

INDICE:

Resumen	i
Glosario	ii
1. Introducción	1
1.1. Antecedentes.....	1
1.2. Planteamiento del problema	3
1.3. Justificación	4
1.4. Objetivos	4
1.4.1 Objetivo General	4
1.4.2 Objetivos específicos.....	4
2. Características del área en que se participó	5
2.1. Introducción	5
2.2. Ubicación geográfica	6
2.3. Misión	7
2.4. Visión.....	7
2.5. Valores	7
2.6. Ingeniería en sistemas computacionales.....	7
2.6.1 Objetivo general de la carrera.....	7
2.6.2 Objetivo.....	8
2.6.3 Quehacer profesional.....	8
2.6.4 Campo de trabajo	8
2.7 Organigrama del área en que se trabajó.....	9
3. Alcances y limitaciones	11
4. Estado del arte.....	12
4.1 Net Bus	12
4.2 VNC.....	14
5. Marco teórico	16
5.1 Java RMI	16
5.2 Arquitectura cliente – servidor	17
5.3 Sockets	19
5.4 Puerto de comunicación	21

5.5 Protocolo de comunicación	21
6. Análisis del sistema.....	25
6.1 Diagrama de procesos	25
6.2 Diagrama de actividad	26
6.3 Diagrama de casos de usos.....	27
6.4 Diagrama de estados	28
7. Diseño y desarrollo del sistema	29
7.1 Diagrama de clases	29
7.2 Bloqueo de teclado	29
7.3 Bloqueo de mouse	30
7.4 Proyección de diapositivas.....	30
8. Casos de pruebas.....	31
9. Conclusiones y trabajos futuros	34
10. Bibliografía	35
11. Anexos	36
Anexo A	36
Anexo B.....	37

Resumen:

Se realizó un sistema (software) con la capacidad de abrir archivos en PowerPoint en formato .pps además de poder bloquear el teclado y mouse de los equipos de cómputo que se encuentren ubicados dentro una red local en una sala de cómputo y así lograr que los estudiantes tengan un menor índice de distracción.

Para desarrollar este software se utilizó la arquitectura cliente servidor, de igual modo se implementó con Java RMI y lenguaje C para desarrollar los diferentes componentes del sistema.

Con este software se logró evitar el uso del cañón de proyección (video proyector) y así reducir el índice de distracción por parte de los alumnos al estar frente a un equipo de cómputo y combatir las posibles deficiencias visuales que pueden tener los alumnos ya que tendrán en primer plano las diapositivas brindando así la misma calidad de información a todos los estudiantes.

Glosario de términos

ARP: Son las siglas en inglés de Address Resolution Protocol (Protocolo de resolución de direcciones). Es un protocolo de nivel de red responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP.

ASN: Abstract Syntax Notation One (notación sintáctica abstracta 1, ASN.1) es una norma para representar datos independientemente de la máquina que se esté usando y sus formas de representación internas. Es un protocolo de nivel de presentación en el modelo OSI. El protocolo SNMP usa el ASN.1 para representar sus objetos gestionables.

ATM: Modo de Transferencia Asíncrona o Asynchronous Transfer Mode es una tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.

CDP *Cisco Discovery Protocol*, 'protocolo de descubrimiento de Cisco', es un protocolo de red propietario de nivel 2, desarrollado por Cisco Systems y usado en la mayoría de sus equipos. Es utilizado para compartir información sobre otros equipos Cisco directamente conectados, tal como la versión del sistema operativo y la dirección IP. CDP también puede ser usado para realizar encaminamiento bajo demanda (ODR, *On-Demand Routing*), que es un método para incluir información de encaminamiento en anuncios CDP, de forma que los protocolos de encaminamiento dinámico no necesiten ser usados en redes simples.

CRC: comprobación de redundancia cíclica es un tipo de función que recibe un flujo de datos de cualquier longitud como entrada y devuelve un valor de longitud fija como salida.

FDDI: Fiber Distributed Data Interface es un conjunto de estándares ISO y ANSI para la transmisión de datos en redes de computadoras de área extendida o local

(LAN) mediante cable de fibra óptica. Se basa en la arquitectura token ring y permite una comunicación tipo Full Duplex. Dado que puede abastecer a miles de usuarios, una LAN FDDI suele ser empleada como backbone para una red de área amplia (WAN).

FTP: File Transfer Protocol - Protocolo de Transferencia de Archivos en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor.

HDLC: High-Level Data Link Control, control de enlace síncrono de datos) es un protocolo de comunicaciones de propósito general punto a punto y multipunto, que opera a nivel de enlace de datos.

HTTP: Hypertext Transfer Protocol en español *protocolo de transferencia de hipertexto* es el protocolo usado en cada transacción de la World Wide Web

IANA: Internet Assigned Numbers Authority es la Agencia de Asignación de Números de Internet. Era el antiguo registro central de los protocolos Internet, como puertos, números de protocolo y empresa, opciones y códigos. Fue sustituido en 1998 por ICANN.

ICMP: El Protocolo de Mensajes de Control de Internet (por sus siglas de *Internet Control Message Protocol*) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP).

IGMP: Internet Group Management Protocol se utiliza para intercambiar información acerca del estado de pertenencia entre enrutadores IP que admiten la multidifusión y miembros de grupos de multidifusión

IMAP: Internet Message Access Protocol, es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

IP: Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (*Internet Protocol*), que corresponde al nivel de red del protocolo TCP/IP.

IPX/SPX: Del inglés *Internetwork Packet Exchange/Sequenced Packet Exchange*, Protocolo Novell o simplemente IPX es una familia de protocolos de red desarrollados por Novell y utilizados por su sistema operativo de red NetWare.

IRC: *Internet Relay Chat* es un protocolo de comunicación en tiempo real basado en texto, que permite debates entre dos o más personas. Se diferencia de la mensajería instantánea en que los usuarios no deben acceder a establecer la comunicación de antemano, de tal forma que todos los usuarios que se encuentran en un canal pueden comunicarse entre sí, aunque no hayan tenido ningún contacto anterior

JDK: Java Development Kit, es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red.

LADP: *Lightweight Directory Access Protocol* (en español *Protocolo Ligero de Acceso a Directorios*) que hacen referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos, aunque su sistema de almacenamiento puede ser diferente, a la que pueden realizarse consultas.

NFS: Network File System (*Sistema de archivos de red*), es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

NNTP: Network News Transport es un protocolo inicialmente creado para la lectura y publicación de artículos de noticias en Usenet. Su traducción literal al español es "protocolo para la transferencia de noticias en red".

OSI: Open System Interconnection (*Modelo de interconexión de sistemas abiertos*), que es la propuesta que hizo la Organización Internacional para la Estandarización (ISO) para estandarizar la interconexión de sistemas abiertos.

POP3: En informática se utiliza el Post Office Protocol (*Protocolo de la oficina de correo*) en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. Es un protocolo de nivel de aplicación en el Modelo OSI.

RARP: Reverse Address Resolution Protocol (Protocolo de resolución de direcciones inverso). Es un protocolo utilizado para resolver la dirección IP de una dirección hardware dada (como una dirección Ethernet).

RMI: *Java Remote Method Invocation* es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y provee de un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI.



RPC: *Remote Procedure Call*, Llamada a Procedimiento Remoto es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos. El protocolo es un gran avance sobre los sockets usados hasta el momento. De esta manera el programador no tenía que estar pendiente de las comunicaciones, estando éstas encapsuladas dentro de las RPC.

SMB: Server Message Block o SMB es un Protocolo de red (que pertenece a la capa de aplicación en el modelo OSI) que permite compartir archivos e impresoras (entre otras cosas) entre nodos de una red. Es utilizado principalmente en ordenadores con Microsoft Windows y DOS.

SMTP: Simple Mail Transfer Protocol, Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.). Está definido en el RFC 2821 y es un estándar oficial de Internet.

SNMP: El Protocolo Simple de Administración de Red es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

SSH: Secure SHell, en español: intérprete de órdenes segura es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo.

SSL: Secure Sockets Layer -Protocolo de Capa de Conexión Segura- y Transport Layer Security -Seguridad de la Capa de Transporte- , su sucesor, son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

TCP: Transmission Control Protocol (en español Protocolo de Control de Transmisión), es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.

UDP: User Datagram Protocol es un protocolo del nivel de transporte basado en el intercambio de datagramas (Paquete de datos). Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

UTP: Unshielded Twisted Pair, par trenzado no apantallado es un tipo de cable de par trenzado que no se encuentra apantallado y que se utiliza principalmente para comunicaciones. Se encuentra normalizado de acuerdo a la norma estadounidense TIA/EIA-568-B y a la internacional ISO/IEC 11801.

VNC: es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. También llamado software de escritorio remoto

1. INTRODUCCION

1.1 Antecedentes

El proyector de acetatos es un aparato que nos permite ver en una pantalla, lo que está plasmado en acetato (transparente) sea a color o blanco y negro, quizás sea el aparato más común y de mayor utilidad en las presentaciones o exposiciones. Con respecto al cañón, es de más fácil manejo, la única función prácticamente es la de enfoque.

Este tipo de equipos fueron muy comunes desde la década del 50 hasta mediados de la del 90 utilizándose principalmente en:

a) En el hogar como forma de entretenimiento; los miembros de una familia o amigos se reunían a ver diapositivas, generalmente de fotografías familiares.

b) En los colegios y universidades como proyecciones audiovisuales académicas: eran utilizados por los profesores y alumnos para el dictado de clases, conferencias y exposiciones, pues la proyección ampliada de diapositivas en colores constituye uno de los auxiliares más útiles y prácticos en la didáctica.

Un proyector de vídeo o cañón proyector es un aparato que recibe una señal de vídeo y proyecta la imagen correspondiente en una pantalla de proyección usando un sistema de lentes, permitiendo así visualizar imágenes fijas o en movimiento.

Todos los proyectores de vídeo utilizan una luz muy brillante para proyectar la imagen y los más modernos pueden corregir curvas, borrones y otras inconsistencias a través de los ajustes manuales. Los proyectores de vídeo son mayoritariamente usados en salas de presentaciones o conferencias, en aulas docentes, aunque también se pueden encontrar aplicaciones para cine en casa. La señal de vídeo de entrada puede provenir de diferentes fuentes, como un sintonizador de televisión (terrestre o vía satélite), un ordenador personal, etc.

Otro término parecido a proyector de vídeo es retroproyector el cual, a diferencia del primero, se encuentra implantado internamente en el aparato de televisión y proyecta la imagen hacia el observador.

El primer concepto de la proyección se utilizó en 1646 por un sacerdote alemán, Atanasio Kircher. Se utiliza para proyectar el pintado a mano las imágenes en una Pantalla en blanco con la ayuda de una vela o lámpara de aceite. Su concepto sirvió de base para otras personas para inventar un dispositivo que se conoce como un "proyector". Más tarde, en 1879, Eadweard Muybridge desarrolló un dispositivo llamado "Zoopraxiscope". Esto se usa para mostrar las películas e imágenes mediante la creación de una ilusión de movimiento. Que se conoce generalmente como "Padre de la película". Más tarde se reunió y propuso Thomas Edison para inventar un dispositivo que se derivarían de la colaboración de su "Zoopraxiscope" y Edison del "fonógrafo". Edison no estaba interesado en participar en el proyecto. Sin embargo, presentó una solicitud con la oficina de Patentes en 1888 la descripción de sus puntos de vista para un dispositivo que denominó como "Kinetoscope", que sería utilizado para la proyección. Kinetoscope es una combinación de dos palabras griegas, "Kineto" y "Scopos". "Kineto" significa "movimiento" y "Scopos" significa "ver". Describió que el dispositivo de grabación y reproducción de los objetos en movimiento.

William Kennedy Laurie Dickson, un fotógrafo de profesión ayudante de Edison y se le asignó la tarea. Dickson comenzó a experimentar en la idea concebida por Edison. En 1895, Woodville Latham creó Eidoloscope proyector. Latham es técnicamente asesorado por Dickson en la máquina. Este episodio dio lugar a Dickson dejando la empresa de Edison. Al mismo tiempo, C. Francis Jenkins y Thomas Armat inventó un proyector de cine llamado "Phantoscope", que se puso de manifiesto al público en 1895. Pronto, ambos partidos y sus formas comenzaron afirmando el mérito de la invención. Thomas Armat mostraron que el dispositivo Raff Gammon y que fueron los propietarios de la empresa Kinetoscope. Además, negoció el dispositivo con Thomas Edison en la fabricación el proyector con su nombre. Edison de acuerdo con la condición de que Phantoscope se anunció como una nueva invención de Edison. Este invento fue de su nombre

como "Vitascope". La primera demostración de "Vitascope" se hizo el 23 de abril de 1896, y en Koster Bial del Music Hall de Nueva York. Vitascope muy pronto se convirtió en famoso. Más tarde, en noviembre de 1896, un nuevo proyector llamado Projectoscope o Proyección Kinetoscope fue desarrollado por "La Compañía Edison", que fue un éxito comercial.

1.2 Planteamiento del problema

Con base en los altos costos tanto para adquirir los equipos de video proyección (cañón), las refacciones y/o consumibles (lámparas) y los altos costos de reparación de dichos equipos, además de lo delicado de dichas refacciones y que hoy en día la mayoría de las clases que se imparten en las escuelas utilizan uno de estos equipos y que en la mayoría de los casos las escuelas no cuentan con el número de equipos necesarios o los recursos para adquirirlos y/o repararlos por tal motivo es necesario contar con herramientas que permitan impartir las clases de manera eficiente y de calidad simulando el uso del cañón, logrando así poder impartir una clase de manera eficiente sin la necesidad de depender de este equipo de apoyo.

De acuerdo con el proceso de observación (ver anexo A) realizado a los estudiantes y profesores del Instituto Tecnológico de Tuxtla Gutiérrez (ITTG) detectamos que el principal problema en el empleo de instrumentos tecnológicos en la educación (cañón de proyección, proyector de acetatos etc.) es el mal manejo de dichos equipos. De acuerdo con la opinión expresado por los maestros, los alumnos al contar con un equipo de cómputo los alumnos se distraen con facilidad, en contraparte los alumnos opinan que el contenido de las "laminas" (diapositivas) no se alcanza a apreciar con claridad (ver anexo I). Entonces ¿Cuál es el problema?

1.3 Justificación

En el aspecto económico al contar con una herramienta de apoyo para evitar el uso del cañón de proyección, se reducirían los costos tanto de adquisición y reparación de dichos equipos, por otro lado se podrían aprovechar los recursos en otras actividades como son la actualización de los equipos de cómputo, compra de materiales para el laboratorio, etc.

En el aspecto educativo al contar con un sistema capaz de mostrar las diapositivas de la clase que el maestro este impartiendo en los equipos que se encuentren en red, los alumnos podrán tener un menor índice de distracción haciendo otras cosas ajenas a la clase ya que alcanzan a ver el material o contenido de la clase en donde se encuentren, se combatirán las posibles deficiencias visuales y todos los alumnos contarían con la misma calidad de información.

1.4 Objetivos

1.4.1 Objetivo General

- Desarrollar una herramienta capaz de bloquear y controlar los equipos de cómputo dentro de una red local y al mismo tiempo proyectar la clase que el profesor este impartiendo en los diferentes equipos de cómputo.

1.4.2 Objetivos específicos

1. Bloqueo del teclado del equipo de cómputo
2. Bloqueo del mouse del equipo de cómputo
3. Proyectar las presentaciones del docente hacia los equipos de la red local.

2. CARACTERISTICAS DEL AREA EN QUE SE PARTICIPO

2.1 Antecedentes

En la década de los 70's, se incorpora el estado de Chiapas al movimiento educativo nacional extensión educativa, por intervención del Gobierno del Estado de Chiapas ante la federación.

Esta gestión dio origen a la creación del Instituto Tecnológico Regional de Tuxtla Gutiérrez (ITRTG) hoy Instituto Tecnológico de Tuxtla Gutiérrez (ITTG).

El día 23 de agosto de 1971 el Gobernador del Estado, Dr. Manuel Velasco Suárez, colocó la primera piedra de lo que muy pronto sería el Centro Educativo de nivel medio superior más importante de la entidad.

El día 22 de octubre de 1972, con una infraestructura de 2 edificios con 8 aulas, 2 laboratorios y un edificio para talleres abre sus puertas el Instituto Tecnológico de Tuxtla Gutiérrez con las carreras de Técnico en Máquinas de Combustión Interna, Electricidad, Laboratorista Químico y Máquinas y Herramientas.

En el año 1974 dio inicio la modalidad en el nivel superior, ofreciendo las carreras de Ingeniería Industrial en Producción y Bioquímica en Productos Naturales. En 1980 se amplió la oferta educativa al incorporarse las carreras de Ingeniería Industrial Eléctrica e Ingeniería Industrial Química.

En 1987 se abre la carrera de Ingeniería en Electrónica y se liquidan en 1989 las carreras del sistema abierto del nivel medio superior y en el nivel superior se reorientó la oferta en la carrera de Ingeniería Industrial Eléctrica y se inicia también Ingeniería Mecánica.

En 1991 surge la licenciatura en Ingeniería en Sistemas Computacionales.

Desde 1997 el Instituto Tecnológico de Tuxtla Gutiérrez ofrece la Especialización en Ingeniería Ambiental como primer programa de postgrado.

En 1998 se estableció el programa interinstitucional de postgrado con la Universidad Autónoma de Chiapas para impartir en el Instituto Tecnológico la Maestría en Biotecnología.

En el año 1999 se inició el programa de Maestría en Administración como respuesta a la demanda del sector industrial y de servicios de la región.

A partir de 2000 se abrió también la Especialización en Biotecnología Vegetal y un año después dio inicio el programa de Maestría en Ciencias en Ingeniería Bioquímica y la Licenciatura en Informática.

2.2 Ubicación geográfica

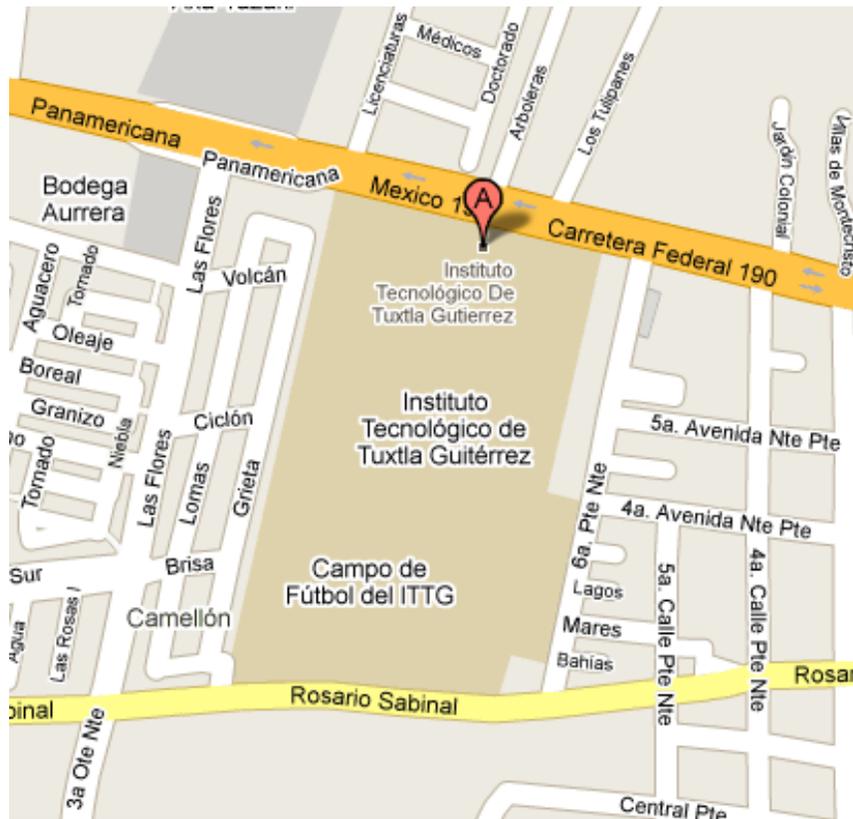


Fig. 2.1 – Ubicación del ITTG

Actualmente las instalaciones del ITTG se encuentran ubicadas en la Carretera Panamericana Km. 1080, C.P. 29000, apartado postal 599 en la ciudad de Tuxtla Gutiérrez en el estado de Chiapas

2.3 Misión

Formar de manera integral profesionales de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

2.4 Visión

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

2.5 Valores

- El ser humano
- El espíritu de servicio
- El liderazgo
- El trabajo en equipo
- La calidad
- El alto desempeño

2.6 Ingeniería en sistemas computacionales

2.6.1 Objetivo General de la Carrera

Formar profesionales capaces de diseñar y desarrollar sistemas de software que les permitan propiciar el fortalecimiento de la tecnología nacional; administrar proyectos de desarrollo de software y especificar y evaluar configuraciones de sistemas de cómputo en todo tipo de organizaciones donde se utilicen sistemas computacionales.

2.6.2 Objetivo

Formar profesionales capaces de diseñar y desarrollar sistemas de software que les permitan propiciar el fortalecimiento de la tecnología nacional; administrar proyectos de desarrollo de software y especificar y evaluar configuraciones de sistemas de cómputo en todo tipo de organizaciones donde se utilicen sistemas computacionales.

2.6.3 Quehacer Profesional

Las necesidades actuales de desarrollo hacen necesario que en el estado se cuente con profesionistas capaces de diseñar, crear y dar mantenimiento a modernos e innovadores productos. La transformación de los sistemas computacionales ha sido muy rápida; la combinación de software y hardware con tecnología de comunicación, de interface con el medio ambiente y otras han permitido el desarrollo de poderosos sistemas. Se espera, que los tres sectores productivos del estado de apoyen aún más en los sistemas computacionales, telecomunicaciones y redes.

2.6.4 Campo de Trabajo

El ingeniero en sistemas computacionales puede prestar sus servicios de manera independiente, trabajar en todo tipo de empresas industriales, de servicios, públicas o privadas como podrían ser industrias extractivas, de transformación, empresas de servicios, comerciales, exportadoras, de distribución, de desarrollo informático, de inversión o crédito.

2.6 Organigrama del área en que se trabajo

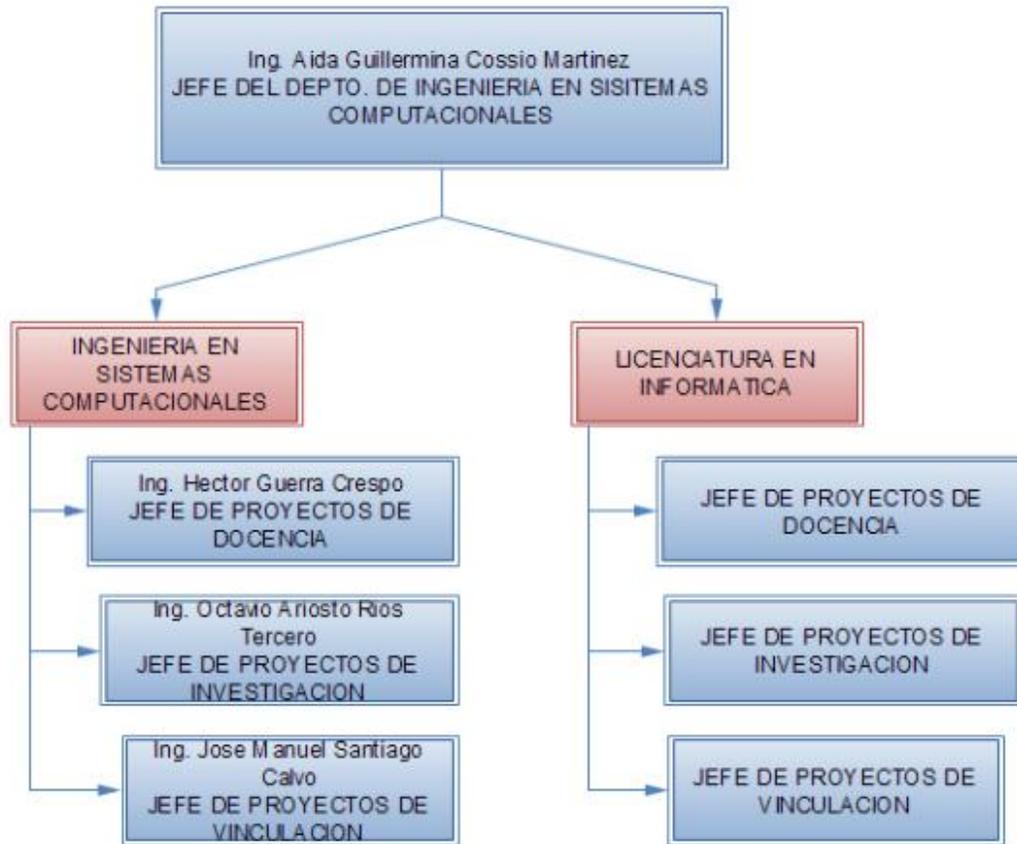


Fig. 2.2 – Organigrama del depto. De sistemas



Fig. 2.3 – Laboratorio de cómputo

3. ALCANCES Y LIMITACIONES

Alcances

La herramienta permitirá abrir archivos de PowerPoint con extensión pps debido a que de esta manera las diapositivas se abren automáticamente sin necesidad de la intervención del usuario, también contará con un módulo de bloqueo de teclado y mouse para que el alumno al estar frente al equipo de cómputo se concentre única y exclusivamente en el contenido de las diapositivas.

Limitaciones

Para que el sistema pueda funcionar es necesario tener instalado al menos la versión de Java JDK 1.1 o posterior (versión 1.6 recomendada), también es necesario configurar las variables de entorno tanto en el cliente como en el servidor.

El desarrollo de la herramienta está pensado para que pueda actualizarse de tal manera que se le puedan implementar más módulos que se listan a continuación:

- Sistema de transmisión de video (ver en tiempo real la pantalla de profesor)
- Sistema de envío de mensajes (parecido a un chat)
- Sistema de envío de archivos (para facilitar la evaluación de las tareas)
- Sistema de envío de cuestionarios (para evaluar en tiempo real el contenido de la clase)

4. ESTADO DEL ARTE

4.1 NetBus

Es un software para el control de una forma remota de sistemas informáticos Microsoft Windows a través de una red. Fue escrito en Delphi por el sueco Carl-Fredrik Neikter en 1998 y ha sido muy controvertido por su potencial de ser utilizado como una puerta trasera.

NetBus se escribió en Delphi por Carl-Fredrik Neikter, un programador sueco en marzo de 1998. Entro en circulación antes que Back Orifice, fue liberado en agosto de 1998. El autor afirmó que el programa estaba destinado a ser usado para bromas, no para irrumpir ilegalmente en los sistemas informáticos. Traducido del sueco, el nombre significa "NetPrank". Sin embargo, el uso de NetBus ha tenido graves consecuencias. En 1999, NetBus se utilizó para la pornografía infantil en el equipo de trabajo de Magnus Eriksson, un erudito en Derecho Universidad de Lund. Las 3500 las imágenes fueron descubiertas por los administradores del sistema, y Eriksson se supone que lo había descargado a sabiendas. Eriksson perdió su puesto de investigador en la facultad, y tras la publicación de su nombre huyo del país y tuvo que buscar atención médica profesional para hacerle frente al problema. Fue absuelto de los cargos penales a finales de 2004, cuando un tribunal dictaminó que se había utilizado NetBus para el control de su ordenador.

Existen dos componentes para la arquitectura cliente-servidor. El servidor debe ser instalado y ejecutado en el equipo que quiere ser controlado a distancia, el tamaño de archivo es de casi 500 KB. El nombre y el icono han variado mucho de versión a versión. Nombres comunes eran "Patch.exe" y "SysEdit.exe". Cuando se inicia por primera vez, el servidor se instala en el ordenador host, incluyendo la modificación de Windows del Registro para que se inicie automáticamente en cada inicio del sistema.

El servidor atiende las conexiones en el puerto 12345 (en algunas versiones, el puerto es configurable), el puerto 12346 se utiliza para ciertas tareas, así como el puerto 20034.

El cliente es un programa separado de la presentación con una interfaz gráfica que permite al usuario realizar una serie de actividades en el equipo remoto, como pueden ser:

- Registro de las pulsaciones del teclado (Keylogging)
- Inyección de teclas de manera remota
- Captura de pantalla, permitiendo descargarla al cliente
- Ejecución remota de aplicaciones
- Navegación por los archivos y carpetas del servidor
- Apagado del sistema
- Abrir / cerrar la bandeja del CD
- Tunneling

NetBus hace conexiones a través de un sin número de sistemas, el cliente de NetBus fue diseñado para las siguientes versiones de Windows:

- Windows 95
- Windows ME
- Windows NT 4.0
- Windows 2000 y Windows XP (NetBus client v1.70)

La mayor parte del protocolo utilizado entre el cliente y el servidor es textual. Por lo tanto, el servidor puede ser controlado por comandos, permitiendo administrar equipos con NetBus desde sistemas operativos distintos a Windows.

Características

(Tales como la captura de pantalla)

Requieren una aplicación con capacidad de aceptar datos binarios, tales como netcat. La mayoría de los protocolos más comunes (como el Internet Relay Chat protocolo POP3, SMTP, HTTP) también se puede utilizar con conexiones de una manera similar.

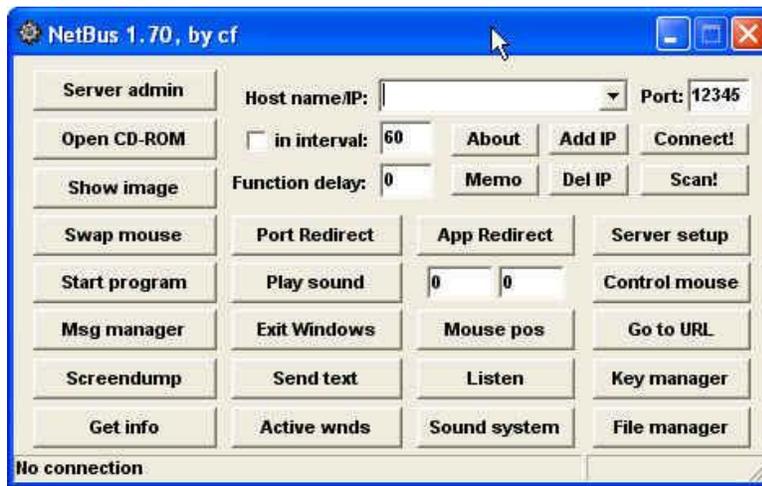


Fig. 4.1 – Interfaz de NetBus

NetBus Pro 2.0 fue liberado en febrero de 1999. Se comercializa como una poderosa herramienta de administración remota, pero existen versiones especiales hacker que permiten utilizarlo con fines ilegales.

Todas las versiones del programa han sido ampliamente utilizadas por los "script Kiddies" y fue popularizado por el lanzamiento de Back Orifice. Debido a su pequeño tamaño, Back Orifice se puede utilizar para obtener acceso y control a una máquina remota. El atacante puede utilizar Back Orifice NetBus para instalar el servidor en el equipo de destino. La mayoría de antivirus detecta y elimina estos programas NetBus.

4.2 VNC:

VNC son las siglas en inglés de *Virtual Network Computing* (Computación Virtual en Red).

VNC es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. También llamado software de escritorio remoto. VNC no impone restricciones en el sistema operativo del ordenador servidor con respecto al del cliente: es posible compartir la pantalla de una máquina con cualquier sistema operativo que soporte VNC conectándose desde otro ordenador o dispositivo que disponga de un cliente VNC portado.

La versión original del VNC se desarrolló en Reino Unido, concretamente en los laboratorios AT&T, en Cambridge. El programa era de código abierto por lo que

cualquiera podía modificarlo y existen hoy en día varios programas para el mismo uso.

En la enseñanza VNC sirve para que el profesor comparta su pantalla con los alumnos, por ejemplo en un laboratorio. También puede usarse para que un técnico ayude a un usuario inexperto, el técnico ve remotamente el problema que informa el usuario.

El programa servidor suele tener la opción de funcionar como servidor HTTP para mostrar la pantalla compartida en un navegador con soporte de Java. En este caso el usuario remoto (cliente) no tiene que instalar un programa cliente de VNC, éste es descargado por el navegador automáticamente.

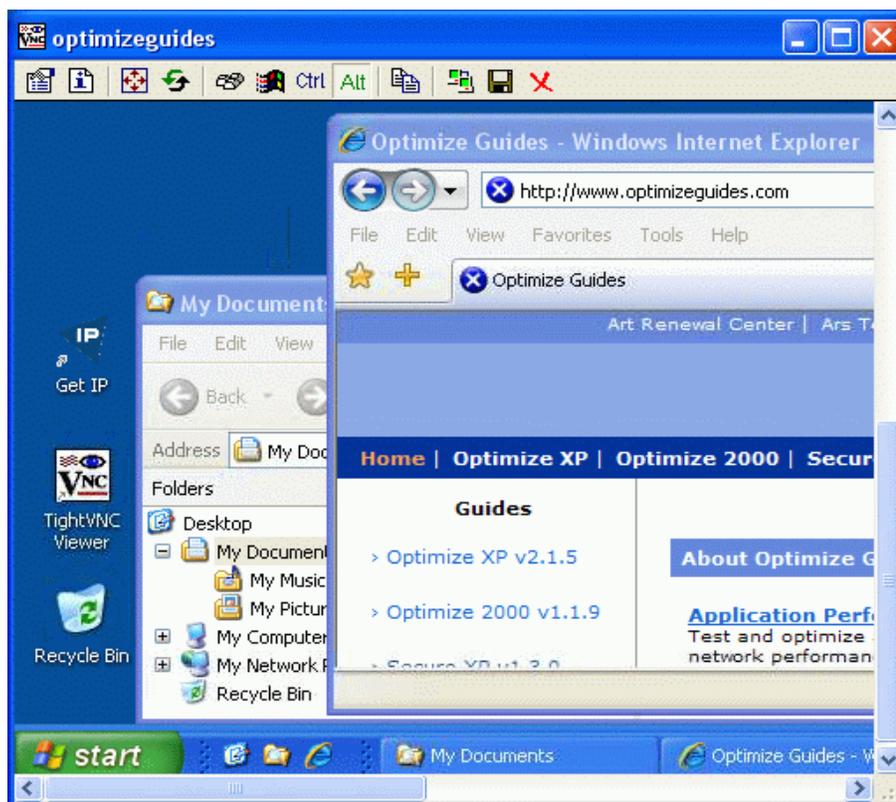


Fig. 4.2 – Demo de VNC

Cabe destacar que hasta la fecha no existe ningún programa que permita además de ver el escritorio del equipo servidor bloquee los dispositivos de entrada como el teclado y mouse.

5. MARCO TEORICO

5.1 Java RMI

RMI (java Remote Method Invocation) es un mecanismo ofrecido por java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de java y provee un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías se debe utilizar CORBA o Soap en lugar de RMI.

Por medio de RMI, un programa Java puede explorar un objeto, lo que significa que este queda accesible a través de la red y el programa permanece a la espera de peticiones en un puerto TCP. A partir de este momento, un cliente puede conectarse e invocar los métodos proporcionados por el objeto.

La invocación se compone de los siguientes pasos:

- Encapsulado (marshalling) de los parámetros
- Invocación del método (del cliente sobre el servidor). El invocador se queda en espera de una respuesta
- Al terminar la ejecución, el servidor serializa el valor de retorno (si lo hay) y lo envía al cliente.
- El código cliente recibe la respuesta y continúa como si la invocación hubiera sido local.

La arquitectura de RMI se puede ver como un modelo de cuatro capas.

La primera capa es la de aplicación y se corresponde con la implementación real de las aplicaciones cliente-servidor. Aquí tienen lugar las llamadas a alto nivel para acceder y exportar objetos remotos. Cualquier aplicación que quiera que sus métodos estén disponibles para su acceso por clientes remotos debe declarar dichos métodos en una interfaz que extienda `java.rmi.Remote`

La segunda capa es la capa proxy, o capa stub.skeleton. Esta capa es la que interactúa directamente con la capa aplicación, todas las llamadas a los objetos remotos y acciones junto con sus parámetros y retorno de objetos tienen lugar en esta capa.

La tercera capa es la de referencia remota y es responsable del manejo de la parte más semántica de las invocaciones remotas. También es responsable de la gestión de la replicación de objetos y realización de tareas específicas de la implementación con los objetos remotos, como el establecimiento de las persistencias semánticas y estrategias adecuadas para la recuperación de conexiones perdidas. En esta capa se espera una conexión de tipo stream (stream-oriented connection) desde la capa de transporte.

La cuarta capa es la de transporte. Es la responsable de realizar las conexiones necesarias y manejo del transporte de los datos de una máquina a otra. El protocolo de transporte subyacente para RMI es JRMP (Java Remote Method Protocol), que solamente es comprendido por programas Java.

5.2 Arquitectura Cliente – Servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- Al contratar un servicio de redes, se tiene que tener en la velocidad de conexión que le otorga al cliente y el tipo de cable que utiliza, por ejemplo: cable de cobre ronda entre 1 ms y 50 ms.

Al receptor de la solicitud enviada por cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.

- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Modelo Cliente-Servidor

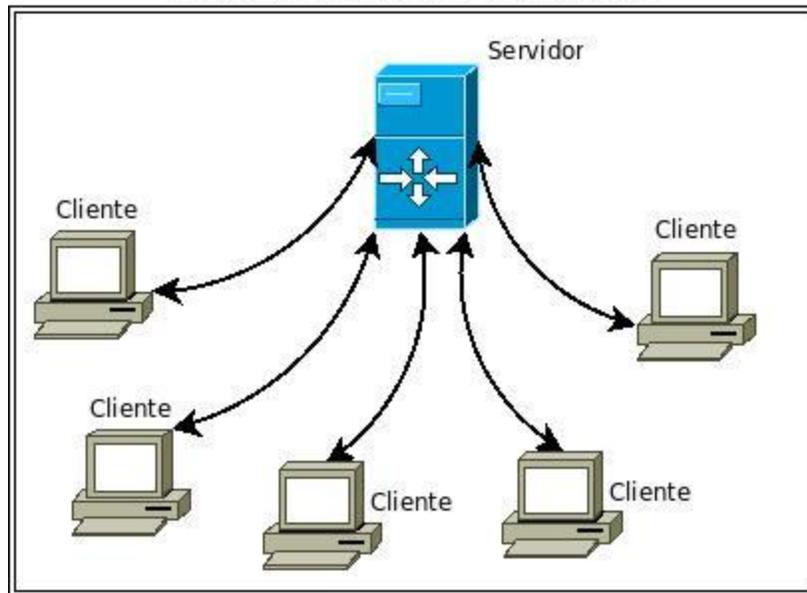


Fig. 5.1 – Arquitectura Cliente – Servidor

5.3 Sockets

Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

Un *socket* queda definido por una dirección IP, un protocolo de transporte y un número de puerto.

Para que dos programas puedan comunicarse entre sí es necesario que se cumplan ciertos requisitos:

- Que un programa sea capaz de localizar al otro.
- Que ambos programas sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad.

Para ello son necesarios los tres recursos que originan el concepto de *socket*:

- Un protocolo de comunicaciones, que permite el intercambio de octetos.

- Una dirección del Protocolo de Red (Dirección IP, si se utiliza el Protocolo TCP/IP), que identifica una computadora.
- Un número de puerto, que identifica a un programa dentro de una computadora.

Los *sockets* permiten implementar una arquitectura cliente-servidor. La comunicación ha de ser iniciada por uno de los programas que se denomina programa *cliente*. El segundo programa espera a que otro inicie la comunicación, por este motivo se denomina programa *servidor*.

Un socket es un fichero existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

Las propiedades de un *socket* dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es Transmission Control Protocol, aunque también es posible utilizar UDP o IPX.

Cuando se implementan con el protocolo TCP, los sockets tienen las siguientes propiedades:

- Orientado a conexión.
- Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
- Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.

Estas propiedades son muy importantes para garantizar la corrección de los programas que tratan la información.

El protocolo UDP es un protocolo *no* orientado a la conexión. Sólo se garantiza que si un mensaje llega, llegue bien. En ningún caso se garantiza que llegue o que lleguen todos los mensajes en el mismo orden que se mandaron. Esto lo hace adecuado para el envío de mensajes frecuentes pero no demasiado importantes, como por ejemplo, mensajes para los refrescos (actualizaciones) de un gráfico.

5.4 Puerto de comunicación

En la informática, un puerto es una forma genérica de denominar a una interfaz a través de la cual los diferentes tipos de datos se pueden enviar y recibir. Dicha interfaz puede ser de tipo físico, o puede ser a nivel de software (por ejemplo, los puertos que permiten la transmisión de datos entre diferentes ordenadores), en cuyo caso se usa frecuentemente el término puerto lógico.

Se denomina puerto lógico a una zona, o localización, de la memoria de un ordenador que se asocia con un puerto físico o con un canal de comunicación, y que proporciona un espacio para el almacenamiento temporal de la información que se va a transferir entre la localización de memoria y el canal de comunicación.

En el ámbito de Internet, un puerto es el valor que se usa, en el modelo de la capa de transporte, para distinguir entre las múltiples aplicaciones que se pueden conectar al mismo host, o puesto.

Aunque muchos de los puertos se asignan de manera arbitraria, ciertos puertos se asignan, por convenio, a ciertas aplicaciones particulares o servicios de carácter universal. De hecho, la IANA (Internet Assigned Numbers Authority) determina, las asignaciones de todos los puertos comprendidos entre los valores [0, 1023], (hasta hace poco, la IANA sólo controlaba los valores desde el 0 al 255). Por ejemplo, el servicio de conexión remota telnet, usado en Internet se asocia al puerto 23. Por tanto, existe una tabla de puertos asignados en este rango de valores. Los servicios y las aplicaciones que se encuentran en el listado denominado Selected Port Assignments. De manera análoga, los puertos numerados en el intervalo [1024, 65535] se pueden registrar con el consenso de la IANA, vendedores de software y organizaciones. Por ejemplo, el puerto 1352 se asigna a Lotus Notes.

5.5 Protocolo de comunicación

En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales. En su forma más simple, un protocolo puede ser definido como las reglas que dominan la sintaxis, semántica y

sincronización de la comunicación. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos. A su más bajo nivel, un protocolo define el comportamiento de una conexión de hardware.

Los protocolos son reglas de comunicación que permiten el flujo de información entre equipos que manejan lenguajes distintos, por ejemplo, dos computadores conectados en la misma red pero con protocolos diferentes no podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma. El protocolo TCP/IP fue creado para las comunicaciones en Internet. Para que cualquier computador se conecte a Internet es necesario que tenga instalado este protocolo de comunicación.

- Estrategias para mejorar la seguridad (autenticación, cifrado).
- Cómo se construye una red física.
- Cómo los computadores se conectan a la red.

Si bien los protocolos pueden variar mucho en propósito y sofisticación, la mayoría especifica una o más de las siguientes propiedades:

- Detección de la conexión física subyacente (con cable o inalámbrica), o la existencia de otro punto final o nodo.
- *Handshaking*.
- Negociación de varias características de la conexión.
- Cómo iniciar y finalizar un mensaje.
- Procedimientos en el formateo de un mensaje.
- Qué hacer con mensajes corruptos o formateados incorrectamente (corrección de errores).
- Cómo detectar una pérdida inesperada de la conexión, y qué hacer entonces.
- Terminación de la sesión y/o conexión.
- En el campo de las redes informáticas, los protocolos se pueden dividir en varias categorías, una de las clasificaciones más estudiadas es la OSI.
- Según la clasificación OSI, la comunicación de varios dispositivos ETD se puede estudiar dividiéndola en 7 niveles, que son expuestos desde su nivel más alto hasta el más bajo:

Nivel	Nombre	Categoría
Capa 7	Nivel de aplicación	Aplicación
Capa 6	Nivel de presentación	
Capa 5	Nivel de sesión	
Capa 4	Nivel de transporte	
Capa 3	Nivel de red	Transporte de datos
Capa 2	Nivel de enlace de datos	
Capa 1	Nivel físico	

Tabla 1 – Modelo OSI

- A su vez, esos 7 niveles se pueden subdividir en dos categorías, las capas superiores y las capas inferiores. Las 4 capas superiores trabajan con problemas particulares a las aplicaciones, y las 3 capas inferiores se encargan de los problemas pertinentes al transporte de los datos.
- Otra clasificación, más práctica y la apropiada para TCP/IP, podría ser esta:

Nivel
Capa de aplicación
Capa de transporte
Capa de red
Capa de enlace de datos
Capa física

Tabla 2 – Modelo TCP/IP

- Los protocolos de cada capa tienen una interfaz bien definida. Una capa generalmente se comunica con la capa inmediata inferior, la inmediata superior, y la capa del mismo nivel en otros computadores de la red. Esta división de los protocolos ofrece abstracción en la comunicación.
- Una aplicación (capa nivel 7) por ejemplo, solo necesita conocer cómo comunicarse con la capa 6 que le sigue, y con otra aplicación en otro computador (capa 7). No necesita conocer nada entre las capas de la 1 y la

5. Así, un navegador web (HTTP, capa 7) puede utilizar una conexión Ethernet o PPP (capa 2) para acceder a la Internet, sin que sea necesario cualquier tratamiento para los protocolos de este nivel más bajo. De la misma forma, un router sólo necesita de las informaciones del nivel de red para enrutar paquetes, sin que importe si los datos en tránsito pertenecen a una imagen para un navegador web, un archivo transferido vía FTP o un mensaje de correo electrónico.

Ejemplos de protocolos de red

- Capa 1: Nivel físico
 - Cable coaxial o UTP categoría 5, categoría 5e, categoría 6, categoría 6a Cable de fibra óptica, Cable de par trenzado, Microondas, Radio, RS-232.
- Capa 2: Nivel de enlace de datos
 - Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI, ATM, HDLC.,cdp
- Capa 3: Nivel de red
 - ARP, RARP, IP (IPv4, IPv6), X.25, ICMP, IGMP, NetBEUI, IPX, Appletalk.
- Capa 4: Nivel de transporte
 - TCP, UDP, SPX.
- Capa 5: Nivel de sesión
 - NetBIOS, RPC, SSL.
- Capa 6: Nivel de presentación
 - ASN.1.
- Capa 7: Nivel de aplicación
 - SNMP, SMTP, NNTP, FTP, SSH, HTTP, SMB/CIFS, NFS, Telnet, IRC, POP3, IMAP, LDAP.
- Capa 8: Nivel de correccion
 - CRC.

6. ANALISIS DEL SISTEMA

A esta actividad se le dio una mayor importancia sobre las demás ya que en esta actividad se determinan las necesidades y componentes del sistema que se va a realizar, por otro lado se desarrollaron diferentes diagramas para poder conocer y comprender el funcionamiento del sistema así como los requisitos para que el software funcione a la perfección.

6.1 Diagrama de procesos

En este diagrama se determinó el funcionamiento general del software con el fin de darnos una idea general de la estructura del sistema. También fue de gran ayuda para determinar cómo se desarrollaría cada parte del sistema

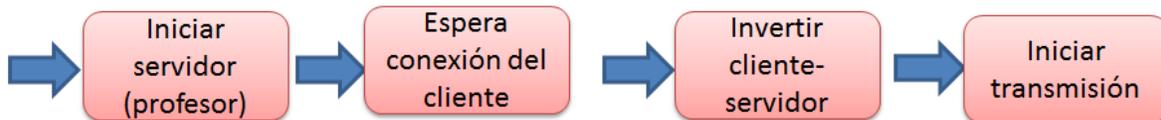


Fig. 6.1 – Procesos del servidor

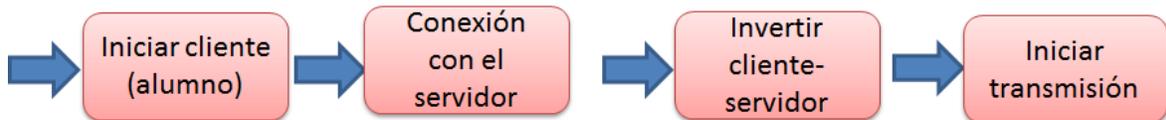


Fig. 6.2 – Procesos del cliente

Como se observa en la fig. 6.1 en las actividades realizadas por el servidor el profesor tiene que hacer las siguientes actividades:

- ✚ Iniciar el servidor
- ✚ El programa queda esperando las conexiones del cliente (alumnos)
- ✚ El programa de manera automática invierte al cliente y al servidor, es decir, ahora el profesor es el cliente
- ✚ El profesor da inicio a la transmisión de las diapositivas

En contraparte como se observa en la fig. 6.2 el cliente el alumno debe de realizar las siguientes actividades:

- ✚ Iniciar el cliente
- ✚ Realizar la conexión con el servidor
- ✚ El programa de manera automática invierte al cliente y al servidor, es decir, ahora el alumno es el servidor
- ✚ El programa de manera automática inicia la transmisión de la diapositiva enviada por el profesor

6.2 Diagrama de procesos:



Fig. 6.3 – Diagrama de actividades

En la figura 6 del diagrama de actividades se explica de manera más detallada cada una de las actividades que el sistema hace de manera interna.

6.3 Diagrama de casos de uso:



Fig. 6.4 – Diagrama de casos de

- ✚ Al presionar el botón de abrir puerto tanto en el cliente como en el servidor, el programa abre la consola del sistema y abre el puerto de comunicación para poder lograr la ejecución del programa
- ✚ Al presionar el botón iniciar programa el programa cliente obtiene su IP y se abre una ventana que indica el No. De conexiones que se realizaran.
- ✚ Cuando se indican las conexiones, el programa se queda esperando las conexiones y una vez completadas ese número de conexiones el programa empieza a correr.
- ✚ Cuando el servidor presiona el botón ejecutar programa se abre una ventana en donde debe de indicar la IP proporcionada por el profesor.
- ✚ Después de introducir la IP del profesor el servidor presiona el botón enviar IP para realizar la conexión con el cliente para poder iniciar la el bloqueo del equipo.

6.4 Diagrama de estados:

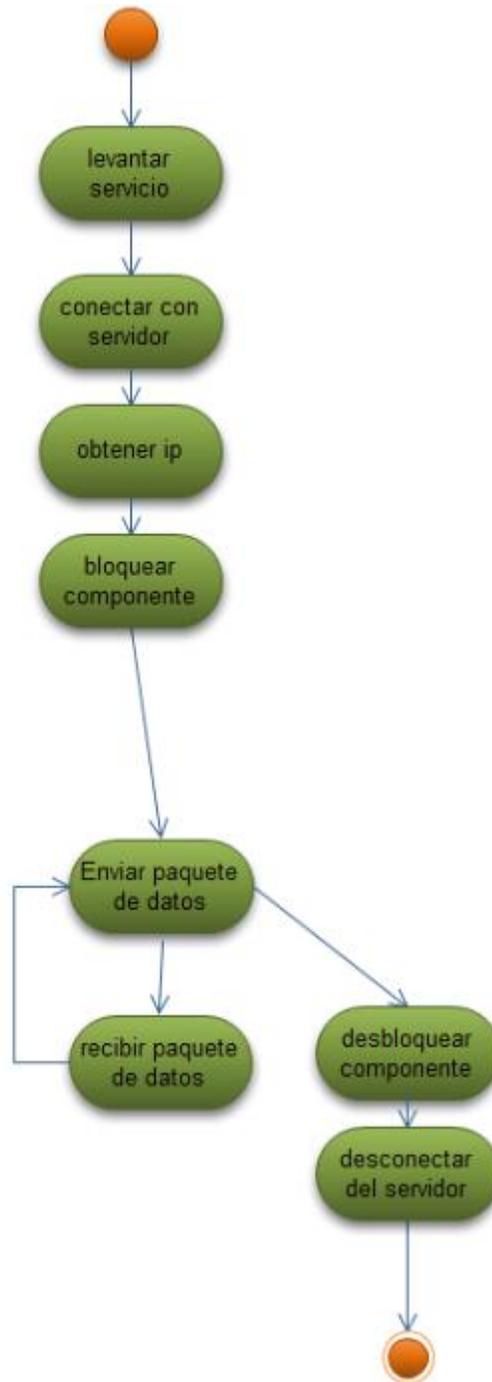


Fig. 6.5 – Diagrama de estados

7. DISEÑO Y DESARROLLO DE LA HERRAMIENTA

7.1 Diagrama de clases

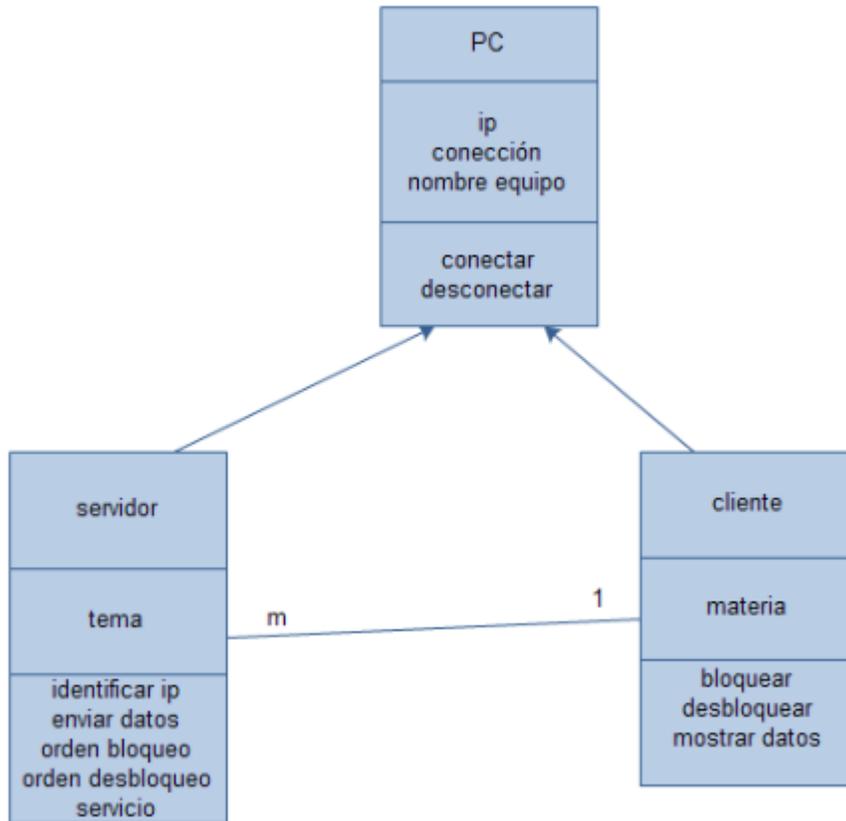


Fig. 7.1 – Diagrama de Clases

Como se observa en la fig. 12, aquí definimos la estructura del sistema basándonos en la arquitectura cliente-servidor, utilizando la técnica para el acceso de objetos remotos proporcionada por java y denominada java RMI en la cual se genera una clase cliente, una clase servidor y una especie de proxy que se sitúa en medio de las dos clases la cual permite la comunicación entre las diferentes clases.

7.2 Bloqueo de teclado

Para bloquear el teclado del equipo de cómputo utilizamos el lenguaje de programación C++ con la finalidad de poder implementar un hook, este programa

deshabilita los eventos del teclado, es decir, al presionar las teclas el programa las detecta y anula la instrucción.

Para poder habilitar el teclado se generó una pequeña interfaz en la cual al hacer un simple clic se termina el proceso y se puede volver a utilizar el teclado de manera normal.

7.2 Bloqueo de mouse

El bloqueo del mouse no se pudo realizar debido a que si bloqueamos el mouse no podemos simular los eventos del teclado para poder avanzar las diapositivas, por tal motivo este objetivo no se cumple por el momento quedando para una actualización al sistema.

7.2 Transmisión de diapositivas

Para proyectar las presentaciones del docente hacia los equipos de los alumnos utilizamos un pequeño programa en java en el cual indicamos la una ruta especifica en donde se encuentran previamente cargadas las diapositivas y con una instrucción para poder abrir ese archivo.

La instrucción para abrir el archivo es la siguiente:

```
Desktop.getDesktop().open(new File(ruta1));
```

8. CASOS DE PRUEBAS

Caso de prueba 1:

Una vez teniendo cada uno de los elementos anteriormente mencionados los unimos de manera que se realiza en modo consola para realizar las primeras pruebas y las posibles correcciones, una vez superada la primera etapa de prueba se realizaron las interfaces tanto para el cliente como para el servidor, las cuales se muestran a continuación.

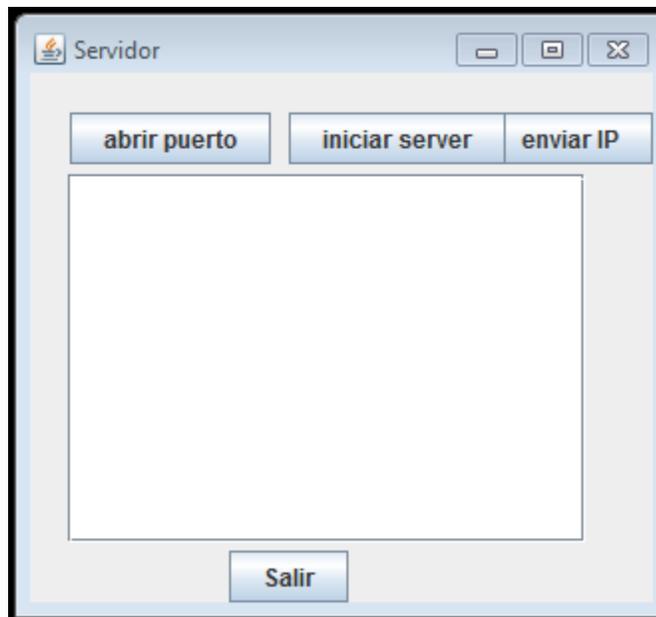


Fig. 8.1 – Interfaz del servidor

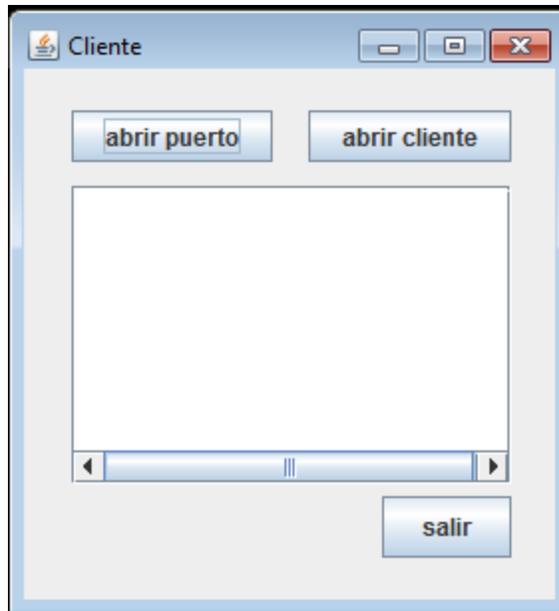


Fig. 8.2 – interfaz del cliente

En esta serie de pruebas se pudo realizar una conexión entre el cliente y el servidor pero no se pudo realizar el envío de las direcciones IP de los equipos de cómputo por lo cual se tuvieron que realizar pequeñas demoras al sistema.

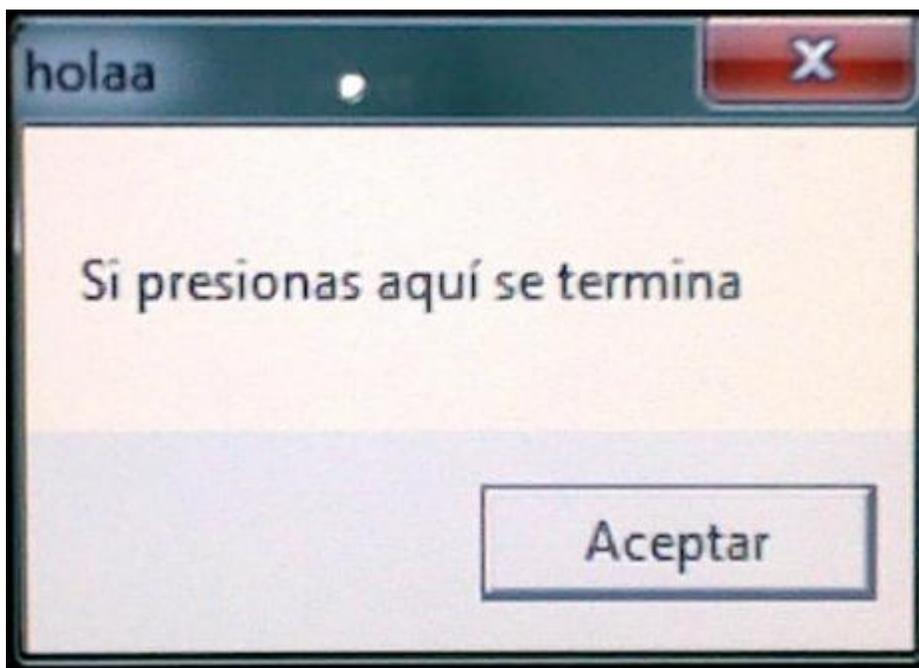


Fig. 8.3 – Interfaz para desbloquear el teclado

Como se observa en la figura 8.3 se obtiene una pequeña interfaz para poder desbloquear el teclado utilizando la siguiente sentencia para lograrlo:

```
HHOOK keyboardHook =  
SetWindowsHookEx(WH_KEYBOARD_LL, HookKey, hInstance, 0);  
    MessageBox(NULL, "Si presionas aquí se termina", "holaa", MB_OK);
```

Caso de prueba 2:

Después de haber realizado la mejora a la interfaz se realizaron pruebas para probar las correcciones realizadas las cuales se muestran a continuación:

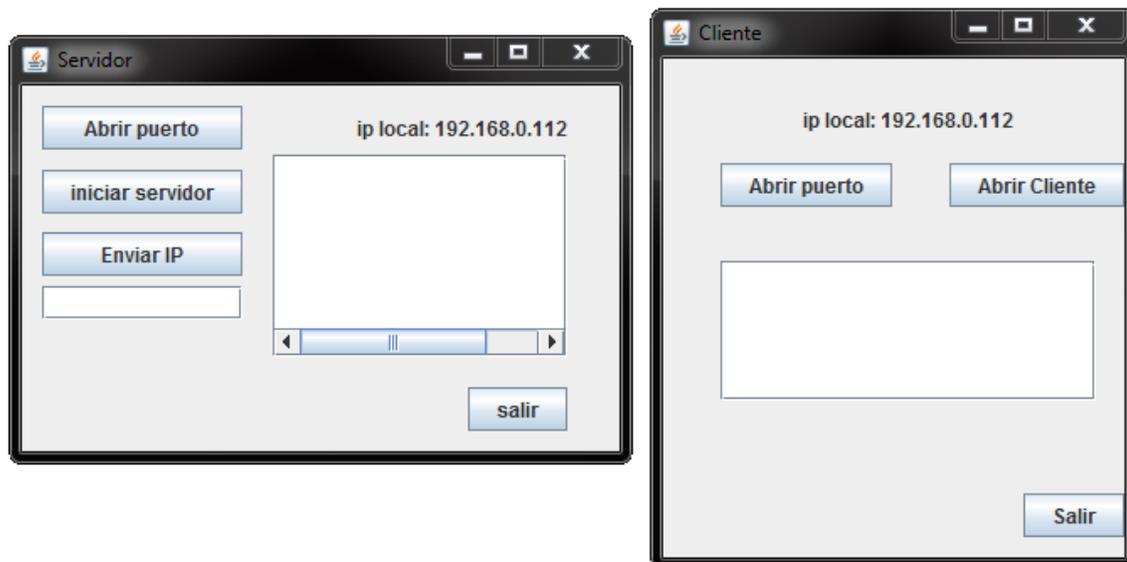


Fig. 8.4 – Interfaz del cliente y servidor

9. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha observado en los casos de prueba que se cumplen con los objetivos 1 y 3 quedando pendiente el objetivo 2 debido a que si se bloquea el mouse el profesor no podrá tener control de las diapositivas.

Al utilizar este sistema es posible combatir las posibles deficiencias visuales que puedan presentar los alumnos, también se brinda la misma calidad de la información a todos debido a que todos los alumnos tendrán en primer plano las diapositivas.

Cabe destacar que el sistema cumple con el objetivo general y cumple con dos de los tres objetivos específicos, esto es debido a que como el bloqueo del mouse se hace a nivel interrupciones del sistema al realizarlo nos resultaría imposible poder emular de manera remota los eventos del mouse, por tal motivo se optó por no deshabilitar este componente.

De manera general el sistema es totalmente funcional con el sistema operativo Windows ya que para poder bloquear el teclado mandamos llamar un archivo con extensión .exe el cual solo corre bajo la plataforma Windows, aunque la transmisión de las diapositivas se puede realizar en cualquier sistema operativo en el cual este instalado Java, esto es debido a que el archivo ejecutable con extensión .exe solo funciona en Windows.

A modo de sugerencia como posibles actualizaciones el sistema podrá contar con diferentes módulos para hacer más dinámica la clase al existir mayor interacción entre el alumno y el profesor, por otro lado el profesor podrá evaluar el desarrollo de la clase.

10. BIBLIOGRAFIA

- ✚ Deitel, Paul J. Y Harvey Deitel, Como Programar en Java, Editorial Pearson Educación.
- ✚ Casad, Joe y Willsey Bod, Aprendiendo TCP/IP en 24 horas, Editorial Prentice-Hall.
- ✚ López, José, Domine PHP y MySQL, Editorial, Alfa omega grupo editor.

Referencias de internet

- ✚ <http://etcandelariacomunica.files.wordpress.com/2009/02/proyectores-de-video.pdf>
- ✚ <http://es.wikipedia.org/wiki/NetBus>
- ✚ <http://es.wikipedia.org/wiki/VNC>
- ✚ http://es.wikipedia.org/wiki/Java_Remote_Method_Invocation

11. ANEXOS

Anexo A: Imágenes de una clase en el laboratorio de sistemas:



Fig. Anexo A.1 – Clase en el laboratorio del ITTG



Fig. Anexo A.2 – Clase en el laboratorio del ITTG

Anexo B: Manual de usuario

Instalar Java

Guía para instalar y configurar Java y sus variables de entorno PATH y CLASSPATH en Windows XP y Linux

¿Qué se necesita para instalar java?

En la página de Sun podemos encontrar varios tipos de descarga. Si sólo queremos ejecutar aplicaciones Java, necesitaremos instalar JRE (Java Runtime Environment). JRE es un entorno de ejecución que sirve para poder ejecutar programas java ya hechos. Útil (por ejemplo) si queremos que nos funcionen los applets de las webs.

Por el contrario, si lo que queremos es crear, compilar y ejecutar nuestras aplicaciones, necesitaremos el SDK (Sun Development Kit). SDK es el entorno de desarrollo que incluye varias herramientas además del compilador de java y el debugger.

También podemos descargarnos el SDK+netbeans si queremos un entorno visual para desarrollar nuestras aplicaciones. Netbeans es especialmente útil para diseñar interfaces gráficas para usuarios.

Puedes descargar estas aplicaciones en la página de sun. Las descargas están agrupadas por Java EE (Enterprise Edition: Desarrollo de aplicaciones web) , Java ME (Micro Edition: Desarrollo para dispositivos móviles y PDA's) y Java SE (Standard Edition: Nos decantaremos por esta si queremos crear aplicaciones normales).

Configurar Java

Una vez instalada la versión SDK (Java SE), deberemos configurar las variables de entorno para poder ejecutar aplicaciones JAVA desde cualquier ruta de tu sistema.

Windows 2000/XP: pulsaremos el botón derecho del ratón sobre MI PC, y seleccionaremos propiedades. En la pestaña de Opciones Avanzadas, hay un botón que te permite configurar las "Variables de Entorno". Haz click en este botón y modifica la variable "Path" del sistema, de modo que también contenga la ruta del ejecutable Java. Por ejemplo: si has instalado Java en c:\Java y la variable del sistema tiene el valor C:\WINDOWS\SYSTEM32, le cambiaremos el valor para que se lea: C:\WINDOWS\SYSTEM32;C:\Java\bin . Ahora, desde la línea de comandos podrás compilar (Ej: javac Miprograma.java) y ejecutar (Ej: java Miprograma) tus aplicaciones Java desde cualquier carpeta del sistema.

Linux: Deberás cambiar el valor de la variable del sistema PATH para que apunte a la ruta donde se han instalado los binarios. Si usas bash como shell, deberías poner la siguiente línea al final del archivo .bashrc: export PATH=/ruta/a/java:\$PATH.

Configurar la variable CLASSPATH

Además de configurar la variable de entorno PATH, también deberás decirle a Java dónde encontrar los archivos de clases compiladas al ejecutar tus aplicaciones. Para ello deberemos modificar la variable CLASSPATH para que al menos incluya el directorio actual (.)

Ej: Desde la línea de comandos escribiremos:

```
SET CLASSPATH=%CLASSPATH%;.
```

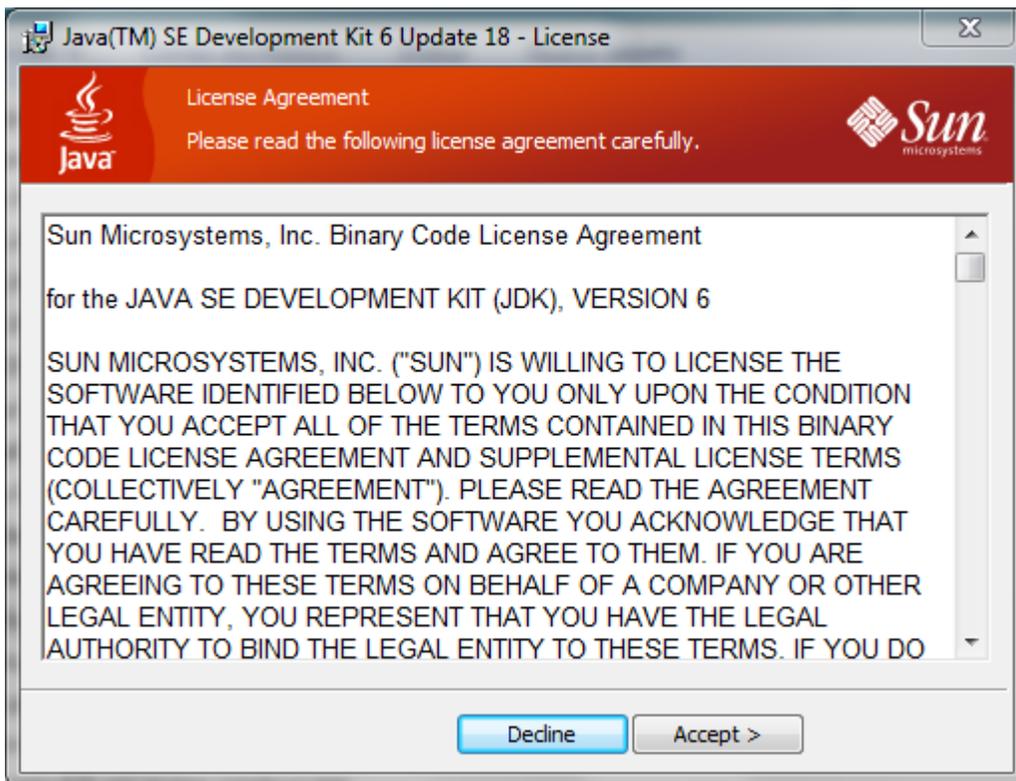
Ésta variable también puede contener otros directorios que contengan clases compiladas.

El tutorial sigue siendo el mismo, claro algunas cosas pueden cambiar, veamos el tutorial:

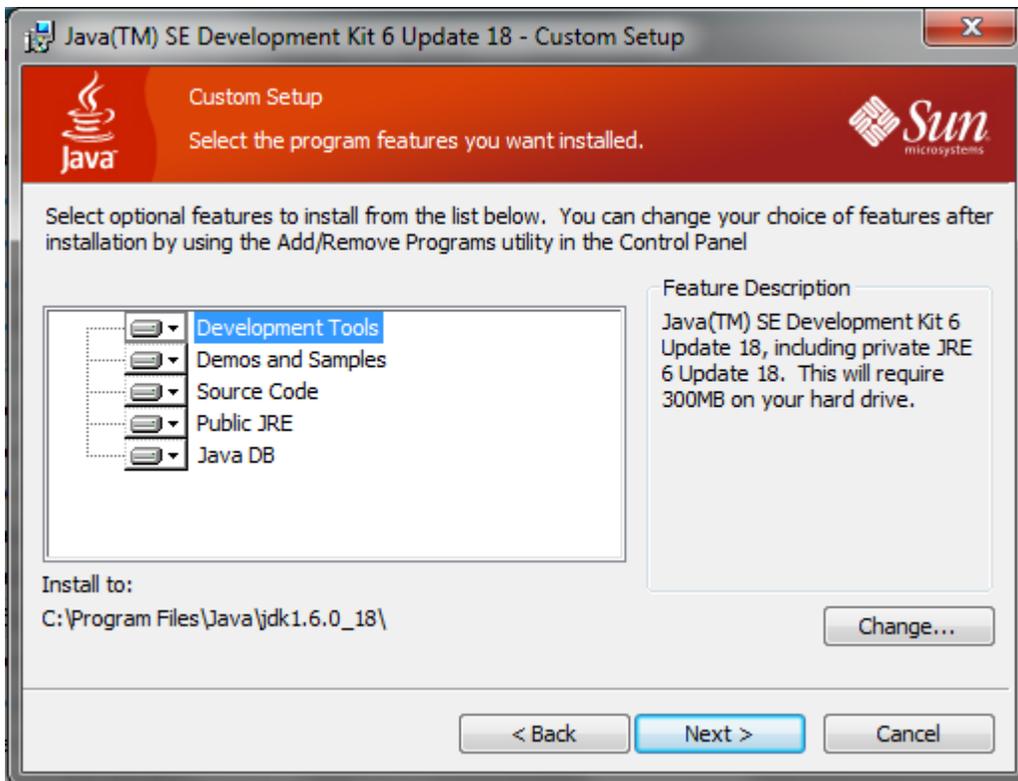
Antes de empezar necesitamos descargar el Java Development Kit o mejor conocido como el Java JDK desde la página de Java o mejor dicho de la página de Sun.

Descargar JDK – Java Development Kit

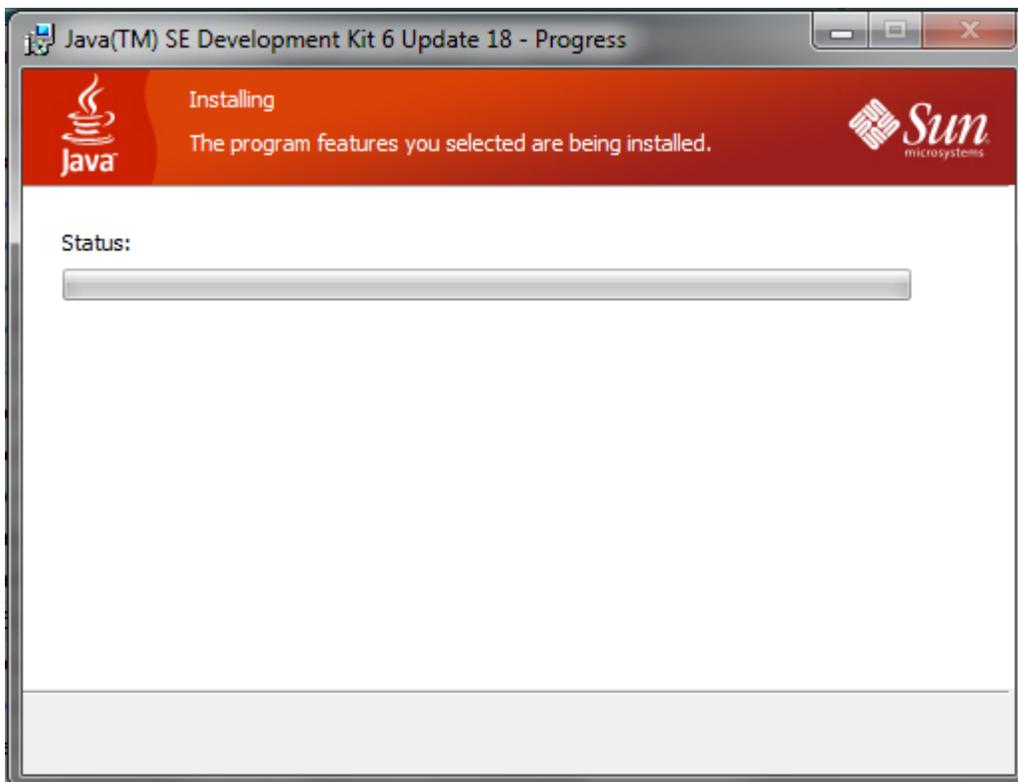
Una vez descargado dicho archivo, le damos doble clic para que empiece la instalación del mismo. Y nos aparecerá una pantalla como esta:



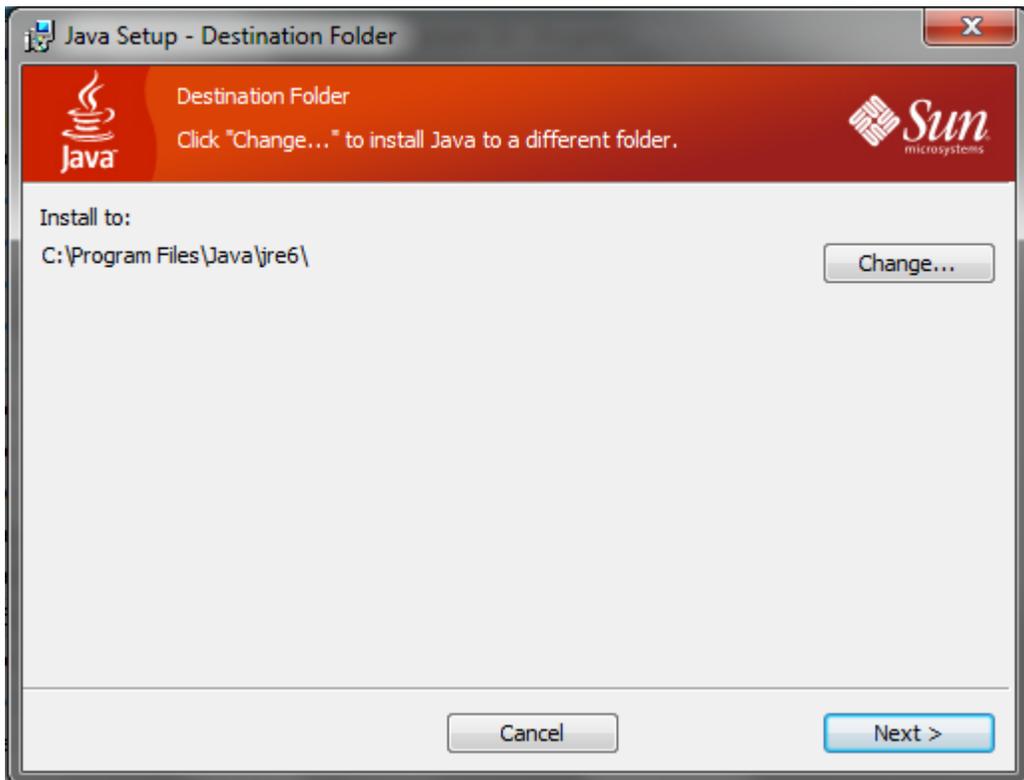
“Leemos” el contrato de Usuario Final, y le damos Clic en Accept >



Si queremos cambiar la capeta de instalación le podemos dar clic en Change... e instalarla donde quiera, Yo la dejé en la ruta por default. Y después le damos Next.



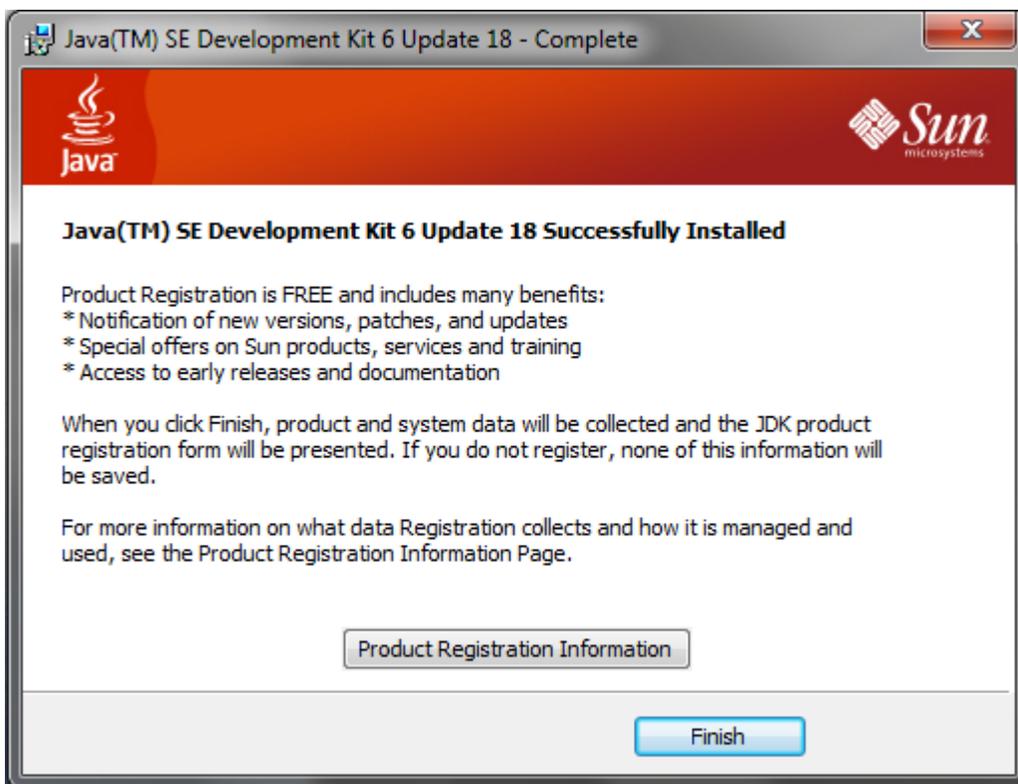
Empezará la instalación del Java JDK y tendremos que esperar unos minutos, dependiendo del rendimiento de tu equipo de computo.



Al terminar la instalación del Java JDK, nos pedirá la carpeta de destino donde se tiene que instalar el JRE de Java, este es importante instalarlo, para que se ejecuten los programas hecho en Java. Yo dejé la ruta por default. Damos Clic en Next >.



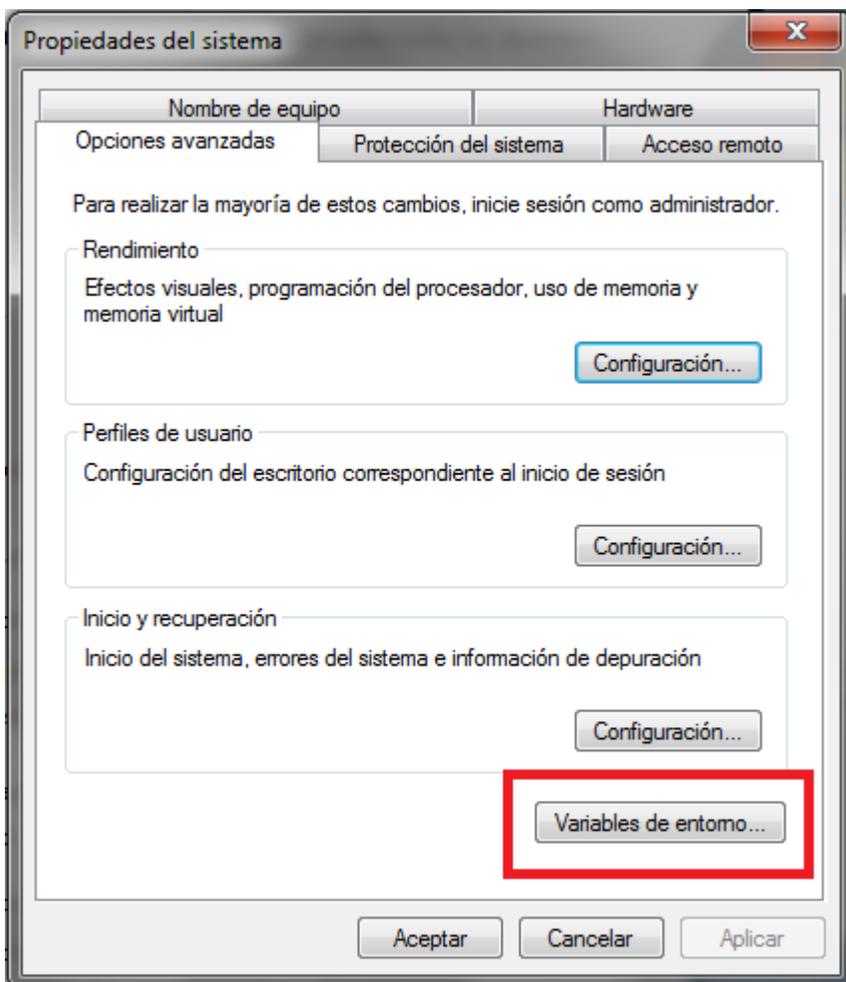
Empezará la instalación de Java JRE y dependerá del funcionamiento de nuestro equipo, cuanto dure esta instalación, por lo general dura unos pocos minutos.



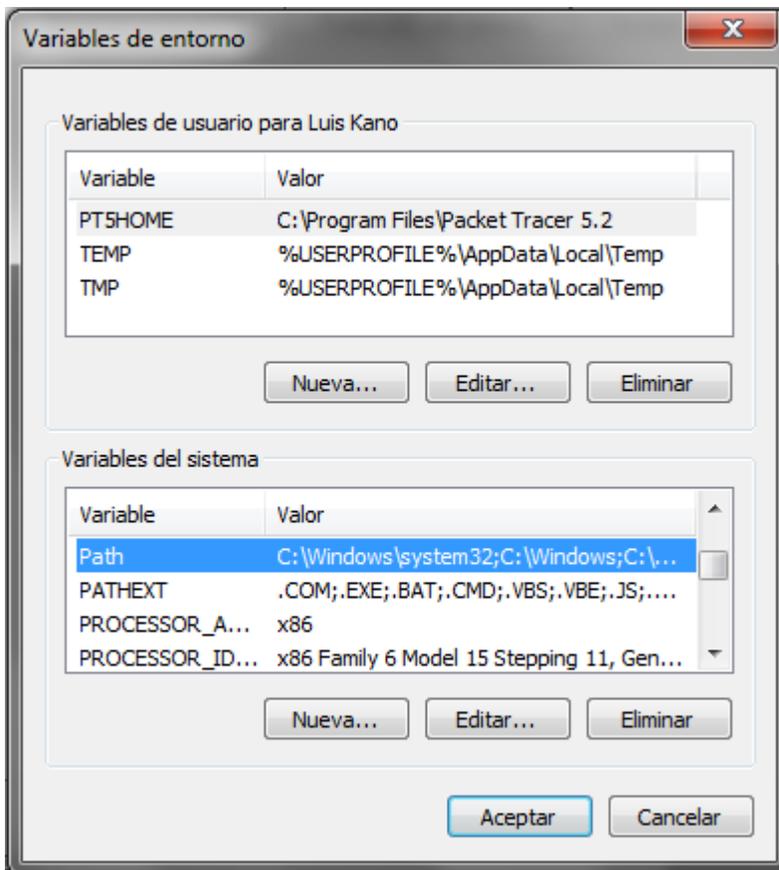
Una vez que haya terminado la instalación del Java JRE, le daremos clic en Finish y terminará la instalación de Java JDK y Java JRE.

Todavía aquí no termina esto, si no que hay que configurar las variables de entorno para que podamos “compilar” desde la Consola de Comandos de Windows 7.

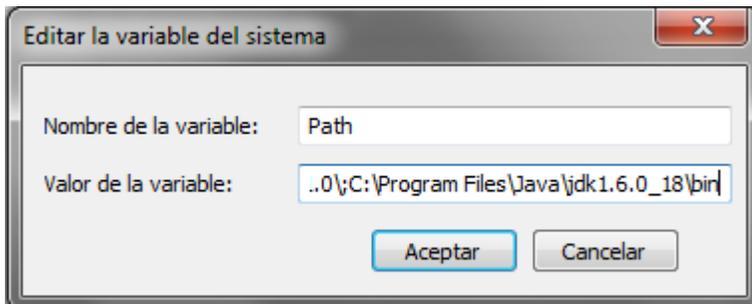
Damos Clic Derecho a Mi PC y nos vamos a Propiedades. Y nos aparecerá una ventana donde aparece la información de nuestro sistema, Damos clic en Configuración Avanzada del Sistema, que se encuentra a la izquierda. Y nos debe de aparecer una ventana como esta:



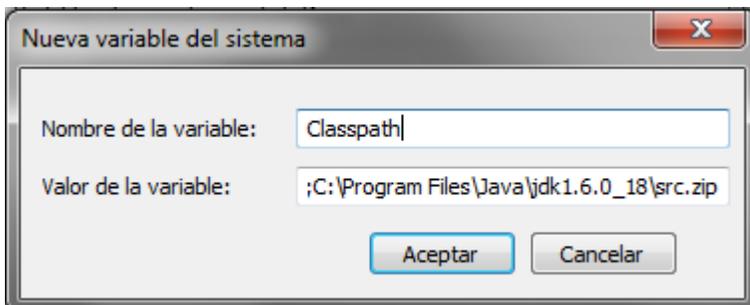
Nos vamos a la pestaña de Opciones Avanzadas y damos clic en Variables de Entorno



Buscamos en las Variables del sistema, la variable llamada Path y le damos clic en Editar



Se abrirá una nueva ventana y le agregamos esto: ;C:\Program Files\Java\jdk1.6.0_18\bin si es que dejamos la ruta por default del instalador, si lo metiste en otra carpeta, deberás poner la dirección donde se encuentra instalado el JDK. Y damos Clic en Aceptar.



Después ahí mismo en Variables del Sistema, crearemos una nueva variable llamada Classpath y tendrá la dirección de nuestro src.zip en este caso sería esta: ;C:\Program Files\Java\jdk1.6.0_18\src.zip y le damos clic en Aceptar.

Le damos Clic en Aceptar en Variables de Entorno y en la Ventana de Opciones Avanzadas.

Ahora nos iremos a la Consola de Comandos de Windows 7, o al famoso CMD, una vez abierto escribimos javac y deberá mostrar este contenido.

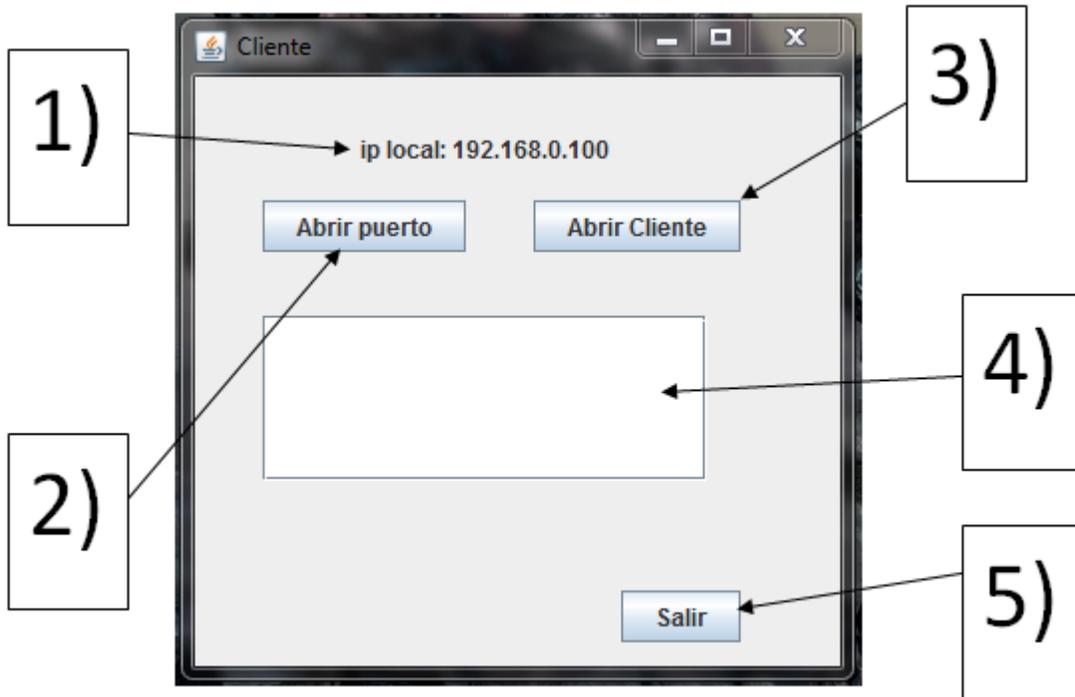
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. R

C:\Users\Luis Kano>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all de
  -g:none           Generate no de
  -g:<lines,vars,source>  Generate only s
  -nowarn           Generate no wa
  -verbose          Output messages
  -deprecation     Output source
sed
  -classpath <path>  Specify where
on processors
  -cp <path>         Specify where
on processors
  -sourcepath <path> Specify where
  -bootclasspath <path> Override locat
  -extdirs <dirs>   Override locat
  -endorseddirs <dirs> Override locat
  -proc:<none,only> Control whethe
ation is done.
  -processor <class1>[,<class2>,<class3>...
o run; bypasses default discovery process
```

Y si se muestra esto, quedó instalado el Java JDK en nuestro sistema de Windows 7, y podemos crear los programas y compilarlos desde la consola de comandos. O lo que es mejor, instalar un IDE como Netbeans o JCreator. En lo personal uso el Netbeans.

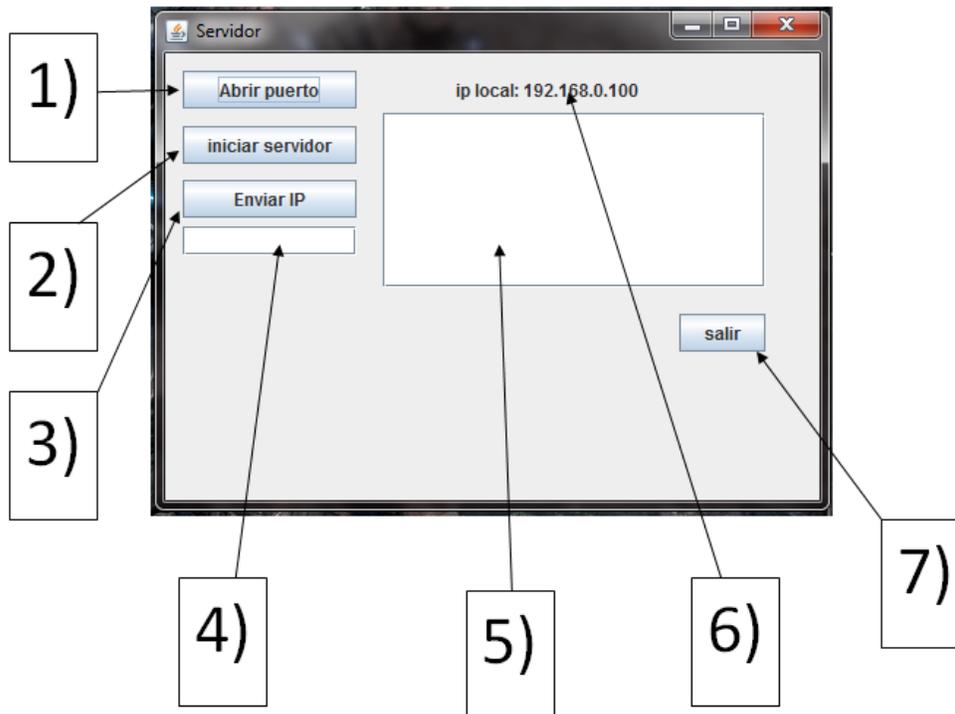
Espero les funcione este tutorial.

Cliente



- 1.- Muestra la ip local de la maquina. Esta puede ser asignada por la red local o configurarse para ip estática.
- 2.- Abre una consola de Windows y ejecuta el comando "rmiregistry 3000" con el cual registramos o "abrimos" el puerto de comunicación y llamada a los procedimientos remotos
- 3.- Abre una consola de Windows y se posiciona en la carpeta "c:\residencia-copia". Ejecuta la instrucción "java OperacionesClient" y se nos pedirá el número de conexiones entrantes que se permitirán.
- 4.- Área de texto donde se mostrara el estatus de la aplicación.
- 5.- cerrar la aplicación.

Servidor



- 1.- abre una consola de Windows, se posiciona en la ruta: "c:/residencia-copia" y ejecuta el comando "rmiregistry 3000" para abrir el puerto de comunicaciones.
- 2.- abre una consola de Windows y se posiciona en la ruta: "c:/residencia-copia" y ejecuta el comando "java OperacionesServer" el cual abre el servidor.
- 3.- envía la dirección ip de la maquina cliente anotada en el cuadro de texto(4) al cliente para que este se reconecte y ejecute las funciones programadas
- 4.- aquí es donde se debe de anotar una dirección ip del formato 255.255.255.255 la cual deberá corresponder a la dirección ip de la maquina en donde se esta ejecutando la aplicación "Cliente.jar".
- 5.- muestra el estatus de la aplicación.
- 6.- nos muestra la dirección ip correspondiente a la maquina en donde se ejecuta la aplicación "Servidor.jar".
- 7.- salir de la aplicación.

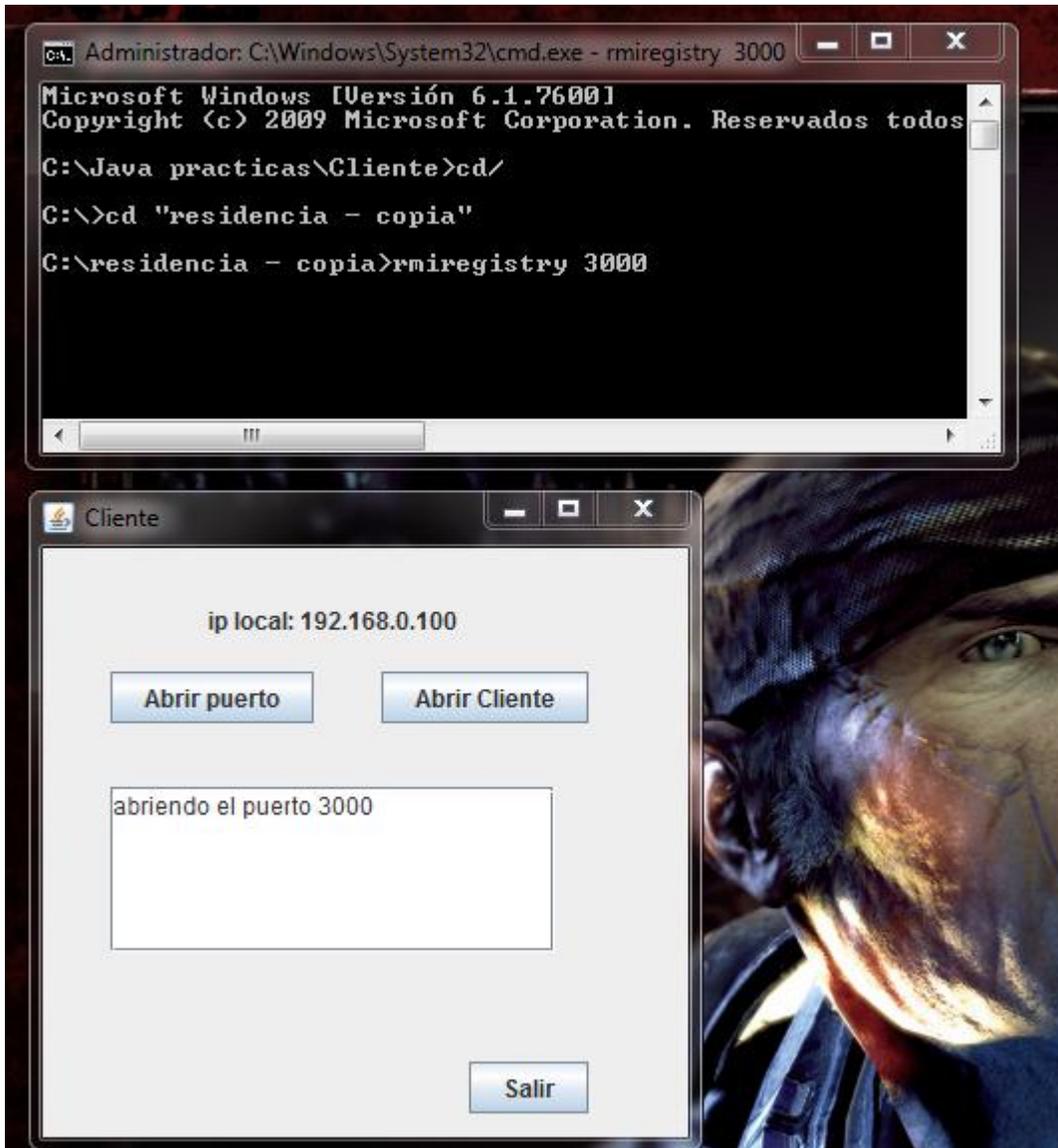
Instrucciones de ejecución

- 1.- verificar que se cuente con los archivos ".java" siguientes:

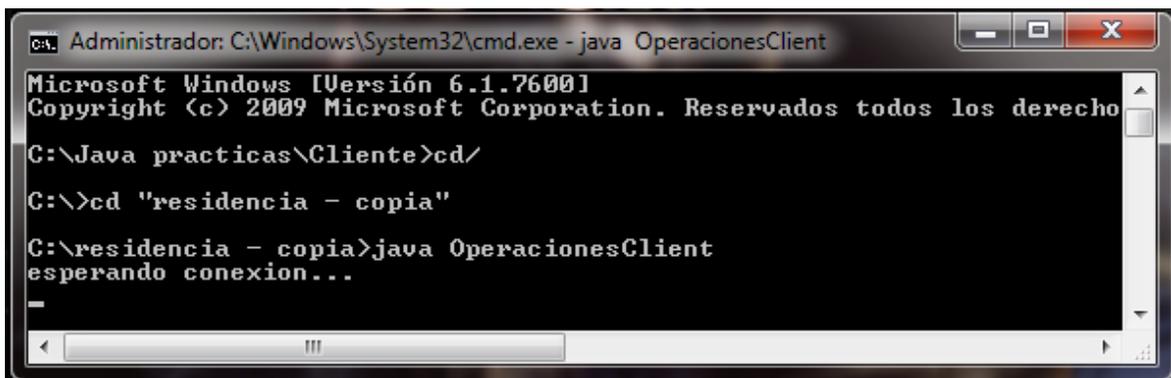
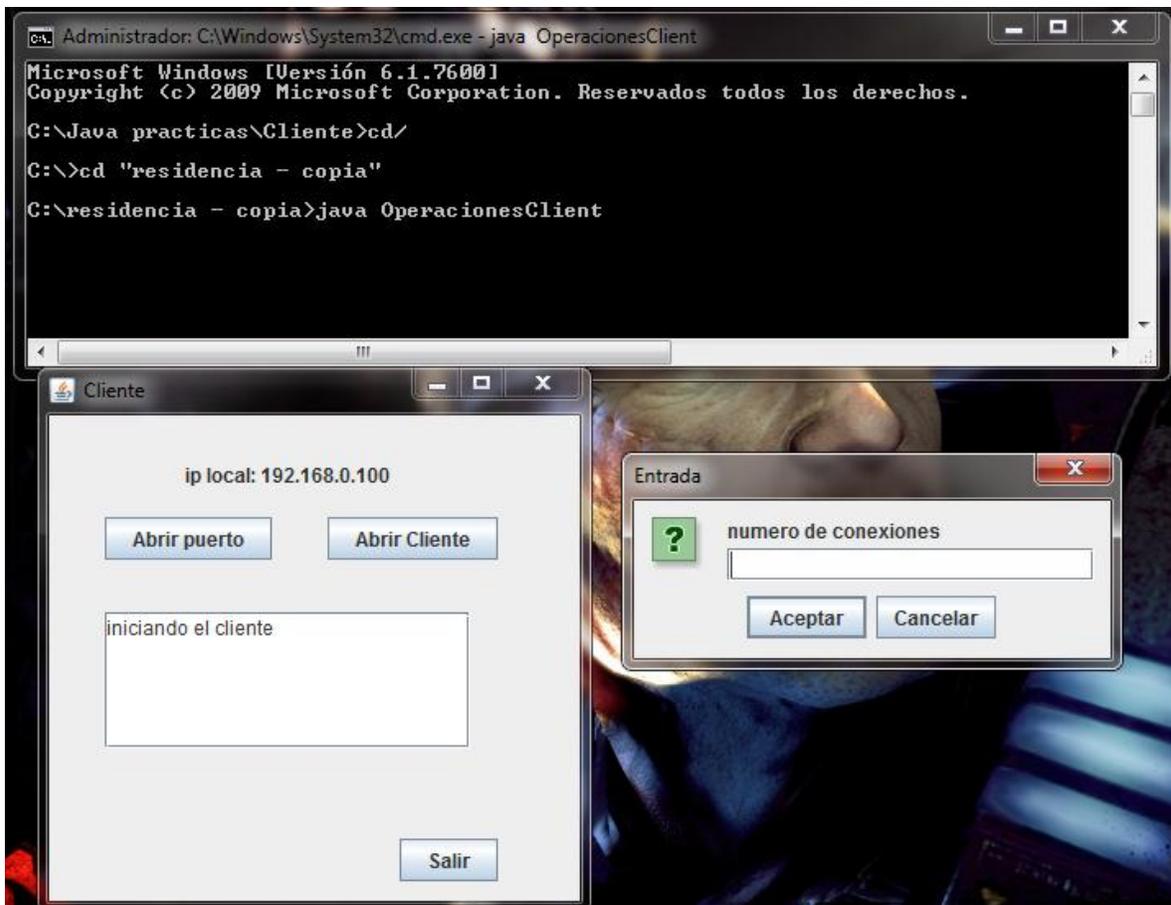
- 1.- Operaciones.java
- 2.- OperacionesClient.java
- 3.- OperacionesServer.java

2.- una vez nos cercioremos de que existan estos archivos y hayamos configurado nuestro classpath podremos ejecutar nuestras aplicaciones.

3.- en la aplicación Cliente se debe de oprimir el botón “abrir puerto” y nos aparecerá la siguiente captura de pantalla.

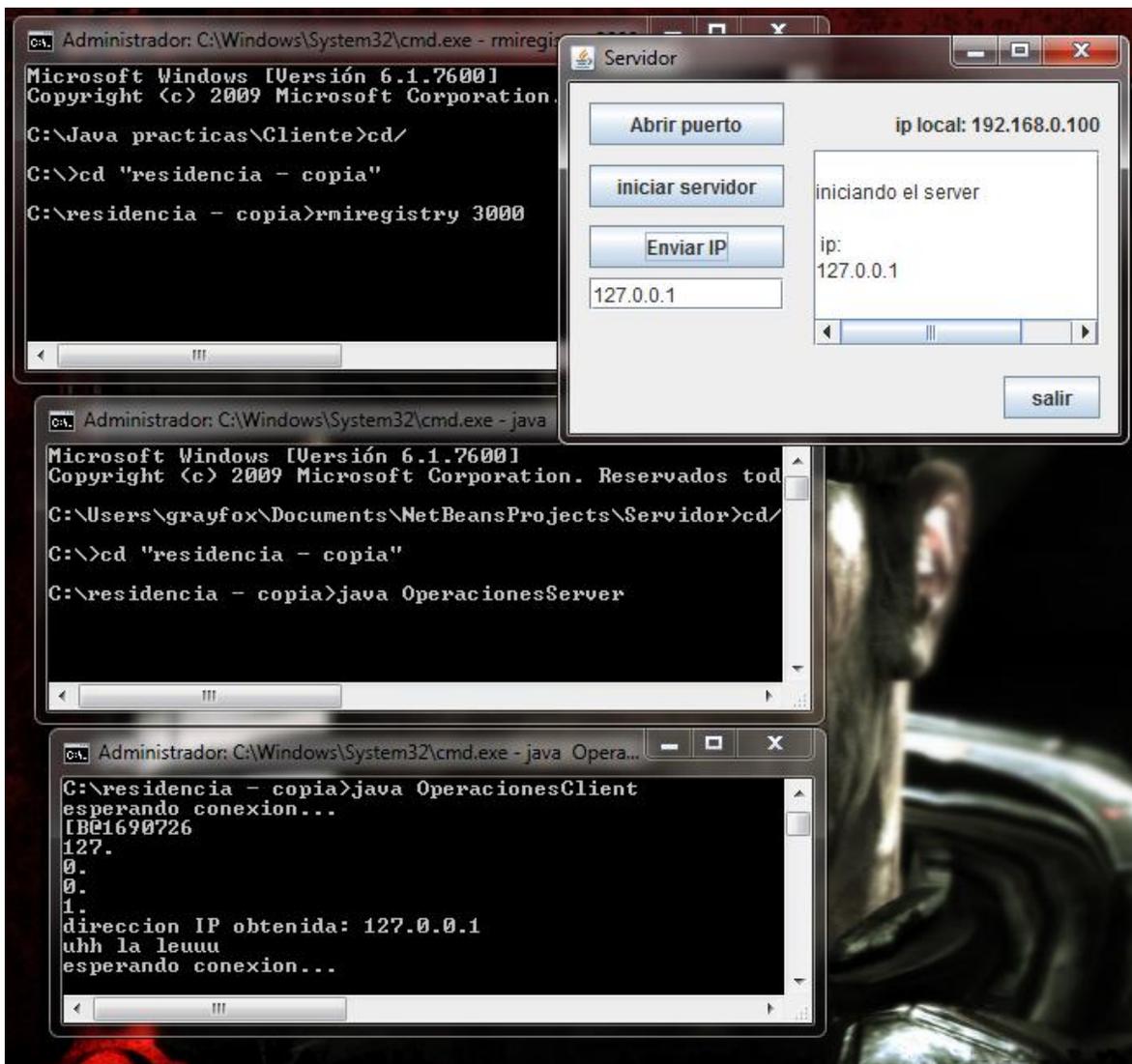


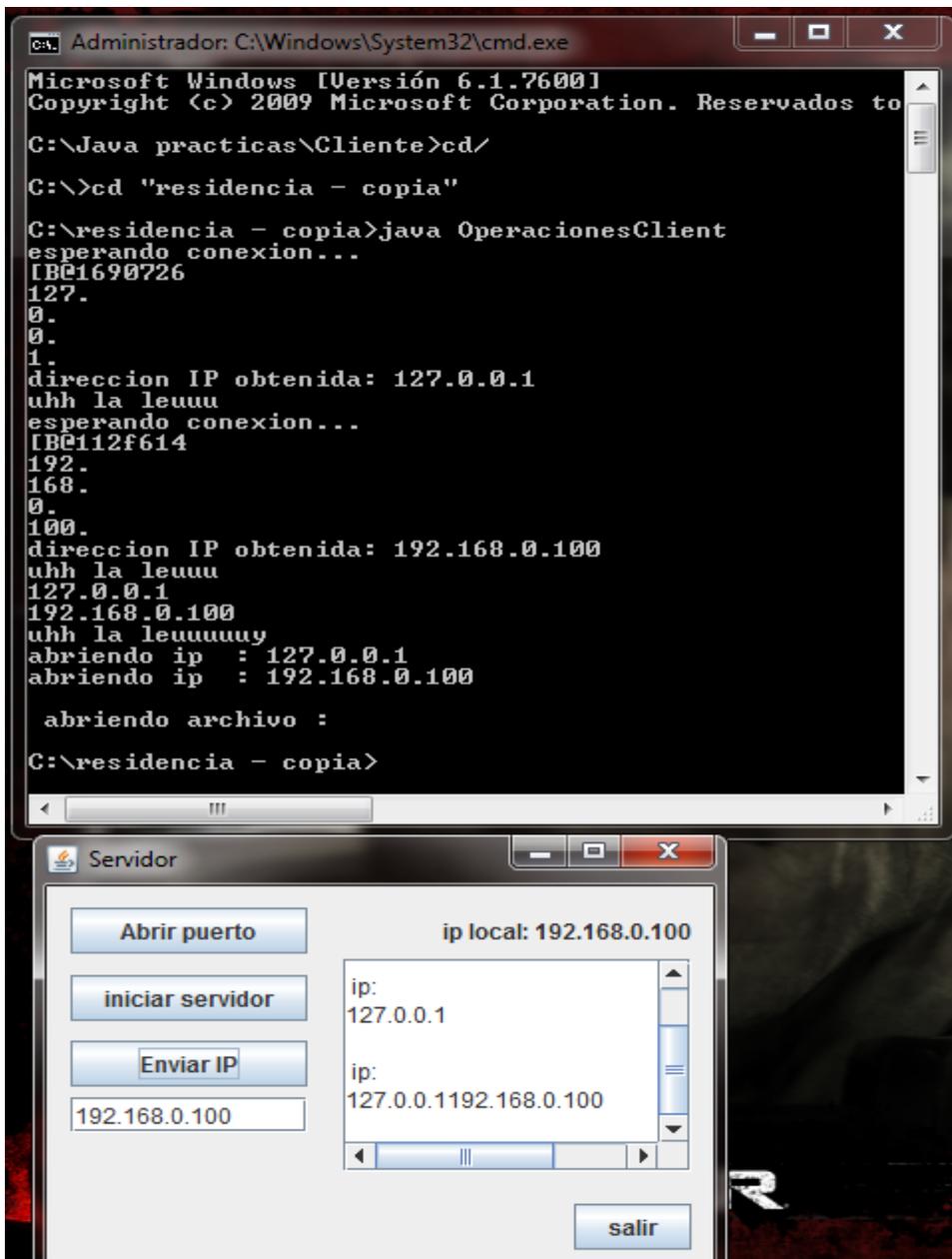
4.-oprimir el botón “ abrir cliente” pedirá el numero de direcciones ip que esperara para seguir con el siguiente paso.



En estas alturas de ejecución, la aplicación Cliente está a la escucha de servidores que se conecten. Después de que se hayan recibido todas las conexiones en espera, se ejecutará la conexión inversa, en donde la aplicación tomará el control de la máquina donde se ejecute la aplicación Servidor. Ahora por el lado del servidor se ejecutan las siguientes instrucciones:

- 1.- ejecutamos el archivo "Servidor.jar" y oprimimos el botón "abrir puerto" y ejecutara en una consola de Windows la instrucción "rmiregistry 3000".
- 2.- presionamos el botón "iniciar servidor" y ejecutara en otra consola la aplicación del servidor.
- 3.- ingresamos en el cuadro de texto la dirección ip de la maquina en donde se ejecuta la aplicación Cliente y después presionar el botón de "enviar ip" cuya función es tomar esa ip e intentar y conectarse a la maquina cliente
- 4.- una ves presionado el botón, la aplicación Cliente registrara la dirección ip y la almacenara para asi hacer una conexión hacia todos los server y abrir los archivos una ves recibidas todas las conexiones en espera.





Y después de todo esto... en la maquina Servidor, se deben de abrir los archivos .ppsx que contiene las diapositivas y un .exe que bloqueara el teclado.

Cualquier duda sobre este proyecto favor de contactarse a:

re3nemesi@hotmai.com

tigre687@hotmai.com