“Tecnología Cliente Servidor con Arquitectura Corba”

Sistema de una Empresa Telefónica

Rodrigo Berrezueta A.[1], Carlos Guzmán B[2], Carlos Valero D.[3]

[1]Ingeniero Eléctrico en Computación 2000
[2]Ingeniero Eléctrico en Computación 2000
[3]Director de Tópico. Ingeniero Eléctrico en Computación, Escuela Superior Politécnica del Litoral, 19 , Postgrado EEUU, Profesor de la Espol desde 19

RESUMEN

El proyecto desarrollado en este tópico, aplica la tecnología Cliente Servidor con una nueva forma de comunicación (Corba). Este proyecto consiste en un sistema de facturación para una empresa telefónica que hace uso de dos tecnologías Corba y Java.

Hacemos notar que el objetivo principal de este proyecto es mostrar el uso de Java y Corba como una nueva forma de desarrollo de aplicaciones distribuidas, luego demostrar como trabaja el sistema de facturación.

Los procesos intermedios entre Cliente y Servidor, que son necesarios para soportar las interacciones entre estos, se denomina Middleware. Pero existen varias formas de middleware, por lo que se intenta establecer un estándar en la forma de comunicación haciéndola orientada a objetos. Este middleware es CORBA [Common Object Request Broker Architecture], que es independiente de lenguajes y sistemas operativos.

Java es el lenguaje de programación que más se acopla a Corba, dado que Java está concebido para trabajar en redes amplias y a su independencia de la máquina donde se ejecutará su código.

Aunque nuestra aplicación trabaja con tecnología de internet este proyecto fue realizado principalmente para que se ejecute en una intranet debido a las opciones que ofrece. Sin embargo ciertas opciones del programa pueden ser ofrecidas para que se ejecuten en la red internet, como son las opciones de consulta y las opciones de recepción de pagos con ciertas modificaciones.

Introducción

En los comienzos de internet surgió una forma de interactuar entre el cliente y el servidor llamado CGI. Estos se ejecutan totalmente en el servidor tomando peticiones a manera de texto y devolviendo resultados al cliente en un formato Html.

Con el pasar del tiempo surgió un lenguaje de programación que permite desarrollar aplicaciones que se ejecutan en un browser, este lenguaje es Java. Java es un lenguaje de programación con todas las herramientas con las que un programador puede contar, enfocado principalmente a que las aplicaciones realizadas en Java se ejecute en la máquina cliente, esta nueva forma de interacción tiene un costo, ya que lleva un cierto tiempo en descargar código Java de la máquina servidora a la máquina cliente, además del tiempo de conversión de código Java en código que la máquina cliente entienda.

A pesar de que los clientes ya podían ejecutar verdaderas aplicaciones con Java, la interacción con el servidor seguía siendo obsoleta ya que no existía un protocolo desarrollado especialmente para una comunicación orientado a objetos entre el cliente y el servidor.

Ante esta situación surge CORBA una tecnología middleware con la cual se optimizan las aplicaciones, poniendo a disposición de los programadores una herramienta moderna para desarrollar aplicaciones cliente/servidor en sus distintos modelos.

Existen varios Middleware con sus propias especificaciones en el mercado como son: Dcom/ActiveX, RMI, Servlets, Corba, Sockets y HTTP/CGI. Entre los Middleware mencionados se destacan Corba y Dcom siendo este último respaldado por Microsoft

Se intenta establecer un estándar en la forma de comunicación cliente/servidor por medio de objetos, para esto se realizarón especificaciones sobre CORBA [Common Object Request Broker Architecture], dadas por la OMG (Object Management Group) grupo formado por mas de 800 empresas de computación.

Corba es una tecnología moderna que hace de bus de objetos para distintos lenguajes de programación.

Corba brinda independencia de lenguajes y sistemas operativos, todo esto por medio de su IDL [Interface Definition Languaje] que permitirá la escritura de código sin importar el Sistema Operativo o el lenguaje de programación en el que se usarán.

Dado que las aplicaciones están diseñadas con una orientación a objetos, estos son fáciles de mantener y actualizar, pues los objetos pueden ser modificados o creados de acuerdo a sus necesidades.

CONTENIDO

Objetivos

Nuestro objetivo es realizar la simulación de un sistema de facturación para una empresa telefónica en el que se muestre claramente la aplicación del lenguaje Java en conjunto con la arquitectura Corba.

Este sistema estaría ubicado en un Servidor Web de la empresa Telefónica, para permitir a los empleados realizar las transacciones básicas de las que debe disponer este sistema.

Nuestro sistema cliente/servidor mostraría su funcionalidad al poner la aplicación como una pagina Web y dar acceso de este servicio a los usuarios.

Pasos a Seguir.

Para el funcionamiento de este sistema se supondrá la existencia una red Intranet, formado por lo menos de un servidor y un cliente, en el cual un supuesto empleado de una empresa telefónica x, correrá la aplicación en un cliente Web.

Algunos autores sugieren hacer un análisis y diseño de objetos de todo el proyecto, pero desde la perspectiva de que estos objetos van a vivir en una red. Con este modelo los objetos vendrían a ser como componentes individuales y autosuficientes que se compilan y ejecutan por separado, teniendo una interface corba para cada objeto. Con este esquema un programador desarrolla los objetos como piezas individuales que se compilan y se hacen ejecutable por separado manteniendo una total independencia. Al tratarse de objetos individuales se podrán cambiar detalles internos de estos sin necesidad de que los objetos estén afectados por los cambios que se realicen en ellos. De la misma forma se podrán distribuir como piezas individuales a través de una red o en una misma máquina, sin la necesidad que el cliente que usa estos objetos sepa donde estos residen.

El otro esquema de desarrollo, y el que hemos escogido para desarrollar este proyecto consiste en la realización de un análisis y diseño orientado a objetos desde el punto de vista clásico, así todos los objetos que obtengamos de nuestro análisis y diseño de objetos existirán como tales desde la perspectiva de un programador solo en tiempo de compilación y los clientes se comunicarán con estos objetos por medio de una sola interface corba (en la interface van los métodos públicos que un cliente ve y que son con los que este va a interactuar). En resumen existirá solo una interface corba para todos los objetos que residirán en el servidor y no una interface corba para cada objeto.

Por medio de este esquema la interface corba levantará los objetos que hemos obtenido de nuestro análisis y diseño de objetos. Este esquema nos brindará las facilidades de modelar un sistema cliente/servidor three tier con servidores livianos.

CORBA

Cliente /Servidor al estilo Corba.

Lo que hace a Corba tan importante es que define su middleware con el potencial de soportar independencia de cualquier tecnología existente. En otras palabras corba utiliza objetos con el fin de unificar la metáfora de traer aplicaciones existentes al bus Corba. Y lo más importante es que las especificaciones del servicio están siempre separadas de la implementación.

Corba esta diseñado para tener componentes inteligentes que puedan descubrir e ínter operar con cada uno de los otros componentes en el bus ORB.

Corba le permite crear objetos ordinarios, y luego hacerlos transaccionales, seguros y persistentes en tiempo de ejecución. Permitiendo a los objetos pasar los parámetros, invocar su método y retornar los resultados.

IDL

EL secreto para el éxito de la OMG es crear especificaciones de Interfaces y no código. Las especificaciones que utiliza Corba, son escritas en la Interface de Lenguaje neutral, mejor conocido como IDL (Interface Definition Languaje). Los componentes escritos en IDL son portables a través de Lenguajes, herramientas, sistemas Operativos, y redes, debido a que el IDL provee independencia del sistema operativo y lenguaje de programación escogidos. El IDL de Corba es puramente declarativo, esto significa que no provee detalles de implementación.

El IDL puede ser escrito e invocado por cualquier lenguaje que provea enlazamiento Corba, así como C, C++, Ada, Smalltalk, COBOL ,Java y el resto de los lenguajes que se vayan acoplando a este estándar.

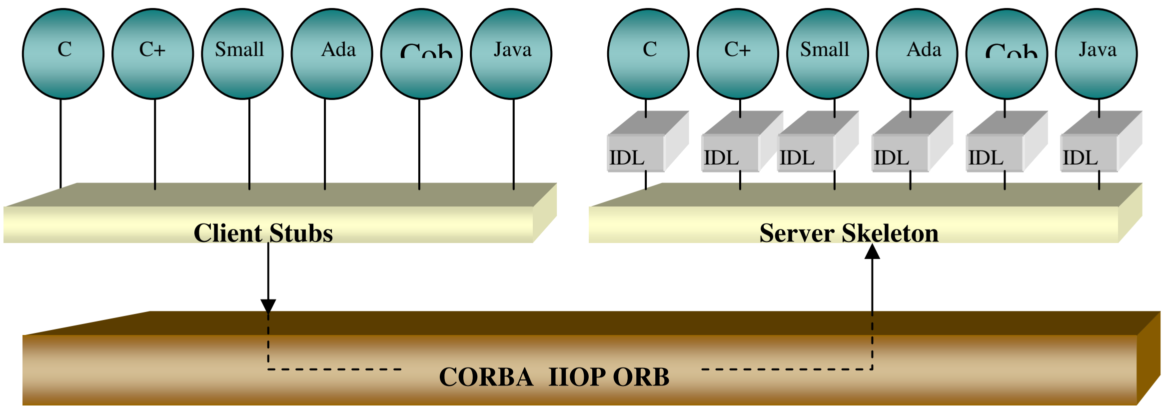

Figura 1 : Corba (IDL) sobre IIOP

ORB (Object Request Broker)

Es el bus de comunicación que permite a los objetos hacer requerimientos desde y hacia otros objetos localizados en forma local o remota, estableciendo la comunicación entre Clientes y Servidores por medio de Objetos.

El ORB es el middleware dentro de la arquitectura Corba, que nos permite comunicar a los Clientes con el Servidor y viceversa. El Cliente no necesita saber dónde reside el objeto, así como, no necesita saber como o en que lenguaje está implementado el objeto servidor. Solo necesita conocer la interface del objeto.

Protocolo (IIOP)

La implementación de ORB trabaja sobre un protocolo denominado Internet Inter ORB Protocol (IIOP), que es básicamente TCP/IP con mensajes de intercambio definidos por Corba, sirve como Protocolo Principal (Backbone). De esta manera cualquier propietario ORB puede conectarse con el Universo ORB haciendo requerimientos en el IIOP backbone.

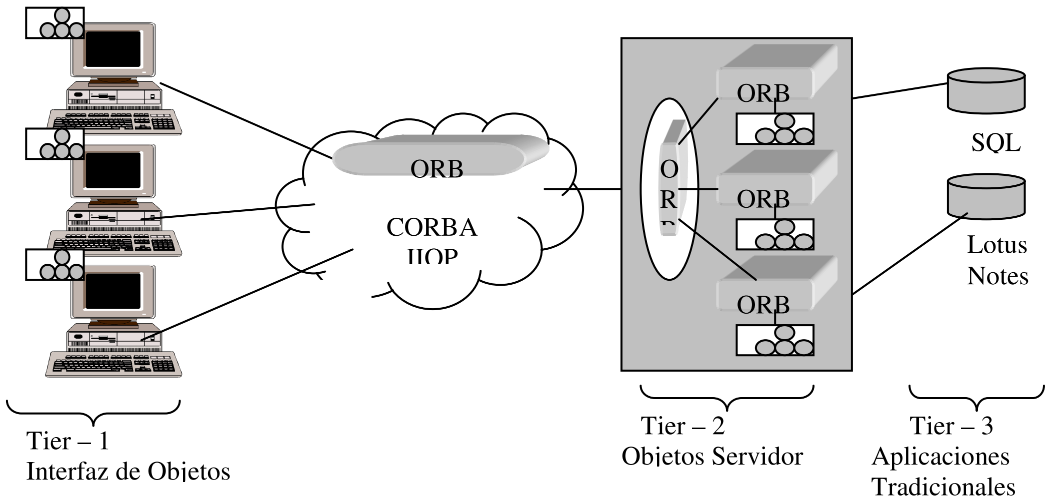

Figura 2: Funcionamiento de Corba Three Tier

ORB es mas sofisticado que otros middleware Cliente/Servidor incluyendo los tradicionales Remote Procedure Call (RPC), Message Oriented Middleware (MOM), Database Stored Procedure y Servicios Peer to Peer .

Applet/Java

El Applet es una pequeña aplicación pretendidamente corta basada en un formato gráfico sin representación independiente, que puede ser accesible desde un servidor

Internet, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento Web.

Los applet invocados por el cliente, viajan en la red en formato .class, al llegar a la máquina cliente, el browser interpreta este código y lo transforma a código binario para que el procesador pueda interpretarlo y ejecutarlo. Este paso que realiza el browser dependerá del sistema operativo así como del procesador. Por ejemplo en Win95/98/NT el browser por medio de su máquina virtual de Java convertirá el código en formato binario para que lo entienda el procesador Intel.

La creación de los Applets de Java fue el primer paso en la creación de un Objeto Web en el sistema Cliente/Servidor. Pero Java necesitaba ser complementado de una infraestructura de Objetos distribuidos, tal como es Corba.

Corba/Java

Corba se relaciona y enfatiza con transparencia en la red, debido a que provee una infraestructura de objetos distribuidos que le permite a las aplicaciones extender su alcance a través de redes, lenguajes y sistemas operativos.

Java se relaciona y enfatiza con la transparencia en la implementación de la Interface debido a que provee una infraestructura de objetos portables que trabaja en todos los Sistemas Operativos Principales.

Por lo que se puede decir, Java inicia donde Corba termina. Ambos, Corba y Java se complementan el uno al otro.

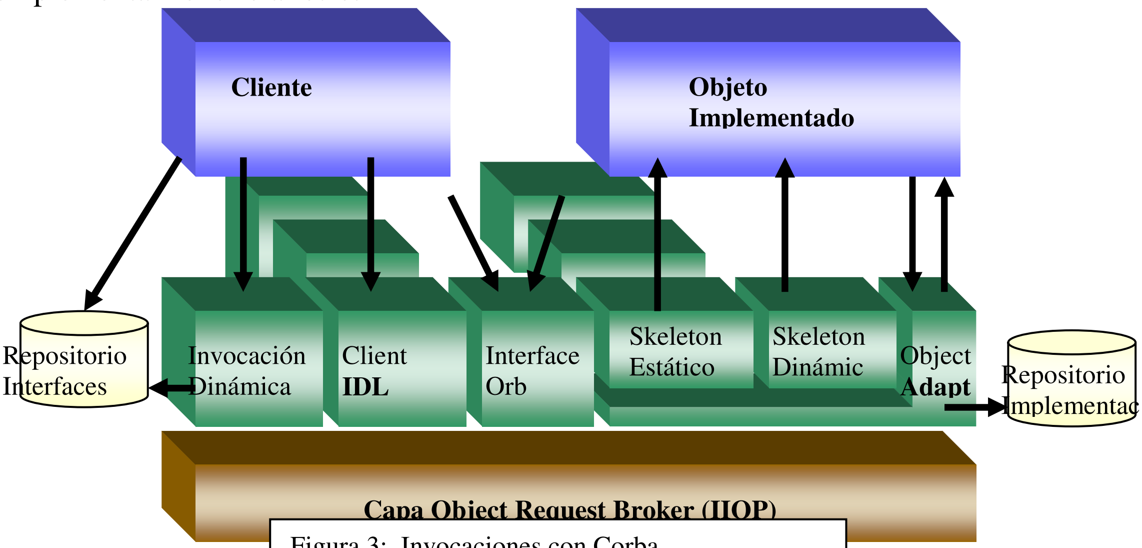

Figura 3: Invocaciones con Corba

Invocaciones estáticas

Una invocación Estática es igual a cualquier invocación a un método, pero por tratarse de una invocación remota con corba, este debe cumplir con lo siguiente:

Se define el IDL que será nuestra estructura para la creación de la interface con que podrá el cliente comunicarse con el objeto que se encuentra en el servidor.

Después de definir el IDL, sé lo pre compila produciendo 6 archivos principales, de los cuales 3 son para el cliente Stub, Helper, Holder, y 3 son para el servidor Skeleton , Interface. A cada uno de estos archivo .java se los compila produciendo los archivos.class.

Descripción del Proyecto

Se desea crear una aplicación para que sea accesible desde el Web, para lo cual se utilizará applets de Java. Los applets de Java podrán ser ejecutados desde distintas máquinas clientes.

El tema principal del proyecto es el desarrollo de un sistema cliente/servidor para una empresa telefónica usando tecnología de internet. Las máquinas clientes deberán contar con la existencia de browsers los cuales descargarán del servidor un applet de java en el cual se brindan las opciones de los distintos tipos de transacciones con los que contará el sistema.

Las disposiciones para realizar este proyecto son las de hacer uso del lenguaje de programación Java y el uso de Corba como middleware de objetos.

Funciones y Objetivos

El sistema de Facturación de la Empresa Telefónica tiene como objetivo:

- Permitir a los usuarios el manejo de las distintas opciones ofrecidas por el sistema.
- Mostrar un nuevo esquema de desarrollo de aplicaciones usando Corba y Java como tecnologías principales en la realización de proyectos que podrán ser ejecutados en una intranet o extranet y con la posibilidad de que ciertas opciones sean ejecutadas en la red internet.

Para poder cumplir con estos objetivos se dispone de las siguientes opciones:

- **Ingreso**. Esta transacción es para poder ingresar el consumo de un cliente a la Base de datos.
- **Consulta**. Es para mostrar al cliente cual es el consumo que ha tenido, mostrando totales de consumo, así como deudas pendientes y detalle de llamadas.
- **Recepción de Pagos**. Es para realizar el pago de lo adeudado por el servicio telefónico prestado. Este se lo puede hacer en efectivo o en cheque.
- **Detalle de Consumo**. Esta transacción nos muestra como su nombre lo indica, el detalle del Consumo, siendo local, nacional, regional, celular, internacional,
- **Asignación de Números**. Es para la asignación de números a nuevos abonados o clientes.
- **Baja de Números.** Es una transacción para retirar de su uso a usuarios, debido a problemas ya sea monetario o personal del cliente.

Arquitectura general de la Aplicación

Es una arquitectura basada en la Arquitectura Cliente/Servidor.
El usuario accesa por medio de una pagina HTML al Applet que trae consigo partes de Corba que permiten crear el puente ORB y así operar con objetos (Corba) en la red.

El servidor esta compuesto principalmente por un servidor Web, este servidor web almacenará a las aplicaciones cliente y servidor esperando a que un browser las descargue. Dado esto el cliente como el servidor se comunicarán con Corba sobre el protocolo IIOP.

La siguiente figura muestra como se realiza esta interacción cliente/servidor usando Objetos Corba.

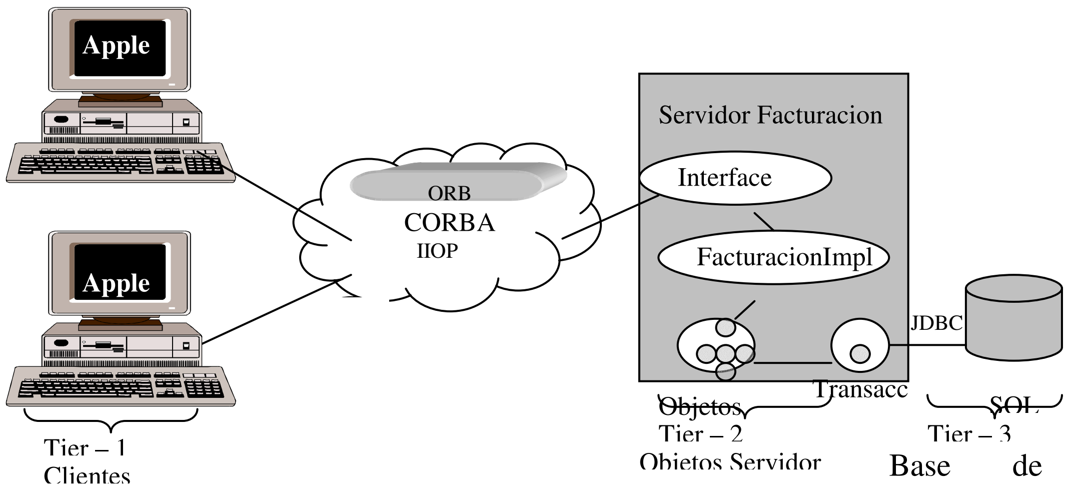

Figura 4: Aplicación vista en Three Tier

El funcionamiento de la aplicación.

Para poner en funcionamiento un sistema que haga uso de Corba y Java deberán seguirse pasos parecidos a los que nosotros hemos realizado para la puesta en funcionamiento de nuestro proyecto:
Se crea en el disco duro un Folder o carpeta donde se guarda:
- La pagina Html que llamará al applet, en nuestro caso la página Html se llama : " Principal.html"
- El Applet que correrá en el browser del cliente, en nuestro caso el applet se llama: "Principal.class"
- Las clases obtenidas en nuestro análisis y diseño orientado a objetos. Para nuestro sistema cada clase va guardada en un único archivo con extensión .class (formato de un archivo Java compilado) y cada archivo .class corresponde de manera directa a un objeto de los obtenidos en nuestro análisis y diseño orientado a objetos.
- Los archivos resultantes de la precompilación del Idl (Stub, skeleton, helper, holder, Impl, example) estos archivos son los que nos permiten tener el puente ORB.

Cuando se carga el ORB en el Servidor nos referimos a que el servidor muestra los métodos con los que dispone la aplicación, cuando se carga el ORB en el cliente nos referimos a que el cliente busca en el servidor, dentro del servicio de nombramiento los métodos con que dispone la aplicación.

Después de haber realizado la instalación previa se procede a la puesta en funcionamiento de los procesos que correrán en el servidor. Para esto se realizan los siguientes pasos:

- Se levanta el Os Agent. El Os Agent es un proceso que viene implementado con un fabricante de ORB y es el encargado de encontrar los objetos servidores Corba.
- Se levanta el Naming Service, que permite registrar el nombre de la interface de la aplicación con el servicio de nombramiento, quedando en el NameSpace la relación nombre a objeto.
- Se levanta el SystemmServer. El SystemmServer es una clase escrita en Java encargada de iniciar la aplicación, este se encargará de levantar nuestros objetos Corba y a su vez el objeto Corba Facturación se encargará de levantar a los objetos obtenidos en nuestro análisis y diseño de objetos.
- Se levanta el Gatekeeper. El Gatekeper también es un proceso que lo trae implementado el fabricante del Orb y se encarga de iniciar el IIOP el que permite la comunicación entre cliente y servidor. El Gatekeeper además sirve como servidor Web permitiendo a la máquina servidora interpretar y ejecutar las paginas Html.
- Finalmente se levanta el Browser, al cual indicamos la dirección del servidor y de la pagina que buscamos.

La pagina Html se encarga de buscar en el servidor el Applet, para mostrarlo en el Browser, pero primero este es bajado (copiado) al cliente, junto con los archivos necesarios como es el Stub, que nos permitira establecer la comunicación ORB.

El siguiente gráfico demuestra lo anteriormente expuesto.

Diagrama de funcionamiento.

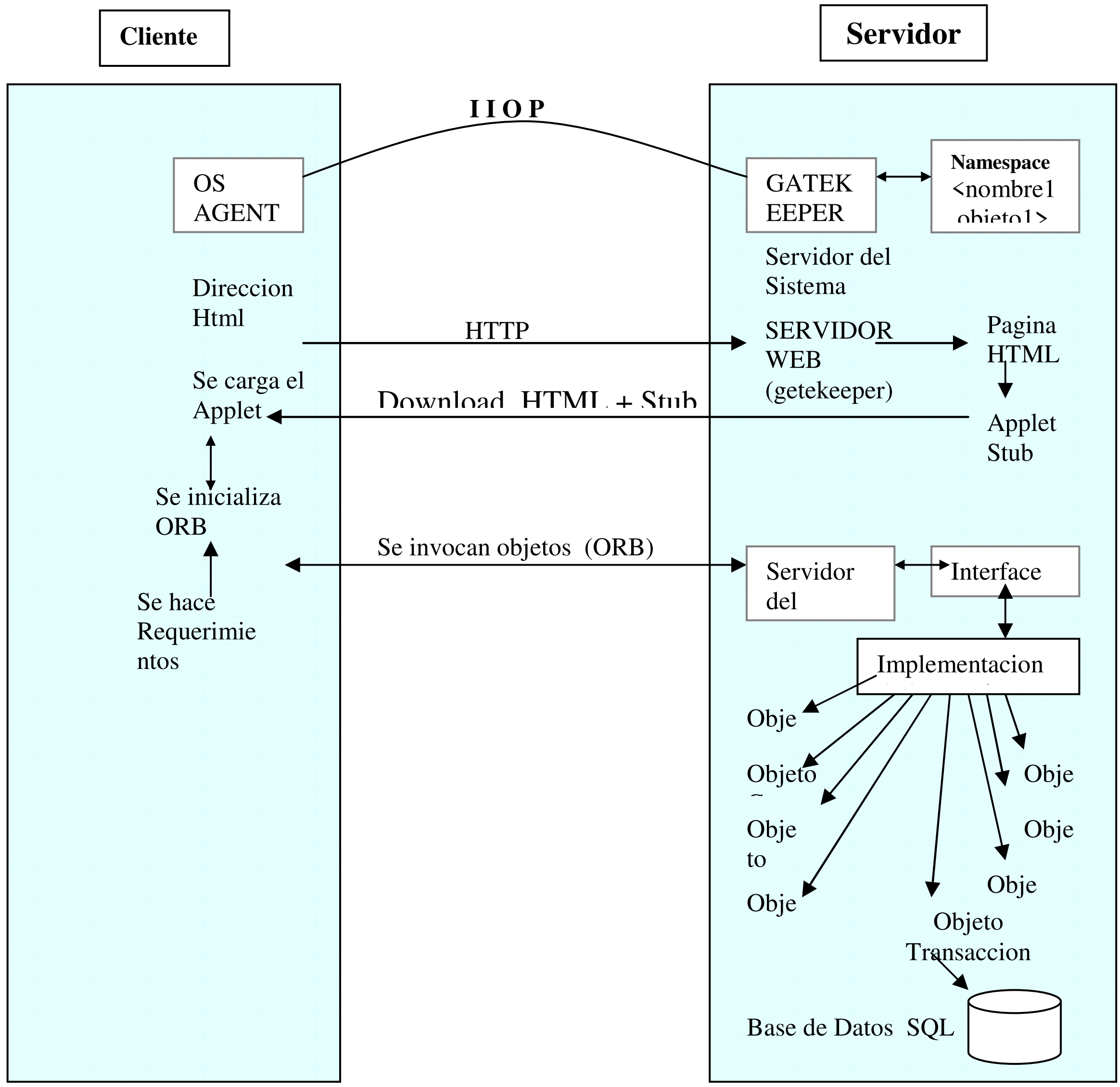

Figura: Funcionamiento de la aplicación Facturacion de Empresa Telefonica.

Referencias

a) Libros de Java

1. GRIFFITH & CHAN & F.ISAI. 1001 Tips para Programar con Java(1ra edición en español México 1998; Mc Graw Hill
2. JAMSA K. Ph. D. , Java Now, (Mc Graw Hill, New York)

b) Libros de Corba

3. ORFALI R. & HARKEY D., Client / Server Programing with JAVA and CORBA (2da Edicion, New York: John wiley & Sons, Inc, 1998)
4. ORFALI R. & HARKEY D & EDWARDS J. Essencial Client / Server Survival Guide (2da Edicion, New York: John wiley & Sons, Inc, 1998)