



Universidad Complutense

Facultad de Informática



eGorilla

Especificación de Requisitos

Asignatura: Ingeniería del Software

Curso Académico: 2008/2009

Grupo: 4º B

Historial del Documento

Persona/s que redactó el documento originalmente:	Todos los miembros del grupo
Persona/s que han sometido a revisión el documento:	Javier Salcedo, David Fernández, Qiang Zhou, Mercedes Bernal
Persona que ha validado el documento:	Javier Salcedo

Índice

1. Introducción	4
1.1. Propósito	4
1.2. Ámbito del Sistema	4
1.3. Definiciones, Acrónimos y Abreviaturas	5
1.4. Referencias.....	6
1.5. Visión General del Documento.....	6
2. Descripción General	7
2.1. Perspectiva del Producto.....	7
2.2. Funciones del producto.....	7
2.3. Características de los usuarios	7
2.4. Restricciones:	8
2.5. Suposiciones y Dependencias	10
2.6. Requisitos Futuros:	10
3. Requisitos Específicos.....	11
3.1. Interfaces Externas.....	11
3.3. Requisitos de Rendimiento	15
3.4. Restricciones de Diseño	15
3.5. Atributos del Sistema	16
4. Apéndices	18



1. Introducción

1.1. Propósito

Desarrollar una **aplicación software P2P usando protocolo IP (TCP/UDP)** para las transmisiones de los paquetes enviados entre los distintos componentes de la aplicación y **un único servidor** encargado de gestionar todos los datos relativos a los elementos manejados por la aplicación **que funcionará en los laboratorios de la facultad.**

1.2. Ámbito del Sistema

El sistema a desarrollar es una aplicación encargada de la compartición (subida y descarga) de archivos que va a funcionar en una red P2P semidistribuida formada por nodos tipo cliente-servidor. Cada cliente hará de servidor a su vez a los otros clientes de dicha red, sirviendo sus archivos.

A continuación se detalla la lista de características que definen de una forma más específica el ámbito del sistema:

- Será desarrollado con **carácter de multiplataforma (Linux, Windows)**, es decir será soportada por varios sistemas operativos haciendo de la aplicación un software portable. Para ello se empleará tecnología **Java**.
- Cada nodo estará **conectado simultáneamente a un único servidor que además será el mismo**. La aplicación **constará de un único servidor** en principio, pudiendo ser ampliado el número de los mismos en un futuro.
- Se utilizará un **sistema de identificación de archivos independiente del nombre**. Para ello se empleará **el algoritmo md5** que asigna a cada archivo con un determinado nombre un **identificador único e inequívoco** dentro de la red P2P.
- Se permitirá la compartición de archivos no completos así como la descarga simultánea de archivos desde varios nodos con un sistema de recuperación de errores en dichas descargas, haciendo posible la reanudación de dichas descargas. Para ello se utilizarán ficheros de **seguridad XML y properties que guardarán los datos relevantes de la aplicación**.
- El sistema comprobará mediante una búsqueda consultando al servidor si el identificador ya existe en la red.





1.3. Definiciones, Acrónimos y Abreviaturas

- **Sistema, Aplicación, eGorilla:** Se hará referencia a estos tres términos en todo el documento tratándolos como uno sólo, es decir como **sinónimos**.
- **Fichero, Archivo:** Se hará referencia a estos dos términos en todo el documento tratándolos como uno sólo, es decir como **sinónimos**.
- **P2P:** Acrónimo de Peer to Peer. Se refiere a una red que no tiene clientes ni servidores fijos, sino una serie de nodos que se comportan simultáneamente como clientes y como servidores respecto de los demás nodos de la red.
- **SVN:** Acrónimo de Sistema de Control de Versiones. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como **svn** por ser ese el nombre de la herramienta de línea de comandos. Los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. Para la gestión del svn se emplearán las herramientas **TortoiseSVN** y **RapidSVN**.
- **Nodo:** Componente activo de la red P2P desarrollada ya sea cliente, servidor o las dos cosas.
- **Md5:** Algoritmo que dado un nombre asigna un identificador único al mismo garantizando (en un 99%) que no existan dos identificadores de archivo iguales en la red P2P.
- **IP, TCP, UDP:** Protocolos asociados a teoría de redes. Son protocolos estándares que se emplean en las tecnologías de redes actuales. Se empleará el protocolo **IP** para las comunicaciones dentro de la red.
- **Máquina Virtual de Java (JVM):** es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.
- **Jdk:** Plataforma de desarrollo de la tecnología Java.
- **Paquete:** Unidad mínima de transferencia de información en la red P2P.
- **NetBeans:** IDE de Sun para desarrollo de aplicaciones principalmente de tecnología Java.
- **Archivo:** Sinónimo de fichero. Es la unidad básica en la red P2P susceptible de ser compartida o de de ser descargada. Cada archivo





será gestionado por el módulo **GestorArchivos** encargado del fragmentado, reensamblado y de la gestión de los diferentes tipos de archivo que se pueden compartir en la red P2P.

- **XML:** Lenguaje de programación basado en etiquetas empleado para desarrollar diferentes aplicaciones web así como otros usos.
- **Properties:** Fichero de configuración donde se especifican los valores o características de un determinado sistema. Su contenido está formado por cláusulas del estilo *nombreCampo = valor*.

1.4. Referencias

- “The eMule Protocol Specification”, <http://www.sourceforge.net>
- Especificación de Requisitos según el **estándar de IEEE 830**.
- **Transparencias** de la asignatura de Ingeniería del Software de curso académico 2008/2009 de la Facultad de Informática en la Universidad Complutense de Madrid.

1.5. Visión General del Documento

En este documento se plasmarán las **ideas generales del sistema**, concretamente las funcionalidades y requisitos básicos deseables del mismo.

No tendrá contenidos relativos a implementaciones concretas ni de los algoritmos utilizados así como de las atribuciones y la distribución del equipo de trabajo asociado al sistema aunque sí se hará referencia a algunas consideraciones en cuanto al carácter modular de la aplicación (correspondiente a la parte de **Arquitectura del Sistema**) y al nombre de alguno de estos módulos concretos remitiendo al lector a la lectura de dicho documento.

En la primera parte del mismo, el lector podrá encontrar una breve **Introducción** donde se introducen diversos conceptos generales al lector para que se sitúe dentro del marco de la aplicación eGorilla.

Después se procede a detallar una **Descripción General** del Sistema ayudando al lector a fijar los conceptos necesarios sobre el sistema que se van a abordar durante el desarrollo de la misma.

Por último se especificarán los **Requisitos Funcionales Específicos** del sistema, donde el lector puede encontrar y entender los requisitos principales del sistema y que se tendrán que cumplir y seguir durante todo el proceso de desarrollo del sistema.





2. Descripción General

2.1. Perspectiva del Producto

Se diseñará un sistema que permita la **compartición e intercambio de archivos** entre usuarios en los equipos de la Facultad de Informática de la Universidad Complutense de Madrid basado en la filosofía P2P.

Para ello el sistema se dividirá en dos partes bien diferenciadas: Una **aplicación Servidor** y una **aplicación Cliente**.

2.2. Funciones del producto

❖ Aplicación Servidor.

- **Gestión de Usuarios y archivos de los mismos** en la aplicación Servidor.
- **Configuración de las opciones** de la aplicación Servidor.

❖ Aplicación Cliente.

- **Compartición de archivos** entre clientes de la red P2P asociada al sistema. Para ello la aplicación Cliente extraerá toda la información de los archivos que tiene el usuario mediante una carpeta de **Archivos Compartidos**.
- **Descarga de archivos** de otros clientes de la red P2P asociada al sistema.
- **Búsqueda de archivos** en el servidor único de la red P2P.
- **Atención preferente** de las solicitudes de los usuarios con **carácter prioritario**. Por el momento no se contempla.
- **Sistema de ayuda** acerca del funcionamiento del sistema.
- **Generación de estadísticas** en tiempo real de la actividad de la aplicación, a petición del usuario.
- **Configuración de las opciones** de la aplicación Cliente.

2.3. Características de los usuarios

Dado que el sistema está dividido en dos subsistemas: Cliente y Servidor, la aplicación irá orientada a usuarios con **conocimientos avanzados** de informática para la **aplicación Servidor**, y **se requiere**





conocimientos básicos de informática para los **usuarios de la aplicación Cliente**.

La aplicación Servidor no está orientada a ser utilizada por todo tipo de usuarios por lo que es necesario recalcar al lector que los usuarios comunes y generales usarán la aplicación Cliente y **en ningún caso** podrán usar la aplicación Servidor ya que ésta está únicamente destinada al uso por parte de los desarrolladores del sistema o personas con altos conocimientos informáticos.

2.4. Restricciones:

- *Política y licencias:*

Se empleará la **Licencia Pública General ó GPL**, con ello el propósito es declarar que el software cubierto por esta licencia es **software libre** y se protege de intentos de apropiación que restrinjan esas libertades a los usuarios.

- *Restricciones HW:*

Limitaciones propias de la red o de los equipos en los que se utiliza la aplicación, en concreto los equipos de los laboratorios de la facultad.

- *Restricciones SW:*

Se deberá disponer de la versión de la **máquina virtual de Java correcta** instalada en los equipos en los que se quiera ejecutar la aplicación para su correcto funcionamiento.

- *Actividades Simultaneas:*

- Proyectos y/o prácticas correspondientes a otras asignaturas de la carrera debido al **carácter docente** del desarrollo de la aplicación.
- **Prácticas laborales** de algunos de los miembros del grupo de desarrollo.

- *Auditoría:*

El profesor de la asignatura se encargará de revisar, examinar y evaluar los resultados relativos al sistema a implementar.





- *Funciones de Control:*

Se emplearán las siguientes funciones de control en el sistema:

- **Repositorio:** Empleado para la correcta actualización de las diferentes versiones del sistema. El enlace correspondiente se adjunta a continuación:

<http://code.google.com/p/egorilla/>

- **Grupo de Google:** Empleado para la **facilitación de la comunicación** entre todos los integrantes del grupo de desarrollo del sistema. El enlace correspondiente a dicho grupo se detalla a continuación:

<http://groups.google.es/group/egorilla>

- **Reuniones semanales:** Se procederá tal y como su propio nombre indica a una o varias reuniones semanales de los miembros del grupo de desarrollo.
- **Clientes de Mensajería Instantánea:** Empleados para **una comunicación más eficiente y rápida** entre los diferentes miembros del grupo de desarrollo. Por ejemplo **Windows Messenger** o equivalentes en otros sistemas operativos.
- **Jefe de Proyecto:** Se encargará de la coordinación y la asignación de las distintas tareas a realizar en cada momento del proyecto dependiendo de las diferentes necesidades que el mismo exija.
- **Revisiones** de cada documento por cada miembro del proyecto.

- *Lenguaje(s) de programación:*

Para el desarrollo de la interfaz del sistema así como para su implementación interna correspondiente al dominio de las redes se empleará el **lenguaje de programación Java exclusivamente**.

- *Protocolos de Comunicación:*

Para la comunicación entre los distintos nodos del sistema se empleará **el protocolo TCP/IP**.

Dentro de la aplicación se podrán distinguir los siguientes protocolos:

- **Cliente-Servidor:** Define los protocolos de comunicación entre el cliente y el servidor. Se pueden distinguir los siguientes subprotocolos:
 - **ProtocoloDeDescargas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo





solicita la descarga de un archivo.

- **ProtocoloDeBusquedas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo solicita la búsqueda de un archivo al servidor.
- **ProtocoloDeSubidas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo sube (o comparte) un archivo al servidor que puede ser descargado por parte de otro cliente.
- **Cliente-Cliente:** Define los protocolos de comunicación entre dos clientes. Se pueden distinguir los siguientes protocolos:
 - **ProtocoloDeDescargas:** Encargado de definir el protocolo a seguir entre dos clientes cuando uno le solicita al otro la descarga de un archivo.

- *Requisitos de Habilidad:*

Los desarrolladores tendrán conocimientos sobre aplicaciones **Cliente-Servidor** así como uso de **herramientas CASE** para la realización de la parte relativa a la documentación del sistema.

- *Criticidad de la aplicación:*

Criticidad baja. Al tratarse de una aplicación P2P cualquier fallo en la aplicación solo afectará a los propios usuarios de la aplicación.

- *Consideraciones acerca de la seguridad:*

Dado que el sistema trata la compartición de recursos en la red, el mismo **garantizará la seguridad** de los equipos involucrados, evitando el acceso desde el exterior exclusivamente a los recursos autorizados (**Ficheros, puertos, Direcciones IP**).

2.5. Suposiciones y Dependencias

La aplicación dependerá de la **versión 1.6 de jdk** que es la versión sobre la cual está desarrollada la aplicación.

2.6. Requisitos Futuros:

- **Comunicación entre diferentes servidores** no limitando el funcionamiento a la existencia de un solo servidor.
- **Conexión a diferentes redes P2P existentes.**





- **Empleo del protocolo UDP** para emplear la aplicación fuera de la red local de la Facultad de Informática de la Universidad Complutense de Madrid.
- Diseño de una **interfaz simple** para la **aplicación Servidor**.
- **Previsualización** de los archivos incompletos.
- Integración de una **aplicación chat** entre los clientes del sistema.
- **Ayuda Online** de la **aplicación Cliente**.
- **Internacionalización de la aplicación**. Multilenguaje (español, inglés,...).
- **Diferentes temas** para la interfaz.
- **Creación de una interfaces alternativas** (modo mascara).

3. Requisitos Específicos

3.1. Interfaces Externas

Se describen los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

- **IE_REQUISITO1: Interfaz usuario**

La interfaz de usuario será simple dado su carácter general y orientado a usuarios sin habilidades avanzadas en el manejo de aplicaciones para que éstos se familiaricen con su uso **de forma rápida**. La aplicación contendrá **manuales de ayuda** y una **selección de controles** (botones, cuadros de texto, etc.) con títulos significativos.

Las diferentes áreas del cliente estarán separadas **de forma clara por pestañas o ventanas diferentes** dado el carácter independiente de cada una de ellas, pudiendo ser seleccionada cada una de ellas mediante un simple **“click”** de ratón.

La interfaz se divide en diferentes paneles:

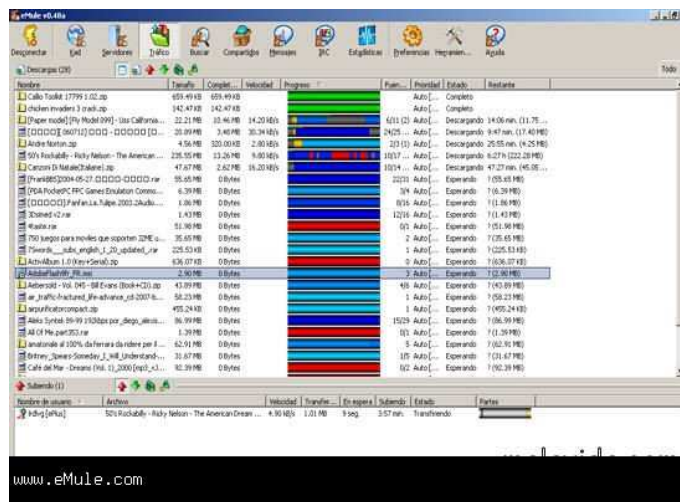
- Un panel superior donde se encuentran los botones principales de cada una de las vistas de la aplicación Cliente.
- Un panel central encargado de la visualización de las diferentes vistas seleccionadas mediante los botones del panel superior.
- Un panel inferior encargado de mostrar el estado de la aplicación Cliente con respecto a la red P2P.



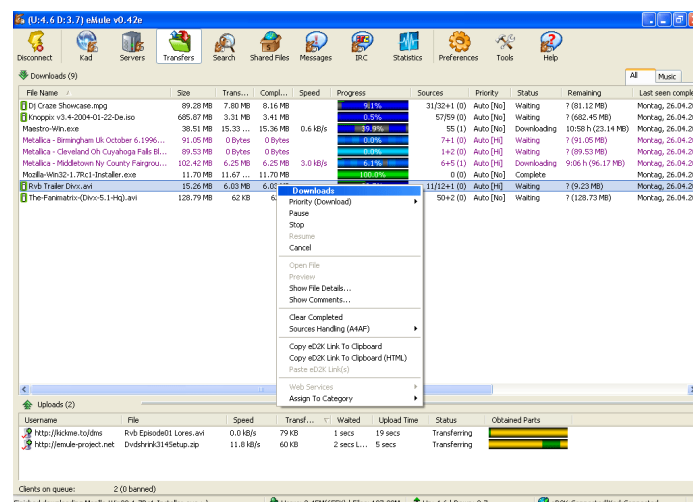


A continuación se describe con más detalle cada una de las pestañas de la misma:

La **pestaña de descargas** muestra las descargas del usuario en una lista, diferenciándose el estado de las descargas mediante un **código de colores**. Dicha lista de descargas también contendrá información de dichas descargas separadas en **diferentes campos de información** tales como **prioridad, tiempo restante, velocidad de descarga, estado de la descarga, etc.** al estilo de la captura siguiente:



Además de la anterior visualización, el usuario podrá realizar acciones extra sobre cada elemento de la lista de descargas accediendo mediante un simple click con el botón derecho al menú contextual de las descargas.



Nota: Las capturas anteriores han sido tomadas del cliente P2P eMule pero la interfaz gráfica del cliente eGorilla no tendrá ese

Especificación de Requisitos

Aplicación P2P eGorilla



**aspecto.**

- **Pestaña de Búsqueda:** permite realizar búsquedas de los archivos que el usuario desee descargarse. Estas búsquedas puede ser filtradas según varios parámetros: tipo de fichero y nombre.
- **Pestaña de Servidores:** permite visualizar la lista de servidores a la que el usuario puede conectarse. En un principio se mostrará el servidor por defecto de la aplicación al cual se podrá conectar el usuario a través de esta pestaña.
- **Pestaña de Configuración:** permite modificar los principales parámetros de la aplicación, tales como velocidad de descarga, subida, puertos, etc.
- **Pestaña de Estadísticas:** se mostrarán las estadísticas en un gráfico según los valores introducidos de los diferentes parámetros (fechas, horas,...).
- **Pestaña de Ayuda:** La pestaña de ayuda corresponde a una ventana informativa con toda la ayuda necesaria sobre cada uno de los módulos en los que se divide la aplicación (significado, modo de uso, etc.)

- **IE_REQUISITO2:** *Interfaz de administración de servidor*

La gestión se realizará a través de un **fichero de configuraciones (XML o Properties)**, donde aparecerán las opciones a modificar. Igualmente se le dotará de un intérprete de comandos para poder visualizar el uso y estado durante la ejecución. **Carecerá de interfaz gráfica** limitando su visualización **por consola**.

- **IE_REQUISITO3:** *Interfaz de comunicación Cliente – Servidor*

La conexión de las aplicaciones cliente al servidor se realiza mediante un protocolo propio de red. Para ello se procederá al envío de la información necesaria desde dichas aplicaciones cliente para que el servidor pueda responder a las peticiones o solicitudes propuestas. Dichas peticiones o solicitudes serán resueltas y contestadas por el propio servidor y serán enviadas a las aplicaciones clientes usando un protocolo propio de red.

Se definen los siguientes protocolos:

- **ProtocoloDeConexion:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo solicita la conexión al primero.





- **ProtocoloDeDescargas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo solicita la descarga de un archivo.
 - **ProtocoloDeBusquedas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo solicita la búsqueda de un archivo en el Servidor.
 - **ProtocoloDeSubidas:** Encargado de definir el protocolo a seguir entre el servidor y el cliente cuando el segundo sube (o comparte) un archivo al servidor que puede ser descargado por parte de otro cliente.
- **IE_REQUISITO4:** *Interfaz de comunicación Cliente – Cliente*

Un cliente podrá realizar la conexión con otro cliente para la transmisión de fragmentos de ficheros que el otro cliente haya solicitado con anterioridad.

A su vez un cliente podrá realizar la conexión con otro cliente para comenzar a recibir fragmentos de un fichero solicitado por dicho cliente implicando una comunicación bidireccional con independencia del sentido del tráfico de datos entre ambos clientes que será resuelta por un protocolo propio de la red.

Este protocolo es el **ProtocoloDescargas** ya que los clientes se comunicarán entre ellos a la hora de descargar un archivo entre ellos.

3.2. Funciones

Las funciones del sistema vienen **ampliamente explicadas** en el documento adjunto de casos de uso, aún así ciñéndonos al estándar se detallan brevemente a continuación:

- *Funciones Cliente:*
 - Conexión con la red P2P. Se empleará para ello el protocolo **ProtocoloConexion**.
 - Visualización del **estado de los archivos pendientes de descarga**.
 - Visualización del **estado de los archivos compartidos**.
 - Visualización de **estadísticas**. Para ello se implementará el módulo **GestorEstadísticas** que recibirá eventos en aquellos puntos del sistema susceptibles de ser mostrados en estadísticas.





- **Búsqueda en el sistema de archivos.** El servidor único recibirá dichas consultas por parte de los clientes lo que hace imprescindible que el mismo deba tener conocimiento de los archivos registrados por parte de los otros clientes. Usará por tanto el protocolo **ProtocoloBúsquedas**.
 - Visualización de la **Ayuda de la aplicación**.
 - **Configuración de la aplicación Cliente** mediante un formulario de configuración. Dicha configuración **se guardará en un fichero XML de configuración**.
- *Funciones Servidor:*

Debe ser conocedor **de los usuarios y de los archivos de los mismos registrados en el sistema**. Para ello:

- Debe **registrar** a cada uno de los clientes que quiera participar en el sistema. El servidor tendrá una **tabla** donde guardará la información relativa a todos los clientes conectados al sistema en cada momento.
- Debe **conocer y organizar** los archivos compartidos por los clientes. Para ello tendrá una **tabla de archivos** donde guardará toda la información relativa a los archivos del sistema.
- Debe **resolver las consultas de los clientes**, informando de la ubicación de estos tanto al solicitante como al propietario de la solicitud de este. Para ello consultará la **tabla de archivos** y devolverá la respuesta indicada dependiendo del estado de dicha tabla.
- Debe **actualizar los datos** de estas estructuras de forma periódica. Esto se realizará mediante **listeners** que actualizarán los módulos afectados en tiempo real cuando se produzcan.
- Se podrán **configurar los diferentes parámetros** del funcionamiento del programa mediante el empleo de un **fichero de properties**.

3.3. *Requisitos de Rendimiento*

El sistema maximizará, en la medida de lo posible, el uso de los recursos de red del usuario de la aplicación para impedir limitaciones en la velocidad de su conexión.

3.4. *Restricciones de Diseño*





La aplicación será programada siguiendo el paradigma de **programación orientada a objetos**. Para ello se empleará la **versión jdk 1.6 de Java** y el IDE **NetBeans 6.1** para el desarrollo del código fuente de la aplicación así como para la documentación de la misma mediante el **plug-in de UML** disponible para dicha versión del IDE.

El uso de la aplicación en los laboratorios de la facultad restringirá el acceso a la red externa.

3.5. *Atributos del Sistema*

- *Fiable:*

El sistema evitará fallos provocados por la propia aplicación empleando para ello mecanismos de **copias de seguridad** para la posterior recuperación del sistema ante cualquier eventualidad externa al mismo como por ejemplo un fallo en la conexión de la red, sistema operativo, etc.

De todo ello se encargará el módulo **GestorDeConfiguración** que será el encargado de guardar toda la información del cliente de forma periódica en **archivos de configuración XML** relativa al estado de las descargas, configuración del cliente, etc.

- *Mantenible:*

El diseño de la aplicación permite la fácil integración de futuros módulos para la aplicación. Esto es posible por la **división modular** que en la que se divide la aplicación, posibilitando que cada módulo sea una **caja negra** donde simplemente se comunica con el resto de los módulos a través de interfaces y de los métodos que proporcionan las mismas.

- *Portable:*

Dado el **carácter multiplataforma** del lenguaje así como de las herramientas empleadas en el desarrollo de la aplicación garantizan que la misma sea portable en sistemas como **Windows y Linux**.

- *Seguro:*

Garantiza el anonimato del usuario y la confidencialidad de su sistema. Se limitará exclusivamente a informar del estado de los archivos compartidos y en ningún caso mostrará información adicional de los usuarios que realizan la compartición de los archivos en la red.

Para ello el servidor no almacenará información personal ni





comprometida de ningún usuario para evitar posibles ataques externos a la aplicación contra su sistema.





4. Apéndices

Se reservan los apéndices a posibles limitaciones, modificaciones y particularidades que puedan aparecer durante el modelo evolutivo de desarrollo del producto a seguir.

