

COMUNICACIÓN CLIENTE => SERVIDOR

Mensaje al servidor	Respuesta del servidor	Descripción
<pre>{ request: CREATE_ROOM, body: { "username": string, } }</pre>	<pre>{ response: ROOM_CREATED, code: 200 body: { room_id: "string" owner: string } }</pre>	<p>Método para crear una sala de juegos</p>
<pre>{ request: JOIN_ROOM, body:{ "room_id":string "username": string } }</pre>	<pre>{ response: JOINED_ROOM, code: 200, body: { "owner": string, "room_id": string, "players": [object] } }</pre>	<p>Método para unirse a una sala de juegos.</p> <p>Players es un objeto donde cada llave es el username de cada jugador. Solamente el owner de la sala va a tener como llave "owner"</p>
<pre>{ request: START_GAME, body: { "room_id": string } }</pre>	<pre>{ response: GAME_STARTED, code: 200, body: { next_card:string next_player: string (username) players: [{ user: string pos: number, cards: [string] }], room_id: string } }</pre>	<p>Método para iniciar el juego.</p> <p>El round es un arreglo con el orden de turnos. Este arreglo tiene el username de cada jugador en la sala.</p> <p>El game es un objeto que nos va a traer como llave el nombre del jugador y como valor un arreglo con las cartas que le tocaron a cada uno.</p> <p>EJEMPLO:</p> <pre>body: { players: { user:'raul' pos:0 cards: ['CLUB_A, SPADE_4] } }</pre>
<pre>{ request: FINISH_GAME body: { winner:string (username) room_id: string } }</pre>	<pre>{ response: GAME_FINISHED, code: 200 body: { winner:string (username), room_id: string } }</pre>	<p>Método para saber que un juego fue terminado.</p> <p>Se envía el username del ganador y el id de la sala en la que gano.</p>

<pre>{ request: NEXT_TURN body: { card: string room_id: string username: string } }</pre>	<pre>{ response: NEXT_TURN, code: 200 body: { next_card: string next_player: string (username) last_player: string (username) players: [{ user: string pos: number, cards: [string] }] room_id: string } }</pre>	<p>Método para pasar al turno siguiente. Una vez el jugador coloca su carta en la mesa. Se elimina la carta que seleccionó y se devuelve el nuevo objeto de players</p>
<pre>{ request: FAROL body: { telltale: string (username) accused: string (username) room_id : string } }</pre>	<pre>{ response: FAROL, code: 200, body:{ answer: boolean telltale: string (username) accused: string (username) players: [{ user: string pos: number, cards: [string] }] room_id: string } }</pre>	<p>Método para decir que un jugador mintió. Se envía quien es el que está acusando y a quien se está acusando.</p> <p>El servidor responde verdadero o falso si el acusador tenía razón o no. El servidor tiene control de la carta que toca y la última carta puesta.</p> <p>El atributo de cards es un arreglo con las cartas que le toca agarrar al jugador.</p>
<pre>{ request: SEND_MESSAGE body:{ username: string room_id: string message: string } }</pre>	<pre>{ response: MESSAGE_RECIEVED code:200, body:{ room_id:string chat: [{ username: string, message: string }] } }</pre>	<p>Método para enviar un mensaje en una sala. Se envía el mensaje, la sala en la que se está y quien lo envió.</p>

COMUNICACIÓN SERVIDOR => CLIENTE

Mensaje al cliente	Descripción
<pre>{ response: GAME_STARTED, code: 200, body: { next_card:string next_player: string (username) players: [{ user: string pos: number, cards: [string] }] } }</pre>	Avisa a todos los jugadores que la partida ya inicio.
<pre>{ response: GAME_FINISHED, code: 200 body: { winner:string (username), room_id: string } }</pre>	Le avisa a los jugadores que ya hubo un ganador.
<pre>{ response: NEXT_TURN, code: 200 body: { next_card: string next_player: string (username) last_player: string (username) players: [{ user: string pos: number, cards: [string] }] } }</pre>	Le avisa a los jugadores que toca
<pre>{ response: FAROL, code: 200, body:{ answer: boolean telltale: string (username) accused: string (username) players: [{ user: string pos: number, cards: [string] }] } room_id: string }</pre>	Le avisa a los jugadores que alguien acuso de farol a un jugador.

}	
{ response: MESSAGE_RECIEVED code:200, body:{ room_id:string chat: [{ username: string, message: string }] } }	Le avisa a los jugadores que alguien envio un mensaje.

TABLA DE ERRORES

Mensaje que envía el server ante un error	Descripción
<pre>{ code: 101 response: "room_full" }</pre>	Error al tratar de unirse a la sala, sala ya llena con máximos jugadores.