

ANEXO A

Puentes Java-COM

En este documento se presenta una comparación de los diferentes puentes Java-COM, así como también los aspectos más importantes de los mismos. Además, se hace una descripción detallada de J-Integra, la herramienta seleccionada para desarrollar este proyecto, y de Ja.NET, que por sus características puede ser de gran importancia en futuros desarrollos.

A continuación se presenta una descripción y una comparación de los aspectos más importantes de los principales puentes Java-COM, teniendo en cuenta la información disponible de cada uno de ellos. En la comparación de los puentes Java-COM se consideraron los siguientes aspectos: tipos de elementos que comunican, modo por el que es posible la comunicación, plataformas en las que corren, si son gratis o comerciales.

1.1 PUENTES JAVA-COM

1.1.1 Jawin

- ❖ Permite que programas Java utilicen objetos COM (Component Object Model) o DLLs (Dynamic Link Library) Win32 sin tener que escribir código JNI (Java Native Interface).
- ❖ Corre sobre cualquier plataforma Windows.
- ❖ De fuente abierta, gratis.
- ❖ Comunicación:
 - El generador de código lee la librería de tipo del componente COM y genera los “stubs”¹ Java necesarios.
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Jawin está construido utilizando Ant, una herramienta de construcción Java de fuente abierta. Además para proporcionar soporte para compilar código Java, Ant también hace fácil lanzar la mayoría de herramientas de línea de comandos.
 - El generador de “stubs” utilizado por Jawin, jawingen/jawinGenUI.exe, tiene tanto una interfaz gráfica como de línea de comandos.
 - Jawin automáticamente convierte errores COM y Win32 a instancias Jawin.
 - A diferencia de Java o Win32, COM proporciona una protección para componentes que solamente se pueden llamar seguramente desde ciertos hilos. La arquitectura para soportar esto está basada en “apartments”². Un “apartment” es un grupo de componentes e hilos corriendo con características de “threading”³ similares. Jawin utiliza “apartments”.
 - La arquitectura, se describe a continuación.

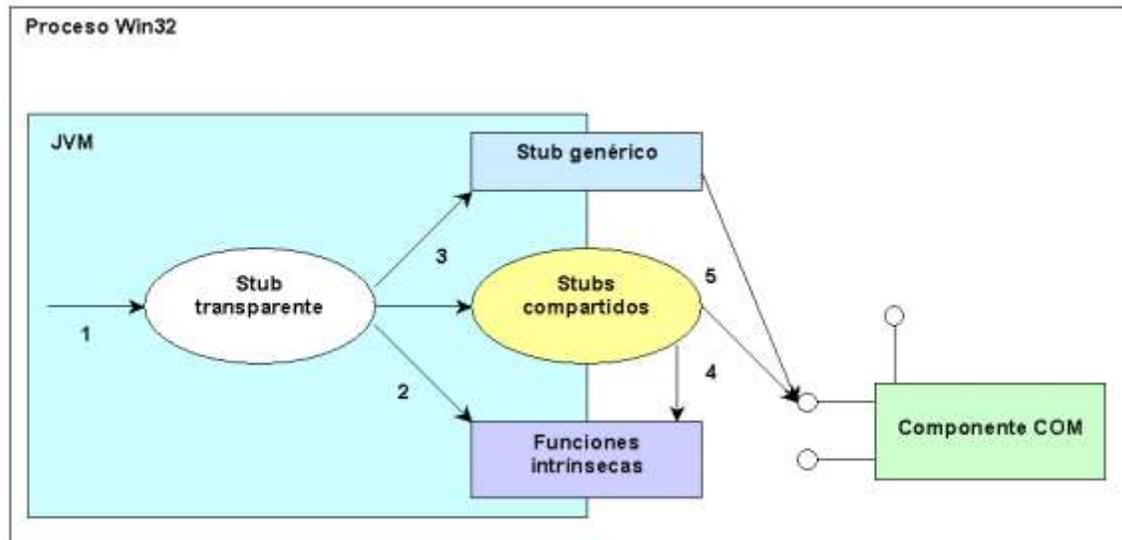
¹ Stub, representante local del extremo remoto. Se prefiere su uso en Inglés.

² Apartment, grupo de componentes e hilos.

³ Threading, hilado. Se prefiere su uso en Inglés.

Arquitectura. Un “stub” transparente se genera a partir de la información de tipo, y es diferente para cada objeto COM o DLL. Los “stubs” compartidos manejan declaraciones de métodos comunes llamando métodos nativos con declaraciones JNI correlacionadas. El “stub” genérico maneja los métodos que tienen declaraciones menos comunes y por lo tanto no son “stubs” compartidos. Tanto el “stub” genérico como los “stubs” compartidos utilizan funciones de ayuda (llamadas funciones intrínsecas) para convertir tipos de datos particulares. La arquitectura de Jawin se muestra en la Figura 1.1.

Figura 1.1 Arquitectura de Jawin



1. El llamante invoca un método.
2. El “stub” del componente serializa los argumentos.
3. Y entonces los envía.
4. El “stub” genérico o compartido des-serializa los argumentos.
5. Y llama el componente Win32 o COM.

La secuencia completa corre en sentido contrario para convertir el valor de retorno de regreso al llamante.

- ❖ **Intrínsecos Jawin.** Las funciones intrínsecas de Jawin son los átomos del “marshalling”⁴. Una función intrínseca sabe cómo convertir uno o más tipos de datos a un “wire format”⁵, o cómo recuperar tipos de datos desde un “wire format”.
- ❖ **Stubs compartidos.** Jawin utiliza “stubs” compartidos para convertir métodos COM y Win32 que tienen declaraciones comunes.

Semánticamente, los tipos de los argumentos y los valores de retorno de estos métodos son todos muy diferentes. Sin embargo, todos estos tipos se convierten a los mismos valores. Así, es posible implementar todos estos métodos con un solo punto de entrada JNI.

⁴ Marshalling, transformación. Se prefiere su uso en Inglés.

⁵ Wire format, formato a bajo nivel. Se prefiere su uso en Inglés.

Esto reduce enormemente el número de puntos de entrada que se deben implementar para manejar un conjunto de APIs (Application Program Interface) dado. En lugar de un punto de entrada por método, solamente se necesita un punto de entrada por declaración única.

Por supuesto, no todas las funciones se pueden implementar con un número finito de “stubs” compartidos. Para métodos con declaraciones más exóticas, Jawin también proporciona un “stub” genérico.

- ❖ **Stub genérico Jawin.** El “stub” genérico Jawin es un “marshaller” verdadero, similar a RMI (Remote Method Invocation) o DCOM (Distributed Component Object Model). El “stub” genérico ve una llamada a una función como una secuencia de eventos. Las funciones intrínsecas serializan una función dentro de un mensaje de requerimiento, y el “stub” genérico mueve este mensaje dentro del espacio nativo. Allí, otro conjunto de funciones intrínsecas convierten el requerimiento serializado a una pila de llamadas e invocan la función. La secuencia completa hace “backwards”⁶ para serializar un mensaje de respuesta con valores o excepciones de retorno y lo envía de regreso al llamante.

El “stub” genérico conoce cómo enviar mensajes hacia atrás y hacia adelante de Java a Win32, pero no conoce las especificaciones de ningún método particular. En el lado Java, el “stub” transparente conoce estos detalles. Sin embargo, no hay un equivalente en el lado de Win32 del “stub” transparente. De acuerdo a la convención de nombramiento RMI y CORBA, tal componente se llamaría “skeleton”⁷. Jawin no utiliza “skeletons” debido a que el mensaje de requerimiento lleva con él la información de tipo.

Además del requerimiento serializado, el “stub” genérico también pasa un flujo de instrucciones que describen cómo deserializar el requerimiento en el lado win32/COM. El flujo de instrucciones es una secuencia de “bytecodes”⁸ que se procesa por un interpretador simple para reconstruir el stack de llamadas. Estos no son bytecodes en el sentido de una clase binaria Java; ellos son específicos y arbitrarios para Jawin.

El “stub” genérico de Jawin es primitivo comparado con RMI o DCOM. Se están haciendo mejoras al “stub” genérico para soportar casos de uso particulares.

1.1.2 Jace

- ❖ Permite manipular objetos Java como si fueran objetos C++ e incluir código C++ en Java.
- ❖ Jace ha sido construido para Windows, Solaris, HP-UX y Linux.
- ❖ De fuente abierta, gratis.
- ❖ Comunicación:
 - Las herramientas Java de Jace generan clases “proxy”⁹ C++ y clases “peer”¹⁰ C++ de las clases Java.
 - La librería de runtime de Jace utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.

⁶ Backward, proceso inverso. Se prefiere su uso en Inglés.

⁷ Skeleton, es el representante remoto del extremo remoto. Se prefiere su uso en Inglés.

⁸ Bytecode, código compilado. Se prefiere su uso en Inglés.

⁹ Proxy, representante. Se prefiere su uso en Inglés.

¹⁰ Peer, par. Se prefiere su uso en Inglés.

- ❖ Otras características:
 - Las clases “proxy” generadas permiten a los desarrolladores instanciar y manipular objetos Java en tiempo de ejecución, como si fueran clases C++ nativas. Las clases “peer” generadas proporcionan un método fácil para que los desarrolladores implementen métodos nativos declarados en sus clases Java, es decir implementen cualquier función nativa Java en C++.
 - Realiza gestión de hilos, gestión de excepciones y conversión automática de tipos, entre otras.
 - El futuro de Jace está abierto, basado principalmente en la realimentación y necesidades de su comunidad de usuarios.
 - Las herramientas, se describen a continuación.

Herramientas

- ❖ **ProxyGenerator** genera las clases “proxy” C++ para una clase Java simple. Se puede especificar a qué nivel de acceso el ProxyGenerator genera campos y métodos miembros. Por defecto, el ProxyGenerator solamente genera miembros públicos, pero dependiendo del nivel de acceso especificado, él también incluirá campos y métodos protected, package, o private. Normalmente, los desarrolladores deberían preferir utilizar AutoProxy en lugar del ProxyGenerator, ya que el AutoProxy caminará por el árbol de dependencias para generar todas las clases dependientes.
- ❖ **AutoProxy.** Esta herramienta escanea en los archivos encabezamiento y los archivos fuente C++ #includes que referencien “proxies” C++ Jace. Él entonces genera los archivos de encabezamiento y fuente para todo el árbol de dependencia para esos “proxies” C++. AutoProxy se utiliza también en conjunto con otras herramientas como BatchGenerator y PeerGenerator.
- ❖ **BatchGenerator.** BatchGen es similar a ProxyGen en que él genera archivos de encabezamiento y fuente para clases “proxy” C++. Sin embargo, en lugar de leer desde un archivo de clase Java, BatchGen lee desde un archivo jar o zip de múltiples archivos de clase. BatchGen se utiliza para generar todas las clases “proxy” C++ dentro de un archivo jar. Esta herramienta es útil cuando se está intentando crear una API C++ para una API Java existente. También se puede ejecutar AutoProxy sobre las clases “proxy” C++ resultantes, así que todas las clases “proxy” C++ dependientes se generen.
- ❖ **PeerEnhancer** se utiliza para mejorar el “bytecode” de las clases “peer” Java con el código de gestión de tiempo de vida automático para “peers” C++ nativos.
- ❖ **PeerGenerator** genera el archivo de encabezamiento y el código fuente C++ requerido para implementar el “peer” C++ nativo para un “peer” Java.

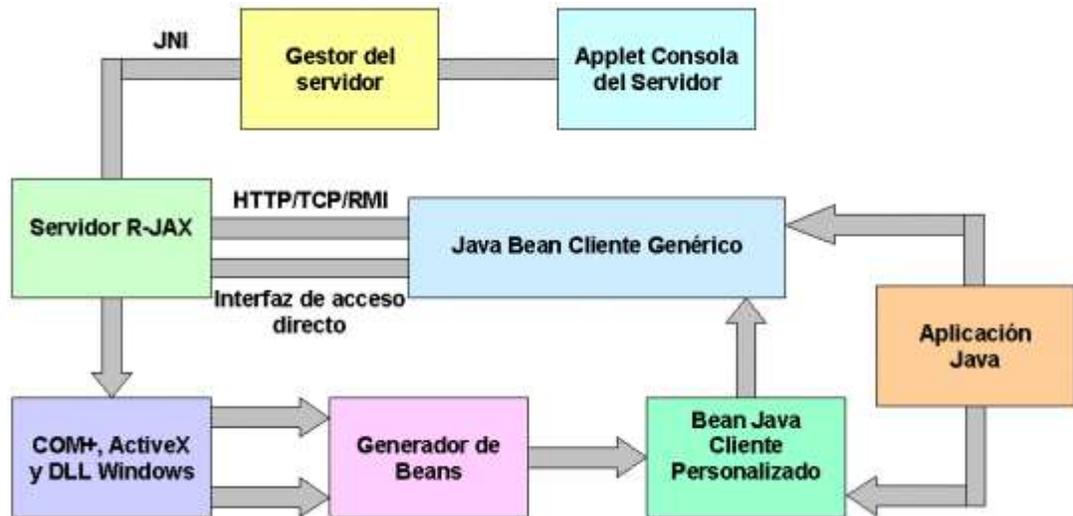
1.1.3 R-JAX

- ❖ Permite que programas Java se comuniquen con objetos COM y DLLs localizados en servidores locales y remotos.
- ❖ Su fabricante es Stryon.
- ❖ Tiene una versión de evaluación válida por 30 días.
- ❖ Comunicación:
 - El generador de Beans lee la librería de tipo del componente COM y genera JavaBeans.
 - Los Beans se utilizan en una aplicación Java que corre en cualquier plataforma.
 - El servidor R-JAX se instala en el equipo Windows donde están los componentes COM y DLLs.
 - Tiene tres posibles adaptadores de comunicación remota: RMI, TCP o HTTP.

❖ Otras características:

- R-JAX tiene componentes que son independientes unos de otros y se pueden ajustar individualmente. Estos componentes se pueden observar en la Figura 1.2.

Figura 1.2 Arquitectura de R-JAX



- ✓ **Bean Cliente:** punto de conexión para aplicaciones Java que necesitan conectarse a objetos COM/DLL nativos. El Bean se puede incorporar en cualquier aplicación Java.
- ✓ **Generador de Beans:** genera JavaBeans para un componente COM al leer su librería de tipo (archivos .olb, .ocx, .exe, .tlb y dll). Los JavaBeans generados son realmente “proxies”, que incluyen todo el código de conectividad para lograr la comunicación con el Servidor R-JAX.
- ✓ **Adaptador de comunicación:** sirve como un conducto para RMI, TCP, o HTTP entre el programa Java y el componente COM/DLL remoto. Los desarrolladores pueden conmutar entre diferentes adaptadores de comunicación. Se proporciona una interfaz de acceso directo para los programas Java y COM que residen en la misma máquina Windows del servidor R-JAX.
- ✓ **Servidor R-JAX:** corre en el servidor Windows donde residen COM/DLLs. Es una combinación de binarios Java y nativos. El código nativo se utiliza para la comunicación con DLLs y componentes COM disponibles. La porción Java es responsable de todas las otras tareas incluyendo comunicaciones y gestión del servidor.
- ✓ **Gestor del servidor:** hace posible administrar remotamente el servidor R-JAX a través de un applet Java.

- El funcionamiento, se describe a continuación.

Funcionamiento. La aplicación cliente Java puede crear entonces un objeto de la clase “proxy”, e invocar métodos en ese objeto.

El mecanismo necesario para convertir los tipos de datos Java a tipos de datos R-JAX se incorpora en las clases “proxy” generadas y por lo tanto es transparente a la aplicación cliente.

Al utilizar el bean cliente R-JAX, las aplicaciones cliente Java se conectan al servidor R-JAX. El servidor R-JAX primero valida el cliente. Después de la validación, el cliente Java puede acceder el objeto COM, DLL o componentes ActiveX según el requerimiento. Esta comunicación se hace a través de los adaptadores cliente y servidor R-JAX diseñados especialmente.

La clase “proxy” convierte el nombre del método y los parámetros, y pasa esta información al servidor R-JAX. El Servidor R-JAX invoca directamente los métodos en el componente ActiveX y recibe el valor de retorno. Él entonces convierte el valor de retorno y lo pasa de regreso a la clase “proxy” Java, la cual a su vez lo pasa de regreso a la aplicación cliente que hizo la llamada. Este proceso es transparente al programador de la aplicación cliente, así que llega a ser posible para los usuarios acceder a componentes ActiveX remotos como un Java Bean en sus aplicaciones Java.

iASP. iASP es una implementación Java de la infraestructura ASP (Active Server Page) de Microsoft. Con iASP (Instant ASP) de Stryon, se pueden escribir aplicaciones web basadas en ASP (Active Server Pages) una vez y correrlas en cualquier parte. iASP permite construir rápidamente aplicaciones escalables, de clase de empresa, cross-plataforma para Internet, Intranet, o Extranet.

Chilisoft. Chilisoft es otra compañía que ofrece la tecnología ASP a través de Chilisoft ASP, que tiene el mismo funcionamiento que ASP de Microsoft pero corre en sistemas operativos como Unix, Solaris entre otros.

ADO R-JAX. Una característica avanzada de R-JAX es el soporte para conexiones remotas a bases de datos a través de scripts ASP corriendo en iASP. Los scripts ASP utilizan ADO (ActiveX Data Object) para manipulaciones de bases de datos. ADO es uno de los componentes más utilizados en aplicaciones ASP.

Mientras iASP proporciona una implementación de ADO, en algunos casos se requiere utilizar el componente ADO de Microsoft. R-JAX proporciona el Bean ADO R-JAX, RJAXADODB. Al utilizar el componente RJAXADODB de RJAX, los usuarios pueden acceder remotamente bases de datos y realizar operaciones de manipulación de datos o creación de datos a través de sus aplicaciones ASP.

RJAXADODB es un conjunto de componentes JavaBeans, cada uno de los cuales actúa como un “proxy” para el componente ActiveX ADO de Microsoft correspondiente. Este paquete se utiliza en scripts ASP corriendo en plataformas no Windows, que necesitan utilizar componentes ADO de Microsoft remotamente.

1.1.4 Java2COM

- ❖ Permite que las aplicaciones Java utilicen objetos COM y hace posible exponer objetos Java como si fueran objetos COM.
- ❖ Soporta todas las máquinas virtuales de Java basadas en Win32.
- ❖ Su fabricante es Neva Object.
- ❖ Tiene una versión de evaluación válida por 30 días.
- ❖ Comunicación:
 - Java2COM Automation Wizard lee la librería de tipo de los componentes COM y genera “wrappers”¹¹ Java.
 - La aplicación Java IBuilder genera “skeletons” Java para los componentes COM.
 - Las comunicaciones Java-COM y COM-Java utilizan una interfaz Java-a-nativo-a-Java llamada Coroutine para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:

¹¹ Wrapper, proxy. Se prefiere su uso en Inglés.

- La jerarquía de clases Java2COM está modelada según la jerarquía de interfaces COM. Por lo tanto, un desarrollador que conozca los principios COM puede extender fácilmente estas clases para adicionar soporte para las interfaces COM que actualmente no están incluidas en Java2COM.
- Ya que Java2COM fue diseñado según COM, tener un conocimiento de este modelo permite entender fácilmente Java2COM.
- La librería de clases Coroutine para Java proporciona el mecanismo para que Java llame funciones externas exportadas desde librerías de enlace dinámico de Windows. Coroutine también permite construir dinámicamente “wrappers” alrededor de métodos Java, así que funciones externas pueden llamar métodos Java como si fueran funciones nativas.
- El acceso, se describe a continuación.

Acceso

- ❖ **Java-COM.** Automation Wizard lee la librería de tipo asociada con el servidor de automatización y genera una clase “wrapper” Java para cada interfaz Dispatch expuesta por el servidor de automatización COM. Antes de correr el cliente se debe haber registrado el servidor.
- ❖ **COM-Java.** Utiliza clases “skeleton” Java generadas para todas las interfaces COM expuestas a través de Java2COM. Una clase “skeleton” Java, que convierte llamadas hechas a través de la interfaz COM en llamadas a métodos Java, corre en el servidor y tiene tres importantes responsabilidades. Primero, crea una tabla de funciones virtuales COM basada en la definición de la interfaz. Segundo, invoca el método apropiado en la implementación de la interfaz. Finalmente, “marshal”/“unmarshal” parámetros de métodos y maneja excepciones Java.

Para generar un “skeleton”, se utiliza la aplicación Java IBuilder. IBuilder es una utilidad de línea de comandos que requiere JRE (Java Runtime Environment) 1.2 (o más alto) y Java2COM para ejecutarse.

IBuilder genera tres archivos. El primero es una clase Java “skeleton”. El segundo es una librería de tipo COM que contiene la definición de la interfaz COM. El tercer archivo es un ejecutable que se utilizará para registrar la interfaz y la librería de tipo. Se lo debe ejecutar en el servidor y en cualquier máquina cliente. En el siguiente paso, se debe implementar la interfaz COM en Java.

En seguida se construye una aplicación servidor COM en Java cuya responsabilidad principal es exponer el componente COM construido en Java.

Finalmente, se construye un archivo Java y se adicionan todos los archivos de clases: la aplicación servidor COM, la descripción de la interfaz COM, la implementación de la interfaz COM, la clase “skeleton” Java. También se pueden adicionar otros archivos de clases que sean necesarios.

El archivo Java se debe instalar en la máquina servidor junto con Coroutine y Java2COM. Cada Servidor COM se debe registrar con COM en la máquina que contiene el ejecutable del servidor. Típicamente, el registro incluye la creación de entradas en el Registro Windows.

La utilidad de Ayuda para el Registro del Servidor Java2COM, j2cnfg.exe permite registrar el servidor con facilidad. Para registrar la interfaz y la librería de tipo se utiliza el archivo ejecutable producido por IBuilder. Éste se copia a la máquina servidor y se ejecuta.

Durante la ejecución, la librería de tipo será extraída al directorio sistema de Windows y será registrado con COM. Se debe registrar la interfaz y la librería de tipo en la máquina servidor y en las máquinas clientes.

Para correr el servidor remotamente se debería copiar el archivo Java que contiene el código del servidor a la máquina remota. En seguida se debe registrar el servidor utilizando j2cnfg.exe y

registrar la interfaz COM utilizando los ejecutables binarios producidos por IBuilder. JRE, Coroutine y Java2COM deben estar instalados en la máquina remota.

Finalmente, se necesitan establecer los permisos de acceso y lanzamiento en la máquina servidor para permitir acceso al servidor desde el cliente, para esto se comienza la utilidad de Configuración de DCOM dcomcnfg.exe en el servidor.

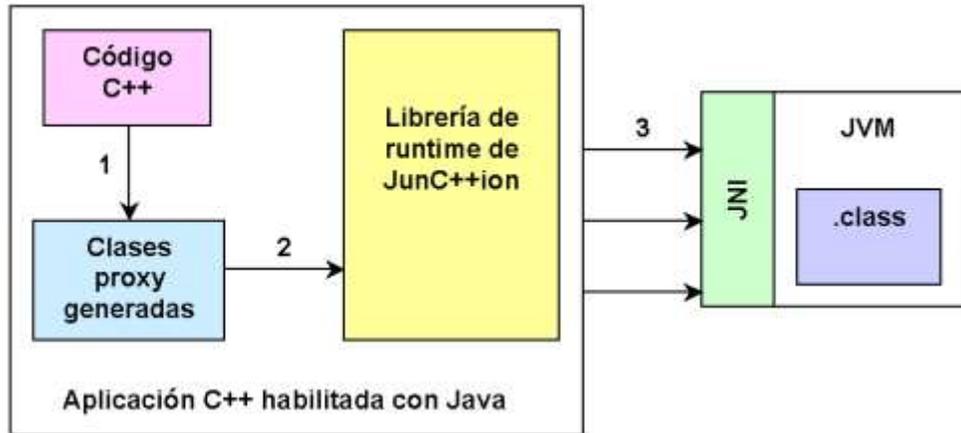
Para exponer el servidor COM construido en Java a .NET se deberían convertir las interfaces y clases contenidas en la librería de tipo a metadatos CLR .NET. Típicamente, se utiliza The Type Library Importer (Tlbimp.exe), una herramienta de línea de comandos que genera un assembly .NET a partir de una librería de tipo.

1.1.5 JunC++ion

- ❖ Permite utilizar objetos Java en C++ y C++ en Java.
- ❖ Corre sobre Windows, Solaris y Linux .
- ❖ Su fabricante es Codemesh.
- ❖ Tiene una versión de evaluación con un período de prueba de 30 a 60 días.
- ❖ Comunicación:
 - El generador de código genera clases “proxy” C++ para las clases Java.
 - Las clases “proxy” C++ utilizan la librería de runtime que utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - JunC++ion permite a los programadores utilizar clases Java en programas C++ simplemente como si fueran otras clases C++, y utilizar C++ en Java a través de implementaciones de métodos nativos.
 - La librería de runtime de JunC++ion proporciona una conexión de bajo nivel entre una JVM (Java Virtual Machine) y la aplicación C++. Las clases “proxy” C++ generadas utilizan clases de la librería de runtime, las cuales a su vez utilizan JNI para comunicarse con una JVM.
 - Las clases “proxy” C++ no solamente pueden extender otras clases “proxy” C++, las clases C++ estándar (no “proxy”) pueden extenderlas también. Esta característica permite a una clase C++ implementar una interfaz Java en C++.
 - Las excepciones Java se mapean a excepciones C++, y las excepciones C++ se mapean a excepciones Java.
 - Las tecnologías disponibles solamente para Java, tales como EJB (Enterprise JavaBeans), JDBC (Java Database Connectivity) y JNDI (Java Naming and Directory Interface), pueden trabajar con componentes C++.
 - El acceso, se describe a continuación.

Acceso. La forma en la que JunC++ion trabaja se puede ver en la Figura 1.3.

Figura 1.3 Funcionamiento de JunC++ion



1. La aplicación C++ realiza una acción que utiliza clases Java.
2. La clase “proxy” llama la librería de runtime de JunC++ion.
3. La librería de runtime de JunC++ion llama la JVM, la lanza (si aún no se ha iniciado), y produce llamadas JNI para la clase “proxy”.

❖ **Java-C++.** JunC++ion permite a clases Java ser utilizadas en C++ como si fueran otras clases C++:

- El Generador de Código de JunC++ion genera clases “proxy” C++ para clases Java.
- Un programador utiliza las clases “proxy” generadas en un programa C++.
- En tiempo de ejecución, la librería de runtime de JunC++ion traslada las llamadas a “proxy” C++ a sus llamadas Java correspondientes.

❖ **C++-Java.** Con JunC++ion, los programadores pueden implementar métodos Java nativos en C++, utilizando las clases “proxy” C++ generadas, que corresponden a las clases Java, y su propio código C++.

- Un programador escribe una clase Java con métodos nativos.
- El Generador de Código genera clases “proxy” C++ de todas las clases Java que el programador utilizará, incluyendo aquellas con métodos nativos.
- Un programador implementa los métodos nativos en C++, sin utilizar JNI.
- En tiempo de ejecución, la librería de runtime de JunC++ion maneja todas las operaciones para transferir información entre los componentes Java y C++, así como también maneja excepciones y chequeo de errores.

1.1.6 xFunction

❖ Permite hacer llamadas desde Java, a funciones externas, implementadas en cualquier lenguaje y que estén en DLLs o librerías compartidas, o realizar llamadas a métodos Java desde funciones externas.

❖ Corre sobre Windows y Linux.

❖ Su fabricante es Excelsior.

❖ Tiene una versión de evaluación válida por 30 días.

❖ Tiene documentación poco clara.

- ❖ Comunicación:
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Se extienden y se instancian clases xFunction para crear objetos Java convencionales, que representan funciones externas, estructuras de datos, punteros, entre otras.
 - Todas las conversiones de datos necesarias y las llamadas a funciones externas son hechas transparentemente por la librería de xFunction.

1.1.7 JCOM

- ❖ Permite que programas Java utilicen objetos COM.
- ❖ Corre sobre Windows95/98/2000 y NT.
- ❖ Tiene licencia LGPL (Lesser General Public License)
- ❖ No tiene documentación acerca del funcionamiento de la herramienta pero tiene algunos ejemplos.

1.1.8 JACOB

- ❖ Permite llamar componentes COM/ActiveX y librerías Win32 desde Java.
- ❖ Corre sobre cualquier plataforma Windows.
- ❖ Tiene una licencia que es una ligera modificación de la licencia BSD proporcionada por el opensource.org.
- ❖ Comunicación:
 - Para llamar un componente COM/DLL se tiene que instanciar el componente COM utilizando ya sea clases ActiveX o Dispatch. Cuando se tiene el objeto instanciado, se puede llamar Dispatch.call() para invocar un método COM simple pasando los parámetros que el método necesita.
 - También se pueden utilizar “wrappers”. Estos son clases Java que exponen un modelo de objetos similar al de los componentes COM (el mecanismo base para Dispatch no es tan amigable para los desarrolladores Java). Se pueden generar “wrappers” para componentes COM utilizando Jacobgen, pero éste todavía no está completo.
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - La modificación a la licencia solamente coloca una restricción extra: no se puede distribuir JACOB libremente como parte de ningún producto comercial que esté dirigido principalmente a los desarrolladores de Java. Esta modificación intenta prevenir que los vendedores de productos creen libremente una versión de JACOB que sea soportada solamente en su entorno de desarrollo particular.
 - La meta fue hacer a Jacob tan compatible como fuera posible, con la API de Microsoft que implementa COM en la JVM de Microsoft, así que se implementaron algunas interfaces COM utilizando JNI.
 - No tiene documentación, pero ya que las clases Jacob son casi compatibles con las clases COM para Java de Microsoft se puede revisar su documentación.

- Con Jacob no se pueden llamar Componentes COM a través de máquinas o sistemas operativos. JACOB está diseñado para la plataforma Windows. COM es un estándar binario definido por Microsoft que no ha sido portado a otras plataformas. Jacob define cómo los componentes se comunican con los otros cuando ellos están en la misma máquina. El estándar DCOM define cómo los componentes se comunican a través de redes, pero JACOB no tiene soporte para DCOM.
- Jacobgen es la abreviatura para JACOB Generator. Es un generador de código para JACOB que genera automáticamente clases “wrapper” para componentes COM/ActiveX, y que por tanto permite a estos componentes ser accedidos como si fueran componentes Java puros. Jacobgen analiza archivos dll, exe, tlb u olb solicitando información de interfaces COM. Genera un archivo intermedio con las descripciones de las interfaces, similar en contenido a un archivo IDL (Interface Definition Language) o MIDL (Microsoft Interface Definition Language), pero con un formato diferente. Finalmente convierte las definiciones de las interfaces y genera los archivos fuente de las clases de salida.
- Jacobgen está licenciado bajo Licencia Pública GNU. El autor retiene el copyright del código fuente de Jacobgen, pero la licencia es muy permisiva. Se utiliza Jacobgen bajo el propio riesgo. No se dan garantías acerca del software.
- No se pueden desplegar componentes COM visuales ya que JACOB no proporciona soporte para tener controles ActiveX gráficos dentro de componentes Java.
- El modelo de “threading” está basado fuertemente en COM “apartments”.
- La gestión del tiempo de vida se basa en la gestión del tiempo de vida del hilo antes que en la colección de basura.

1.1.9 JWindows

- ❖ Permite que un programa Java acceda al entorno de programación de Microsoft Windows debido a que es básicamente un “wrapper” de la librería MFC (Microsoft Foundation Classes) escrita por Microsoft.
- ❖ JWindows está dirigido principalmente a Windows98/95 pero trabaja bien bajo Windows NT/ME/2000. JWindows no correrá en Windows CE. JWindows trabaja con cualquier compilador y máquina virtual.
- ❖ Tiene licencia GNU Lesser Public.
- ❖ Comunicación:
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Una librería de clases que realiza esto, ya está disponible en Microsoft. Se llama SDK (Software Development Kit) de Microsoft para Java y se proporciona con MS J++ (Microsoft J++). Sin embargo esa librería requiere el uso de tanto el compilador Java de Microsoft como la máquina virtual Java de Microsoft, y por lo tanto no es para nada portable. Además, debido a un acuerdo entre Sun y Microsoft, a partir de septiembre de 2004, Microsoft no está autorizado para continuar soportando su JVM.
 - No soporta controles ActiveX.
 - JWindows actualmente sólo soporta un subconjunto del conjunto de características Windows, pero suficiente para permitir la creación de las aplicaciones Windows más básicas.
 - Si hay alguna función o una clase que falte se puede adicionar fácilmente al código fuente y recompilar.
 - No se esperan actualizaciones en el futuro.

1.1.10 jacoZoom

- ❖ Permite utilizar objetos COM desde Java.
- ❖ Corre sobre cualquier plataforma Windows.
- ❖ Su fabricante es InfoZoom.
- ❖ Tiene una versión de evaluación por 30 días.
- ❖ Comunicación:
 - La utilidad de línea de comandos produce clases “wrapper” Java que corresponden directamente a objetos e interfaces COM.
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Soporta controles ActiveX.
 - Soporta interfaces vTable y Dispatch.
 - Se puede utilizar para acceder al Directorio Activo.
 - Incluye un paquete que permite a los programadores Java acceder el Registro de Windows.
 - Soporta el manejo de eventos COM por clientes Java.
 - Paquetes Java, se describen a continuación.

Paquetes

- ❖ **com.inzoom.comjni.** El puente Java-COM. El paquete comjni contiene clases e interfaces, las cuales corresponden directamente a interfaces y structs COM.

Típicamente las clases de este paquete no se llaman directamente sino solamente indirectamente a través de Clases Wrapper Java, JCWs (Java Classes Wrapper) de una librería de tipo importada. El paquete es necesario no solo para crear las JCWs de una librería de tipo sino también para compilarlos y correrlos.

- ❖ **com.inzoom.comtlb.** El convertidor de la librería de tipo y la utilidad de línea de comandos necesarios para generar los “wrappers” que se llaman desde Java.

El paquete comtlb contiene un convertidor de librería de tipo y una clase main, que se invoca por la utilidad de línea de comandos para generar los “wrappers” Java. Depende de comjni ya que él no llama a COM directamente sino solamente a través de JCWs localizados en comjni. Este paquete no es necesario para compilar y correr los “wrappers” generados.

Además de la generación de código Java este paquete se puede utilizar para generar sintaxis COM-IDL. Él expone librerías de tipo en una jerarquía de objetos intuitiva, mucho más fácil de explorar que las interfaces COM subyacentes. El paquete se puede utilizar ya sea a través de la utilidad de línea de comandos o directamente desde Java. Cuando se utiliza desde dentro de Java proporciona opciones adicionales con respecto a la generación de código Java.

- ❖ **com.inzoom.axjni.** Soporte para controles ActiveX.
- ❖ **com.inzoom.winreg.** Acceso al registro Windows.

1.1.11 Bridge between Java and Windows

- ❖ Permite que programas Windows y programas Java se comuniquen.
- ❖ El paquete es gratis (para uso comercial y no comercial) pero el código fuente no se proporciona al público en este momento. El código que se proporciona son versiones simplificadas de los componentes clave en ese paquete. Se puede utilizar y modificar el código en cualquier forma que se desee si se reconoce en el trabajo al autor.
- ❖ Comunicación:
 - Existe un control ActiveX y una clase Java llamados XYMessenger, que tienen métodos casi idénticos y pueden enviar y recibir mensajes uno al otro.
 - El control ActiveX debe estar en un programa que utiliza MFC y corre en Windows, y la clase Java en una aplicación que corre sobre cualquier plataforma.
 - La comunicación se realiza por medio de sockets.
 - Permite comunicaciones remotas.
- ❖ Otras características:
 - La idea básica es insertar el control XYMessenger.ocx en el programa que utiliza MFC, para que pueda actuar tanto como cliente como servidor para otros programas que también estén utilizando XYMessenger. Para los programas Java solamente se necesita derivar una clase de XYMessenger para lograr el mismo objetivo. Una llamada a un método es todo lo que se necesita para lograr la comunicación. El primer programa que llama al método estará actuando como el nodo raíz en el árbol llamado árbol de comunicación. El nodo raíz necesita utilizar un puerto local para que otros programas se conecten. Otros programas se conectarán a este nodo raíz utilizando la dirección IP y el número del puerto (del nodo raíz). Estos programas serán los clientes del nodo raíz, sin embargo, cada uno de ellos también puede ser un servidor, así que más y más programas se pueden conectar.
 - Existe un método que permite enviar mensajes de tamaño arbitrario a otros programas conectados. Cada nodo en el árbol de comunicación está identificado por un identificador de usuario. Existen métodos que se pueden utilizar para recuperar los identificadores de usuario de todos los nodos.
 - Los mensajes entrantes se almacenan en un arreglo de mensajes interno. Para los programas Windows, un evento ActiveX se lanzará cuando un nuevo mensaje llegue, se supone que el desarrollador implementa un manejador de eventos y procesa el mensaje. Para programas Java, se supone que el desarrollador sobrescribe la función manejadora de mensajes virtuales en la clase base XYMessenger para procesar el mensaje. En ambos casos los manejadores de mensajes serán invocados por el sistema automáticamente. Existen varios métodos para obtener información de los mensajes del arreglo y para manipularlos.
 - Cada instancia de XYMessenger.ocx solamente se puede utilizar en un hilo simple (Apartment Model Threading). Se pueden utilizar múltiples instancias de XYMessenger.ocx en múltiples hilos. No hay nada para prevenir el uso de múltiples instancias de XYMessenger.ocx en un hilo simple. Incluso si una instancia de XYMessenger.ocx está esperando por un mensaje de respuesta desde otro nodo, el hilo no se bloquea, él puede procesar mensajes entrantes desde otros nodos.
 - La clase Java XYMessenger es realmente multi-hilo en el sentido de que cada conexión (entre dos nodos) está corriendo en un hilo separado.
 - El control ActiveX XYMessenger.ocx está implementado con VC ++5.0 utilizando la clase CSocket de MFC. La clase Java XYMessenger está implementada con el JDK1.2.2 utilizando las clases Socket y ServerSocket de java.net.
 - El código simplificado que se proporciona tiene un límite de conexiones de máximo 10 clientes a un servidor (sin embargo, cada uno de los 10 clientes pueden tener 10 clientes conectados a él.). No es difícil incrementar el límite a 1000, por ejemplo. Otra limitación es, cada nodo en el

árbol de comunicación mantiene una lista de identificadores de usuario de todos los nodos. Esto puede ser ineficiente si hay cientos de nodos conectados.

1.1.12 The JavaBeans Bridge for ActiveX

- ❖ Proporciona a los usuarios de contenedores OLE (Object Linking and Embedding)/COM/ActiveX legados la habilidad para incorporar y utilizar componentes JavaBeans portables en el mismo modo en que ellos previamente incorporarían y utilizarían componentes OLE/COM/ActiveX específicos a la plataforma.
- ❖ Corre sobre cualquier plataforma.
- ❖ Su fabricante es Sun Microsystems
- ❖ Gratis.
- ❖ Comunicación:
 - El Packager utiliza el archivo jar que contiene el(los) componente(s) JavaBean(s). El Packager crea una librería de tipo OLE y un registro Win32 para el componente. Esto permite a los contenedores OLE/COM/ActiveX analizar y presentar correctamente un componente JavaBean.
 - El puente JavaBeans para ActiveX es parte del producto Java Plug-in. Java Plug-in está disponible para entornos Java2, así como también para el entorno JDK1.1.
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Los beans pueden lanzar eventos que pueden ser capturados por contenedores OLE/COM/ActiveX.
 - Los beans pueden actuar como servidores para la invocación de métodos de OLE/COM/ActiveX. Esto significa que programas Windows pueden invocar métodos en Beans.

1.1.13 Bridge2Java

- ❖ Permite a un objeto ActiveX ser tratado como un objeto Java.
- ❖ Corre sobre cualquier plataforma.
- ❖ Su fabricante es IBM AlphaWorks.
- ❖ Tiene una versión de evaluación válida por 90 días.
- ❖ Comunicación:
 - Se crean “proxies” Java a partir de la librería de tipo del control ActiveX por medio de una herramienta de generación de “proxies”.
 - Utiliza JNI para lograr la interoperabilidad.
 - No permite comunicaciones remotas.

1.1.14 JNI++

- ❖ Permite acceder clases C++ desde Java y viceversa.
- ❖ Corre sobre Windows2000/NT y Linux.
- ❖ Tiene licencia LGPL.

❖ **Comunicación:**

- Tiene dos utilidades de generación de código, la primera de estas utilidades se utiliza para generar clases “proxy” C++ dada una clase o interfaz Java. La segunda utilidad se utiliza para simplificar la creación y exposición de código nativo a Java. Su entrada es una interfaz Java, y su salida incluye una clase “proxy” Java que implementa la interfaz así como también una clase “peer” C++ y código de mapeo.
- Tiene una librería núcleo nativa o C++ que proporciona una interfaz para JVM simplificada, además de clases “wrapper” para los tipos de datos JNI.
- Utiliza JNI para lograr la comunicación.
- No permite comunicaciones remotas.

❖ **Otra características:**

- El código que mapea las llamadas a métodos y los accesos a campos desde las clases “proxy” C++ generadas a sus clases “peer” Java, y desde las clases “proxy” Java generadas a sus clases “peer” C++ generadas es generado. Todo lo que se deja es llenar la implementación para las clases “peer” C++. Las clases “proxy” C++ generadas se pueden utilizar como si estuvieran implementadas en C++, aunque las llamadas a los métodos son realmente delegadas a la clase “peer” Java correspondientes en el otro lado de JNI.
- Acceso, se describe a continuación.

Acceso

❖ **C++-Java**

- El generador de “proxies” C++ es responsable de generar código para permitir al código fuente C++ instanciar y utilizar clases Java como si ellas estuvieran implementadas en C++.
- El generador de “proxies” JNI++ acepta un archivo proyecto y una o más clases e interfaces java como entrada, y genera todo el código que se necesita para utilizar convenientemente las clases java como si estuvieran escritas en C++.
- El primer paso es crear un archivo proyecto que lista las clases java de entrada y establece varias opciones de generación de código.
- En seguida, las clases java de entrada son compiladas. Luego, el generador de “proxies” JNI++ se invoca para generar clases “proxy” C++ para las entradas definidas. Las clases “proxy” C++ generadas son compiladas y enlazadas con el código escrito por el usuario y la librería de runtime JNI++.
- Las clases “proxy” C++ generadas son utilizadas por el código cliente, el cual utiliza la interfaz JVM simplificada proporcionada por la librería de runtime núcleo de JNI++. La librería de runtime núcleo, a su vez, translada las llamadas a la JVM cargada.

❖ **Java-C++**

- El Generador Peer JNI++ genera código fuente C++ y Java que permite el uso de clases C++ como si estuvieran implementadas en Java.
- El generador Peer JNI++ acepta un archivo proyecto y una o más interfaces Java como entrada. Él entonces toma las interfaces Java de entrada y genera el código necesario para implementar de una forma fácil cada una en C++.
- El primer paso es crear un archivo proyecto que lista las interfaces Java de entrada y establece varias opciones de generación de código.
- En seguida, las interfaces Java de entrada se compilan. Luego, el generador “peer” JNI++ se invoca para generar clases “peer” C++ para las interfaces de entrada. Después de llevar a cabo la implementación de las clases “peer” C++ generadas, se compilan y enlazan con código escrito por el usuario y la librería de runtime JNI++.

- El código Java cliente corriendo dentro de la JVM invoca métodos en la clase “proxy” Java generada. Las llamadas a los métodos nativos de la clase “proxy” Java son pasados por la JVM a través del código de mapeo generado al lado C++ de JNI. El código de mapeo, a su vez, convierte la llamada a la instancia “peer” C++ asociada con la clase “proxy” Java. Tanto el código de mapeo como la instancia “peer” C++ pueden hacer uso de la librería de runtime núcleo para realizar sus tareas.

1.1.15. The Win32-Java hybrid, WJH

- ❖ Permite la comunicación entre programas Windows y programas Java.
- ❖ Corre sobre cualquier plataforma.
- ❖ Gratis, de fuente abierta.
- ❖ Comunicación:
 - No permite comunicaciones remotas.
- ❖ Otras características:
 - Una aplicación WJH está compuesta por un ejecutable Win32 y al menos un archivo de clase Java. El ejecutable Win32, entre otras cosas, carga la DLL de la JVM y crea una instancia de la máquina virtual. El archivo de la clase Java de inicio y su método “main”¹² son identificados. Esta información y sus argumentos de línea de comandos asociados se pasan a la JVM. La JVM entonces carga este archivo de clase e interpreta las instrucciones localizadas en el archivo. Para ayudar a ejecutar estas instrucciones, la JVM puede llamar la funcionalidad de la API Win32 o las DLLs Win32 que soporta la JVM.
 - La idea detrás de WJH es escribir una pequeña cantidad de código en C o C++ (el lanzador de la JVM Win32) y escribir la mayor parte del código (la porción de aplicación) en Java.

1.1.16. J-Integra

- ❖ Permite que objetos Java se comuniquen con objetos COM y viceversa.
- ❖ Corre sobre cualquier plataforma.
- ❖ Su fabricante es Intrinsic.
- ❖ Tiene una versión de evaluación válida por 60 días.
- ❖ Comunicación:
 - Ofrece herramientas tales como: com2java, java2com, regtlb, regjvm, setdllhost entre otras, que permiten lograr la comunicación entre los mundos Java y COM.
 - A nivel más bajo, J-Integra utiliza las clases de “networking”¹³ de Java totalmente estándar.
 - Tiene dos modos de comunicación: modo DCOM que permite comunicaciones locales y remotas, y modo nativo, que sólo permite comunicaciones locales.
- ❖ Otras características:

¹² Main, principal. Se prefiere su uso en Inglés.

¹³ Networking, trabajo en red. Se prefiere su uso en Inglés.

- J-Integra incorpora una implementación Java pura del protocolo DCOM de Microsoft, lo que significa que se puede utilizar en cualquier plataforma y con cualquier JVM.
- J-Integra internamente utiliza DCOM protocolo para proporcionar acceso a componentes COM tanto locales como remotos desde un entorno Java puro. Es decir, el runtime Java puro de J-Integra se comunica con componentes COM utilizando DCOM que está sobre RPCs (Remote Procedure Call), que a su vez están sobre TCP/IP. Por lo tanto, a más bajo nivel, J-Integra utiliza las clases de “networking” de Java totalmente estándar.
- DCOM es una parte integral de Windows NT y Windows 98, y está disponible en forma gratuita para Windows 95 en el sitio web de Microsoft. Esto significa que no es necesario instalar software especial en la máquina que tiene el componente COM.
- J-Integra permite el manejo de eventos, excepciones e hilos, así como también recolección de basura en Java y COM.

1.1.17. WebLogic. Intrinsic Software, anunció el 25 de septiembre de 2003, su tercer contrato de mantenimiento y soporte anual con BEA Systems. El contrato es una extensión de los servicios de soporte adquiridos por BEA en julio del 2001, cuando licenció el toolkit de desarrollo de J-Integra de Intrinsic para poder incluirlo en el Servidor WebLogic de BEA.

1.2 OTROS PUENTES

Antes de entrar a detallar otros puentes es necesario tener claros los siguientes conceptos.

COM+/MTS. COM+/MTS (Microsoft Transaction Server) es el servicio de transacción en el sistema operativo Windows NT. EJB es una especificación para un servicio de transacción basado en Java. Tanto COM+/MTS como EJBs dirigen la creación de aplicaciones basadas en componentes, orientadas a transacción. Existen similitudes y diferencias entre estos dos modelos de componentes.

Una transacción es un conjunto de eventos que son llevados a cabo como una unidad, ya sea todos los eventos suceden, o ninguno de ellos lo hace.

Servicios Web. Los Servicios Web proporcionan la integración de aplicaciones Web diferentes así como también de datos y servicios sobre Internet. Es un conjunto de aplicaciones modulares e independientes que se pueden publicar, descubrir e invocar utilizando protocolos estándares de la industria. Es un modelo de computación distribuida que representa la interacción entre programas.

Los Servicios Web proporcionan un estándar y una plataforma para desarrollar programas y aplicaciones (escritas en diferentes lenguajes y plataformas de diferentes vendedores), corriendo en diferentes máquinas, dispositivos y sistemas operativos. Los Servicios Web describen una arquitectura de aplicación orientada a servicios y basada en componentes.

.NET. Existen tres principales lenguajes de desarrollo en .NET: VisualBasic.NET, C#, y Visual C++.NET, y las aplicaciones y los Servicios Web escritos en estos lenguajes deben ir a través un proceso de dos pasos para ser compilados a código ejecutable por la máquina. Primero, los compiladores del “framework”¹⁴ .NET toman el código fuente y generan un conjunto de instrucciones escritas en un lenguaje común que es interno al “framework”. Este lenguaje se llama Lenguaje Intermedio de Microsoft, MSIL (Microsoft Intermediate Language), y es independiente de la CPU, así que antes de que pueda ser ejecutado, el CLR (Common Language Runtime) debe convertirlo a nativo, código específico a la CPU.

El “framework” .NET incluye un conjunto de tecnologías software que realizan funciones requeridas comúnmente. Estos módulos, las librerías de clase .NET, están agrupadas por categorías funcionales, e incluyen ADO.NET, acceso y gestión de datos, ASP.NET, construcción, despliegue, y ejecución de aplicaciones y Servicios Web, XML (Extensible Markup Language), etiquetamiento de datos, SOAP

¹⁴ Framework, infraestructura. Se prefiere su uso en Inglés.

(Simple Object Access Protocol), intercambio de datos, WebForms (aplicaciones Web), y WinForms (aplicaciones Windows).

En teoría, las aplicaciones .NET pueden correr en múltiples plataformas Windows, pero para que esto suceda, el CLR y las clases del “framework” .NET deben ser portados a las plataformas objetivo. Esto es una tarea significativa, así que todas las aplicaciones de propósitos prácticos escritas en .NET deben correr en .NET.

1.2.1 iNET

- ❖ Se puede escribir una aplicación en .NET y utilizar iNET de Stryon, para convertirla a Java así que ella puede ejecutarse transparentemente en un entorno habilitado con Java, incluyendo sistemas operativos Unix y Linux y los principales servidores de aplicaciones tales como IBM WebSphere, BEA WebLogic, SunONE, Oracle 9i, and Jboss.
- ❖ iNET de Stryon ha re-creado el “framework” de desarrollo .NET completo enteramente en Java. La aplicación .NET se convierte a Java nativo antes del runtime, así que él se ejecuta en Java en cualquier JVM. EL “framework” implementado en Java ofrece ejecución .NET cross-plataforma junto con todos los beneficios y soporte intrínseco para Java y otros “frameworks” del lado del servidor.
- ❖ Existen tres principales componentes del sistema iNET de Stryon:
 - **IL2Java (Intermediate Language to Java).** Este módulo convierte código MSIL a Java nativo. Esto significa que iNET puede convertir cualquier código fuente .NET, en Visual Basic, C#, o Visual C++, en Java nativo.
 - **Librería de clases de runtime.** Las librerías de clases del runtime iNET es un conjunto de módulos software, escritos en Java, que replican la funcionalidad proporcionada por el “framework” .NET, ADO.NET, ASP.NET, WinForms, WebForms, XML, SOAP, todas corriendo en Java nativo.
 - **Herramientas de despliegue.** Varias herramientas iNET dan a los usuarios un modo conveniente para desplegar aplicaciones iNET en plataformas J2EE (Java 2 Platform Enterprise Edition).
- ❖ **Funcionamiento.** Antes del runtime, el compilador .NET compila el código fuente a MSIL, el cual el convertidor IL2Java de iNET convierte a Java nativo. Ya que las librerías de clases del runtime de iNET son totalmente compatibles con las librerías de clases de .NET, el código fuente Java que IL2Java genera contiene referencias a las librerías de clases del runtime de iNET en lugar de a las de .NET.

Una vez la aplicación se convierte totalmente a código fuente Java, corre como cualquier aplicación Java. Se compila por la JVM a “bytecode” Java, y se ejecuta. El paquete del runtime de la aplicación incluye librerías de clases del runtime iNET.

- ❖ IL2Java de iNET está disponible como un plug-in para Visual Studio.NET, así que es fácil para un desarrollador convertir el código a Java y desplegarlo ya sea en el computador objetivo local o remoto, todo dentro de Visual Studio.NET. IL2Java también se puede invocar desde la utilidad de línea de comandos simple al llamar il2java.exe.

1.2.2 iHUB

- ❖ iHUB de Stryon, permite a objetos remotos en “frameworks” disparejos interactuar con otros como si se estuvieran comunicando con objetos que son locales y de su propia especie. El término objeto aquí se aplica a aplicaciones, Servicios Web, servlets, EJB's, componentes COM y MTS, ActiveX, JavaBeans, etc. El soporte para más “frameworks” y tecnologías se adicionará en un futuro. La actual integración de “frameworks” de iHUB tiene soporte para las siguientes tres áreas:

- Integración de la plataforma J2EE y el “framework” .NET.
 - Integración de la plataforma J2EE y el “framework” Windows legado.
 - Integración con XML.
- ❖ IHUB incluye los siguientes puentes:
- **Puente tecnología Java a .NET (J2N):** permite a entidades Java integrarse, incorporar y utilizar aplicaciones, objetos y servicios .NET transparentemente. Él comprende un generador de “proxies”, APIs cliente (APIs remotas para la tecnología Java), canales y una máquina de comunicación.
 - **Puente .NET a tecnología Java (N2J):** la contraparte de J2N que permite a entidades .NET acceder los binarios de Java como si fueran componentes .NET locales. Los objetos de tecnología Java pueden ser desde clases simples Java a JavaBeans, Servlets a incluso EJBs. El puente N2J incluye el generador de “proxies”, el servidor iHUB y un wizard de despliegue.
 - **Puente tecnología Java a Windows (J2W):** proporciona a los objetos Java acceso completo a binarios Windows tales como DLLs, objetos COM/DCOM/MTS o COM+ y componentes ActiveX.
 - **Puente Windows a tecnología Java (W2J):** la contraparte de J2W que abre el mundo de la tecnología Java a todas las aplicaciones Windows.
 - **Puente para Servicios Web:** proporciona un conversor WSDL2Java (Web Services Description Language to Java) el cual permite a los usuarios generar archivos Java a partir de sus archivos de Servicios Web .NET (WSDL). El usuario puede acceder Servicios Web .NET en su código sin la necesidad de un conversor de tercera parte. Las librerías iHUB hacen toda la conversión requerida. Contrariamente, el generador WAR (Web ARchive) permite a los usuarios generar archivos WAR y WSDL a partir de cualquier objeto de tecnología Java incluyendo JavaBeans. Los usuarios pueden entonces publicar y consumir cualquier objeto Java como un Servicio Web.

1.2.3 JuggerNET

- ❖ JuggerNET, es la solución de integración Java-.NET de Codemesh. Con JuggerNET, Java y .NET pueden trabajar juntos al nivel más fino de integración: clase por clase, método por método, campo por campo. JuggerNET genera clases “proxy” .NET a partir de clases Java. Se utilizan las clases “proxy” para lograr que cualquier programa .NET trabaje con Java.
- ❖ **Generador de código.** El Generador de código genera automáticamente “proxies” .NET para cualquier clase Java, las librerías de runtime para las interoperaciones Java-.NET, la interoperabilidad COM, y la generación automática de información de referencia acerca de “proxies”.
- ❖ **Funcionamiento.** Las clases “proxy” .NET utilizan clases utilidad de la librería de runtime de JuggerNET, la cual a su vez utiliza código JNI para comunicarse con una JVM. Como resultado, los programadores no tienen que ver código JNI debido a que JuggerNET maneja todo el código de integración.
- ❖ **Proxies .NET.** Los “proxies” .NET son fáciles de leer y están completamente documentados para proporcionar información de referencia detallada a los programadores.

1.2.4 Ja.NET

- ❖ Ja.Net de Intrinsic, es un puente entre el mundo de Java y el mundo de Microsoft.NET, este puente es de Intrinsic.
- ❖ Ja.NET impulsa NET Remoting, el nuevo protocolo de objetos distribuido de Microsoft .NET Remoting combina lo mejor de las tecnologías distribuidas más nuevas, Servicios Web, con lo

mejor de las tecnologías distribuidas anteriores tales como DCOM, CORBA y RMI. .NET Remoting se utiliza dentro de .NET para permitir que componentes CLR (Common Language Runtime) en diferentes dominios de aplicación se comuniquen con otros. CLR soporta muchos lenguajes tales como: C#, Visual Basic.NET, ASP.NET y C++. Al utilizar Ja.NET, los componentes Java parecen ser componentes CLR, y los componentes CLR parecen ser componentes Java.

- ❖ Una de las características más interesantes de .NET Remoting es que el protocolo de transporte y el formato de datos son configurables y extensibles. Actualmente .NET y Ja.NET soportan los protocolos de transporte HTTP y TCP/IP, y los formatos de datos SOAP y binario.

1.2 RESULTADO DE LA COMPARACIÓN ENTRE PUENTES JAVA-COM

Teniendo en cuenta, que la API Scripting WMI está implementada en una DLL, e incluye una librería de tipo que se encuentra en un archivo .tlb, los puentes que se podrían utilizar son: Jawin, R-JAX, Java2COM, xFuncion, Jacob, jacoZoom, Bridge between Java and Windows, y J-Integra. Con base en el estudio presentado anteriormente, el puente escogido fue J-Integra, por sus mejores características con respecto a los otros puentes.

Los otros fueron descartados principalmente por las siguientes razones:

- ❖ Jawin, debido a que no soporta comunicaciones remotas, no permitiría la gestión de equipos remotos.
- ❖ R-JAX, porque sería necesario instalar el Servidor R-JAX en cada uno de los equipos que se van a gestionar, algo muy dispendioso en una red con un gran número de equipos.
- ❖ Java2COM, porque además de no soportar comunicaciones remotas, está basado en COM que lo hace un poco complicado de utilizar.
- ❖ xFuncion, ya que además de no permitir comunicaciones remotas, tiene una documentación poco clara.
- ❖ Jacob, debido a que no tiene documentación propia y no soporta comunicaciones remotas.
- ❖ JWindows, debido a que no permite comunicaciones remotas.
- ❖ jacoZoom, porque con él no se pueden comunicar objetos remotamente.
- ❖ Puente entre Java y Windows, debido a que sería necesario instalar el programa que contiene el control ActiveX XYMessenger en cada uno de los equipos Windows a gestionar, algo que no es muy óptimo en una red de gran tamaño.

Por estas razones y porque J-Integra permite comunicaciones remotas a través de DCOM, y debido a que con J-Integra no es necesario instalar ningún software especial en los equipos Windows, y tiene una documentación muy clara y muy completa, J-Integra es la mejor elección para realizar la gestión de estaciones de trabajo de usuario utilizando WMI y Java.

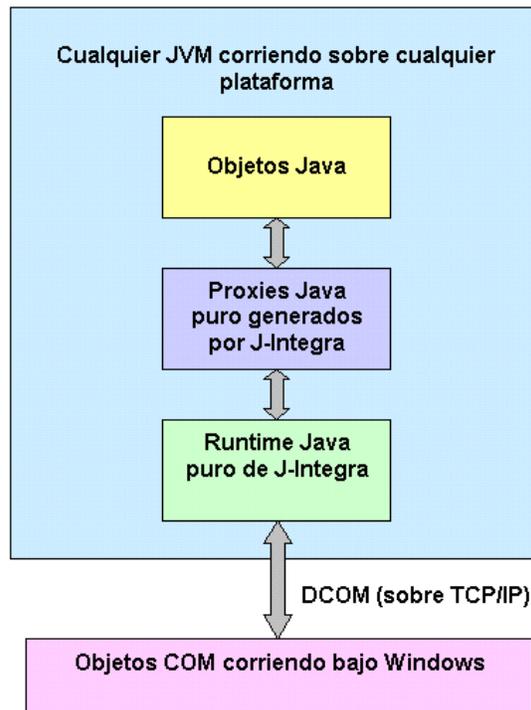
A continuación se hace una descripción detallada de J-Integra:

1.3 J-INTEGRA

J-Integra es puente Java-COM que se puede utilizar para acceder componentes COM como si fueran objetos Java y objetos Java como si fueran componentes COM.

J-Integra incorpora una implementación Java pura del protocolo DCOM de Microsoft, lo que significa que se puede utilizar en cualquier plataforma y con cualquier JVM. J-Integra internamente utiliza este protocolo para proporcionar acceso a componentes COM tanto locales como remotos desde un entorno Java puro. Es decir, el runtime Java puro de J-Integra se comunica con componentes COM utilizando DCOM que está sobre RPCs, que a su vez están sobre TCP/IP. Por lo tanto, a más bajo nivel, J-Integra utiliza las clases de “networking” de Java totalmente estándar. La arquitectura de J-Integra se muestra a continuación. La arquitectura de J-Integra se muestra en Figura 1.4.

Figura 1.4 Arquitectura de J-Integra



DCOM es una parte integral de Windows NT y Windows 98, y está disponible en forma gratuita para Windows 95 en el sitio web de Microsoft. Esto significa que no es necesario instalar software especial en la máquina que tiene el componente COM.

1.3.1 Autenticación. J-Integra soporta autenticación tanto NT Challenge-Response (Desafío-Respuesta) como Legacy LanMan (que se pueden deshabilitar si no se requieren). Estos mecanismos de autenticación hacen que los passwords no sean enviados a través de la red. Actualmente J-Integra solamente soporta Autenticación, no Cifrado.

La herramienta estándar de Microsoft para la configuración de DCOM se utiliza para especificar exactamente quién puede acceder un componente. Es posible comunicarse seguramente con un servidor en Windows 95 o Windows 98, siempre que DCOM haya sido configurado para utilizar un dominio NT para autenticación, es decir J-Integra soporta comunicaciones autenticadas con servidores en Windows 95/98.

1.3.2 DCOM. Cuando el código Java se corre bajo Windows, J-Integra puede utilizar código nativo para tomar la identidad del login actual. Si se prefiere que J-Integra no haga esto, o si no se está loggeado en un dominio NT (por ejemplo si el código Java está en UNIX), entonces se debe configurar el acceso DCOM, y se debe especificar el dominio NT, el usuario y el password que serán utilizados por J-Integra.

Si el código Java se está corriendo bajo Windows 95, entonces se debe descargar e instalar tanto las herramientas DCOM como DCOMCNFG desde el sitio web de Microsoft.

1.3.3 Eventos COM. Si un componente COM puede generar eventos (en términos COM, tiene una interfaz fuente, llamada, por ejemplo, xyz), J-Integra permite a objetos Java implementar la interfaz del evento xyz, y entonces suscribirse a ese evento, utilizando las semánticas estándar de Java de `addXYZListener` y `removeXYZListener`.

1.3.4 Excepciones. J-Integra mapea transparentemente entre excepciones COM y excepciones Java.

Excepciones originadas en objetos Java. Los componentes COM pueden invocar métodos en objetos Java utilizando J-Integra. Si tal método genera una excepción, J-Integra captura la excepción y la traslada a la excepción COM apropiada.

Si se desea establecer explícitamente la información de error, en lugar de aceptar la información por defecto de J-Integra, entonces se puede lanzar una instancia de `com.linar.jintegra.AutomationException`.

Excepciones originadas en componentes COM. Cuando un cliente Java invoca un método en un componente COM, ese componente puede producir una excepción. Las excepciones COM tienen un número de error asociado, una fuente y una descripción.

Cuando un componente COM lanza una excepción, una instancia de `com.linar.jintegra.AutomationException` se lanza en el cliente Java. La clase `AutomationException` expone métodos para obtener el número de error, la fuente y la descripción de la excepción COM que fue lanzada.

Si no se desea utilizar la clase `AutomationException`, entonces se puede capturar la `java.io.IOException` en su lugar, ya que, `AutomationException` se deriva de `IOException`. En este modo se puede reducir la dependencia de código con las clases relacionadas con COM, a expensas de no poder acceder el código de error, la fuente y la descripción.

1.3.5 Colección de basura y cuenta de referencia

COM-Java. Cuando los clientes COM mantienen referencias a un objeto Java, el runtime de J-Integra mantiene una referencia dentro de la JVM a ese objeto Java en nombre de los clientes COM. Él también mantiene la cuenta del número de referencias COM exportadas a ese objeto Java, y libera su referencia cuando la cuenta de referencias COM llega a cero. Cuando se corre en modo DCOM, J-Integra recibe mensajes ping DCOM informándole que un cliente COM todavía está en uso. Si no se reciben tales mensajes ping por seis minutos (por la especificación DCOM), el runtime de J-Integra libera todos los objetos Java sin ping.

Si se desea recibir notificación cuando objetos Java ya no son referenciados por clientes COM, se puede llamar un método pasando una referencia a una instancia de una clase Java que implemente una interfaz de no referencia. Cuando ya no se desea ser notificado, se llama otro método.

Java-COM. La máquina virtual de Java realiza la colección de basura de referencias Java a un objeto COM cuando tales referencias ya no se puedan acceder. Cuando tales referencias se colectan como basura, J-Integra adiciona los detalles del objeto DCOM a una lista interna de referencias de objetos DCOM que se deberían liberar. Cada diez segundos un hilo demonio de J-Integra libera estas referencias a objetos DCOM, que se liberan a través de la colección de basura.

Si se prefiere liberar explícitamente una referencia a un objeto, entonces se puede llamar un método, pasando la referencia al objeto como un parámetro.

Cuando la JVM está a punto de cerrar el sistema se debe llamar un método que libera todas las referencias a objetos COM. Esto libera cualquier referencia a un objeto COM que aún no ha sido liberada a través de la colección de basura. Una vez se ha llamado este método no se puede hacer uso de cualquier objeto COM accedido a través de J-Integra.

Cuando se corre en modo DCOM, el runtime de J-Integra envía mensajes ping DCOM por el protocolo DCOM para informarle al servidor COM que el cliente todavía está en uso. De acuerdo con la especificación DCOM, los clientes DCOM envían estos mensajes ping cada dos minutos. Si tres intervalos ping consecutivos (seis minutos) pasan sin que el servidor COM reciba un mensaje ping, el servidor asume que el cliente ha colapsado o se ha vuelto inalcanzable debido a una falla en la red, y libera y limpia sus referencias.

1.3.6 Hilos

COM-Java. J-Integra mantiene un pool de hilos para manejar requerimientos desde objetos COM. Por defecto, J-Integra permite un máximo de 20 hilos para manejar tales requerimientos, sin embargo se puede cambiar el máximo al establecer una propiedad de J-Integra al número que se quiere.

J-Integra crea tales hilos como y cuando se requieren. Sin embargo, esto no significa que se crean 20 hilos al inicio. J-Integra crea el primer hilo cuando el primer requerimiento remoto llega. Si un segundo requerimiento llega y el primer hilo está ocupado sirviendo un requerimiento, J-Integra crea un segundo hilo para manejar el segundo requerimiento y así sucesivamente, hasta el máximo de 20 (por defecto).

Estos hilos normalmente terminarán después de 5 minutos de no ser utilizados, sin embargo este tiempo se puede incrementar al establecer una propiedad de J-Integra.

Si se desea evitar que el mismo método en un objeto Java se invoque más de una vez al mismo tiempo por objetos COM, entonces simplemente se debe utilizar la palabra clave estándar de Java, `synchronized`, que producirá que invocaciones de métodos se bloqueen si otro objeto ya está invocando el método.

Si se desea tener múltiples llamadas desde el mismo hilo cliente COM, el mismo hilo Java del runtime de J-Integra se utiliza para invocar los métodos correspondientes en objetos Java, entonces se puede establecer una propiedad de J-Integra que produce un hilo dedicado asignado a cada hilo cliente COM. Se pueden modificar el número máximo de hilos por defecto y el tiempo que los hilos establecen como desocupado antes de terminar, al establecer las propiedades apropiadas. Estas propiedades no son soportadas actualmente cuando se hacen llamadas IDispatch (Late-Bound) en modo nativo.

Java-COM. J-Integra está diseñada para manejar múltiples requerimientos simultáneos a objetos COM desde Java.

La máquina DCOM del runtime de J-Integra permite múltiples requerimientos simultáneos al mismo objeto COM, así como también a diferentes objetos COM, y ya que J-Integra se comunica con DCOM, y DCOM está sobre RPCs, J-Integra divide grandes invocaciones de métodos en múltiples RPC PDUs, y reensambla las respuestas que se dividen.

Si el objeto COM remoto maneja correctamente múltiples requerimientos simultáneos es enteramente dependiente de su modelo e implementación de hilos.

1.3.7 Acceso

COM-Java. J-Integra soporta acceso Late Bound (IDispatch) y Early Bound (vtbl). El acceso Late Bound significa que no hay información disponible en tiempo de compilación acerca del objeto que se está accediendo, todo se evalúa dinámicamente en tiempo de ejecución. Esto significa que sólo hasta que se corre el programa se sabe si los métodos y propiedades que se están accediendo realmente existen. El acceso Early Bound, por el contrario, proporciona información acerca del objeto que se está accediendo mientras se está construyendo el programa.

El acceso Late Bound es más lento que el acceso Early Bound, y ya que no hay información disponible acerca de los objetos que se están accediendo, es más propenso a errores de usuario.

Desde una perspectiva COM, los objetos COM son instanciados ya sea utilizando una cadena legible por el usuario llamada "moniker"¹⁵, o utilizando un identificador único legible por la máquina, llamado Class ID (Identificador de Clase).

J-Integra mapea monikers y Class IDs a clases Java que corren en una JVM específica, identificada por un JVM ID. Un JVM ID es simplemente un nombre corto.

1. Cuando una JVM se inicia, se registra el hecho de que está corriendo utilizando un método con el JVM ID (Identificador de Máquina Virtual de Java).
2. Cuando la JVM se corre, se define una propiedad Java que le dice al runtime de J-Integra por qué puerto TCP/IP se escuchan los requerimientos DCOM.
3. En cada cliente Windows que necesita acceder la JVM, se registra el nombre y la localización de la JVM utilizando la herramienta regjvmsvc o regjvm de J-Integra.

Al hacer esto, se pueden crear instancias de cualquier clase Java pública cargable por la JVM, que tenga un constructor por defecto. Se podrían entonces, invocar todos los métodos públicos y acceder todas las propiedades públicas (incluyendo los estáticos) en el nuevo objeto.

❖ **Early Binding (Static Access).** La herramienta java2com de J-Integra analiza clases Java y produce:

- Un archivo IDL (Interface Definition Language) COM, y
- Código "marshalling" DCOM Java puro utilizado por el runtime de J-Integra para facilitar el acceso a objetos Java desde COM utilizando acceso vtable (Early Binding).

Por cada interfaz Java pública que java2com descubre, crea una definición de interfaz COM correspondiente. Por cada clase Java pública que java2com descubre, crea una definición de interfaz COM correspondiente. Además si la clase Java tiene un constructor público por defecto, entonces java2com genera una clase COM.

En seguida se compila el archivo IDL generado utilizando la herramienta MIDL de Microsoft. Ésta viene con Visual C++ y se puede descargar desde el sitio web de Microsoft. Esto producirá una librería de tipo que se debe registrar con la herramienta regtlb. Esta herramienta registra una librería de tipo en un cliente Windows COM que desea acceder objetos Java utilizando el mecanismo Early Binding de COM.

Para registrar una librería de tipo se especifica el nombre del archivo de la librería de tipo a ser registrada y el JVM ID de la JVM en la que se encuentran las clases COM descritas en la librería de tipo.

Si la librería de tipo se genera a partir de un archivo IDL, que a su vez es generado por la herramienta java2com de J-Integra, entonces el comando regtlb puede determinar

¹⁵ Moniker, cadena que permite instanciar un objeto . Se prefiere su uso en Inglés.

automáticamente el nombre de la clase Java que corresponde a cada clase COM en la librería de tipo. Para desregistrar una librería de tipo se especifica el nombre del archivo de la librería de tipo. También es posible desregistrar todas las librerías de tipo registradas por regtlb. Esta herramienta no sirve para registrar librerías de tipo estándar.

- ❖ **Late Binding (Dynamic Access).** Si se utiliza el mecanismo Late Binding no se utilizan las herramientas java2com ni regtlb.

Java-COM. Por cada clase COM que la herramienta com2java encuentra en un librería de tipo, genera una clase Java que se puede utilizar para acceder la clase COM. Estas clases generadas tienen varios constructores:

- ❖ El constructor por defecto, que crea una instancia de la clase COM en el host local, sin autenticación.
- ❖ Un segundo constructor, que crea una instancia de la clase COM en un host específico, sin autenticación.
- ❖ Un tercer constructor, que crea una instancia de la clase COM en el host local, con autenticación específica.
- ❖ Un cuarto constructor, que crea una instancia de la clase COM en un host especificado, con autenticación específica.
- ❖ Un constructor final, que se puede utilizar para wrappear una referencia a un objeto retornado, la cual referencia una instancia de una clase COM. En otras palabras, se puede utilizar para acceder una referencia a una clase COM que fue retornada por otra clase COM (por una llamada a un método, o a través de una propiedad o evento). Si un método o propiedad retorna una referencia a una clase COM, entonces la herramienta com2java genera automáticamente una declaración del método, que retorna la clase Java apropiada, si la clase COM retornada está definida en la misma librería de tipo.

Por cada interfaz definida en la librería de tipo, la herramienta com2java genera dos archivos Java: una interfaz Java y una clase Java, y por cada enumeración, genera una interfaz Java que contiene definiciones constantes de cada elemento en la enumeración.

1.1.4 Despliegue

COM-Java

- ❖ **Modo DCOM.** En este escenario un cliente COM basado en Windows accede objetos Java corriendo en cualquier entorno.
 - **La máquina cliente COM.** Los siguientes archivos se requieren en la máquina cliente COM en tiempo de ejecución:
 - ✓ jintmk.dll. El moniker de J-Integra
 - ✓ Si se utiliza Early Binding, se necesitará la librería de tipo generada a través de la herramienta java2com.

Los siguientes archivos se requieren para configurar la máquina cliente COM, pero no se requieren en tiempo de ejecución:

- ✓ regjvmcmd.exe (y su archivo fuente), o un archivo script generado por la herramienta regjvm.
- ✓ Si se utiliza Early Binding, regtlb.exe (y su archivo fuente).

- **La máquina servidor Java.** Los siguientes archivos se requieren en la máquina servidor Java:
 - ✓ Clases servidor Java.
 - ✓ El runtime de J-Integra (jintegra.jar).
 - ✓ Si se está utilizando Early Binding, las clases “proxy” Java puras generadas por java2com.
- ❖ **Modo nativo (en y fuera de proceso).** En este escenario un cliente COM accede un objeto Java corriendo en la misma máquina del cliente. J-Integra utiliza código nativo para facilitar la integración. Los siguiente archivos se requieren en tiempo de ejecución:
 - jintmk.dll, el moniker de J-Integra.
 - ntvinv.dll (y su archivo fuente), para tomar la identidad de login actual.
 - Las clases servidor Java.
 - El runtime de J-Integra (jintegra.jar).
 - Si se utiliza Early Binding, la librería de tipo generada por la herramienta java2com.
 - Si se utiliza Early Binding, las clases “proxy” Java puras generadas por java2com.

Los siguientes archivos se requieren para configurar la máquina, pero no se requieren en tiempo de ejecución:

- Regjvmcmd.exe (y su archivo fuente), o un archivo script generado por la herramienta regjvm.
- Si se utiliza Early Binding, regtlb.exe (y su archivo fuente).

Si se quiere que la JVM corra fuera de proceso, se debe utilizar el comando regjvm para registrarla como nativa. Además se comienza la JVM en modo nativo, definiendo una propiedad para habilitar este modo.

Si se quiere que la JVM corra en proceso, se debe utilizar el comando regjvm, pero esta vez se especifican parámetros adicionales.

Por defecto, J-Integra llama métodos COM utilizando Early Binding, tanto en modo DCOM como en modo Nativo, pero J-Integra también puede utilizar Late Binding en ambos modos.

Java-COM

- ❖ **Modo DCOM (Java puro).** En este escenario un cliente Java puro corriendo en cualquier plataforma accede un servidor COM basado en Windows. La autenticación se establece en el código cliente. Generalmente no se requiere software J-Integra en la máquina servidor COM.
 - **La máquina cliente Java.** Los siguientes archivos se requieren en la máquina cliente Java:
 - ✓ Las clases cliente Java.
 - ✓ Las clases “proxy” Java puro generadas por com2java.
 - ✓ El runtime de J-Integra (jintegra.jar).

No se requiere ningún otro software (tal como una infraestructura DCOM) en la máquina cliente Java. El cliente Java puede correr en cualquier parte que tenga una JVM.

- **La máquina servidor Windows.** Generalmente no se requiere software J-Integra en la máquina servidor Windows, aunque se puede requerir el comando setdllhost para configurar DLLHOST si el componente COM está en una DLL.

Se necesitará correr la herramienta DCOMCNFG para conceder al usuario cliente permisos de lanzamiento y de acceso por defecto.

- ❖ **Modo DCOM con autenticación nativa.** Este escenario es idéntico al previo, excepto que en lugar de configurar explícitamente el dominio, el usuario y el password en el código cliente Java, J-Integra automáticamente toma la identidad del usuario corriendo el cliente Java. Esto solamente funciona al correr el cliente Java bajo Windows.
 - **La máquina cliente Java.** Además de los archivos listados para el escenario Java puro, se requiere la DLL de autenticación de código nativo, ntvauth.dll.
 - **La máquina servidor Windows.** Igual al escenario Java puro.
- ❖ **Modo nativo.** En este escenario J-Integra utiliza código nativo en lugar de DCOM para invocar métodos en el componente COM de un cliente Java. El cliente Java corre bajo Windows.
 - **La máquina cliente Java.** Los siguientes archivos se requieren en la máquina cliente Java en tiempo de ejecución:
 - ✓ Las clases cliente Java.
 - ✓ Las clases “proxy” Java puras generadas por com2java.
 - ✓ El runtime de J-Integra (jintegra.jar).
 - ✓ El archivo ntvinv.dll.

Además cuando se corre la JVM, se define una propiedad de J-Integra para habilitar el modo nativo desde la línea de comandos o en el código Java.

Un componente COM se debe registrar en la máquina cliente Java utilizando el mecanismo de registro COM estándar. El componente puede ser local o remoto.

J-Integra puede utilizar código nativo (DLLs) para hacer el puente. No se tiene que utilizar el modo nativo. Por defecto, J-Integra utiliza DCOM y el modo nativo se debe habilitar explícitamente. El código Java permanece igual si está utilizando modo nativo o modo DCOM.

1.3.9 Herramientas

CheckConfig. Imprime la información acerca de la plataforma y su configuración. Se puede utilizar la salida de esta herramienta para hacer chequeos de que el sistema Windows está configurado correctamente para correr DCOM.

com2java. La herramienta com2java de J-Integra lee información de una librería de tipo y genera archivos Java que se pueden utilizar para acceder las clases e interfaces COM definidas en esa librería de tipo.

Las librerías de tipo contienen información de clases, interfaces y otros constructores COM. Ellas son generadas típicamente por herramientas de desarrollo tales como Visual C++ y Visual Basic.

La herramienta com2java procesa tres clases de constructores en una librería de tipo: enumeraciones, interfaces COM (incluyendo interfaces fuente -eventos), clases COM.

- ❖ **Enumeraciones.** Si una librería de tipo contiene una enumeración, J-Integra generará una interfaz Java que contiene definiciones constantes para cada elemento en la enumeración.
- ❖ **Interfaces.** Existen dos clases de interfaces COM que J-Integra maneja. Él maneja interfaces Dispatch, cuyos métodos solamente pueden ser accedidos utilizando el mecanismo IDispatch de

COM, y maneja interfaces 'duales', cuyos métodos pueden ser invocados directamente (acceso vtlb).

Por cada interfaz definida en una librería de tipo, la herramienta com2java genera dos archivos Java: una interfaz Java, y una clase Java. El nombre de la interfaz Java generada es el mismo nombre de la interfaz COM. El segundo archivo que com2java genera es una clase Java que contiene código utilizado para acceder los objetos COM que implementa la interfaz, y también código para permitir a objetos COM invocar métodos en clases Java que implementan la interfaz. El nombre de la clase Java generada es el nombre de la interfaz con la palabra Proxy agregada a él.

- ❖ **Clases.** Una clase COM implementa una o más interfaces COM, en el mismo modo en que una clase Java puede implementar una o más interfaces Java.

Para cada clase COM en una librería de tipo, la herramienta com2java genera una clase Java correspondiente con el mismo nombre de la clase COM. Las versiones beta anteriores de J-Integra no manejaron una clase implementando múltiples interfaces, sin embargo esta versión de J-Integra lo hace.

java2com. La herramienta java2com de J-Integra analiza clases y obtiene:

- ❖ Un archivo IDL COM, y
- ❖ Código "marshalling" DCOM Java puro utilizado por el runtime de J-Integra para facilitar el acceso a objetos Java desde COM utilizando acceso vtable (Early Binding).

Él entonces realiza el mismo análisis sobre cualquier clase o interfaz utilizada en parámetros o campos en las clases/interfaces dadas, recursivamente, hasta que todas las clases e interfaces Java accesibles en esta manera han sido analizados.

El nombre del archivo IDL de la interfaz COM también se utiliza para el nombre de la librería de tipo generada cuando se compila el archivo IDL utilizando el compilador MIDL de Microsoft.

Por cada interfaz Java pública que java2com descubre, crea una definición de interfaz COM correspondiente. Por cada clase Java pública que java2com descubre, crea una definición de interfaz COM correspondiente. Además si la clase Java tiene un constructor público por defecto, java2com genera una clase COM. Si una clase Java puede generar eventos Java, la clase COM generada tiene interfaces fuente (eventos COM) que corresponden a los eventos soportados por la clase Java.

Herramienta de licencia. J-Integra tiene un nuevo mecanismo de licenciamiento. La licencia es un archivo XML, se llama jintegra.xml. Este archivo de licencia no es legible por el usuario y su manipulación puede causar que el producto deje de funcionar. Debido a que los productos se pueden instalar y actualizar, una herramienta de gestión de licencia se lanza con el runtime, para asistir al realizar estas tareas. Este archivo no viene con el runtime, sino que es enviado desde Intrinsyc.

La herramienta de licencia J-Integra es una herramienta de línea de comandos utilizada para ver, instalar, y actualizar la información de licenciamiento de J-Integra. Esta herramienta trabaja en tanto la línea de comandos de Windows como en el "shell¹⁶" de UNIX.

La herramienta de licencia tiene tres funciones y opciones de línea de comandos: ver, instalar y actualizar.

regjvm y regjvmcmd. Estas herramientas registran la localización de una JVM en un cliente Windows que necesita crear instancias de clases Java en la JVM. La herramienta regjvm permite crear y

¹⁶ Shell, intérprete de órdenes de un sistema operativo. Se prefiere su uso en Inglés.

gestionar todas las referencias a JVMs en una máquina. regjvmcmd es la versión de línea de comandos y regjvm es una versión gráfica de la misma herramienta.

En la forma más simple, se especifica:

- ❖ Un JVM ID (que corresponde con el nombre utilizado en el método de registro).
- ❖ Y el Binding que se utiliza para acceder la JVM, con el nombre de la máquina que corre la JVM y el puerto TCP/IP especificado cuando se inicia la JVM, al establecer la propiedad del puerto DCOM de J-Integra.

Si ya no se necesita tener registrada la JVM, o si se desea cambiar su registro, primero se la debe desregistrar utilizando regjvmcmd/unregister con el JVM ID.

Algunas plataformas anteriores no soportan la convención de nombramiento de moniker utilizada por la herramienta regjvm. Para estas plataformas, se utiliza la herramienta regprogid.

Una JVM se puede acceder desde clientes COM utilizando uno de tres modos diferentes: modo DCOM, modo nativo fuera de proceso y modo nativo en proceso.

Regprogid. Se debe utilizar un moniker para crear clases Java desde otras plataformas. El moniker utilizado en J-Integra se llama "JintMk.dll". Se debe especificar el nombre basado en la convención de nombramiento de moniker.

Algunas plataformas anteriores no aceptan la convención de nombramiento de moniker, y solamente aceptan un progld (Identificador de Programa). Si se está utilizando un programa que no acepta información de moniker, se tiene que mapear este nombre a un progld para que la plataforma lo acepte. regprogid es la herramienta que se utiliza para hacer este mapeo.

Para mapear un progld a una clase Java (la cual realmente es un moniker), se especifica el nombre progld que se quiere utilizar y en seguida el nombre de la JVM registrada y de la clase Java. Si el progld proporcionado ya ha sido mapeado a un moniker, regprogid no mapeará el progld. Se lo debe desregistrar primero y en seguida re-registrarlo. También se puede remover el mapeo de un progld. Además, existe un comando que lista todas las clases registradas con las herramientas de J-Integra. La lista contiene información de cada clase.

Setdllhost. Las DLLs necesitan ser cargadas en un proceso corriendo ya que no pueden correr por ellas mismas. La infraestructura DCOM en Windows tiene una facilidad para manejar esto y se llama surrogacy. Básicamente se configura el registro para indicar el nombre de un programa sustituto que cargará la DLL bajo demanda. Microsoft ha proporcionado tal programa sustituto que puede tener casi todos los objetos COM que han sido implementados como DLLs, este programa se llama dllhost.exe.

Setdllhost.exe es un programa proporcionado por J-Integra para establecer que el sustituto por defecto para un servidor en proceso (DLL) especificado es dllhost.exe. La razón por la que esto es necesario es porque un cliente COM no puede comunicarse remotamente con un servidor en proceso, debe estar en un ejecutable. DLLHOST hace esto al crear varias entradas en el registro.

El uso de sustitutos para acceder DLLs no es una especificación de J-Integra, sino una especificación COM. Setdllhost simplemente sigue la especificación COM para establecer apropiadamente el sustituto por defecto. Se sugiere utilizar esta herramienta a menos que se conozca qué entradas de registro son necesarias, y cómo trabajar con el registro.

No se tiene que correr setdllhost si se corre en modo nativo. En este caso, la DLL corre en proceso y no necesita un sustituto.

1.3.10 Controles ActiveX. Un control ActiveX es una clase de componente COM que generalmente requiere estar en un contenedor GUI. Los controles ActiveX típicamente tienen la extensión .ocx.

J-Integra permite incorporar un Control COM ActiveX dentro de una GUI Java como si el Control ActiveX fuera un componente GUI. Eso permite acceder los métodos y propiedades del control y suscribirse a sus eventos utilizando los mecanismos estándar de Java.

1.3.11 Servicios Web. Se puede utilizar un toolkit de Servicios Web cualquiera para tomar ventaja del puente java-COM de J-Integra. Para esto, se crea un Servicio Web Java que utiliza el cliente COM y se exponen los servicios proporcionados por la clase Java utilizando el protocolo SOAP. Además se crea una aplicación cliente Java que utiliza SOAP para comunicarse con el Servicio Web.

1.3.12 EJB. J-Integra también se puede utilizar para acceder EJB desde COM. Además Intrinsic proporciona una Arquitectura de Conexión genérica a J2EE (JCA-COM plugin), que permite a EJBs acceder componentes COM existentes, a través de un adaptador de recursos simple desplegado en los servidores de aplicaciones que cumplen con J2EE.

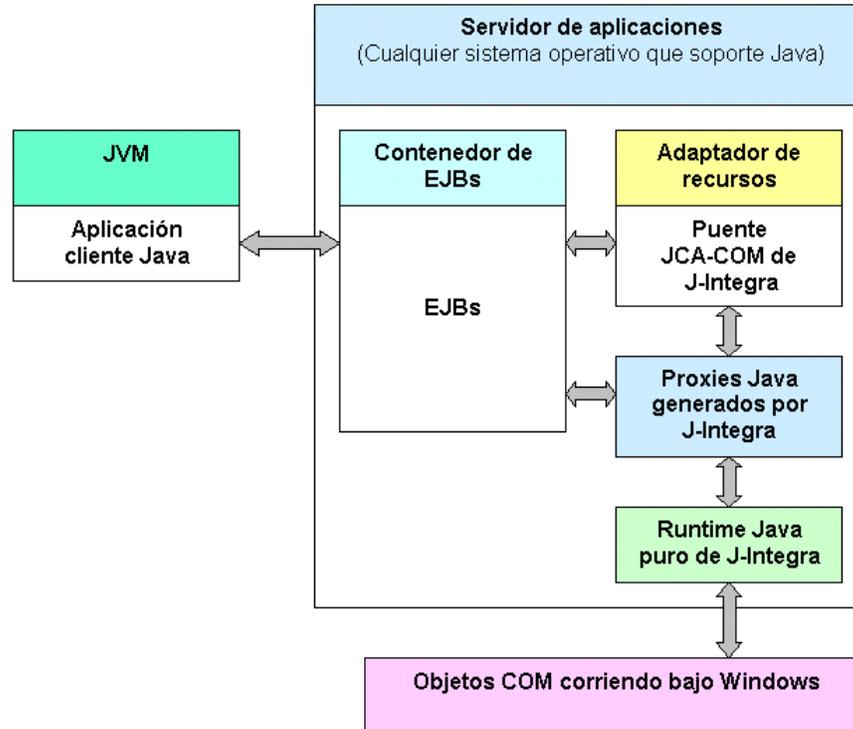
COM-EJB. J-Integra permite al desarrollador COM acceder EJBs como si fueran objetos COM corriendo en el sistema local. J-Integra incluye todas las librerías necesarias para acceder EJBs desde una aplicación COM.

Se pueden acceder EJB que se encuentran en la plataforma HP Bluestone Total-E-Server 7.3 Application Server , iPlanet Application Server, JBoss Application Server, JRun Application Server, Oracle 8i, Oracle9i, BEA WebLogic Platform y IBM WebSphere Application Server.

Java-EJB, JCA-COM. El plugin JCA-COM de J-Integra toma ventaja de JCA (J2EE Connector Architecture) para proporcionar un enlace entre componentes COM y EJBs. Con el plugin JCA-COM, se pueden acceder componentes COM desde EJBs como si estos componentes estuvieran implementados en Java, como se muestra en la Figura 1.5. De esta forma se puede tomar ventaja de la potencia de EJBs sin sacrificar la inversión existente en tecnología COM.

El plugin JCA-COM es un adaptador de recursos genérico. Se despliega el JCA genérico una vez, en un servidor de aplicaciones J2EE, y se puede acceder cualquier aplicación que utilice J-Integra. Se puede utilizar cualquier servidor de aplicaciones, como por ejemplo en BEA Weblogic Application Server y Sun J2EE™ reference application server.

Figura 1.5 Comunicación Java-EJBs utilizando JCA-COM



1.3.13 WMI. WMI ofrece dos APIs que utilizan DCOM. Es posible acceder ambas APIs utilizando J-Integra. Sin embargo, solamente la Interfaz Scripting es accesible utilizando modo DCOM. Esto es, solamente la Interfaz Scripting es accesible desde una máquina no Windows.

Acceder la Interfaz Scripting WMI, a través de J-Integra, desde una máquina no Windows, requiere establecer un ejecutable sustituto (dllhost.exe) para la DLL que contiene los objetos COM WMI (wbemdisp.dll). El sustituto se puede establecer utilizando la herramienta setdllhost de J-Integra. Una vez establecido, DCOM se debe configurar para permitir acceso a este sustituto y WMI se debe configurar para permitir acceso. El sustituto debe ser establecido solamente en una máquina Windows. Ningún software relacionado con J-Integra se requiere en ninguna de las máquinas Windows durante la operación del sistema (el sustituto dllhost.exe es un componente estándar de DCOM).

No es posible acceder la interfaz No Scripting utilizando modo DCOM. Cuando se accede WMI, se necesita obtener el objeto IWbemServices. Esto se hace a través del objeto IWbemLocator. Una vez el objeto ha sido obtenido, se accede Servicios WMI en máquinas remotas a través del mecanismo DCOM normal. El objeto IWbemLocator siempre es un servidor COM en proceso (además, la interfaz IWbemLocator se declara como local en el archivo IDL que viene con el SDK WMI). Otros objetos WMI tienen la misma limitación.

Esto significa que solamente se puede acceder el objeto IWbemLocator si WMI está instalado en la máquina que tiene la aplicación cliente. Por lo tanto, para acceder la Interfaz No Scripting se necesita correr J-Integra en Modo Nativo. En este modo, J-Integra utiliza librerías nativas para instanciar los objetos WMI.

En un futuro, cuando el "framework" .NET esté instalado en la mayoría de equipos Windows, se podría utilizar Ja.NET en lugar de J-Integra, por esto a continuación se detalla esta herramienta.

1.4 Ja.NET

Ja.Net es un puente entre el mundo de Java y el mundo de Microsoft.NET. Ja.Net es un puente bi-direccional.

Ja.NET utiliza .NET Remoting, el nuevo protocolo de objetos distribuido de Microsoft. .NET Remoting combina lo mejor de las tecnologías distribuidas más recientes, Servicios Web, con lo mejor de las tecnologías distribuidas anteriores tales como DCOM, CORBA y RMI. .NET Remoting se utiliza dentro de .NET para permitir que componentes CLR (common language runtime), en diferentes dominios de aplicación, se comuniquen con otros. Es importante recordar que CLR soporta muchos lenguajes, como por ejemplo C#, Visual Basic.NET, ASP.NET y C++. Al utilizar Ja.NET, los componentes Java parecen ser componentes CLR y los componentes CLR parecen ser componentes Java.

Una de las características más interesante de .NET Remoting es que el protocolo de transporte y el formato de datos son configurables y extensibles. Actualmente .NET y Ja.NET soportan los protocolos de transporte tanto HTTP como TCP/IP, y ya sea SOAP o un formato de datos binario.

Existen cuatro escenarios de despliegue:

- ❖ Java accede .NET sobre TCP/IP
- ❖ Java accede .NET sobre HTTP
- ❖ .NET accede Java sobre HTTP
- ❖ .NET accede Java sobre TCP/IP

1.4.1 Componentes. Ja.NET tiene de cinco componentes diferentes.

El runtime de Ja.NET (janet.jar). El núcleo del runtime está almacenado en Janet.jar. Es el único componente de Ja.NET que se requiere en tiempo de ejecución. Requiere un archivo de configuración (por defecto Janet.xml, que se puede sobrescribir) que es generado durante el despliegue por Janetor. El archivo de configuración se requiere en tiempo de ejecución.

Janetor. Janetor se utiliza para crear un archivo de configuración para el runtime de Ja.NET. Este archivo:

- ❖ Tiene la localización, el tipo y el formato del canal de objetos remotos.
- ❖ Tiene el nombre del equipo, el puerto y el tipo de objetos locales.
- ❖ Sirve para instalar una licencia.
- ❖ Permite Habilitar/deshabilitar logging.
- ❖ Permite exportar a un archivo WAR (Web Application Archive), para el despliegue del runtime de Ja.NET en un servidor de aplicaciones basado en J2EE.

GenJava (genjava.jar). GenJava es una herramienta de desarrollo utilizada para generar "proxies" Java a partir de clases .NET. Trabaja en colaboración con la herramienta GenService. GenJava se utiliza cuando se acceden componentes .NET desde Java.

GenNet (gennet.jar). GenNet es una herramienta de desarrollo utilizada para generar "proxies" .NET a partir de clases Java. Trabaja en colaboración con la herramienta GenService. GenNet se utiliza al acceder componentes Java desde .NET.

GenService. GenService es una herramienta de desarrollo que actúa como parte de la generación de "proxies". Es el único componente no Java de Ja.NET y es desplegable solamente en plataformas Windows. Al generar "proxies" él actúa como el puente entre las herramientas basadas en Java, GenJava y GenNet, y la plataforma .NET. GenService nunca es invocado por el usuario final directamente.

1.4.2 Mapeo de eventos Java a eventos .NET. Ja.NET actualmente soporta el mapeo de eventos Java a eventos .NET. Los siguientes constructores Java están mapeados a constructores .NET.

- ❖ Java event listener
- ❖ Java event source

1.4.3 Autenticación. El runtime de Ja.Net soporta autenticación tanto Basic como Digest HTTP para clientes.

1.4.4 Cifrado. Ja.Net ahora soporta cifrado sobre TCP/IP o HTTP en funcionalidad Beta.

1.4.5 .NET Remoting. Microsoft.NET Remoting proporciona un “framework” que permite a objetos interactuar con otros a través de dominios de aplicación. El “framework” proporciona un número de servicios, incluyendo activación y soporte del tiempo de vida, así como también canales de comunicación responsables de transportar mensajes a y desde aplicaciones remotas. Los Formatos se utilizan para codificar y decodificar los mensajes antes de transportarlos por el canal. Las aplicaciones puede utilizar codificación binaria donde el desempeño es crítico, o codificación XML donde la interoperabilidad con otros “frameworks” remotos es esencial. Toda la codificación XML utiliza el protocolo SOAP al transportar mensajes desde un dominio de aplicación a otro. Remoting fue diseñado con seguridad.

Gestionar el tiempo de vida de objetos remotos sin el soporte del “framework” subyacente es a menudo difícil. .NET Remoting proporciona un número de modelos de activación para escoger. Estos modelos son de una de dos categorías:

- ❖ Objetos activados por cliente.
- ❖ Objetos activados por servidor.

Un objeto activado por cliente es un objeto cuya creación y destrucción son controladas por la aplicación cliente. Los objetos activados por cliente están bajo el control de un gestor de tiempo de vida basado en “lease”¹⁷ que garantiza que el objeto es colectado como basura cuando su “lease” expira. El tiempo de vida de un objeto activado por servidor es gestionado por el servidor remoto. En el caso de objetos activados por servidor, los desarrolladores tienen la posibilidad de seleccionar ya sea un modelo Single Call o Singleton. Los objetos Single Call sólo pueden trabajar con una solicitud entrante, mientras que los objetos Singleton sirven a varios clientes. El tiempo vida de los Singletons también es controlado por tiempo de vida basado en “lease”.

¹⁷ Lease, arriendo. Se prefiere su uso en Inglés.

ANEXO B

Modelado

2.1 CASOS DE USO

Actores del sistema

Usuario: persona con derechos de administrador sobre el sistema donde está instalada la aplicación y que tiene derecho de acceder y utilizar la aplicación.

Administrador: persona que configura cuáles cuentas de administrador pueden acceder y utilizar la aplicación.

Caso de uso: VALIDAR DATOS

Actor: administrador y usuario.

Propósito: permitir acceder al sistema únicamente a las cuentas de administrador seleccionadas por el Administrador que se encuentran almacenadas en una base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor entra al sistema.	2. Solicita login y password.
3. Introduce los datos solicitados por el sistema.	4. Valida los datos con la base de datos y presenta el menú principal que tiene las opciones: MACs-IPs, HUBs-SWITCHES, Equipo de usuario, Usuarios y contraseñas.

Cursos alternos:

- ❖ Línea 3: El actor no introduce el login y el password. El sistema solicita nuevamente los datos.
- ❖ Línea 4: Los datos del actor corresponden a un Usuario. El sistema no muestra la opción Usuarios y Contraseñas.
- ❖ Línea 4: Los datos del actor no son válidos. El sistema solicita nuevamente el login y el password.

Caso de uso: CONSULTAR USUARIOS

Actor: administrador.

Propósito: permitir ver el login y los privilegios (Administrador o Usuario de la aplicación) de las cuentas de administrador del equipo en el que está instalada la herramienta, que se encuentran almacenadas en la base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Usuarios y contraseñas del menú principal.	2. Consulta en la base de datos y muestra el login y los privilegios de las cuentas. Además presenta las opciones Adicionar, Modificar y Eliminar.

Caso de uso: ADICIONAR USUARIO

Actor: administrador.

Propósito: permitir adicionar una cuenta de administrador del equipo en el que se encuentra instalada la aplicación y establecer si es un Administrador o un Usuario de la aplicación.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso comienza cuando un actor escoge la opción Adicionar.	2. Solicita los datos de la cuenta.
3. Introduce los datos.	4. Verifica los datos y los almacena en la base

	de datos.
	5. Muestra la información actualizada.

Cursos alternos:

- ❖ Línea 4: El administrador no introduce el login de la cuenta. El sistema solicita nuevamente los datos.

Caso de uso: MODIFICAR USUARIO

Actor: administrador.

Propósito: permitir modificar la información de una cuenta almacenada en la base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la cuenta que desea modificar y la opción Modificar.	2. Consulta la base de datos y presenta los datos a modificar.
3. Modifica los datos.	4. Verifica los datos y los almacena en la base de datos.
	5. Muestra la información actualizada.

Caso de uso: ELIMINAR USUARIOS

Actor: administrador.

Propósito: permitir eliminar una cuenta de la base datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la(s) cuenta(s) que desea eliminar y la opción Eliminar.	2. Elimina de la base de datos la(s) cuenta(s) y despliega la información actualizada.

Caso de uso: OBTENER DIRECCIONES MAC

Actor: administrador y usuario.

Propósito: permitir obtener en forma automática las direcciones MAC de las direcciones IP que previamente hayan sido almacenadas en una base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando la validación de los datos de un actor es exitosa.	2. Obtiene mediante comandos de DOS la dirección MAC del equipo local y de los equipos remotos encendidos, cuya dirección IP se encuentra almacenada en la base de datos.
	3. Almacena las direcciones MAC obtenidas en la base de datos.
	4. Continúa realizando estas operaciones mientras existan en la base de datos direcciones IP sin su correspondiente dirección MAC.

Cursos alternos:

- ❖ Línea 2: El equipo remoto no está encendido. El sistema obtiene la dirección MAC de la siguiente dirección IP que se encuentra en la base de datos.

Caso de uso: CONSULTAR DIRECCIONES

Actor: administrador y usuario.

Propósito: permitir ver las direcciones IP que previamente hayan sido almacenadas en una base de datos con la respectiva dirección MAC obtenida por el sistema.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción MACs-IPs del menú principal.	2. Consulta en la base de datos y muestra las direcciones IP con su respectiva dirección MAC, además de las opciones Adicionar, Modificar, Eliminar e Insertar rango.

Caso de uso: ADICIONAR DIRECCIÓN IP**Actor:** administrador y usuario.**Propósito:** permitir adicionar una dirección IP a la base de datos para que el sistema obtenga la dirección MAC que le corresponde.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso comienza cuando un actor escoge la opción Adicionar.	2. Solicita una dirección IP.
3. Introduce el dato.	4. Verifica el dato y lo almacena en la base de datos.
	5. Muestra la información actualizada.

Cursos alternos:

- ❖ Línea 4: El administrador no introduce el dato. El sistema solicita nuevamente los datos.
- ❖ Línea 4: La dirección IP no es válida. El sistema la solicita nuevamente.

Caso de uso: MODIFICAR DIRECCIONES**Actor:** administrador y usuario.**Propósito:** permitir modificar una dirección IP y/o su respectiva MAC de la base datos.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la dirección IP que desea modificar y la opción Modificar.	2. Consulta la base de datos y presenta los datos a modificar.
3. Modifica los datos.	4. Verifica los datos y los almacena en la base de datos.
	5. Muestra la información actualizada.

Cursos alternos:

- ❖ Línea 4: La dirección IP y/o MAC no son válidas. El sistema solicita nuevamente los datos.

Caso de uso: ELIMINAR DIRECCIONES**Actor:** administrador y usuario.**Propósito:** permitir eliminar una dirección IP y su respectiva MAC de la base datos.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la(s) dirección(es) IP que desea eliminar y la opción Eliminar.	2. Elimina de la base de datos la(s) dirección(es) IP con su correspondiente dirección MAC.
	3. Muestra la información actualizada.

Caso de uso: INSERTAR RANGO**Actor:** administrador y usuario.**Propósito:** permitir adicionar un rango de direcciones IP para que el sistema obtenga la direcciones MAC correspondientes.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce el rango que desea insertar.	2. Verifica el rango introducido.

	3. Determina y almacena en la base de datos las posibles direcciones IP del rango.
	4. Muestra la información actualizada.

Cursos alternos:

- ❖ Línea 2: El rango del rango introducido no es válido. El sistema solicita nuevamente el rango.

Caso de uso: CONSULTAR HUBS-SWITCHES

Actor: administrador y usuario.

Propósito: permitir ver la información, almacenada en la base de datos, de los hubs y de los switches de cada sector de la red.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción HUBS-SWITCHES del menú principal.	2. Despliega los sectores en que se encuentra dividida la red.
3. Selecciona el sector del que desea obtener información sobre los equipos.	4. Consulta la base de datos y muestra información de los hubs y de los switches del sector seleccionado, además de las opciones Adicionar, Modificar, Eliminar, Ver información del equipo y Ver equipos de usuario.

Caso de uso: ADICIONAR HUBS-SWITCHES

Actor: administrador y usuario.

Propósito: permitir adicionar información de un hub o switch a la base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso comienza cuando un actor selecciona la opción Adicionar.	2. Solicita la información del hub o del switch que se almacena en la base de datos.
3. Introduce la información solicitada.	4. Almacena la información en la base de datos.
	5. Muestra la información actualizada.

Caso de uso: MODIFICAR HUBS-SWITCHES

Actor: administrador y usuario.

Propósito: permitir modificar la información de un hub o switch específico que se encuentra almacenada en la base de datos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona el hub o switch que desea modificar y la opción Modificar.	2. Consulta la base de datos y presenta los datos a modificar.
3. Modifica los datos.	4. Almacena la información en la base de datos.
	5. Muestra la información actualizada.

Caso de uso: ELIMINAR HUBS-SWITCHES

Actor: administrador y usuario.

Propósito: permitir eliminar toda la información almacenada en la base de datos de un hub o switch específico.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona los dispositivos de red que desea eliminar de la base de datos y la opción Eliminar.	2. Elimina de la base de datos toda la información asociada con los dispositivos seleccionados.

	3. Muestra la información actualizada.
--	--

Caso de uso: VER INFORMACIÓN SNMP

Actor: administrador y usuario.

Propósito: permitir obtener los datos de mayor importancia y utilidad proporcionados por la MIB II (RFC 1213).

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Ver información del equipo de un dispositivo de red específico.	2. Obtiene a través de SNMP los principales datos del dispositivo proporcionados por la MIB II y los despliega al actor.

Caso de uso: VER EQUIPOS DE USUARIO

Actor: administrador y usuario.

Propósito: permitir obtener en forma automática las direcciones IP, MAC y puerto de los equipos de usuario conectados a un hub o switch específico.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Ver equipos de usuario de un dispositivo de red específico.	2. Obtiene a través de SNMP, utilizando la MIB Bridge, las direcciones MAC y los puertos utilizados por los equipos de usuario conectados al dispositivo de red.
	3. Relaciona las direcciones MAC obtenidas con las direcciones MAC almacenadas en la base de datos y obtiene las direcciones IP correspondientes.
	4. Despliega las direcciones IP, MAC y el puerto utilizado de cada uno de los equipos de usuario conectados al dispositivo.

Caso de uso: CONECTAR A UN EQUIPO DE USUARIO

Actor: administrador y usuario.

Propósito: permitir realizar una conexión a cualquier equipo remoto de la red utilizando una cuenta de administrador de ese equipo.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Equipo de usuario del menú principal.	2. Solicita la dirección IP, el login y el password de una cuenta de administrador del equipo con el que el sistema se va a conectar.
	3. Verifica la dirección IP.
	4. Realiza la conexión con el equipo remoto utilizando la API Scripting de WMI y J-Integra (puente Java-COM seleccionado).

Cursos alternos:

- ❖ Línea 3: La dirección IP no es válida. El sistema solicita nuevamente los datos del equipo remoto.
- ❖ Línea 4: No se puede realizar la conexión. El sistema solicita nuevamente los datos del equipo remoto.

Caso de uso: VER INFORMACIÓN HARDWARE

Actor: administrador y usuario.

Propósito: permitir obtener información acerca de la Motherboard, Procesador, Memoria, BIOS, Disco Duro, CD-ROM Drive, Floppy Drive, Buses, Puerto Paralelo, Puerto Serial, Tarjeta de Red, Teclado, Mouse e Impresora de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Hardware del menú de equipo.	2. Presenta las opciones: a. Motherboard. b. Procesador. c. Memoria. d. BIOS. e. Disco duro. f. CD-ROM drive. g. Floppy drive. h. Buses. i. Puerto paralelo. j. Puerto serial. k. Tarjeta de red. l. Teclado. m. Mouse. n. Impresora.
3. Selecciona una de las opciones de hardware.	4. Obtiene los valores de las propiedades del elemento hardware seleccionado y las despliega.

Caso de uso: CONFIGURAR PROTOCOLO INTERNET (TCP/IP)**Actor:** administrador y usuario.**Propósito:** permitir configurar manualmente la dirección IP, máscara de subred y puerta de enlace u obtener una dirección IP automáticamente a través de DHCP en un equipo remoto. Así como también deshabilitar DHCP o renovar el período de validez de una dirección IP asignada automáticamente.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Tarjeta de red.	2. Obtiene los valores de las propiedades de la(s) tarjeta(s) que tiene el equipo de usuario especificado y las despliega al actor.
	3. Presenta las opciones: a. Habilitar, Deshabilitar y Renovar DHCP, véase sección DHCP. b. Configurar IP y máscara de subred, véase sección Configurar IP. c. Configurar Puerta de enlace, véase sección Puerta de enlace.

Sección: DHCP**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona una tarjeta de red y la opción Configurar IP.	2. Muestra las opciones Habilitar DHCP, Deshabilitar DHCP y Renovar DHCP.
3. Selecciona una opción.	4. Realiza la operación seleccionada por un actor y muestra el resultado de la operación.

Cursos alternos:

❖ Línea 1: El actor no selecciona una tarjeta de red. El sistema le informa al actor sobre esto.

Sección: Configurar IP**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona una tarjeta de red y la opción	2. Solicita la dirección IP y la máscara de subred.

Configurar IP.	
3. Introduce la nueva dirección IP y la máscara de subred.	4. Verifica la dirección IP y la máscara de subred.
	5. Realiza la configuración especificada por el actor.

Cursos alternos:

- ❖ Línea 1: El actor no seleccionó una tarjeta de red. El sistema le informa al actor sobre esto.
- ❖ Línea 4: La dirección IP y/o la máscara de subred no son válidas. El sistema le informa al actor sobre esto y solicita nuevamente los datos.

Sección: Configurar puerta de enlace

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona una tarjeta de red y la opción Configurar IP.	2. Solicita la dirección IP de la puerta de enlace.
3. Introduce la puerta de enlace.	4. Verifica la dirección IP.
	5. Establece la puerta de enlace especificada.

Cursos alternos:

- ❖ Línea 1: El actor no selecciona una tarjeta de red. El sistema le informa al actor sobre esto.
- ❖ Línea 4: La dirección IP no es válida. El sistema le informa al actor sobre esto y solicita nuevamente el dato.

Caso de uso: CONFIGURAR DNS

Actor: administrador y usuario.

Propósito: permitir configurar el DNS preferido y alternativo de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Configurar DNS.	2. Solicita las direcciones IP de los DNSs preferido y alternativo.
3. Introduce los datos solicitados.	4. Verifica los datos introducidos.
	5. Realiza la configuración especificada por el actor.

Cursos alternos:

- ❖ Línea 3: El actor no introdujo la dirección IP del DNS primario. El sistema solicita nuevamente los datos.

Caso de uso: CONFIGURAR NetBIOS

Actor: administrador y usuario.

Propósito: permitir establecer la opción NetBIOS de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la tarjeta de red y la opción NetBIOS.	2. Presenta las opciones para habilitar y deshabilitar NetBIOS
3. Selecciona la opción deseada.	4. Realiza la operación seleccionada y presenta el resultado de la operación.

Cursos alternos:

- ❖ Línea 1: El actor no selecciona una tarjeta de red. El sistema le informa al actor sobre esto.

Caso de uso: CONFIGURAR FILTRADO TCP/IP

Actor: administrador y usuario.

Propósito: permitir establecer los puertos TCP, UDP y los protocolos IP habilitados de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la tarjeta de red y la opción Filtrado TCP/IP.	2. Presenta las opciones: a. Habilitar filtrado, véase sección Habilitar filtrado. b. Deshabilitar filtrado, véase sección Deshabilitar filtrado.

Sección: Habilitar filtrado**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce los puertos TCP y UDP, y los protocolos IP que desea habilitar.	2. Verifica los números de los puertos introducidos.
	3. Realiza la configuración especificada y presenta el resultado de la operación.

Cursos alternos:

- ❖ Línea 2: Los números de los puertos no son válidos. El sistema solicita nuevamente los datos.

Sección: Deshabilitar filtrado**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Deshabilitar filtrado.	2. Realiza la configuración especificada y presenta el resultado de la operación.

Caso de uso: VER INFORMACIÓN SOFTWARE

Actor: administrador y usuario.

Propósito: permitir obtener información acerca del Sistema, Sistema operativo, Particiones, Configuración de arranque, Variables de entorno, Protocolos de red, Grupos de usuarios, Cuentas de usuarios, Perfil de login, Comandos de inicio y Servicios de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Software del menú de equipo.	2. Presenta las opciones: b. Sistema. c. Sistema operativo. d. Particiones. e. Configuración de arranque. f. Variables de entorno. g. Protocolos de red. h. Grupos de usuarios. i. Cuentas de usuarios. j. Perfil de login. k. Comandos de inicio. l. Servicios.
3. Selecciona una de las opciones de software.	4. Obtiene los valores de las propiedades del elemento software seleccionado y las despliega.

Caso de uso: APAGAR/REINICIAR EQUIPO

Actor: administrador y usuario.

Propósito: permitir apagar remotamente un equipo.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Sistema operativo.	2. Obtiene los valores de las propiedades del sistema operativo del equipo remoto y las

	despliega, además de las opciones Reiniciar y Apagar.
3. El actor selecciona una opción.	4. Realiza la operación especificada y presenta el resultado de la operación.

Caso de uso: CREAR PROCESO

Actor: administrador y usuario.

Propósito: permitir crear un proceso en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Procesos.	2. Presenta las opciones: Crear proceso y Terminar proceso.
3. Selecciona la opción Crear proceso.	4. Solicita el nombre del archivo ejecutable del proceso que se quiere crear.
5. Introduce el nombre del archivo.	6. Presenta los archivos con el nombre especificado.
7. Selecciona un archivo.	8. Crea el proceso especificado y muestra el resultado de la operación.

Cursos alternos:

❖ Línea 7: El actor no selecciona un archivo. El sistema le informa al actor sobre esto.

Caso de uso: TERMINAR PROCESOS

Actor: administrador y usuario.

Propósito: permitir terminar procesos en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Procesos.	2. Presenta las opciones: Crear proceso y Terminar proceso.
3. Selecciona la opción Terminar proceso.	4. Muestra los procesos activos del equipo remoto.
5. Selecciona los procesos que desea terminar.	6. Termina los procesos especificados y muestra el resultado de la operación por cada proceso seleccionado.

Cursos alternos:

❖ Línea 7: El actor no selecciona un proceso. El sistema le informa al actor sobre esto.

Caso de uso: CAMBIAR ESTADO DE SERVICIOS

Actor: administrador y usuario.

Propósito: permitir iniciar, pausar, reanudar, detener o eliminar un servicio en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Servicios.	2. Obtiene los valores de las propiedades de los servicios del equipo remoto y las despliega, además de las opciones Iniciar, Pausar, Reanudar, Detener y Eliminar.
3. Selecciona una opción.	4. Realiza la operación especificada y presenta el resultado de la operación.

Cursos alternos:

❖ Línea 4: El actor no selecciona un servicio. El sistema le informa al actor sobre esto.

Caso de uso: BUSCAR ARCHIVOS

Actor: administrador y usuario.

Propósito: permitir buscar en un todos los drives o en un específico, los archivos con un nombre y/o extensión determinados en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Archivos.	2. Presenta las opciones: Buscar archivos, Determinar tamaño, Eliminar todos los archivos.
3. Introduce el nombre del archivo y el drive donde desea realizar la búsqueda.	4. Despliega los archivos con el nombre especificado y presenta las opciones: Copiar, Renombrar, Mover y Eliminar.

Cursos alternos:

- ❖ Línea 3: El actor no introduce el nombre del archivo en forma correcta (nombre.extensión). El sistema le informa al actor sobre esto y solicita nuevamente los datos.

Caso de uso: EJECUTAR OPERACIONES SOBRE ARCHIVOS**Actor:** administrador y usuario.**Propósito:** permitir copiar, renombrar y mover archivos en un equipo remoto.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona un archivo y la opción Copiar, Renombrar o Mover archivos.	2. Obtiene y despliega la ruta del archivo seleccionado.
3. Introduce la nueva ruta.	4. Realiza la operación especificada y despliega el resultado de la operación.

Cursos alternos:

- ❖ Línea 1: El actor no selecciona un archivo. El sistema le informa al actor sobre esto.

Caso de uso: ELIMINAR ARCHIVOS**Actor:** administrador y usuario.**Propósito:** permitir eliminar archivos en un equipo remoto.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona los archivos que desea eliminar y la opción Eliminar.	2. Elimina los archivos seleccionados y despliega el resultado de la operación por cada archivo seleccionado.

Cursos alternos:

- ❖ Línea 1: El actor no selecciona ningún archivo. El sistema le informa al actor sobre esto.

Caso de uso: DETERMINAR TAMAÑO**Actor:** administrador y usuario.**Propósito:** permitir eliminar archivos en un equipo remoto.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce la extensión del archivo y el drive en el que desea realizar la búsqueda.	2. Realiza la búsqueda de los archivos con la extensión especificada.
	3. Obtiene el tamaño total que ocupan los archivos encontrados.
	4. Despliega el tamaño de cada uno de los archivos y el tamaño total.

Cursos alternos:

- ❖ Línea 1: El actor no introduce la extensión del archivo. El sistema le informa al actor sobre esto.

Caso de uso: ELIMINAR TODOS LOS ARCHIVOS**Actor:** administrador y usuario.**Propósito:** permitir eliminar todos archivos con una extensión determinada en todos o en un drive específico de un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Eliminar todos.	2. Elimina los archivos seleccionados y despliega el resultado de la operación de cada archivo.

Cursos alternos:

- ❖ Línea 1: El actor no ha seleccionado la opción Determinar tamaño. El sistema le informa al actor sobre esto.

Caso de uso: INSTALAR PRODUCTO SOFTWARE

Actor: administrador y usuario.

Propósito: permitir instalar un producto software a través del Instalador de Windows (Windows Installer) en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Productos software.	2. Presenta las opciones: Instalar producto y Desinstalar producto.
3. Selecciona la opción Instalar producto.	4. Busca y despliega los archivos que tienen extensión msi.
5. Selecciona el producto que desea instalar.	6. Realiza la instalación y despliega el resultado de la operación.

Cursos alternos:

- ❖ Línea 5: El actor no selecciona un producto. El sistema le informa al actor sobre esto.

Caso de uso: DESINSTALAR PRODUCTO SOFTWARE

Actor: administrador y usuario.

Propósito: permitir desinstalar un producto software a través del Instalador de Windows (Windows Installer) en un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Productos software.	2. Presenta las opciones: Instalar producto y Desinstalar producto.
3. Selecciona la opción Desinstalar producto.	4. Despliega los productos software instalados a través del Instalador de Windows.
5. Selecciona el producto que desea desinstalar.	6. Realiza la desinstalación y despliega el resultado de la operación.

Cursos alternos:

- ❖ Línea 3: El actor no selecciona un producto. El sistema le informa al actor sobre esto.

Caso de uso: VER EVENTOS NT

Actor: administrador y usuario.

Propósito: permitir ver las características de los registros de eventos de un equipo remoto, así como también los eventos registrados en cada uno de ellos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Eventos NT.	2. Despliega las opciones: Registro de aplicación, Registro de seguridad y Registro del sistema.
3. Selecciona el registro del que desea ver información.	4. Despliega los valores de las propiedades del registro seleccionado, además de la opción Ver eventos.
5. Selecciona la opción Ver eventos.	6. Despliega los eventos del registro seleccionado.

Caso de uso: ENVIAR UN E-MAIL POR CAMBIO DE DIRECCIÓN IP**Actor:** administrador y usuario.**Propósito:** permitir que un equipo remoto envíe automáticamente un e-mail a una dirección de correo electrónico especificada cuando se cambia su dirección IP.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Eventos.	2. Despliega la información de configuración para el envío del e-mail, además de las opciones: a. Habilitar evento, véase sección Habilitar evento. b. Modificar evento, véase sección Modificar evento. c. Deshabilitar evento, véase sección Deshabilitar evento.

Sección: Habilitar evento**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Habilitar evento.	2. Solicita los datos de configuración para el envío del e-mail.
3. Introduce los datos.	4. Valida los datos.
	5. Habilita el evento y despliega los datos configurados.

Cursos alternos:

❖ Línea 4: El actor no introduce los datos necesarios para realizar la configuración. El sistema le informa al actor sobre esto.

Sección: Modificar evento**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Modificar evento.	2. Despliega los datos previamente configurados.
3. Modifica los datos.	4. Habilita el evento con los nuevos datos de configuración y despliega los datos configurados.

Sección: Deshabilitar evento**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Deshabilitar evento.	2. Deshabilita el evento.

Caso de uso: CONFIGURAR SISTEMA**Actor:** administrador y usuario.**Propósito:** permitir habilitar o deshabilitar las herramientas de edición del registro, el panel de control, la opción ejecutar, el prompt de comandos, así como también restringir el acceso y esconder los drives en un equipo remoto.**Curso normal de los eventos**

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el actor selecciona la opción Registro del menú de equipo.	2. Presenta las opciones: Sistema, Aplicaciones, Internet y Red.
3. Selecciona la opción Sistema.	4. Muestra el estado actual (habilitado o deshabilitado) de las herramientas de edición

	<p>del registro, panel de control, opción ejecutar y prompt de comandos. Además presenta las siguientes opciones:</p> <ol style="list-style-type: none"> a. Habilitar Herramientas de edición del Registro, veáse sección Configurar Sistema. b. Deshabilitar Herramientas de edición del Registro, veáse sección Configurar Sistema. c. Habilitar Panel de Control, veáse sección Configurar Sistema. d. Deshabilitar Panel de Control, veáse sección Configurar Sistema. e. Habilitar Opción Ejecutar, veáse sección Configurar Sistema. f. Deshabilitar Opción Ejecutar, veáse sección Configurar Sistema. g. Habilitar Prompt de Comandos, veáse sección Configurar Sistema. h. Deshabilitar Prompt de Comandos, veáse sección Configurar Sistema. i. Esconder Drives, veáse sección Drives. j. Mostrar Drives, veáse sección Drives. k. Restringir acceso a Drives, veáse sección Drives. l. Permitir acceso a Drives, veáse sección Drives.
--	---

Sección: Configurar herramientas
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Habilitar o Deshabilitar Herramientas de Edición del Registro, Panel de Control, Opción Ejecutar o Prompt de Comandos.	2. Realiza la operación seleccionada por el actor y despliega el resultado de la operación.

Sección: Configurar drives
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce las letras de los drives a los que desea esconder o restringir el acceso.	2. Realiza la operación seleccionada por el actor y despliega el resultado de la operación.

Cursos alternos:

- ❖ Línea 1: El actor no introduce ningún dato. El sistema le informa al actor sobre esto.

Caso de uso: CONFIGURAR APLICACIONES

Actor: administrador y usuario.

Propósito: permitir obtener las aplicaciones que corren cuando se inicia Windows, restringir al usuario de correr ciertas aplicaciones o restringir las aplicaciones que un usuario puede correr sobre un equipo remoto.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Registro del menú de equipo.	2. Presenta las opciones: Sistema, Aplicaciones, Internet y Red.
3. Selecciona la opción Aplicaciones.	4. Muestra el nombre de las aplicaciones que

	<p>corren cuando se inicia Windows, las aplicaciones restringidas y las aplicaciones que el usuario puede correr. Además le presenta al actor las siguientes opciones:</p> <ol style="list-style-type: none"> Adicionar aplicaciones restringidas, véase sección Adicionar aplicaciones. Eliminar aplicaciones restringidas, véase sección Eliminar aplicaciones. Adicionar aplicaciones permitidas, véase sección Adicionar aplicaciones. Eliminar aplicaciones permitidas, véase sección Eliminar aplicaciones.
--	---

Sección: Adicionar aplicaciones
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce los nombres de los archivos ejecutables de las aplicaciones que desea restringir o permitir.	2. Realiza la operación seleccionada por el actor y despliega el resultado de la operación para cada una de las aplicaciones.

Cursos alternos:

- ❖ Línea 1: El actor no introduce ningún dato. El sistema le informa al actor sobre esto.

Sección: Eliminar aplicaciones
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona las aplicaciones que desea eliminar de la lista de aplicaciones restringidas o permitidas.	2. Elimina las aplicaciones y despliega el resultado de la operación para cada una de las aplicaciones.

Cursos alternos:

- ❖ Línea 2: El actor no selecciona las aplicaciones que desea eliminar. El sistema le informa al actor sobre esto.

Caso de uso: CONFIGURAR PROXY

Actor: administrador y usuario.

Propósito: permitir configurar la dirección y el puerto del proxy, así como también habilitarlo o deshabilitarlo.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Internet.	2. Presenta al actor las siguientes opciones: <ol style="list-style-type: none"> Configurar Proxy. Configurar Restricciones. Configurar Descarga de Archivos.
3. Selecciona la opción Configurar Proxy.	4. Muestra la configuración (dirección y puerto) y el estado (habilitado o deshabilitado) del proxy. Además presenta las siguientes opciones: <ol style="list-style-type: none"> Configurar proxy, véase sección Habilitar proxy. Deshabilitar proxy, véase sección Deshabilitar proxy.

Sección: Habilitar Proxy
Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor introduce la dirección y el puerto del Proxy.	2. Valida el número de puerto.
	3. Habilita el proxy especificado y despliega el resultado de la operación.

Cursos alternos:

- ❖ Línea 2: El actor introduce un número de puerto no válido. El sistema le informa al actor sobre esto y solicita nuevamente los datos.

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Internet.	2. Presenta al actor las siguientes opciones: a. Configurar Proxy. b. Configurar Restricciones. c. Configurar Descarga de Archivos.
3. Selecciona la opción Configurar Restricciones.	4. Muestra el nombre de todas las restricciones y su estado actual (habilitada o deshabilitada). Además presenta las opciones de habilitar o deshabilitar cualquiera de las restricciones.
5. Selecciona las restricciones que desea habilitar o deshabilitar.	6. Realiza la operación seleccionada por el actor y despliega el resultado de la operación para cada una de las restricciones seleccionadas.

Sección: Deshabilitar Proxy

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el actor selecciona la opción deshabilitar Proxy.	2. Deshabilita el proxy configurado y despliega el resultado de la operación.

Caso de uso: CONFIGURAR RESTRICCIONES DE INTERNET

Actor: administrador y usuario.

Propósito: permitir habilitar o deshabilitar las siguientes restricciones: No mostrar Tab de Seguridad, No mostrar Tab de Contenido, No mostrar Tab de Conexiones, No mostrar Tab de Programas, No mostrar Tab de Privacidad, No mostrar Tab de Opciones avanzadas, No modificar Opciones avanzadas, No modificar Configuración Automática, No modificar Configuración LAN, No permitir Asistente para la conexión y No modificar Servidor proxy.

Curso normal de los eventos

Cursos alternos:

- ❖ Línea 5: El actor no selecciona las restricciones que desea habilitar o deshabilitar. El sistema le informa al actor sobre esto.

Caso de uso: CONFIGURAR DESCARGA DE ARCHIVOS

Actor: administrador y usuario.

Propósito: permitir habilitar o deshabilitar la descarga de archivos o restringir al usuario de descargar software de sitios específicos.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Internet.	2. Presenta al actor las siguientes opciones: d. Configurar Proxy. e. Configurar Restricciones. f. Configurar Descarga de Archivos.
3. Selecciona la opción Configurar Descarga de Archivos.	4. Muestra los sitios de los que no se puede descargar software y el estado actual (habilitado o deshabilitado) de la descarga de archivos. Además presenta las siguientes opciones:

	<ol style="list-style-type: none"> a. Adicionar sitios, véase sección Adicionar sitios. b. Eliminar sitios, véase sección Eliminar sitios. c. Habilitar descarga de archivos, véase sección Habilitar/Deshabilitar descarga de archivos. d. Deshabilitar descarga de archivos, véase sección Habilitar/Deshabilitar descarga de archivos.
--	---

Sección: Adicionar sitios

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el actor introduce los nombres de los sitios de los que no se puede descargar software y selecciona la opción Adicionar sitios.	2. Restringe los sitios especificados y despliega el resultado de la operación para cada sitio.

Cursos alternos:

- ❖ Línea 1: El actor no introduce ningún sitio. El sistema le informa al actor sobre esto.

Sección: Eliminar sitios

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona los sitios que desea eliminar de la lista de sitios restringidos y la opción Eliminar sitios.	2. Elimina los sitios y despliega el resultado de la operación para cada uno de los sitios seleccionados.

Cursos alternos:

- ❖ Línea 1: El actor no selecciona los sitios que desea eliminar. El sistema le informa al actor sobre esto.

Sección: Habilitar/Deshabilitar descarga de archivos

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Habilitar o Deshabilitar descarga de archivos.	2. Realiza la operación seleccionada por el actor y despliega el resultado de la operación.

Caso de uso: CONFIGURAR SEGURIDAD DE RED

Actor: administrador y usuario.

Propósito: permitir habilitar o deshabilitar el protocolo SSDP, el fortalecimiento TCP/IP y la protección contra ataques de inundación.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Red.	2. Presenta las siguientes opciones: <ol style="list-style-type: none"> a. Configurar Seguridad de Red. b. Configurar Restricciones de Red. c. Configurar Entorno de Red.
3. Selecciona la opción Configurar Seguridad de Red.	4. Muestra el estado actual (habilitado o deshabilitado) del protocolo SSDP, el fortalecimiento TCP/IP y la protección contra ataques de inundación. Además le presenta las opciones de habilitar o deshabilitar cualquiera de ellas.
5. Selecciona la opción Habilitar o Deshabilitar SSDP, Fortalecimiento TCP/IP o Protección	6. Realiza la operación seleccionada por el actor y despliega el resultado de la operación.

contra ataques de inundación.	
-------------------------------	--

Caso de uso: CONFIGURAR RESTRICCIONES DE RED

Actor: administrador y usuario.

Propósito: permitir habilitar o deshabilitar las siguientes restricciones: No permitir Agregar componentes de red, No permitir Configuración avanzada, No permitir Asistente para nueva conexión, No permitir Habilitar y Deshabilitar la Conexión de área local, No permitir cambiar el nombre de la Conexión de área local, No permitir ver las propiedades de la Conexión de área local, No permitir cambiar las propiedades de la Conexión de área local y No permitir Configuración avanzada del Protocolo de Internet (TCP/IP).

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Red.	2. Presenta las siguientes opciones: a. Configurar Seguridad de Red. b. Configurar Restricciones de Red. c. Configurar Entorno de Red.
3. Selecciona la opción Configurar Restricciones de Red.	4. Muestra el nombre de todas las restricciones y su estado actual (habilitada o deshabilitada). Además presenta las opciones de habilitar o deshabilitar cualquiera de ellas.
5. Selecciona las restricciones que desea habilitar o deshabilitar.	6. Realiza la operación seleccionada por el actor y despliega el resultado de la operación para cada una de las restricciones seleccionadas.

Cursos alternos:

❖ Línea 5: El actor no selecciona las restricciones que desea habilitar o deshabilitar. El sistema le informa al actor sobre esto.

Caso de uso: CONFIGURAR ENTORNO DE RED

Actor: administrador y usuario.

Propósito: permitir ocultar o mostrar el equipo remoto de la lista de equipos próximos, la opción Equipos Próximos o la opción Toda la red.

Curso normal de los eventos

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un actor selecciona la opción Red.	2. Presenta las siguientes opciones: a. Configurar Seguridad de Red. b. Configurar Restricciones de Red. c. Configurar Entorno de Red.
3. Selecciona la opción Configurar Entorno de Red.	4. Muestra el estado actual (visible u oculto) del equipo , de toda la red y de los equipos próximos. Además presenta las opciones mostrar u ocultar para cada uno de ellos.
5. Selecciona la opción Mostrar u Ocultar para el equipo, toda la red o equipos próximos.	6. Realiza la operación seleccionada por el actor y despliega el resultado de la operación.

2.2 PAQUETES

Acceso: Validar Datos.

Hubs-switches: Obtener direcciones MAC, Consultar direcciones, Adicionar dirección IP, Modificar direcciones, Eliminar direcciones, Insertar rango, Consultar hubs-switches, Adicionar hubs-switches, Modificar hubs-switches, Eliminar hubs-switches, Ver información SNMP, Ver equipos de usuario.

Equipo: Conectar a un equipo de usuario.

Hardware: Ver información hardware, Configurar protocolo Internet (TCP/IP), Configurar DNS, Configurar NetBIOS, Configurar filtrado TCP/IP.

Software: Ver información software, Apagar/Reiniciar equipo, Crear proceso, Terminar procesos, Cambiar estado de servicios, Buscar archivos, Ejecutar operaciones sobre archivos, Eliminar archivos,

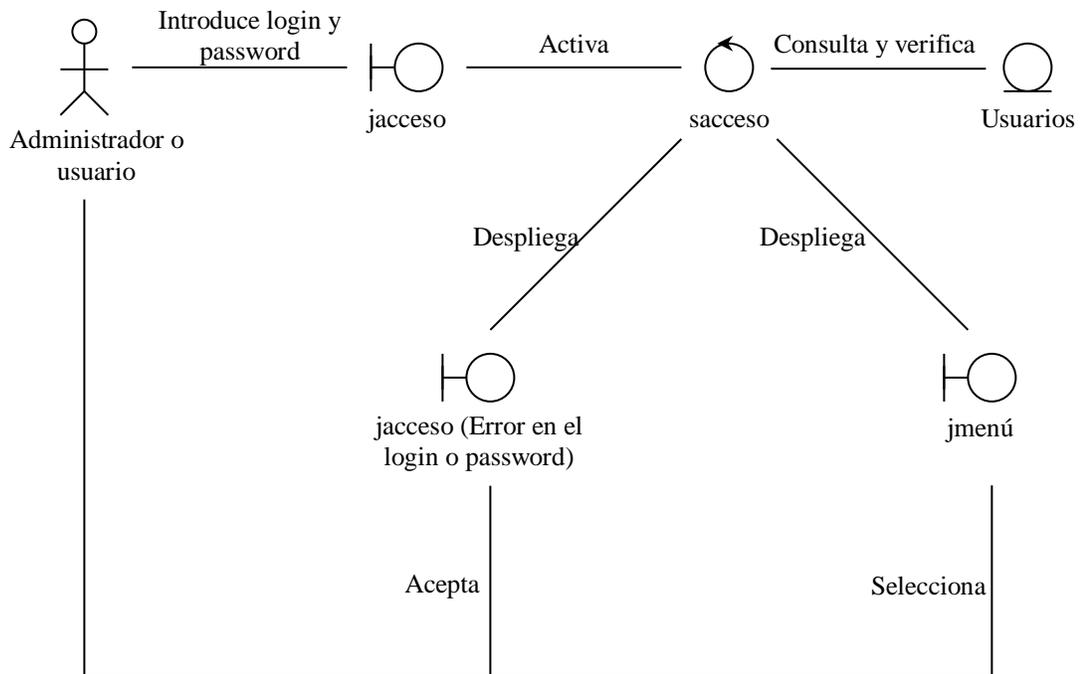
Determinar tamaño, Eliminar todos los archivos, Instalar producto software, Desinstalar producto software, Ver eventos NT, Enviar e-mail por cambio de dirección IP.

Registro: Configurar Sistema, Configurar aplicaciones, Configurar proxy, Configurar restricciones de Internet, Configurar descarga de archivos, Configurar seguridad de red, Configurar restricciones de red, Configurar entorno de red.

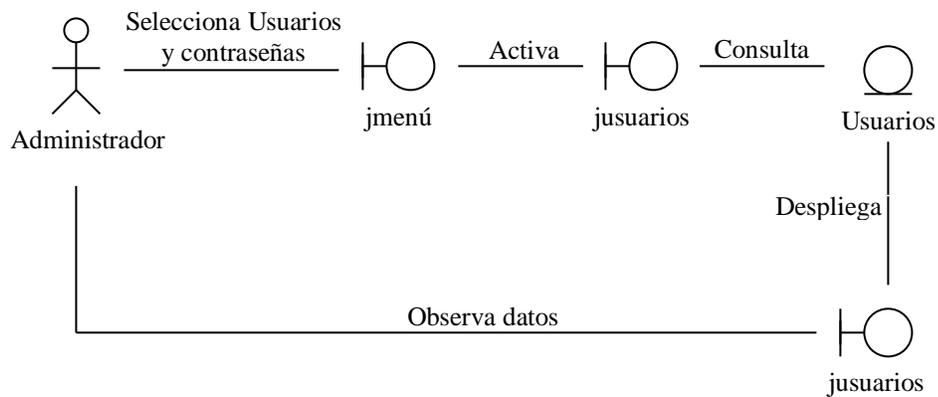
Usuarios: Consultar usuarios, Adicionar usuario, Modificar usuario, Eliminar usuarios.

2.3 DIAGRAMAS DE REALIZACIÓN

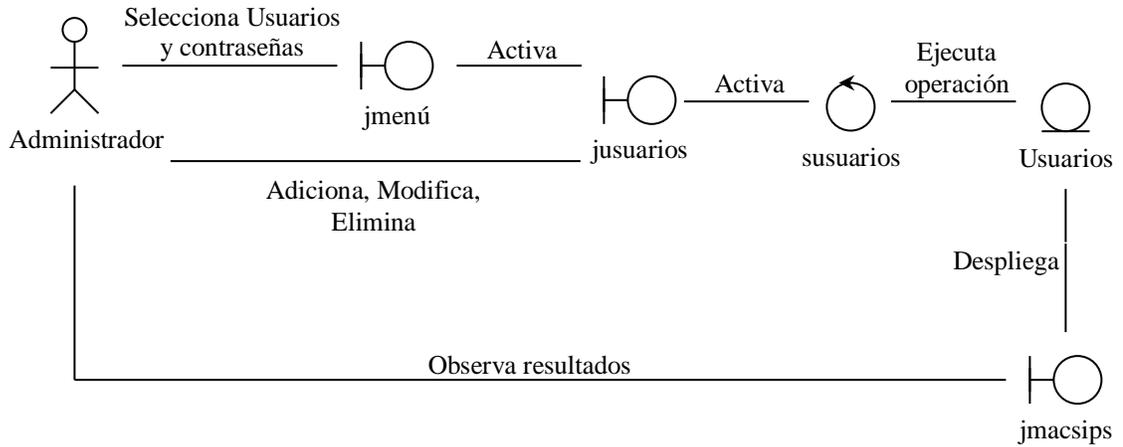
VALIDAR DATOS



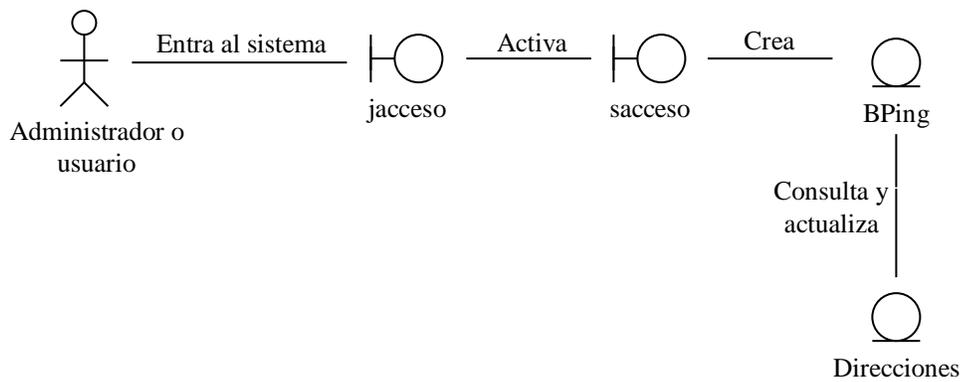
CONSULTAR USUARIOS



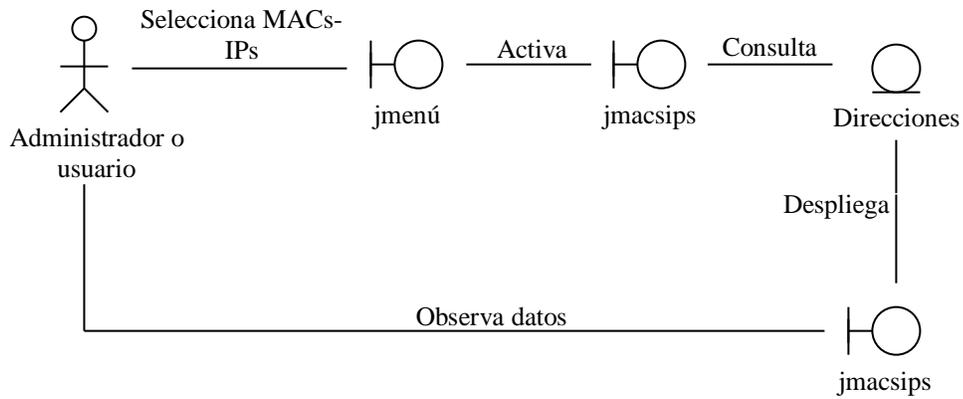
ADICIONAR, MODIFICAR Y ELIMINAR USUARIOS



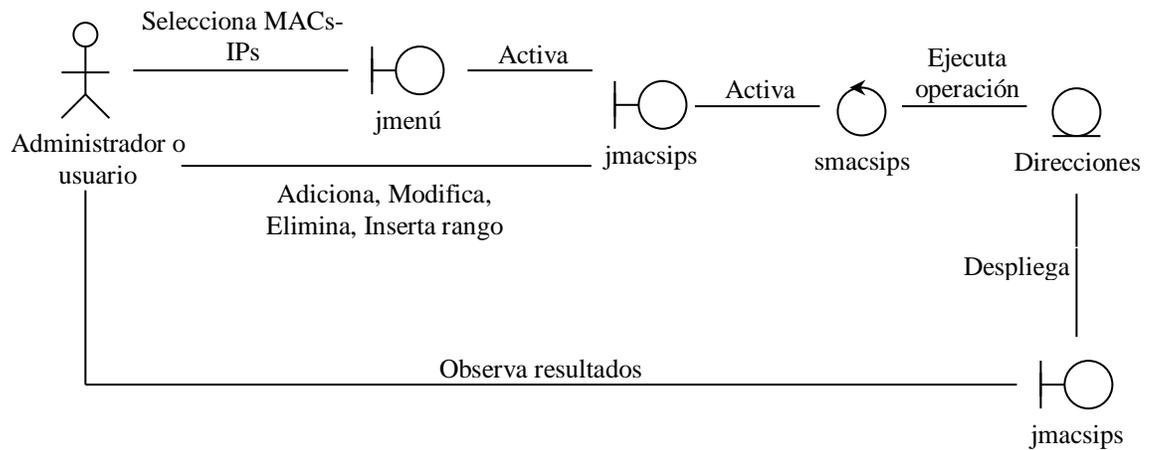
OBTENER DIRECCIONES MAC



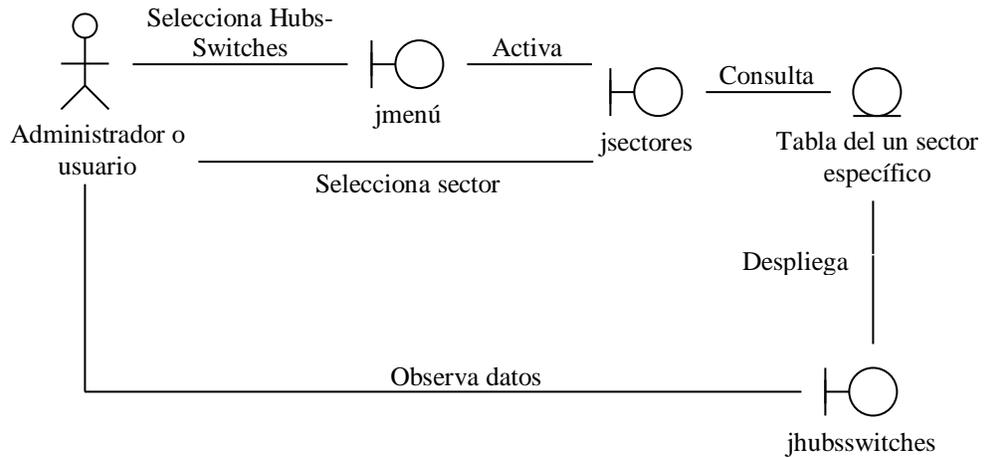
CONSULTAR DIRECCIONES



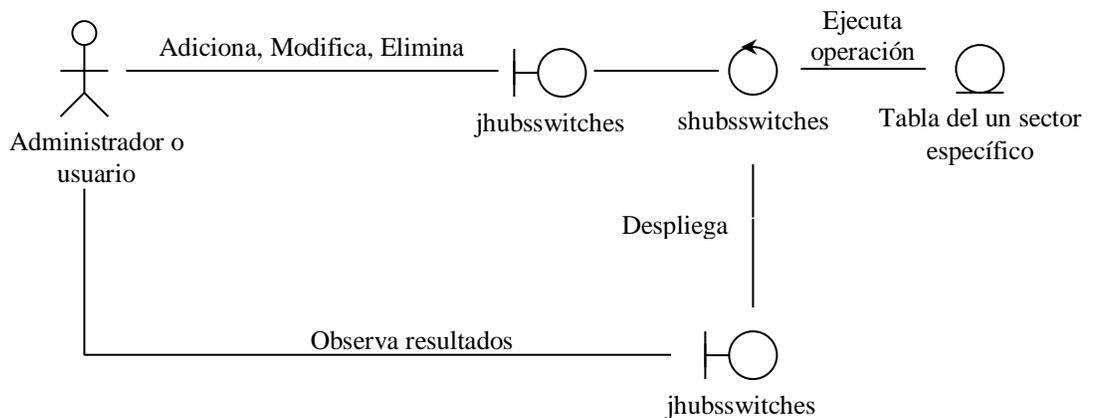
ADICIONAR DIRECCIÓN IP, MODIFICAR DIRECCIONES, ELIMINAR DIRECCIONES E INSERTAR RANGO



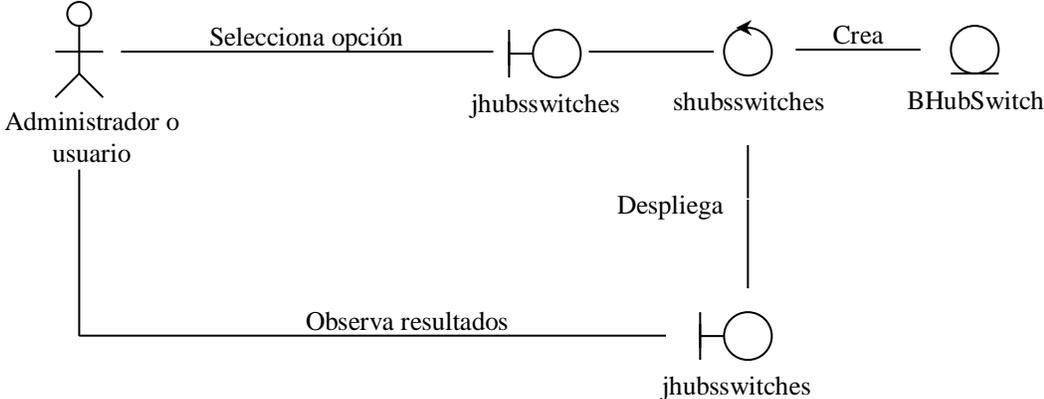
CONSULTAR HUBS-SWITCHES



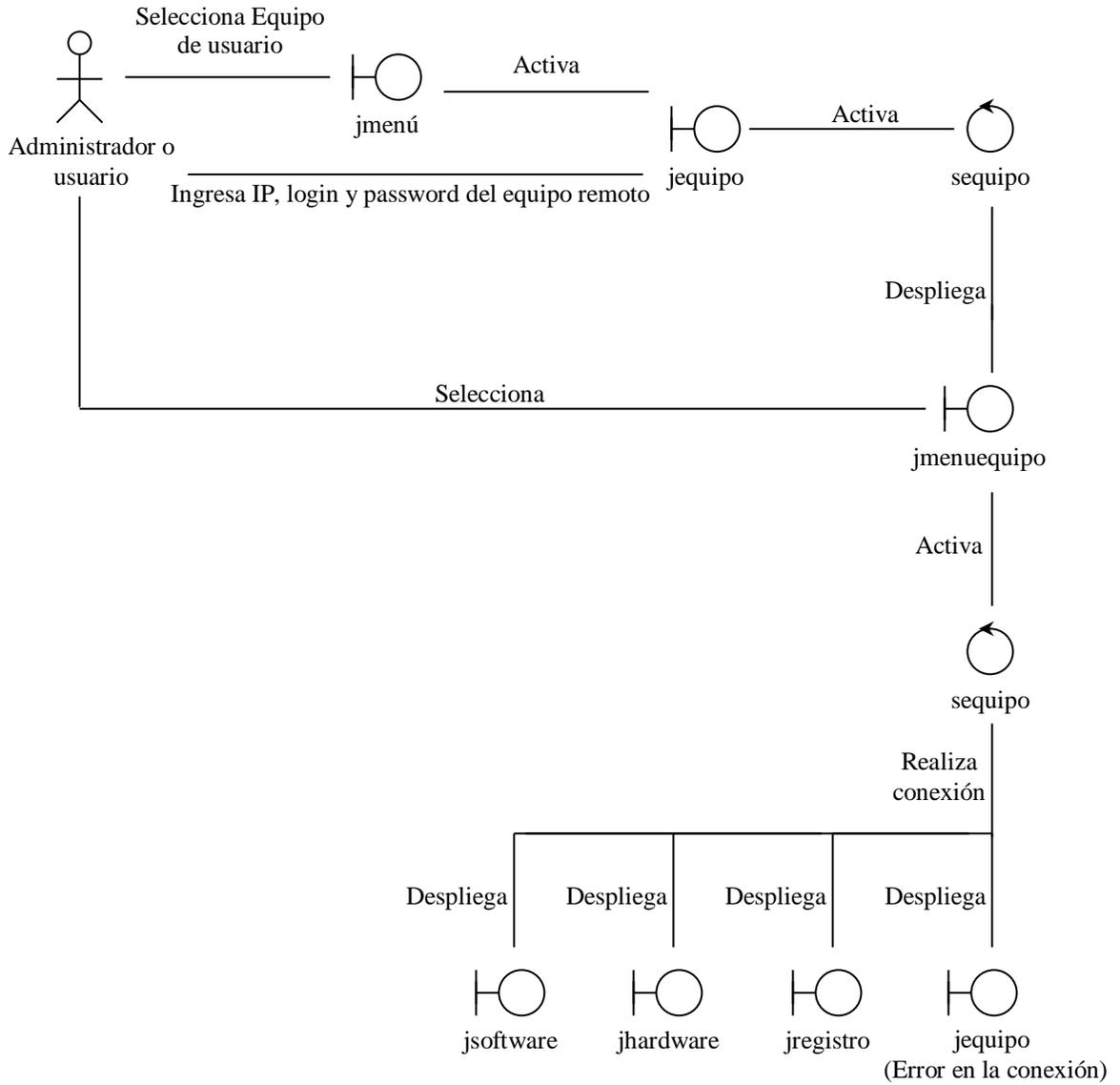
ADICIONAR, MODIFICAR Y ELIMINAR HUBS-SWITCHES



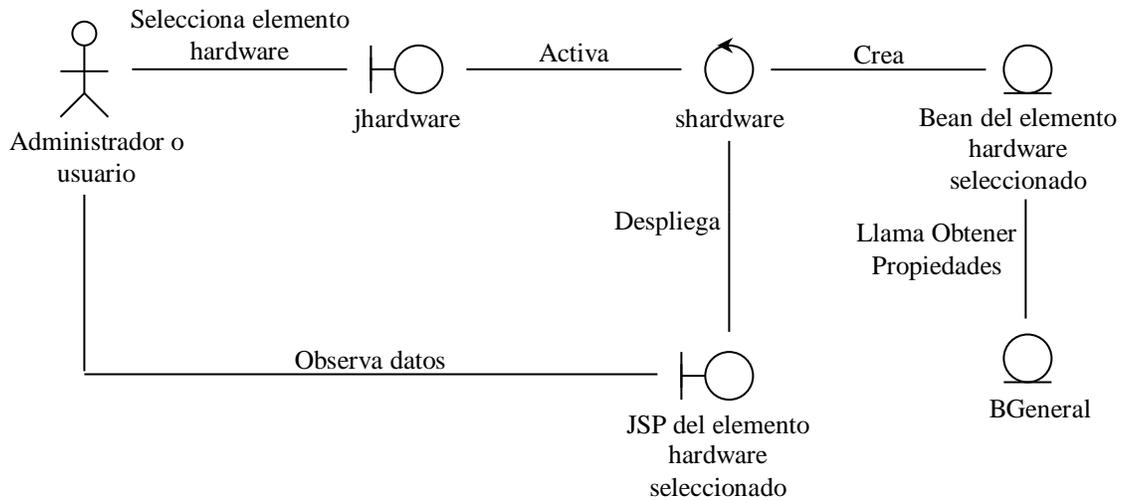
VER INFORMACIÓN SNMP/ VER EQUIPOS DE USUARIO



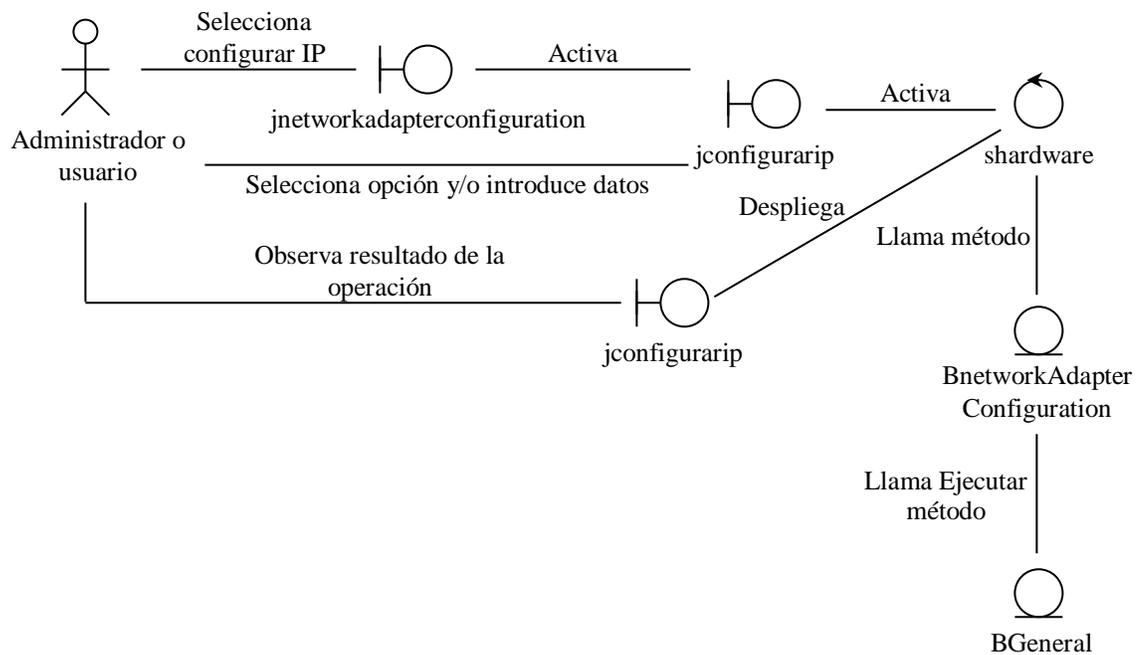
CONECTAR A UN EQUIPO DE USUARIO



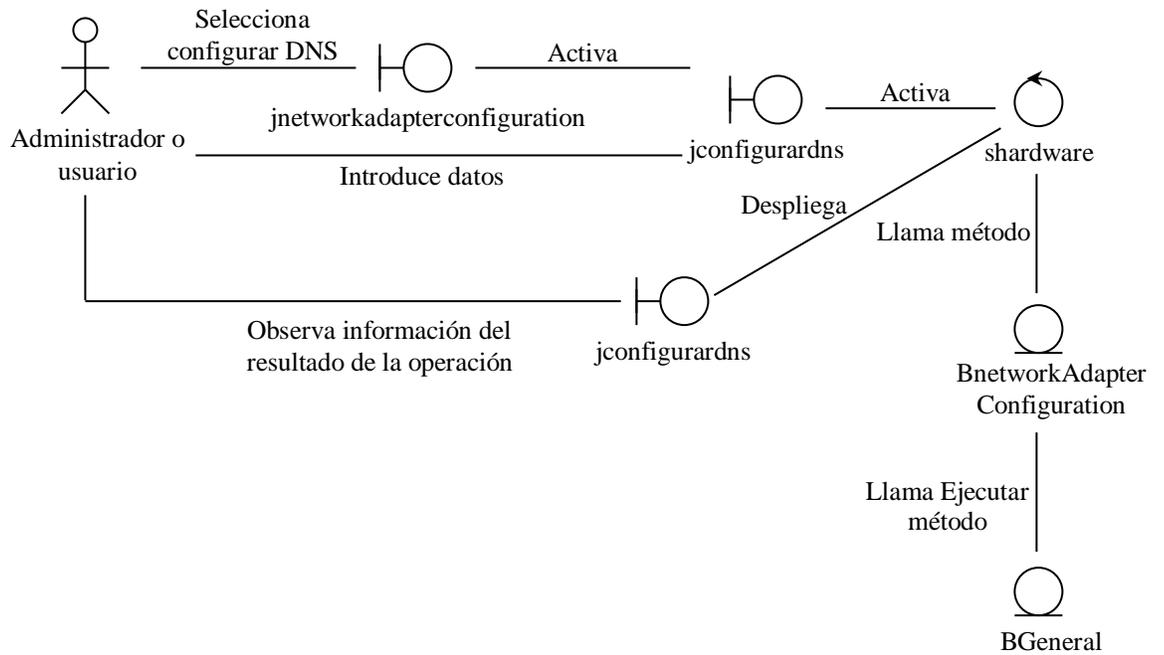
VER INFORMACIÓN HARDWARE



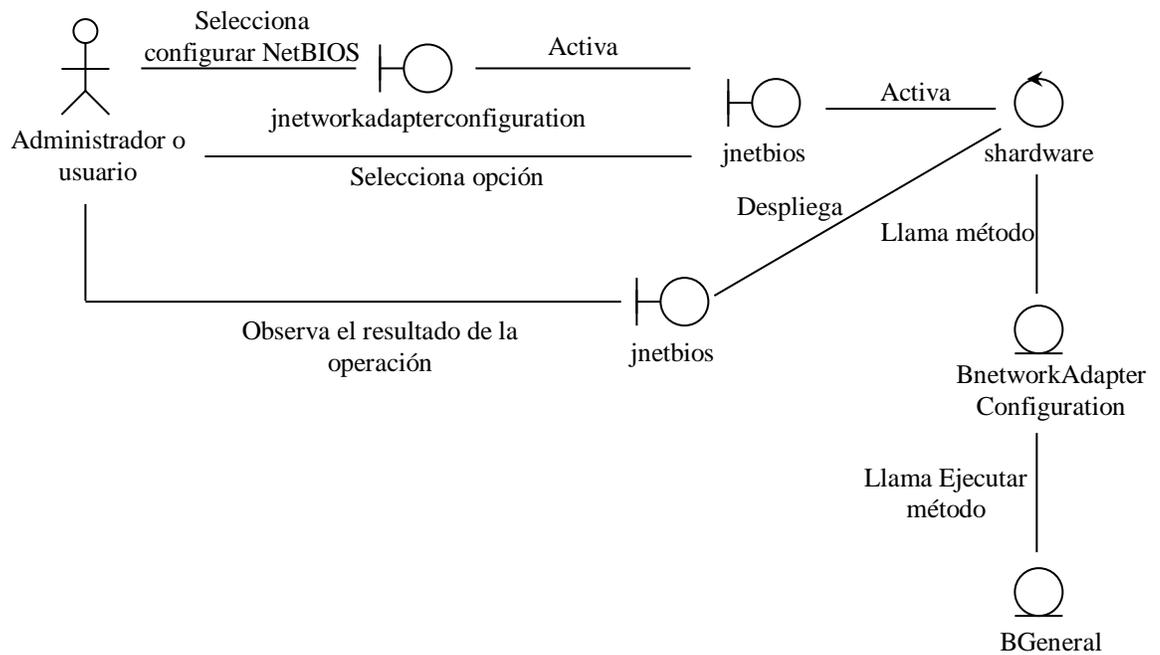
CONFIGURAR PROTOCOLO INTERNET (TCP/IP)



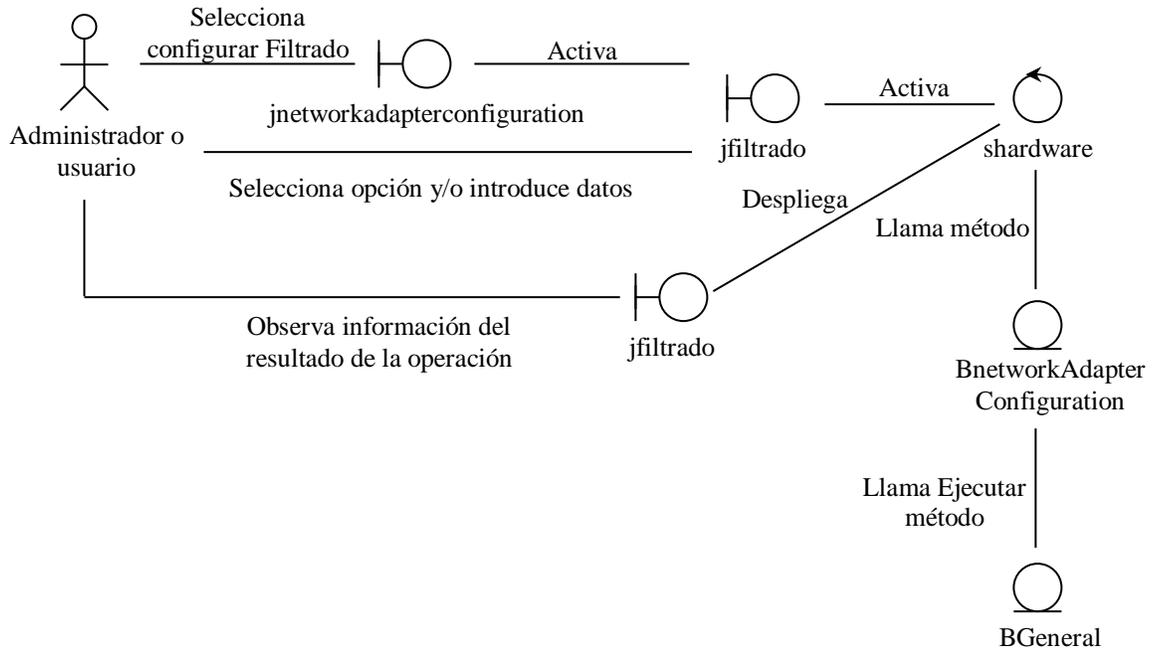
CONFIGURAR DNS



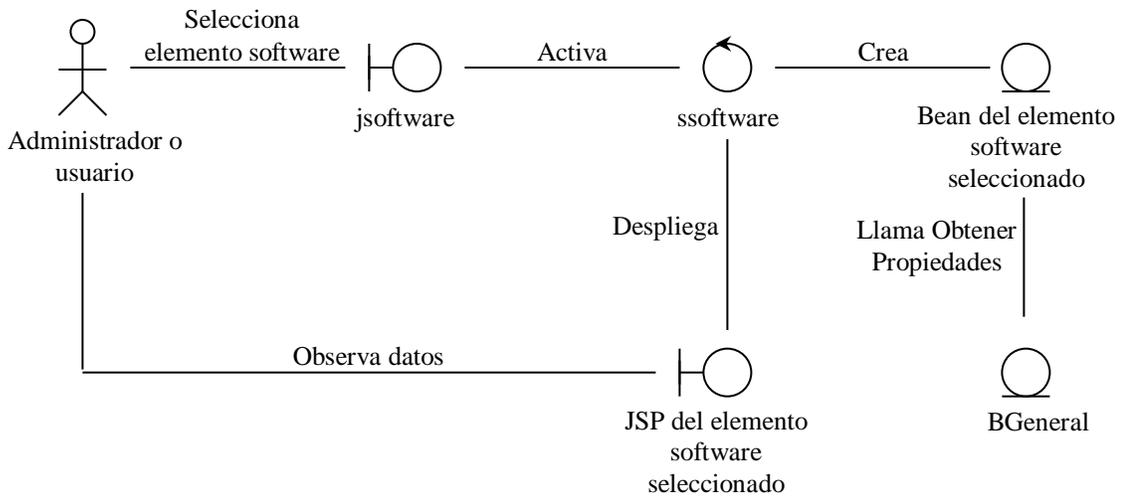
CONFIGURAR NetBIOS



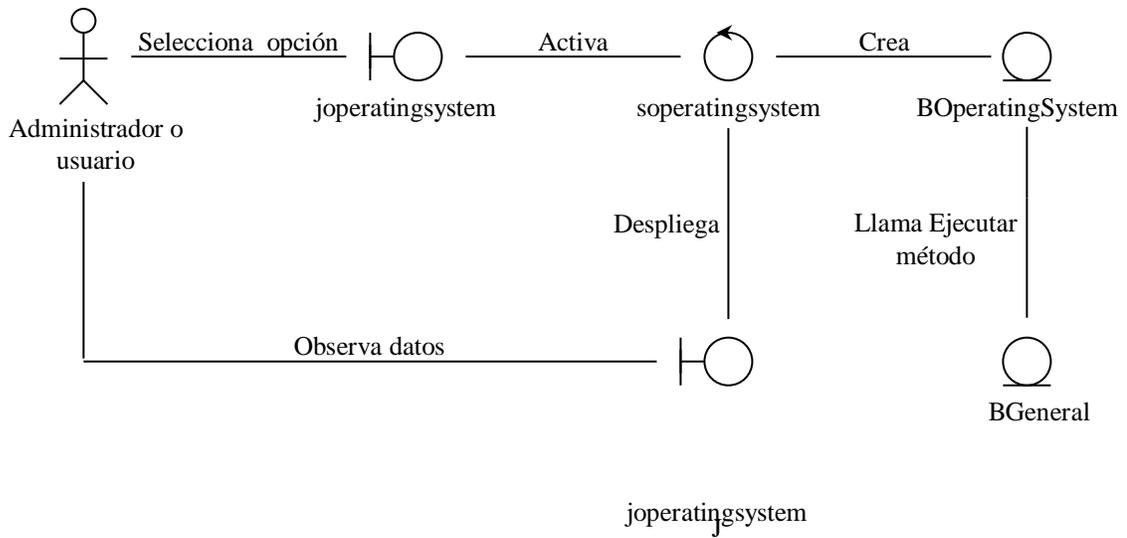
CONFIGURAR FILTRADO TCP/IP



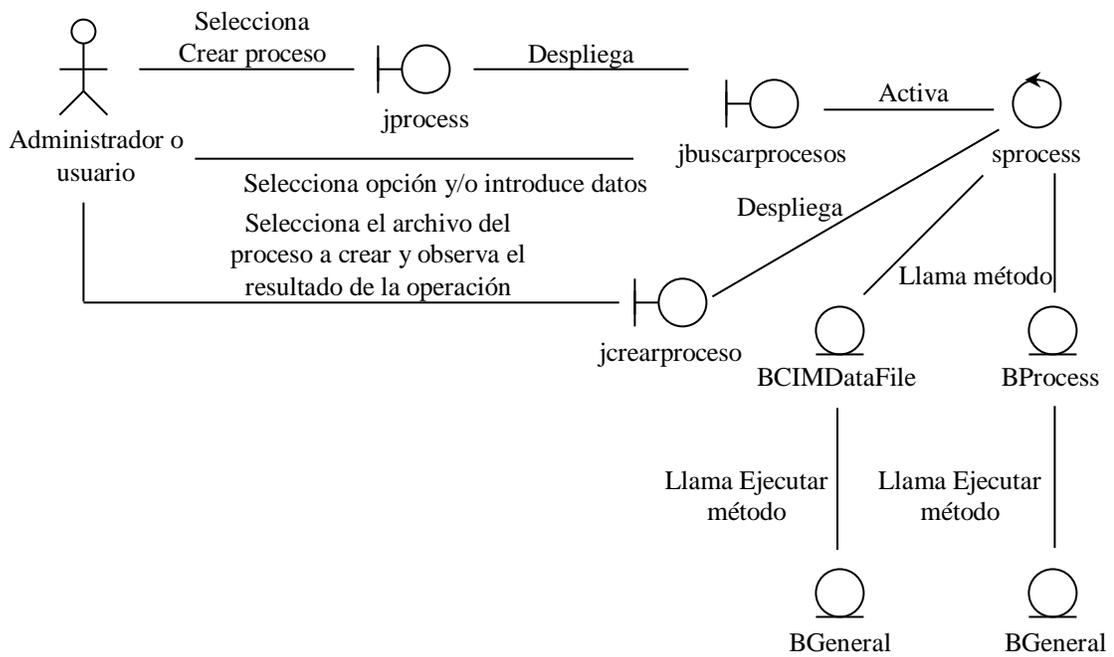
VER INFORMACIÓN SOFTWARE



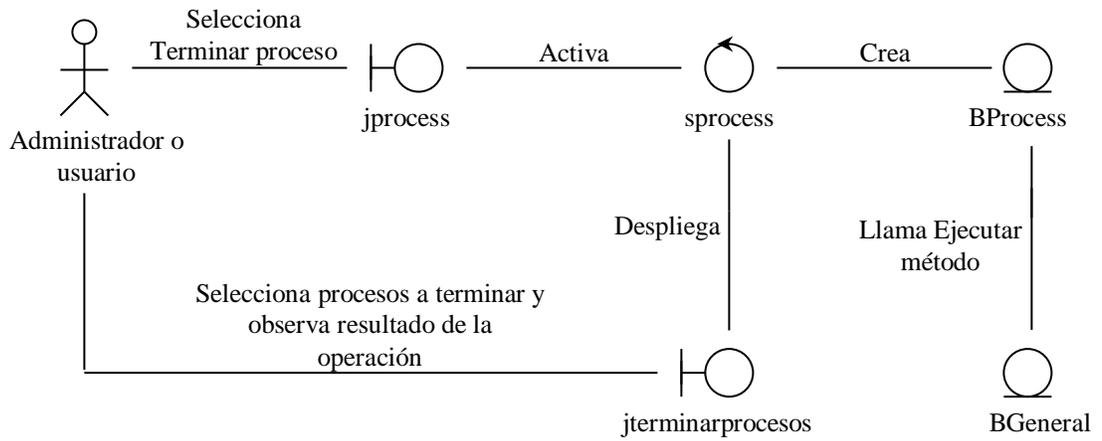
APAGAR/REINICIAR EQUIPO



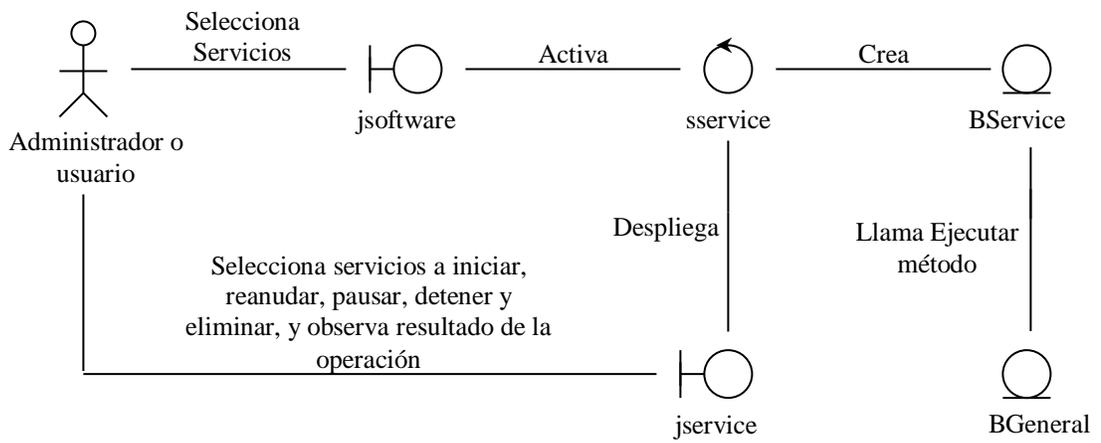
CREAR PROCESO



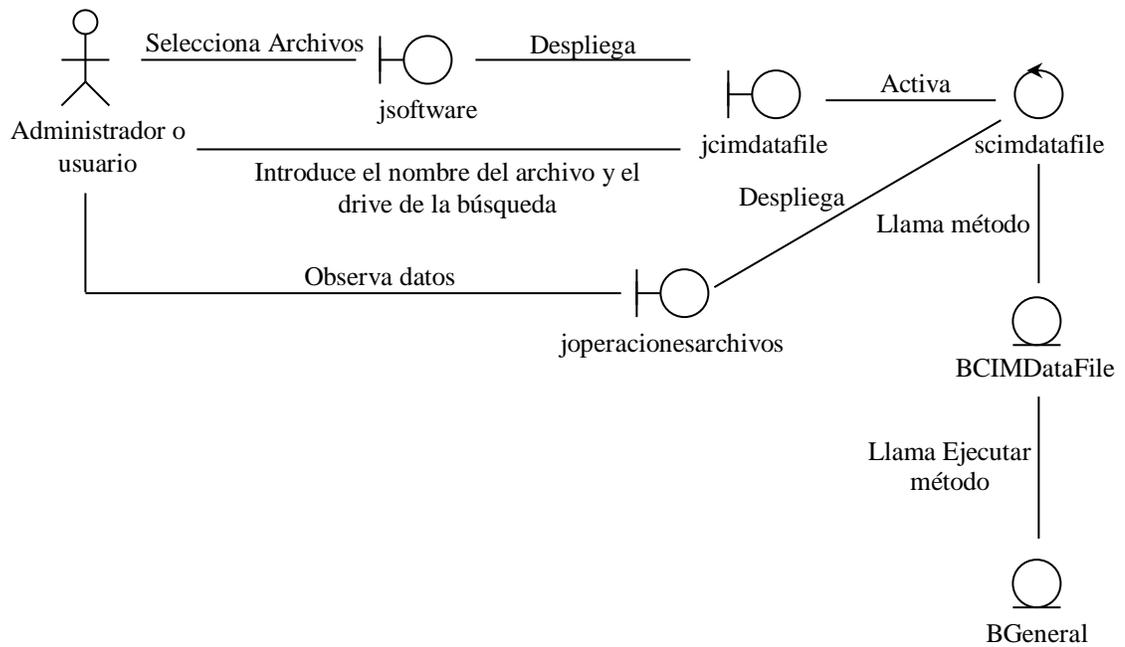
TERMINAR PROCESOS



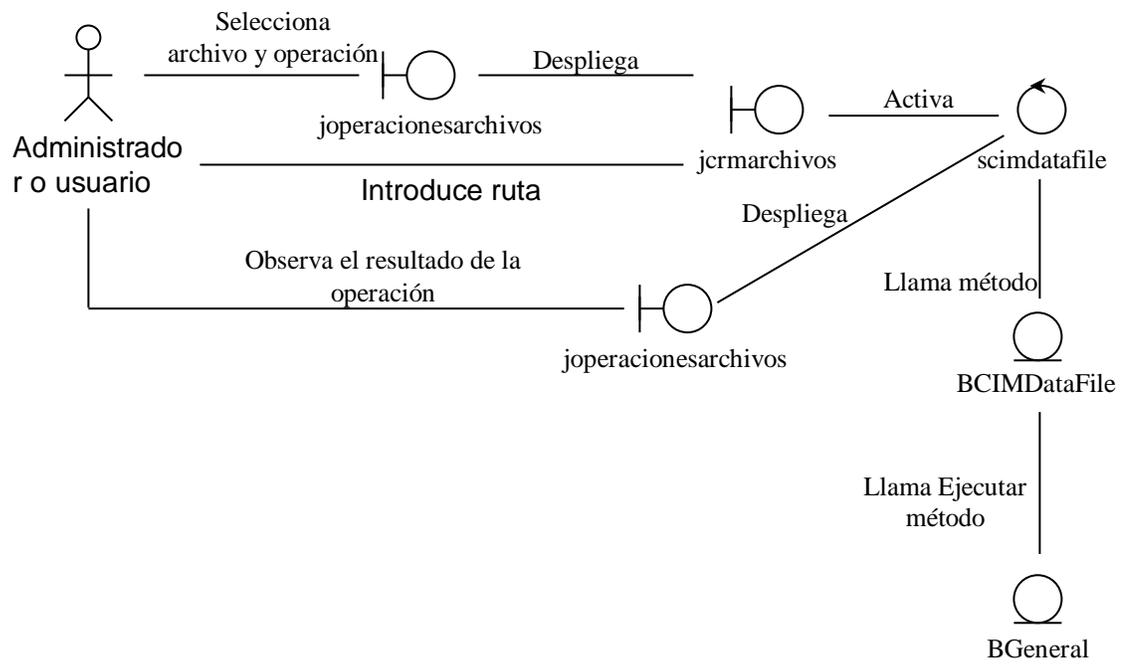
CAMBIAR ESTADO DE SERVICIO



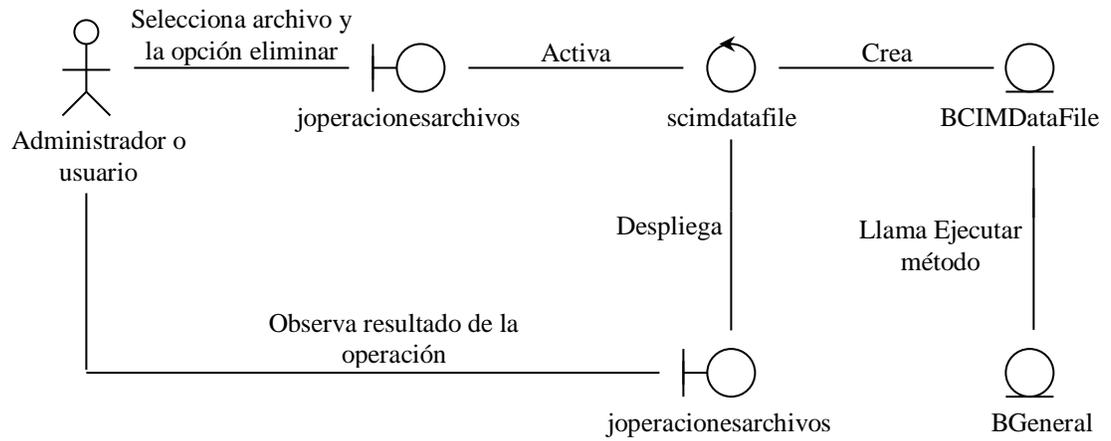
BUSCAR ARCHIVOS



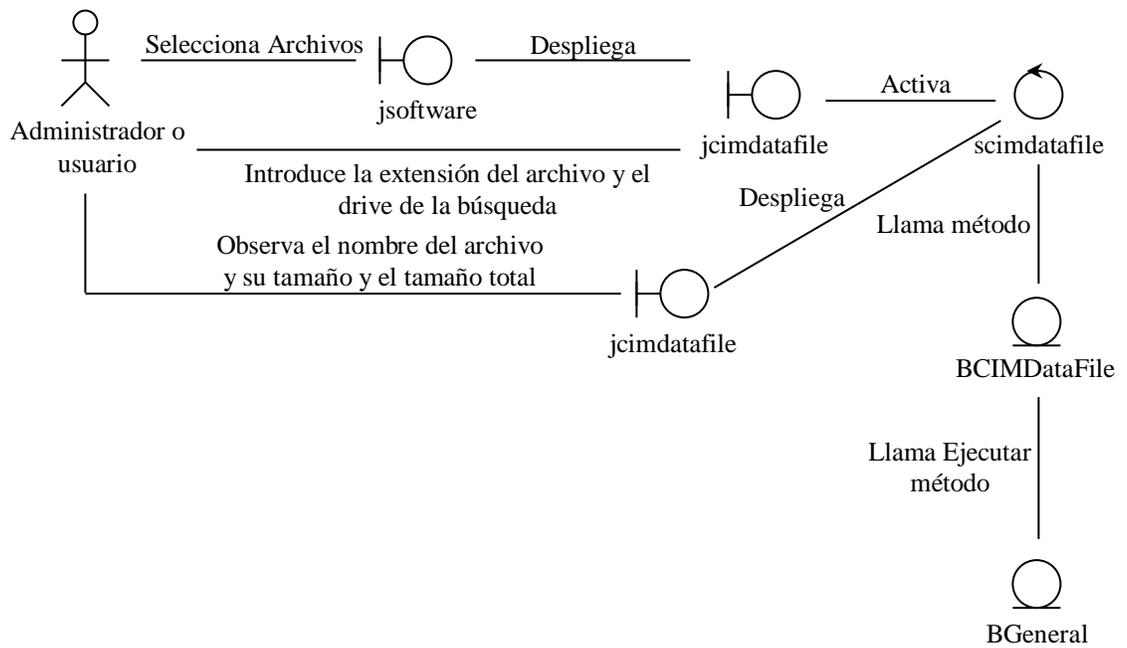
EJECUTAR OPERACIONES SOBRE ARCHIVOS



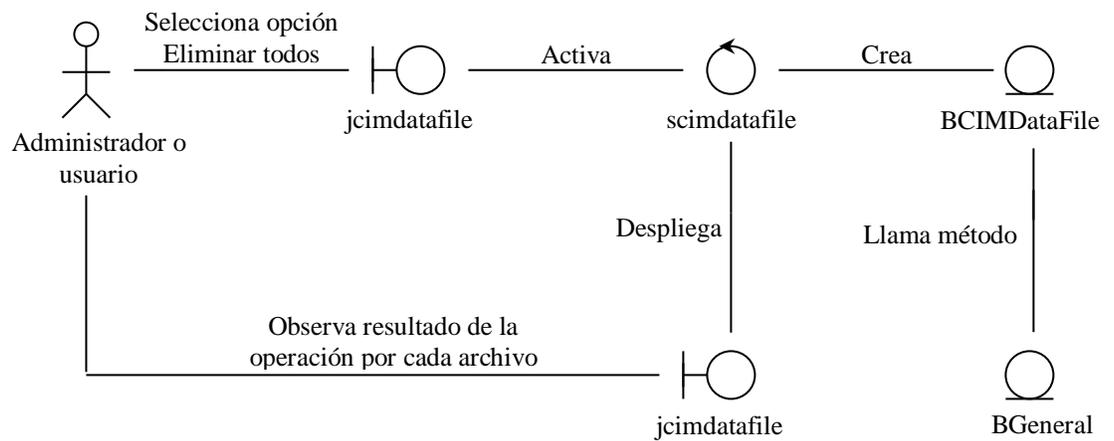
ELIMINAR ARCHIVOS



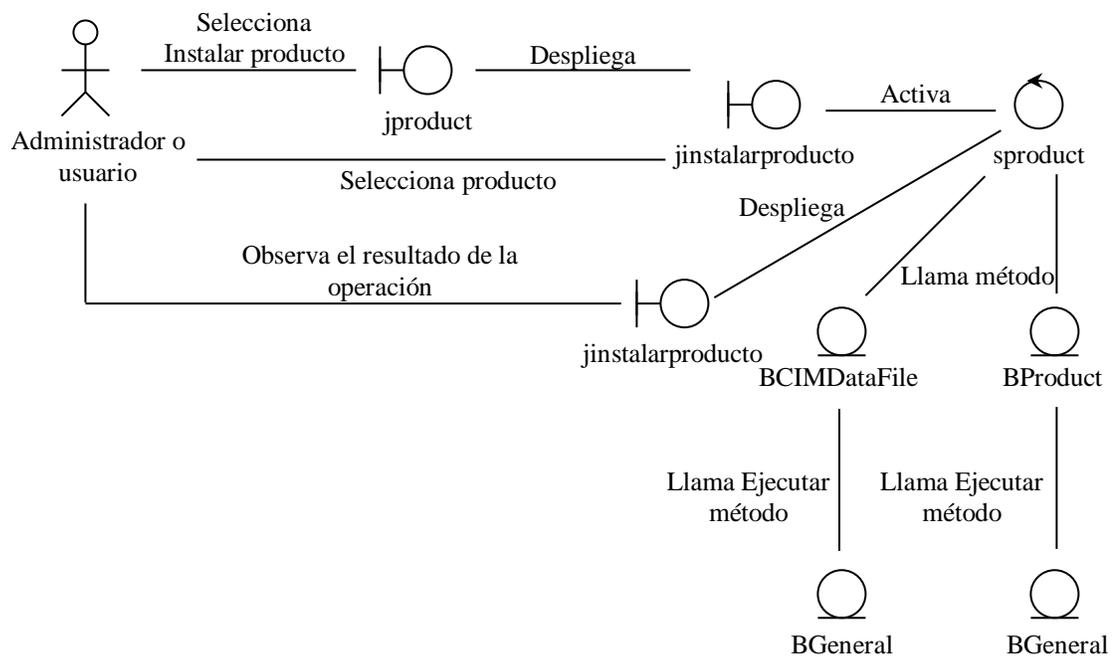
DETERMINAR TAMAÑO



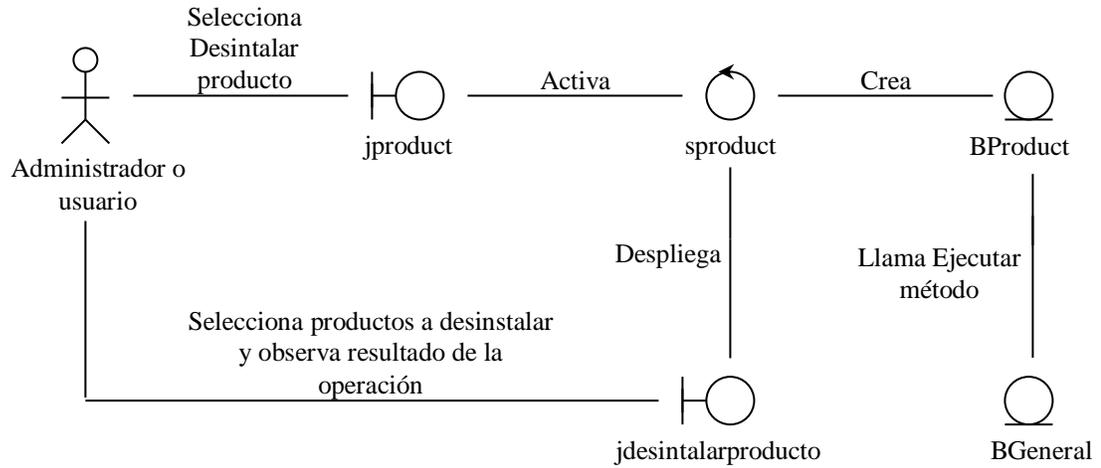
ELIMINAR TODOS LOS ARCHIVOS



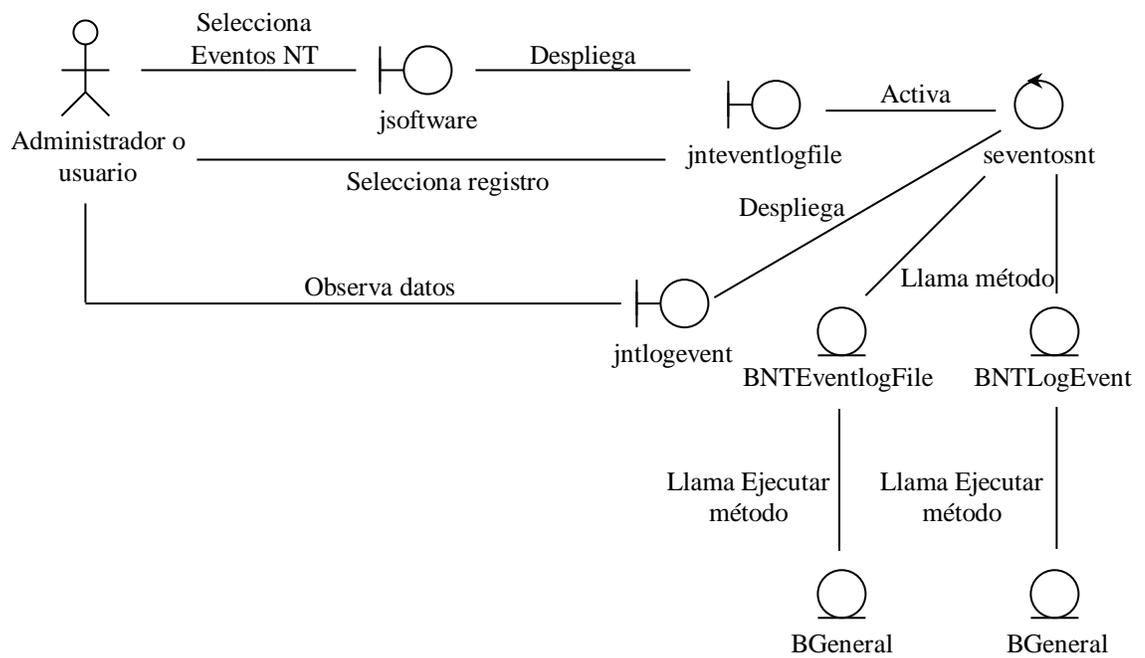
INSTALAR PRODUCTO SOFTWARE



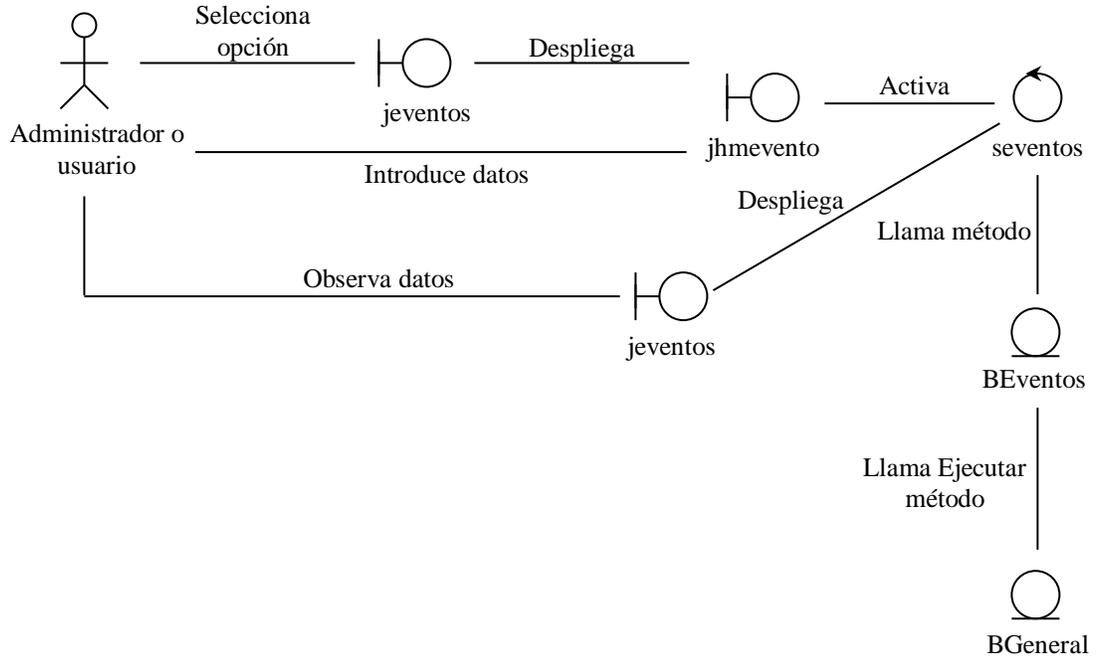
DESINSTALAR PRODUCTO SOFTWARE



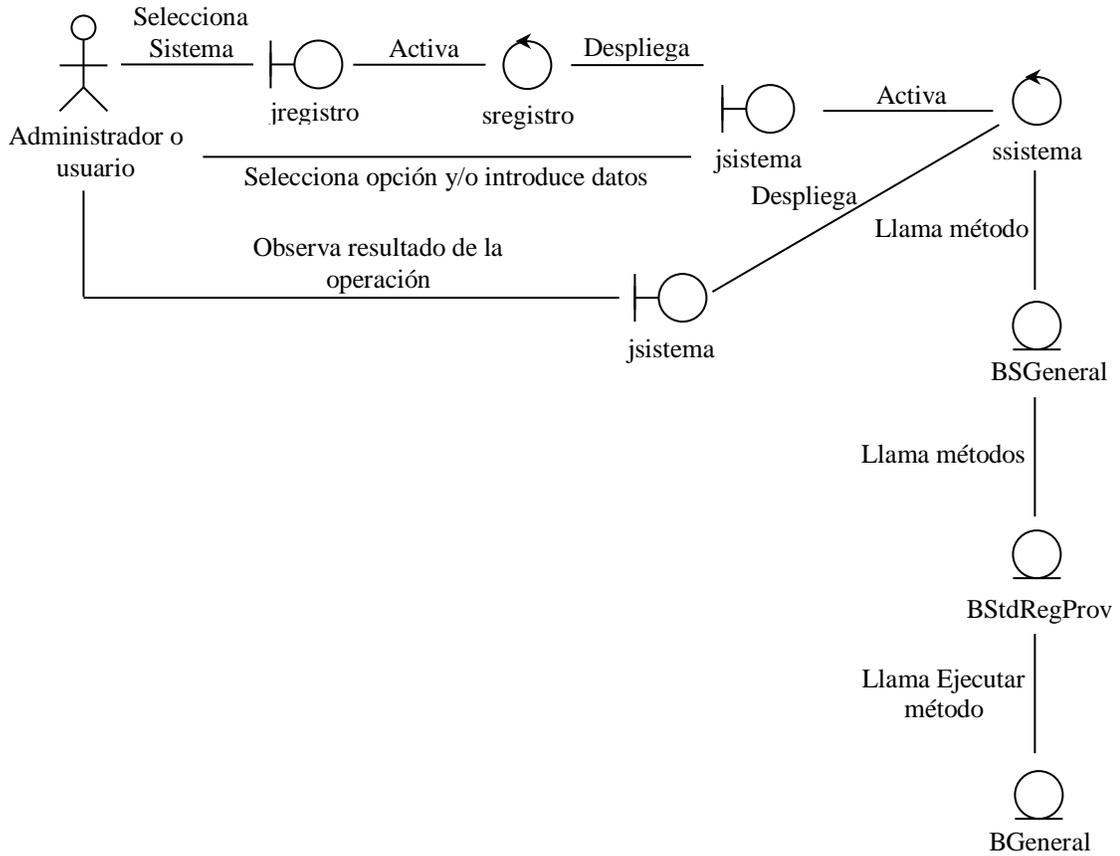
VER EVENTOS NT



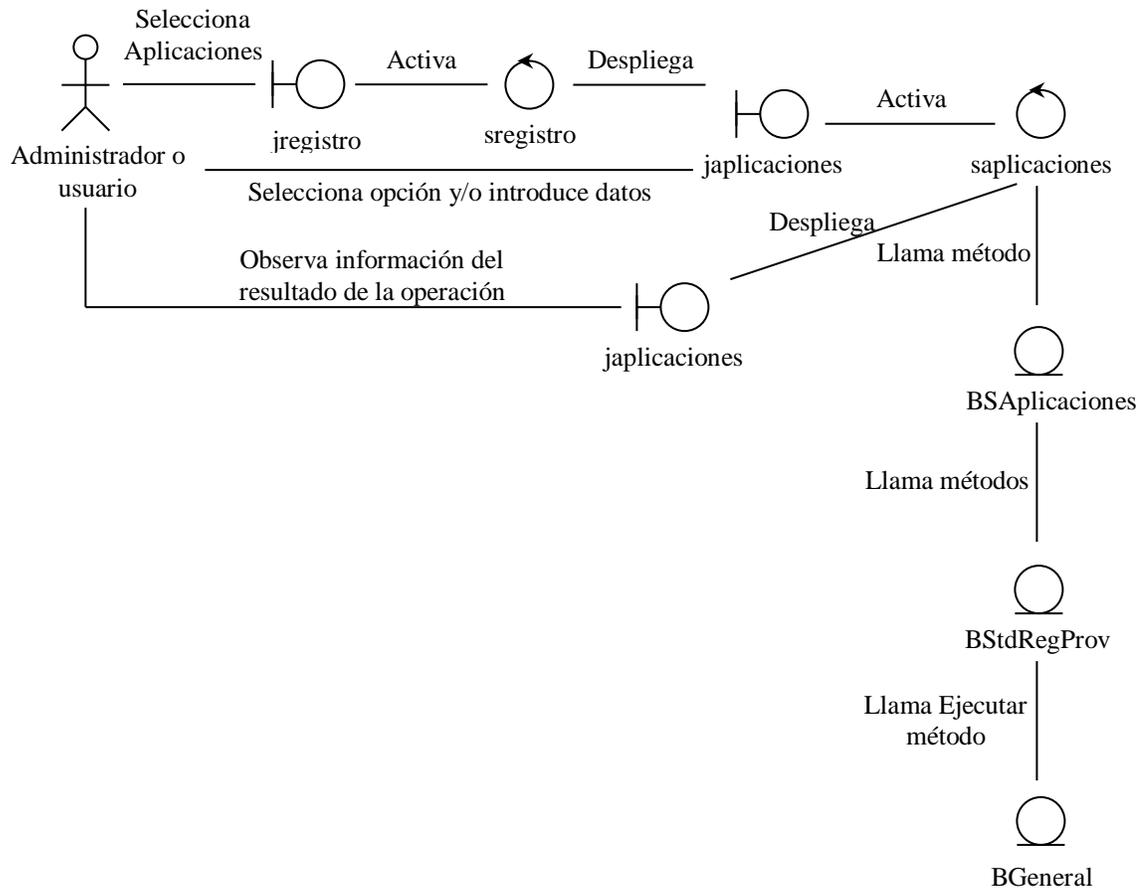
ENVIAR UN E-MAIL POR CAMBIO DE DIRECCIÓN IP



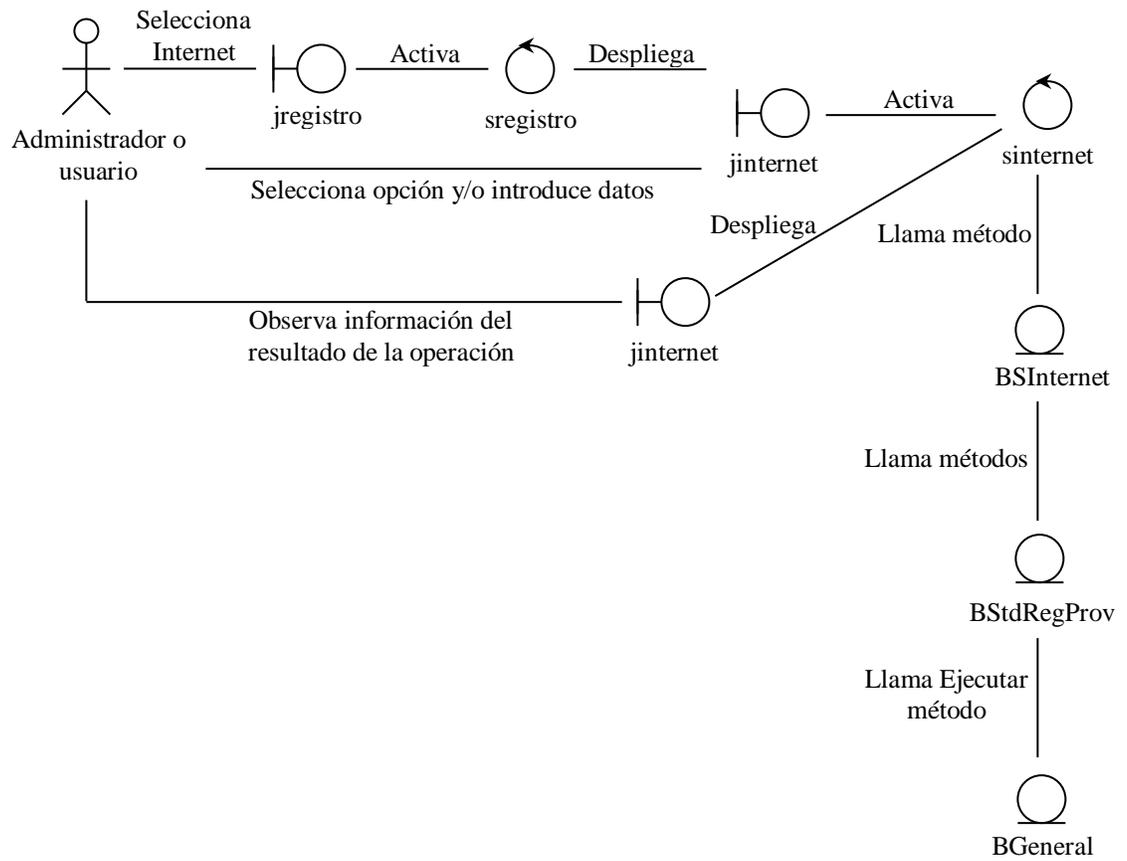
CONFIGURAR SISTEMA



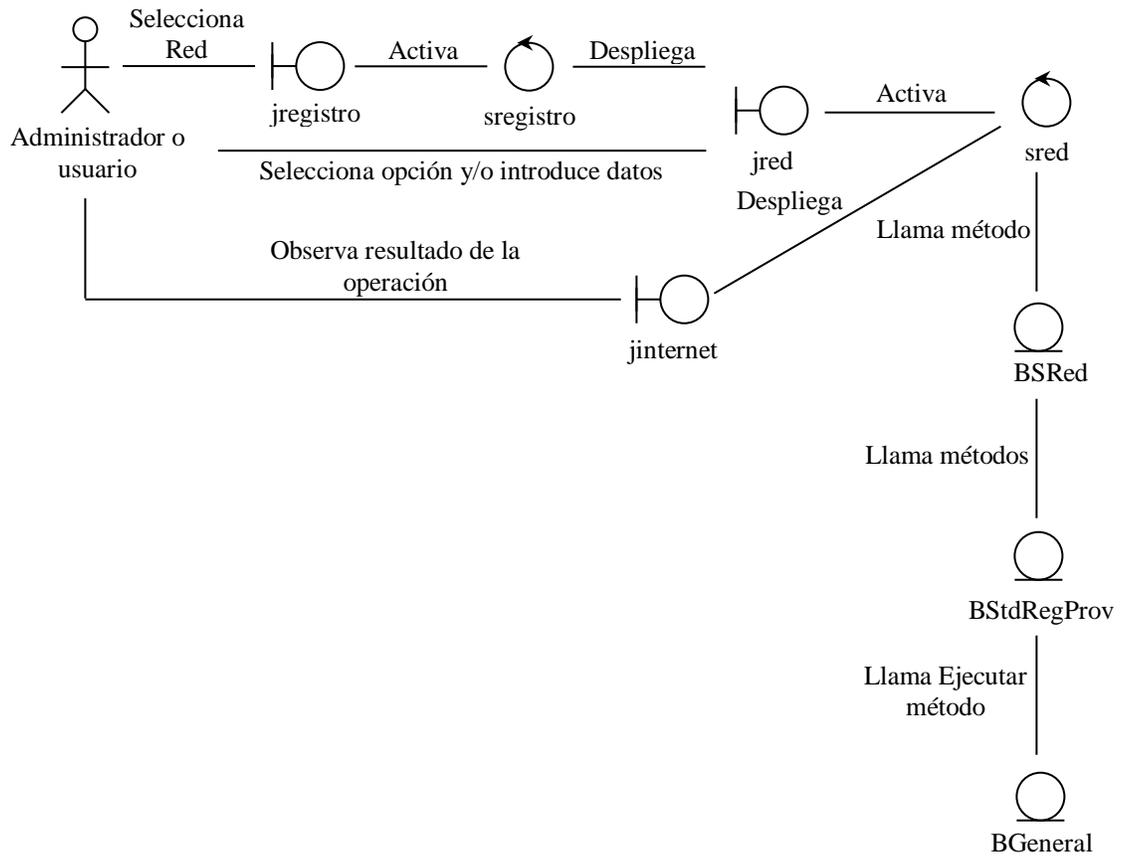
CONFIGURAR APLICACIONES



CONFIGURAR PROXY, CONFIGURAR RESTRICCIONES DE INTERNET, CONFIGURAR DESCARGA DE ARCHIVOS



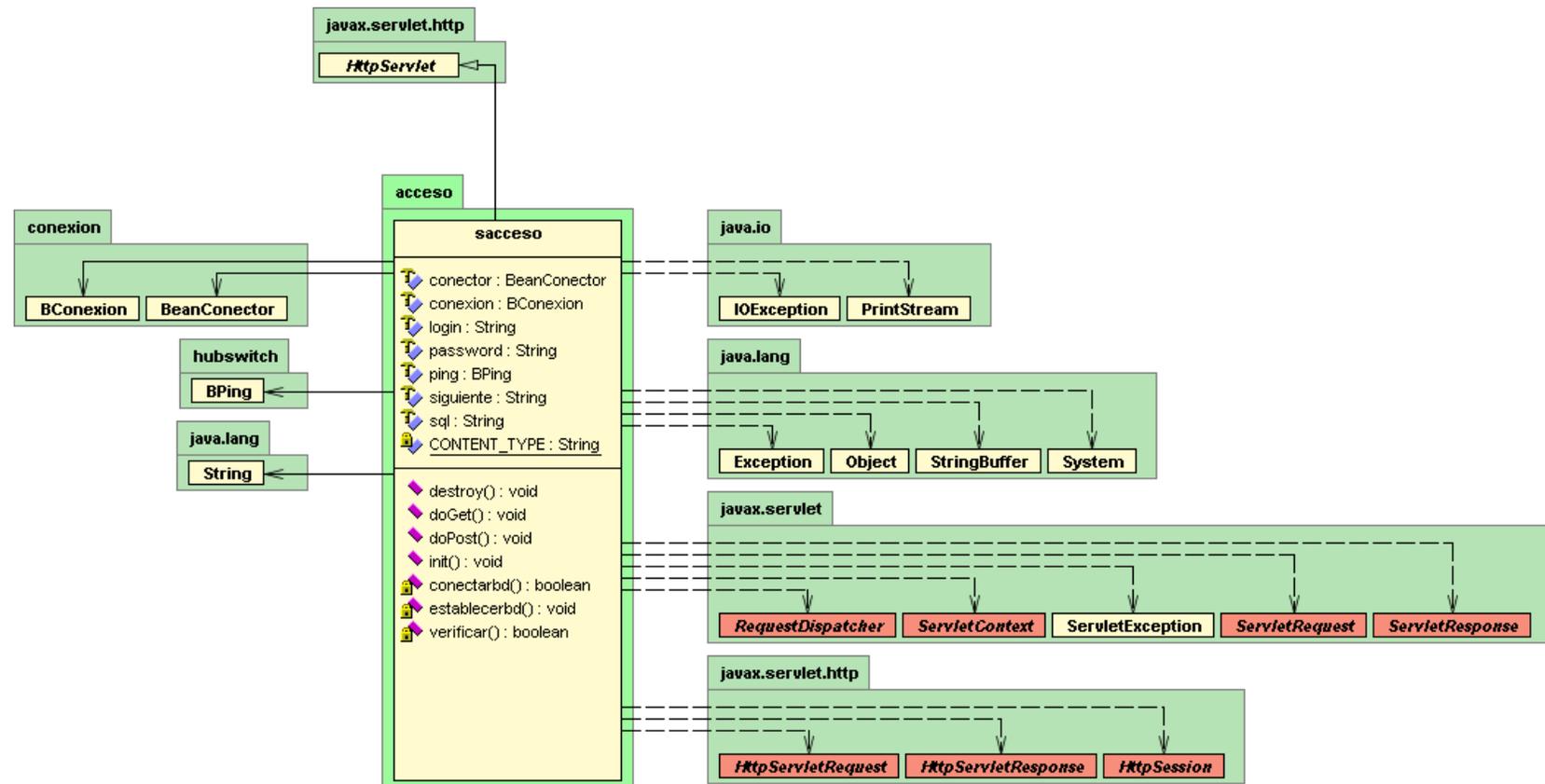
CONFIGURAR SEGURIDAD DE RED, CONFIGURAR RESTRICCIONES DE RED, CONFIGURAR ENTORNO DE RED



2.4 DIAGRAMAS DE CLASE

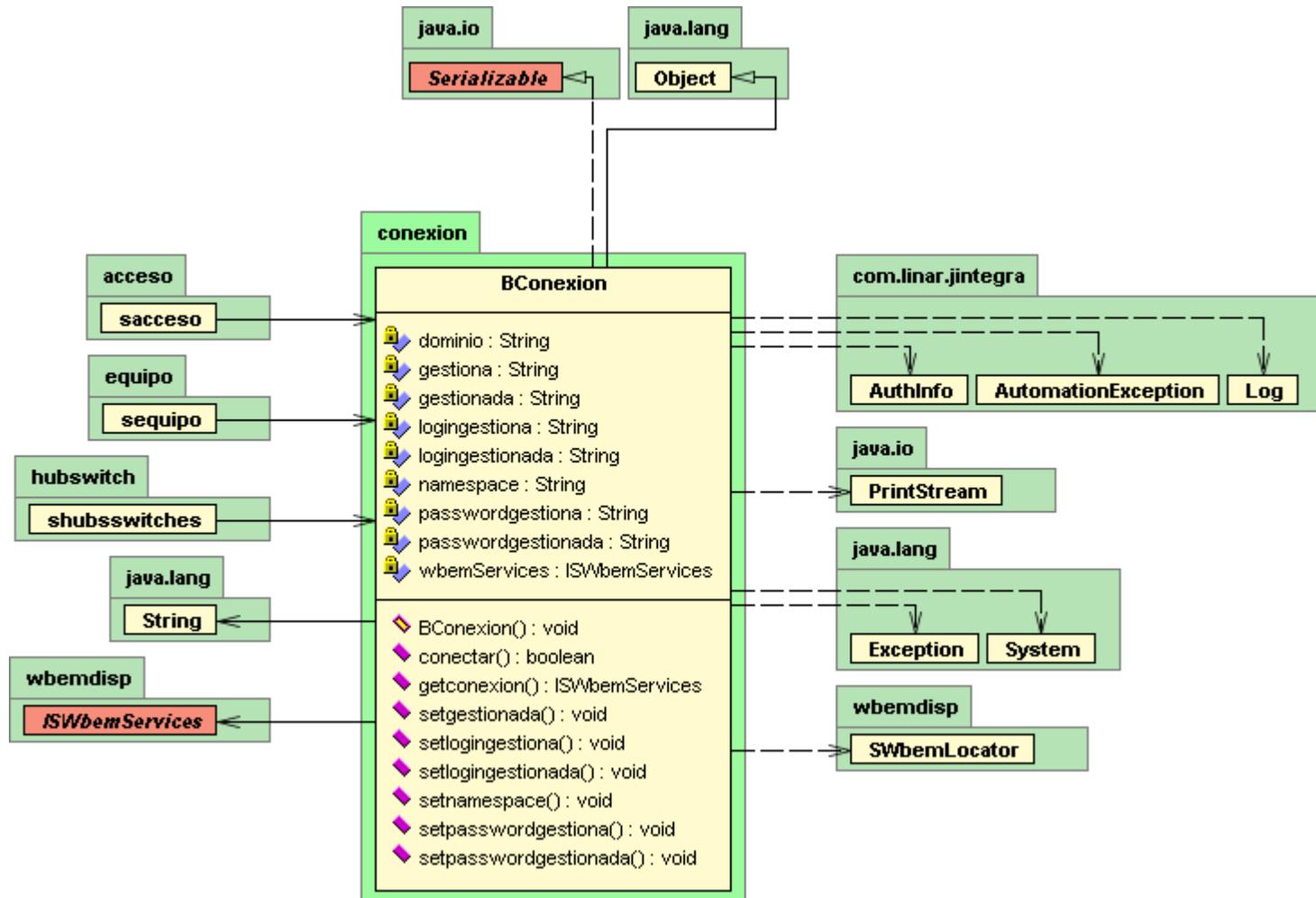
Paquete acceso

sacceso

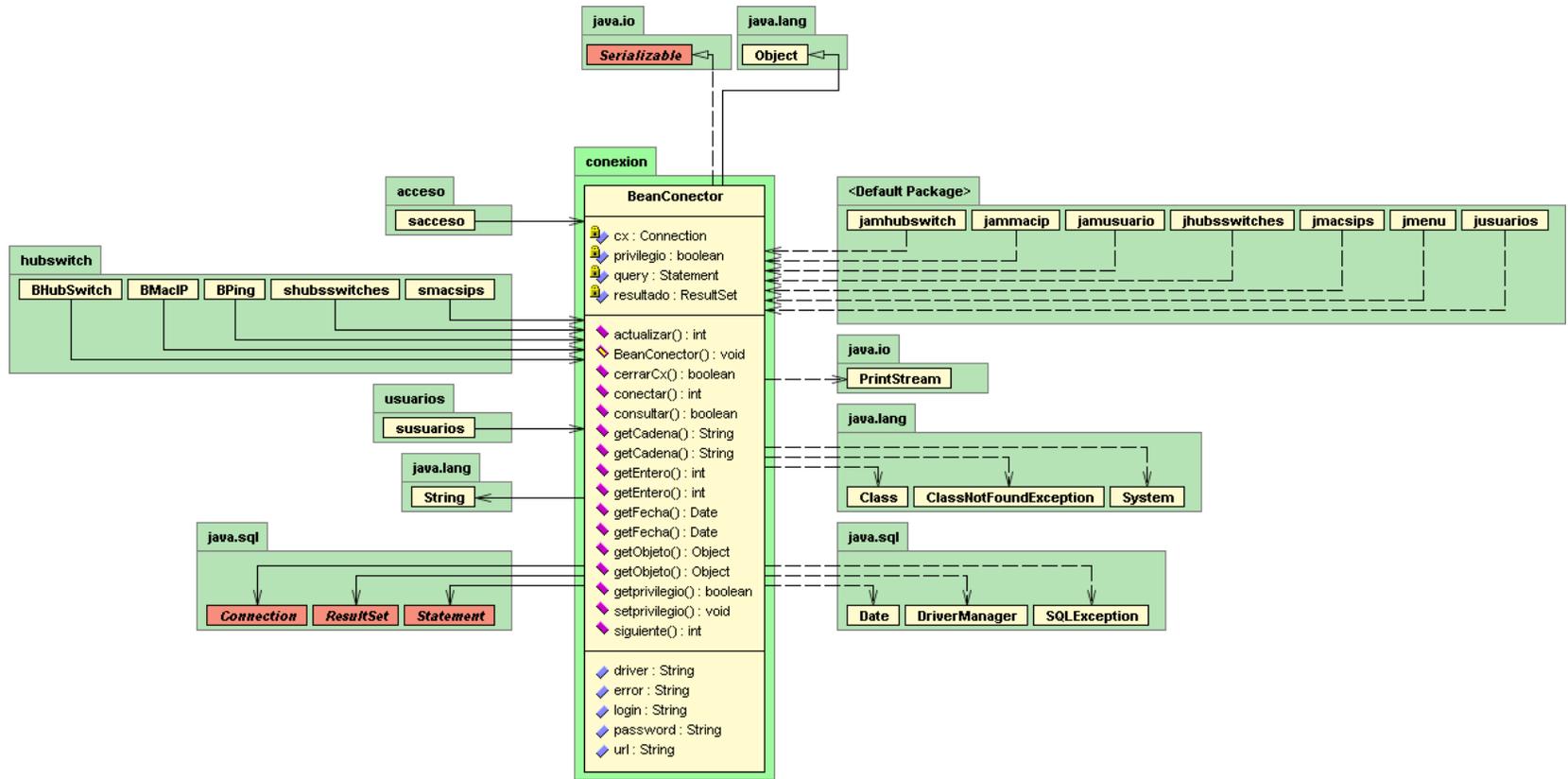


Paquete conexión

Bconexión

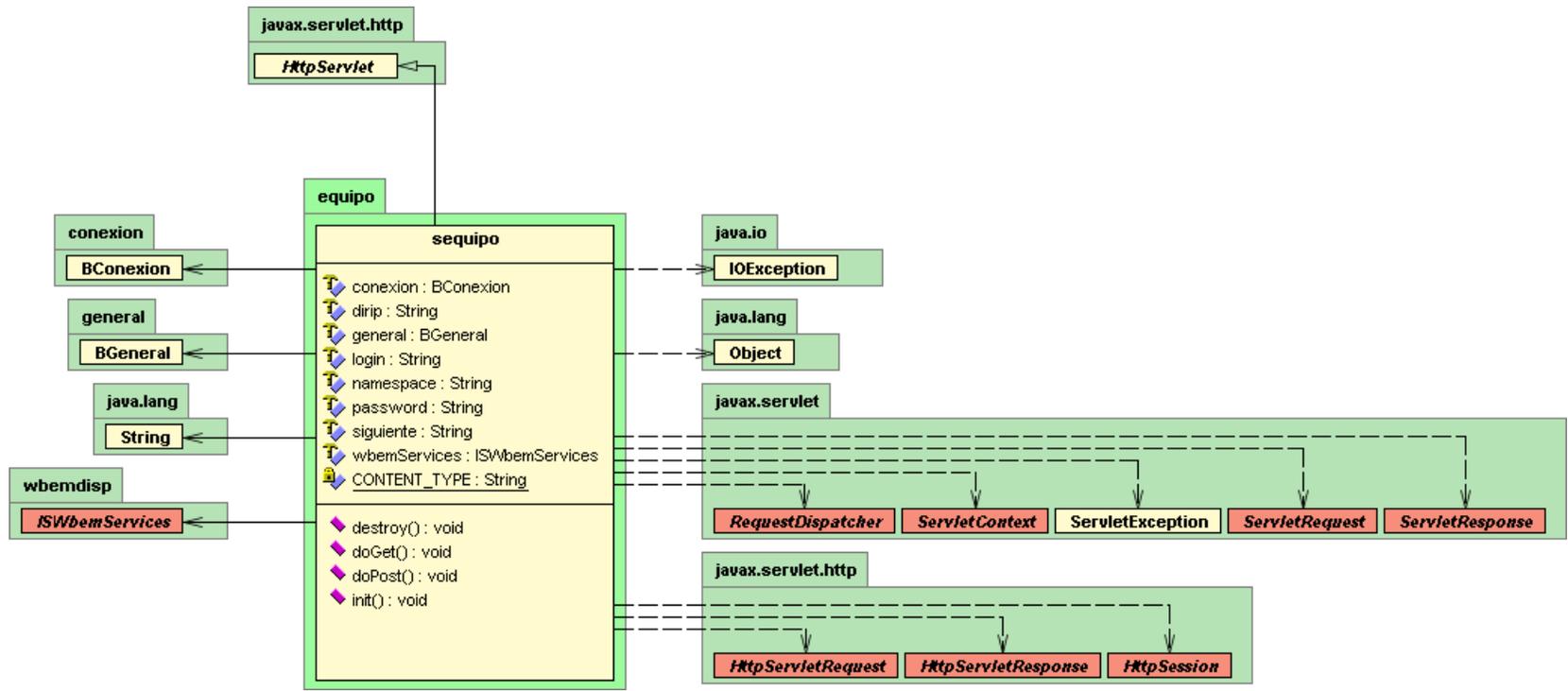


BeanConector



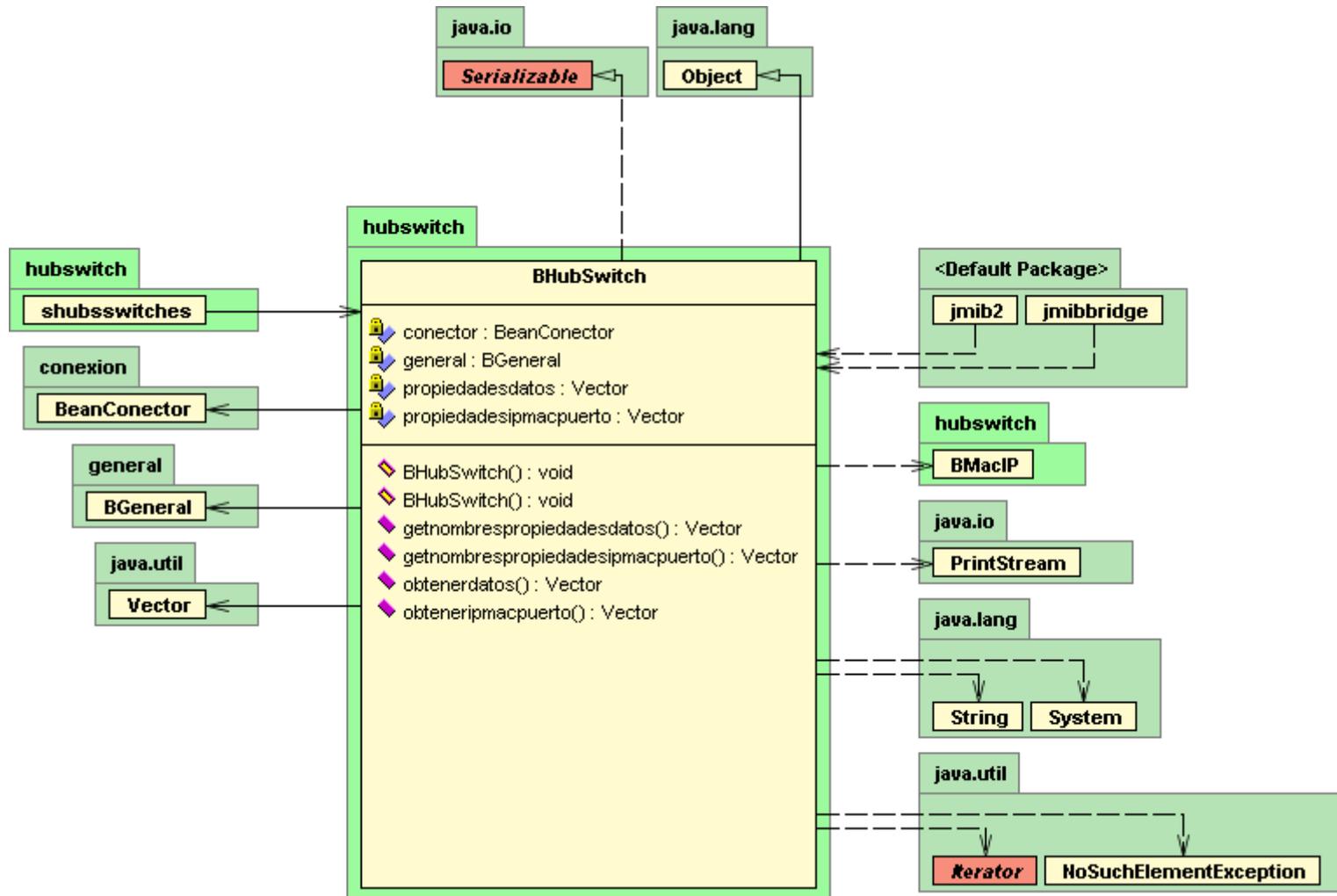
Paquete equipo

Sequipo

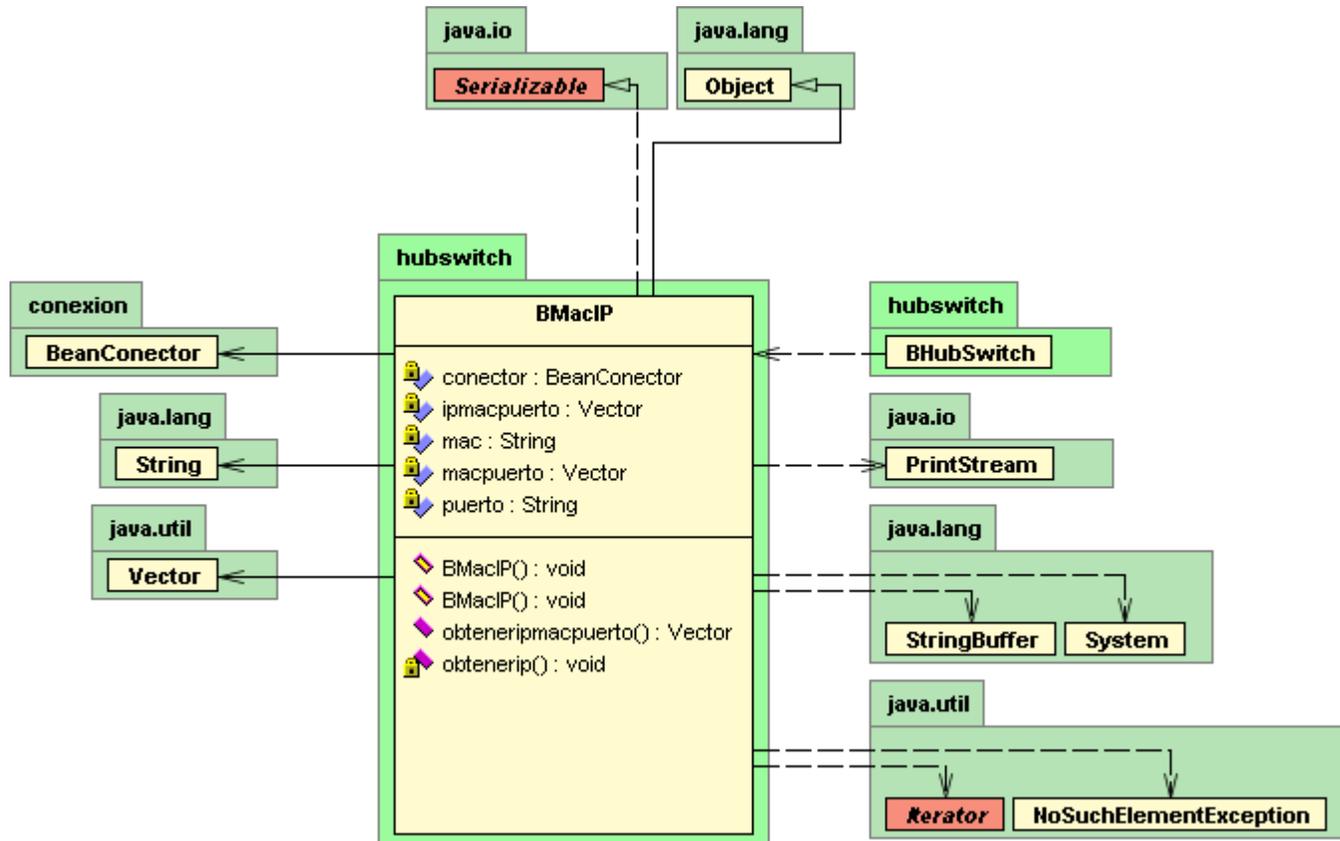


Paquete hubswitch

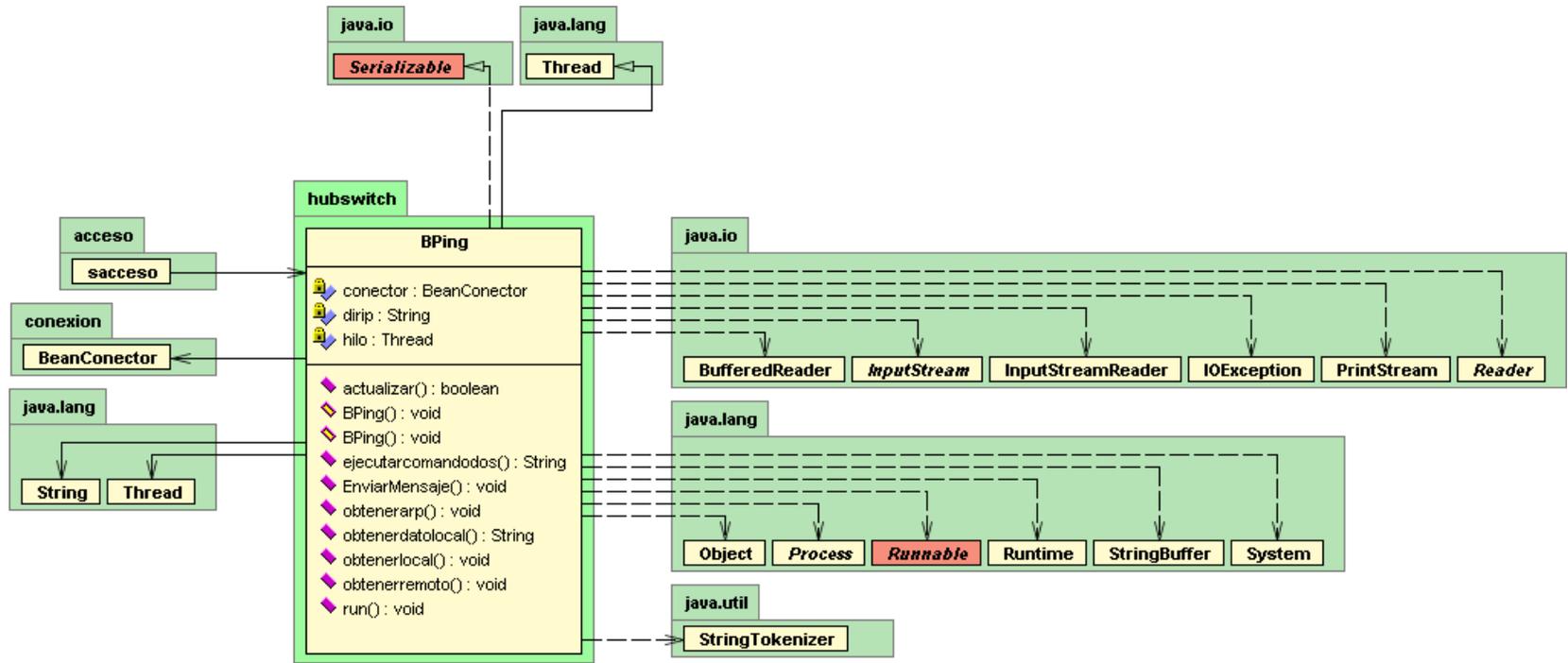
BHubSwitch



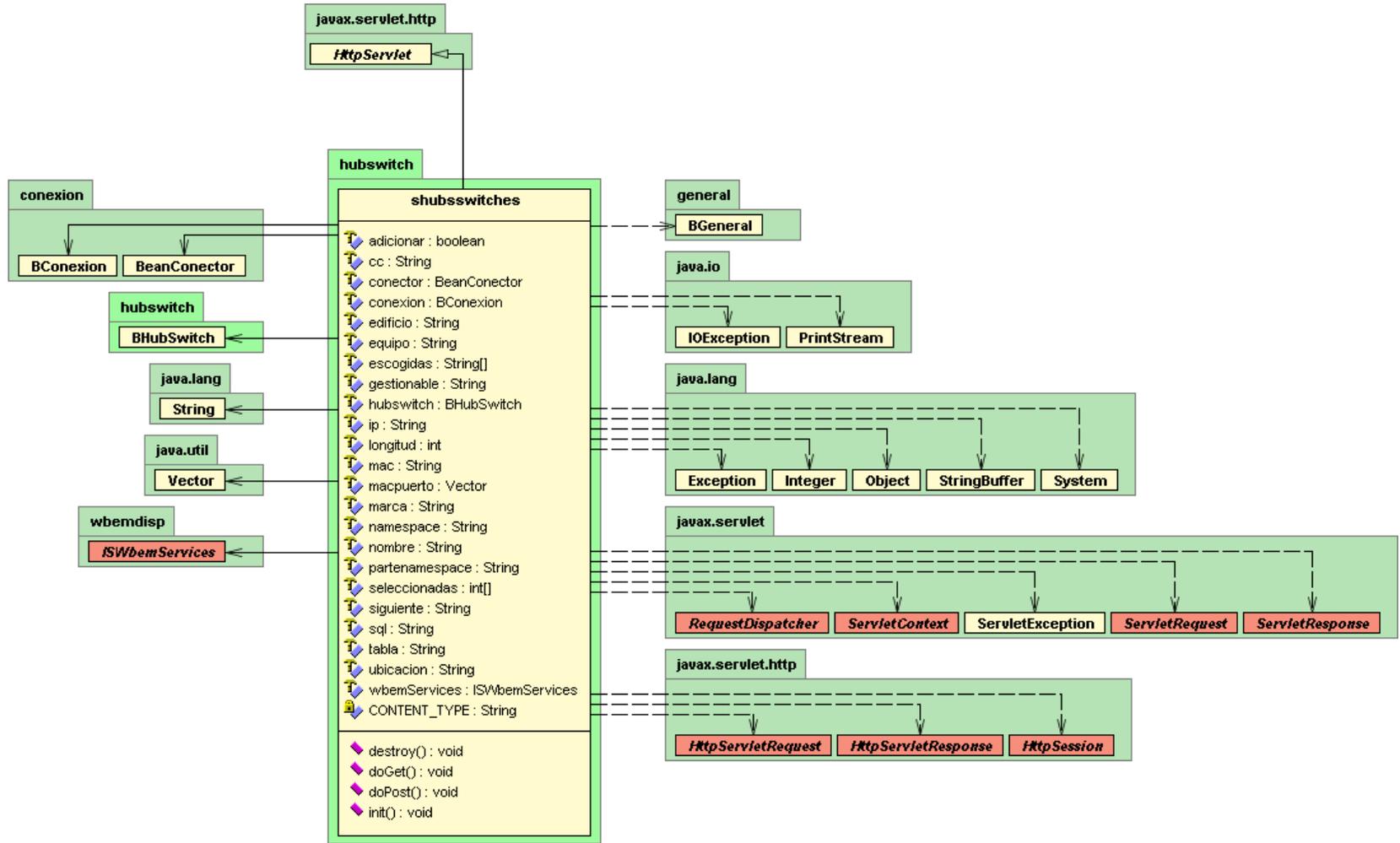
BMacIP



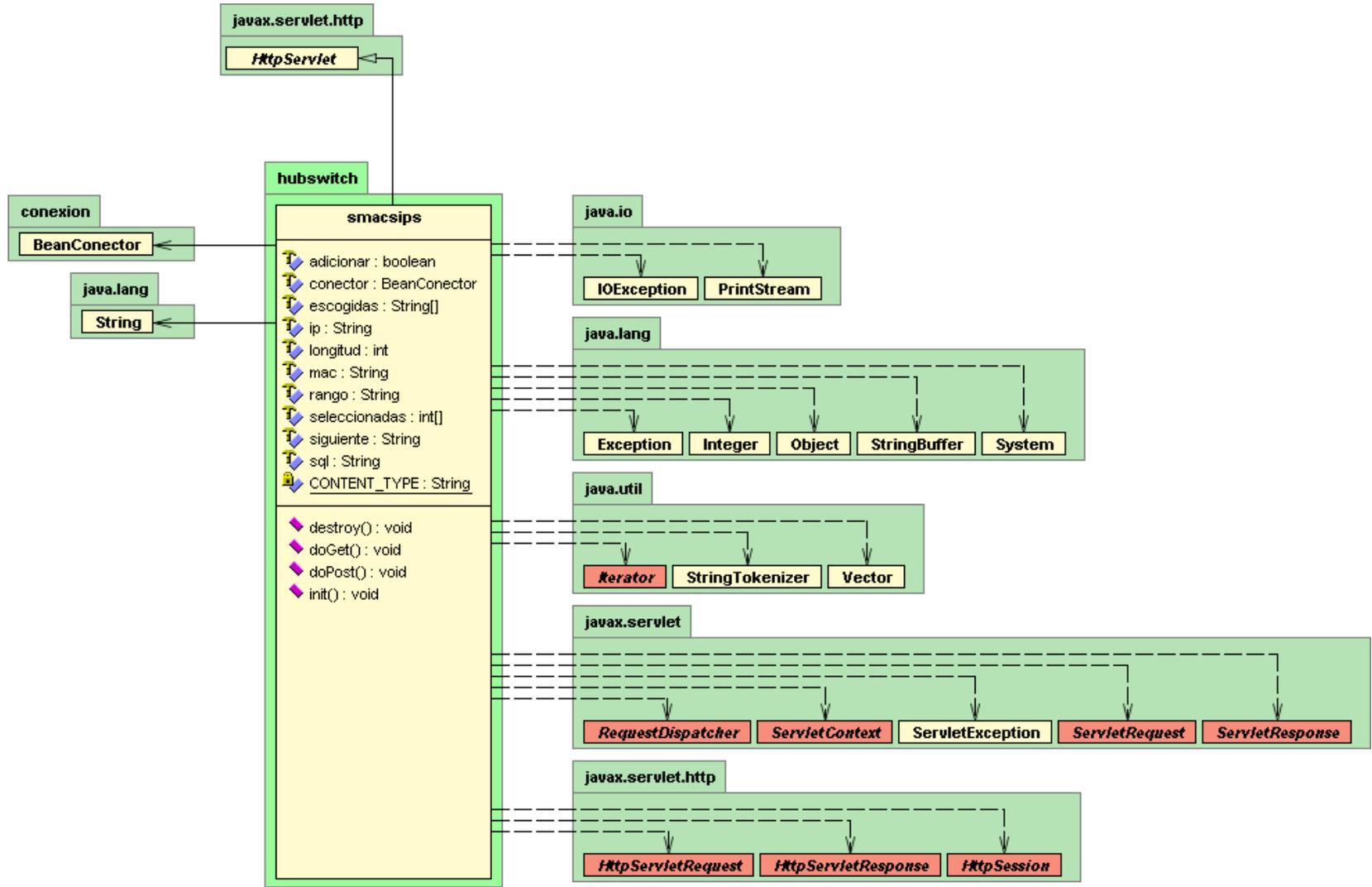
BPing



shubsswitches

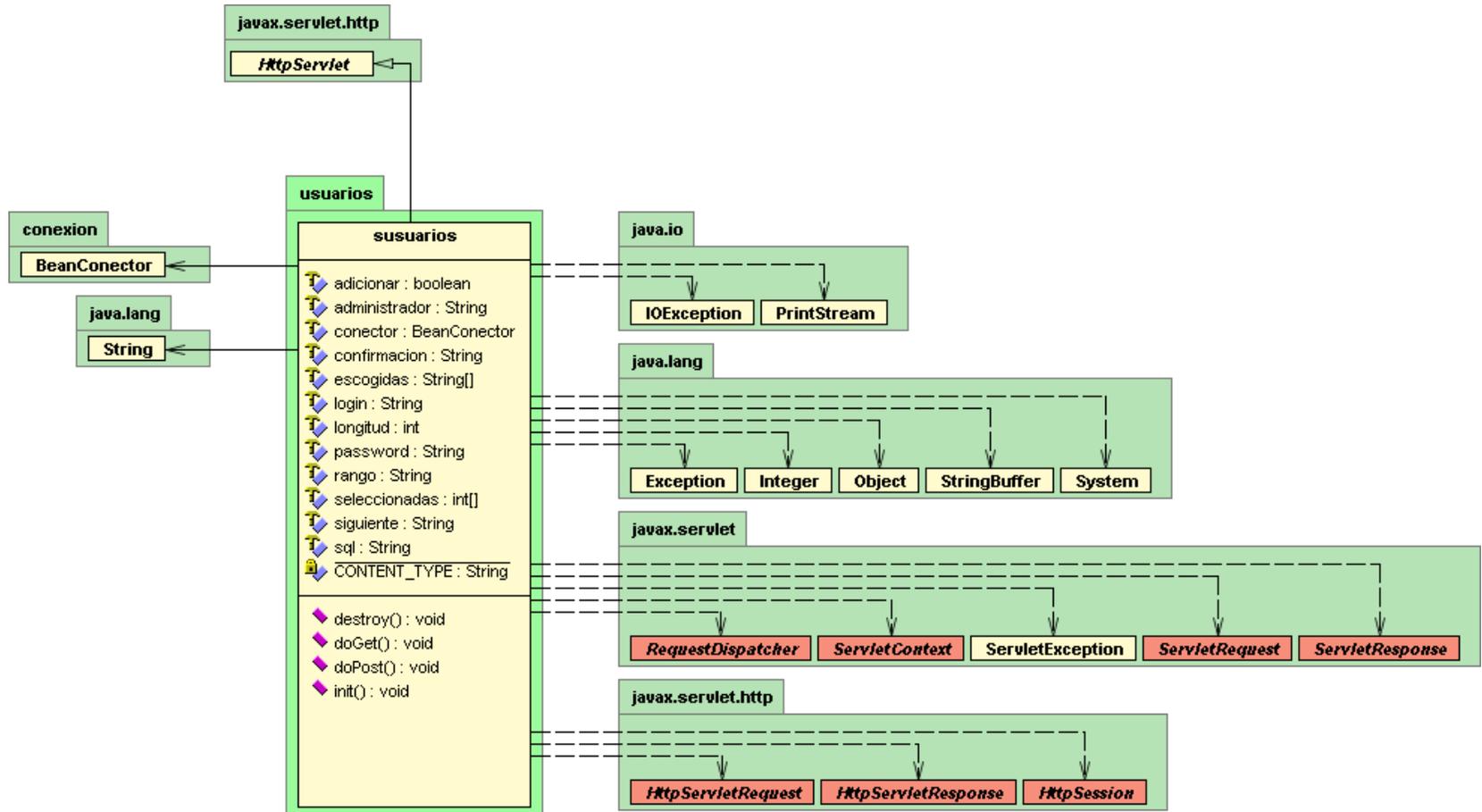


smacsips



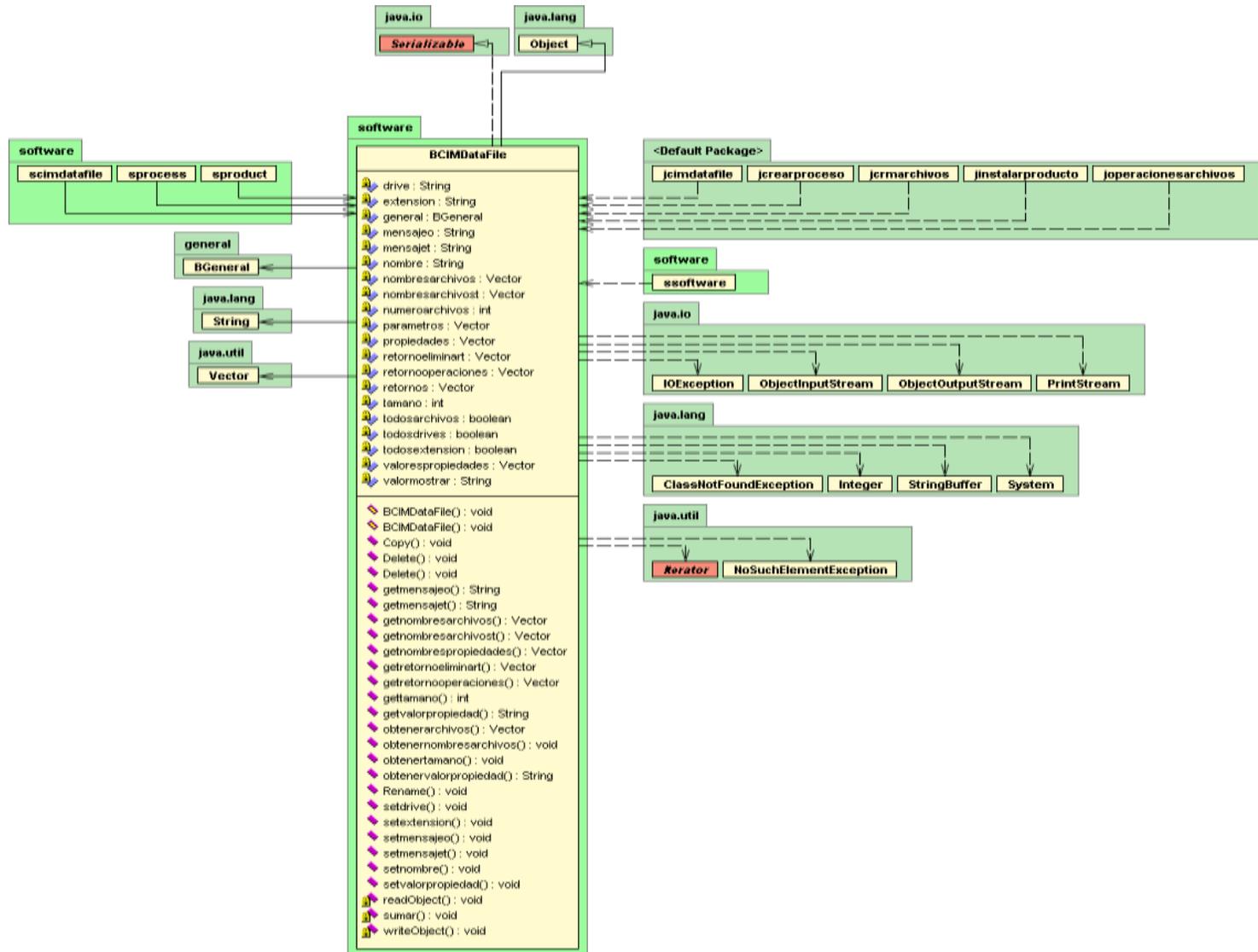
Paquete usuarios

susuarios

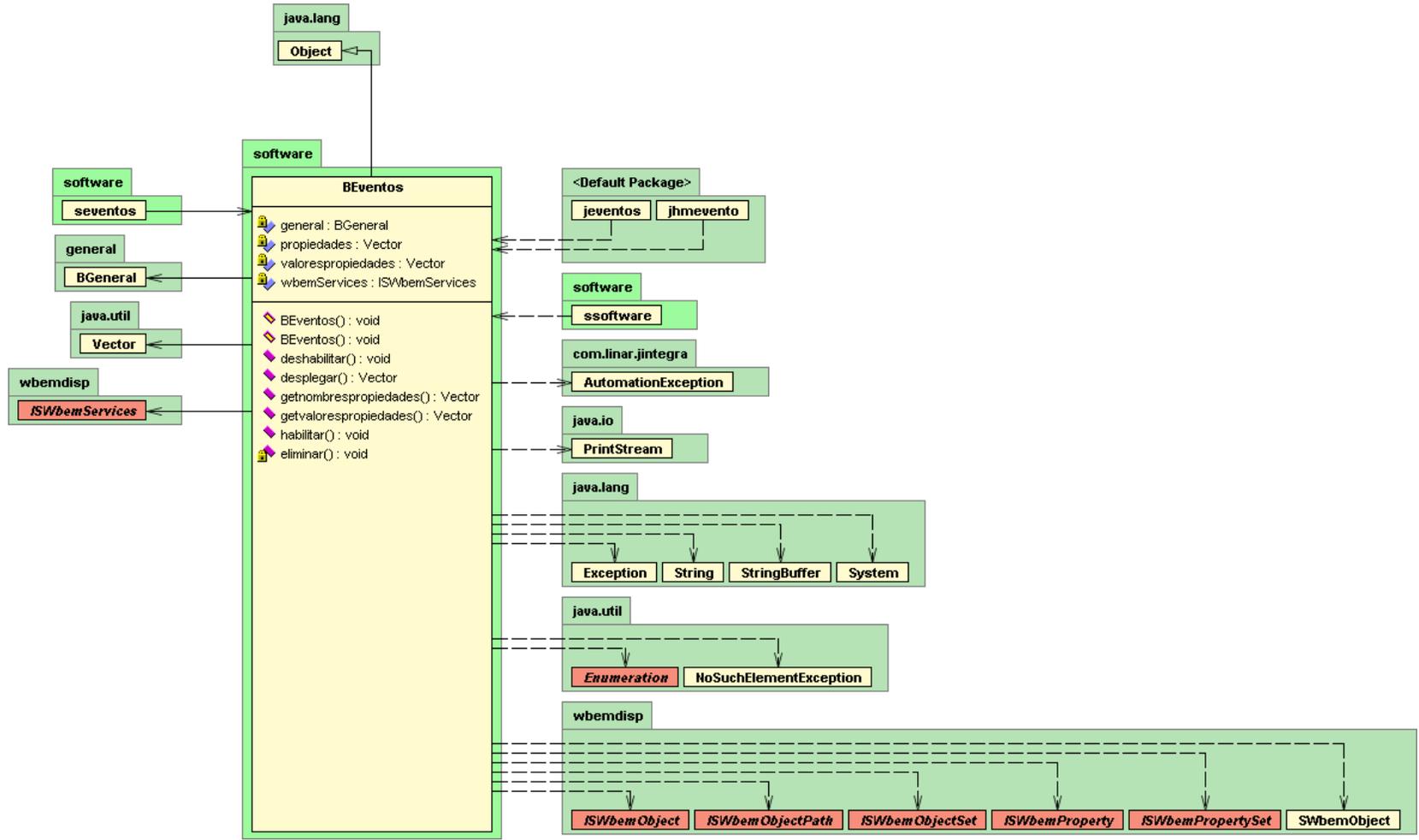


Paquete software

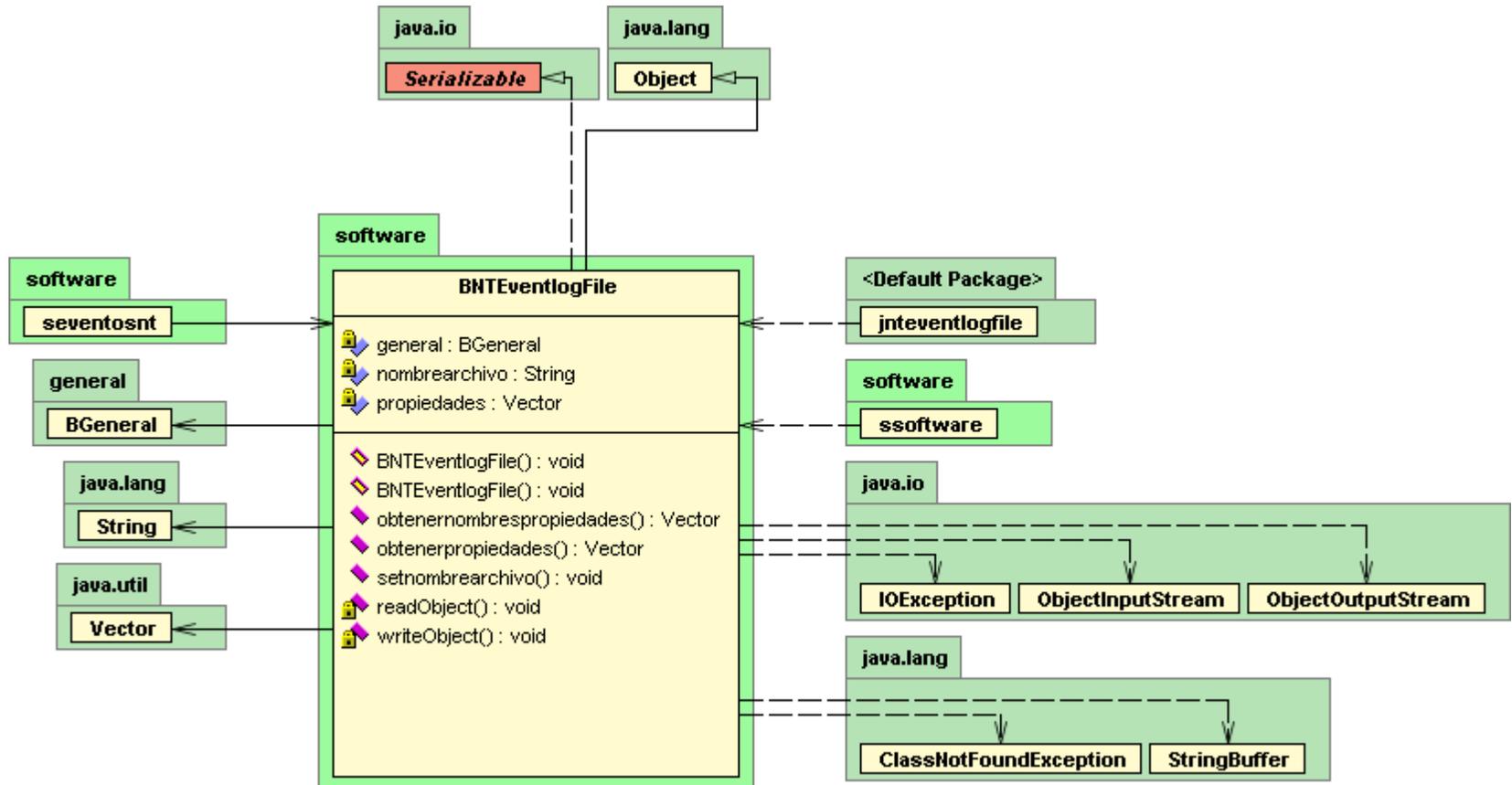
BCIMDataFile



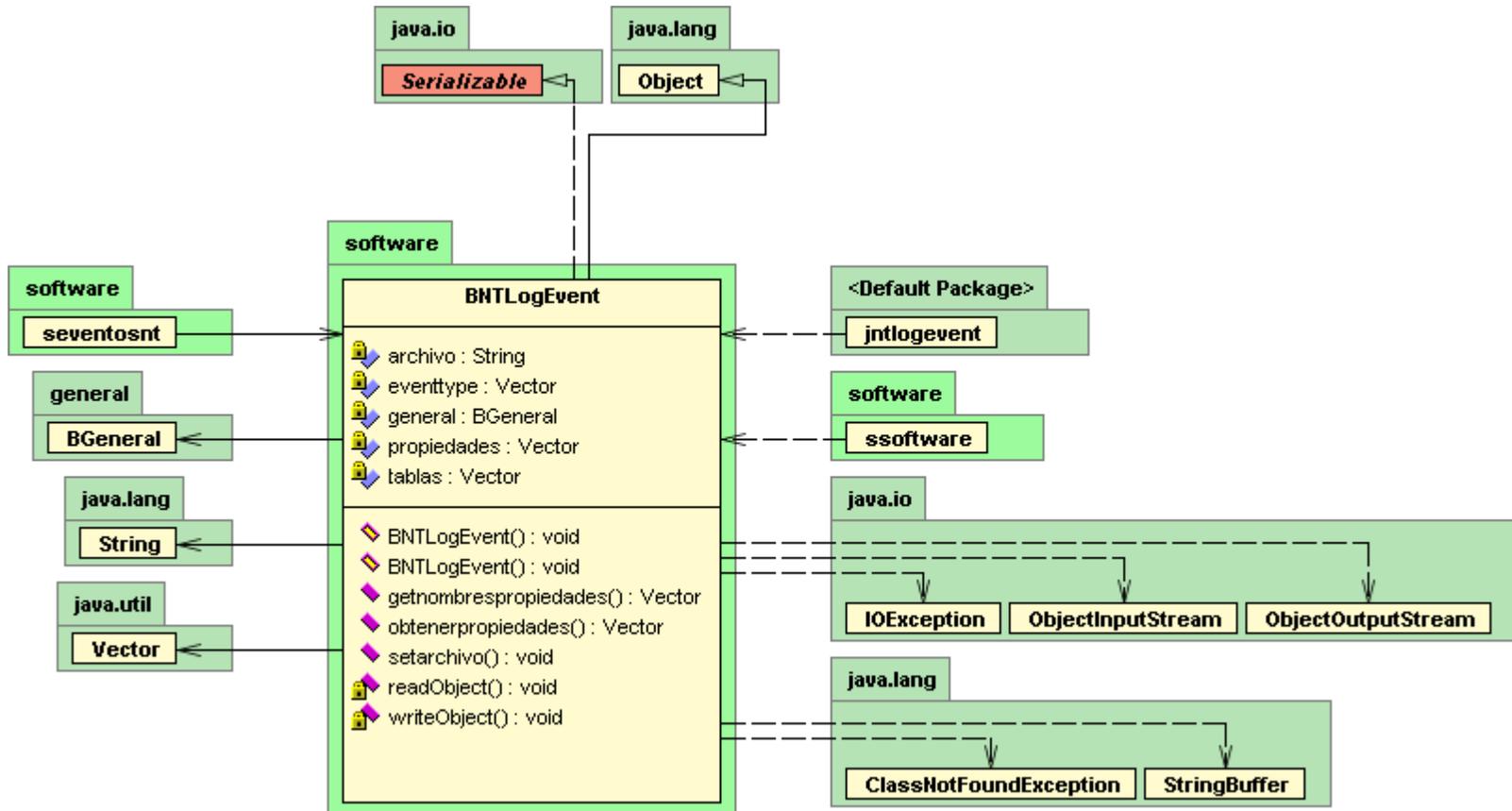
BEventos



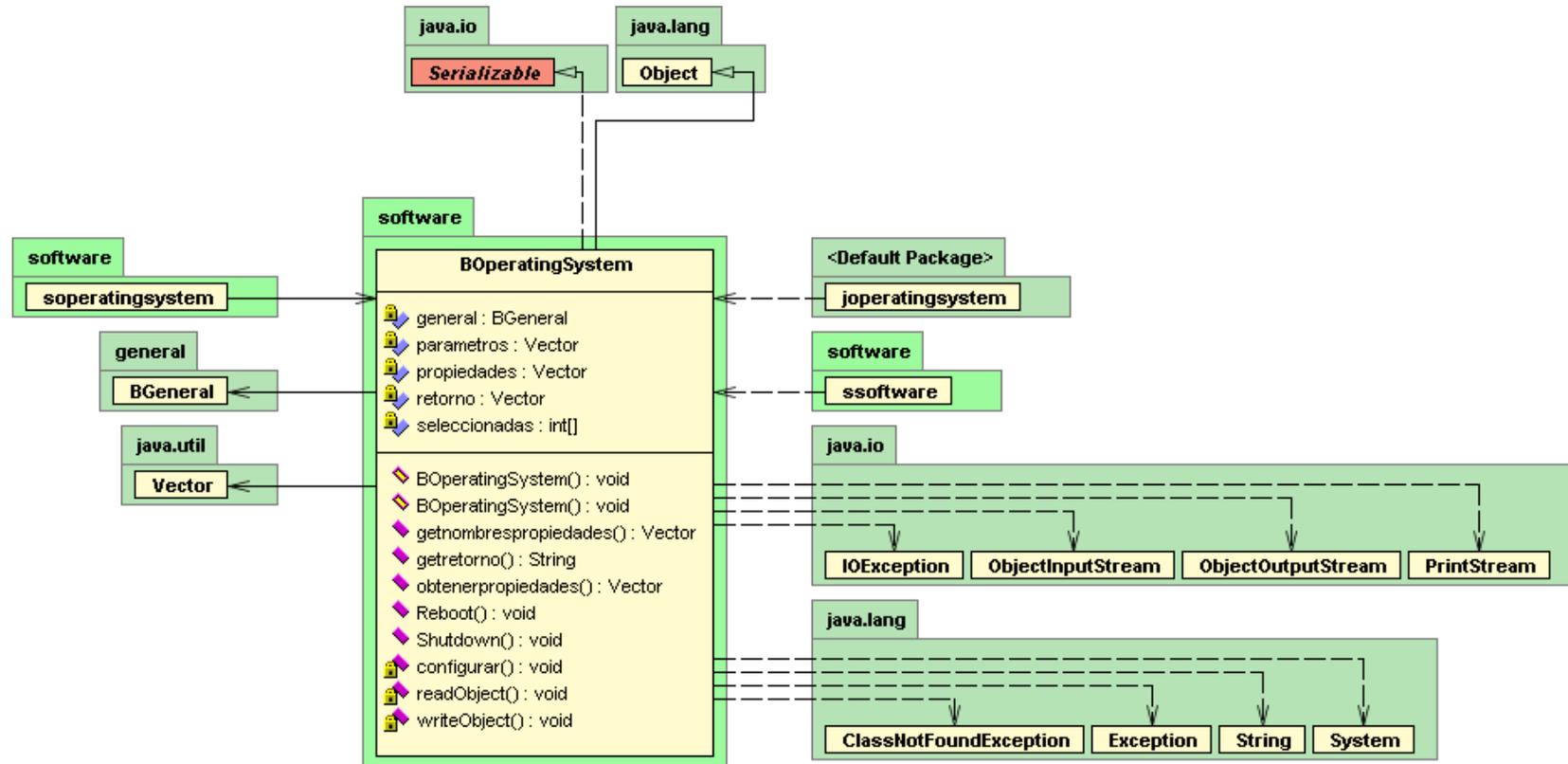
BNTEventlogFile



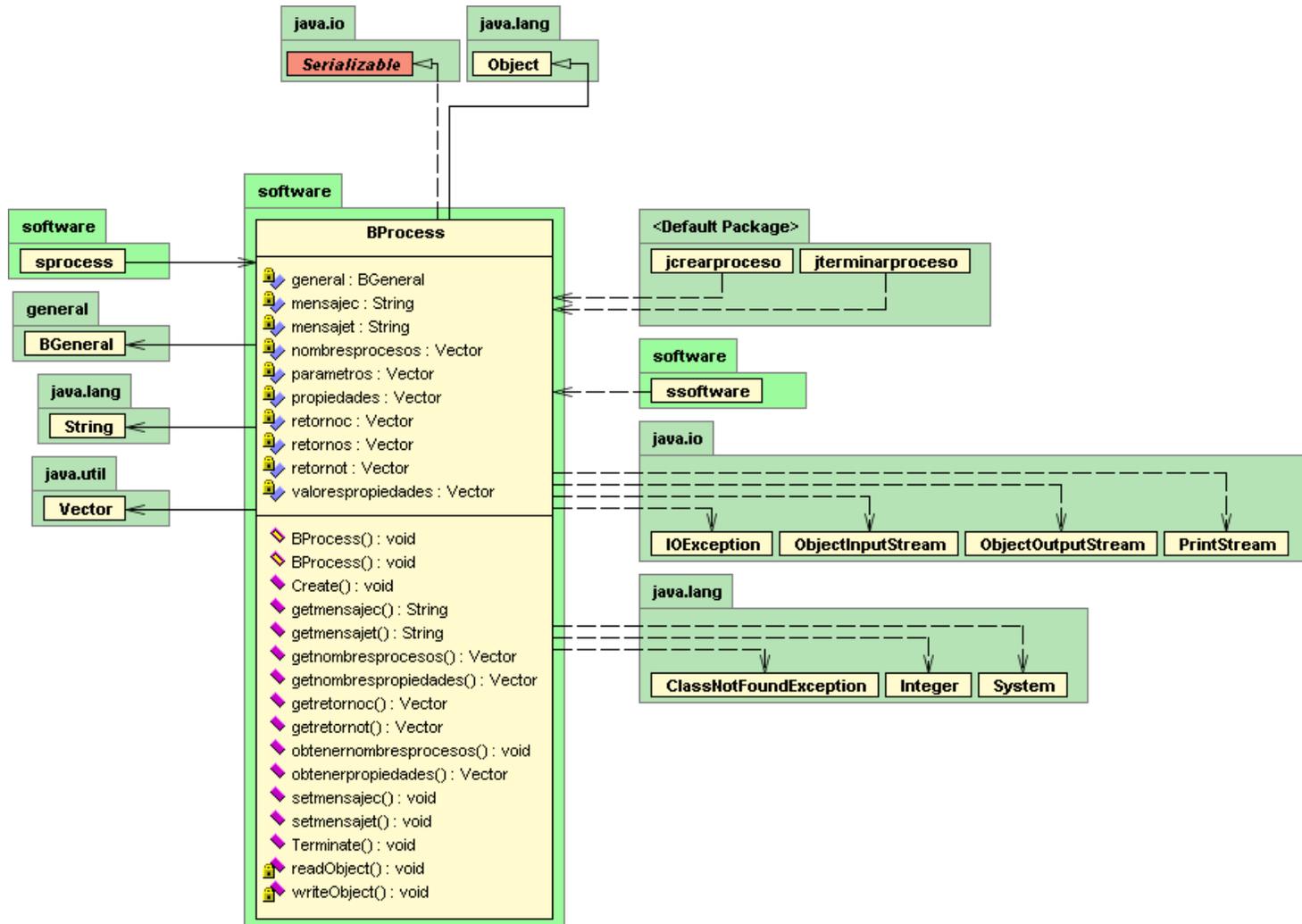
BNTLogEvent



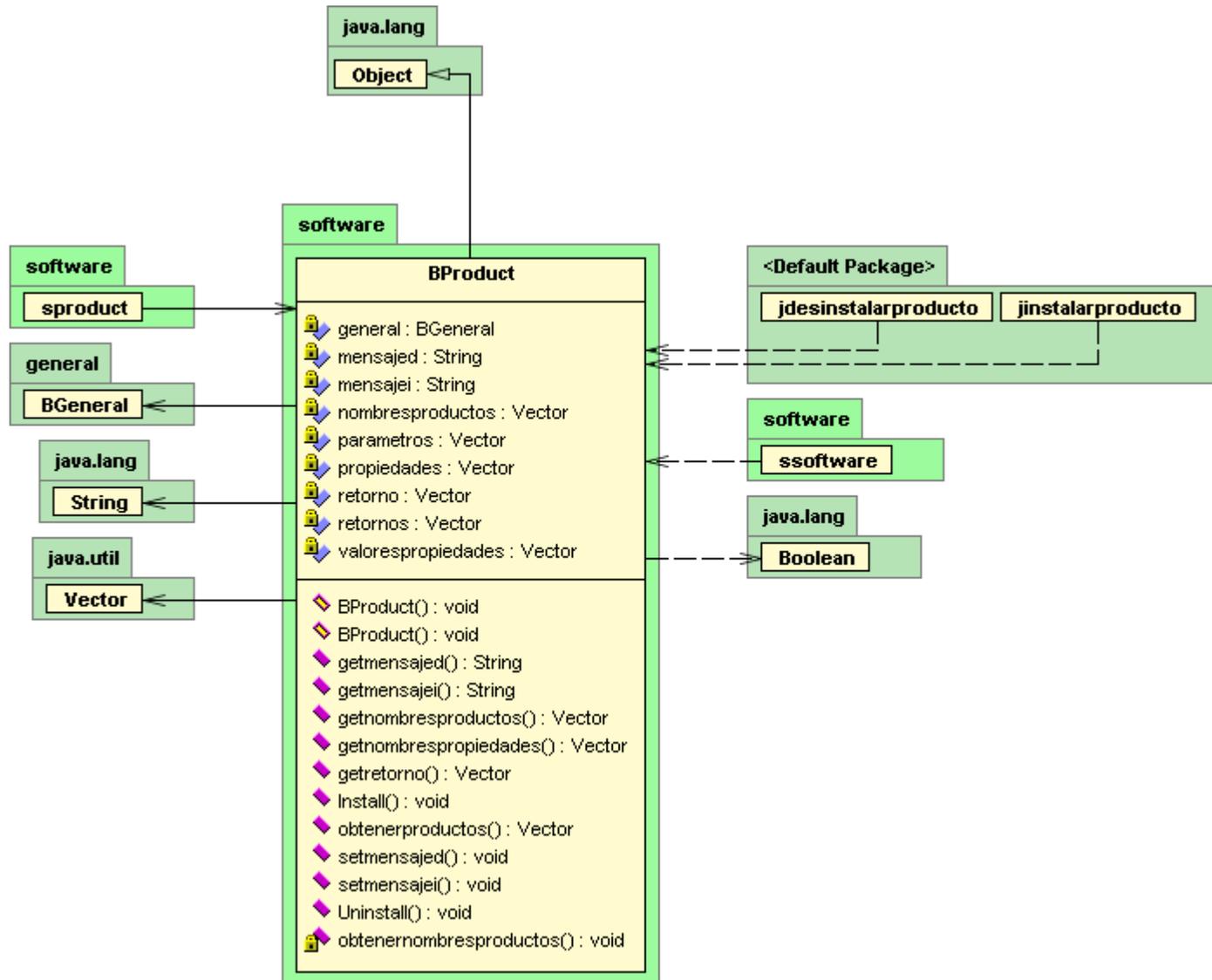
BOperatingSystem



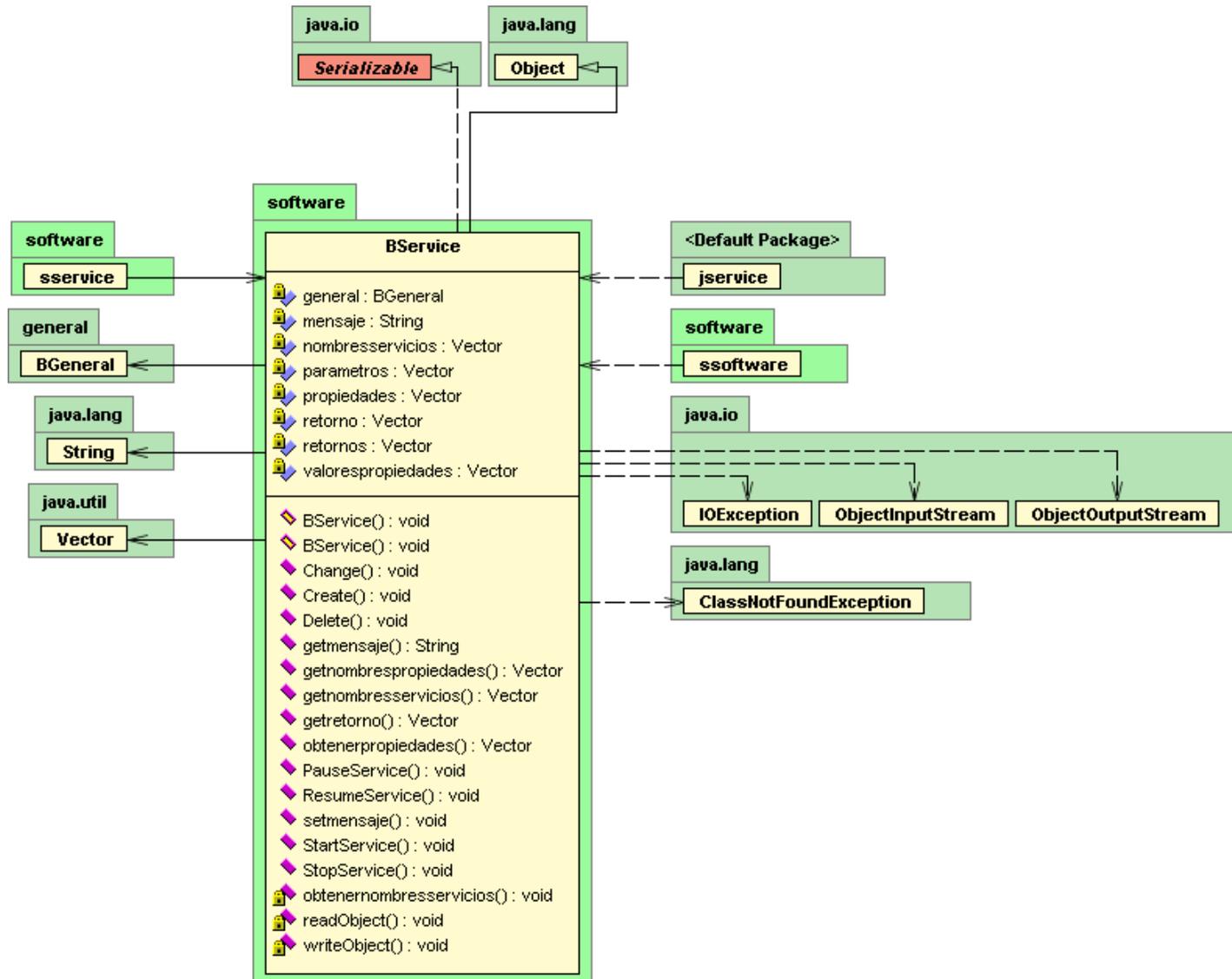
Bprocess



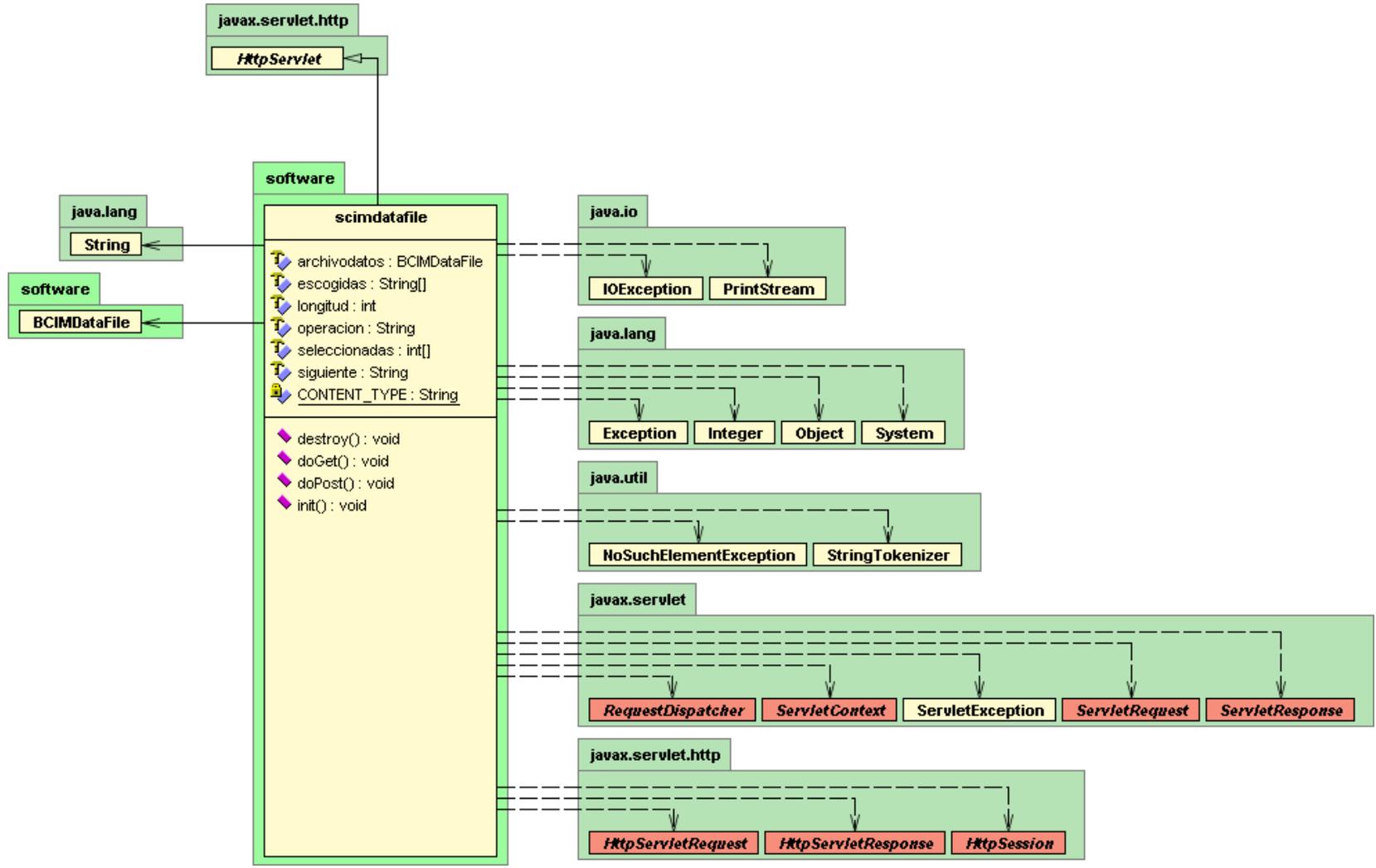
BProduct



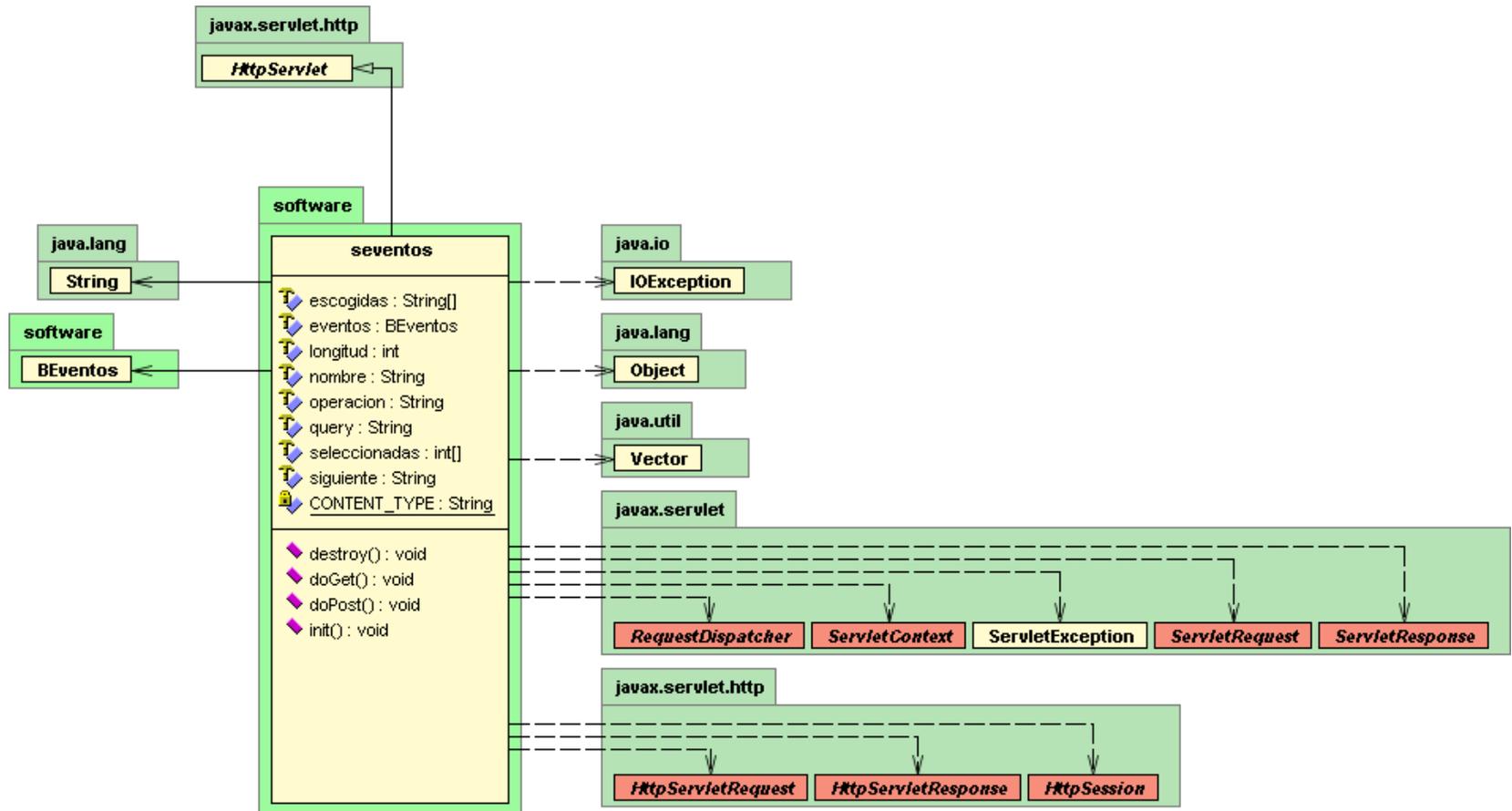
Bservice



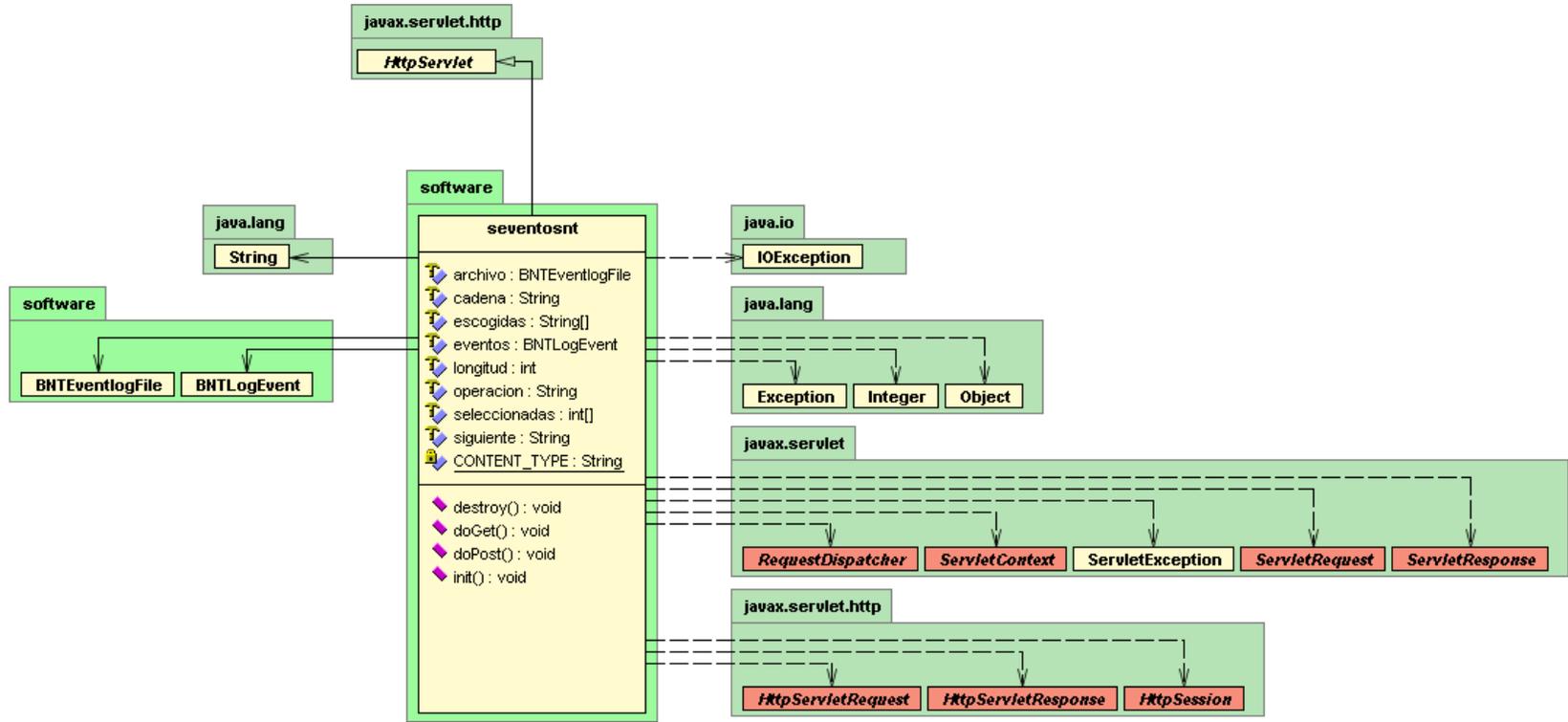
scimdatafile



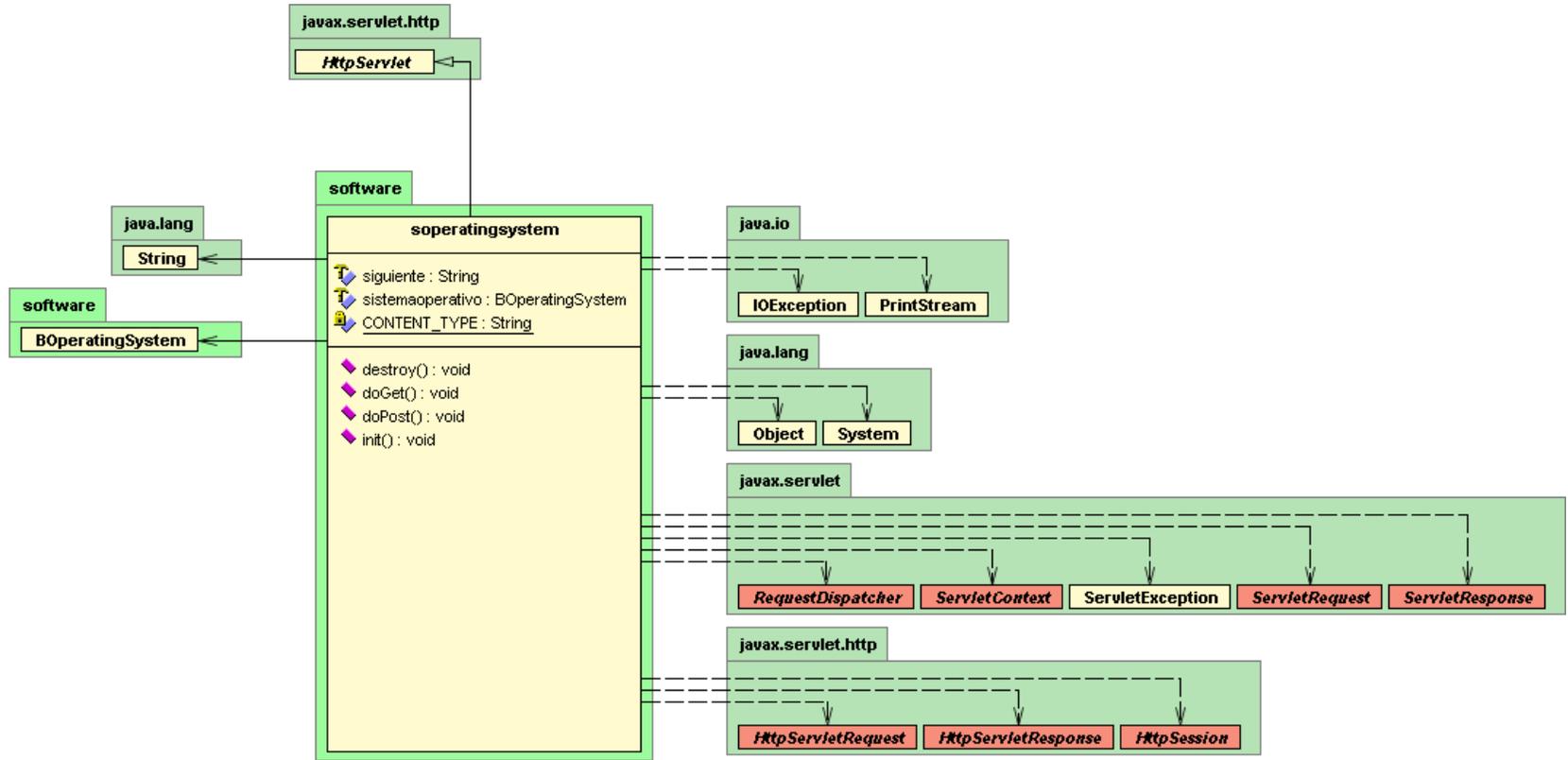
seventos



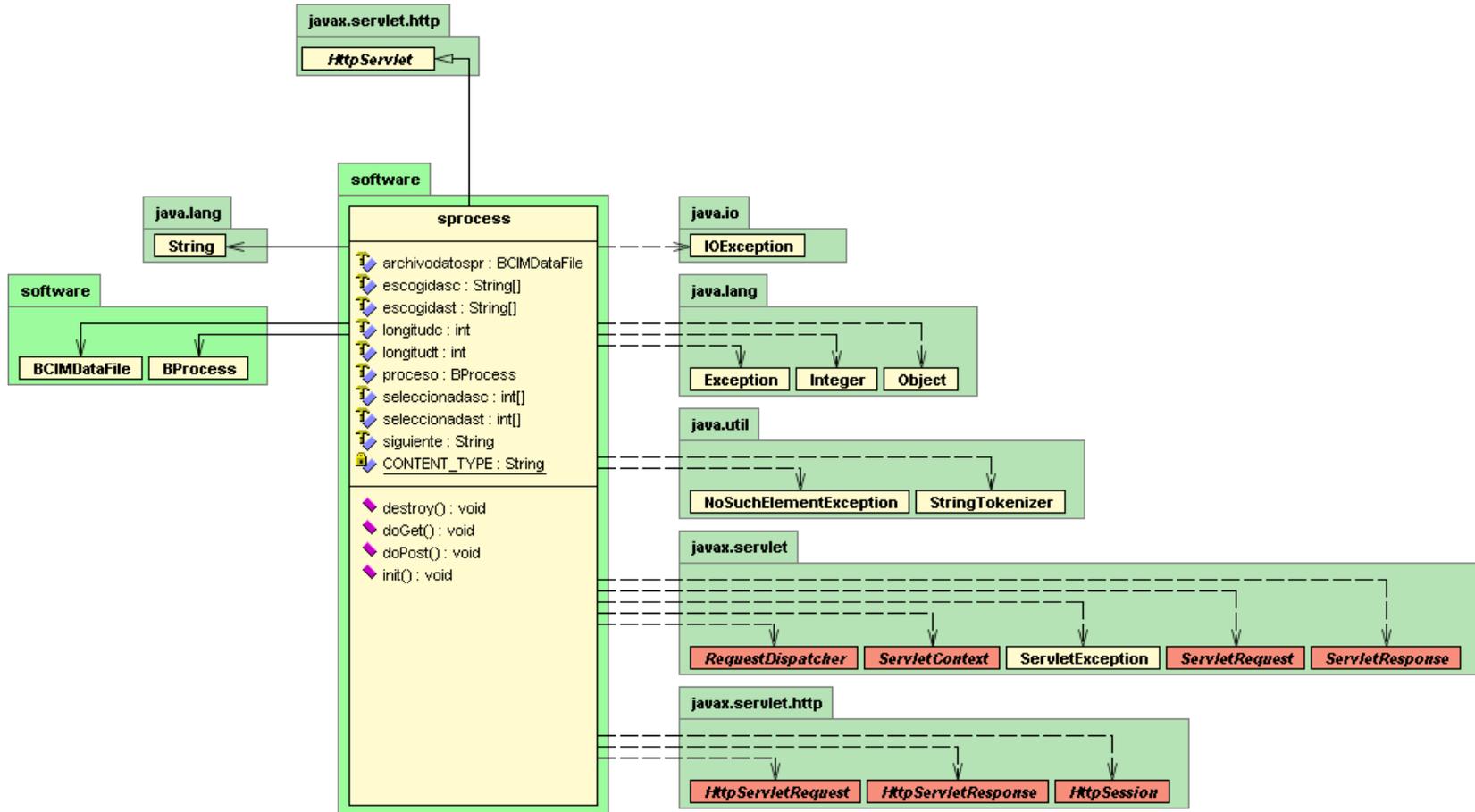
seventosnt



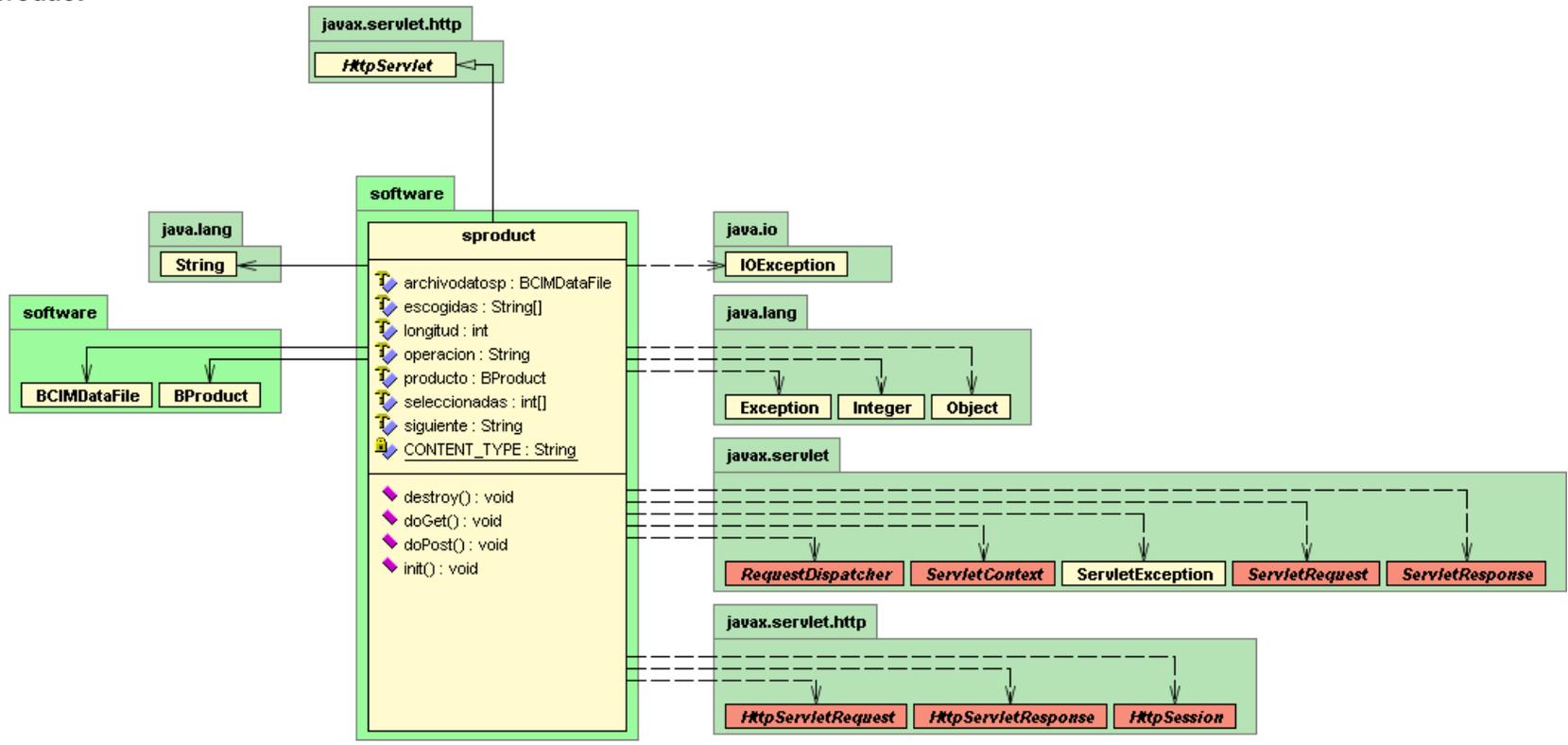
soperatingsystem



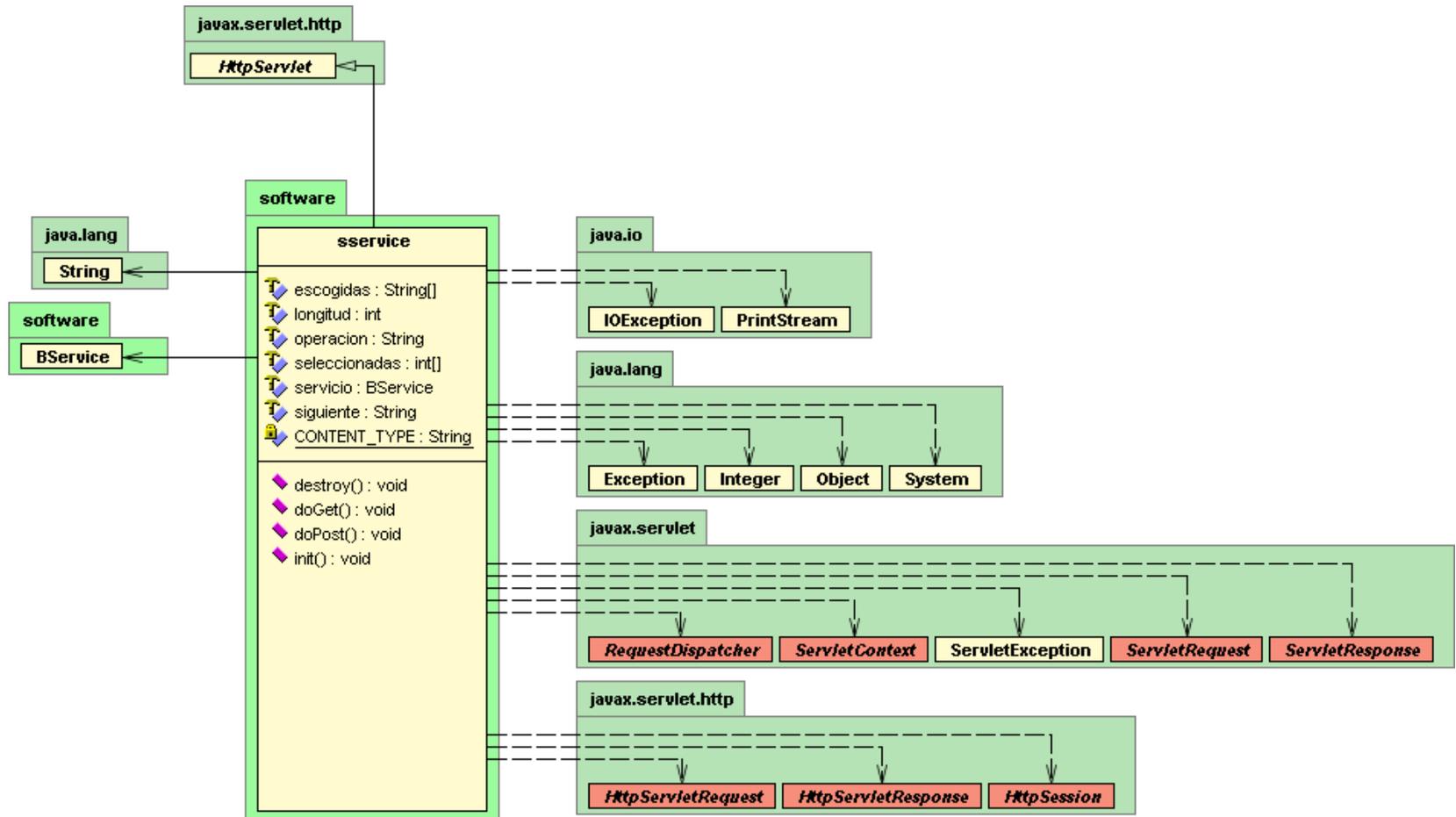
Sprocess



sproduct

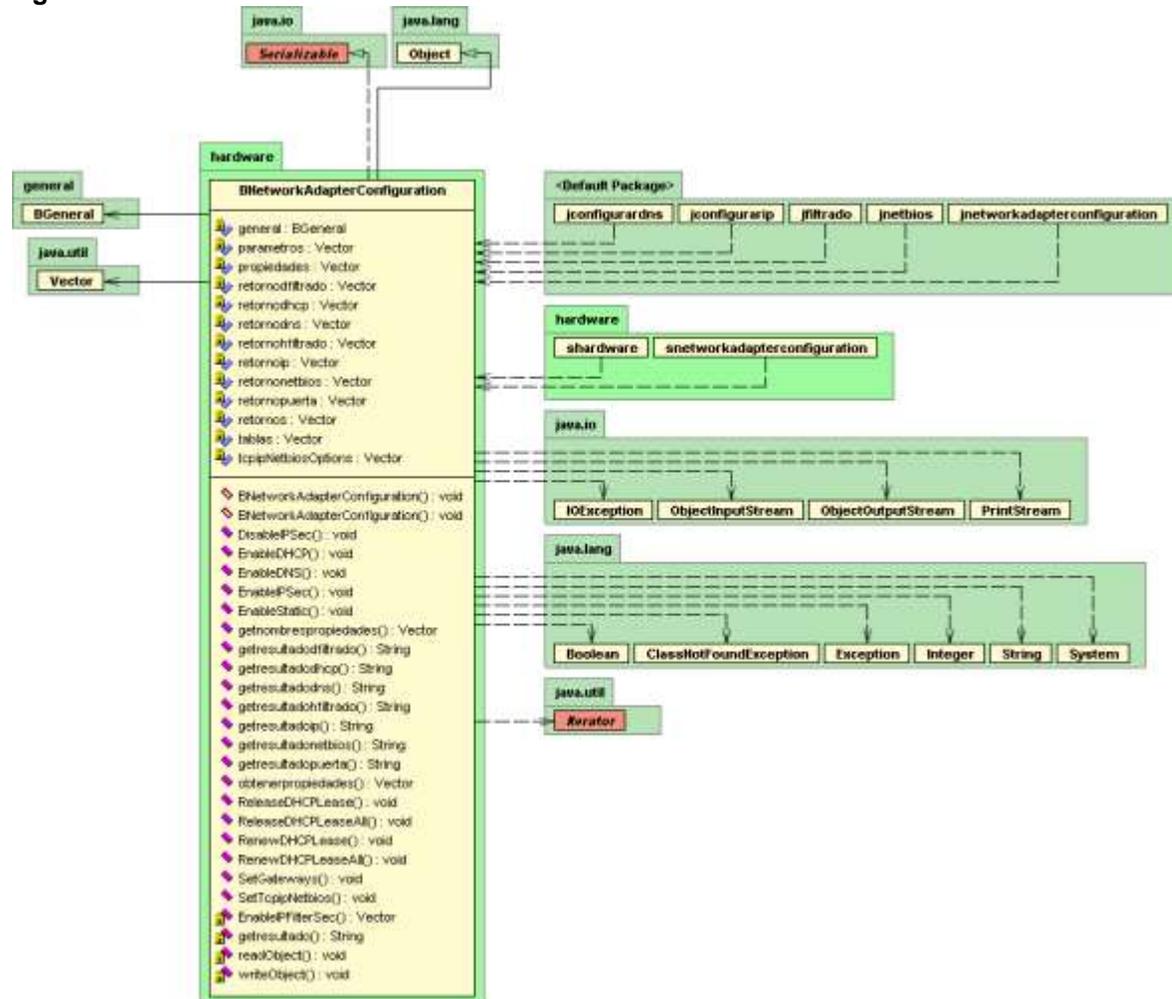


s-service

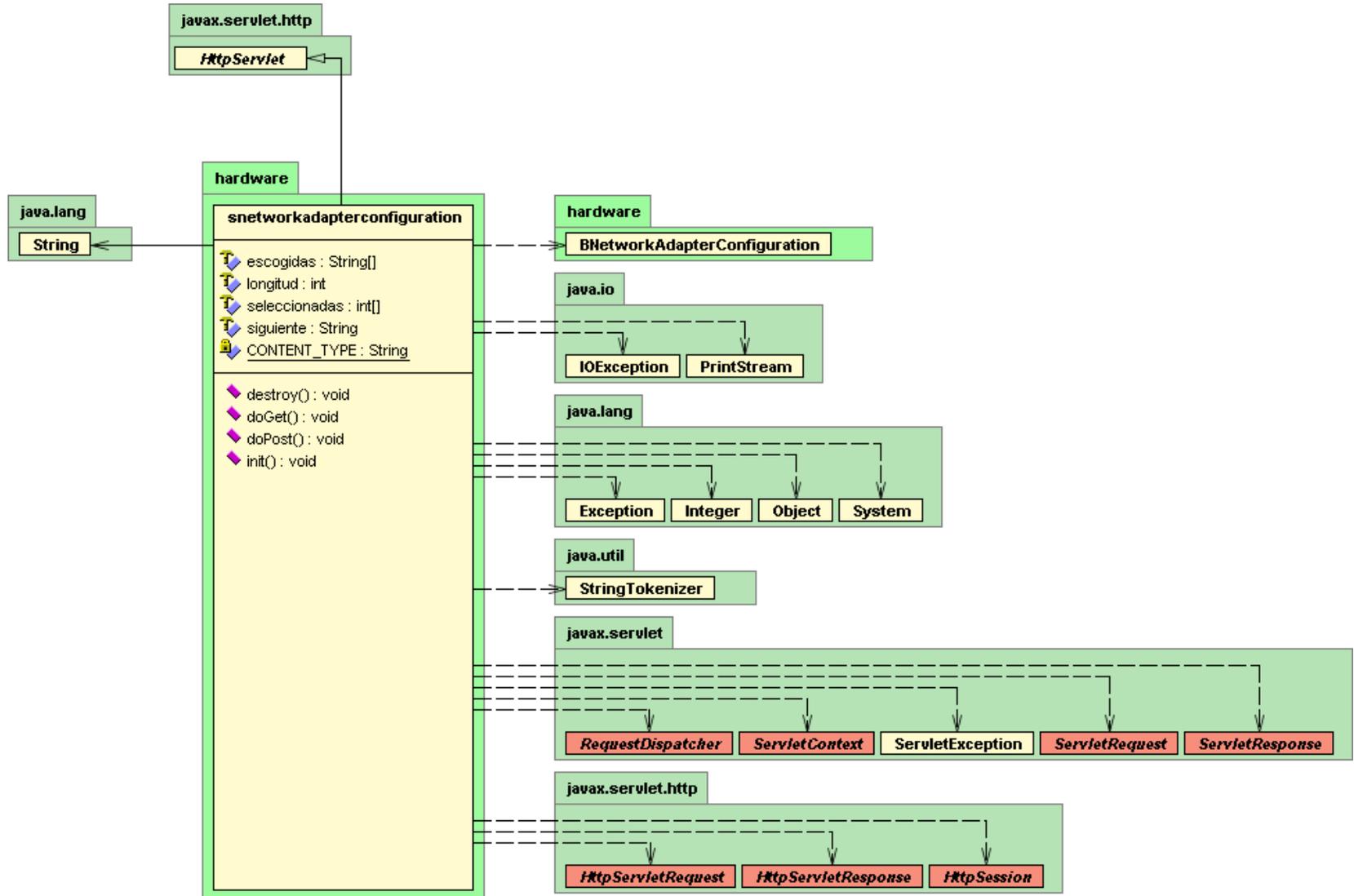


Paquete hardware

BNetworkAdapterConfiguration

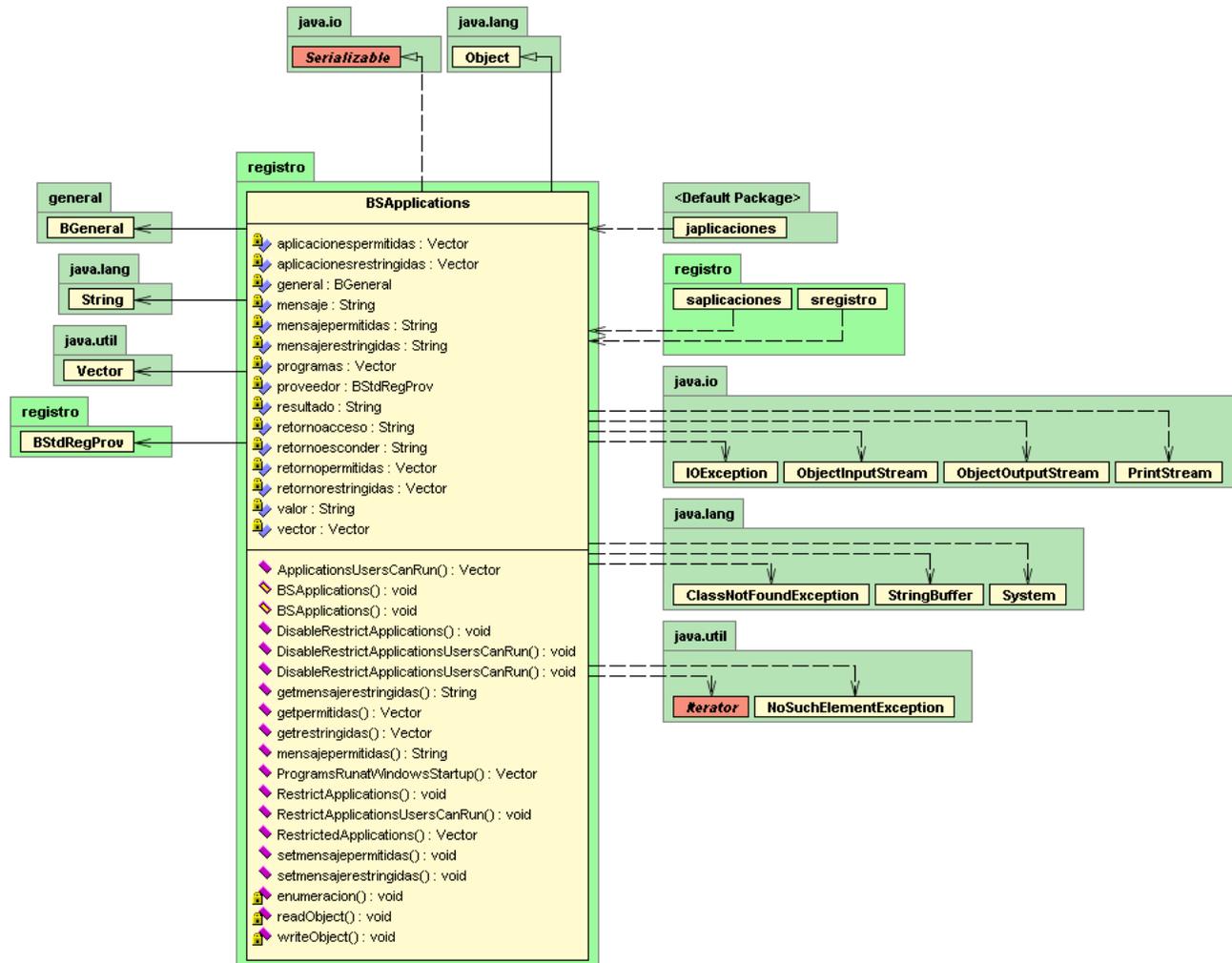


snetworkadapterconfiguration

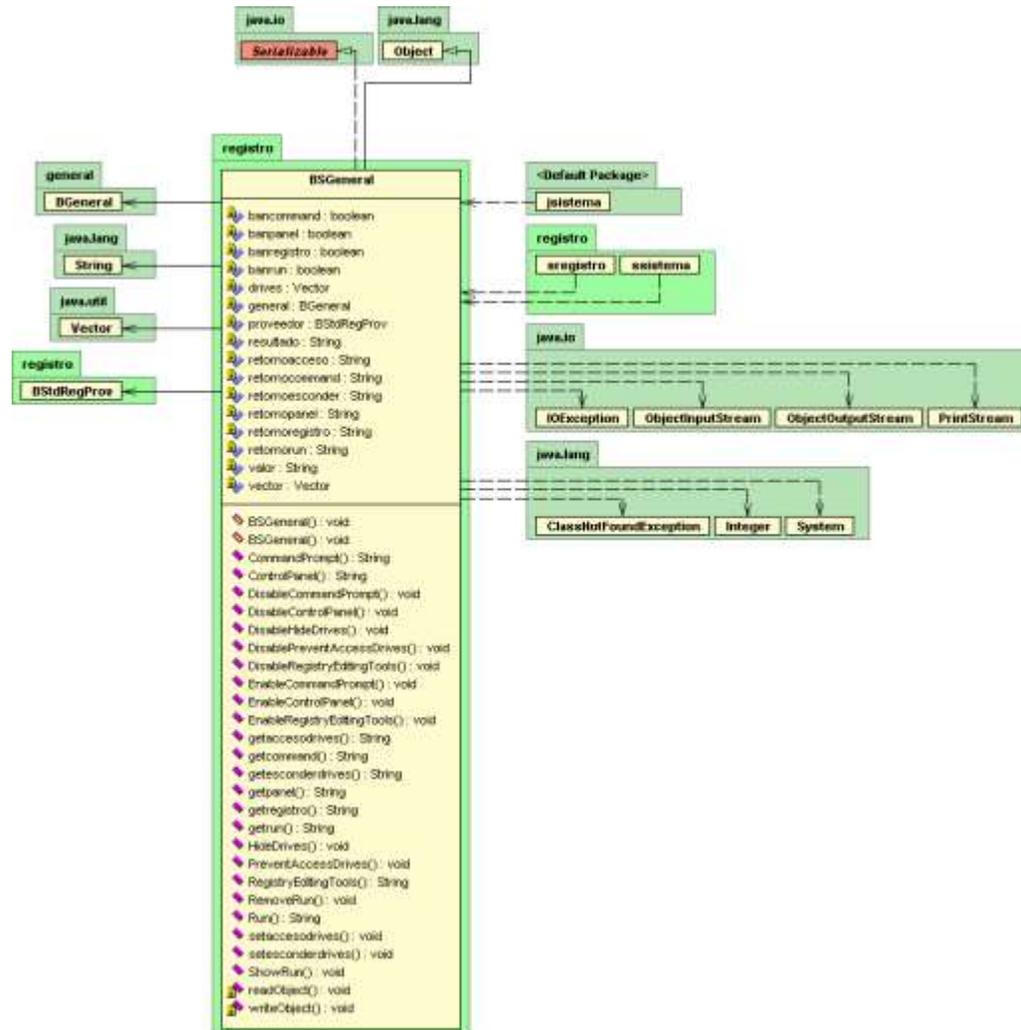


Paquete registro

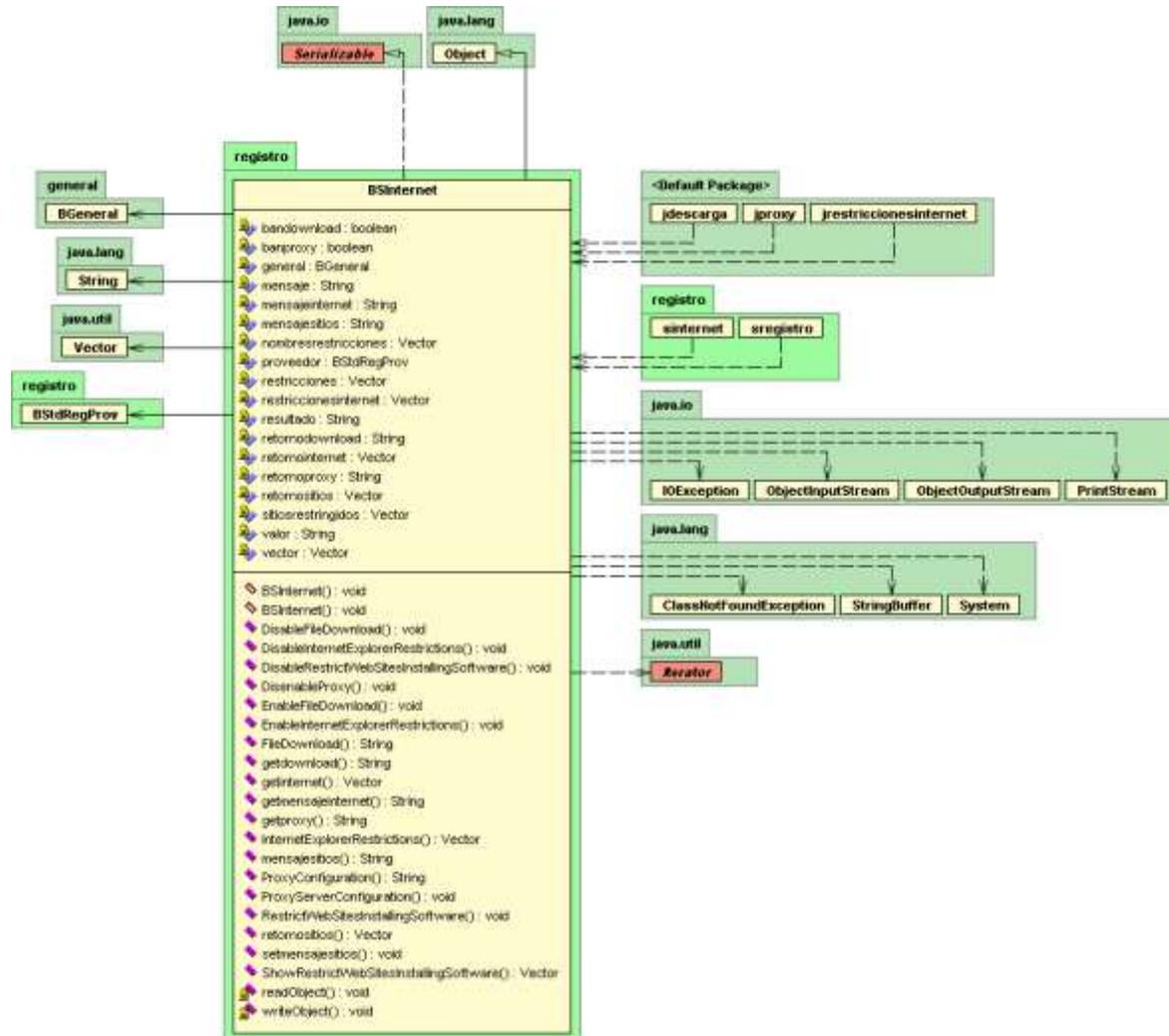
BSApplications



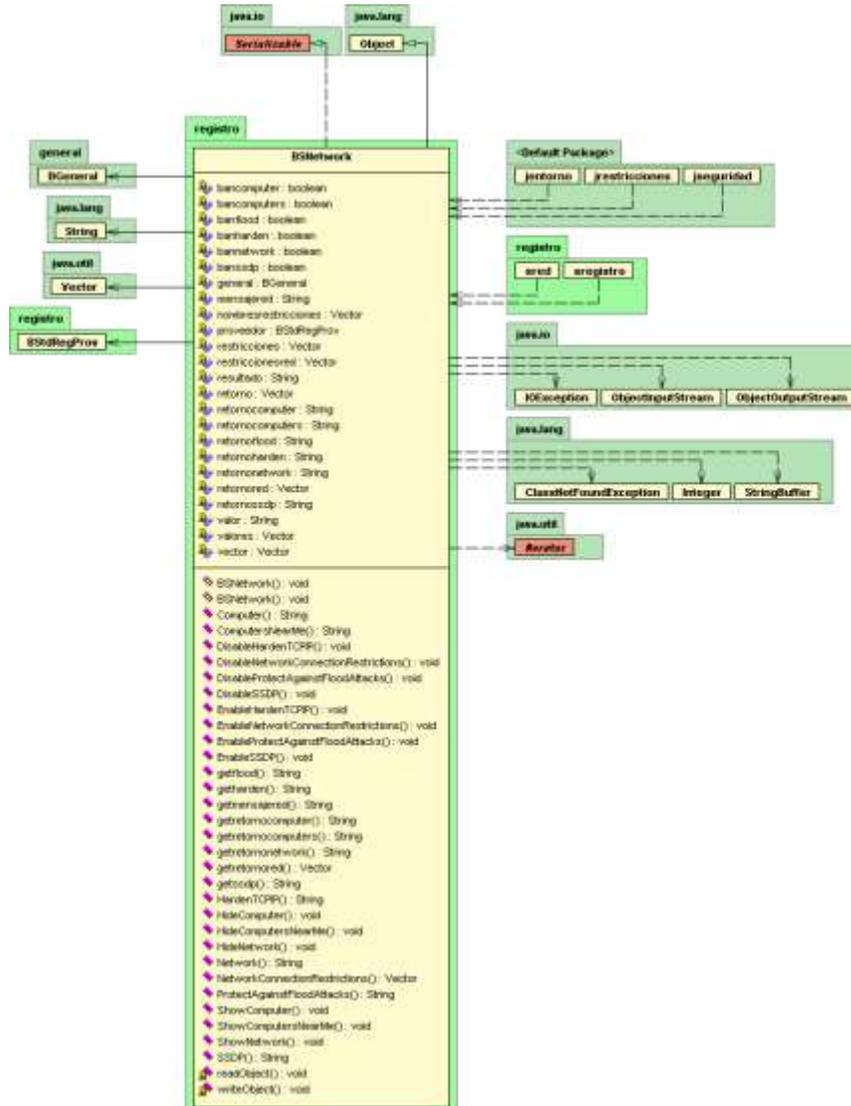
BSGeneral



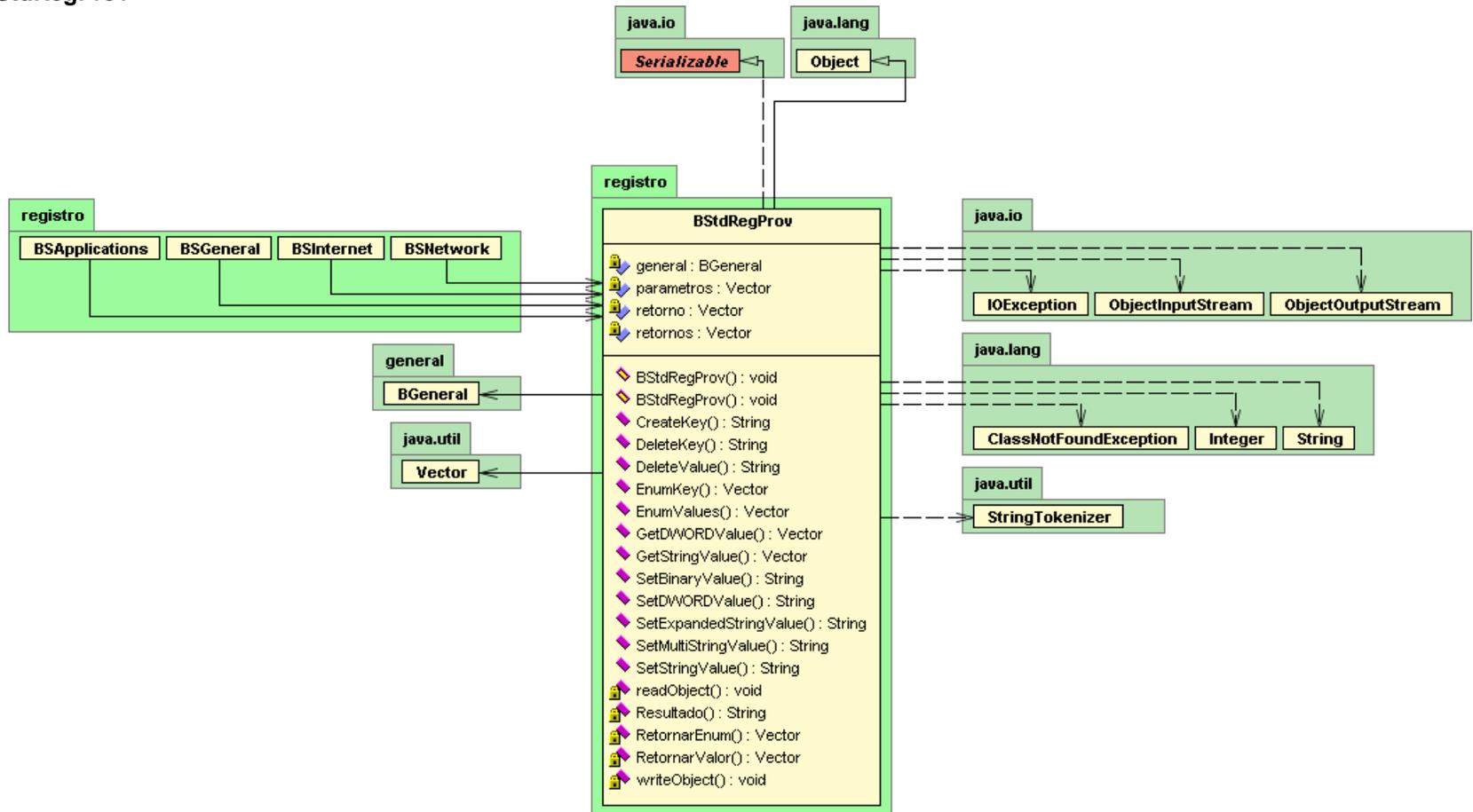
BSInternet



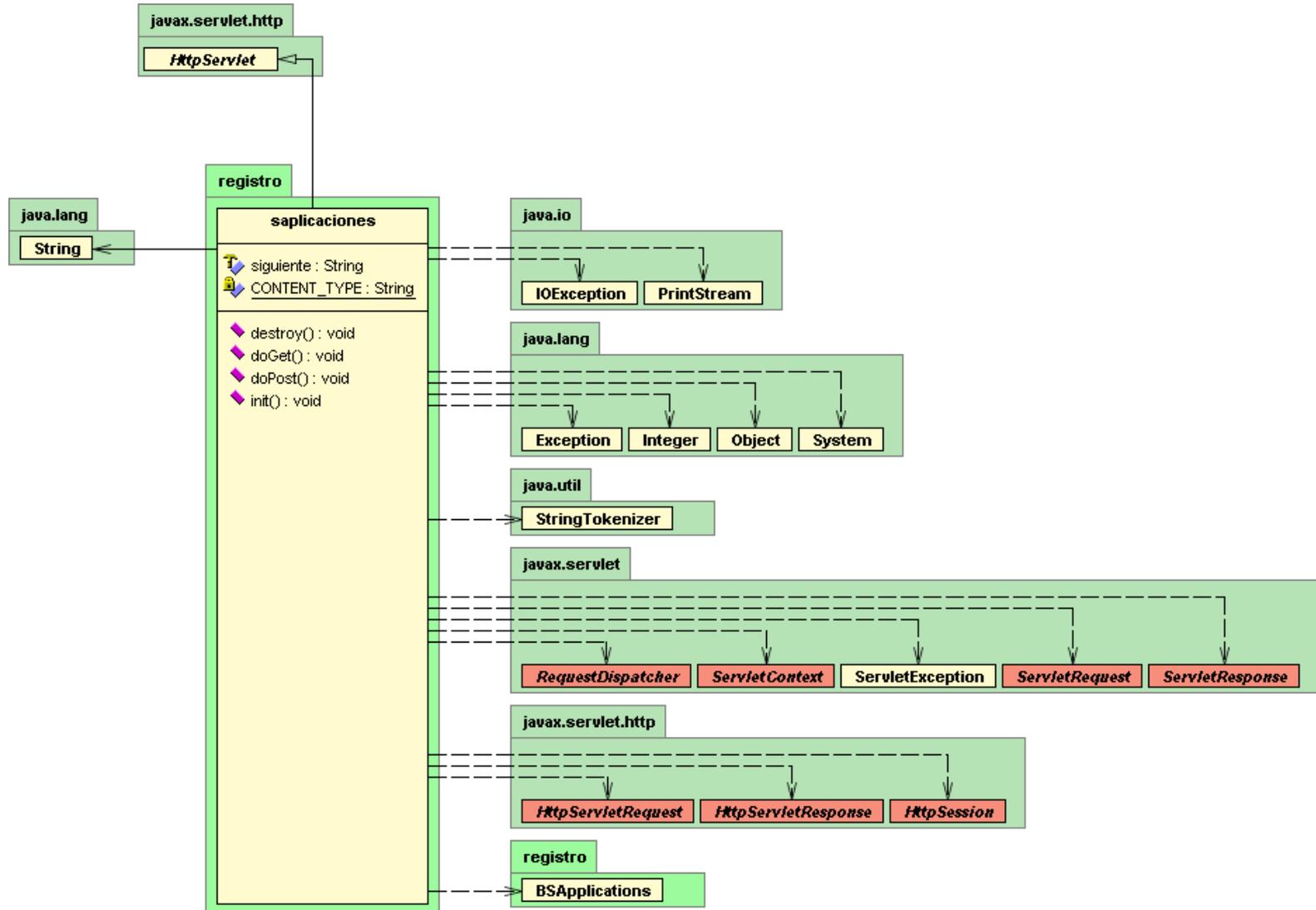
BSNetwork



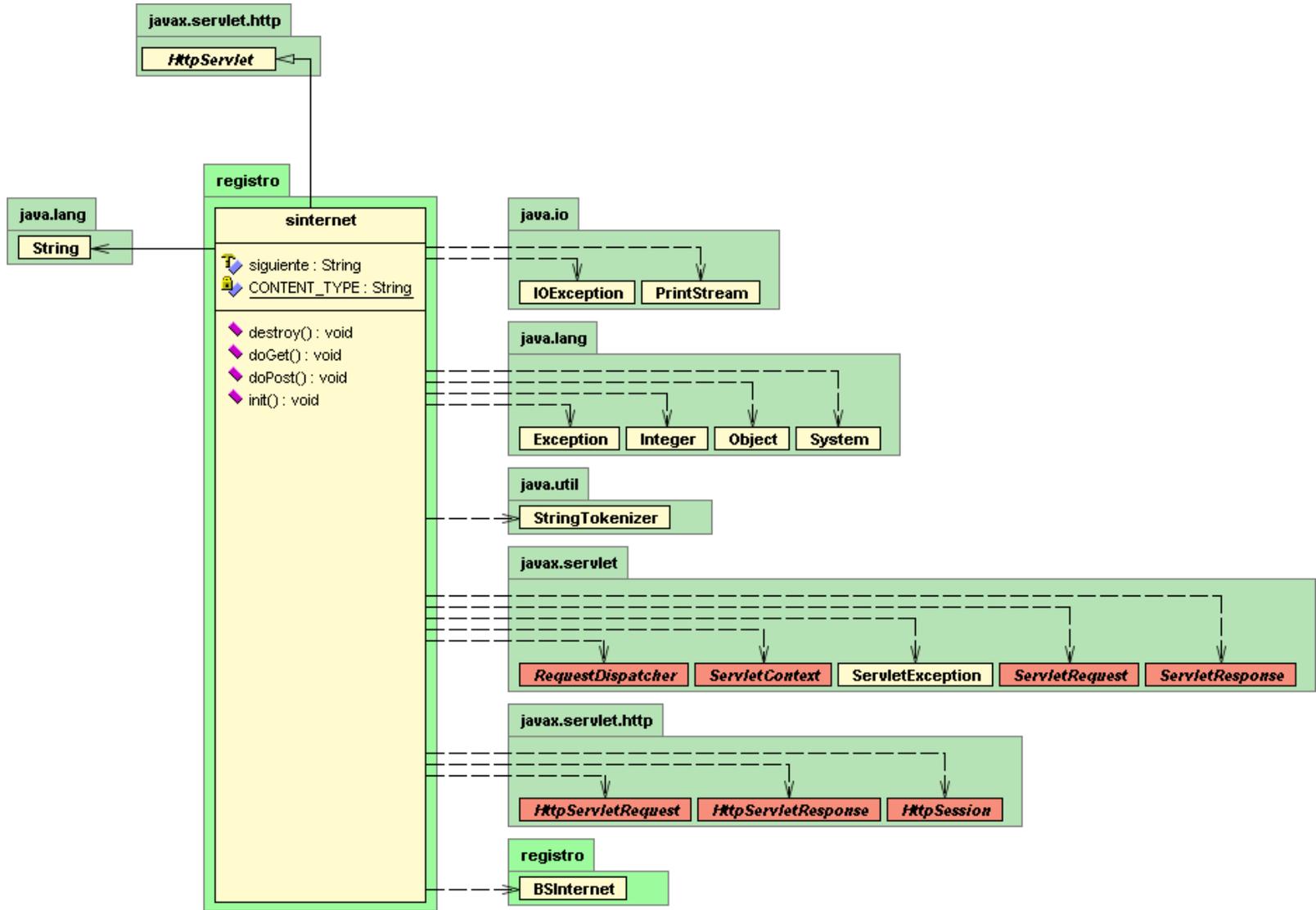
BStdRegProv



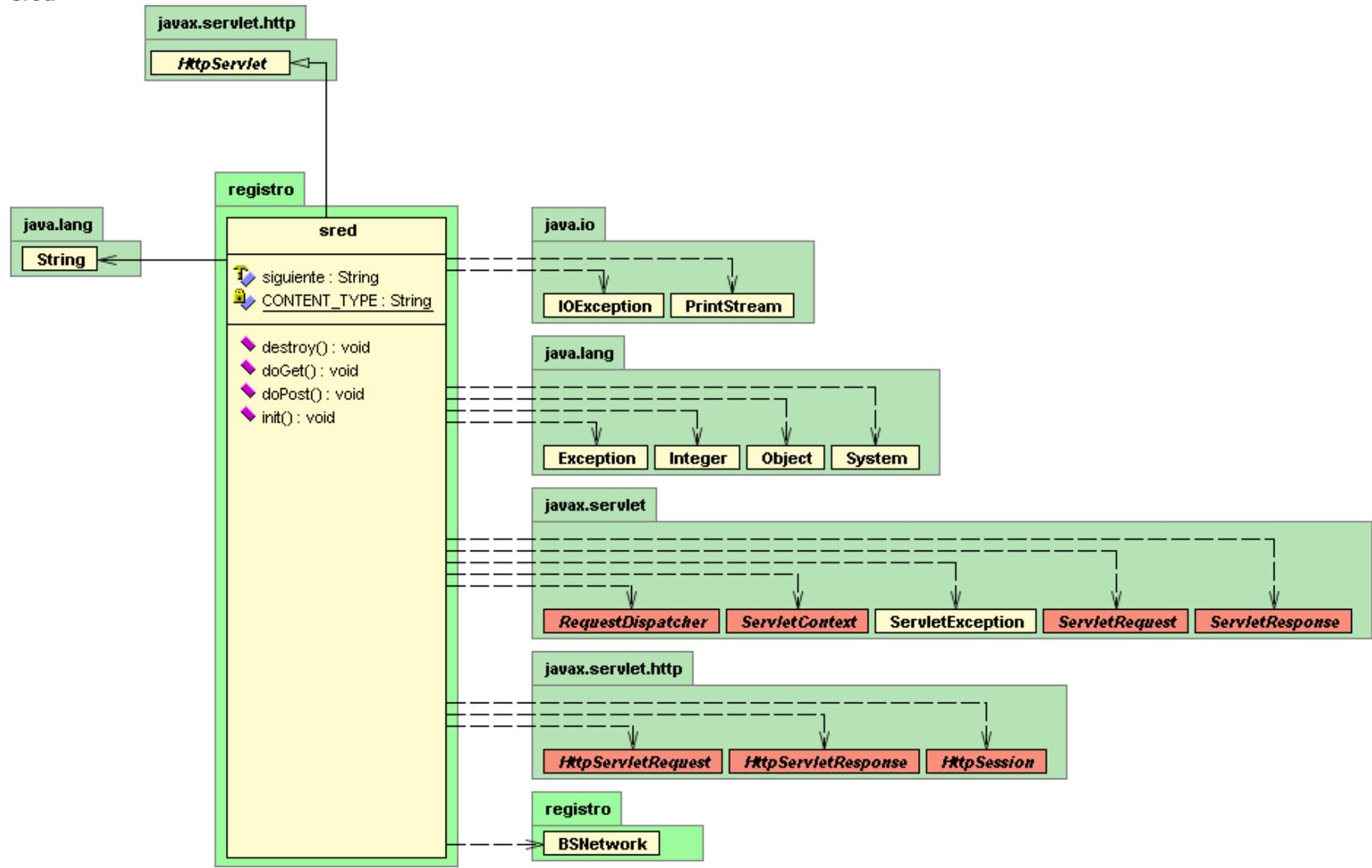
saplicaciones



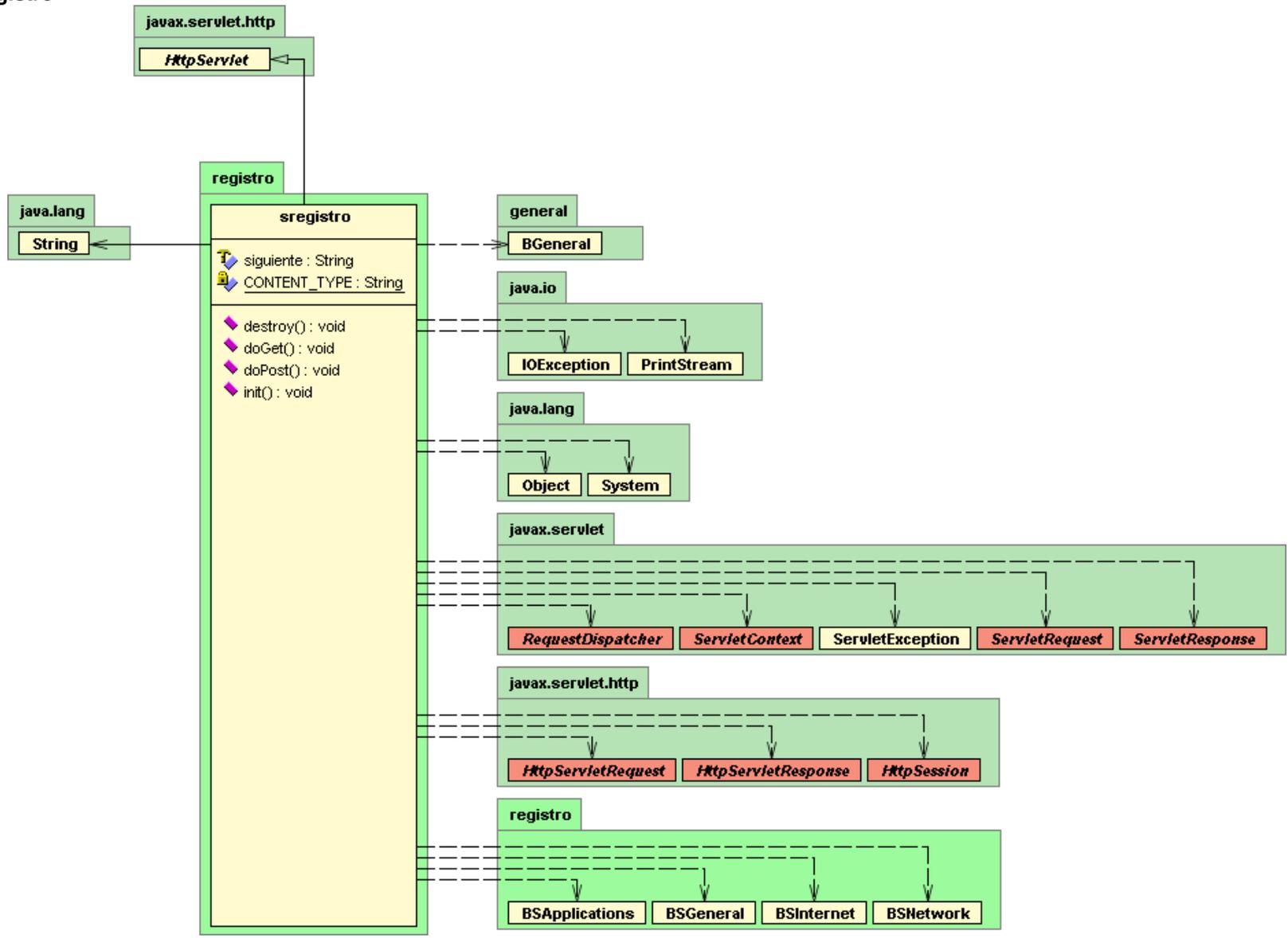
sinternet



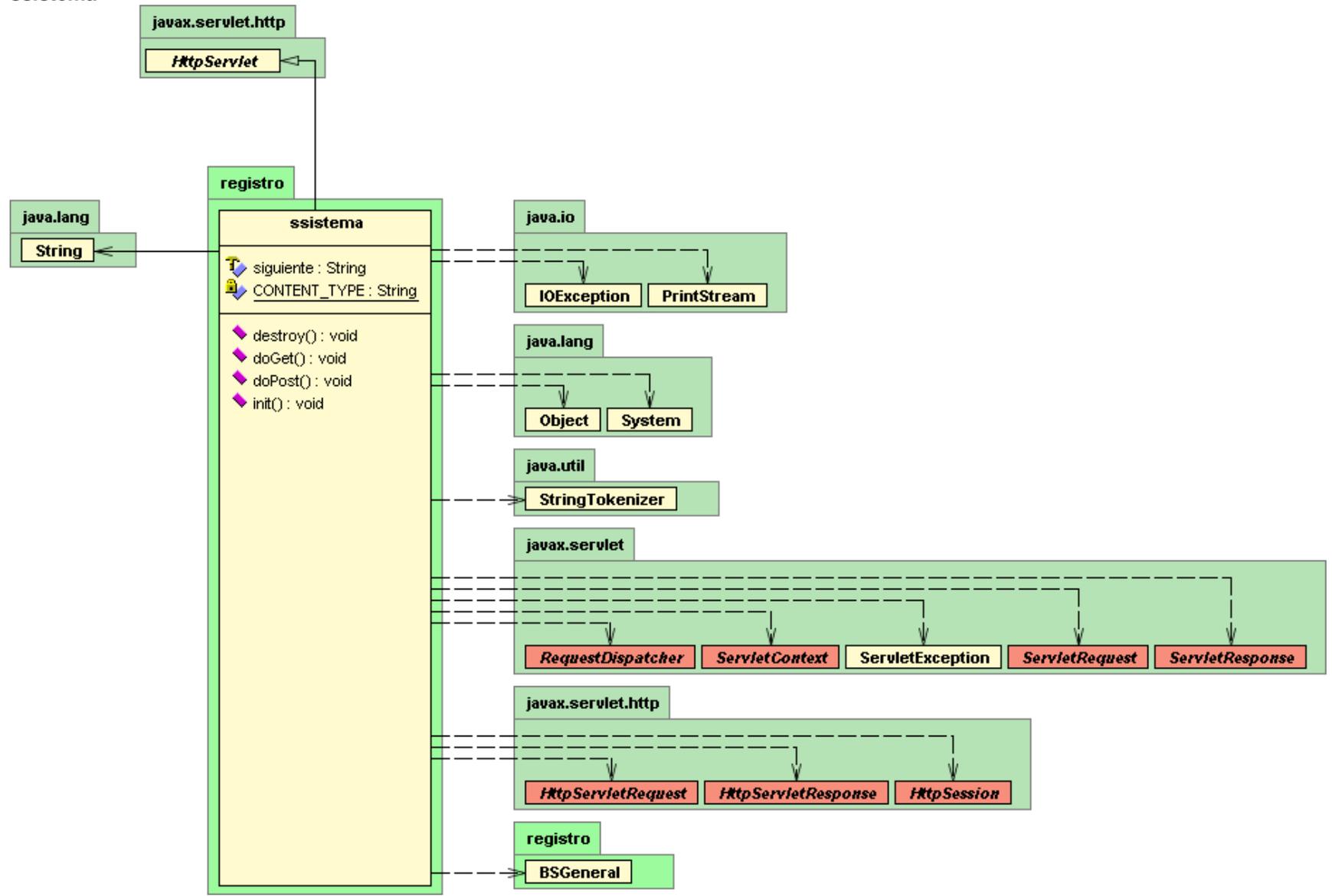
sred



sregistro



ssistema



ANEXO C

Instalación y configuración

El siguiente procedimiento solamente se debe realizar en el equipo que se utilizará para gestionar otros equipos.

J-Integra

En primer lugar es necesario descargar la versión de prueba de J-Integra de la página web de Intrinsyc (<http://www.intrinsyc.com>). En este caso se utiliza el archivo `jintegra155.zip` descargado en el mes de enero del 2003. En seguida se debe:

- ❖ Extraer `jintegra155.zip` al directorio donde se desea instalar J-Integra. Por ejemplo, en Windows, si se quiere realizar la instalación en `C:\jintegra`, el archivo se debe extraer en `C:\`. Esto crea los folders `C:\jintegra`, `C:\jintegra\lib`, entre otros.
- ❖ Adicionar `jintegra\bin` (`C:\jintegra\bin` si J-Integra se instaló en `C:\`) a la variable del sistema PATH (si se está utilizando Windows) y `jintegra\lib\jintegra.jar` (`C:\jintegra\lib\jintegra.jar` si J-Integra se instaló en `C:\`) o `jintegra/lib/jintegra.jar` en Unix, al CLASSPATH.

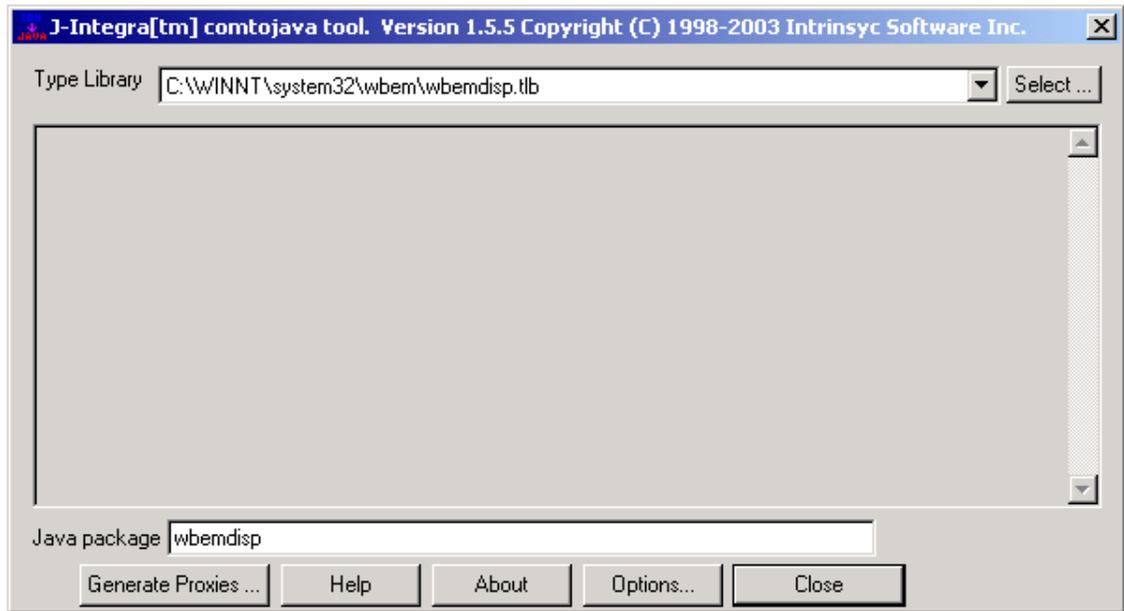
Al descargar la versión de prueba de J-Integra, Intrinsyc envía un e-mail que contiene la licencia de J-Integra (`jintegra.xml`). Con ese archivo se deben realizar el siguiente procedimiento.

- ❖ Salvar el archivo adjunto, `jintegra.xml`, en `<jintegra_install_dir>\lib`.
- ❖ En una ventana de comandos, cambiar el directorio a `<jintegra_install_dir>\lib` y ejecutar los siguientes comandos:
 - `java -cp jintegra.jar com.intrinsyc.license.JintegraLicenseTool clean`
 - `java -cp jintegra.jar com.intrinsyc.license.JintegraLicenseTool install`
- ❖ Si se está usando Outlook:
 - Doble click en el mensaje
 - Cuando el mensaje aparece, hacer click derecho en el archivo XML y seleccionar Save As.
 - Salvarlo al directorio `jintegra\lib`.

J-Integra y WMI

- ❖ En el prompt de comandos o en Explorador de Windows, ir al directorio `C:\jintegra\bin` y ejecutar `com2java` con `wbemdisp.tlb`, localizado en `c:\winnt\system32\wbem`, especificando "wbemdisp" como el nombre del paquete Java, como se muestra en la Figura 1.

Figura 1. Herramienta com2java de J-Integra



- ❖ En el prompt de comandos, ir al directorio C:\jintegra\bin y ejecutar setdllhost con wbemdisp.dll, localizada en c:\winnt\system32\wbem, de la siguiente manera:
 - setdllhost wbemdisp.dll C:\WINNT\system32\wbem

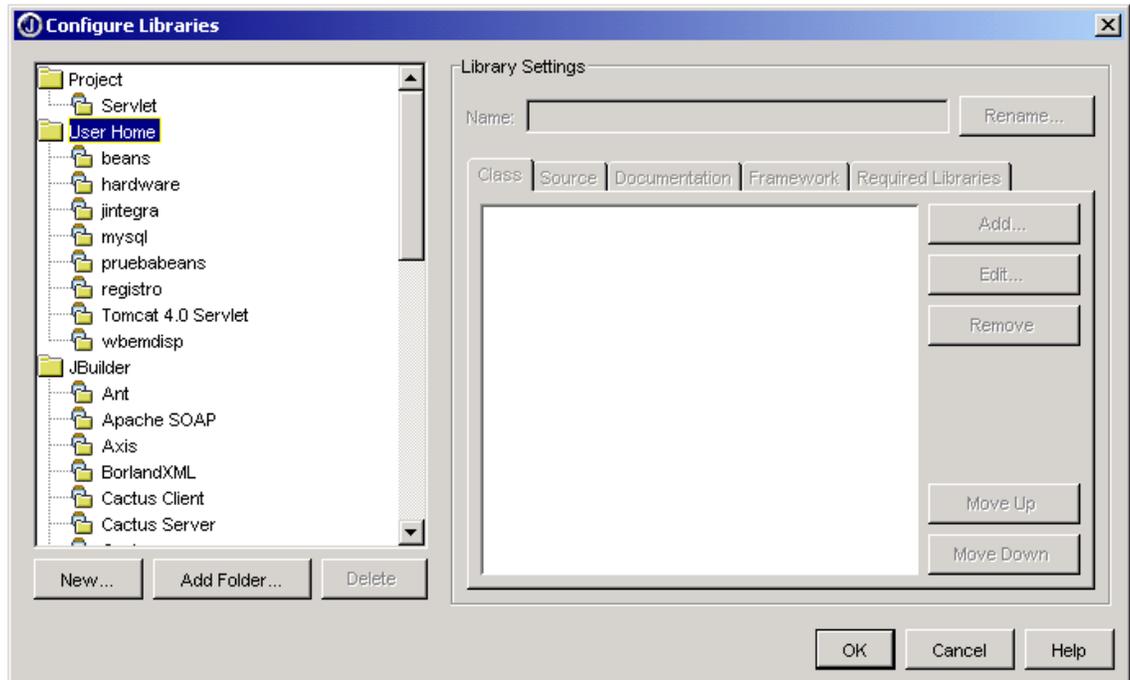
J-Integra, WMI y JBuilder

- ❖ Establecer la variable JAVA_HOME.
- ❖ Adicionar el paquete de Proxies, llamado wbemdisp, generado con la herramienta com2java de J-Integra, a la variable del sistema CLASSPATH.

En JBuilder:

- ❖ Ir a Tools, Configure libraries.
- ❖ Presionar New y crear jintegra con la ruta C:/jintegra/lib/jintegra.jar, en seguida crear wbemdisp con la ruta del paquete de Proxies, wbemdisp, generado con com2java. Como se muestra en la Figura 2.

Figura 2. Librerías de JBuilder



- ❖ Finalmente agregar al proyecto las librerías jintegra y wbemdisp definidas anteriormente.

Proveedor SNMP

En primer lugar se debe ir a Inicio/Configuración/Panel de control/Agregar o quitar componentes Windows e instalar Herramientas de Administración y supervisión como se muestra en la Figura 3.

Figura 3. Configuración SNMP



Además para que la herramienta funcione adecuadamente es necesaria la instalación del proveedor SNMP, `wmisnmp.exe`, que soporta la MIB II (RFC 1213).

- ❖ Se debe copiar `BRIDGE-MIB.my` al directorio `C:\WINNT\system32\wbem\SNMP`.
- ❖ En seguida en el prompt de comandos, ir al directorio `C:\WINNT\system32\wbem\SNMP` y ejecutar
 - `smi2smir /a BRIDGE-MIB.my`

En ese momento el CIMOM del equipo que gestionada tiene la MIB BRIDGE en su repositorio.

- ❖ Compilar el archivo mof de los switches utilizando
 - `mofcomp nombre del archivo.`

Archivo MOF de un switch

```
// Copyright © 1998-1999 Microsoft Corporation. All rights reserved.
```

```
#pragma autorecover
```

```
#pragma namespace("\\.\root")
```

```
instance of __Namespace
{
    Name = "snmp" ;
};
```

```
#pragma namespace("\\.\root\snmp")
```

```
[AgentAddress ( "10.1.13.11" ),
AgentReadCommunityName ( "public" )
]
```

**DIRECCIÓN IP
COMUNIDAD DE LECTURA**

```
instance of __Namespace
{
    Name = "switch11" ;
};
```

NOMBRE DE LA NAMESPACE DEL SWITCH

```
#pragma namespace("\\.\root\snmp\switch11")
NAMESPACE DEL SWITCH
```

(\\.\root\snmp\NOMBRE DE LA

```
instance of __Win32Provider as $PClass
{
    Name = "MS_SNMP_CLASS_PROVIDER";
    Clsid = "{70426720-F78F-11cf-9151-00AA00A4086C}";
};
```

```
instance of __ClassProviderRegistration
{
```

```
    Provider = $PClass;
    SupportsGet = TRUE;
    SupportsPut = FALSE;
    SupportsDelete = FALSE;
    SupportsEnumeration = TRUE;
```

```
    QuerySupportLevels = NULL ;
```

```
    ResultSetQueries = { "Select * From meta_class Where __this isa SnmpMacro" };
};
```

```
instance of __Win32Provider as $EventProv
```

```
{
    Name = "MS_SNMP_REFERENT_EVENT_PROVIDER";
    Clsid = "{9D5BED16-0765-11d1-AB2C-00C04FD9159E}";
};
```

```
instance of __EventProviderRegistration
```

```
{
    Provider = $EventProv;
    EventQueryList = {"select * from SnmpExtendedNotification"};
};
```

```
instance of __Win32Provider as $EncapEventProv
```

```
{
    Name = "MS_SNMP_ENCAPSULATED_EVENT_PROVIDER";
    Clsid = "{19C813AC-FEE7-11D0-AB22-00C04FD9159E}";
};
```

```
instance of __EventProviderRegistration
{
    Provider = $EncapEventProv;
    EventQueryList = {"select * from SnmpNotification"};
};
```

```
instance of __Win32Provider as $PInst
{
    Name = "MS_SNMP_INSTANCE_PROVIDER";
    Clsid = "{1F517A23-B29C-11cf-8C8D-00AA00A4086C}";
};
```

```
instance of __InstanceProviderRegistration
{
    Provider = $PInst;
    SupportsGet = TRUE;
    SupportsPut = TRUE;
    SupportsDelete = TRUE;
    SupportsEnumeration = TRUE;

    QuerySupportLevels = { "WQL:UnarySelect" };
};
```

