

**“Prototipo para la edición y almacenamiento remoto de
aplicaciones Java: OpenJavaDeveloper”**

CAMILO ANDRÉS RODRÍGUEZ VENEGAS
DANIEL EDUARDO RODRÍGUEZ VENEGAS

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD INGENIERIA
PROGRAMA TECNOLOGÍA EN INFORMÁTICA
CENTRO REGIONAL SOACHA
2008

“Prototipo para la edición y almacenamiento remoto de aplicaciones Java: OpenJavaDeveloper”

CAMILO ANDRÉS RODRÍGUEZ VENEGAS
DANIEL EDUARDO RODRÍGUEZ VENEGAS

*Trabajo de Grado para optar el título de:
Tecnólogos en Informática*

IGNACIO MONROY
Asesor Metodológico

MAURICIO BERMUDEZ
Asesor Tecnológico

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD INGENIERIA
PROGRAMA TECNOLOGÍA EN INFORMÁTICA
CENTRO REGIONAL SOACHA
2008

NOTA DE ACEPTACIÓN

Firma del presidente del jurado

Firma de jurado

Firma de jurado

Ciudad fecha (Día Mes Año)

TABLA DE CONTENIDO

	PAG.
0. INTRODUCION	5
1. DESCRIPCIÓN DEL PROBLEMA	7
2. OBJETIVOS	11
2.1 OBJETIVO GENERAL	11
2.2 OBJETIVOS ESPECÍFICOS	11
3. ESTADO DEL ARTE	13
4. MARCO CONCEPTUAL	15
5. MARCO REFERENCIAL	20
6. METODOLOGÍA	22
7. ANALISIS DEL SISTEMA	26
8. DESCRIPCION DETALLADA DEL SISTEMA	28
8.1 MODELO CONCEPTUAL DEL SISTEMA	28
8.2 MODELO CONCEPTUAL DEL SISTEMA PROPUESTO	29
8.3 DIAGRAMA DE SECUENCIA DEL SISTEMA ACTUAL	30
9. ESPECIFICACIONES TÉCNICAS	31
10. DIAGRAMA DE FLUJO DE DATOS	33
10.1 Diagrama de Contexto del Cliente	33
10.2 Diagrama de Contexto del Servidor	34
10.3 DICCIONARIO DE DATOS	35
10.4 DIAGRAMA ENTRADA SALIDA	36
11. DISEÑO DE CAPTURAS DE PANTALLA	37
12. ANALISIS DE RESULTADOS	41
CONCLUSIONES Y RECOMENDACIONES	
BIBLIOGRAFÍA	
ANEXOS	

INTRODUCCIÓN

El proyecto tuvo como finalidad facilitar el ambiente de desarrollo en los cuales se encuentra el estudiante en la universidad Minuto de Dios. Con la realización de un prototipo que ayude e incentive al usuario en la creación de soluciones de software en las clase de programación sobre lenguaje Java (J2SE), así mismo lograr la integración de los estudiantes en diversos proyectos que ayuden a mejorar los procesos investigativos de los mismos en las diversas temáticas de la clase. El aplicativo cuenta con las herramientas necesarias para la realización de soluciones óptimas y simples para los usuarios, a la vez se tuvo en cuenta los recursos que brindan los computadores para lograr un mejor uso de estos y evitar conflictos en la ejecución o elaboración de programas.

La facilidad y usabilidad en el mundo actual es importante ya que el tiempo es un factor valioso a la hora de realizar diversas actividades (trabajo, estudio, etc.) , por ende la implementación de esta solución permite a la Corporación Universitaria Minuto de Dios mejorar los procesos que empiezan los estudiantes en el área de programación sobre lenguaje Java, brindándoles la facilidad y comodidad para que el desempeño dentro de la institución sea lo más óptimo posible y dando más calidad al producto que se ofrece en esta.

El enfoque de desarrollo se lleva de forma tal que los usuarios finales del proyecto se sienten seguros de la utilización de la plataforma ya que

soluciona varios inconvenientes que se presentan actualmente como la perdida de actividades y de tiempo a la hora de empezar a trabajar en el área de programación dentro de la institución.

1. DESCRIPCIÓN DEL PROBLEMA

La universidad Minuto de Dios actualmente maneja las herramientas de desarrollo para programación en lenguaje Java en cada una de las maquinas que se encuentran distribuidas en 3 salas de informática, el trabajo se ve afectado seriamente por algunos inconvenientes que suelen ocurrir frecuentemente como lo son: desconfiguración de las maquinas, eliminación de las herramientas de desarrollo y eliminación de archivos de los estudiantes, siendo así, cuando los estudiantes se disponen a programar tienen que comprobar si todas estas herramientas se encuentran instaladas correctamente, y si habían dejado un archivo en el equipo lo mas probable es que este fuera eliminado por alguna persona.

Como el proceso es una de las bases en la universidad, que se maneja durante toda la carrera y no debería verse afectado por ningún motivo, es aquí donde nuestra aplicación satisface las necesidades de los estudiantes de informática, permitiéndoles llevar un proceso sin ningún tipo de percances, mejorando así la facilidad y el entusiasmo hacia la materia.

Anteriormente se plantearon varias soluciones a este problema, como lo fueron:

- *Congelar los equipos:*

- Se puede hacer uso de programas para congelar los discos duros, y que ningún cambio afecte al sistema en si, un ejemplo de este tipo de software es el *Deep Freeze*.

Deep Freeze:

El Deep Freeze es ampliamente considerado como el líder en la industria del software del tipo "reinicie y restaure". El principal objetivo de Deep Freeze es que, mientras esté instalado, la computadora estará "congelada". Cualquier cambio que se le realice a una computadora "congelada" será eliminado al reiniciar la PC, es decir que el equipo volverá al estado en que se activó la protección.

- *Inconvenientes al hacer uso de "congeladores" en equipos:*

La seguridad que tiene la mayoría de estos programas, en muchos casos no es la más adecuada, y esto presenta un gran inconveniente, porque si se tienen los conocimientos necesarios para vulnerar sistemas o aprovecharse de errores en la programación, se puede vulnerar el software fácilmente, aprovechándose de algún agujero o usando herramientas creadas por terceros.

- *Tarjetas recuperadoras de sistema*

- Estas tarjetas permiten recuperar el sistema a un punto anterior, o cada vez que se reinicie el equipo donde esta instalada, una tarjeta de este tipo muy reconocida es *ResQ card*.

ResQ Card:

Recovery Card es la última tecnología disponible en el mercado mundial para la recuperación de datos. La tarjeta ResQ Pro puede resolver todos estos problemas, sólo reiniciando su computador.

Los errores computacionales como Copiar / borrar / Agregar / Remover / Formatear, operaciones que pueden fácilmente destruir el sistema y el disco, son instantánea y automáticamente recuperados por un simple reinicio del sistema. La Tarjeta ResQ Pro, ayudará al usuario y administrador de sistemas con las tareas de recuperación (reinstalar una y otra vez el sistema operativo y aplicaciones de software) en el tiempo que se demora en reiniciar el equipo.

- Inconvenientes al hacer uso de tarjetas recuperadoras:

El principal inconveniente es el costo, existen en el mercado diversas gamas de tarjetas recuperadoras, pero para obtener un buen rendimiento y durabilidad, lo más recomendable es invertir en calidad y marcas de renombre, esto, conlleva a ver los altos precios que tienen estas tarjetas, y que para la institución seria una gran inversión si se tiene en cuenta que son 3 salas de

informática. Otro problema sería el tiempo que se necesita para hacer el pedido, búsqueda de fondos para realizar la inversión, todo esto podría llevar más de 1 año.

Como se puede observar, las soluciones no son del todo viables ya sea por costos, seguridad etc., nuestro proyecto busca solventar todas estas dificultades, al hacer uso de un sistema operativo como es GNU/Linux, se reducen los costos principalmente por las licencias y la ideología sobre la cual esta basada, y brinda mucha más seguridad para la institución y para el trabajo de los estudiantes, también el uso de Java y MySQL para el desarrollo del prototipo permite que la solución tenga un bajo costo y que la calidad no va a ser inferior.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un editor de archivos para soluciones de software desarrolladas en lenguaje Java (J2SE) en plataformas GNU/Linux y Windows, con control de usuarios en la Corporación Universitaria Minuto de Dios, realizando algunos procesos de forma remota.

2.2 OBJETIVOS ESPECÍFICOS

- Hacer un análisis específico de cada parte del sistema (subsistema), teniendo en cuenta las prioridades y la esencia de cada uno.
- Realizar investigación bibliográfica sobre las diferentes tecnologías y herramientas disponibles sobre las cuales se basa la solución, teniendo en cuenta aspectos como la licencia de cada lenguaje y de cada herramienta de software.
- Definir las herramientas (IDE, OS y demás) sobre las cuales se desarrolla la aplicación.
- Identificar las principales funciones de las herramientas y realizar pruebas con ellas, hasta lograr dominar lo esencial y así desarrollar un trabajo óptimo.

- Emplear las técnicas de ingeniería de software para el desarrollo de la aplicación, teniendo en cuenta desde el análisis hasta la realización de las pruebas sobre el producto final.

3. ESTADO DEL ARTE

En búsqueda de soluciones similares u aproximaciones al proyecto "openJavaDeveloper" hubo una en particular que es muy interesante, ya que aplica una solución aceptable al problema de las plataformas y herramientas en los equipos para el lenguaje C / C++, esta es "WebDevStudio".

WEBDEVSTUDIO (IDE Online)

Desarrollado por:

- Alberto Martínez Fernández
- Israel Ortiz Florez

*** Descripción:**

Actualmente existe un elevado número de empresas dedicadas al desarrollo de proyectos software que requieren de un potente entorno para poder implementarlas. Un entorno de desarrollo integrado (IDE) consiste básicamente en un software que previamente ha sido instalado en la máquina del cliente y cuyo principal objetivo es el desarrollo de otro software. Sin embargo, todos los entornos que se suministran en la actualidad presentan el grave inconveniente de que son dependientes de

la máquina del cliente hasta tal punto de que un software solamente se puede llegar a implementar con un entorno en máquinas con idénticos sistemas operativos. Una herramienta capaz de abstraerse totalmente del cliente permitiría el mantenimiento de proyectos bajo cualquier plataforma sin necesidad de realizar ninguna instalación.

webDevStudio es un entorno de desarrollo integrado (IDE) online basado en la tecnología PHP. Como cualquier otro entorno permite mantener proyectos informáticos implementados en diferentes lenguajes de programación así como realizar una serie de operaciones básicas sobre ellos. La principal diferencia con respecto al resto de entornos reside en que es totalmente independiente de la máquina desde donde ejecute el cliente. El usuario no deberá tener instalado en su máquina nada mas que un navegador Web que le permitirá conectarse directamente al servidor donde se encuentra nuestra aplicación.

4. MARCO CONCEPTUAL

ARQUITECTURA CLIENTE SERVIDOR

¿Que es un cliente?

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

¿Que es un servidor?

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.

Elementos de la arquitectura cliente servidor:

En esta aproximación, y con el objetivo de definir y delimitar el modelo de referencia de una arquitectura Cliente/Servidor, debemos identificar los componentes que permitan articular dicha arquitectura.

[En Línea]<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor.shtml>

Considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- Presentación/Captación de Información
- Procesos
- Almacenamiento de la información

Características del modelo cliente/servidor:

En el modelo cliente/servidor podemos encontrar las siguientes características:

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.

7. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

8. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

REDES DE ÁREA LOCAL (LAN)²

Las LAN constan de los siguientes componentes:

- Computadores
- Tarjetas de interfaz de red
- Dispositivos periféricos

Las LAN permiten a las empresas aplicar tecnología informática para compartir localmente archivos e impresoras de manera eficiente, y posibilitar las comunicaciones internas. Un buen ejemplo de esta tecnología es el correo electrónico. Lo que hacen es conectar los datos, las comunicaciones locales y los equipos informáticos.

Algunas de las tecnologías comunes de LAN son:

- Ethernet
- Token Ring
- FDDI

²En Línea] <http://www.monografias.com/trabajos30/redes-de-datos/redes-de-datos.shtml>

Las WAN interconectan las LAN, que a su vez proporcionan acceso a los computadores o a los servidores de archivos ubicados en otros lugares. Como las WAN conectan redes de usuarios dentro de un área geográfica extensa, permiten que las empresas se comuniquen entre sí a través de grandes distancias. Las WAN permiten que los computadores, impresoras y otros dispositivos de una LAN compartan y sean compartidas por redes en sitios distantes. Las WAN proporcionan comunicaciones instantáneas a través de zonas geográficas extensas. El software de colaboración brinda acceso a información en tiempo real y recursos que permiten realizar reuniones entre personas separadas por largas distancias, en lugar de hacerlas en persona. Networking de área amplia también dio lugar a una nueva clase de trabajadores, los empleados a distancia, que no tienen que salir de sus hogares para ir a trabajar.

5. MARCO REFERENCIAL

CORPORACION UNIVERSITARIA MINUTO DE DIOS CENTRO REGIONAL SOACHA

La universidad Minuto de Dios regional soacha, ofrece diversos tipos de carreras tecnológicas, como son:

- Tecnología en Informática
- Tecnología en Electrónica
- Tecnología en Costos y auditoria
- Tecnología en Logística
- Tecnología en Comunicación gráfica

La Universidad Minuto de Dios, como cualquier organización establecida, tiene una estructura organizacional, que tiene los siguientes cargos:

- Rector General
- Vicerrector Académico
- Vicerrector Administrativo-Financiero
- Vicerrector de Sedes Regionales
- Rector Seccional Bello

- Secretario General
- Decano Facultad Ciencias de la comunicación
- Decano Facultad Educación
- Decano Facultad Ciencias Económicas y Administrativas
- Decano Facultad Ingeniería
- Decano Facultad Ciencias Humanas y Sociales
- Decano Admisiones y Registro
- Decano Desarrollo y Bienestar Estudiantil
- Decano Instituto Bíblico Pastoral Latinoamericano
- Director Centro Regional Soacha
- Coordinador Tecnología En Informática
- Coordinador Tecnología en logística
- Coordinador Tecnología en costos y auditoria
- Coordinador Tecnología en comunicación grafica
- Coordinador Tecnología en electrónica

6. METODOLOGÍA

El proyecto se desarrollo gracias a las observaciones que fueron realizadas durante la estancia en la universidad Minuto de Dios, exactamente en las clases de programación Java.

La metodología del presente proyecto es de tipo cuantitativo porque se hizo uso de recolección de datos y se partió de la creación de graficas estadísticas a partir de los datos obtenidos mediante diferentes pruebas como lo fueron: tiempo de reinstalación de herramientas, inspección de componentes instalados en la maquina. También es cualitativa porque se hizo parte del trabajo haciendo uso de la razón humana y en base a diversos sucesos que permiten que el trabajo del ser humano sea más cómodo y agradable, del mismo modo evaluando las conveniencias de los estudiantes a la hora de obtener un servicio como el propuesto.

El método usado para la realización de la solución fue el siguiente:

- Se evaluó todo el sistema, dividiendo el problema en diferentes procesos y a su vez en subprocesos, de este modo la aplicación quedó totalmente desglosada, posteriormente se realizo la identificación de cada proceso en su totalidad hasta lograr un análisis adecuado que permitió ver el sistema en su totalidad, abarcando cada uno de los aspectos de forma óptima.

- A cada proceso se aplico una prioridad, teniendo en cuenta siempre el proceso más complejo.
- Investigación sobre redes, y sistemas cliente – servidor, que ayudaron a fundamentar la aplicación.
- La búsqueda de información sobre diferentes herramientas de desarrollo fue importante, porque abrió campo a la iniciación del proceso de desarrollo, las herramientas usadas fueron: Eclipse Europa versión 3.3.1, compilador JAVAC de Sun Microsystems.
- El diseño grafico de la aplicación fue un factor primordial, se baso en los modelos actuales de algunos IDE, para así lograr conformidad y usabilidad para el usuario.
- La selección del lenguaje se hizo a partir de la evaluación de diversos aspectos (POO, API, funcional en aplicaciones cliente – servidor, entre otros), un lenguaje que cumplía estos requisitos era: Java, en su versión estándar (J2SE), aprovechando la JVM se logra la ejecución en los OS como: Windows XP/2000 y en distribuciones GNU/Linux como: Debian y Slackware.
- Una vez seleccionado el lenguaje, se estudio parte de la API durante el desarrollo de la aplicación, lo que logro minimizar el código gracias a la funcionalidad de los métodos propios del lenguaje (Java).

- La construcción del software teniendo los modelos realizados con anterioridad, permitió el desarrollo integral de la aplicación, salvaguardando las dificultades como cualquier proyecto, pero que permitieron llegar a un punto final satisfactorio.
- Se probó la aplicación en diferentes OS en busca de posibles errores.
- Con una última revisión se dio por terminada la solución de software.

7. ANALISIS DEL SISTEMA

7.1 Descripción general del sistema

7.1.1 DESCRIPCION DEL SISTEMA ACTUAL:

El sistema actual permite que los usuarios de la Universidad Minuto de Dios puedan trabajar en cada una de las maquinas que se encuentran en las 3 salas de computo de la institución con el lenguaje Java (J2SE).

Como primera medida para poder programar, las maquinas deben tener instalada las herramientas de desarrollo como son: J2SDK (Java SE Development Kit) y alguna IDE, o editor de texto, si no esta alguna de estas herramientas, hay la necesidad de instalarlas y configurarlas para poder usarlas adecuadamente, posteriormente el estudiante puede empezar a programar y debe guardar los archivos de trabajo en un medio extraíble para conservar los datos y la integridad de los mismos.

La solución de software OpenJavaDeveloper tiende a disminuir el tiempo en que los usuarios se disponen a trabajar/programar, con tan solo ejecutar el IDE "OpenJavaDeveloper" y loguearse al servidor puede empezar a desarrollar programas con este editor, este editor da la opción de guardar el archivo en el servidor, al hacer esto la integridad del archivo se mantiene y no puede ser eliminada por un usuario fuera del dueño y/o administrador.

Al ser una aplicación de tamaño considerable, se optó por usar la OOP (Object Oriented Programming) para así conseguir una estructura modular, sostenible y con facilidad en el mantenimiento. El lenguaje apropiado para este tipo de sistema es Java, con el uso de este, la aplicación se transformó en algo verdaderamente estable y seguro.

Haciendo uso de la Ingeniería de Software se pudo realizar la aplicación siguiendo las siguientes etapas.

Análisis de requisitos:

Obtener los requisitos que la solución necesitaba fue el primer paso para su posterior desarrollo. Un buen análisis de requisitos ayuda mucho a agilizar el proceso de desarrollo.

Especificación:

Se describió lo mejor posible todo el sistema, para lograr un sistema acorde a las necesidades.

Diseño y arquitectura:

Se planteó la forma como se iba a estructurar el sistema final, realizando un diseño del mismo con todas sus variables, del mismo modo planteando los agentes externos como la red en la que sería productiva la solución.

Programación:

Con los diseños de la aplicación ya desarrollados, se siguió con la línea de programación, siguiendo puntos importantes como la arquitectura entre otros.

Prueba:

A la solución se le realizaron varias pruebas de estabilidad y control, en busca de posibles errores (bugs) que logran hacer inestable la aplicación.

Documentación:

Se prepararon los manuales técnicos y de usuario para hacer entendible el sistema, y que no se pierda la idea en el tiempo.

Mantenimiento:

El mantenimiento al ser realizada la aplicación en su totalidad usando la OOP facilita mucho el entendimiento y facilidad de actualización de módulos.

8. DESCRIPCION DETALLADA DEL SISTEMA

8.1 MODELO CONCEPTUAL DEL SISTEMA

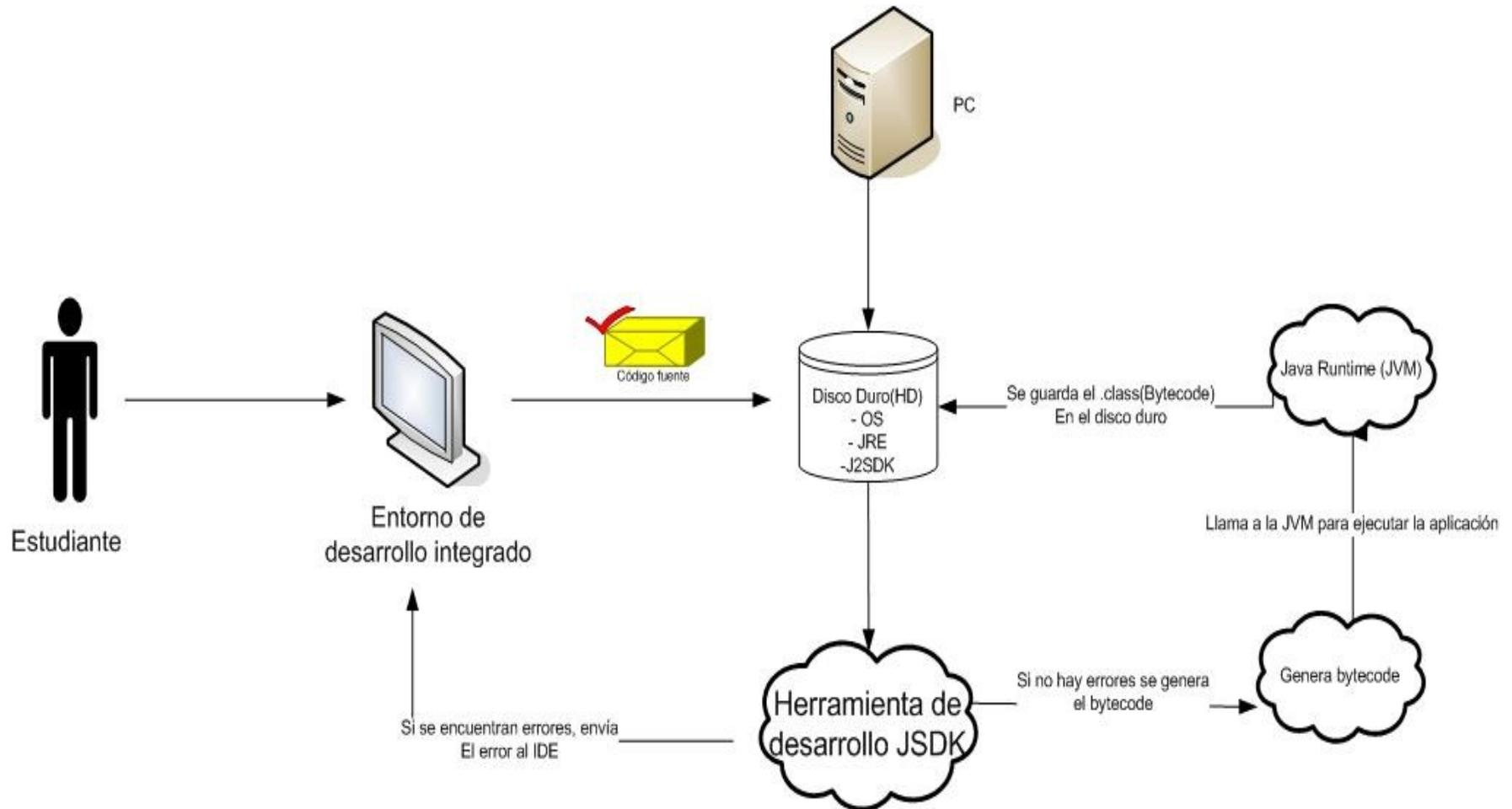


Figura 1. Casos de uso sistema actual

8.2 MODELO CONCEPTUAL DEL SISTEMA PROPUESTO

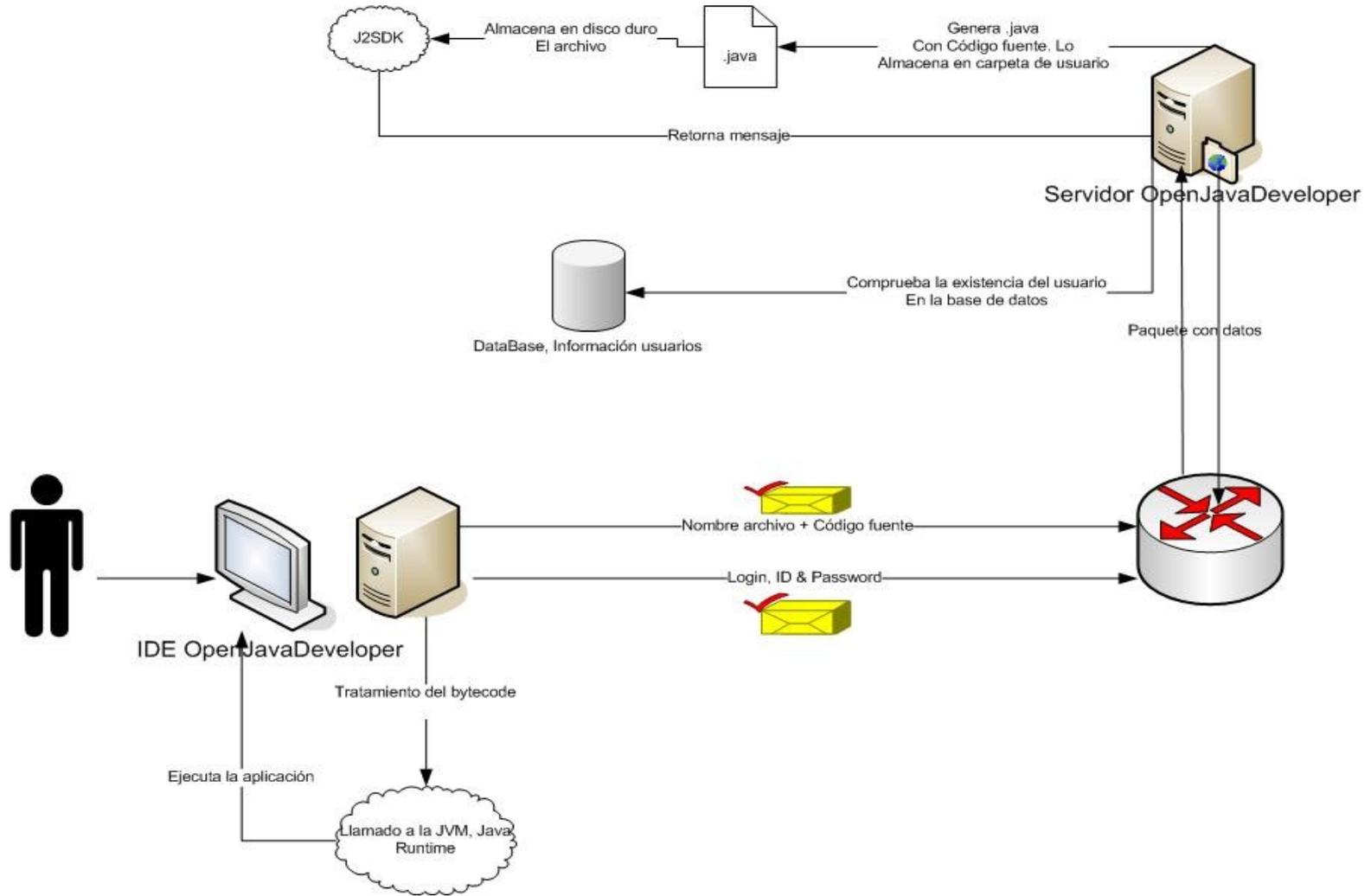


Figura 2. Casos de uso sistema propuesto

8.3 DIAGRAMA DE SECUENCIA DEL SISTEMA ACTUAL

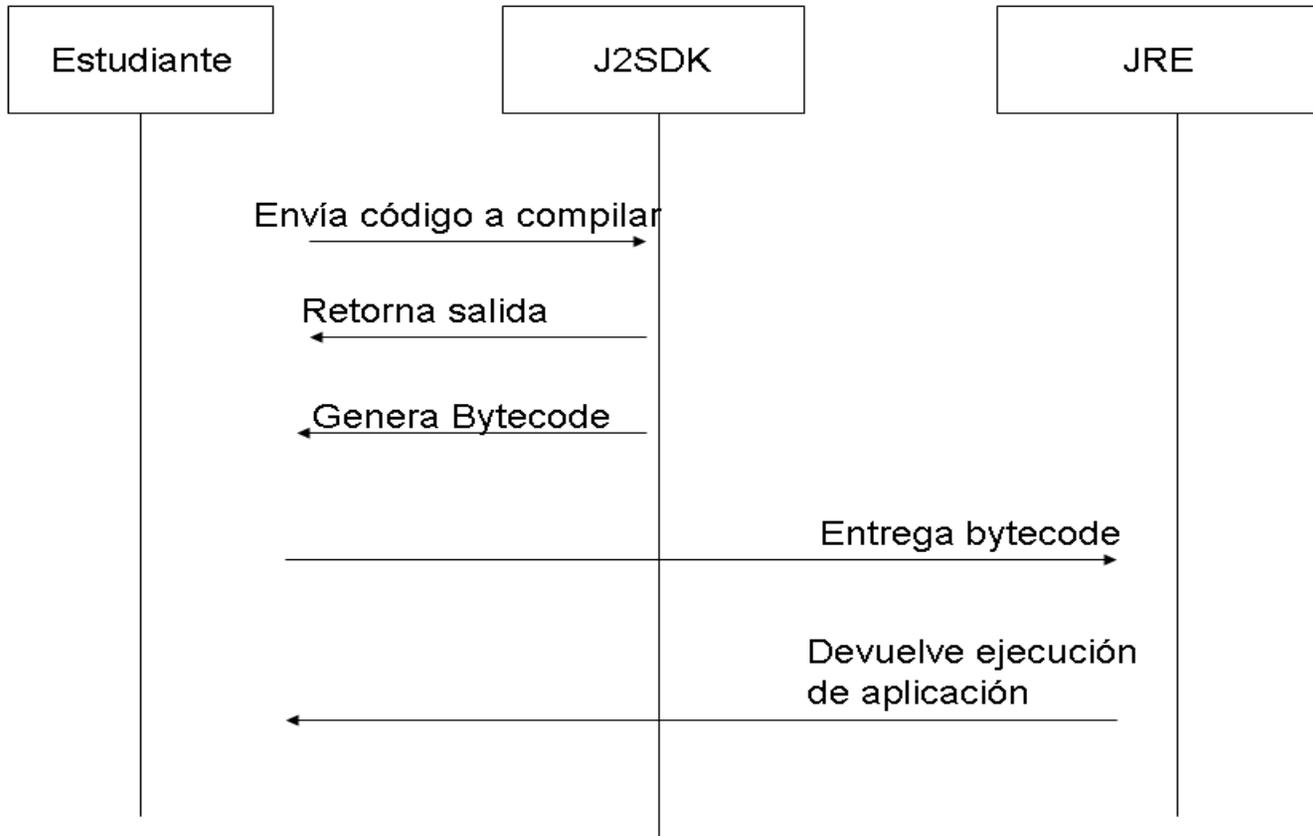


Figura 3. Diagrama de secuencias del sistema actual

9. ESPECIFICACIONES TÉCNICAS

Esta solución como cualquier otra, tiene unas especificaciones para su correcto funcionamiento, y son:

Cliente:

- JRE 1.4 o superior.
- Sistemas operativos windows: windows 2000, windows XP.
- Sistemas operativos libres: GNU/Linux Debian, GNU/Linux Slackware, GNU/Linux Red Hat y cualquier derivado de los mismos con Kernel 2.6.x.
- Debe tener acceso a la red donde se encuentra el servidor "OpenJavaDeveloper".

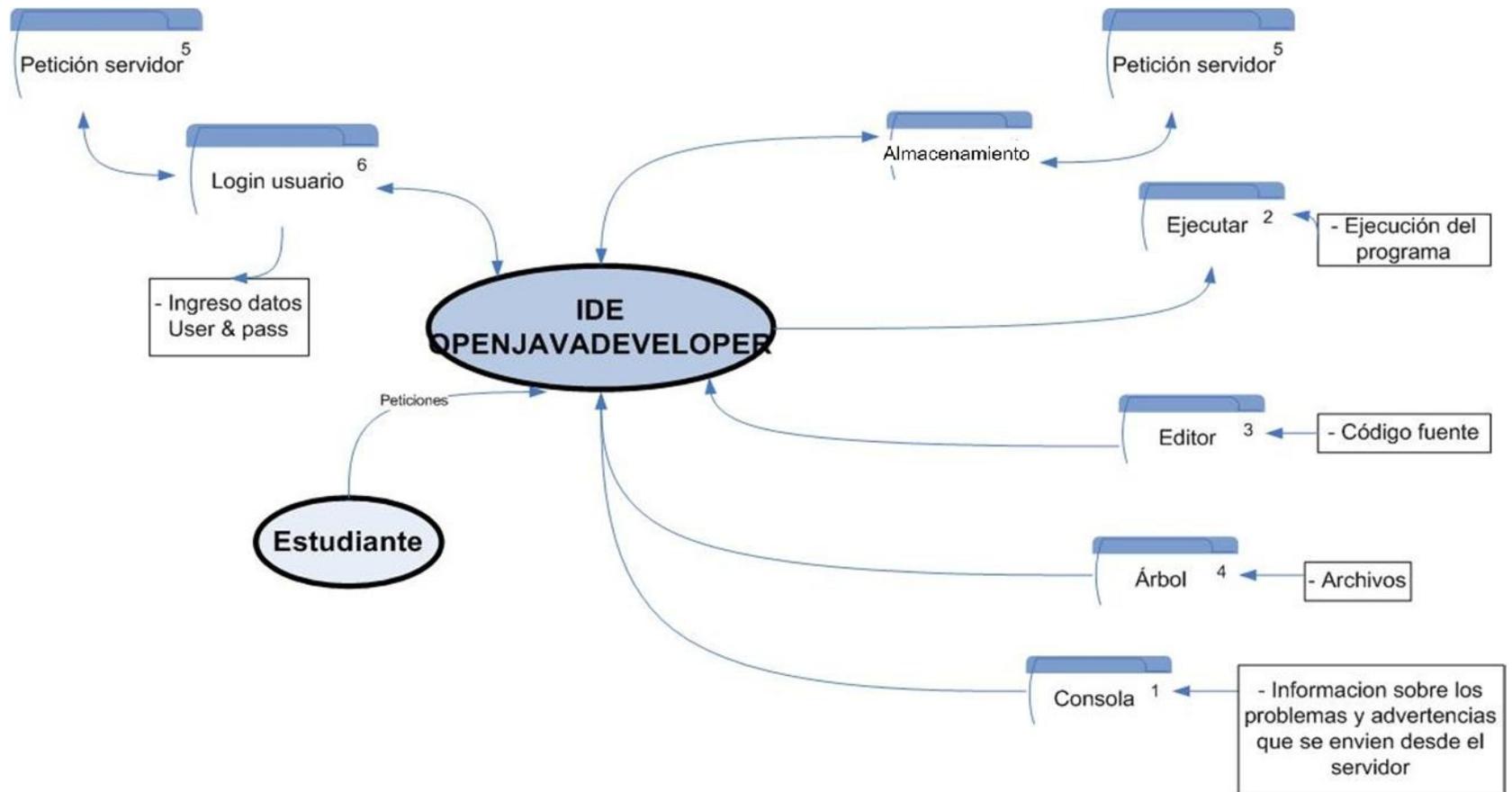
Servidor:

- JRE 1.4 o superior.
- Sistemas operativos windows: windows 2000, windows XP.
- Sistemas operativos libres: GNU/Linux Debian, GNU/Linux Slackware, GNU/Linux Red Hat y cualquier derivado de los mismos con Kernel 2.6.x.
- Tener instalado el motor gestor de base de datos MySQL 4 o superior, y el servidor apache con modulo PHP 4 o superior.

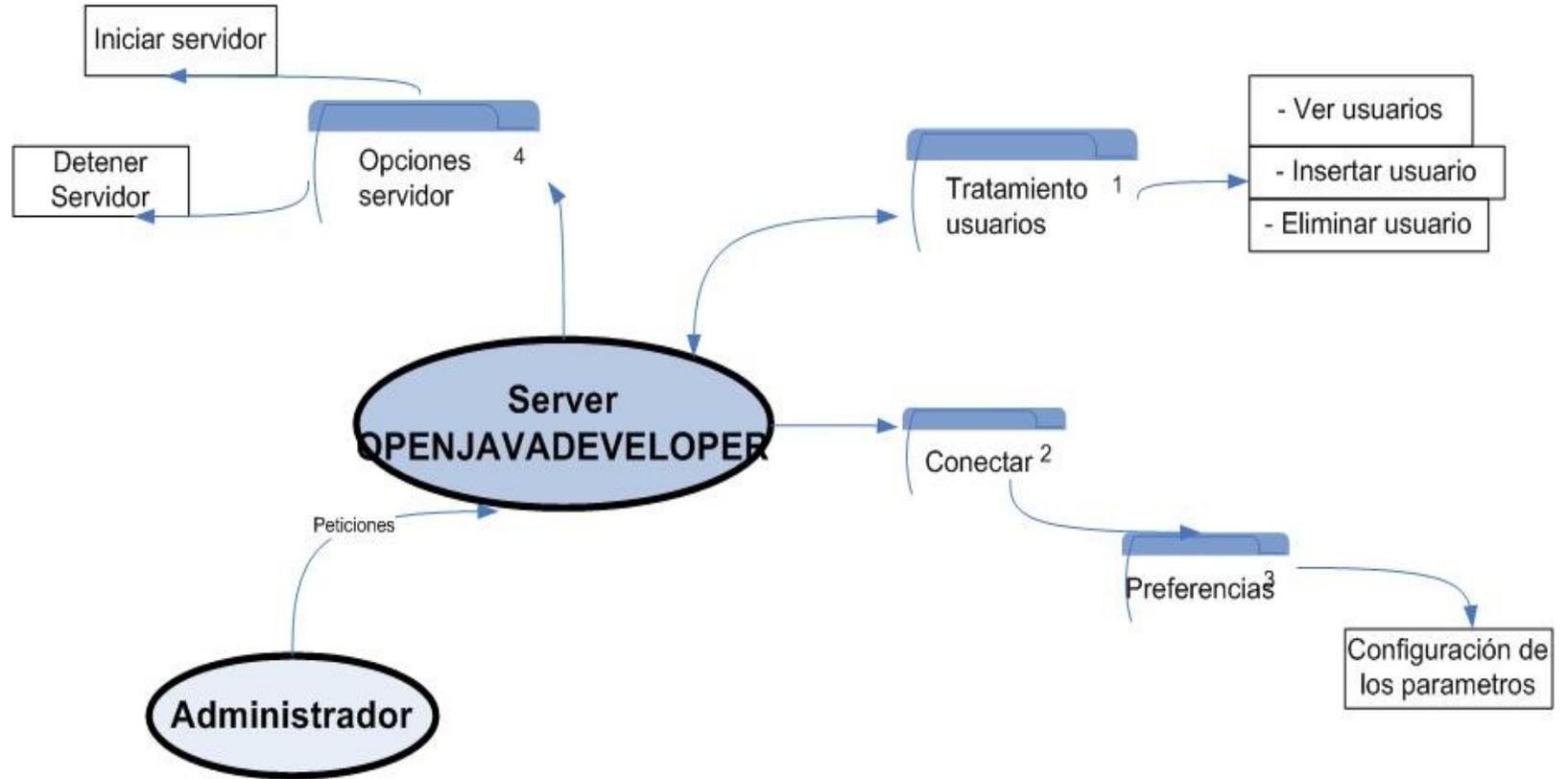
Es importante resaltar el uso de soluciones de software libre, que ayudan al progreso de diversos grupos y a la integridad de la comunidad desarrolladora de software.

10. DIAGRAMA DE FLUJO DE DATOS

10.1 Diagrama de Contexto del Cliente



10.2 Diagrama de Contexto del Servidor

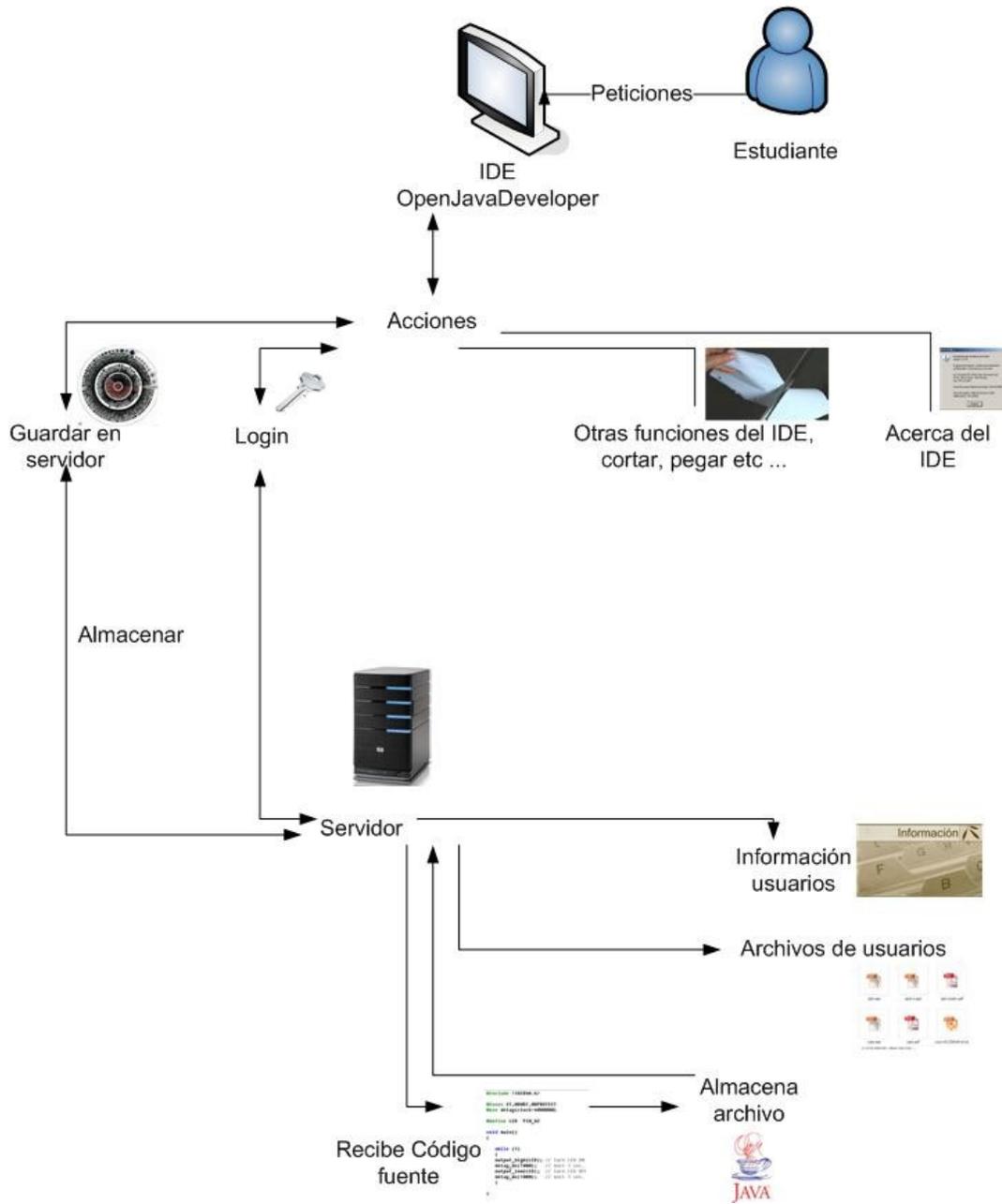


10.3 DICCIONARIO DE DATOS

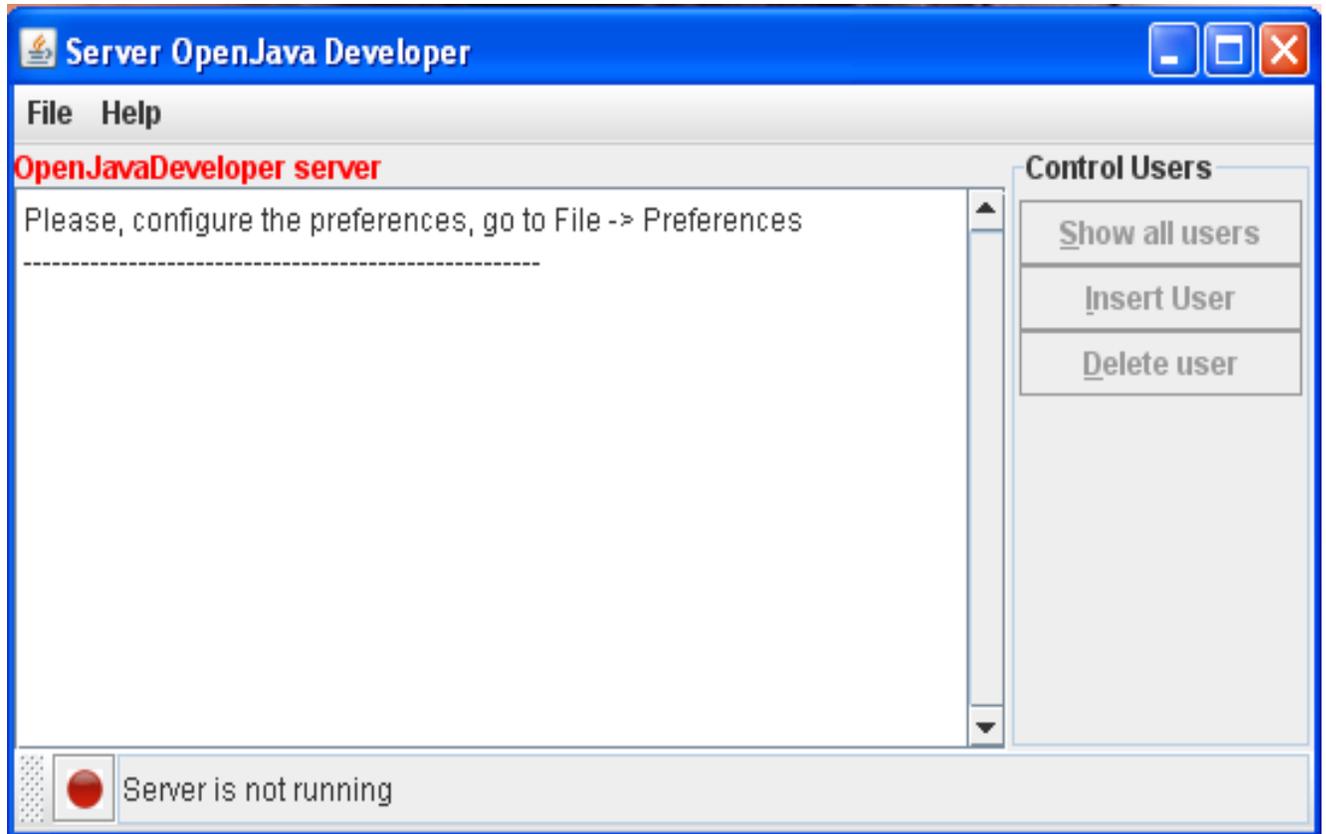
Diccionario de datos tabla "Usuarios".

TABLA USUARIO					
Descripción:				Tabla que contiene los datos básicos del usuario	
Nombre	Llave	Tipo	Longitud	Obligatorio	Descripcion
ID	PK	Entero	9	SI	Número único que identifica al usuario
Name		Cadena de texto	30	SI	Nombre completo del estudiante
Last Name		Cadena de texto	40	SI	Apellidos del estudiante
Password		Cadena de texto	40	SI	Contraseña del estudiante

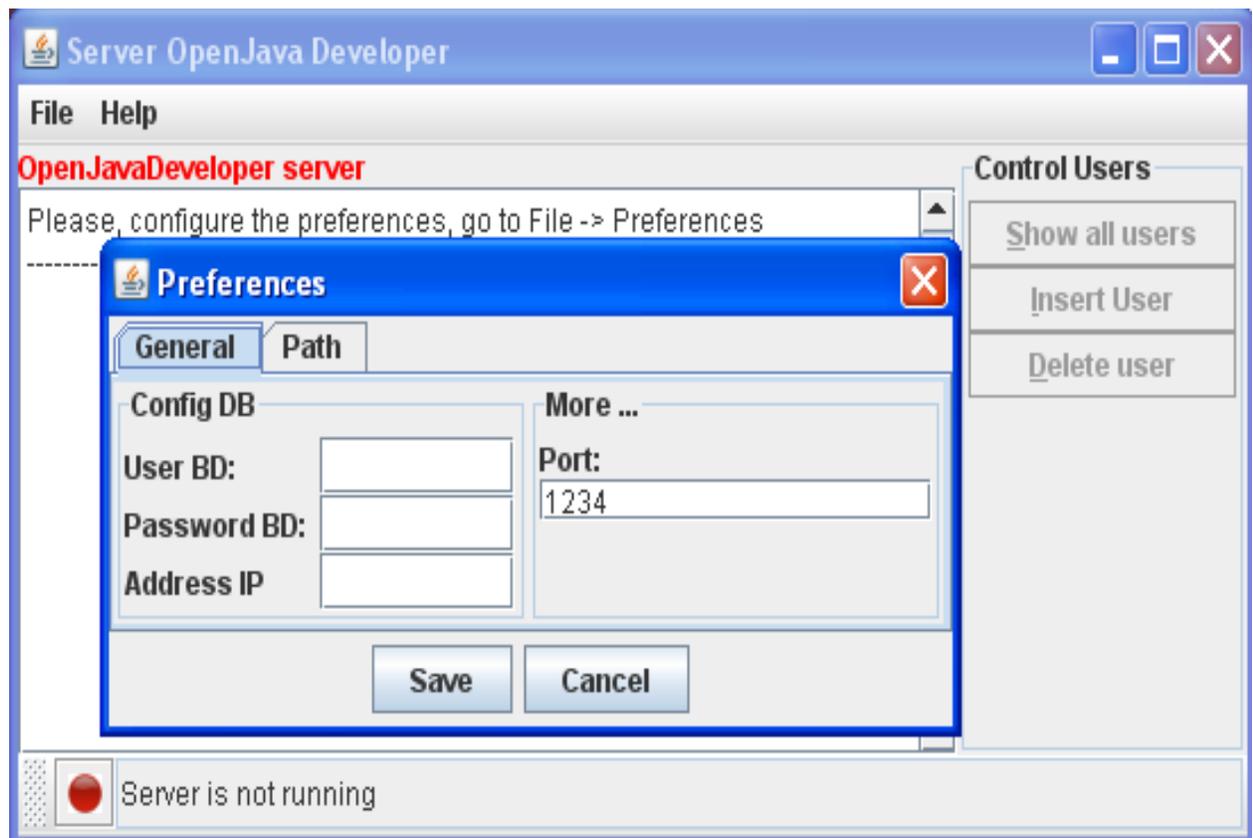
10.4 DIAGRAMA ENTRADA SALIDA



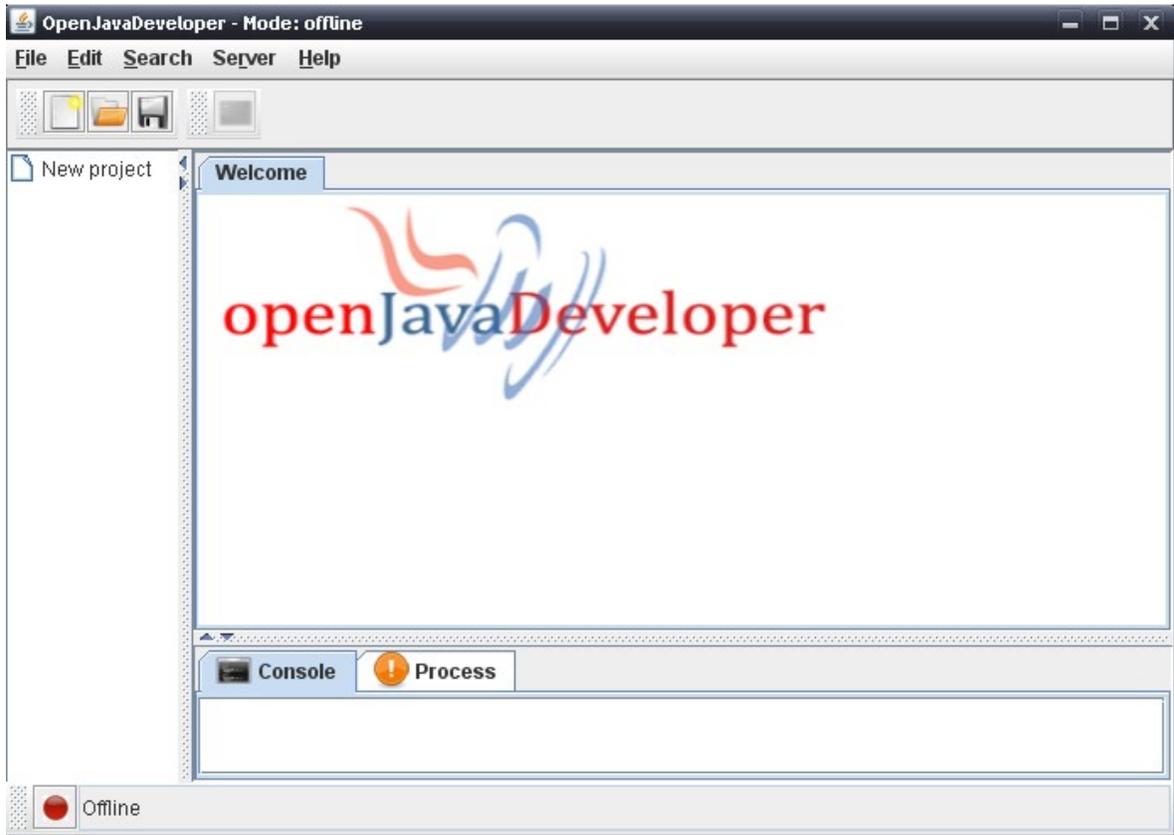
11. DISEÑO DE CAPTURAS DE PANTALLA



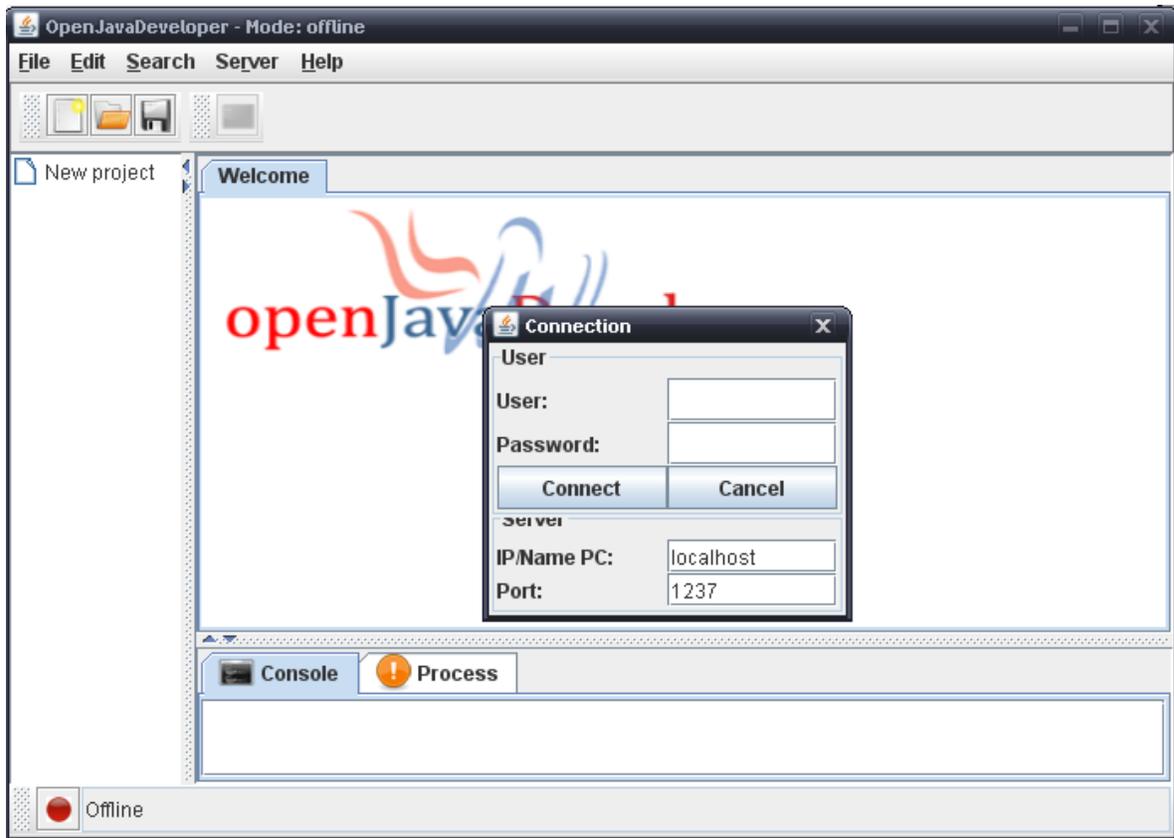
Pantalla de la ventana principal del servidor.



Pantalla para la configuración de las opciones de la Base de datos y el puerto de escucha.



Pantalla de la ventana del cliente (IDE).



Pantalla de la ventana del cliente (IDE) para la conexión al servidor.

12. ANALISIS DE RESULTADOS

Los resultados fueron evaluados y analizados, viendo la usabilidad de cada función implementada en la aplicación, teniendo en cuenta que al ser un prototipo funcional lo adecuado era eliminar las falencias posibles en las funciones de este.

- La aplicación almacena cualquier tipo de archivo java, siempre y cuando no herede de otro creado por el usuario, sin embargo puede heredar de cualquier clase integrada en la API de Java (J2SE).
- El usuario puede conectarse al servidor haciendo uso de un ID y contraseña, que es otorgada por el Administrador del servidor.
- El usuario puede cargar, modificar y salvar archivos con código fuente java o archivos de texto plano.
- Se puede hacer uso del editor y las pestañas, que permiten tener abiertos varios archivos al mismo tiempo, sin problemas de ningún tipo.
- El usuario puede desconectarse del servidor cuando lo desee, sin preocuparse por los programas que ya ha almacenado remotamente, puesto que estos se guardan tanto en el servidor como forma de respaldo y del mismo modo para el cliente.
- Una vez conectado el cliente, no puede realizar otra conexión con datos de otro usuario.

- Las opciones comunes de: cortar, copiar, pegar, buscar, rehacer entre otras, están disponibles sin ningún tipo de fallo.
- El Administrador puede insertar, eliminar, consultar usuarios y ver los archivos que cada usuario ha almacenado.
- Tanto el Administrador como el cliente tiene la opción de configurar el puerto de conexión que considere adecuado, como la dirección IP dentro de la red donde tiene ubicada la base de datos(esto únicamente para el Administrador).
- La conexión y desconexión para el servidor esta completamente funcional, al desconectarse, los clientes quedan en modo offline inmediatamente, sin la posibilidad de almacenar hasta que el Server este de nuevo online.
- El servidor tiene la opción de guardar el log de las conexiones y peticiones que han realizado los clientes.

CONCLUSIONES Y RECOMENDACIONES

- Un buen análisis permite que el desarrollo de una solución se realice con más eficiencia y practicidad.
- Es importante realizar una investigación a fondo sobre el tema, de esta forma se puede sustentar y usar el conocimiento adquirido en otras áreas.
- Todo el software que se desea usar, debe dar libertad de uso, por eso es importante conocer el tipo de licencia sobre el cual se acoge cada uno.
- La ingeniería de software es un campo muy amplio y extenso, pero teniendo los conceptos básicos claros, es posible realizar una aplicación con ciertas normas y estándares.

Se recomiendan algunas mejoras posteriores a este prototipo:

- Encriptación de los datos ID y usuario, esto mejorara la seguridad de la aplicación, ya que evitara que los datos si son robados sean legibles directamente.
- Almacenamiento de archivos que hereden de otro archivo creado por el usuario.

- Añadir al árbol de archivos, los archivos del usuario que están almacenados en el servidor, y que puedan ser usados fácilmente.
- Creación de paquetes en el IDE, y posterior implementación en el servidor.

BIBLIOGRAFÍA

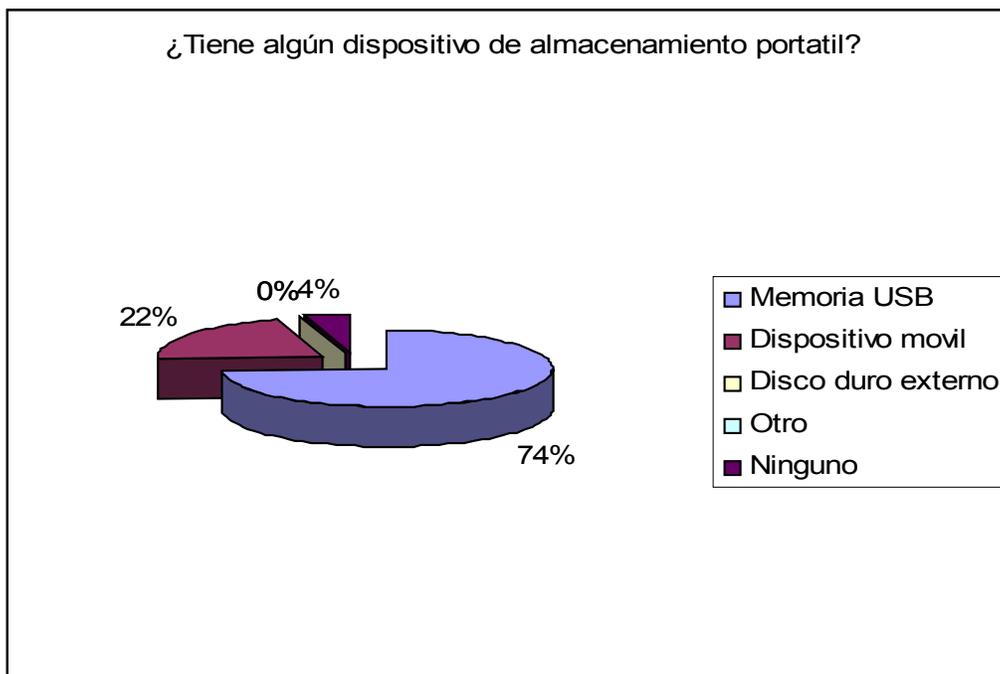
- Como programar en java 5th edición, Deitel & Deitel, Prentice Hall.
- Java cliente servidor, Nicolas Cedric - Avare Christophe - Najaman Frederic, Editorial GESTION.
- FSF (Free Software Foundation), <http://www.fsf.org/>
-
- GNU(GNU not is Unix), <http://www.es.gnu.org>
- Affero GPLv3, <http://www.affero.org/>
- Comunidad JavaHispano, <http://www.javahispano.org>
- Proyecto WebDevStudio, <http://webdevstudio.wordpress.com/>

ANEXOS

Graficas estadísticas realizadas durante la creación de la metodología, para demostrar el problema y otros aspectos importantes.

1. ¿Tiene alguno de los siguientes dispositivos de almacenamiento portátil?

Memoria USB	Dispositivo móvil	Disco duro externo	Otro	Ninguno
17	5	0	0	1



2. ¿Ha tenido inconvenientes a la hora de desarrollar aplicaciones en lenguaje Java dentro de la Universidad?

Si	No	N/R
18	0	2

