



**Universidad Complutense**

Facultad de Informática



**eGorilla**

*Especificación de Requisitos*

*Asignatura: Ingeniería del Software*

*Curso: 2008/2009*

# Índice

1.	Introducción.....	3
1.1.	Propósito.....	3
1.2.	Ámbito del Sistema .....	3
1.3.	Definiciones, Acrónimos y Abreviaturas.....	3
1.4.	Referencias.....	4
1.5.	Visión General del Documento.....	4
2.	Descripción General .....	4
2.1.	Perspectiva del Producto.....	4
2.2.	Funciones del producto:.....	4
2.3.	Características de los usuarios: .....	4
2.4.	Restricciones:.....	4
2.5.	Suposiciones y Dependencias .....	6
2.6.	Requisitos Futuros:.....	6
3.	Requisitos Específicos .....	6
3.1.	Interfaces Externas .....	6
3.2.	Funciones .....	10
3.3.	Requisitos de Rendimiento .....	12
3.4.	Restricciones de Diseño .....	12
3.5.	Atributos del Sistema .....	12
4.	Apéndices .....	13



# 1. Introducción

## 1.1. Propósito

Desarrollar una aplicación software basada en una red P2P con un protocolo adaptado a las necesidades de la misma así como la implementación de los nodos o componentes que conforman dicha red. Se encargará de la gestión del intercambio y compartición de archivos entre los distintos usuarios que utilizan la aplicación.

## 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.

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, es decir será soportada por varios sistemas operativos haciendo de la aplicación un software portable.
- Cada nodo estará conectado simultáneamente a un único servidor.
- Se utilizará un sistema de identificación de archivos independiente del nombre.
- 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.
- Los nodos estarán identificados de manera única.
- Se comportará de forma eficiente de cara a los recursos de la red del usuario, es decir tratará de limitar, en la medida de lo posible un alto consumo de dichos recursos de red relativos al usuario.
- El sistema no enviará información confidencial, delegando al usuario la decisión de lo que desea compartir en la red P2P asociada.

## 1.3. Definiciones, Acrónimos y Abreviaturas

- **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. **Subversion** es un software de [sistema de control de versiones](#) diseñado específicamente para reemplazar al popular [CVS](#), el cual posee varias deficiencias. 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. Una característica importante de Subversion es que, a diferencia de CVS, 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.
- **Nodo:** Componente activo de la red P2P desarrollada ya sea cliente, servidor o las dos cosas.
- **Fake:** Archivo con contenido engañoso a lo indicado por su nombre.





- **IP, TCP, UDP:** Protocolos asociados a teoría de redes. Son protocolos estándares que se emplean en las tecnologías de redes actuales.
- **Máquina Virtual de Java (JVM):** es un [programa](#) nativo, es decir, 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](#).
- **eDonkey 2000 y Kad:** Son redes específicas del cliente eMule.

#### 1.4. Referencias

- "The eMule Protocol Specification", <http://www.sourceforge.net>
- Página oficial del emule, <http://www.emule-project.net>
- Especificación de Requisitos según el estándar de IEEE 830

#### 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.

## 2. Descripción General

#### 2.1. Perspectiva del Producto

Se diseñará un sistema rápido y fiable, que permita la compartición de archivos entre el mayor número de clientes posibles, amén de las características anteriormente mencionadas.

#### 2.2. Funciones del producto:

- Sistema de conexión a la aplicación.
- Compartición de archivos entre clientes de la red P2P asociada al sistema.
- Descarga de archivos de otros clientes de la red P2P asociada al sistema.
- Búsqueda de archivos en la red P2P asociada al sistema.
- Mantenimiento de una lista de usuarios preferentes asociados a cada usuario que se conecta al sistema mediante la aplicación del cliente.
- Atención preferente de las solicitudes de los usuarios con carácter prioritario.
- Previsualización de archivos multimedia que se encuentran en situación de descarga.
- Sistema de ayuda acerca del funcionamiento del sistema.

#### 2.3. Características de los usuarios:

El sistema se pretende dirigir a cualquier tipo de usuarios con conocimientos mínimos de informática, tales como manejo de sistemas operativos e Internet sin ningún tipo de manejo avanzado en ningún campo de la misma en particular.

#### 2.4. Restricciones:





- **Política y licencias:**

Se empleará la Licencia Pública General ó GPL. Por tanto nuestro 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:**

Las restricciones Hardware fundamentales son las relativas a la existencia de una conexión de red, ya sea Internet como cualquier otro tipo de red, así como el uso de los recursos ofertados en los laboratorios de la facultad para la realización de las pruebas de funcionamiento correspondientes.

- **Restricciones SW:**

Se requerirán aplicaciones multimedia para la función de previsualización de fragmentos.

Se deberá disponer de la versión de la máquina virtual de Java instalada en los equipos en los que se quiera ejecutar la aplicación.

- **Actividades Simultaneas:**

Proyectos y/o prácticas correspondientes a otras asignaturas de la carrera.

Por otra parte, algunos desarrolladores compatibilizan los estudios con otro tipo de proyectos, lo cual puede implicar un leve impacto (aunque no crítico) para el tiempo de duración del proyecto. Por tanto aún no ser demasiado importante ha de ser tenido en cuenta igualmente.

- **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.

- **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 Java.





- **Protocolos de Comunicación:**

Se emplearán los siguientes protocolos:

- TCP/IP.
- UDP.
- Protocolo de comunicación sobre la capa de aplicación.

- **Requisitos de Habilidad:**

Los desarrolladores tendrán conocimientos de programación, lenguajes orientados a objetos, aplicaciones cliente servidor así como de herramientas CASE para la realización de la parte relativa a la documentación del sistema.

- **Críticidad de la aplicación:**

La criticidad será máxima ya que el objetivo fundamental es la de ofrecer un software fiable, mantenible y de calidad a los futuros usuarios del sistema.

- **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.

## 2.5. Suposiciones y Dependencias

En la medida que la implementación del sistema requiera recursos no accesibles desde el entorno de programación Java, se incluirán módulos programados en otros lenguajes, tales como C, C++ ó C#.

## 2.6. Requisitos Futuros:

Gracias al diseño modular empleado en el desarrollo del sistema será posible la inclusión de nuevas funcionalidades tales como:

- Posibilidad de conexión a diferentes redes p2p existentes tales como Kad o la red eDonkey2000.
- Analisis del sistema para que no se produzca saturación de los recursos y así pausar ó reducir el consumos de los mismos por parte del sistema.

# 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.

- **RE\_IE\_IUsuario: Interfaz usuario**

- **R1\_U – ConectarServidor**





- *R2\_U – ConsultarDescargas*
- *R3\_U – ConsultarCompartidos*
- *R5\_U – IniciarDescarga*
- *R6\_U – DesconectarServidor*
- *R7\_U – ConsultarEstadísticas*
- *R8\_U – CancelarDescarga*
- *R9\_U – PausarDescarga*
- *R10\_U – ReanudarDescarga*
- *R11\_U – PrevisualizarDescarga*
- *R12\_U – ConfigurarCliente*

La interfaz de usuario debe ser amigable e intuitiva dado su carácter general y orientado a usuarios sin habilidades avanzadas en el manejo de aplicaciones. Por lo tanto la aplicación permitirá al usuario la interacción con el sistema mediante técnicas intuitivas como “click”, “drag and drop”, etc. Así como una descripción clara de uso mediante manuales de ayuda y una selección de controles (botones, cuadros de texto, etc.) con títulos significativos haciéndola amigable y fácil de usar.

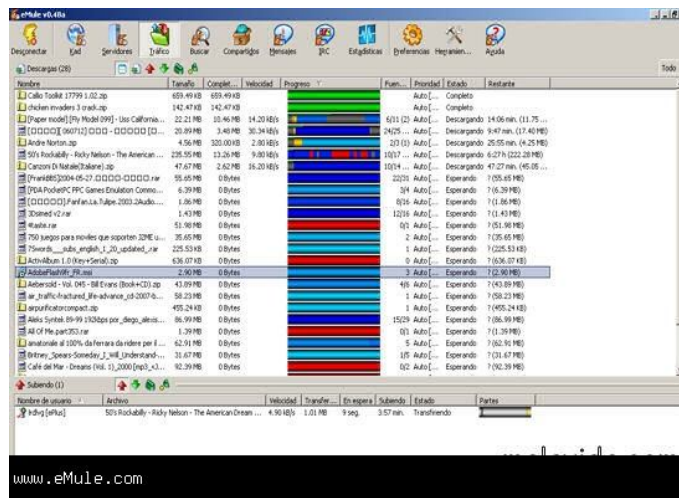
Las diferentes áreas del cliente estarán separadas **de forma inteligente** por pestañas o ventanas diferentes dado el carácter independiente de cada una de ellas, pudiendo ser seleccionada cada una de ellas mediante un simple click de ratón.

La pantalla inicial de la misma ofrecerá información sobre la actividad que se esté desarrollando en ese momento.

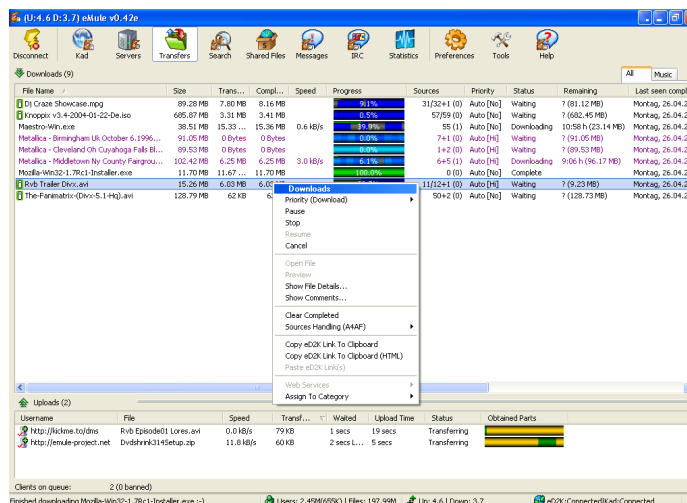
La pestaña de descargas del usuario será junto con la pestaña de búsqueda la parte más importante de la aplicación.

Las descargas del usuario serán mostradas 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 aspecto.

### • **RE\_IE\_IServidor:** Interfaz de administración de servidor

- **R1\_S – PeticiónAceptaCliente**
- **R2\_S – ActualizarTablaArchivos**
- **R3\_S – PeticiónBúsquedaArchivo**
- **R4\_S – DevolverFuentesArchivo**







- *R5\_S – PeticiónDesconectarCliente*
- *R6\_S – IniciarServidor*
- *R7\_S – ConfigurarServidor*
- *R8\_S – ApagarServidor*
- *R9\_S – PeticiónDescargaArchivo*

Se gestión se realizará a través de un fichero de configuraciones, 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 en un principio de interfaz gráfica dejando este cometido sujeto a una posible implementación simple (tal y como la empleada por el client-e eMule).

- ***RE\_IE\_IClienteServidor: Interfaz de comunicación Cliente – Servidor***
  - *R4\_U – BuscarArchivos*

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.

- ***RE\_IE\_IClienteCliente: 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.



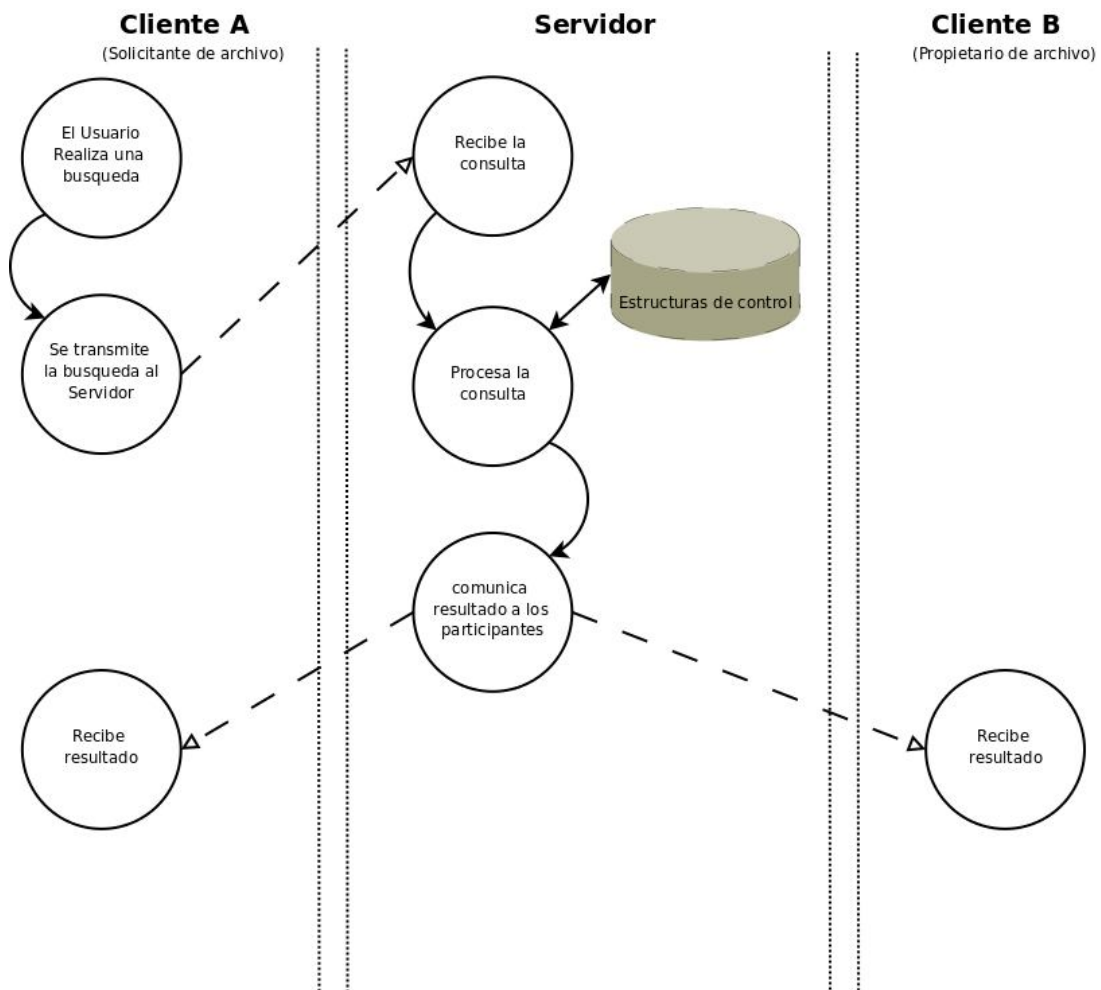


### 3.2. Funciones

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:**

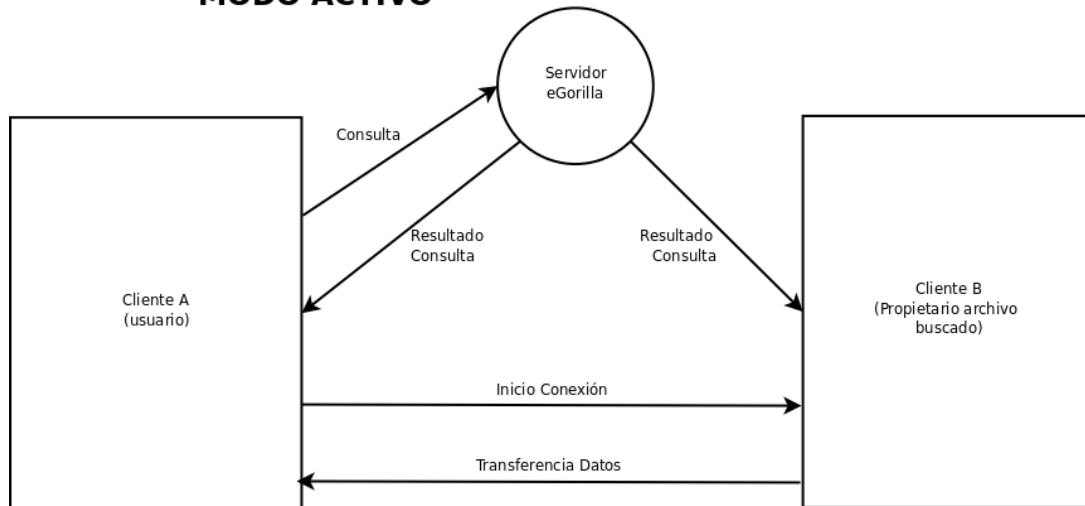
- Conexión con la red P2P.
- Visualización del estado de los archivos pendientes de descarga en todo momento.
- Visualización del estado de los archivos compartidos en todo momento.
- Visualización del consumo de recursos del sistema.
- Consulta de estadísticas del funcionamiento local del sistema.
- Búsqueda en el sistema de archivos. El servidor 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.





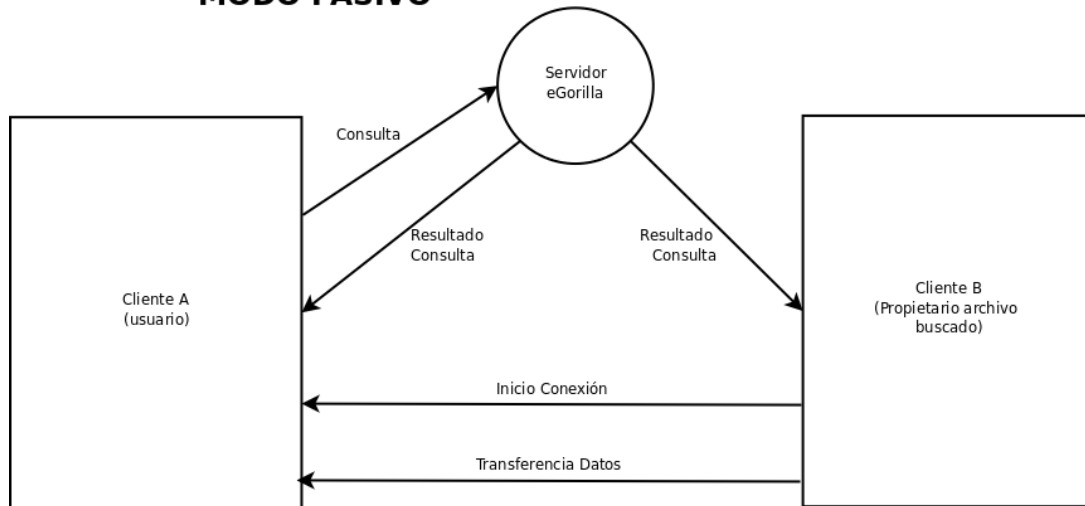
- Descarga de archivos desde otros clientes. Se podrá realizar de las siguientes maneras:
- **Activo:** El cliente A realiza la conexión con otro cliente B y comienza a enviar los fragmentos.

### MODO ACTIVO



- **Pasivo:** El cliente A acepta la conexión por parte de otro cliente B, y comienza a recibir fragmentos.

### MODO PASIVO





- **Funciones Servidor:**

El servidor es un ente muy importante en el sistema, ya que es el punto de contacto entre los usuarios. Debe ser conocedor del estado de los usuarios y de los archivos registrados en el sistema.

- Debe registrar a cada uno de los clientes que quiera participar en el sistema.
- Debe conocer y organizar los archivos compartidos por los clientes.
- Debe poder resolver las consultas de los clientes, informando de la ubicación de estos tanto al solicitante como al propietario de la solicitud de este.
- Debe actualizar los datos de estas estructuras de forma periódica.
- Se podrán configurar los diferentes parámetros del funcionamiento del programa.

### 3.3. *Requisitos de Rendimiento*

El rendimiento del sistema es un asunto crítico ya que el uso o no de este dependerá de el aprovechamiento que haga de los recursos de los usuarios, y con ello el éxito entre el público del sistema.

Por ello el sistema deberá maximizar, 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. Por norma general se limitará el uso de aplicaciones externas a las existentes en los laboratorios para facilitar las posibles pruebas a realizar en los laboratorios de la facultad de informática.

### 3.5. *Atributos del Sistema*

- **Fiable:**

Se asegura la fiabilidad del sistema en cuanto a sus funciones propuestas y desarrolladas ante posibles fallos externos a la aplicación (Reanudación de descargas, etc.).

- **Mantenible:**

El diseño de la aplicación permite la fácil integración de futuros módulos para la aplicación.

- **Portable:**

La plataforma de desarrollo planificada así como diferentes librerías asociadas permite la implantación de la aplicación en multitud de sistemas diferentes dado su carácter de multiplataforma.

- **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 se garantiza que el servidor no almacenará información personal ni comprometida de ningún usuario.





## 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.

