

Como sera la mecanica de autentificación:

La autentificación entre un cliente y un servidor usando TCP es un proceso crucial para asegurar la privacidad y la seguridad de la información que se intercambia entre ellos.

Para la autentificación entre servidor y cliente en un juego de tic tac toe que se actualiza constantemente, se puede utilizar un proceso de autentificación basado en tokens o identificadores únicos para cada sesión de usuario.

Debe incluir los siguientes pasos:

1. *El cliente envía una solicitud de conexión al servidor, que incluye una identificación única generada por el cliente.*
2. *El servidor recibe la solicitud y verifica la identificación única del cliente. Si la identificación es válida, el servidor crea un token de autentificación único para la sesión del usuario y lo envía de vuelta al cliente.*
3. *El cliente recibe el token de autentificación y lo almacena localmente.*
4. *Durante la sesión, el cliente utiliza el token de autentificación para enviar solicitudes al servidor para jugar tic tac toe y para recibir actualizaciones del estado del juego.*
5. *El servidor verifica continuamente la validez del token de autentificación en cada solicitud del cliente y solo responde si el token es válido.*
6. *Si la sesión del usuario termina, el token de autentificación se invalida y se elimina tanto del servidor como del cliente.*

En el caso de un juego de tic tac toe, el token de autentificación se utiliza para verificar que el cliente que está solicitando actualizaciones del juego es un usuario autorizado y legítimo.

```
//BETA PROTOCOL ACCESS TICTACTOE
```

```
Initial
```

```
--> is TCP
```

```
A : mUser, mPass
```

```
B : cPassword, cUser
```

```
Transitions:
```

```
A --> B : mUser, mPass
```

```
B -> : r = Confirmation(mUser, mPass)
```

```
//Confirmation() verify that the mUser and mPass are equal to cPassword, cUser
```

```
B --> A : r
```

```
//s = store() Funcion de guardar para que lo guarde o almacene.
```

Este es un pequeño ejemplo de cómo se vería la autentificación. Agregando una variable extra que deben conocer ambos agentes que es "token".

En la parte de la interpretación de los mensajes del cliente al servidor y viceversa sería:

1. El cliente envía sus credenciales de inicio de sesión al servidor:
CLIENTE -> SERVIDOR: enviarCredenciales(usuario, contraseña)
2. El servidor verifica las credenciales del cliente y crea un token de autenticación único para el usuario:
SERVIDOR -> CLIENTE: enviarToken(token)
3. El cliente envía una solicitud al servidor para iniciar el juego:
CLIENTE -> SERVIDOR: iniciarJuego(token)
4. El servidor responde con el estado actual del juego:
SERVIDOR -> CLIENTE: enviarActualizacion(tableroActualizado)
5. Finalmente, cuando el cliente decide salir del juego, envía una solicitud al servidor para cerrar la sesión:
CLIENTE -> SERVIDOR: cerrarSesion(token)