Minority

Interacciones Android – Metodos API

Activity\_Registracion=

BtnRegistracion:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/AgregarUsuario**](http://apiminorityproyecto.azurewebsites.net/api/usuario/AgregarUsuario) **:**

Se pasa un objeto de clase Usuario, se fija si existe, y si no, lo agrega a la BD.

Activity\_Login=

BtnLogin:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/"**+Mail+**"/"**+Password](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetExisteUsuario/%22+Mail+%22/%22+Password) :

Se pasa el mail y la password de un usuario. Si existe, devuelve el usuario correspondiente con sus propiedades. Si no, devuelve un usuario con las propiedades en 0.

Se guarda en una clase estatica el Id del usuario, su Username, y sus monedas.

Activity\_SeleccionarSala=

TraerEstadosSalas():

Si no trajo los estados:

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

Trae las salas con sus propiedades, entre ellas, si esta “Disponible” o “En juego”.

Si trajo los estados:

Genera un timer cada 15 seg donde se resta un seg a los TextViews que indican cuanto falta para que termine la disponibilidad de una sala o cuanto lleva en juego.

Si termina el tiempo de disponibilidad:

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/salasdejuegos/"**+**NombresSalas**[i](http://apiminorityproyecto.azurewebsites.net/api/usuario/GetIdByNombre/salasdejuegos/%22+NombresSalas%5bi)] :

Trae el Id de una sala (Podría sacarlo ya que lo tengo guardado en una clase estática)

[**http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuego/"**+Id](http://apiminorityproyecto.azurewebsites.net/api/sala/ModificarSalaDeJuego/%22+Id):

Se le pasa un objeto de clase SalasDeJuego, y se modifica la disponibilidad de la sala a la que tiene el objeto. (Tambien podría sacarse)

BtnSala:

. Busca el Id del botón apretado

[**http://apiminorityproyecto.azurewebsites.net/api/usuario/IngresarUserSala**](http://apiminorityproyecto.azurewebsites.net/api/usuario/IngresarUserSala) **:**

Se le pasa un objeto de clase UsuariosxSala, y se fija que el usuario tenga monedas para entrar a la sala, que no haya entrado anteriormente, y que hayan menos de 50 jugadores. Validando esas cosas, se devuelve un mensaje al Android . Si el mensaje es “Ingreso a sala restando moneda” , se le resta una moneda a las del usuario guardadas en la clase estatica y lo lleva a la Activity\_Jugabilidad. Si es “ Ingreso a sala” , lo lleva a dicha Activity sin restarle la moneda, y si no , muestra el mensaje permaneciendo en la misma Activity.

Activity\_Jugabilidad=

[**http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/"**+IdSala](http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/%22+IdSala) :

Trae las propiedades de una sala.

Si no es la primera vez que juega y hay menos de 3 jugadores, volverá a la Actitivty\_SeleccionarSala. Sino, llamara al método SetearTimerSegundosDisponibles().

SetearTimerSegundosDisponibles():

Si los segundos disponibles para que arranque la sala son mayores a 0, se generara un timer en el formato hh/mm/ss que durara dichos segundos, donde en cada segundo se llama a [**http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/"**+IdSala](http://apiminorityproyecto.azurewebsites.net/api/sala/GetSala/%22+IdSala)para ir mostrando los jugadores que van entrando a la sala.

Si termina ese timer y hay menos de 3 jugadores, se dirije a Activity\_SeleccionarSala, sino busca la pregunta mediante [**http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/"**+**IdSala**+**"/"**+**SalaDeJuegoTraida**.**NRonda**](http://apiminorityproyecto.azurewebsites.net/api/pregunta/GetPregunta/%22+IdSala+%22/%22+SalaDeJuegoTraida.NRonda)

Muestra la pregunta y setea un timer de 30 seg para que la conteste. Al terminar dicho timer, se fija si el usuario voto . Si lo hizo, lo lleva a Activity\_Resultados. Sino, inserta una respuesta en blanco y después lo dirije a dicha Activity.

BtnVotar:

Marca que el usuario voto e inserta su respuesta a la BD.

Activity\_Resultados=

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

Se pasa un objeto de tipo VotoACalcular, que contiene el id del usuario, el id de la sala, las opciones de la pregunta, el nro de ronda, y el voto del jugador. Teniendo ese objeto se hace un Select del campo “TerminoRonda” de esa sala .

.Si esta en false:

.Se la modifica a true y se hace un while hasta que la cantidad de jugadores en la sala sea la misma que la de respuestas, en el caso de que haya diferencias de tiempo. Una vez pasado ese while, se obtiene las opciones de la pregunta hecha junto la cantidad de votos de cada una.

.Se calcula la minoría, modificando la cantidad de jugadores de la sala a los que están dentro de la misma, y en el caso de que haya empate, se pone en false el campo “Sigue” de todos los usuarios que están en esa sala. ( Lo hace una sola vez)

. Se hace un foreach de las respuestas teniendo en cuenta

Login 🡪 Entra a sala 🡪 Suma usuario a la sala si no estaba antes 🡪 2 min de recaudación de jugadores 🡪 Trae pregunta 🡪 30 seg de timer 🡪 Calcula resultados