Lista de métodos a api – Minority

Problematicas

Login:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/{Mail}/{Password}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/%7bMail%7d/%7bPassword%7d) **=**

Este método manda como parámetros el mail y la password del usuario a loguearse, hace un Select a la BD, y si devuelve mas de 0 registros, lo deja entrar.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/Get**](http://apiminorityproyecto.azurewebsites.net/api/sala/Get) **=**

Este método hace un select de todas las salas en la BD.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuegoMHC/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuegoMHC/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y un objeto de tipo “SalaDeJuego”(FromBody), y cambia el atributo “ModificarHComienzo” de la sala con dicho a id igual al del objeto.

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/{tabla}/{nombre}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/%7btabla%7d/%7bnombre%7d) **=**

Este método recibe como parámetro una tabla de la BD y un contenido que iria en el campo “Nombre” de las tablas ( a excepción de que sea la tabla “preguntas”, en ese caso el campo es “OpcionA”), y hace un SELECT a esa tabla con ese nombre. Si existe retorna el id del registro, sino, retorna un 0.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuego/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuego/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que si la disponibilidad de la sala esta en “true”, cambia la disponibilidad a “false” y elimina los usuarios de esa sala. Sino, cambia su disponibilidad a false. (Creo que no debería eliminar los usuarios de la sala ya que eso lo hace posteriormente. Ademas, la disponibilidad la puede manejar el server).

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/ModificarUsuario/{id}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/ModificarUsuario/%7bid%7d) **=**

Este método recibe como parámetros el id de un usuario y un objeto de tipo “Usuario” (FromBody). Este le resta una moneda al usuario en la BD e inserta el usuario con dicho id a la tabla “usuariosxsala”(Esto tal vez debería hacerlo después).

[**http://apiminorityproyecto.azurewebsites.net/api/sala/GetCantJugadoresSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/GetCantJugadoresSala/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y devuelve la cantidad de jugadores de dicha sala.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/AnadirJugadorSalaDeJuego/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/AnadirJugadorSalaDeJuego/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es añadir un jugador a la sala sumándole 1 a la CantJugadores y al MontoAGanar.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala. Lo que hace es traer los datos de dicha sala mediante un SELECT.

[**http://apiminorityproyecto.azurewebsites.net/api/pregunta/InsertarPreguntas**](http://apiminorityproyecto.azurewebsites.net/api/pregunta/InsertarPreguntas) **=**

Este método recibe como parámetro un objeto de tipo “Preguntasxjuego” (FromBody), y lo que hace es insertar 4 preguntas a la tabla “preguntasxjuego” en la BD.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/%7bid%7d) **=**

Este método recibe como parámetros el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que si el NRonda del objeto es igual a -1, actualiza el CantJugadores y el MontoAGanar de la sala poniéndolos igual al del objeto. De no ser así, hace lo mismo con el NRonda.

[**http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/{IdSala}/{NRonda}**](http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/%7bIdSala%7d/%7bNRonda%7d) **=**

Este método recibe como parámetros el id de una sala y el número de ronda de la misma. Lo que hace es hacer un SELECT de la tabla “preguntas”, donde el Id= SELECT de la tablas “preguntasxjuego” donde el IdSala y el IdRonda sean iguales a los parámetros.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ActualizarSalaDeJuegoMenosDe3Jugadores/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ActualizarSalaDeJuegoMenosDe3Jugadores/%7bid%7d) **=**

Este método recibe como parámetros el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que en caso de cuando haya terminado el tiempo de reclutamiento de jugadores y haya menos de 3 jugadores en la sala, pone la CantJugadores, el MontoAGanar y el NRonda en 0 de esa sala, y elimina los registros de la tablas “usuariosxsala” y “preguntasxjuego” correspondientes a la sala.

[**http://apiminorityproyecto.azurewebsites.net/api/respuesta/InsertarRespuesta**](http://apiminorityproyecto.azurewebsites.net/api/respuesta/InsertarRespuesta) **=**

Este método recibe como parámetro un objeto de tipo “Respuesta”. Lo que hace es insertar una respuesta a la BD.

[**http://apiminorityproyecto.azurewebsites.net/api/respuesta/GetCantVotos/{Opcion}/{IdSala}**](http://apiminorityproyecto.azurewebsites.net/api/respuesta/GetCantVotos/%7bOpcion%7d/%7bIdSala%7d) **=**

Este método recibe como parámetros una de las opciones de voto y el id de una sala. Lo que hace es hacer un SELECT de la tabla “respuestas” y determina la cantidad de votos de dicha opción en la sala.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/DeleteUsuarioxSala/{IdUsuario}**](http://apiminorityproyecto.azurewebsites.net/api/sala/DeleteUsuarioxSala/%7bIdUsuario%7d) **=**

Este método recibe como parámetro el id de un usuario. Lo que hace es eliminar los registros de la tabla “usuariosxsala” cuyo Usuario sea igual al id.

[**http://apiminorityproyecto.azurewebsites.net/api/Sala/ModificarCantJugadoresYRespuestasSala/{IdSala}**](http://apiminorityproyecto.azurewebsites.net/api/Sala/ModificarCantJugadoresYRespuestasSala/%7bIdSala%7d) **=**

Este método recibe como parámetro el id de una sala y objeto de tipo “SalaDeJuego”. Lo que hace es actualizar la CantJugadores y el NRonda de la sala según lo que ganaron la ronda anterior, y elimina la respuestas que corresponden a dicha sala.

Llamadas API:

Login:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/{Mail}/{Password}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/%7bMail%7d/%7bPassword%7d) **=**

Este método manda como parámetros el mail y la password del usuario a loguearse, hace un Select a la BD, y si devuelve mas de 0 registros, lo deja entrar.

Entrar sala:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/{tabla}/{nombre}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/%7btabla%7d/%7bnombre%7d) **=**

Este método recibe como parámetro una tabla de la BD y un contenido que iria en el campo “Nombre” de las tablas ( a excepción de que sea la tabla “preguntas”, en ese caso el campo es “OpcionA”), y hace un SELECT a esa tabla con ese nombre. Si existe retorna el id del registro, sino, retorna un 0.

INGRESARUSERSALA : jugador ID, sala ID

Devuelve:

Nro jugador

..

…

..

Inserta

Jugador en sala

Saca 1 moneda

Incrmeneta cont sala

Valida q no este repet

Valida q no hay mas 50

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/ModificarUsuario/{id}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/ModificarUsuario/%7bid%7d) **=**

Este método recibe como parámetros el id de un usuario y un objeto de tipo “Usuario” (FromBody). Este le resta una moneda al usuario en la BD e inserta el usuario con dicho id a la tabla “usuariosxsala”(Esto tal vez debería hacerlo después).

[**http://apiminorityproyecto.azurewebsites.net/api/sala/GetCantJugadoresSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/GetCantJugadoresSala/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y devuelve la cantidad de jugadores de dicha sala.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/AnadirJugadorSalaDeJuego/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/AnadirJugadorSalaDeJuego/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es añadir un jugador a la sala sumándole 1 a la CantJugadores y al MontoAGanar.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/%7bid%7d) **=**

Este método recibe como parámetros el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que si el NRonda del objeto es igual a -1, actualiza el CantJugadores y el MontoAGanar de la sala poniéndolos igual al del objeto. De no ser así, hace lo mismo con el NRonda.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ActualizarSalaDeJuegoMenosDe3Jugadores/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ActualizarSalaDeJuegoMenosDe3Jugadores/%7bid%7d) **=**

PASA A BATCH!!!

Este método recibe como parámetros el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que en caso de cuando haya terminado el tiempo de reclutamiento de jugadores y haya menos de 3 jugadores en la sala, pone la CantJugadores, el MontoAGanar y el NRonda en 0 de esa sala, y elimina los registros de la tablas “usuariosxsala” y “preguntasxjuego” correspondientes a la sala

.

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarCantJugadoresONRondaSala/%7bid%7d) **=**

Este método recibe como parámetros el id de una sala y un objeto de tipo “SalaDeJuego” (FromBody). Lo que hace es que si el NRonda del objeto es igual a -1, actualiza el CantJugadores y el MontoAGanar de la sala poniéndolos igual al del objeto. De no ser así, hace lo mismo con el NRonda.

[**http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/{IdSala}/{NRonda}**](http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/%7bIdSala%7d/%7bNRonda%7d) **=**

Este método recibe como parámetros el id de una sala y el número de ronda de la misma. Lo que hace es hacer un SELECT de la tabla “preguntas”, donde el Id= SELECT de la tablas “preguntasxjuego” donde el IdSala y el IdRonda sean iguales a los parámetros.

Votar:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/{tabla}/{nombre}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/%7btabla%7d/%7bnombre%7d) **=**

Este método recibe como parámetro una tabla de la BD y un contenido que iria en el campo “Nombre” de las tablas ( a excepción de que sea la tabla “preguntas”, en ese caso el campo es “OpcionA”), y hace un SELECT a esa tabla con ese nombre. Si existe retorna el id del registro, sino, retorna un 0. (Usuario y preguntas)

GUARDO EL ID en una variable

[**http://apiminorityproyecto.azurewebsites.net/api/respuesta/InsertarRespuesta**](http://apiminorityproyecto.azurewebsites.net/api/respuesta/InsertarRespuesta) **=**

Este método recibe como parámetro un objeto de tipo “Respuesta”. Lo que hace es insertar una respuesta a la BD.

Pollings:

[**http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala. Lo que hace es traer los datos de dicha sala mediante un SELECT.

(Lo hace en cada tick de los segundos donde reclutan jugadores)

TraerEstadosSalas() =

Si no trajo los estados, llama al api/sala/Get

Sino, analiza la disponibilidad de las salas.

En el caso de que se termine o empieze alguna, llama al :

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/{tabla}/{nombre}**](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/%7btabla%7d/%7bnombre%7d) **=**

Este método recibe como parámetro una tabla de la BD y un contenido que iria en el campo “Nombre” de las tablas ( a excepción de que sea la tabla “preguntas”, en ese caso el campo es “OpcionA”), y hace un SELECT a esa tabla con ese nombre. Si existe retorna el id del registro, sino, retorna un 0.

Luego al:

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuegoMHC/{id}**](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuegoMHC/%7bid%7d) **=**

Este método recibe como parámetro el id de una sala y un objeto de tipo “SalaDeJuego”(FromBody), y cambia el atributo “ModificarHComienzo” de la sala con dicho a id igual al del objeto.

Y finalmente al api/sala/Get.

(Lo hace cada 15 seg)