



**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: .....	5
2.4.	Restricciones: .....	5
2.5.	Suposiciones y Dependencias .....	7
2.6.	Requisitos Futuros: .....	7
3.	Requisitos Específicos .....	7
3.1.	Interfaces Externas .....	7
3.2.	Funciones .....	10
3.3.	Requisitos de Rendimiento .....	13
3.4.	Restricciones de Diseño .....	13
3.5.	Atributos del Sistema .....	13
4.	Apéndices .....	15



# 1. Introducción

## 1.1. Propósito

Desarrollar una **aplicación software P2P usando protocolo TCP** 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**, 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. La aplicación constará de un único servidor.
- 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**.
- Los nodos estarán identificados de manera única. El sistema comprobará mediante una búsqueda consultando al servidor si el identificador ya existe en la red.

## 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. 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 **TCP** 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 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 (**en nuestro caso como fichero de configuración**).
- **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 del curso académico de Ingeniería del Software 2008/2009.

#### 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 mediante el empleo de **un solo servidor** y mediante **protocolo TCP** haciendo uso de una **interfaz simple e intuitiva**.

### 2.2. Funciones del producto:





- Sistema de conexión y desconexión a la aplicación.
- Compartición de archivos entre clientes de la red P2P asociada al sistema. Para ello el usuario empleará 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.
- Sistema de ayuda acerca del funcionamiento del sistema.
- Generación de estadísticas en tiempo real de la actividad de la aplicación.
- Configuración de las opciones de la aplicación cliente mediante una interfaz sencilla e intuitiva.

### 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 como por ejemplo sobre teorías de redes, seguridad informática o similares.

### 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:**

Limitaciones propias de la red en particular asociada a la aplicación.

- **Restricciones SW:**

Se requerirán **aplicaciones multimedia** para la función de previsualización de los fragmentos de los archivos actualmente en descarga. Se empleará el reproductor **VCL** por su versatilidad y completitud ante diversos formatos de dichos archivos

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.
- Prácticas laborales.

- **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.
- **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.

- **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:**

Para la comunicación entre los distintos módulos de la aplicación 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 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.

- **Criticidad 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 (**Ficheros, puertos, Direcciones IP**)

## 2.5. Suposiciones y Dependencias

La aplicación dependerá de la **versión 1.6 de jdk**.

El código será generado exclusivamente utilizando el lenguaje Java mediante la herramienta **NetBeans 6.1** (dado que permite la incorporación del editor UML para implementar la documentación asociada al proyecto)

Se seguirá un **manual del programador** donde se especifican claramente el estándar a seguir a la hora de la implementación por parte de los desarrolladores del proyecto.

Además para evitar problemas relativos al uso de acentos, etc. En el código entre distintas plataformas se usará la codificación **UTF-8**.

## 2.6. Requisitos Futuros:

- Posibilidad de conexión a diferentes redes p2p existentes.
- Diseño de la interfaz del servidor.
- Previsualización de los archivos incompletos.

# 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**

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.

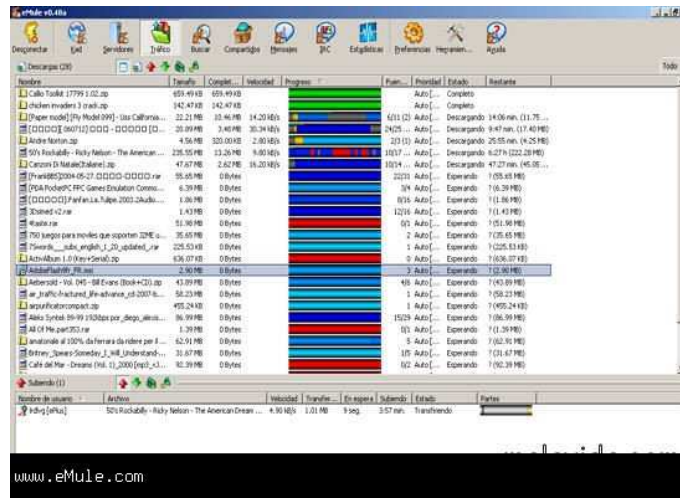
Las descargas del usuario serán mostradas en una lista, diferenciándose el estado de las descargas mediante un código de colores. Dicha lista de descargas también contendrá información de dichas descargas separadas en diferentes campos de información tales como prioridad, tiempo restante, velocidad de descarga, estado de la descarga, etc. al



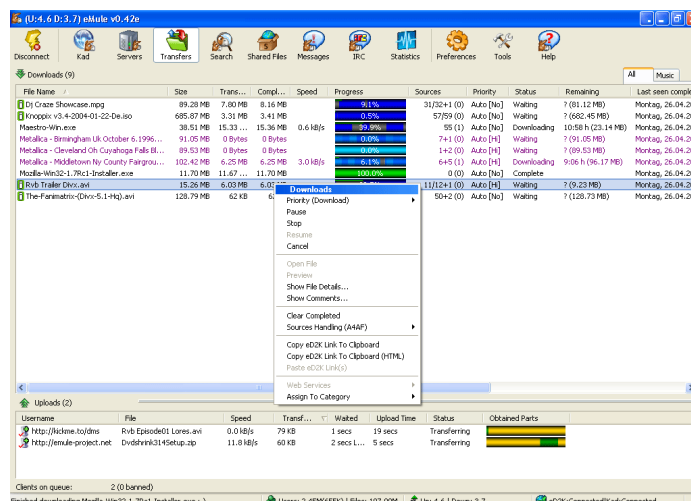




estilo de la captura siguiente:



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

Se 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 por tanto de interfaz gráfica.

- **RE\_IE\_IClienteServidor:** 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 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.

- ***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.

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





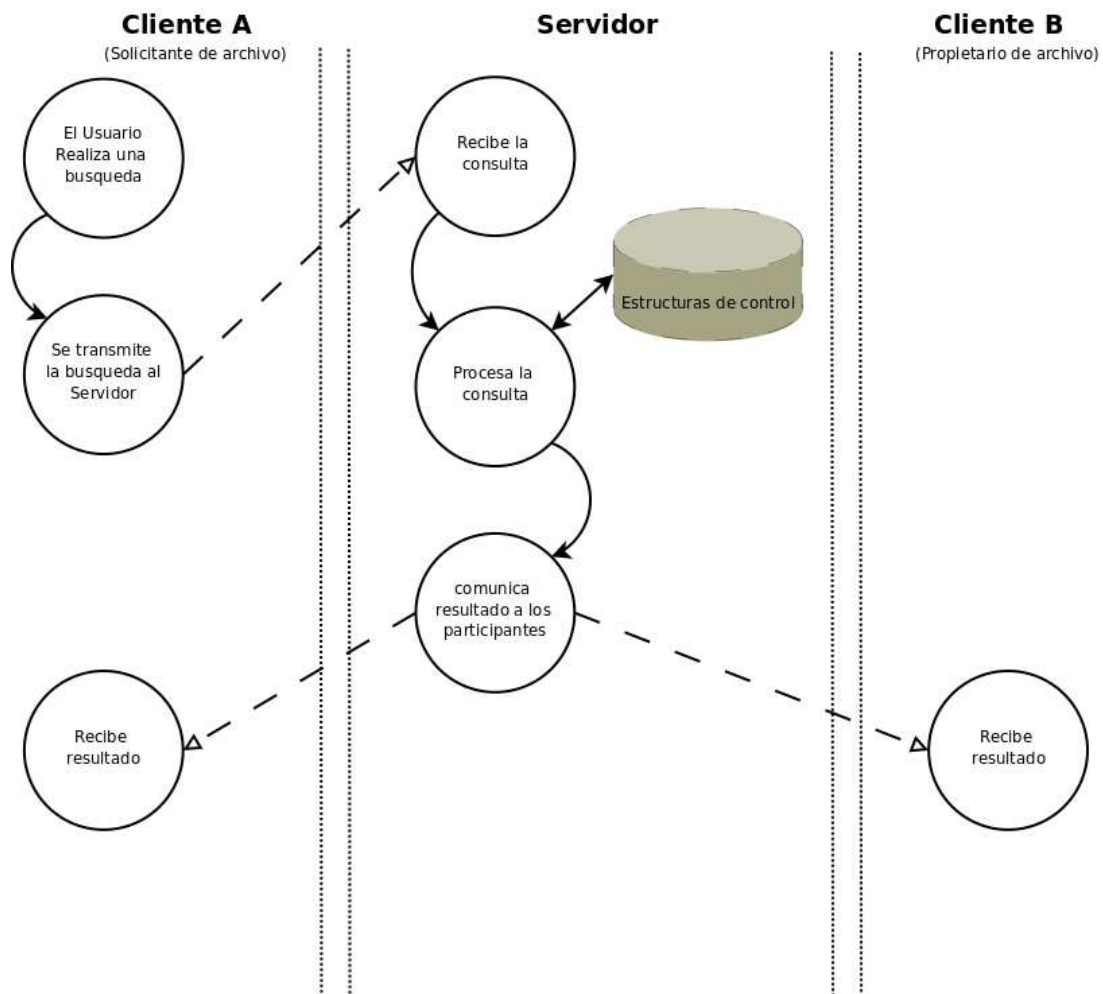
### 3.2. Funciones

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:**

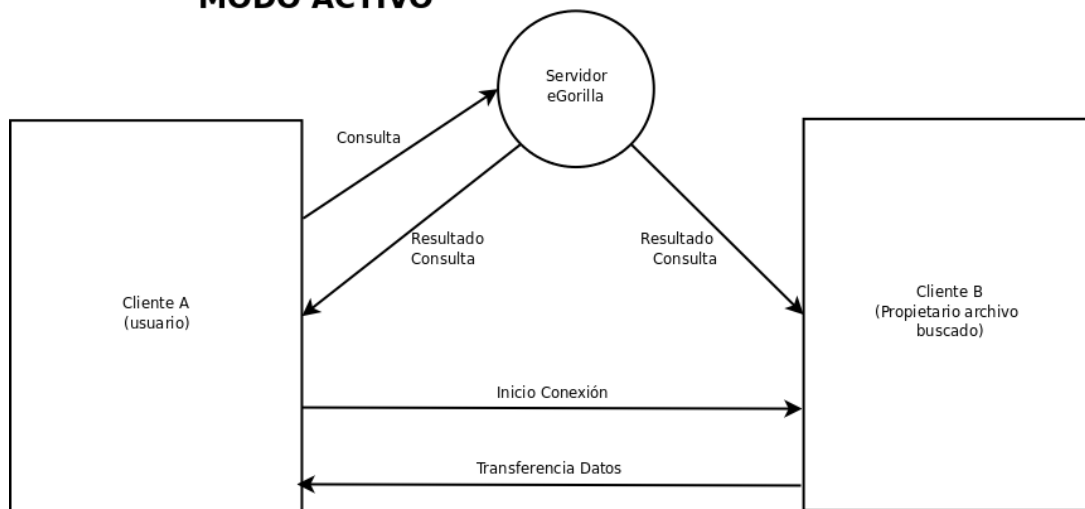
- Conexión con la red P2P. Se empleará para ello el protocolo **ProtocoloConexion**.
- Visualización del **estado de los archivos pendientes de descarga** en todo momento mediante la pestaña de descargas.
- Visualización del **estado de los archivos compartidos** en todo momento.
- Visualización de estadísticas en tiempo real del estado del sistema. 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 **ProtocoloBusquedas**.
- Visualización de la **Ayuda de la aplicación**.
- **Configuración de la aplicación** mediante un formulario de configuración. Dicha configuración **se guardará en un fichero XML de configuración**.



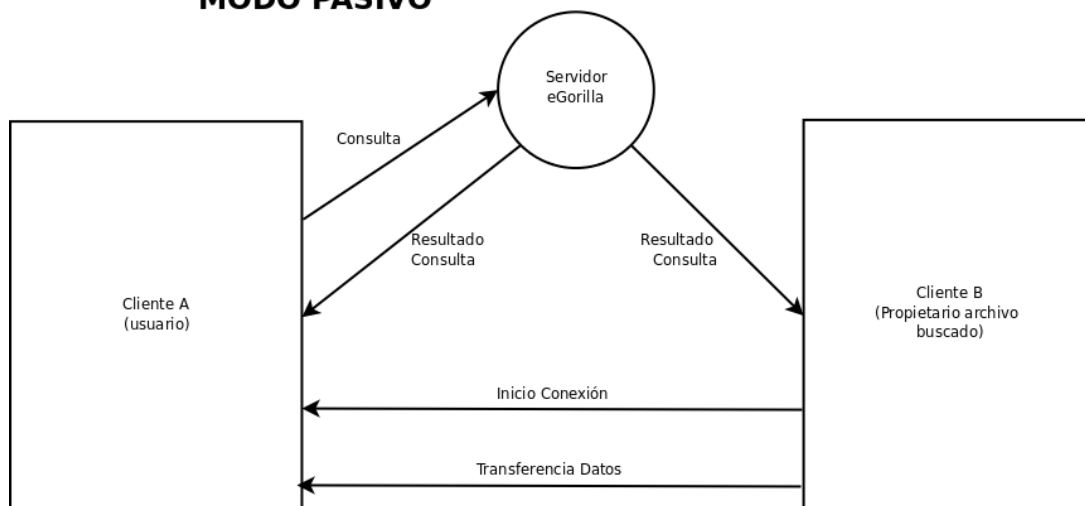


- Descarga de archivos desde otros clientes. Se podrá realizar de las siguientes maneras:
  - **Activo:** El cliente A realiza la conexión con otro cliente B y comienza a enviar los fragmentos.



**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. 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** 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**. 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.

Se empleará **jdk 1.6** y la herramienta **NetBeans 6.1**.

### 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.

