

ESCUELA SUPERIOR POLITECNICA DEL LITORAL



T
004.68
CISE
V.1

PROGRAMA DE TECNOLOGIA EN COMPUTACION

PROYECTO DE GRADUACION

PREVIO A LA OBTENCION DEL TITULO DE:

PROGRAMADOR EN ANALISIS DE SISTEMAS

TEMA : ESPOLCHAT v1.0

MANUAL DE DISEÑO ORIENTADO A OBJETO

AUTORES

**LUZ/CISNEROS CALDERON
HERNANDO ROJAS MONROY**

DIRECTOR

ANL. WILSON COBEÑA LEYTON

AÑO

1999

AGRADECIMIENTO

Agradecemos por esta meta alcanzada a Dios, a nuestras familias y amigos por su invaluable apoyo; al Analista Wilson Cobeña L., al Programa de Tecnología en Computación, al Instituto de Ciencias Químicas y a la Escuela Superior Politécnica del Litoral, por las enseñanzas impartidas y las facilidades brindadas durante la elaboración de nuestro proyecto.

Luz Cisneros Calderón
Hernando Rojas Monroy



DEDICATORIA

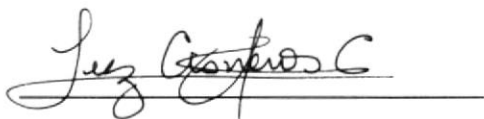
Ofrecemos el presente trabajo, fruto de muchos sacrificios, a nuestros padres, esposa y hermanos, quienes nos han brindado su apoyo en todo momento a lo largo de nuestros estudios.

Luz Cisneros Calderón
Hernando Rojas Monroy

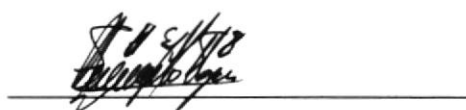
DECLARACION EXPRESA

La responsabilidad por los hechos, ideas y doctrinas expuestas en este proyecto, corresponden exclusivamente a los autores; y el patrimonio intelectual de los mismos a la Escuela Superior Politécnica del Litoral (Reglamento de Exámenes y Títulos Profesionales de la ESPOL).

LUZ CISNEROS CALDERON



HERNANDO ROJAS MONROY



DIRECTOR DEL PROYECTO

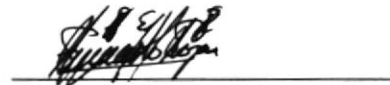
ANALISTA WILSON COBEÑA LEYTON

AUTORES DEL PROYECTO

LUZ CISNEROS CALDERON



HERNANDO ROJAS MONROY



CONTENIDO

	Página
CAPITULO 1	
1 INTRODUCCION	1
1.1 OBJETIVOS GENERALES	2
1.2 OBJETIVOS PARTICULARES	2
1.3 AUDIENCIA	2
 CAPITULO 2	
2 AMBIENTE OPERACIONAL	1
 CAPITULO 3	
3 ESTANDARIZACION	1
 CAPITULO 4	
4 MODELO VISUAL DE OBJETOS	1
 CAPITULO 5	
5 DISEÑO DE PANTALLAS	1
5.1 PANTALLA INICIAL DE ESPOLCHAT	2
5.2 PANTALLA DE CHARLA	3
5.3 PANTALLA DE AYUDA	4
5.4 PANTALLA DE AVISO	5
5.5 PANTALLA DEL SERVIDOR	6
 CAPITULO 6	
6 NARRATIVA DE PROGRAMAS ORIENTADA A OBJETO	1
6.1 SEGURIDADES DEL SISTEMA	21
 BIBLIOGRAFIA	

CAPITULO 1



1 INTRODUCCION

Posiblemente lo más importante acerca del Internet y sus efectos en el mundo de hoy es la certeza de que se trata de una increíble herramienta social. Las personas están hablando, compartiendo, conociendo y aprendiendo de otras, gracias al Internet; el cual es un lugar donde personas de todas las razas, credos, países y géneros pueden encontrarse y discutir ideas. Una forma central de Internet para el encuentro de mentes constituye el Internet Relay Chat.

Charlar es una de las posibilidades más agradables y adictivas del Internet; ya que esto le permite conocer gente con quien compartir intereses comunes, hacer nuevos amigos y algunas veces hasta pretender ser alguien que no es.

Existen algunas variedades para charlar en Internet, que incluye formas en HTML y Java, que nos brindan las herramientas necesarias para implementar el chat en nuestro propio sitio web. En realidad la mejor forma de charlar es implementando Java.

Entre las facilidades que brinda están:

- Puede correr bajo múltiples sistemas operativos y plataformas.
- Es fácil de usar y tiene todas las agradables interfaces que no posee el sistema en HTML, tales como el desplazamiento ligero y ambientes en tercera dimensión.
- No requiere de software preinstalado en la máquina cliente.
- Puede ser fácilmente adaptado en una página web, lo cual lo hace más flexible y fácil de usar.

La arquitectura del sistema chat en Java es similar a aquella empleada en un sistema escrito en lenguaje C++ u otro lenguaje que implique una arquitectura cliente/servidor. Esto significa que el sistema de charla tiene un programa ejecutable corriendo en el lado del cliente y también tiene un programa ejecutable corriendo en el lado del servidor. Los conceptos básicos del sistema de charla tipo cliente/servidor son los mismos si la interfaz es en texto, tercera dimensión o gráfica con muchos avatares.

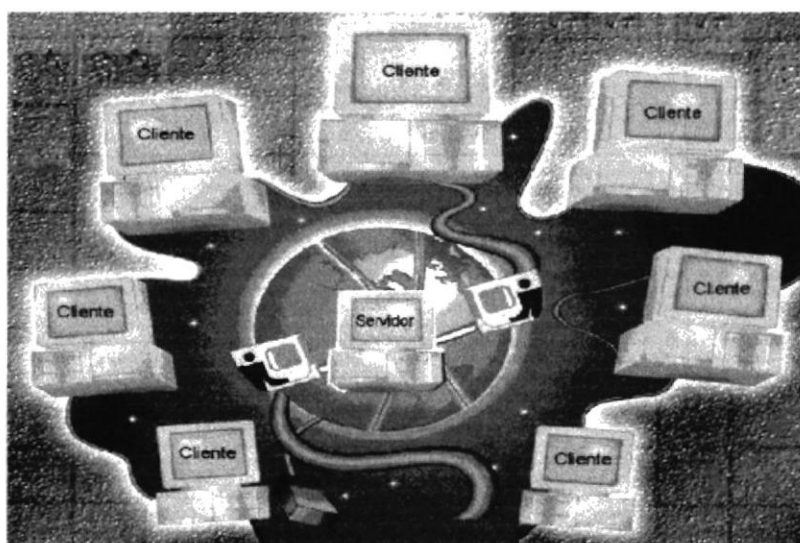


Gráfico 1.1.- Principio IRC utilizado a través de Internet.

1.1 OBJETIVOS GENERALES

- Implementar un sistema de charla en Internet propio de la ESPOL, utilizando las características de la programación orientada a objetos que ofrece el lenguaje Java.
- Cumplir con uno de los requisitos previa obtención del título de Programadores en Análisis de Sistemas.

1.2 OBJETIVOS PARTICULARES

Analizar, determinar y desarrollar las características básicas de un sistema de charla en Internet y ponerlas a disposición de otros programadores que, posiblemente en el futuro, se interesen en agregar otras que hagan a ESPOLChat cada vez más interesante, útil y entretenido para los usuarios.

1.3 AUDIENCIA

Este manual está dirigido a programadores con experiencia básica en programación orientada a objetos, sea ésta en lenguaje C, C++ o Java; ya que los fundamentos de estos lenguajes son similares.



CAPITULO 2

2 AMBIENTE OPERACIONAL

El presente trabajo constituye un sistema básico de charla vía Internet, en el cual se han aplicado los conocimientos adquiridos a lo largo del Tópico de Graduación en lenguaje JAVA, realizado desde Enero 16 a Mayo 22 de 1999, como requisito previo a la obtención del título de Programador en Análisis de Sistemas.

Esta aplicación, funciona bajo la arquitectura Cliente/Servidor y consta de los siguientes elementos:

- Una ventana principal, donde encontramos: los Cuartos de Charla (General, Política, Deportes, Música y Farándula); y los botones de Charlar, Desconectar y Ayuda.
- Una ventana de charla propiamente dicha, donde hallaremos los campos para ingreso de Nickname y Mensaje, así como los botones de Conectar, Enviar, Otro Cuarto y Salir.

El usuario tendrá acceso a esta aplicación desde cualquier Navegador de Internet que posea soporte para Java, desde Netscape Navigator 4.5 e Internet Explorer 3.0 o superiores.

El cliente o usuario podrá visualizar en el área de charla, tanto los nombres de los usuarios conectados, así como sus respectivos mensajes, a partir de que se realice la conexión con el servidor web que posee la aplicación de charla.

Para que la aplicación se ejecute de manera óptima, se necesita contar con lo siguientes elementos de hardware y software:

Para el Servidor Web:

Hardware

- Computador con procesador Pentium II o III, RAM de 64 MB.
- Un UPS.

Software

- Sistema Operativo MS Windows NT versión 4.0 o superiores.
- Lenguaje de programación Java JDK 1.1.7b o superior.

Para el Cliente (Usuario):

Hardware

- Computador con procesador 486 o Pentium II, RAM de 32 MB.
- Tarjeta Fax-Modem.



Software

- Sistema Operativo Windows NT Workstation, Windows 95, Windows 98 o cualquier otro tipo de sistema operativo con características similares o superiores.
- Navegador con soporte para Java (Netscape Navigator 4.5 o Internet Explorer 3.0 o superiores).
- Proveedor de servicio de Internet.



CAPITULO 3

3 ESTANDARIZACION

Nombre de la aplicación

ESPOLChat v1.0.

Clases

Deben empezar con letra mayúscula y su nombre debe describir al objeto que representa. Ejemplo:

Canal

Constructores

Deben tener el mismo nombre que la clase a la que pertenecen. La cantidad de parámetros que usen es independiente. Ejemplo:

Canal (tipo parámetro)

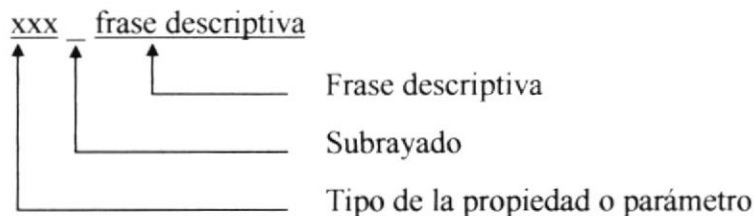
Métodos

Deben empezar con letra minúscula y su nombre debe describir su función. La cantidad de parámetros que usen es independiente. Ejemplo:

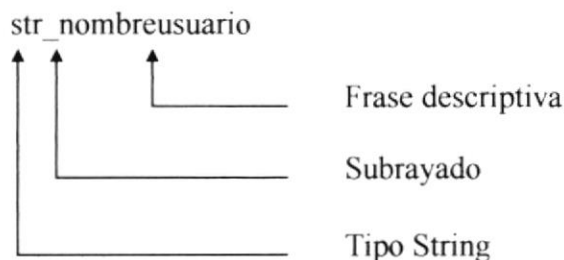
difundir (tipo parámetro)

Propiedades y Parámetros

Deben empezar con letra minúscula, los tres primeros caracteres describirán en forma abreviada el tipo de la propiedad o parámetro seguido de un carácter de subrayado y luego una frase descriptiva de la propiedad o parámetro.



Ejemplo:



Diseño de pantallas

DISEÑO DE PANTALLA	
Nombre de la forma:	Autores:
Fecha de diseño:	
Versión :	
Descripción:	
Labels in the screenshot: - Barra de título - Area de texto - Ventana - Cuadro de diálogo - Botón minimizar - Botón maximizar - Botón cerrar - Etiqueta - Nickname - Conectar - Charla - Enviar - Otro Cuadro - Salir - Cuartos - General - Deportes - Música - Paradoja - Botón - Barra de desplazamiento - Item - Lista - Campo de texto	
Lista de elementos de datos:	
Comentario:	

Formato de Pantallas

Las barras de título de cada pantalla deben tener en la esquina superior izquierda lo siguiente: el nombre del sistema seguido de 4 espacios, el nombre de la forma en mayúsculas seguido de 10 espacios, el nombre del PROTCOM en mayúsculas seguido de 1 espacio, 1 guión y 1 espacio; y, el nombre de la ESPOL en mayúsculas. Ejemplo:

ESPOLChat v1.0 CHARLA PROTCOM - ESPOL



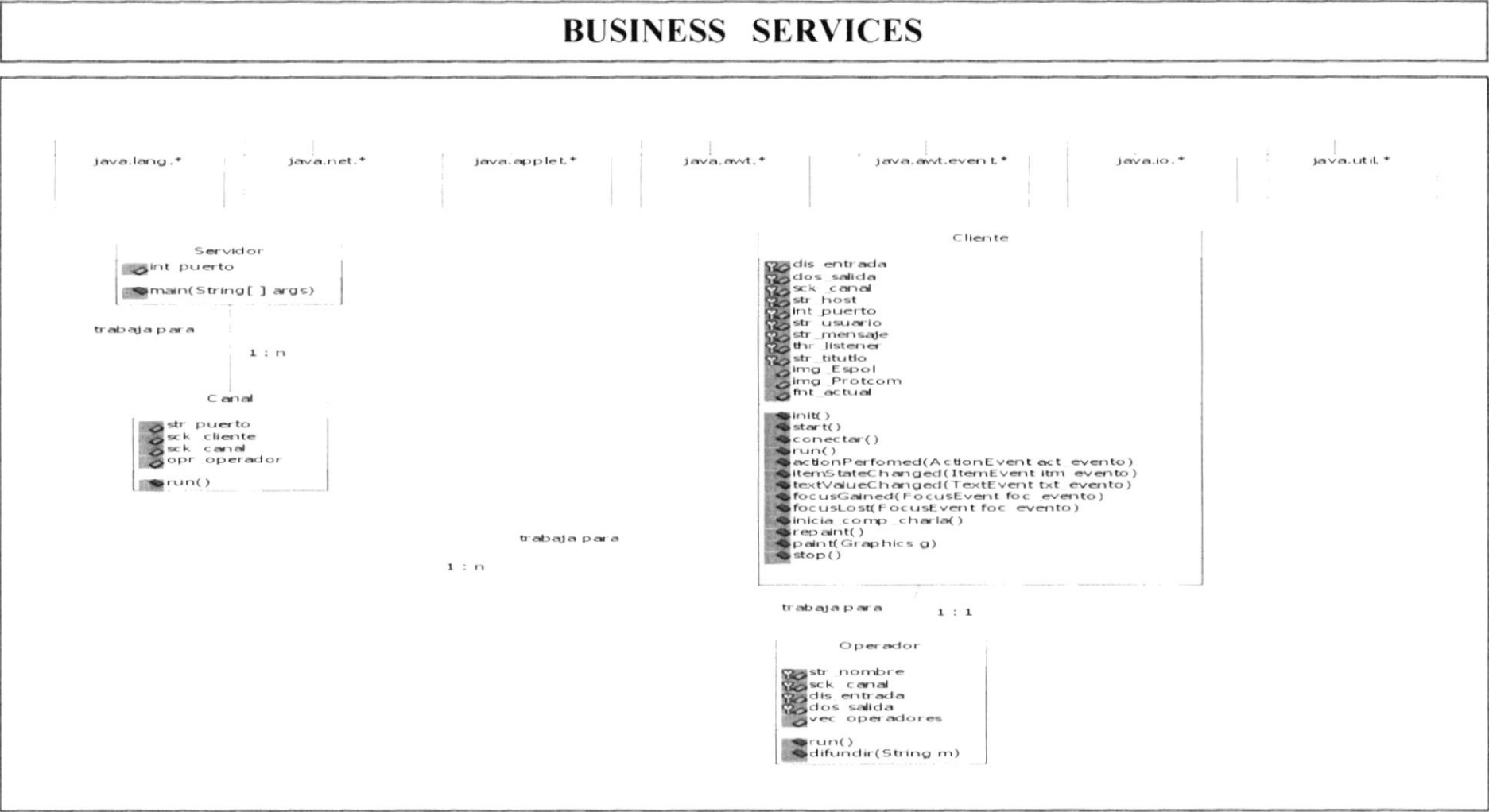
CAPITULO 4

4 MODELO VISUAL DE OBJETOS

El objetivo de este modelo, es el de hacer más fácil la comprensión de cómo esta estructurado el sistema, al describir en forma gráfica todos los componentes del mismo, tales como:

- Los paquetes de java utilizados por el sistema
- Las clases propias del sistema con sus respectivos métodos y propiedades
- El ámbito de los métodos y propiedades
- La relación existente entre las clases, y
- Visual servicio del negocio de la aplicación





CAPITULO 5



5 DISEÑO DE PANTALLAS

Especifica los datos sobre el diseño de cada una de las pantallas del sistema, tales como:

- **Nombre de la forma**

Nombre que se le ha dado a cada pantalla.

- **Fecha de diseño**

Fecha en que se diseñó cada pantalla.

- **Versión**

Versión de la pantalla.

- **Autores**

Se nombra a los autores de cada pantalla.

- **Descripción**

Breve descripción de la pantalla.

- **Lista de elementos de datos**

Enumera los elementos de datos y especifica su tipo.

- **Comentario**

Comentario referente al manejo de cada pantalla.

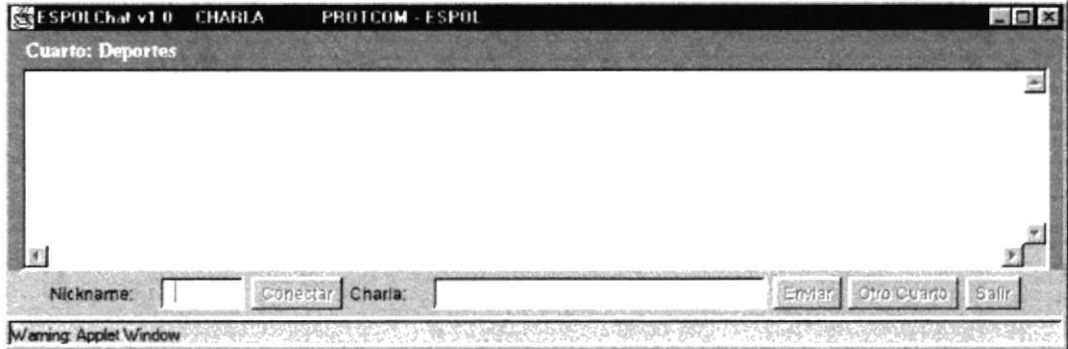


5.1 PANTALLA INICIAL DE ESPOLCHAT

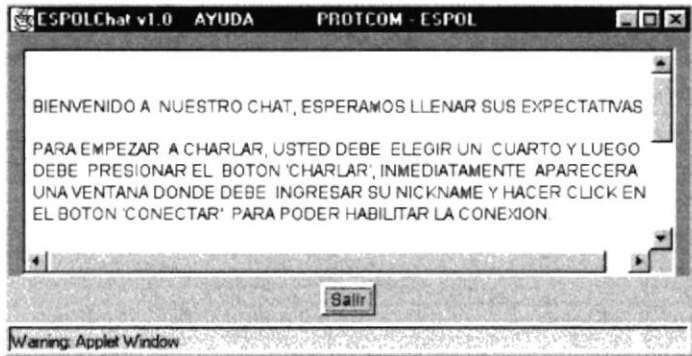
DISEÑO DE PANTALLA									
Nombre de la forma: ESPOLChat	Autores: Luz Cisneros Calderón								
Fecha de diseño: mayo 4 de 1999	Hernando Rojas Monroy								
Versión : 1.0									
Descripción: Pantalla inicial de ESPOLChat, para el cliente.									
Lista de elementos de datos: <table border="0"> <tbody> <tr> <td>lst_cuartos</td> <td>List</td> </tr> <tr> <td>btn_charlar</td> <td>Button</td> </tr> <tr> <td>btn_desconectar</td> <td>Button</td> </tr> <tr> <td>btn_ayuda</td> <td>Button</td> </tr> </tbody> </table>		lst_cuartos	List	btn_charlar	Button	btn_desconectar	Button	btn_ayuda	Button
lst_cuartos	List								
btn_charlar	Button								
btn_desconectar	Button								
btn_ayuda	Button								
Comentario: Se debe elegir un cuarto de charla y luego presionar el botón “Charlar”. Si desea salir presionar el botón “Desconectar”. Para mejor orientación presione el botón “Ayuda”.									



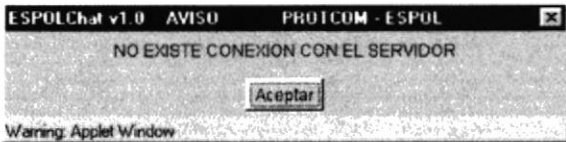
5.2 PANTALLA DE CHARLA

DISEÑO DE PANTALLA															
Nombre de la forma: CHARLA	Autores: Luz Cisneros Calderón Hernando Rojas Monroy														
Fecha de diseño: mayo 4 de 1999															
Versión : 1.0															
Descripción: Pantalla de charla de la aplicación.															
															
Lista de elementos de datos: <table> <tbody> <tr> <td>txa_areacharla</td> <td>TextArea</td> </tr> <tr> <td>txf_nombre</td> <td>TextField</td> </tr> <tr> <td>txf_mensaje</td> <td>TextField</td> </tr> <tr> <td>btn_conectar</td> <td>Button</td> </tr> <tr> <td>btn_enviar</td> <td>Button</td> </tr> <tr> <td>btn_otrocuarto</td> <td>Button</td> </tr> <tr> <td>btn_salircharla</td> <td>Button</td> </tr> </tbody> </table>		txa_areacharla	TextArea	txf_nombre	TextField	txf_mensaje	TextField	btn_conectar	Button	btn_enviar	Button	btn_otrocuarto	Button	btn_salircharla	Button
txa_areacharla	TextArea														
txf_nombre	TextField														
txf_mensaje	TextField														
btn_conectar	Button														
btn_enviar	Button														
btn_otrocuarto	Button														
btn_salircharla	Button														
Comentario: Se debe digitar el nickname (nombre con el cual se reconocerá al usuario durante la sesión de charla) en el campo "Nickname" y luego presionar el botón "Conectar". Luego escribir el mensaje en el campo "Charla" y presionar el botón "Enviar". Para cambiar de cuarto de charla presionar el botón "Otro Cuarto". El botón "Salir" se activa cuando se produce una excepción.															

5.3 PANTALLA DE AYUDA

DISEÑO DE PANTALLA					
Nombre de la forma: AYUDA	Autores: Luz Cisneros Calderón Hernando Rojas Monroy				
Fecha de diseño: mayo 4 de 1999					
Versión : 1.0					
Descripción: Pantalla de ayuda de la aplicación.					
					
Lista de elementos de datos: <table> <tbody> <tr> <td>txa_areaayuda</td> <td>TextArea</td> </tr> <tr> <td>btn_ayuda</td> <td>Button</td> </tr> </tbody> </table>		txa_areaayuda	TextArea	btn_ayuda	Button
txa_areaayuda	TextArea				
btn_ayuda	Button				
Comentario: Utilizar las barras de desplazamiento horizontal y vertical, o maximizar la pantalla para leer todo el texto de la ayuda. Para salir presione el botón “Salir”.					

5.4 PANTALLA DE AVISO

DISEÑO DE PANTALLA	
Nombre de la forma: AVISO	Autores: Luz Cisneros Calderón
Fecha de diseño: mayo 4 de 1999	Hernando Rojas Monroy
Versión : 1.0	
Descripción: Pantalla que muestra los mensajes de la aplicación.	
	
Lista de elementos de datos:	
dlg_aviso	Dialog
lbl_aviso	Label
btn_aviso	Button
Comentario: Presionar el botón “Aceptar” después de haber leído el mensaje.	

5.5 PANTALLA DEL SERVIDOR

DISEÑO DE PANTALLA

Nombre de la forma: SERVIDOR

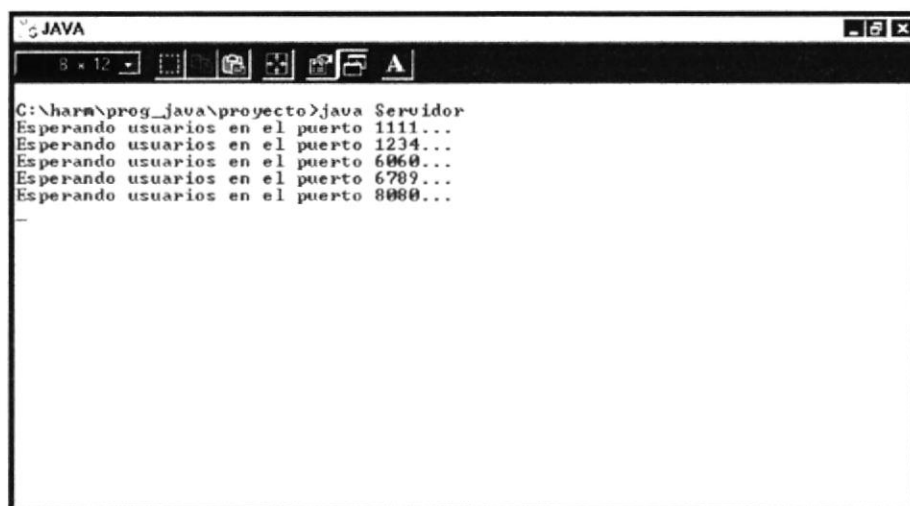
Autores: Luz Cisneros Calderón

Fecha de diseño: mayo 4 de 1999

Hernando Rojas Monroy

Versión : 1.0

Descripción: Pantalla de la aplicación Servidor.



```
C:\harm\prog_java\proyecto>java Servidor
Esperando usuarios en el puerto 1111...
Esperando usuarios en el puerto 1234...
Esperando usuarios en el puerto 6060...
Esperando usuarios en el puerto 6789...
Esperando usuarios en el puerto 8080...
```

Lista de elementos de datos:

No tiene.

Comentario: Este programa debe ser ejecutado en el servidor con el objeto de que los clientes lo invoquen.

CAPITULO 6



6 NARRATIVA DE PROGRAMAS ORIENTADA A OBJETO

Describe cada una de las clases del sistema especificando sus características, las cuales son:

- **Tipo de control**

Permite controlar la visibilidad de, y el acceso a, variables y métodos dentro de las clases.

- **Extensión**

Indica la superclase de la cual la clase actual hereda sus métodos y propiedades.

- **Relación**

Indica el tipo de relación que una clase tiene con otra u otras clases.

- **Descripción**

Describe brevemente la función que cumple la clase.

- **Ubicación**

Determina la jerarquía de clases de la clase actual.

- **Constructores**

Métodos especiales para inicializar las propiedades de una clase.

- **Métodos**

Funciones propias de una clase que se ejecutan según sea necesario.

- **Propiedades**

Variables que utiliza la clase.

- **Nombre**

Nombre de la variable.

- **Tipo**

Tipo de la variable.

- **Valor Inicial**

Valor con el que se inicializa la variable al declararla.

- **Descripción**

Breve descripción de la aplicación que se le da a la variable.

- **Interfaces**

Describe las colecciones de métodos que pueden añadirse a una clase para proveerla de mayor funcionalidad.



NOMBRE CLASE: Servidor									
TIPO DE CONTROL: Público EXTENSION: application RELACION: Asociación Canal	PAGINA: 1 de 2								
DESCRIPCION: Ofrece varios canales y espera por peticiones de conexión.	FECHA ELABORACION: abril 15 de 1999								
UBICACION: Clase actual Servidor	FECHA ACTUALIZACION: junio 17 de 1999								
INDICE									
<u>CONSTRUCTORES:</u> No tiene.									
<u>PROPIEDADES:</u> static int_puerto: Número del puerto para la conexión.									
<u>METODOS:</u> main () Método principal que se ejecuta al correr la aplicación.									
<u>INTERFACES:</u> IOException Construye una excepción de Input Output sin dar un mensaje. RuntimeException Superclase de las excepciones lanzadas durante la operación normal de la máquina virtual de java.									
DETALLE									
<u>CONSTRUCTORES:</u> No tiene.									
<u>PROPIEDADES:</u>									
<table border="1"> <thead> <tr> <th><u>Nombre</u></th> <th><u>Tipo</u></th> <th><u>Valor Inicial</u></th> <th><u>Descripción</u></th> </tr> </thead> <tbody> <tr> <td>int_puerto</td> <td>int</td> <td>0</td> <td>Número de puerto para un cuarto de charla</td> </tr> </tbody> </table>	<u>Nombre</u>	<u>Tipo</u>	<u>Valor Inicial</u>	<u>Descripción</u>	int_puerto	int	0	Número de puerto para un cuarto de charla	
<u>Nombre</u>	<u>Tipo</u>	<u>Valor Inicial</u>	<u>Descripción</u>						
int_puerto	int	0	Número de puerto para un cuarto de charla						

NOMBRE CLASE: Servidor	
TIPO DE CONTROL: Público EXTENSION: application RELACION: Asociación Canal	PAGINA: 2 de 2
DESCRIPCION: Ofrece varios canales y espera por peticiones de conexión.	FECHA ELABORACION: abril 15 de 1999
UBICACION: Clase actual Servidor	FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u> main Sintaxis: <pre>public static void main (String[] str_args) throws IOException</pre> Descripción: Inicia los hilos de ejecución de cada canal de charla.	
Parámetros: str_args Cadena ingresada en la línea de comando.	
Retorna: Nada.	
Manejo de errores: IOException	



NOMBRE CLASE: Canal**TIPO DE CONTROL:** Público**EXTENSION:** application**RELACION:** Asociación Servidor
Asociación Operador**DESCRIPCION:** Crea conexión para establecer
enlace entre operador y cliente.**UBICACION:** Clase actual Canal**PAGINA:**

1 de 2

FECHA ELABORACION:

abril 15 de 1999

FECHA ACTUALIZACION:

junio 17 de 1999

INDICE**CONSTRUCTORES:****Canal (int int_puerto)** Crea un canal que espera por peticiones de conexión**PROPIEDADES:**

str_puerto: Cadena que guarda el número del puerto del cuarto elegido
opr_operador: Hilo para conectarse con el servidor.
sck_cliente: Socket para conectar al cliente con el servidor.
sck_canal: Puerto en el servidor para conectar al cliente.

METODOS:**run()** Realiza la mayor parte del trabajo en la aplicación.**INTERFACES:**

Runnable Método para clases que quieren correr como hilos.
IOException Construye una excepción de Input Output sin dar un mensaje

DETALLE**CONSTRUCTORES:****Sintaxis:** Canal (int int_puerto)**Parámetros:****int_puerto** Variable tipo entero para el número del puerto (cuarto elegido).**Descripción:**

Crea una instancia ServerSocket que espera por una petición de conexión.

NOMBRE CLASE: Canal**TIPO DE CONTROL:** Público**EXTENSION:** application**RELACION:** Asociación Servidor
Asociación Operador**DESCRIPCION:** Crea conexión para establecer
enlace entre operador y cliente.**UBICACION:** Clase actual Canal**PAGINA:**

2 de 2

FECHA ELABORACION:

abril 15 de 1999

FECHA ACTUALIZACION:

junio 17 de 1999

DETALLE**PROPIEDADES:**

<u>Nombre</u>	<u>Tipo</u>	<u>Valor Inicial</u>	<u>Descripción</u>
str_puerto	String	Null	Puerto de conexión
opr_operador	Operador		Hilo de enlace
sck_canal	ServerSocket		Puerta de enlace en servidor
sck_cliente	Socket		Puerta enlace de cliente en servidor

METODOS:**run****Sintaxis:**

public void run ()

Descripción:

Acepta la petición de conexión del cliente y lo enlaza al operador.

Parámetros:

No tiene.

Retorna:

Nada.

Manejo de errores:

IOException

NOMBRE CLASE: Operador	
TIPO DE CONTROL: Público EXTENSION: java.lang.Thread RELACION: Asociación Canal	PAGINA: 1 de 3
DESCRIPCION: Crea instancias de entrada y salida de datos para cada cliente.	FECHA ELABORACION: abril 15 de 1999
UBICACION: Clase actual Operador	FECHA ACTUALIZACION: junio 17 de 1999
INDICE	
<u>CONSTRUCTORES:</u>	
Operador (Socket sock_conexion)	Recibe un objeto Socket e inicia instancias de entrada y salida para el Socket.
<u>PROPIEDADES:</u>	
protected static vec_operadores:	Vector que guarda a los clientes conectados.
protected str_nombre:	Cadena que guarda el nombre del usuario.
protected sock_canal:	Puerta de enlace para el operador.
protected dis_entrada:	Flujo de entrada de datos para el operador.
protected dos_salida:	Flujo de salida de datos para el operador.
<u>METODOS:</u>	
run ()	Realiza la mayor parte del trabajo en la aplicación.
difundir (String str_msje)	Envia los mensajes a todos los clientes.
<u>INTERFACES:</u>	
Enumeration	Método para la enumeración de sets de valores
IOException	Construye una excepción de Input Output sin dar un mensaje
DETALLE	
<u>CONSTRUCTORES:</u>	
Sintaxis:	Operador (Socket sock_conexion)



NOMBRE CLASE: Operador**TIPO DE CONTROL:** Público**EXTENSION:** java.lang.Thread**RELACION:** Asociación Canal**PAGINA:**

2 de 3

DESCRIPCION: Crea instancias de entrada y salida de datos para cada cliente.**FECHA ELABORACION:**

abril 15 de 1999

UBICACION: Clase actual Operador**FECHA ACTUALIZACION:**

junio 17 de 1999

DETALLE**CONSTRUCTORES:****Parámetros:**

sck_conexion Inicializa el valor de la propiedad sck_canal.

Descripción:

Crea una conexión de tipo socket hacia el cliente que pasó Como parámetro y obtiene los streams de entrada y salida para esa conexión.

PROPIEDADES:

<u>Nombre</u>	<u>Tipo</u>	<u>Valor Inicial</u>	<u>Descripción</u>
vec_operadores	Vector		Arreglo que guarda a los clientes
sck_canal	Socket		Puerta de enlace para el operador
dis_entrada	DataInputStream		Línea entrada datos de operador
dos_salida	DataOutputStream		Línea salida datos de operador

METODOS:**run****Sintaxis:**

public void run ()

Descripción:

Método donde se ejecutan las principales acciones de la aplicación.

Parámetros:

No tiene.

Retorna:

Nada.

NOMBRE CLASE: Operador	
TIPO DE CONTROL: Público EXTENSION: java.lang.Thread RELACION: Asociación Canal	PAGINA: 3 de 3
DESCRIPCION: Crea instancias de entrada y salida de datos para cada cliente.	FECHA ELABORACION: abril 15 de 1999
UBICACION: Clase actual Operador	FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u>	
Manejo de errores: IOException	
difundir	
Sintaxis: public static void difundir (String str_msje)	
Descripción: Método que distribuye los mensajes entre los clientes.	
Parámetros: str_msje Cadena que contiene el mensaje a difundir.	
Retorna: Nada.	
Manejo de errores: IOException.	

NOMBRE CLASE: Cliente**TIPO DE CONTROL:** Público**EXTENSION:** Applet**RELACION:** Asociación Canal
Asociación Operador**DESCRIPCION:** Crea la interfase para el usuario
e inicia la conexión con el
servidor.**UBICACION:** Clase actual Cliente**PAGINA:**

1 de 11

FECHA ELABORACION:

abril 15 de 1999

FECHA ACTUALIZACION:

junio 17 de 1999

INDICE**CONSTRUCTORES:**

No tiene.

PROPIEDADES:

protected dis_entrada:	Flujo de entrada de datos
protected dos_salida:	Flujo de salida de datos
protected sock_canal:	Puerta de enlace para la comunicación
protected str_host:	Nombre del servidor
protected int_puerto:	Canal de comunicación del servidor
protected str_usuario:	Nombre del cliente
protected str_mensaje:	Mensaje del usuario en formato UTF-8 (encoded)
protected thr_listener:	Hilo de conexión para el cliente
protected str_titulo:	Título del cuarto de charla
img_espol:	Logotipo de la ESPOL
img_protcom:	Logotipo del PROTCOM
fnt_actual:	Fuente actual del texto del applet

METODOS:

init ()	Crea la interfase del usuario.
start ()	Obtiene parámetros de conexión para el applet.
conectar ()	Trata de conectarse al servidor e inicia el hilo de ejecución.
run ()	Realiza el trabajo principal del applet.
accionPerfomed(ActionEvent act_evento)	Reconoce el tipo de acción del evento.

NOMBRE CLASE: Cliente**TIPO DE CONTROL:** Público**EXTENSION:** Applet**RELACION:** Asociación Canal
Asociación Operador**DESCRIPCION:** Crea la interfase para el usuario
e inicia la conexión con el
servidor.**UBICACION:** Clase actual Cliente**PAGINA:**

2 de 11

FECHA ELABORACION:

abril 15 de 1999

FECHA ACTUALIZACION:

junio 17 de 1999

INDICE**METODOS:**

textValueChanged(TextEvent txt_evento)	Evalúa cambios en texto de un área o campo de texto.
itemStateChanged(ItemEvent itm_evento)	Evalúa cambios en items de un componente tipo lista.
focusGained(FocusEvent fgn_evento)	Evalúa cuál componente ha ganado el foco.
focusLost (FocusEvent fls_evento)	Evalúa cuál componente ha perdido el foco.
inicia_comp_charla()	Reinicializa los componentes del área de charla.
captura_excepcion()	Presenta los mensajes de las excepciones.
repaint()	Llama al método paint().
paint (Graphics g)	Crea la pantalla del applet.
stop ()	Finaliza el hilo de comunicación y todos sus procesos.

INTERFACES:

ActionListener:	Método para atender las acciones de los eventos.
ItemListener:	Método para atender un evento de item
TextListener:	Método para atender un evento de texto

BIBLIOTECA
CAMPUS
PEÑAS

NOMBRE CLASE: Cliente**TIPO DE CONTROL:** Público**EXTENSION:** Applet**RELACION:** Asociación Canal
Asociación Operador**DESCRIPCION:** Crea la interfase para el usuario
e inicia la conexión con el
servidor.**UBICACION:** Clase actual Cliente**PAGINA:**

3 de 11

FECHA ELABORACION:

abril 15 de 1999

FECHA ACTUALIZACION:

junio 17 de 1999

INDICE

FocusListener: Método para atender un evento de foco
Runnable: Método para clases que quieren correr como hilos.
NullPointerException: Detecta una referencia a un objeto nulo.
IOException: Construye una excepción de Input Output sin dar un mensaje.

DETALLE**CONSTRUCTORES:**

No tiene.

PROPIEDADES:

<u>Nombre</u>	<u>Tipo</u>	<u>Valor Inicial</u>	<u>Descripción</u>
dis_entrada	DataInputStream		Flujo de entrada de datos
dos_salida	DataOutputStream		Flujo de salida de datos
sck_canal	Socket		Puerta para la comunicación
str_host	String	Null	Nombre del servidor
int_puerto	int	0	Canal de comunicación
str_usuario	String	Null	Nombre del cliente
str_mensaje	String	Null	Mensaje en formato UTF-8
img_Espol	Image		Logotipo de la ESPOL
img_Protcom	Image		Logotipo del PROTCOM
thr_listener	Thread		Hilo de conexión para cliente
fnt_actual	Font		Fuente actual del applet

METODOS:**init****Sintaxis:**

public void init ()

NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público	PAGINA: 4 de 11
EXTENSION: Applet	
RELACION: Asociación Canal Asociación Operador	FECHA ELABORACION: abril 15 de 1999
DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor.	
UBICACION: Clase actual Cliente	FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u>	
Descripción: Crea la interfase para el usuario.	
Parámetros: No tiene.	
Retorna: Nada.	
Manejo de errores: No.	
start	
Sintaxis: public void start ()	
Descripción: Obtiene parámetros de conexión para el applet.	
Parámetros: No tiene.	
Retorna: Nada.	
Manejo de errores: No.	

NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor. UBICACION: Clase actual Cliente	PAGINA: 5 de 11 FECHA ELABORACION: abril 15 de 1999 FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u>	
conectar	
Sintaxis: public void conectar ()	
Descripción: Para conectarse al servidor e iniciar hilo thr_listener.	
Parámetros: No tiene.	
Retorna: Nada.	
Manejo de errores: IOException	
run	
Sintaxis: public void run ()	
Descripción: Crea el área funcional principal del applet.	
Parámetros: No tiene.	
Retorna: Nada.	



NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor. UBICACION: Clase actual Cliente	PAGINA: 6 de 11 FECHA ELABORACION: abril 15 de 1999 FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u> Manejo de errores: IOException actionPerfomed Sintaxis: Public void actionPerfomed (ActionEvent act_evento) Descripción: reconoce el tipo de acción del evento. Parámetros: act_evento Retorna: Nada. Manejo de errores: IOException NullPointerException itemStateChanged Sintaxis: public void itemStateChanged (ItemEvent itm_evento) Descripción: Evalúa cambios en estado de items en componente tipo lista.	

NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor. UBICACION: Clase actual Cliente	PAGINA: 7 de 11 FECHA ELABORACION: abril 15 de 1999 FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
METODOS: Parámetros: itm_evento Retorna: Nada. Manejo de errores: NumberFormatException textValueChanged Sintaxis: public void textValueChanged (TextEvent txt_evento) Descripción: Evalúa cambios en texto de un área o campo de texto. Parámetros: txt_evento Retorna: Nada. Manejo de errores: No.	



NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público	PAGINA: 8 de 11
EXTENSION: Applet	
RELACION: Asociación Canal Asociación Operador	FECHA ELABORACION: abril 15 de 1999
DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor.	
UBICACION: Clase actual Cliente	FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u>	
Inicia_comp_charla	
Sintaxis: public void inicia_comp_charla ()	
Descripción: Inicializa los componentes del área de charla	
Parámetros: No tiene.	
Retorna: Nada.	
Manejo de errores: No.	
captura_excepcion	
Sintaxis: public void captura_excepcion (String str_excepcion)	
Descripción: Presenta los mensajes de las excepciones en una caja de diálogo.	
Parámetros: str_excepcion	
Retorna: Nada.	



NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador	PAGINA: 9 de 11
DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor.	FECHA ELABORACION: abril 15 de 1999
UBICACION: Clase actual Cliente	FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u>	
Manejo de errores: No.	
focusGained	
Sintaxis: public void focusGained (FocusEvent fgn_evento)	
Descripción: Evalúa qué evento ha ganado el foco.	
Parámetros: fgn_evento	
Retorna: Nada.	
Manejo de errores: No.	
focusLost	
Sintaxis: public void focusLost (FocusEvent fls_evento)	
Descripción: Método que debe estar presente para hacer perder foco a un componente	
Parámetros: fls_evento	

NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor. UBICACION: Clase actual Cliente	PAGINA: 10 de 11 FECHA ELABORACION: abril 15 de 1999 FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
METODOS: Retorna: Nada. Manejo de errores: No. repaint Sintaxis: public void repaint () Descripción: Llama al método paint. Parámetros: No tiene. Retorna: Nada. Manejo de errores: No. paint Sintaxis: Public void paint (Graphics grp_pantalla)	



NOMBRE CLASE: Cliente	
TIPO DE CONTROL: Público EXTENSION: Applet RELACION: Asociación Canal Asociación Operador DESCRIPCION: Crea la interfase para el usuario e inicia la conexión con el servidor. UBICACION: Clase actual Cliente	PAGINA: 11 de 11 FECHA ELABORACION: abril 15 de 1999 FECHA ACTUALIZACION: junio 17 de 1999
DETALLE	
<u>METODOS:</u> Descripción: Dibuja la ventana con sus componentes Parámetros: grp_pantalla Retorna: Nada. Manejo de errores: No. stop Sintaxis: public void stop () Descripción: Finaliza la sesión (hilo de comunicación) y todos sus procesos. Parámetros: No tiene. Retorna: Nada. Manejo de errores: No.	

6.1 SEGURIDADES DEL SISTEMA

- El sistema es de uso público ya que cualquier persona está en condiciones de usarlo.
- El usuario se deberá registrar al inicio de la sesión.



BIBLIOGRAFIA

- Teach Yourself JAVA 1.1 in 21 Days.
Laura Lemay, Charles L. Perkin.
Second Edition.
- The JAVA 1.1 Programmer's Reference.
Daniel I. Joshi & Pavel A. Vorobiev, PhD.
First Edition.
- <http://www.java.sun.com/products/jdk/1.1/docs/api/packages.html>
- <http://www.members.es.tripod.de/froufe/index.html>
- <http://www.casarramona.com/mt/programador/dirmanual.html>

