 Proyecto CUPi2	ISIS-1205 Algorítmica y Programación Descripción
Ejercicio:	n12_adivinaQuien
Autor:	Equipo CUPi2
Semestre:	

Enunciado

Se quiere construir una aplicación que permita jugar Adivina Quién desde diferentes computadores. Este es un juego de dos jugadores, cuyo objetivo es adivinar la persona asignada del oponente dentro de un tablero de personas a través de preguntas entre los jugadores.

De cada persona del tablero de juego se conoce:

- **Nombre.** Es único, no hay dos personas con el mismo nombre.
- **Estado.** Indica si la persona ha sido descartada del tablero por el jugador o no.
- **Imagen.**

Al iniciar el juego, cada jugador se identifica con su login y contraseña o se registra como nuevo usuario en el servidor. Una vez identificado, el usuario se conecta al servidor para comenzar el juego y el servidor le asigna un oponente para la partida. Una vez comienza la partida, el servidor le asigna a cada jugador una persona del tablero de juego.

El jugador con el primer turno es escogido aleatoriamente.

Después de asignar el turno comienza el juego. En cada turno el jugador puede realizar una pregunta, que el oponente debe responder. En su turno, el jugador puede intentar adivinar el nombre de la persona asignada a su oponente.

Cada jugador empieza con un tablero de 24 personas y debe adivinar el nombre de la persona asignada a su oponente. La idea del juego es que cada jugador debe formular preguntas acerca de las características físicas que cree que la persona asignada a su oponente puede tener; el oponente debe responder cada pregunta únicamente con 'SI' o 'NO'; una vez el oponente da su respuesta, el jugador puede descartar a las personas que considere necesario dependiendo de la respuesta que dio su oponente.

Como un servicio adicional, el programa debe almacenar información sobre la cantidad de partidas perdidas y ganadas por cada uno de los jugadores.

El programa distribuido debe constar de dos partes: un programa servidor, encargado de mantener la información estadística del juego y permitir a los jugadores encontrarse para una partida; y un programa cliente, a través del cual un usuario puede jugar Adivina Quién.

El programa cliente debe ofrecer las siguientes funcionalidades:

1. Cargar la información de las personas disponibles de un archivo de texto.
2. Conectarse al servidor. Para esto el usuario tiene dos alternativas:
 - a. **Registrarse:** La primera vez que un usuario ingresa debe registrarse para poder jugar. Para esto debe suministrar todos sus datos: nombre completo, login (único), contraseña, dirección IP del servidor al cual se quiere conectar y el puerto por el cual dicho servidor se encuentra esperando conexiones.
 - b. **Iniciar sesión:** Si el usuario ya se encuentra registrado, debe iniciar sesión. Para esto debe suministrar su login, su contraseña, la dirección IP del servidor y el puerto.
3. Visualizar el nombre y la imagen de la persona que le asignó el servidor.
4. Visualizar información de la partida. El jugador podrá visualizar la siguiente información:
 - a. **Información del oponente:** Login, cantidad de partidas ganadas y perdidas.
 - b. **Tablero de juego:** Los nombres y las respectivas fotos de las 24 personas del tablero.
 - c. **Preguntas:** Listado de preguntas realizadas con la respectiva respuesta dada por el oponente.
 - d. **Estado de la partida:** Este puede ser: en espera del oponente, en espera de una pregunta, realizando una pregunta o partida terminada.
5. Realizar una pregunta al oponente
6. Recibir una pregunta. Una recibida, debe contestar 'SI' o 'NO'.
7. Intentar adivinar el nombre de la persona asignada al oponente.

El programa servidor, por su lado, debe esperar a que los jugadores se vayan conectando. Cada vez que se conecten dos jugadores, inicia una partida. Esto implica que el servidor debe:

1. Verificar la información de cada jugador cuando se conecte.
 - a. Si el jugador inicia sesión, verifica su login y contraseña. El login debe estar registrado y la contraseña dada debe corresponder a la que se encuentra registrada con ese, si esto no ocurre, se le notifica al usuario con un mensaje de error y no se conecta.
 - b. Si el jugador se registra, verifica el login dado. No debe existir un jugador con el mismo login, si esto ocurre, se le notifica al usuario con un mensaje de error y no se registra ni se conecta.
2. Enviar a cada jugador la información de su oponente, una vez se establece una partida.

Una vez establecida la partida, el servidor es el responsable de la administración del juego. Esto implica que el servidor debe:

1. Determinar aleatoriamente los turnos e informarle a los jugadores su turno y su persona asignada.
2. Notificar la pregunta de un jugador a su oponente y la respuesta respectiva del segundo al primero.

Una vez finalizada la partida (cuando un jugador adivine el nombre de la persona asignada de su oponente), el servidor debe cerrarla. Esto implica que el servidor debe:

1. Registrar un juego ganado al jugador victorioso.
2. Registrar un juego perdido al jugador derrotado.
3. Notificar el fin del juego a los dos jugadores.
4. Notificar a los dos jugadores el login del jugador victorioso.

Finalmente, el servidor debe ofrecer las siguientes opciones al usuario:

1. Mostrar las partidas que se encuentren en curso. Por cada partida se debe mostrar cuantas personas tiene cada jugador en el tablero.
2. Mostrar las estadísticas históricas del juego. Se debe mostrar el login de cada jugador que haya participado en una partida, la cantidad de partidas ganadas y perdidas. La información histórica de las batallas debe ser persistente y se debe almacenar en una base de datos, de manera que cada vez que se ejecute de nuevo el servidor, las estadísticas sean presentadas al usuario.

Interfaz Cliente

Diálogo de inicio. El usuario puede elegir en el menú Opciones, si desea iniciar sesión o registrarse.



Si el usuario desea registrarse, se desplegará el diálogo que se muestra en la siguiente imagen.



Por otro lado, si el usuario desea iniciar sesión, se muestra el siguiente diálogo:

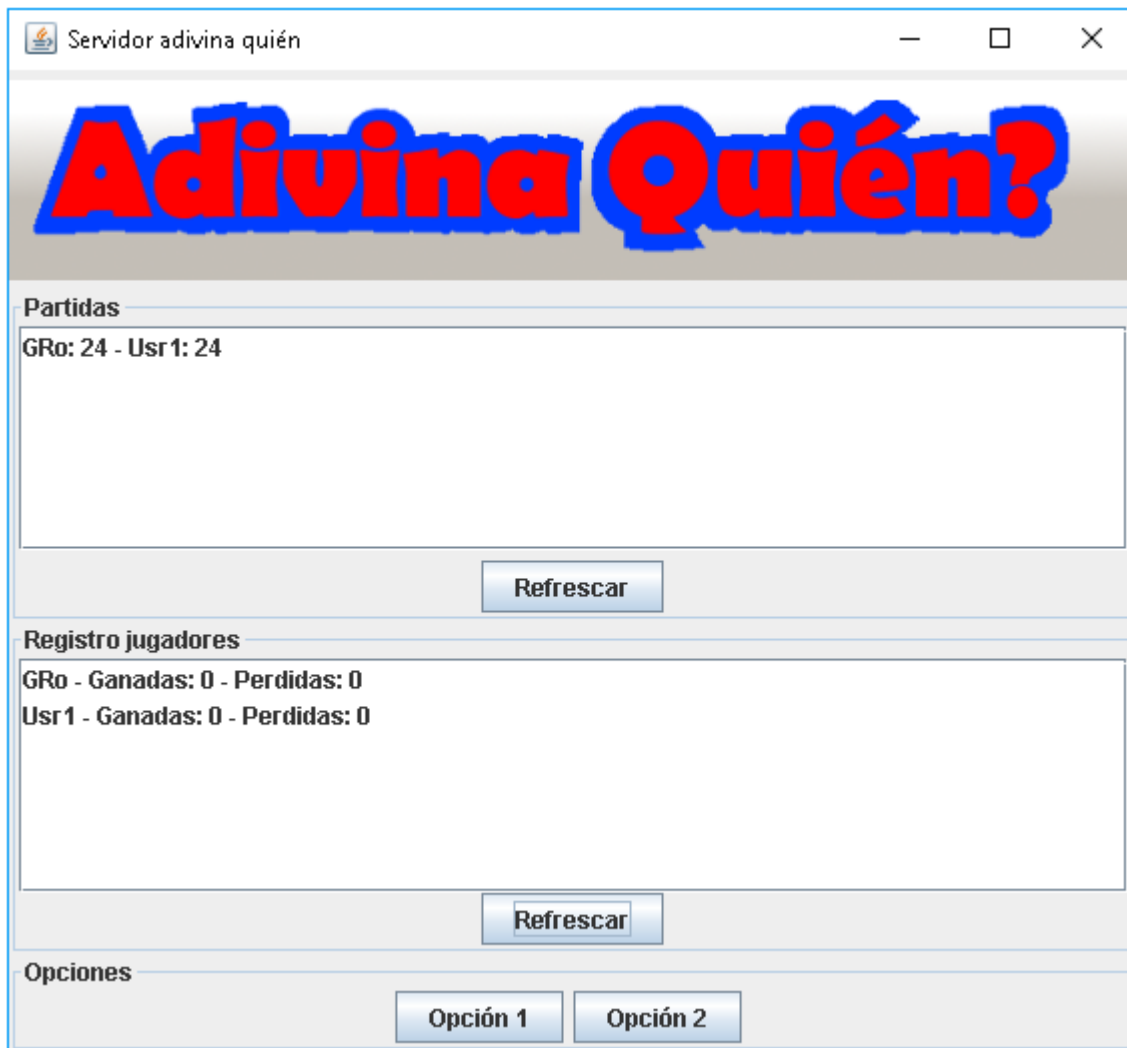


Una vez se realiza la conexión de dos jugadores, se muestra la interfaz principal al momento de realizar una partida.



Interfaz Servidor

La interfaz del servidor permite refrescar la lista de partidas actuales y la lista de jugadores que han realizado una partida.



The screenshot shows a web browser window titled "Servidor adivina quién". The main heading is "Adivina Quién?" in large, stylized red and blue letters. Below this, there are three sections:

- Partidas**: A box containing the text "GRo: 24 - Usr1: 24". Below this box is a "Refrescar" button.
- Registro jugadores**: A box containing the text "GRo - Ganadas: 0 - Perdidas: 0" and "Usr1 - Ganadas: 0 - Perdidas: 0". Below this box is a "Refrescar" button.
- Opciones**: A section with two buttons labeled "Opción 1" and "Opción 2".

Persistencia

Para el desarrollo de la aplicación Adivina Quién, se utilizará como motor de base de datos el sistema (*open source*) llamado Derby, desarrollado en el proyecto Apache Derby.

Las ventajas de utilizar un motor de base de datos Derby son:

- Está desarrollado completamente en Java y tiene un sistema de persistencia basado en archivos.
- Permite crear un servidor Derby independiente de la aplicación y establecer la comunicación con esta a través de sockets y desde diferentes máquinas.
- Es una base de datos embebida, es decir, que se puede utilizar como si fuera parte de la aplicación.

Para la aplicación se usará la base de datos embebida que consta de una tabla para almacenar la información de los jugadores desde la primera vez que el servidor es ejecutado y otra tabla para almacenar la información de las partidas ganadas y perdidas. La primera tabla incluye los siguientes campos: alias del jugador, nombre

del jugador, apellidos del jugador, contraseña del jugador, victorias y derrotas. La llave primaria de la tabla es el alias del jugador. Esto implica que no pueden existir dos jugadores con el mismo alias).

En la tabla a continuación se muestra el diseño de las tablas de la base de datos con sus campos y llave primaria.

Tabla	Campos	Tipo	Llave primaria
usuarios	login	varchar(32)	login
	nombre	varchar(32)	
	contrasenia	varchar(12)	
resultados	login	varchar(32)	
	cantidadGanadas	int	
	cantidadPerdidas	int	

Tabla 1. Diseño de la tabla para la base de datos.

Protocolo de comunicación

A continuación, se presenta el protocolo de comunicación que establece cuáles mensajes (y en qué orden) se deberán enviar para realizar cada una de las tareas del sistema. Es responsabilidad tanto del servidor como de los clientes ser capaz de interpretar este protocolo y realizar las tareas necesarias.

En términos generales, los mensajes que se envían tienen el siguiente formato:

<COMANDO>;;<PARÁMETRO1>::<PARÁMETRO2>:: ... ::<PARÁMETROn>

En dónde:

- <COMANDO>: Indica la solicitud del usuario al servidor o la respuesta del servidor al usuario.
- <PARAMETRO>: Indica la información necesaria para resolver la petición.
- Para diferenciar el comando de los parámetros, se utiliza el separador “;;;” (3 “punto y coma”).
- Para diferenciar los parámetros, se utiliza el separador “::” (3 “dos puntos”).

Protocolo para iniciar sesión:

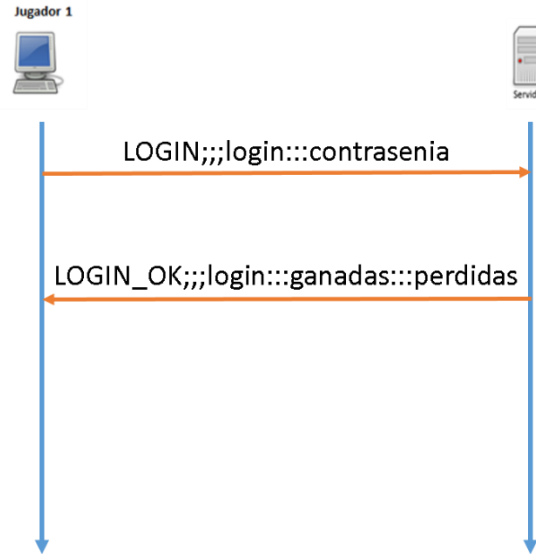


Figura 1. Protocolo de inicio de sesión.

Caso 1: El jugador inicia sesión correctamente en el servidor.

En la Figura 1 se aprecia el protocolo de inicio de sesión que consiste en que el programa cliente envía un mensaje al servidor con el comando LOGIN, seguido del login y la contraseña del jugador que desea iniciar sesión. El servidor verifica la existencia del login y correspondencia de la contraseña en la base de datos y responde con un mensaje de LOGIN_OK con el login del jugador, la cantidad de partidas ganadas y la cantidad de partidas perdidas.

La sintaxis de cada mensaje es explicada en la **Tabla 2** y **Tabla 3**:

Comando cliente	Parámetros	Descripción
LOGIN	login	Representa el login del jugador que se va a conectar al servidor.
	contrasenia	Representa la contraseña del jugador que se va a conectar al servidor.

Tabla 2. Sintaxis del mensaje del cliente para el protocolo de inicio de sesión.

Comando servidor	Parámetros	Descripción
LOGIN_OK	login	Representa el login del jugador que se conectó al servidor.
	ganadas	Representa la cantidad de partidas ganadas por el jugador.
	perdidas	Representa la cantidad de partidas perdidas por el jugador.

Tabla 3. Sintaxis del mensaje del servidor para el protocolo de inicio de sesión.

A continuación, un ejemplo concreto:



Figura 2. Ejemplo del protocolo de inicio de sesión

En la Figura 2 se ilustra el escenario en el que el jugador con el alias “GRo” desea iniciar sesión, por lo tanto, el programa envía el mensaje con comando LOGIN, el alias “GRo” y la contraseña “qwe123”. El servidor encuentra en su base de datos un jugador con el login y contraseña dada. El servidor responde al usuario con el comando LOGIN_OK, indicando que los datos proporcionados son correctos, en su mensaje incluye el login del usuario que se conectó y el número 5 como la cantidad de partidas que ha ganado anteriormente el usuario “GRo”, y 1 la cantidad de partidas que ha perdido anteriormente el usuario “GRo”.

Caso 2: El jugador tiene problemas para iniciar sesión:

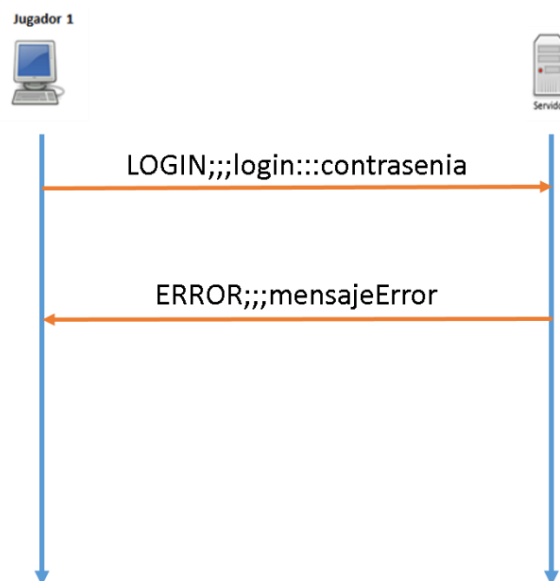


Figura 3. Protocolo de inicio de sesión cuando se presenta un error.

En la 3 se aprecia el protocolo de inicio de sesión en presencia de un error. Ocurre cuando el programa cliente envía un mensaje al servidor con el comando LOGIN, seguido del login y la contraseña del jugador que desea iniciar sesión. El servidor no encuentra en su base de datos un jugador con el login dado, o la contraseña dada no corresponde a la de la base de datos, o se presenta un error en la conexión, por lo tanto le envía al programa cliente un mensaje con el comando ERROR y el mensaje de error.

La sintaxis de cada mensaje es explicada en la **Tabla 4** y **Tabla 5**.

Comando cliente	Parámetros	Descripción
LOGIN	login	Representa el login del jugador que se va a conectar al servidor.
	contrasenia	Representa la contraseña del jugador que se va a conectar al servidor.

Tabla 4. Sintaxis del mensaje del cliente para el protocolo de inicio de sesión.

Comando servidor	Parámetros	Descripción
ERROR	mensajeError	Representa el mensaje que indica el error que se presentó.

Tabla 5. Sintaxis del mensaje del servidor para el protocolo de inicio de sesión en caso de error.

A continuación, un ejemplo concreto:

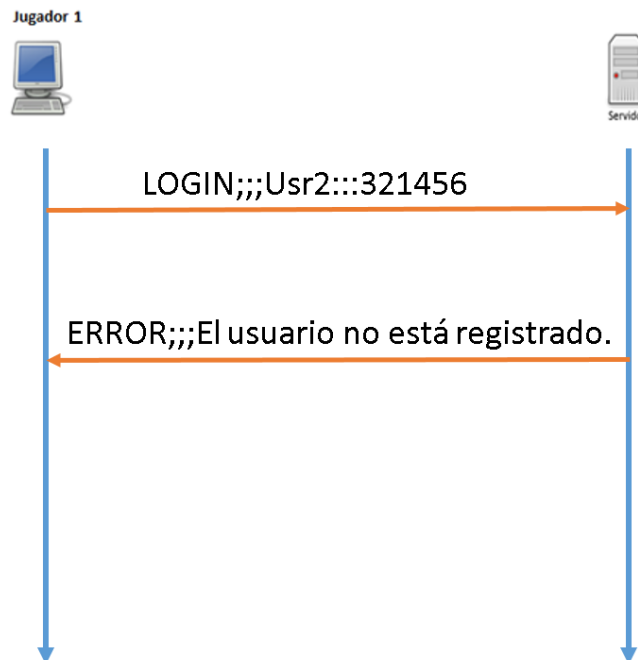


Figura 4. Ejemplo de protocolo de inicio de sesión cuando se presenta un error.

La Figura 4 muestra el siguiente ejemplo: el jugador envía una petición de inicio de sesión con el comando LOGIN y el login "Usr2", contraseña "321456". El servidor no encuentra en su base de datos un jugador con el login dado, por lo tanto envía el mensaje de error con comando: ERROR y el mensaje "El jugador no está registrado".

Protocolo para registrar un jugador:

Caso 1: Se registra un jugador que no se encuentra en la base de datos.

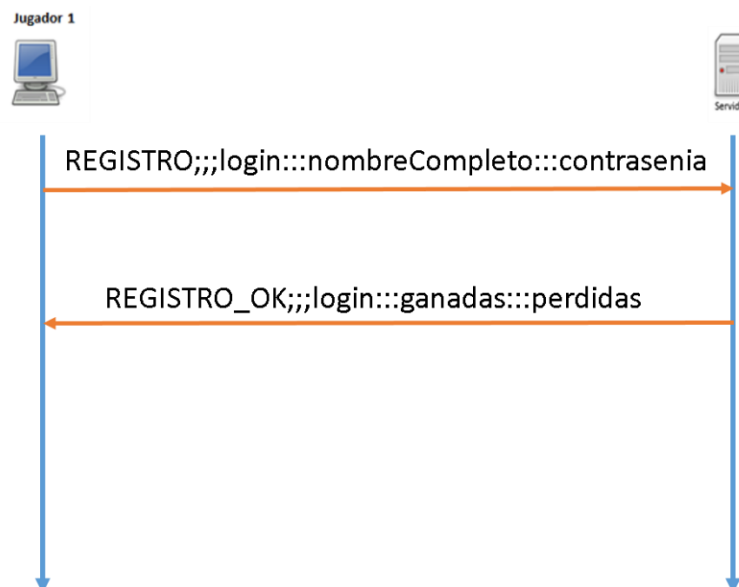


Figura 5. Protocolo para registrar un nuevo usuario.

En la Figura 5 se aprecia el protocolo de registro que consiste en que el programa cliente envía un mensaje al servidor con el comando REGISTRO, seguido de los datos del jugador que desea registrarse: login, nombre completo y la contraseña. El servidor verifica que no exista un jugador con el mismo login. Posteriormente el servidor envía un mensaje al cliente con el comando REGISTRO_OK con el login, la cantidad de partidas ganadas y perdidas del usuario.

La sintaxis de cada mensaje es explicada en la **Tabla 6** y **Tabla 7**:

Comando cliente	Parámetros	Descripción
REGISTRO	login	Representa el login del jugador que desea registrarse.
	nombreCompleto	Representa el nombre completo del jugador que desea registrarse.
	contrasenia	Representa la contraseña del jugador que desea registrarse.

Tabla 6. Sintaxis del mensaje del cliente para el protocolo de registro de un jugador.

Comando servidor	Parámetros	Descripción
INFO	login	Representa el login del jugador.
	ganadas	Representa el número de partidas ganadas del jugador.
	perdidas	Representa el número de partidas perdidas del jugador.

Tabla 7. Sintaxis del mensaje del servidor.

A continuación, un ejemplo concreto:



Figura 6. Ejemplo del protocolo para registrar un nuevo usuario.

En el ejemplo que muestra la Figura 6, el usuario Equipo cupi2 desea registrarse en el sistema, por lo tanto, el programa cliente envía el mensaje con comando REGISTRO seguido del login “cupi2”, nombre completo: Equipo cupi2 y la contraseña “psw123”. El servidor revisa en la base de datos si existe un jugador con el login “Brock, si no lo encuentra procede a agregar el usuario. El servidor responde con un mensaje con el comando REGISTRO_OK seguido del login “cupi2”, cantidad de partidas ganadas 0 y cantidad de partidas perdidas 0.

Caso 2: El jugador tiene problemas para registrarse:

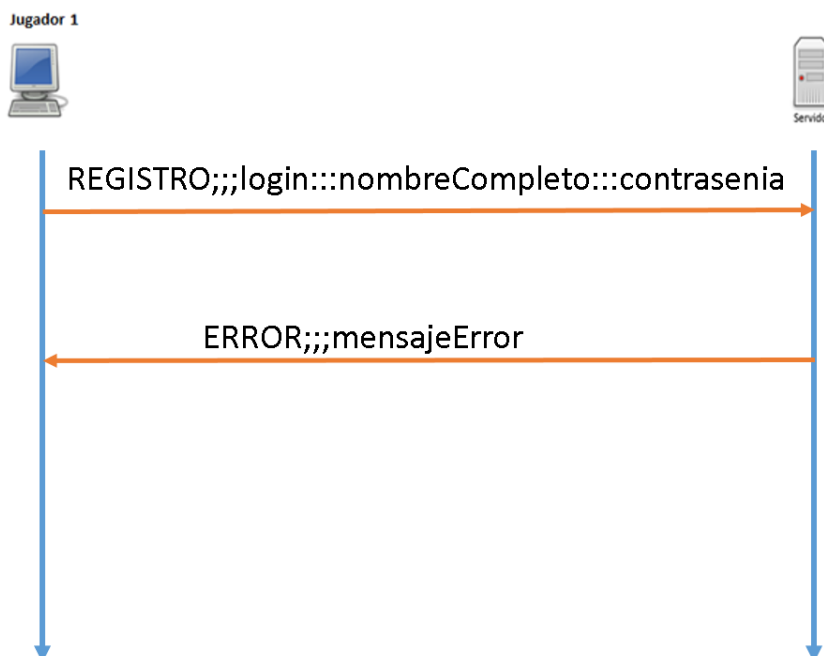


Figura 7. Protocolo de registro de un usuario cuando se presenta un error.

En la Figura 7 se muestra el protocolo de registro de usuario en caso de error, en donde el programa cliente envía la solicitud de registro al servidor con un mensaje con comando REGISTRO, seguido del login, nombre completo y la contraseña del usuario que desea registrar. El servidor revisa en su base de datos en caso de encontrar un usuario con el login dado, o de que se presente un problema en la conexión, envía un mensaje de error al programa cliente con comando ERROR seguido del mensaje de error.

La sintaxis de cada mensaje es explicada en la **Tabla 8** y **Tabla 9**:

Comando cliente	Parámetros	Descripción
REGISTRO	login	Representa el login del jugador que desea registrarse.
	nombreCompleto	Representa el nombre del jugador que desea registrarse.
	contraseniaJugador	Representa la contraseña del jugador que desea registrarse.

Tabla 8. Sintaxis del mensaje del cliente para el protocolo de registro de un jugador.

Comando servidor	Parámetros	Descripción
ERROR	mensajeError	Representa el mensaje de error que se requiere informar al usuario.

Tabla 9. Sintaxis del mensaje del servidor para el protocolo de registro de un usuario en caso de error.

A continuación, un ejemplo concreto:

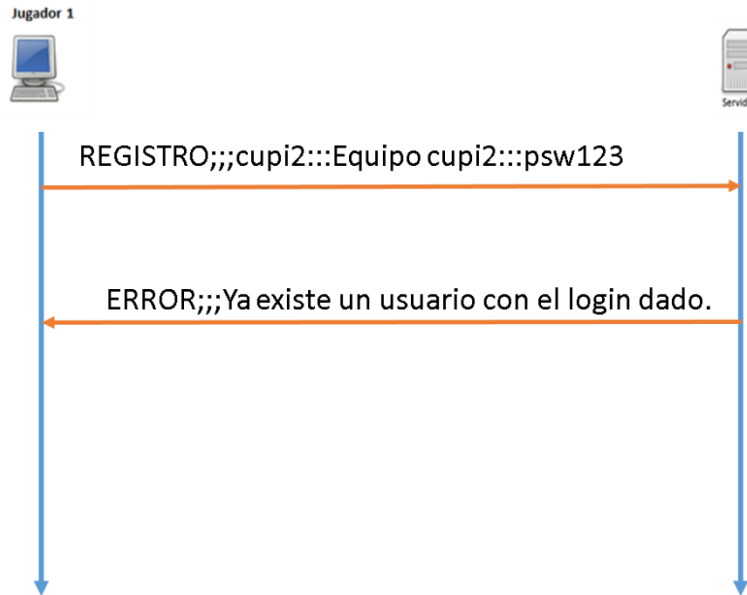


Figura 8. Ejemplo del protocolo de registro de usuario en caso de error.

La Figura 8 muestra el siguiente ejemplo: el jugador envía una petición de registro de jugador con el comando REGISTRO seguido del login “cupi2”, nombre completo “Equipo cupi2” y contraseña “psw123”. El servidor al revisar la base de datos encuentra un jugador con el login, por lo tanto envía al programa cliente el mensaje de error.

Protocolo para iniciar la partida:

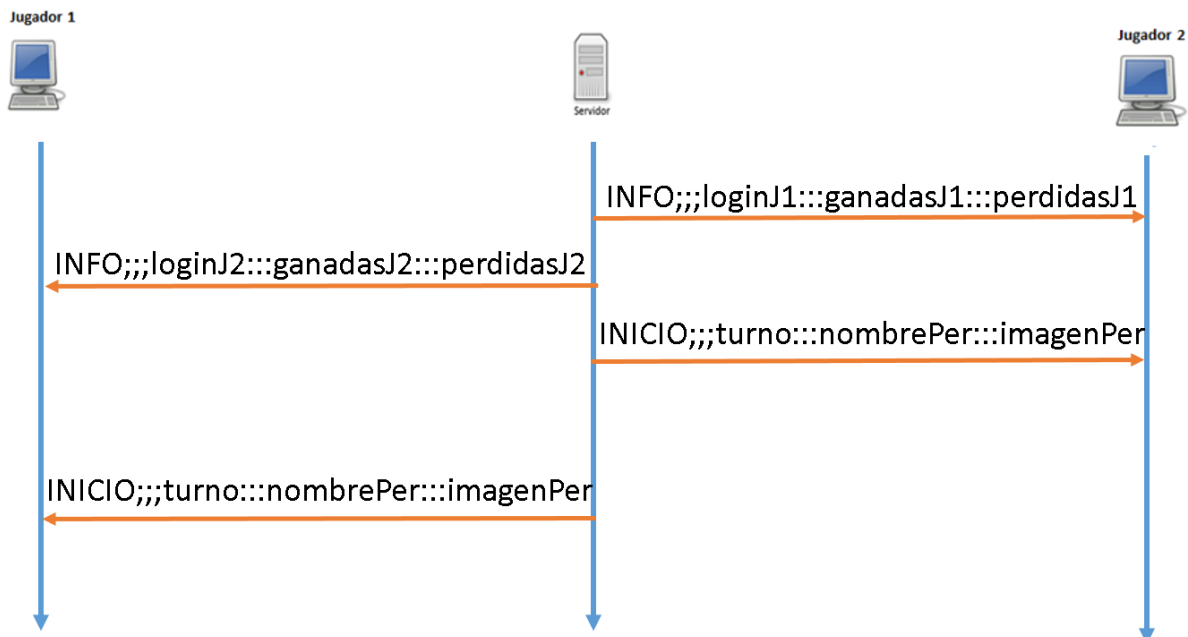


Figura 9. Protocolo para iniciar el encuentro.

En la Figura 9 se aprecia el protocolo para iniciar el encuentro entre dos jugadores. Primero el servidor notifica a cada jugador la información de su oponente a través de los mensajes con comando INFO. El servidor elige aleatoriamente el turno y la persona asignada que tendrá cada jugador, enviando un mensaje al cliente de cada jugador con el comando inicio, seguido del turno (que puede ser 1 o 2) y del nombre de la persona asignada y su imagen respectiva.

La sintaxis de cada mensaje es explicada en la **Tabla 10**:

Comando servidor	Parámetros	Descripción
INFO	loginJx	Login del jugador oponente.
	ganadasJx	Cantidad de partidas ganadas del oponente
	perdidasJx	Cantidad de partidas perdidas del oponente
INICIO	turno	Turno asignado al jugador, puede ser 1 o 2.
	nombrePer	Nombre de la persona asignada al jugador.
	imagenPer	Imagen de la persona asignada al jugador.

Tabla 10. Sintaxis del mensaje del servidor para el protocolo de inicio de partida.

A continuación, un ejemplo concreto:

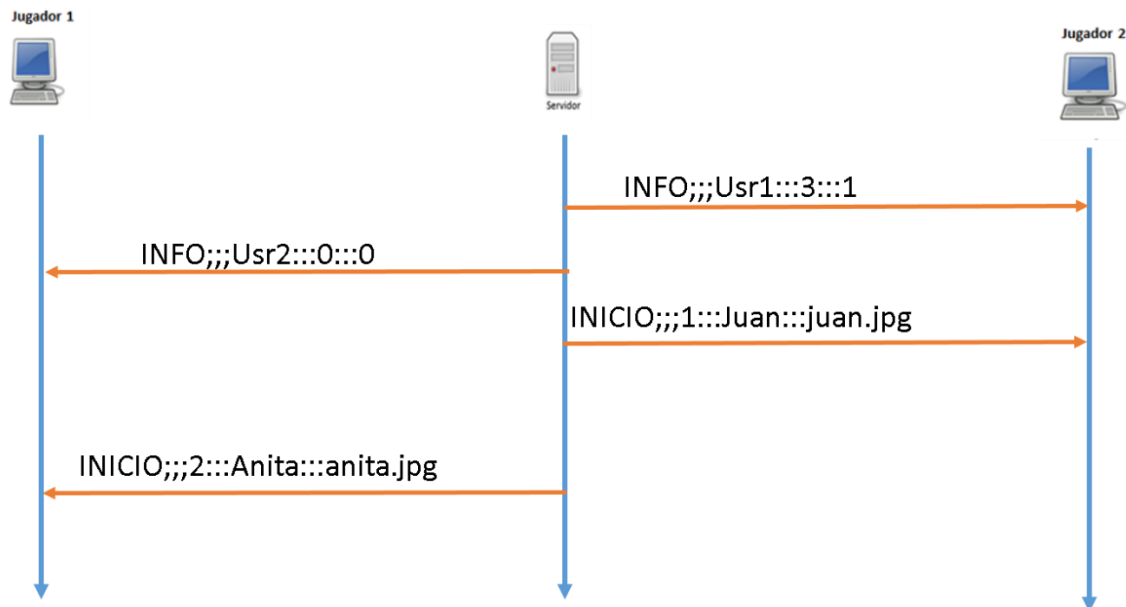


Figura 10. Ejemplo del protocolo para iniciar un encuentro.

En la Figura 10, se muestra el siguiente ejemplo: el servidor notifica al Jugador 2 la información de su oponente: login “Usr1”, cantidad de partidas ganadas 3 y cantidad de partidas perdidas 1. El servidor también notifica al Jugador 1 la información de su oponente login “Usr1”, cantidad de partidas ganadas 0 y cantidad de partidas perdidas 0. Posteriormente, el servidor escoge aleatoriamente los turnos y las personas asignadas, en este caso el Jugador 2 tiene el primer turno y su persona asignada es Juan, en cambio el Jugador 1 tiene el segundo turno y su persona asignada es Anita.

Protocolo para realizar el turno de un jugador:

Caso 1: El jugador realizar la jugada normalmente y le quedan 2 o más personas en el tablero.

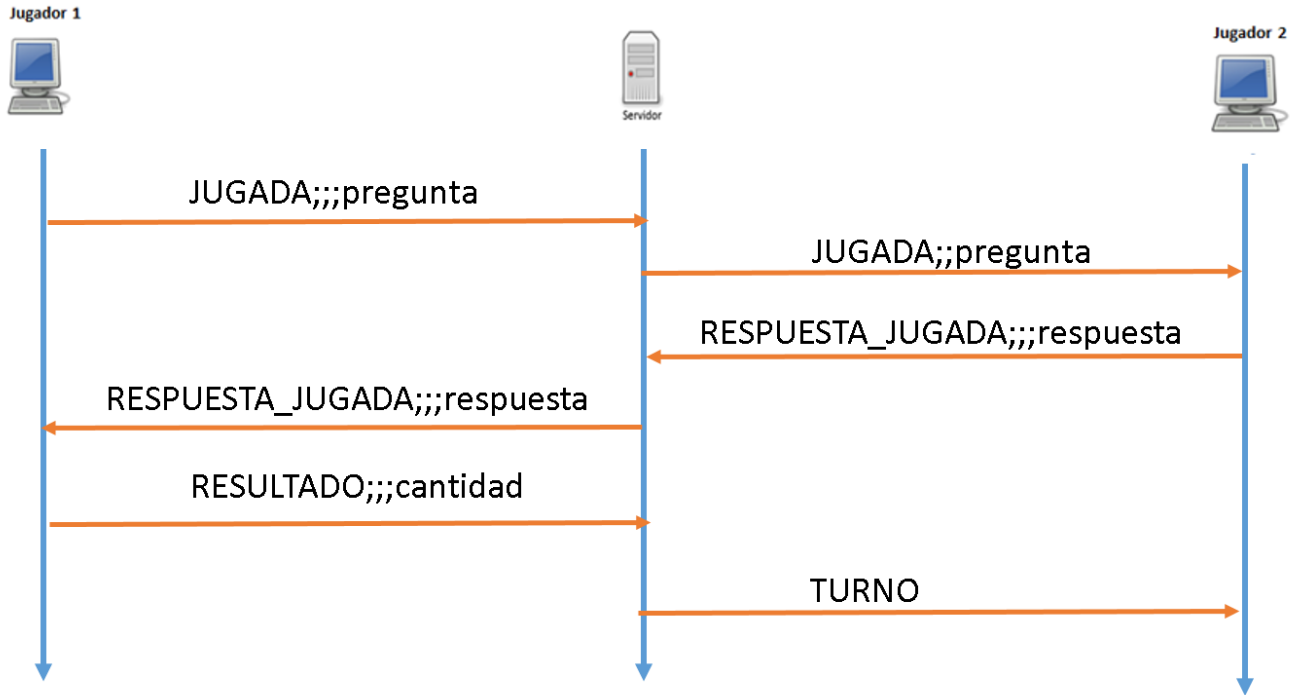


Figura 11. Protocolo para realizar una jugada.

En la Figura 11 se aprecia el protocolo para realizar una jugada. Ocurre cuando el cliente del jugador en turno, envía un mensaje con comando **JUGADA** seguido de la pregunta y el servidor recibe el mensaje y lo envía la pregunta a su oponente. El oponente, recibe la pregunta y envía una respuesta con el comando **RESPUESTA_JUGADA** seguida de la respuesta a la pregunta anterior. El servidor recibe el mensaje de respuesta y lo envía al jugador que hizo la pregunta. Cuando el jugador recibe la respuesta, envía un mensaje al servidor con el comando **RESULTADO** seguido de la cantidad de personas restantes en su tablero. El servidor notifica al oponente que ahora es su turno de jugar a través del mensaje con comando **TURNO**.

La sintaxis de cada mensaje es explicada en la **Tabla 11**:

Comando servidor	Parámetros	Descripción
JUGADA	pregunta	Representa la pregunta que realiza el jugador a su oponente.
RESPUESTA_JUGADA	respuesta	Representa la respuesta que da el oponente al jugador. Debe ser Si o NO.
RESULTADO	cantidad	Representa la cantidad de personas que quedan en el tablero del jugador.
TURNO		Indica que es el turno del oponente.

Tabla 11. Sintaxis del mensaje del cliente y servidor para el protocolo para realizar una jugada.

A continuación, un ejemplo concreto:

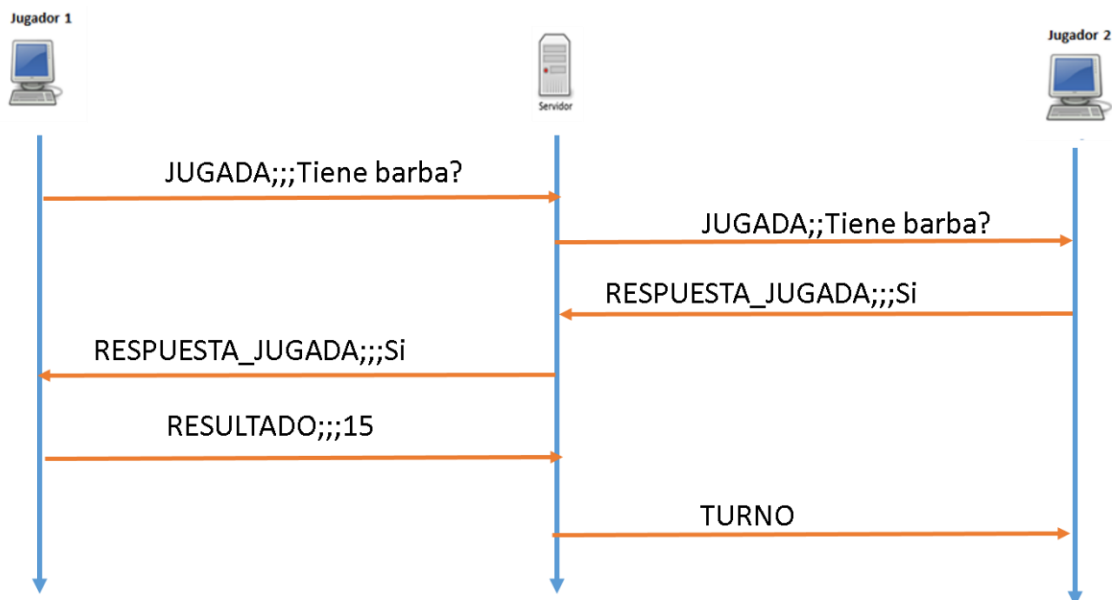


Figura 12. Ejemplo de protocolo para realizar una jugada.

En la Figura 12, se muestra el siguiente ejemplo: El Jugador 1 envía el mensaje con comando JUGADA preguntando si la persona de su oponente “Tiene barba?”, el oponente responde con el comando RESPUESTA_JUGADA que su personaje “Si” cumple con la condición preguntada anteriormente. Al recibir la respuesta del oponente, el jugador envía el comando RESULTADO indicando que le quedan 15 personas en su tablero. Por último, el servidor le indica al oponente que es su turno a través del mensaje con comando TURNO.

Caso 2: El jugador realizar la jugada normalmente y le queda 1 persona en el tablero pero no logra adivinar la persona del rival.

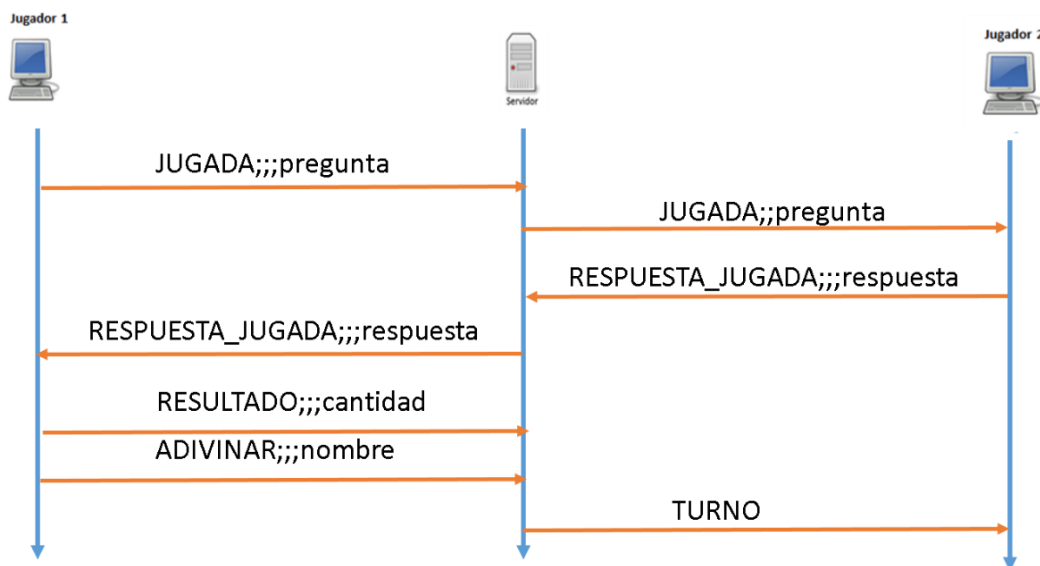


Figura 13. Protocolo para realizar una jugada.

En la Figura 13 se aprecia el protocolo para realizar una jugada. Ocurre cuando el cliente del jugador en turno, envía un mensaje con comando JUGADA seguido de la pregunta y el servidor recibe el mensaje y lo envía la pregunta a su oponente. El oponente, recibe la pregunta y envía una respuesta con el comando RESPUESTA_JUGADA seguida de la respuesta a la pregunta anterior. El servidor recibe el mensaje de respuesta y lo envía al jugador que hizo la pregunta. Cuando el jugador recibe la respuesta, envía un mensaje al servidor con el comando RESULTADO seguido de la cantidad de personas restantes en su tablero. Como sólo queda 1 persona en el tablero, el cliente del jugador intenta adivinar con el mensaje con comando ADIVINAR seguido del nombre de la persona a adivinar. Como no logra adivinar, el servidor notifica al oponente que ahora es su turno de jugar a través del mensaje con comando TURNO.

La sintaxis de cada mensaje es explicada en la **Tabla 12**:

Comando servidor	Parámetros	Descripción
JUGADA	pregunta	Representa la pregunta que realiza el jugador a su oponente.
RESPUESTA_JUGADA	respuesta	Representa la respuesta que da el oponente al jugador. Debe ser Si o NO.
RESULTADO	cantidad	Representa la cantidad de personas que quedan en el tablero del jugador.
TURNO		Indica que es el turno del oponente.
ADIVINAR	nombre	Representa el texto del intento de adivinar el nombre de la persona del oponente

Tabla 12. Sintaxis del mensaje del cliente y servidor para el protocolo para realizar una jugada.

A continuación, un ejemplo concreto:

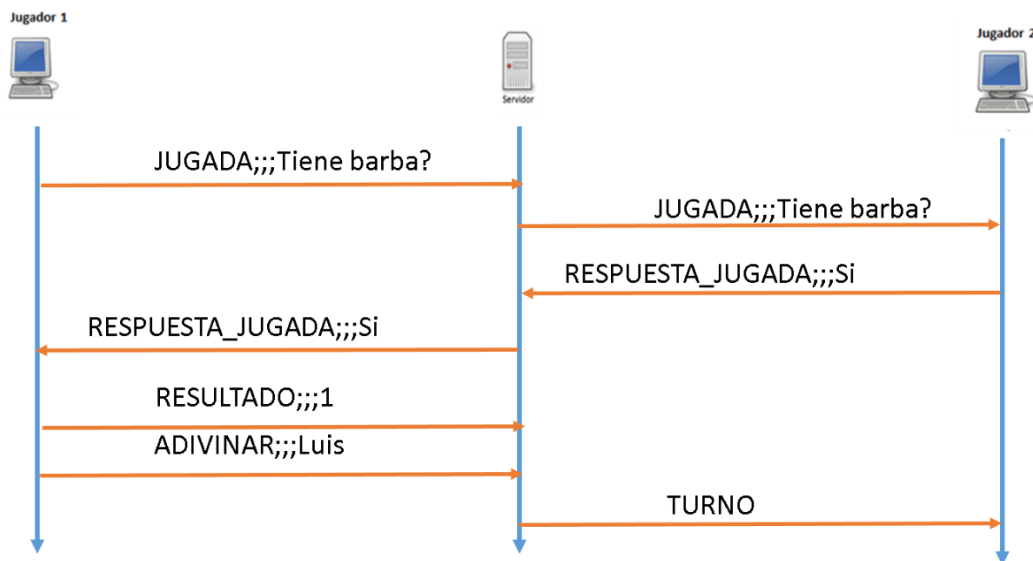


Figura 14. Ejemplo de protocolo para realizar una jugada.

En la Figura 14, se muestra el siguiente ejemplo: El Jugador 1 envía el mensaje con comando JUGADA preguntando si la persona de su oponente “Tiene barba?”, el oponente responde con el comando RESPUESTA_JUGADA que su personaje “Si” cumple con la condición preguntada anteriormente. Al recibir la respuesta del oponente, el jugador envía el comando RESULTADO indicando que le queda 1 persona en su tablero. Por lo que intenta adivinar el nombre de la persona del oponente con el comando ADIVINAR y el

nombre “Luis”. Por último, el servidor le indica al oponente que es su turno a través del mensaje con comando TURNO.

Caso 3: El jugador realizar la jugada normalmente y le queda 1 persona en el tablero y logra adivinar la persona del rival.

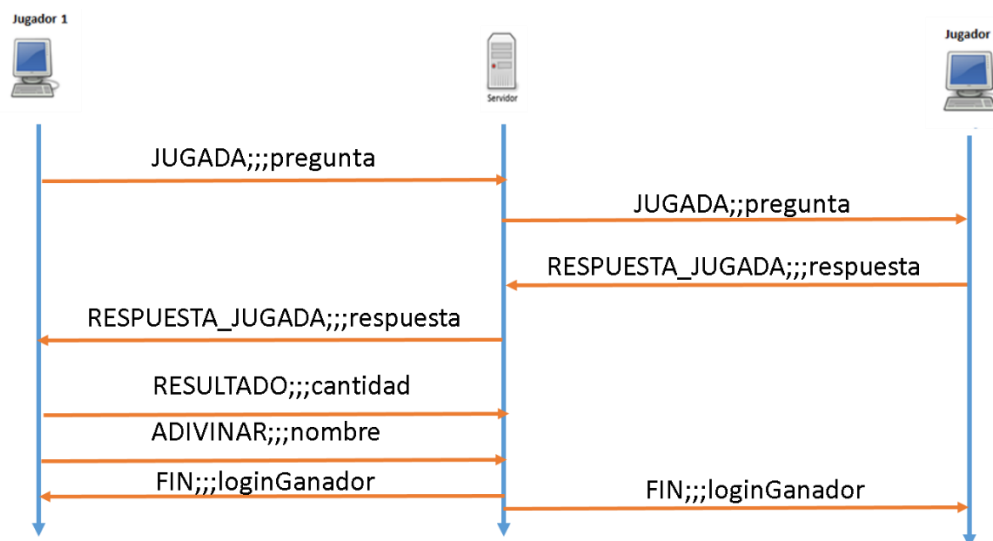


Figura 14. Protocolo para realizar una jugada.

En la Figura 14 se aprecia el protocolo para realizar una jugada. Ocurre cuando el cliente del jugador en turno, envía un mensaje con comando JUGADA seguido de la pregunta y el servidor recibe el mensaje y lo envía la pregunta a su oponente. El oponente, recibe la pregunta y envía una respuesta con el comando RESPUESTA_JUGADA seguida de la respuesta a la pregunta anterior. El servidor recibe el mensaje de respuesta y lo envía al jugador que hizo la pregunta. Cuando el jugador recibe la respuesta, envía un mensaje al servidor con el comando RESULTADO seguido de la cantidad de personas restantes en su tablero. Como sólo queda 1 persona en el tablero, el cliente del jugador intenta adivinar con el mensaje con comando ADIVINAR seguido del nombre de la persona a adivinar. Como logra adivinar, el servidor notifica ambos jugadores el fin de la partida con el mensaje FIN seguido del login del usuario que ganó.

La sintaxis de cada mensaje es explicada en la **Tabla 13**:

Comando servidor	Parámetros	Descripción
JUGADA	pregunta	Representa la pregunta que realiza el jugador a su oponente.
RESPUESTA_JUGADA	respuesta	Representa la respuesta que da el oponente al jugador. Debe ser Si o NO.
RESULTADO	cantidad	Representa la cantidad de personas que quedan en el tablero del jugador.
ADIVINAR	nombre	Representa el texto del intento de adivinar el nombre de la persona del oponente
FIN	loginGanador	Representa el login del jugador ganador.

Tabla 13. Sintaxis del mensaje del cliente y servidor para el protocolo para realizar una jugada.

A continuación, un ejemplo concreto:

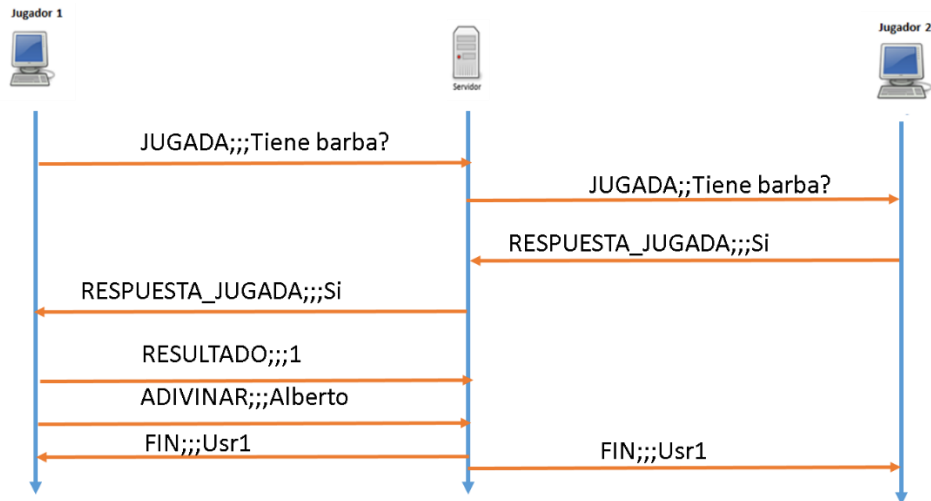


Figura 15. Ejemplo de protocolo para realizar una jugada.

En la Figura 15, se muestra el siguiente ejemplo: El Jugador 1 envía el mensaje con comando JUGADA preguntando si la persona de su oponente “Tiene barba?”, el oponente responde con el comando RESPUESTA_JUGADA que su personaje “Si” cumple con la condición preguntada anteriormente. Al recibir la respuesta del oponente, el jugador envía el comando RESULTADO indicando que le queda 1 persona en su tablero. Por lo que intenta adivinar el nombre de la persona del oponente con el comando ADIVINAR y el nombre “Alberto”. Dado que el jugador logró adivinar el nombre de la persona del oponente, el servidor notifica a los 2 jugadores el fin del juego con el comando FIN y el login del ganador “Usr1”.

Protocolo para intentar adivinar el nombre de la persona del oponente:

Caso 1: El jugador no logra adivinar el nombre de la persona del rival.

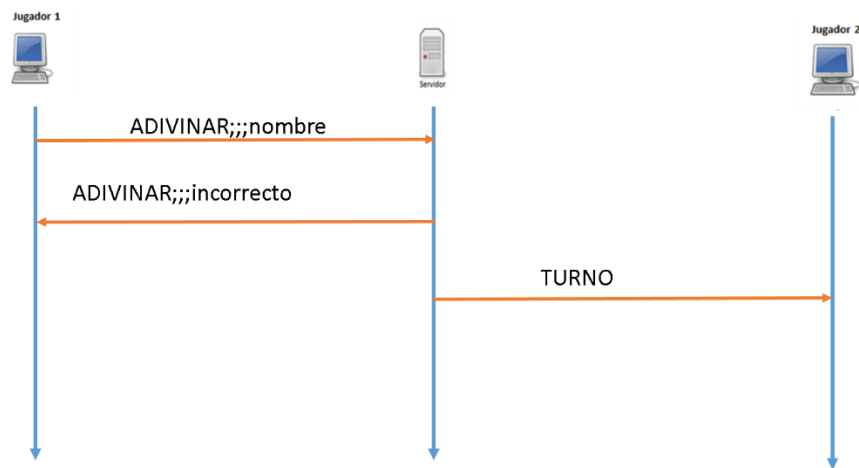


Figura 16. Protocolo para adivinar.

En la Figura 16 se aprecia el protocolo para adivinar. Ocurre cuando el cliente del jugador en turno, envía un mensaje con comando ADIVINAR seguido del nombre de la persona y el servidor le informa que no adivinó con el comando ADIVINAR seguido del mensaje incorrecto. El servidor notifica al oponente que ahora es su turno de jugar a través del mensaje con comando TURNO.

La sintaxis de cada mensaje es explicada en la **Tabla 14**:

Comando	Parámetros	Descripción
ADIVINAR	nombre	Representa el nombre que intenta adivinar de la persona asignada al oponente.
ADIVINAR	incorrecto	Representa el mensaje que indica que el jugador no adivinó.
TURNO		Indica que es el turno del oponente.

Tabla 14. Sintaxis del mensaje del cliente y servidor para el protocolo para realizar una jugada.

A continuación, un ejemplo concreto:

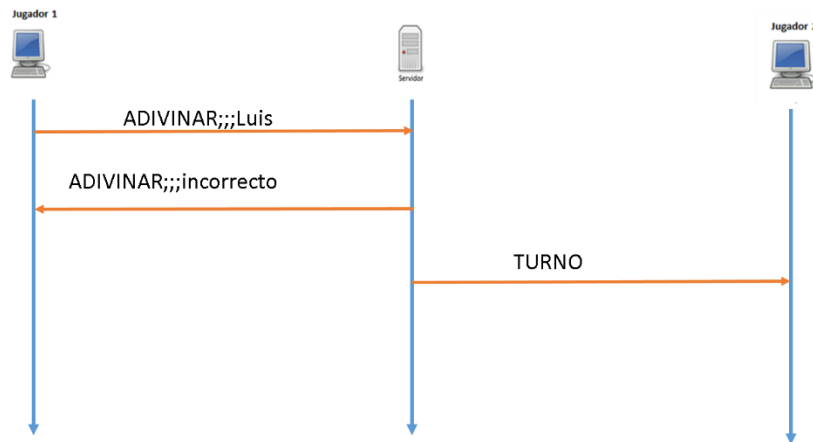


Figura 17. Ejemplo de protocolo para Adivinar.

En la Figura 17, se muestra el siguiente ejemplo: El Jugador 1 envía el mensaje con comando ADIVINAR preguntando si la persona de su oponente se llama “Luis”, el servidor le informa que no adivinó con el mensaje ADIVINAR con comando “incorrecto”. Por último, el servidor le indica al oponente que es su turno a través del mensaje con comando TURNO.

Caso 2: El jugador logra adivinar el nombre de la persona del rival.

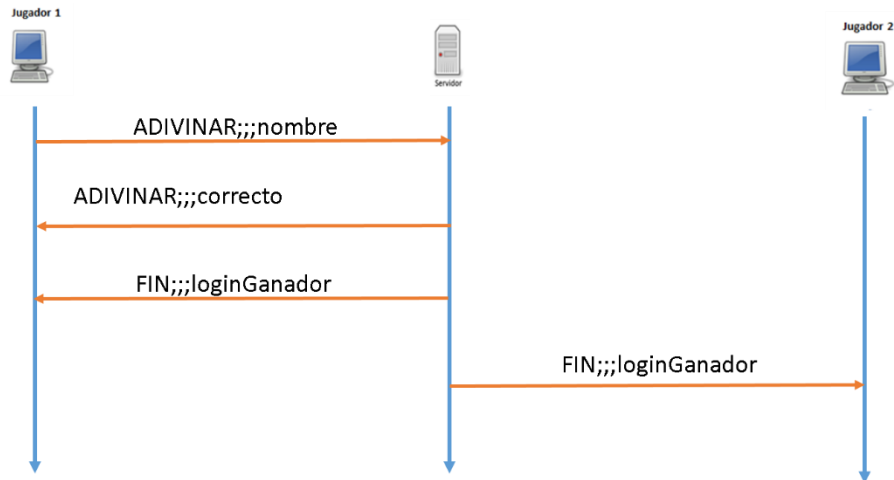


Figura 18. Protocolo para adivinar.

En la Figura 18 se aprecia el protocolo para adivinar. Ocurre cuando el cliente del jugador en turno, envía un mensaje con comando ADIVINAR seguido del nombre de la persona y el servidor le informa que no adivinó con el comando ADIVINAR seguido del mensaje correcto. Como logra adivinar, el servidor notifica ambos jugadores el fin de la partida con el mensaje FIN seguido del login del usuario que ganó.

La sintaxis de cada mensaje es explicada en la **Tabla 15**:

Comando	Parámetros	Descripción
ADIVINAR	nombre	Representa el nombre que intenta adivinar de la persona asignada al oponente.
ADIVINAR	correcto	Representa el mensaje que indica que el jugador adivinó.
FIN	loginGanador	Representa el login del jugador ganador.

Tabla 15. Sintaxis del mensaje del cliente y servidor para el protocolo para realizar una jugada.

A continuación, un ejemplo concreto:

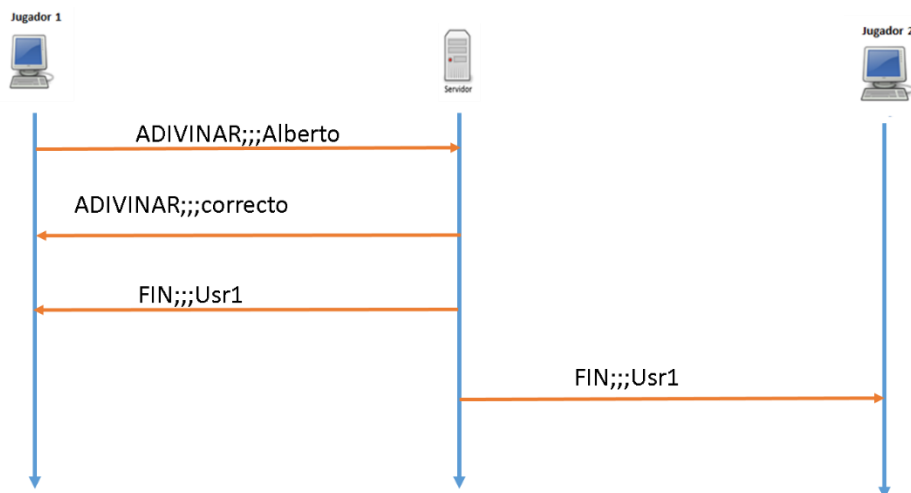


Figura 19. Ejemplo de protocolo para Adivinar.

En la Figura 19, se muestra el siguiente ejemplo: El Jugador 1 envía el mensaje con comando ADIVINAR preguntando si la persona de su oponente se llama “Alberto”, el servidor le informa que adivinó con el mensaje ADIVINAR con comando “incorrecto”. Dado que el jugador logró adivinar el nombre de la persona del oponente, el servidor notifica a los 2 jugadores el fin del juego con el comando FIN y el login del ganador “Usr1”.