Proyecto Final “Minority” – Programación y procesos

Problemáticas

Activity\_Login:

Ingresa su mail y su contraseña. Verifica que ambas no sean nulas, y si no lo son, Verifica que ambas existan en la bd. Si existen, pasa a la Activity\_SeleccionarSala.

Activity\_SeleccionarSala:

Se compone de 2 textviews que indican el nombre del usuario y sus monedas, guardadas en una clase estática de la aplicación. Abajo, hay 6 salas compuestas de un textview que indica si esta “En Juego” o, en el caso de estar disponible, cuanto tiempo falta para que termine de estarlo. Al arrancar la activity, se produce el método TraerSalas(), que en el caso de que no las haya traído aun, las trae de la BD, checkea su disponibilidad, cambia el atributo “ModificarHComienzo” a “false” de la BD y vuelve a llamar al TraerSalas().

Al traer las salas, se inicia un timer de 15 segundos donde se pregunta si la sala está o no disponible. Si lo está, se pregunta si su estado que cuenta el tiempo hasta que termine su disponibilidad este en 00:00:00, si lo está, cambia su disponibilidad cuando termina el timer inhabilitando el botón que conduce a dicha sala, y se vuelve a traer a las salas de la BD. De no estar en 0, se le resta 1 segundo al contador. Por otro lado, si esta “En juego”, verifica que el “TiempoALlegar” que es cuando termina su “no disponibilidad” sea igual a la hora actual, lo que indica que ha terminado, y cambiara su no disponibilidad cuando termina el timer. Si al terminar el timer se llega a la última sala y no hubo cambios, se llama de nuevo al TraerSalas().

Una vez que aprieta el botón para ir a la sala busca el id del botón que fue apretado para así encontrar el índice del vector que contiene el nombre de la sala a dirigirse. Luego de obtener el nombre, se busca el id de la sala en la BD, y posteriormente, el del usuario. Una vez que se obtuvieron dichos ids, se valida que el usuario no haya entrado anteriormente a la sala, que yo lo había hecho con una propiedad de la clase estática que tengo en el Android que se llama GetEntroSala(). El problema con esto, es que si entro desde mi celu y desde el Genymotion, me toma el mismo usuario como dos distintos, con lo cual, podría cambiarlo a una validación a la tabla “usuariosxsala” que tengo en la BD, pero tardaría mucho eso.

Si entro anteriormente, se dirige a la Activity\_Jugabilidad. Sino, se le resta una moneda al usuario, lo inserta en la tabla “usuariosxsala”, y pregunta por la cantidad de jugadores. Si esta es menor a 50, añade un jugador a la BD y se dirige a la Activity\_Jugabilidad. Sino, no puede entrar a dicha sala. Una problemática con esto es que tal vez debería insertarlo en la tabla “usuariosxsala” cuando añade el jugador pero no se si cuando hago un PUT puedo pasar más de un parámetro que no sea el id.

Terminando con esta activity, el método IrAActivityJugabilidad() consiste de los parámetros IdSala y el tiempo en que termina la disponibilidad de la misma. En el método, se pone en un Bundle el IdSala, el nombre del Usuario, si es o no la primera vez que juega, y la diferencia de tiempo entre el TiempoALlegar y la hora actual pasado a segundos.

Activity\_Jugabilidad:

Recibe los objetos traídos en el Bundle y en base a eso pregunta si es la primera vez que juega para saber si pone los botones habilitados o inhabilitados. Luego, trae la información de la sala, y pregunta si ya había traído la sala antes. Si no la trajo, pregunta si la CantJugadores<3 y si no juega por primera vez, cosa que si se cumplen dichas condiciones, se dirige a la Activity\_SeleccionarSala. Si ya trajo la sala antes, realiza el método IniciarTimerSegundosDisponibles().

Si los SegundosDisponibles son <=0, se pregunta si es la primera vez que juega. De ser así, se genera un numero random de Jugadores, se inserta dicho número en la BD,(se insertan 4 ids de pregunta en la BD), se cambia el ModificarHComienzo de dicha sala, se pone el NRonda=1, se busca la pregunta de la BD, y una vez que se encuentra la pregunta, se muestran los botones y se pone un timer de 30 segundos para votar(Es muy poco probable que pase esto). Si no es la primera vez que juega, directamente se busca la pregunta, y si los segundos disponibles son mayor a 0, se hace un timer de dichos segundos +1, donde en cada tick se hace el GetSala() (Si la cantidad de jugadores=1 y todavía no inserto ids de pregunta a la BD, lo hace y luego llama al GetSala()), para mostrar la variable cantidad de jugadores. Al terminar ese timer, se pregunta si la cantidad de jugadores es menor a 3. Si es así, actualiza la cantidad de jugadores, el monto a ganar de la sala y su NRonda a 0, y se dirige a la Activity\_SeleccionarSala. Sino, inserta el número de jugadores en la BD, cambia el ModificarHComienzo de dicha sala, se pone el NRonda=1, se busca la pregunta de la BD, y una vez que se encuentra la pregunta, se muestran los botones y se pone un timer de 30 segundos para votar. Todo el proceso tarda bastante

Una vez que vota, se inhabilita el botón de votar, se inserta una respuesta del usuario a la base de datos, y se espera a cuando se terminen los 30 segundos para ir a la Activity\_Resultados, mediante el método IrAActivityResultados(), que contiene un Bundle con el IdSala, CantJugadores, el nombre del usuario y su id, el NRonda, el voto del usuario y las opciones de voto.

Activity\_Resultados:

La Activity, luego de obtener las referencias de los textViews, arranca con el método GenerarResultadoUsuarioParte1(). En este método se toman los datos que vinieron en el Bundle, y se obtiene la cantidad de votos de la opción 1 y de la opción 2. Luego, sigue el método GenerarResultadoUsuarioParte2(). Aquí se empiezan a realizar los resultados de la ronda en base a la cantidad de votos calculada. Si el voto del jugador en la ronda fue nulo, se toma que perdió. Si no es así, se pregunta si la cantidad de votos a las 2 opciones es igual, lo que haría que sea un empate. Si no es así, en base a cual es mayor a una variable boolean MayoriaOpcion1. Está en conjunto con el voto del usuario son parámetros de la función CheckearResultados(), que va a determinar en función de sus parámetros si el usuario “Gano” o “Perdió”. Dicho resultado es el parámetro de la función ImprimirResultadosPantalla(). Esta se encarga de mostrar cuantos jugadores votaron cada cosa, mostrando en verde la opción minoritaria, y en rojo la otra. Si perdió o empato, elimina al usuario de la tabla “usuariosxsala”, espera 3 segundos con un timer, actualiza los datos en la BD, y se dirige a la Activity\_SeleccionarSala. Si gano, hace lo mismo desde el timer de los 3 segundos, pero su destino es la Activity\_Jugabilidad. Se dirige a esta con el método IrAActivityJugabilidad, donde pondrá en el Bundle el Id de la sala, el nombre del usuario, los segundos disponibles de la sala, donde pondrá un 0, y la si es la primera vez que juega, donde pondrá un “false”.