

| | |
|--------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|  Proyecto CUPi2 | ISIS-1205 Algorítmica y Programación Descripción |
| Ejercicio: | n12_brainsDuel |
| Autor: | Equipo CUPi2 |
| Semestre: | 2017-2 |

Enunciado

Se quiere construir una aplicación que permita jugar Brains Duel desde diferentes computadores. Este es un juego de trivia para jugar en parejas.

El servidor del juego almacena todas las preguntas del juego con sus respectivas respuestas. De cada pregunta se conoce:

- **Categoría:** Categoría a la que pertenece la pregunta
- **Pregunta:** Texto de la pregunta
- **Opciones de respuesta:** Hay 4 opciones de respuesta posibles
- **Respuesta:** Respuesta de la pregunta.

Existen 6 categorías de pregunta en el juego: Arte, Entretenimiento, Deportes, Cultura, Geografía e Historia. Cada jugador tiene una lista de 6 trofeos que puede ganar, estos trofeos corresponden a las categorías de las preguntas. El objetivo del juego será obtener 3 de los 6 trofeos disponibles antes que el oponente.

El jugador con el primer turno es escogido aleatoriamente.

Después de asignar el turno comienza el juego. El jugador en turno podrá **girar la ruleta** o **enviar un reto** a su oponente. En caso de que elija **girar la ruleta**, el servidor enviará una pregunta aleatoria de la categoría que cayó en la ruleta al jugador que se encuentra en turno. El jugador tendrá que responder la pregunta correctamente; si así lo hace podrá seguir girando la ruleta hasta el fin del juego. Si el jugador responde de forma incorrecta, el turno pasará a ser del oponente.

Hay dos formas de ganar un trofeo: La primera consiste en contestar 3 preguntas seguidas correctamente. La categoría de la tercera pregunta define el trofeo que se ganará. Si ya tiene el trofeo, entonces no podrá ganarlo de nuevo. Si se contesta correctamente la tercera pregunta se ganará el trofeo, de lo contrario se reiniciará el contador de preguntas acertadas.

La segunda forma de ganar un trofeo consiste en caer en la corona de la ruleta. En este caso, aparecerá una pregunta de categoría aleatoria. Si se contesta correctamente esta pregunta, el jugador podrá ganar un trofeo aleatorio independiente de la pregunta.

Si el jugador decide **enviar un reto** a su oponente, el servidor elegirá una pregunta aleatoria y se la enviará a los dos jugadores. El servidor esperará la respuesta de ambos jugadores y les informará quién ganó el reto. En caso de que los dos hayan respondido bien la pregunta, informará que hubo empate. Cuando el reto finalice, el jugador que envió el reto perderá el turno. Si alguno de los dos jugadores acierta en el reto, su número de preguntas contestadas correctamente aumentará a tres.

Como un servicio adicional, el programa debe almacenar información sobre la cantidad de duelos perdidos y ganados 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 un duelo; y un programa cliente, a través del cual un usuario puede jugar BrainsDuel.

El programa cliente debe ofrecer las siguientes funcionalidades:

1. 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: Nombres, apellidos, alias (no pueden haber dos jugadores con el mismo alias), 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 alias, su contraseña, la dirección IP del servidor y el puerto.
2. Girar la ruleta. Debe permitir girar la ruleta cuando se tiene el turno asignado y calcular una categoría de forma aleatoria.
3. Enviar y recibir retos. Enviar un reto a su adversario y recibir retos del mismo.
4. Responder una pregunta. Responder una pregunta enviada por el servidor.
5. Visualizar información del duelo. El jugador podrá visualizar la siguiente información:
 - a. **Información del oponente:** Alias, cantidad de victorias y cantidad de derrotas.
 - b. **Trofeos ganados:** Trofeos ganados tanto por el jugador como por el oponente.
 - c. **Estado del duelo:** Este puede ser: en espera de oponente, en espera de turno, en turno o juego terminado.

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

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

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

1. Determinar aleatoriamente los turnos e informarlo a los jugadores.
2. Validar la respuesta de las preguntas enviadas por los clientes.
3. Elegir una pregunta aleatoria
4. Determinar, a partir de los trofeos enviados por los clientes, el fin del juego.

Una vez finalizado el duelo (cuando un jugador obtenga los 3 trofeos), el servidor debe finalizar el juego. 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 alias del jugador victorioso.

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

1. Mostrar los duelos que se encuentren en curso.
2. Mostrar las estadísticas históricas del juego. Se debe mostrar el alias de cada jugador que haya participado en un duelo, la cantidad de victorias y de derrotas, y su porcentaje de efectividad. Este último se define de la siguiente manera:

$$\% \text{ Efectividad} = \text{Duelos ganados} * 100 / \text{Total de batallas}$$

La información histórica de los duelos 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 si desea iniciar sesión o registrarse (Figura 1).



Figura 1. Diálogo de inicio

Si el usuario desea registrarse, se desplegará el diálogo que se muestra en la Figura 2.

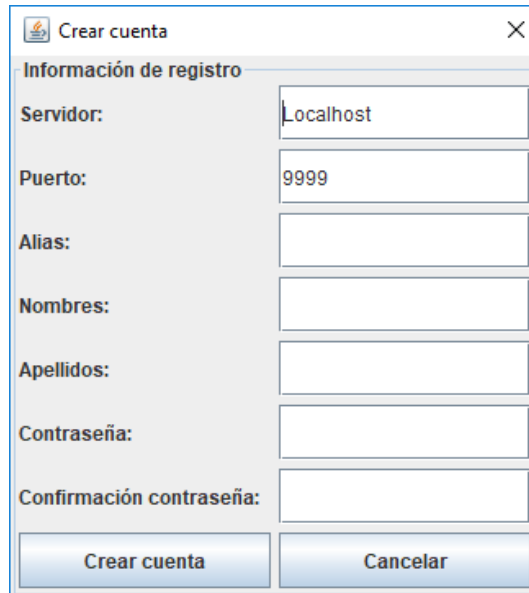


Figura 2. Diálogo de registro.

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



Figura 3. Diálogo de inicio de sesión.

Finalmente, en la Figura 4 se muestra la interfaz principal al momento de girar la ruleta.

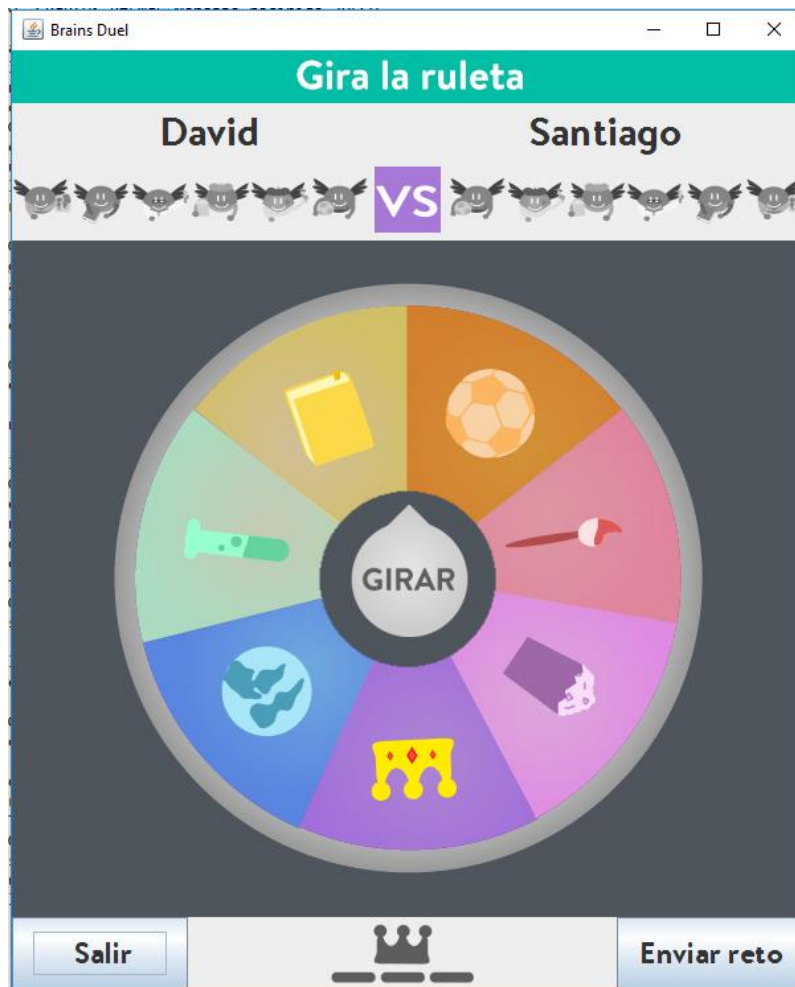


Figura 4. Interfaz principal en turno.

Interfaz Servidor

La interfaz del servidor (Figura 5) permite refrescar la lista de duelos y la lista de jugadores con el mismo botón.

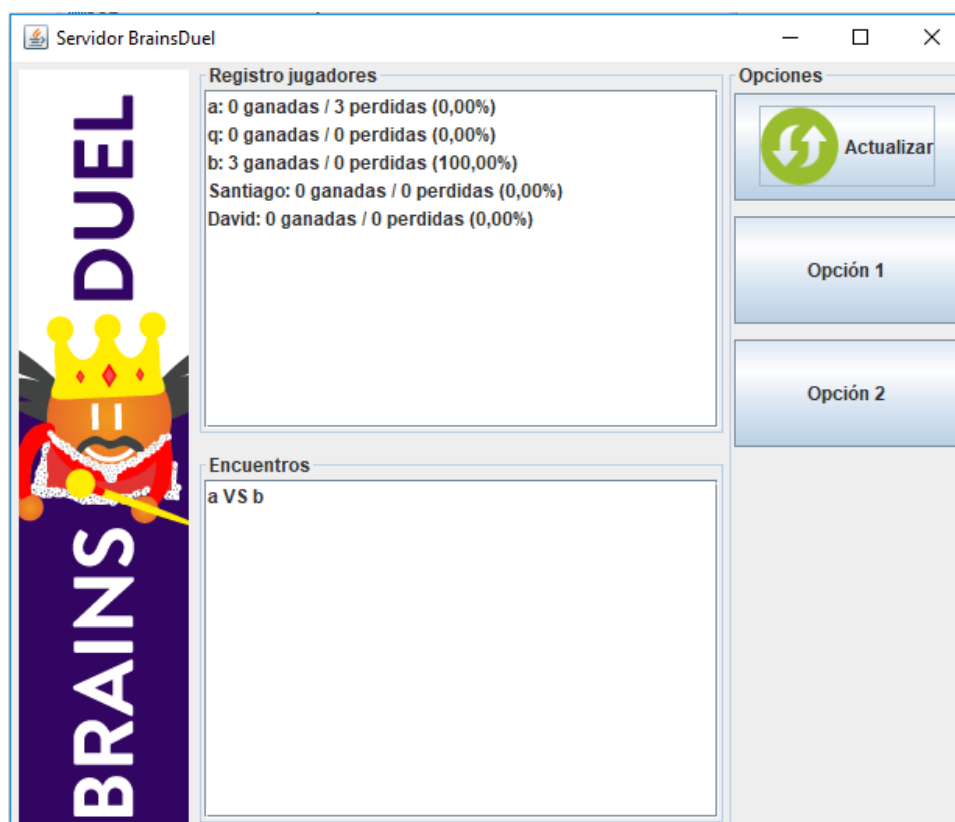


Figura 5. Interfaz del servidor.

Persistencia

Para el desarrollo de la aplicación BrainsDuel, 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 única tabla para almacenar la información de los jugadores desde la primera vez que el servidor es ejecutado. La 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 1 se muestra el diseño de la tabla de la base de datos con sus campos y llave primaria.

| Tabla | Campos | Tipo | Llave primaria |
|-----------|-------------|-------------|----------------|
| jugadores | alias | varchar(32) | alias |
| | nombre | varchar(32) | |
| | apellidos | varchar(50) | |
| | contrasenia | varchar(12) | |
| | victorias | Int | |
| | derrotas | 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:

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

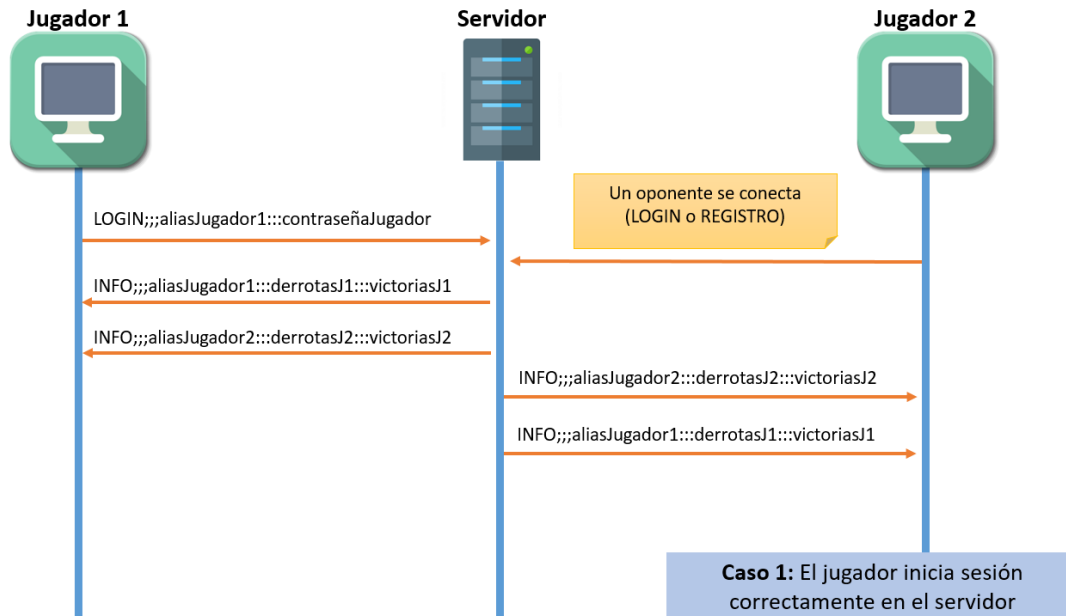


Figura 6. Protocolo de inicio de sesión.

En la Figura 6 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 alias y la contraseña del jugador que desea iniciar sesión. El servidor verifica la existencia del alias y correspondencia de la contraseña en la base de datos. Posteriormente, una vez se inicie el encuentro con otro jugador, el servidor envía un mensaje con el alias y las estadísticas (victorias y derrotas) del jugador y un mensaje con la información del oponente, con el comando INFO.

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

| Comando cliente | Parámetros | Descripción |
|-----------------|-------------------|------------------------------------------------------------------------|
| LOGIN | aliasJugador | Representa el alias del jugador que se va a conectar al servidor. |
| | contraseñaJugador | 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 |
|------------------|--------------|------------------------------------------------|
| INFO | aliasJugador | Representa el alias del jugador. |
| | derrotas | Representa el número de derrotas del jugador. |
| | victorias | Representa el número de victorias del jugador. |

Tabla 3. Sintaxis del mensaje del servidor cuando existe un oponente.

A continuación, un ejemplo concreto:

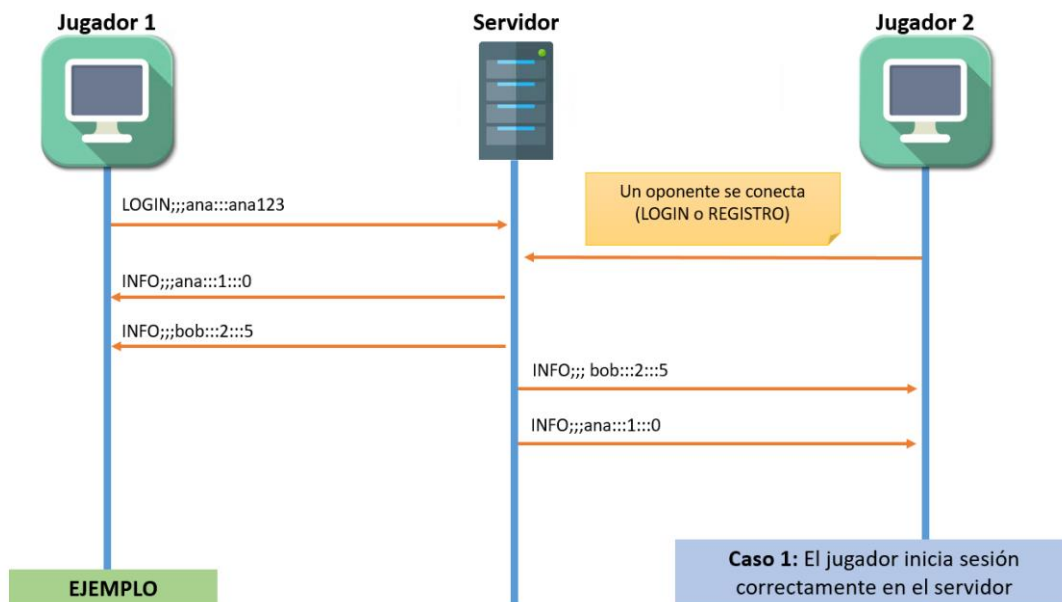


Figura 7. Ejemplo de protocolo de inicio de sesión.

En la Figura 7 se ilustra el escenario en el que el jugador con el alias “ana” desea iniciar sesión, por lo tanto, el programa envía el mensaje con comando LOGIN, el alias “ana” y la contraseña “ana123”. El servidor encuentra en su base de datos un jugador con el alias y la contraseña dados. Una vez se inicie el encuentro con otro jugador, envía un mensaje con comando INFO indicando la información del jugador y otro mensaje con comando INFO indicando la información del oponente: alias “bob”, derrotas “2” y victorias “5”.

Caso 2: Un jugador con un alias que no está registrado trata de iniciar sesión:

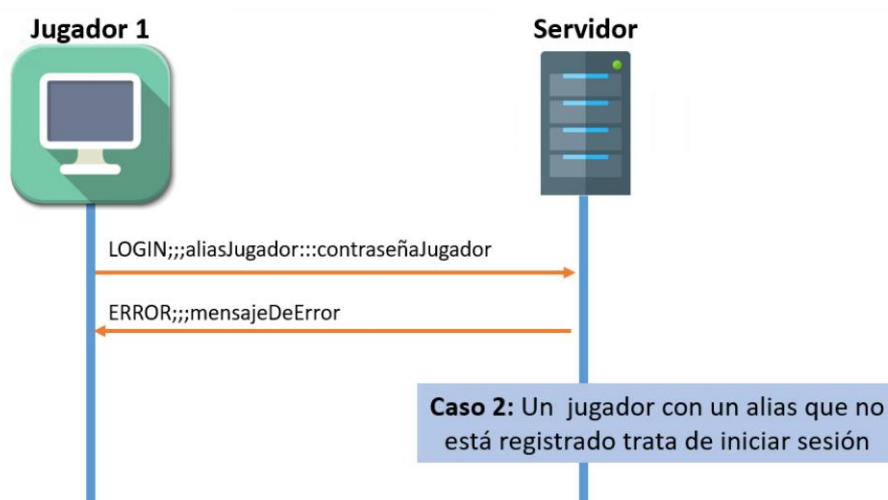


Figura 8. Protocolo de inicio de sesión cuando ocurre un error.

En la Figura 8. e 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 alias y la contraseña del jugador que desea iniciar sesión. El servidor no encuentra en su base de datos un jugador con el alias dado, 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 | aliasJugador | Representa el alias del jugador que se va a conectar al servidor. |
| | contraseñaJugador | 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 | mensajeDeError | Representa el mensaje de error que se desea informar al usuario. |

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

A continuación, un ejemplo concreto:

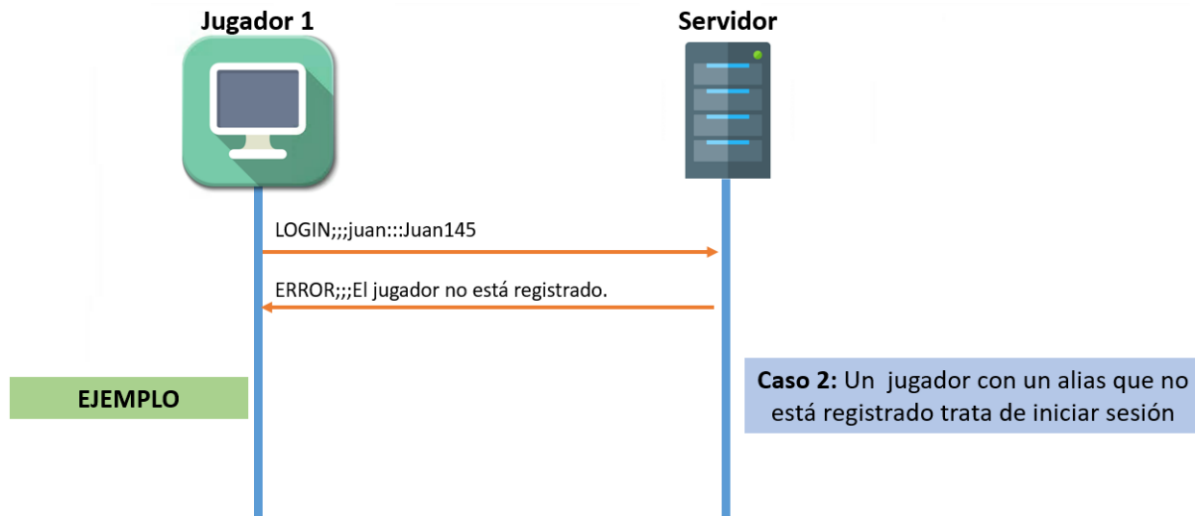


Figura 9. Ejemplo del protocolo de inicio de sesión cuando ocurre un error.

La Figura 9 muestra el siguiente ejemplo: el jugador envía una petición de inicio de sesión con el comando LOGIN y el alias “juan”, contraseña “Juan145”. El servidor no encuentra en su base de datos un jugador con el alias dado, por lo tanto envía el mensaje de error con comando: ERROR y el mensaje “El jugador no está registrado”.

Caso 3: Un jugador se conecta al servidor para iniciar sesión y la contraseña no coincide con el alias ingresado.



Figura 10. Protocolo de inicio de sesión cuando ocurre un error.

En la Figura 10 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 alias y la contraseña del jugador que desea iniciar sesión. El servidor encuentra en su base de datos un jugador con el alias dado, pero la contraseña no coincide. 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 6y en la Tabla 7:

| Comando cliente | Parámetros | Descripción |
|-----------------|-------------------|------------------------------------------------------------------------|
| LOGIN | aliasJugador | Representa el alias del jugador que se va a conectar al servidor. |
| | contraseñaJugador | Representa la contraseña del jugador que se va a conectar al servidor. |

Tabla 6. Sintaxis del mensaje del cliente para el protocolo de inicio de sesión.

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

Tabla 7. Sintaxis del mensaje del servidor para el protocolo de inicio de sesión en caso de error.

A continuación, un ejemplo concreto:

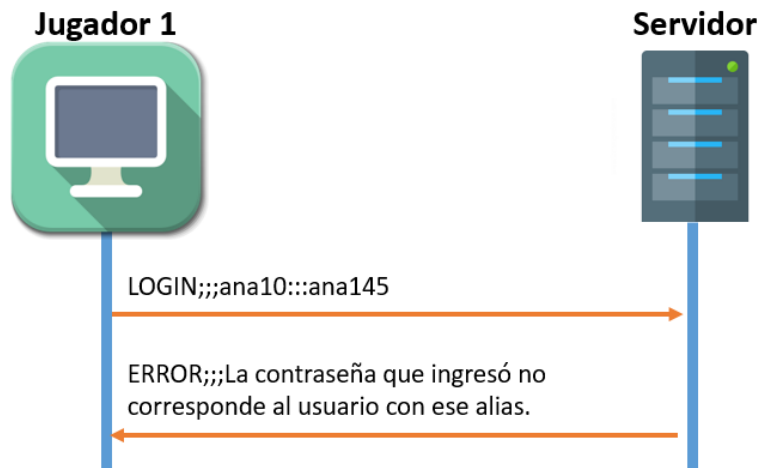


Figura 11. Ejemplo del protocolo de inicio de sesión cuando ocurre un error.

La Figura 11 muestra el siguiente ejemplo: el jugador envía una petición de inicio de sesión con el comando LOGIN y el alias “ana10”, contraseña “ana145”. El servidor encuentra un jugador con el alias dado, pero con una contraseña diferente, por lo tanto, envía el mensaje de error con comando: ERROR y el mensaje “La contraseña que ingresó no corresponde al usuario con ese alias.”

Protocolo para registrar un jugador:

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

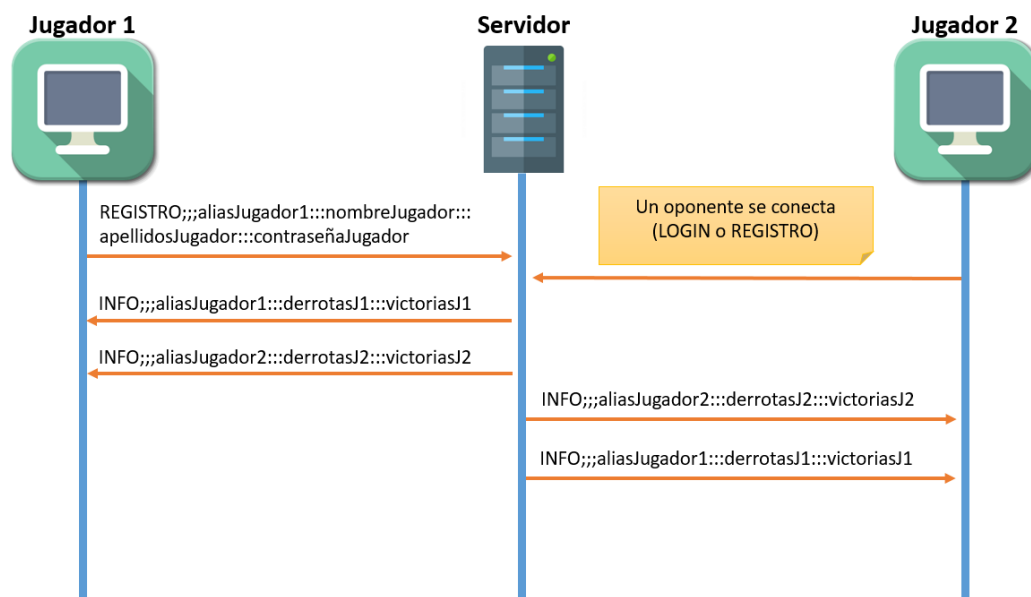


Figura 12. Protocolo de registro de un jugador.

En la Figura 12 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: alias, nombre, apellidos y la contraseña. El servidor verifica que no exista un jugador con el mismo alias. Posteriormente, una vez se inicie el encuentro con otro jugador, envía un mensaje con el alias y las estadísticas (victorias y derrotas) del jugador y otro mensaje con la información del oponente, con el comando INFO.

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

| Comando cliente | Parámetros | Descripción |
|-----------------|-------------------|-------------------------------------------------------------|
| REGISTRO | aliasJugador | Representa el alias del jugador que desea registrarse. |
| | nombreJugador | Representa el nombre del jugador que desea registrarse. |
| | apellidosJugador | Representa los apellidos del jugador que desea registrarse. |
| | contraseñaJugador | 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 |
|------------------|--------------|------------------------------------------------|
| INFO | aliasJugador | Representa el alias del jugador. |
| | derrotas | Representa el número de derrotas del jugador. |
| | victorias | Representa el número de victorias del jugador. |

Tabla 9. Sintaxis del mensaje del servidor.

A continuación, un ejemplo concreto:

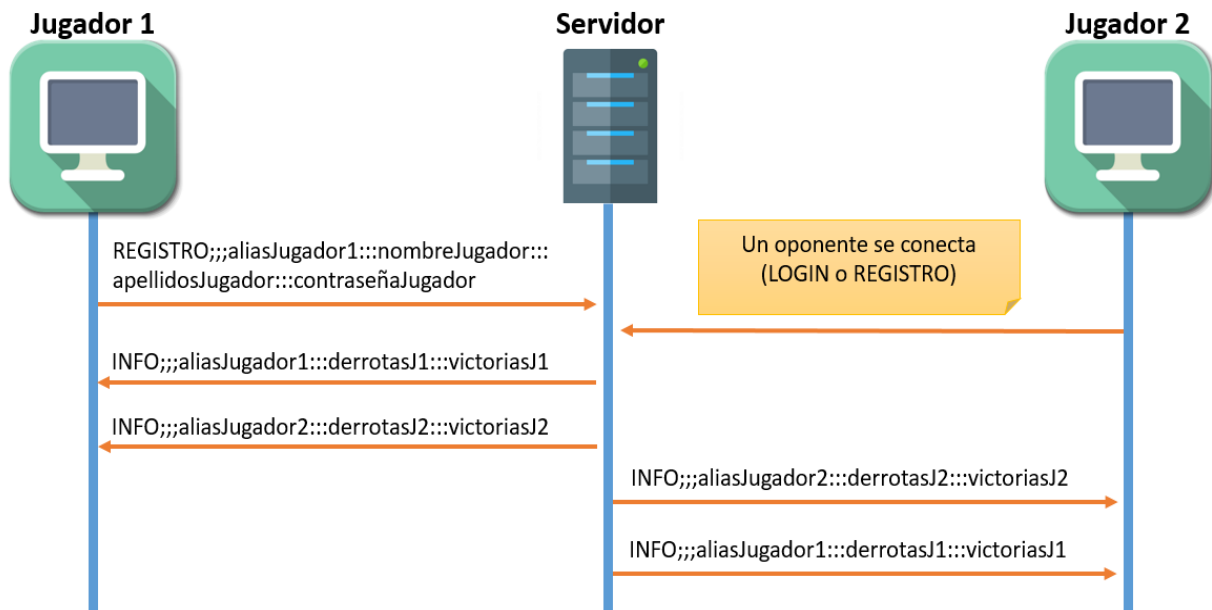


Figura 13. Ejemplo del protocolo de registro de un jugador.

En el ejemplo que muestra la Figura 13, la usuaria Ana Romero desea registrarse en el sistema. Por lo tanto, el programa cliente envía el mensaje con comando REGISTRO seguido del alias “ana”, nombre, apellidos y contraseña:”. El servidor revisa en la base de datos si existe un jugador con el alias “ana”. Si no lo encuentra, procede a agregar el usuario. Una vez se conecta un oponente, en este caso “Bob”, el servidor le envía un mensaje con comando INFO, indicando la información del jugador (alias: ana, derrotas y victorias: 0) y uno con la información de su oponente: alias “Bob”, derrotas “2” y victorias “5”.

Caso 2: Se registra un jugador que ya existe en la base de datos:

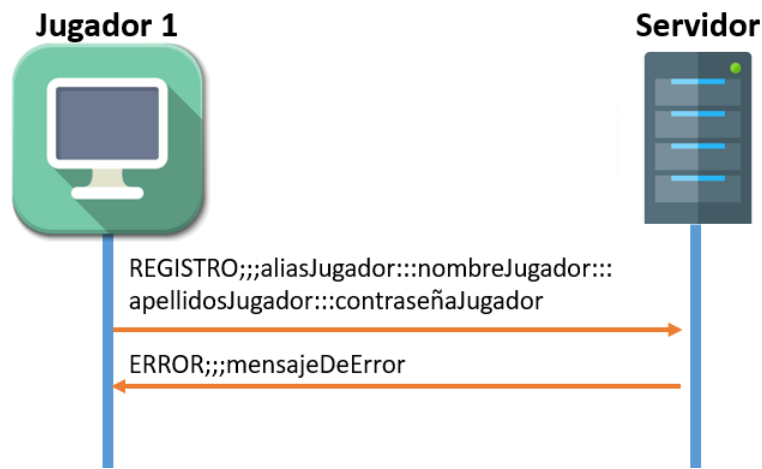


Figura 14. Protocolo de registro de un jugador cuando ocurre un error.

En la Figura 14 se aprecia 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 alias, nombre, apellidos y la contraseña del usuario que desea registrar. El servidor revisa en su base de datos por un usuario con el alias dado, en caso de encontrarlo entonces 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 10 y Tabla 11:

| Comando cliente | Parámetros | Descripción |
|-----------------|-------------------|-------------------------------------------------------------|
| REGISTRO | aliasJugador | Representa el alias del jugador que desea registrarse. |
| | nombreJugador | Representa el nombre del jugador que desea registrarse. |
| | apellidosJugador | Representa los apellidos del jugador que desea registrarse. |
| | contraseñaJugador | Representa la contraseña del jugador que desea registrarse. |

Tabla 10. Sintaxis del mensaje del cliente para el protocolo de registro de un jugador.

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

Tabla 11. Sintaxis del mensaje del servidor para el protocolo de registro de un usuario en caso de error.

A continuación, un ejemplo concreto:

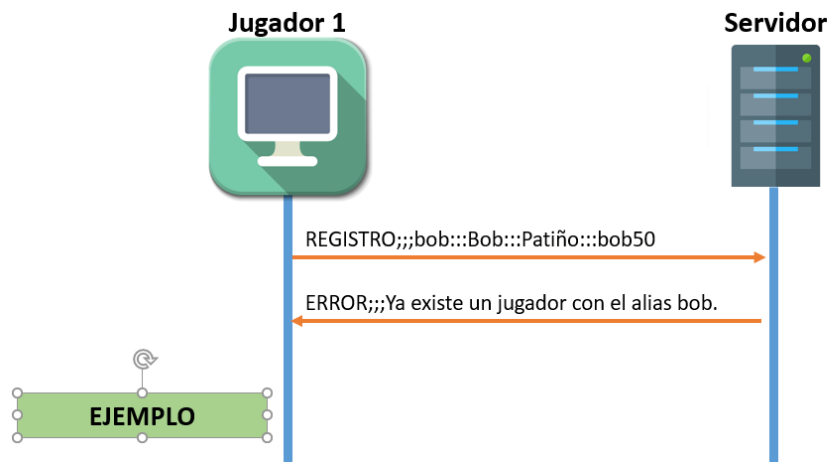


Figura 15. Ejemplo del protocolo de inicio de sesión cuando ocurre un error.

La Figura 15 muestra el siguiente ejemplo: el jugador envía una petición de registro de jugador con el comando REGISTRO seguido del alias, nombre, apellidos y contraseña del jugador: “bob”, “Bob”, “Patiño” y “bob50” respectivamente. El servidor, al revisar la base de datos, encuentra un jugador con ese alias, y por lo tanto envía al programa cliente el mensaje de error informando que existe un jugador con el alias especificado.

Protocolo para asignar turno:

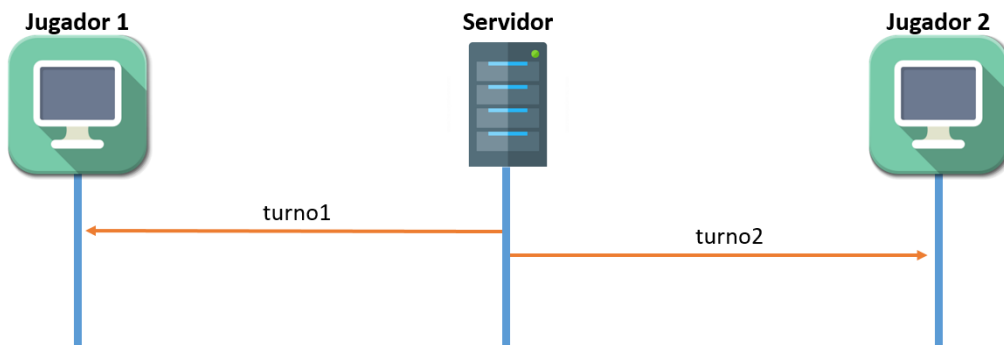


Figura 16. Protocolo para asignar turno.

En la Figura 16 se aprecia el protocolo para asignar turno. Ocurre cuando el servidor inicia una batalla con dos jugadores. El servidor elige aleatoriamente el turno que tendrán los jugadores, enviando un mensaje al cliente de cada jugador con el parámetro 1 y el parámetro 2, para informar el primer y segundo lugar, respectivamente.

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

| Comando servidor | Parámetros | Descripción |
|------------------|------------|------------------------------|
| | 1 | Representa el primer turno. |
| | 2 | Representa el segundo turno. |

Tabla12. Sintaxis del mensaje del servidor para el protocolo de asignación de turnos.

A continuación, un ejemplo concreto:

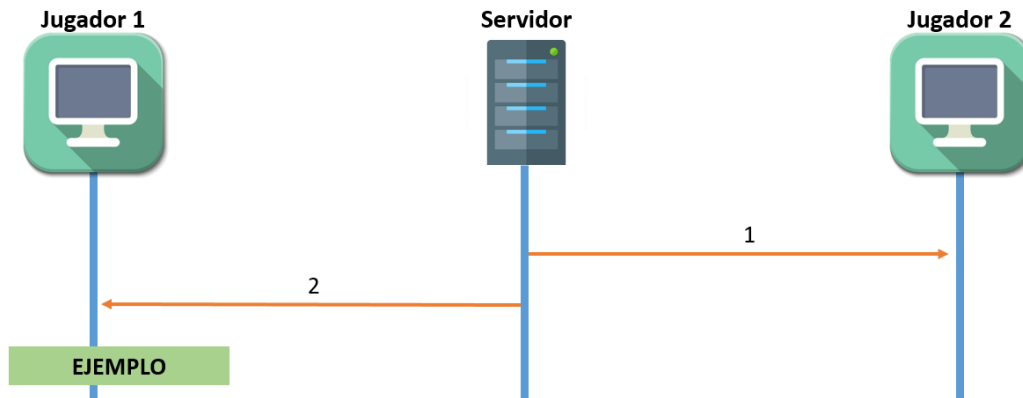


Figura 17. Ejemplo del protocolo para asignar turno.

En la Figura 17, se muestra el siguiente ejemplo: el servidor escoge aleatoriamente los turnos y envía un mensaje al Jugador 1 informando que tendrá el segundo turno y uno al Jugador 2, informando que tendrá el primer turno.

Protocolo para enviar y responder una pregunta (Girando la ruleta):

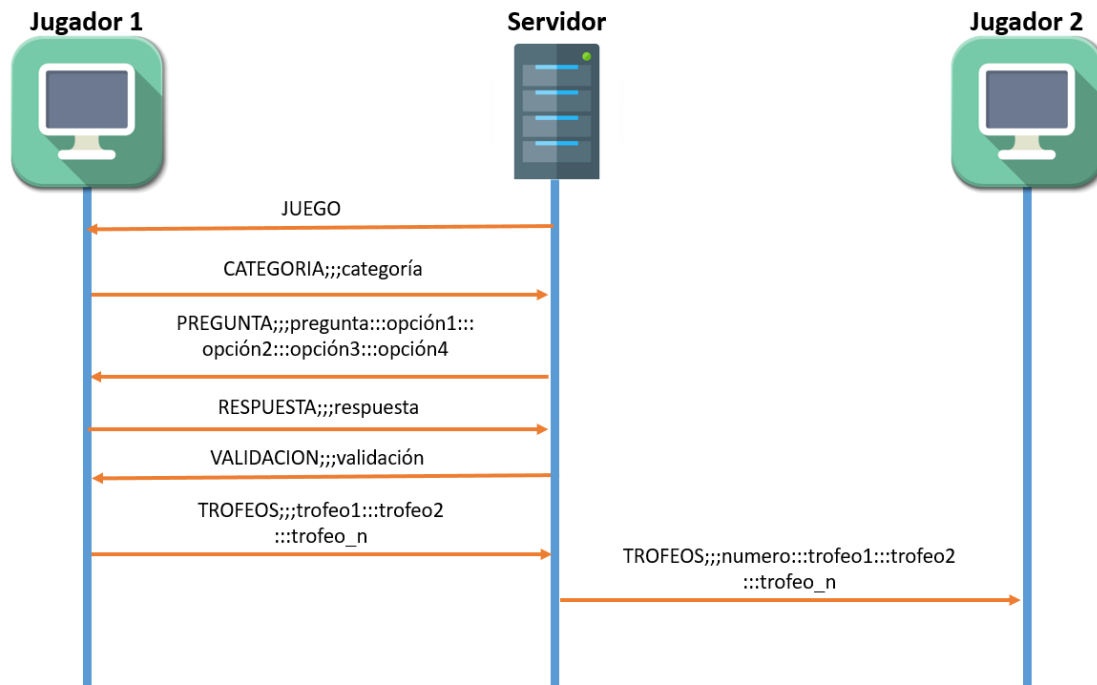


Figura 18. Protocolo para enviar la selección de un jugador.

En la Figura 18 se presenta el protocolo para enviar y responder una pregunta. En este caso el servidor envía el comando JUEGO para informar al jugador que tiene el turno. El jugador gira la ruleta y obtiene una categoría aleatoria que deberá enviar al servidor con el comando CATEGORIA. El servidor deberá buscar una pregunta aleatoria de esa categoría y deberá responder con esta al jugador. **Si se pide una pregunta con categoría “CORONA”, se deberá responder con una pregunta de cualquier categoría.** La pregunta se identifica por el comando PREGUNTA, e incluye la pregunta y las cuatro opciones de respuesta. El cliente debe contestar al servidor con el comando RESPUESTA y la respuesta a la pregunta. El servidor luego validará que la respuesta a la pregunta enviada sea correcta y enviará la validación con el comando VALIDACION. Finalmente, el jugador en turno debe enviar la lista de trofeos que tiene con el comando TROFEOS, y el servidor deberá enviarla al jugador oponente. **Si el jugador no tiene trofeos, igualmente se deberá enviar el comando; en este caso sin parámetros ni separadores.**

Si la respuesta del jugador fue correcta, se repite este protocolo y el jugador 1 mantiene su turno. De lo contrario, se envía el comando al jugador 2.

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

| Comando | Parámetros | Descripción |
|-----------|------------|----------------------------------------------------------------------------------------------------------------------|
| JUEGO | | Indica que el jugador tiene el turno y puede girar la ruleta o enviar un reto. |
| CATEGORIA | categoría | Categoría de la pregunta. Las posibles categorías son: “ARTE”, “DEPORTES”, “CIENCIA”, “HISTORIA”, “ENTRETENIMIENTO”, |

| | | |
|------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | "GEOGRAFIA" y "CORONA". |
| PREGUNTA | pregunta | Texto de la pregunta que se va a mostrar |
| | opción1 | Opción 1 de la pregunta |
| | opción2 | Opción 2 de la pregunta |
| | opción3 | Opción 3 de la pregunta |
| | opción4 | Opción 4 de la pregunta |
| RESPUESTA | respuesta | Respuesta del jugador a la pregunta |
| VALIDACION | validación | Validación de la pregunta. Indica si la respuesta del usuario fue correcta o incorrecta. Los posibles valores de validación son: "CORRECTO" e "INCORRECTO" |
| TROFEOS | trofeo_n | Corresponde a uno de los trofeos ganados por el jugador. El comando puede tener mínimo 0 parámetros y máximo 6 parámetros de trofeo. Los posibles valores que puede tomar cada trofeo son: ARTE, "DEPORTES", "CIENCIA", "HISTORIA", "ENTRETENIMIENTO", "GEOGRAFIA". No hay un orden particular para el envío de los trofeos |

Tabla 13. Sintaxis del mensaje del servidor para el protocolo para enviar y responder una pregunta.

A continuación, un ejemplo concreto:

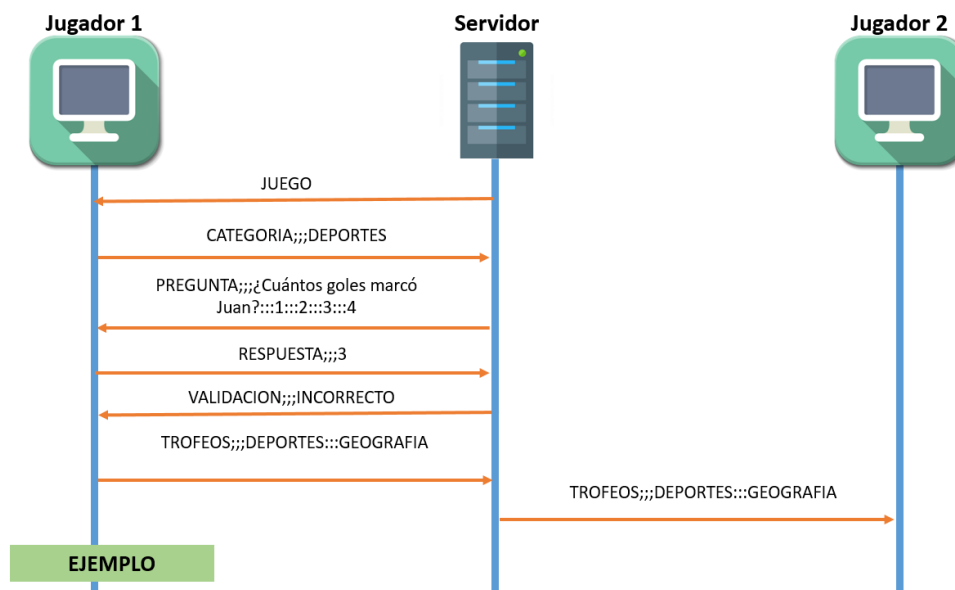


Figura 19. Ejemplo del protocolo para enviar y responder una pregunta.

En la Figura 19 se presenta el ejemplo del protocolo. El servidor envía al jugador 1 el comando JUEGO. El jugador 1 gira la ruleta y obtiene la categoría de DEPORTES, que es enviada al servidor. El servidor busca una

pregunta de deportes y la envía al cliente. El cliente contesta “3” y envía la respuesta al servidor. El servidor valida la respuesta y es incorrecta. El jugador 1 tiene los trofeos DEPORTES y GEOGRAFIA por lo que son enviados al servidor y luego al jugador 2.

En este caso, el jugador 1 volvería a recibir el comando JUEGO para repetir este protocolo.

Protocolo para enviar y recibir un reto:

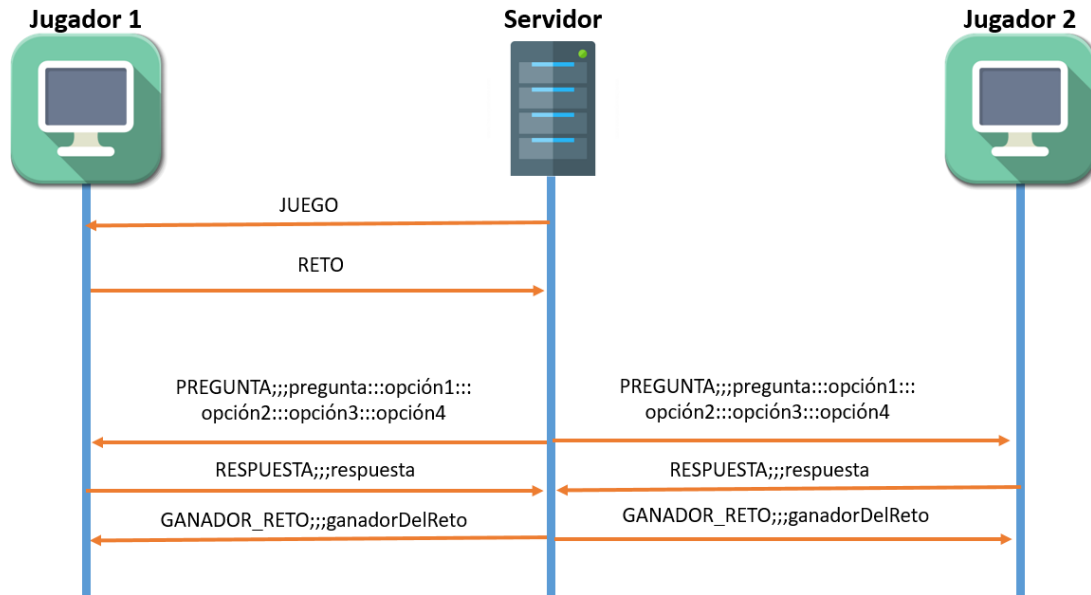


Figura 20. Protocolo para enviar y recibir un reto.

En la Figura 20 se muestra el protocolo para enviar y recibir un reto. En este caso el jugador recibe el comando JUEGO. Luego, envía el comando RETO. El servidor recibe el comando de un cliente. El servidor elige una pregunta aleatoria entre todas las preguntas y la envía a los dos jugadores con el comando PREGUNTA. El servidor se queda esperando la respuesta de ambos jugadores. Una vez ambos jugadores responden, el servidor informa quién ganó el reto con el comando GANADOR_RETO. El jugador que envió el reto pierde el turno, por lo que el comando JUEGO deberá ser enviado al oponente después de terminar este protocolo.

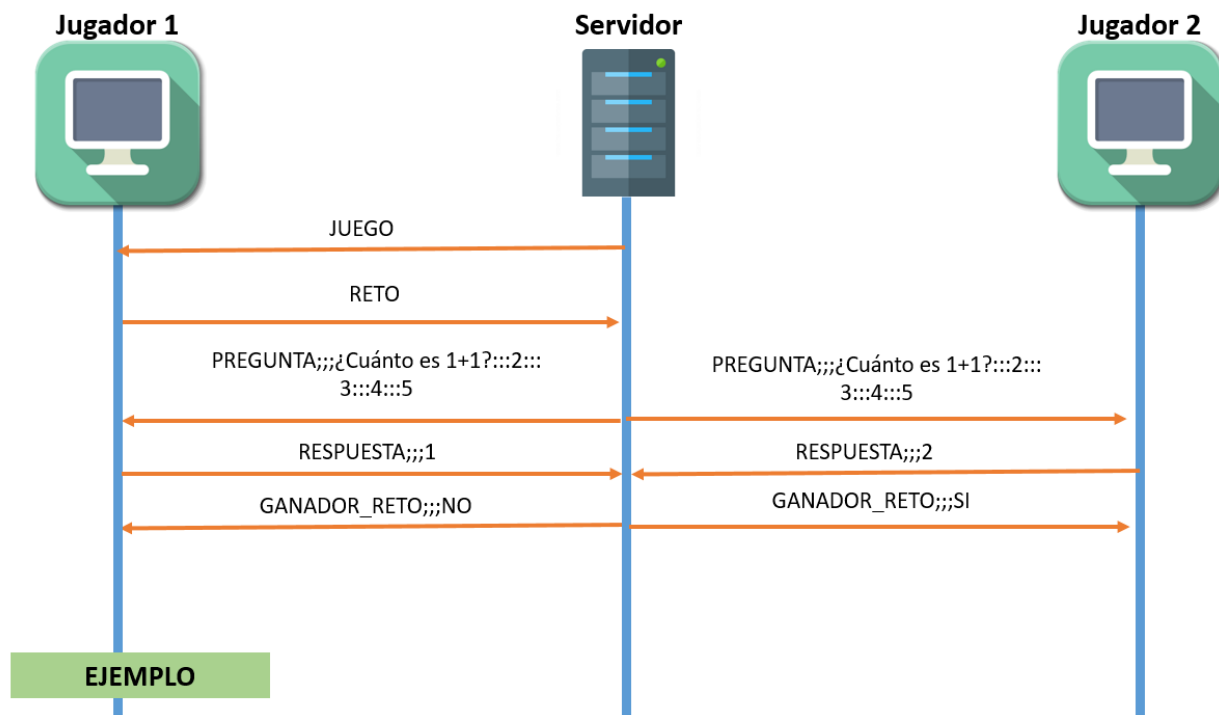
La sintaxis de cada mensaje es explicada en la Tabla 14:

| Comando | Parámetros | Descripción |
|----------|------------|--------------------------------------------------------------------------------|
| JUEGO | | Indica que el jugador tiene el turno y puede girar la ruleta o enviar un reto. |
| RETO | | Indica que se envía un reto. |
| PREGUNTA | pregunta | Texto de la pregunta que se va a mostrar |
| | opción1 | Opción 1 de la pregunta |

| | | |
|--------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| | opción2 | Opción 2 de la pregunta |
| | opción3 | Opción 3 de la pregunta |
| | opción4 | Opción 4 de la pregunta |
| RESPUESTA | respuesta | Respuesta del jugador a la pregunta |
| GANADOR_RETO | ganadorDelReto | EMPATE si ambos jugadores tuvieron la misma respuesta. De lo contrario, le envía SI al jugador que acierta y NO al jugador que pierde. |

Tabla 14. Sintaxis de los mensajes para enviar y recibir un reto.

A continuación un ejemplo concreto:



En la Figura 21 se muestra el siguiente ejemplo: El jugador 1 tiene el turno y envía un reto al servidor. El servidor envía la pregunta ¿Cuánto es 1+1? Con sus respectivas opciones. El servidor recibe la respuesta de ambos clientes y las valida. Se informa al jugador 2 que fue el ganador del reto y al jugador 1 que no lo fue.

Protocolo para finalizar el juego:

Uno de los jugadores consiguió 3 trofeos y envía el mensaje de trofeos al servidor dentro de su turno.

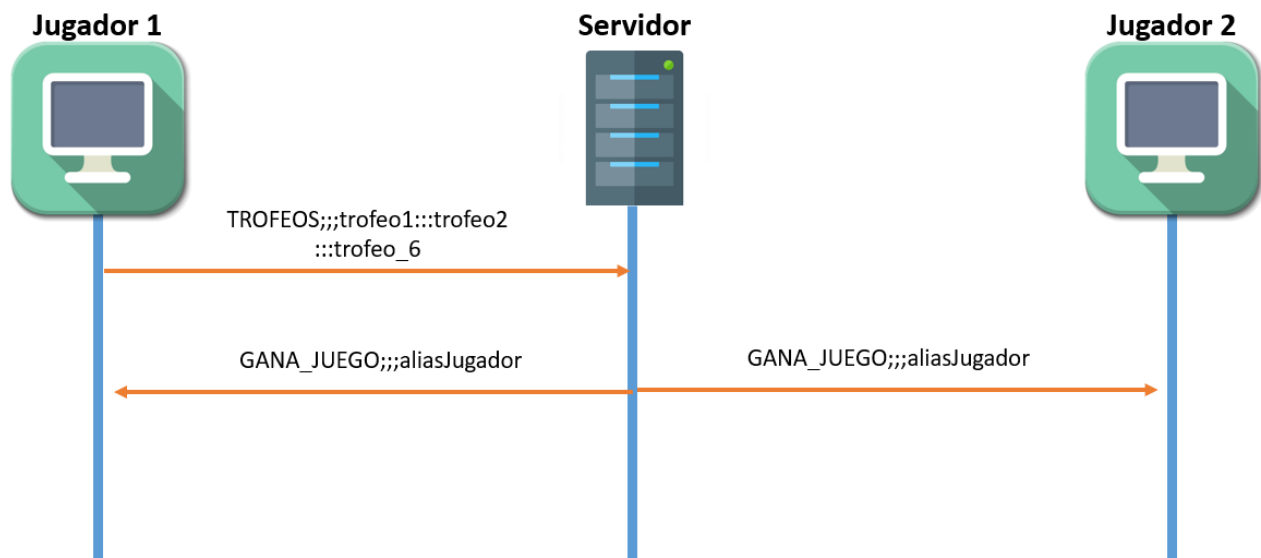


Figura 22. Protocolo para finalizar el juego.

La Figura 22 muestra el para finalizar el juego. Durante su turno, uno de los clientes envía la lista de trofeos. El servidor se da cuenta que el cliente tiene 3 trofeos, por lo que usa el comando GANA_JUEGO para informar a los dos jugadores el alias del vencedor.

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

| Comando | Parámetros | Descripción |
|---------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TROFEOS | trofeo_n | Corresponde a uno de los trofeos ganados por el jugador. El comando puede tener mínimo 0 parámetros y máximo 6 parámetros de trofeo. Los posibles valores que puede tomar cada trofeo son: ARTE, "DEPORTES", "CIENCIA", "HISTORIA", "ENTRETENIMIENTO", "GEOGRAFIA" |

| | | |
|------------|--------------|--------------------------------------------------------------------------------------------------------|
| GANA_JUEGO | aliasJugador | Mensaje que indica que el juego se terminó, y el usuario identificado con aliasJugador fue el ganador. |
|------------|--------------|--------------------------------------------------------------------------------------------------------|

Tabla15. Sintaxis del mensaje del cliente para el protocolo para realizar un ataque.

A continuación, un ejemplo concreto:

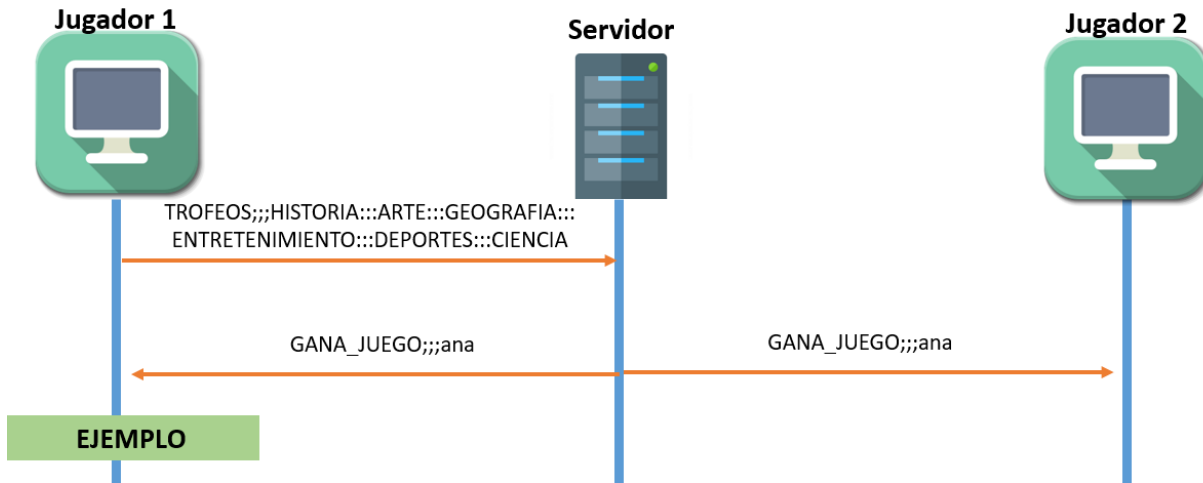


Figura 23. Ejemplo del protocolo para realizar un ataque, en caso que finalice el juego.

En la Figura 23 se presenta el siguiente ejemplo: El jugador 1 se identifica con el alias “ana”. Ella se encuentra terminando su turno, sin embargo, envía la lista de trofeos y ya los tiene completos. En este caso, el servidor se da cuenta de esto y declara a “ana” como ganadora, enviando GANA_JUEGO;ana a ambos contendientes.

Carga del archivo de preguntas

El archivo de preguntas se encuentra en data/preguntas.txt. Cada línea del archivo contiene una pregunta diferente en el formato mostrado a continuación:

Categoría;Pregunta;Opción1; Opción2; Opción3; Opción4;Respuesta

Las partes de la pregunta se encuentran divididas por punto y coma (;).

Categoría: Categoría de la pregunta

Pregunta: Texto de la pregunta

Opción1: Opción 1 de la pregunta

Opción2: Opción 2 de la pregunta

Opción3: Opción 3 de la pregunta

Opción4: Opción 4 de la pregunta

Respuesta: Respuesta de la pregunta. Corresponde a alguna de las opciones anteriores.

Ejemplos: Las siguientes son preguntas del archivo

DEPORTES; ¿Cuál es el equipo con más campeonatos ganados de la NBA?; Lakers; Bulls; Celtics; Magic; Celtics

ENTRETENIMIENTO; ¿Cuántos hermanos tiene Ron Weasley?; 4; 7; 6; 3; 6

CIENCIA; ¿Quién es el encargado por el color verde de las plantas?; Mitocondrias; Celulosa; Clorofila; Citoplasma; Clorofila

HISTORIA; ¿Hacia dónde se dirigía Cristóbal Colón cuando descubrió América?; India; África; América; Oriente medio; India

GEOGRAFIA; ¿Cuál es la capital de Venezuela?; Bogotá; Maracaibo; Caracas; Lima; Caracas