

Acceso a Servicios Web Basados en SOAP desde Redes de Baja Capacidad



ANEXO C

Johana Quiñonez Collazos
Víctor Garzón Marín

Director
Francisco Orlando Martínez Pabón
Magister en Ingeniería, área Ingeniería Telemática

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación de Servicios Avanzados de Telecomunicaciones
Popayán, Junio de 2012



TABLA DE CONTENIDO

ANEXO C.	ANÁLISIS Y MODELADO DEL SISTEMA MEDIADOR	1
C.1.	DIAGRAMAS DE CASO DE USO DEL SISTEMA	1
C.1.1.	Actores del Sistema	1
C.1.2.	Descripción de Casos de Uso	2
C.2.	DIAGRAMAS DE PAQUETES	5
C.2.1.	Mediador del lado del proveedor: Servidor	5
C.2.2.	Mediador del lado del consumidor: Cliente	6
C.2.3.	Motor de Compresión: CompressionMotor.jar	6
C.2.4.	Gestor de Metadatos: MetadataAnalyzer.jar	7
C.3.	DIAGRAMAS DE CLASES.....	7
C.3.1.	Mediador del lado del proveedor: Servidor	7
C.3.2.	Mediador del lado del consumidor: Cliente	8
C.3.3.	Motor de Compresión: CompressionMotor.jar	9
C.3.4.	Gestor de Metadatos: MetadataAnalyzer.jar	9



LISTA DE FIGURAS

FIGURA C-1 CASOS DE USO: CONSUMIDOR DEL SERVICIO	1
FIGURA C-2 CASOS DE USO: ADMINISTRADORCLIENTE	2
FIGURA C-3 CASOS DE USO: ADMINISTRADORSERVIDOR.....	2
FIGURA C-4 DIAGRAMA DE PAQUETES: SERVIDOR	5
FIGURA C-5 DIAGRAMA DE PAQUETES: CLIENTE	6
FIGURA C-6 DIAGRAMA DE PAQUETES: MOTOR DE COMPRESIÓN	6
FIGURA C-7 DIAGRAMA DE PAQUETES: GESTOR DE METADATOS.....	7
FIGURA C-8 DIAGRAMA DE CLASES: SERVIDOR	7
FIGURA C-9 DIAGRAMA DE CLASES: CLIENTE	8
FIGURA C-10 DIAGRAMA DE CLASES: MOTOR DE COMPRESIÓN	9
FIGURA C-11 DIAGRAMA DE CLASES: GESTOR DE METADATOS.....	9



ANEXO C. ANÁLISIS Y MODELADO DEL SISTEMA MEDIADOR

El propósito de éste anexo, es realizar un análisis de implementación del sistema mediador. Dicho análisis incluye: diagrama de casos de uso, que permitirá determinar las acciones que los usuarios pueden realizar con el sistema; y diagramas de paquetes y clases, con los que se enseña la estructura interna del código fuente del mediador, tanto del lado del consumidor (Cliente), como del lado del proveedor (Servidor).

C.1. DIAGRAMAS DE CASO DE USO DEL SISTEMA

C.1.1. Actores del Sistema

Consumidor del Servicio: como su nombre lo indica es la persona que realiza el consumo de un servicio Web. Este consumo genera en el mediador del lado del consumidor un proceso de captura de la petición SOAP del consumidor, su envío al mediador del lado del proveedor y todo el procesamiento de la respuesta SOAP resultante, que garantiza el consumo efectivo del servicio.

Administrador del Servidor: este actor debe encargarse de nutrir el repositorio de metadatos con la información de los clientes, necesaria para la selección del mecanismo de compresión. También configurar algunos parámetros utilizados por el Servidor.

Administrador del Cliente: su función es configurar los parámetros utilizados por el cliente para su funcionamiento.

Figura C-1 Casos de uso: Consumidor del servicio

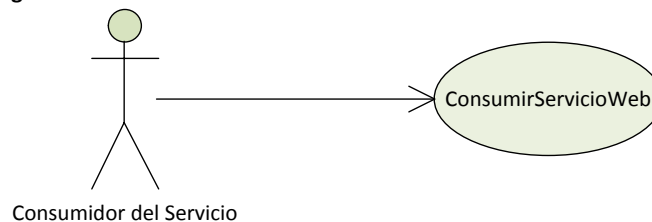




Figura C-2 Casos de uso: AdministradorCliente

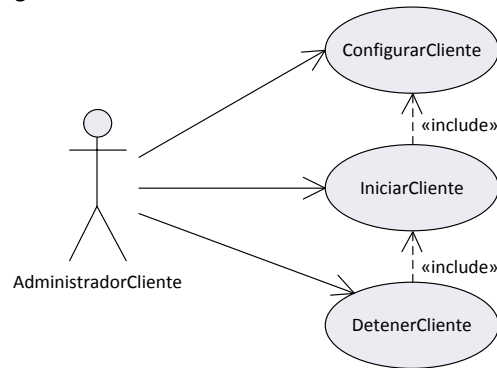
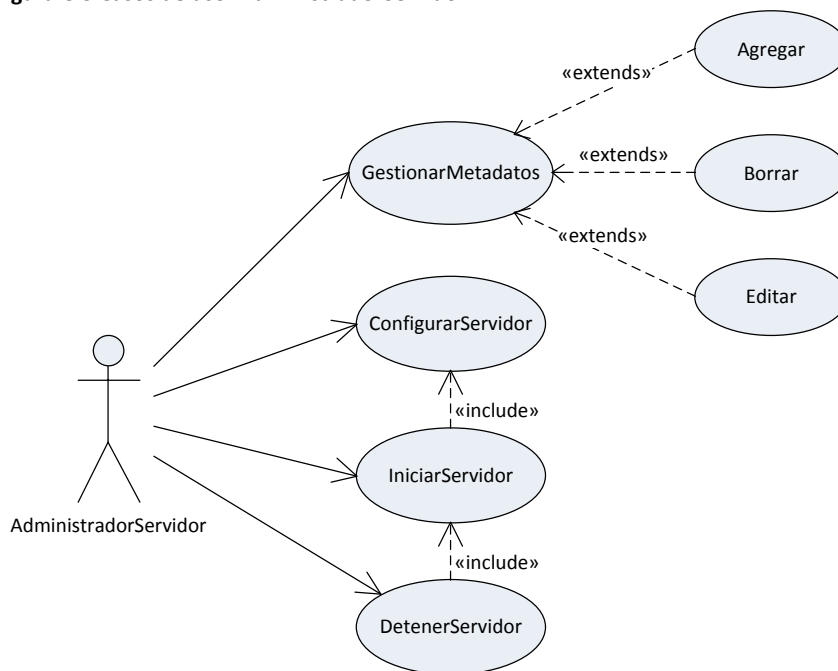


Figura C-3 Casos de uso: AdministradorServidor



C.1.2. Descripción de Casos de Uso

Caso de Uso:	Consumir Servicio Web
Actores:	Consumidor del Servicio
Tipo:	primario
Descripción:	Permite realizar el consumo de un servicio Web
Precondición:	ninguna
Flujo Principal:	<ol style="list-style-type: none"> 1. El consumidor ejecuta un cliente del servicio Web, configurado de forma tal que accede al proveedor del servicio a través del sistema mediador.
Flujo Alternativo:	ninguno



Caso de Uso:	Gestionar Metadatos
Actores:	Administrador del Servidor
Tipo:	primario
Descripción:	Permite la gestión (creación, edición y eliminación) de recursos en el repositorio de metadatos.
Precondición:	ninguna
Flujo Principal:	
<ol style="list-style-type: none"> 1. El Administrador del Servidor presiona el botón “Gestión de Metadatos” en la interfaz principal. 2. El sistema despliega la GUI para gestión de metadatos. 3. El Administrador del Servidor ingresa la dirección IP del recurso que desea gestionar. 4. El sistema determina si dicho recurso ya existe y habilita los botones “Editar” y “Eliminar”. De no existir el recurso, habilita el botón “Crear”. 5. El Administrador presiona en los botones habilitados. 6. Si el Administrador del Servidor presionó el botón “Crear” o “Editar”, y todos los campos fueron llenados correctamente, el Sistema realiza la acción solicitada. 	
Flujo Alternativo:	
<ul style="list-style-type: none"> • Si en 6 el Administrador del Servidor no ha llenado correctamente los campos, el Sistema despliega una advertencia y solicita la corrección de los datos. • Si en 6 el Administrador del Servidor selecciona “Borrar” el sistema elimina el recurso del repositorio. 	

Caso de Uso:	Configurar el Servidor
Actores:	Administrador del Servidor
Tipo:	primario
Descripción:	Permite la configuración de los parámetros de entrada del Servidor.
Precondición:	ninguna
Flujo Principal:	
<p>El Administrador a partir de la interfaz GUI principal, configura parámetros como: la ruta donde serán almacenadas las peticiones y respuestas, el mecanismo de compresión a utilizar (parámetro incluido solo para el desarrollo de pruebas, es recomendable dejar que el mediador decida el mecanismo de compresión) y los puertos de comunicación utilizados.</p> <ol style="list-style-type: none"> 1. El Administrador desbloquea la opción que desea configurar utilizando el botón de bloqueo en frente de cada una de ellas. 2. El Administrador modifica la opción que desea configurar. 3. El Administrador bloquea de nuevo la opción que desea configurar. 4. El Sistema modifica el parámetro editado por el Administrador. 	
Flujo Alternativo:	
ninguno	

Caso de Uso:	Iniciar el Servidor
Actores:	Administrador del Servidor
Tipo:	primario
Descripción:	Permite iniciar la escucha de peticiones provenientes de los Clientes.



Precondición:	Configurar correctamente todos los parámetros del Servidor.
Flujo Principal:	<ol style="list-style-type: none"> 1. El Administrador del Servidor presiona el botón “Iniciar Servidor” en la GUI principal. 2. El Sistema inicia todos los hilos de procesamiento de colas del Servidor.
Flujo Alternativo:	ninguno

Caso de Uso:	Detener el Servidor
Actores:	Administrador del Servidor
Tipo:	primario
Descripción:	Permite detener la escucha de peticiones provenientes de los Clientes y el envío de respuestas.
Precondición:	Iniciar el Servidor
Flujo Principal:	<ol style="list-style-type: none"> 1. El Administrador del Servidor presiona el botón “Detener Servidor” en la GUI principal. 2. El Sistema detiene todos los hilos de procesamiento de colas del Servidor.
Flujo Alternativo:	ninguno

Caso de Uso:	Configurar el Cliente
Actores:	Administrador del Cliente
Tipo:	primario
Descripción:	Permite la configuración de los parámetros de entrada del Cliente.
Precondición:	ninguna
Flujo Principal:	<p>El Administrador a partir de la interfaz GUI principal, configura parámetros como: la ruta donde serán almacenadas las peticiones y respuestas y los puertos de comunicación utilizados.</p> <ol style="list-style-type: none"> 1. El Administrador desbloquea la opción que desea configurar utilizando el botón de bloqueo en frente de cada una de ellas. 2. El Administrador modifica la opción que desea configurar. 3. El Administrador bloquea de nuevo la opción que desea configurar. 4. El Sistema modifica el parámetro editado por el Administrador.
Flujo Alternativo:	ninguno

Caso de Uso:	Iniciar el Cliente
Actores:	Administrador del Cliente
Tipo:	primario
Descripción:	Permite iniciar la escucha de peticiones provenientes de los Consumidores.
Precondición:	Configurar correctamente todos los parámetros del Cliente.
Flujo Principal:	<ol style="list-style-type: none"> 1. El Administrador del Cliente presiona el botón “Iniciar Cliente” en la GUI principal. 2. El Sistema inicia todos los hilos de procesamiento de colas del Cliente.



Flujo Alternativo: ninguno

Caso de Uso:	Detener el Cliente
Actores:	Administrador del Cliente
Tipo:	primario
Descripción:	Permite detener la escucha de peticiones provenientes de los Consumidores y la escucha de respuestas provenientes del Servidor.
Precondición:	Iniciar el Cliente
Flujo Principal:	<ol style="list-style-type: none"> 1. El Administrador del Cliente presiona el botón “Detener Cliente” en la GUI principal. 2. El Sistema detiene todos los hilos de procesamiento de colas del Cliente.
Flujo Alternativo: ninguno	

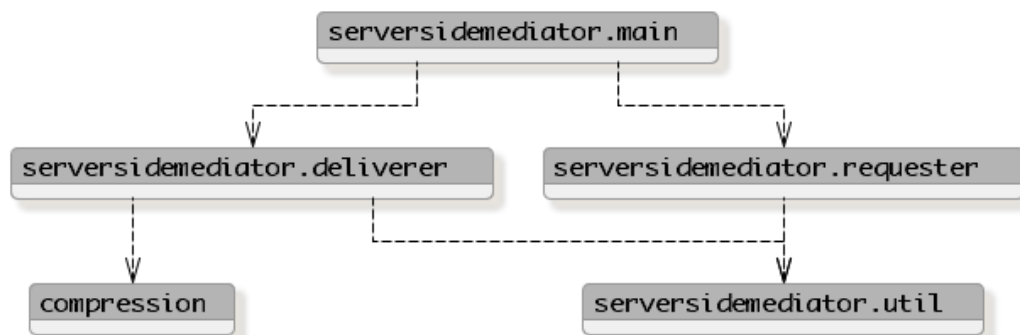
C.2. DIAGRAMAS DE PAQUETES

En esta sección se presentan los componentes generales de cada una de las aplicaciones desarrolladas.

C.2.1. Mediador del lado del proveedor: Servidor

La Figura C-4 presenta el diagrama de paquetes general del Servidor. El código fuente del Servidor se encuentra estructurado en 5 paquetes de la siguiente manera:

Figura C-4 Diagrama de paquetes: Servidor



- **main:** el paquete *main*, contiene las clases necesarias para iniciar el servidor. Tanto desde sus interfaces gráficas, que permiten realizar la configuración del Servidor; como desde una consola con la configuración por defecto.
- **requester:** contiene las clases necesarias para el tratamiento de solicitudes y el envío de las mismas hacia el proveedor.

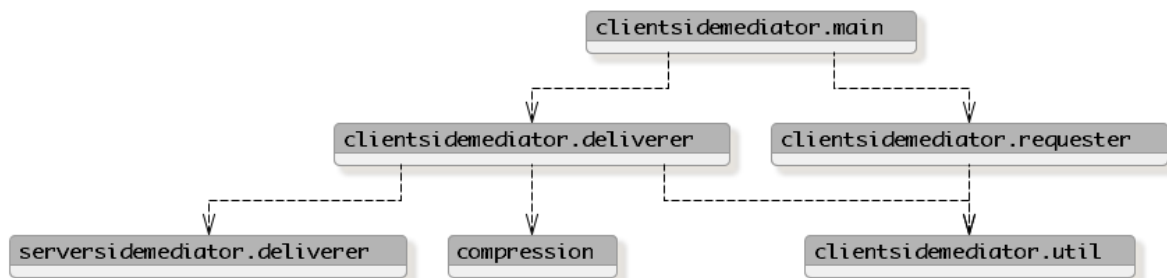


- **deliverer:** en éste paquete se realiza el procesamiento de las respuestas SOAP provenientes del proveedor del servicio. Utiliza el paquete *compression* para realizar todos los procesos de compresión.
- **compression:** este paquete tiene una conexión directa con la librería *CompressionMotor.jar*, la cual, implementa clases que utilizan las herramientas de compresión empleadas en el proyecto.
- **util:** en este paquete se almacenan clases utilizadas dentro de todo el proyecto, en el se encuentran, los parámetros de configuración del Servidor y la implementación de las colas.

C.2.2. Mediador del lado del consumidor: Cliente

La Figura C-5 presenta el diagrama de paquetes utilizado en el Cliente. El Cliente y el Servidor han sido diseñados siguiendo el mismo esquema arquitectural, es por esto, que en ambos se pueden encontrar los mismos paquetes. La principal diferencia entre el Servidor y el Cliente, es que en éste último se ha agregado un paquete llamado *serversidemediator.deliverer*, en él se ha incluido la clase *ResponseFile* utiliza para mapear los objetos serializados provenientes del servidor.

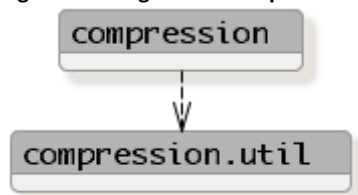
Figura C-5 Diagrama de paquetes: Cliente



C.2.3. Motor de Compresión: CompressionMotor.jar

La librería *CompressionMotor.jar* fue desarrollada para facilitar la utilización de las herramientas de compresión dentro del Cliente y el Servidor. La Figura C-6 presenta el diagrama de paquetes general que representa la estructura de la librería.

Figura C-6 Diagrama de Paquetes: Motor de Compresión



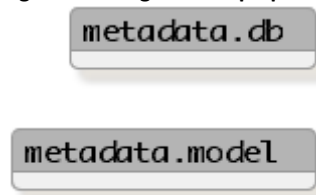


Este diagrama se encuentra distribuido en dos paquetes: *compression*, que contiene las clases necesarias para utilizar las herramientas de compresión y *compression.util*, donde se encuentran parámetros como las extensiones utilizadas por las herramientas de compresión o identificadores que permiten determinar la herramienta utilizada.

C.2.4. Gestor de Metadatos: MetadataAnalyzer.jar

Esta librería fue desarrollada para facilitar el trabajo con los metadatos. Contiene las clases para establecer la conexión con la base de datos utilizando JDBC, clases para el mapeo de los equipos y sus características, y clases para la gestión de recursos, es decir, creación, edición y eliminación de los mismos. La Figura C-7 presenta el diagrama de paquetes de la librería.

Figura C-7 Diagrama de paquetes: Gestor de Metadatos

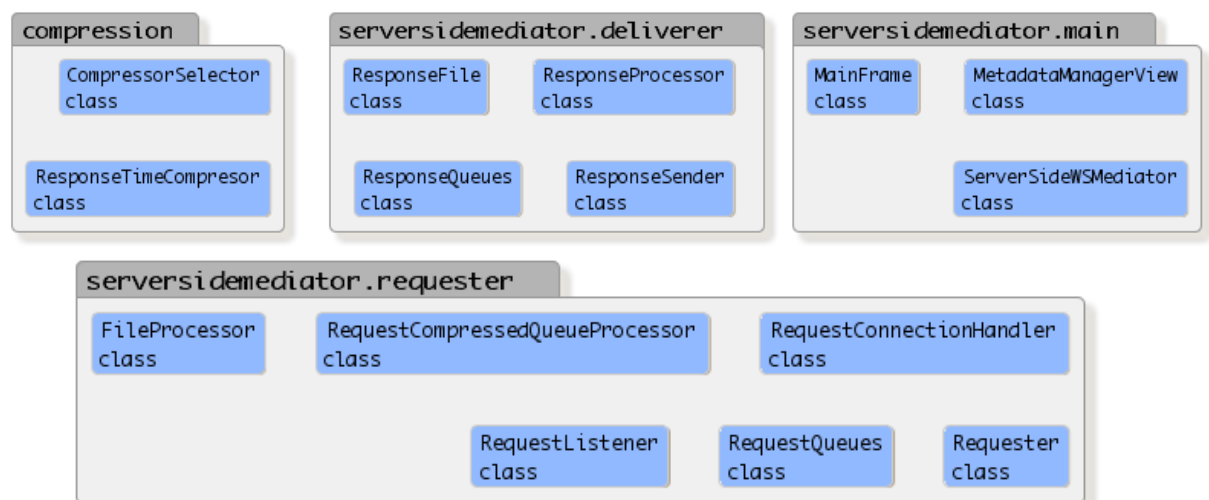


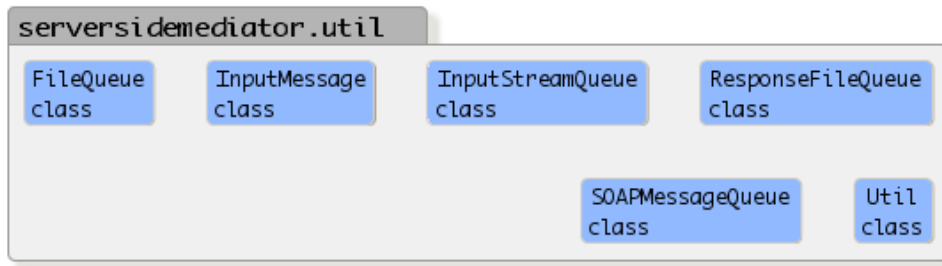
C.3. DIAGRAMAS DE CLASES

A continuación se presentan las clases utilizadas en cada una de las aplicaciones desarrolladas. La explicación de cada una de las clases puede ser encontrada en la documentación incluida con los ejecutables.

C.3.1. Mediador del lado del proveedor: Servidor

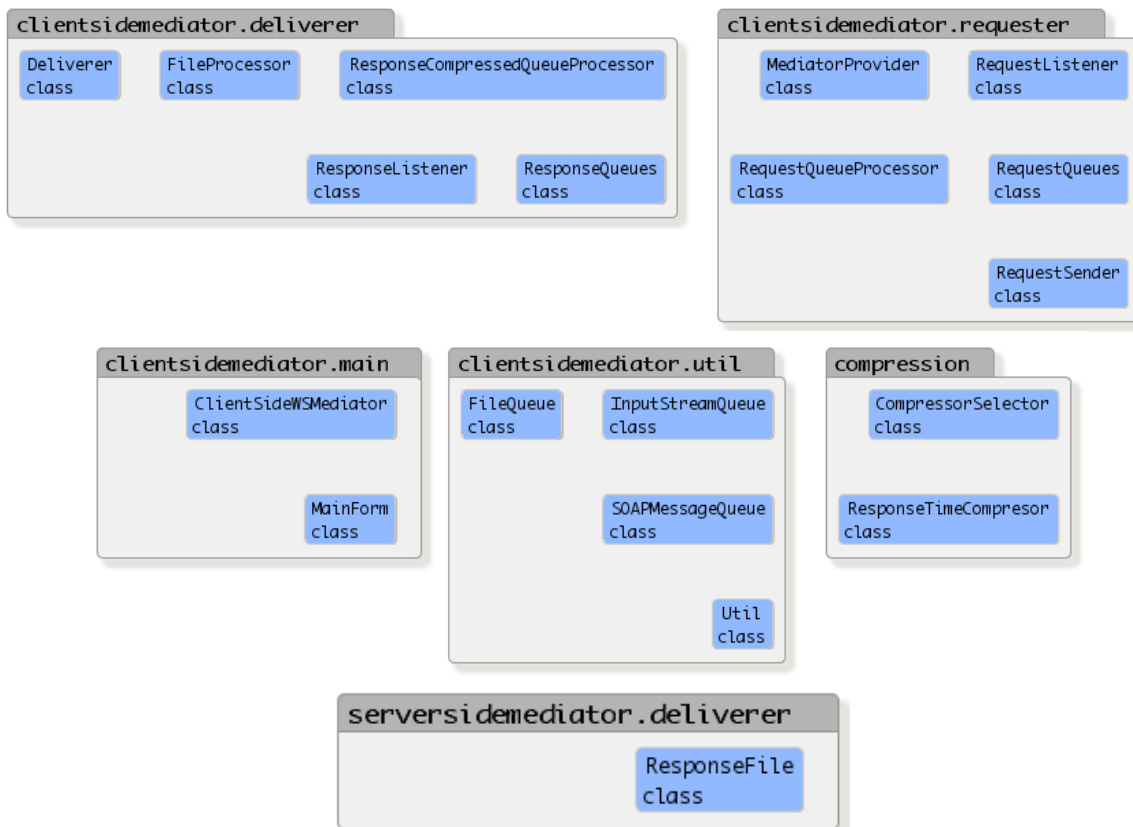
Figura C-8 Diagrama de Clases: Servidor





C.3.2. Mediador del lado del consumidor: Cliente

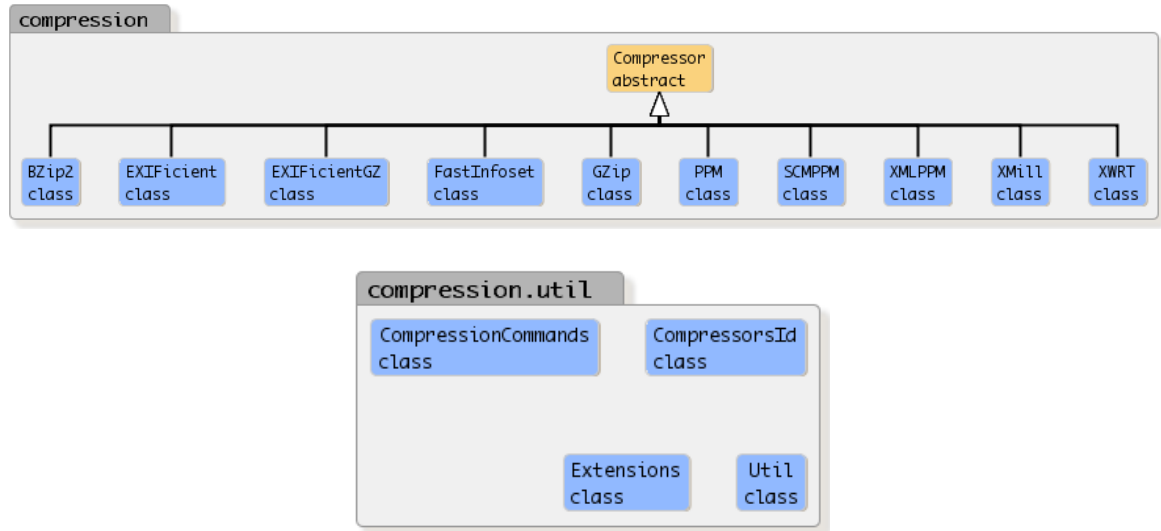
Figura C-9 Diagrama de Clases: Cliente





C.3.3. Motor de Compresión: CompressionMotor.jar

Figura C-10 Diagrama de clases: Motor de Compresión



C.3.4. Gestor de Metadatos: MetadataAnalyzer.jar

Figura C-11 Diagrama de Clases: Gestor de Metadatos

