ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

PROYECTO DE GRADUACION

"Juego de Damas"

Previa a la obtención del Título de

INGENIERO EN COMPUTACION

PRESENTADA POR:

Guillermo Araujo

José Luis Loaiza

César Páez

Grace Pastorelly

Guayaquil - Ecuador

-:- 1 9 9 5 -:-

INDICE

1. ESPECIFICACIONES

1.1 *DESCRIPCIÓN GENERAL DEL PROYECTO*

Este proyecto consiste en un juego de damas que será implementado como un sistema CLIENTE-SERVIDOR. La parte servidora estará implementada en lenguaje C bajo una plataforma UNIX y la parte cliente en Visual Basic bajo la plataforma windows.

Se hará uso de Sockets para la comunicación entre nuestros programas(cliente y servidor) aprovechando su característica de usar protocolo TCP/IP. Para el desarrollo de la parte de comunicación del CLIENTE se usará DISTINCT.

1.2 *REQUERIMIENTOS FUNCIONALES*

Este sistema realizará las siguientes funciones:

- Mantendrá una lista de jugadores inscritos al club de juego de damas.
- Podrán jugar al mismo tiempo varios jugadores.
- Se llevará registro de los partidos .
- Cada jugador podrá invitar a otro siempre que este esté inscrito.
- Cada jugador podrá retirarse de cualquier partido cuando guste.
- Cada jugador podrá retirarse del juego cuando guste(solicitar ser eliminado de la lista de jugadores).
- Se mantendrá una cola de clientes.
- Se mantendrá un registro de los jugadores que han ganado.
- Implementará medidas de seguridad para que cada cliente accese al juego por medio de una contraseña.

1.3 REQUERIMIENTOS DE RENDIMIENTO Y CONFIABILIDAD

El rendimiento y la confiabilidad de nuestro sistema dependerá de dos factores : Primero del porcentaje de carga donde el programa servidor este corriendo. Segundo del tipo de enlace existente entre el CLIENTE y el SERVIDOR, por ejemplo:

Si el cliente está corriendo en una PC y el enlace es vía telefónica trabajando a 9600 bps, entonces el tiempo de respuesta para cada jugada no será más allá de 30 segundos. En cuanto a la confiabilidad, podemos decir que nuestro sistema es muy confiable (los protocolos TCP/IP son bastante confiables en la transmisión de datos) y solo se verá restringido por la calidad de la señal telefónica.

El performance del sistema se mantendrá en un nivel óptimo siempre y cuando el número de jugadas simultáneas sea hasta un máximo de 10, luego de lo cual este empezará a disminuir.

2. DISEÑO DEL PROTOCOLO

2.1 *ARQUITECTURA CLIENTE - SERVIDOR DE LA APLICACIÓN*

El cliente será realizado en Visual Basic 3.0, ya que este presenta gran facilidad de programación. La programación esta orientada a eventos, permite mantener funcionalidad en la implementación, así como facilidad en el manejo de archivos y uso de base de datos (ACCESS 1.1).

Se determinó que el CLIENTE mantendrá un archivo de configuración, que contiene información tanto del directorio en el cual se encuentra la base de datos, así como de la dirección ip del servidor con el que se desea comunicar y respectivo well-known port.

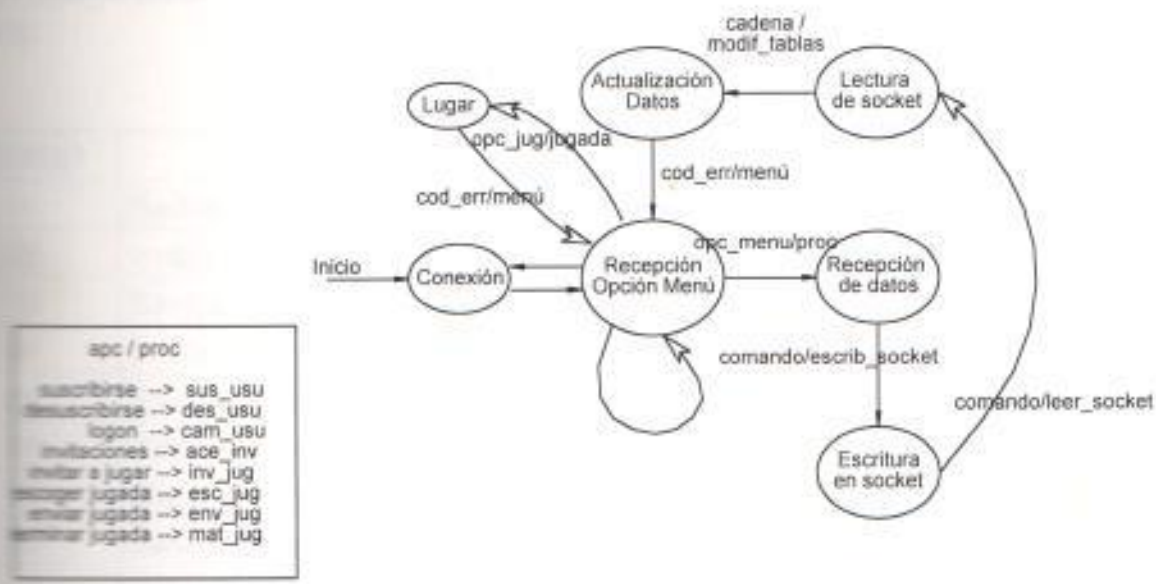
Otra de las razones por la cual el CLIENTE será desarrollado en V.B., es que la interfaz con el usuario es en ambiente gráfico.

Por otro lado, el SERVIDOR será implementado en lenguaje C con la característica que será concurrente para ofrecer servicio a diferentes usuarios al mismo tiempo. Del tipo stateless, y será orientado a conexión es decir utilizando los protocolos TCP/IP, para ofrecer seguridad en la transmisión de datos.

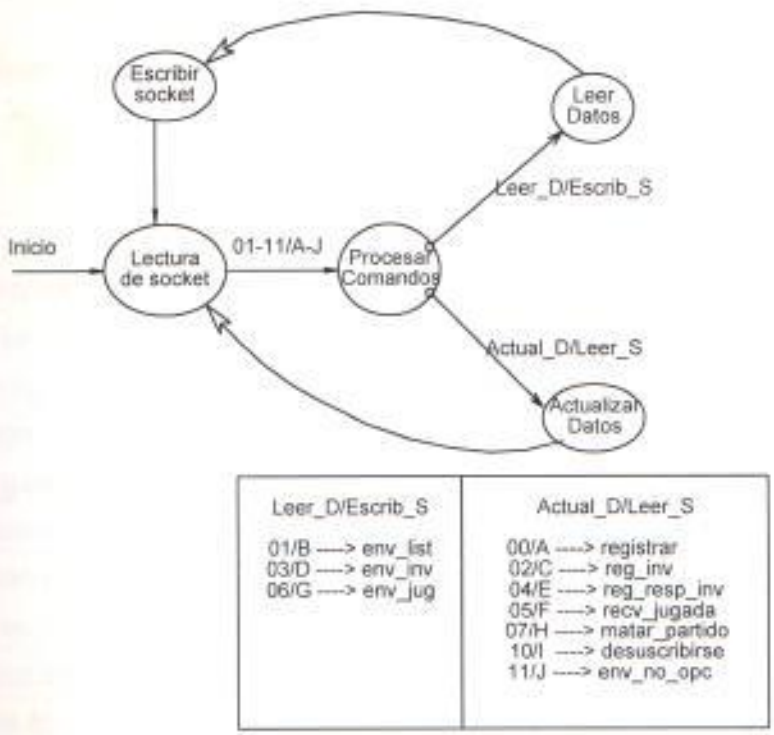
Para su implementación se usará lenguaje "C", que trabaje sobre el sistema operativo UNIX. , y para la comunicación con el cliente se lo hará a través de los sockets.

Así mismo, utilizará diferentes tablas para el almacenamiento de los datos de los jugadores y el estado de cada uno de sus partidos.

2.2. MAQUINA DE ESTADOS DEL CLIENTE Y DEL SERVIDOR



MAQUINA DE ESTADOS DEL SERVIDOR



2.3 SINTAXIS Y SEMANTICA DEL PROTOCOLO EN EL QUE SE BASAN EL CLIENTE Y EL SERVIDOR

Las opciones con que dispone el cliente se encuentran representadas por un código único. Este código es de tipo string, y puede tomar los siguientes valores:

CODIGO	DESCRIPCION
01	Recibir la lista de jugadores ya suscritos, sin incluir la propia.
02	Invitar a alguien a jugar.
03	Recibir invitaciones de otros suscritores.
04	Enviar respuesta de invitaciones.
05	Enviar jugadas (el cliente envía).
06	Recibir jugadas (el cliente recibe).
07	Matar partidos en los que el suscritor esta involucrado.
10	Desuscribirse.
11	Obtener lista de los partidos que el subscriptor esta jugando.
12	Logon, Verificación de password
13	Suscribirse al juego

Tabla 2.3.1 Protocolo de comunicación de datos entre el servidor y el cliente

Si ocurriese un error en el servidor, este informa al cliente el tipo de error ocurrido, para lo cual se han definido un conjunto de posibles mensaje y errores. Los mensajes constan de 4 dígitos, que se encuentran representados por la serie 99xx. Los que se encuentran codificados mediante la siguiente tabla:

9900	El cliente fue registrado con éxito.
9901	El alias a usar esta siendo usado. Alias duplicado.
9902	Ud. ya se encuentra registrado. No puede ser registrado nuevamente.
9904	Opción no implementada.
9905	El contrincante se ha retirado del partido.
9906	Ud. ha ganado el juego.
9907	El contrincante ha ganado el juego.
9909	Clave Incorrecta.
9910	Todo fue realizado exitosamente.
9911	No se podido realizar la operación.
9912	No tiene Invitaciones.
9913	Ya está jugando con todos los suscriptores.

Tabla 2.3.2 Protocolo de comunicación de errores entre el servidor y el cliente

Se ha determinado que para facilidad de envío y recepción entre el cliente y servidor, se usara un delimitador de campo, así como un delimitador de fin de buffer.

;	delimitador de campo
#	delimitador de fin de buffer

es decir un típico buffer ya sea de envío o recepción tendrá un formato como el que se muestra:

cod_oper;campo_uno;campo_dos;campo_tres;campo_cuatro;.....;campo_n;#

Las fichas del tablero son representadas en el buffer mediante un identificador de 2 caracteres; donde el caracter mas significativo representa el color de la ficha, y el caracter menos significativo el tipo de ficha (dama o reina).

El color rojo será representado por un 1, mientras que el de color azul mediante un 1. Una ficha dama tiene un código 0 y una reina será representada por el código 1.

Código	Significado
00	azul dama
10	roja dama
01	azul reina
11	roja reina

Tabla 2.3.3 Color y tipo de ficha

Las posiciones de las fichas en el tablero serán representadas desde la parte superior izquierda del tablero; y la parte inferior derecha como 8,8.

Un subscriptor puede encontrarse registrado solo una vez. Esto implica que no puede existir dos subscriptores con un mismo nombre o identificador (alias).

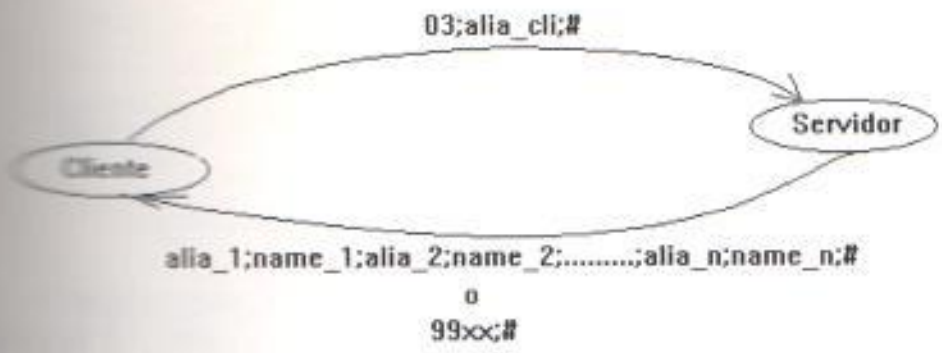
El alias de un subscriptor puede ser alfanumérico (a-z|A-Z|0-9) de longitud máxima de 8 caracteres. El nombre solo puede ser alfabético (A-Z|.) y de una longitud máxima de 35 caracteres.

Recibir lista de jugadores (01):

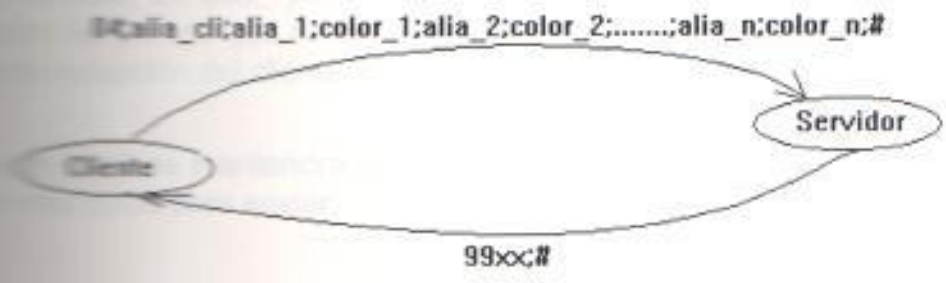
Una vez recibido el buffer del servidor, el cliente envía el correspondiente mensaje 99xx al servidor.

Invitar a alguien a jugar (02):

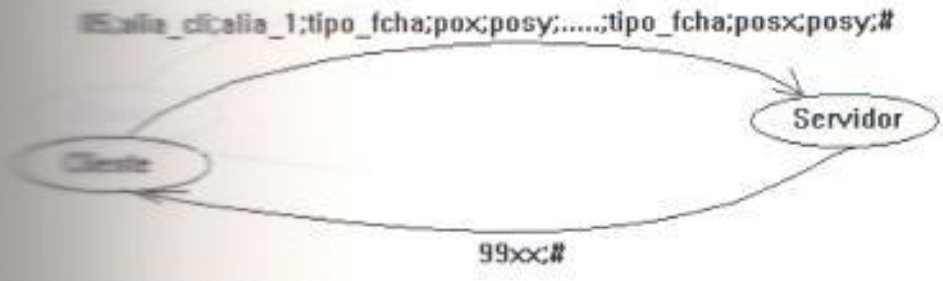
Recibir Invitaciones (03):



Enviar Respuesta de Invitación (aceptar invitación) (04):

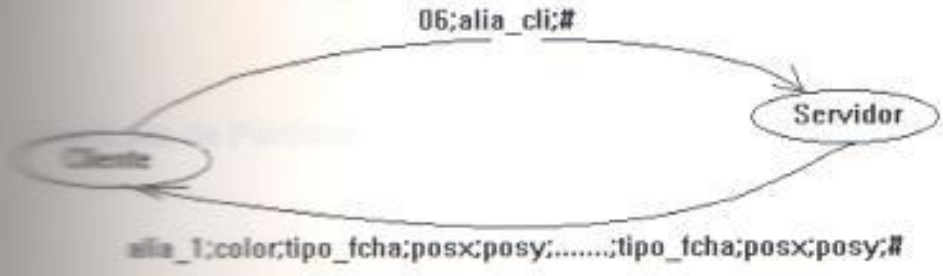


Enviar jugada (05):



Las jugadas por el cliente serán almacenadas en la tabla fichas_out hasta el momento de su envío, lo que se hará jugada por jugada, hasta que hayan sido enviadas todas.

Recibir jugada (06):



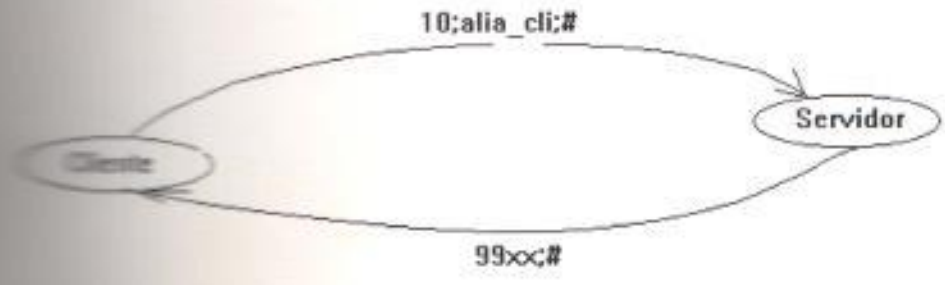
ASAET + DELIMITADOR + COLOR + TIPO FICHA + DELIMITADOR + POSX + DELIMITADOR + POSY + DELIMITADOR + TIPO FICHA + DELIMITADOR + POSX + DELIMITADOR + POSY + DELIMITADOR + ...+ CHR(13) + ALIAS2 + DELIMITADOR + TIPO FICHA + DELIMITADOR + POSX + DELIMITADOR + POSY + DELIMITADOR + TIPO FICHA + DELIMITADOR + POSX + DELIMIATDOR + POSY + DELIMITADOR + FIN BUFFER

Una vez enviado el primer tramo al servidor, el cliente espera por el buffer indicado arriba, y este es grabado en la tabla fichas y contrin de la base para mayor facilidad de uso.; y será mostrado al cliente para que este realice la selección del partido que desea realizar la jugada. El color especificado en la trama de recepción del cliente, es el color con que el cliente realiza la partida.

El cliente leerá se mantendra en lectura hasta detectar que el servidor ya no posee mas datos para enviar.

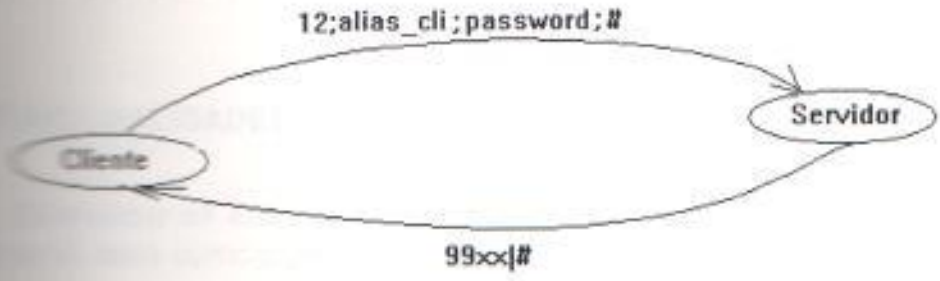
Mostar partidos (07):

Desuscribirse (10):

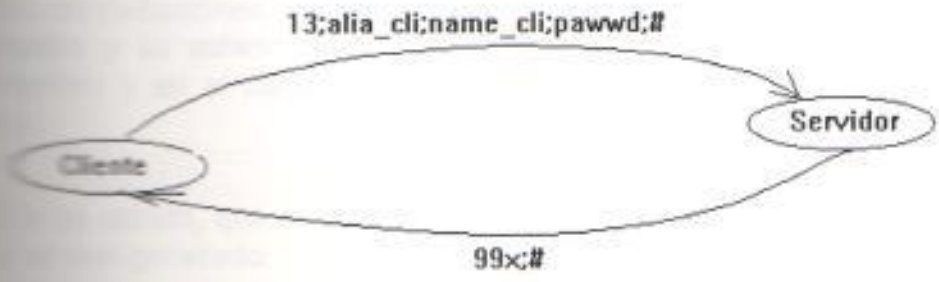


Traer Lista de Partidos del Suscriptor (11):

Logon(12)



Subscribirse (13):



3. DISEÑO DEL SERVIDOR

3.1 FUNCIONALIDADES DEL SERVIDOR

El servidor se encargará de mantener tablas con datos relacionados al juego en sí, tales como jugadores, invitaciones ,etc. Las tablas son :

1. Tabla de jugadores, con su nombre, alias y password.

2. Tabla de invitaciones, conteniendo a los jugadores que invitan, los que son invitados y su estado (esto es: si no recibe respuesta, si ya aceptó la invitación) y en el caso de haber terminado el juego registrará quien lo ganó.

3. Tabla de estado, que registre el estado actual de cada uno de los partidos que se han generado.

De esta forma podrá dar servicio al cliente respondiendo cada uno de sus requerimientos. Estas respuestas que dará el servidor se traducen en las siguientes funciones:

- Comprobar que el cliente está autorizado para jugar y verificar su contraseña.
- Inscribir a un jugador cuando el cliente lo solicite, asi pasa a formar parte del club de juego de damas.
- Enviar una lista de todos los jugadores inscritos cuando el cliente lo solicite
- Registrar una invitación cada vez que el jugador_a invita al jugador_b e informar sobre el particular al jugador_b.
- Registrar las jugadas que cada jugador realiza, esto es, mantener el estado actual de cada partido.
- Matar los partidos cuando cualquier jugador asi lo solicite, esto no implica que el jugador deje de formar parte del club de juego de damas.
- Eliminar a un jugador del club de juego de damas cuando lo solicite.

Tipo de Servidor y su Justificación

Puesto que queremos desarrollar un sistema "JUEGO DE DAMAS" en el que cualquier persona pueda jugarlo sin importar el lugar en el que se encuentre, necesitamos usar ciertos protocolos de comunicación que nos provea de mecanismos básicos para la transferencia de datos, y sobre todo, que dicha transferencia se realice de manera confiable y con un alto grado de performance, además que pueda trabajar tanto en redes LAN como en WAN, estos protocolos son los protocolos TCP/IP.

Además queremos que los jugadores se conecten en el momento que deseen, teniendo presente que usaremos los protocolos TCP/IP y considerando que tales protocolos no dictaminan cuándo o porqué las aplicaciones interactúan , debemos organizar nuestra aplicación en un ambiente distribuido. Quien nos ofrece este ambiente es el modelo CLIENTE-SERVIDOR, de tal manera que para implementar nuestro sistema "Juego de Damas" usaremos el modelo cliente-servidor.

Así tendremos un servidor que esté escuchando continuamente para determinar cuándo arriva un requerimiento de conexión y establecer el enlace.

Servidor no orientado u orientado a conexión?

Dentro de la serie de protocolos TCP/IP tenemos dos tipos de protocolos de transporte, estos son : TCP y UDP .

El protocolo de transporte TCP provee toda la confiabilidad necesaria para comunicarse a través de una red internet, sus características son las siguientes:

- Este verifica que los datos arriven y automáticamente retransmite los datos que no lo hacen.
- Contabiliza un checksum sobre los datos para garantizar que no fueron corrompidos durante la transmisión.
- Utiliza un número de secuencia para asegurar que los datos arrivan en orden y automáticamente elimina paquetes duplicados.
- Provee control de flujo para asegurar que el emisor no transmita datos más rápido de lo que el receptor pueda consumir.
- Informa a ambos el cliente y al servidor si la red llega a ser inoperable por alguna razón.

El protocolo de transporte UDP depende del hardware de la red y de los gateways intermedios. Si trabaja en una red de área local entonces lo harán correctamente. Pero si es en una WAN puede tener problemas puesto que los datos pueden ser perdidos, duplicados, retardados o distribuidos fuera de orden.

Con este antecedente y tomando en cuenta que a nuestro juego pueden conectarse de diferentes lugares en forma confiable entonces necesitamos usar un protocolo de transporte orientado a conexión, de esta manera utilizaremos TCP.

¿Servidores *Stateless o Statefull?*

Un servidor statefull es aquel que guarda información de estado acerca de las interacciones que está haciendo con el cliente, de esta forma se gana eficiencia pero no es muy confiable porque esa información puede llegar a ser incorrecta si los mensajes son perdidos, duplicados o distribuidos fuera de orden o si el cliente corta bruscamente la conexión y reinicia (reboot) la máquina. Un servidor stateless es aquel que no mantiene información de estado.

En nuestro caso haremos al servidor stateless debido a que este consume menos recursos que el statefull y puesto que será instalado en la Sun Sparc Station 2 de CESERCOMP debemos cuidar de sus recursos, ya que esta provee muchos servicios más, que necesitan de esos recursos (no queremos alterar el performance de los sistemas que en él están corriendo).

¿Servidor *Concurrente o Iterativo?*

El término CONCURRENCIA se refiere a un procesamiento simultáneo aparente.

El procesamiento concurrente es fundamental en un ambiente distribuido y puede ocurrir en muchas formas. Por ejemplo este ocurre entre máquinas sobre una red, en la cual muchas parejas de aplicaciones pueden comunicarse concurrentemente, "compartiendo la red que las interconecta". La concurrencia también ocurre tanto en CLIENTES como en SERVIDORES.

Para efectos de programación del cliente, este no necesita realizar ninguna consideración especial para desarrollar su CLIENTE CONCURRENTE puesto que es el sistema operativo quien se encarga de permitir que múltiples clientes invoquen al cliente concurrentemente.Los servidores para conseguir

concurrencia requieren de un esfuerzo considerable. El SERVIDOR CONCURRENTE se encarga de permitir que múltiples usuarios se conecten a él para solicitar un servicio. De esta forma atiende a varios usuarios simultáneamente, y lo puede hacer gracias a la ayuda que le provee el Sistema Operativo.

Los SERVIDORES ITERATIVOS manejan un solo cliente a la vez, lo que produce que existan varios clientes que tendrán que esperar un tiempo considerable para poder ser atendidos. En otras palabras, una vez que un cliente se conecta con el servidor, se apropia de él y no es sino, hasta que termina todas sus transacciones y lo libera, que otro cliente puede accederlo.

Como nuestro juego de damas debe permitir que varios clientes jueguen simultáneamente, debemos implementar un SERVIDOR CONCURRENTE.

DISEÑO DE LOS DATOS MANEJADOS EN EL SERVIDOR

Usaremos los siguiente tipos de datos:

Estructuras

sockaddr_in	Estructura que contiene una dirección IP y un número de puerto de protocolo
sockaddr_in.sin_family	tipo de dirección
sockaddr_in.sin_port	número del puerto del protocolo
sockaddr_in.sin_addr	dirección IP
sockaddr_in.sin_zero[8]	no usado (seteado a 0)

El cliente usará esta estructura como cli_addr, y el servidor como ser_addr.

Variables de *Tipo Entera*

argc	es el número de parámetros pasados a la función main
port	es el número del puerto
sockfd	es un descriptor entero para el socket maestro
newsockfd	es un descriptor entero para el socket esclavo
idProceso	es el identificador del proceso padre
clilen	longitud de la estructura cli_addr

Variables de Tipo Char

contiene los parámetros pasados a la función main

contiene la dirección ip del host

ARCHIVOS

cada línea contiene la lista de los jugadores inscritos en el club de damas

nombre_jugador ; partidos_ganados ; partidos_perdidos
nombre_jugador ; partidos_ganados ; partidos_perdidos

nombre_jugador ; partidos_ganados ; partidos_perdidos"

cada línea contiene las invitaciones hechas entre jugadores, si ha sido aceptada la invitación, el color elegido por el invitado

alias_invitador ; alias_invitado ; V
alias_invitador ; alias_invitado ; F

alias_invitador ; alias_invitado ; V

cada línea contiene el alias de quien le toca jugar primero,el alias del otro jugador,el color de quien le toca el turno, el tipo de ficha (si es azul o roja y simple o reina), la posición en X , la posición en Y

alias1 ; alias2 ; color1; tipo_ficha ; pos_x ; pos_y ; tipo_ficha ; pos_x ; pos_y

alias1 ; alias2 ; color1 ; tipo_ficha ; pos_x ; pos_y ; tipo_ficha ; pos_x ; pos_y

alias1 ; alias2 ; color1 ;tipo_ficha ; pos_x ; pos_y ; tipo_ficha ; pos_x ; pos_y

DISEÑO DE LA APLICACION SERVIDORA.-

Nuestro diseño se basa en un sistema del tipo Top-Down, con una función principal que se encarga de llamar a las otras, a medida que las necesite.

Algoritmo *PRINCIPAL*

1. Recibir los parámetros del host y del puerto
2. Crear un socket y enlazarlo a una dirección conocida para el servicio que será ofrecido.
3. Poner el socket en modo pasivo, habilitándolo para que sea usado por el servidor.
4. Hacer una llamada a *accept* para recibir el próximo requerimiento de un cliente.
5. Crear un nuevo proceso esclavo para manejar su respuesta. Use *fork()*
6. Si esta en el proceso padre, cerrar el socket del proceso hijo
7. Si No, cerrar el socket del proceso padre y llamar al algoritmo *cliente* para procesar el requerimiento solicitado.
8. Regresar al paso 4.

Algoritmo *RECV_CLI*

1. Crear un puntero a char *cadena* en donde almacenaremos el string enviado por el cliente.
2. Leer un caracter del socket, haciendo un read (socket, buff, buflen)
3. Si el caracter es diferente de '#', concatenarlo a *cadena*.
4. Si No, retornar *cadena* y terminar
5. Regresar al paso 2.

Algoritmo *CORTAR_CAMPO*

1. Recibe un string
2. Crear un puntero a char *cadena* en donde almacenaremos el comando enviado en el string.
3. Leer por caracteres hasta encontrar el delimitador de campo (;), y almacenarlo en *cadena*
4. Cortar del string dado lo que contiene cadena
5. Retornar *cadena* y terminar

Algoritmo *CLIENTE*

1. Recibir el socket del esclavo que procesará este requerimiento.
2. Obtener el string enviado por el cliente llamando al algoritmo *recv_cli*.
3. Obtener el comando del requerimiento llamando al algoritmo cortar_campo.
4. Si el comando es 13, registrar al jugador en el club de juego de damas. *registrar*
5. Si el comando es 1, Enviar la lista de los jugadores ya suscritos. *env_list*.
6. Si el comando es 2, Registrar invitación para un partido. *reg_inv*.
7. Si el comando es 3, Enviar invitaciones para iniciar un partido. *env_inv*.
8. Si el comando es 4, Registrar la respuesta a una invitación.*reg_resp_inv*.
9. Si el comando es 5, Recibir una jugada, *recv_jugada*
10. Si el comando es 6, Enviar jugadas, *env_jugada*.
11. Si el comando es 7, matar las partidas que el usuario desee. *matar_partido*.
12. Si el comando es 8, Saltar al paso 16.
13. Si el comando es 10, Desuscribirse del club de juego de damas. *desuscribir*
14. Si el comando es 12, Verificar la contraseña (el password) llamando al algoritmo Logon
15. Si el comando no es ninguno de los anteriores, Enviar un mensaje al cliente. *env_no_opc*.
16. Regresar al paso 2.
17. Concluir la sesión con ese cliente haciendo un *shutdown*.

Algoritmo *REGISTRAR*

1. Recibe un string
2. Crear un puntero a char *cadena* para almacenar cada uno de los campos
3. Leer un caracter del string y eliminarlo del string
4. Si el caracter es el delimitador de buffer (#), saltar al paso 8
5. De lo contrario, Si el caracter es diferente del delimitador de campo (;), almacenarlo en *cadena*
6. De lo contrario, añadir *cadena* y un delimitador de campo (;) al archivo players.txt
7. Regresar al paso 3.
8. Añadir un retorno de línea al archivo players.txt, luego cerrarlo .
9. Enviar un OK al cliente y retornar.

Algoritmo *ENV_LIST*

1. Recibe un string
2. Crear un puntero a char *lista_f* para almacenar los alias de los jugadores que no serán enviados ya que ellos ya fueron invitados por el solicitante o ellos ya lo invitaron.
3. Crear un puntero a char *cadena* para almacenar la lista de los alias y sus respectivos nombres que serán enviados al solicitante.
4. Obtener el *alias* del jugador que solicita la lista
5. Abrir el archivo invita.txt y ubicarse en la primera línea.
6. Si es fin de archivo, Saltar al paso 11
7. Leer una línea del archivo invita.txt y sacar *alias1* y *alias2* de esa línea
8. Si *alias1* es igual a alias, añadir alias2 en *lista_f*
9. Si *alias2* es igual a alias, añadir alias1 en *lista_f*
10. Regresar al paso 6
11. Cerrar el archivo invita.txt
12. Abrir el archivo players.txt y posicionarse en la primera línea.
13. Si es fin de archivo, Saltar al paso 18
14. Leer una línea del archivo players.txt y sacar alias1 y nombre1
15. *alias1* es diferente de alias y no consta en *lista_f*, añadir alias1 y *nombre1* a cadena
16. Añadir un delimitador de campo (;) a cadena
17. Regresar al paso 13
18. Cerrar el archivo players.txt
19. Añadir un delimitador de fin de buffer (#) a cadena
20. Enviar cadena al cliente y retornar.

Algoritmo *REG_INV*

1. Recibe un *string*
2. Obtener el *alias* del jugador que realiza las invitaciones y eliminarlo de *string*
3. Obtener el *alias1* del jugador a quien invita *alias* y eliminarlo de *string* .
4. Añadir una línea al archivo invita.txt que contenga *alias, alias1, F*
5. Enviar un OK al cliente y Retornar.

Algoritmo ***ENV_INV***

1. Recibe un *string*
2. Crear un puntero a char *lista* que guardará la lista de los *alias_invitador* que han invitado a *alias* y que aún no han recibido respuesta.
3. Crear un puntero a char *cadena* que guardará los *alias_invitador*, el delimitador, y su respectivo *nombre*
4. Obtener el alias del jugador que quiere revisar la lista de los jugadores que han invitado
5. Abrir el archivo invita.txt y obtener todos los *alias_invitador* que han invitado a *alias* y aún no han recibido respuesta, luego guardarlo en *lista*.
6. Abrir el archivo players.txt y para cada *alias_invitador* que se encuentre en *lista*, obtener su *nombre*, luego añadir: *alias_invitador*, delimitador de campo (:) y *nombre*, a *cadena*.
7. Añadir el delimitador de fin de buffer (#) a *cadena*.
8. Cerrar los archivos, invita.txt, players.txt
9. Enviar *cadena* al cliente y retornar

Algoritmo ***REG_RESP_INV***

1. Recibe un string
2. Abrir los archivos invita.txt y damas.txt
3. Obtener el alias del jugador que envía respuestas a las invitaciones y eliminarlo del *string*
4. Si encontró el delimitador de fin de buffer (#), Saltar al paso 9
5. Obtener *alias1* y *color1* del *string* y luego eliminarlo del *string*.
6. Buscar en el archivo invita.txt la línea que contiene *alias1* ; *alias* , y en el mismo campo, en lugar de F ponga V .
7. Añadir una línea al archivo damas.txt que contenga el "estado inicial del nuevo partido", es decir, el alias de quien le toca jugar de acuerdo al color escogido por el invitado, el alias del otro jugador, el color de quien le toca jugar primero, el tipo de ficha, pos_x, pos_y..., con sus respectivos delimitadores de campo.
8. Regresar al paso 4
9. Cerrar los archivos invita.txt y damas.txt .
10. Enviar un OK al cliente.

Algoritmo *RECV_JUGADA*

1. Recibe un *string*
2. Abrir los archivos invita.txt y damas.txt
3. Obtener el *alias1* y el *alias2* del *string*
4. Verificar en el archivo invita.txt si existe un partido entre *alias1* y *alias2* .*Si existe continuar, sino saltar al paso xxx*
5. Añadir un línea al archivo damas.txt eliminando los delimitadores de fin de campo y de fin de buffer finales.
6. Cerrar los archivos invita.txt y damas.txt .
7. Enviar un OK al cliente.

Algoritmo *ENV_JUGADA*

1. Recibe un *string*.
2. Declarar un puntero a char *cadena* que guardará las jugadas de los diferentes partidos en los que está jugando *alias*
3. Obtener el *alias* del jugador que solicita el envío de las jugadas
4. Abrir el archivo damas.txt y posicionarse al principio del archivo.
5. Buscar en el archivo damas.txt la siguiente línea que contengan como primer alias el *alias* obtenido en el paso anterior y guardarla en cadena.
6. Si encuentra la línea anterior eliminarla del archivo , enviarla al cliente y saltar al paso 5 ,sino continuar.
7. Retornar

Algoritmo *DESUSCRIBIR*

1. Recibe un *string*
2. Obtener el *alias* de quien quiere desuscribirse del club de juego de damas
3. Buscar en el archivo damas.txt todas las líneas que contienen a *alias* y eliminarlas, luego por cada línea eliminada añadir una línea que contenga, *alias, alias,color_cualquiera, codigo (9905).*
4. Eliminar del archivo invita.txt la línea que contenga *alias*
5. En el archivo players.txt eliminar el jugador *alias*
6. Retornar

3.3.13 *Algoritmo MATAR_PARTIDOS*

1. Recibe un *string*
2. Obtener el *alias* de quien quiere matar los partidos, y eliminarlo de *string*
3. Si obtiene el delimitador de fin de buffer (#), saltar al paso 9
4. Obtener *alias1* y eliminarlo de *string*.
5. Buscar en el archivo damas.txt la línea que contiene *alias* y *alias1* y eliminarla, luego añadir la línea que contenga,*alias1,alias,color cualquiera,codigo (9905).*
6. Eliminar del archivo invita.txt la línea que contenga *alias* y *alias1*
7. En el archivo players.txt, aumentar en uno el tercer campo (partidos_ganados) del jugador *alias1*, y aumentar en uno el cuarto campo (partidos_perdidos) del jugador *alias*
8. Regresar al paso 3
9. Retornar

3.3.14 *Algoritmo ENV_NO_OPC*

1. Enviar el siguente comando al cliente: "9999 | #"
2. Retornar

3.3.15 *Algoritmo LOGON*

1. Recibe un string
2. Obtener el *alias* y el *password* del jugador que quiere jugar.
3. Abrir el archivo players.txt y obtener la linea que empieza con *alias*
4. De la linea anterior comparar el tercer campo con *password. Si son iguales enviar al cliente un OK, de lo contrario enviar un codigo de error.*
5. Retornar

3.4 COMPROMISOS DEL DISEÑO DEL SERVIDOR

En el servidor se utilizarán archivos para el almacenamiento de todos los datos concernientes a los jugadores y el estado de sus partidos. Con esto se pretende facilitar la administración del servidor y dar mayor facilidad al desarrollo de la aplicación.

Todas las transacciones hechas por el servidor serán realizadas de manera coherente para evitar conflictos con los datos de los jugadores, se mantendrá un estricto control de acceso para dar mayor seguridad a los jugadores y sus partidos evitando asi filtraciones de información.

Entre los archivos que usaremos tenemos a players.txt en el cual se guardarán los usuarios, sus nombres, sus contraseñas y el número de partidos ganados y perdidos.

El archivo invita.txt que contiene las distintas invitaciones realizadas entre los jugadores, asi como también la respuesta dada, sea esta afirmativa o negativa, y el color escogido por el jugador invitado.

Finalmente tendremos el archivo damas.txt quien se encargará de mantener el estado en el que se encuentran todos los partidos existentes en el juego, y para cada partido se indicará el jugador a quien le corresponde el turno de juego junto con su color. En este mismo archivo se encontrarán los partidos que hayan sido eliminados por cualquiera de los jugadores propietarios de aquel partido.

Para la comunicación con el cliente se definirá un protocolo de comunicación orientado a cadena de caracteres, el cual será armado por el cliente para realizar todos sus requerimientos. De igual forma el servidor retornara al cliente los datos solicitados por el mismo, un código para indicarle que su solicitud ha sido realizada, o un código de error. Todo esto se encuentra detallado en el protocolo de comunicaciones mencionado anteriormente.

ADMINISTRACIÓN DEL SERVIDOR

Realmente la tarea que tendrá el administrador del sistema para controlar y mantener el servidor de damas será mínima, ya que todas las transacciones se las realiza de manera automática con el cliente. Por ejemplo , cuando el cliente hace un requerimiento de suscripción, el servidor obtiene el string de datos enviado por el cliente, realiza la suscripción y le devuelve un mensaje al cliente para indicarle que su pedido fue ejecutado.

Entre los puntos que tendrá el administrador que realizar se encuentran los siguientes.

1. Verificar la correcta configuración del puerto de acuerdo a los pasos indicados en el MANUAL DE USUARIO DEL SERVIDOR.

2. Levantar el servidor indicando la dirección IP y el puerto que va a ser utilizado por el mismo.

3. Informar a los usuarios clientes del puerto que deberán utilizar para conectarse con el servidor de damas, y sobre todo , tener presente que ese puerto debe ser utilizado única y exclusivamente para brindar ese servicio, evitando asi conflictos con otros servicios.

4. Tomar en cuenta que el servidor utiliza un número de archivos temporales para ejecutar ciertas transacciones con los clientes , los mismos que son borrados automáticamente por el servidor una vez que haya terminado su transacción con aquel cliente. Estos archivos temporales toman diferntes nombres de acuerdo al cliente con el que estén trabajando.

5. Si por alguna razón, los archivos temporales no se están borrando, entonces el administrador deberá hacer limpieza con determinada frecuencia, para esto, primero debe bajar el servidor , luego verificar que solamente deben existir los siguientes archivos:

   damas.exe

   players.txt

   invita.txt

   damas.txt

   El resto tendrá que ser borrado.

6. Todos los archivos .txt están conformados por líneas, las mismas que contienen un determinado número de campos, los cuales a su vez, están separados por el delimitador de punto y coma **(;)**.

ADMINISTRACIÓN DEL SERVIDOR

Realmente la tarea que tendrá el administrador del sistema para controlar y mantener el servidor de damas será mínima, ya que todas las transacciones se las realiza de manera automática con el cliente. Por ejemplo , cuando el cliente hace un requerimiento de suscripción, el servidor obtiene el string de datos enviado por el cliente, realiza la suscripción y le devuelve un mensaje al cliente para indicarle que su pedido fue ejecutado.

Entre los puntos que tendrá el administrador que realizar se encuentran los siguientes.

1. Verificar la correcta configuración del puerto de acuerdo a los pasos indicados en el MANUAL DE USUARIO DEL SERVIDOR.

2. Levantar el servidor indicando la dirección IP y el puerto que va a ser utilizado por el mismo.

3. Informar a los usuarios clientes del puerto que deberán utilizar para conectarse con el servidor de damas, y sobre todo , tener presente que ese puerto debe ser utilizado única y exclusivamente para brindar ese servicio, evitando asi conflictos con otros servicios.

4. Tomar en cuenta que el servidor utiliza un número de archivos temporales para ejecutar ciertas transacciones con los clientes , los mismos que son borrados automáticamente por el servidor una vez que haya terminado su transacción con aquel cliente. Estos archivos temporales toman diferntes nombres de acuerdo al cliente con el que estén trabajando.

5. Si por alguna razón, los archivos temporales no se están borrando, entonces el administrador deberá hacer limpieza con determinada frecuencia, para esto, primero debe bajar el servidor , luego verificar que solamente deben existir los siguientes archivos:

   damas.exe

   players.txt

   invita.txt

   damas.txt

   El resto tendrá que ser borrado.

6. Todos los archivos .txt están conformados por lineas, las mismas que contienen un determinado número de campos, los cuales a su vez, están separados por el delimitador de punto y coma **(;)**.

Las líneas del archivo players.txt contienen 5 campos, estos son:

alias;nombre;password;partidos ganados;partidos perdidos

En este se encuentran todos los jugadores que se han suscrito al clud de juego de damas.

Las líneas del archivo invita.txt contienen los siguientes campos:

alias del que invita ;alias del invitado;respuesta

el campo *respuesta* será V o F dependiendo de si ha sido aceptada o no la invitación.

Las líneas del archivo damas.txt contienen el alias del jugador al que le tocaría jugar , luego el alias del jugador que estaría en espera, el color de quien le toca el turno y finalmente las fichas y sus posiciones de acuerdo a un cierto formato.

Solamente en el caso de que uno de los jugadores decide matar un partido, aparecerá una línea conteniendo los dos alias y un código (9905), el mismo que sirve para informarle al otro jugador que su partido ha sido muerto por su contrincante. Una vez informado sobre este particular, la línea desaparece del archivo.

CAMBIO DE PASSWORD.- Si por algún motivo uno de los jugadores desea cambiar el password, entonces el administrador deberá ir al archivo players.txt y cambiar el tercer campo por el nuevo password en la línea que corresponde al usuario que lo solicita.

4. DISEÑO DEL CLIENTE

4.1 FUNCIONALIDAD DEL CLIENTE

El CLIENTE será realizado en Visual Basic versión 3.0, por presentar gran comodidad de programación, así como el hecho de que ya sabíamos manejarlo, esto se debe a que la programación esta orientada a eventos, permite definir funciones, así como tipos de datos, facilidad en el manejo de archivos y de la base de datos en ACCESS 1.1.

El CLIENTE será implementado de tal manera que este use eficientemente el espacio en disco, es decir se eliminara código redundante, para lo cual se hará uso de funciones, manteniendo así funcionalidad.

Para la implementaron del cliente, se usará un archivo de inicialización, tarmas.ini, donde se especifica el número de puerto y dirección ip del servidor, lo cual hace que este sea portable, y fácil de mantener en caso de que el número de puerto o dirección del servidor sea alterada, así como el path del directorio donde se almacenan las tablas de la base.

Las opciones del menú estarán disponibles al cliente conforme este se introduzca en el juego.

El manejo de datos de las jugadas serán almacenado en tablas, lo que permitirá que el acceso a las jugadas sea rápida y eficiente.

Para manejar con mayor comodidad los errores que se presentasen durante el uso de las funciones de WINDOWS SOCKET, así como los obtenidos durante la transmisión o implementación, se usara una tabla, error_code. Esta tabla permitirá obtener el significado del error obtenido, lo que hace que futuros errores que surgiesen sean ingresados en esta, evitando así que la codificación sea alterada.

Las posiciones de las fichas en el tablero de los juegos serán almacenadas en un tabla de entradas, fichas; luego de realizar el movimiento, estas serán trasladadas a una tabla de salida, fichas_out.

Los movimientos de las fichas dentro del tablero serán validadas en lo permisible, así como que el usuario no realice mas de un movimiento consecutivo.

Será capaz de mantener información de varios partidos que el usuario desea mantener simultáneamente.

Deberá establecer la conexión con el servidor, así como la consistencia de los datos que serán enviados según el protocolo propuesto. Mantendrá de igual manera, la autenticación y autorización del usuario.

Proveerá al usuario de ayuda en línea en cada una de las opciones posibles del juego, así como de un archivo de ayuda.

4.2 DISEÑO DE DATOS MANEJADOS EN EL CLIENTE

4.2.1 *Archivo DAMAS.INI*

Contiene variables de inicialización del sistema, como son:

well_port

host_ip

definen las características de la comunicación como son: numero de puerto donde escucha el servidor y dirección del servidor(en notación decimal).

PATH

que contiene el directorio donde se almacena la tabla de datos, así como otros archivos propios de la aplicacion cliente.

4.2.2 *Base de Datos DAMAS.MDB*

Usada para almacenar datos sobre los diferentes juegos pendientes que pueda tener el jugador, códigos de error, invitaciones recibidas.

Tabla FICHAS, FICHAS_OUT

Perteneciente a la base de datos damas.mdb, sus campos son:

usuario	String(8)	Alias del contrincante
ficha	Integer	Identifica tipo de ficha: 1=roja,2=azul
posx	Integer	posición en x de la ficha en el tablero
posy	Integer	posición en y de la ficha en el tablero
reina	Integer	bandera: 1=reina, 0=no reina

TABLA ERROR_CODE

Contiene el código de error y su respectiva descripción.

err_code	string (5)	Código de error
descripcion	string(100)	Descripción del error

esta tabla contiene tanto posibles errores de las funciones de WINSOCK como errores definidos en la aplicación.

TABLA INVITACIONES

Contendrá las invitaciones que el usuario hubiera recibido de otros suscriptores y que no han sido contestadas.

alias: string(8) Contiene el alias del suscriptor del cual se obtuvo la invitación.
name: string(35) Nombre del suscriptor.

TABLA CONTRIN

En esta tabla se almacena el nombre al contrincante con quien el usuario tiene partidos en los que puede realizar jugada con otros jugadores, color de ficha con la que el usuario mantiene la partida, asi como el estado de si se ha efectuado el movimiento o no.

alias	string(8)	Alias del jugador con quien el usuario mantiene partido.
color	integer	0 si el usuario juega con las rojas y 1 si es con azul.
val	boolean	True si el movimiento de la jugada ya fue efectuado, False si aun no se efectua

Las datos de las tablas de un usuario específico son devueltas al servidor si este termina la ejecución del programa o si se decide cambiar de usuario.

4.2.2 *Estructuras*

Estructura Tipoficha

Esta estructura sirve para llevar el estado de las fichas en el momento mismo de la jugada con un contrincante específico, con estos datos se realizan las validaciones necesarias para hacer de una jugada válida. Esta estructura es llenada de datos(traidos de una base de dato) antes de la jugada, y luego terminada misma los valores almacenados en la estructura son copiados en la base de datos.

coordx	Integer	*posición en x de la ficha en el tablero
coordy	Integer	*posición en y de la ficha en el tablero
reina	Integer	bandera: 1=reina, 0=no reina
estado	Integer	bandera: 1=jugando,0= perdida

[illegible]_alias

Define el alias del cliente, con su respectivo nombre.

[illegible] String(8) Contiene el alias del cliente

[illegible] String(35) Define el nombre del cliente

[illegible] ***Variables Globales***

NOMBRE	TIPO	DESCRIPCION
[illegible]	Single	distancia entre los contornos d un cuadro del tablero y la ficha (objeto).
[illegible]	Single	dimensión de un cuadro del tablero
		es la ordenada de la esquina superior izquierda del tablero con respecto a la forma ficha.frm
[illegible]	String	alias del contrincante
[illegible]	String	valor índice de los arreglos (ficha_roja, ficha_azul) escogido para la jugada
[illegible]	String	tipo de ficha escogida a la que le toca jugar
[illegible]	String	guarda el tipo de ficha a la que le toca jugar
[illegible]	Tipo_ficha	almacena estado actual de las fichas azules
[illegible]	Tipo_ficha	almacena estado actual de fichas rojas
[illegible]	String	contiene el delimitador de campo
[illegible]	String	contiene el delimitador de fin de buffer
[illegible]	Integer	contiene la cantidad de partidos en los que el usuario se encuentra
[illegible]	Integer	contiene cantidad de invitaciones recibidas por el usuario
[illegible]	Integer	cantidad de jugadores a los que se puede invitar
[illegible]	String	contiene el nombre del archivo de inicialización
[illegible]	String	contiene el nombre de la tabla de errores
[illegible]	String	contiene el path del directorio del cliente
[illegible]	String * 8	contiene alias de partidos posibles a matar

4.2 DISEÑO DE LA APLICACION CLIENTE

4.2.1 *SUSCRIBIRSE*

1. Ingreso el nombre del usuario, alias a usar y password.
2. Si alguno de los campos a ingresar se encuentra en blanco o el alias comienza con caracter numérico, presenta mensaje de error respectivo, regrese al 1
3. Armar buffer de envío.
4. Enviar al servidor el buffer
5. Recibir el buffer del servidor
6. Obtener el código del buffer recibido, su significado y presentarlo.
7. Actualizar archivo de inicialización "damas.ini"
8. Actualizar opciones del menú.

4.2.2 *VERIFICACION DE USUARIO*

1. Ingrese el password con el que se ha suscrito.
2. Armar buffer de envío al servidor.
3. Enviar el buffer a servidor.
4. Recibir buffer de verificación del servidor.
5. Obtener el código de respuesta.
6. Si el código es 9910, presentar menú principal.
7. Si el código es 9911, regrese al paso 1.

4.2.3 *RECIBIR LISTA DE JUGADORES*

1. Armar buffer de envío.
2. Enviar al servidor el buffer
3. Recibir el buffer del servidor.
4. Si el buffer recibido comienza con 99, obtener significado y presentarlo, continuar al paso 6.
5. Desencadenar el buffer recibido y almacenarlo en el arreglo.
6. Actualizar archivo de inicialización y las opciones del menú.

INVITAR A ALGUIEN A JUGAR

1. Seleccionar el alias con quien se desea mantener partido.
2. Si desea seleccionar otro jugador, regresar al paso 1.
3. Si se desea enviar la lista y no se ha seleccionado algún alias ir al paso 8 .
4. Armar el buffer a enviar.
5. Enviar el buffer al servidor.
6. Recibir buffer del servidor.
7. Obtener el código contenido en el buffer, seleccionar el significado y presentarlo.
8. Actualizar opciones del menú.

RECIBIR INVITACIONES

1. Armar el buffer con código y alias del usuario.
2. Enviar el buffer al servidor.
3. Recibir el buffer del servidor.
4. Si se trata el buffer de un código de error, obtener el significado del código y presentarlo, ir al paso 6.
5. Obtener el alias y nombre de cada invitación en el buffer y almacenarlos la tabla invitaciones.
6. Actualizar archivo "damas.ini" y opciones del menú.

ENVIAR RESPUESTA DE INVITACION

1. Mostrar invitaciones recibidas que son cargadas de la tabla de invitaciones.
2. Seleccionar la invitación a la que se desea aceptar y el color con el que se desea jugar y eliminarlo de la lista.
3. Añadir al buffer el alias seleccionado, si se desea seleccionar otro regresar al paso 2.
4. Enviar el buffer al servidor.
5. Recibir el buffer del servidor.
6. Obtener código de retorno del buffer, buscar su significado en la tabla de error_code y mostrarlo.
7. Si el código obtenido es 9910, actualizar tabla de invitaciones, eliminar las invitaciones aceptadas.

ENVIAR JUGADA

1. Obtener la jugada de la tabla ficha_out y de la de contrin y armar el buffer.
2. Enviar al servidor el buffer.
3. Recibir del servidor el buffer.
4. Obtener código del buffer.
5. Si se trata del código 9908 continuar al paso 8.
6. Si se trata del código 9910 eliminar la jugada de la tabla de fichas_out y de contrin.
7. Si se trata de error en comunicación ir al paso 9.
8. Regresar al paso 1 hasta que la tabla este vacía.
9. Actualizar opciones del menú.

RECIBIR JUGADA

1. Armar buffer con código de opción y alias.
2. Enviar el buffer al servidor.
3. Recibir del servidor el buffer. si el buffer esta vacío ir al paso 6.
4. Levantar información a las tablas de fichas_in y contrin.
5. Actualizar archivo de inicialización, damas.ini, y las opciones del menú.
6. Salir.

MATAR PARTIDOS

1. Armar buffer de recibir lista de partidos en los que el usuario se encuentra involucrado (11;alias;#).
2. Enviar al servidor el buffer.
3. Recibir el buffer del servidor.
4. Si el buffer recibido es del tipo error, ir al paso .
5. Armar lista (arreglo) con los alias recibidos del buffer.
6. Presentarlos en pantalla.
7. Seleccionar el partido a matar.
8. Añadirlo al buffer.
9. Si se desea añadir otro partido, ir al paso 7.
10. Enviar el buffer al servidor.
11. Recibir buffer del servidor.

Obtener código del buffer, buscar significado de la tabla y mostrarlo.

DESUSCRIBIRSE

Armar buffer conteniendo el código y el alias.
Enviar el buffer hacia el servidor.
Recibir buffer del servidor.
Obtener código del buffer y buscar su significado en la tabla de error_code y presentarlo.
Si el código es 9910, actualizar archivo.ini y las opciones del menú.

ALGORITMO GRABAR TABLERO

Abrir base de datos *damas.mdb*
Abrir las tablas: *fichas_in, fichas_out*
Borrar todos los datos con el alias del contrincante de la tabla *ficha_in*
Tomar primer registro de estructura *f_roja*
Si estado de la ficha *f_roja(indice).estado=1* entonces crear un registro con los datos de la ficha en la tabla *ficha_out*
Tomar siguiente registro de estructura *f_roja* e ir al paso 5 hasta terminar de leer la estructura
Tomar primer registro de estructura *f_azul*
Si estado de la ficha *f_azul(indice).estado=1* entonces crear un registro con los datos de la ficha en la tabla *ficha_out*
Tomar siguiente registro de estructura *f_azul* e ir al paso 5 hasta terminar de leer la estructura

ALGORITMO CARGAR_TABLERO

Abrir base de datos *damas.mdb*
Abrir la tabla *fichas_in*
Inicializar las estructuras *f_roja y f_azul*
Tomar primer registro de tabla *fichas_in*
Si el registro tomado corresponde al contrincante deseado tomar estos datos y llenarlos en la estructura *f_roja o f_azul*
Posicionar en pantalla la ficha
Tomar siguiente registro de la tabla y repetir paso 5 hasta llegar al final de la misma

ALGORITMO VALIDAR_JUGADA

1. Verificar si corresponde turno al jugador, sino ir a paso 10
2. Validar la ficha que hizo la jugada, si es roja ir a paso 3 sino al paso 6
3. Obtener las posiciones inicial(xi,yi) y final(xf,yf) del movimiento de la ficha
4. Si el ABS(xf-xi)=1 y (yf-yi)=1 entonces mover la ficha a la posición final, actualizar la estructura *f_roja(indice actual)* y retornar.
5. Si el ABS(xf-xi)=2 y (yf-yi)=2 entonces verificar si ficha entre posición final e inicial es del contrincante, si lo es entonces ocultar ficha contrincante, actualizar la estructura *f_azul(ficha comida)*, actualizar estructura *f_roja* con posición final y retornar
6. Obtener las posiciones inicial(xi,yi) y final(xf,yf) del movimiento de la ficha
7. Si el ABS(xf-xi)=1 y (yf-yi)=1 entonces mover la ficha a la posición final, actualizar la estructura *f_azul(indice actual)* y retornar.
8. Si el ABS(xf-xi)=2 y (yf-yi)=2 entonces verificar si ficha entre posición final e inicial es del contrincante, si lo es entonces ocultar ficha contrincante, actualizar la estructura *f_roja(ficha comida)*, actualizar estructura *f_azul* con posición final y retornar

COMPROMISOS DEL DISEÑO DEL CLIENTE

Por la facilidad de programación, se ha tenido que utilizar tablas, lo que trae que estas a pesar de estar vacías ocupen espacio en disco, lo cual se hubiese evitado si se usaban archivos, los que solo existirían si estos contuviesen datos, pero por otra parte el uso de archivos introduciría complejidad en la programación en el momento de eliminar columnas de estos.

El hecho de utilizar un archivo de inicialización, damas.ini, hace que la información que este contiene pierda confiabilidad, ya que pueden ser editados o alterados fácilmente.

Permite que una persona se pueda cambiar en un momento dado de usuario sin necesidad de salir y volver a ejecutar el programa cliente.

Si un usuario desea conectarse a un servidor para el cual deberá realizar una llamada telefónica o conectarse mediante una tarjeta de red, se deberá configurar el software DISTINCT de acuerdo al servidor o enlace a usar.

DISEÑO DE INTERFACES DEL USUARIO

En el diseño de las interfaces con el usuario, se ha tratado de mantener el estándard establecido en WINDOWS.

Las opciones del Menú seran visualizadas conforme el usuario haya navegado por estas. Es decir si para realizar la ejecución de una opción se debe previamente haber ejecutado otra, la primera no podrá ser accesada por el usuario. Además se dispone de hot-keys para las opciones que más frecuentemente son usadas.

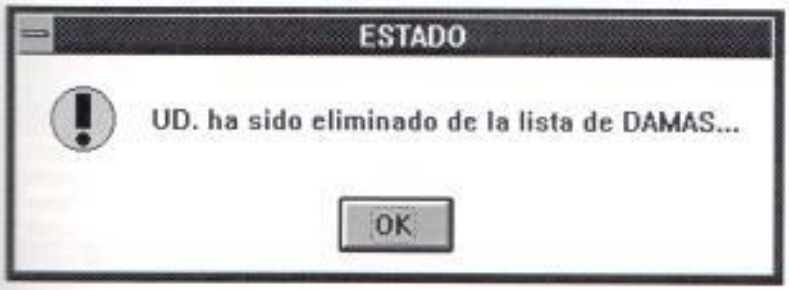
Se contará tambien con una Ayuda, tanto acerca de la realización de la aplicación así como del contenido de cada una de las opciones y campos de la aplicación.

Se dispone de una barra de estado, que se visualizará en la parte inferior de la pantalla, en la que se mostrará una ayuda en línea del campo sobre el cual el cursor del mouse se encuentre.

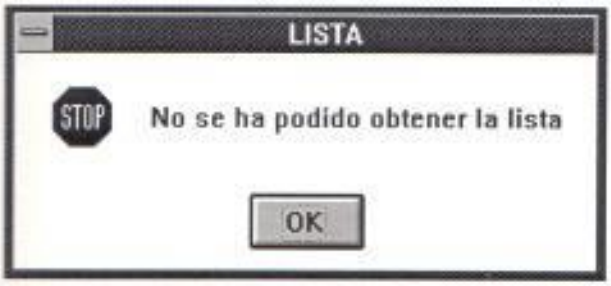
Las pantallas con que se disponen durante la aplicación tendrán un título descriptivo de esta. Tendrán deshabilitados los botones de maximización y minimización.

Las pantallas de indicación de estado, errores, poseerán: título de pantalla a la que corresponde el error obtenido, icono representativo de la gravedad del error causado y una breve descripción de este.

Las pantallas de cada opción, así como de los mensajes al ser mostrados al usuario serán visualizados en el centro del monitor.


ESTADO
UD. ha sido eliminado de la lista de DAMAS...
OK


SUSCRIBIRSE
No se ha especificado el alias del cliente
OK


LISTA
STOP
No se ha podido obtener la lista
OK


INVITAR A JUGADOR
Alias a Invitar
Alias Invitados
cpu
KARINA
MATAGATO
Nombre
Jugador: STEFANIE ALBAN
Aceptar
Cancel

MANUALES

MANUAL DEL USUARIO DE LA APLICACION CLIENTE

Instalación

Previo a la instalación, se deberá tener en cuenta que para obtener un buen rendimiento en la ejecución de la aplicación, se deberá contar con los siguientes requerimientos básicos:

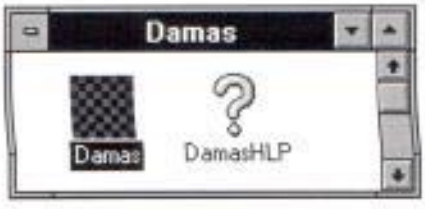
- Windows versión 3.0 o superior.
- Mouse
- Espacio disponible en disco mínimo 2M.

Se deberá ejecutar desde el File Manager el comando ***a:\instalar.exe***, o realizar doble click en el nombre del archivo.

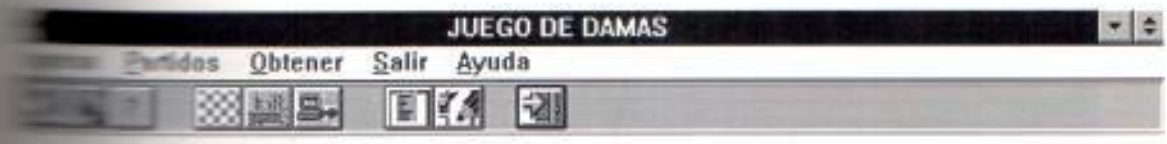
Como resultado de esto, se le solicitara el nombre del directorio destino. Una vez ingresado este, presione continuar.

Como iniciarse en el Juego DAMAS?

Presione consecutivamente dos veces el botón izquierdo del Mouse sobre el icono DAMAS del grupo DAMAS.



La pantalla que aparecerá es la que contiene el menu principal, como se muestra en la figura inferior.



La mayoría de las opiones del menú en este momento no se encuentran aún disponibles. Sólo las opciones de Suscribirse, Usuario, Salir y el Help se encuentran disponibles. Las emás opciones del menú estarán disponibles cuando Ud. haya ejecutado la opción Suscibirse o Usuario satisfactoriamente.

Suscribirse

Para suscribirse al juego de DAMAS, Ud. cuenta con 3 métodos.

- Seleccionar del menú principal la opción Ingreso y de este Suscribirse.
- Presionando las teclas ***Ctrl+S***.
- Seleccionarlo de la barra de Herramientas (el primer botón a la izquierda).

selecionada la opcion de culquiera de los métodos antes descritos, ingresar sus datos como son: nombre, alias con el cual desea ser en el juego, y la clave de acceso.

digite su clave, en lugar de esta, aparerá en ella una mascará (*) que ninguna otra persona pueda verla. Asegurese de que la clave sea fácil de recordar pero al mismo tiempo difícil para las demas de obtener.

crea que los datos ingresados son corectos presione el botón de si ya no desea ingresar o suscribir un usuario presione el botón de

Conectarse con un usuario

Ud. ya se encuentra suscrito al juego, podra usar la opción de Usuario del submenú de Ingreso del menú principal, o presionndo la tecla ***Ctrl+U***. Para así poder realizar todas las tareas que considere necesarias. Al realizar esta acción correctamente, le permitirá visualizar las demás opciones del menú y podrá hacer uso de ellas.

Una vez seleccionada la opción, una ventana como la que se muestra a continuación le será presentada. Se deberán ingresar tanto el alias como la clave con la cual se suscribio, y presionar el botón de Aceptar.

Si los datos que ingreso son correctos, podrá Ud. hacer uso de las demás opciones del menú.

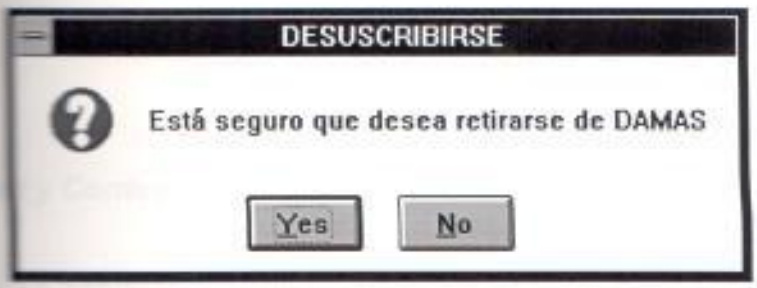
Desuscribirse

Esta opción se encuentra disponible una vez se haya ejecutdo la opión de Registro de Suscribirse satisfactoriamente.

Para ejecutar esta opcón se disponen de 3 métodos para ello.

- Seleccione del menú principal la opción Ingresar y de este Desuscribirse.
- Presionando las teclas ***Ctrl-D***.
- Realizando doble Clik sobre el 2do botón de la izquierda de la barra de Herramientas (al posicionar el mouse sobre esta aparecerá una leyenda indicando su función).

Si está seguro de que desea desuscribirse presione Yes o de lo contrario No en la casilla que se muentra (que le será visualizada).



El resultado de la operación será mostrada.

Determinar el usuario en uso

algún motivo Ud. olvidó con que usuario está trabajando, puede obtener de el alias en uso de cualquiera de una de las siguientes mneras.

- Presionendo el 3er botón de la izquierda de la barra de Herramienta (donde al pocisionar al mouse sobre esta aparece la leyenda "Mi alias").
- Selecione del menú principal la opcion Ingreso y de este "Mi alias".
- Presione las teclas Ctrl+M.

seleccionada cualquiera de las alternativas antes expuestas, aparecerá ventana proporcionandole la información deseada.

y Contestar Invitaciones Recibidas

se disponen de tres métodos para ello.

- Seleccionando del menú principal la opción Obtener y de este invitaciones.

- Realizar doble Clik sobre el 2do botón a la derecha de la barra de Herramientas.

- Presionando las teclas ***Ctrl+I***.

Si dispone de invitaciones que otros suscritores le hayan solicitado y que estas hayan sido aceptadas, estos serán visualizados en una ventana como la que se muestra a continuación.

Luego seleccione de la lista de Contrincantes el oponente al cual se desea aceptar, el nombre del mismo podra ser visualizado en la parte inferior de dicha pantalla, realice doble Clik, lo cual hara que inmediatamente este este sea trasladado a la lista de Seleciconados. Seleccione también el color con que Ud. desea jugar antes de realizar el doble Clik.

Realice la operación anterior con cada uno de los oponentes a los cuales se les desea responder.

Si por cualquier motivo Ud. decide no responder a uno de los alias ya seleccionados antes de presionar Aceptar, estos pueden ser trasladados a la lista de Contrincante de la misma manera en la que fueron colocados en la lista de Seleccionados.

aceptar los oponentes ya seleccionados presione Aceptar.

regresar al menu principal presione Cancelar.

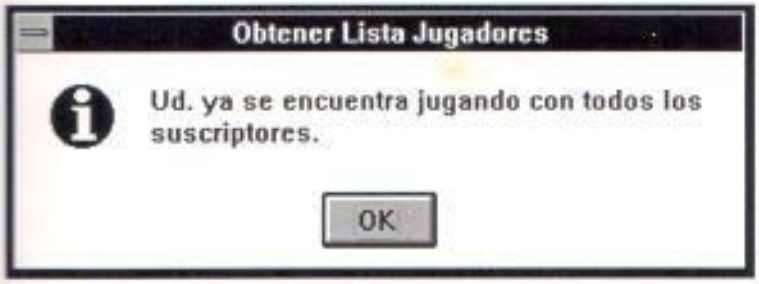
Invitar a otro Suscritor a mantener partido

de tres métodos para ello.

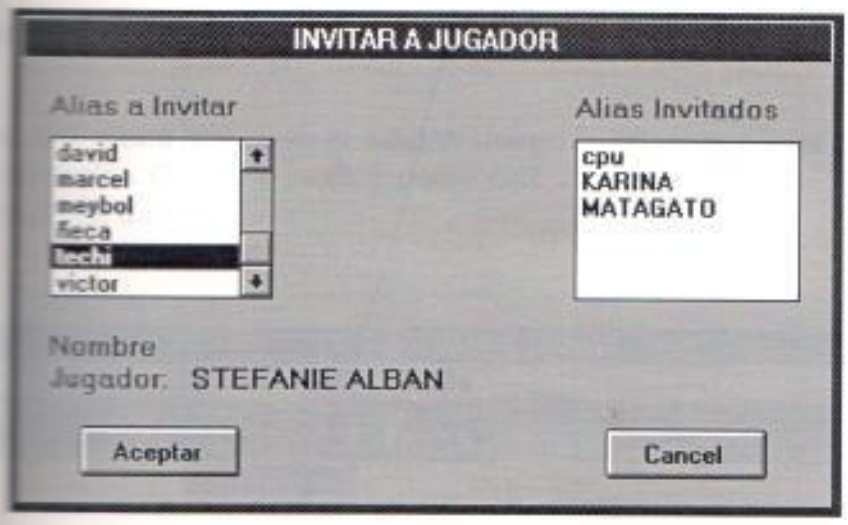
- Seleccionando del menú principal la opción Obtener y de este Jugadores a Invitar.
- Realizar doble Clik sobre el 2do botón a la derecha de la barra de Herramientas.

- Presionando las teclas ***Ctrl+J***.

ya se encuentra jugando con todos los suscritores, le será preentada como se muestra en l figura inferior.



no se encuentra jugndo con todos os suscritores, Ud. podrá seleccionar de ellos de lapantalla que le será mostrada.



Para seleccionar un contrincante al que se desea invitar, realice doble click en el alias que desea invitar de la lista de Alias a Invitar, este será trasladado a la lista de Alias Invitados.

Si desea eliminar alguno de los usuarios ya selecionados, realice la operación inversa a la de selecionar, lo que hará que dicho usuario sea trasladado nuevamente a la lista de Alias a Invitar.

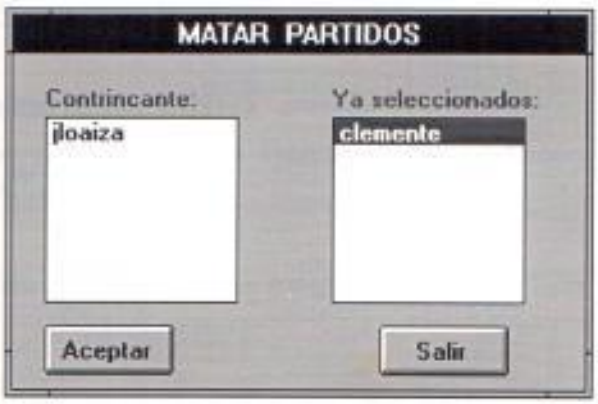
Cuando Ud. decida que ya han sido seleccionados todos los alias a quien desea invitar, presione Aceptar. Si desea anular o abandonar la operación presione Cancel.

Matar Partidos

Seleccione del menú principal la opción Juego y de este Matar Partidos o presione las teclas ***Ctrl-K,*** o realice doble Clik sobre el botón de la barra de herramientas que dice "kill".

Si usted posee algun partido con otro suscritor, le será visulizada la pantalla siguiente. Seleccione de la lista contrincante aquel con el cual se desea matar el partido realizando doble Clik sobre el alias de la lista de Contrincante y luego presione el botón Matar.

Si por cualquier motivo desea ya no matar algún partido, realice la operación anterior en orden inverso, es decir doble Clik sobre el alias de la list de Ya seleccionados, lo quehará que este regrese nuevamente a la lista de Contrincante.



Una vez que ya haya seleccionado todos los partidos que desea eliminar, presione el botón Aceptar. Si desea anular la operación y abandonar la selección de la opción presione Salir.

Enviar Partidos

Seleccione del menú principal la opción Juego y de este Enviar Partidos o presione ***Ctrl-E,*** o de la barra de Herramientas el 6to botón del lado izquierdo (su leyenda es "Enviar Partidos").

Mover Ficha de Jugada

Si desea realizar un movimiento de alguna jugada o partido que Ud. tenga, seleccione cualquiera de las siguientes alternativas

- Seleccione del menú principal la opción Partidos y de este Obtener.

- Presione las teclas ***Ctrl+O***.
- Selecione de la barra de Herramientas el botón cuyo dibujo es un tablero (4to a la izquierda)

Seleccione el partido del que desee mover ficha de la lista de contrincantes y presione Aceptar.

regresar al menú principal, presione Salir.

una jugada tiene que posicionar el mouse sobre la ficha(de su que desea mover y hacer un click. Entonces un recuadro aparecerá en escogida. Ahora debe mover el mouse a la posición donde desea la ficha y hacer nuevamente un click del mouse.

es lícita la ficha se moverá a la posición deseada y el recueadro de escogida desaparecerá , si la jugada no es válida entonces aparecerá indicándole lo sucedido.

conlleva a comer varias fichas contrincante entonces se tiene que esto ficha por ficha, esto es, primero seleccionará la ficha a mover un click del mouse luego seleccionará la casilla del tablero (con otro mouse) donde se moverá la ficha para comer a una de las contrincante moverá a la casilla correspondiente para comer la siguiente ficha y así sucesivamente.

Salir

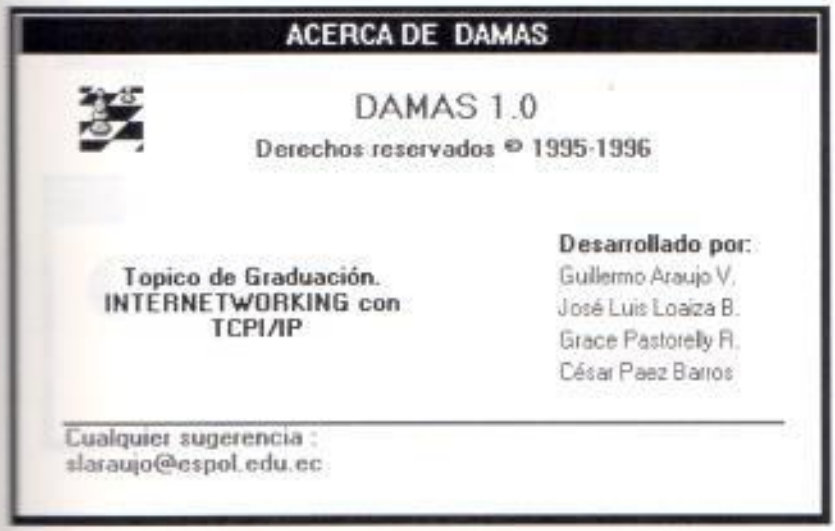
del menú principal la opción Salir o presionando ***Alt-S,*** o de la Herramientas el 1er botón del lado derecho.

Contenido de la Ayuda

la tecla ***F1*** o seleccione del menú principal la opción Ayuda y de este

Acerca de

Seleccione del menú principal la opción Ayuda y de este Acerca de.

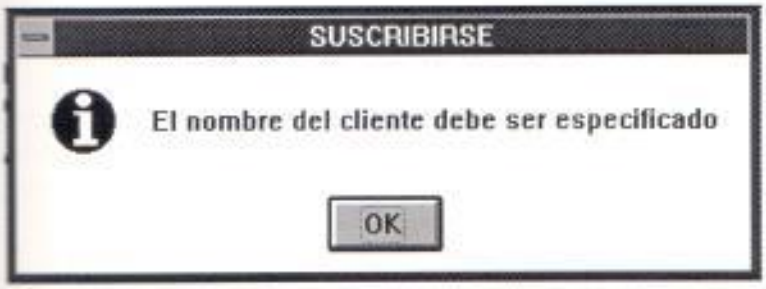


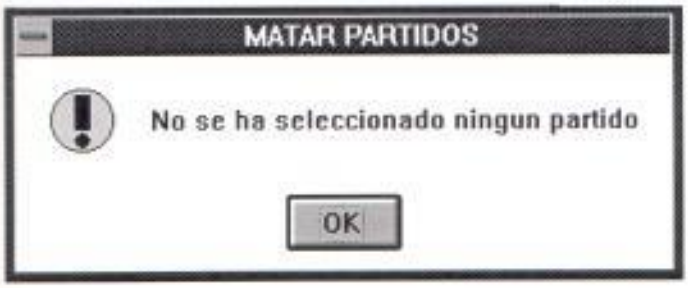
Tipos de Errores

Los errores posibles durante la ejecución de la aplicación son:

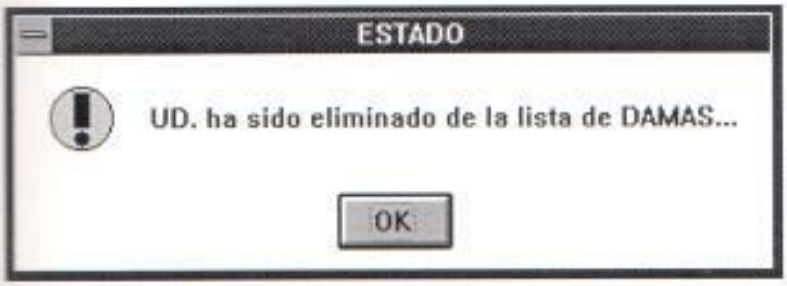
De comunicación:

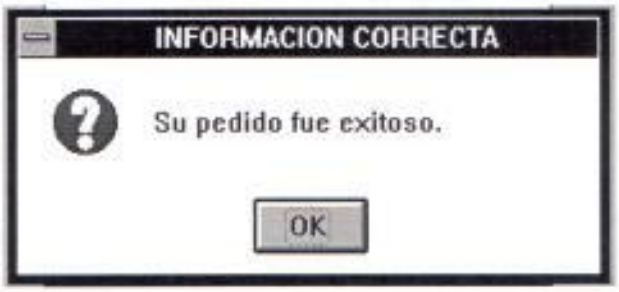
Validación de datos:





Estado o respuesta:




INFORMACION CORRECTA
Su pedido fue exitoso.
OK

MANUAL DEL ADMINISTRADOR DE LA APLICACION SERVIDORA

Instalación

Para la instalación del servidor, el administrador del centro de cómputo debe seguir los siguientes pasos distribuídos en dos etapas:

PRIMERA ETAPA (Preparación del puerto)

Realizarlo en el modo mantenimiento

1. Configurar uno de los puertos para establecer el enlace con el software cliente. Por ejemplo configurar el puerto COM2 a una velocidad de 9600 bps mediante el administrador del shell **sysadmsh--systems--hardware--serial/parallel port.**

2. Reconstruir el nuevo kernel, las variables de ambiente y que se levante por default.

3. En el modo de mantenimiento configurar el puerto anterior para que trabaje en red con el protocolo tcp a través del comando **netconfig**

4. Seleccione lo siguiente:

   Add a Chain

   sco_tcp

   interface SLIP **sl0** o **sl1.**

   tty2a o **tty1a** (para COM2 o COM1 según sea el caso)

   9600 bps

   Dirección de la interface local donde estará el servidor, ejm., 192.192.192.2

   Dirección de la interface destino, es decir de donde proviene el cliente, ejm.

   192.192.192.3

5. Reconstruir el nuevo kernel, sus variables de ambiente y que se levante por default.

SEGUNDA ETAPA (Instalación del software servidor)

1. Crear el directorio ***/home/fie/estud/jloaiza*** para almacenar todos los archivos del servidor .

2. Si desea almacenar el servidor del juego de damas en otro directorio como por ejemplo en el directorio /damas, entonces debe crear ese directorio además editar el archivo *damas.c* en el que se encuentran las variables globales *players, inivita, damas,*las cuales contienen los path de acceso, las mismas que deben ser cambiadas por el nuevo path de acceso. Ejemplo:

Antes:

players = */home/fie/estud/jloaiza/players.txt*
invita = */home/fie/estud/jloaiza/invita.txt*
damas = */home/fie/estud/jloaiza/damas.txt*

Ahora:

players = */damas/players.txt*
invita = */damas/invita.txt*
damas = */damas/damas.txt*

Finalmente, compilar el archivo damas.c para que el nuevo cambio tome efecto, para esto en el shell del unix escriba el siguiente comando.

cc damas.c .o damas.exe -lsocket

3. Grabar los siguientes archivos en el directorio que creó en el paso anterior
   damas.exe
   players.txt
   invita.txt
   damas.txt

4. Levantar el software damas.exe indicando la dirección IP y el puerto que usará para comunicarse con el software cliente (Lo puede hacer en modo background si lo desea). Por ejemplo con el siguiente comando lo haremos con la dirección 192.192.192.2 y el puerto 1995 (Recuerde que el software cliente enviará sus datos a la dirección y al puerto que aqui se indican).

./damas.exe 192.192.192.2 1995

Administración del software servidor de damas

Realmente la tarea que tendrá el administrador del sistema para instalar y mantener el servidor de damas será mínima, ya que todas las transacciones se las realiza de manera automática con el cliente. Por ejemplo, cuando el cliente hace un requerimiento de suscripción, el servidor obtiene el string de datos enviado por el cliente, realiza la suscripción y le devuelve un mensaje al cliente para indicarle que su pedido fue ejecutado.

Entre los puntos que tendrá el administrador que realizar se encuentran los siguientes.

1. Verificar la correcta configuración del puerto de acuerdo a los pasos indicados anteriormente.

2. Levantar el servidor indicando la dirección IP y el puerto que va a ser utilizado por el mismo.

3. Informar a los usuarios clientes del puerto que deberán utilizar para conectarse con el servidor de damas, y sobre todo , tener presente que ese puerto debe ser utilizado única y exclusivamente para brindar ese servicio, evitando asi conflictos con otros servicios.

4. Tomar en cuenta que el servidor utiliza un número de archivos temporales para ejecutar ciertas transacciones con los clientes , los mismos que son borrados automáticamente por el servidor una vez que haya terminado su transacción con aquel cliente. Estos archivos temporales toman diferntes nombres de acuerdo al cliente con el que estén trabajando.

5. Si por alguna razón, los archivos temporales no se están borrando, entonces el administrador deberá hacer limpieza con determinada frecuencia, para esto, primero debe bajar el servidor , luego verificar que solamente deben existir los siguientes archivos:

   damas.exe

   players.txt

   invita.txt

   damas.txt

   El resto tendrá que ser borrado.

6. Todos los archivos .txt están conformados por lineas, las mismas que contienen un determinado número de campos, los cuales a su vez, están separados por el delimitador de punto y coma (;).

Las líneas del archivo players.txt contienen 5 campos, estos son:

alias;nombre;password;partidos ganados;partidos perdidos

En este se encuentran todos los jugadores que se han suscrito al clud de juego de damas.

Las líneas del archivo invita.txt contienen los siguientes campos:

alias del que invita ;alias del invitado;respuesta

el campo *respuesta* será V o F dependiendo de si ha sido aceptada o no la

invitación.

Las líneas del archivo damas.txt contienen el alias del jugador al que le tocaría jugar , luego el alias del jugador que estaría en espera, el color de quien le toca el turno y finalmente las fichas y sus posiciones de acuerdo a un cierto formato.

Solamente en el caso de que uno de los jugadores decide matar un partido, aparecerá una línea conteniendo los dos alias y un código (9905), el mismo que sirve para informarle al otro jugador que su partido ha sido muerto por su contrincante. Una vez informado sobre este particular, la línea desaparece del archivo.

CAMBIO DE PASSWORD.- Si por algún motivo uno de los jugadores desea cambiar el password, entonces el administrador deberá ir al archivo players.txt y cambiar el tercer campo por el nuevo password en la línea que corresponde al usuario que lo solicita.