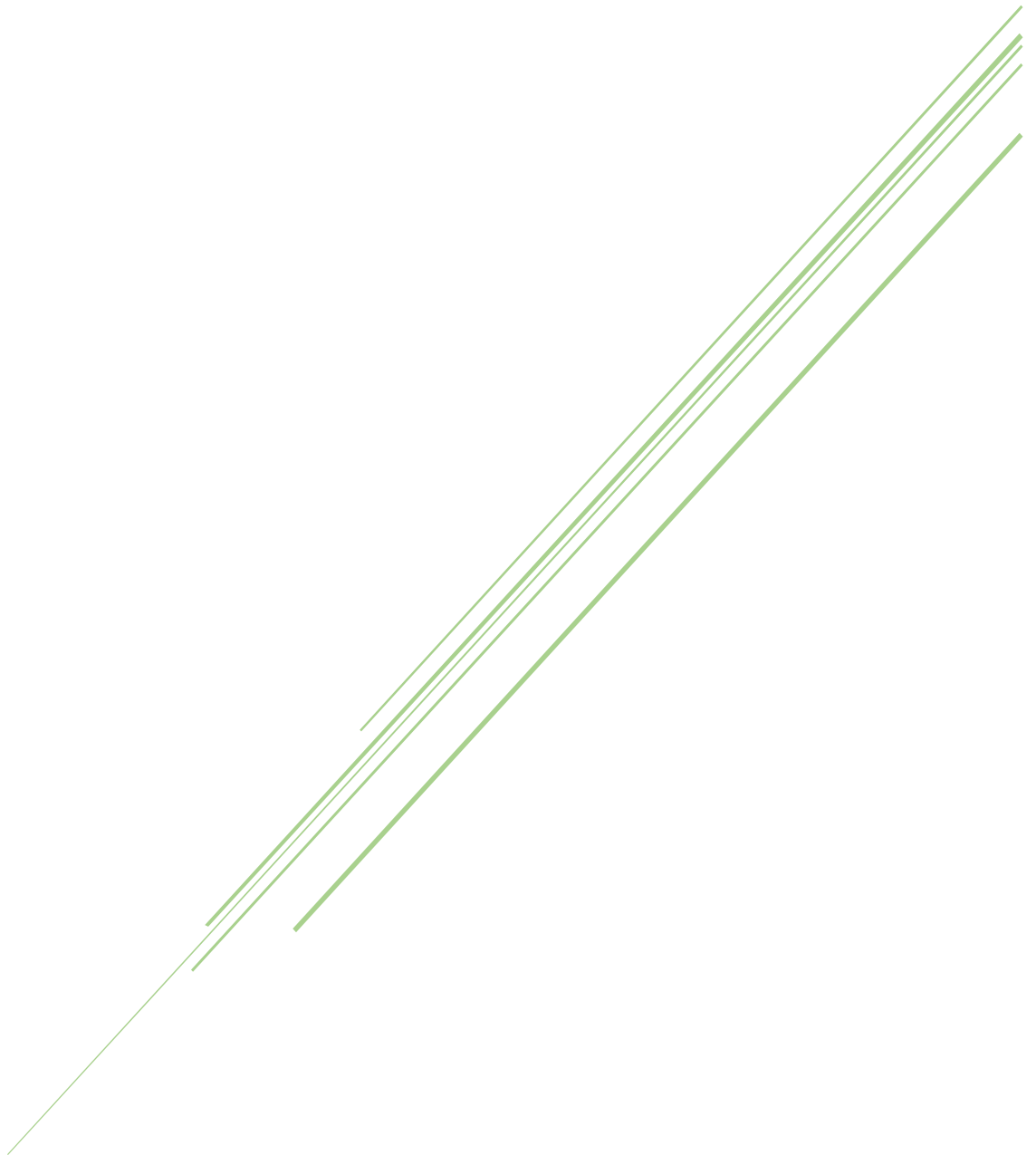


PROYECTO 2

Generala



Departamento de Cs. e Ingeniería de la Computación
Antonini Antonella – Kestel Guido

Proyecto 2

Generala

Ubicación

Proyecto2 > Ejercicio1

Códigos

Makefile
generala.h
generala.x
generala_client.c
generala_server.c

Compilación

make

Ejecución

`./generala_server`

`./generala_client <IP>`

Modo de uso

Se debe ejecutar en primer instancia el **servidor**, y luego los distintos **clientes**. Cada cliente recibe como parámetro la dirección ip del servidor, ya que debe comunicarse con él.

Una vez ejecutado el cliente, se le da la alternativa al participante de comenzar el juego mediante la *opción 1* (podrá solo si hay 2 o más jugadores) o esperar a que otro lo inicie mediante la *opción 0*.



```

Bienvenido a la
Generala

Usted es el jugador número 1.

Elija una opcion:
0-Esperar a que otro inicie el juego
1-Iniciar el juego

Opcion: █
```

Recuerde que sólo se podrán anotar jugadores hasta el instante en que un jugador indique que quiere empezar el juego.

Comunicación

La lógica del juego se simuló utilizando un único servidor centralizado y dos o más clientes. Los mismos se comunican y sincronizan sus turnos mediante el modelo de comunicación RPC.

Servidor

El servidor es el encargado de mantener toda la información de cada uno de los participantes como su estado, puntajes obtenidos, turnos, etcétera. Además, es el responsable de la sincronización de los turnos de jugada.

El servidor mantiene un identificador por participante el cual es otorgado al momento de que este inicia el juego (*función anotarse()*). Luego espera por la llegada de otros participantes, indicándole al mismo que no puede iniciar el juego hasta que haya al menos dos jugadores.

Cuando un jugador se anota para jugar, se le pregunta si quiere aguardar por más jugadores o si quiere comenzar la partida de inmediato. Una vez que se da inicio al juego, no pueden anotarse más jugadores y se establece la cantidad total de jugadores de la partida.

Una vez comenzado el juego, el servidor es el encargado de sincronizar los turnos para lanzar los dados, por parte de cada participante. El servidor llama a cada uno de los participantes cuando es tu turno de lanzar los dados. Si el participante correspondiente no responde (se realizan 1000 llamadas), el jugador se da como caído y se pasa automáticamente el turno a uno de sus contrincantes.

Al momento de que el jugador obtiene el turno, el servidor ofrece ciertas funciones al mismo para llevar a cabo su jugada:

- *jugar()*: devuelve al jugador 5 valores aleatorios entre 1 y 6 correspondientes a los valores de los dados, simulando la jugada del participante en cuestión.
- *tirar()*: devuelve un valor aleatorio entre 1 y 6 representando un dado. Se invoca por cada dado que el jugador desee tirar nuevamente.
- *anotar()*: anota en la tabla del jugador actual la jugada que éste quiera anotar, otorgándole los puntos correspondientes.
- *tachar()*: tacha en la tabla del jugador actual la jugada que éste quiera tachar. A la categoría correspondiente le asigna el valor cero.
- *termine()*: tras jugar sus 11 turnos, el jugador indica al servidor que ha finalizado, y que se encuentra a la espera de la finalización de sus contrincantes.
- *terminar()*: una vez que todos los participantes han finalizado el juego, el servidor informa los resultados obtenidos (puntajes correspondientes y ganador del juego).

```
antonella@Anto ~/Desktop/Proyecto 2 - Distribuidos/Ejer1
File Edit View Search Terminal Help


¡Ahora sí! Que comience el juego ...
Usted arrancara el juego!



| categoria      | resultado |
|----------------|-----------|
| 1              | 0         |
| 2              | 0         |
| 3              | 0         |
| 4              | 0         |
| 5              | 0         |
| 6              | 0         |
| Escalera       | 0         |
| Full           | 0         |
| Poker          | 0         |
| Generala       | 0         |
| Generala Doble | 0         |



Es su turno. Se tiraran los dados...
Han salido los dados:



Elija una opcion:
0-Volver a tirar
1-Anotar
2-Tachar
3-Resultados Parciales
4-Ver Dados
Opcion: 1
Elija una categoria para jugar...
```

Orden de jugada

El orden establecido de los turnos, es el orden en que los jugadores se anotaron en el juego. Cada jugador pide permiso al servidor para jugar hasta que el servidor se lo otorgue (provisionalmente, si no es su turno, los demás jugarán hasta que sea su turno).

El servidor eventualmente le otorgará el permiso, momento en el cuál registra quién es el jugador que está en turno, y le cede el permiso exclusivo a él.

Cuando un jugador **obtiene** el permiso, antes de comenzar el lanzamiento, pide al servidor la última jugada (realizada por el jugador anterior) y el valor actual de las tablas de puntajes, ya que puede servirle de ayuda en la decisión de la elección realizada por el participante. Esto lo realiza mediante las funciones *jugadaAnterior()* y *resultadosParciales()*.

```
antonella@Anto ~/Desktop/Proyecto 2 - Distribuidos/Ejer1
File Edit View Search Terminal Help

Resultados Parciales:
Jugador 0:
Unos      0
Dos       0
Tres      X
Cuatros   0
Cincos    0
Seis      0
Escalera  0
Full      0
Poker     0
Generala  0
Generala Doble 0

Usted:
Unos      0
Dos       0
Tres      0
Cuatros   0
Cincos    0
Seis      0
Escalera  0
Full      0
Poker     0
Generala  0
Generala Doble 0
```

```
antonella@Anto ~/Desktop/Proyecto 2 - Distribuidos/Ejer1
File Edit View Search Terminal Help

Ultima jugada del oponente:
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]

Anoto los Cuatro

categoria | resultado
-----|-----
1         | 0
2         | 0
3         | 0
4         | 0
5         | 0
6         | 12
Escalera  | X
Full      | 0
Poker     | 0
Generala  | 0
Generala Doble | 0

Es su turno. Se tiraran los dados...
Han salido los dados:
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]
[1][2][3][4][5]

Elija una opcion:
0-Volver a tirar
1-Anotar
2-Tachar
3-Resultados Parciales
4-Ver Dados
Opcion: 1
```

Una vez **finalizada** la jugada, el participante le indica al servidor, mediante la invocación de la función *permiso()*, que ha finalizado su turno y que puede continuar. Sólo aquí el servidor actualiza el siguiente jugador que puede jugar y cuando éste lo invoque podrá darle el permiso.

Ciente

El cliente en este modelo de comunicación, es el representante del jugador. Permite la interacción con el mismo, le ofrece la interfaz correspondiente para llevar a cabo la jugada, y solicita al servidor las funciones anteriormente mencionadas.

Lleva el control de los valores anotados y tachados por el participante para no anotar dos veces la misma categoría. Además, cuando el cliente solicita anotar la jugada, corrobora que se haya logrado una categoría válida (Escalera, Full, Poker, etcétera).

```
antonella@Anto ~/Desktop/Proyecto 2 - Distribuidos/Ejer1
File Edit View Search Terminal Help

Elija una opcion:
0-Volver a tirar
1-Anotar
2-Tachar
3-Resultados Parciales
4-Ver Dados
Opcion: 1
Elija que categoría desea anotar:
1 - Unos
2 - Dos
3 - Tres
4 - Cuatros
5 - Cincos
6 - Seis
7 - Escalera
8 - Full
9 - Poker
10 - Generala
11 - Generala Doble

Ingrese categoría a anotar: 7
Usted va a anotar la categoría 7
No tiene escalera. Se ha tachado la escalera
Es el turno de otro jugador
```

Una vez que el cliente se anota en el juego, el servidor lo registra, asignándole un identificado, y espera a que inicie el juego (o puede comenzarlo, si así lo desea).

A continuación, le solicita al servidor la cantidad total de jugadores, valor utilizado para mostrar ciertos datos como resultados parciales y totales del juego. Posteriormente, se implementa un ciclo, en el que constantemente el cliente le pide permiso al servidor para jugar. Este ciclo dura hasta obtener 11 permisos, que es lo que indica que se han jugado 11 turnos ¹.

Durante cada turno la interacción es la que se mencionó en el caso del servidor, eligiéndose las operaciones a ejecutar en función de las decisiones del usuario.

¹ Existen 11 posibles categorías, por lo tanto cada jugador tendrá 11 tiros posibles en el juego.