

75.42 Taller de Programación I  
TP N°5: Archivos Ubicuos  
*Grupo 04*

*“Documentación Técnica”*

# Índice

<b>1. Requerimientos de software</b>	<b>1</b>
<b>2. Descripción general</b>	<b>1</b>
<b>3. Aplicación <i>Servidor</i></b>	<b>1</b>
3.1. Clases . . . . .	1
3.2. Diagramas UML . . . . .	1
3.3. Descripción de archivos protocolos . . . . .	2
<b>4. Aplicación <i>Cliente</i></b>	<b>2</b>
4.1. Clases . . . . .	2
4.2. Diagramas UML . . . . .	2
4.3. Descripción de archivos protocolos . . . . .	2
<b>5. Aplicación <i>Monitor</i></b>	<b>2</b>
5.1. Clases . . . . .	2
5.2. Diagramas UML . . . . .	3
5.3. Descripción de archivos protocolos . . . . .	3
<b>6. Programas intermedios y de prueba</b>	<b>3</b>
<b>7. Código Fuente</b>	<b>3</b>

## 1. Requerimientos de software

Se listan a continuación los distintos requerimientos mínimos y necesarios para poder compilar, desarrollar, probar y depurar las aplicaciones que conforman al proyecto:

- *Sistemas operativos*: GNU/Linux (x86 y x86-64, distribuciones Linux basadas en RPM y DEB);
- *Controlador de versiones*<sup>1</sup>: GIT (<http://git-scm.com/>);
- *Compilador*: g++ (<http://gcc.gnu.org/>);
- *Herramientas*: Make (<http://www.gnu.org/software/make/>).

## 2. Descripción general

El proyecto se encuentra dividido en tres aplicaciones principales: *cliente*, *servidor* y *monitor*. El corazón central del servicio se encuentra establecido sobre el servidor, ya que las demás aplicaciones tienen como tarea mantenerse en contacto con este. Tal comunicación se realiza a través de sockets, permitiéndose así la ejecución remota de los programas, instanciados en distintos equipos.

Para lograr tal fin, se han identificado ciertos módulos que dotarán de distintos tipos de funcionalidades a nuestras aplicaciones. Además de ello, y no menos importante, se establecerá la existencia de un protocolo conocido por los tres entes, de manera de poder entenderse entre si. Cabe señalar que la aplicación monitor no es capaz de interactuar con la aplicación cliente, sino que simplemente todas interactúan con el servidor.

## 3. Aplicación *Servidor*

Al ser la parte central y motor del servicio, el servidor deberá constar de un conjunto de módulos los cuales permitirán dividir las distintas ocupaciones y eventos que deben atenderse en este.

Como avance de lo que será la explicación detallada del propósito a cumplir por cada clase, veamos los tipos de eventos y como son manejados a grandes rasgos por el servidor.

En primer lugar, el servidor atenderá conexiones entrantes. Estas últimas son derivadas a un ente que se encargará de mantener una interacción con el cliente que se encuentra al otro lado del canal de comunicación. El primer paso que se realiza con este usuario es el inicio de sesión. Esto le permitirá identificarse y ser derivado de acuerdo a la credencial (nombre de usuario) presentada.

Cada usuario se encuentra vinculado a un ente que denominaremos *carpeta*. Al producirse una conexión entrante, cada usuario es derivado a la carpeta que le corresponde, agrupándose así en cada uno de los directorios las conexiones vinculadas a un mismo nombre de usuario.

Llegado a este paso, es decir, una vez que una conexión es derivada a la carpeta correspondiente, se inicia la sincronización con el host de dicha comunicación. Esto se logra a través del intercambio de mensajes los cuales llegan a cada ente que representa una conexión, y los cuales son todos depositados en un ente *receptor*.

Cada carpeta posee un módulo *sincronizador* el cual se ocupa de tomar uno a uno los mensajes provenientes del receptor, los procesa y da la orden de realizar una acción en base a esto último. Entre estas acciones se encuentran: recibir un archivo, recibir una notificación de archivo eliminado, recibir partes correspondientes a un archivo modificado, etc.

Al llegar un mensaje que solicita agregar un archivo, modificarlo o eliminarlo, se deriva con dicha notificación en el módulo *manejador de archivos*, quien se encarga de lidiar con los archivos físicos.

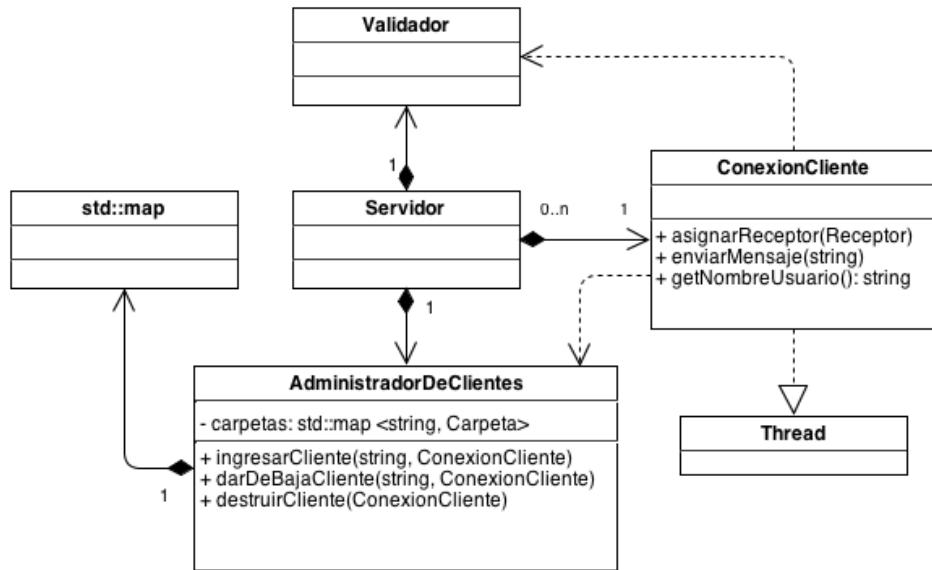
### 3.1. Clases

[ Colocar texto aquí ]

### 3.2. Diagramas UML

---

<sup>1</sup>Este requerimiento es de carácter opcional ya que solo es necesario en caso de desear clonar el proyecto desde el repositorio del grupo.



**Figura 1:** *Diagrama de clases principal del servidor.*

### 3.3. Descripción de archivos protocolos

[ Colocar texto aquí ]

## 4. Aplicación *Cliente*

[ Colocar texto aquí (descripción general)]

### 4.1. Clases

[ Colocar texto aquí ]

### 4.2. Diagramas UML

### 4.3. Descripción de archivos protocolos

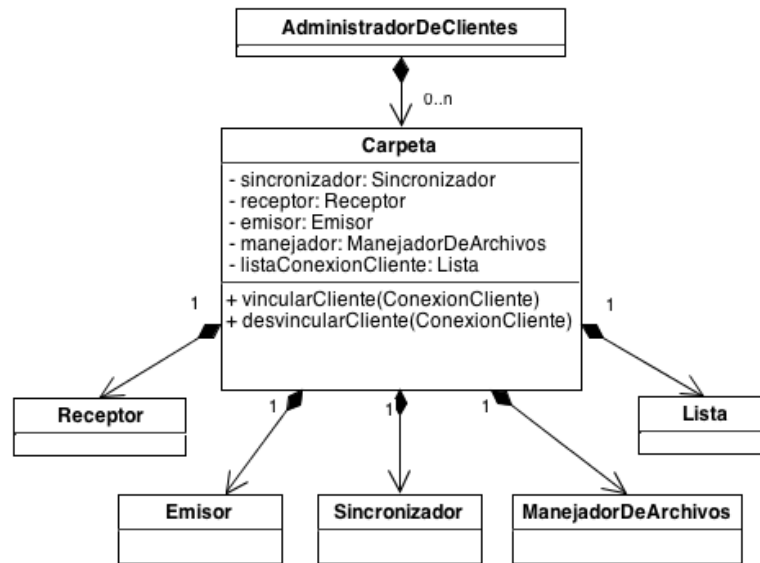
[ Colocar texto aquí ]

## 5. Aplicación *Monitor*

[ Colocar texto aquí (descripción general)]

### 5.1. Clases

[ Colocar texto aquí ]



**Figura 2:** Diagrama que muestra la jerarquía de clases principales.

## 5.2. Diagramas UML

## 5.3. Descripción de archivos protocolos

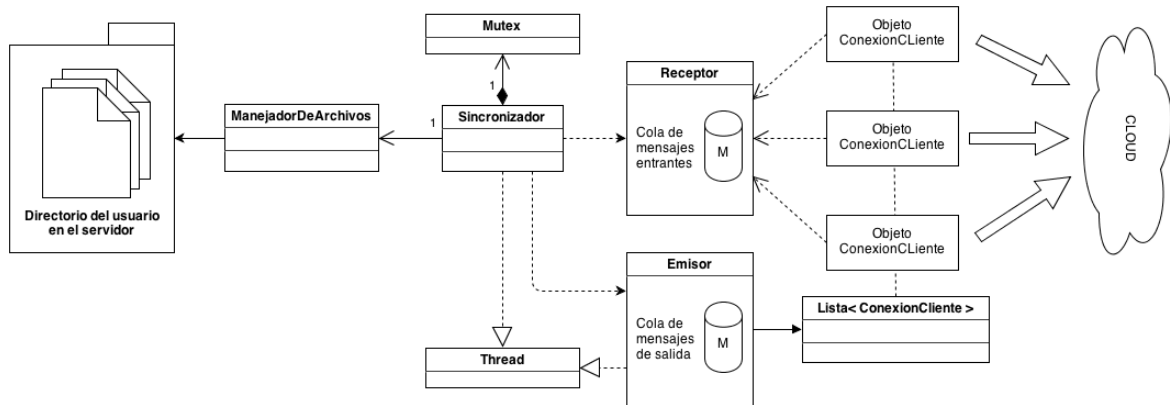
[ Colocar texto aquí ]

## 6. Programas intermedios y de prueba

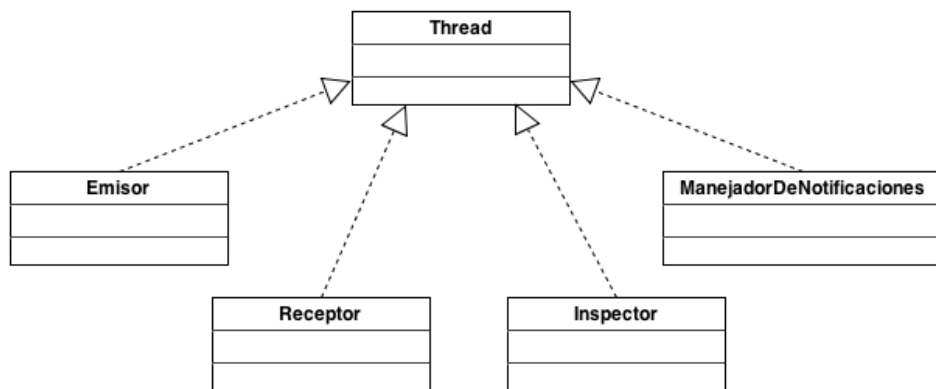
[ Colocar texto aquí ]

## 7. Código Fuente

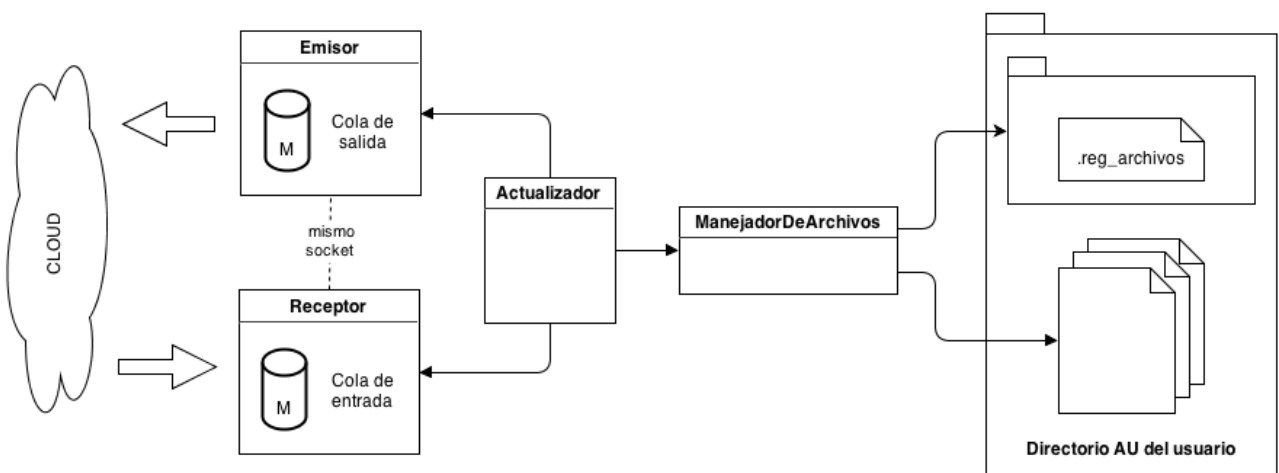
[ Colocar texto aquí ]



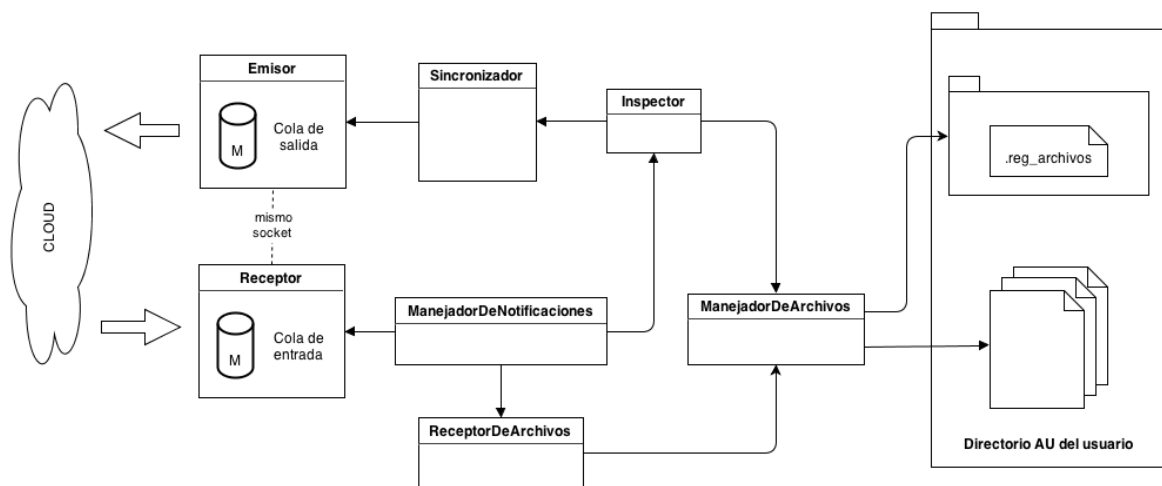
**Figura 3:** Diagrama de representación de una Carpeta con clientes conectados.



**Figura 4:** Diagrama de clases que indica aquellas que son un Thread.



**Figura 5:** Diagrama de clases que muestra de qué forma se relacionan en el proceso de actualización inicial.



**Figura 6:** Diagrama de clases que muestra de que forma se relacionan en el proceso de sincronización.