



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

**Universidad Nacional de Colombia - sede Bogotá**  
**Facultad de Ingeniería**  
**Departamento de Sistemas e Industrial**  
**Ingeniería de Software 1 (2016701)**

## Historia de Usuario #2

### Anexo de Documentos Relacionados:

### Descripción conceptual

<b>Módulo</b>	Emparejamiento automático de jugadores para partidas en tiempo real
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	El sistema de emparejamiento conectará automáticamente a dos jugadores disponibles en partidas en tiempo real. Si hay un jugador en espera, se le asignará un oponente y la partida comenzará. Si no hay jugadores disponibles, el usuario permanecerá en la cola de espera hasta que llegue otro jugador.

### Backend

URL	Método	Código html
localhost:8080/api/matchmaking	POST	200, 202, 400, 500
<b>Caso de uso técnico</b> Cuando un jugador envía un POST con su ID, el servidor revisa en Firebase si hay otro jugador en espera. Si lo encuentra, crea una partida, los asigna y responde con 200 ("matched"). Si no hay nadie disponible, lo coloca en la cola y responde con 202 ("waiting"). Si la solicitud es inválida, devuelve 400 ("bad request"), y si ocurre un error en el servidor, responde con 500 ("internal server error").		
<b>Datos de entrada</b>	<b>Datos de salida</b>	
{	500: {	

<pre>"playerId": "123456" }</pre>	<pre>"status": "error", "message": "No se pudo conectar con Firebase, inténtalo de nuevo" }</pre>
<b>N/A</b>	<pre>400: {   "status": "error",   "message": "playerId es requerido" }</pre>
<b>N/A</b>	<pre>202: {   "status": "waiting",   "playerId": "123456" }</pre>
<b>N/A</b>	<pre>200: {   "status": "matched",   "matchId": "abc123",   "player1": "123456",   "player2": "654321" }</pre>

URL	Método	Código html
localhost:8080/api/matchmaking	DELETE	200 404
<b>Caso de uso técnico</b> Permite a un jugador salir de la cola de espera antes de emparejarse.		
<b>Datos de entrada</b>	<b>Datos de salida</b>	
<pre>{   "playerId": "123456" }</pre>	<pre>200{   "status": "canceled",   "playerId": "123456" }</pre>	
<b>N/A</b>	<pre>{   "status": "error",   "message": "El jugador no estaba en la cola" }</pre>	

**Interacción esperada:**

El usuario hace clic en el botón "Buscar partida" dentro del menú principal. Si ya estaba en una partida previa, debe haber finalizado antes de poder buscar otra.

**Flujo visual y eventos:**

1. El usuario presiona "Buscar partida".
2. Se muestra un mensaje "Buscando oponente..." y se desactiva el botón para evitar múltiples solicitudes.
3. Si se encuentra un oponente, se muestra una transición a la pantalla de juego con ambos jugadores.
4. Si no hay oponente disponible, el usuario permanece en espera con el mensaje "Esperando jugador..." hasta que alguien se una.
5. Si el usuario cancela la búsqueda, la interfaz vuelve al menú principal.
6. Si ocurre un error, se muestra un mensaje indicando que la solicitud falló.

**Mockups/Prototipos:**



**Notas Adicionales:**

Cancelación instantánea: Si el usuario cancela la búsqueda, la animación y el mensaje de espera desaparecen de inmediato, pero en una implementación real también se debe eliminar al jugador de la cola en Firebase.

Persistencia del estado: Si la búsqueda es cancelada o el jugador encuentra una partida, la información podría guardarse en localStorage para mejorar la experiencia de usuario en caso de recarga de la página.