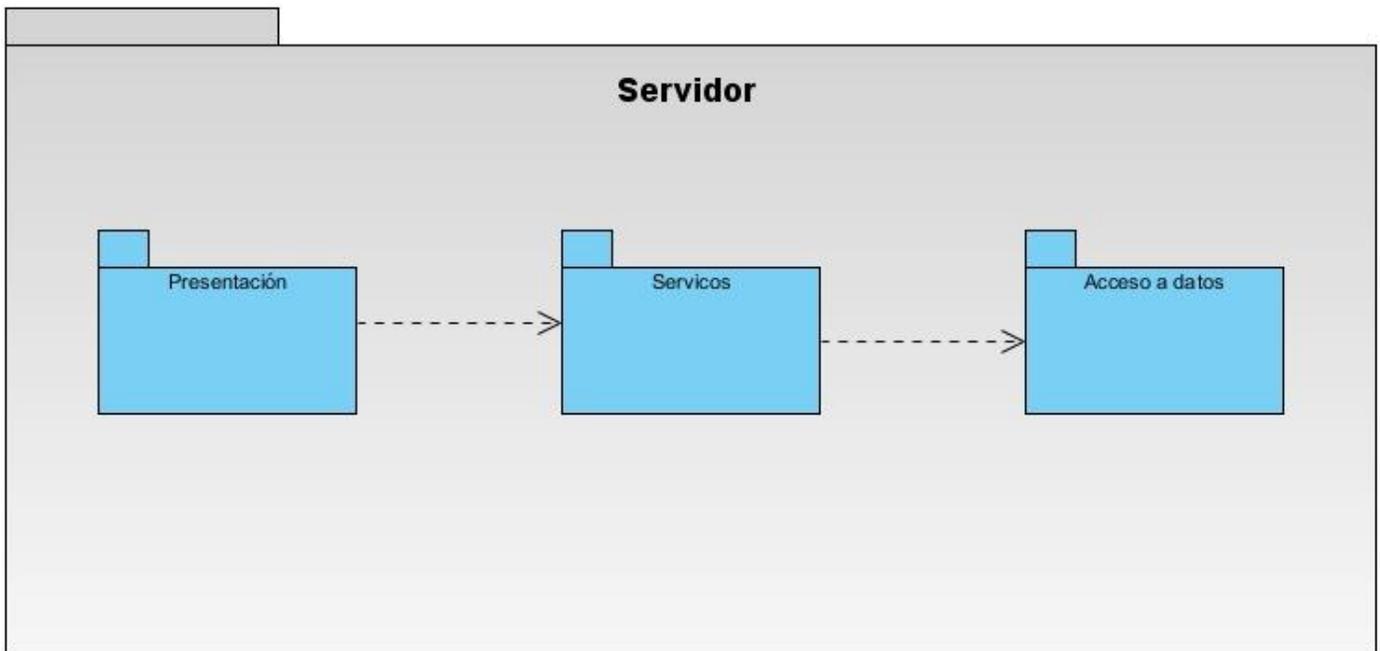


Figura 5: Diagrama de paquetes.

2.6.1 Descripción de los paquetes.

Paquete Presentación: Esta capa posee las interfaces de usuario de la aplicación. En este caso las interfaces de usuario serán implementadas por la biblioteca GTK+. En la sección de los anexos se encuentran todos los prototipos de interfaces.

Paquete Servicios: Esta capa contiene la clase **ControladoraPrincipal** que es la que controla todo el flujo de datos de la aplicación también posee las clases: **ControladorGestionarUsuario**,

ControladoraDispositivo, ControladoraPermiso que son las encargada de realizar todas las operaciones con las clases de acceso a datos para ya sea insertar, eliminar, consultar o modificar algún dato en la base de datos.

Paquete Acceso a datos: Esta capa controla todo el acceso a la base de datos del sistema posee las clases: **Usuario** contiene los datos de un usuario, **Permiso** contiene los datos de un permiso, **Dispositivo** que contiene los datos de un dispositivo, estas incluyen a la clase **Conexión** que es la que permite la conexión a la base de datos.

2.7 Diagrama de clases del diseño.

Los diagramas de clases se utilizan para representar: la estructura estática del sistema modelado, las clases y objetos del sistema y su estructura interna así como las relaciones que existen entre las distintas clases y objetos del sistema. Se concentran en los elementos del sistema de forma independiente del tiempo. Estos diagramas sirven a su vez para realizar la abstracción de un dominio y formalizar el análisis de los conceptos relacionados al mismo (Modelo de dominio), también sirve para definir documentar una solución de diseño, es decir, la estructura del sistema que se va a implementar en términos de clases y objetos (27).

2.7.1 Diagrama de clases del diseño del Servidor.

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

Capa Presentación

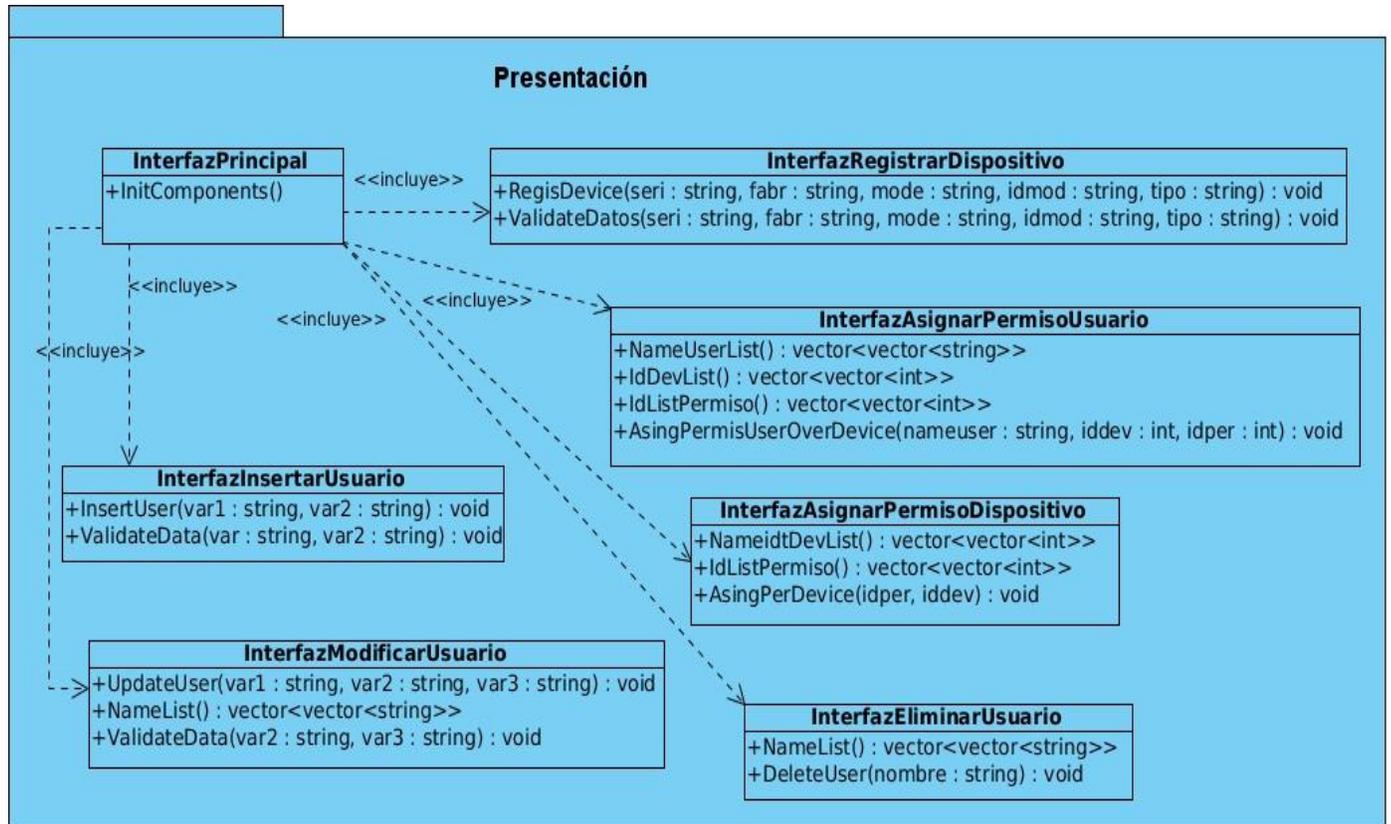


Figura 6. Diagrama de clases del diseño capa Presentación.

Capa Servicios.

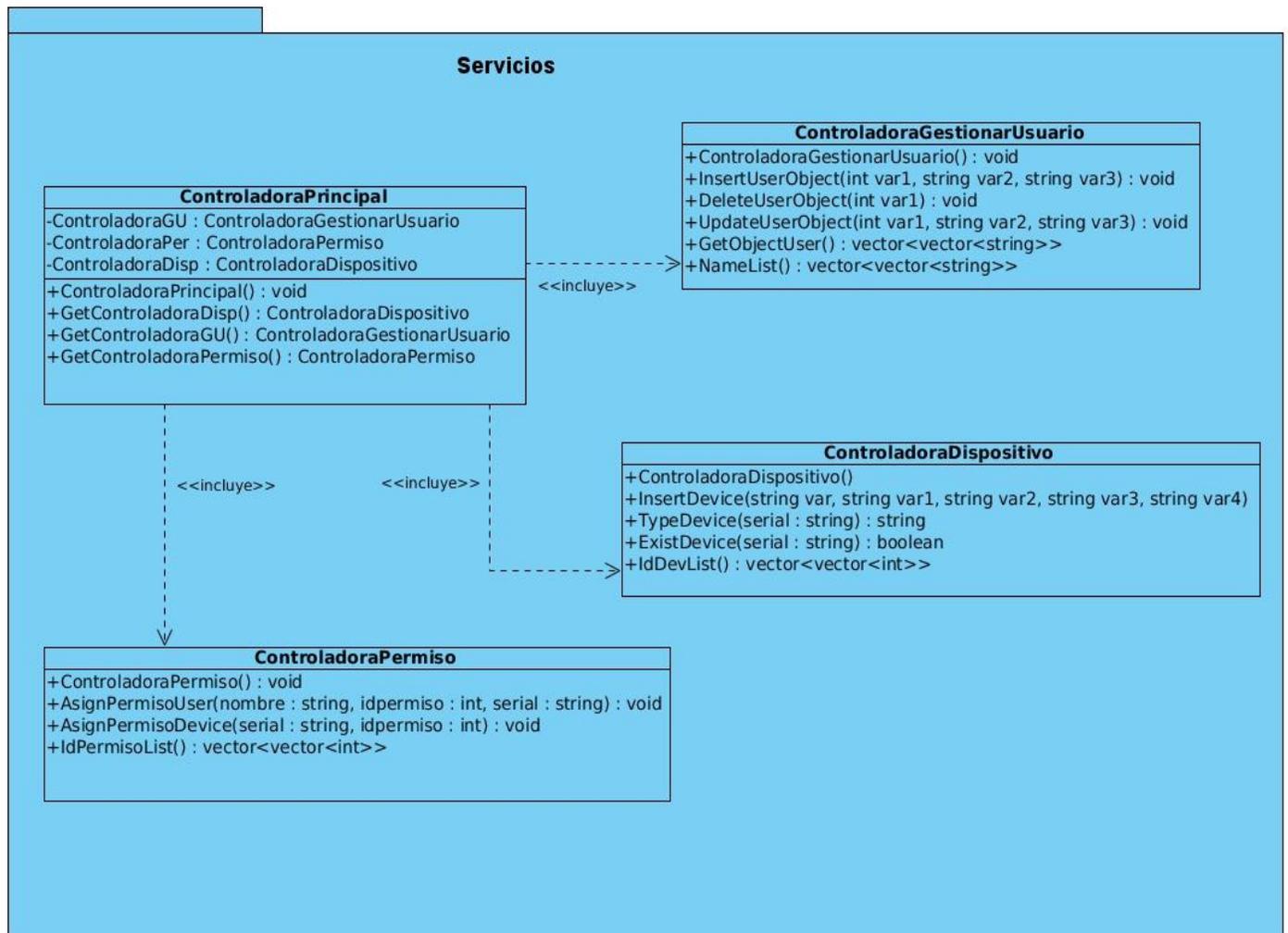


Figura 7. Diagrama de clases del diseño capa Servicios.

Capa Acceso a datos.

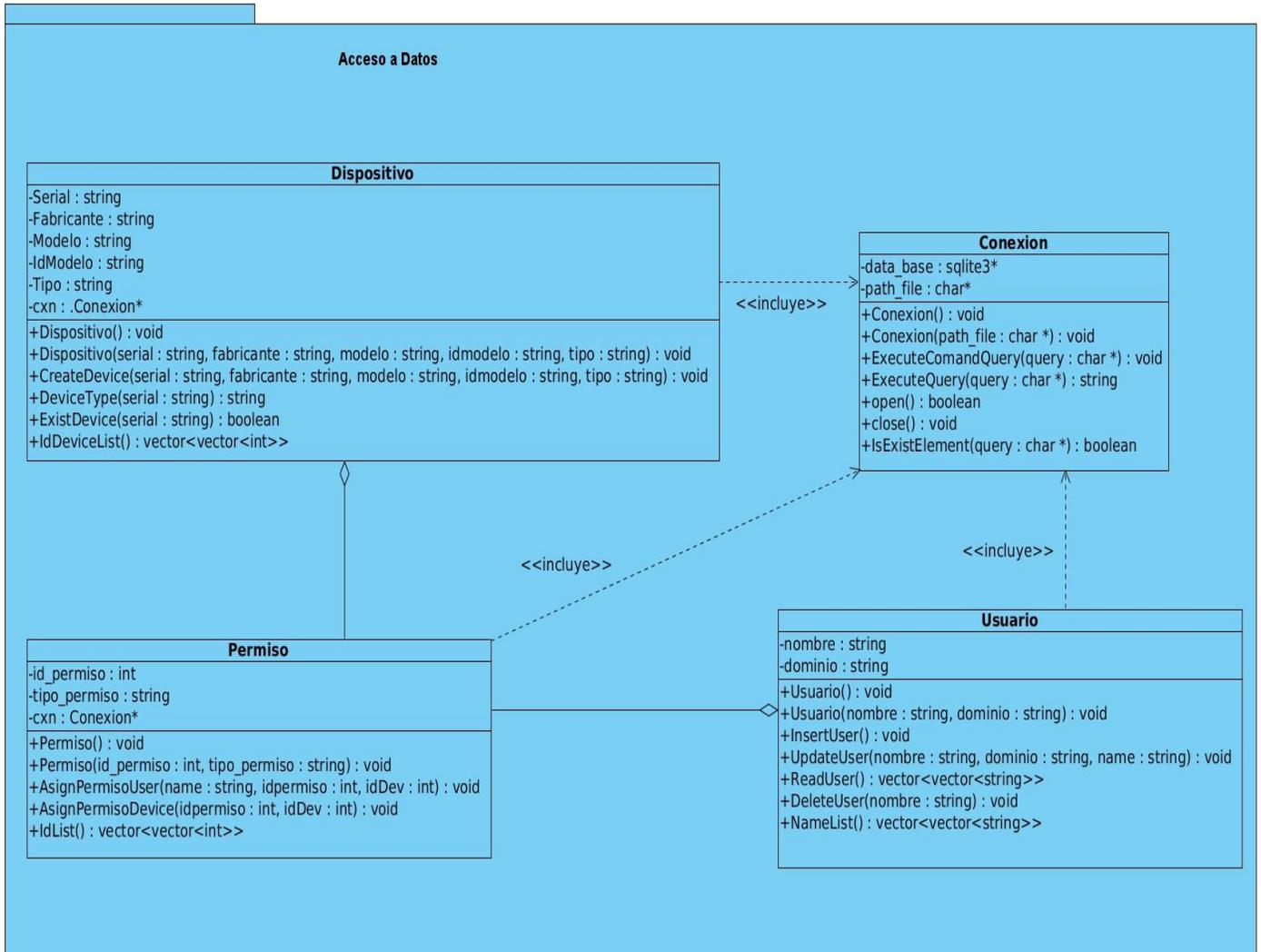


Figura 8. Diagrama de clase del diseño capa Acceso a datos.

2.7.2 Descripción de las clases del sistema.

A continuación se describen las clases: **Conexión**, **Usuario**, **ControladoraPrincipal**, las demás se describen en la sección de los anexos.

Nombre: Conexion	
Tipo de clase: Pública	
Atributo	Tipo
<i>data_base</i>	<i>sqlite3*</i>
<i>path_file</i>	<i>char*</i>
Funciones	
<i>ExecuteComandQuery</i> : Función para ejecutar una consulta sin resultado.	
<i>ExecuteQuery</i> : Función para ejecutar una consulta con resultado.	
<i>open</i> : Función para abrir la conexión a la base de datos.	
<i>close</i> : Función para cerrar la conexión a la base de datos.	
Descripción	Esta clase establece la conexión a la base de datos y ejecuta las consultas

Tabla 6: Descripción de la clase Conexión.

Nombre: Usuario	
Tipo de Clase: Pública	
Atributo	Tipo
Nombre	<i>string</i>

Dominio	<i>string</i>
cxn	<i>Conexion *</i>
Funciones	
<i>InsertUser</i> : Función para registrar los datos de un usuario en la base de datos.	
<i>UpdateUser</i> : Función para actualizar los datos de un usuario registrado en la base de datos.	
<i>ReadUser</i> : Función para mostrar los datos de los usuarios registrados en la base de datos.	
<i>DeleteUser</i> : Función para eliminar los datos de un usuario registrado en la base de datos.	
<i>NameList</i> : Función para mostrar los nombres de los usuarios registrados en la base de datos.	
Descripción	Esta clase maneja toda la información con respecto a un usuario

Tabla 10: Descripción de la clase Usuario.

Nombre: ControladoraPrincipal	
Tipo de Clase: ControladoraPrincipal	
Atributo	Tipo
ControladoraGU	<i>ControladoraGestionarUsuario</i>
ControladoraPer	<i>ControladoraPermiso</i>
ControladoraDisp	<i>ControladoraDispositivo</i>
Funciones	
<i>GetControladoraDisp</i> : Este método devuelve una instancia de la clase <i>ControladoraDispositivo</i>	
<i>GetControladoraGU</i> : Este método devuelve una instancia de la clase <i>ControladoraGestionarUsuario</i> .	
<i>GetControladoraPermiso</i> : Este método devuelve una instancia de la clase <i>ControladoraPermiso</i>	

Descripción	Esta clase es la encargada de manejar todo el flujo de información en todo el sistema.
--------------------	--

Tabla 12: Descripción de la clase ControladoraPrincipal

2.7.3 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.

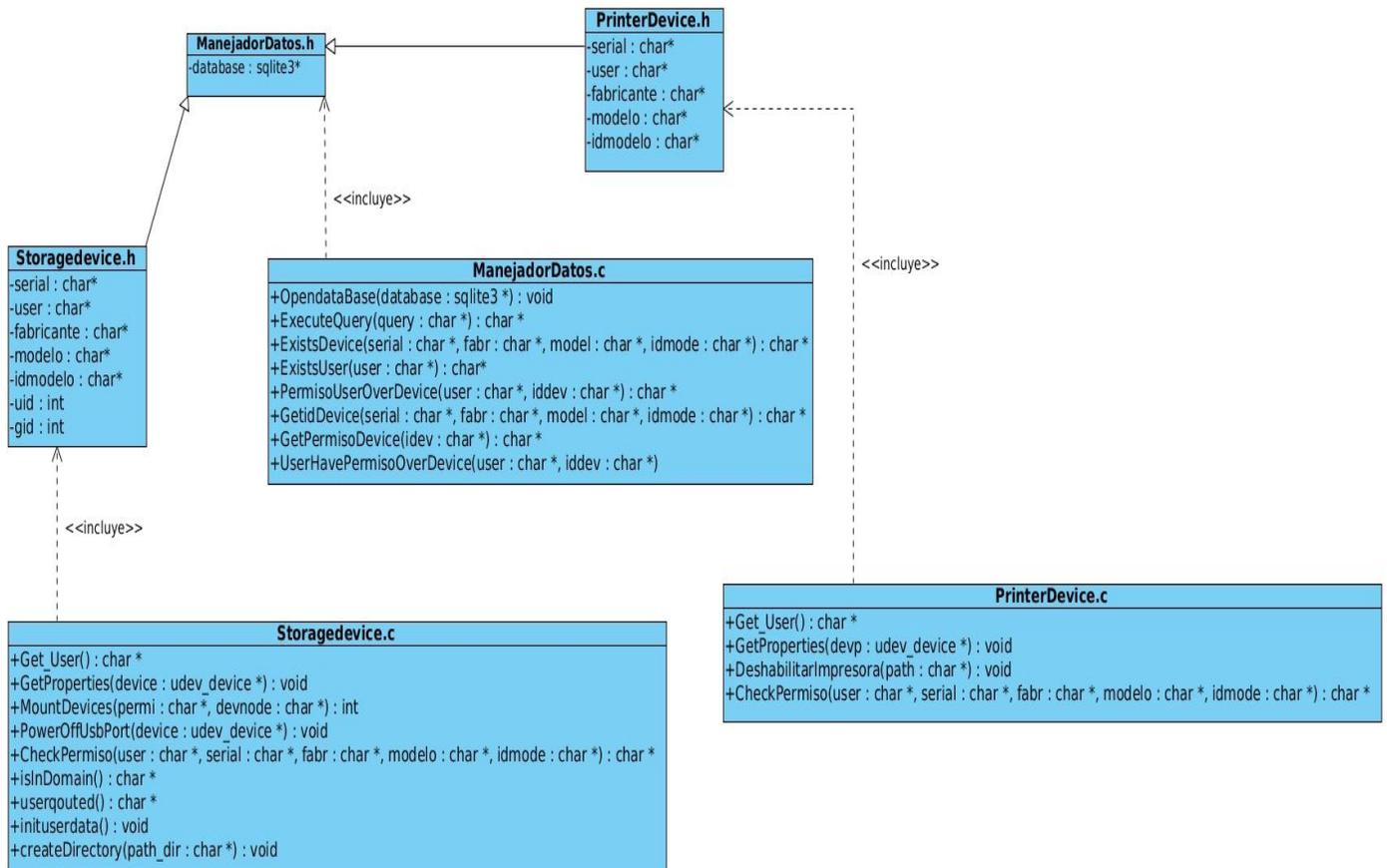


Figura 9. Diagrama de estructuras de datos para el cliente.

2.7.4 Patrones de diseño.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*. Es decir, brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un

patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (28).

Uno de los patrones de diseño que se utilizó para la solución fue el patrón experto que indica que la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento) (28).

Un ejemplo donde se utilizó este patrón fue en la clase **Usuario**, para saber todos los nombres de los usuarios registrados a la base de datos es necesario implementarlo en esta, pues esta clase es la que posee todos los métodos referentes a la gestión de un usuario.

Otro de los patrones que se utilizó fue el patrón controlador que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado (29).

Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (29).

Un ejemplo donde se utilizó este patrón fue en la clase **ControladoraGestionarUsuario** pues esta clase sirve como intermediario entre la interfaz visual **InterfazInsertarUsuario** y la clase acceso a datos **Usuario**.

2.8 Diseño de la base de datos.

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.

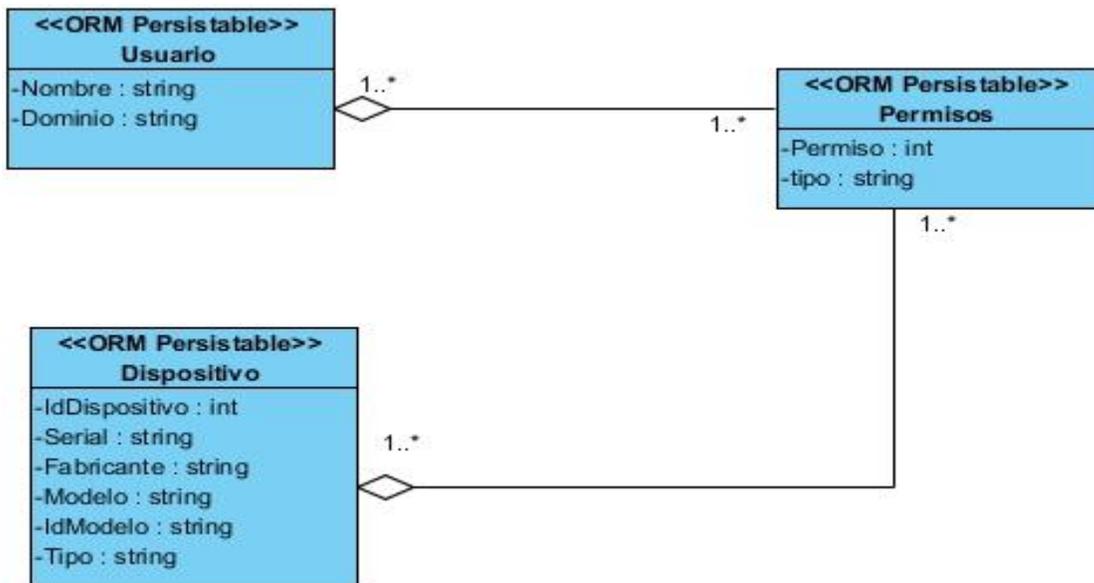


Figura 10. Diagrama de clases persistentes.

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.

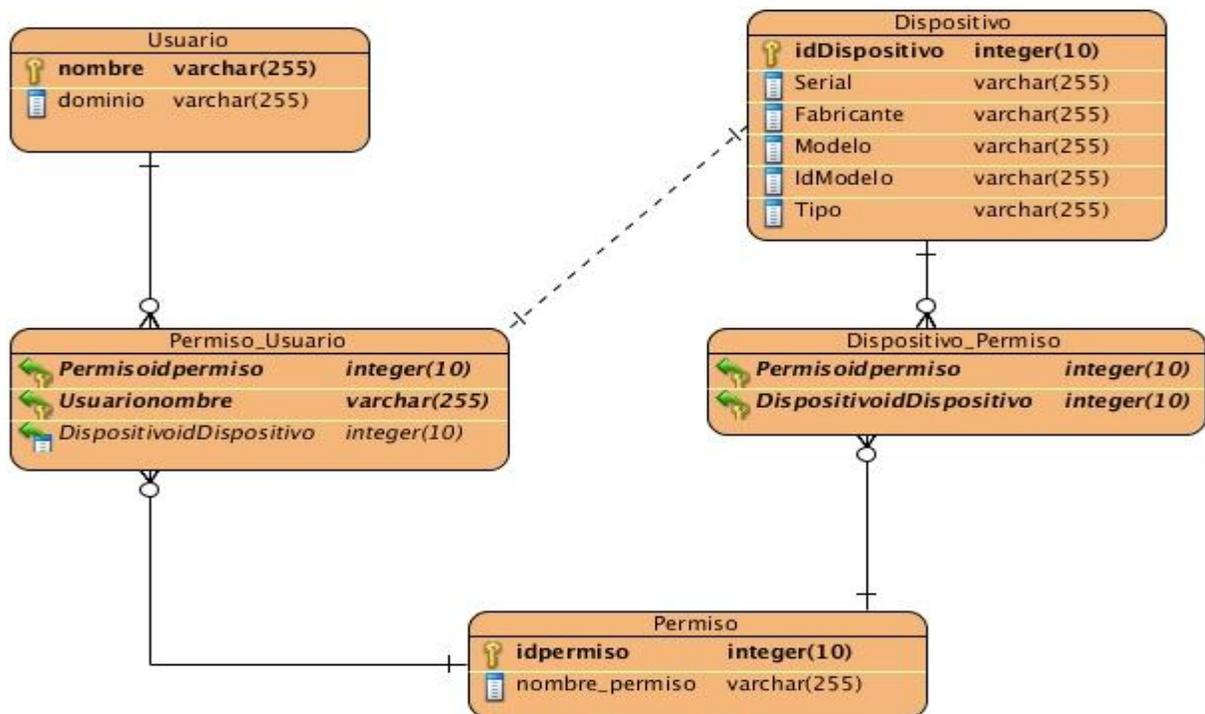


Figura 11. Diagrama entidad-relación.

2.8.2.1 Descripción de las tablas.

Nombre: Usuario

Descripción: Tabla para almacenar los datos de los usuarios

Atributo	Tipo	Descripción
----------	------	-------------

nombre	<i>varchar</i>	Nombre del usuario.
dominio	<i>varchar</i>	Dominio al que pertenece el usuario.

Tabla 21: Entidad Usuario.

Nombre: Dispositivo		
Descripción: Tabla para almacenar los datos de los dispositivos.		
Atributo	Tipo	Descripción
idDispositivo	<i>integer</i>	Identificador del dispositivo.
Serial	<i>varchar</i>	Serial que identifica al dispositivo.
Fabricante	<i>varchar</i>	Fabricante del dispositivo.
Modelo	<i>varchar</i>	Modelo del dispositivo.
IdModelo	<i>varchar</i>	Identificador del modelo del dispositivo.
Tipo	<i>varchar</i>	Tipo de dispositivo.

Tabla 23: Entidad Dispositivo.

Nombre: Permiso		
Descripción: Tabla para almacenar los datos de los permisos.		
Atributo	Tipo	Descripción
idpermiso	<i>integer</i>	Identificador del permiso.
nombre_permiso	<i>varchar</i>	Nombre del permiso.

Tabla 24: Entidad Permiso.

2.9 Conclusiones parciales.

Este capítulo abordó las principales características del sistema a desarrollar, se definió los principales requerimientos con los que va a contar el sistema, así como la arquitectura cliente-servidor para el sistema en general y 3 capas para la aplicación del lado del servidor. Se muestra también el diagrama de casos de

uso del sistema, el diagrama de clases persistentes, así como el modelo de dominio y sus entidades, el modelo de diseño entre otros. Además se escogieron los patrones de diseño a utilizar que fueron los patrones de asignación de responsabilidades experto y controlador. 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 PRUEBAS.

Introducción.

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

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 (2).

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 interfaces (30).

A continuación se representa el diagrama de componentes del servidor y del cliente respectivamente.

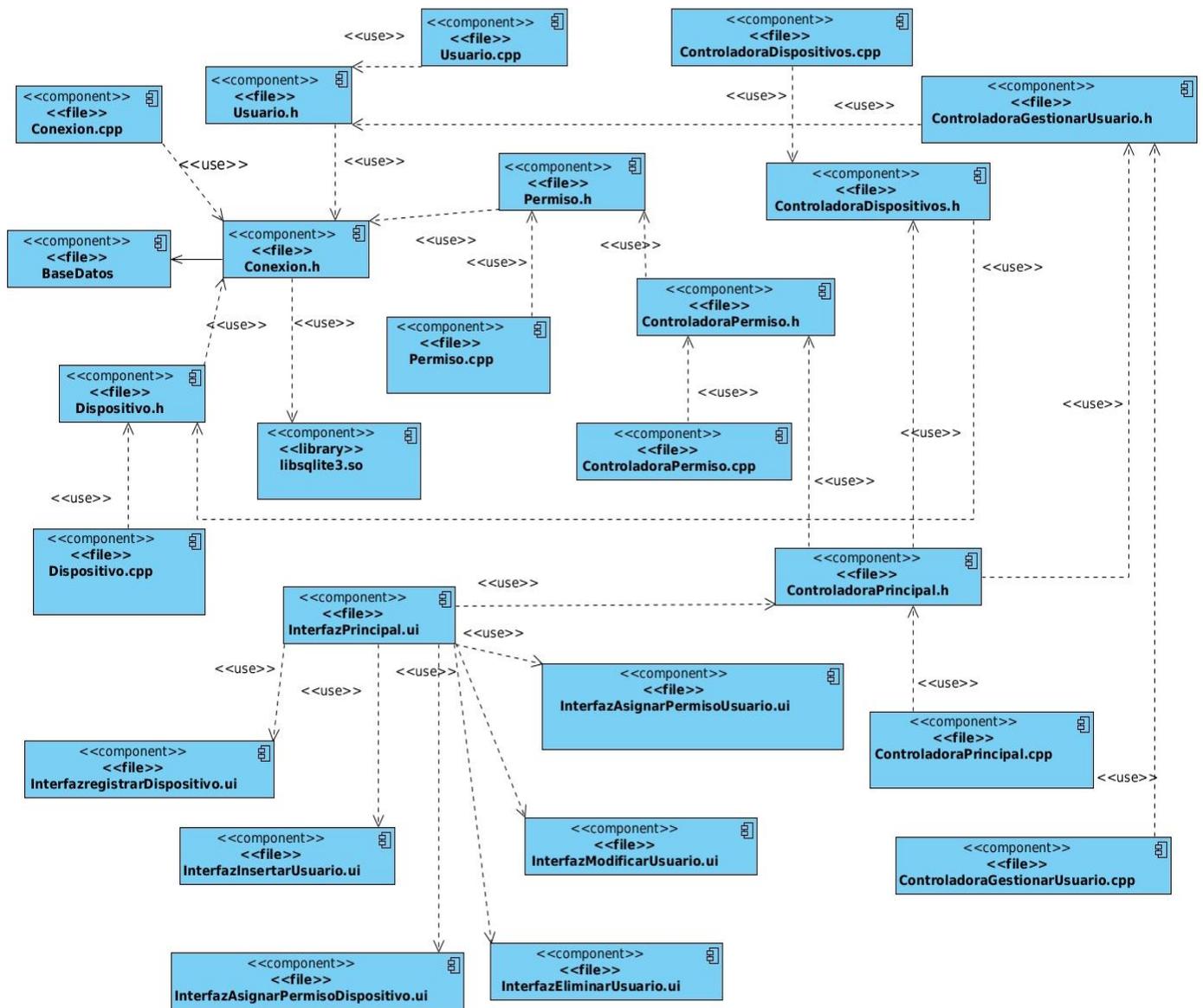


Figura 12. Diagrama de componentes del servidor.

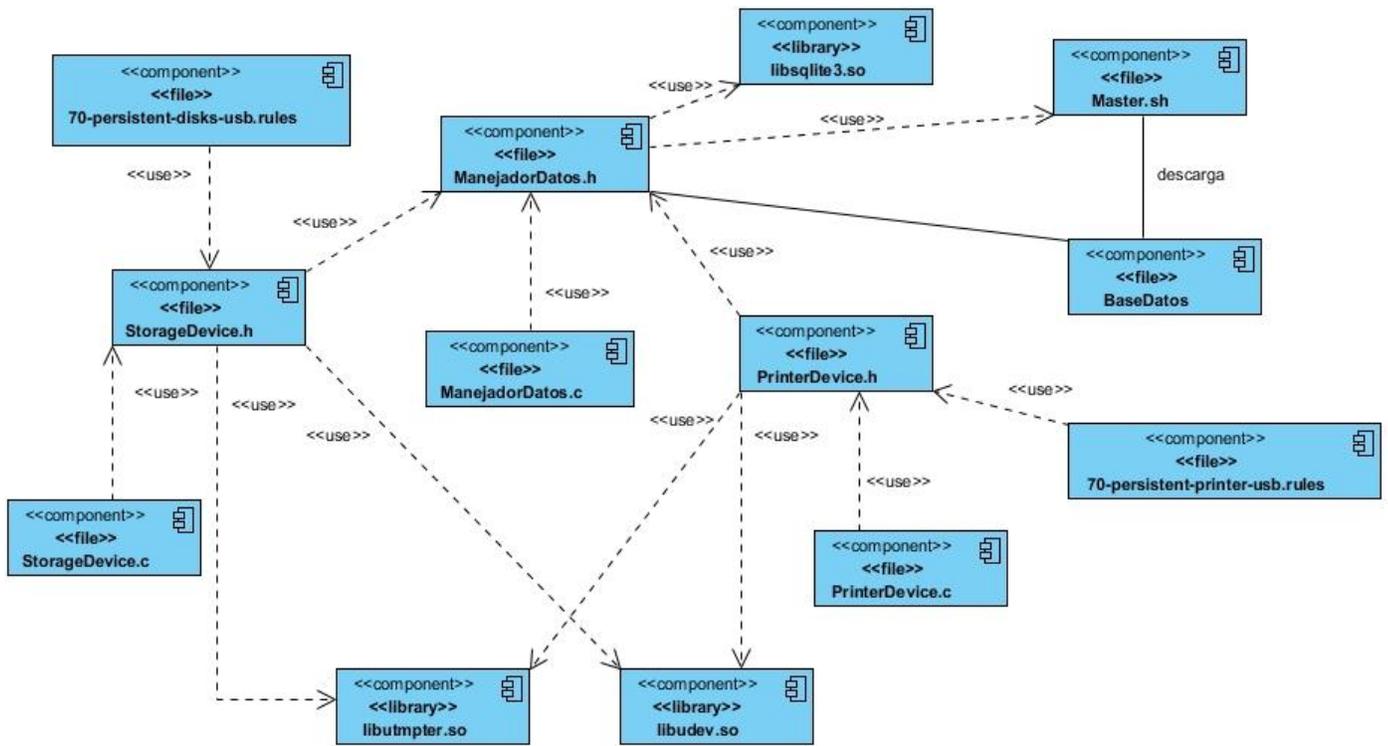


Figura 13. Diagrama de componentes del cliente

3.2 Estándares de codificación.

Los estándares de codificación son un conjunto de reglas o pautas que están enfocados a la estructura de la lógica del código, para facilitar su lectura, comprensión y mantenimiento para que cualquier persona que se desempeñe como codificador de dicho lenguaje, pueda interpretar de manera eficiente la escritura del código (31).

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (32).

A continuación se presenta estas reglas para la programación en los lenguajes C/C++

Identificador de variables

- Cuando las variables están compuestas por más de una palabra la primera 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.

Identificador de funciones

- La primera letra deberá ser mayúscula

Ejemplo

- *void* Function();
- *string* GetResult();

Identificadores de tipos definidos por el usuario

- La primera letra será mayúscula.

Ejemplo

- *class* Clase.
- *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 crítico. En este proceso el probador que es el responsable de llevar a cabo las pruebas, somete al producto a una serie de acciones y este a su vez responde con un comportamiento determinado (2).

3.3.1 Pruebas de caja negra.

Las pruebas de caja negra son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro (Caja negra). Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba (33).

Descripción de las variables.

	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	nombre	<i>string</i>	no	Debe contener sólo letras
2	dominio	<i>string</i>	no	Debe contener sólo letras y puntos.

Tabla 25 Descripción de las variables caso de pruebas gestionar usuario.

Caso de pruebas Gestionar Usuario

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 comprueba que los datos estén correctamente y el usuario no esté en la base de datos.	El administrador escoge la opción "Insertar Usuario". El sistema le permite introducir los datos.
		"afcespedes"	"uci.cu"		
EC 1.2 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario 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.	
		(vacío)	(vacío)		
EC 1.3 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario incorrectamente.	I	V	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
		(vacío)	"uci.cu"		
EC 1.4 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario incorrectamente.	V	I	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
		"afcespedes"	(vacío)		
EC 1.5 Insertar usuario incorrectamente.	El administrador introduce los datos del usuario incorrectamente.	I	I	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que	
		"12afce"	"12.uci"		

Escenario	Descripción	nombre	dominio	Respuesta del sistema	Flujo central
				los datos están incorrectos.	
EC 2.1 Modificar usuario correctamente.	El administrador introduce los datos del usuario correctamente	V "afcespedes"	V "uci.cu"	El sistema comprueba que los datos estén correctamente.	El administrador escoge la opción "Modificar Usuario". El sistema le permite escoger el usuario a modificar.
EC 2.2 Modificar el usuario incorrectamente.	El administrador introduce los datos del usuario incorrectamente.	I "12afce"	I "12.uci"	El sistema verifica que los datos estén correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.	
Escenario	Descripción	nombre		Respuesta del sistema	Flujo central
EC 3.1 Eliminar usuario correctamente.	El administrador escoge el nombre del usuario correctamente	V "afcespedes"		El sistema comprueba que los datos estén correctamente.	El administrador escoge la opción "Eliminar Usuario". El sistema le permite escoger el usuario a eliminar.
EC 3.2 Eliminar el usuario incorrectamente.	El administrador escoge el nombre del usuario incorrectamente.	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.	

Tabla 26 Caso de pruebas Gestionar Usuario.

Descripción de las variables.

Nombre del campo	Clasificación	Valor Nulo	Descripción

1	Serial	<i>string</i>	no	Puede contener letras y números
2	Fabricante	<i>string</i>	no	Puede contener letras y números
3	Modelo	<i>string</i>	no	Puede contener letras y números
4	IdModelo	<i>string</i>	no	Puede contener letras y números
5	tipo	<i>string</i>	no	Debe contener sólo letras.

Tabla 27 Descripción de las variables caso de pruebas registrar dispositivo.

Caso de pruebas Registrar dispositivo.

Escenario	Descripción	Serial	Fabricante	Modelo	IdModelo	tipo	Respuesta del sistema.	Flujo central
EC 4.1 Insertar los datos de un dispositivo correctamente	El administrador introduce los datos del dispositivo correctamente.	V "2030E"	V "SanDisk"	V "CruzeR Blade"	V "201"	V Memoria USB	El sistema comprueba que los datos estén correctamente y el dispositivo no esté en la base de datos.	El administrador escoge la opción "Registrar Dispositivo". El sistema le permite introducir los datos.
EC 4.2 Insertar los datos de un dispositivo incorrectamente	El administrador introduce los datos del dispositivo incorrectamente.	I (Vacío)	I (Vacío)	I (Vacío)	I (Vacío)	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.	

Tabla 28. Caso de pruebas Registrar dispositivo.

Descripción de las variables.

	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Usuarionombre	<i>string</i>	no	Debe contener sólo letras
2	DispositivoidDispositivo	<i>int</i>	no	Debe contener sólo números
3	Permisoidpermiso	<i>int</i>	no	Debe contener sólo números

Tabla 29. Descripción de las variables caso de asignar permiso a usuario.

Caso de pruebas Asignar permisos a usuario.

Escenario	Descripción	Usuarionombre	DispositivoidDispositivo	Permisoidpermiso	Respuesta del sistema.	Flujo central
EC 4.1 Asignar permisos a usuarios correctamente	El administrador escoge los datos del usuario, dispositivo y permiso correctamente.	V	V	V	El sistema verifica que los datos estén correctos, y le asigna los permisos al usuario sobre el dispositivo.	El administrador escoge la opción "Asignar Permiso". El sistema le permite introducir los datos.
		"afcespedes"	"1"	"7"		
EC 4.2 Asignar permisos a usuarios	El administrador escoge los datos del	I	I	I	El sistema verifica que los datos estén	
		(Vacío)	(Vacío)	(Vacío)		

incorrectamente	usuario, dispositivo y permiso incorrectamente.			correctos, de no ser así, emitirá un mensaje de error indicando que los datos están incorrectos.
-----------------	---	--	--	--

Tabla 30. Caso de pruebas Registrar dispositivo.

3.4 Resultados de las pruebas.

Durante el proceso de pruebas realizadas al sistema se detectaron un grupo de no conformidades de los casos de pruebas anteriores, los cuales arrojaron 7 no conformidades en la primera iteración las cuales fueron resueltas y en la segunda iteración se detectaron 2 no conformidades las cuales fueron resueltas también. A continuación se muestra un gráfico que representa las no conformidades.

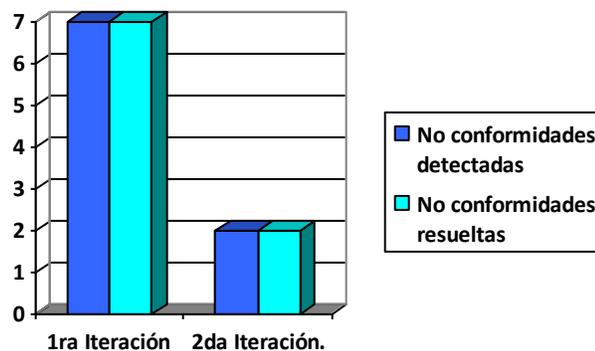


Figura 13. Resultados de las pruebas

3.5 Modelo de 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 (2).

Computadora Cliente: computadora a la cual un usuario del dominio conecta un dispositivo y descarga la base de datos del servidor.

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.



Figura 14. Diagrama de despliegue

3.6 Conclusiones Parciales.

En el presente capítulo se realizó el diagrama de componentes, también se definió los estándares de codificación a utilizar. 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

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.
- El estudio de sistemas homólogos permitió obtener las principales funcionalidades con que cuenta el sistema.
- La arquitectura de *software* utilizada permitió lograr un sistema escalable.
- La utilización de protocolo SSH permitió implementar mecanismos que hacen más segura la comunicación entre el cliente y el servidor.
- El sistema implementado permite la administración exitosa de dispositivos USB.
- Las pruebas realizadas permitieron lograr un producto que cumple con los requisitos planteados inicialmente.

RECOMENDACIONES

Una vez concluida la presente investigación, se recomienda para futuras versiones del sistema:

- Dotarle de nuevas funcionalidades al sistema para darle soporte a otros tipos de dispositivos USB.
- Continuar ampliando el sistema para que gestione los permisos, no sólo a nivel de usuarios del dominio, sino también a nivel de usuarios locales y a nivel de computadoras.

REFERENCIAS BIBLIOGRÁFICAS

1. Entorno visual de aprendizaje. [En línea]. [Citado el: 21 de noviembre del 2012]. Disponible en: http://eva.uci.cu/file.php/92/Tema_1_Introduccion/Materiales/Apoyo/Conceptos_basicos_v2.0.pdf.
2. MORA Vega, Yusmila. Sistema para la administración remota de dispositivos USB. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas, 2012. 81 h.
3. “Definición de dispositivos de almacenamiento - Qué es, Significado y Concepto.” [En línea]. [Citado el: 11 de febrero del 2013]. Disponible en: <http://definicion.de/dispositivos-de-almacenamiento/>.
4. informaticamoderna.com. [En línea]. [Citado el: 12 de febrero del 2013]. Disponible en: http://www.informaticamoderna.com/Memorias_USB.htm#defm.
5. “impresoras: DEFINICION DE IMPRESORA.”. [En línea]. [Citado el: 12 de febrero del 2013]. Disponible en: <http://impresoras-teffy.blogspot.com/2007/09/definicion-de-impresora.html>.
6. articulosya.com. [En línea]. [Citado: 13 de febrero del 2013]. Disponible en: <http://www.articulosya.com/article/9031/que-es-una-impresora-multifuncion.aspx>.
7. “Definición de administración - Qué es, Significado y Concepto”. [En línea]. [Citado el: 13 de febrero del 2013]. Disponible en: <http://definicion.de/administracion/>.
8. Buenas Tareas. [En línea]. [Citado el: 13 de febrero del 2013]. Disponible en: <http://www.buenastareas.com/ensayos/Concepto-De-Administracion-Importancia-Caracteristicas/2594302.html>.
9. Manual de Ayuda de la Aplicación .*Manual*

10. appsmashups.com. [En línea]. [Citado el: 14 de febrero del 2013]. Disponible en: <http://www.appsmashups.com/protege-tu-informacion-y-puertos-usb-con-myusbonly/>.
11. NirSoft. [En línea]. [Citado el: 15 de febrero del 2013]. Disponible en: http://www.nirsoft.net/utills/usb_devices_view.html.
12. Softpedia en español. [En línea]. [Citado: 15 de febrero del 2013]. Disponible en: <http://www.softpedia.es/programa-USB-Over-Network-30314.html>.
13. adictosalared.com. [En línea]. [Citado el: 15 de febrero del 2013]. Disponible en: <http://www.adictosalared.com/2008/02/seguridad-truecrypt/>.
14. Gestión de Proyectos. [En línea]. [Citado el: 18 de febrero del 2013]. Disponible en: <http://kasyles.blogspot.com/2008/09/openup-como-alternativa-metodolgica.html>.
15. DoCirs. [En línea]. [Citado el: 18 de febrero del 2013]. Disponible en: <http://www.docirs.cl/uml.htm>.
16. latinowebs.com. [En línea]. [Citado el: 18 febrero del 2013]. Disponible en: <http://recursotec.latinowebs.com/lecciones/c/source/capitulo1-1.htm>.
17. Recipes For Linux. [En línea]. [Citado el: 18 de febrero del 2013]. Disponible en: <http://www.recipesforlinux.com/2011/03/25/trabajando-con-udev-i-introduccion/>.
18. GTK+. [En línea]. [Citado el 18 de Febrero del 2013]. Disponible en: <http://www.calcifer.org/documentos/librognome/gtk.html>.
19. Protocolo SSH. [En línea]. [Citado el 18 de febrero del 2013]. Disponible en: <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>.

20. cavsi.com. [En línea]. [Citado el: 18 de febrero del 2013]. Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
21. SQLite Home Page. [En línea]. [Citado el 18 de febrero del 2013]. Disponible en: <http://www.sqlite.org/>.
22. Slideshare.net. [En línea]. [Citado el 2 de abril del 2013]. Disponible en: <http://www.slideshare.net/lloga7/caso-de-uso-presentation>.
23. Patrones Arquitectónicos. [En línea]. [Citado el 3 de Abril del 2013]. Disponible en: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>.
24. Blog de Julio Tentor. [En línea]. [Citado el 3 de Abril del 2013]. Disponible en: <http://jtentor.com.ar/post/Arquitectura-de-N-Capas-y-N-Niveles.aspx>.
25. Babel. [En línea]. [Citado el 3 de Abril del 2013]. Disponible en: http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas_sw.pdf.
26. Arquitectura cliente servidor - Oposiciones TIC. [En línea]. [Citado el 4 de Abril del 2013]. Disponible en: <http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html>.
27. Codecompiling. [En línea]. [Citado el 4 de Abril del 2013]. Disponible en: http://www.codecompiling.net/files/slides/UML_clase_04_UML_clases.pdf.
28. ¿Qué es un Patrón de Diseño? [En línea]. [Citado el 4 de Abril del 2013]. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
29. Slideshare. [En línea]. [Citado el 4 de Abril del 2013]. Disponible en: <http://www.slideshare.net/SergioRios/unidad-9-patrones-de-diseo>.

30. Microsoft. [En línea] [Citado el: 6 de abril del 2013]. Disponible en: <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
31. Estándares De Codificación. [En línea]. [Citado el: 10 de abril del 2013]. Disponible en: <http://www.buenastareas.com/ensayos/Estandares-De-Codificacion/3297956.html>.
32. Estándares de Codificación. [En línea]. [Citado el: 10 de abril del 2013]. Disponible en: https://docs.google.com/document/d/1rbxDfM0zsbFDNRZeM2FoXfRDbYSiSt6tCdbYPA0qdzs/edit?hl=en_US&pli=1.
33. Ingeniero de Gestion: Pruebas de Caja Negra y Caja Blanca. [En línea]. [Citado el: 4 de mayo del 2013]. Disponible en: <http://ingenierogestion.blogspot.com/2009/06/pruebas-de-caja-negra-y-caja-blanca.html>.

BIBLIOGRAFÍA

1. Graut et al (2004). "Metodología de la Investigación". S.I.: Fondo editorial Coruniversitaria, 2da ed, 2004.
2. Hill, Mc Graw.Sampieri et al (2006) "Metodología de la Investigación". S.I.: 4ta ed, 2006.

ANEXOS

Anexo 1: CU Asignar permisos a dispositivo.

Caso de uso:	Asignar permisos a dispositivo.	
Actores:	Administrador	
Propósito:	Este CU se lleva a cabo con el objetivo de que el administrador pueda asignar un permiso por defecto a un dispositivo.	
Resumen	El CU se inicia cuando el administrador asigna el permiso al dispositivo y termina cuando el sistema realiza la acción.	
Precondiciones:	El administrador tiene que estar autenticado en el sistema.	
Referencias:	RF9	
Prioridad:	Crítico	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El CU se inicia cuando el administrador selecciona la opción "Asignar permiso a dispositivo".	2. El sistema muestra la lista de permisos a asignar, y los dispositivos que están en la BD.	
3. El administrador escoge el identificador del dispositivo y el identificador del permiso que desea asignar a dispositivo escogido.	3. El CU termina cuando el sistema procesa la información según la acción realizada por el administrador y asigna los permisos al dispositivo.	

Tabla 31. CU Asignar permisos a dispositivo.

Anexo 2: CU Registrar datos de dispositivos.

Caso de uso:	Registrar datos de dispositivos
Actores:	Administrador

Propósito:	Este CU se lleva a cabo con el objetivo de que el administrador registre los datos de un dispositivo que le haya sido notificado.
Resumen	El caso de uso se inicia cuando el administrador registra en la base de datos los datos del dispositivo que le fue notificado para luego asignarle los permisos.
Precondiciones:	El administrador tiene que estar autenticado en el sistema.
Referencias:	RF6
Prioridad:	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. El CU se inicia cuando el administrador selecciona la opción "Registrar dispositivo".</p> <p>2. El administrador introduce los datos del dispositivo:</p> <ul style="list-style-type: none"> -Serial -Fabricante -Modelo -Identificador del modelo -Tipo 	<p>3. El sistema comprueba que los datos introducidos están correctos y que el dispositivo no está registrado en la base de datos.</p> <p>4. El CU termina cuando el sistema procesa la información según la acción realizada por el administrador y asigna los permisos al dispositivo.</p>
Flujos alternos	
Acción del actor	Respuesta del sistema
	<p>4. Si los datos están incorrectos o el dispositivo se encuentra registrado dentro de la base de datos, el sistema muestra un mensaje de error.</p>

Tabla 32. CU Registrar dispositivo.

Anexo 3: Diagrama de procesos del CU utilizar dispositivo.

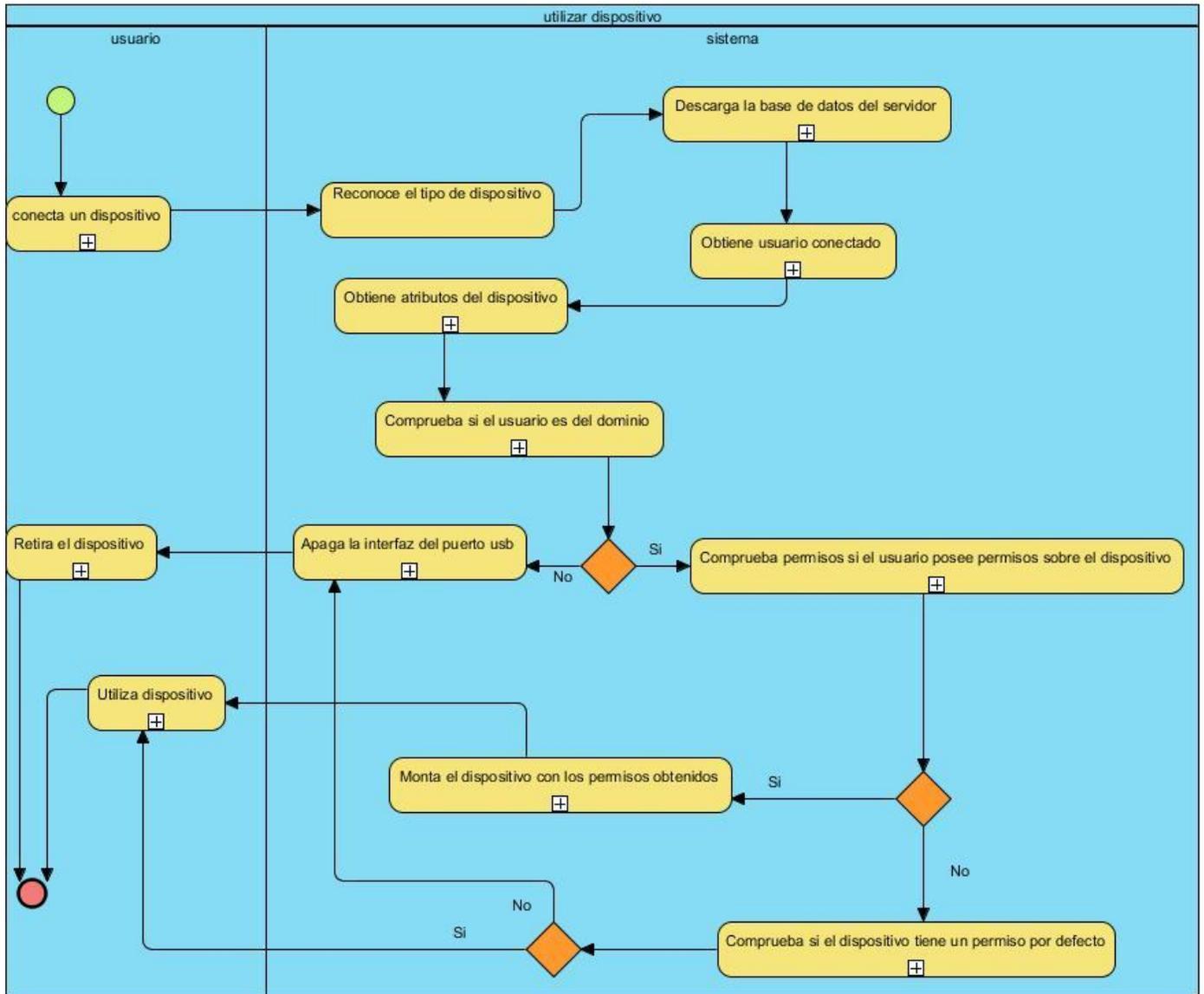


Figura 15. Diagrama de procesos CU utilizar dispositivo.

Anexo 4: Prototipo de interfaz “Insertar usuario”.

Insertar usuario

Nombre

Dominio

Detailed description: This is a user interface prototype for adding a user. It features a light gray background. At the top center, the title "Insertar usuario" is displayed. Below the title, there are two input fields. The first is labeled "Nombre" and the second is labeled "Dominio". At the bottom of the form, there are two buttons: "Aceptar" on the left and "Cancelar" on the right.

Figura 16. Prototipo de interfaz “Insertar usuario”.

Anexo 5: Prototipo de interfaz “Modificar usuario”.

Modificar usuario

Nombre

Dominio

Detailed description: This is a user interface prototype for modifying a user. It features a light gray background. At the top center, the title "Modificar usuario" is displayed. Below the title, there are two input fields. The first is labeled "Nombre" and the second is labeled "Dominio". At the bottom of the form, there are two buttons: "Aceptar" on the left and "Cancelar" on the right.

Figura 17. Prototipo de interfaz “Modificar usuario”.

Anexo 6: Prototipo de interfaz “Eliminar usuario”.

Eliminar Usuario

Nombre

Detailed description: This is a user interface prototype for deleting a user. It features a title 'Eliminar Usuario' at the top center. Below the title, there is a label 'Nombre' followed by a text input field. At the bottom of the form, there are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Figura 18. Prototipo de interfaz “Eliminar usuario”.

Anexo 7: Prototipo de interfaz “Asignar permiso a usuario”.

Asignar permiso a usuario

Usuario

Dispositivo

Permiso

Detailed description: This is a user interface prototype for assigning a permission to a user. It features a title 'Asignar permiso a usuario' at the top center. Below the title, there are three labels: 'Usuario', 'Dispositivo', and 'Permiso', each followed by a text input field. At the bottom of the form, there are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Figura 19. Prototipo de interfaz “Asignar permiso a usuario”.

Anexo 8: Prototipo de interfaz “Registrar dispositivo”.

Registrar dispositivo

Identificador

Serial

Fabricante

Modelo

IdModelo

Tipo ▼

Aceptar Cancelar

Detailed description: This is a form titled "Registrar dispositivo" (Register device). It contains six input fields: five text boxes for "Identificador", "Serial", "Fabricante", "Modelo", and "IdModelo", and one dropdown menu for "Tipo". At the bottom, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 20. Prototipo de interfaz “Registrar dispositivo”.

Anexo 9: Prototipo de interfaz “Asignar permiso a dispositivo”.

Asignar Permiso a dispositivo

Dispositivo ▼

Permiso ▼

Aceptar Cancelar

Detailed description: This is a form titled "Asignar Permiso a dispositivo" (Assign permission to device). It contains two dropdown menus: "Dispositivo" (Device) and "Permiso" (Permission). At the bottom, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 21. Prototipo de interfaz “Asignar permiso a dispositivo”.

Anexo 10: Descripción de la clase Dispositivo.

Nombre: Dispositivo	
Tipo de Clase: Pública	
Atributo	Tipo
Serial	<i>string</i>
Fabricante	<i>string</i>
Modelo	<i>string</i>
IdModelo	<i>string</i>
Tipo	<i>string</i>
cxn	<i>Conexion *</i>
Funciones	
<i>CreateDevice</i> : Función para registrar los datos de un dispositivo en la base de datos.	
<i>Devicetype</i> : Función para devolver el tipo de dispositivo.	
<i>ExistDevice</i> : Función para comprobar si el dispositivo está registrado en la base de datos.	
<i>IdDeviceList</i> : Función para devolver los identificadores de los dispositivos en la base de datos.	
Descripción	Esta clase maneja toda la información relacionada con los dispositivos USB

Tabla 33: Descripción de la clase Dispositivo.

Anexo 11: Descripción de la clase Permiso.

Nombre: Permiso

Tipo de Clase: Pública	
Atributo	Tipo
id_permiso	<i>int</i>
Permiso	<i>int</i>
tipo	<i>string</i>
cxn	<i>Conexion *</i>
Funciones	
<i>AsingPermisoUser:</i> Asigna un permiso a un usuario sobre un dispositivo determinado.	
<i>AsingPermisoDevice:</i> Asigna un permiso por defecto a un dispositivo determinado.	
<i>IdList:</i> Devuelve una lista de los identificadores de los permisos registrados en la base de datos.	
Descripción	Esta clase maneja toda la información con respecto a los permisos que se van a utilizar en el sistema.

Tabla 34: Descripción de la clase Permiso.

Anexo 12: Descripción de la clase ControladoraGestionarUsuario.

.Nombre: ControladoraGestionarUsuario	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>InsertUserObject:</i> Función para registrar un usuario en la base de datos.	
<i>DeleteUserObject:</i> Función para eliminar un usuario de la base de datos.	

<i>UpdateUserObject</i> : Función para modificar un usuario de la base de datos.	
<i>GetUserObject</i> : Función para mostrar los usuario de la base de datos.	
<i>NameList</i> : Función para devolver un listado de nombres de los usuarios registrados en la base de datos.	
Descripción	Esta clase es la encargada de controlar todo el flujo de información con respecto a un usuario.

Tabla 35: Descripción de la clase ControladoraGestionarUsuario.

Anexo 13: Descripción de la clase ControladoraDispositivo.

Nombre: ControladoraDispositivo	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>InsertDevice</i> : Función para registrar un dispositivo en la base de datos.	
<i>TypeDevice</i> : Función para devolver el tipo de dispositivo.	
<i>ExistDevice</i> : Función para comprobar si el dispositivo está registrado en la base de datos.	
<i>IdDevList</i> : Función para devolver un listado con los identificadores de los dispositivos registrados dentro de la base de datos.	
Descripción	Esta clase es la encargada de controlar todo el flujo de información con respecto a un dispositivo.

Tabla 36: Descripción de la clase ControladoraDispositivo

Anexo 14: Descripción de la clase ControladoraPermiso.

Nombre: ControladoraPermiso

Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>AsignPermisoUser:</i> Asigna un permiso a un usuario sobre un dispositivo determinado.	
<i>AsignPermisoDevice:</i> Asigna un permiso a un usuario sobre un dispositivo determinado.	
<i>IdPermisoList:</i> Devuelve una lista de los identificadores de los permisos registrados en la base de datos.	
Descripción	Esta clase es la encargada de controlar todo el flujo de información con respecto a un permiso.

Tabla 37: Descripción de la clase ControladoraPermiso

Anexo 15: Descripción de la clase InterfazPrincipal.

Nombre: InterfazPrincipal	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>InitComponents:</i> Función que inicializa todos los componentes visuales.	
Descripción	Esta clase es la encargada de la interfaz visual principal.

Tabla 38: Descripción de la clase InterfazPrincipal.

Anexo 16: Descripción de la clase InterfazInsertarUsuario.

Nombre: InterfazInsertarUsuario
--

Tipo de Clase: InterfazDispositivo	
Atributo	Tipo
Funciones	
<i>InsertUser</i> : Función para insertar los datos de un usuario en la base de datos	
<i>ValidateData</i> : Función para validar los datos de un usuario a insertar en la base de datos.	
Descripción	Esta clase es la encargada de la interfaz visual para insertar los datos de un usuario en la base de datos.

Tabla 39: Descripción de la clase InterfazInsertarUsuario.

Anexo 17: Descripción de la clase InterfazModificarUsuario.

Nombre: InterfazModificarUsuario	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>UpdateUser</i> : Función para modificar los datos de un usuario en la base de datos	
<i>ValidateData</i> : Función para validar los datos de un usuario a insertar en la base de datos.	
<i>NameList</i> : Función para mostrar el listado de nombres de los usuarios registrados en la base de datos.	
Descripción	Esta clase es la encargada de la interfaz visual para modificar los datos de un usuario en la base de datos.

Tabla 40: Descripción de la clase InterfazModificarUsuario.

Anexo 18: Descripción de la clase InterfazEliminarUsuario.

Nombre: InterfazEliminarUsuario	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>DeleteUser</i> : Función para eliminar los datos de un usuario en la base de datos	
<i>NameList</i> : Función para mostrar el listado de nombres de los usuarios registrados en la base de datos.	
Descripción	Esta clase es la encargada de la interfaz visual para eliminar los datos de un usuario en la base de datos.

Tabla 41: Descripción de la clase InterfazEliminarUsuario.

Anexo 19: Descripción de la clase InterfazRegistrarDispositivo.

Nombre: InterfazRegistrarDispositivo	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>RegisDevice</i> : Función para registrar los datos de un dispositivo en la base de datos	
<i>ValidateDatos</i> : Función para validar los datos de un dispositivo a insertar en la base de datos.	
Descripción	Esta clase es la encargada de la interfaz visual para registrar los datos de un dispositivo en la

base de datos.

Tabla 42: Descripción de la clase InterfazRegistrarDispositivo.

Anexo 20: Descripción de la clase InterfazAsignarPermisoUsuario.

Nombre: InterfazRegistrarDispositivo	
Tipo de Clase: Pública	
Atributo	Tipo
Funciones	
<i>NameUserList:</i> Función para mostrar el listado de nombres de los usuarios registrados en la base de datos.	
<i>IdDevList:</i> Función para devolver un listado con los identificadores de los dispositivos registrados dentro de la base de datos.	
<i>IdListPermiso:</i> Devuelve una lista de los identificadores de los permisos registrados en la base de datos.	
<i>AsingPermisUserOverDevice:</i> Asigna al usuario un permiso sobre determinado dispositivo	
Descripción	Esta clase es la encargada de la interfaz visual para asignar un permiso a un usuario sobre determinado dispositivo.

Tabla 43: Descripción de la clase InterfazAsignarPermisoUsuario.

Anexo 21: Descripción de la clase InterfazAsignarPermisoDispositivo.

Nombre: InterfazAsignarPermisoDispositivo
Tipo de Clase: Pública

Atributo	Tipo
Funciones	
<i>IdDevList</i> : Función para devolver un listado con los identificadores de los dispositivos registrados dentro de la base de datos.	
<i>IdListPermiso</i> : Devuelve una lista de los identificadores de los permisos registrados en la base de datos.	
<i>AsingPerDevice</i> : Asigna al dispositivo un permiso por defecto.	
Descripción	Esta clase es la encargada de la interfaz visual para asignar un permiso por defecto a un determinado dispositivo.

Tabla 44: Descripción de la clase InterfazAsignarPermisoDispositivo.

GLOSARIO DE TÉRMINOS.

Glib: es una biblioteca de propósito general, provee tipos de datos en C que usualmente se incluyen de forma estándar a otros lenguajes.

GNOME: es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos GNU/Linux, compuesto enteramente de software libre.

GNU: proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

IDE: Integrate Development Enviroment: Entorno de desarrollo integrado. Herramienta que se usa para facilitar el desarrollo de software.

RUP: Proceso unificado de desarrollo.

Sistema operativo: Un sistema operativo es un software de sistema, es decir, un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos. Comienza a trabajar cuando se enciende el computador y gestiona el hardware de la máquina desde los niveles básicos, permitiendo también la interacción con el usuario.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.