



# Universidad Complutense

Facultad de Informática



**eGorilla**

*Diseño del Sistema*

*Asignatura: Ingeniería del Software*

*Curso Académico: 2008/2009*

*Grupo: 4º B*

# Índice

<b>1. INTRODUCCIÓN .....</b>	<b>3</b>
<b>2. CONTRATOS DE OPERACIONES .....</b>	<b>4</b>
2.1. INTRODUCCIÓN.....	4
2.2. CLIENTE .....	4
2.2.1. Contrato para caso de uso conectarServidor.....	4
2.2.2. Contrato para caso de uso buscarArchivos .....	5
2.2.3. Contrato para caso de uso iniciarDescarga.....	5
2.2.4. Contrato para caso de uso consultarEstadísticas .....	6
2.3. SERVIDOR .....	7
2.3.1. Contrato para caso de uso IniciarServidor.....	7
<b>3. CASOS DE USO REALES .....</b>	<b>8</b>
3.1. CLIENTE .....	8
3.1.1. Servidores .....	8
3.1.2. ConfigurarCliente.....	9
3.1.3. BuscarArchivos.....	10
3.1.4. ConsultarEstadísticas .....	11
3.2. SERVIDOR .....	12
3.2.1. GestorDeConfiguración .....	12
<b>4. DIAGRAMAS DE DISEÑO.....</b>	<b>13</b>
4.1. DIAGRAMAS DE CLASES .....	13
4.1.1. PROTOCOLO EGORILLA .....	13
4.1.2. GESTORDERED .....	14
4.1.3. CONFIGURARCLIENTE.....	15
4.1.4. GESTORDECONFIGURACIÓN.....	16
4.2. DIAGRAMAS DE ESTADOS .....	19
<b>5. DIAGRAMA DE COMUNICACIÓN ENTRE MÓDULOS.....</b>	<b>20</b>



FACULTAD DE INFORMÁTICA	UNIVERSIDAD COMPLUTENSE DE MADRID	
Ingeniería del Software	Curso 2008/2009	Grupo: 4º B

# 1. Introducción

En este documento se procede a la descripción del **diseño** del sistema, una vez revisada casi por completo la etapa de **análisis** (siempre habrá algo por revisar correspondiente a ésta etapa) siguiendo el orden establecido por el estándar del **Proceso Unificado de Desarrollo ó RUP**.

El contenido del presente documento se puede clasificar en los siguientes conceptos:

- **Contratos de operaciones:** Se analizan y describen las operaciones más críticas del sistema.
- **Casos de uso reales:** Se analizan y describen las acciones reales producidas por el sistema mediante diagramas de casos de uso en los que se refleja precisamente dicho comportamiento real. Un claro ejemplo para ilustrar la idea al lector es el funcionamiento de la interfaz de la aplicación en la que un caso de uso concreto sería **pulsarBotónServidores**.
- **Diagramas del diseño:** Se describen los diagramas correspondientes al diseño, principalmente **diagramas de clases**, de actividad ó secuencia y de estados.
- **Diagrama de Clases de Comunicación entre Módulos:** En éste diagrama se ilustran la clase ó clases dentro de cada módulo que son las encargadas de la comunicación con el resto de los módulos de la aplicación.





## 2. Contratos de operaciones

### 2.1. Introducción

Los contratos de operaciones contribuyen a definir el comportamiento de un sistema en términos de los cambios de estado que éste experimenta cuando se invoca una operación. Éste no profundiza en detalles de funcionalidad asociadas a las operaciones invocadas sino que interpreta al sistema como una caja negra.

Éste documento estará redactado declarativamente expresándose a partir de los cambios de estado de las precondiciones y las poscondiciones.

Se describirán las operaciones de mayor relevancia para la funcionalidad principal de la aplicación, éstas estarán condicionadas por el diseño del modelo del dominio y los diagramas de secuencia del sistema.

### 2.2. Cliente

#### 2.2.1. Contrato para caso de uso *conectarServidor*

Operación	<b>conectarServidor ()</b>
<b>Responsabilidades</b>	Establece la conexión con el servidor, se identifica en él e informa de los archivos compartidos.
<b>Tipo</b>	Sistema
<b>Referencias Cruzadas</b>	Casos de uso: BuscarArchivos, IniciarDescarga, DesconectarServidor.
<b>Excepciones</b>	Si la versión del cliente no es válido, indicar que hay un error de versión.
<b>Salida</b>	Información al usuario mediante mensaje de información.
<b>Precondiciones</b>	No existe una conexión activa con ningún servidor.
<b>Postcondiciones</b>	<ul style="list-style-type: none"><li>• Fue creada una instancia de ServidorCliente.</li><li>• Fue creada una cola de peticiones de InicioDeDescarga.</li><li>• Se inicializó estadísticas de sesión.</li><li>• La conexión con el servidor quedó establecida.</li></ul>





FACULTAD DE INFORMÁTICA	UNIVERSIDAD COMPLUTENSE DE MADRID	
Ingeniería del Software	Curso 2008/2009	Grupo: 4º B

## 2.2.2. Contrato para caso de uso *buscarArchivos*

Operación	buscarArchivos (nombre, tipo)
Responsabilidades	Realiza una búsqueda de los archivos que están en la red que cumplan los requisitos de los parámetros.
Tipo	Sistema
Referencias Cruzadas	
Excepciones	Si no tiene establecida una conexión con el servidor se notificará al usuario con un error.
Salida	Información al usuario mediante una tabla con los archivos resultantes de la búsqueda.
Precondiciones	Se ha establecido conexión con el servidor.
Postcondiciones	Se creó una lista de archivos coincidentes a los parámetros buscados.

## 2.2.3. Contrato para caso de uso *iniciarDescarga*

Operación	iniciarDescarga(identificador)
Responsabilidades	Realiza una serie de peticiones para descargar el archivo solicitado.
Tipo	Sistema
Referencias Cruzadas	
Excepciones	Archivo erróneo, Muestra un mensaje de error.
Salida	Información al usuario mediante una tabla con los archivos resultantes de la búsqueda.
Precondiciones	Se ha establecido conexión con el servidor y además posee el identificador único del archivo a descargar.
Postcondiciones	<ul style="list-style-type: none"><li>Se obtuvo una lista de clientes poseedores del archivo deseado.</li><li>Se creó una lista de peticiones de las partes del archivo deseado.</li></ul>





FACULTAD DE INFORMÁTICA	UNIVERSIDAD COMPLUTENSE DE MADRID	
Ingeniería del Software	Curso 2008/2009	Grupo: 4º B

## 2.2.4. Contrato para caso de uso *consultarEstadísticas*

Operación	<b>consultarEstadísticas(parametros)</b>
Responsabilidades	Realiza una petición para obtener las estadísticas conforme a la información solicitada por el usuario.
Tipo	Sistema
Referencias Cruzadas	
Excepciones	No existen datos estadísticos, Muestra un mensaje de error.
Salida	Información al usuario mediante una gráfica (de diferente tipo: barras verticales, horizontales, pie) con los datos obtenidos.
Precondiciones	Existencia de datos para mostrar.
Postcondiciones	Se obtuvo una gráfica de con la información solicitada por el usuario.





## 2.3. Servidor

### 2.3.1. Contrato para caso de uso *IniciarServidor*

Operación	<b>iniciarServidor()</b>
<b>Responsabilidades</b>	Iniciar el sistema Servidor de la aplicación.
<b>Tipo</b>	Sistema
<b>Referencias Cruzadas</b>	
<b>Excepciones</b>	Puerto de escucha ocupado, Muestra un mensaje de error.
<b>Salida</b>	Información al usuario mediante una tabla con los archivos resultantes de la búsqueda.
<b>Precondiciones</b>	No puede haber más de un servidor eGorilla en una misma máquina ejecutándose a la vez.
<b>Postcondiciones</b>	<ul style="list-style-type: none"><li>• Se crea una instancia de Servidor.</li><li>• Se crea una instancia de ArchivosDisponibles.</li><li>• Se crea una instancia de UsuariosConectados.</li></ul>

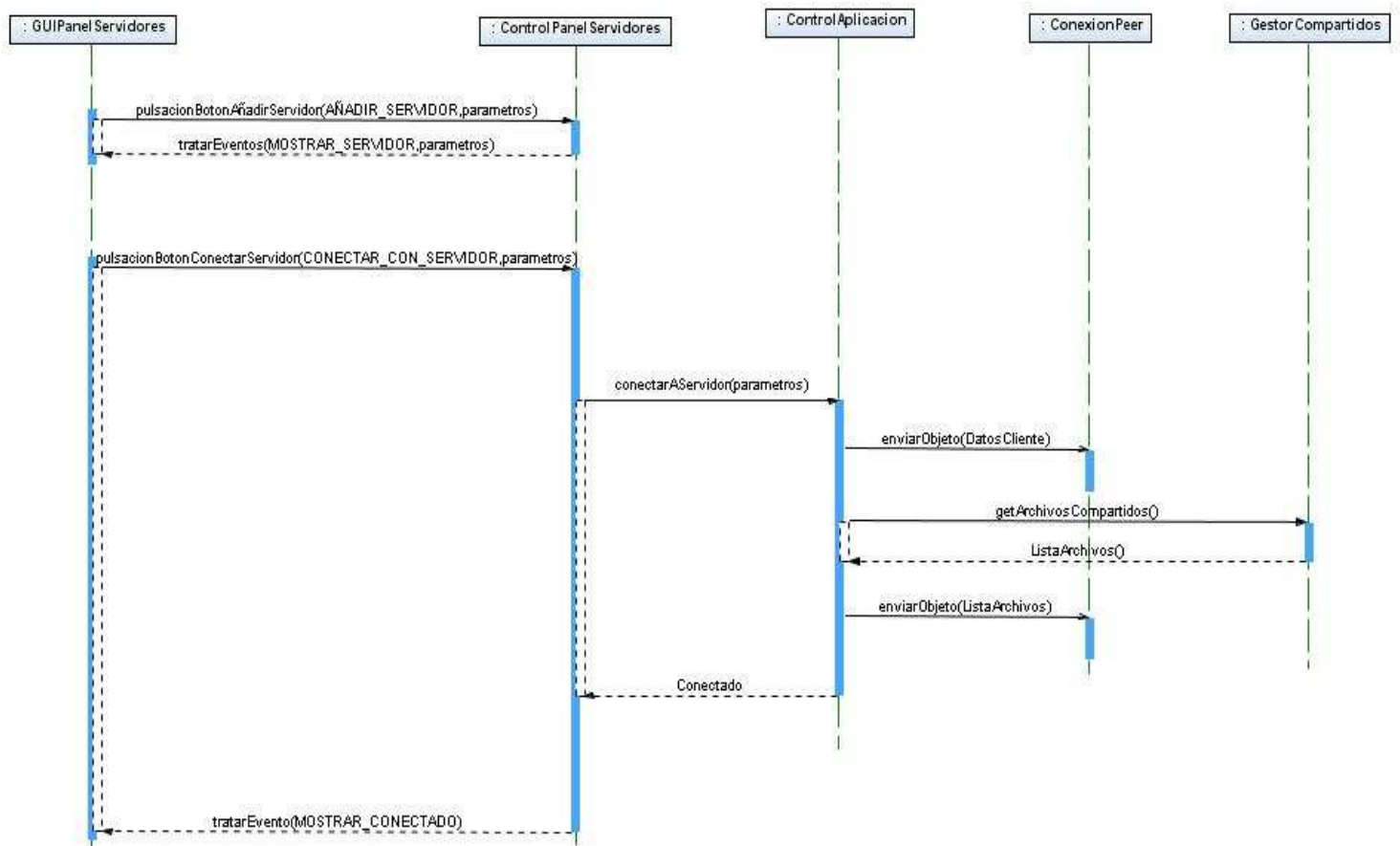




## 3. Casos de uso reales

### 3.1. Cliente

#### 3.1.1. Servidores

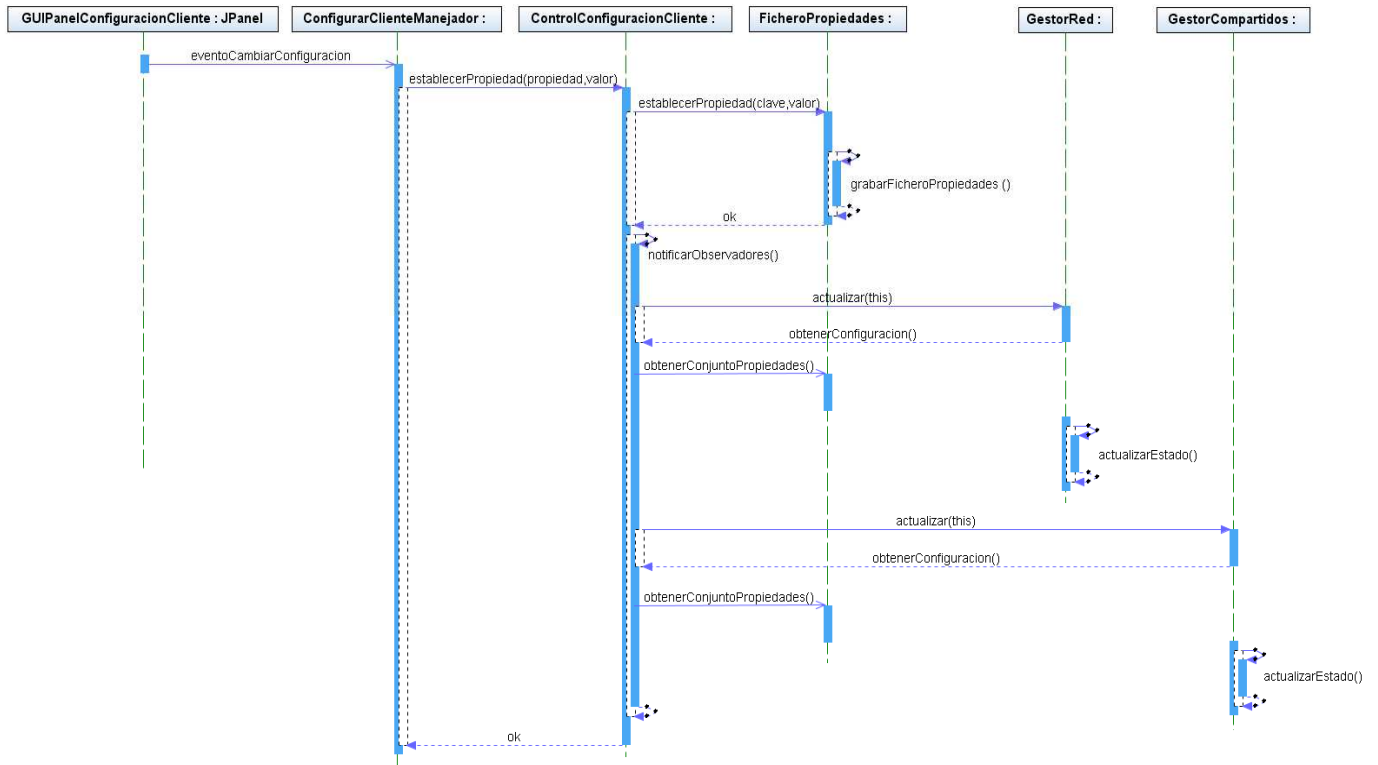




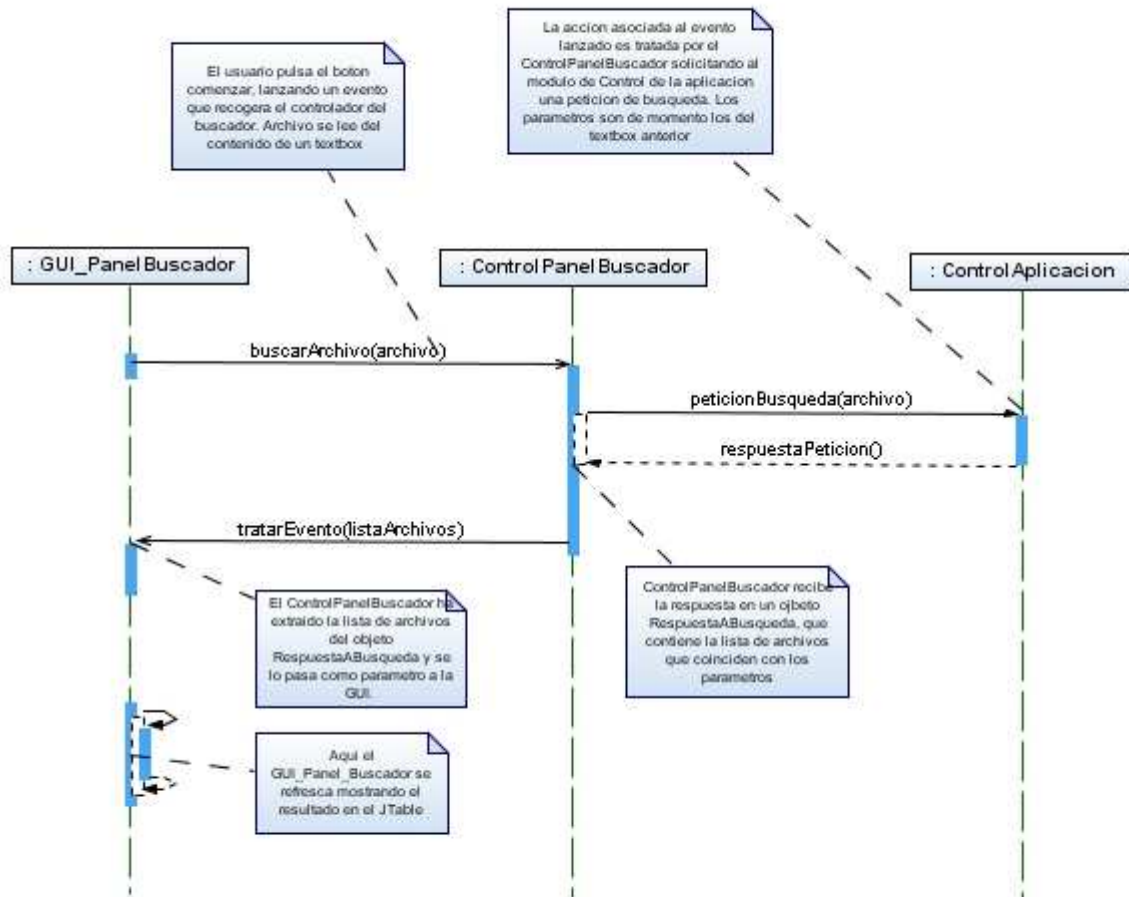


### 3.1.2. ConfigurarCliente

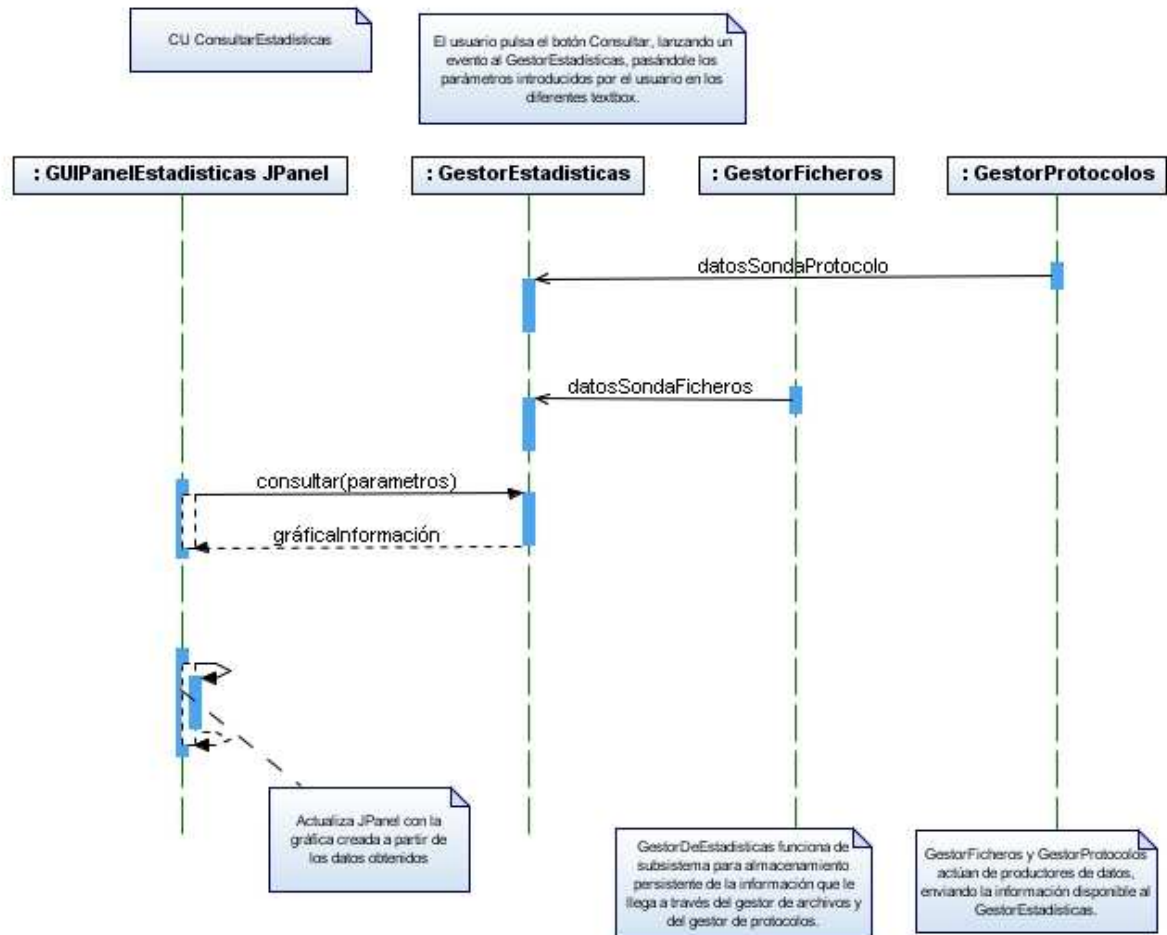
Diagrama Secuencia CU  
ConfigurarCliente:  
establecerConfiguracion



### 3.1.3. *BuscarArchivos*

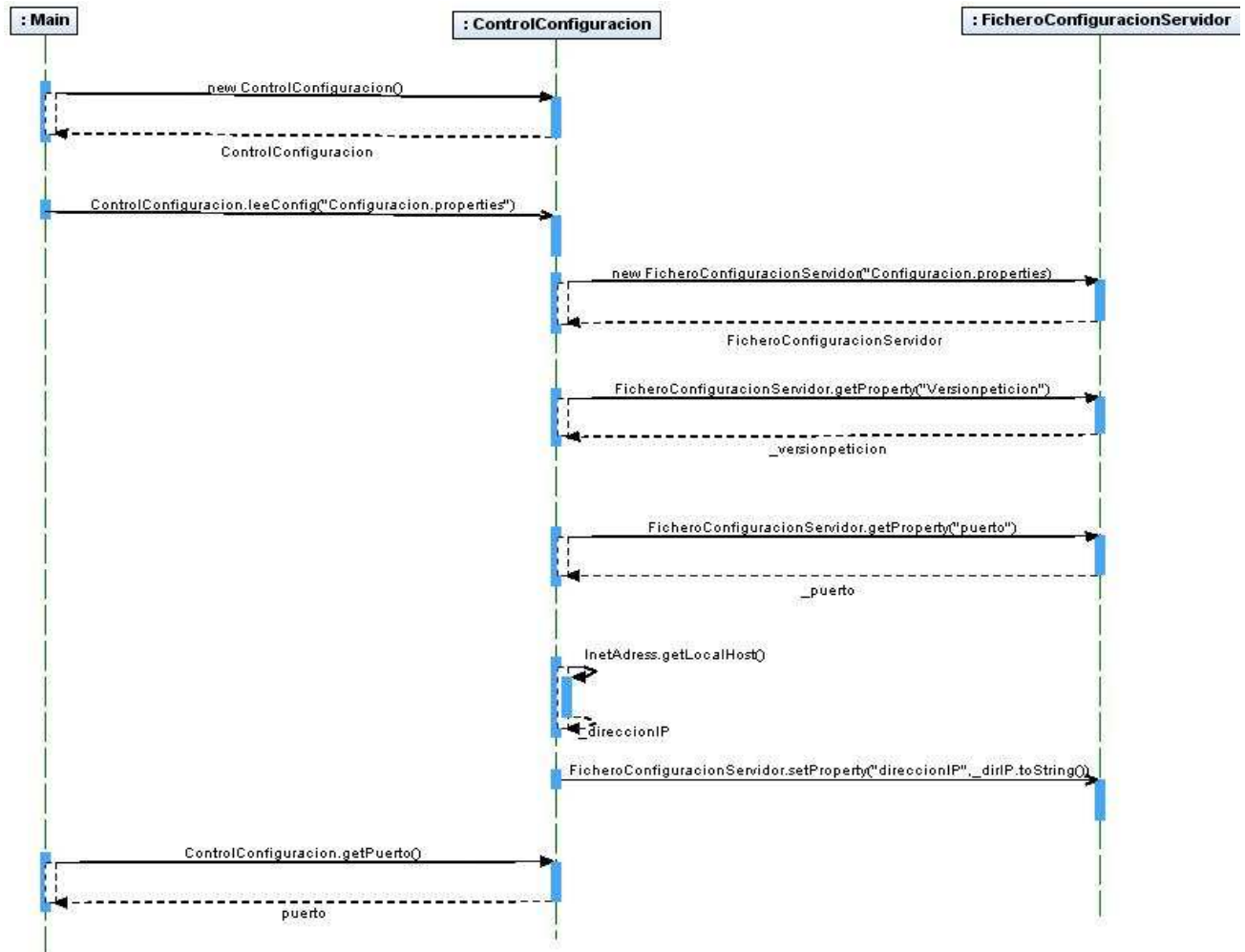


### 3.1.4. ConsultarEstadísticas



## 3.2. Servidor

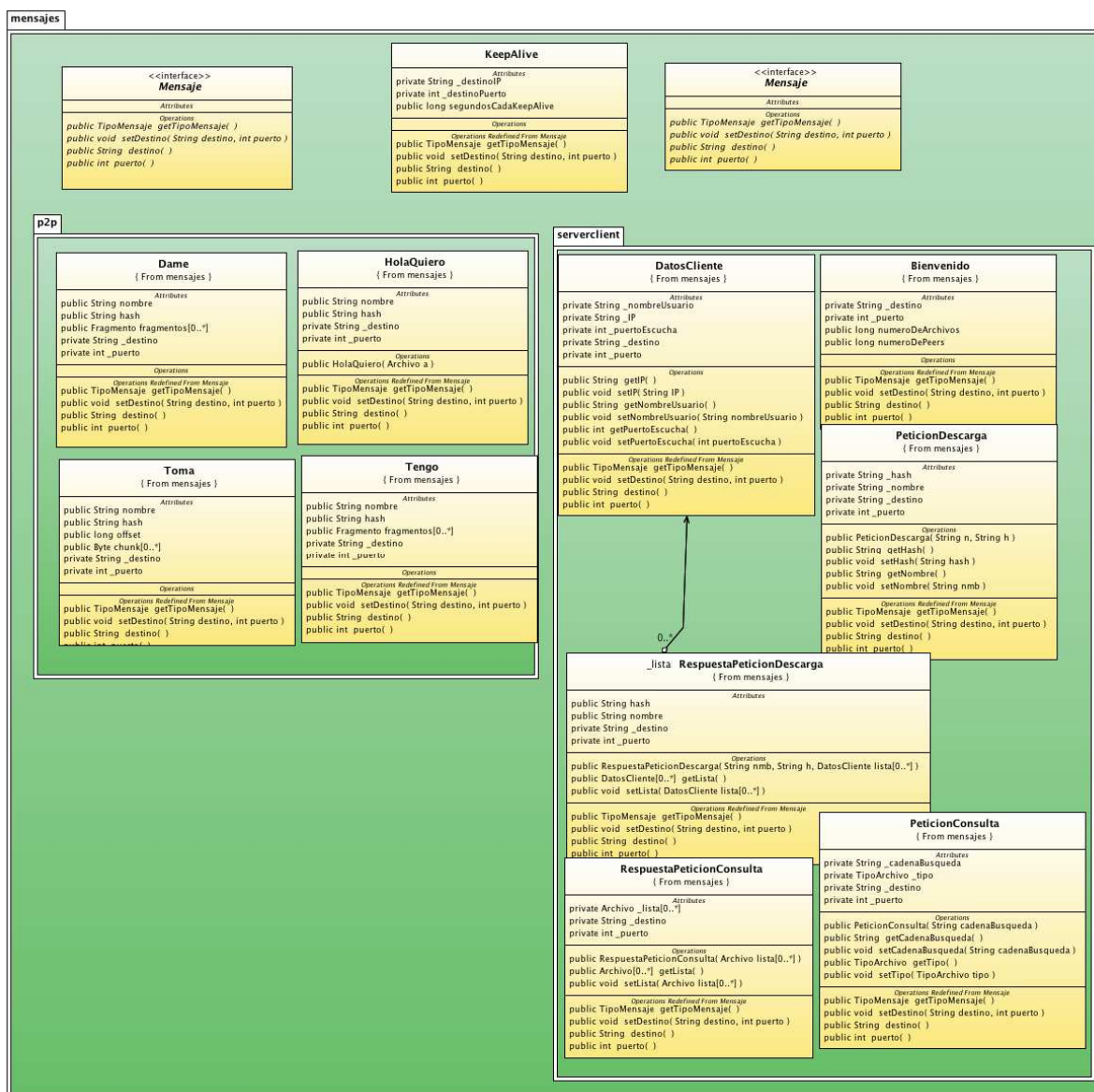
### 3.2.1. GestorDeConfiguración



## 4. Diagramas de diseño

### 4.1. Diagramas de Clases

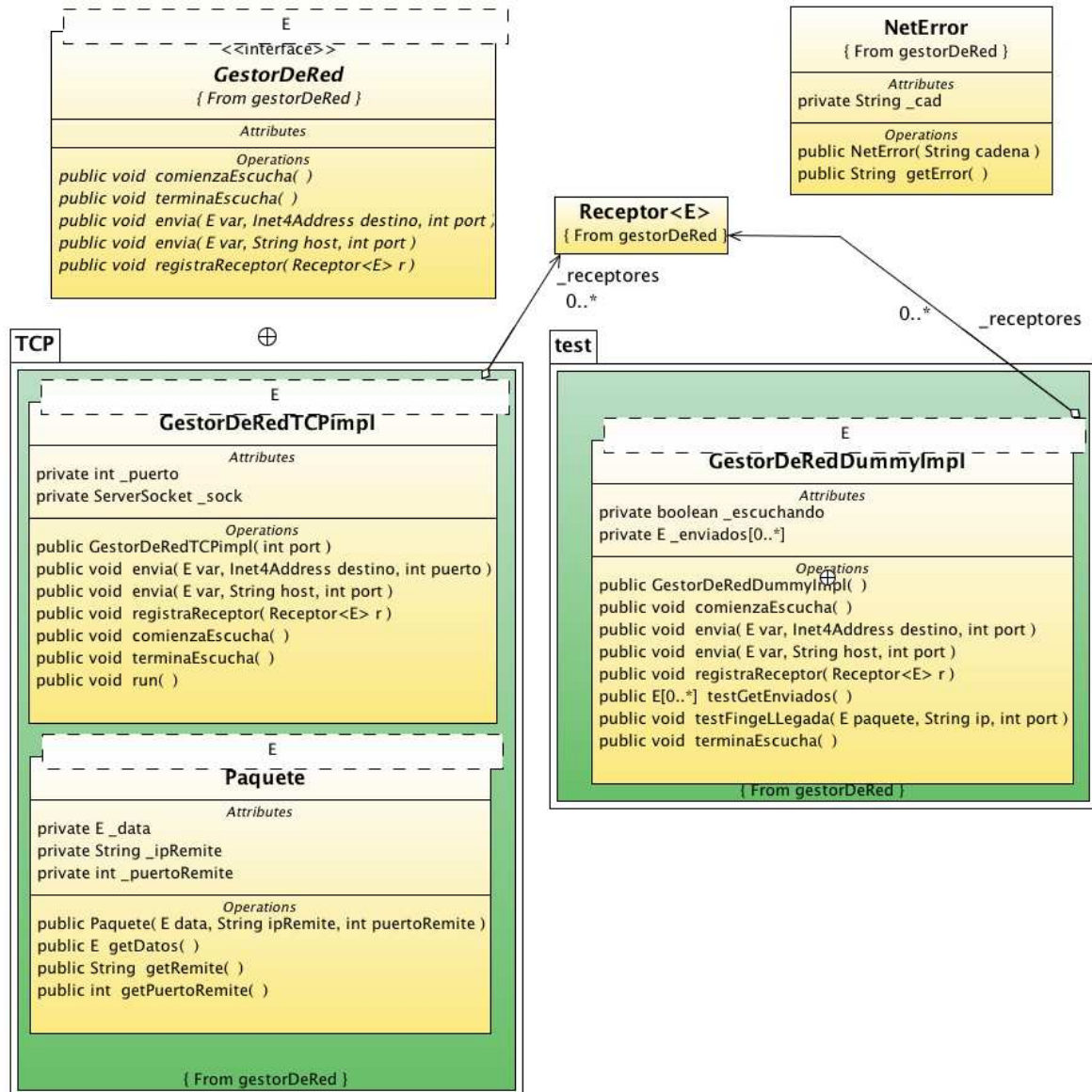
#### 4.1.1. Protocolo eGorilla



El protocolo eGorilla se comunica compartiendo mensajes ya definidos en un orden establecido. La comunicación puede ser entre dos clientes o entre un cliente y un servidor, por lo que el diálogo es diferente.



## 4.1.2. GestorDeRed



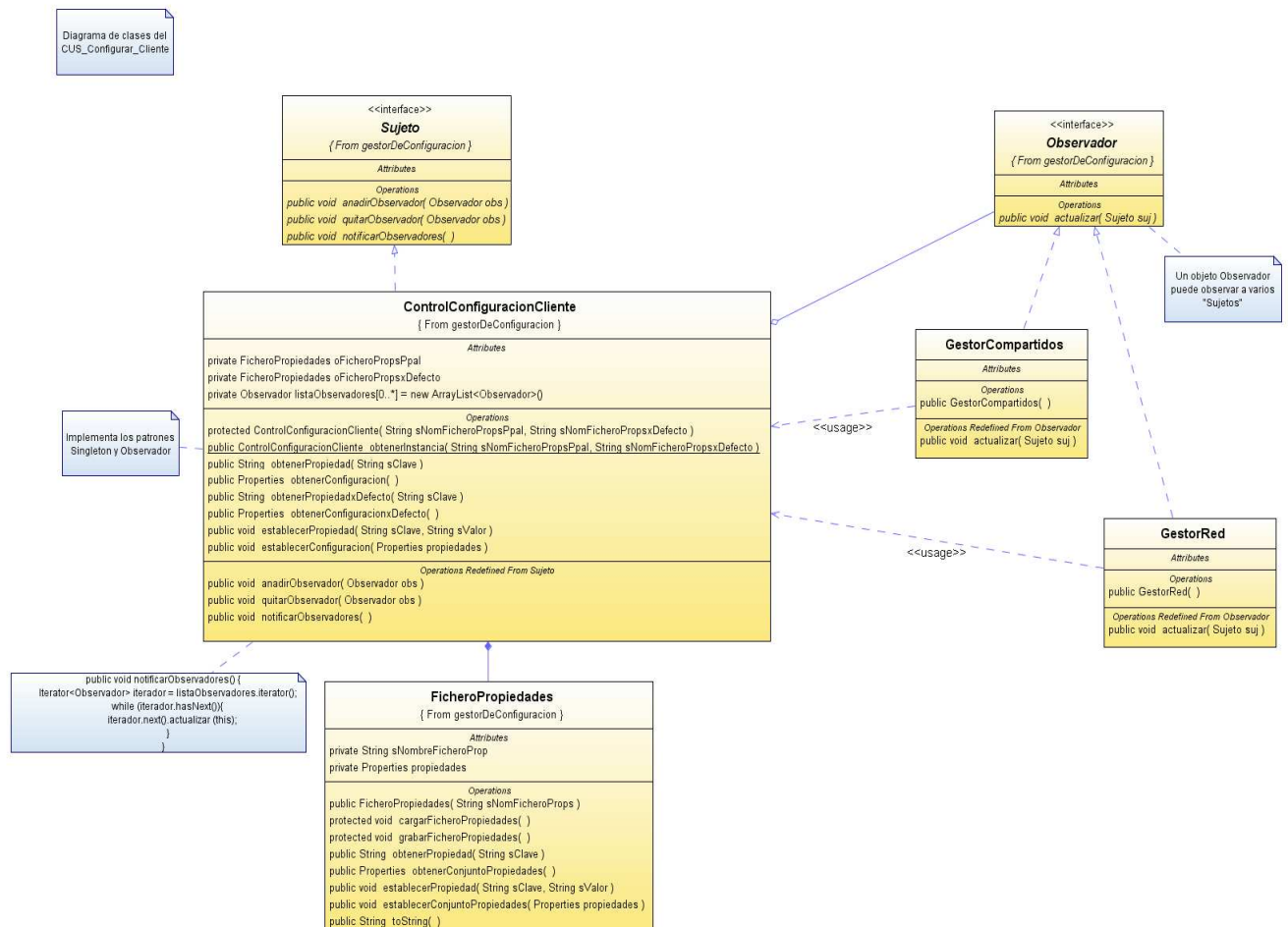
Todas las comunicaciones de red se realizan mediante una interfaz que realiza la abstracción del uso de **sockets** en la aplicación. La **parametrización** de la clase hace que solo sea posible enviar un tipo de objetos. El uso práctico se realizará instanciando la plantilla con la clase Mensaje permitiendo enviar cualquier tipo de mensajes, en nuestro caso se enviarán **mensajes eGorilla**.





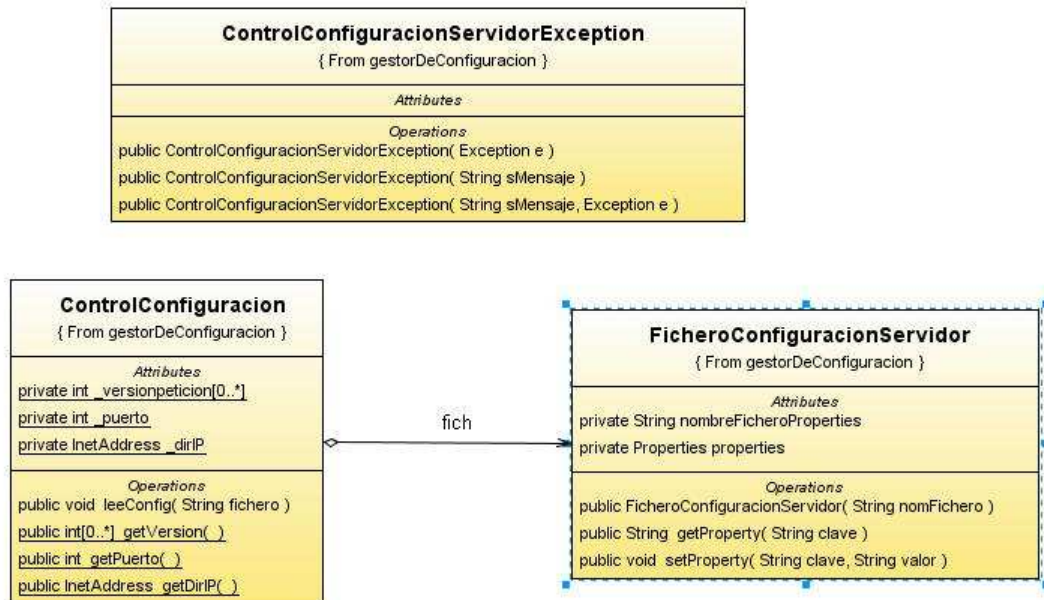
Para la realización de pruebas se ha creado una implementación del gestor que no realiza ninguna comunicación de forma que se pueden realizar pruebas sin la necesidad de montar la infraestructura de red.

### 4.1.3. ConfigurarCliente





## 4.1.4. GestorDeConfiguración





## 4.1.5. GestorDeFicheros

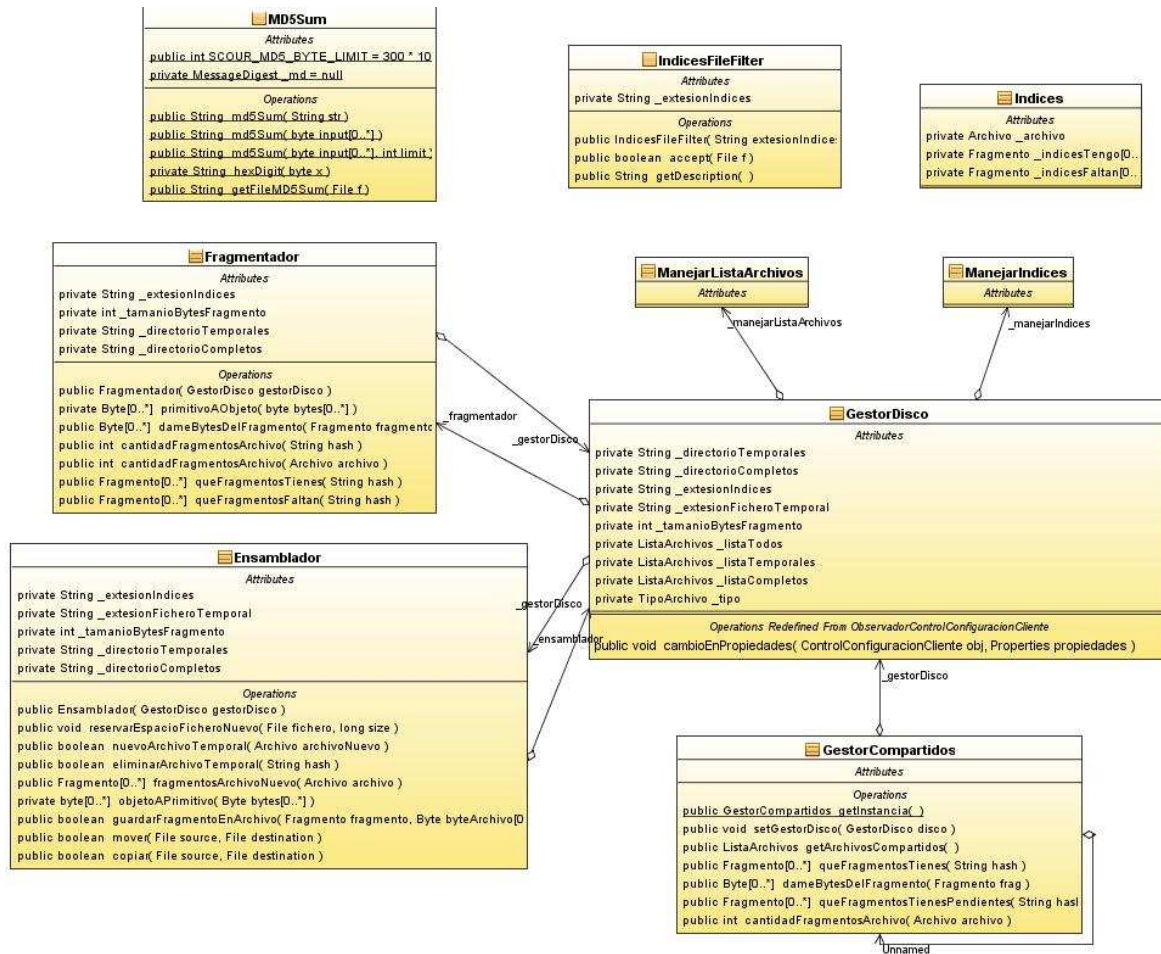


Ilustración – Diagrama de clases para el *GestorDeFicheros*.



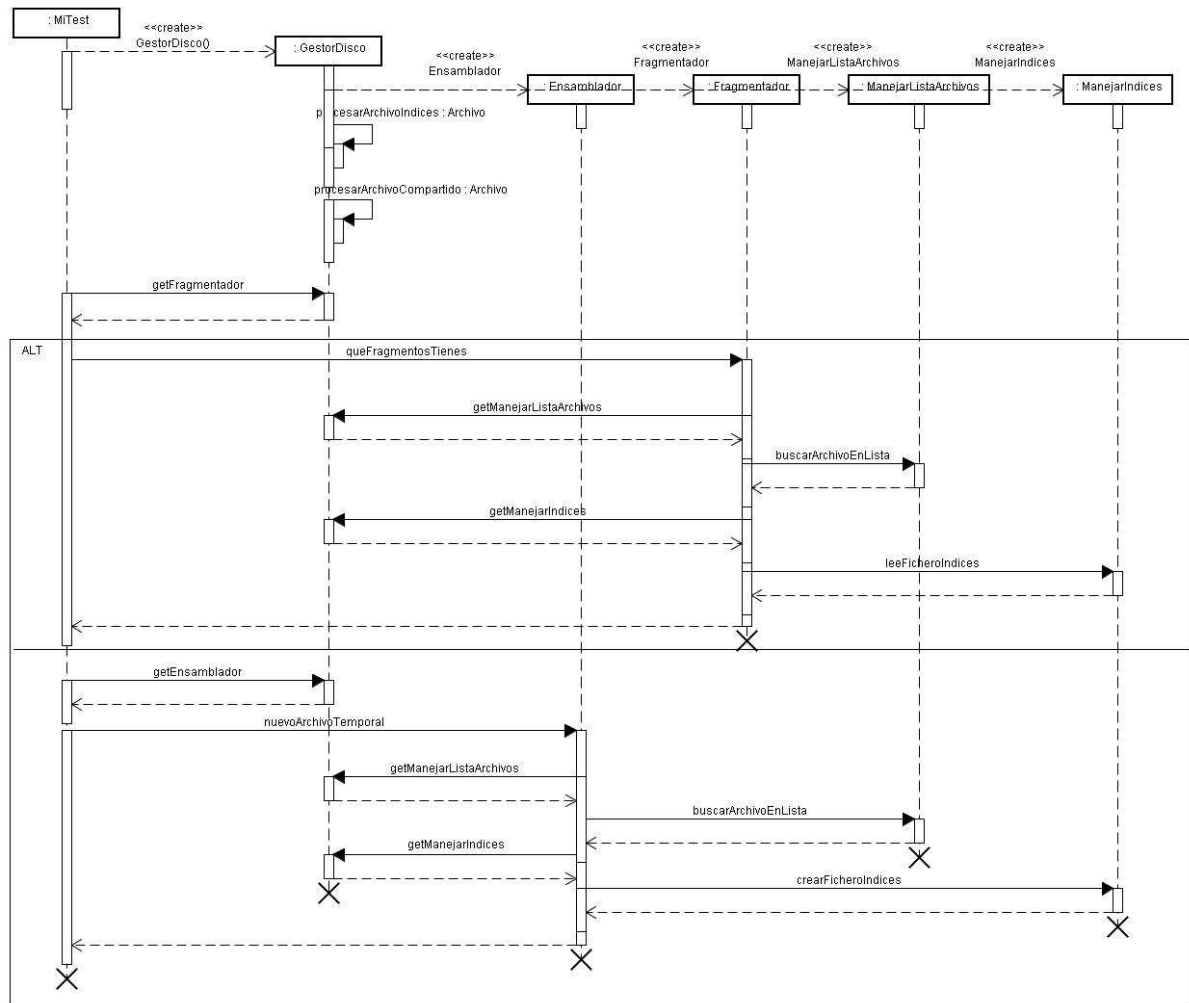


Ilustración – Diagrama de secuencia para el *GestorDeFicheros*.





## 4.2. *Diagramas de Estados*





## 5. Diagrama de comunicación entre módulos

